The Lost Art Of Sampling: Part 6

By Steve Howell

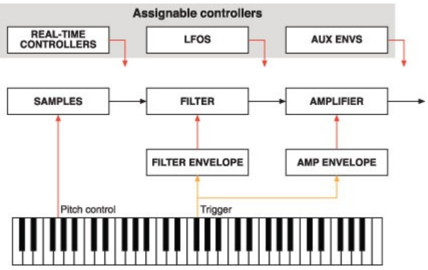
We continue looking at what your sampler's synth engine can do to liven up bland self-sampled sounds, and explain the concepts of layering and multitimbrality.

In *Part 5*, I introduced the powerful synth engine features present in most samplers. Ironically, these days, when samplers have more sound-shaping capabilities than ever before, these features are remaining unused by many modern sample-library developers, who are increasingly using multi-layered velocity-switched samples instead of the built-in synthesis capabilities. The increasing realism of these sounds comes at a price, of course, in terms of memory load and processor strain, and it has also contributed to many people feeling there's no point sampling their own sounds any more, as their own efforts inevitably sound 'flat' and 'lifeless' compared to those in commercial libraries. However, by using the oft-ignored synth facilities, even short and often static-sounding self-made samples can be brought to life quite dramatically, as we'll see in this article.

From Dull To Dynamic

Most samplers present users with a fairly traditional synth-type signal path, except that samples are used in place of the usual oscillators — see the block diagram on the right. They're not shown, but typically the final output amplifier will also feed a set of multi-effects for adding the final polish to any sound.

If I have a dull, bland-sounding self-made sample, the first thing I do to make it more interesting is give it some 'shape', a sense of development over time, by reaching for the amplitude envelope controls. Surprisingly enough, simply adjusting these, combined with mapping playing velocity to amplitude, can often transform an otherwise 'lifeless' sound into something quite playable. If the sample you've taken is very short, and you've artificially extended it by using some of the looping techniques described in the fourth instalment of this series, you can even make a difference to how good the loop sounds with the envelope. If the loop is imparting a 'static' quality to the sustain portion of the sound, nudging the level of the sustain portion of the envelope down will help to make it less so.



A highly simplified block diagram of a sampler (or indeed any modern sample-based synth or workstation). Many also feature multi-effects before the final audio output.

Once I have something that's playable, the filter is usually my next port of call. Whether the sounds are sets of acoustic samples or samples of synths, real drums or drum machines, I've found that they nearly always benefit from having playback velocity applied to control the filter cutoff. This is simple to achieve by reducing the filter's cutoff frequency a little, and slightly increasing the extent to which playback velocity affects the cutoff. Working with (say) a control parameter range of 0 to 100, setting the cutoff to 80 and the velocity-to-cutoff amount to +20 will give a modest degree of tonal control, and you could be more extreme if the situation demands it.

These programming routings give a dull, flat-sounding sample a pronounced envelope, with velocity control of amplitude and timbre — much like most acoustic instruments! Additionally tweaking the filter's envelope can also further help to minimise any dodgy-sounding loop points. Sweeping the filter cutoff very slightly with an envelope varies the tone of the sound over the course of a note, and this timbral movement can help to obscure the loop point, or even render it inaudible. If this helps, but the 'static' nature of your loop is once more revealed when the filter goes into the sustain portion of its envelope, you can reach for LFOs to add some amplitude and timbral movement to the sound during the sustain period.

For example, by applying some slow sine-wave modulation to the amplifier, the sustained level will rise and fall slightly, adding some subtle interest to the sound. To prevent the modulation being present all the time, you can usually set an LFO delay, so that it only kicks in during the sustain cycle of the sound. Only small amounts of LFO amplitude modulation should be applied, just to make the sound subtly fluctuate in level and add some movement and interest without it sounding too artificial. The same subtle LFO modulation trick can be applied to the filter in moderation if you wish to create a similarly subtle variation in tone during the course of a sustained note. However, it's important not to modulate amplitude and filter cutoff with the *same* LFO — the ear can quickly pick up on this repetition and, in some cases, it will actually highlight a loop rather than suppressing or masking it. Fortunately, most modern samplers offer you at least two LFOs to play with — Native Instruments' *Kontakt* allows you to add as many as you wish! If you use separate LFOs for filter and amplitude modulation, and set them to slightly different rates, you can achieve asynchronous tonal and amplitude modulation that rarely (if ever) repeats in an audible fashion.

