How to make a noise

sound design and synthesizer programming

by

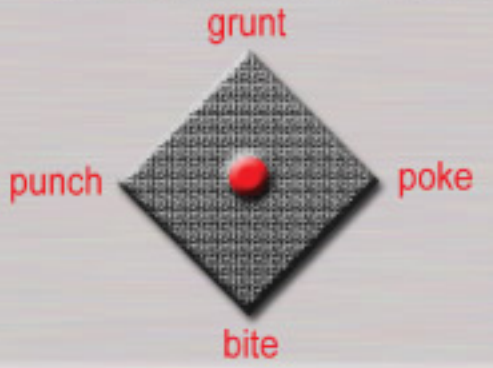
Simon Cann

Sound source

Free Version

analogue | digital | FM | additive

Vector control

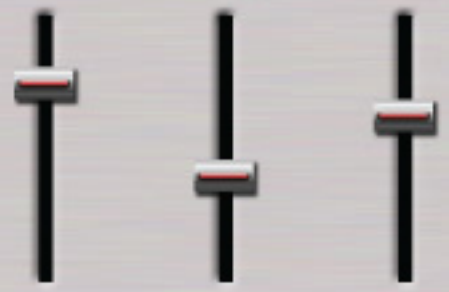


Modulation matrix

source	amount	destination
talent		volume
hope		presence

Super smooth vintage filter

Mixer



How to make a noise: sound design and synthesizer programming

How to make a noise: sound design and synthesizer programming

Welcome

Thank you for downloading this book. I hope you enjoy it.

How To Make A Noise is a comprehensive guide about how to program synthesizers. In particular it focuses on how to program and get usable sounds, and then how to use your sounds in an appropriate context.

Before we get going, a few pieces of housekeeping, please.

- This is the free edition of How To Make A Noise and was first released in early 2005. Since then the book has been completely revised and updated, and is available in hard copy format from leading book retailers such as Amazon for £9.95/$14.95. If you want to get hold of the printed (and updated!!) version of this book, the easiest way is to go to my website www.noisesculpture.com/htman and follow one of the links to Amazon.
- The updated version of the book is the same page size as my Cakewalk Synthesizers and Building a 21st Century Music Career books, and with the new content coupled with the new layout, the printed version is nearly 300 pages long. When compared with this free version, in the printed version you'll find:
  - more content, including four refills
  - more graphics (and all of the existing graphics in this edition have been completely redrawn in the printed version), and
  - more sounds (over 300 patches are featured in the updated book).

  If you want to have a look at the table of contents and a sample chapter to see what is covered by the printed edition and how it now looks (it does look *very* different from this version), then you will find a copy of these in the zip file that contained this book.
- This free version has not been updated (although refills are available which are discussed in a moment). Accordingly, some of the details about the synthesizers may no longer be up-to-date. Most immediately, you will see that some of the screenshots in this book are slightly out-of-date.
- This book relies heavily on examples. The example patches are available from www.noisesculpture.com/htmanpatches at a cost of $10. You don't need the patches to read and understand the book: in most cases, the construction of the patches is explained. However, it will save you a lot of hard work if you get them. Even if you don't own the synthesizers mentioned in this book, you can download a demo and use the patches with the demo (although at the time of writing, the patches could not be loaded into the demo of Rhino). The patches are compatible with both this version and the printed version of the book.
- This edition of the book is free of charge but not free of copyright. This book is my copyright. You may not copy it, nor may you redistribute it, without my explicit permission. However, you may print a copy for your own personal use. As noted above, the patches are available – these too are copyright and you may not redistribute these either.

The featured synthesizers

This book focuses on, and provides sonic examples from, five of the current leading software synthesizers:

- Cameleon 5000 from Camel Audio.
- Rhino from Big Tick
- Vanguard from reFX
- Wusikstation from Wusik dot com, and
- Z3TA+ from Cakewalk

In addition, in the printed version of the book and in the refills, Surge from Vember Audio is featured.

These synthesizers are used to illustrate certain points. Each synth is capable of much more than I am demonstrating in this book. There are other great synthesizers out there too – the techniques set out in this book can be applied equally to many of them. However these six have been chosen to illustrate the points raised in this book.

If you don't already own all of these synthesizers, I suggest you do. If nothing else, download a demo and try them out. I am also assuming that you will at least be familiar with the manuals for these synths: this book is not a substitute for reading these manuals and it certainly does not attempt to explain how to access any of the synthesizers' features.

The accompanying patches

This free book and the printed version both rely heavily on examples – there are now over 300 patches covered in the printed version of the book. You don't need the patches to read and understand the book: in most cases (but not all), the construction of the patches is explained. However, it will save you a lot of hard work if you get them.

Detail of how to load the patches are given in the notes accompanying the patches. Within this book, presets are identified by their name which is shown in **bold** for clarity.

Refills

This book is not the end of the story. As with all good synthesizers, refills are available. As you can see from the table of contents for the printed version of the book, four refills are available. These refills:

- look at more sounds – many more example patches are constructed: most of the patches are practical, usable patches
- feature more synthesizers (in particular, Surge from Vember Audio is now featured), and
- consider more features – such as arpeggiators and other playback features, and other sound generating techniques, including wave-shaping, ring modulation and feedback (to give three examples).

There are further patches to go with the refills which are included with the patches package. Anyone who has already bought the patches is entitled to an update package free of charge.

Enough introduction…

OK. On to the music.

Chapter 1: getting started

Read this first!!

The whole of this chapter could alternatively be titled "read this first". If you already have some knowledge about how to program synthesizers this chapter may seem over-simplistic.

If you haven't got a clue about programming, first make sure you are familiar with the user manual which comes with Vanguard (the synthesizer we will be using in this chapter) and then persevere with this chapter – it will all become clearer. For now I would like you to be happy tweaking knobs and making sounds even if you don't know what you're doing or why you're doing it. I will explain the whys and wherefores in the subsequent chapters.

Whether you're an experienced programmer or completely new to the subject, please don't skip this chapter as it raises some of the main themes I want to carry through this book.

In this chapter we are going to build three very simple patches in Vanguard:

- a bass sound – **first bass**
- a lead sound – **lead whine**, and
- a "plucky" sound, **arpeggio** – you can guess how this is going to be used.

I do not hold these sounds out as being the best you will ever hear – indeed, I am specifically designing them to be quite average (even though Vanguard is capable of making much more interesting sounds). In this chapter I am also limiting myself by basing each of the three sounds on a single sawtooth waveform and will only use a 24dB low pass filter.

However, I do want to illustrate two points in particular here:

- programming for a purpose – if you want/need a sound for a specific piece you should be able to design a sound to fit rather than hope you can find a preset that is "satisfactory" for the job, and
- how the sounds work in context – you will hear that on their own these three patches sound pretty uninspiring, however, together they work well.

On the former point, I would include tweaking an existing preset as part of the process of designing your own sound. Also, I wouldn't wish to rule out the notion that a preset can provide the correct sound for any track – I just suggest that it may often be easier to create your own sound rather than spend hours looking through your existing banks of presets. With all of these considerations, please do remember that the issue that matters is whether the track you are producing is any good, not how your came about your sound sources.

Synthesizer architecture

The next chapters will discuss the architecture of synthesizers in greater detail. However, the vast majority of synthesizers, whether hardware or software, produce sounds in a similar way. There are many factors in a sound's design, but in essence a sound will usually have three main elements present in its sound:

- the sound generator (usually an oscillator)

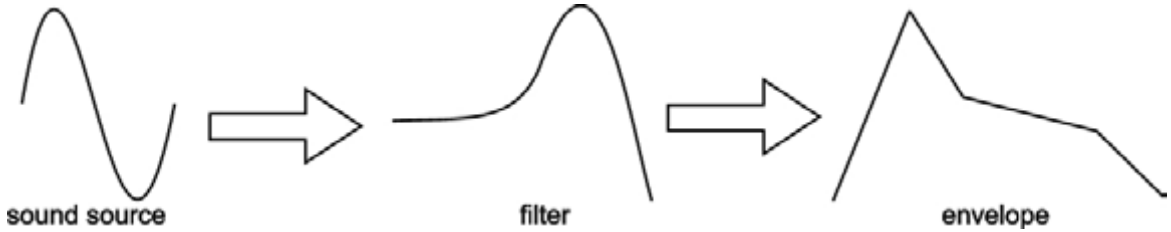


figure 1: the basic signal flow in the sound creation process

- the filter (to shape the tone), and
- the envelope (to control the volume, and perhaps the tone, over time).

All synthesized sounds emanate from a sound generator of some sort. In the world of software synthesis the sound source is some form of computer code.

Once the sound has been created, it is then "shaped" – both in terms of the tone and the volume over time – to provide a (hopefully) pleasing tone. It is quite possible that the pitch of the sound may be changed over time too. The control sources which effect these changes may include:

- envelopes, and
- oscillators (often low frequency oscillators or LFOs).

The term that is applied when one device controls (or changes) another is modulation. So if an LFO controls a filter, perhaps to create a wah-wah type effect, then we would say that the LFO modulates the filter (or more accurately, the LFO modulates the filter's cut-off frequency).

This is just a brief introduction to the workings of synthesizers, we will discuss the various elements in greater detail as we progress.

Start point for programming

Most sound designers like to have their own starting point when they begin programming a new sound. For most designers this will be a simple patch set up in a way that they know how everything works.

The patches we are going to make in this chapter are all quite simple and are based on a sawtooth wave and a 24dB low pass filter which will be my starting point. For the three sounds programmed in this chapter I will describe the construction process starting from my favoured Vanguard blank patch, **vanguard blank**. As you can see in figure 2, only a few controls are going to be tweaked to make these patches.

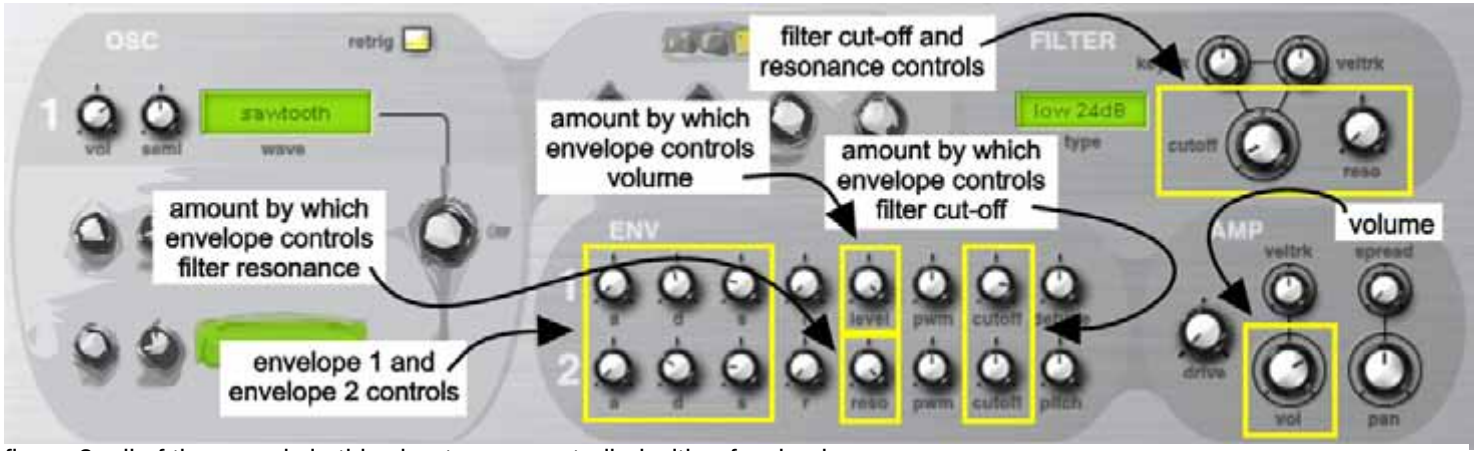


figure 2: all of the sounds in this chapter are controlled with a few knobs

In the zip file that contained the PDF of this book, you will find four .fxp files. These fxp files are the Vanguard patches for the three sounds described below. For those of you who have purchase the patch banks, these patches are also included in **vanguard HTMAN bank.fxb**.

The patches for the remainder of this book are available from www.noisesculpture.com. You do not need these patches to follow the examples, but you will find that a lot of the monotony in programming the examples can be avoided if you do purchase the patches. Also, many of the fine details of the more complicated patches are not set out in explicit detail.

First bass

The bass sound I want

The intention is that this bass sound will hold a track together but will not dominate. Therefore I am looking for something that is not too aggressive but not too soft. I want the low end but without any bite so it shouldn't be too bright but neither should it be too dull. I want something really average, but not bland.

Elements of the first bass sound

The volume envelope is used to give some shape to the **first bass** sound, so the note is more than just on or off. I have set the envelope as follows:

- I am using envelope one as the volume envelope . The attack of the volume envelope is slowed slightly – not by much, but by just enough to take some of the aggression out of the attack. In this case the attack time (ie the time it takes the sound to go from zero to its maximum) was set to 14ms.
- The decay time is also slowed down enough that its effect can be heard – here the decay has little audible effect unless the sustain level is set considerably below its maximum. In this instance I have set the sustain level to 42 (where it has a maximum possible value of 127). As a general rule, you will find it much easier to adjust the decay time after you have set the sustain level and if you're unsure about the sustain level, set it too low and then increase it once you have sorted the decay time.
- So now we have set the volume envelope we need to apply it (ie make it have some effect on the volume). To do this adjust the "level" knob. In this case, I set the level knob to 127 so that the level is fully controlled by the envelope.

The filter was closed down to 963Hz. This gives quite a dull, lifeless sound so I did two things to improve it. First, the volume envelope was set to slightly modulate (in this context, open up) the filter. For this patch, I set the envelope control of the cut-off filter to 34 (with a maximum possible value of 127). I also increased the resonance of the filter to 36% – this gives a brighter, slightly less thick quality to the bass note. The effect of the resonance is subtle but it is more noticeable when the envelope modulation is added.

Looking at the modulation of the filter by the volume envelope, the decay time that has been chosen is important:

- if it is very fast you won't hear the effect of the envelope
- if it is quicker than ideal and you'll get a bit of emphasis

- just right (I chose 136ms) and you'll hear enough emphasis on the attack of the note (much like hitting a bass guitar string with a plectrum)
- too slow (say over 250ms) and you get a squelchy sort of sound which may be fine in other circumstances but is not what I'm after here. Equally too much filter resonance would have made this sound squelchy – there are several places you need to look if you're trying to dry out "squelch" (for instance, as in this example the decay control or the way the envelope controls the filter – equally, the filter cut-off and resonance control will all have an effect).

Arpeggio

The arpeggio sound I want

The sound I want for this arpeggio is a thin, bright, staccato (almost percussive) sound. Think of a harpsichord, but not so harsh, and you will be getting close.

Elements of the arpeggio sound

For **arpeggio**, as I'm going to want the volume envelope (envelope one) to have a significant effect, I set the level control in envelope one to the maximum (127). To get the attack I am after, the attack in the volume envelope was then set to zero (ie the fastest attack).

I set the sustain level to 27 (out of 127) – any lower and the sustain portion of the patch would not have been audible, thereby making the sound too staccato for my taste in this context. Having set the sustain level, I set the decay time. In this instance, I set the decay time to 357ms – long enough to hear the note, but short enough to give that staccato harpsichord type sound.

The sustain portion of each note is too bright for my taste, so I fixed this with the filter by turning the filter down to 112Hz. I could have gone lower (ie had even more effect), but then the sound lost some of its clarity when I applied the envelope (see below).

Now we have the right envelope but the sound is quite dull and is virtually silent because the filter is closed. So what I did was to apply the volume envelope to modulate the filter. To do this I set the cut-off control in envelope one to 89 (out of 127). This gives the sound some brightness during its initial attack and decay stages.

While this sound is better it still wasn't bright enough so I added some resonance in the filter – this makes the sound brighter and thinner. I could have simply turned up the resonance control in the filter section – this certainly gives a brighter sound. However, because envelope one (the volume envelope) is modulating the filter it would also mean that the sound gets more reminiscent of a laser gun being shot in a science fiction movie. So I didn't take that option.

Instead, I modulated the filter resonance with the second envelope: as a first step the "reso" control in the second envelope bank was turned up to its maximum (127). I then set the envelope – first I set the attack to zero (ie the fastest time). Next I set the sustain level – here I wanted to ensure there is a bit more brightness on the sustain portion of the note, but not too much, so I set the level at 22 (on a scale of 127). Lastly we needed to set the decay time – too fast and the effect of the envelope would be lost, too slow and we would get the laser gun effect. 108ms sounded and felt right to me.

The final thing I did with this patch was turn down the volume to 90 – this balances the levels of the three patches.

Lead whine

The lead sound I want

For this lead sound, I was looking for a very muted, almost whining, tone with a characteristic much like a voice through a vocoder or a guitar talk box. I'm not looking for a burning/searing lead type sound.

Elements of the lead whine sound

As a first step in creating **lead whine** I turned down the filter cut-off to 2.74kHz and turned up the resonance to its maximum. If you hold a note and play with the resonance control, as you sweep it from zero to full you will hear a vocal-type sound. I wouldn't go as far as saying one extreme (no resonance) gives an "oooh" type sound and the other extreme (maximum resonance) gives and "aaah" type sound, but we are getting there, slowly. Anyway, leave the resonance at the maximum.

I wanted a gentle volume envelope (envelope one), but it must be effective, so I set the level control in envelope one to the maximum (127). The attack time was set at its fastest (zero) and I turned the sustain level down to 29 (out of 127). Finally I adjusted the volume envelope decay time to 2.82 seconds. This gives a fairly plain sound.

To complete the patch I wanted to capture the character of a guitar talk box – to me this is best represented by the classic sound of morphing vowels. To produce this effect (albeit in a small way) I am modulated the filter cut-off frequency with envelope two. For a start I set the attack time of envelope two to 923ms and gently applied the envelope to modulate the cut-off. To my mind setting the envelope to modulate the cut-off at 13 (on a scale of -127 to 127) is just right – any more and the sound becomes too bright and loses that talk box quality.

To complete this sound, I turned down the sustain level – this means that after the attack and decay portions of envelope two, the filter is less modulated. With the sustain level set, I tweaked the decay time – at about 2.63 seconds it felt right to me.

As a final step, the volume was reduced to 65 to stop the output distorting.

Using the three sounds in a mix

In the zip file that came with this book in addition to the four fxp files you will find a midi file called **chapter1.mid**. Load the midi file into your sequencer and load up three instances of Vanguard:

- Assign the first midi track (called "lead") to the first instance of Vanguard and load the patch **lead whine**.
- Assign the second midi track (called "arpeggio") to the second instance of Vanguard and load the patch **arpeggio**.
- Assign the third midi track (called "bass") to the third instance of Vanguard and load the patch **first bass**).

First play each track solo. You will hear that the sounds are individual not very inspiring. Now play them together and you will hear that the track works well. You are not aware of the static quality of the bass (its sustain phase is masked by the arpeggio). The brightness of the arpeggio is balanced and rounded out by the bass and both of these parts support the lead part.

Why did we make these sounds?

You can hear that these sounds are quite simple and are not that interesting on their own. However, I want to illustrate a few points.

Designed for purpose

I have created three sounds here for a specific purpose. I think the end result (ie the patches used in the context of a track) has been good. If you are creating/editing sounds for a specific purpose you stand a far greater chance of getting the right sound which then fits in the mix.

Although these sounds are not interesting on their own, in the context of the track they work well together. While I wouldn't claim that any musical virtuosity is displayed in the midi file, the sounds and the track are a perfect marriage. With other sounds – for instance, a searing lead, and a thundering bass – the track probably would not have worked.

Arrangement

Note the arrangement of three parts. You can hear that each sound occupies a different area of the sound spectrum:

- the bass fills the lower frequencies
- the arpeggio fills the higher frequency range, and
- the lead fills the mid range (which is comparatively uncluttered by the other two parts).

When each sound has its own place in the frequency range, it will be clear and distinct – you may not always want that clarity (for instance, if you are layering), however, you will be more likely to have your mixes criticized for being "muddy" than for being "too sparse".

You can also hear that I have a certain luxury here – I have full control over both the music and the sound. You may not always have this level of control. However, if you do have this control, then the arrangement of your track will allow you more flexibility – especially if you want to use some "big" sounds that fill a large elements of the sonic spectrum.

Modern music is highly compressed – if you want your music to stand out, then you will probably compress your music too. If one area of the frequency spectrum is dominant this will mean that the compressors will be disproportionately affected by that part of the spectrum making the rest of the mix comparatively quieter. This is the case even if you're using a multi-band compressor where you will still have one band (or more likely several bands) dominated by one sound. The net effect will be to take energy out of your mix and make it sound comparatively quieter.

Editing sounds – balancing parameters

I see no reason why you shouldn't edit an existing sound to make it suitable for your purposes – there is no law that says all patches have to be created from a blank patch each time. However, I hope these three sounds illustrate that there is generally no one knob that can be tweaked to change a sound. If someone tells you just to change a filter to make a sound less bright, they are either lazy or ignorant. However, you may achieve the effect by just tweaking the filter knob: it all depends on the structure of the patch.

Every patch will be a combination of many factors and you will have to balance several controls to change your patches in a suitable manner. One aspect that these patches particularly highlight is the effect of the decay time on the sounds – this is particularly so for any patch where the envelope controls a filter or a filter's resonance. Play with the decay control in the bass and arpeggio patches and notice the effect it has.

Simplicity

These sounds all use:

- the same filter, and
- the same oscillator

and yet they sound different.

The main differences arise because of the envelope and also the application of the envelope to the filter parameters.

Remember too that these sounds are all monophonic and have no FX.

Programming in context

None of these sounds is interesting if you listen to it in isolation. None of the sounds works on its own (although the lead might just). There is no doubt that these sounds could be made more interesting, but as I have said, in the context of this track, they work together well.

The three patches are therefore perfect. If I had to suggest improvements, it would be to make the patches more playable – for instance to add some dynamics so that the tone and/or volume change with different levels of velocity.

If you are programming out of context (by which I mean you are programming without your track playing and so you cannot hear all of the other parts and how your patch sounds when it is played and in the context of the mix), then you have a difficult job. You may get a brilliant sound, but I would question whether it will work in the track without further editing.

Let me get to the heart of what I am going to tell you in the rest of the book. In the days before digital audio workstations when hardware ruled the earth, you had to go into a recording studio to make a professional recording. Those days have gone – it is now quite possible for one person to produce a record on a computer at home which sounds as good as a record recorded in the most expensive recording studio.

In the days of hardware and recording studios, there were two separate processes: recording and mixing. In today's computer based studios, the processes of writing, recording and mixing have merged. There is now no longer an engineer on hand to ensure that the frequencies in the mix do not clash nor a producer to hold your hand through the process.

Instead with computers there are usually one or two people making all of the decisions – without the producer and engineer to guide this leads to second rate mixes. I have a simple suggestion: design your sounds properly and your tracks will mix themselves (in terms of the respective sounds working together). I am not trying to imply that you won't have to do anything with the mix, simply that if you get your sounds right and working within their own areas of the sound spectrum so that they don't fight each other, then it is much easier to balance the respective levels of the elements.

I will return to many of these themes throughout the book.

Chapter 2: envelopes

I now want to introduce the elements of sound design in a bit more detail. You might be expecting to look at sound sources first – there would be a certain logic given that the sound comes at the start of the signal. However, you're going to have to wait a little while longer, because first we're going to talk about envelopes which can have an effect on a sound which is more dramatic than a simple waveform tweak.

Volume envelopes

The most immediate use for envelopes is controlling volume.

Think of a note played on a piano. When a key is struck, the note goes from silence to the maximum volume instantaneously. From this peak, ie from the moment of impact when the hammer comes into contact with the string, there is an immediate rapid reduction in the volume and then the note reaches a level from which it gradually fades to nothing. A picture of the volume of a piano note over time is set out in figure .

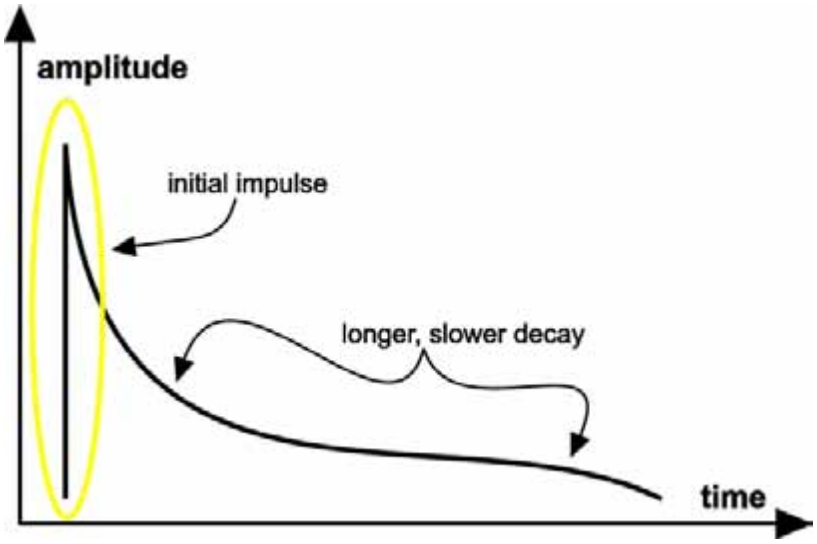


figure 4: the volume envelope of a piano note

Now if you think about a violin note which slowly fades in and then stays at its maximum volume until the note ends, at which point the note gently decays, this may look like the image in figure .

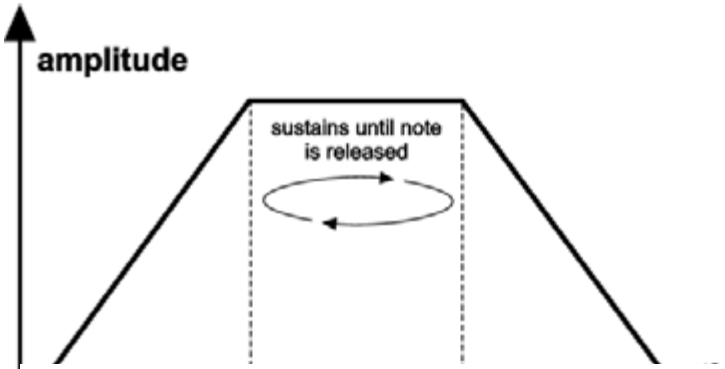


figure 3: a typical sustained string envelope

What else can envelopes do?

Getting more complicated, an envelope changes level over time – its effect will depend on how it is applied. For instance:

- An envelope could control volume (as we have seen above). Depending on the architecture of the synthesizer, the envelope may control the level of an individual oscillator or the level of a whole patch.
- An envelope may control a filter. If an envelope does control a filter, it will (generally) control the cut-off frequency and so make the sound brighter or (more usually) duller over time. If an envelope modulates the filter's resonance then it will control the amount of the resonance – you could set one envelope to close the filter over time and another envelope to increase the amount of resonance while the filter is closing.
- An envelope can also modulate pitch, a common use for this would be to give a short (and subtle) pitch wobble at the start of a note to give the sound more emphasis.

As we will see in chapter 5: modulation and other ways of messing with things and chapter 6: modulation in practice, these are not the only uses for envelopes.

Different synthesizers are designed in different ways. Let's look at some of the more common types of envelope.

ADSR envelopes

The ADSR envelope is the "classic" envelope. This envelope is used in Vanguard and in Wusikstation. There are four main controls:

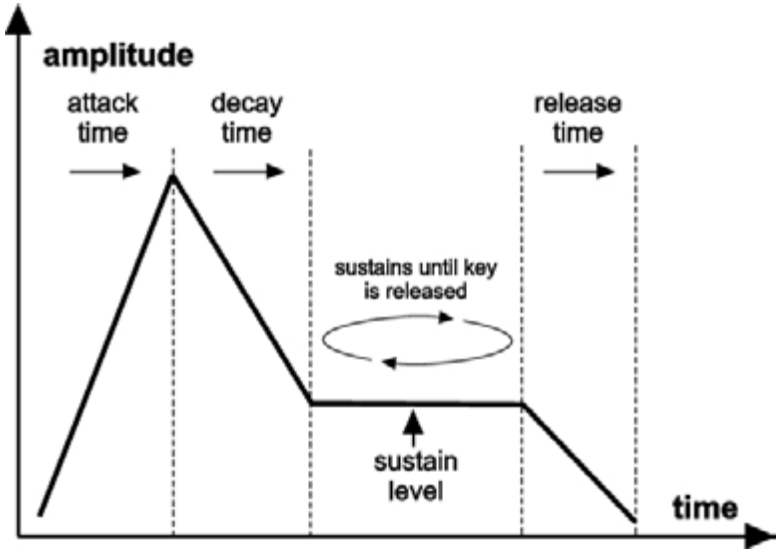


figure 5: the ADSR envelope

- A: attack time – this governs the time it takes for the sound to reach its maximum volume after the note is triggered. Using the example of a piano, the attack time would be zero – ie it would take no time for the to go from nothing to the maximum volume. For a string type sound the attack may be slower.
- D: decay time – this controls how quickly the sound drops (to the sustain level) after it has reached its maximum volume). Again, using the example of the piano, the decay time would be fast, but it would be longer than the attack time.
- S: sustain level – this is the volume of the sound (or the level of the envelope) while a key is held. This level stays constant until the key is released – this may be perceived as a weakness if you are using this type of envelope to mimic the behavior of a real instrument where the volume will continue to gently decay over time.
- R: release time – this is the time it takes the sound to decay to zero after a key is released.

You will notice that with this envelope:

- once the sustain part of the envelope has been reached the note does not decay until the key is released, and
- there is no function in the envelope to determine how long the note sustains (the only control over this is by releasing the key).

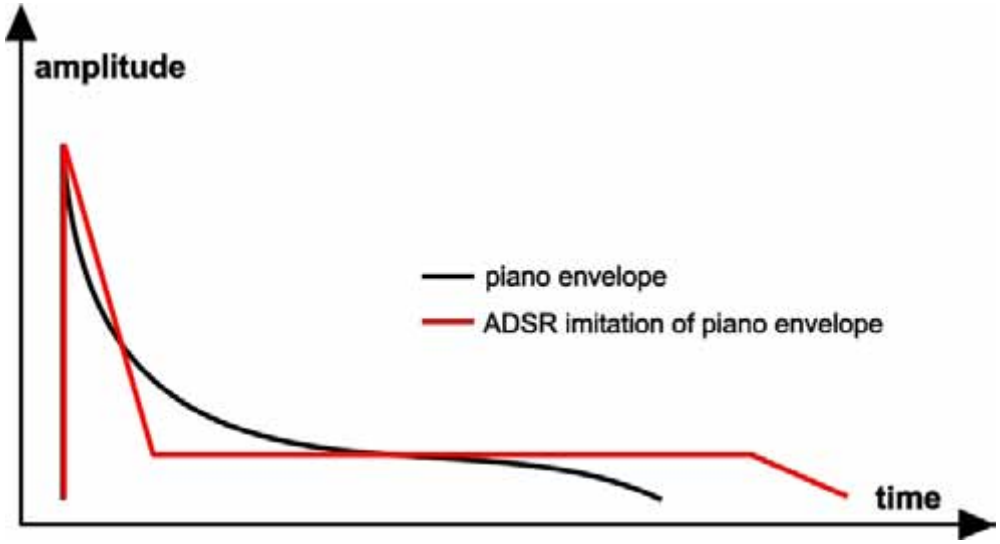


figure 6: the ADSR type envelope may not be the ideal choice for piano type patches

So, assuming you can get the attack and the decay right for a piano type patch, you would not be able to accurately replicate the piano since the envelope does not decay to zero over time – figure 6 illustrates the differences in crude terms.

The next weakness with this type of envelope (and this applies to all envelopes) is that real sounds do not necessarily increase or decrease in a linear manner. Take the example of a slow swelling violin – in practice the attack of the note is likely to have two phases:

- first the note will go from nothing to a very quiet level very quickly, then
- the note volume may increase exponentially.

You could almost see the attack as having three phases – a fast phase, followed by a slow phase, followed by another fast phase: figure 8 illustrates this point. It also shows why it may be difficult to use a synthesizer to accurately replicate natural instruments.

So if the ADSR envelope has limitations, what other choices are there? Lots, but let me just stop you there. I don't want you to think of the ADSR envelope as being limiting – many synthesizers have this style of envelope for good reasons: it works and it is easy to use.

I should also point out at this stage that if you want a piano sound, the optimal solution would be to hire a studio with a piano and a good recording room and get

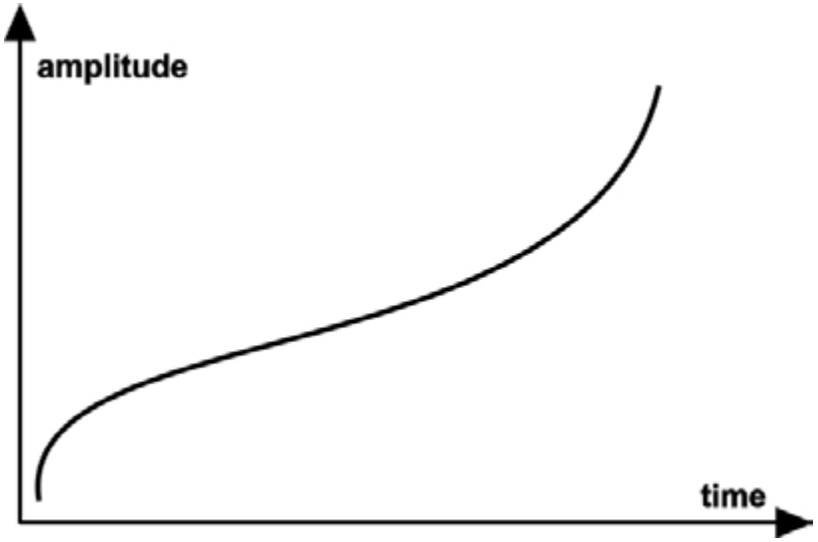


figure 8: the attack phase of a "slow" string sound

an experienced piano player to record the part. Failing that, there are some excellent sample libraries with highly detailed and playable pianos available, but again I suggest you find a skilled piano player.

DASSDSR (and more) envelopes

If you look at figure 7, you will see that Z3TA+ takes a different approach to envelope design – at first it may appear complicated, however this just means it is more flexible. This is what the Z3TA+ envelope does:

- Del: the delay before the envelope begins – generally, this isn't used, but it is useful, especially if you want different elements of a sound to come in after the initial attack.
- Att: attack time – this controls the time it takes the note to reach its maximum level once the envelope cycle has begun (ie after the delay has ended).
- Slt: slope time – after the completion of the attack phase, the envelope enters the slope stage, this control governs the time it takes the envelope to decay from its maximum level (at the end of the attack phase) to the slope level.
- Sll: slope level – this sets the level that the envelope will reach at the end of the slope stage.
- Dec: decay time – the time it takes the note to change from the slope level to the sustain level (note that unlike an ADSR envelope, this change could be an increase or a decrease in level).

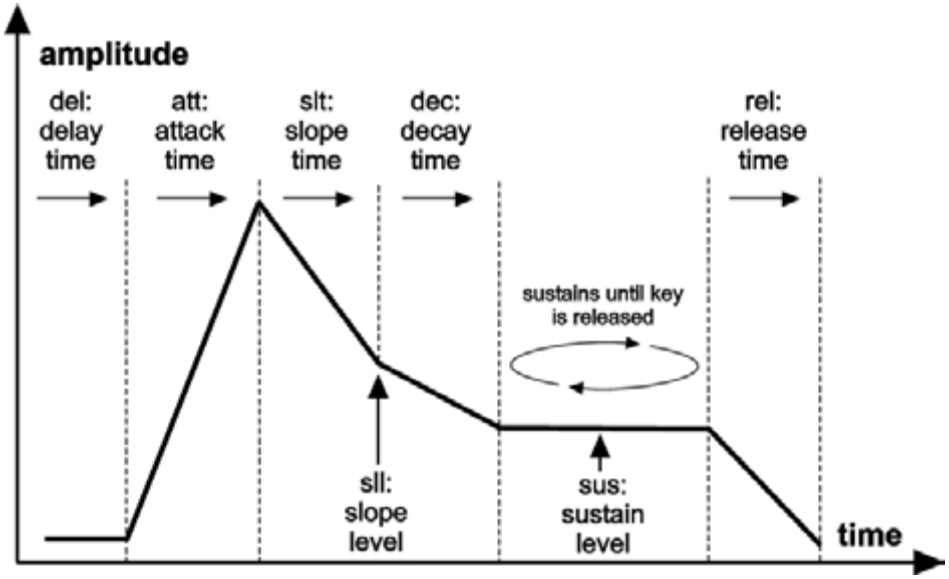


figure 7: the Z3TA+ envelope offers more flexibility than the conventional ADSR envelope

- Sus: sustain level – this sets the level at which the envelope will remain until the key is released.
- Rel: release time – the time it takes the note to reach zero after the key is released.

Not only does Z3TA+ give a more flexible envelope, it also gives the sound designer three options about how a level increases or decreases over time. So for the attack time, slope time, decay time and release time you can choose:

linear change, so the level changes uniformly over time

exponential change, so the change is slow to start but gets more dramatic over time, and

"power" (or logarithmic) change so the level initially changes quickly, getting slower over time (think of it as being the inverse of the exponential change).

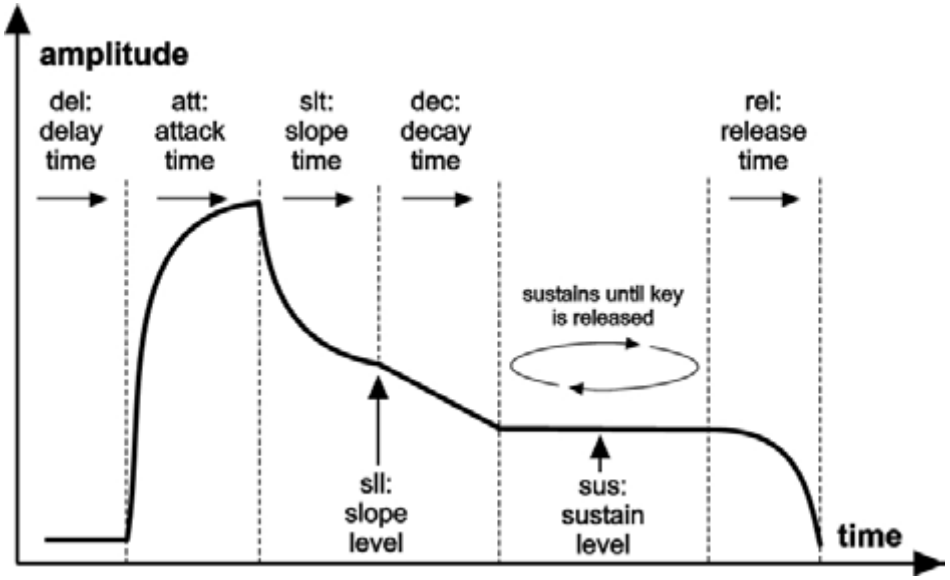


figure 9: the Z3TA+ envelope also gives you control over the slopes

The Z3TA+ envelope can act like an ADSR envelope (indeed, that may be a good starting point for programming sounds) – to do this, set the delay time to zero, the slope time to zero and the slope level to maximum.

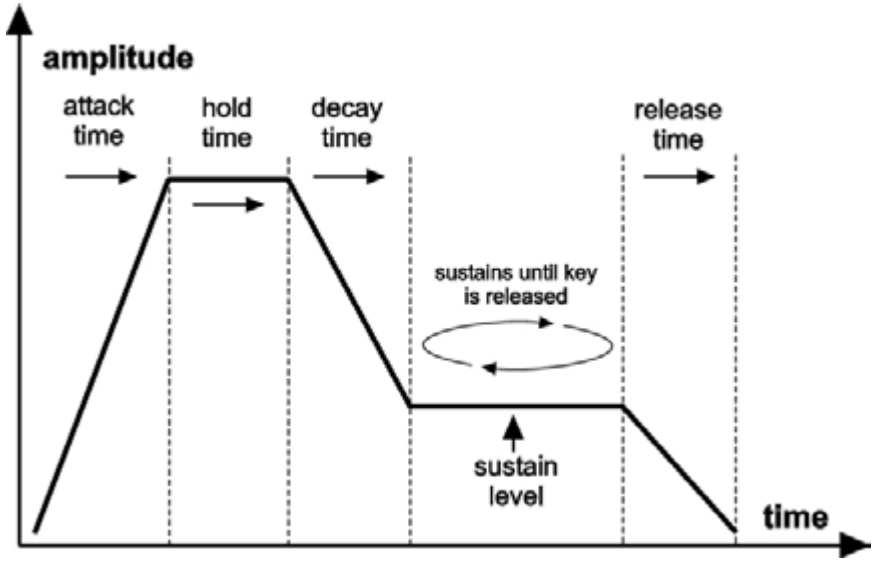


figure 11: an AHDSR envelope

Another use for this envelope's flexibility is to increase the time that the sound stays at the level reached after the attack phase has been completed – this can make the sound behave as if the signal is being constrained by a compressor or a limiter. This stage of the envelope is often called the "hold" stage – maintaining this level (only for a short period) can give a sound more "punch". Try it. Figure 11 shows a typical AHDSR envelope – attack, hold, decay, sustain, release. The AHDSR envelope is not used by any of the synths mentioned in this book, but it is used in Pentagon I from Cakewalk.

Z3TA+ to imitate a piano

Z3TA+'s DASSDSR envelope does not provide a perfect envelope for a piano – however, it can make a reasonable imitation.

When imitating a piano (or any other acoustic instrument which decays over time), with the Z3TA+ envelope you can:

- set the attack time, slope time and slope level to resemble the initial impact of the hammer (remembering especially that you can change the character of the slope), and
- then set a long decay time with a sustain level below the slope level – this will give the character of the note decaying over time.

This still doesn't make a perfect emulation of a piano's envelope (not least since this envelope still does not decay to zero – it will always remain at the sustain level until the key is release). However, it does give the sound designer more flexibility.

Let's build a simple piano type patch in Z3TA+ – the emphasis here is on simple:

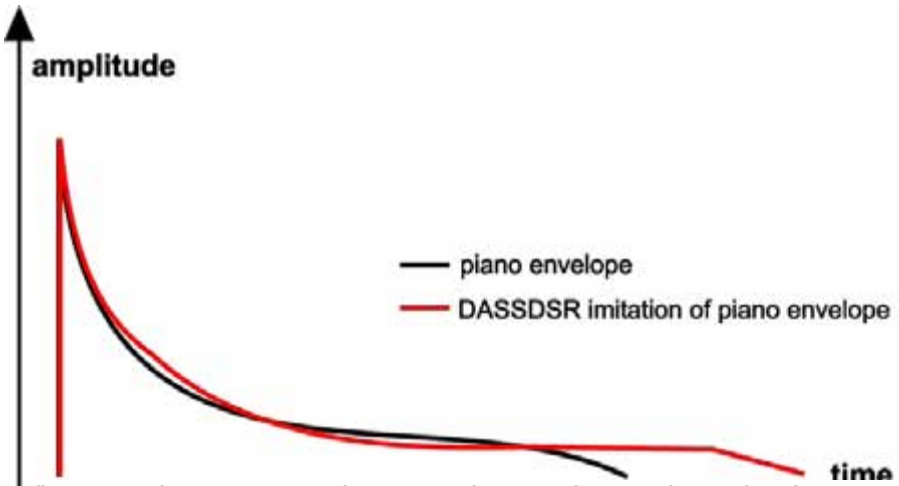


figure 10: the Z3TA+ envelope gets closer to the envelope of a piano

we're not on trying to fool any concert pianist. If you have purchased the patches, this patch is **simple Z3TA+ piano**. If you haven't purchased the patches (yet?!?) the settings are listed in the text below.

- First we will load up two oscillators – in the first oscillator load the piano wave and in the second load a sine wave. To my mind the piano wave sounds alright in the lower registers but in the higher registers it sounds a bit too sharp – I used the sine wave to give the tone a bit more roundness (you will find I add pure sine waves to patches quite often). We will look at layering sounds in greater detail in chapter 4: sound sources.
- Next set the amplitude envelope – this envelope always applies, you do not need to do anything in the modulation matrix to make it have effect:
  - delay and attack are both set to zero
  - the slope curve is set to exponential, the slope time to 0.39ms and the slope level 51%
  - the decay time is 0.39ms and the sustain level 23%, and
  - the release time is 0.05ms.
- Lastly we will add a bit or reverb and for this
  - engage the plate reverb
  - set the size to 45%

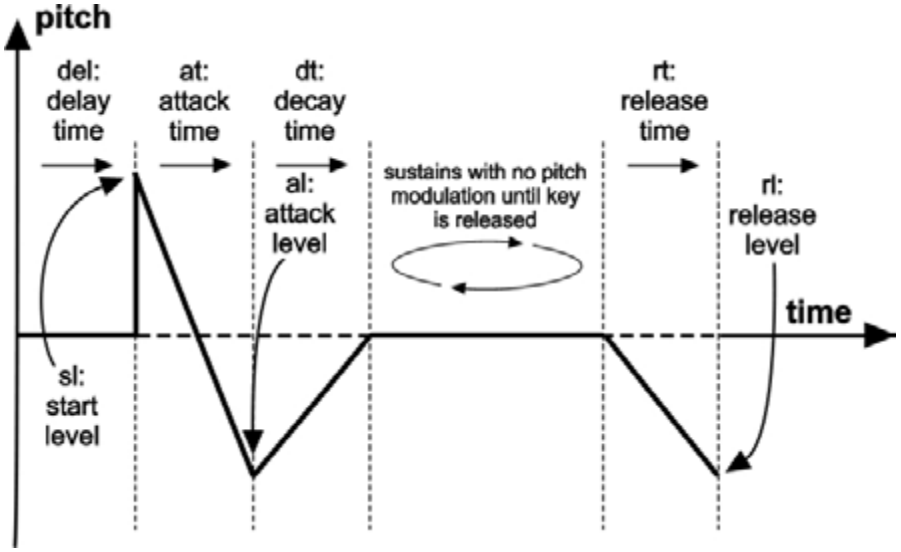


figure 12: the Z3TA+ pitch envelope

- set damping to 50%
- set the high and low EQ both to -6dB, and
- set the wet/dry slider to 60%.

And there we have it, a rather crude piano type patch. If you play this it will sound more realistic in the lower registers. The higher registers may be useful if you play a clavinet type part.

Other envelopes

Z3TA+ pitch envelope

Z3TA+ also comes with an envelope dedicated to controlling the pitch. The other seven envelopes can control the pitch too, but they are unipolar (ie they only give positive values). The pitch envelope is bipolar – this means it can give positive and negative values in a single cycle. In practice this means that the pitch envelope can raise AND lower a note (where a regular envelope could raise OR lower a note).

Rhino envelopes

Rhino takes a different approach and allows you to draw every point on your envelope and to precisely control the shape of the curve between each point. This has two main advantages:

control – you have very precise control over the design of your envelope, and

rhythm – you can draw rhythmic envelopes which can be synchronized to the tempo of your track.

The disadvantage is complexity, however, this is largely outweighed by having a graphical interface (see figure 13).

Rhino also offers the facility to save envelopes and comes with a bank of envelope presets which can be really useful for creating patches quickly (as some of the examples will demonstrate).

Rhino piano

Let's create a simple piano patch in Rhino: for those of you with the patches, this is **Rhino piano**. The purpose of this patch is twofold – first to introduce Rhino's envelopes and second, to demonstrate how these envelopes can create a more convincing envelope than some of the other options.

To create the sound we will use two oscillators – in oscillator one we will load the waveform Hard 88 and in oscillator two we will load the waveform FM tines. Both of these waves can be found under the "electric piano" group of waves. Once loaded, we will drop the pitch of oscillator one by an octave.

The two waveforms are samples of real instruments, so even with a simple envelope (with no volume control) the sound already resembles that of an electric piano.

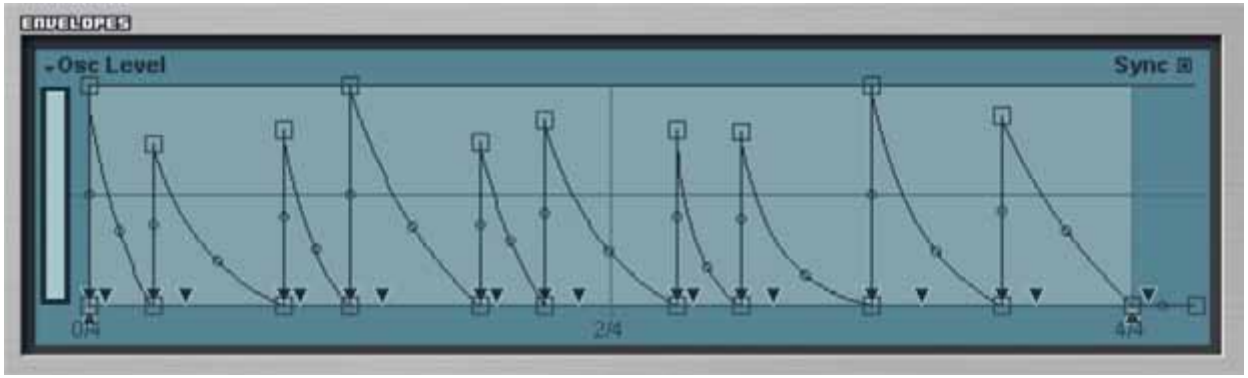


figure 13: Rhino allows you to draw rhythmic envelopes

However, if you sustain a note, then it starts to sound unnatural – the volume does not decay as it would in a natural instrument. To remedy this we will load some envelopes.

We could draw some envelopes, but it is much easier and much quicker to load some of Rhino's preset envelopes. In the envelope settings/level folder there is an envelope called "level piano.env" – I have loaded it for both oscillators. In oscillator one, I then adjusted the curve of the decay (to 25) to give the impression of a slightly faster decay.

As you can hear, a sample and a preset envelope can be called into action to produce a natural tone quickly and easily.

Cameleon 5000 envelopes

The Cameleon 5000 takes a slightly different approach to envelopes, but this reflects its different way of creating sound. Cameleon does have a fairly (but not entirely) conventional amplitude envelope and a dedicated modulation envelope, but this is not where the interesting stuff takes place.

Cameleon 5000 is primarily an additive synthesizer, although it does have subtractive synthesis features. With additive synthesis, the sound is made up by a number of sine waves – the combination of sine waves varies over time. Cameleon envelopes have a number of breakpoints: at each breakpoint the combination of sine waves is reconfigured. Between the breakpoints the sine wave configuration morphs from one combination to the other. A similar process takes place for the noise element of the Cameleon sound.

A fuller explanation of additive synthesis is given in chapter 9: additive synthesis.

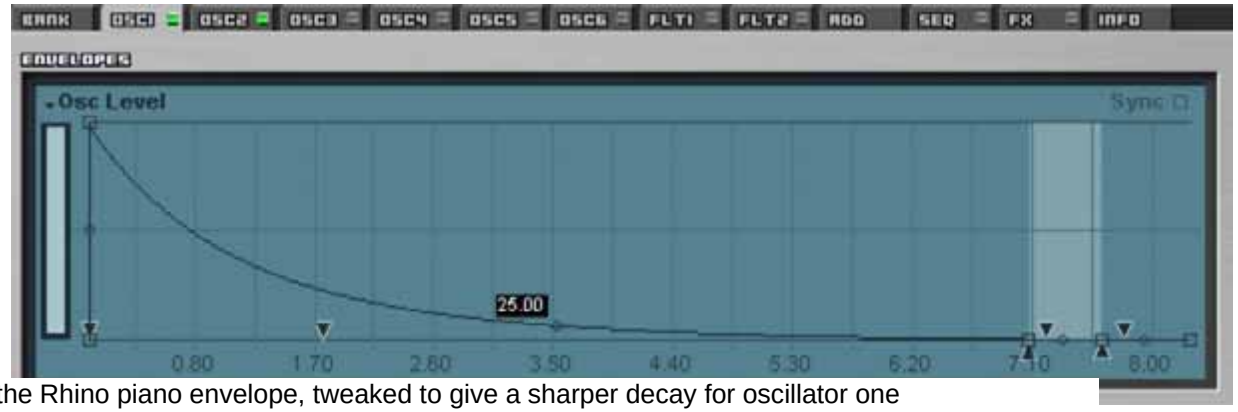


figure 14: the Rhino piano envelope, tweaked to give a sharper decay for oscillator one

Envelopes and samples

For sample based synthesizers, such as Wusikstation, the samples will have their own volume envelope. In this case you should think about the interaction between samples and the envelopes. If the wave has a slow attack time, you cannot make it faster simply by choosing a fast attack with your envelope. If you want to get more attack in this situation, you are going to have to change the place that the sample starts to play from – this will have a secondary effect (which may or may not be desirable) of changing the sound and feel of the sample.

Conversely, you can take a sample with a fast attack time and apply a slower envelope. Again, this will change the sound of the sample (which is sort of what we are trying to do…).

For the purpose of this book, when I talk about "sample based" synthesizers, I am referring to machines that can play the whole length of a sample or load a multi-sample (such as Wusikstation or Rhino). I am not calling machines such as Z3TA+, which has the facility to load single cycle waveform, a sample based synthesizer. Equally I will not refer to Cameleon 5000 as a sample based synthesizer as it resynthesizes. The distinction between the terms is a fine one and only intended for clarity rather than as qualitative assessments on the strengths of any of the featured synthesizers.

Key tracking and envelopes

I want to introduce a new concept here: key tracking. This concept is used in several areas. Essentially key tracking means dynamically changing an element of the sound depending on the pitch. So if we think about the volume envelope of a piano, as the notes get higher they sustain for a shorter period and conversely, as the notes get lower they sustain for longer.

If we want to replicate this behavior on a synthesizer we would use key tracking.

Effect of envelopes on sound

So why did we start by talking about envelopes and not oscillators? Quite simply because the envelope is perhaps one of the most important tools in creating sounds. A lot of nonsense is talked about sound sources – and don't get me wrong, these are important – however, if you use your ears, I think you will find an envelope can change a sound just as much (if not more) than changing an oscillator.

You should also be aware that the volume envelope of a real instrument is a ferociously complex thing which can be affected by many factors. No synthesizer envelope will come close to replicating the complexity of the envelope of a real instrument.

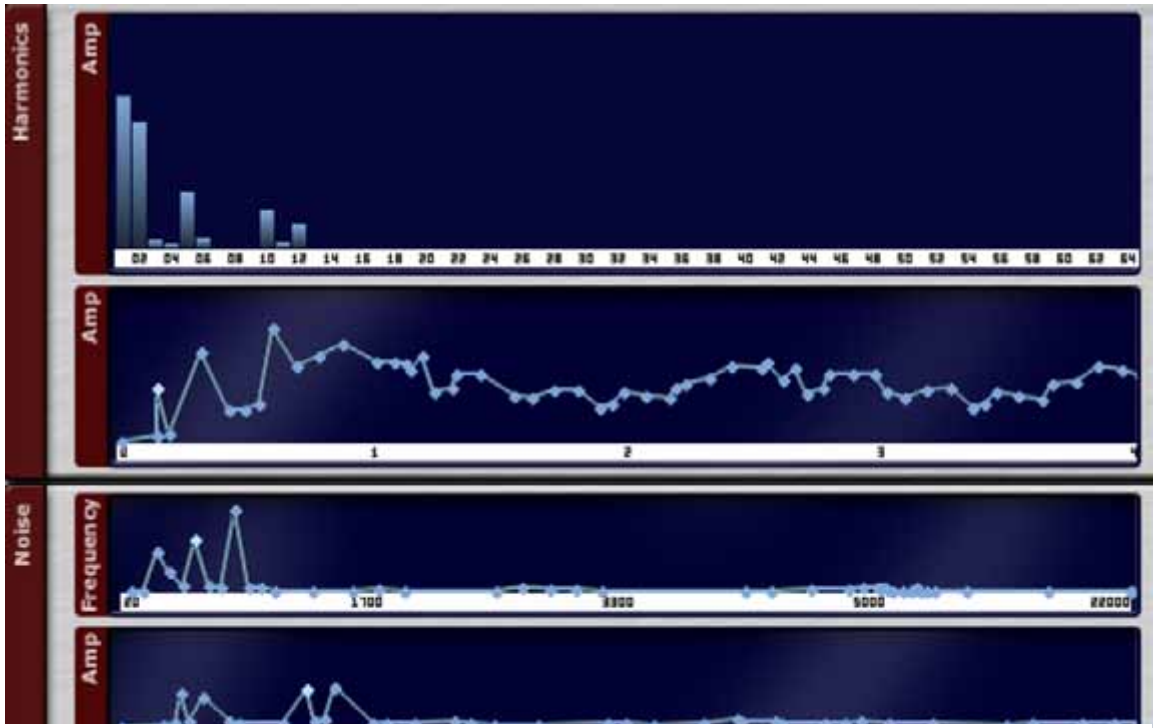


figure 15: Cameleon 5000 envelopes – too difficult to explain in a caption

Chapter 3: filters

At its most basic you can think of the filter as being a tone control. A low pass filter is like the treble control on your stereo – turn it down and the sound gets "duller". However, a filter can do much more for you.

Vanguard gives us one filter – but with lots of options (in fact, probably enough options to suggest that it really gives us two filters in one block). Cameleon gives us two filters, but that's not really the real point of the synth. Rhino gives us two filters as does Z3TA+ (although Z3TA+ gives us two stereo filters so that probably makes four). Oh yeah, Wusikstation gives us 28 filters.

All of the filters are designed to work in slightly different ways: this will affect the sound you hear. One 24dB low pass filter will not sound the same as another 24dB filter, so don't try to make them sound the same – its just too much of a dull job to try.

Filter types

Low pass

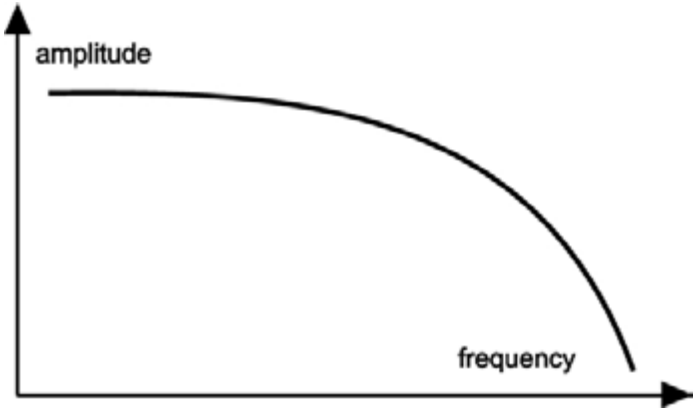


figure 16: a low pass filter

The low pass filter (or if you prefer, high cut filter) allows low frequencies to pass through it.

When a low pass filter is fully open, all frequencies can pass through it (although some filters do cut the signal even when they are fully open).

As the filter is closed it progressively allows less sound to pass through – you will hear this as the sound becoming duller as the higher frequency elements of the spectrum are filtered out. When the filter is nearly completely closed only the very lowest elements of the frequency spectrum can pass.

The effect that the low pass filter will have on a sound will vary depending on the waveform you have selected. If you choose a sine wave, then the effect of the filter will be limited. The sine wave comprises only the fundamental frequency, therefore if the filter cuts this, it cuts the whole sound. However, if you choose a sawtooth wave which has a lot of high frequency information, then the low pass filter will have a much greater perceived effect on the sound.

However, don't think that if you're using sine waves you won't want to use a filter. For instance, if you are playing chords with a sine wave patch then the individual sine wave notes will interact to produce frequencies beyond the range of the individual sine waves – the filter will affect these frequencies.

High pass

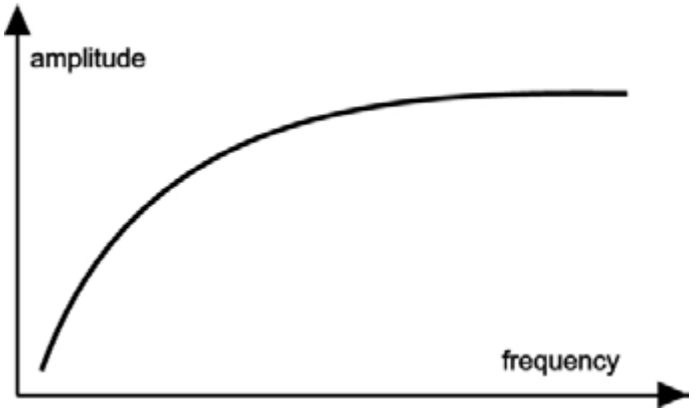


figure 17: a high pass filter

The high pass filter (or low cut filter) is the reverse of the low pass filter – it allows high frequencies to pass and progressively cuts out the lower frequencies.

As well as sound shaping, high pass filters have another use: to filter out the junk in the lower end of the mix spectrum. How often have you listened to one of your tracks and find it sounds muddy or dull. That could be too much bass. You only get so much dynamic range and without filtering you may be filling your low end needlessly. This means that the key elements – your bass and kick – can't shine.

While high pass filtering may be noticeable if you play a patch on its own, in the mix it is unlikely to be noticeable except with more extreme cuts. However, the net result of the low end filtering may be to give a cleaner/fuller low end when the bass elements are allowed to come through.

Some sound designers also use high pass filters with resonance (see below) to boost the fundamental tone of a patch.

Band pass

The band pass filter acts like a combination of a low pass filter and a high pass filter by cutting the frequency spectrum at both ends to only allow a narrow band of sound to pass. The frequency control determines the centre frequency where the full signal is allowed to pass – from that point outwards, the spectrum of sound is

progressively cut. At extremes of frequency, the band pass filter will sound similar to either a high pass or a low pass filter.

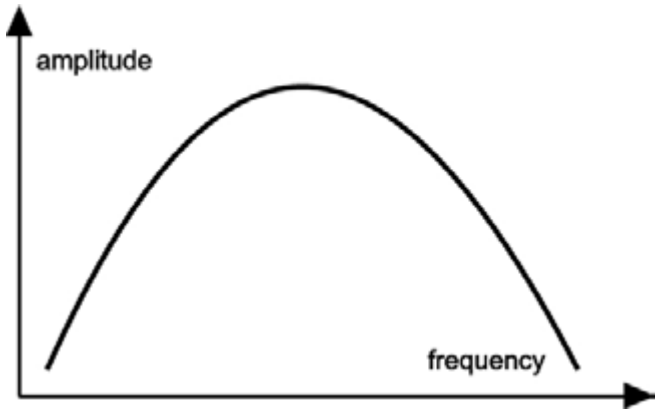


figure 18: a band pass filter

Band pass filters tend to take a lot of energy out of the signal so you may have to boost the level after the filter. They are also responsible for the classic "telephone" sound.

Notch

If the band pass filter is the equivalent of burning the candle at each end, then the notch filter equates to burning it in the middle. The notch filter cuts the frequencies at its current value.

The notch filter can be used for effect and it can be used surgically in the mix. If you're trying to mix two sounds and they don't sit well together it may be that they're both trying to operate in the same frequency range. In this case, you can "notch out" one of the sounds to allow the other to sit properly – you can do this in your patch design or, as many mixing engineers do, with EQ in the mix.

Formant

Formant filters are usually used to emulate vowel sounds – this is what Z3TA+ does. The Vanguard formant filter has a character that produces more resonant peaks and the Cameleon formant filter is described as being like a powerful multi-band graphic equalizer – it does have the facility to make vocal type sounds, but it can also do more.

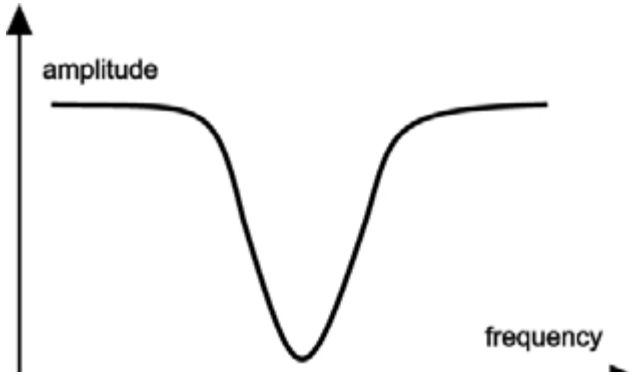


figure 19: a notch filter

Comb filters

Comb filters work by adding a slightly delayed version of a signal to itself. This causes phase cancellations and can give a slightly "chorused" or metallic type of sound.

The spectrum produced by these filters looks like a comb, hence the name.

Of the five synths considered in this book, only Rhino has a comb filter.

Combination filters

Combination filters are not a different filter type in their own right – instead they are combinations of existing filters. The most obvious example of a combination filter is the notch and low pass filter in Vanguard.

However, there are other examples that could fall to be considered as combination filters, for instance, Z3TA+'s 24dB and 36dB filters are actually stacked 12dB filters. In normal use this doesn't make a difference, however, using the separation control, the cut-off frequencies of the stacked filters can be separated (ie one will be raised in relation to the other) – this can give differing resonant peaks (see "resonance" below) and a different tone to the filter.

Filter parameters

Filters have several parameters and the controls available differ between the five synthesizers featured in this book.

Cut-off frequency

The cut-off frequency is the point at which the filter starts to have effect. So if you have a low pass filter and set the cut-off frequency to 8kHz, the sound spectrum above 8kHz will be progressively reduced. However, if you are using a high pass filter, sounds below the cut-off frequency (8kHz) will be reduced.

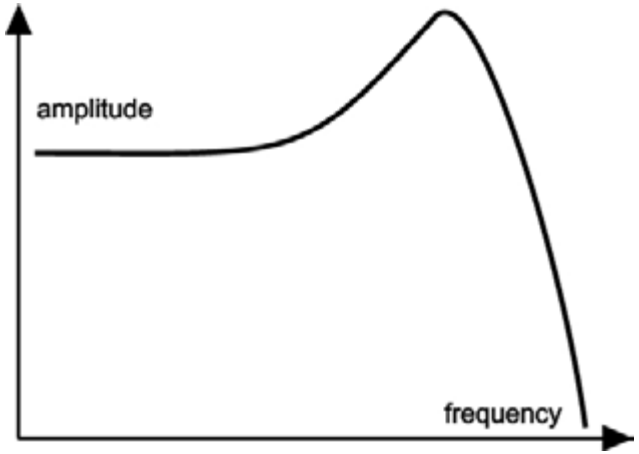


figure 20: a low pass filter with a resonant peak

Resonance

Resonance adds some bite to a filter. It works by boosting the sound spectrum around the filter cut-off frequency. Used in moderation, the effect is subtle and can make a sound appear brighter and/or slightly thinner (or less fat if you prefer). When used to the extreme the effect is noticeable – most dance records use filter sweeps with high levels of resonance.

At very high resonance settings, some of the filters can exhibit quite extreme behavior – if you're looking for an example, turn on the resonance boost in Z3TA+

and push the resonance right up. Make sure you turn down the output before you try this or you are likely to burn your ears off.

Wusikstation and Z3TA+ have limiters on their filters because of the extreme nature of sounds that can be produced at high filter resonance settings.

Filter slopes

A low pass filter progressively reduces the volume of a sound above the cut-off point. The rate at which the sound wave is reduced above the cut-off frequency is determined by the slope of the filter.

If you have a 6dB/octave filter, it will reduce the level of the sound source by 6dB at one octave above the cut-off point, 12dB at two octaves above the cut-off frequency, 18dB at three octaves above the cut-off.

The effect of a 6dB filter on a sound wave is quite subtle.

A 12dB/octave filter (sometimes called a 2 pole filter) will reduce the level of the sound wave's volume by 12dB for each octave for each octave above the cut-off point. This type of filter was often used in some of the Japanese synthesizers from the 1980s.

A 24dB/octave filter (sometimes called a 4 pole filter) will reduce the level of the sound wave's volume by 24dB for each octave above the cut-off point. This type of filter was often used in some of the classic US analogue synthesizers.

You will see some different filter slopes in figure 21. This is an image drawn by hand and so is not to scale, however, you can see the differing result of the filter slopes.

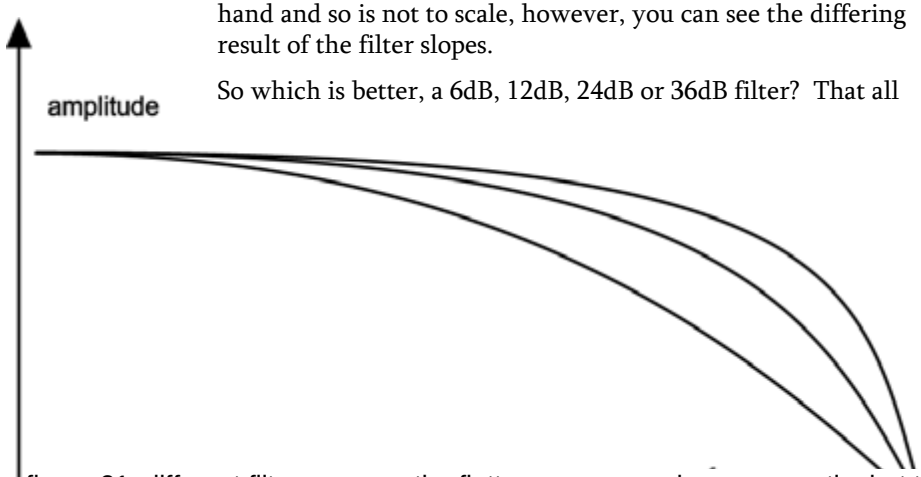


figure 21: different filter curves – the flatter curves may be more gentle, but the affect a large portion of the sonic spectrum

So which is better, a 6dB, 12dB, 24dB or 36dB filter? That all depends. Take a comparison between a 12dB and 24dB filter – you need to think of the context in which the sound will be used. On the face of it, the 24dB filter might get faster results. However, remember that a 12dB filter has to work twice as hard and will affect a much greater proportion of the signal to achieve the same reduction that a 24dB filter will make at any given frequency, so the 12dB filter may be more appropriate.

There are no hard and fast rules, but you may find that, for instance, a 24dB filter could be a better starting point when designing bass sounds, whereas a 12dB filter produces better results with pads or sample based sounds. Then again, you're going have your own considerations and will need to decide whether you're trying to shape the sound or whether you're getting surgical, so you may find another combination that works for you.

Different synthesizers – different filters

Different filter designs operate in different ways and give different results – it is important that you have some idea of what a filter is trying to do to the signal. For instance, some filters (not shown here) will allow you some control over the bandwidth of the resonance – this is quite a fine change, but definitely has an affect on the sound's character.

If you get two synthesizers each with a 12dB/octave filter, they won't necessarily sound the same. They might initially, but if you start apply envelopes and other modulation sources to the filters, they are likely to work in different ways.

Think of filter labels a bit like the badges on the back of a car. "2.0" on the back of a car may indicate a two liter engine. Two cars with "2 liter" engines may in practice have engine capacities of 1998 cc and 2021 cc – its all a matter of design and branding.

You wouldn't expect two cars from different manufacturers with a 2.0 badge on the back to have the same performance. It wouldn't be unreasonable to expect that if you have two cars in the same manufacturer's range and one has a 2.0 badge and the other a 3.0 that the bigger engine will accelerate more quickly. However, you would not necessarily expect it to accelerate 50% faster.

Use the same logic with filters: the numbers are labels which don't necessarily translate between different synths – use them as an indication of the differences within the individual synthesizer. At the end of the day, if you want to know how the filters sound and feel, you're just going to have to take the synth out for a test drive.

In practice you will find filters can be arranged in parallel (for instance Rhino) or in series (Z3TA+ and Wusikstation both give the option to arrange filters in parallel or series). When filters are run in series, the output of each filter is fed into the next filter giving a cumulative effect: two 12dB filters in series gives the effect of a 24dB filter.

However, remember that this book is about making music, not mathematics. By design, different filter slopes will achieve their results in different ways. Some will work in a pure linear, mathematical way. Some will cut the sound more sharply after the cut-off point. Some will cut less at the cut-off point, but will cut more (in an inverse exponential curve) as the sound gets higher above the cut-off point. Others will try to emulate classic gear – and some will claim to emulate classic gear knowing that virtually no one has the gear to make an A/B comparison.

Using filters in practice

One of the key issues for making a filter sound realistic is modulation which is discussed in greater detail in chapters 5 and 6. A filter which is not modulated will be, and will usually sound, static. The human ear is attuned to natural instruments which have a constantly shifting character and will expect similar characteristics in synthesized sound.

Controlling filter cut-off

When using a filter to shape tone, the main (but not the only) modulation controls are:

- envelopes – these provide an ideal control source to shift the filter over time, and
- velocity – this allows a filter to mimic the properties of natural instruments so that the sound gets brighter with more expressive playing.

You will find that velocity and envelopes are often used in combination when controlling filters.

Filter as a volume control

The filter also acts as a volume control – the more a filter closes, the greater its affect on attenuating the amplitude of the sound. You can use this to your advantage by setting an envelope to completely close a filter. If we return to our example of a piano note – the note should have a finite length, but an ADSR envelope cannot mimic this behavior. However, a filter controlled by another ADSR envelope could cut the volume completely. Therefore, the combination of two envelopes, one controlling the volume and the other controlling the filter could create a more accurate representation of an acoustic sound.

Key tracking and filters

You may want to use key tracking when setting the filter to mimic what happens naturally. If you are designing a patch to resemble the behavior of an acoustic instrument, a single cut-off point may be unnatural: the high notes would be too dull and the low notes to bright. To remedy this you could use key tracking with the filter: this would open the filter at higher frequencies and close it at lower frequencies giving a more natural response. If you want you can have an unnatural response where the filter closes *more* at higher pitches.

Controlling resonance

Resonance is generally used in three main ways:

- to give a brighter sound
- to give a "thinner" sounds – this is partly a result of making the sound brighter, and
- as an effect, whether that be making the filter scream or giving sounds a really squelchy character.

Controlling resonance with an envelope will allow for more subtle nuances to be introduced into a sound.

Filters: some sonic examples

We have talked about filters. Let's build a few patches and listen to the effect of filters in practice.

All of the patches in these first examples are built using Z3TA+. Similar results can be obtained with any of the other featured synthesizers (except for the patches using the formant filter).

These examples are built around a single sawtooth wave and use some of the filters that are available in Z3TA+ – there are more filter options available, both in Z3TA+ and from the other synthesizers featured in this book. Some of the techniques used to build these patches have yet to be discussed, but will be addressed later in the book.

The unfiltered waveform

raw saw

This isn't difficult …. **raw saw** gives you the sound of a sawtooth wave without any filtering. It is included for comparison with the filtered sounds – it is not the most engaging sound you will ever hear.

Low pass filtering

low pass

Low pass takes the sawtooth wave and runs it through a 24dB low pass filter. As you can hear, the sound is much duller and much quieter than **raw saw**. The volume reduction is not surprising – a significant proportion of the waveform has been removed.

12dB low pass sweep

12dB low pass sweep puts a sawtooth wave through a 12dB low pass filter and then sweeps the cut-off frequency of the filter. A filter "sweep" is another way of saying "adjusts the cut-off frequency from one extreme (for instance fully open) to another extreme (for instance closed)". Filter sweeps can be restricted to a much narrower

range and sweeps can also follow rhythmic patterns (see **stuttering low pass filter** below).

At the start of the note, the filter is open and so its effect will not be heard. As the note is held the filter will close down until it reaches a point where it is slightly open and the tone becomes constant. Some resonance has been added to the filter to make the effect of the sweep a bit more extreme (and noticeable), so please ensure the volume is not too loud when you first try this patch.

As you listen to this patch, you will hear the sound is quite bright to start with. As the filter starts to close you will hear the effect of the resonance. Towards the end of the sweep the effect is quite extreme – a few steps further and this patch could be distorting or screaming (turn down your speakers, push up the resonance and listen).

36dB low pass sweep

This patch adopts exactly the same parameters as **12dB low pass sweep** with one difference – the filter is replaced with a 36dB low pass filter.

If you compare **36dB low pass sweep** and **12dB low pass sweep**, you will hear two main differences. First, with the 36dB filter the sweep is far smoother and sounds much more controlled: the sound does not come close to screaming or distorting. This smoothness will benefit some patches, but not all – there will always be times when you really want to create a screaming lead sound. Perhaps then you call up a 12dB filter.

Secondly, you will notice that the sustain sound after the sweep ends is both quieter and more dull. This is a result of using a filter with a sharper slope.

One other matter I want to draw to your attention. Listen to the **12dB low pass sweep** and then **36dB low pass sweep** – is the effect of the filter in the second patch three times that of the filter in the first? In other words, does a 36dB/octave filter SOUND three times as effective as the 12dB/octave filter? Remember, you program with your ears not your eyes. Don't let the specifications fool you about the sonic results.

stuttering low pass filter

Stuttering low pass filter takes a slightly different approach to filter sweeping and uses a low frequency oscillator to modulate the cut-off frequency of the filter where the frequency of the LFO is linked to the tempo of the track, thereby creating a rhythmic effect. The output level of the LFO is randomized, so the cut-off frequency is randomized too. The effect of these two factors gives the stuttering effect.

High pass filtering

high pass

High pass takes the sawtooth wave and runs it through a 24dB high pass filter. As you can hear, the sound becomes much thinner and much quieter. While this sound is much thinner than the unfiltered wave, tonally it sounds much closer to the sound of the unfiltered sawtooth wave (listen to **raw saw**) than when the wave is run through the low pass filter. This is not surprising: much of the information about a sound's character is contained in its higher frequencies.

Again, the volume of the sound is considerably reduced when compared with the unfiltered sound because a large chunk of the sound spectrum has been removed.

high pass sweep

With the **high pass sweep**, the sawtooth wave runs through a 24dB high pass filter. At the start of the note, the cut-off frequency is low, so as this is a high pass filter, the effect of the filter will not be heard. As the note is held, the filter sweeps from a low cut-off frequency to a higher cut-off frequency. The sound becomes thinner and quieter.

At the end of the sweep, a constant, quite fizzy tone is heard. If the sweep continued, then all of the frequencies would be filtered and no sound would be heard.

Band pass filtering

combo low + high pass

Combo low + high pass is constructed by running two filters in series (ie the output from filter one is fed directly into filter two). The first is a low pass filter and the second is a high pass filter. If you compare this patch with **band pass**, which uses a band pass filter to create its sound, you will hear that the results are very similar.

band pass

A band pass filter cuts high and low frequency elements simultaneously and so, as would be expected, with the **band pass** patch the sound becomes:

- duller
- thinner, and
- even more quiet than when a high pass or low pass filter is used on its own.

band pass sweep

With **band pass sweep**, the band pass filter's cut-off frequency is swept from its highest setting to its lowest setting. At both extremes, no sound is heard due to the filtering effectively removing all elements of the sound spectrum.

Near the top extreme, the sound is reminiscent of a high pass filter with a high cut-off frequency. Whereas, towards the lower extreme the sound is reminiscent of a low pass filter with a low cut-off frequency. In between these two extremes you get the effect of the band pass filter working at different frequencies.

Notch filtering

notch

It is often quite difficult to hear when a notch filter is being used. If you compare this patch, **notch**, to **raw saw** you will hear that there is a slight thinning of the sound in the mid range. If you played both patches in the context of a track you would be hard pressed to hear the difference.

notch sweep

As may be expected, **notch sweep** is constructed by sweeping the cut-off frequency of a notch filter. With this patch you can hear the effect of a filter sweep but without too much energy being robbed from the sound. Where notch filtering is quite a dull and subtle effect, notch filter sweeping can be a far more useful programming technique.

Formant filtering

formant

The format filter in Z3TA+ effectively has five positions, corresponding to the vowels a, e, i, o and u. This patch, **formant**, has been constructed with the format filter set to the "e" position. It is a testament to the power of the filter that a sawtooth wave can at least bear a passing resemblance to the "e" sound.

formant sweep

With **formant sweep**, the filter steps through the five vowels – a, e, i, o and u – in turn. This is a long, *long* way from a talking synthesizer. However, the tonal variations are interesting.

Using filters to create patches

We will end this chapter with three simple, but usable patches built using Vanguard. These patches are included as sonic examples of how a filter can be used. As they employ techniques that have yet to be discussed, I will not explain the construction of these patches in great detail (which may make it harder to replicate the sounds if you haven't purchased the patches).

bass + stab

This first patch, **bass + stab**, is a straightforward bass patch. The sound is quite full and rich. Having called up the oscillators and closed down the filter (giving quite a dull sound), I then made two main changes to make this patch more playable:

First, the filter is (slightly) velocity sensitive – hit the keys harder and the sound gets brighter. The filter is also controlled by the volume envelope and so the note is (slightly) brighter in the initial stages.

Second, in the higher keyboard regions this patch sounded too bright to me so I have applied some key scaling to reduce the brightness of the patch in the higher octaves. The key scaling, in conjunction with the filter, works to almost completely cut the sound at higher octaves. With the key scaling the bass patch becomes quite usable as a stab type sound.

gentle stab

Keeping with the stab theme, **gentle stab**, uses key scaling but with this patch the sound gets brighter as the patch is played in the higher octaves. This behavior may be more intuitive as it mimics the behavior of natural instruments.

As with the previous patch, the filter is also controlled by an envelope (to give slightly more bite in the attack phase) and by velocity giving the player real time control over the patch.

There is also another very subtle element to this sound – the resonance of the filter is controlled by the second envelope. The effect of this resonance is to add a touch of brightness to the attack of the note.

resonant bass

Resonance bass takes a slightly less subtle approach to the use of resonance. In this patch the filter is controlled by two sources:

- key tracking – as the patch is played at higher pitches, the filter cut-off is reduced making the upper octaves less bright (which would be consistent with a bass patch), and
- velocity – as this patch is played with greater velocity, the filter opens up to give a brighter sound. This gives the player real-time control over the filter.

The resonance of the filter is controlled by an envelope. When the note is struck this envelope works to increase the resonance to the maximum amount and then cut the resonance. This gives a "squelchy" type sound – the tone of the sound is determined by the filter's cut-off frequency which is controlled by velocity.

You may also notice that in the higher keyboard regions, this patch takes on almost a vocal quality.

Chapter 4: sound sources

We've looked at envelopes and we've looked at filters. Both of these have a fundamental effect on the tone of a sound and in many instances can have a greater effect on the perception of a sound than the actual sound source itself. However, the sound source is still significant.

As a first point, don't get too hung up with how different manufacturers label different wave shapes (whether in the five featured synthesizers or with other synths). When viewed on an oscilloscope most waveforms (especially analogue or analogue-emulating waveforms) do not look like their mathematically generated forms. However, while the vintage waves may not look like the mathematically generated waves, they do usually sound great.

As I say, don't concern yourself with labels, but do concern yourself with how individual waves in different synthesizers sound and how you can use each wave in an appropriate context.

All of the synthesizers featured in this book can produce all of the main wave shapes. However, they may produce the waves in different ways, for instance Wusikstation is a sample based synthesizer and so uses samples whereas Cameleon 5000 is an additive synthesizer (see chapter 9: additive synthesis for further detail) and so can create a sound from first principles.

This chapter only looks at the main sound sources. It does not attempt to look at all (or even most) of the waves that are available from the featured synthesizers.

Basic wave shapes

Let's first look at some of the wave shapes that are common to virtually every synthesizer that has ever been produced.

Sine wave

The sine wave is perhaps the most basic element in a sound. It is the purest form of tone you can have – it consists of the fundamental note and has no overtones. If you run a sine wave through a filter, there are no overtones to filter out – therefore the only effect that a filter would have would be to reduce the volume of the note itself. If you put any sound through a low pass filter, as you take out the harmonics it will tend to sound like a sine wave.

On its own a sine wave can sound quite dull and is not often a first choice for programming. However, as a waveform it is often used to thicken up patches. Where a waveform sounds weak on its own, particularly if it is based on a sampled wave, adding a sine wave can give a depth to a patch and add a roundness/fullness to the sound.

Sine waves are also often added to bass patches to give a sub-sonic, foundation shaking, quality. If you are doing this, please check the patch on full range monitors and take care not to blow your speakers.

A final frequent use for a sine wave is in FM synthesis – it is quite common (indeed, it was originally the only option) to build FM patches solely with sine waves (see chapter 8: frequency modulation synthesis).

As we will discuss later (see chapter 9: additive synthesis), sine waves are the components of all other waves.

Sawtooth wave

The sawtooth may be the closest that we will come to a general purpose wave.

It gives a bright sound which is often used as the basis for brass and string sounds as well as general "fat" synthesizer sounds (such as stabs and basses). It is rare, but not unknown, to hear a raw sawtooth wave: because of the bright quality of the wave it is usually filtered.

While a sawtooth wave is a bright wave, filtering can add warmth and depth to a sound. However, the wave also tends to dominate a broad proportion of the sound spectrum. This may not be a problem, but if your arrangement contains several patches based on sawtooth waves, you may find your mix starts to get muddy.

Square and pulse waves

The square waveform has a hollow quality and is often used to create "woody" or "reedy" tones such as those found in woodwind instruments. It is also frequently used in bass sounds, either on its own or to fatten up a sound often acting as a sub-oscillator (ie a note pitched below the fundamental).

A pulse wave with a value of 50% (ie when both sides of the wave are balanced) is a square wave. A pulse with a 0% width is just noise. Some synthesizers separate the square and pulse others provide one wave and a facility for pulse width modulation (which is discussed below).

Triangle wave

A triangle wave gives a sound that is slightly less reedy or perhaps less sharp than a square wave. If you want to stretch a point, you could alternatively think of the triangle wave as being like a sine wave but somewhat sharper in its output (but please do not try to relate a wave's shape on an oscilloscope to its tone).

A triangle wave is often used as a low frequency oscillator (LFO) waveform.

Noise

You might think that noise is just noise, however, you would be wrong. It comes in different colors – each color depends on the composition of frequencies in the sound

spectrum. I hope you won't think it too much of a cop out if I don't try to use words to describe the sonic differences in great detail. However, in summary, white noise is the brightest, pink noise is slightly less bright and brown noise is duller still.

Complex wave shapes and variations

Other wave forms

We've looked at the basic waves. All of the featured synths have many more waves available. Some offer many variations on a basic theme (for instance Z3TA+ gives you 11 sawtooth variants) and others offer a range of different waves. For all of the featured synthesizers (with the exception of Vanguard) the waveform options are effectively limitless since it is possible to import waveforms.

It would be pointless to try to describe the differences between the waves in words – you really need to spend some time listening to the differences.

You may be wondering why so many waves are offered. There are several reasons:

- to give different tones and shades – the broader the palette, the more colors the musician can paint with (and the more dilemmas for the sound designer)
- efficiency – some wave forms can be achieved by combining two other waves: by making the wave available without having to use two waves, the developer is giving you the advantage of reduced CPU load and the flexibility to keep other wave slots free
- because everyone else does – would you buy a synthesizer which only offered four waveforms (unless it was a very good emulation of a piece of vintage gear)?

However, please do remember that a wide range of waveforms alone does not make a good synthesizer.

Pulse width modulation

Pulse width modulation (often called PWM) is a technique most associated with square and pulse waves. With a square wave, the positive and the negative phases of the wave are balanced. When the pulse width is modulated, this balance changes to give a different shaped wave.

The different waveforms are not simply different shapes on an oscilloscope, but contain different spectral components, hence their different tone. These components are discussed in greater detail in chapter 9: additive synthesis.

PWM can either be static, for instance a 50% wave is modulated to give a 40% wave, or it can be a continuous change (for instance when being modulated by an LFO). As a technique, PWM is generally used for one of two reasons. First, it changes the tone of the waveform. Second when two waves which have been modulated in different ways are combined, there can be a fattening effect (but see "doubling oscillators" below).

Wave shaping

As well as giving a wide range of waveforms, Rhino and Z3TA+ also allow you to further shape the waves. The synthesizers take different approaches but the tools both work to distort a raw waveform's shape. The effect of these wave shaping devices can be subtle but can also be more extreme producing FM-like tones (without using FM) and distorted sounds.

With Rhino, if you:

- put the shaper line horizontally, the resulting wave will be a flat line
- put the shaper line diagonally – bottom left to top right – the shaper will have no effect, however
- put the shaper line diagonally – top left to bottom right – the shaper will invert the wave.

By contrast, Z3TA+ performs a variety of transformations on its waves: there are many more options, but it does feel more controllable. An explanation of the transformations is set out in the Z3TA+ handbook and is not repeated here.

Although it doesn't add anything to the sound, one interesting feature of Z3TA+'s shaper is the waveform display (see figure22) which shows the transformations in real time.

figure 22: the effect of Z3TA+'s shaper on a vintage sawtooth wave

Combining sounds

You can't have failed to notice that all of the synthesizers featured in this book allow you use more than one waveform at a time and yet so far this book has largely talked about the possibilities offered by using single waveforms.

The reason I have done this is simple – it is easier to explain the operation of single waves. I have already shied away from describing many of the waveforms. If it is hard to find words to describe sounds, it is harder to describe the variation between

the many permutations of combinations of oscillators working together. Therefore for the rest of this book there will be a greater reliance on sonic examples than has been the case so far.

There are many reasons for sound combining (including to create new tones, to augment a weak tone or simply to get a smoother sound) and there are many ways that sounds are combined, for instance two similar sounds could be doubled or two different sounds can be layered to create a wholly new sound.

Whatever the reason, sounds created with multiple oscillators will generally fill more of the sound spectrum. Accordingly, care needs to be taken to ensure that these sounds do not come to dominate a mix (unless that is the intention).

Let's now look at some ways that sounds can be combined in a bit more detail.

Doubling oscillators

The most simple combination of oscillators is to take one oscillator and clone its settings to a second oscillator. Simply cloning the oscillator may not do much more than increase the volume. However, the addition of a second oscillator does give many more sonic opportunities.

Oscillator phase

When we talk about the "phase" of an oscillator we generally mean the position in a wave's cycle see figure 23. With one oscillator, the phase matters little, however with two oscillators the effect of phase can be dramatic.

If you have two waveforms that are totally in phase then you will get reinforcement of the signal. Reinforcement is perceived as an increase in volume. If your waves are out of phase then you will get cancellation. The effect of the cancellation depends on the individual waves: if you have two sine waves that are 180 degrees out of phase, then you will get total cancellation.

Cancellation caused by putting waves out of phase is generally perceived as a change in tone, usually making the sound thinner and sharper. This may be a great result if you are after a plucked sound or a more metallic sound – it may be less impressive if you are after a really fat sound.

If we look at Z3TA+ it gives us several options to play with the phase which may be a useful illustration of the sonic possibilities of phase. In Z3TA+ all of these changes can be applied on a per oscillator basis (so each of the six oscillators can have different combinations of options):

- The phase of a note can be synchronized with the start of a note (ie the key strike). This means that every note from that oscillator will start from the same position in its phase – the starting phase position will be determined by the phase control for that oscillator.

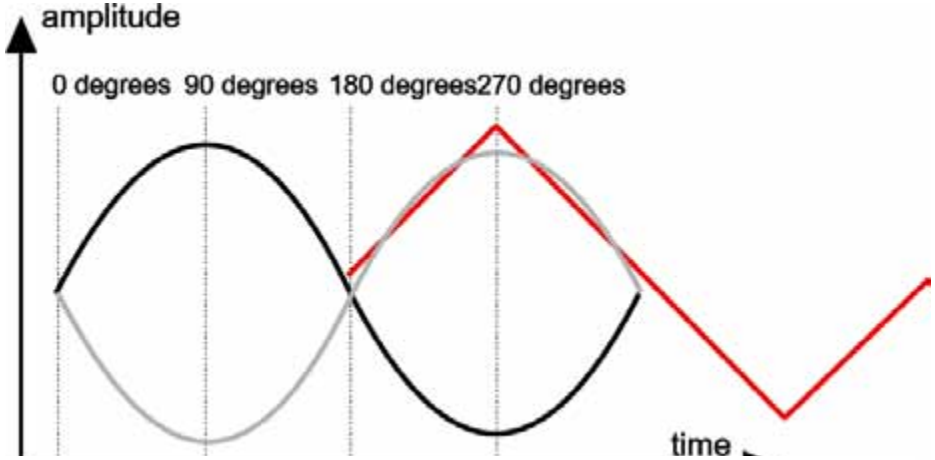


figure 23: the phase of an oscillator – the gray sine wave has reversed polarity and the red triangle wave is 180 degrees out of phase

- The phase of each note can be synchronized but the polarity reversed – mix a wave with positive polarity and a wave with negative polarity, both with the same phase, and there will be total cancellation.
- Alternatively the phase of each note (again on a per oscillator basis) can be allowed to run freely. This means that each new note has a different phase and each note from a different oscillator can have a different phase. This can result in some interesting effects (or annoying effects, depending on your perspective).
- As mentioned earlier, the phase of each note can be shifted – this can give consistency of tone when using two oscillators each of which is key strike phase synchronized.

As you will hear, the effect of detuning an oscillator can lessen the more extreme effects of phase shifts.

One thing to note if you are playing with the phase in Z3TA+ – it does not make phase changes in real time. Instead, the new phase is introduced with each newly triggered note.

Oscillator detuning

Another common technique used with doubled oscillators is to detune one of the oscillators (or to detune each of them but in different directions). Subtle detuning can give a natural chorusing effect which is perfect for creating fullness/roundness/smoothness or just adding fatness to the sound.

There is a balance to be struck when using this technique – if the detuning goes too far then the resulting sound can become flabby and/or out of tune. This may be a good or bad thing, depending on the effect you're after.

Rhino, Z3TA+ and Wusikstation all allow you to fine tune individual oscillators. Cameleon 5000 does not allow for detuning of individual oscillators, however, each partial can be individually detuned (detune them all and you *might* get a similar effect). Vanguard doesn't allow this fine detuning as such (but see below for how it addresses this issue).

Multi-oscillators

So far we have looked at doubling oscillators. However, there is no reason why you shouldn't triple or quadruple oscillators and separately detune each oscillator. Three of the synthesizers offer six oscillator slots which would allow you to build a patch with six slightly detuned oscillators.

Or you could use one of the multi-oscillators that are available in Z3TA+ and Vanguard and in one step save yourself a lot of time and perhaps get a better sound.

While different in their implementation, both Z3TA+ and Vanguard have the same essential characteristics. Instead of there being one wave, with the multi-oscillators up to eight waves will be called up (in one slot). These waves will be spread across the stereo spectrum and slightly detuned to give a very big sound from one oscillator. For Z3TA+ this is called multi-mode. However, I do admire Michael Kleps' simple honesty in labeling the knob to control multi-mode in Vanguard "fat".

Both machines allow control over the detuning of the waves. Z3TA+ also allows some control over the number of waves and the phase of the waves.

Although Wusikstation and Rhino don't offer a multi-oscillator option, both are capable of importing waves so you could load some samples of multi-oscillators and create the same effect. Indeed, taking this further (and I haven't tried this out, so if it doesn't work, I apologize) you could use the multi-oscillator in Z3TA+ to create a multi-oscillator wave which you then render and import back into Z3TA+ and again engage the multi-oscillator feature. My hunch is that this wouldn't sound as good as I hope.

Layering oscillators

There is a fine difference between layered and doubled oscillators – I am only using different terms so that I can be clear about the different concepts.

With layered oscillators, you again take two (or more) oscillators acting together. However, the oscillators are different, so for instance, you may combine a saw and a square wave. Alternatively you may combine two square waves, one being an octave higher than the other.

The combinations may or may not be detuned – that is all a matter of taste as we shall hear later in this chapter. However, it is quite common to change the octave of a layered oscillator so that it can add some high end bite or low end punch.

For a simple practical example of layered oscillators, take a sawtooth wave and a square wave. On their own, both waves have a certain sound: both are quite bright and in certain circumstances this may mean that some richness and some depth of tone is lacking. You could double either of these oscillators to get a richer tone. Alternatively, you could layer these two oscillators.

Layering gives a new tone that is neither sawtooth nor square – the layered sound still has the brightness of the two components, but perhaps more weight. If you then want to thicken things up considerably, drop the square by an octave and engage the multi-mode oscillator for the sawtooth and put the result through a filter. This is a very quick and dirty way to get a fat sound.

Layering sounds

If there was a fine difference between layered oscillators and doubled oscillators, the difference between layered oscillators and layered sounds is even more tenuous.

With layered oscillators you take two (or more) oscillators and route them together through the same modifiers and controls (filters, envelopes etc) – in effect, the two oscillators are working together as one, creating a whole new sound.

With layered sounds, each oscillator has its own filter and its own envelopes etc. The advantage of a layered sound (when each layer has a similar sound) over a layered (or doubled) oscillator is that you can get a thicker, fuller sound which is more controllable. However, one of the advantages of layered oscillators is speed of working for nearly the same sound.

As with doubling and layered oscillators, layered sounds can be detuned.

Whether you can layer the sounds or layer the oscillators depends on the particular synthesizer's architecture. For instance with Vanguard you can only layer (or double) the oscillators but with Wusikstation you can only layer the sounds. Rhino and Z3TA+ allow you to layer both (or either) sounds and oscillators. Neither option is right or wrong (or better or worse) – its just how the synthesizers have been designed.

Layering sounds is more of a programming technique rather than being a separate sound source and so there are no sonic examples with this chapter. However, the technique will be demonstrated in chapter 12: building patches. Also, as you will see in chapter 7: FM synthesis, layering is one of the key building blocks for FM sounds.

You may come across synthesizers that offer "vector" synthesis (for instance Wusikstation). Vector synthesis is another way of layering and fading between sounds. With the original hardware vector synthesizers there was a joystick that gave the sound designer control over the mix of a number of sounds (usually four).

The days of hardware are largely gone, but some software synthesizers (for instance Wusikstation and Z3TA+) still allow this kind of control to mix sounds together.

Syncing oscillators

We've already discussed the phase of oscillators. There is a further step you can take (in Z3TA+) and hard synchronize two oscillators together. When hard sync is engaged, the slave oscillator restarts its phase each time the master oscillator starts its phase.

If the two oscillators are pitched at the same level, the effect is comparatively mild (perhaps giving some cancellation). If the two oscillators are pitched differently then one oscillator will complete its phase before the other – this means that the slave will be part way through its phase when it is retriggered. This can result in a very hard sound that is often used for creating cutting lead sounds.

Combining sounds: creating new tones

That's enough theory, let's look at some practical patches. These patches have all been built with Z3TA+ since this synth has the most easily accessible flexibility for the points I am trying to demonstrate and if we stick with one machine we can make a consistent comparison.

We're going to look at – or rather listen to – the effect of combining two sounds. This is much like the musical equivalent of paint color charts when you want to redecorate your home. On their own each color looks fine – it is only when you put the color next to a similar color that you can see the similarities and the differences. And as with color charts, when you have a range of tones, there is a dilemma about choosing the right one.

In listening to these combinations, we're trying to achieve several things:

- first to hear the added weight of combining two oscillators
- second to hear a completely new tone, and
- third to hear whether the component parts can be separately identified.

We're also going to listen to how the combinations react to filtering.

The comments in this section relate solely to Z3TA+ – the sounds which are created are based on the interaction of Z3TA+'s oscillators and filters. While many comments may be equally applicable to many other synthesizers, these views should not be regarded as "rules" and the interaction of the same elements in different synthesizers may give differing results.

Sawtooth waves – same octave

single saw and single saw filtered

These two patches are included as a reference point for later comparisons.

Single saw gives the sound of a single sawtooth waveform. **Single saw filtered** gives a single sawtooth wave filtered through a 24dB/octave filter which is velocity sensitive (as are all of the filtered patches in this section).

On its own, the sawtooth wave sounds rich and bright. Through a filter, there is a progressive cut in the brightness of the wave.

two saw running free

With **two saw running free** you can hear two sawtooth waves running free – in other words, the phase of each of the two waves is arbitrary. This means the waves could both be in sync or could be totally out of sync with reference to each other.

If you strike the same note repeatedly, you will hear different tones caused by the differing phase cancellation of the two oscillators.

two saw phase sync

In **two saw phase sync**, the phase of each of the oscillators is synchronized to the key strike. Because of the synchronization, this patch has a consistency that is not heard in **two saw running free**: each time you strike the key you will get exactly the same note. As both oscillators are running exactly in sync the effect is to increase the volume output without changing the tone.

two saw 90 phase

Two saw 90 phase has the same set up as **two saw phase sync**, however, the phase of oscillator two is 90 degrees out. This means that there will be some reinforcement and some cancellation which together result in a different tone. You can hear that this tone is much thinner and has more of a plucked quality – perhaps the sort of sound that could be used as the basis for a harpsichord type patch.

As the phases of the two waves are synchronized there will be a consistent tone to this note. It would have been possible to have two free-running oscillators and to put one 90 degrees out of phase by reference to the other. However, that would have been a fairly pointless exercise as the waves would have been free-running in the first place (a random phase that is 90 degrees different is still a random phase).

two saw 180 phase

Two saw 180 phase takes the previous patch and pushes the phase slider in the second oscillator to 180 degrees. The effect – especially in the last few degrees – is radical. A thin plucked tone loses even more of its bass element to end up sounding as if the patch has been raised by an octave. Indeed, if you drop this patch by an

octave and compare it with **two saw phase sync** you will hear similarities between the sounds.

two saw inverted 1 phase

If we take to sawtooth waves and invert them without any phase differential there would be total cancellation. I guess that you know what silence sounds like and so wouldn't appreciate a demonstration. Instead with **two saw inverted 1 phase**, the second wave is inverted by reference to the first and its phase is moved by one degree to ensure there isn't total cancellation.

You can hear that there is still considerable cancellation and the sound that is audible has little tone and a lot of noise. If you move the phase fader the tone will shift, gradually becoming richer. If you move the fader to 180 degrees you will hear another shift in tone with the sound of a square wave becoming audible. This is not surprising – if you add two sawtooth waves together with one having an inverted polarity and being 180 degrees out of phase, the result will be a square wave.

two saw detune running

Two saw detune running takes two sawtooth waves, both with the same polarity, but neither synchronized, and slightly detunes the first oscillator.

As you can hear, especially if you compare the patch with **two saw running free**, the detuning results in a much warmer tone which is constantly shifting. Where **two saw running free** may not have provided many usable tones, this patch could find many uses.

While there have been no changes to the envelope settings, this patch almost appears to fade in (especially when compared with the next patch).

two saw detune phase sync

Two saw detune phase sync takes the previous patch and synchronizes the phase of the oscillators when a key is struck. This gives three main differences when compared with **two saw detune running**:

- the tone of this patch is different having a slightly brighter quality (the comparison obviously depends on how closely synchronized the oscillators are at any stage in **two saw detune running**)
- the texture of this patch is different – with the phase start synchronized, the resulting sound has a far more aggressive quality, and
- by comparison with **two saw detune running**, there is a real "spit" to this patch in the attack phase.