Sometimes, depending on the nature of your sample, it can even be appropriate to add further movement to a sound by using an LFO to slightly modulate pitch. However, our ears are particularly sensitive to such changes, even very small ones, and it can be over-obvious that the sound is being artificially 'wobbled', so this technique has to be used with care. Similarly, sounds can be enlivened and loop points further disguised or 'smudged' by adding multi-effects such as reverb, chorus or delay, but care must be taken not to overdo it and make the sound too 'artificial' or 'processed' (unless that's what you're after).

When coupled appropriately with the kind of synth-engine tricks I described in the third instalment of this series, such as mapping playback velocity to the sample's start point, or velocity to attack time (and modern samplers allow you to control all kinds of other sound-shaping parameters, such as decay, sustain level, release times, and resonance), even the dullest of source samples can be made much more expressive and interesting, and without having to use umpteen memory-hungry velocity layers in the sound.

A Little History — Multitimbrality & Samplers

Samplers were fairly slow to adapt to what we would now regard as multitimbrality, although of course you could map different samples across the span of a keyboard from the Fairlight onwards. When Akai first dabbled with the concept on the S900, they implemented it in a very strange (and not very intuitive) way.

You had to set up a Program that contained all the samples you wanted in your 'Multi' and then set their Keygroups to the MIDI channel you wanted them to play on! It was very tedious and difficult to use.

Things improved with the S1000 — you could renumber Programs so that they had the same Program number, and then set each Program to have its own MIDI channel. Thus, if 'Drums', 'Bass' and 'Piano' were all renumbered to be Program number 1, when this Program was selected, all three Programs would become active at once. It was fairly cumbersome to set up, but it worked well enough, especially live, where a single remote MIDI Program Change command could recall a complete multitimbral/layered collection of Programs.

However, synth manufacturers had already established the concept of the Multi mode, where Programs or Patches are assigned to 'slots' on different MIDI channels, and Roland introduced the same simple concept on their S-series samplers in 1986. Other sampler manufacturers followed suit, but it took Akai longer to come around. A 'proper', synth-like Multi mode wasn't introduced until the S3000XL arrived in 1995, and even then, an old-style 'renumbering' multitimbral mode was retained until the release of the S5000 and S6000 in 1998, which offered only a 'proper' Multi mode capable of handling up to 128 parts. Even then, many die-hard Akai users complained about the 'new' implementation, and a conversion function was provided to convert the old Multis into the new format.

Killer Composites

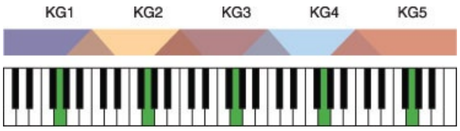
Creators of synth and sample libraries often layer several samples together to create great composite sounds — a killer orchestral string sound may well be made up of several different string samples, detuned, panned and mixed to create a lush ensemble sound — and there's no reason why you shouldn't do the same with your own self-made samples. Such layering is easy in a modern sample-based synth, where you just select different (say) sampled string sounds in several different oscillators. With a sampler, it's not quite so straightforward, but it's far from impossible.

We've already looked at Keygroups as a means of creating multisampled sounds, where you might have one instrument sample handling a range of three or four semitones across the keyboard, and then a different source sample handling the next few semitones in the next Keygroup up. But by deliberately overlapping the Keygroups, you can layer several samples so that they are triggered together by the same keys.

A good way of building a bigger, lusher sound is to place samples on top of each other, detune each one (one slightly up, one slightly down, for example) and maybe pan each one to the left and right respectively to create a 'pseudo-stereo' effect, especially if the samples are mono. This technique works well to fatten electronic-sounding synth tones, but it tends to sound artificial if you try it on 'acoustic' sounds such as string or choral ensembles, especially if your self-made samples have very similar attack portions and loop points. In such cases, it's better to layer two different but similar sounds, such as an orchestral string sound with a smaller ensemble string sound, 'real' strings with a warm synth string sound, or a male choir with a female or synth choir patch. When you try this, you often find that detuning (and the phasing/chorus effect it inevitably creates) is no longer necessary to fatten the sound, and you can create composite samples that sound much greater than the sum of their parts. It's by using such techniques that the creators of factory libraries for synths can make such great sounds using such relatively minimal sample-sets as their source material.