The differences between **two saw detune running** and **two saw detune phase sync** are significant when building patches. The free-running patch may give a sound that is far more suited to a soft pad where the phase synchronized sound may have more of a use in a stab type sound.

two saw big detune phase sync

Developing the previous patch further, **two saw big detune phase sync** takes **two saw detune phase sync** and shifts the tuning of oscillator one putting it 50 cents out of tune with oscillator two.

The result, to my ears at least, is a not very usable tone. However, even if we can agree that this sound is not particularly usable, I think we could probably find a range of opinions disagreeing about where the useful tone ends and the "out of tune" tone begins.

two saw detune 90 phase and two saw detune 180 phase sync

In the same way that **two saw 90 phase** and **two saw 180 phase** have a thinner sound than **two saw phase sync**, these two patches have a thinner sound than **two saw detune phase sync**, but they obviously have a fuller, richer sound than their comparators with equally tuned oscillators. There is not quite such a marked tonal difference when the waves are 180 degrees out of phase, however, if you adjust the tuning, as the waves get closer in pitch the thinning effect becomes far more pronounced.

You can also hear that as the phase difference gets closer to 180 degrees, some of the edge gets taken off the attack.

two saw detune inverted

If you simply invert the polarity of a wave and then play it with a normal polarity version of the same wave, you will get total cancellation. This is not the case when two opposite polarity, slightly detuned waves are played.

However, you do get quite a thin sound and irrespective of the envelope settings, the sound will fade in slowly – if you adjust the fine tuning, the closer the two oscillators are pitched together, the longer the fade in time.

saw multi free filter

Taking a slightly different approach, with **saw multi free filter** we have:

- a single oscillator but with multi-mode called into action, in other words you are not listing to one oscillator but many, each slightly detuned
- the many waves created by the multi-mode oscillator do not have their phases in sync, and
- the output then passes through a 24dB/octave low pass filter which is controllable with velocity.

If you compare the sound of this patch with **two saw running free** you will hear a very different tone – this patch is much richer and doesn't have the sharpness associated with phase cancellation which is present in the two saw patch (although there will be phase cancellation in this patch).

If you experiment with the velocity you will also hear that the filter will smoothly control the tone of the oscillator.

2 x saw multi detuned

Building on this, **2 x saw multi detuned** uses two multi-mode sawtooth waves – one slightly detuned and both with the waves synchronized to the key strike. No filtering is used in this patch.

You will notice several different characteristics to this sound:

- most immediately, there is a real spit to this patch – that is a function of having all of the waves synchronized to the key strike, and
- second the tone is much harder (even though there are two oscillators and therefore double the number of waves. There is also (slightly) less movement in the tone.

To my mind there may not always be much advantage to using two multi-mode oscillators instead of one. I'm not suggesting you should not use two multi-mode oscillators, just that the tonal differences may be quite subtle. However, it is less subtle if when you use two oscillators you pan them in different directions.

So what?

We've listened to some sawtooth waves in various combinations. What have we learned from this? With sawtooth waves, there seem to be a few important principles:

- two, slightly detuned oscillators give a bigger sound than one
- keeping two oscillators at the same pitch gives a sharper sound, especially when the two waves' phases are synchronized
- synchronising the phase starts gives a much harder quality to the sound
- synchronising the start of two waves and then starting their phases at different stages in the cycle generally gives a thinner sound
- too much difference in the tuning and the sound starts to lose focus and "sounds" detuned
- 1 sawtooth wave + 1 sawtooth wave = 1 square wave (provided you understand phase and polarity).

Square waves – same octave

Let's look at some square waves and see if the principles are similar to those we have found for sawtooth waves.

single squ and single squ filtered

Like their single saw equivalents, these patches are here for comparison purposes.

Single squ gives the sound of a single square waveform. **Single squ filtered** gives a single square wave filtered through a 24dB/octave filter which is velocity sensitive.

On its own, the sawtooth wave sounds bright with a certain sharpness. However, when listening to this sound through a filter there is not much of a progressive change to the square wave's sound – it is either bright or dull: there is little in between.

two squ running free

Because square waves are either at their positive maximum or at their negative maximum, phase cancellation has quite a stark effect. With **two squ running free**, when the two waves are not in phase, the sound can take on an FM-like, almost metallic type quality.

two squ phase sync

In **two squ phase sync**, the phase of each of the oscillators is synchronized to the key strike. Because of the synchronization, this patch has a consistency that is not present in **two squ running free**: each time you strike the key you will get exactly the same note. As both oscillators are running exactly in sync the effect is to increase the volume output without changing the tone.

two squ 90 phase

With **two squ 90 phase**, the phase of the waves is synchronized to the key strike, but the phase of the second wave is 90 degrees behind the first wave. While there is a different tone from two squ phase sync – a thinning of the sound – there isn't the same difference in tone as occurred in the sawtooth version of this patch.

two squ 179 phase

If a square wave is added to another square wave which is 180 degrees out of phase, it has the same effect as mixing two square waves which have opposing polarities.

Again adopting the logic that everyone knows what silence sounds like, this wave mixes two square waves the second of which is 179 degrees out of phase. The result is a very thin and noisy sound: there is also a lot of cancellation going on so the sound is very quiet.

If you check out some of the different tones when different phases are selected, you will notice that the tone does not vary as much as it does if you do a similar exercise with sawtooth waves. While the sound does get thinner, its essential tone remains.

two squ detune running

Two squ detune running takes two square waves with free-running phase and slightly detunes one. The sound produced by this patch is a square wave with chorusing.

two squ detune phase sync

Two squ detune phase sync builds on the last patch and synchronizes the phase of both oscillators to the striking of the key. The note has a natural chorusing effect, but the sound is much harder and the attack of the note is far more pronounced.

two squ big detune phase sync

When the pitch difference between the two phase synchronized oscillators is increased to 50 cents, the chorusing effect becomes unpleasant and is perceived as dissonance. If you add a slow LFO modulating the pitch of both oscillators simultaneously you might be able to use this tone to create a siren effect – if you push the mod wheel up a bit you will hear that this effect has been programmed.

two squ detune 90 phase

Unsurprisingly, **two squ detune 90 phase** sounds very much like **two squ phase sync** but with some added natural chorusing effect. And like its non-detuned counterpart, the phase difference has little effect on the tone, in fact, with the detuning it is even harder to tell the tonal difference between this patch and **two squ detune phase sync**. If you do a direct A/B comparison, you will hear the difference. I suspect that within the context of a patch or mixed in a track the difference would not be noticed.

two squ detune inverted

If you take two waves both having their phase key synchronized but with their polarity inverted, you would expect there to be total cancellation. However, in this patch one wave is slightly detuned and so there isn't complete cancellation. Instead the attack portion of the wave sounds as if the attack has been slowed so that the patch fades in.

square multi free filter

As we did with the sawtooth waves, **square multi free filter** is based on:

- a single oscillator with multi–mode engaged
- all of the waves created by the multi–mode oscillator having their phase in sync, and
- the output then passing through a 24dB/octave low pass filter which is controllable with velocity.

To my mind a square wave on its own gives a very sharp sound. However, when multi-mode is engaged, the quality of the sound changes completely, almost giving a vocal type quality to the sound: put the wave through the formant filter to hear what I mean.

What I find particularly interesting about this patch is how the multi-mode wave reacts to the filter. Generally I find that with a square wave the filter is hard to use – the filter either has too little effect or it has too much effect (and even then you seem to get the worst of both worlds – a dull characterless tone that is still too angular).

I much prefer how the multi-mode square wave reacts to a filter: there is a progressive (and musically useful) change in the tone as the filter closes. In other words, the combination works just as you would hope.

2 x squ multi detuned

2 x squ multi detuned uses two multi–mode square waves – one slightly detuned and both with the waves synchronized to the key strike. No filtering is used in this patch.

As might be expected the tone of this patch is much harder and there is also less movement in the tone. However, if you do call up a filter, it works as well as for a single multi-mode square wave oscillator.

Sawtooth waves – octave separation

So far we have looked at the basic tones and only considered the effect of the filter in a few instances. We are now going to look at more complex combinations and listen to how the filter works with these combinations.

To try to keep the permutations within the range of human tolerance, all of the following group of patches have the oscillators' phases synchronized to the key strike.

These patches are all based on two waves tuned to different octave intervals. The octave intervals were chosen as they are the most easily usable and to try to reduce the number of permutations. You should feel free to experiment with all and any interval differences.

two saw +1 octave

This first patch takes two sawtooth waves – the second oscillator is tuned an octave higher than the first.

The effect of the second oscillator on the tone is radical when compared with **two saw phase sync**. You no longer hear the sound of a sawtooth wave, but instead you hear something brighter and much sharper – almost with a plucked tone, perhaps beginning to resemble a harpsichord wave or similar.

As you can hear, adding two bright but not exceptional waves together gives us a completely different tone which has no resemblance to the component parts.

two saw detune +1 octave

Taking the previous patch a step forward, **two saw detune +1 octave** slightly detunes the base oscillator (the effect is the same whichever oscillator is detuned). The detuning gives a chorusing effect as occurred when the oscillators where both at the base pitch. However, this chorusing effect is much more subtle – more in the way of a warming/thickening of the tone.

two saw +1 filter octave

Two saw +1 filter octave, takes **two saw +1 octave** (ie with no detuning) and runs both waves together through a 24dB/octave low pass filter. The filter cut-off frequency is controlled by velocity.

While the waveform of the combined oscillators without filtering sounds nothing like a sawtooth wave, the combined waves react to filtering in the same favorable way that individual sawtooth waves do, in other words there is a constant shaping of the tone and a range of tone colors can be achieved.

You can also hear that with more extreme filtering the resulting sound is close to the sound of a similarly filtered single sawtooth wave.

two saw +2 octaves

When two sawtooth waves are tuned two octaves apart, the sound takes on a brighter and thinner tone still. There is still one distinct unified sound – you do not perceive this to be two waves playing together at the same time.

The difference between this patch and **two saw +1 octave** is noticeable, however it is nowhere near as radical as the difference between **two saw +1 octave** and **two saw phase sync**.

two saw detune +2 octaves

With a slight bit of detuning, the sound of the combined waveform in **two saw detune +2 octaves** is quite clear but the detuning adds a bit of brightness and thickness to the tone. It would be hard to describe this sound as natural chorusing.

If the amount of detuning is increased, the effect is to make the sound take on more of a "drone" and for the two sound sources to become individually identifiable.

two saw +2 filter octaves

As may be expected, when the two sawtooth waves separated by two octaves are put through a filter, the sound again tends to become reminiscent of a filtered single sawtooth wave, but as it is slightly brighter, it also sounds slightly thinner. With less extreme filtering, the sound reacts well to filtering providing a broad range of changing tones.

two saw +3 octaves

With two saw +3 octaves we have raised the second oscillator by another octave so that there is a three octave gap between the oscillators. Two separate tones can now be clearly heard. The sound has a quite distinct bright organ-like quality with a lower tone and a higher tone being heard simultaneously.

two saw detune +3 octaves

As you can hear, when two sawtooth waves which are three octaves apart are combined and one wave slightly detuned, the detuning has little effect on the sound. For this reasons we won't listen to the difference with larger octave intervals as the tonal effects become meaningless.

However, if you detune one oscillator significantly (say by 50 cents) then the effect becomes quite unpleasant.

two saw +3 filter octaves

If you liked the combination of the raw waveforms in **two saw +3 octaves** you are more likely to appreciate the filtered version – if you didn't, you may not like the filtered version. As with all of the sawtooth based combinations, the filter is quite effective and as the filter closes the sound becomes more like a filtered single sawtooth. While the filter is quite effective, it removes the effect of two sounds by filtering out the higher sound.

Since we are using a low pass filter, as the filter closes it has more effect on the higher pitched oscillator. This means that it cuts the higher pitched oscillator disproportionately leaving the regularly pitched oscillator to dominate. If you are modulating a filter with an envelope this could mean that the higher oscillator can be heard during the attack phase with only the regularly pitched oscillator sounding during the sustain phase. This combination could be used to give a regular sawtooth sound more bite. This effect is demonstrated in **harpsi two saws +2** later on.

While a similar principle is true for smaller intervals, the effect is less noticeable.

two saw +4 octaves

With the interval set to four octaves, the resulting sound of **two saw +4 octaves** is quite sharp and buzzy. I'm not sure that you would want to use this waveform combination without filtering.

two saw +4 filter octaves

Many of the principles that applied in two saw +3 filter octaves are relevant here. Again, the raw tone combination may not be very useful, however as an example, with an envelope modulating the filter you may be able to create an organ-type click on at the start of a note.

compare A and compare B

Before we move on I want to highlight some of Z3TA+'s more useful waveforms.

Listen to **compare A** and **compare B** and compare the sounds. Do you notice any difference?

To my mind they are very close although I think **compare B** has a more cohesive tone than **compare A** (in other words you hear only one sound, not two) and also **compare B** has a slightly thinner sound than **compare A**.

However, there is a far more practical difference between these two patches:

- **compare A** is made up of two vintage saw one waves, one pitched three octaves above the second and at a lower volume (35% against 60% for the base oscillator, so there is a different balance from that found in **two saw +3 octaves**)
- **compare B** only uses one wave – octaved saw two.

There are some reasons why you might want to use **compare A**:

- you prefer the tone (unlikely), or
- you want the flexibility to control each wave separately whether in terms of pitch or volume (for instance so that you can mix them in different proportions or change the levels individually over time).

However, there are three reasons why you might want to use **compare B**:

- a single oscillator uses less CPU
- using one oscillator leaves more oscillator slots free for further sonic possibilities, or
- you prefer the tone of the combined wave.

You may find that Z3TA+ doesn't offer the wave combination you want. You can still take advantage of this sort of combined waveform by calling up the waves you want to combine, sampling the output of the combined wave (you can do this by rendering a wave in your host) and then importing the sample back into Z3TA+.

Combining sawtooth waves – a few thoughts

We have now been through a large chunk of sawtooth wave combinations. There are a few thoughts that we can draw from what we have heard:

- combinations of waves pitched at different octaves give radically different tones
- sawtooth waves will still react to a filter when they are combined with other waves, and
- you don't necessarily need FM or additive synthesis to get brighter tones. This is not to say that you can dispense with FM and additive – the combinations shown here are used in a subtractive synthesis context and do not necessarily offer the full range of sonic possibilities which are available with additive synthesis and other sound creation methods not based around tone shaping with a filter.

Square waves – octave separation

We've looked at the sounds produced by combinations of sawtooth waves at different octaves. We're now going to undertake a similar exercise for the square waves and listen to the new tone variations that these combinations offer.

two squ +1 octave

Where two square waves at the same pitch will give quite a hollow, wooden tone, pitching one square wave an octave higher than the other gives the combination a much thinner sound with some of the characteristics of a sawtooth wave but somewhat brighter. If you drop the level of oscillator two to about 25% you will hear the sawtooth sound more clearly.

Tonally the combination of these two square waves gives a new sound – the individual elements cannot be separately identified.

two sq detune +1 octave

If you listen to and compare **two squ +1 octave** and **two sq detune +1 octave** you will hear very little difference – there is perhaps a slight change in the tone, but certainly none of the chorusing effect that we have come to associate with detuning. The effect is nowhere near as dramatic as it is for the corresponding patch constructed with sawtooth waves nor as dramatic as it is when both the square waves are evenly pitched.

At higher octave separations, the detuning effect becomes even less significant so we won't proceed further with examples of combinations of detuned square waves.

two squ +1 filter octave

As I mentioned earlier, I am not a great fan of filtering square waves. The effect of the filter when one square wave is pitched an octave above another gives a better filtering result than when a single square wave is filtered. However, I'm still not knocked out by the results.

With **two squ +1 filter octave** the filter does work to progressively change the tone, however, in the same way that when square waves are filtered the essential tone does not change that much, with certain filter settings you can still get the effect of a waveform that is dull but with too many overtones.

two squ +2 octave

With the second square wave being tuned two octaves above the base oscillator, you start to hear two tones although the effect is not noticeable to a distracting point.

As you would expect, the sound is brighter and even thinner, getting still further away from the tone of a square wave.

two squ +2 filter octave

Two squ +2 filter octave reacts well to a filter (subject to my usual comments about the difficulties when filtering square waves). You do need to exercise some caution when filtering this combination as it has a ready tendency to sound like an organ.

However, this patch (along with **two squ +3 filter octave**) does illustrate an important point for additive waves with dominant harmonic content where you will find similar difficulties if you want to filter the wave.

two squ +3 octave and two squ +4 octave

When one square wave is raised by three or four octaves, two sounds can be clearly heard. The sound is brighter than if than if you listen only to the base oscillator, however, as the two sounds remain distinct, the tone does not get thinner through adding the higher pitched oscillator.

The key difference between **two squ +3 octave** and **two squ +4 octave** is the quality of the higher pitched element – as might be expected, for the higher pitched second wave, the tone of the patch is much sharper.

two squ +3 filter octave and two squ +4 filter octave

As we found with earlier examples using two oscillators with a large interval between their tunings, when passing both oscillators through a low pass filter, as the filter closes it has more effect on the higher pitched oscillator. This means that it cuts the higher pitched oscillator disproportionately leaving the regularly pitched oscillator to dominate.

If you are modulating a filter with an envelope you could get the higher oscillator to sound during the attack phase with only the regularly pitched oscillator being audible during the sustain phase. This combination could be used to simulate the key click of an organ – the quality of the click could be adjusted by changing the tuning of the higher pitched oscillator.

Combining sawtooth and square waves

We've listened to combinations of sawtooth waves and combinations of square waves separately and so now we're going to listen to combinations of the two waves working together. For this group of examples we are going to listen to the effect when the sawtooth and the square wave work together and in particular, the effect when one is pitched higher than the other.

saw squ

The first patch we will listen to takes a sawtooth wave and a square wave both at the same pitch. The effect is to create a wholly new sound which is:

- brighter than a sawtooth alone
- fuller than a square wave alone, and
- more aggressive than either.

So is this the ideal wave combination? Perhaps. Perhaps not. It all depends on how you want to use the combination and whether you agree with my assessment of the sound. However, if you are trying to get a big, dominating sound this combination works well and is a good starting point.

saw squ detune

When the sawtooth wave and the square wave are used in combination but with a slight detune, there is some natural chorusing, but not a significant amount.

saw squ filter

The combination of sawtooth wave and square wave reacts well to being filtered, giving a broad range of tones. However, the tone of the square wave does tend to become clearer as the filter is closed.

saw +1 octave squ

When the sawtooth wave is raised to be an octave higher than the square wave the result is a thicker tone than when the two oscillators are pitched together.

My preference – and you may disagree – is not to mix these two waves in equal measures. When you allow one wave to dominate:

- if the square wave is the quieter, the tone of the sawtooth can predominate with the square wave being used to fatten out the sound, but
- if the sawtooth is the quieter, the square wave can predominate but some of its harshness is counteracted by the sawtooth.

saw squ+1 octave

By contrast, when the square wave is raised an octave above the sawtooth wave, the result is a thinner tone than the tone when the sawtooth is higher. When the volumes of these levels is changed so that one is quieter than the other, the louder wave tends to dominate and the quieter wave adds little to the tone.

saw +1 squ detune and saw squ +1 detune

There is a very slight effect when one of the oscillators is detuned against the other. When the sawtooth wave is an octave higher, the effect is more in the nature of a slight tone change. When the square wave is higher, the effect is more of a chorusing effect.

With higher octave intervals the detuning has progressively less effect and so we won't look at any further examples of detuning in this section.

saw +1 squ filter and saw squ +1 filter

Both of these patches react well to the filter and allow a progressive tone change without either wave coming to dominate. You will notice that as the filter is closed it becomes harder to differentiate between these two sounds.

saw multi +1 squ and saw multi +1 squ filter

For the final patch of the one octave separation group, I have taken a multi-mode sawtooth wave and combined it with a single square wave (set at a lower volume). For the second patch, I have run this combination through a 24dB/octave filter.

For both patches the square wave works to really thicken up the multi-mode sawtooth giving some punch to the patch.

However, the combination of the two waves is not quite a seamless as was the case when neither wave engaged multi-mode (listen to **saw +1 octave squ** and **saw squ +1 octave** for a comparison). In **saw multi +1 squ** and **saw multi +1 squ filter** the square wave can be heard as a separate component to the sound – it is almost a low level buzz. This effect is less pronounced if you use the patch in a track, however if you just hold a single note, you can hear the effect quite clearly. You can also try this patch with a multi-mode square wave. To my ears, engaging the multi-mode square loses the effect of the square wave and does not add much to the patch.

This combination does filter well and as the filter closes, the sound takes a more cohesive tone with the low level buzz of the square wave still being present but being less obvious as the contrasting sawtooth wave is reduced.

saw +2 octave squ and saw squ+2 octave

When one oscillator is pitched two octaves above the other, the two waves can start to be heard as distinct elements. The tone gets brighter but this is more a reflection of the dominance of the higher note. As the two elements are being heard separately, there is little change to the tone when their respective levels are shifted.

saw +2 squ filter and saw squ +2 filter

Both of these patches react well to the filter. When the filter is closed there is a cohesive tone (rather than there being two separate sound sources). When the square wave is pitched higher, the effect of the filter is less significant for the tone – it readily cuts the sawtooth element but only has a dulling effect on the square wave.

saw +3 octave squ, saw squ+3 octave, saw +3 squ filter and saw squ +3 filter

When a sawtooth wave and a square wave are pitched three octaves apart, the sound starts to become less useful as a singular tone and is much more a combination of two elements. That being said, the combination does give some good organ type tones and the sound does react quite well to filtering.

So why have we listened to all those combinations of waves?

We have taken two waves and gone through around 70 permutations of how those waves can be combined. Why?

There are many reasons – the first was to introduce you to an element of the sound palette that is available to you. The tones and colors are not explained anywhere else – you need to listen to them to become aware of the possibilities. Given the components, some of the tones and colors are neither obvious nor intuitive.

Once you are aware that there is a range of tones beyond those offered by the basic waveforms, you can become familiar with the palette and begin to understand how the basic elements interact.

Only once you have a knowledge of the broader tones that are available and how the elements interact will you be able to call up the sounds you need when designing a patch rather than hoping for a happy coincidence. You will understand how to create the sound you need for your track and you will also find that programming becomes much faster.

Another aspect of listening to all of these sounds is that we have ruled out some options. For instance, detuning an oscillator when the other oscillator is several octaves higher does not necessarily give a useful tone. Now we know this information, we don't need to spend time experimenting.

With the combinations we have looked at so far, there have only been two waves. When you design sounds you can of course use more waves, for instance you may want to use a combination of two slightly detuned sawtooth waves with a square wave dropped an octave lower to fatten up the sound.

You should also remember that with modern synthesizers offering many oscillator slots you don't have to use all of the slots when the key is struck. If you want you could allow some elements of the sound to fade in gently over time.

Combining waves – practical patches

If you have worked your way through the examples, you've probably had enough of listening to detailed nuances for a lifetime. Let's move on and create a few practical patches in Z3TA+ using some of the new tones we've created by combining waves.

The following group of sounds has been built with some of the wave combinations we have just listened to. You will also see that I have used some of the modulation options which are discussed in the next chapter.

lead saw squ +1 filter

For this first patch I took the patch we looked at earlier called **saw squ +1 filter**. The original patch didn't necessarily appear to be most immediately useful, however it did have a tone that would cut through a mix without totally dominating, making the wave combination useful as the basis for a lead sound.

To make the patch, I made the following changes to the earlier patch:

- first I reduced the polyphony to one and engaged portamento ("fing, var", setting the glide time, ie p.time, to about 0.4 seconds)
- I set the bend range to four semitones up (a major third) and 12 semitones down (an octave)
- to make the filter a bit more interesting I increased the filter resonance to around 12dB – to balance this I set the cut-off slope to 36dB/octave
- to complete the patch I made it velocity responsive in two places (both set through the modulation matrix):
    - filter cut-off (see below)
    - filter level, so the volume of the patch is controlled by the filter's level which is controlled by key velocity.

The next two chapters will explain more about modulation. Let me explain how the filter in this patch operates – you can figure out why it works once you have read the next chapter.

The filter cut-off reacts to both velocity and the modulation wheel. For normal playing I suggest you set the modulation wheel to its minimum – this will allow you to control the cut-off frequency by using velocity alone. However, this is not the only control you have – you can control the filter in real time with the modulation wheel, so when a note is sustaining you can close the filter by pushing up the wheel and then reopen it as the note sustains.

The velocity controlled filter cut-off combined with the high resonance gives a wide range of playable tones. The modulation wheel then allows for far more extreme (and potentially unnatural control over the filter cut-off).

harpsi two saws +2

This is a synthetic harpsichord based on **two saw +2 filter octaves**. I chose this combination because of the way that it interacts with the filter. I could have used other combinations (such as **two saw phase 90** for different tonal variations).

This patch is unrealistic when compared to a real harpsichord in two areas:

- first, its sound only bears a passing resemblance to a harpsichord (in that it is bright and quite staccato), and
- second, this synthetic patch is touch sensitive.

The first change from the basic patch is that the filter cut-off is now controlled by envelope one (and the filter has the secondary effect of controlling the volume of the patch – when the envelope reaches the zero level the filter is closed and so no note is audible).

The second change is that the volume of oscillator two is modulated by velocity. This means that at quieter volumes the oscillator cannot be heard. At higher velocities the oscillator can be heard as an added brightness in the attack of the note. With the operation of the filter the second oscillator is only heard during the attack phase of the note – once the filter envelope has reached its sustain phase, the second oscillator cannot be heard.

organ two squ +3 filter

This patch is based on **two squ +3 octave filter** and takes advantage of the organ like quality of that earlier patch.

From the base patch I made two main changes, the first to the filter and the second to add some vibrato. I will deal with each in turn.

For the filter, I wanted to do two things:

- first add an element of velocity sensitivity so that the tone can be controlled by the player, and
- second to set the filter so that it closes as notes get higher: this will ensure that the tone is more consistent across the keyboard.

I wanted to add some vibrato to simulate the effect of a rotating speaker. To do this I set two local LFOs (ie polyphonic LFOs – see chapter 5: modulation and other ways of messing with things) to modulate the two oscillators (one LFO per oscillator) and set the frequency of the LFOs to be slightly different. The amount of modulation was set to be the minimum that could be heard.

Other sound sources

Most of the sound sources we have considered so far are likely to be found in conventional subtractive synthesizers. However, there are many other sound sources (using this term in the widest possible sense). Some of these other sound sources are a combination of existing sounds and some are sounds created in wholly new ways.

Samples

All of the synthesizers featured in this book (apart from Vanguard) have the ability to use samples as wave sources. Z3TA+ allows only single cycle waves to be imported whereas Rhino and Wusikstation allow full length multi-samples to be used. Cameleon 5000 resynthesizes sounds which gives other sonic option.

The ability to include other waveforms effectively makes the sonic options limitless.

The book does not cover sampling. However, if you do want to learn more about sampling, then I recommend you check out "Sample this!!" written by me and Klaus P. Rausch.

Frequency modulation (FM)

Highly complex waveforms can be created by taking two sine ways and using one to modulate the frequency of the other.

FM has developed considerably since the earliest FM synthesizers – different wave forms can now be used and FM synthesizers have filters to give even more sonic options.

FM is described in greater detail in chapter 7: frequency modulation synthesis.

Wave-sequencing

In basic terms, wave-sequencing is stepping through different waveforms in a predetermined order. You could have a sequence which plays a sawtooth wave for a beat, then a square wave for a beat, a sine wave for a beat and finally a triangle wave for a beat before looping back to the start of the sequence.

However, simply cycling through four or five basic waveforms is unlikely to give a sonically rich results. Hence most synthesizers that are capable of wave-sequencing (such as Wusikstation) contain a wide variety of waveforms to allow more possibilities and greater sonic nuances. The waves can then be stepped through to give rhythmic effects or cross-faded between to give constantly shifting sounds.

Wave sequencing is described in greater detail in chapter 8: wave-sequencing.

Additive

Additive synthesis often sounds simple in theory – you add some sine waves together to create a new waveform. It is easy to pick a few sine waves and create a new waveform. However, the difficulty comes in choosing the right sine waves and then controlling them over time.

Additive is described in greater detail in chapter 9: additive synthesis.

Chapter 5: modulation and other ways of messing with things

So far in this book we've looked in turn at each of the main element that are pulled together when constructing a sound – the oscillators, the envelopes and the filters. We're now going to move on and look at *how* these elements fit together.

Modulation goes to the heart of creating sounds. It is the technique by which a synthesizer creates the varied nuances of sound. Without modulation sounds may be rich, thick, fat, bright or delicate. However modulation will take a flat sound and animate it, giving it life in the hands of a skilled player.

Modulation: the concept

At its most basic, the concept of modulation in a synthesizer is simple – something is changed by something else and usually in real time.

Conventionally, a "source" will modulate a "destination". So if we think about vibrato, a low frequency oscillator would act as the source and the pitch would be the destination.

In this chapter, we will get in some detail around modulation in synthesizers.

Modulation in the real world

Before we get to the detail, let's think about an acoustic instrument such as the piano. Taking a simplistic view, the factors that have an effect on the sound are velocity, pitch and time.

Effect of velocity on a sound

Velocity is another way to describe loudness, which in the case of a keyboard would be determined by how hard the key is hit. Classically trained musicians are used to seeing markings to indicate pianissimo (very soft) ppp, fortissimo (very loud) fff and so on. With midi we have 127 levels of loudness – this range is called "velocity". So instead of fff you may have a note with velocity in the range of 113 to 127.

With a piano, the velocity (ie the loudness) of a note has an effect in three main areas:

- the volume – there is a direct link between the velocity and how hard a note is struck
- the tone – the harder the key is hit, the brighter the tone, although the range of tones between the softest and the loudest notes may not be as dramatic as the volume differences
- the sustain time of the note – the louder the note, the longer it sustains.

Effect of key pitch on a sound

Clearly the piano keyboard controls the pitch of the note played. However, there are other factors affected by the pitch:

- the tone – the higher the note, the brighter it sounds
- the sustain time of the note – the higher the note, the less time that it sustains.

Effect of time on a sound

The length of time that a note is held is also a factor in the tone of a piano. Over time:

- the sound will get duller (however bright it may have been initially), and
- the volume will decay until it finally reaches zero.

Combining the effect of velocity, pitch and time

A piano note is far more complex than may be suggested here, however, in broad terms you can see that:

- The initial volume of the note is affected by one factor only – velocity.
- Tone is affected by three factors – the initial velocity of the note, the pitch of the note and the length of time it is held. However, you should also note that:
    - the tone increases in line with the increase in pitch (ie the higher the note, the brighter it becomes), but
    - the sustain time decreases in line with the increase in the pitch (ie the higher the note, the less time that it sounds for).
- Sustain time (and also volume over time) is affected by two factors – the starting volume (which is in turn controlled by the initial velocity) and the pitch.

To mimic some of the characteristics of an acoustic instrument, we're going to need to use modulation to control the synthesizer.

Modulation destinations

We will approach modulation in a slightly backwards way and look at modulation destinations before we look at modulation sources. Once we understand what can be modulated, we can work back to choose the best source to create the modulation.

Volume

Volume is likely to be one of the main destinations you modulate. You may want to:

- control the volume of your whole patch – for instance to make the patch touch sensitive or to add a tremolo effect, or
- control the volume of individual oscillators so you can cross–fade between two sounds.

Filter

The two main modulation destinations in the filter block are:

- the cut-off frequency, and
- the resonance.

Often musicians will want to control both in real time – the cut-off frequency to make a sound brighter or duller and the resonance to add some sharpness and bite to the sound.

Some synthesizers (such as Z3TA+) allow you to control the separation of the filters. This is where the filters are a combination of several other filters – with the separation control, the cut-off frequencies of the stacked filters can be separated so that one will be raised in relation to the other (or others) giving differing resonant peaks and a different tone to the filter.

Pitch

The effects that can be obtained by modulating pitch include vibrato and trills as well as weird special effects. Subtle pitch modulation is also important when trying to mimic the properties of natural instruments – for instance a slight and very short rise in pitch at the start of a note will often help recreate part of the attack of a brass instrument or a plucked string instruments.

Vanguard makes a distinction between "pitch" and "detune" and Wusikstation makes a distinction between "pitch" and "fine" – in essence there is no difference between these as destinations, but "detune"/"fine" does allow a greater level of detail in the control.

When using an LFO to introduce vibrato you will probably have to apply less vibrato than a player would apply to a natural instrument. LFO vibrato usually has a fixed frequency (but see below) and so the effect will quickly sound mechanical. Reducing the depth of the effect helps to stop some of the mechanical feel becoming too prominent.

Vibrato speed

When a musician applies vibrato to a real instrument, both the speed and the depth of the vibrato increase – it therefore make sense that LFO speed should be a modulation destination (and the depth of the effect should also be controllable).

Pulse width

A fast way to change the tone of an oscillator or to fatten up a sound is to play with the pulse width modulation – this was discussed in greater detail in chapter 4: sound sources.

Pan

The classic use for panning is auto-pan type effects where the sounds moves regularly between the left and right channels. There are however more subtle uses of panning, for instance panning can be related to the pitch of a note with low notes panned towards the left and higher notes towards the right as if you were sitting in front of a piano.

FX

FX sends and returns can be modulated. Sends can be controlled so that FX is added for emphasis on certain notes. Returns can be controlled so that, for instance, the FX signal is suppressed while the signal that is being fed to the FX is present – this "ducking" effect means that a patch will sound less muddy when it is played since the FX won't swamp the sound.

If you want to get more complicated, you may want to change controls in real time – for instance increasing chorus depth as vibrato increases.

You can also modulate other effects such as EQ or delays.

Sample start point

Wusikstation allows you to modulate the sample start point. You may use this to cut the start of a sample, which may be useful if either:

- you want a sharper start – perhaps the sample fades in, or
- you want to avoid a sound at the start of the sample, for instance if you want to bypass a guitar string being picked but keep the sound of the string sustaining.

Envelope speed

Key tracking was mentioned earlier – the ability to control envelope speed is useful, even for entirely synthesized sounds, to give the sound a more natural quality.

Modulation sources

There are many modulation sources within synthesizers – the most common are LFOs and envelopes. It is also possible for external sources to act as modulators – this is the case with velocity, the pitch bend wheel and the modulation wheel which are all dependent on the musician. However, player expression controllers can be used either as modulation sources or as modulation source controllers. For instance, the modulation wheel could be used:

- to control the filter cut-off – in this case, the modulation wheel would be acting as a modulation source, or
- to control the amount of vibrato applied to a note by an LFO – in this case the modulation wheel would be acting as the modulation source controller.

The main purpose of player expression controllers is, as the name would suggest, to allow musicians greater control over the notes they play. The earliest synthesizers essentially gave the option of note on and note off and little more – today there are more options, although in practice the main controllers that are used are velocity, the pitch bend wheel and the modulation wheel. However, the choice of controllers used is usually quite specific to individual musicians.

The main modulation sources are summarized below – LFOs are explained separately in the following section.

Velocity

Velocity is often used to control:

- the loudness of a note
- the tone of a note by modulating the filter cut-off, and
- the volume envelope attack time (although from the five synthesizers considered in this book, only Rhino and Cameleon 5000 allow you this level of control).

Velocity can either be applied directly to modulate the destination, or it can be used as a modulation source controller to control another source (for instance, an envelope) which is modulating the destination. When velocity is used as a source controller, you can often achieve smoother results.

Velocity layering

With Wusikstation and Rhino there is an additional aspect to consider: velocity layering. This is the process where multi-samples are constructed with different samples assigned to different velocity ranges, so for instance you could layer a soft, a medium and a loud piano sample. This will give a different source tone for the patch at different velocity levels. Velocity layers can be used in conjunction with velocity being used as a modulation source so, for instance, you could use a delicate amount of filter modulation to hide the steps between your three sample layers.

figure 24: Z3TA+ XY controller

Envelopes

There are many things you can do with envelopes. If you apply the envelope to the filter negatively it will close down the filter rather than open it up. Starting a note with the filter closed and then opening it quickly may be another useful technique if you are trying to simulate the "spit" of a brass instrument. For further simulation, you might also want to add a pitch changing envelope at the start of a brass note.

Envelopes can act as modulation sources or can work as modulation controllers – a typical use as a modulation controller would be to slowly fade in and then fade out some LFO vibrato.

Pitch bender

The pitch bender is often hardwired to the pitch bend function ie you won't be able to assign it to control volume for instance. That being said, some synths allow the wheel to be assigned to any control.

Modulation wheel

Conventionally the modulation wheel is wired to the vibrato depth and so acts as a modulation controller.

However, it doesn't have to be so – the modulation wheel can also be used as a modulation source (for instance, with the current version of Vanguard at the time of writing the modulation wheel is hardwired to the filter cut-off).

Key tracking

Key tracking is the process where changes are made according to the pitch of a note.

Take an example, if you pluck the bottom E string of an acoustic guitar, the sound will last for over 10 seconds. However, if you play the top E string at the highest fret, the sound will barely last for 3 seconds. With key tracking we can seamlessly control the length of an envelope to reflect the behavior of the acoustic instrument.

Key tracking can be applied in many ways – for instance, the cut off point of a filter can be controlled by key tracking so that higher notes are brighter than lower notes.

XY controllers

Both Wusikstation and Z3TA+ have XY pads (see figures 25 and 24). These allow you to control two destinations from one controller – for instance, you could control a filter's cut-off frequency and resonance simultaneously. Alternatively you could mix the output of four oscillators.

figure 25: the Wusikstation XY controller

XY pads can work as a modulation source or as a modulation source controller.

Cameleon 5000 gives similar functionality with its morph square. However, the morph square offers additional functionality in that it can be controlled by dedicated multi-part envelopes.

Aftertouch

If your keyboard (or other midi controller) gives aftertouch messages, then the pressure on the keyboard while notes are sustained can be used to control your patch. You could, for instance, use aftertouch to control volume to give a swell to a held string chord.

Aftertouch can either be applied directly to modulate the destination, or it can be used as a modulation source controller to control another source which is modulating the destination.

Expression pedal

Often the expression controller is hardwired to a synthesizer's volume. As with the modulation wheel, it can also be used as a modulation controller.

figure 26: using Wusikstation's controller lane to create gating effects

Wusikstation wave-sequence layers

Wusikstation has two wave layers, each of which has 12 sequence lanes which are conventionally used for wave-sequencing. However, these can also be used as a modulation source (or a modulation source controller) – for instance, if a lane is assigned to control the volume, then gating type effects can be generated easily.

Oscillator output

Rhino, Wusikstation and Z3TA+ all allow the output of an oscillator to be used as a modulation source. The most frequent use for this would be for creating FM sounds.

Random generators

Both Wusikstation and Z3TA+ allow a random generator to be included as a modulation source. This can often help to remove the mechanical quality of some sounds.

Midi control code

Lastly, midi control code can be used as a source or as a source controller.

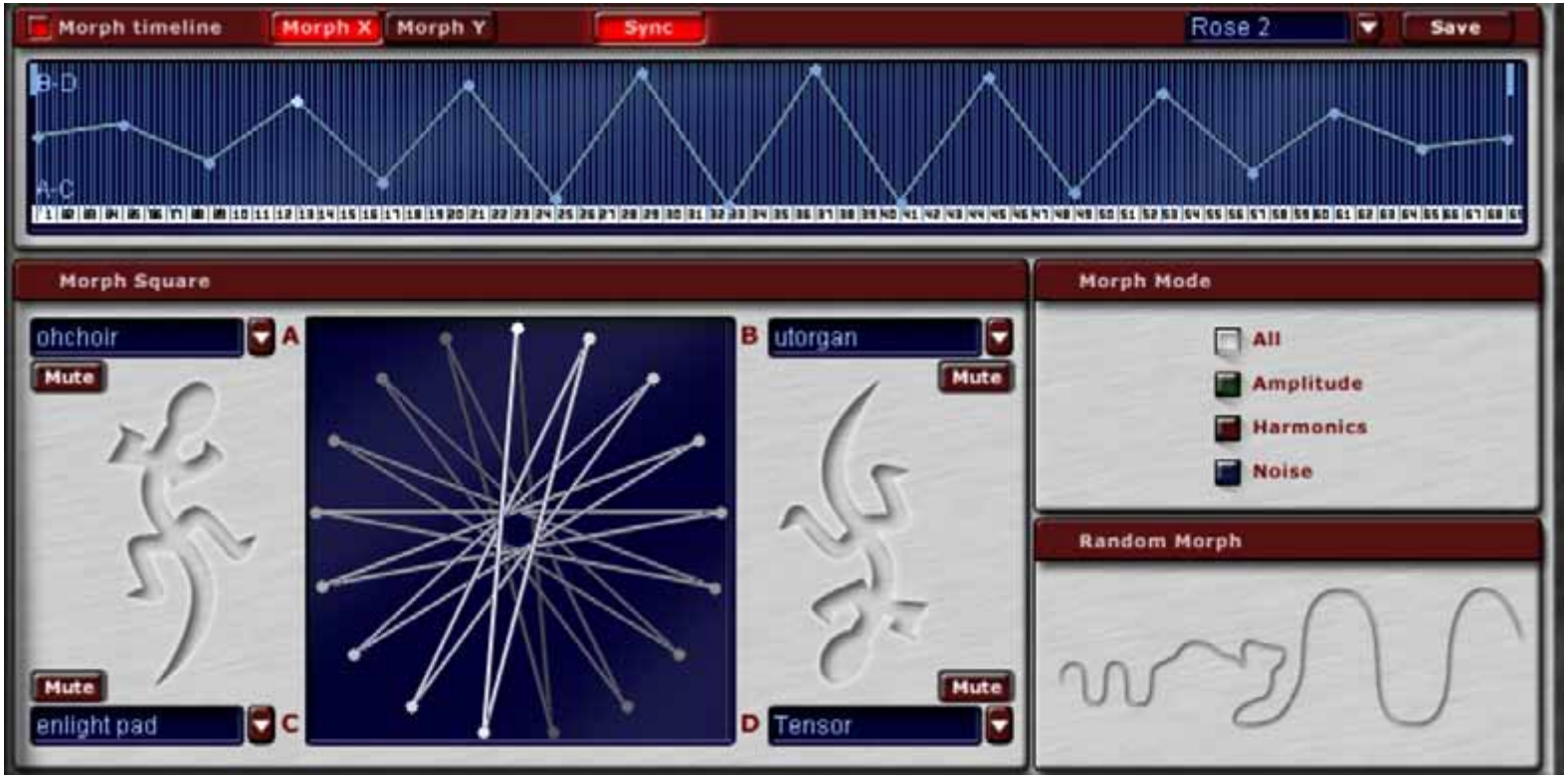


figure 27: Cameleon 5000's morph square with (one of two) controlling envelopes

LFOs as a modulation source

A whole book could be written about LFOs – especially with all of the options that are available under Z3TA+.

The low frequency oscillator is an oscillator which oscillates at a low frequency – for our purposes it oscillates below the audio threshold. As you know, in addition to working as a vibrato source, the LFO can also modulate:

- volume to give tremolo effects, and
- the filter cut-off to give wah-wah or gating/stuttering type effects – especially when a square or pulse wave is selected as the modulation source.

Main LFO controls

The main controls of an LFO are wave shape and depth.

Wave shape

Most synthesizers offer several LFO shapes. Vanguard only offers one (but that's all it needs), Cameleon 5000 offers three, Wusikstation offers six and both Rhino and Z3TA+ effectively have limitless options.

The most common LFO shapes are:

- **Sine** – the sine wave produces an even and rounded modulation source.
- **Triangle** – the triangle wave has virtually the same effect as the sine wave, but the changes are constant. This may give a crisper sound when compared to the sine wave. Both the sine and triangle are ideal waves for vibrato, tremolo or wah-wah effects.
- **Sawtooth** – depending on whether the wave is applied positively or negatively, the sawtooth provides either
  - a rising modulation source which drops abruptly after reaching the maximum value, or
  - a falling modulation source, rising to a peak and then gently falling.

  The sawtooth wave is often used for modulating the filter cut-off.
- **Square** – the square wave is either at its maximum or its minimum. This wave tends to be used as an LFO to give rhythmic effects. It can also be used to modulate the pitch to give trill type effects.
- **Random** – the output from the LFO is a random value. Again this wave is often used for creating rhythmic effects.
- **Noise** – noise is a less commonly used LFO source.

If you look at Rhino and Z3TA+ you will see that you have many more LFO shape options – these are mentioned at the end of this section.

Speed

The speed control affects the frequency of the LFO. If an LFO is used as a vibrato source, then the speed control will affect the frequency of the vibrato. Most LFOs have a range of around 0.1 cycles per second to about 20 cycles per second.

Speed: LFO synchronization

The LFO speed can be synchronized with the tempo of a track so that the time it takes the LFO to complete a whole cycle is a subdivision of the track's tempo. This option is often used in conjunction with rhythmic effects.

Depth

The depth of an LFO's modulation (ie how much it affects the modulation destination) needs to be carefully balanced so that an appropriate range of its effect is controlled. For instance, if you are using an LFO to give some vibrato, you probably want a range which is less than (plus or minus) one semitone. With Z3TA+ it is easy to select the range by using an appropriate curve, for the other synthesizers you will have to judge the range by ear.

Phase

LFO phase may be important. If the LFO phase is synchronized to the start of a note then when the note is triggered the effect of the LFO will be known. By contrast, if the LFO is running freely it may be at any part of its cycle when the note is triggered (this random element may be desirable for ensemble type effects).

Z3TA+ also allows you to control where the LFO starts its cycle, so for instance you could select a triangular wave and direct it to start at the top of its cycle (ie where the wave will have maximum effect immediately).

Monophonic/polyphonic LFOs

LFOs can be monophonic or polyphonic (if you are using Z3TA+: monophonic = global, polyphonic = local).

Monophonic (global) LFOs

In Wusikstation and Z3TA+ (LFOs 1 to 4) the LFOs are monophonic.

With monophonic LFOs, a single oscillator is used whenever that LFO is invoked. So if you are using an LFO to add some vibrato to a pad, when you play a chord all of the notes will have the rise and fall of their pitch synchronized. The effect may not be particularly natural in this instance.

Polyphonic (local) LFOs

In Vanguard, Cameleon 5000, Rhino and Z3TA+ (LFOs 5 and 6) there are polyphonic LFOs. With a polyphonic LFO, each separate note has its own LFO which will start its cycle at a different time from any other LFO.

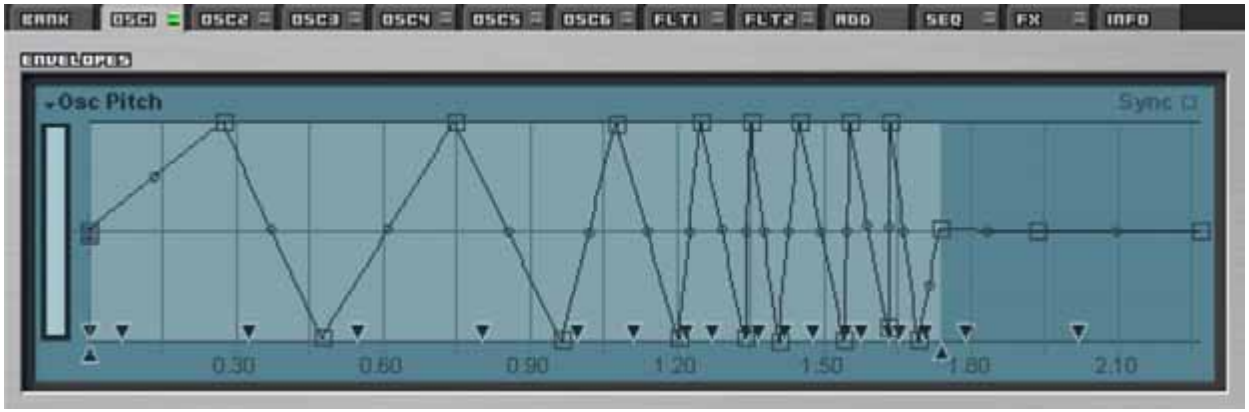


Figure 29: an LFO envelope created in Rhino

Using our pad example again, this could mean that if you are using a polyphonic LFO to modulate a chord, some notes could have their pitch raised while others could have their pitch flattened. This may give a more desirable result for pad, but is less use for rhythmic sounds.

Rhino LFOs

Rhino gives you two options for an LFO source:

- you can use any oscillator as a wave which means you have a limitless LFO shapes available as you can import your own waves, or
- you can draw your own modulation envelope (noting, of course, that there are many LFO envelopes that come with Rhino). The deployment of the LFO envelope is discussed in the next chapter.

Z3TA+ LFOs

As with Rhino, Z3TA+ gives you a wide range of waveforms to choose from, including the option to import user waves. Unlike Rhino, Z3TA+ has a bank of six dedicated LFOs with a host of very powerful features.

In addition to the wave selector and the speed slider, you can also sync an LFO to the track's tempo and control the phase of an LFO. So far, so good. But Z3TA+ can do more.

Delay and fade

- Delay sets the time between the note being struck and the LFO starting its cycle.
- Fade sets the time (after the delay phase has been completed) over which the LFO fades in – the LFO reaches its full value at the end of the fade phase.

If the LFO is controlled in the control column of the matrix (more later) then the delay and fade times are not affected by whether the controller allows the LFO to have effect.

Morphing

The Z3TA+ LFO isn't simply one LFO: there are two waveforms (both running at the same frequency). Z3TA+ gives you the opportunity to

- combine these two waves in various ways
- morph between the waves, and
- join part of one wave with the other.

Lastly there is the "one-shot" option so instead of working as a repeating cycle the LFO will complete one cycle and then cease to have effect. This may be useful if you want some pitch bend at the start of a note.

Figure 28: the Z3TA+ LFO block

Chapter 6: modulation in practice

Using modulation in synthesizers

Modulation is implemented in different ways in different synthesizers. For some there is a dedicated control for any available modulation, for others there is a matrix. We will look at some of the featured synthesizers in a bit more detail to see how they implement modulation. First we will look at the functions of the synthesizers – later we will consider how to use these controls in a practical situation.

Modulation in Vanguard

Vanguard offers a range of dedicated modulation controls – this means that each modulation source and the modulation destination has been set. The advantage of this design is speed when using the synthesizer – the potential disadvantage is a lack of flexibility (although you would be hard pressed to find a modulation option that isn't available as the child's coloring box graphic in figure 30 shows).

Vanguard allows you modulation control over:

- pitch
- pulse width
- filter cut-off
- levels, and
- resonance.

These modulation destination can be controlled by these sources:

- velocity
- envelopes
- LFOs, and
- key position

Let's look at the options in more detail.

Modulating volume in Vanguard

- Volume can be controlled over time by using envelope one. The volume envelope will control the volume of all of the oscillators and the depth of its effect is controlled by the level knob in the envelope one lane. For negative values (ie the level knob is turned to the left) the effect of the envelope will be to cut the volume.
- The second way to control the volume is by velocity, however, velocity has no effect on volume over time. The veltrk knob in the amplifier section controls the effect that velocity has on the volume of the patch – turned to the right and the keyboard has to be hit harder to increase the volume. If the knob is turned to the left, then higher velocities cause the volume to be cut.

Modulating pitch in Vanguard

- Each oscillator has a dedicated LFO to apply vibrato separately to each corresponding oscillator. The speed of the vibrato is controlled by the speed knob and the depth of the vibrato (ie how far from the centre frequency the note is wobbled) is controlled by the detune knob (both knobs in the LFO bank).
- Envelope one can be set to modulate the pitch (plus or minus one semitone) of all oscillators simultaneously – this is controlled by the detune knob in the envelope one lane.
- Envelope two can be set to modulate the pitch (plus or minus 4 octaves) of all oscillators simultaneously – this is controlled by the pitch knob in the envelope two lane.

Modulating filter cut-off in Vanguard

There are eight controls which affect how the filter cut-off is modulated (and there is the cut-off knob itself).

- Both envelope one and envelope two can modulate the filter cut-off. The effect

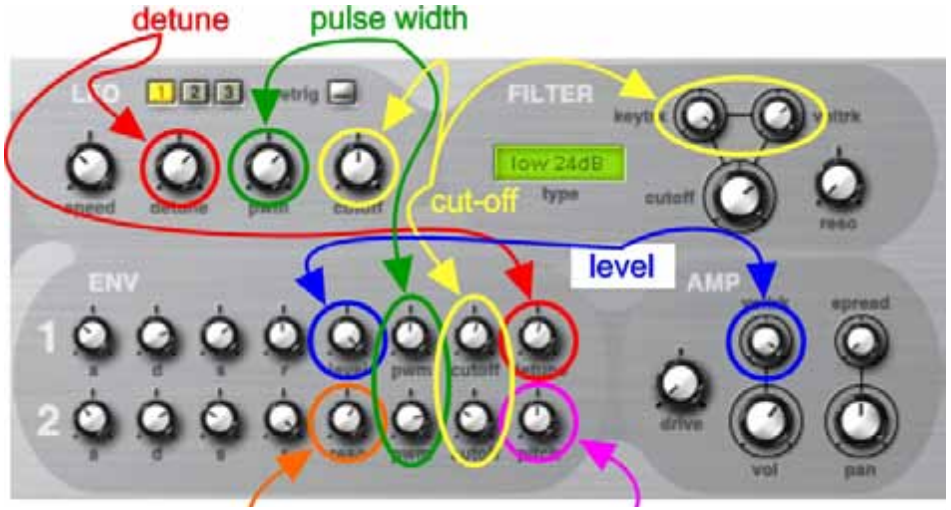


figure 30: the modulation options in Vanguard

of the envelope is controlled by the cut-off knob in each envelope's lane. The envelope can be applied to open the filter (if the cut-off knob is turned to the right) or to close the filter (if the knob is turned to the left).

- Each LFO can be applied to modulate the filter (for instance, to give a wah-wah effect). The effect is controlled by the detune knob in the LFO bank – turned to the right and the LFO opens the filter in relation to the cut-off frequency; turn the knob to the left and the LFO works to close the filter in relation to the cut-off frequency.
- The keytrk knob in the filter bank determines how the pitch of a note affects the filter cut-off frequency. When the knob is turned to the right, the filter opens up as higher notes are played. As the knob is turned to the left, the filter closes as higher notes are played, giving brighter bass sounds and muffled treble sounds.
- The veltrk knob in the filter bank determines how the velocity of a note affects the filter – this knob works in the same way as the veltrk knob in the amplifier section. As there are separate veltrk knobs in each section, a greater degree of control can be exercised over the volume and filter modulation. As with the amplifier section, turn the knob to the right and the keyboard has to be hit harder to open the filter. If the knob is turned to the left, then higher velocities will cause the filter to be closed.
- In the current version of Vanguard (version 1.11) the modulation wheel is hardwired to the filter cut-off and will open the filter.

Modulating resonance in Vanguard

There is only one modulation source for resonance – envelope two. The effect that the envelope has on the resonance is controlled by the "reso" knob in the envelope two lane. Turned to the right and the envelope will increase the resonance, but turned to the left and envelope two will decrease the resonance.

Modulating pulse-width in Vanguard

There are several options for modulating the pulse width of the oscillators. For any of the controls to have an effect, a waveform must be selected that is capable of having its pulse-width modulated – these waves are indicated by "pwm" in their name.

- Each LFO can be applied to modulate the pulse-width of its respective oscillator. The effect is controlled by the pwm knob in the LFO bank.
- Both envelope one and envelope two can modulate the pulse-width of all of the oscillators. The effect of the envelope is controlled by the pwm knob in each envelope lane.

Working with Vanguard's modulation structure

Vanguard is designed to be simple (and therefore fast) to operate. It succeeds in this aim and therefore cannot be criticized for not having some of the more complicated features of the other synthesizers in this book.

Only having two envelopes may be seen as limiting, but in practice, this is unlikely to be an issue. Often envelopes are used in conjunction with velocity controls. As the velocity based modulation destinations are separately controlled, there is sufficient control.

The number of envelopes may also have been a problem if each of the oscillators could be separately controlled – the oscillators cannot be separately controlled and so there is no necessity for one envelope per oscillator.

Three separate, polyphonic LFOs give considerable flexibility. If LFOs are applied to more than one oscillator it is a fiddly task to precisely replicate the LFO settings. This means that you are more likely to get the settings close but not perfect – this lack of perfection tends to add more movement to a patch.

It is unfortunate that vibrato (as an example) cannot be controlled by the modulation wheel – as part of the simplicity of Vanguard, the modulation wheel is hardwired to the filter cut-off. At the time of writing it is understood that assignable controllers may be implemented in Vanguard.

Modulation in Rhino

Rhino takes a unique approach to modulation, which is not surprising for a synthesizer that uses modulation (in Rhino's case, frequency modulation) to produce sounds.

The modulation for each oscillator and each filter is individually set on that oscillator or that filter's page, so the effect of velocity, key tracking and envelopes etc are set individually for each oscillator. The only exception to this is where an oscillator is modulating the frequency of another oscillator (whether in the normal FM manner or acting as an LFO). Where the oscillator is the modulation source, the routing and the depth of the modulation is determined in a routing matrix.

A practical example may help to explain how the modulation system works in Rhino.

Using modulation in Rhino

For this example we will take a simple patch, **modulation example**, which has one sawtooth oscillator running through a filter. Velocity will affect the patch in two ways:

- first, it will make the sound brighter as a key is struck harder, and

- second, it will change the attack time of the volume envelope – at lower velocities the sound will fade in slowly while at higher velocities, the attack will be much faster.

Modulating attack time

In Rhino, each oscillator's level envelope is always applied with 100% depth. If you look at the oscillator's level envelope in figure 31, you will see two time markers have been indicated:

- The most obvious marker is the breakpoint at the end of the attack phase. If there is no modulation of the attack time, this is the attack time that you will hear. For this patch, the default attack time is fairly slow.
- The other time marker is the circled downward arrowhead – this indicates the attack time at maximum modulation. You will see that at maximum modulation, the attack time is quite fast.

Next look at the velocity/aftertouch window. You can see a diagonal line – this controls how the modulation attack time (in that window's dropdown menu, "time" has been chosen) is affected by velocity. The grid works as follows:

- velocity values count from left to right – from 0 (on the left) to 127 (on the right)
- the top axis represents no change (ie the time marker in the envelope takes its full value)
- the bottom axis represents the downward arrow taking its full value.

So if you set the line horizontally at the top of the square, velocity will have no effect – the attack time will always be that set by the envelope. If you set the line horizontally at the bottom of the square, velocity will have no effect – the attack time will be that set by the downward arrowhead. Indeed, whenever the line is horizontal, velocity will have no effect.

In this patch, the line goes diagonally downwards. This means:

- at lower velocities, the attack time set by the envelope predominates (and so the sound fades in slowly)
- at higher velocities, the attack time set by the downwards arrowhead predominates (and so the attack is much faster), and

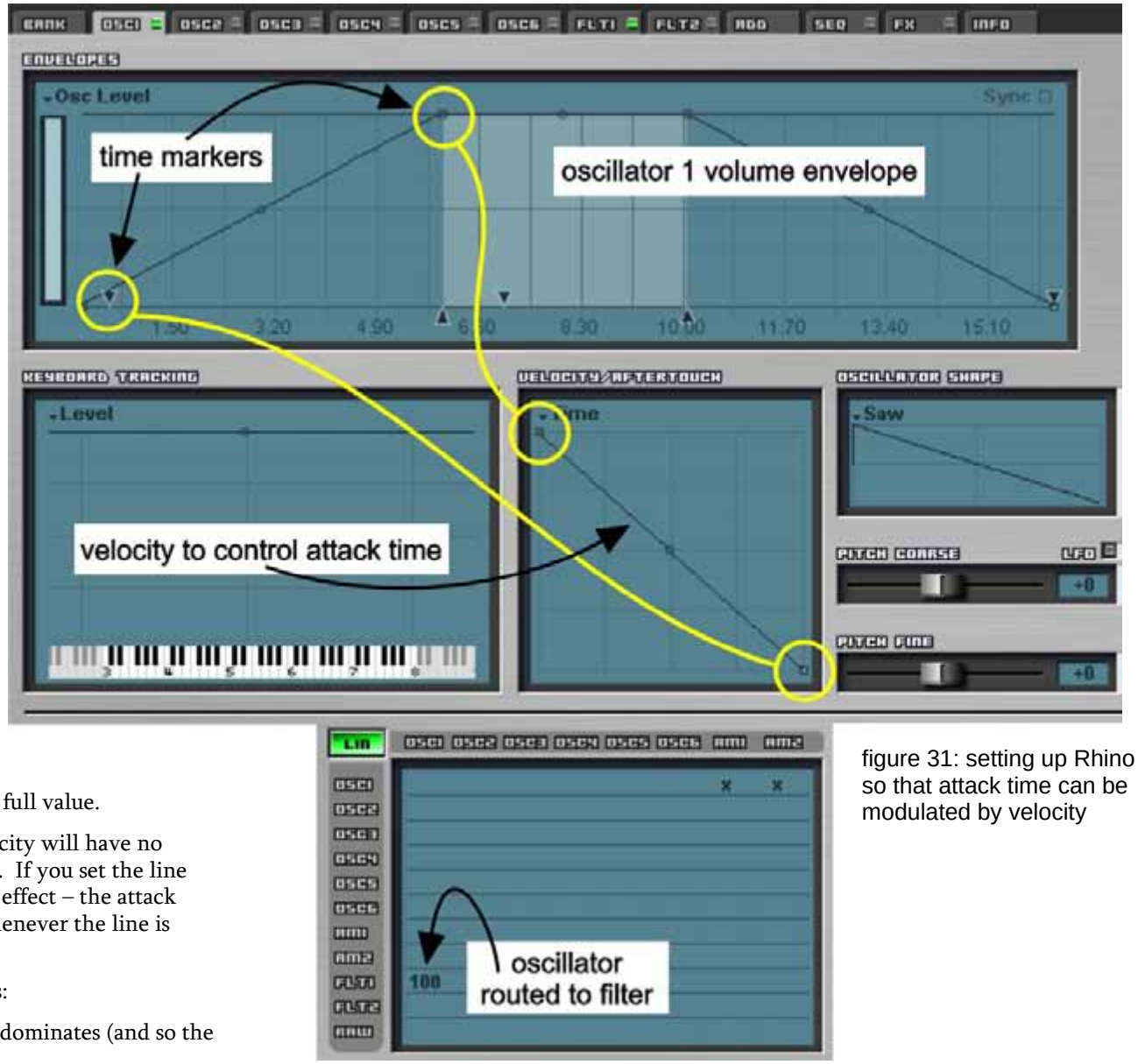


figure 31: setting up Rhino so that attack time can be modulated by velocity

- at velocities between the two extremes, the attack time varies depending on the velocity.

If the line were to go diagonally upwards, then the behavior of the modulation would be reversed (ie the attack time would be faster at lower velocities and slower at higher velocities).

You will notice that in this patch velocity does not affect the volume of the oscillator's level (although this would be easy to achieve using the velocity/aftertouch window and choosing "level").

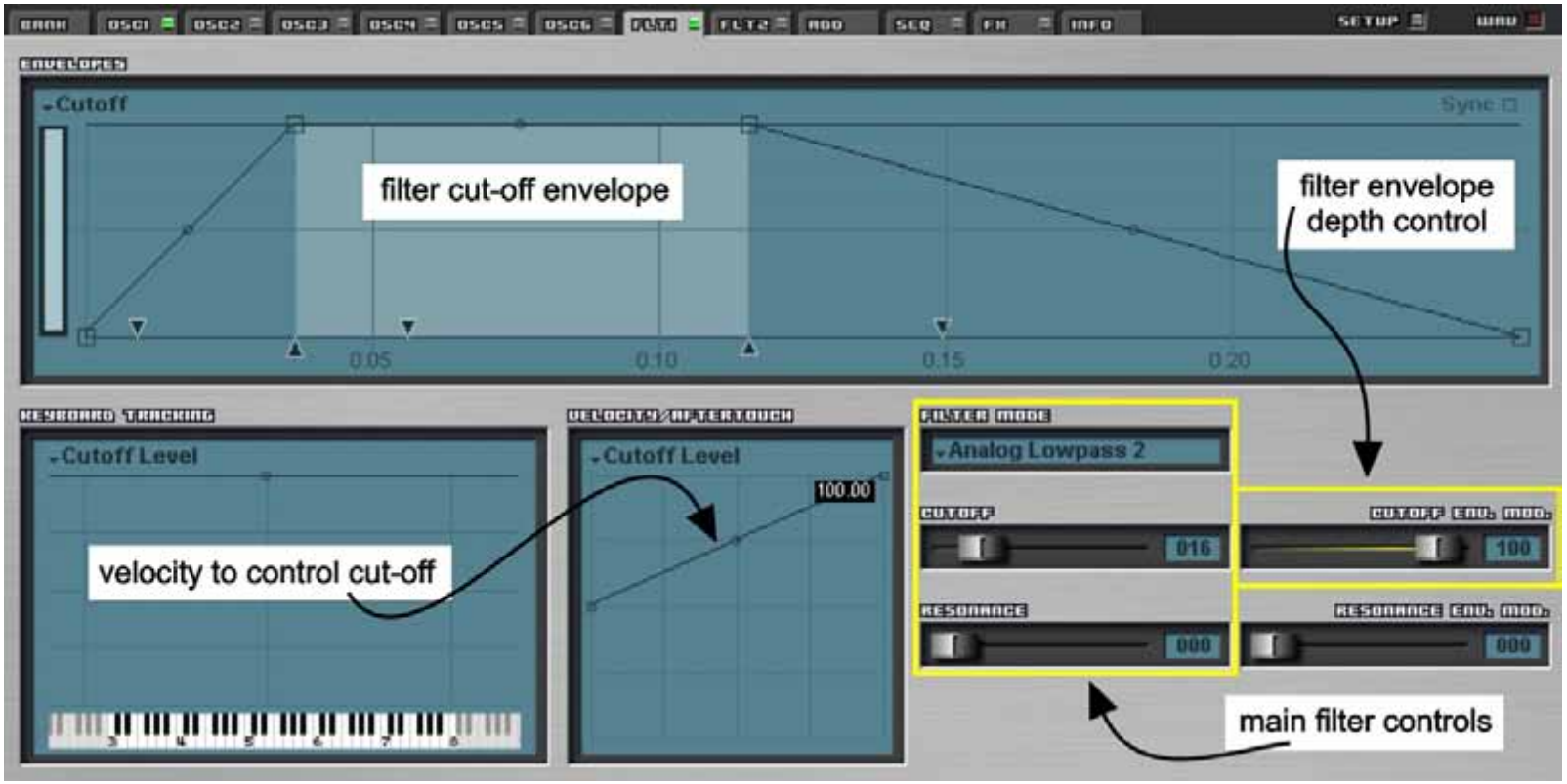


figure 32: setting up Rhino so that the filter cut off can be modulated by the filter envelope and velocity

Modulating the filter cut-off

There are four factors controlling the filter cut-off:

- First (and most obviously), the cut-off slider in the main filter control is set to 16 (on a scale that goes to 100).
- The second factor controlling the filter cut-off is the filter envelope depth slider (labeled "cutoff env mod"). This works in a slightly different way to the way that this type of control works on other synths in that it allows the envelope to effectively cut the filter. Unlike other synthesizers, the envelope depth control does not work to further open the filter from the selected cut-off frequency.

So for instance, if the filter is fully open and the envelope depth control is set to zero, there will be no effect on the filter. However, if the envelope depth is increased to 100%, then the envelope will reduce the cut-off frequency and only when the envelope is at its maximum will there be no effect on the filter. Equally, if the filter cut-off is at 50%, then this limit will not be exceeded when the envelope is at its maximum.

- The third factor is the envelope itself – as the envelope depth control has been set to 100%, this means that the envelope level controls the filter cut-off directly (subject to the maximum level set by the cut-off control).
- The final factor controlling the filter is velocity.

As with the attack time, the way that velocity controls the filter cut-off is set in the velocity/aftertouch window:

- The top of the grid represents the cut-off frequency set by the cut-off slider.
- The bottom of the grid represents a fully closed filter. Like the cut-off envelope, this modulation controller works to close the filter from the cut-off frequency selected by the cut-off slider.

figure 33: Wusikstation modulation matrix

In this patch, the cut-off level will be:

- unaffected at higher volumes, and
- cut a lower volumes – with a velocity of zero the filter will be half closed.

This setting means that the patch will sound brighter at higher velocities.

Summary of modulation in Rhino

You can see that Rhino offers a huge number of modulation options and gives the musician detailed control over every aspect of a sound. The downside to this approach is, as always, complexity. If you want to change the way that velocity affects a patch you may have to change it in eight different places (six oscillators and two filters). If you are modulating a patch with several sources, this could become quite tiresome.

The other disadvantage to this option is a certain lack of flexibility. For instance you cannot set an oscillator acting as an LFO to modulate the filter's cut-off. Instead you would have to draw an LFO type envelope (or use one of the presets) to modulate the cut-off and achieve a wah-wah type sound.

There is also a different approach to assigning controllers under Rhino where this must be done by using "midi learn" to control the parameters.

We will look at some more practical modulation options in Rhino, specifically when creating FM sounds, in chapter 7: frequency modulation synthesis.

Modulation in Wusikstation and Cameleon

Wusikstation and Cameleon 5000 both adopt a "modulation matrix". This is a flexible grid where you can choose:

- the modulation source
- the modulation destination
- the minimum amount of modulation, and
- the maximum amount of the modulation.

Let's look at the Wusikstation matrix in greater detail. There are 36 slots for possible modulations – each slot has five controls:

- source: selects the modulation source, so for instance if you want to add vibrato this would be an LFO
- destination: selects the target to be modulated – using our vibrato example, a layer's pitch would be the destination (of an LFO modulation source)
- min and max: select the minimum and maximum extent that the modulation source affects its destination – if the minimum level is set to be less than the maximum, then the modulation will negatively affect the destination (for instance, if velocity modulates a filter to make a sound brighter at higher velocities, the sound would get duller at higher velocities if the minimum is higher than the maximum)

- amount – governs the extent to which the modulation source affects the modulation destination and also how it affects the destination. To elaborate, if an envelope is set to modulate the filter and the amount is set to 127, then the envelope may fully open the filter. If the amount is set to 65, then the envelope will only partially open the filter. However, if the amount is set to -127, then the envelope will operate to close the filter. The amount control operates in conjunction with the min and max controls and so you may have to balance the three controls to get the effect you want.

This form a matrix gives considerable flexibility and control to the modulation routings.

Wusikstation also gives another option which isn't immediately obvious from looking at the matrix – if two modulation sources are both set to control the same destination, then the effect of the sources is multiplied. So for instance, if an LFO and the modulation wheel are both set to control the pitch of an oscillator, when the modulation wheel is set to zero the LFO will have no effect (ie its effect will be multiplied by zero). However when the modulation wheel is set to its maximum extent then the LFO will have its full effect. This process allows greater control of the modulation.

Wusikstation also has three dedicated controls on each of the oscillator envelopes and modulation envelopes. These hardwire three destinations to the velocity control (as an example, one links the velocity to that layer's/envelope's level).

Modulation in Z3TA+

Like Wusikstation and Cameleon, Z3TA+ adopts the modulation matrix approach. However, the matrix here is more flexible/complicated and therefore gives you many more options as to how you want to apply your modulation.

Z3TA+ gives us five columns in the modulation matrix. Source, range and destination act in a similar manner to the Wusikstation and Cameleon matrixes. There are two main differences with the Z3TA+ matrix: curve and control.

Curve

Curve affects both how the source affects the destination and also the range of control. Some explanation might help.

Let's start by looking at the pitch curves – there are four choices: 1 semitone, 1 tone, 1 octave and 4 octaves. If you apply an LFO as a modulation source, then the curve will determine the maximum amount of pitch variation – so for a gentle vibrato you may select "1 semitone". This would give a vibrato range of +/- one semitone and so you might want to move the range slightly off the maximum.

The pitch curves do not have to be used exclusively with LFOs. For instance, you could equally apply a pitch curve in conjunction with the pitch envelope. Also, there is no rule that the modulation destination has to be pitch – you can use this curve in connection with any destination. And for completeness, you can use the other curves with pitch destinations – the pitch curves just make it easier to understand what the modulation is doing.

The next group of curves is the speed curves – these allow you to select curves which have greater effect either sooner in the curve (fast curves) or later in the curve (slow curves). Both of these curves have two options: positive and negative – if you're using the modulation wheel to control a filter's cut-off, then "positive" would open the filter and "negative" would close the filter.

The last group of curves we will consider are the linear curves – again these come in positive and negative flavors, but now we have another option: bipolar and unipolar. Bipolar means that the curve can generate negative AND positive values. Unipolar means that the curve can be EITHER negative OR positive. Bipolar positive also has the effect of converting a bipolar source to unipolar.

Control

The second feature of the Z3TA+ modulation matrix which isn't (obviously) present in the Wusikstation matrix is the control column.

Control allows a musician to "control" how the modulation source modulates the destination. So taking the example of an LFO acting as a vibrato source, if the modulation wheel is set as the control, then the depth of the LFO vibrato will be directly controlled by the modulation wheel (within the parameter set by the range control).

The range of controls is wide, ranging from velocity, to midi control code (cc) data through to key position.

This functionality can be applied in the Wusikstation matrix, it is just a bit more fiddly and takes two lines (which isn't a problem being as the Wusikstation matrix has 36 lines).

Z3TA+ modulation matrix working in practice

Here's an example of the Z3TA+ modulation matrix being used in practice. A similar (but not quite identical) functionality could be achieved in Wusikstation.

figure 34: Z3TA+ modulation matrix

Less functionality would be available in Rhino, Cameleon 5000 and Vanguard.

We're going to create a bright bass sound (which will take up quite a bit of space in the mix). If you've got the presets, then load up **bass modulation**.

This patch is based on two slightly detuned multi-mode (synchronized) oscillators (using the vintage saw one wave) running into a 36dB low pass filter. The filter is fully closed (requiring quite a bit of modulation to open it) and the resonance is set at quite a high level (10dB). Finally, some reverb is added – the parameters are explained later.

If you manually input these parameters into Z3TA+ you will find that the patch doesn't make any noise. The silence is because the filter is fully closed. So the first thing we're going to do is to modulate the filter.

Filter cut-off modulation

The first step is to make this patch have some sound – to do this we need to open up the filter. So we will set filter one as a modulation destination. If you set the source to "on", then the range control will directly open the filter.

However, I wanted some velocity control here, so I selected velocity in the control column and I also chose the U-lin+ curve. Next I set the limits of the filter modulation ie the maximum and the minimum amount that the filter will be opened (so that the sound is neither too bright nor too dull). For this patch a

minimum of 26% and a maximum of 67% felt right to me (left click and drag in the range to control the maximum and right click and drag to control the minimum).

Finally, I added a bit of envelope control here – this adds a bit of bite to the sound by ensuring the filter opens a bit more at the start of the note and then becomes slightly duller. This behavior mimics real instruments.

To add the envelope control, I will EG1 into the source column. I am using envelope one as a basic ADSR envelope (ie slope time is set to the minimum and slope level to the maximum). Attack is as fast as possible (ie zero), decay is set to 0.18ms and the sustain level is set to 70%. This gives a good sound to the filter modulation.

If you compare the with/without EG1 sounds: without the envelope, the sustain portion of the note is considerably brighter. With the envelope, the start of the note has a more distinct attack and is less bright during the sustain phase – we will fix this dulling of the sound in a moment.

If you play this patch in the bass region it sounds fine. If you play the patch in the higher regions of the keyboard it sounds somewhat indistinct. Although this patch is a bass patch, I want to give it a bit more flexibility, for instance, so that it could be used as a stab (even though I am a keen advocate of designing patches for a specific purpose). This is where we start using key tracking.

As my modulation source I selected U-note# with U-lin+ as the curve and again filter one cut-off as the destination. The range was adjusted to taste and in this instance a value of 38% felt right to me. This key tracking also has the effect of reversing some of the dulling effect of bringing EG1 as a modulation source.

Also note that if you want to use this patch as a stab you will need to increase the polyphony – I kept it at one to ensure the patch is not played in a way that would lead it to being muddy.

The filter is now highly dependent on the velocity of the incoming note and the high resonance value exaggerates the effect of the filter modulation.

Controlling the volume

The amplitude envelope has no direct effect on the volume over time of this patch – it has the characteristic of an organ's envelope (ie on or off). Instead the volume of the patch is controlled by modulating the level of filter one.

To set up this modulation:

- envelope generator two was chosen as a modulation source (and the envelope is set much like envelope one, although the sustain level is slightly lower)
- the full range was selected
- velocity was engaged in the control column, and
- the modulation destination is filter 1 level.

The volume modulation is now as sensitive as the filter modulation – this gives the patch a very playable feel to it.

Adding some drift/detuning

So far the sound of the patch is a bit lifeless to my mind so I added a bit of detuning. The oscillators were already detuned with reference to each other, so I added a bit of subtle vibrato. To make the effect more interesting, I set a different LFO speed for each of the oscillators – this ensures that the LFO does not set up a repetitive pattern that could become fatiguing on the ears.

To set this up:

- LFO one was assigned as the source modulating oscillator one's pitch
- LFO two was set as the source modulating oscillator two's pitch
- the one semitone pitch curves were selected for each of the two assignments
- a sine wave was chosen as the wave for both LFO one and LFO two
- the speed of LFO one was set to 0.58 Hz and the speed of LFO two to 0.43 Hz
- the LFOs were applied to taste by tweaking the range control in the modulation matrix – to my ears 9.6% sounded right for LFO/oscillator one and 13.7% sounded right for LFO/oscillator two.

The sound of the patch has now taken on a richness.

Adding vibrato

This patch is already very playable, but I also wanted to add some controlled vibrato. To do this I:

- called up LFO three and set the wave to triangle and the speed to 4.93 Hz
- assigned LFO three as the modulation source which modulates the pitch of all oscillators
- set the range to full and the curve to pitch whole note, and
- set the modulation wheel to be the control.

This allows the player to add vibrato without affecting the modulation introduced by LFO one and LFO two.

Reverb without the mush

As a final step I wanted to add some reverb, however, I was concerned that this could add some low end "mush", so I needed to ensure this reverb is controlled.

For this patch I used the large hall algorithm with the size set at 75%, the damping at 85%, low at -3.6dB, high at -1.97dB and wet/dry at 75%. This gives a very noticeable wash of reverb, but if I cut the time, cut the mix level or added more damping I didn't like the effect, so did two things to control the reverb.

First I made the amount of reverb touch sensitive, in other words, at lower volumes the reverb will be proportionately less prominent. This is quite simple to achieve – in the matrix:

- the modulation source is "on" and the range is full
- the control is velocity, and
- the destination is reverb level.

Now, with louder playing the reverb effect increases. However, there is a leveraging effect – more signal is being sent to the reverb because the signal is louder and secondly, the reverb itself becomes louder with harder playing.

The second thing I did was to gate the reverb (ie cut the reverb output signal) while the patch is being played. This is quite simple:

- first, a new envelope (in this case envelope generator three) was engaged as the modulation source
- envelope three was set to act and an ADSR type envelope with the sustain set to the full level – the only tweak was to set the attack time and the release time to 0.09 seconds
- the modulation destination is reverb level, and
- the curve is U-lin-. The negative curve means that when a key is held, envelope three has a negative effect on the reverb level (ie it cuts the level completely).

While the notes are sounding, reverb will not be heard. As the notes are released, a halo of reverb is left. In this way the reverb adds effect, but doesn't make the mix muddy. The attack and release times could be less, but they are set just off zero to ensure that the transitions are not too unnatural to the ear.

Chapter 7: frequency modulation synthesis

Background and history to FM synthesis

I promised that there would be no science and no history in this book. If you want to understand more about FM theory and its history (and I would encourage you to dig deeper) then I suggest you start by using the following terms to search the internet:

- "john chowning" – Dr Chowning was the guy who first got to grips with the concept of FM synthesis: all musicians owe a great debt to him
- "dx7" – the Yamaha DX7 was the first mass market FM synthesizer (if you have heard any pop music produced during the 1980s you *will* have heard the DX7)
- "fm synthesis" – this should be self explanatory.

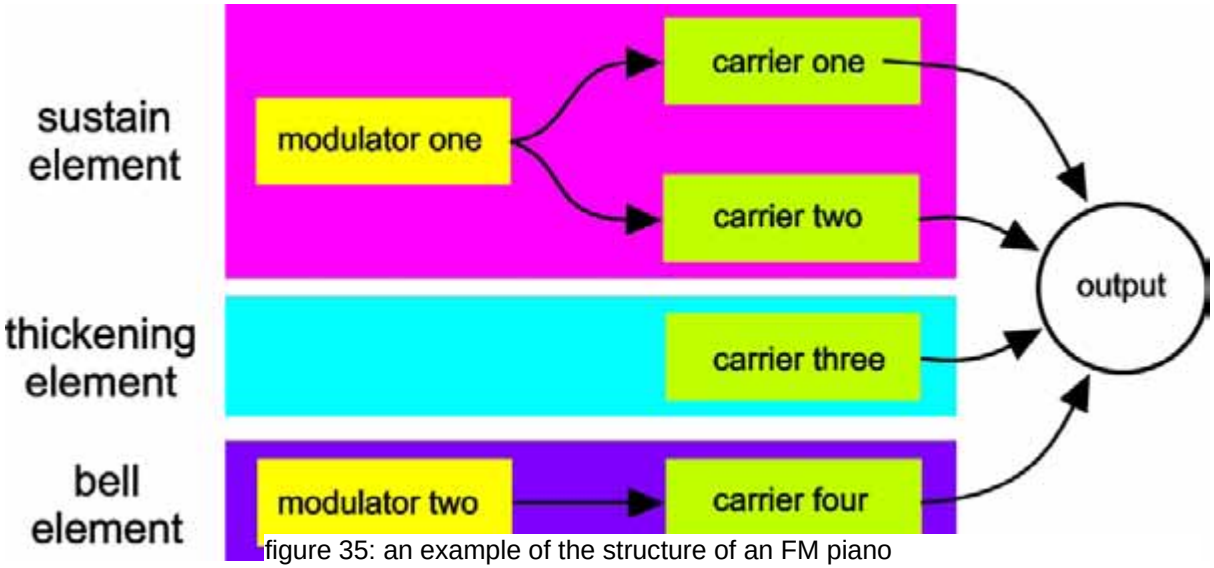


figure 35: an example of the structure of an FM piano

Elements to the FM sound

For many people, FM synthesis falls into the category of "too difficult" – it seems both cumbersome and complicated giving unpredictable results.

However, FM is comparatively simple in concept – it just has a lot of possibilities which make it appear quite complicated. It also requires a different mindset from that needed for subtractive synthesis. So forget every way of working we have discussed so far and let me explain.

An example patch structure

Before going any further, I want to have a quick look at an example patch structure. This should give a context to how an FM patch may be constructed.

It is quite usual for an FM patch to include several elements. For instance, a classic electric piano type patch will have two main elements – the bell tone (heard when the key is struck) and the sustain sound. As an example, this sort of sound could be structured as follows:

- a first modulator/carrier group (these terms are explained below) creates the sustain sound
- a second carrier (modulated by the first modulator) doubles the sustain sound created by the first carrier – using the same modulator for both carriers is more efficient but restricts the flexibility of the sound
- a third carrier (this time without a modulator) is used to thicken up the sound – the three carriers together create a cohesive foundation for the sound
- another modulator/carrier pairing then creates the bell sound.

One of the difficulties when creating a sound of this nature is to ensure that the two separate parts of the sound (ie the sustain part and the bell part) work together to create a convincing sound.

Operators

Instead of having oscillators, FM synthesizers have "operators". The first mass market FM synthesizer, the Yamaha DX7, had six operators – each of which was a sine wave. To create an FM sound, two operators are needed – a modulator and a carrier.

A modulator can be thought of as acting in a similar manner to a low frequency oscillator (LFO). It modulates the frequency of the carrier – the carrier is the operator that is connected to the output and is therefore heard. However, the frequency of the modulator is much higher than would be conventional for an LFO. Indeed, the frequency of the modulator is in the audio spectrum so the modulator could be heard if it were to be connected to an output.

The effect of modulating a carrier with a signal in the audio spectrum is that the effect is not heard as vibrato, but instead the tone of the carrier is changed, quite often significantly.

Software synthesizers which have an FM function (such as Rhino, Wusikstation and Z3TA+) allow the option of including different waves as operators. These different waves give different sounds. You could probably spend a lifetime without exhausting all of the possibilities offered for FM synthesis using sine waves alone. To my mind other waves just give too many complications without giving such pleasing results so are not a practical proposition – your opinion on this matter may be different.

The DX7 had six operators – these were arranged in algorithms. Rhino and Wusikstation also have six operators, however, their routing options are fully flexible so there is not the restriction imposed by hardwired algorithms. Z3TA+ also has six operators, but its routing options are slightly less flexible.

Filters

The original FM synthesizers did not have filters. All of the synthesizers featured in this book which offer some form of FM also give the opportunity to filter FM sounds. This gives yet more possibilities. As you will see later, it is also possible to create hybrid FM/subtractive patches to take advantage of the best of both worlds.

There has been a long (and continuing) debate about the respective merits of digital and analogue in the production of music. For this book, the argument is somewhat irrelevant – all of the synthesizers featured here are by their nature digital, even if they do have the facility to recreate some of the behaviors of analogue machines. Accordingly, I would like to move the conversation a bit further and talk about additive versus subtractive.

Subtractive synthesis can be characterized as creating "fat" or "warm" sounds, where digital is often characterized as being "thin" and "cold". Both characterizations are generalizations which have elements of truth and falsehood (it really all comes down to the quality of the programming).

However, in some circumstances:

- a subtractive sound can dominate a broader element of the frequency range than an analogue sound, but
- an FM sound can be brighter/sharper, cutting through a mix.

Provided you have programmed your sounds in an appropriate way, this can means that FM and subtractive sounds are the perfect compliment for each other, with the characteristics allowing the respective sounds to work together perfectly into a mix.

Controls

The essence of the sound produced by an FM synthesizer is not simply a combination of the individual elements, but the interaction between the modulator(s) and the carrier(s) over time. As one element changes in relation to another there are tonal shifts which gives FM its unique sound. The aspects that affect a sound are:

- the frequency of the modulator relative to the carrier – as a general rule, the higher the frequency of the modulator relative to the carrier, the brighter (or more metallic) the sound, and
- the amount of modulation: this is controlled by the output level of the modulator – the greater the output, the greater the effect of the modulator (again, this affects the brightness).

To achieve the constant shifts and design the desired sound, each element must be controlled: usually (but not always) the pitch relationship between each operator is fixed (and controlled by the keyboard), but the level of each operator is dynamic. There are several tools in the armoury to bring about these controls – envelopes, velocity scaling and key scaling – we have used them all in subtractive synthesis.

Envelopes

Much of the character of a sound – any sound – is captured in the note's attack. Conventional ADSR envelopes may not give sufficient range of control and so since the earliest FM synthesizers, more complicated envelopes have been used. Rhino, offers highly flexible envelopes (which offer a much greater degree of control than was available on the original DX7).

FM sounds where the modulators are static can often be quite harsh and fairly uninteresting.

When sounds with many layers are created, then envelopes are key to achieving the necessary control over the sound. Take the example of the electric piano (set out above): when constructing the envelopes, the sustain portions should (as would be expected) sustain. The bell elements of the sound should have quite a percussive attack and a swift decay. However, the decay should be sufficiently smooth that the change between hearing the bell element and the sustain elements is imperceptible to the listener (and the musician).

As you will see later, the attack signature of a sound can be readily controlled by having different attack times for the modulator and carrier envelopes. For instance, to create the "spit" of a brass note, a slower attack time can be selected for the modulator (when compared to the attack time for the carrier).

Velocity scaling

When creating "pure" FM sounds (ie when not using filters) the tone is often controlled by using velocity scaling to control the output level of the modulators. The output of the whole patch is controlled by using velocity scaling to control the level of the carriers.

Even more than is the case for subtractive synthesis, velocity scaling does not have to have a range from zero to the maximum. For instance if velocity scaling is controlling the tone of a sound, then you may want to set one end of the velocity range so that the sound is dull and the other so that the sound is bright – dull and bright may not correspond with no modulation and maximum modulation.

Velocity scaling can give the musician immense amounts of control over an instrument, perhaps to a level where the synthesizer can come close to mimicking some of the behaviors of a real instrument.

Key scaling

With subtractive synthesis key scaling is usually only used to open up the filter a bit when higher notes are played. With FM synthesis, key scaling is far more important.

It is very easy to create metallic sounds with FM synthesis. It is harder to create usable metallic sounds. One of the keys to designing FM sounds is to ensure that the overtones are appropriate. It is quite easy to create a sound which works within a limited key range, but does not work outside of that range. Often this will manifest itself as a sound that begins to sound too harsh. The harshness can be addressed in two ways:

- the modulation can be reduced – this may make the sound work outside of the first range, however, it will probably affect the original sound, perhaps making it too dull
- the level of the operator can be reduced in certain key ranges – this is more likely to obtain the desired result. This is what we mean when we talk about key scaling (or key level scaling as it is sometimes called).

Tuning

With FM sounds tuning can be quite a challenge. First, the pitch can be affected by the interaction of the modulator and the carrier. Secondly, the resulting waveforms can include many harmonious and inharmonious elements – if the inharmonious elements are predominant, then the resulting sound (especially if a chord is struck) may not be pleasant.

Getting to grips with FM programming

Patch structure

The essential element of the FM sound is a modulator and a carrier working together. Sounds can have more than one modulator: there may be several carriers each with their own modulator or several modulators modulating one carrier – these possibilities are described below.

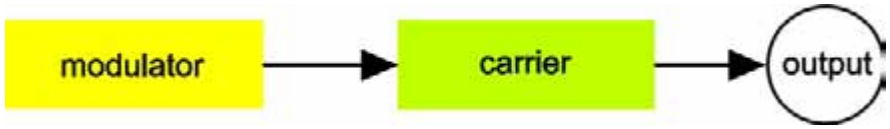
Combining operators

As we have mentioned, there are many ways that operators can work together to create a sound. For the sake of clarity, let me explain the terms I am going to use in this chapter.

Simple FM

Simple FM is where one modulator drives one carrier.

This arrangement is the very essence of FM sound. Most of the classic FM sounds can be built around this arrangement, although often several simple FM stacks will



be layered together to achieve a thicker sound. Alternatively, some patches may be built around several simple FM stacks providing different elements of the sound: the classic example of this is the FM electric piano sound where the bell and the sustain portions can be built from separate simple FM stacks which are then layered.

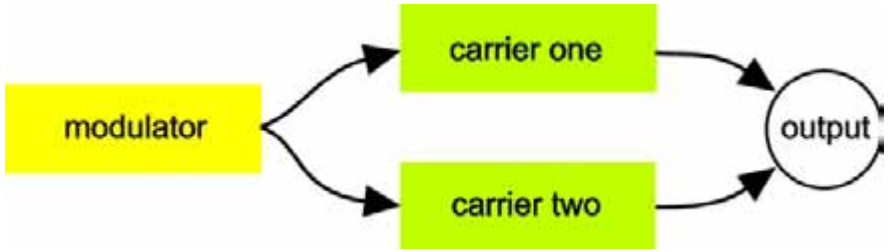
Often one modulator/carrier stack alone will give a weak sound, much as a single oscillator in a subtractive synthesizer can give a weak sound. Modulators and carriers can be doubled to thicken up a sound – again think of how oscillators are doubled in a subtractive synthesizer.

The examples in this chapter use Rhino, however, two of the other synthesizers (Z3TA+ and Wusikstation) have FM capabilities. Both of these other synthesizers can easily implement simple FM stacks and can have up to three simple FM stacks arranged in parallel. However, Z3TA+ does not have the routing options to allow either of the parallel operator arrangements described below.

Parallel carriers

With parallel carriers, one modulator drives two or more modulators.

Parallel carriers allow the effect of two simple FM stacks to be achieved by using



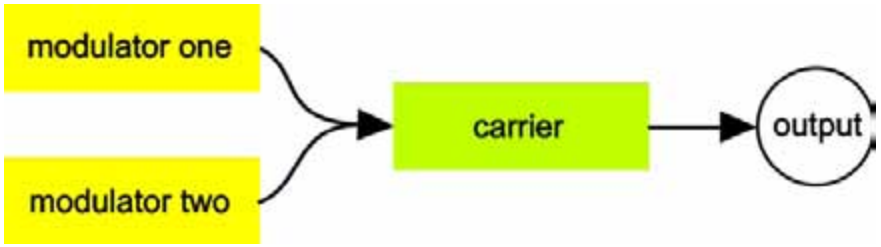
three operators rather than four. This leads to greater programming efficiency and marginally less CPU load.

The downside of parallel carriers becomes apparent if you detune one carrier (say by a couple of cents) to get a thicker sound. While the sound is thicker, there is also phase cancellation which is primarily heard as a rise and fall in the volume.

However, if you use two simple FM stacks and detune both the modulator and the carrier in one stack, a much richer sound can be achieved without such apparent phase cancellation (but you are using more operators and losing the advantage of parallel carriers).

Parallel carriers do not need to be at the same pitch. For instance, if you take one carrier at the base frequency and another seven semitones higher, you have the basis for a Wurlitzer type electric piano sound. Equally this arrangement can yield many wooden (as in tuned percussion) type sounds.

As a side note, just because two unmodulated parallel carriers sound odd when they



are played together, that does not mean they will not work together when they are both modulated by the same modulator.

Parallel modulators

With parallel modulators, two or more modulators drive one carrier.

Parallel modulators offer far more complicated and rich sounds than are available from a simple FM arrangement. However, more options give more complications: the relationship between the two modulators will have a considerable effect on the sound.

If the modulators are of the same pitch, then the effect of two modulators will be to enhance the amount of modulation. Tonally this means that the brightening effect of the modulation will be increased.

Very simplistically, if the modulators are all set at intervals which correspond to integer multiples of the carrier's frequency, then the sound with parallel carriers will give a harmonious tone. Different intervals can give many different tones – some of which may be useful in a musical context, other of which may fall more to be considered as sound effects.

Another simple, but effective use for parallel modulators is to apply different envelope attack times – used in conjunction with differently tuned modulators, this can give a very natural sound. Again, taking the example of a simple electric piano:

- the first modulator could have a very fast attack and a short decay – this could be tuned to a higher pitch than the carrier to give the bell sounds, and
- the second modulator could have a slower attack and a sustain portion – this could be tuned to a pitch below the carrier to give a more bassy/wooden type tone.

If two modulators are very slightly detuned, then this will lead to interference between the modulators (in the same way as would be the case if two oscillators in a subtractive synth are detuned). However the net effect is that the amount of modulation (and therefore the tone) will constantly shift as the operators cancel each other out. Depending on the particular patch, the results may or may not be desirable.

Cascade

With a cascade, one modulator drives another modulator which itself then drives the carrier.

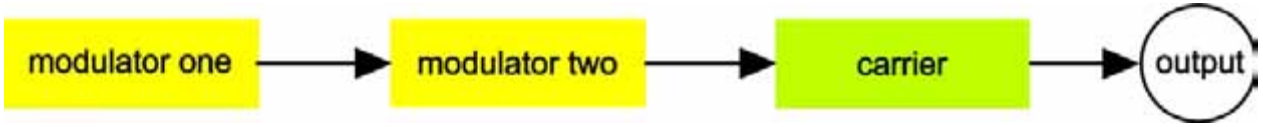
With a cascade, the first modulator outputs a sine wave. However, the second and subsequent modulators are themselves being modulated. Accordingly their output is a different waveform. Cascades can produce the richest and most complex FM timbres: a similar (but less controllable) effect can be created in a simple FM stack by modulating with waveforms other than the sine wave.

Although not illustrated in the graphic, it is possible to have more than three operators in a cascade, however the results become less predictable and there is a greater tendency to producing noise.

Both Wusikstation and Z3TA+ can readily implement FM cascades – indeed, Z3TA+'s architecture was specifically designed to allow this sort of FM arrangement.

Building a first FM patch

As a first step, let's look at an FM patch. Once we have done this we will then go back and look at the elements of the FM sound in greater detail. If you have the



patches, load the bank **rhino chapter 7 FM synthesis bank**, and call up the patch **FM bass** (which should be at the top of the list).

All of the example patches in this chapter are built with Rhino. You will make your life much easier if you familiarize yourself with the Rhino user manual. We are also going to be using many of the modulation concepts which were discussed in chapter 5, so please check out that chapter before you get too deeply into this chapter.

If you play **FM bass** you will hear only one operator: operator one, which is a sine wave. This patch will end up as a three operator patch (and two further operators can be called into action for illustrative purposes).

Setting up the patch

For this patch, each of the five operators has a sine wave selected. An envelope has been drawn (it is the same envelope for all operators) with an immediate attack and an exponential decay, so the sound decays swiftly initially but the rate of decay slows. This envelope mimics the envelope of a piano or guitar. For this example there is no key scaling or velocity scaling. This is intended as a bass sound, so please play the patch in the lower ranges (although the patch is quite playable at higher pitches).

If you look at the Rhino matrix, you will see that some of the numbers are "grayed" out. This means that the operator is working, but that routing has temporarily been muted. To turn the routing on again, simply right click on the grayed out number. If you didn't know about this, go back and read the Rhino user manual.

For anyone who has not purchased the patches but who wants to create this patch you will need the following settings (if you have the patches, skip this section):

- level envelope (for all oscillators): fastest attack time, decay time 2 seconds (after which volume = zero), curvature control set to approximately 25 (see Figure 36)
- all of the oscillators have sine waves
- oscillator one has its coarse pitch set to zero, its raw output should be set to 100, it is modulated by oscillator two and oscillator three, both modulation levels should be set to 60

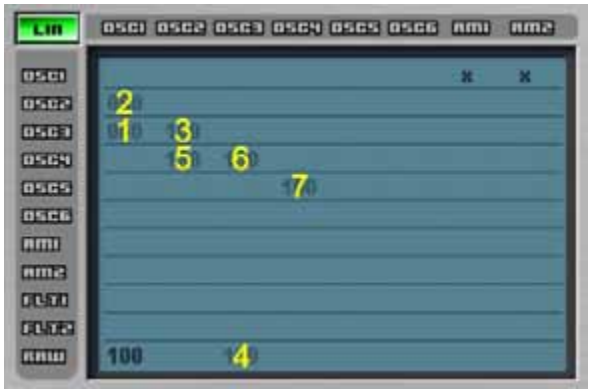


figure 37: the routing matrix for FM bass

Figure 36: FM bass envelope

- oscillator two has its coarse pitch set to -19, it does not have any output (ie it only acts as a modulator), it is modulated by oscillator three and oscillator four, both modulation levels should be set to 100
- oscillator three has its coarse pitch set to -12, its raw output should be set to 100, it is modulated by oscillator four – this modulation level should be set to 100
- oscillators four and five both have their coarse pitch set to -12 and their raw outputs set to zero, oscillator four is modulated by oscillator five – this modulation level should be set to 100

Once you have set this patch, all of the modulations should be switched off (by right clicking in the matrix). Only the raw, unmodulated, output from oscillator one should remain.

Initial modulation: simple FM

Once you have loaded **FM bass** and made the mutings so that only operator one is heard, listen to the sound of that operator. Play some notes in the bass region: you will hear a pure sine wave. Then right click where the figure "1" shows in figure 37 above (ie where the osc 1 column and the osc 3 row intersect) – this will engage operator three as a modulator for operator one, and the figure will no longer be grayed out. You will hear that the sound immediately becomes:

- louder
- lower in pitch

- darker in tone.

Operator three has its pitch set one octave (12 semitones) below operator one's pitch. When an operator with a lower pitch modulates a higher pitched carrier, the sonic result is a louder, lower, darker note.

Switch off that modulation and switch in the modulation indication by the number "2" in the image (ie right click where the oscillator one column and the oscillator two row intersect). This will engage operator two as a modulator for operator one. In comparison to the sound when operator three was the modulation source, this sound will be even lower in pitch (and may subjectively be darker in tone).

The modulator in this case (operator two) has been pitched 19 semitones (an octave and a fifth) below the pitch of the carrier. You should note that as a side effect, the perceived pitch of the sound when the modulator is pitched below the carrier is more likely to be governed by the modulator rather than the carrier.

Go wild – bring in operator three as well by right clicking as indicated by the figure "1". You will hear that the tone changes quite significantly and the pitch again feels lower.

Developing the patch: cascading operators

Switch out operator three as a modulation source for operator one: you should now have a simple FM sound of operator one being modulated by operator two. Let's cascade the modulators – set operator three to modulate operator two (which is modulating operator one). Right click (as indicated by figure "3") where the oscillator 2 column and the oscillator 3 row intersect.

You will hear quite a radical change in tone. The sound becomes much brighter and more aggressive without losing its bass weight.

Now also add operator 3 as a modulator to operator one (right click where indicated by the number "1").

Adding weight

To really add some weight to this sound, operator three can also act as a carrier. Right click where indicated by number "4" and you will hear the sine wave generated by operator three being added directly to the output, as well as acting as a modulator to operators one and two.

Dominant operators

Take some time to listen to the interaction of these three operators. The dominant element of the patch is the three cascaded operators – operator three modulating operator two which in turn is modulating operator one. This is also the dominant element in the pitch. If you play this patch with other patches it will sound in the wrong key: this is the effect of operator two. To reach regular pitch, all of the operators should have their pitch raised by a fifth (ie seven semitones).

First patch: next steps

You could take this patch further and add more modulators. While interesting, I preferred the simplicity of the three operator patch. However, you may disagree. In any case, here are a few simple additional modulations you could add:

- Operator four could be set to modulate operator two (number "5" in the graphic). This adds an interesting element to the tone.
- Alternatively, operator four could be added to the start of the cascade and so modulate operator three (right click where indicated by the number "6" in the graphic). On its own I don't particularly like this tone, but I'm sure I could find a use for it.
- Finally, operator five could be added at the start of the cascade (right click where indicated by the number "7"). I don't like this sound – however, it does illustrate the downside of adding the additional operators: the noise. The patch goes from having a rich sound to having annoying elements of noise.