Positional Crossfading

One of the biggest problems with multisampled instruments and mapping them out across the range of a keyboard is that there can often be an abrubt tonal change as you pass from one Keygroup to another. If your sampler permits this, one way around this is to use a Positional or Keygroup crossfade (as usual, the precise terminology varies from sampler to sampler). When you do this, the ranges of your Keygroups overlap, and as you play up or down the keyboard in the area of the transition from one Keygroup to another, one Keygroup sample fades out as the other fades in, resulting in a smoother transition between Keygroups.



A diagrammatic representation of a simple Program with samples on the 'G' of every octave, but using Keygroup overlap/crossfade to smooth the transition between adjacent Keygroups.

This feature is implemented differently on different samplers. On some, such as Akais, you simply overlap the Keygroup ranges and switch on the Keygroup Crossfade function. Then, if your Keygroups overlap, they are automatically crossfaded, and if they don't, they behave as normal. Modern software samplers allow you to overlap the Keygroups and then drag handles with the mouse so that crossfades can be set independently for each Keygroup as required. This is more flexible, but equally, it's more time consuming to set up.

As always, however, there can be pitfalls for the unwary. Firstly, if there is a slight pitch difference between the samples in the overlapping, crossfading Keygroups, you might hear anything from a mild flanging effect to a stronger chorus effect in the overlapping region. You can overcome this by tuning one of the Keygroups slightly, but then you run the risk of trouble with the tuning of all the other Keygroups!

Secondly, if there is any 'movement' in the sound (such as natural vibrato, for example), then when the sample is transposed as you play higher up the Keygroup, that movement speeds up. The problem is that in a Keygroup crossfade, the adjacent sample is likely to be playing more slowly, so you can get a jumbled mess as the two interact. In difficult cases like this, you simply have to decide whether this is preferable to the abrupt transition that might occur without any Keygroup crossfade in place.

Multi Mode

I mentioned in ***Part 5*** that I felt samplers to be the ultimate synthesizers, because they can employ any sound as their sonic starting point. They can replay acoustic sounds very accurately, or process and transform them, they can operate like complete synthesizers in their own right, and they can be used to chop up, stretch and mangle drum (and other) loops, and trigger long recordings or spot sound effects. And all of this power can be brought together in the Multi Mode found in all modern samplers.

In this mode, the various single sample-based Programs we have created can be combined, and each one can be played on a different MIDI channel. For example, you could have a sampled piano sound on part 1, on MIDI channel 1; sampled drums in Multi part 2, on MIDI channel 2; bass (synthesized in the sampler from raw synth waveforms) in Multi part 3, on MIDI channel 3... and so on. But unlike on a typical multitimbral synth, these individual multitimbral Parts are not limited to purely instrumental sounds, and could contain anything: *Recycle* loops, vocal 'spin-ins'... whatever you like.

Unlike other software instruments, where you insert different instances on different sequencer tracks each time you want to use them, most modern software samplers operate in a form of Multi Mode by default, and Programs are played simply by assigning them to a Part (usually a 'drag and drop' operation) and addressing them on the appropriate MIDI channel. However, unlike hardware samplers, which typically have to share their internal effects complement across all their multitimbral parts, multitimbral software samplers typically allow you to assign effects at the Program level, so that each part can have its own effects tailored for the sound (provided the power of the host CPU can handle the processing load of all the effects, of course!).

However, whilst most people associate Multis with multitimbral sequencing (where different sounds are played on different MIDI channels), they can also be used more simply to layer sounds. If you assign two Programs to two Parts, and set those parts to the same MIDI channel, the two Programs will play together. This is a great way to create composite sounds, as described earlier, and it saves the tedium of having to layer individual samples at the Program and/or Keygroup level. Most samplers have the ability to individually tune and pan the multitimbral Parts (the latter function allowing you to create 'pseudo-stereo' effects on mono samples). The individual Parts in the Multi can, of course, be mixed and balanced, and most samplers also allow you to set key ranges for each Part, too, so you can set up keysplits, where different ranges of the keyboard trigger different sounds. Whilst this may all seem a little complicated at first, it allows a great deal of flexibility when combining sounds in different ways.