Velocity and key scaling

If you wanted to take this patch further, then by applying some velocity scaling to operator one, the volume could be controlled by velocity. The tone can be controlled by tweaking the velocity response of some of the modulators – ideally operator three.

However, the challenge here with velocity scaling comes because the effect of the modulators is to make the sound darker – reduce the volume of the modulators and the sound gets less duller (a good thing in this context). Remove the modulators and you are left with a sine wave at a higher pitch than the fundamental of the bass note (a bad thing). Also, operator three is both a carrier and a modulator: if you cut its volume too radically you will lose even more of the weight of the sound. As with all sounds, your job is to balance the various elements to produce a musically usefully sound.

Building blocks of FM sound

I now want to walk you through the basic elements of the FM sound. For this you will find it much easier if you have the accompanying patches. The bank accompanying this chapter, **rhino chapter 7 FM synthesis bank**, contains approximately 60 Rhino patches. If you don't want to buy the patches you can still follow the chapter. However, you may find it harder to pick up some of the nuances in the detailed patches towards the end of the chapter.

1:1 ratio – FM using different modulators

This first group of patches illustrate the basic tones that are available in a simple FM stack by modulating a sine wave with various different waveforms. I have used five different waves – Rhino has over 100 waves so there is a lot of room for experimentation if you are curious.

For these examples the modulator in each patch is touch sensitive. This will give you an idea of the effect of the level of the modulator on the tone of the patch. You should also note that for these patches the envelopes are very basic (the operator level does not change over time).

For those of you without the patches, you can set up Rhino for this group of sounds in the following way:

- operator one (to act as the carrier) – coarse pitch set to 0
- operator two (to act as the modulator) – coarse pitch set to 0
- both level envelopes – attack time: 0, no decay phase, sustain at full level
- amount by which modulator modulates carrier = 100
- the modulator is fully velocity sensitive. To set this, go to operator two's page – in the centre you will see a box labeled "velocity/aftertouch". Choose "level" from the dropdown (if this is not already selected) and then drag the left hand square on the velocity scale to the bottom left corner. Ensure that the right square is in the top right corner.

The wave forms are suggested by the patch names.

sine to sine

This is the most elementary FM configuration. The sound produced has a reedy/woodwind quality that is perhaps reminiscent of an organ's tone.

triangle to sine

When a triangle wave is set as the modulator the result is a much brighter tone than when the sine wave is used. However, at lower levels of modulation, either when less velocity is used or if the output of the modulator is reduced, the sound of **sine to sine** and **triangle to sine** are similar.

square to sine

Square to sine gives a brighter sound again and the overtones generated by frequency modulation can be more clearly heard. However, this patch shows an interesting behavior – at higher levels of modulation (in other words at higher velocities for this patch) the sound gets thinner and quieter.

saw to sine

Like **square to sine**, **saw to sine** sounds thinner and quieter with higher levels of modulation. With higher levels of modulation the sound also tends towards noise: generally this may be unwanted, however, it may be useful, for instance when creating a woodwind type patch.

At mid levels of modulation, the sound is quite warm and rich and is reminiscent of an overdriven-type sound.

With the exception of **sine to sine**, these patches all exhibit (to a greater or lesser extent) the following behaviors:

- at low levels of modulation, the sine wave alone is heard
- as the effect of the modulator starts to become apparent, two separate elements are heard – the sine wave and some higher frequency "distortion"
- at mid levels of modulation two elements are present but the sound becomes warmer
- at higher levels of modulation only one element of sound is heard (largely because the distortion sound drowns the sine wave)
- as the levels of modulation increase further, the sound tends to noise and becomes quieter.

brown to sine

Brown to sine takes a different approach and uses noise (in this case brown noise) as the modulation source.

This arrangement gives a bit of a strange result. The sine wave is clearly audible and does not seem to be affected much by the process. However, the brown noise is amplified by using it as a modulation source. If you switch off the brown noise as a modulator and engage it as a carrier you will hear the natural tone of the noise (being less strident than when used as a modulator) and the volume will be reduced.

1:1 ratio – FM using different carriers

The next group of patches illustrate the basic tones that are available in a simple FM stack when different waves are modulated with a sine wave.

For these examples the modulator in each of the patches is again touch sensitive. This will give you an idea of the effect of the level of the modulator on the tone of the patch. As before, the envelopes are very basic (the operator level does not change over time).

sine to triangle

When the sine wave modulates the triangle wave, the sound gets brighter. It takes on the quality of a slightly nasal sawtooth wave.

sine to square

When a sine wave modulates the triangle wave, the resulting sound is closer to two pulse waves which have been phase synchronized – the tone, as you would expect, is brighter and harder but still very musical.

sine to saw

The sawtooth wave is already harmonically rich. When modulated by a sine wave, the effect is quite subtle: at the maximum amount of modulation, the sawtooth has a thinner, perhaps slightly sharper, quality.

sine to brown

The effect of the sine wave modulating brown noise is to make the noise output brighter and louder. If you want a brighter noise source, it may be quicker to choose a different color noise (for instance, pink noise).

Comparison: sine wave as carrier or modulator

While rather a sweeping generalisation, it can be heard that a key difference between the patches where the sine wave is the modulator and the patches where the sine wave is the carrier being modulated by a different wave form is the consistency of the output signal. Where the sine wave is the modulator, one sound is heard (irrespective of the level of the modulator). Where the sine wave is the carrier, there are generally two sounds, the sine wave and some distortion – there is also a tendency to noise.

However, do not let these comments rule out the other wave forms as modulators (or carriers for that matter). These different waveforms give different shades of tone – their results will only be unpredictable until you become familiar with their operation.

1:1 ratio – FM using same modulator and carrier

This last group of patches illustrates the tones that are available in a simple FM stack when an operator is modulated by the same waveform. You should note that the results here differ from those that will be achieved by creating a feedback loop in an operator. With a feedback loop the sound will get brighter but, as might be expected, at higher levels of feedback there is far more of a tendency to noise.

For these examples the modulator in each of the patches is again touch sensitive. This will give you an idea of the effect of the level of the modulator on the tone of the patch. As before, the envelopes are very basic (the operator level does not change over time).

triangle to triangle

As may be expected, when a triangle modulates a triangle, the tone becomes sharper and more "plucked". As the modulation level increases or decreases, the tone takes on the quality of a sound where the pulse-width of the oscillator is being modulated. This could provide a quite usable musical tone.

square to square

As the square wave modulates another square wave, the resulting sound becomes thinner and sharper. At higher ends of modulation, noise is introduced.

saw to saw

As the sawtooth wave modulates another sawtooth wave, the sound gets noisier and quieter. From a practical perspective, there is little to commend the use of this combination.

brown to brown

Your ears may be better than mine and you may have a better listening environment, but to my mind there is no significant difference in tone when brown noise is modulated by brown noise.

Practical uses for the same modulator and carrier

As can be heard the range of musical options that are available when the same wave is used for both the modulator and the carrier in a simple FM stack is far more limited than may at first be expected. For this reason, the rest of this book will use sine waves as FM operators (except in one or two cases), however, this should not discourage you from developing a greater understanding of the sonic possibilities offered by different waves.

All of these examples have been illustrated using Rhino, there is no reason why you shouldn't create similar sounds with Z3TA+. Equally Z3TA+ offers different options from Rhino – for instance different factory waves and more wave shaping options – there is no reason why you should not use these changes for creating FM tones.

1:1 ratio – FM using parallel operators

This group of patches illustrates the tones that are available when parallel operators are used. For these examples the modulator(s) in each patch is/are touch sensitive. This will give you an idea of the effect of the level of the modulator(s) on the tone of the patch. As always, the envelopes are very basic (the operator level does not change over time).

For reference, you may want to listen to **sine to sine** before you listen to these patches.

sine2 to sine

Sine 2 to sine is built around one sine wave with two parallel modulators at the same pitch as the carrier.

Two (equal pitched) sine waves both modulating another sine wave at 50% will give the same sound as one sine wave modulating another at 100%. The advantages of having parallel modulators are:

- the carrier can be modulated to a greater extent than is possible with one modulator (thereby giving the possibilities for a brighter tone still)
- the modulators can be tuned differently, giving greater sonic options (this is discussed in greater detail below), and

- the different modulators can have different envelopes allowing them to control the carrier in different ways at different stages of the note – often different envelopes will be used when the modulators are tuned differently.

sine to sine2

Sine to sine2 is a basic parallel carrier arrangement where one modulator modulates two carriers.

There is little tonal difference between **sine to sine** and **sine to sine2** – the key difference is the increase in volume. Parallel carriers are most useful where the carriers are tuned to different frequencies (for instance, an octave apart).

1:1 ratio – FM using cascading operators

Moving on, let's look at some cascading operator patches and also consider the role of operator levels in cascades. Again, the envelopes used here are very basic (the operator levels do not change over time).

sine to sine to sine

This first patch is the most simple cascade – operator three modulates operator two which then modulates operator one which then feeds the output.

Operator two and operator three are both velocity sensitive. Therefore, as the velocity increases, operator two will modulate operator one (the carrier) to a greater extent. However, operator two will also have its wave shape changed by operator three so as the velocity increases, two factors are changing the tone.

At the maximum extent of modulation, the tone is brighter than **sine to sine**. This sound could perhaps be used for a clavinet or a bass type patch. In the mid range, there is a woodwind type tone, reminiscent of a clarinet or an oboe.

sine to sineX to sine

With this second cascade patch, the effect of operator two on operator one is constant and is not controlled by velocity (although operator three is still velocity sensitive). What changes is the wave shape of operator two which is controlled by the extent to which it is modulated by operator three.

At lower volumes, the sound is the same as that produced by **sine to sine** at maximum velocity (since in this case operator two is not being modulated). The effect of fixing the relationship of operator two to operator one is to give this patch a generally brighter tone which is controlled by velocity

sineX to sine to sine

With this third cascade patch, the effect of operator three on operator two is constant and is not controlled by velocity. However, operator two is controlled by velocity and so velocity changes the extent to which operator one is modulated by operator two.

In the mid range this patch has a more woody tone than **sine to sineX to sine**.

Sonic nuances of the cascade patches

At maximum modulation (ie at maximum velocity) all three of these patches give the same tone.

However, at lower volumes, **sine to sine to sine** gives the least bright sound (since both operator two and operator three will have their levels reduced. While **sine to sine to sine** may appear to have most flexibility as two operators are controlled by velocity, it cannot reproduce some of the mid range tones of the other two patches and so the velocity control (which could equally be any other form of control such as keyboard scaling) works to the detriment of the sound.

The last two patches illustrate how fixing the relationship between operators at different parts of the cascade chain can produce markedly different sounds.

More unusual FM

At this point I want to look at a few examples of the more unusual applications of FM. None of the following four patches uses any form of velocity or key scaling.

mock sawtooth

It is possible to create the sound of a sawtooth wave with a sine wave and some FM. You can do this by setting an operator with a sine wave and feeding the operator back into itself – you will see that I have set a value of 57. If you want to compare this constructed sawtooth with the real thing, in **mock sawtooth** oscillator two has been set with a sawtooth wave (and muted). You will hear that the constructed version is quite close.

I think it is unlikely that you will want to use FM to create a sawtooth wave. However, you may want to be aware of this possibility. More likely you may find it preferable to replace a sine wave creating a sawtooth with a sawtooth wave.

sort of square

In a similar manner to **mock sawtooth**, you can create a sort of square wave sound from sine waves. To achieve this, the modulator should be set an octave above the carrier and the modulation should be set to full. Finally I set the feedback in the modulator to 21. Again, to compare this constructed wave with the real thing, in **sort of square** I have set up oscillator three with a square wave and muted it. You will hear that the real square is a bit brighter than this constructed wave.

We have now looked at two recreations of standard synthesizer waves. However, please do remember that at the end of the day this is rather pointless exercise –

whether or not we can create a square wave or a sawtooth wave is not significant. What is important is whether any sound will work in the patches we will build.

many mods

We have already mentioned parallel modulators, and will discuss this idea further when we look at patch building later in this chapter. **Many mods** is an example of a carrier being modulated by five operators all of the same frequency. In essence, this is the **sine2 to sine** patch taken to the extreme.

I think the most noteworthy thing about this patch is how unnoteworthy it is. I cannot see any practical situation where you are likely to try to construct a patch of this manner.

many carriers

Many carriers takes **sine to sine2** to the extreme: operator six modulates operators one to five which are all of broadly the same pitch. You will see that each operator has been slightly detuned. This has the effect of thickening the sound a bit, but it also has the side effect of creating phase cancellations which lead to the fluctuating volume you will hear.

If you have the patches you will see that some feedback in operator six has been muted out. If you right click this to activate the feedback, you will hear a change in tone – you will also hear that the effect of the volume fluctuations becomes more pronounced (and more intrusive).

Again, this patch is more important as an example of something you probably don't want to bother pursuing as it produces an uninteresting sound.

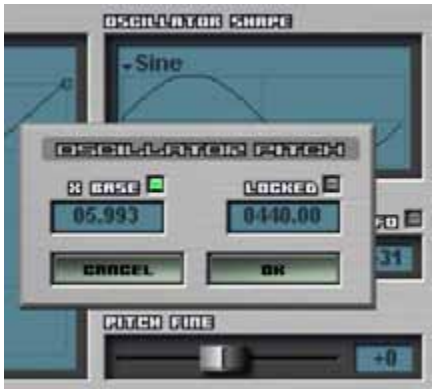
Varying carrier : modulator ratio

We have already touched on the effect of having the modulator and carrier at different frequencies. At this point, I feel I am already running out of ways to describe the different nuances which arise with different modulator/carrier relationships. For this I apologize, however, I do feel we are reaching a point where words alone cannot convey sounds. If you haven't done so already, I suggest you get hold of Rhino (or a demo of Rhino) and work through some of the examples given below. It will yield far better results than reading this book alone.

This section describes a range of sounds that can be achieved with a simple FM stack where the modulator is at a different pitch by reference to the carrier. There are also a few parallel operator examples. I should point out that the carrier/modulator pitch intervals illustrated below are not the only options, and there is no reason for you to stick with the semitone intervals that I have used.

On the subject of tuning, I have described the interval between the modulator and the carrier in terms of semitone steps. I have done this because it is the most immediate way to describe the interval. Many other people talk in terms of carrier : modulator ratio (or C:M ratio) where the modulator frequency is a multiple of the carrier's frequency, so for instance:

- with a 1:2 ratio the modulator is 12 semitones higher than the carrier (the modulator would be twice the frequency of the carrier)
- with a 1:3 ratio the modulator is 19 semitones higher than the carrier (an octave plus a fifth – the modulator would be three times the frequency of the carrier)
- with a 1:4 ratio the modulator is 24 semitones (or two octaves) higher than the carrier.



I think you get the idea. If you right click on the "pitch coarse" slider in Rhino it will display the pitch as a multiple of the base frequency – this is an easy way to set a ratio as you are working with frequency ratios rather than semitones. This is also a more convenient way to sweep through frequencies without being bound by the semitone steps of the coarse pitch slider.

You will notice that the integer ratios give a cleaner (less clangorous) form of FM. In particular you will find that 1:integer C:M ratios will not give you that classic metallic sound.

Let's look at what can be achieved in a bit more detail. For these simple FM patches, there is no velocity sensitivity.

Sine -48 to sine

In **sine -48 to sine** the modulator is pitched considerably below the carrier – in this instance it is pitched four octaves below, or if you want to consider the relationship of the frequencies, the carrier is approximately 16 times the frequency of the modulator. In this case, the modulator is acting more as a fast LFO than as an FM modulator. As would be expected, the result sounds more like very fast vibrato rather than a usable musical sound.

Sine -36 to sine

When the modulator is pitched three octaves below the carrier (which gives a C:M ratio of 8:1) a usable tone starts to develop. The tone is engine like in the lower ranges and almost has a vocal quality in the upper ranges. The pitch of the note is low in comparison to the carrier.

sine -24 to sine

At two octaves below the carrier (C:M ratio of 4:1) a richer tone emerges which has more weight than the tone produced by **sine -36 to sine** or **sine -12 to sine** (which is considered below).

sine -19 to sine

At nineteen semitones below the carrier the C:M ratio is 3:1. The tone of this combination is very similar to that given by **sine -24 to sine**.

sine -12 to sine

As mentioned above, **sine -12 to sine** (C:M ratio 2:1) has a thinner tone than **sine -24 to sine**, but it does have a thicker tone than **sine to sine**.

However, if you just compare the sounds (of **sine to sine**, **sine -12 to sine** and **sine -24 to sine**) by playing the same note you may give yourself an unrealistic impression of the sonic differences. Play **sine to sine**, then play **sine -12 to sine** one octave higher and then play **sine -24 to sine** another octave higher still. You will still hear differences but they will be less pronounced.

sine to sine

We have already looked at this patch (and have already considered many C:M 1:1 ratio patches) – a tweaked version (to remove the velocity scaling) is included here only to give a comparison with the other modulator/carrier relationships. When compared to the earlier tones (ie with a C:M ratio in the format X:1) this patch sounds quite thin. When compared to some of the later tones discussed here (ie with a C:M ratio in the format 1:X) this patch sound quite dull and uninteresting.

sine +5 to sine and sine +7 to sine

OK. This is where things start to get metallic and my inability to explain fine sonic differences in words becomes even more noticeable. This is also where all of the patches will sound quite sharp in the top octaves of your keyboard.

With both of these patches, the sound is brighter than **sine to sine** and there is a distinct metallic edge to the sound. As would be expected, **sine +7 to sine** is brighter than **sine +5 to sine** and has more of a metallic edge. However, for both patches the sound is cohesive: you cannot hear a separate sine wave and metallic element.

We are using a different type of carrier : modulator ratio here. For the first time, the modulator is pitched above the carrier. However, more significantly we are not using an 1:integer ratio:

- **sine +5 to sine** has the modulator pitched five semitones above the carrier – this gives a frequency equivalent to 1.33 x the carrier frequency which could be expressed as a C:M ratio of 3:4, and
- **sine +7 to sine** has the modulator pitched seven semitones above the carrier – this gives a frequency equivalent to 1.5 x the carrier frequency which could be expressed as a C:M ratio of 2:3.

Another issue to note here: the pitch of both of these patches is determined by the modulator, not the carrier. The modulator will determine the pitch when the C:M ratio is less than 1:2. When the C:M ratio is equal to or above 1:2 (ie the modulator is at least twice the frequency of the carrier) then the carrier will determine the frequency.

sine +12 to sine

Sine +12 to sine uses a modulator an octave above the carrier, this gives a C:M ratio of 1:2. As you would expect with a 1:integer ratio, the sound is not particularly metallic but more resembles a (slightly dull) square wave. If you look at **sort of square** you will see that these two patches have a very similar architecture.

sine +16 to sine

If you weren't convinced by **sine +5 to sine** and **sine +7 to sine**, this is where you can have no doubt about the ability for FM patches to have a metallic quality.

For this patch the modulator is pitched 16 semitones above the carrier: this gives a C:M ratio of 2:5.

sine +17 to sine

If you need more convincing about the metallic qualities of FM, then **sine +17 to sine** should do it.

With the modulator pitched at 17 semitones above the carrier, this simple FM stack has a C:M ratio of 3:8.

The reason I am stressing the metallic qualities at these frequencies is that with a large gap between the carrier and the modulator it becomes far easier to generate a metallic sounds. However the timbre of the metallic sound becomes more shrill and piercing. With these lower ratios a clearer, more usable sound can be obtained.

sine +19 to sine

With **sine +19 to sine** we have a C:M ratio of 1:3. As with all of the 1:integer ratios, the tone of this patch is free from overtones and has no metallic element to it. However, it is quite a sharp tone, which may sound a bit thinner than **sine +12 to sine** (the previous 1:integer sound).

sine +21 to sine

With the modulator pitched 21 semitones above the carrier, **sine +21 to sine** has a C:M ratio of 3:10.

To my mind this is the first "bell"-like tone, rather than metallic tone. You may disagree. Even if you agree that this patch's tone has a bell like quality, that does not necessarily mean it will be ideally suited for the creation of bell like patches.

sine +24 to sine

Hopefully you're getting the idea now about what to expect. **Sine +24 to sine** has a modulator that is pitched two octaves above the carrier. This gives a C:M ratio of 1:4. As you would expect, the tone is bright but does not have the overtones that you associate with a metallic sound.

sine +31 to sine

Passing over the 1:5 ratio, **sine +31 to sine** which gives us a C:M ratio of 1:6. Like the 1:4 ratio (and the 1:5 ratio which could be achieved by tuning the modulator to 28 semitones above the carrier), this sound is progressively more piercing. You may also describe this as sharper, harsher or thinner.

To my ear, and again you may disagree, there are two elements to the sound from this patch – the sine wave element and a distortion (almost metallic) element.

sine +36 to sine and sine +48 to sine

When I said that 1:integer ratios don't have a metallic tone, I lied. As you can hear from these two patches, it is possible to hear a metallic tone in both of these patches. However, we have reached high C:M ratios here (1:8 and 1:16, respectively). Most significantly, the difference in sonic terms between these sounds and the sound that can be achieved when using ratios that lie between these two ratios is comparatively subtle given the intensity of the sound.

sine +21 and sine +33 to sine

sin +21 and sine +33 to sine is the first example in this group of a patch with parallel modulators. The two modulators are pitched:

- 21 semitones above the root (ie a C:M ratio of 3:10), and
- 33 semitones above the root (in other words a C:M ratio of 3:20).

As you can see one modulator (operator three) is an octave above the other (operator two), and so has a frequency which is twice that of the other. While the sound of this patch is quite sharp and very bright, the two modulators work sympathetically to develop the tone of the patch.

sin +21 andto sine +21

sine +21 andto sine +21 uses parallel modulators and a cascade as follows:

- operator two modulates operator one
- operator one is also modulated by operator three
- operator three also modulates operator two – this last modulation means that the effect of operator two on operator one is to change its wave shape so operator one is not being modulated solely by a sine wave.

You can hear the effect of the cascade by right clicking in the matrix where the oscillator two column intersects with the oscillator three row – this will "gray out" the number and will cut the feed from operator three to operator two. The effect of operator three on operator one will remain.

At lower ranges, this patch sounds warmer with the cascade in place. This is perhaps counter-intuitive. However, in the higher ranges the sound is distorted when the cascade is in place (as would probably be expected). For this reason it would probably be wise to use some key scaling to cut back operator three in the higher ranges.

Envelopes

All of the sounds so far have been produced without using envelopes – the note has been either on or off and the modulator has had a constant effect on the carrier after the key has been struck. This gives a consistent, but harsh and uninteresting sound. The real magic of FM arises when one operator changes in level. Here are a few examples (ranging from bad to good) of how envelopes can be used when creating FM sounds.

level drop

This patch, **level drop**, illustrates the effect of the level of the modulator falling over time.

Operator one is modulated by operator two. The C:M ratio is 1:4 (the coarse pitch for operator one is set to zero and the coarse pitch for operator two is set to +24). The modulator is controlled by a volume envelope which opens at its maximum when the key is struck and then falls to zero over a period of about 10 seconds.

As the key is struck the modulator modulates the carrier to the maximum extent. This gives the same sound as was produced in **sine +24 to sine**. However over time the tone softens and tends to a sine wave. This is unsurprising – the effect of the modulator drops over time. When the carrier is no longer modulated, all you will hear is a sine wave.

This mimics natural instruments which tend to a sine wave as they sustain and their overtones decay.

pitch drop

This patch, **pitch drop**, illustrates the effect of the pitch of the modulator falling over time. You will hear that its effect is more extreme (and less controllable) than the effect produced by **level drop**. The levels of both operators are stable for this patch.

As before, operator one is modulated by operator two. The coarse pitch for operator two is set to -48 giving an apparent C:M ratio of 16:1. However, operator two has its pitch modulated by a pitch envelope (check out the Rhino manual if you want to read more about this). The pitch envelope is very much like the level envelope used in level drop – the pitch drops over a period of about 10 seconds. I have also set the pitch envelope slider to the maximum.

When the note is struck, the modulator is pitched up to the maximum extent, so instead of being modulated by a low note as may be suggested by the modulator's coarse pitch, the carrier is modulated by a very high note – this gives a fairly bright sound.

The pitch of the modulator then falls over the next 10 seconds, finally reaching a point where the note is so low it ceases to have an effect on the carrier (and only the sine wave of the carrier will be heard). As you hold the note, you may well hear several notes descending in pitch as the C:M ratio relationship continues to shift.

Try playing a chord – the initial sound is really quite wacky.

You may find using pitch envelopes difficult if you are programming sounds to be used in a musical context. However, if you are programming sound effects, this sort of sound may be ideal.

pitch and level drop

If that last example wasn't enough, **pitch and level drop** illustrates the sound if the pitch and level of the modulator drop simultaneously. This patch has the same setup as **pitch drop** but adds a level envelope to the modulator too.

The sound is similar to **pitch drop**. However as the modulator has less effect on the carrier over time, the effect of the lower pitch of the modulator becomes less dramatic (for instance you don't get the "echo" or "triple falling" type effect just before the modulator ceases to have effect.

sine +17 to sine w/env

Let's move on a step and look at two musical examples.

For this first example we will take **sine +17 to sine** that we used earlier and tweak the level envelope of the modulator. The modulator envelope has been set to open to its full extent immediately. It then decays over about four seconds with a curvature of about 25, so it looks like an exponential curve – it falls very quickly at the start of the decay but the rate of decline levels out: think of a piano's volume envelope and you will be getting close.

Without the envelope the patch sounds like a metallic noise. With the envelope it is transformed into a simple bell. The effect of the envelope is to modulate the carrier to the maximum extent for a very short period and then to remove this modulation source very smoothly.

However, as the carrier has a basic envelope, the sound can sustain infinitely which detracts from some of the bell quality of the patch.

sine +21 to sine w/env

Sine +21 to sine w/env takes **sine +21 to sine** we used earlier and adds an envelope to the modulator and an envelope to the carrier. The envelopes are very similar to the ones used for **sine +17 to sine w/env**, but with one difference. In the modulator envelope a "hold" stage has been added so that after the attack has reached the maximum level, the envelope stays open at its maximum level for approximately 15ms before it begins its decay phase.

The sound of the patch with the envelopes added is classic 1980s DX bells.

The effect of adding the hold stage to the modulator's envelope is to give a bit more impact to the attack of the note. The effect is subtle, but is noticeable.

In the lower key ranges this patch feels to detuned, so what I have done is a bit of key scaling on the modulator to reduce its effect for the lower keys. This makes the sound less bright but it also makes the sound cleaner.

Slowing the modulator envelope

Before we move on I want to cover one last aspect: slowing the modulation envelope with reference to the carrier envelope.

There isn't an example patch here, but these techniques are used in **synth brass** which is described below under "Building usable FM patches". As the patch name may indicate, the techniques can be used to create brass sounds.

If you create a simple FM stack (C:M = 1:1) and set the carrier with a fast attack and set the attack for the modulator to be slightly (and only slightly) slower – for instance 5ms slower – the effect of the delay will be imperceptible if the operator is listened to in isolation. However, the interaction between the two operators works in such a way that the "spit" in the attack of a brass note is created. The brass effect can be enhanced by adding a bit of feedback to the modulator.

If the envelope's attack is slowed further then the modulator's attack can be heard – the spit is lost along with the brass effect. If the modulator's envelope's attack is increased then the spit is also lost.

A brass tone can also be accomplished with a single operator by slowing the attack and adding a bit of feedback to the signal. This gives a more muted tone.

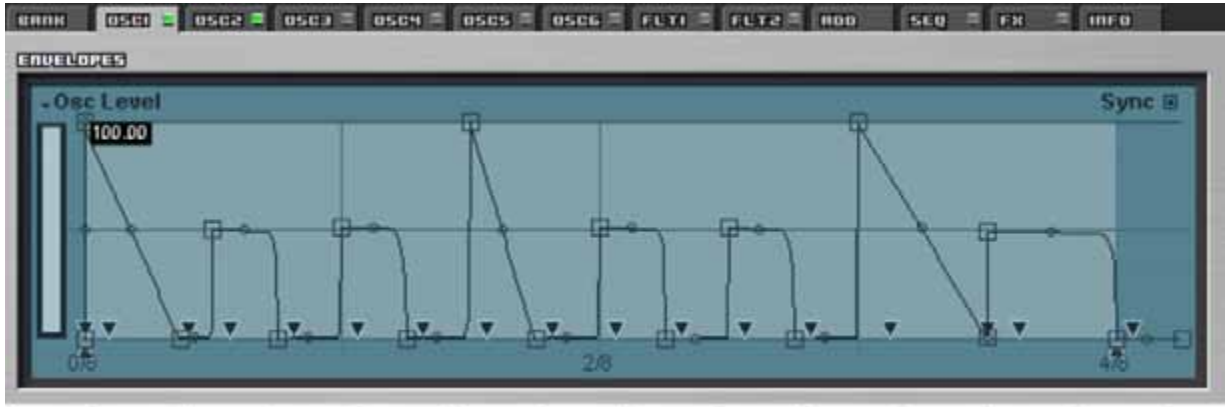
Building usable FM patches

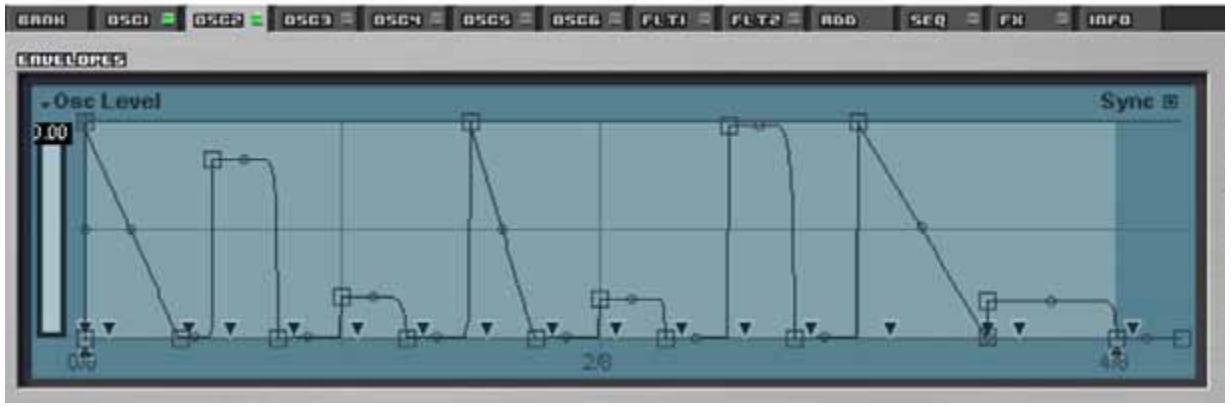
To round off this chapter we are going to build some patches – you can find these patches in the patch bank under the heading "first patches". All of these patches are intended to illustrate:

- how you can bring the various elements together to create a sound which is usable in a musical context, and
- how to move from getting a sound at random and calling that a patch, to creating sounds by design that you need for a specific project.

As with all of the patches in this book, my aim is to show how these sounds can be created from a clean slate rather than to illustrate how to reverse engineer a patch. Many decisions I have taken when designing the sounds are a matter of taste – you many not agree with my choices. That is fine: you will need to make decisions based on what works for your particular track.

These patches are presented in a finished state. To explain how they are constructed, the first step will be to switch off the modulators and most of the carriers so that we can consider the individual ingredients.





pulsing square to square

This first patch takes advantages of Rhino's flexible envelopes to create a pulsating rhythm. Other than that, the patch is quite straightforward using a simple FM stack of two square waves (with the modulator also having some feedback to brighten the sound. If you play more than one note with this patch, be careful to synchronize all of the note starts or the rhythm will get fairly complicated (and whatever you do, don't try to play arpeggios).

Take a look at the images of the two envelopes (above, right).

These envelopes create a quite simple rhythm. The first was created as a one bar rhythm with beats on the eighth notes. The first, four and seventh notes (ie the 1, 2 and, 4 beats) had their volumes emphasized. The decay on the emphasized beats was made more staccato than for the non emphasized beats – for the non-emphasized beats, the gate time (ie the length for which the note sounds) was generally set to be around half of the note's length. This gives the basic pulsating rhythm.

This envelope was then copied to operator two and the level of each beat was then tweaked so that the tone shifts (slightly) as the rhythm plays.

The only velocity scaling that is used is in operator one. The effect is to control the volume of the patch with velocity. Velocity has no effect on the tone of the this patch.

…. to ruin

This patch is a take on the classic FM electric piano sound. You will hear that at lower volumes it is quite close to the classic sound, however at higher volumes it is brighter with more percussive bite.

For those of you who do not have the patches, you are going to need to set your envelopes to look like the following image (note that in the patch each envelope is individually set, for instance, operators five and six have decay times of around four and two seconds respectively).

You will also need to set the matrix to look something like the image below.

Finally set the operators to the following coarse pitches: operator one: -12, operator two: -12, operator three: -12, operator four: 0, operator five: -12, operator six: +31.

Note that all of the carriers (operators one, three and five) are tuned to the same pitch (-12).

There are three elements to this patch:

- the "bell" sound, and
- two parts to the sustain portion of the sound
  - one part is more metallic
  - the other more wooden

  together they all blend to form a cohesive sound.

Let's look at the elements individually.

First the bell tone. If you mute the outputs of the operators one and three, you will hear the bell tone on its own. This tone is created by operator six modulating operator five with a C:M ratio of (approximately) 1:11 – I didn't use a scientific ratio here, I tuned operator six until it sounded right.

The output of the carrier is set to 45 – this is the level that sounded right to my ear when mixed with the other carriers. However, the keyboard tracking (for operator five) has also been tweaked so the volume reduces slightly in a linear manner as the pitch of the note increases. Both operators five and six are fully responsive to velocity information, so not only does the volume of the bell change with velocity but also the quality of the tone changes, becoming brighter at higher velocities.

The bell element has a comparatively short decay time. To create the full sounds it is necessary to add the sustain portion. The tone I am after is quite complex and is very velocity responsive: at lower volumes it could be characterized as being more woody, at higher volumes more metallic. Accordingly there are two elements to the sound.

Turning to the metallic sound first, unmute operator three's output and mute the output from the bell tone. Now also mute the modulators of operator three and mute the feedback in operator four.

You will now have a sine wave which we want to convert into a metallic electric piano element. The basic tone is created by operator four modulating operator three with a C:M ratio of 1:2. Operator four is fully velocity sensitive so the tone is effected by velocity.

If you listen to this simple FM sound on its own, it doesn't sound that metallic – remember this is the sustained part of the metallic sound. To give the sound more bite and a metallic attack portion, operator six modulates operator three slightly – you will hear that the sustain section appears to sound more metallic when the new attack portion is added although there is no change to the sustain sound.

Finally, the feedback in operator four is increased. In isolation the feedback does not add a particularly pleasant quality to the sound. However, when the complete patch is played it does give some emphasis to the metallic element without changing the tone of the patch much.

Even at the lowest velocity levels I want some of this element to be present, so the lower end of the velocity scaling is set to around 20. In the context of the patch as a whole, I set the level of operator three to 67.

Lastly we will look at the wooden element which is created by operator two modulating operator one (C:M is 1:1) with a bit of feedback being added to operator one. Operator two is fully velocity sensitive. Like operator three, operator one has its velocity range slightly restricted.

The character of the two sustain elements when played together gives a broad range of tone from wooden to metallic. At higher velocity ranges they both take a more metallic tone. At lower ranges, the effect of the modulators is less pronounced and so the warmer tone can be heard.

Bringing all of the elements together gives a cohesive electric piano sound which does not have the feel of three elements glued together.

You may also notice that this patch uses one of the FX units: the rich chorus unit. This gives the patch a softer sound but with a bit more sparkle (and to my mind makes the sound slightly less digital).

You may note that operator four has been set to modulate operator one. This modulation is muted but is there as a further option. To my mind bringing in this element makes the tone too sharp. However, you may disagree or find a context where this tone may be appropriate.

As an alternative patch design, try the following. Mute output of operator six and mute all the modulators of operator one. This gives an alright but not overly interesting sound. Now increase the modulation effect that operator six has on operator one. Increase the level so that operator one is about 70% modulated by operator six. We now have a very usable four operator FM piano.

My opinion – and you may disagree – is that it is a fairly pointless exercise to create an FM electric piano patch from scratch: I have done it here to illustrate the general principles of FM patch design. There are many good patches out there (the internet is your friend). Do a search, audition some patches and then tweak your favorites to respond in a manner that is appropriate to your playing style. This is one areas where I think originality could lead to a lower quality result.

organized

Organized creates an organ type sound by using five parallel carriers which are all modulated (to a greater or lesser extent) by the sixth operator.

For this patch, all of the envelopes are simple on/off affairs. For those of you without the patches, this is how to set up the sound:

	osc 1	osc 2	osc 3	osc 4	osc 5	osc 6
coarse pitch	-12	0	+12	+19	+24	-12
fine pitch	0	0	0	-19	0	-10
modulation amount by oscillator 6	100	100	15	14	11	0
output level	100	100	100	27	35	0

If you are familiar with drawbar organs (or their software incarnations such as LinPlug's daOrgan or Native Instruments' B4) this patch will make a lot of sense to you. In this patch:

- operator one acts as the root
- operator two is the second harmonic (ie double the frequency of the root)
- operator three is the third harmonic (three times the frequency of the root)
- operator four is the fourth harmonic (four times the frequency of the root)
- operator five is the fifth harmonic (five times the frequency of the root), and
- operator six is the modulator with a C:M ratio of 1:1.

Operator four has been detuned slightly to add a bit of movement to the sound: given the low level of this operator, the effect is quite subtle.

While operator one acts at the root, and operator two is tuned to the second harmonic, this relationship changes when the modulation is introduced. The effect of the modulator on operator two is that it sounds at the same pitch as the root. In practice, playing just operators one and two together (both with modulation) gives quite a rich tone.

Without the FX unit the sound is a bit intense. Switching in the FX makes the sound smoother and more organic – unsurprisingly, the rotary speaker simulator helps the patch to sound like an organ played through a rotary speaker. The chorus unit has also been added to soften some of the effects of detuning earlier in the sound chain.

wind instrument

Wind instrument is designed to recreated some of the tone of a wind instrument: it is not intended to replicate the precise characteristics of any specific instrument.

There are two main elements to this sound:

- the breath noise – which is added in several places, and
- the sustain tone.

Both elements of the sound are intended to be velocity sensitive – the breath noise in particular is intended to be noticeable at higher velocities.

As a first step, mute all of the outputs except for operator one. Also mute all of the modulations. You will now hear operator one alone. If you look at the envelope you will see that the operator has a fairly quick attack (about 3ms) and then a fairly quick (exponential) decay to its sustain level. For the sustain portion of the envelope you will see that the level varies slightly to give a tremolo type effect.

Operator three and operator five are both noise sources – white and brown noise respectively. If you look at the envelopes you will see that operator three (white noise) gives a pulse of noise – it fades in and out over approximately 5ms. By contrast, operator five has an envelope which resembles operator one's envelope (in broad terms). The two noise sources are used in different ways in this patch:

- white noise (which is the brighter) exaggerates the breath noise at the start of a noise, where
- brown noise is used to give coarseness to the note's tone to resemble the quality of a woodwind instrument.

Unmute operators three and five as modulators of operator one and unmute the output of operator three. Listen to the effect of the noise which at this stage will have a disproportionate effect on the sound of the patch.

Now mute the noise sources and unmute the modulation of operator one by operator two and unmute the feedback on operator two. You will hear a sustained woodwind sound – you will also hear that the level envelope gives the effect of breath at the start of a note. There are two envelopes affecting this sound: operator one's envelope which we have already discussed and operator two's envelope.

Operator two's envelope has an attack and sustain which is similar to operator one's. However, as can be seen in the image below, during the decay phase its level falls to zero and then increases again to the sustain level. In isolation this effect may be too subtle to hear. However, when listened to as part of the whole patch, the effect is to allow the sound of the noise sources to be heard more distinctly.

This operator also has an element of velocity scaling and key scaling – this time gently cutting the volume of the operator as the velocity increases and cutting the volume quite sharply above note F5. Both of these scalings have been designed to allow the "chiff" of the woodwind to come through more clearly.

OK, mute everything apart from the output to operator four. This operator is intended to thicken up the sound from operator one and so it is pitched 12 cents below that operator. It also has a much more simple envelope: a slow attack and a constant sustain phase.

Operator four is modulated by two operators:

- operator two which changes the tone of the operator to be more flute like (the attack envelope of operator two is largely ignored due to operator five's slower envelope), and
- operator five which adds a bit of breath noise.

Operator four is fully velocity sensitive and has its level set at 60 which is where it sounded right to me – it gave depth to the sound without creating the impression that there were two sound sources.

Finally a very small amount of operator three (white noise) is also fed to the output just to add another shade to the breath noise.

synth brass

This patch takes a different approach to the other patches in this section in that it creates its sound by a combination of FM and subtractive synthesis with two of the operators using a sawtooth wave fed through a filter.

This patch is slightly counter-intuitive: the fatter element of the sound is created by the FM elements and the brighter elements of the sound are created by the subtractive synthesis. However, this shouldn't really be that surprising: subtractive sounds are built around very bright sounds being filtered to remove the unpleasant high frequency element. FM sounds start dull and become brighter when modulated.

Let's look at the subtractive elements first. Operators four and five both have sawtooth waves selected. These are slightly detuned and fed into filter one which has a low pass filter selected ("analog lowpass 2"). You will not hear the raw output of these oscillators in this patch.

Each of the operators is controlled by an envelope. The attack time is fast (but not zero, so there is a brass quality to the attack). The envelopes then enter a short hold stage so the envelope will be fully open for a few milliseconds. This hold stage gives the sound a bit more punch when compared to an envelope where the decay phase begins immediately. The envelope then drops to, and remains at, the sustain level. There is no velocity or key scaling applied to either of these operators.

The filter cut-off frequency is also controlled by an envelope. It also has a fast (but not zero) attack time, but there is no hold phase. The filter is velocity sensitive so it opens up more at higher velocities.

The sound so far is acceptable, but not particularly impressive. As I have mentioned earlier in the book, adding a sine wave to a patch is a simple way to add weight. Operator six does this without detracting from the essential tone of the patch.

Its time to move on to the FM elements, so mute operators four, five and six.

I am going to create an FM brass sound which will fatten up the sound when added to programming we have already completed. As I want this sound to have a fattening role, it is crucial that it doesn't have too bright a tone.

Brass sounds can be created with a simple FM stack using a C:M ratio of 1:1. The key to the brass tone is ensuring that the modulator's attack envelope is slightly slower than the carrier's attack envelope. In this patch we are going to use a parallel carrier arrangement to create the brass sound.

Operators one and three (the carriers) are set with a very basic envelope – it is essentially on/off but with the attack phase slowed slightly to give a brass like character. Both carriers are fully velocity responsive. To add some movement to the sound, carrier three is detuned slightly (-3 cents).

Operator two acts as the modulator. Its level envelope has a slightly slower attack than either of the carriers' envelopes. It also has a hold phase which, in conjunction with the slowed attack, gives some emphasis to the brass attack sound. After this the envelope falls to the sustain level. This modulator is largely (but not entirely) velocity sensitive to allow for tonal variation to be controlled by velocity.

To give a tonal variation between the carriers, different amounts of modulation are applied – operator one's modulation is set to 100 and operator three's to 59. To give the modulation a bit more brightness a very small amount of feedback is added to the modulator. To further change the tone of operator three, an amount of feedback is added.

Once this FM section has been set up, add the other three operators and travel back to the 1980s.

clangy bell

As you will have realized, FM synths are great for metallic sounds. However, controlled and usable FM sounds take some work. For this patch I want to create a bell sound, but not one that is too clean: I am interested in a sound that has a good clang to it and can be used over a wide range of the keyboard. I am not concerned about the velocity scaling or the nuances of the bell, nor whether it can be played as a chord.

In essence, this patch is going to be built around two simple FM stacks creating a fairly deep bell and a third simple FM stack giving a brighter overtone.

First let's look at operator six which modulates carriers one, three and five. This modulator is working as a low frequency oscillator using a triangle wave. However, instead of having a fixed frequency for the LFO, I have set it with a C:M ratio of 3:1 – in this way the rate of the LFO is dependent upon the pitch of the note. This will ensure that the if two notes ring the LFO effects do not work in sympathy (which really wouldn't sound natural for a clanging bell). Now mute the LFO modulations until we have finished building the patch. Apart from the LFO, all of the level curves follow the familiar pattern of rapid decay that we have used earlier.

Operator one and operator five work as parallel carriers – to give their bell-like tone, they are both modulated by operator two which is pitched at a 1:9 C:M ratio. This tuning was determined by ear as was the feedback which is added to operator two to give some brightness. Operator five was then slightly detuned. If you engage only operators one and five and only modulate them with operator two you will hear the volume cancellation of this effect. This effect will be counteracted with:

- a further change to the tone of operator five (when it is modulated slightly by operator four)
- the LFO, and
- the sound output from operator three which disguises some of the volume fluctuation.

Operator three is intended to add some more brightness, but given that the patch is intended to be a "clangy bell" (rather than a crisp, clean, clinically sharp bell) it is important that this operator doesn't get too bright and so the ratios will be kept low.

For now, mute the other two carriers and unmute operator three (and mute its modulators if you haven't done so already).

Operator three is primarily modulated by operator four. To ensure that the sound does not become too shrill I selected a low C:M ratio – in this case 1:2 and added a reasonable amount of feedback to the modulator to dirty up the signal a bit. To also add a bit more bite, a small amount of operator two has also been fed in – this gives a touch more bell-like attack to the sound.

On its own the sound of operator three isn't particularly pleasing now: with the modulation it gives a buzzy almost distorted sound. However, when you add the bell sounds back (and the LFO), the sound is much more pleasing and realistic with a far more cohesive tone than may be expected if you listen to the elements individually.

The last task before this patch is finished is to sort the key tracking. The patch will benefit from some tweaking of operators four and five. For operator four its effect is cut (quite drastically below octave five and for operator five, its effect is cut equally drastically above octave seven. Without these changes the sound in the lower key ranges had too many overtones which to my mind made the bell indistinct and unfocussed, rather than clangy. Without the change in the upper key range the sound had an unpleasant, fizzy attack.

another bell

While **clangy bell** has a great sound it may not always fit with a track. First, it does not have a modern pop type sound and so perhaps it would be more useful as a special effect and secondly, it takes up a lot of the sonic range, in other words it could drown out other important aspects of your production.

If you can't use **clangy bell**, then you may need **another bell**. This is a much brighter and more modern bell sound (although its heritage is something of a 1980s

DX cliché now which may discourage its use) and you can play chords without too much dissonance.

The difficulty when trying to construct a bell is that the very essence of the bell sound is the many different frequency components which are not necessarily harmonically related. Without these element a bell is just a high pitched noise, but with too many of these elements the sound becomes difficult to pitch. While FM can readily create the elements, it is important to balance them so that the sound is useable but retains that bell like quality.

The foundation for **another bell** is four parallel carriers (operators one, three, four and five), three of which are modulated by one modulator (operator two). The four carriers are all pitched at the same level, although operators four and five are pitched 32 cents below and 16 cents above the pitch to give a bit more ring to the bell (and introduce a bit of vibrato and tremolo).

The envelopes all follow the form used in **clangy bell** – for operators one, four and five the decay time is around seven seconds. For operator two (the modulator) the decay time is around one second and for operator three the decay time is just over a second.

Operators one, four and five are all modulated by operator two. The amount of modulation varies to change the tonal quality of each of the carriers: the respective modulation amounts are 47, 74 and 100. The modulator is pitched to a level where it sounded right (ie there was enough bell sound without there being too many overtones which would have made pitching the sound difficult). In this case the C:M ratio is 1:7.

Operator three has been added to round out the sound during the attack phase. While subtle, it does give the sound a slightly fuller tone.

A further subtle change to vary two of the three main tonal elements is the additional modulation of operators one and five (by operator three). As I said, this is subtle, but it does add a slightly brighter quality to the attack of the note (which is balanced with the roundness added by operator three's raw output).

Operators one, four and five have been made fully velocity sensitive. At quieter volumes the effect of operator three, which has no velocity scaling, becomes more apparent. The modulator, operator two, has also been given an element of velocity scaling to reduce some (but not all) of the brightness of the tone at lower levels.

Finally, at higher key ranges (above octave six), the effect of the modulator has been reduced in order to tone down some of the harshness that would otherwise be apparent in the note.

bass

The skip from a bell to a bass sound may seem odd, however, the foundation of **bass** is the bell element – this is what gives the sound its bright attack. **Bass** takes the next step forward after **FM bass** which was constructed at the start of this chapter.

The challenge with any bass sound is to capture the bass element (obviously) of the sound and the brightness of the sound without these two elements being heard as two distinct sounds – in a bass patch, you should hear a cohesive sound.

So how do you get these two elements in one sound? Simple (almost) – use parallel modulators.

Open up **bass** and mute all of the modulations and all of the raw outputs apart from operator one's output. You will hear a plain sine wave with a decay of about 7 seconds. Now unmute the modulation of operator one by operator three. Operator three is pitched an octave below operator one and has the effect of giving a bass tone – there are many circumstances where this simple FM two operator sound alone would be sufficient as a bass sound. However, our goal is a fuller, touch sensitive, bass sound.

You will also hear that operator three has a much faster decay envelope than operator one. Operator three's decay lasts for just over a second so you hear the bass tone initially and as the note sustains you hear a much higher sine tone. You could extend the decay time of the modulator to give a longer deep tone. However, I don't want to for several reasons:

- First, I don't want the bass sound to take up too much of the lower frequency spectrum – with this patch design the bass element is heard when the note is struck. This design makes the bass have more impact but without making it staccato and so it should fit into a mix more smoothly.
- Also, to my ear the tone sounds duller if the envelope for this modulator is extended. As always, you may disagree with this subjective assessment.

Now mute the modulation of operator one by operator three and unmute the modulation by operator two. Immediately you will hear the bell tone I mentioned earlier (we are using a C:M ratio of 1:7 – this is the same ratio that we used in **another bell**). This is not the greatest bell tone we could design, however, it does give some brightness to the sound.

Unmute operator three so we have the parallel modulator arrangement in place. Now we have achieved our goal – a bass sound with some brightness which retains its cohesion. However it still needs a lot of work.

On a side note, it would have been possible to pitch operator three an octave lower. This would have given an even more "bassier" sound. However, this would have led to two undesirable side effects:

- first, the two elements of the sound are heard separately so the whole cohesion of the sound that I've been trying to achieve would be lost, and
- second, the bass starts to sound flabby.

The first thing I want to add to the sound is an additional bell element to keep the brightness of the attack. To ensure the patch remains as a bass patch and there are no weird metallic overtones, I have duplicated the existing bell element (operator one modulated by operator two) and pitched it an octave lower.

This lower bell element is created with operators four and five. Operator four is tuned to an octave below operator one. Operator five is then tuned to be an octave below operator two – this maintains the C:M ratio of 1:7. Operator four is then very slightly detuned to take off some of the sharpness of the duplicated bell sound.

The sound needs some rounding out – it is after all intended as a bass sound. To do this, I will bring in operator three as a raw feed. You will remember that operator three is pitched an octave below operator one and already modulates operator one to create the bass tone. Earlier I considered increasing the time of operator three's decay envelope: if we had done that it would have had a negative impact here too.

Now that the key elements are in place we can add some scaling. The only key scaling I want to add is to operator two so that its effect is reduced at higher key ranges (above octave five). A similar process could be applied to operator five above the sixth octave, however, as that range is getting out of the usual range of play, I am not too concerned to make the change.

I do want to add some velocity scaling to control both the volume and the tone of the patch. To do this I have made all of the operators velocity sensitive to a greater or lesser extent. The only place where I have done something out of the ordinary is with operator two where the velocity scaling is not linear. Instead the velocity scaling is curved so the effect of the operator becomes much more apparent at higher velocities.

You will hear that this patch can be played in higher ranges. However, if I was going use this patch in this way, I would want to tweak it a bit to get a more even response across the keyboard.

FM patches: what did we do?

It is not the style of this book to summarize the content of the chapter that you have just read. However, it may be useful to draw together some of the main design ideas that were touched upon in these sound examples:

- a C:M ratio of 2:1 (or even 3:1) will generate a very usable bass tone which can be used as a foundation for many patches
- parallel modulators can create a cohesive sound that has both a deep tone and a bell-like sparkle
- sine waves are great for rounding out sounds
- high C:M ratios (where the modulator is pitched above the carrier) are where the metallic sounds are found
- sounds from 1:integer C:M ratios produce cleaner tones
- increase modulation for brighter sounds and cascade modulation for more detailed and complex sounds
- the pitch of a patch can be affected by C:M ratios
- fixing the level relationships in cascades (giving less velocity sensitivity) can be good if you are after a specific tone
- envelopes need attention – in particular, think about
    - fast decay time to give emphasis to the transients at the start of the note
    - the attack "spit" with brass sounds, and
    - introducing a hold phase for emphasis

FM patches – what's next

While we have built a fair number of FM patches in this chapter, this isn't the end of the FM examples in this book. Chapter 8: wave-sequencing gives many more examples of FM patches. In particular, that chapter introduces the concept of FM wave-sequencing.

You should also check out chapter 9: additive synthesis, which features more examples of patches built with Rhino but this time without using FM techniques.

There are also some more FM patches (using Rhino and Z3TA+ as well as Rhino) set out in chapter 12: patch building.

Chapter 8: wave-sequencing

What is wave-sequencing?

As we mentioned earlier, in essence, wave-sequencing is the process of stepping through a series of waves in a predetermined order. This allows you to:

- create rhythmic patterns by varying the waveform, and/or
- constantly shift the tone as a note is sustained.

Wave-sequencing is not a different way to create a sound source in the same way that FM is different from subtractive synthesis. However, with a constantly shifting tone, you can create sounds which cannot be created with either subtractive synthesis or FM. For instance you can:

- cross-fade between waves to get an effect close to morphing, and
- create "infinite" waves, for instance you could line up 30 waves each subtly different so that the sound changes gradually over time, giving you a highly unique oscillator.

These two effects can be created using additive synthesis, however, that technique requires far more work to create the result which can be quickly achieved using wave-sequencing.

Basic wave-sequencing

It may be as impractical to teach someone how to "wave-sequence" as it is to teach them how to write a song – at the end of the day, it all comes down to knowing the basic techniques and a matter of taste. However, a few very simple wave-sequences may help to illustrate the power of the technique and some of the difficulties with sequences.

The most obvious synthesizer to use to demonstrate wave-sequencing is Wusikstation – it was designed to wave-sequence. However, I first want to demonstrate the technique with Rhino. Hopefully this slightly more "hand cranked" demonstration will make it easier to understand what is going on. As you would expect, you will find more sophisticated wave-sequences in the patches we build with Wusikstation.

For all of these patches, I suggest you set the tempo of your host to around 110bpm. While this first group of patches use Rhino, none of them uses any FM.

sinsqutrisaw

This first patch is a four beat wave sequence:

- on the first beat a sine wave sounds
- on the second beat a square wave sounds
- on the third beat a triangle wave sounds, and
- on the fourth beat a sawtooth wave sounds.

The construction of the patch is quite straightforward – the different waves are allocated to the four oscillators. To create the basic rhythm, the "basic 4 env" envelope in the rhythms folder was selected and its length stretched by 200%. This causes all of the oscillators to sound together on the beats.

To get the sequence to play, you can edit the envelopes by dropping the volume peaks where the wave is not wanted so that the peaks that are left play the waves in the correct sequence. In other words, the envelope in oscillator one peaks on beat one, the envelope in oscillator two peaks on beat two and so on.

This first sequence illustrates much of what is interesting with wave-sequences and much of what is disappointing with sequences. On the plus side, four simple waves have been taken and an interesting rhythmic sound has been created. However, on the negative side, the sound soon becomes quite repetitive.

It is also interesting that the comparatively weak sine and triangle waves seem to fit the sequence better than the unfiltered sawtooth wave which comes across as being slightly buzzy to my ear.

sinsqutrisaw fade

This second patch is a copy of the first, but with a few edits made to the envelopes. First the length of the envelopes is increased by 200% and secondly I have adjusted the peaks so that instead of each wave sounding separately, they cross-fade. While this gives the sound of a shifting tone rather than four distinct waves, you can clearly hear each separate wave – by no stretch of the imagination is this one constantly shifting tone.

It is also interesting that in this patch, the square wave and the sawtooth wave both seem for more dominant than they were in the more rhythmic patch.

sinsqutrisaw 8ths

So far the sequences have been fairly linear, playing one wave after another. With this next patch we will again take the first patch and will edit it to do two things differently:

- first the order of the waves will not be sequential, and
- second, more than one wave may play at once.

To liven things up a bit, I loaded the "basic 8" envelope for each of the four oscillators and I have set the levels as described in the table below.

		step one	step two	step three	step four	step five	step six	step seven	step eight
oscillator one	sine	full volume			full volume		full volume	full volume	full volume
oscillator two	square		full volume	half volume	half volume	full volume			full volume
oscillator three	triangle			full volume				full volume	full volume
oscillator four	saw	full volume		half volume	full volume	full volume		full volume	

As you can hear this gives a rhythm that is far more interesting than that in the first patch.

additive seq

For **additive seq**, I have taken six sine waves. Waves two to six are pitched a frequencies which are the multiples of the first. As a side note, these sine waves have been created using Rhino's additive wave generator – the basis of this patch will be used again in the next chapter (additive synthesis).

Once the waves are set up, this sequence then creates an arpeggio pattern from the waves.

The prime purpose of this patch is to illustrates my frustration with melody auto-play patches. While the effect is interesting, the result is highly limiting – for instance you can't play chords and (in this case) there isn't a minor arpeggio. Hopefully you will see why I am happy to have patches with rhythmic elements, but not melodic elements (and why there is nearly no mention of melody auto-play techniques in this book).

additive seq II

Moving on, **additive seq II** takes the foundation of the previous patch but creates a more rhythmic rather than a melodic sequence. This latter patch uses the same sine waves, at the same pitch, as are used in **additive seq**. However, the higher pitched waves are largely used as harmonics shaping the tone rather than as a fundamental note and so simple chords are possible with this patch.

The rhythm was created in a similar way to how the rhythm in **sinsqutrisaw 8ths** was constructed. Those of you with the patches will see that with oscillators five and six some notes are held (ie they are not just a rhythmic beat) and that some of the beats fade in (giving a "wine glass" type bell tone).

For wave-sequence patches which create rhythmic patterns, the midi part playing the synthesizer must be simple: programming the patch is more akin to programming another rhythm part. You are going to need to think how the wave-sequence will work with the other rhythmic elements (primarily drums and bass) when you are programming.

For tone shifting wave-sequence patches, players can often adopt a conventional playing technique. However, this style of patch is often more suited to sustained chords and so perhaps some aftertouch could introduce some interesting modulation effects.

More advanced wave-sequencing

There are several limitations in using Rhino for wave-sequencing. These can all be overcome by using Wusikstation which, having two wave-sequence layers each with 32 wave slots and 12 controller lanes (which can also act as modulation sources), was designed for wave-sequencing.

However, this flexibility with Wusikstation illustrates one of the key challenges with wave-sequencing: the endless permutations. Wusikstation has several hundred waves available. The permutations of these several hundred waves are endless. Once the waves have been selected, they must be sequenced – again the permutations can be quite daunting.

However, once a series of waves has been chosen, the process for selecting the waves is quite straightforward in Wusikstation (even if the permutations are numerous). If nothing else it is much easier to choose a wave in a Wusikstation wave-sequence layer than it is to draw a wave-sequencing envelope in Rhino.

The following example patches are all created in Wusikstation. The patches have all been created using the Famous Keys waves that are bundled with Wusikstation – the full waveset does not come with the demo and so if you are using the demo of Wusikstation you may find that these patches do not work as intended.

waveseq one

This first patch sequences 16 waves together. The waves are DW8001 to DW8016 which can be found in the famous keys three soundset.

The sequence has been set to step forward on each quarter note and plays wave 1 to 16 in order.

For me, the main purpose of this patch was to audition the waves and to listen to how they sound when put into a wave-sequence.

waveseq two

Waveseq two is very similar to **waveseq one** – the waves are the same and the order in which they are played is the same. There are two key differences:

- first, the waves cross-fade (in the second controller lane xFade has been set to the maximum for each step), so instead of there being a step in tone with each

new wave as in the first patch, with this sequence there is a constant shift in tone

- second, the time for each step has been increased to be equivalent to a half note. Without this increase, the fade between waves felt too abrupt.

As with **waveseq one**, the purpose of this sequence is to listen to how waves sound when put into a wave-sequence.

waveseq sustained pad

Now that the waves have been auditioned, I want to construct a pad sound. I will use some of the waves from the previous two patches to create this pad, but so that I don't have to reload the slots, I have simply copied the previous two patches and will only choose the waves I want.

The step length is set to 1/8 and as you will see, each step has its time set individually (in a separate controller lane) so that this does not become a regularly changing pattern. Also, as this is a pad, I don't want there to be any stepping so I will ensure xFade is set to the maximum for each step.

The first wave is DW8010 (which is in slot 10). This has a sort of buzzy electric piano type tone – I will use this to give the wave some brightness in its attack. As you will see later on, I have used the filter to slow the attack of the brightness and so the piano like envelope will be masked. I have set the time for this first step to three, so it will last 3/8.

To follow this wave I chose DW8004 – there was no real logic here, it just sounded right to me and tonally it fitted well after the previous wave. The time for this wave was set at two.

The next wave is DW8007 – as with all of these waves it was chosen for its tone. The time allocated to this wave is four.

The fourth wave is DW8013 and the time allocated is 10.

The final wave is DW8016. As the intention is that this wave sounds for the whole of the sustain portion of the note, the wave has therefore been included in several slots each with the maximum time selected – this way the sound will sustain rather than loop back to the beginning of the sequence.

Now that I have set the waves, I am going to add a filter to give the player a bit more control over the sound. The filter is going to do two things:

- first, it will allow the player to control the tone of the pad, particularly during the attack phase, and
- it will also act as a quasi volume control to quieten the pad during its sustain phase – you will have noticed that there is no conventional envelope settings for this sequence.

Mod env one is set up to control the filter – the modulation of the filter is controlled through the modulation matrix. The mod envelope achieves three things:

- first it makes the envelope's control of the filter fully velocity sensitive, so at higher velocity levels the filter opens more
- second, the mod envelope opens the filter slowly, having the effect of slowing the attack of the wave, and
- third the envelope closes slowly (the decay time is set to the longest possible time and the sustain level is set to zero) using the filter as a volume control. However, the filter does not fully close as the minimum amount of modulation in the matrix is set to 25.

Lastly, to add a bit of movement to the pad, I have added an LFO to give some vibrato to the sustain portion of the pad. This is done with mod LFO one which is controlled through the modulation matrix by mod env one set with a long attack time so that the effect of the vibrato can be faded in.

waveseq rhythm pad

Now that we have created a sustaining pad with a shifting tone, it might be interesting to create a rhythmic pad as a contrast.

Rhythmic wave-sequences can be quite effective for single notes. For chords (as would be typical with a rhythmic pad) the effect can be overpowering, so to get a rhythmic wave-sequence pad to work it has to be restrained. This patch has been designed to be played with chords in the mid-range of the keyboard: you will find it doesn't sound great when played outside of this range.

You will also find that this patch leaves lots of room for other elements in the production. However, it does not work well if you have a rhythm that simply plays eighth beats. To use this sequence in a track you will need to be a bit more creative in your rhythm programming.

The waves for this pad are again the same as those in **waveseq one** and as before I have only selected those that I want. However, unlike with **waveseq sustained pad**, the waves in this patch have been selected on the basis that they give a rhythmic transition from the previous wave. And of course, to keep the rhythmic effect, xFade is not used.

The other key feature of this sequence is the two silences. These are an integral part of the rhythm which you will have noticed is not a regular beat.

I won't take this pad apart and describe the individual waves in detail – those of you who have purchased the patches can take a look.

waveseq bass

We've built a rhythmic pad, so now it is time to build a rhythmic bass patch. This patch, **waveseq bass**, is built around the waveforms chosen from the same range of waveforms selected for **waveseq one**. However, for this patch, I am also going to thicken up the sound by adding two conventional oscillators – I will explain this later.

The pattern for this wave-sequence is a regular sixteenths rhythm. The waves that I have chosen were those that gave an element of drive, but in the lower regions of the keyboard.

To my ear this choice of waves gives a good rhythm, but the sound lacks real weight. The sequence is great in a certain range of the keyboard – above the range and the patch sounds a bit thin and metallic while below the range the patch sounds to muddy and ponderous. For me the way to add weight is to bring in another tone.

To add this extra weight, in oscillators one and two I called up the saw waveform – both oscillators have had their pitch dropped by an octave and are then slightly detuned by reference to each other and panned slightly to the left and slightly to the right.

However, these two saw waves dominate the sound too much and cut in too quickly. To counter this I have closed down a low pass filter for each of the oscillators. These filters are then controlled by mod env one which takes some of the bite off the filters – the envelope is set to slowly open. In the modulation matrix, the amount that the filters are opened is slightly different for each filter giving a slightly different tone for each layer. However, the amount that the filter opens is not that much – this dull tone ensures the saw oscillators add weight without becoming a dominant feature.

The last tweak I have added is to drop the pitch of oscillator one when the key is first struck and allow the pitch to rise as the note fades in. This pitch change is controlled by mod env two. The effect it gives is like an engine revving. When the oscillator reaches the right pitch, the sound of the oscillators acting together gets fatter and warmer, making an excellent backing for the rhythmic wave-sequence which sounds from the moment the key is pressed. You will find that this is the sort of patch where notes need to be held and any changes between notes should be carefully planned.

FM wave-sequencing

If you're really up for a programming challenge, then Rhino allows you to achieve something more subtle than wave-sequencing: FM wave-sequencing!! Instead of creating regular FM sounds you can sequence the modulators. This has a lot of advantages:

- you can smoothly transition from one sound to another
- you can achieve the wave-sequencing effect with sine waves alone – you don't need to use other waveforms (although that being said, if you're into experimentation, you can try FM wave-sequencing with different waves)
- you are not limited to a set number of tones where if you use Rhino conventionally, you can wave-sequence with only six waves.

With FM wave-sequencing in Rhino you are have a maximum of five modulators. However, you have control of the levels, so immediately you have more than five basic tones. Secondly, you can arrange the modulators in parallel which gives you a much wider range of tone and you can modulate the modulators – I wouldn't say you have an infinite number of tones available, however, you certainly have more possibilities than you could work through in a couple of days.

Also, with the constantly variable envelopes, you have much more flexibility in how your tone changes – for instance if you have two modulators acting at once, one could stop immediately and the other could gently fade: this sort of tonal change cannot be replicated with conventional wave-sequencing techniques. You can, of course, perform FM wave-sequencing with Wusikstation, however, its FM implementation is not as readily usable as that under Rhino.

If you want to take the concept further and are confident about addressing the tuning challenges with FM, then you can even create FM wave-sequencing arpeggios (although I have already expressed my dislike of auto-play patches).

A few examples may help to explain the technique.

FM wave-sequencing patches

If you haven't done so already, I suggest you get hold of the patches before you read this section. The patches all use quite detailed envelopes to achieve their effect and it may be quite tiresome to manually recreate this programming.

The following patches are all variations on a theme. When building these patches I was aiming to create a driving bass rhythm that needs to work in the context of a fairly sparse track – a simple drum pattern, this bass part and a lead line.

With a track this sparse, the bass needs to fill a big space. This means that the bass part will need to be (comparatively) busy and contain a fair amount of mid-frequency information. However, if the bass is to take up such a big space in the sonic spectrum, then is doesn't need really low elements otherwise it may smother some of the drums (the kick drum in particular).

Setting up the patches

The patches are built around one carrier and four parallel modulators which have individual envelopes applied to give the wave-sequencing rhythm. For the first patch the fourth parallel modulator is not included. No FX and no filters were used in the building of these patches.

- Operator one is set as the carrier. There is no adjustment to the coarse pitch and the envelope is a simple on/off setting. If you mute the modulators, you will hear a plain sine wave.
- Operator two is set at the same pitch as the carrier – this gives the tone considerable weight (although of itself, this tone is not that interesting). The fun stuff happens when the envelope is introduced into the proceedings – this is described below.
- Operator three is pitched one octave below the carrier (ie the coarse tune is set to -12). The effect of this modulator is to add depth and bass to the tone of the note. Other modulators pitched below the carrier could be used to give a fuller bass sound, however, they may also lead to tuning difficulties, so for this patch I have chosen not to include any difficult elements.
- Operator four is pitched 19 semitones above the carrier (so the C:M ratio is 1:3). The purpose of this modulator is to bring some bite to the sequence.
- Operator five is pitched 24 semitones above the carrier (giving a C:M ratio of 1:4). This adds a touch of brightness (especially in the later patches).

fm wave seq 1

We've seen the main elements for the sequence. This is how the sequence is put together.

Operators two and three are set to fully modulate the carrier, operator one. The amount by which operator four modulates operator one is set to 75. For each modulator the "basic 16" envelope was selected and then tweaked. Mute the effect of operators three and four on the carrier before continuing to look at this patch.

The effect of operator two is to give the patch tone and drive: it doesn't add much bass to the sound. The sixteenth rhythm plays fairly consistently, although on steps two and six the amplitude is reduced to 50% and on steps four and five the envelope is set to zero (so only the tone of the carrier is heard). The pattern played by operator two is quite repetitive – I felt the patch needed this to keep its drive. However, you will hear that some of the repetition is counteracted by the effect of operator four.

Where operator two is busy, operator three creates a pattern which is quite sparse, only playing on four individual steps. While only playing on four beats, there is another change to the rhythm with this operator in that some of the steps sound for the whole length of the beat (ie rather than being simply an impact).

The sequence created by operator three may be sparse, but when operators two and three work together it creates a good rhythm, but the end of the pattern is still a bit repetitive. The tone of the pattern also lacks some variation. To address these shortcomings, operator four is called in.

Operator four has five steps and some of these last for the beat (rather than just for the impact). Operator four also acts to break some of the monotony of operator two without detracting from the rhythm established by that operator by putting all of its beats (except the first) towards the end of its sequence.

The final tweak I made to this patch was to assign a user slider. I'm not a great fan of this feature if it is just linked to one control. However, it can be really useful when several parameters are controlled simultaneously by the slider (especially if the slider is then controlled by some external hardware with the midi learn function). In this case, the slider marked "intensity" controls the amount of modulation by operators two, three and four.

You will see that each operator has a different range: operator two's modulation goes from 45 to 100, operator three's from 65 to 100 and operator four's from 15 to 75. To my ear, these ranges gave the broadest spectrum of usable tones. A practical use for this feature might be at the start of a track – if you set the intensity slider to be controlled by a track envelope, then at the start of the sequence you can gradually increase the intensity, perhaps over two bars when the drums could then join the track.

fm wave seq 2

This next sequence builds on **fm wave seq 1**. The key changes are:

- operator five has been added as a fourth parallel modulator
- all modulation levels have been set to 100, and
- the intensity slider has been dropped.

In this patch, operator five adds a touch of emphasis to some of the beats. Its effect is quite subtle having more of an effect on the tone and texture of the patch rather than the rhythm. If you listen closely you will be able to hear that on one beat the operator fades in (very quickly) – again the effect is very subtle but it is almost like applying reverse reverb to one beat.

fm wave seq 3

Tonally **fm wave seq 3** is much brighter than **fm wave seq 2**. There are two reasons for this brightness:

- In operator five, the sine wave has been replaced with a triangle wave – this gives the effect of operator five more sizzle.
- Operator two (which modulates operator one) is now modulated by operators four and five. This changes the waveform in operator two that is modulating operator one and so brings more tonal variations to the pattern. In effect we're wave-sequencing twice in a three operator cascade, which has parallel modulators at its source. See figure 38 for an illustration of the signal flow.

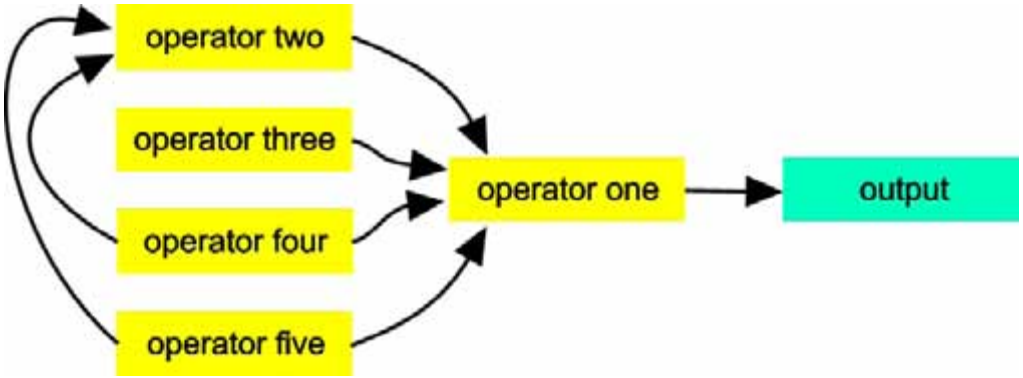


figure 38: the routing adopted in fm wave seq 3

I have assigned some sliders to allow for some tonal variations and so that you can easily hear the effect of the extra modulations.

- The slider labeled "raw 2 & 5" allows you to directly feed the raw outputs of operators two and five. I was hard pressed to hear any difference in the tone by introducing these raw waves – however, you can hear the effect when some of the other modulations are reduced in level.
- "Operator 2 mod" allows you to reduce the amount by which operator two is modulated by operators four and five. When this is set to zero, you can make a direct comparison with **fm wave seq 2** to hear the effect of changing operator five's wave to a triangle.
- "3, 4 & 5 mod 1" controls the amount by which operators three, four and five modulate operator one. If you set this to zero it is easier to hear the effect of the changes controlled by the operator 2 mod slider.

fm wave seq 4

This is the final patch in the series. It has the most tonal movement and is the brightest. However, that does not necessarily mean it would be right for your project. Within the track for which this patch is intended, the brightness works.

As you would expect, **fm wave seq 4** is a development from **fm wave seq 3**. The main changes are:

- The envelope for operator two (which gives much of the drive to the patch) has been restructured. The rhythm has been kept but with a few tweaks added here and there (especially to some of the curves). My method for deciding what I wanted to change was to listen to the sequence looping and then make a slight change where the patch felt repetitive. As always, you may not agree with the choices I made.
- Three new modulations have been added:
  - feedback has been added to operator two (which modulates the carrier)
  - a touch of feedback has also been added to operator three (which also modulates the carrier), and
  - operator five modulates operator three.

The effect of these modulations – particularly on operator two – is to give a greater range of tone shift as the sequence plays.

- The three modulation tweaks can be controlled by a new slider labeled "new mods".
- There is also another slider, labeled "raw 2, 3 & 5", which controls the raw output of operators two, three and five. As you increase this, the sound gains a slight element of presence: my preference is to set the slider to zero and just rely on the carrier to provide the whole sound.

Once you have listened to the patch, if you have the feeling that this could have been inspired by a well known keyboard player of east European origins who scored a popular 1980s US tv cop show, you would be correct.

fm no seq 4

To give you an idea of the sound of this patch without the wave-sequencing, **fm no seq 4** takes the previous patch and replaces the rhythmic envelopes with piano type envelopes. You will see that a range of sliders have also been assigned to control the tone.

FM arpeggio patches

With a simple FM stack, when the carrier : modulator ratio is higher than 1:2 (for instance 2:3, 1:3 etc) the pitch of the FM patch is determined by the carrier. When the C:M ratio is less than 1:2 (for instance 1:1.5, 3:1 etc) the pitch is more influenced by the modulator. This gives us the opportunity to create melodic figures (such as arpeggios) in FM patches and at the same time have the tone shifts offered by FM wave-sequencing. However, please remember that the perception of pitch can be quite subjective, especially when an FM wave contains many frequencies which may not be harmonically related.

It is probably easiest to illustrate the principle with a simple patch.

fm wave seq octaves

This patch is based on a parallel modulator arrangement.

- Operator two is the first modulator It is pitched 12 semitones below the pitch of the carrier, operator one – its effect is to darken the tone and to drop the pitch of the resulting note by an octave. The "basic 8" envelope has been used and this has been tweaked so that the operator kicks on the beats.

- Operator three is the second modulator. It is pitched at the same frequency as the carrier and so when this note sounds it will be an octave above the note produced when operator two modulates the carrier. Again the "basic 8" envelope has been called up, but this time the peaks have been left on the "and" beats (ie peaks one, three, five and seven have been deleted).

Hit any note and you will get that classic root/octave bass line. Find a 1970s disco track and you're made for life.

fm wave seq arpeggio

Its time to get a bit more adventurous: **fm wave seq arpeggio** creates a simple arpeggio using the same principles established in **fm wave seq octaves**.

Operator one is set as the carrier. Operators two to five act as modulators and create the arpeggio tuning as follows:

- operator two is tuned two octaves and five semitones below the carrier (giving a C:M ratio of 7:1) and acts as the root of the arpeggio
- operator three is tuned a major third above the root modulator (giving a C:M ratio of roughly 5:1)
- operator four is tuned a perfect fifth above the root modulator (giving a C:M ratio of 4:1), and
- operator five is tuned an octave above the root modulator (ie an octave and five semitones below the pitch of the carrier, giving a C:M ratio of 3:1).

Operator six is tuned to the same pitch as the carrier and only modulates it to give a slightly brighter tone to the arpeggio, in this case the modulation amount is set at 30.

Once the modulators had been set up and the matrix was adjusted so that each of the four main modulators modulates the carrier by 100. The sequence was set up with the envelopes in each operator so that the modulators play in sequence to create the arpeggio.

Those of you that have the patches can hear the results. A very basic, quite uninteresting arpeggio is played. You can make the tone somewhat more interesting by playing with the two sliders:

- "Weight" increases the modulation of each of the four main modulators by operator six. It also feeds the raw output of the four main modulators to the output. You will see that this slider operates to increase the modulation only slightly (the maximum increase is 25) but the increase for the raw outputs is much more significant (70).
- "Brightness" increases the modulation of the carrier (operator one) by operator six, and introduces some feedback into the carrier. As you would expect from the label on the slider, the effect of these two additional modulations is to make the sound brighter, however, I find the brightness works better when the weight slider is also increased – my preference is to set the weight slider to 40 and the brightness slider to 30.

A few thoughts about wave-sequencing

Wave-sequencing is somewhere between painting and sculpture – you keep adding shades and tones while chipping other pieces away. Frequently you need to stand back and look at what you're doing to make sure the overall effect is right.

It can be one of those instance gratification techniques. However, the technique does have its limitations. For instance, you can usually only use one rhythmic wave-sequence in an arrangement at any one time or else the track will just become too cluttered and confusing.

I like the technique because it can create rhythmic patterns which can only be created by using this technique: you cannot program a sequencer to create these sounds. I find the constant shifting/cross-fading techniques can generate interesting textures, however, often trying to sequence together several waves to create an interesting and cohesive texture just gets the same effect that can be achieved with conventional FM or subtractive synthesis (or a combination).

Wave-sequencing can take a while to achieve satisfactory results. However, I would encourage you to think of wave-sequences as being an integral part of your track's arrangement. If you follow this logic, then you should apply the same care to creating the sequence as you would when programming another other part within your track.

I think the FM wave-sequences add another option for creating interesting and compelling rhythm tracks. However, I am less convinced about whether the FM wave-sequenced arpeggios have much of a future. The programming of the arpeggios is long winded and tedious and as the example above demonstrates, the tunings can be quite imprecise and then the tonal variations are limited as the tuning of the arpeggio can be affected by the modulation. At the end of the day, I do wonder why you would do this in Rhino when it already has a step sequencer and pitch envelopes.

Chapter 9: additive synthesis

Additive synthesis: what and why?

We spent a lot of time looking at subtractive synthesis where you take a sound source which is rich in harmonic content and reduce those harmonics with a filter.

Additive synthesis takes the opposite approach – it takes a bunch of sine waves at different frequencies and puts them together to create a new sound. The sine waves (or "partials" as they are called – the term partials and harmonics are pretty much interchangeable) are all multiples of the fundamental frequency.

Additive synthesis requires patience to construct these new waves, but the results can be excellent. There are many advantages to this approach:

- each partial can be individually controlled
- the amplitude of each partial can be controlled over time (depending on the architecture of the individual synthesizer)
- true morphing can be achieved
- individual partials can be detuned (to introduce more metallic elements into a sound) and can have their phase manipulated.

None of these features can be created by a conventional subtractive synthesizer or sampler. In addition, some synthesizers (an example being Cameleon 5000) allow you to resynthesize sounds.

Morphing and cross-fading

There is an important difference between morphing and cross-fading.

With cross-fading there are two sound sources – the effect is achieved by the level of one source being reduced while the level of the other source is increased. It is therefore possible to hear two distinct sounds during the process.

With morphing, there is one sound source – initially it has the characteristics of the first source and at the end it has the characteristics of the second. During the morphing transition, the composition of the partials making up the wave changes. So when the sound is halfway through its morph, there is a wholly new sound rather than there being the two existing sounds as would be the case if the sounds were simply cross-faded.

In short, morphing allows for completely new sounds to be created. These sounds cannot be created by other means.

Warning!! Mathematics ahead

I promised that this wasn't a book about mathematics. It isn't and if you want you can skip these paragraphs.

I mentioned earlier that waves are made up from a combination of sine waves – different frequencies and different proportions of waves will give different results (or as we prefer to call them, different sounds). Without getting into to much detail of the mathematics this is how the three most common waves can be constructed using additive principles.

For those of you who are really keen on mathematics, you can work these formulae through and prove that the addition of these sine waves creates the resulting waves. For the less mathematically inclined (this author included), I have included graphs to show the amplitude of each harmonic within a wave – compare these graphs to some of the sine wave compositions in Rhino and Cameleon 5000 when we start creating waves from first principles.

Additive square wave

A square wave is made up of only odd-numbered harmonics (sine waves) with decreasing amplitudes in the ratio 1/n (where n is the number of that harmonic). The harmonics are therefore:

- the first harmonic (the fundamental) which has its full amplitude
- the third harmonic which has an amplitude of one-third of the maximum

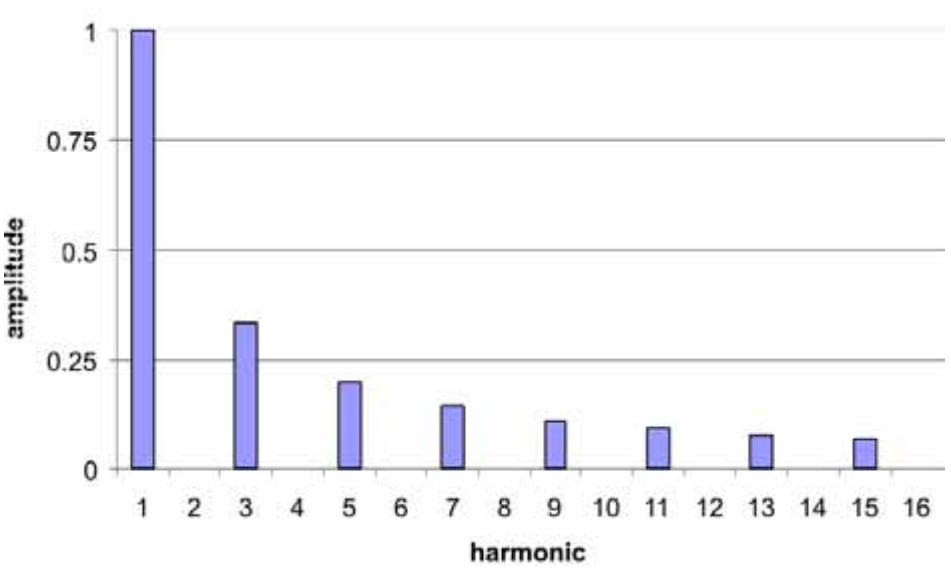


figure 39: amplitude of the first eight partials of a square wave

- the fifth harmonic which has an amplitude of one-fifth

and so on.

A graph plotting the amplitude of each harmonic against the harmonic's position would look like figure 39. However, you will note that this graph only shows the first sixteen harmonic positions where there should be an infinite number of harmonics – later on we will look at the relationship between the number of partials and the tone.

Additive sawtooth wave

A sawtooth wave has both odd-numbered and even-numbered harmonics with amplitudes decreasing in the ratio 1/n. The harmonics making up the sawtooth wave are therefore:

- the first harmonic at its full amplitude
- the second harmonic at half its amplitude
- the third harmonic at one-third of its full value
- the fourth harmonic at one-quarter of its full value

and so on.

Additive triangle wave

A triangle wave has odd harmonics with decreasing amplitudes in the ratio $1/n^2$. Therefore a triangle wave has the following harmonics:

- the first harmonic at its full amplitude
- the third harmonic at one-ninth of its full value
- the fifth harmonic at one-twenty-fifth of its full value
- the seventh harmonic at one-forty-ninth of its full value

and so on.

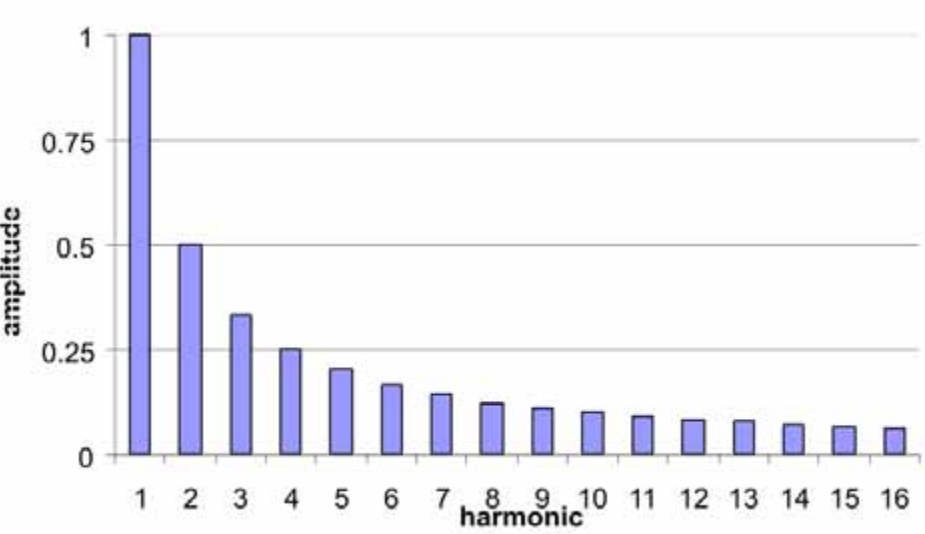


figure 41: amplitude of the first sixteen partials of a sawtooth wave

As you will read, one of the themes of this chapter is to program with your ears, not your eyes. If you look at the triangle wave graph and the square wave graph, you will see some similarities – they both have odd harmonics declining in amplitude. Now listen to a square wave and a triangle wave – the difference is far more marked than the variations in the graphs may suggest.

If you want to delve further into the science of waveforms, then go back to the internet and search for "fourier synthesis".

Difficulties with additive synthesis

Like FM, additive synthesis has a reputation for being "difficult". "Difficult" may be an understatement – additive can be potentially hugely complex. Take sound creation in Cameleon 5000 as an example:

- each wave can have 64 partials – each partial has its own level and its own tuning
- at each point in the envelope, the partial levels and their tunings can be repositioned, and

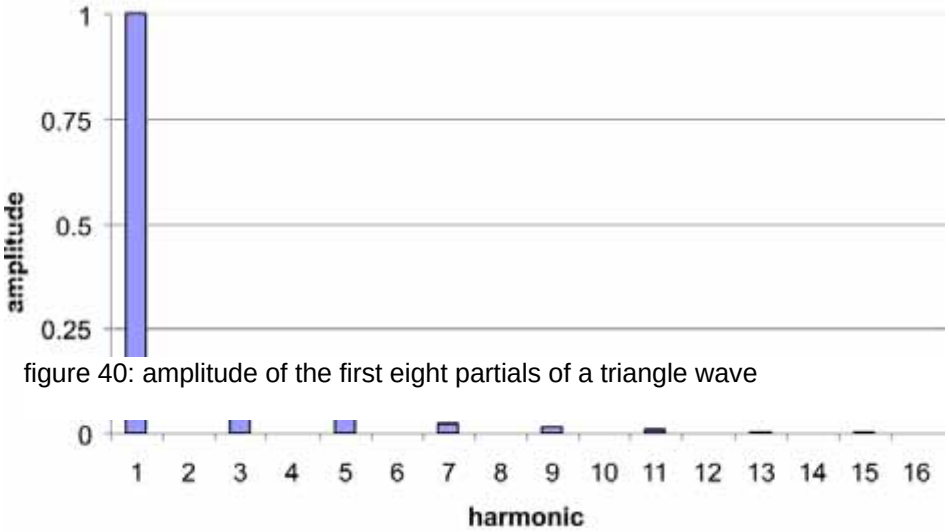


figure 40: amplitude of the first eight partials of a triangle wave

- for each wave, there is an accompanying noise source with many options for control.

To then multiply the confusion there are four oscillators which you can morph between, remembering that the sound for each oscillator is constantly shifting.

If you adjust only 10 partials at three envelope breakpoints for the four waves, and add some morphing then you will have made nearly 150 adjustments.

Before you can get to the complexity and the endless permutations, there is a far more straightforward challenge – creating an interesting sound. If you can't create an interesting sound, then the issues of how you control it over time and how you balance several sound sources are irrelevant.

Unfortunately there is no magic formula for creating an interesting sound from first principles. You really need to sit down and listen and build up some experience in creating these sounds. However, you can always import waves (with Rhino you can import single cycle waves, with Cameleon 5000 you can import multi-samples).

If you're looking for a real challenge, then try creating some FM patches with additive waves: use an additive wave as a modulator and listen to how the changes to the wave affect tone.

Example additive patches

Simple additive patches

To help understand what additive can do and the tones it can create, it may be helpful to create some sounds using additive synthesis techniques. This first group will be built using Rhino. While Rhino's additive synthesis features are nowhere near as controllable as those under in Cameleon 5000, they are still very powerful and very usable making it the ideal tool to demonstrate some of the basics of additive synthesis.

The purpose of these patches is not to create great examples of sounds that can be made with additive synthesis. Instead, these sound are here to illustrate some of the fundamentals around creating additive waves.

These patches are all created with additive waves and envelopes – no filters or FX are used here.

additive build

This first patch, **additive build**, uses wave-sequencing techniques to develop the tone of the note over time. The patch will sound rather like an arpeggio except that all of the earlier notes are held as higher harmonics are introduced to change the tone.

When you strike the note, you will hear the first oscillator – in this oscillator an additive wave has been created by choosing the first harmonic (ie the fundamental) only. You will next hear the second harmonic added – the level of this second oscillator has been set in the matrix at 55%.

The third harmonic (ie three times the frequency of the fundamental – 19 semitones above the fundamental) is the third wave to be added in the series and comes in at 40% of the level of the fundamental. The fourth wave is the fourth harmonic (four times the frequency of the fundamental or 24 semitones above the fundamental) and comes in at 35% of the level of the fundamental. The fifth and sixth harmonics then follow at 20% and 15% of the level of the fundamental.

So what have we done here? We have created a wave by layering harmonics – you can hear that as each harmonic is added the tone of the wave becomes brighter. If you want to hear the tone of the note once the harmonics have been added but without the sequential build up, you can listen to the next two patches:

- **additive build [tone]** is a copy of **additive build**, the only difference is that the envelopes are all the same and now all of the waves all play together at the same time (with the same mix levels as before)
- **additive build [wave]** recreates the tone of **additive build [tone]** by selecting the individual partials in the additive wave generator. You can hear the sound is the same as **additive build [tone]**, but you will notice that the main volume fader has been increased to balance the single oscillator against the six oscillators in the previous patch. The particular advantage of this approach (compared with **additive build [tone]**) is that only one wave is used – this uses less CPU and leaves the other five waves free to create tones (or to layer/detune the same wave in another oscillator).

So as you can hear, we have taken six harmonics and created something that sounds quite similar to a sawtooth wave. Who said additive was difficult?

You will notice if you look at the matrix in **additive build** that each of the waves is progressively quieter. However, if you listen to the waves as they enter in sequence there is little perceptible difference in volume of each of the waves – certainly oscillator six does not sound as if its volume is 15% of oscillator one's volume.

This highlights two issues:

- first, the disproportionate effect the higher harmonics can have, and
- second, it is really difficult to adjust those high harmonics – you only want to move them one or two clicks and this is difficult with a mouse and the GUI doesn't really give enough immediate feedback.

One other thing you may notice – if you listen to **additive build** (when the six oscillators are heard) and **additive build [tone]** it may not be immediately apparent that they are the same sound. With any shifting sound source it is always difficult to isolate one element of the sound for comparison.

additive square and better drawn add sq

Here are two examples to illustrate how you can spend a long time trying to make an additive wave look right when you should be listening to how it sounds.

First listen to **additive square** – this wave was created by importing a square wave. As you would expect, it sounds like a square wave. Now take a look at figure 43 and notice the harmonic composition of the waveform. You will see that the wave includes all of the odd harmonics (with the amplitudes described earlier).

Now take a look at figure 42. You will see that this square wave has a different harmonic composition – in particular, none of the partials above the 43rd partial are included. However, you will also see that the image of the square wave (below the partials) is far smoother and much more closely resembles a square wave.

If you listen carefully you can hear that this new wave has a different tone. I am not sure that the effort need to redraw the wave justifies any tonal changes – the change here is probably to the detriment of the sound.

While I can hear the difference when the waves are played on their own, I'm not sure that there would be much difference in a patch, let alone in the context of a fully orchestrated track.

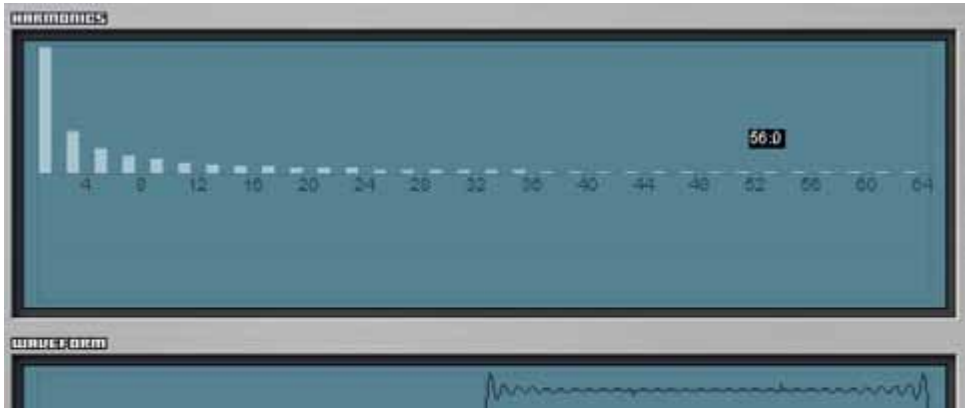


figure 43: an additive square wave

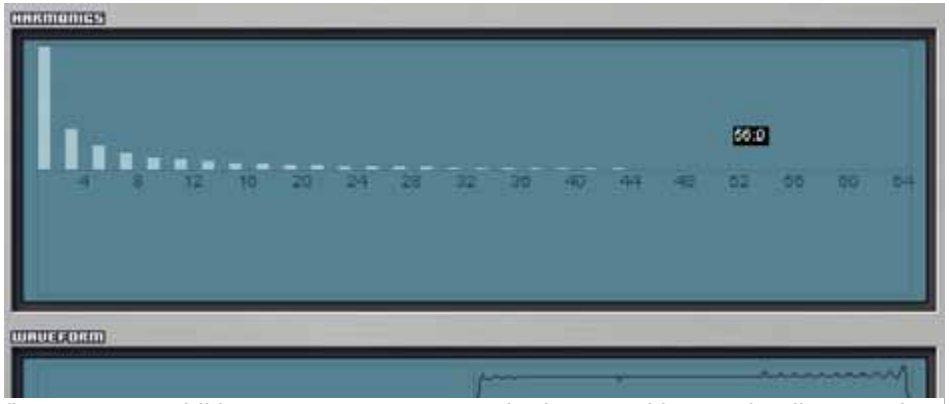


figure 42: an additive square wave – note the improved image, but listen to the degraded sound

square type sound

For this next patch, my intention was to create something from scratch that sounds (vaguely) like a square wave but only using three partials – I set myself the three partial limit largely because I can get quite frustrated when trying to manipulate more partials. As I was after a square wave sound, I chose the first, the third and the fifth partial – the first partial is set to the full amount, the third partial to about halfway and the fifth partial to around 15%.

If you listen to this patch the sound does resemble a square wave. However, to my mind, this tone sounds more like a square wave through a filter, or rather more like a square wave *should* sound through a filter given that square waves don't react that well to filtering.

You will also remember that with **additive seq** it was harder to hear the final tone when the waveform was shifting (due to notes being added) – you may find a use for this truncated version of a square wave if you use it in passing rather than as a feature.

reed squ type sound

For this next patch, the mission I set myself was to take the previous three partial patch, **square type sound**, and to make it more "reedy". To achieve this, I added an

element of the second and fourth harmonics. The second harmonic was set to around 20% and the fourth harmonic to about 30%.

These two extra harmonics also lost some of the character of the square so I increased the amplitude of the third harmonic to around 75% and increased the amplitude of the fifth harmonic by a few percent. For both this patch and the previous patch, these levels were set by ear – there is no scientific logic called upon and the waveform display is of little interest.

As with the previous patch, this patch was not intending to replicate a sound, merely to create a tone color using a minimal number of partials. In this it has succeeded. Hopefully these two waves have illustrated that it is possible to create additive waves quite quickly and easily.

phase I, phase II and phase shifting

Listen to **phase I** and then listen to **phase II**. Now tell me, can you honestly tell the difference between these two waves? I certainly couldn't.

Now look at figure 44 and figure 45. You should notice two things – the harmonics are exactly the same, but the phase of the harmonics is different (the phase of each harmonic is controlled by the lower half of the top pane). The result of tinkering

figure 45: the additive wave used in the patch phase I

with the phase is that the resulting waveforms look wholly different, but still sound the same.

figure 44: the additive wave used in the patch phase I

As you might suspect, this is a patch which has been contrived to illustrate the point that you need to use your ears and not yours eyes. Also, to my mind it suggests that when you program additive waves it is fairly pointless to muck around with the phase of individual harmonics.

So if the waves sound the same, are they the same? No – listen to **phase shifting**. This has the **phase I** wave loaded into oscillator one and the **phase II** wave loaded into oscillator two and the oscillators' envelopes have been set to cross-fade between the two waves. As you hold a note, you will hear tonal shifts. If these waves were identical, there would be no shift in tone.

If you're feeling adventurous, you can set the waves to play together (just tweak the envelopes to be identical) – here you will hear a tone that is different from the tone you would get if the waves were the same: in particular, if you listen to the higher frequencies, when the waves play together the organ-like tone takes on somewhat more of a vintage character. The reason for the difference is the waves that are out of phase are partly cancelling each other out – this gives a different tone (this cancellation works very much like conventional oscillators that are not working in phase sync).

bright additive

We have created some additive waves, but with all of the control and tone shaping options we have only made dull imitations of conventional waves. Let's make a brighter wave that doesn't resemble any wave we have considered so far.

One downside with additive waves is that it is very easy to create organ type tones – another problem is that if you get away from an organ-like tone you can start to create something really wild. While a wild sound may be interesting, it is not necessarily useful nor controllable in a patch.

To keep this wave controllable, I'm only going to adjust three partials – the fundamental, another to give a bit of body and the last (the highest) to give the bright tone. As a first step, I will increase the volume of harmonic one (the fundamental) to 100.

To get the metallic brightness I'm after, I'm going to increase harmonic 20 to 45%. I chose harmonic 20, because the harmonics below this gave more of an organ like tone when working in combination with the fundamental and the higher harmonics made the sound too piercing – harmonic 20 seemed to be the ideal compromise. I then increased it to a level that sounded right to my ear – in this case it was 45%. You may or may not agree with the choices I have made here.

Finally, to add a bit of weight to the sound I am going to add a third harmonic. As with the choice of the brightness element, the choice of a harmonic to add weight was a compromise. Lower harmonics gave more of an organ quality while higher harmonics didn't add weight and could interfere with the sound created by harmonic 20. I did consider harmonic 12, but felt that gave too much of a reedy tone and so in the end I chose harmonic 8 (ie three octaves above the fundamental) and increased it to 60%.

As you can hear, with just three harmonics this wave has created a tone that has an FM-like quality. While the results were obtained to a certain extent by trial and error (at least in the early stages) creating a wave this way is far easier than creating a wave with FM.

If you are creating from first principles (ie starting with an empty wave and adding harmonics) you will be able to clearly hear when any addition creates dissonance and does not add to the tone and remedy the situation immediately. The down side to this form of wave creation in Rhino is that all you are doing is creating a waveform – these waves are not controllable in the way that they are in Cameleon 5000.

That being said, the downside in Cameleon 5000 is that controlling the waves becomes a bit of a hit and miss affair – it is not immediately evident how each partial affects the tone when you are tweaking partials that occur later in a note's passage.

additive bell ep

Although it is interesting to create tone with the additive wave generator in Rhino, the results still haven't passed the "so what?" test. To be useful as a technique, the waves need to be capable of being used to create a sound which could be used in a musical context.

I'm going to use the waveform I created in **bright additive** to create an electric piano type sound. For this patch, I will load the waveform into oscillators one, two and three and will then tweak the additive waveform as necessary to get the tonal changes. I will not use a filter to create this sound but will instead rely on the three oscillators alone. Equally, no FX will be used to create this sound.

The first thing I'm going to do is mess around with the waves:

- oscillator one is left untouched – this will be used to give the bell tone in the attack phase of the note
- oscillator two has harmonic 20 removed – this gives the note a more bright organ like quality
- oscillator three has harmonic 20 removed and harmonic 8 reduced to 15 – while this does give an organ like tone, when used in conjunction with the bright attack sound from oscillator one, this will give the ideal sustain sound for our electric piano type patch.

Now the waves are set up, the first task is to take oscillator one and adjust its envelope to give the attack portion of the note and then to cross-fade that into the sustain portion of the note created by oscillator three. For both oscillator one and oscillator three I took one of the standard piano waves and tweaked it.

For oscillator one, the envelope falls rapidly to zero over a period of about 1.2 seconds. With a steep curve (in this case set to around 25), this gives the classic bell like tone to the attack of the note. By contrast, oscillator three has a similar curve, but its level falls to about half over a much shorter period of time – around 0.25 seconds. This envelope also has a sustain portion (which is not surprising as this wave is used for the sustain portion of the note).

The operation of these two envelopes together gives me a seamless cross-fade between the two sounds. The cross-fade has the smoothness of an FM tone change.

Both operators one and three also have an element of touch sensitivity which controls the volume of each oscillator.

These two operators alone give a serviceable, but simple, electric piano sound. However, I want a slightly more sophisticated tone and to give the player more control. This is where we call upon oscillator two, which has a tone that is brighter than oscillator three but duller than oscillator one.

A very simple envelope is used for oscillator two – it fades in and then remains at its sustain level. However, the velocity curve for this oscillator is set so that it is only heard at the higher volumes, at lower volumes the oscillator has no effect on the

sound. Finally, I have added a bit of detuning to thicken the sound with a natural chorus effect.

With the second oscillator, the patch has a thicker tone with more of a sparkle at higher velocities.

You may be wondering why I created *another* electric piano sound, given that we can already achieve this sound using FM. There are several reasons:

- this method gives a different tone – it is thinner and brighter – which may suit a particular arrangement, especially one which doesn't want to be dominated by an FM electric piano sound
- this sound is far less clichéd than an FM electric piano
- the use of less oscillators means that less system resources are called upon – also, as I've only used three of the six oscillators, it would be possible to double this sound for a thicker tone or to explore other sonic possibilities, and
- it is simpler to design this sound and for the musician, it is easier to understand what is going on – the tone can be easily changed: there are no complex FM cross-relationships that mean one tweak could wreck a whole patch.

To give a more realistic emulation of an electric piano it would have been possible to route the three oscillators through a filter and use a filter envelope as a volume control. However, this option didn't seem to add much to the sound of the patch, so I didn't do it.

More complex additive patches

A few words about Cameleon 5000

So far this chapter has looked at some of the things that additive synthesis can do. The examples have used Rhino as the sound source and so we have created static waves whose composition of partials does not change over time.

Cameleon 5000 gives significantly more control to the additive sound creation process. For instance Cameleon allows:

- each element to be individually controlled over time, so for instance, individual partials can be controlled over time, meaning single wave can get "duller" as it sustains without using a filter
- noise elements to be added to the sound for more realistic sound creation possibilities
- detuning of individual partials
- touch sensitive (or other modulation controller) morphing between two tones
- multi-sample resynthesis, for the (re)creation of more interesting instruments allowing their timbre to change over the whole keyboard

The focus of this book is on ground-up synthesis, rather than sampling/resynthesis, so many of the great features of Cameleon 5000 are being ignored (but many of the features of the other synthesizers are being ignored too). In particular, this book is not going to look at creating sounds on the basis of presets or resynthesized samples. Instead, it will focus on creating sound from scratch by manipulating of the individual partials and also creating sounds by using the limited supply of stock waves.

There are three disadvantages to the approach taken by this book:

- first it is difficult to get usable results
- second, the stock waves are limited (for instance, none includes any FM or metallic type tones), and
- third, there is a tendency to create organ type sounds and if you wanted an organ, you would have bought an organ.

So why are we doing this? Simple – if you understand and can control the waves in this situation, it becomes much easier to control the more interesting waves.

So let's get on and make a few patches. For anyone pondering the naming convention I have adopted in this section, there is no facility in Cameleon 5000 to create banks and so to put the patches in the correct order within Cameleon 5000 I have put a letter at the start of each preset.

a saw to square [envelope]

One of the key features of Cameleon is its ability to morph between sounds. This first patch, **a saw to square [envelope]** demonstrates that morphing. As you strike a key you will hear a sawtooth wave (the wave I have used is the "saw-bright" wave which was chosen from the list of waves in the harmonics drop down). If you continue to hold the key this sound will progressively morph into the sound of a square wave (the square wave I have chosen in the harmonic dropdown is "square-J"). As the key remains held, the sound will morph back to a sawtooth wave and the loop will begin again.

This morphing is achieved by drawing an envelope in the morph timeline.

While this may not be the most interesting sound, listen to the morphing. Then fire up another synthesizer and cross-fade between a sawtooth wave and a square wave and note the difference. Listen to the smoother tonal shifts in Cameleon 5000 – with a cross-fade you can hear the separate elements whereas with the morph you hear one far more cohesive tone (albeit, in this case, one that could do with a good bit of filtering).

b saw to square [velocity] and c square to saw [velocity]

Instead of using an envelope to control the morphing, the tone shifts in these patches are controlled by velocity.

For **b saw to square [velocity]**, at the lower velocity levels you will hear a sawtooth wave. At the higher velocity levels you will hear a square wave. Between the extremes you will hear the morphed sound. Listen to the different nuances as you play the patch at different velocities.

The second patch, **c square to saw [velocity]**, follows the same logic as the previous patch however, the morphing goes from the square wave at lower velocities to the sawtooth wave at higher velocities.

Instead of using a sawtooth and square wave, you could set up two variations on a sound – a bright sound and a dull sound – and then use velocity scaling to morph between these two extremes creating a natural response without having the sonic intrusions that occur when you can hear a different velocity layer being selected in a sampler. As we are morphing, there are no steps to worry about.

d additive electric piano

You will remember that earlier in this chapter we built the **additive bell ep patch** with Rhino. This patch uses a similar simplicity to create a similar kind of patch.

The patch uses four breakpoints – at each breakpoint the composition of the harmonics is redefined. As the patch plays and a note passes from breakpoint to breakpoint, the harmonic composition of the note morphs between the tones set at each breakpoint. The result is a consistently changing tone.

As a first step, I created four breakpoints and dragged the envelope to vaguely follow the shape of a piano envelope – the four points were set:

- at the peak of the attack
- immediately after the attack
- about two seconds into the decay (in other words after the initial tone shaping of the attack had ceased to have effect),
- and after about seven seconds.

These points were chosen to broadly mimic the four main stages of the note in a natural instrument.

At the first point – which corresponds to the peak of the attack of the note – I drew some harmonics which broadly followed those that I used in the additive bell ep patch in Rhino. I used the same harmonics (first, eighth and twentieth) and set the levels to broadly the same level as they were in Rhino. If I were trying to create every nuance of a realistic instrument, I would also have added some noise in the attack phase.

The second breakpoint marks then end of the decay phase which comes immediately after the attack phase – in a conventional ADSR envelope this point would determine the decay time and set the sustain level. For the tone to reflect the characteristics of a real instrument, it should be quite bright. Compare this approach to what would happen if we were using a conventional subtractive synthesizer – at this point the filter would generally have been closed down to the sustain level. However, as we are using additive synthesis, we can adopt an appropriate volume envelope while ensuring that the tone does not change significantly.

To give a bright tone, I have again used the same three harmonics, but to take account of there not being a hammer impact at this phase of the note, I reduced the level of the twentieth harmonic. If you look at the harmonic content in Cameleon 5000 (by engaging "breakpoint" and clicking on the breakpoint) you will see that all of the harmonics are lower – this is to be expected as this phase of the note is quieter. However, if you lift the level of this breakpoint to be equal to the attack breakpoint and then compare the respective harmonic compositions, you will see that this breakpoint does have less high frequency information.

The third breakpoint is intended to mimic the tone about two seconds into the note when the effect of the hammer action has passed. There are two differences between this and the second breakpoint. First, there is no twentieth harmonic and second the eighth harmonic has been reduced further. As before, since this point is quieter in volume than the second breakpoint, both of the remaining harmonics have had their amplitude reduced.

The effect of these first three breakpoints is to give a constantly shifting tone to mimic the attack phase of an electric piano. By gently shifting the harmonic composition of the note and its volume a natural effect can be achieved. After this I have only added one further breakpoint at about seven seconds. The waveform at this point only contains the fundamental and so allows for the note to morph over time and lose its brightness as it decays.

I quite like the attack of this patch – in a track it could work fairly well. However, there are some shortcomings here:

- There is no player control over the tone. A copy of this wave could be created and the harmonics tweaked at each breakpoint to give a duller sound – velocity scaling could then be used to morph between the two waves to give a far more naturally responsive sound.
- While the attack is good, the sustain portion of the note is quite dull. This could be rectified by adding more breakpoints and changing the tonal composition of the note. Also an element of wobble could be introduced with an LFO to give a bit more life to the sustain portion of the note.

You may think it is too much of a delicate approach to cut out or reduce the higher harmonics and you may also rightly think that it is quite time consuming. Perhaps a preferable method of operation is to use the filter to remove some of the main high frequency elements and to "thin" the sound by cutting individual harmonics

over time. Alternatively, you may think that using a filter for an additive synthesizer is cheating!

e organ

I've pointed out at several points that one of the drawbacks (no pun intended) of additive synthesis is the inadvertent ease with which organ sounds can be created. The flip side is that additive is a fast way to create interesting organ tones.

I have created this patch with three breakpoints. The first at the attack stage, the second immediately after the decay (ie at the start of the sustain phase) and the third at the end of the sustain phase. The waveform at the first breakpoint is created by combining the first four harmonics to give an organ like tone.

At the second breakpoint (at the start of the sustain phase), the overall volume of the envelope is reduced, but a fifth harmonic is added to give a touch more brightness to the organ tone. The third break point (at the end of the sustain phase) is set at the same level as the second breakpoint (so there is no change is volume as the note sustains), however its tonal content is much closer to the first breakpoint, in other words it is duller than the second breakpoint's wave.

Within the sustain phase I have then created a loop. During the sustain phase, the note changes its tone due to the differing harmonic compositions at each breakpoint. The loop takes a short phase of that constantly shifting tone and repeats it to mimic something of the character of an organ.

To then give more of an organ like quality to this sound I have also added a low frequency oscillator to give some vibrato.

As with the previous patch, I quite like the attack phase of the notes, but I am less sure about the sustain phase – I feel it could get quite wearing if you used it for a sustained pad. However, you should remember that these patches are intentionally made quickly – the main reasons for this was to show that additive can be a fast way of working and to demonstrate the basic principles. To really gain some benefit of additive synthesis you need to invest far more time than these patches would imply.

FM or additive?

One question you may have is when do you use FM and when do you use additive? To my mind it is really a matter of what you feel easier with and how much control you want to have.

If you want to create a tone and to control how that tone changes over time and to be quite precise in the changes you are going to make, then an additive wave in Cameleon 5000 is likely to be the right answer.

On the other hand, if you want a tone that shifts over time, but you don't have the patience to adjust the wave at each breakpoint, then perhaps FM is the ideal starting point. However, if you are looking for an unusual waveform which may be layered and filtered, then perhaps an additive wave or two created in Rhino may be your ideal starting point.

As with all of these matters, the final decision is a matter of your taste and finding the tools that you feel comfortable using.

Chapter 10: FX

FX: good or bad?

There is often a debate about the inclusion of FX in a synth. There are several good reasons for including FX and there are equally some good reasons not to include on-board synthesizer FX if you are programming a patch:

- First, the FX units in most synthesizers are not of the same quality as commercially available plug-ins (although the quality is improving and this is not a universal damnation of all built in FX units). However, if you don't like the quality of the FX, remember that you don't have to use them.
- By comparison, the built in FX units are usually designed for a specific purpose and therefore don't have unnecessary features. Most built in FX units usually work well with the synthesizer to which they are attached.
- It is much easier to select an internal FX unit than to insert and route an external unit.
- Most built in FX are coded with a lower hit on the CPU which is partly a reflection of their reduced feature sets. Then again, if you want CPU efficiency, it may be preferable to have ten synthesizer all feeding into one high quality/high CPU using reverb unit rather than have 10 synthesizers all running their own lower quality reverb unit.
- Built in FX units may not be as flexible, nor as controllable as some external plug-ins.
- Built in FX units cannot be configured in the same way as external units. As an example, with most synthesizers it is difficult to insert an EQ unit before a reverb, so that the EQ only affects the signal sent to the reverb unit and the dry signal is not affected by the EQ.
- There are some FX units that are integral to a patch's design – for instance, distortion.
- Integral FX units allow a sound and the effect to be stored as one patch, making the sound more transportable. If external FX are used then each element would have to be separately stored if the sound is to be reproduced on a different system. However some VSTis (not covered by this book) do allow VST FX units to be loaded with the VSTi and stored as part of the patch.
- Lastly you can do some really slick things with built in FX units that you cannot achieve with external units. For instance, you can:
  - control an FX send with velocity
  - vary the depth on an effect with aftertouch
  - "duck" an effect so that it is not heard while the note is heard while the key is held.

One other personal thought – I've got to admit that, while I admire some people's ability to stick with an idea and explore every possibility with a set concept, I get pretty bored with all of the patches based on two slightly detuned sawtooth waves where the only tone shaping is achieved with the FX. I'm also of the opinion that a patch should be strong enough on its own without any FX – the FX should then be the added spice.

That's enough of the arguments. For the purpose of this book, I will acknowledge that there are internal FX units that can be programmed as part of a patch and I will use them in the patches I program at the end of this book. You can decide whether you want to program FX units in your own patches.

Why use FX

FX units are generally used in two ways:

- as an integral part of the sound for instance, a distortion unit, or
- to enhance a sound, for instance, reverb to give a sound some space and depth.

The disadvantages of FX include:

- over use (often clichéd use)
- inappropriate use for a particular track
- excessive demands on the CPU, and
- muddying of the sound.

Deployment of FX

Insert FX

With insert FX the whole of the audio signal passes through the FX unit. So if you want to add a distortion effect, then you will want the whole signal to be distorted by the fuzz box – it would be rare (but not impossible) to mix a clean and distorted signal together when you have full control over the amount of the distortion. EQ and compression are other typical examples of insert FX.

Send FX

With send FX, you send part of your signal to the FX unit and then add a purely effected signal back to the dry sound. Examples of FX units that are grouped under the send FX heading are modulation effects (chorus etc), delays and reverbs.

Using FX in practice

Most of the FX units in the five featured synthesizers work as insert FX. Generally they are inserted after the sound has been created (ie after the oscillators, envelopes and filters) and before it reaches the output. For FX units that are commonly deployed as send FX (such as delays and reverbs) the FX units are still deployed as insert FX and the amount of the effect is controlled by a wet/dry mix control.

FX units

All of the five synthesizers offer built in FX units – some have more, some less. Some have different flavors of similar units, others just give one option. Here is a brief summary of the generic units and an explanation of some of the uses you can put the FX units to.

Distortion

Most people are used to hearing distortion used in an extreme way. However, depending on the amount of control you are given over the distortion, it can be used in a much more subtle manner. Typically, you should be able to squeeze the following spectrum of tone color from most distortion units:

- at very low levels, you are unlikely to hear much effect, but the distortion can have the effect of warming up a sound and perhaps compressing it a bit
- with a touch more, you will often find that a sound can feel more "vintage" as the distortion effect takes out some of the top end of the signal
- as the effect increases, the signal will start to break up – this might be a useful setting for sound effects
- the next step is to move into proper overdrive (start thinking of rock guitars and Deep Purple type organs and you will get the picture) – this overdrive type effect can be quite controllable and quite musical
- finally you will get out–and-out distortion which may be hard to control or use with any subtlety.

All of the synthesizers featured in the book offer some sort of distortion (coupled with filtering):

- Vanguard takes the most straightforward approach, having a drive knob in the amplifier section.
- Wusikstation and Rhino both offer a fairly basic sort of digital distortion unit. Both also offer the facility to balance the clean and distorted signals together.
- Cameleon 5000 offers slightly more options, including the choice between digital and analogue-emulated distortion. One other useful feature within the Cameleon is a compressor (these are discussed in more detail below) and a control to thicken up the sound. Cameleon then offers the option to feed the distorted signal into a separate filter to smooth out the sound a bit more (Rhino and Wusikstation have a more restrictive filter built in to their respective distortion units).
- Z3TA+ offers a choice of five distortion modes: soft drive, hard drive, valve amp, smart shaper and heavy metal. Each has a different character and is more suited for specific tasks. In addition to the different modes, there is also a sample rate reduction stage within the distortion unit. One other useful feature of the Z3TA+ distortion unit is the facility to route the distorted signal to a filter to further control the sound.

When you use a distortion unit it is usually very easy to get a piercing sound which dominates the track, rather than a warm overdriven sound that may fit in context. If you are trying to recreate the sound of overdriven guitars, remember that all the rockers play their guitars through speakers – inefficient speakers that cut off most of the signal above 4kHz. To recreate this effect you may need to use a filter, some EQ or perhaps some form of speaker simulator. Z3TA+ offers a form of speaker simulator.

Also, if you are looking for realism, then the distortion needs to be very controlled. Even heavily overdriven guitars can still sound clean (ish) if they are played very lightly. Ideally you should be able to control the amount of distortion by the velocity of the notes. To get more control and more "gearing" in the distortion (ie a much greater range of overload), modulate the distortion depth with velocity as well as controlling the volume of the signal with velocity.

One other thing you can do if you are trying to create a screaming lead guitar-type sound is add a high pitched sine wave to simulate feedback – give this sine wave a very slow attack (perhaps five seconds) or make it separately controllable (for instance, assign its level to the modulation wheel) and you will have another level of control.

Compression

Compressors were originally used to prevent a signal overloading an input, and indeed, compressors still are used for this purpose. In essence, a compressor is an automatic gain control: when a signal gets to a certain level, the compressor restricts how much louder the signal can get.

In practice, you're quite unlikely to want to use the compressor as a volume control and instead will use it as an effect where it can help to:

- make a sound fatter and/or smoother
- enhance the perceived loudness of the sound, and
- to give the sound more punch.

The disadvantages of using a compressor are that the resulting sound can tend to dominate a mix. With more extreme compressor settings, the sound will usually

have a more uniform level and will fill a broader frequency range. Both of these factors can make it harder to mix a highly compressed synthesizer sound. Another downside of compressor abuse is that the resulting sound tends to be less interesting – the delicate harmonic shifts of a patch tend to get covered up as all of the harmonics take a similar level.

Z3TA+ has a compressor and Cameleon 5000 also includes a compressor as part of its distortion unit although the compressor part can be used without the need to add distortion.

EQ

The purpose of equalization is to make certain elements of the sound spectrum louder and other elements quieter. You can use EQ creatively or surgically. However, remember that you still need to get your sound to sit in a mix, so go carefully with the EQ. Make sure you are not using the EQ to try to cover the faults of poor programming.

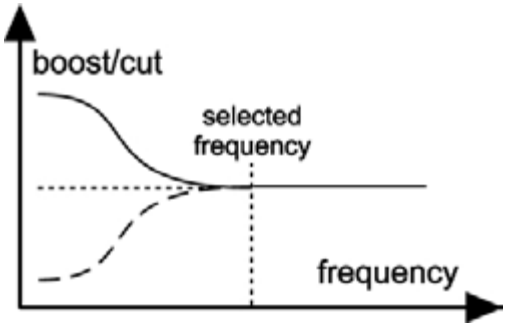


figure 47: low shelf EQ

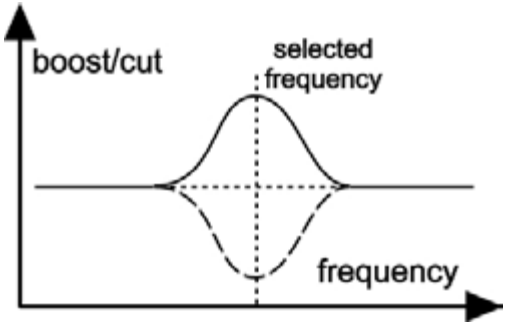


figure 46: semi-parametric EQ

EQ sections are available in Rhino, Wusikstation and Z3TA+.

EQ in Rhino

Rhino has a three band EQ section, or rather three resonant filters which can cut high, mid or low frequencies. The base range of the EQ is controlled by a cut-off slider which controls all three bands.

EQ in Wusikstation

Wusikstation provides seven four-band equalization sections – one for each of the six layers and a further master EQ section. Within each four band section:

- one band is a dedicated low shelf EQ which cuts or boosts all frequencies under the a specified frequency
- two bands are semi-parametric bell curve EQs which can boost or cut frequencies in a selected range, and
- the final EQ is a high shelf EQ which boosts or cuts the sound spectrum above a specified frequency.

For each of the three types of EQ, the bandwidth (or in the case of the shelf EQ, the slope) is preset.

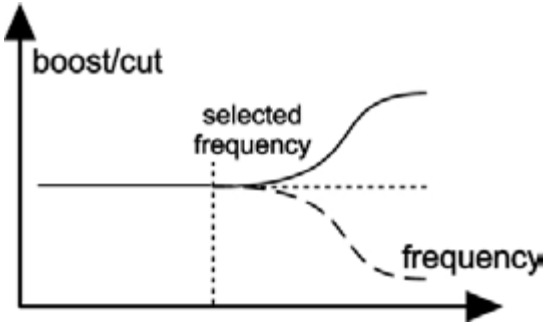


figure 48: high shelf EQ

EQ and simul-mode in Z3TA+

Z3TA+ includes a seven band graphic equaliser. The bandwidths are preset and there are 13 different options for the centre frequencies that are controlled by the EQ section – this allows the equaliser to be more focussed on different areas of the sound spectrum.

In addition, other FX units within Z3TA+ (such as the modulation effects and the delay) also have EQ included.

Within the Z3TA+ EQ section there is a simulator with 30 different amp/cabinet responses to provide equalization control that cannot be achieved with conventional EQ units.

Filter

As mentioned above, Cameleon 5000 also has a separate filter in the FX section. This is particularly useful when coupled with the distortion unit. None of the other synthesizers offers a regular filter in their FX section, however, Z3TA+ has the option to route the distortion signal back to the filter and Wusikstation has a filter stage in its master section.

However, Rhino does offer the "crazy comb" effect which is based on two comb filters. This tends to give more of a chorus type effect rather than act as a filter.

Modulation effects

These units (chorus/flanger/phaser/ensemble etc) all do a similar thing but the sound they produce can be markedly different. In essence they mix a delayed, modulated signal with the original signal to give a fuller/warmer/richer/whooshier (insert whatever description you want) sound.

Used sparingly, these effects can give an element of spark or richness to a sound. Used to excess and you will create a sound that quickly becomes passé.

Who does what?

So which synthesizers offer which modulation effects? Take a look at the table on the next page.

Phaser/flanger

Words cannot readily describe the difference between the sound produced by a phaser and the sound produced by a flanger, however, to my mind a phaser gives a more nasal sound where a flanger gives a more whooshy type sound.

Flangers and phasers are often used on percussion sounds and with plucked-type sounds (such as guitars and electric pianos). The sounds is very distinctive and is hard to use in a subtle way, so you are likely not to want to use the effect too frequently.

Chorus/ensemble

Chorus and ensemble units also mix a delayed signal with the original signal, however, they use a longer delay than would be typical with a phaser or flanger.

The result can be a much more lush/smooth sound – at extremes however, it can sound like vibrato played to fast or just bad tuning.

Used in moderation, the effect can either add sparkle or thicken up a sound without noticeably changing the sound's characteristics. As an example, it is quite common to use a chorus with a bass guitar to provide a more rounded and brighter sound.

When used to the extreme you can get some serious thickening – you can also get a seriously mushy sound, so be careful if this is not your intention.

Rotary speaker

A rotary speaker effect is intended to replicate the behavior of a rotary speaker. The only synth with this effect is the Rhino and it does give you a rotaryish sort of effect.

A general word of caution about all modulation effects

All modulation effects tend to have both a brightening and a thickening effect on a sound. This means that the effected sound can dominate the sound spectrum to a greater extent than the dry sound. Be judicious in your use of these effects if you are applying them to a sound that is intended to be used in the background – it may become more of a feature than you intended. If you are applying modulation effects to a background sound, then look at your arrangement and consider thinning out the arrangement to ensure the modulation effect does not become dominant.

Delay

All of the units offer some kind of delay. In each case, the delay time can be linked to the tempo of the track, being defined as a fraction of the beat (for instance, with delay of 1/4, the signal is delayed by a quarter note).

	Vanguard	Rhino	Wusikstation	Z3TA+	Cameleon
phaser	none	one option available	None	four units are available: >mono phaser >stereo phaser >quad phaser, and >chorus phaser	none
flanger	none	one unit available	None	two units are available: >mono flanger, and >stereo flanger	none
chorus	none	two chorus units available: >chorus 1, a simple chorus, and >rich chorus, a stereo chorus	two chorus units available: >double chorus, and >quad chorus	three units are available: >mono chorus >stereo chorus >6 voice chorus	one chorus unit is available
ensemble	none	three options available: >stereo ensemble >cross ensemble >rich ensemble	none	none	none
rotary speaker	none	one option available	none	none	none

As you would expect, the featured synthesizers offer several types of delay.

Mono delay

A mono delay adds a simple delay to the input signal. The only synth to offer a mono delay is Vanguard. However, a mono delay can be achieved in the other units if the delay time is equal for both channels in a stereo delay.

Stereo delay

With a stereo delay, there are two separate delay lines which may (or may not) have different delay times. Conventionally the delay lines are hardwired left and right. All of the featured synths offer some form of stereo delay.

Cameleon 5000, Rhino and Z3TA+ all allow for the left and right channels to be individually adjusted. Vanguard links the delay time for both channels and Wusikstation gives the option to control the right channel to give a wider spatial effect.

Cross delay

Like a stereo delay, a cross delay has two channels. However, the output from the first delay is fed into the second and the output from the second is fed into the first. This can give the effect of spreading the delays across the stereo spectrum or ping-ponging the delays between the two channels.

Vanguard, Z3TA+ and Rhino all offer some form of cross delay.

Other common variations

You may come across other variations of delay. These don't do much that is different from the three types of delay described above, instead they mix up the various elements, perhaps giving more delay lines or bouncing the output around the stereo spectrum in different ways. Z3TA+ offers a "ping delay" and an "LRC delay" where Vanguard offers a "widen" delay which slightly delays the right channel given a broader stereo spectrum.

Common delay controls

The most common controls that all of the delay units have are:

- the delay time – as noted earlier, this is often (but not always) synchronized to the host's tempo and expressed in terms of beats
- feedback – this controls how many times each delay repeats (quite literally, it controls how much of the delayed signal is sent back to the input), and
- EQ, damping or a filter to control the sound of the echo – typically this will be used to cut the high frequency elements of the delayed signal to make the sound duller, resulting in a more natural sounding echo.

Using delay lines

When adding delay, you need to consider what you are trying to achieve. If you are adding the slightest touch of slap-back just to give a bit of spark and a bit of space, then make sure you have done that. If you want to add a bit of echo, make sure that the delays are duller than the original source. If you are after an effect, then, as with the modulation effects, moderation is key.

Like modulation effects (and reverb), you also need to be cautious about how much space in the sound spectrum an echo will fill. There are several ways to tame the echo so that the sound doesn't dominate:

- first (and most obviously) by reducing the feedback and cutting the volume of the echo
- filtering (both high pass and low pass) the delayed signal so that the echo fills a narrower portion of the sound spectrum
- "ducking" so that the echo does not sound while the main sound source (that is being echoed) sounds – this effect was used conjunction with reverb in **bass modulation** (see chapter 6: modulation in practice).

While an echo may dominate the sound spectrum, I often find that it dominates less than reverb while giving more character to the sound than reverb. I also find echo brighter and less mushy. For this reason I will often use echo rather than reverb when building patches. However, this is very much just a preference of mine. Ironically, I often then like to add some reverb to the delayed signal alone (ie no reverb to the dry signal) to smooth out the echoes and to move the echoes further backwards in the sound spectrum. Again, this is just a preference of mine – you may not like this technique.

Reverb

Reverb can give a sound some spatial context.

All of the featured synthesizers offer a reverb unit, but the quality varies from average to quite good. However, none of the reverb units offers as much control as a good external reverb (whether hardware or software) not least since you cannot put EQ before a reverb.

From a practical perspective, most reverb units use considerable system resources. If you are using (for example) six synths each with some reverb, it would be far more efficient to send all of the synths to a single external reverb unit. This may have the added bonus of giving a higher quality reverb.

All of the synthesizers featured in this book offer reverb – several offer different types and different algorithms.

With reverb units, there is significant interaction between the controls, so even more than normal it is necessary to balance several parameters when shaping the sound. The most common controls offered by the built in reverb units are:

- "pre delay" – this controls the time between the original signal sounding and the reverb taking effect. Smaller values give the perception of a smaller room while longer delays give the perception of larger rooms.
- "size" (or "room size") – this controls the size of the room being simulated by the reverb unit. Usually this control has the same effect as a "decay" control and affects the length of the reverberation (however, Cameleon 5000 offers separate "size" and "decay" controls).
- "damp" – damping works by cutting the higher frequencies in the reverb signal. One of the reasons why electronic reverb sounds artificial (ie clangy and metallic) is because it has too much high frequency information. The damping control can help to create a more realistic simulation.
- "width" – often controls the stereo width of the reverb.

Reverb in Rhino

Rhino gives three reverb programs – each has different controls and behaves in a slightly different manner.

- Stereo reverb is a quite conventional unit with fairly simple controls.
- 8-tap reverb bases its sound on two eight-tap delay lines. It gives a wider range of parameters (including pre-delay and diffusion) and gives a more natural reverb sound (to my mind).
- Stereo oktaverb is more of an effect rather than a reverb unit – it provides a limited reverb sound on the input signal raised an octave.

Reverb in Z3TA+

Z3TA+ takes a slightly different approach from Rhino and provides four different reverb algorithms, each of which combines a different grouping of parameters (such as pre-delay and reverb density) behind the scenes, but still allows the reverb to be edited by comparatively simple parameters. Each of the four algorithms has its own character suggested by the algorithm's name:

- small room
- medium hall
- big hall, and
- plate – a simulation of plate reverb.

Pan

Only one of the synths includes a pan effect – Rhino with "autopan". This effect does what you would expect it to – it pans the signal between the left and right channels and gives you control over the panning.

A similar effect can be obtained with most of the other synths by using an LFO to modulate a signal's position in the stereo spectrum. The option in Rhino makes this effect easier to achieve (and is perhaps more controllable in certain circumstances).

Trancegate

This is one effect that has led to a series of imitators (or perhaps I'm too cynical and other developers had already had this idea, but didn't get to the market before reFX – I hasten to add that none of the imitators is featured in this book).

In concept the trancegate is quite simple, when the effect is switched on, a 16 step gate is engaged. The gate is directly linked to the tempo of the track with each gate step being a division of the beat, so for instance, if the speed is set to 1/16, each step will last for 1/16th of a beat and the trance gate cycle will last one bar.

figure 49: Vanguard's trancegate

The audio signal passes through the 16 steps – at each step, the gate is either open or closed. This creates rhythmic patterns to the sound which still continues to progress through its cycle (as determined by the envelopes and other modulators). Vanguard offers separate gates for the left and right channels which further increase the options for this tool.

Z3TA+ takes a different approach to gating – when the note C0 is struck, the sound is gated. This means that the gate pattern can be constantly shifted or controlled in real time.

Chapter 11: putting it all together

Design philosophy

In the next chapter we will look at some practical examples of sound design but before we get there I want give something of an overview of the whole process.

The comments below are my personal opinions and should not be taken as facts. If you disagree with me, then your opinion is probably correct – I cannot know the context in which you will be using your sounds and so it would be presumptive of me to suggest that I could ever know better. So if you disagree, that is fine with me, but please don't write and tell me.

Programming with a purpose

There may be many reasons to design a sound. For me the most pressing reason for building a patch is the need to find a specific sound that works for a specific track (or a specific sound that is *needed* in some other specific context). If you stumble across a sound that you think is interesting, it doesn't really mean much unless you can then use that sound appropriately. While you may program "just for fun", the remainder of the book is predicated on the assumption that you will have a specific goal when creating any sound.

Only you know how and where a sound is going to be used. And while it is something of a chicken and egg conundrum, I always find that it is more difficult to program the nuances of a sound without having a rough midi part playing. In practice, if you separate the arrangement of the part from the programming of the sound, you will always be at a disadvantage.

By equal measure, I would encourage you to try sounds in a different context from the one for which they were originally intended. For instance, many bass patches make great stab sounds. Don't do this as a last ditch measure when nothing else works, but rather as an exercise to find how different sounds can fit in a different context.

Many people will also encourage you to "just experiment" with programming. Personally, I dislike the notion behind this comment. I'm not saying that experimentation is bad or that you shouldn't try things. My only argument here is that the "just experiment" philosophy is used by those who can't be bothered to learn how to use a tool properly. Equally this argument is used by developers who can't be bothered to explain to people how to use a tool properly or who have developed a tool which may be interesting but is ultimately without purpose.

Arrangement of the track

There are many ways to get to a sound – use a preset, use a commercially available sound bank or even hit the random button until you find something interesting. All of these are valid ways of getting a sound, so why would you bother programming your own sound?

As I have already mentioned, there is one simple reason why you would program your own sounds (and within the scope of programming, I would include tweaking a sound from another source) – to get the right sound for your production.

When you are mixing a track, each instrument needs its own space or else the mix may get be cluttered, indistinct and generally bad. You may have the best bass, pad, stab and lead sound available in your synth, but do these elements all fit together? Does your bass muffle the kick drum? Do the bass and the stab occupy different sonic ranges? Does the stab stand out over the pad and does it then clash with the vocal?

These are all elementary problems associated with the mix which are often addressed with equalization.

Even if the sonic ranges occupied by your synthesizer parts aren't the problem, have you found the right sound for the track? Does the bass fit that pumping rhythm you've worked at so hard? Does that lead cut the top of your head off or is it just a flaccid preset that seemed alright when you sequenced the riff? Does that stab really lock in with the bass to give a perfect performance or is it an unequal match for that bass? And what about that pad? Does it sound like something from a 1970s string machine? Or worse still, does it sound like a cheap VSTi when you want it to sound like a 1970s string machine?

Automation is an important part of the mixing process, so why not also consider automation in the production of your sound? Don't just automate a patch's level, but think how the timbre can change over time – control this with player controls and with track automation (for instance, open and close the filter slightly as the track plays).

Can't I just go and buy something?

A simple question that is often asked is "what is wrong with commercial banks and the banks that come with my synthesizer?"

As simple question deserves a simple answer: there is nothing wrong with presets. The only downside to presets is that they're just not designed to fit your specific track and you may not know how to control them: for instance you may sweep the filter when it may be more appropriate for the sound you are after to sweep the depth of the envelope modulation of the filter.

People (including me) use presets for many reasons – some are excellent, they save programming time (so you can make music), they sound good, the results are better than you can program yourself, the style is outside of your usual programming style or the presets give you inspiration. As I said, I use presets and will continue to use presets in addition to programming my own sounds.

However, it seems that you're being very hopeful that you could take a sound that has been programmed in a vacuum (ie the sound was programmed by someone who knows nothing about your track) and that the sound will then be used in a context of a mix and will work perfectly. I can believe that a patch programmed out of context might be right for your track – I find it hard to conceive that it would fit the mix perfectly *and* would fit the playing style of the midi track where it will be responding to velocity and other playing techniques.

I think you also need to consider the purpose of factory presets and what sound designers have to do to persuade people to purchase a bank of patches. Presets that come with a synth have one purpose – to display the capabilities of that synth. This is a sensible thing to do and is exactly what I would do if I were in the business of making synths.

However good the presets are, however well they demonstrate the synth in question, they are not necessarily right for your track. So what do you do? Most people consider buying some presets from somewhere else. Which presets do they choose? Usually the one that sounds most flashy – not necessarily the one that fits with the track they are trying to construct.

Some (but not all) commercial banks are not very playable (for instance they don't react to velocity or other playing nuances and they don't have controllers set up). They are also often drenched with effects (where this may not be wise/appropriate) – for instance you will often get a bass swimming in reverb. This may sound good when one or two notes are tried, however, in the context of a track it often just sounds muddy.

Don't get me wrong. Some commercially available presets are genuinely inspiring – especially some of the more modern dance focused banks made by guys who are actually making modern dance music. These are patches that have been designed for a purpose by people who actually understand how to deploy the patches – that is why their demos sound so good.

If you have a choice between bad programming done by yourself and a good preset, then the good preset should win every time. Sound programming should *never* detract from the music and the music making process. You should also remember that there is no rule that the sound design has to be undertaken by the creator/producer of a composition: it is a perfectly valid decision to use external sound design, provide the result works for the track.

Finally, please don't form the view that I think you have to program your own sounds to create a musically valid piece. I'm just as critical of the use of presets you have created yourself if they are inappropriate for a track.

Knowing the limitations

Use of samples/real instruments

I have made many references to real instrument sounds. This has been done because most people have an idea of a "piano sound" or a "brass sound". However, I would not recommend using any of the techniques in this book if your goal is to capture the nuances of a real instrument.

If you are after authenticity, then I recommend you hire a studio (with an good room and an engineer), hire the instrument you want to record and find the best session musician you can get. This is most likely to give you satisfactory results. As a second option, find a quality sample library and use samples of the real instrument – again, find a session musician as they are likely to give you the best performance.

That being said, don't be shy of using synthesizers – just remember, they are not the real thing (unless your goal is synthetic sounds).

Polyphony limit and note priority

Any machine you use to create music is going to have limitations. This is true whether you are using hardware or software. Generally there will be a finite number of notes you can play. With hardware, this limit is usually a known number (often hardware will be described as being, say, 64 note polyphonic). With software, the system limitations are usually affected by a range of factors, such as how much system resource is used by other functions (such as the VST host, the effects in the synthesizer, the number of filters you are using etc).

With hardware synthesizers the limitations on voices may lead to voice stealing – when you reach the maximum polyphony, the synthesizer will cut off a previous note as a new note is triggered. With software, when the system resources are reached then your system is likely to freeze or fall over.

However, there are many things you can do with software to reduce the load of your CPU.

Freeze tracks

Most hosts will allow you to freeze tracks and free up resources. This may be the easiest way to prevent system overload.

Limit polyphony

With software you have the option to limit polyphony. By limiting polyphony you will have to address voice stealing rather than look at complete system failure (which is probably the less preferable option). As the voice stealing can then be performed on a per patch basis, you have the option to control the voice allocation so that key parts are not robbed.

All of the featured synthesizers allow you to set the polyphony limits on a per patch basis. With Wusikstation you can even limit voices on a per layer basis and so

ensure that the more important elements of your sound are not compromised. Cameleon 5000 also allows you to limit the number of partials used, however, this may have an effect on the tone.

Simplify the patch

With software there are many things you can do to simplify a patch and reduce its CPU load, for instance:

- Reduce the number of oscillators. Is that sixth oscillator really adding to the sound or is it just there for good luck? Equally, as we mentioned in chapter 4: sound sources, it is possible to use one wave where it creates the sound of two waves in combination.
- Reduce the filters. Either reduce the number of filters used and/or reduce the slope of the filters (for instance in Z3TA+ the 36dB filters use more system resource than the 12dB filters).
- Cut the release time of your envelopes. A shorter release time will mean that the total number of notes still sounding at any one time may be reduced. This can be a very effective way to reduce total system load without losing polyphony especially with very busy playing. However, it may make your patch sound awful.
- Turn off the FX units. As noted in chapter 10: FX it can be a more efficient use of system resources to use external FX units.

Principles of sound design

How do you create a patch?

Unfortunately there is no magic formula to creating a patch. However, I would suggest a few basic procedures before you jump in:

- understand what each knob does
- understand the signal flow in the synthesizer and how each element interacts, and
- have a definite aim in what you're trying to program – don't just fiddle with the knobs and hope something useful comes along.

Once you have these basics sorted, I suggest a two stage approach:

- get something that is "alright" – in other words, a sound that is functional and is reasonably close to where you want to end up – you should be able to get to this point quickly, then
- undertake the detailed tweaking to make sure the patch is exactly right for your needs – this will be the time consuming part of the programming.

When it comes to getting something that is "nearly" right, I see no reason not to use a preset, although the subsequent editing may take longer as you will have to familiarize yourself with the workings of the patch. But then again if you're using one of the synthesizers with a more straightforward architecture or you know what you're doing, you may be happier starting from a clean sheet.

So you know how your synthesizer works, you have a grasp of what all the knobs do, how do you get the basic sound together. For me, I then tend to think in terms of the main food groups.

Main food groups

Most sounds can be characterized by two key elements – their brightness and the envelope (primarily the attack of the envelope, but often you need the decay and sustain level to be right for the contrast of the attack to be clear). Once these two elements have been addressed, any tonal nuances after that may be affected by a range of factors.

Brightness

If you are trying to categorize the brightness of a sound, it will generally fall within one of four descriptions:

- very bright
- bright
- dull, or
- very dull.

You can try to subdivide further, but realistically, it is already quite difficult to try to find the line between these four categories.

Attack

If you try hard, you can find four categories of attack time:

- fast – percussive (for instance a xylophone hit or a guitar string being plucked)
- fast – but not percussive, for instance an organ
- medium, and
- slow

Again, you can try to subdivide the categories further, but it is a fairly pointless exercise.

Getting the combination right

Once you understand what you want in the way of brightness and envelope, getting the right sound is dependent on the right elements being drawn together. For instance, here are some combinations may want to look at:

To get brightness

Consider the oscillator (or oscillators) and filter combination. If they don't work together you will either get a shrill sound or a thin sound or a dull sound, but never a bright sound. For instance, if you are creating a sound based on a square wave, you may find it preferable to choose 12dB filter. A 24dB or 36dB filter may work, but to my ear they ultimately rob the square wave of much of its character, which you may want to keep if you are trying to create a bright patch.

To get richness

One of the key elements to a rich sound is its movement. You can achieve this with careful selection of your sound source (which will usually involve several slightly detuned oscillators and perhaps some delicate use of a sub-oscillator). Alternatively, you can just slap on loads of chorus. My preference is for the former option.

To get attack

Most obviously, to get some attack in a note it is important that the envelope opens quickly. Depending on the tonal quality you are after, it is often important that the note then decays quickly to the sustain level – this will give you a percussive envelope reminiscent of a plucked guitar string or similar.

However, to ensure that the attack is emphasized, another significant factor is the waveform. Some waves can sound slower than others. For instance, all other factors being equal, a square wave can sound like it has a faster attack than a sawtooth wave.

To get warmth

A warm sound is subtly different from a rich sound, for instance, it is much less bright. Warmth is also a comparative term – if a colder sound starts a patch and is then washed over by a warm sound, the contrast can give the listener the illusion of warmth.

Any warm sound will probably have a slow attack and a slow release. There must be thickness to the sound which will come from detuned oscillators. As noted above, one of the key elements of a warm sound is the changing tone which can be achieved with another oscillator with an even slower envelope.

To get punch

It is hard to get some punch into a sound without adding too much brightness. First you need a fat sound source, then you need a fast attack and finally you need a hefty filter. The filter can be modulated with the envelope so that the attack is emphasized but the brightness is controlled.

Choosing the right tools

When you are programming take some time to make sure you choose the right tool. It is easy to make a few glib comments about which is the right tool for any job. For instance if you want a fat "analogue" sounding pad, then perhaps your first choice should be Vanguard – that will likely give you the fastest and the best results.

However, Vanguard is not the only tool you can choose – Rhino makes excellent FM sounds, but in the hands of a skilled sound designer it is equally capable of making great analogue sounding patches. Which is right for you is a matter of your taste, your skill and what is right for the particular track you are working on. However, don't let your (or my) preconceptions about any machine's strength let you rule out a synthesizer from any task.

More importantly, choosing the right tool is more a function of having the knowledge to control that instrument. The only way to learn how to use an instrument is to keep using it over a long time – don't keep buying new instruments, pick one and learn every aspect of its character.

In chapter 12: building patches I will try and explain some of my reasoning for choosing a specific synth to create a specific sound.

Playability

Finally, any patch has to be highly playable. The playability of the patch is one aspect that separates the average sound from the professional. Use your knowledge of how the sound will be deployed to set the modulation sources.

For instance, if the sound is going to be played by a live musician, then you might ensure that velocity has an effect and that the modulation wheel and expression pedal both offer a wide range of control. However, if the midi track controlling the sound is going to be programmed, then you might want to make sure that external controllers, such as track envelopes may have a significant effect, so for instance, you may add an additional filter which has little effect in the normal course but can be readily controlled by a track envelope.

Chapter 12: building patches

So far we have designed some sounds, but many of these patches have been used to illustrate specific points. This chapter focuses on creating sounds that are usable in a musical context and which give the musician a lot of control over the sound.

One of the best ways to learn about sound design is to look at existing patches and take them apart piece by piece. However, this may not always be the most straightforward way to learn. To draw a parallel, you don't learn to do jigsaw puzzles by taking completed jigsaw puzzles apart. For the same reason, reverse engineering patches may not always be the easiest way to learn how to program. In my view, you learn how to program by programming and with experience you improve your skills.

That being said, once you have worked your way through this chapter I would encourage you to dissect other patches to understand how other sound designers create their sounds. Then try to recreate those sounds and learn from your mistakes.

So far in this book many of the example patches have been fairly dry, lacking effects. For this chapter I will be adding effects. However, my taste is (generally) to use less rather than more and as I have mentioned there are many reasons why you would want to rely less on built in effects and instead use master effects in your host (for instance to save CPU usage and to use higher quality effects).

Accordingly, in this chapter you may feel that I have either used too much or to little in the way of effects. As always, there is the question of taste – these patches have been designed to my taste and for the uses I want to put the patches to: your tastes will be different from mine and so you may find some of these patches more or less useful.

As I mentioned, the patches in this chapter are all intended to be used in musical situations. However, if you have the banks you will see that I have added a few patches to help illustrate how I started to build particular sounds. These patches will be mentioned, but will not be deconstructed. Lastly, there are also one or two patches mentioned (but again not deconstructed) where the result was interesting, but was not the sound I intended. These are included so that you can learn by my mistakes.

One other point – just because I have demonstrated a patch with one synth, there is nothing to stop you apply the principles with another synth although you may have some difficulties translating some of the FM principles to all of the other synths. For some patches you will also have to make substitutions – for instance, only Z3TA+ has a 36dB filter and a six voice chorus. These may make some patches harder, but not impossible, to replicate. Perhaps you could use different filters/FX etc and take the patches in a different direction.

If you've got this far and are still reading, I would highly recommend you get hold of the patches that accompany this book – it will make this chapter far easier for you.

Basses

Z3TA+ bass patches

1970s bass

For this first Z3TA+ patch I wanted to build a simple fat synth bass sound which is reminiscent of the synth bass sounds that were popular in the 1970s. I chose to make this patch in Z3TA+ as it gave a thick and rich sound. I did try this patch in Vanguard (listen to **slider**), but couldn't get the effect I was after (in terms of sound) and also Vanguard didn't give me the control over the filter envelope that I was looking for.

The heart of the sound is created by oscillator one and oscillator two (both key synchronized). Oscillator one uses vintage saw one and oscillator two uses vintage square one, but dropped an octave below oscillator one. The level of oscillator one was set at 75% – oscillator two was then increased until it had a thickening effect on oscillator one, but not so much that it overpowered the sound.

When the patch was finished, the final audio test I did to balance the level for oscillator two was to ensure it was loud enough to add weight to the attack portion of the sound, but quiet enough that it had little effect during the sustain portion of the note. To my ears, setting the level of oscillator two at 32% gave the right balance.

Once oscillator two had been sorted, I added oscillator three to double oscillator one. To thicken the sound, I slightly detuned oscillator three. The effect of these three oscillators together is very bright and very dominating. The sound is also a bit fizzy and lacking any control by the player.

The next step was to engage the filter. For this I called up the 24dB low pass filter. The filter block does two things in this patch:

- first, it shapes the tone of the patch, and
- second, it controls the level of the patch.

The tone is controlled by envelope one and the velocity, but the volume is controlled by the velocity alone. As I'm only using one filter, I set the oscillator busses to send 100% of their signal to filter one (bus one).

The interaction between the modulation matrix and the filter is (to my mind) slightly counter-intuitive in Z3TA+ unless you are using the curves. The filter cut-off slider determines the level FROM which the filter is cut-off is modulated. Whereas the filter level slider determines the level TO which the filter level is

modulated. From a practical perspective I can understand the different operations, however, when setting up the modulation matrix you need to be aware of that difference.

To set the filter for this patch, I put the cut-off slider to zero so that the cut-off level would be fully controlled through the modulation matrix. In the matrix, one row controls the filter:

- the modulation source is envelope one
- the control is velocity (ie the extent to which the filter cut-off is modulated by the envelope – between the minimum and the maximum set by the range – is controlled by velocity), and
- the destination is filter one cut-off, as you would expect.

There is a balance to be struck when setting the sustain level and the top end of the range control. My approach is to set the sustain level to a place where it is too low and then to tweak the range control. This allows me to adjust the point to which the envelope's attack opens the filter to the maximum extent. In this case, I was listening for a sound that wasn't too bright.

Once the maximum extent of the range has been set I go back and adjust the sustain level of the envelope to get the brightness (or dullness) during the sustain portion of the envelope right. With this patch I disengaged the velocity control of the modulation matrix when setting the maximum extent of the range.

For this patch, setting the maximum in the range control to 80% sounded right to me. I then re-engaged the velocity control in the modulation matrix and set the minimum in the range. In this case 25% sounded right to me.

However, although the velocity gave me the control I wanted over the filter, there were still two problems:

- first, the sound was still too thick, and
- second, the filter envelope was wrong, which wasn't surprising as the only adjustment I had made to the envelope was to set the sustain level to be too low.

The thickness of the sound was rectified by pushing up the resonance slider in the filter. At around 9.44dB this gave the right sound and also emphasized the effect of the variable filter cut-off. In setting the filter envelope I wanted to do two things:

- first emphasize the attack of the note – this meant that the level of the envelope had to drop swiftly after the attack portion of the envelope, and
- then have the envelope continue to decay gently over time so the filter continues to close as the note is held.

To achieve this effect, I needed to use one of Z3TA+'s multi-stage envelopes. As the modulation matrix had been set up, all I needed to do was to set the envelope – this was done by ear: I listened until is sounded right.

In this case, the attack was set to its fastest (ie zero). To give the bite to the attack, I set the slope level to 75% and the slope time (ie the time it takes the envelope to go from the maximum at the end of the attack phase to the slope level) to 0.05 seconds. This gave some bite to the sound.

So that the sound would progressively continue to lose its brightness I set the sustain level below the slope level. In this case I set it to 62.5% and the decay time to 5.2 seconds. This means that the note will not continue to lose its brightness after is reaches the sustain level. However, in practice, it is unlikely to continue to be held for much after this point so I am not too concerned about this.

With the filter we have done two things here. Firstly, we have set the tone of the patch. However, there is a second effect – the sound of the patch now fills up far less of the sound spectrum: it is still a big bass sound, however, now the patch clearly works as a bass sound and does not dominate areas outside of the bass range. That being said, you will still need to ensure that it works in the low end of the spectrum with the kick drum and any other sound elements in that region.

Linking the filter cut-off to velocity does give velocity some control over the volume. However, I wanted more and so I set up another line in the modulation matrix. This time the source was "on", the control was velocity and the destination filter one level. I didn't want the volume to go from zero to maximum, instead I was looking for a slight reduction at lower velocities. To achieve this I set the minimum and maximum range in the modulation matrix to 70% and 100%.

With the level of filter one set to 75% the patch has a tendency to overload. Ideally the level of the filter (or the oscillators) would be turned down until the overloading is removed. However, I'm lazy so I engaged the master limiter. You should be aware that this uses additional CPU resources and so may not be the most efficient programming technique.

If this patch had been programmed in a synthesizer from the 1970s, the synth would almost certainly have been monophonic. Accordingly for this patch I set the polyphony to one and added some portamento. The portamento works so that when one note is held the next note slides to its pitch: there is no retriggering. The effect is sufficiently subtle that is can be heard for small intervals – at larger intervals the effect is not perceived. Finally, staying with the 1970s theme, no FX are added to the sound.

mwuah

This next bass patch is based on a vintage sawtooth one wave dropped an octave and then fed through a 36dB low pass filter with the cut-off set to 983 Hz and the resonance set to 7.43dB. This gives quite a dull sound – for this patch I want to

create a “slow” bass sound: as you can hear, the oscillator/filter set up gives us a fast and uninteresting sound.

The first thing I wanted for this patch was for the pitch to rise slightly when the note is struck. To do this I used the pitch envelope and set the start level to -8.75% and the attack time to 1.79 seconds. Using the pitch envelope as the source to modulate oscillator one and having a maximum range selected, when the “pitch 1O” curve is selected, the effect of the pitch rising when the note is first struck can be heard.

The second tweak was to open the filter slightly when the note is struck and then close it very slowly. To do this I called up envelope two to modulate the filter cut-off – in the modulation matrix, the maximum range was set to 6.8%. There are two stages to the envelope – the attack time which is set to 0.75 seconds and the slope time which is set to 10 seconds. At the end of the slope time, the slope level is set to zero (ie the envelope has no further effect on the filter).

So far, the sound is not particularly interesting and it is too fast for a slow bass sound.

The third change was to separate the filters. There is no separation slider – this change has to be effected through the modulation matrix. The filter separation destination in the modulation matrix is controlled by envelope one acting as the source. The range was set to the full extent and the curve selected was U-LIN- in other words, at zero modulation the filters are separated to the maximum extent (unless they are further separated by adding another line in the matrix), while at maximum modulation there is no separation (ie the filter acts as a conventional 36dB filter.

The patch illustrates filter separation in practice – or more accurately, it illustrates recombining separated filters. The 36dB filter is actually three filters stacked together. Each filter has its own cut-off and resonant peaks: in normal mode these resonance peaks stacked and so are quite prominent.

With separation, the cut-off frequencies of the filters are spaced apart (ie successive cut-off frequencies are progressively raised). As the filters are separated, there are still three filters each with their own resonant peaks, however the peaks are at different frequencies and so are less prominent.

The net effect of separating the filters is a change in tone without too much of a significant change in the brightness. This gives the sound more of a vocal quality. Try playing the patch in higher keyboard regions to hear this effect more clearly.

Envelope one which controls the separation takes advantage of Z3TA+’s multi-point envelopes. The attack time was set to 0.26 seconds and having reached its peak, the level then falls over a slope time of 0.34 seconds to a slope level of 80%. This portion of the envelope gives the patch its characteristic “m-wow” sound. After this the level of the envelope rapidly decays to a sustain level of zero over a decay time of 0.95 seconds. At this sustain level, in the higher ranges of the keyboard, the filter is only kept open by the effect of envelope two.

So there you have it, a slow bass which in the higher keyboard regions gives quite a vocal quality. I like the sound as it is and so I decided not to add any FX.

You will notice that there are no player controls here. This is quite a slow patch and there was only one thing I wanted to do, however, unfortunately Z3TA+ doesn’t allow it. Ideally I would like to modulate the attack time in envelope one (with velocity) so that the time over which the filters are recombined could be controlled. As I said, you can’t do this in Z3TA+ and so there are no player controls here.

Bells and chimes

Z3TA+ bells

bells

It is quite possible to create bell-like tones without FM. I quite like the non-FM way of creating bell sounds because the timbre is less sharp than with FM and there is almost a wooden tone to the sound in certain key ranges. As this patch is created with only two of Z3TA+’s six oscillators, it would be possible to layer some FM bell sounds to give a hybrid sound. Personally, I think this sound is good on its own and so I haven’t followed that option.

This bell sound is based on two multi triangle waves – one is clean and the other is put through a filter with lots of resonance. In the patch oscillator two is fed to bus two (ie filter two) – as no filter is selected, the signal passes through without being affected except where the level of oscillator two (which was set to 100%) is modulated by envelope two.

Oscillator one is fed into bus one and passes through the 24dB filter. The cut-off frequency was set to 4,800Hz and the resonance set to 26.88dB (ie the maximum). The level of this oscillator was set to 85% and this level is modulated by envelope one.

The filtered wave has a bright metallic tone with an almost breathy quality. The unfiltered wave is less bright, but bringing it together with the filtered wave gives a warmer, ringing tone. However, there is no volume control and the percussive attack which is so characteristic of a bell is missing.

Both envelopes have a fast decay although envelope two has faster decay. Both envelopes also use Z3TA+’s multi-point envelope. Envelope one falls to the slope level (of 67%) over 0.13 seconds and then decays to the sustain level (of 17%) over 0.58 seconds. Envelope two falls to the slope level (of 30%) over 0.45 seconds and

then decays to the sustain level (of 26%) over 0.1 seconds. The combined effect of the envelopes takes a metallic sound and gives it a bell like quality.

To achieve the final sound, three effects units were added:

- the six-voice chorus which adds some "shimmer" to the sound
- one ping delay which gives the initial attack of the bell more emphasis
- the large hall reverb which, in addition to adding reverb, also smoothes the sound giving a more coherent feel.

Keys

Wusikstation piano patches

piano

Piano is a comparatively simple piano patch. The basis of the patch is the piano grand waveform (from the famous keys four bank) loaded into layers one and three. These have then panned hard left and hard right.

The output volume of each layer has been linked to pitch by selecting pitch as the modulation source and the layer's volume as the destination. As the pitch increases, layer three's volume increases, where layer one's volume decreases. The effect of these volume changes is to spread the piano across the stereo spectrum so that lower notes are heard in the left channel and higher notes are heard in the right channel.

It would have been possible to use one layer with pitch as a source and pan as a destination. However, I preferred the sound created by having two layers.

For both waveforms, the layer envelope was set with a zero attack and decay time, the sustain was set to its maximum and the release time to around 60. The velocity was set to 86 giving quite a responsive patch.

Each waveform passes through a 2 pole (12dB) filter which is fully closed (with no resonance). The filters are both controlled by modulation envelope one. With the envelope, the attack time was set to zero, the decay time to its maximum and the sustain level to zero. The vel knob was set to 76. The effect of the envelope is that it introduces velocity control to the filter and it also allows the brightness to fade as a note decays.

While interesting, to my mind, the samples need a bit of weight. The first bit of weight that I added was layer two with the pure sine wave selected. Unlike the other two waves, this decays over time – the layer envelope was set with an attack time of zero, a decay time of 94 and sustain level of zero. The velocity knob was set to 90.

The final bit of weight I added was in layer four where I called up the 202 vel bass 01 wave, again from the famous keys four bank. The layer envelope's attack time was set to zero, the decay time to 87, the sustain level to zero and the vel control to 127. The layer's volume was then set to 61 and in the modulation matrix, pitch was set as a modulator negatively controlling the volume (ie so the layer gets quieter at higher key ranges).

To complete the patch, layers one and three were then sent to the reverber.

electric piano

Much like **piano**, **electric piano** is a very fast, effective piano patch in Wusikstation. In many ways this patch is probably simpler.

In layer one the Rhodes 73 wave is loaded and in layer two the Rhodes vel wave is loaded. Both waves come from the famous keys four bank. The velocity range for layer one was set from 0 to 116 and for layer two from 117 to 127. Layer one runs through a 4 pole (24dB) filter which is controlled by velocity.

Mod LFO one was set to 1/8 and introduces vibrato (when the modulation wheel is pushed up). In the modulation matrix, the amount (of modulation) was set to 62: much more and the vibrato would sound unnatural.

Once set up, FX were added to the patch. In this case I added a Quad Chorus and a short echo set to 1/16 with a small amount of feedback and high damping so the echo is acting more like a reverb unit.

The final tweak was to control how much signal was sent to the Quad Chorus unit – the idea was to have more chorusing at lower velocities but less at higher velocities so the aggressive tone could shine. This took two lines in the modulation matrix – one for each layer. The source was velocity and the destination layer one or layer two FX1. The minimum for each line was set to 113 and the maximum to 31 with an amount setting of 127. To my ear, this gives the appropriate scaling.

Z3TA+ piano patches

One of the key themes I have emphasized is programming with a purpose. However, I stumbled on this patch when I was trying to do something totally different. From an interesting sound I then developed the patch into a highly usable Wurlitzer-type electric piano sound.

The basis of the sound is contained in **Z3TA+ piano root**. Those of you with the patches can look at this patch in more detail. In essence, the sound is created by oscillator one acting as modulator for oscillator two (the carrier). This simple FM stack creates a tone which can be changed in a very controllable manner (in this case by velocity controlling the output of oscillator one).

Oscillator two is then doubled with oscillator three (which is not frequency modulated) and the resulting sound is then fed through a low pass filter. This architecture means that the tone is changed in two ways:

- the sound source changes (quite radically) by velocity controlling the amount by which oscillator one frequency modulates oscillator two, and

- the filter's cut-off frequency is also modulated.

Z3TA+ piano takes this simple idea and builds a usable patch. We will use a similar technique with **male voice choir** in the pads section, below.

Z3TA+ piano

Z3TA+ piano takes **Z3TA+ piano root** as its basis, makes a few tweaks and then builds. Again, the essence of the sound is oscillator one working as the frequency modulator of oscillator two which acts as the carrier. Oscillator three then doubles oscillator two (but is not frequency modulated so it adds much more to the weight of the tone than simply thickening it through doubling).

As with the earlier patch, oscillators two and three use the vintage saw one wave. However, in a change from the earlier patch, oscillator one uses vintage square one rather than the sine wave as the modulator. The change is subtle, but to my ears it gives the sound a bit more bite. In chapter 7: frequency modulation synthesis, I expressed the opinion that it is probably best to stick with sine waves for FM – I still hold with this view, however, in this patch there is extensive use of the filter and I feel that the change in modulator wave adds to the tone of the patch.

The two sound generating oscillators are then fed into a 36dB low pass filter (filter one) with the cut-off set at 540Hz and the resonance to 1.26dB. This has the effect of robbing the sound of any real tone and so to rectify this and give a bit more shape to the sound, the filter is modulated by envelope one – this modulation is controlled by velocity and the range is set to a minimum of zero and a maximum of 70%.

Envelope one, which controls the filter, has its attack time set to zero, the slope time to 0.11 seconds and the slope level to 43%. This gives a brightness to mirror the hammer strike of an electric piano – the envelope then decays over 4.5 seconds to a zero sustain level. Finally the curve I selected was U-LIN+ as this gave the smoothest control over the widest usable range of tones.

The envelope is essentially a simple ADSR envelope – the second part of the envelope (the decay time and sustain level) allows the filter to be used largely as a volume control so that a note does not sustain indefinitely (however, this element of the envelope does change the tone too).

The reason for the simplicity in the filter is that much of the tone of this patch is created by the frequency modulation of oscillator two by oscillator one. The output level of oscillator one has a modulation source of envelope two. In the modulation matrix, this modulation source is controlled by velocity.

The output level of oscillator one is quite low at 25%, however, this is sufficient to give some real bite. As mentioned this output level is modulated by envelope two where the attack time is set to zero, the slope time to 0.66 seconds and the slope level to 75%. This gives an effect similar to a "hold" stage in an envelope and emphasizes the attack of the note – as the slope time in this envelope is longer than the slope time in envelope one which controls the filter, this gives a longer emphasis to the attack while the tone is controlled by the filter. Having reached the slope level, the envelope then falls to the sustain level 31% over 6.5 seconds.

So far we have created a serviceable electric piano type patch. However, I wanted to make a very playable patch, so this is what I did.

The first thing I wanted to do was to fatten up the patch. This was quite simple – I called up oscillators five and six, selected a vintage square one wave in each and routed them to bus one (ie so that their full output goes through the filter). You will see that oscillator six has been raised by an octave.

Adding these two oscillators gives the patch is wonderful tone with the nuances still being controlled by the simple FM stack.

The final tweak I made was to add some effects. In this case, I added a quad phaser and some plate reverb. The phaser slightly thins the tone giving a bit of movement and the reverb adds some space to the sound.

I know I'm biased, but I love this patch. I think that by coupling FM and subtractive synthesis you can create really playable sound that uses the best elements of FM and subtractive without having the disadvantages of sounds created using only one technique.

Z3TA+ piano developed

I then developed **Z3TA+ piano** by making two changes:

- adding oscillator four to act as a modulator for oscillator five, and
- adding LFO five and LFO six to modulate the filter and oscillator three's volume respectively.

When listening to the patch on its own (ie not within the context of a track) I prefer **Z3TA+ piano** to **Z3TA+ piano developed** – you may have a different view. To my ear the former is more playable and this latter patch is too bright. However, within a track this developed patch may work better. Anyway, let me explain what I did in a bit more detail.

Oscillator four is loaded with a sine wave that has its level set to 62.5%. This frequency modulates oscillator five. The output level of oscillator four is modulated through the modulation matrix by envelope three with a range of 0 to 34% and, as with the modulation of oscillator one's level, the modulation of oscillator four's level is controlled by velocity.

Envelope three (which acts as the modulation source for oscillator four's output level) is set up as an ADSR type envelope. The attack time is set to zero and the decay time to 0.03 seconds with the sustain level at zero. The effect of oscillator

four is to give more bite in the attack and to make the tone thinner. It has no effect on the sustain portion of the note.

The other main change is the addition of two LFOs. These add a slight wobble to the tone and an element of tremolo. For these elements I used LFO five and LFO six which are both polyphonic LFOs and each has its phase set to free. This means that their effect on each separate note will be slightly different.

LFO five was set as the modulation source for the filter cut-off. The speed was set to 9.4Hz and the fade time to 0.28 seconds. In the modulation matrix, the range was set to 21.9% and the U-LIN- curve selected so as the LFO wobbles it reduces the cut-off frequency. As the note sustain, this addition gives a slight wobble to the tone and a very slight tremolo effect.

However, although a polyphonic LFO has been selected, the tremolo is rather uniform. To keep the effect but remove the uniformity, LFO six modulates the level of oscillator three. The speed of this LFO was set to 2.21Hz, the fade to 1.31 seconds, the range to 43.8% and again the U-LIN- curve was used.

The combination of the two LFOs gives a far more natural tremolo type effect.

Lead

Z3TA+ lead

lead gtr

This patch makes extensive use of Z3TA+'s effects units, however, the other elements of the sound are still important.

The sound is intended to play lead guitar-type parts and single note guitar-type riffs. While this patch is intended for single notes, the polyphony has been set to four so that sustained notes are not unnaturally cut off. Like a real guitar, this patch is highly touch responsive – at lower velocities, the tone is muted (although clearly very overdriven) while at higher velocities the sound screams.

The root of this patch is fairly simple: two sawtooth waves (one two octaves above the other) which are routed into a 36dB filter. This gives a slightly distorted tone and nothing more. The filter is then closed down to 48Hz and the resonance increased to 8.17dB. At this point no sound is audible. To get some sound and to give some shape to that sound, the filter cut-off was modulated twice in the modulation matrix:

- first by envelope one – here the envelope adopts a fairly conventional ADSR envelope with a decay time of 2.41 seconds and sustain level of 70%, and
- second by the key position so that the envelope opens up more in the higher key ranges. In the modulation matrix, the source is "on" – the range 7% to 70%, the curve slow+ and the control is U-UNOTE#.

Finally some vibrato is added (which can be controlled by the modulation wheel). The modulation source for the vibrato is LFO one which uses the triangle wave and had its speed set to 7.68Hz. The range is 35% and the curve is pitch 1S (ie one semitone if the full range is selected).

For this patch, I set the vibrato separately for oscillator one and oscillator two. It would have been possible to select "all osc pitch" as the modulation destination. However, that would rob the patch of the flexibility when adding other oscillators. In addition, this architecture also means that different ranges can be selected (which can dirty up the vibrato).

At this point you can hear that there is a broad spectrum of tone ranging from almost a harmonic like sound at low velocities to quite a gentle tone in the mid-range through to a more biting tone at the higher velocity ranges.

However, we don't really have the scream of a lead guitar, so this is where we call on the FX units.

There are three key elements that take the basic sound and give it the characteristic of the finished sound – the heavy metal distortion unit, the six voice chorus and the mid wide EQ.

The heavy metal distortion was chosen for its tone. The gain was set fairly high at 12dB, but not as high as it could be and the level was set at 71%. To control the tone, the output was routed to filter one: without this routing, the tone is just thin and buzzy and really unusable. To finish of this stage, the tone was set at 78% and the decimator to 7%.

After going though the distortion unit, the sound needed to have some of the thickness taken out and to be made brighter. For this the mid wide EQ was called upon – the 1.2k band was dropped by 5dB, the 3.2k band was boosted by 12dB and the 5k band was boosted by 2.25dB.

However, even with the distortion and the EQ, the key to taking the distorted tone and making is thick and smooth is the six voice chorus which also has the effect of spreading the patch over the stereo field which also makes the patch very dominant over a wide area of the sound spectrum. With the six voice chorus, the depth was set to 62%, speed to 0.09Hz, delay to 9ms, feedback to 30%, the EQ mode was hi+ and the low control was set to -12dB with the high to 0.63dB and finally the level was set to 60%.

You will hear that even with these effects being added – and some are quite extreme – the patch is still very velocity responsive.

Once the distorted lead guitar tone had been set it was then embellished with a ping delay and a plate reverb. Both of these were added to taste, in particular the echo was set to fit a riff for which this sound was programmed.

Pads

First a word of caution. We're going to create some pads here – many of these are very thick and will fill up huge amounts of the sonic range.

In my personal opinion, one of the key differences between professionally and non-professionally produced tracks is the use of pads: professionals tend not to use them (or if they do, they are used with subtlety). By contrast, the amateur producer will often use a pad as a first step to fill out a track.

Feel free to use pads in your productions, but please, think how you are going to use the pad and don't use it to cover other shortcomings in your production.

Rhino pad

fm pad

We'll create some incredibly thick pads later with Z3TA+. I wanted **fm pad** to be a more delicate pad which could therefore be used in more situations.

This pad is essentially two simple FM stacks where the level of the modulator fluctuates to give subtle tone shifts over time – we will use a similar technique in **male voice choir** (see below) but with this latter patch the tremolo will be introduced by an LFO. In this patch I won't try to describe the volume envelopes in detail: you'll have to get the patches if you want to take a look.

For the first stack, operator one is modulated by operator two with a 1:1 ratio – in the routing matrix, the modulation amount was set to 47. In the second stack, operator three is modulated by operator four with a 2:1 ratio and in the routing matrix, the modulation amount was set to 34. This second stack was slightly detuned (both operators by +9).

To give a slight bit of extra tone to the first stack, operator five modulates operator one with a 1:6 ratio – the modulation amount was set to 3. To give this tone a bit more movement, operator five is modulated by operator six which acts as a low frequency oscillator.

The outputs from operators one and three are then fed into both the raw output and filter one – the respective levels are balanced by the "filtered" slider. You will see that I have set this slider at 70. The filter selected is analog lowpass 3 and the cut-off slider is set to 4.

As you can hear, the result is a simple, but effective pad with the ability to shift the tone by moving the "filtered" slider.

Vanguard pad

thick pad

With **thick pad**, I wanted to create a simple, but quite dense pad. The end result gives a tone similar to **lush pad** (which we will look at later) but uses considerably less system resource.

This patch uses three waves, a sawtooth, pwm saw and a square wave – the pwm saw was tuned an octave higher than the other two and the fat knob was turned to 96. This combination gives a very thick, bright sound.

To take this from a bright patch to a pad, the 12dB low pass filter was called up with the cut-off set to 37Hz, the resonance to 20% and the keytrk to 55%. This makes the sound dull. To give some life, envelope one was called up and the cut-off knob (in the envelope lane) turned to 22. The envelope was then set with an attack time of 2.2 seconds, a decay time of 4 seconds, a sustain level of 104 and a release time of 2.9 seconds.

To give the patch volume some shape, the level knob in the envelope one lane was set to 127. In the amp section, the volume was set to 40, the veltrk to 32 making the patch quite touch responsive, the spread knob was set to 80 and the drive knob to 28 to add a bit of dirt.

To give the sound a bit more movement, oscillators one (sawtooth) and two (pwm saw) are modulated by their LFOs. LFO one was set to 8.96Hz with the detune knob to 19. LFO two was set to 3.44Hz with the detune knob to -14 and the pwm knob to 44.

I could have called up some FX for this pad, however, I didn't feel that they would have added much and also, they would have used more system resource.

Z3TA+ pads

male voice choir

I wanted to create a patch that was evocative of the sound of a male voice choir. It took me quite a while to get the precise sound I was after. For those of you with the patches I have included two patches which, although I really like the sound, I rejected because they were not the precise sound I was searching for.

The two patches I rejected **voices** and **voices II** are both based on a square wave pushed first through a 24dB low pass filter and then through a formant filter. **Voices II** has more layers, more vibrato and a higher pitched ghost-like element.

The two patches have quite an ethereal, haunting quality to them. However, although they are breathy and might in certain circumstances sound like an effected human voice, they are not close enough to the male voice choir sound I wanted.

With **MVC root**, I found the essence of the sound I was looking for. Again, a square wave is pushed through a format filter. However this time the square wave is frequency modulated by a sine wave. This element of frequency modulation changes the tone radically and gives it that "male voice" quality that I was looking for.

If you listen to **MVC root** you may not be that impressed with it – especially after **voices** and **voices II** which are both much more developed sounds. You may also disagree with my assessment that the sound is reminiscent of a male voice. Push the modulation wheel and play the note and you will hear a radically different quality: the amount of FM is controlled by the modulation wheel.

I suggest you move the modulation wheel before you play the note – if you move it too quickly while the note is sounding the effect is not particularly pleasant. Much like a formant filter, there are a few quite distinct tones to be found in the travel of the modulation wheel – shifting too obviously from one tone to another does not necessarily give a pleasing result. If you want to add a bit more life to the sound, you can use your expression pedal to control the vibrato.

In this patch I use FM to create the root of the sound. To create the source sound I could alternatively have used the wave-shaping tool or imported my own wave. However, I chose FM first because it can be controlled over time (which the other two options cannot) and more importantly because I'm happier using FM.

Anyway …. that was how I got to the root for **male voice choir**. This is how I worked from that point to create a patch. When auditioning this patch, the range when an acceptable male choir sound can be obtained is quite narrow – if you play outside of this range, the effect may be lost. This reflects some of the limitations of the human voice.

Oscillators one and two create a simple FM stack with a C:M ratio of 1:1. This whole stack is doubled by oscillators three and four which create another simple FM stack which uses a C:M ratio of 1:2. In both cases, the modulator's wave is a sine wave and the carrier wave is a vintage square three wave. Both of the stacks then feed directly into the filter block.

In the first FM stack, the level of oscillator one was set at 10.9%. However, this level is then modulated by LFO one (which was set to a triangle wave with a speed of 3.85Hz and the range was set to 11% in the modulation matrix). This subtle modulation of the modulator causes the level of oscillator one to fluctuate gently. This variation in the volume of oscillator one causes the tone of oscillator two (which is frequency modulated by oscillator one) to waver slightly. The sonic result is that the output from oscillator two takes on some of the inconsistency of the human voice. In other words, this modulation of a the modulator allows us to achieve precisely the result we are looking for.

In the second FM stack, oscillator three's level is set at 30% and is modulated by LFO two (which was set to a triangle wave with a speed of 2Hz and the range was set to 30% in the modulation matrix). Again this modulation gives the inconsistency to mimic the human voice.

With the two FM stacks (created from the first four oscillators) working together, oscillator two was detuned by 12.5 cents to give a bit more movement to the tone and oscillator two and four had their levels set at 64% and 54%, respectively.

To give some further movement to the sound, the pitch of oscillators two and four are then each modulated by a triangle wave LFO. Oscillator two is modulated by LFO five with a speed of 1.14Hz and a range of 20%, using the pitch 1S curve, and LFO six modulates oscillator four with a speed of 2.96Hz and a range of 19% using the same semitone curve. The polyphonic LFOs have been engaged so that each note is separately modulated giving more of a choral effect.

The filter that is used is comparatively simple – there is no modulation. The filter for the FM elements (filter one) uses the format filter with its cut-off frequency set to 53Hz and its resonance set to 0dB. This cut-off frequency gives an "A" (as in cat) tone. The second filter is used later and does not affect this element of the tone.

Before I continue with the construction of this patch, let me meander for a moment and explain a bit further (with an audio example) why I used FM for this patch. As you will have noticed, the FM effect we introduced was very subtle. Indeed, I would not blame you for being sceptical about its effect. If you are sceptical, then turn off oscillators one and three and listen to the results. Without these two FM modulators, the sound takes on a totally different tone becoming far more metallic and harsh. There is absolutely no way that the sound without the FM could be described as having a vocal quality, even with the formant filter.

Returning to the patch, we need to give it a bit of shape – this is where we tweak two of the envelopes. First, to set the volume of the whole patch, the amplifier envelope was edited. A fairly simple ADSR envelope was set up with the attack time set to zero, the decay time set to 0.51 seconds, the sustain level set to 68% and the release time set to 2.9 seconds.

These settings were largely chosen so that they didn't have too much effect on the settings chosen for the envelopes which modulate the oscillators – in particular, the release time was set so that it didn't cut the release time of the oscillator envelopes. It would have been possible to use the amplifier envelope to control the output of oscillators two and four. However, I didn't particularly like that effect, so I used the separate envelope. Equally, I felt it was important for this patch to use the amplifier envelope to give a bit more shape to the sound.

For the outputs of oscillators two and four, I set envelope two as the modulation source using the full range in the modulation matrix (ie the levels of oscillators two and four are fully controlled by envelope two up to each oscillator's output level). Envelope two was set as an ADSR type envelope with the attack time set to 1.05 seconds, the decay time set to 4.4 seconds, the sustain level set to 68%, and the release time set to 1.18 seconds. This gave a feel reminiscent of a choir.

At this point, the sound of the patch was interesting but I wanted to add some thickness, particularly in the lower registers. To do this I called up oscillator five and loaded a vintage saw one wave in multi free mode (ie multi-mode with the phase of each oscillator running freely). The phase slider was set to 104 degrees to give a fairly fat sound.

Simply adding an oscillator didn't do enough – this gives a fizzy sound which detracted from the choir sound created by the first four oscillators and filter one. To remedy this I routed the new oscillator through filter two which is a 36dB low pass filter with its cut-off set to 675Hz and its resonance set to 3.4dB. This routing means there are now two separate elements to the patch which are layered together.

While the filtering of this fifth oscillator improves the sound, it still didn't quite get there in terms of fitting with the rest of the patch. To help it fit, I did two things:

- First, I added some key scaling to the output level of filter two. This has the effect of reducing the volume of this fifth oscillator/second filter combination as higher notes are played – this gives the lower range thickness while keeping the higher range tone. To do this I set B-NOTE# as the source in the modulation matrix with filter two level as the destination: the range was set to maximum and the curve to U-LIN-. It is this negative curve that leads to the damping of the volume at higher pitches.
- Second, I added a volume envelope to modulate the output of oscillator five. In the patch this is envelope generator one and in the modulation matrix, the range is full and the control is velocity. The envelope is very simple: the attack time was set to 1.3 seconds (ie slightly slower than for the other two sounding oscillators) and the sustain set to the maximum level (hence the decay time has no effect). The release time set to 3.16 seconds and so sounds for slightly longer than the "vocal" oscillators after the note is released.

There is only one velocity sensitive feature in this patch: oscillator five's output. The velocity control means that with harder playing the patch sounds thicker in the lower regions. Given the nature of the patch, I didn't want there to be much in the way of velocity modulation.

The effects are comparatively subtle for this patch. First there is the six voice chorus which adds smoothness, brightness and stereo breadth to the sound. Secondly, there is some plate reverb which adds some smoothness and space.

lush pad

Here's a paid that will melt your CPU. It is also a very dominant pad and so you may not be able to add it to every track you create. This is a comparatively simple pad but it does use the effects units fairly heavily. However, the effects are used to enhance the tone rather than to radically reshape the sound.

The basis of the pad is three oscillators, each with the vintage saw one wave selected. For each of the waves, multi-mode (multi, free) has been selected and the phase set to 90 degrees, so from the start we're using a fair amount of CPU. One oscillator is split between the two busses, one turned predominantly to the first bus and the last oscillator turned predominantly to the second bus. There are two, parallel 36dB low pass filters: each with their cut-off set to zero and the resonance set to 2.59dB. One filter is then panned left and the other right.

To complete this basic set up – as there is no sound with the filters being closed – envelope one was set as the modulation source modulating all of the filters' cut-off frequencies. The range was set to a minimum of 21% and a maximum of 89%. Finally velocity was set as the control. Envelope one was set up as a simple ADSR envelope with attack time set to 0.04 seconds, decay time to 0.51 seconds, the sustain level to 87% and the release time to 3.45 seconds. In addition, in the amplifier envelope the attack time was set to 1.46 seconds and the release time to 3.45 seconds.

This gives the main sound to the patch – to give some more thickness and change the tone a bit a fourth oscillator, vintage square two, was added. This time a single wave, not a multi-mode wave, was called up which is tuned to the same pitch as the sawtooth waves and balanced between the busses.

I only wanted this fourth oscillator to thicken up the tone a bit in the lower regions so I added some key scaling to reduce the level at higher pitches. To do this, in the modulation matrix I set the source to B-NOTE#, the range to full, the curve to B-LIN- and oscillator four's level as the destination.

This is the main sound. To give it a bit more richness and movement and to emphasize some of the inherent characteristics of the patch, in particular some of the velocity nuances, I added some effects.

First, to keep some of the higher frequency elements in check (so keeping the pad thick) I selected the soft drive distortion unit. Here I added some drive (7dB) and some level (23%), kept the tone full and routed the output back to the filters. Next I added the six-voice chorus to give some more swirl to the sound and finally I added a ping delay and some plate reverb.

lush pad developed

I like **lush pad**, but I also wanted to take it in a slightly different direction, hence **lush pad developed** which makes three changes to the basic theme:

- The first change I made was to arrange the filters in series and pan them to the centre.
- Secondly, I swapped the second filter for a 12dB high pass filter.

- Lastly I set the low pass filter to open gently during the release stage giving the impression of a lighter sound and longer sustain.

With the changes in the filter block I switched the modulation matrix destination so envelope one only controls the cut-off of filter one. I then introduced envelope two as a modulation source for the cut-off of filter two.

Filter two's cut-off was kept at its minimum but the curve for envelope two was set to U-LIN-. This means that the envelope works to keep the cut-off low (ie so that the sound can be heard). Envelope two had its attack time set to 2.4 seconds – you will hear that the sound fades in: this is the high pass filter having its cut-off frequency swept from high to low giving the fade effect. The sustain level was set to 100% so the decay time has no effect. Finally the release time was set to 5.2 seconds so that the sound gets progressively thinner during the release phase.

To open the low pass filter during the release stage, thereby making the sound progressively brighter as it decays, I called up the pitch envelope. The pitch envelope does not have to be used only for pitch – in this instance I took advantage of its bipolar behavior. It has no effect until the release stage when it comes into action as a modulation source with a release time of 2 seconds and a release level of 100%.

The pitch envelope acts as a modulation source to filter one's cut-off through the modulation matrix where the range was set to 100%. Unlike all of the other envelopes the pitch envelope's release level can increase during the release phase so the modulation routing in this instance is quite simple.

Pluck and stabs

Rhino plucks

guitaresque

Those of you with the patches will see that I have included a patch called **guitar-unesque**: this is my failed patch. For **guitaresque** I wanted to create something close to the tone of a clean guitar played through a chorus pedal – ideally I was looking for the kind of tone that could be used to play guitar like arpeggios. You will understand that **guitar-unesque** didn't go in this direction so I gave up.

guitaresque has two main elements – a thin, almost sitar like sound and another sound which in certain regions sounds rather like tuned wooden percussion. In combination, we get a wholly new sound which is much closer to a clean guitar sound.

The first part is created by a parallel modulator block. Operator one is modulated by operator two (the modulation amount is 72 with a C:M ratio of 1:3) and operator three (the modulation amount is 74 with a C:M ratio of 2:1). In addition, operator three also modulates operator two with a C:M ratio of 7:1, here the modulation amount is 68.

All of the envelopes in this patch decay swiftly with a curve set to about 25. For operator one, the level envelope decays to zero after about six seconds. For operator two, the level envelope decays to zero after about 0.8 seconds and for operator three, the envelope decays to approximately 50% (where it sustains) after about 1.2 seconds.

For all three operators there is some velocity scaling. The scaling in operator one ranges from approximately 20 to 100, in operator two from approximately 25 to 100 and for operator three from approximately 50 to 100.

The second part of the patch comprises two elements. First operator four is routed through the filter and to the raw output (both set to 100). The waveform selected for this operator is the sine wave tuned an octave below operator one (although not modulating that operator). The envelope for operator four decays to zero over about six seconds.

The prime purpose of the operator is to thicken up the sound – routing it through the filter doesn't change its tone much, but it does double its volume.

The last part to this sound is operator six which is modulated by operator five with a C:M ratio of 1:15 and the modulation level set to 68. The envelope for operator six decays to 15% over approximately 1.2 seconds whereas the envelope for the modulator, operator five, decays to zero over approximately 0.8 seconds (although this operator uses a much steeper curve). This last combination gives a the tuned wooden percussion type sound.

This final simple FM stack is then routed through filter one where analog lowpass 1 has been selected. The cut-off slider was set to 28 and the envelope slider ("cutoff env. mod.") was set to 55 and the cut-off envelope decays to zero over approximately 6 seconds.

To round off the patch, the stereo ensemble and 8 taps reverb were called up in the FX section.

It would have been possible to use a pluck type wave to get plectrum sound, however, I'm not a big fan of this effect. First, I find it hard to control the waves in Rhino to give a plausible effect and second I find that the picking sound can then dominate a patch.

Vanguard pluck

thin

With **thin** I wanted, quite literally, a very thin sound that would work as a plucky sound but wouldn't dominate the mix too much. I wanted a sort of clavinet type tone, but more subdued and with far less bite.

The sound source for this patch is combed square two and combed square three: these are quite bright waves with almost a metallic quality. To simply tame this sound with a low pass filter would not work with this patch – it would only make

the sound dull rather than thin, so for this patch I chose the filter – M – which is a bandpass filter. The cut-off was set to 2kHz, the resonance to 32% and veltrk to 100%.

This filter setting gave quite a dull sound, so envelope two was engaged to modulate the filter cut-off. The attack time was set to zero, the decay time was set to 91ms, the sustain level was set to 7 (on a scale to 127), the release time was set to 142ms and the cut-off amount in the envelope two lane (ie the amount by which the envelope modulates the cut-off frequency) was set to 58.

As a last step, to give the patch a bit more shape, envelope one was called up to modulate the level. The attack time was set to 0ms, the decay time to 57ms, the sustain level to 62, the release time to 104ms and the level knob to 114.

Wusikstation stab

saw stab

With this patch, I wanted to create a big multi-mode type oscillator patch in Wusikstation. There are several reason why you might want to do this. First Wusikstation generally uses less CPU than many synthesizers (this is especially so as Wusikstation has the same CPU hit whether it is using a sample of a single wave or a multi-mode wave) and the second reason for this patch is that I like the sound.

For this patch, the super saw wave from the famous keys two bank was loaded into the first three layers. These are panned left, centre and right – one is kept at its pitch, one fine tuned up by 17 and the other fine tuned down by 23.

Each layer is fed through a low pass 4 pole (24dB filter) with the cut-off set to zero – in layer one, a small amount of resonance (11) was added. The filter cut-off for all layers is controlled by modulation envelope one through the modulation matrix: the source and destination, should be obvious, the minimum and maximum were set to 0 and 127 respectively and the amount for layer one was set to 108, for layer two 99 and for layer three 96.

Modulation envelope one which controls the three layer filters is a simple affair: the attack and decay time were set to zero and the sustain level set to 127. The vel knob was set to 114 which gives a wide range of control over the cut-off of the filters.

Layer one has some resonance set in the filter. I wanted to add some resonance for the filters in layers two and three. However, I wanted the resonance to be greater at lower velocities and less at higher velocities. To do this I called up modulation envelope two and set it very much like modulation envelope one but with the vel knob set to 43. I then set the envelope to modulate the resonance in layers two and three – to increase the resonance at lower levels, I set the minimums to 127 and the maximums to 0. The amounts were then set to 51 (layer two) and 49 (layer three).

As a final tweak modulation LFO one was called up to modulate the fine pitch of layer one (which was the layer without any detuning). The LFO wave is a sine wave and the speed was set to 1/32. In the modulation matrix, the minimum and maximum were set to 0 and 127 and the amount to 58.

The patch was finished off with some effects. All of the layers were sent to FX1 (127) – in this slot the stereo echo was selected with a delay time set to 1/12. No layer was sent to FX2, however, the full output of the echo was fed into the reverber which was called up in FX2. The FX were structured in this way so that they add weight to the patch but are kept further back in the sound field.

Z3TA+ plucking and stabs

bad guitar

This is a very simple patch which has a tone almost reminiscent of a bass guitar being slapped in its higher registers. As a patch it could be developed further, but I like the simplicity and so don't want to change it. However, please don't let me stop you.

The sound source for this patch comprises two waves: in oscillator one the fret wave has been selected and in oscillator two, a sine wave has been selected. The fret wave gives the patch its tone and the sine wave gives the warmth and thickness to the sound without obviously being present.

The waves then run into the filters which are arranged in series (by clicking on the dual button). Filter one is a 36dB low pass filter with the cut-off set to 16Hz (ie fully closed) and the resonance set to 0dB.

To open up the filter, envelope one was set as the modulation source in the modulation matrix with filter one's cut-off frequency as the modulation destination. The range was set to a minimum of 42% and a maximum of 100% with velocity being set as the control.

In envelope one, the attack time was set to zero. To mimic the behavior of the envelope when a string is picked, the slope time was set to 0.34 seconds and the slope level to 54%. To create the decay section of this patch (which does not mimic the decay of a real instrument) the decay time was set to 1.32 seconds and the sustain level to 32%. A bit of shape was also added in the amplifier envelope.

The second filter is a 12dB high pass filter – its function is to remove all of the muddy elements from the patch. The cut-off was set at 180Hz and the resonance at 1dB (which gives a subtle emphasis to the fundamental tone).

The effect of the two filters is to create a great tone, however, they have also taken a lot of energy out of the patch which is remedied in the in the effects section. The tool for this job is the compressor – which also has the effect of evening out some of the harmonics giving a brighter but more rounded tone.

The use of the compressor in this patch lacks any subtlety, however, I like the result. The fast mode was selected, the threshold set to 0dB and the ratio to 50:1 (in other words, the compressor is acting as a limiter). The gain was then pushed up to 12dB and the level set at 100%.

Finally the six voice chorus was added to give some brightness and sparkle. To ensure the chorus added brightness, the hi+ EQ was selected and the hi control pushed up to 4dB.

muted stab

For this pad, I was trying to create a muted stab with a lot of tonal variation. For me, it was important that this pad be muted but still have character.

I did initially start to build this patch in Vanguard – listen to **unwanted stab**. However I abandoned this patch because I didn't like the character and it was too close to **gentle stab**. One factor to note with Z3TA+ is that its patches tend to have quite a dominant character that cut through mixes – you can hear **muted stab** clearly in a mix even though it is a comparatively subdued patch.

The basis of the patch is two slightly detuned vintage saw four waves each with multi-mode (free) engaged (each with the phase slider pushed up to 90 degrees) and panned in opposite directions to about 70/80%. These are then fed into two 36dB low pass filters in series: I'm not certain how often you're going to need a 72dB/octave filter (as is created here). However, for the tone I was after at lower velocities, this combination worked for me. You should, of course, be aware that this combination will use more CPU than a single filter with a gentle filter slope.

In both filters, the cut-off was set at 16Hz (ie no sound passes through) and the resonance was set to 4.94dB. To give the filters some movement, envelope one was set at the modulation source with all filters cut-off as the destination. The range was set to 83% in the modulation matrix and the control is velocity. Finally, I selected the U-LIN+ curve.

The effect of the U-LIN+ curve is subtle – to my mind its effect makes the control of the filter easier. At higher velocities there is little difference, however, at lower velocities it is easier to get (and control) that "muted" tone that I am after.

The envelopes were set as simple ADSR envelopes. Envelope one which controls the filter was set with an attack time of zero, a decay time of 0.21 seconds, a sustain level of 60% and a release time of 0.33 seconds. The amplifier envelope was also set with an attack time of zero and then a decay time of 0.06 seconds, a sustain level of 75% and a release time of 0.33 seconds

To then give the patch some more width, I then panned the first filter mostly to the left and the second filter mostly to the right. This also has the effect of changing the tone of the patch slightly – it is still muted, but probably brighter/warmer with the panning.

Finally to round off the patch I added some plate reverb.

separated stab

On first hearing, **separated stab** sounds similar to **muted stab** which is not surprising being as I copied the earlier patch and tweaked it. However, from a playability perspective, the patches are quite different. The components of the patch are also different.

With separated stab, I was looking for a brighter tone that would be more dominant. However, I also wanted to have more control over this patch.

The setup for **separated stab** is similar to **muted stab**, the key differences are that the octaved square one waves are used (instead of vintage saw one waves) and the filters are run in parallel (not in series).

Envelope one uses the same settings as in **muted stab** although there is a change to its deployment through the modulation matrix with the range being set to a minimum of 29% and a maximum of 84% (and remember that the filters work in parallel, not in series for this patch).

The tone of this patch was good, but it lacked some of the thickness of **muted stab**. To rectify this I added a third oscillator using the vintage saw one wave. This gave a thicker tone, but to my mind the effect was more pleasing when the pitch of this new oscillator was raised by two octaves. The third oscillator also robbed the patch of some of its subtlety. To remedy this and control the operation of oscillator three, I turned to the modulation matrix.

In the modulation matrix, I added a line with the source set to "on", the range to maximum, the curve to slow+, the control to velocity and the destination to oscillator three. This means that oscillator three has less effect at lower velocities (so the sound is less fat) but more effect at higher velocities (making the patch thicker).

To give some more tonal variation to the patch – and to reduce some of the power of the resonance – I added another line to the modulation matrix. The source was set to "on", the range to 52%, the control to modulation wheel and the destination was all filters separation. By separating the filters more tonal changes can be uncovered without robbing the sound of its essential tone and volume. This change can be controlled in real time with the modulation wheel or automated.

The sound was still a bit thick too my ears. As there were no filters available, I turned to the EQ section and selected the wide one EQ mode and dropped the 80Hz slider by -18dB. The effect of this tweak is to take out some of the low end from the patch leaving more room for the bass and the kick drum in the mix.

Finally I added some plate reverb to taste.

Chapter 13: the synths

The examples in this book have been based on five software synthesizers.

I have focused purely on software for several reasons. First, this is the way the world is going: even "hardware" synthesizers are usually just software with a box wrapped around them. Secondly, software is far more accessible for most readers.

The synthesizers I have chosen are some of the best that are currently available. They have all been built by talented, passionate developers who have designed something unique and brilliant. These developers are all musicians who have built a synthesizer that they want to see built. These are not instruments that have been built by some faceless corporation to fit a perceived need or to come in at a certain price point. You will find that if you need help with your synthesizer, you will get help directly from the developer: usually in hours, sometimes in minutes.

This book is not a review. You will see that I have highlighted many of the positive aspects about all of the synthesizers: this book is about how to do things – specific aspects of the synths in question have been used to illustrate my specific points. It is not a review and my intention is not to highlight any features that you may perceive as a weakness.

I am sure that if you talk to the five developers they will tell you that I haven't highlighted some really cool features of their synths and they are right in this assertion – this is not a book about synths: it is a book about sound design. If you are looking for book about how not to do things, the limitations of software synthesizers and what you wish software would do for you, then this is the wrong book for you.

If you don't already own all of the synths, I suggest you do. If nothing else, get hold of the demos and find out how each synth really works. The patches will work with the demos (although for Wusikstation some of the waveforms used in the patches are only available in the purchased version).

The summary below only lists the main features of each of the synths. The prices listed are those quoted on website at time of publication – note, some prices are quoted in US$ and others are quoted in EUR€.

Cameleon 5000

developer: Camel Audio

website: http://camelaudio.com

price: $199/€159

Cameleon 5000 was one of the first software based additive synthesizers and, I believe, was the first additive synthesizer to offer practical resynthesis.

At its heart is the morph square which allows users to morph between four sounds at once (including user imported sounds). Once imported, sounds can then be edited in ways which are impossible with a conventional sampler, for instance there is control over each of the harmonics. As you would expect there are the usual LFOs, envelope controls and FX units.

Definitely one for the creative and the curious.

Rhino

developer: Big Tick

website: http://bigtick.pastnotecut.org

price: €100

Rhino comes with six oscillators (offering more than 130 built-in waveforms and the ability to import your own) and a built in additive waveform generator. It has two filters and countless graphical multipoint envelopes with curvature controls. Add in a step sequencer and one variable wave shaper per oscillator and a visible output routing matrix and you have a hugely powerful machine.

However, even with all this power, for me, Rhino is still the ideal FM machine.

Vanguard

developer: reFX

website: www.refx.net

price: $89.99

At first look, Vanguard appears to be a simple subtractive synthesizer and it is certainly capable of producing warm, rich sounds. However, it comes with 31 different oscillators and a selection of 13 filters (including some combined dual filter variations) giving it a wide range of sounds.

A key advantage of Vanguard is that you can see what's going on – virtually all of the controls are accessible at the same time. The only controls that aren't immediately accessible are the second and third LFOs.

Vanguard is incredibly easy and fast to program and creates great sounds: for me that often makes it a first call synthesizer.

Wusikstation

developer: Wusik dot com

website: www.wusik.com

price: $99.95

Wusikstation is quite a unique synthesizer with six sample based layers. Each layer can load separate multi-sampled soundsets which is can then be processed through up to four filters. As it is sample based, the waveform options are effectively limitless. Each layer has its own envelope and there are also eight modulation envelopes and eight modulation LFOs which can control a variety of destinations through the modulation matrix.

The two wave-sequence layers offer possibilities for the creation of unique and complex sounds. Its low CPU usage and ability to create compelling sounds makes Wusikstation a versatile synthesizer which can be used in many musical situations.

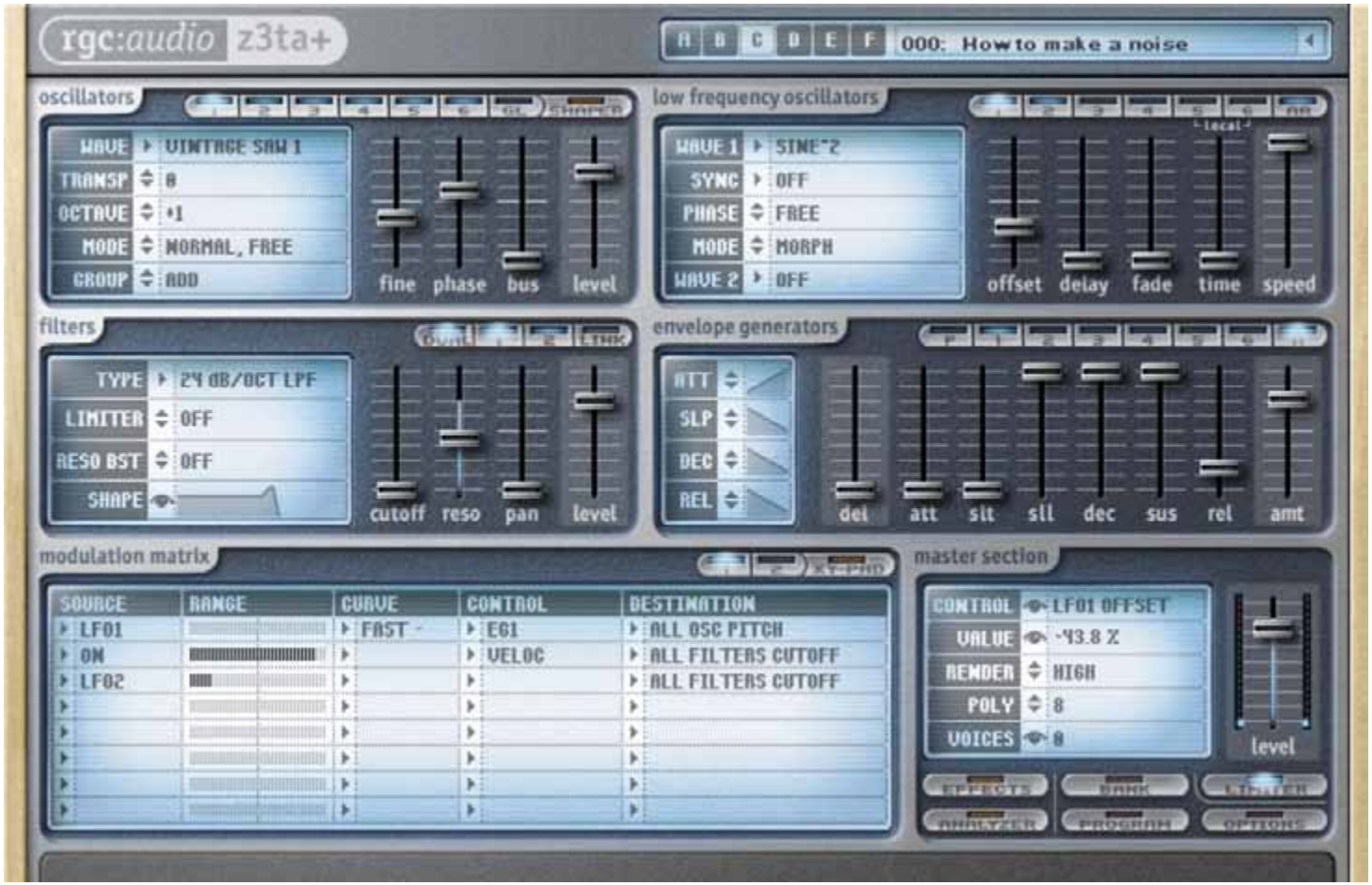
Z3TA+

developer: Cakewalk

website: www.cakewalk.com

price: €149

You could pretty much fill a book just talking about the features of Z3TA+. Behind the unassuming GUI is a real heavyweight synthesizer.



So what does Z3TA+ offer? Full stereo processing. 6 oscillators, 60 built-in waveforms, 6 user waveforms, PWM possible for all oscillators with any waveform. Independent waveshaper for each oscillator with 14 wave transformations. Multi-mode for any oscillator. Individual settings for pitch bend up and down (-12 to +12 semitones). 2 Stereo filters. 6 morphing capable LFOs. Eight 6-stage envelope generators. Arpeggiator/sequence player. Modulation matrix. Many effects. 768 program capacity (in six banks). To read the full list of features, check out the website.

There are two main reasons to use Z3TA+. First, and most obviously, the sound: this is perhaps the benchmark against which all other software synthesizers are judged. Second, it can do pretty much anything. Its depth (in sound and features) and usability for musicians make it a synthesizer that should be in every musicians arsenal.