

SimSynth

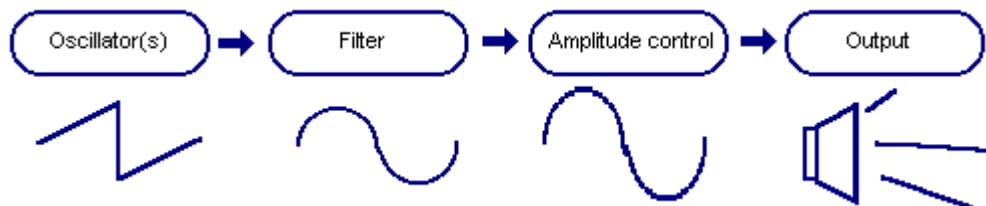
Copyright© 1994 by David Billen

How to Program a Synthesizer

This is a "HOW TO" for general synthesizer programming. For an introduction and overview to SimSynth, see the INTRO.WRI file. For technical reference, see the help file which is accessible from within the program...

Sound synthesis comes in many forms, but each form can be placed into one of two categories: additive synthesis and subtractive synthesis. We use subtractive.

Subtractive synthesis is like making a clay sculpture. The artist starts with an ugly blob of clay, and then removes, (subtracts), all of the clay which doesn't belong in the final piece. So it is with analog synthesis. The artist, (that's you), begins with an ugly blob of sound created by an oscillator. You then chisel away the undesirable bits of sound using a filter. The amplitude of the sound is often altered by something called an envelope, and finally it is sent to the output.

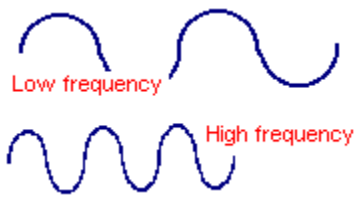


Wave shapes...

When thinking about sound synthesis, it helps to picture sound waves as plots on a two dimensional graph. This is the way that waves are usually shown by wave editors, (and SimSynth). This is the way that technicians always think of sound. Not like the expanding circle of vibrations from a science class, but rather like the scribbling from a seismograph. Technicians use the phrase wave shape. They are referring to the picture that would be drawn by a seismograph, or the picture that might be shown on an oscilloscope if you were monitoring the sound passing through an electronic circuit.

The greater the width, or horizontal distance between up and down cycles of the wave shape, the deeper the pitch of the sound. If you looked at the wave produced by one second of high pitched sound, the up and down cycles would be more frequent than they would be for one second of low pitched sound. Therefore, low pitched sounds have a lower frequency, and high pitched sounds have a higher

frequency.



The greater the height, or vertical distance between high and low points of the wave shape, the louder the amplitude of the sound.



Oscillators...

Traditional analog oscillators are electronic circuits which produce a wave with a very simple shape. It is standard to select from one of three: a square, (or pulse) wave, a sawtooth wave, or a triangle wave. If you listen to an oscillator by itself, all of the wave shapes make a belligerent sound, but the square wave is the brightest, the sawtooth is a little darker, (has a lower tone), and the triangle is the darkest. It is also common to select noise as the wave shape. This is random "white" sound, and is mostly useful for special effects.



Note: You may be wondering what it is about these waveshapes, square, sawtooth, and triangle, that make them the natural building blocks of sound. Probably nothing! They are simply easy to generate with cheap circuits. Consider generating a square wave as a project for your Archer 101 in 1 electronic projects kit. You need to create a high signal, then a low signal, then a high signal, then a low signal. In fact, This may be the Archer 1 in 1 electronic project kit.

There are some standard settings which are normally available to configure the sound of an oscillator. As discussed, one must first select the wave shape. Here are some of the other settings:

Pulse width is sometimes only available if the selected wave shape is a square wave. It alters the wave shape by adjusting the ratio between up and down cycles. Note that this is not the same as adjusting the frequency, (because the overall distance between full cycles is not affected). The reason for wanting to adjust the pulse width, is that it gives the oscillator a slightly different sound. (Any type of control that changes the sound is automatically a useful thing on a synthesizer).



Frequency is usually an amount to add or subtract from a base frequency. In traditional synthesizers, (which are controlled by a music keyboard), the base frequency is the note that the musician is playing, and the frequency setting is added to it. Musicians rarely want to be out of tune, but the frequency setting is useful because most synthesizers, (including SimSynth), have more than one oscillator.

LFO and envelope are discussed later. They concern the changing of the frequency over time.

Filters...

It's been mentioned that the oscillators make a harsh, ugly sound. This is intentional. The oscillators are the initial blob of clay from our sculptor metaphor. They are full of extra sound, which you can remove using a filter.

There are several different types of filters. Far and away the most common is the low pass filter. Indeed, early synthesizers and inexpensive synthesizers often have only a low pass filter. The name is "low pass" because it lets low frequencies pass through, and filters out high frequencies. (This is the function of a basic "tone" control found on old radios).

It is likely that the majority of sounds you create for a synthesizer will use only a low pass filter. Why? Because the "extra sound" made by an oscillator is of a higher frequency than the specified frequency.

Tones are made up of a *pure* tone at a base frequency, plus some number of pure tones at higher frequencies. Any pure tone has a specific shape called a sine wave. A square wave, for example, can even be thought of as a sine wave of the specified frequency, plus a number of quieter, higher frequency sine waves. If you add all of these sine waves together, and look at the resulting wave shape on an oscilloscope, it will look like a square wave. (In fact, there's a mathematical formula for analyzing any tone, and finding the sine waves it is made of).



Lucky for us, we don't have to care. This has only been explained to make sense of that fact that even though you set a specific frequency for an oscillator, there are other, higher frequencies present in the resulting sound. But, these can be gotten rid of using a low pass filter.

Note: If you wonder what a sine wave might sound like: The test pattern which is broadcast by television stations after hours, or for tests of the Emergency Broadcast System, are sine waves.

There are some standard settings which are normally available on a synthesizer to configure the sound of a low pass filter:

Frequency, (or **cutoff**), is the frequency at which the cut off occurs. In other words,

frequencies beneath the frequency specified for the low pass filter are passed through, and higher frequencies are filtered out. (Actually, most low pass filters are "soft". Beginning with the specified frequency, high frequencies are merely reduced in amplitude, more so the higher the frequency. A frequency usually has to get really high to be filtered away completely.)

Emphasis, (sometimes called resonance, and sometimes called "Q"), allows you to amplify frequencies which are near the cutoff frequency. Turning the emphasis very high results in a wave shape that is not common in nature, and therefore has an electronic sound. (In fact, boosting this setting probably gives the most characteristic sound of an analog synthesizer.)

LFO and envelope are discussed later. They concern the changing of the frequency over time.

There are other types of filters. A High pass filter is the opposite of a low pass filter. it allows high frequencies to pass through, and blocks out low frequencies. A Band pass filter passes frequencies near its specified frequency, and blocks out frequencies that are both higher and lower. The most obscure is the Notch filter. It is the opposite of a band pass filter.

Note: SimSynth uses something called a State Variable Filter. This generates low pass, high pass, band pass, and notch all at the same time. You get sliders to specify the mix of filter outputs that you want to hear.

Envelopes and LFOs...

Armed with oscillators that are sending their output through a filter, you can create a great variety of sounds. Each will be more irritating than the last! This is because no sound which simply plays at a set frequency and amplitude is interesting, (regardless of how skillfully filtered). Envelopes and LFOs are the answer.

Envelopes allow you to specify how the sound changes through time. Envelopes can be used to specify oscillator frequency changes, filter frequency changes, and overall amplitude of the sound changes. For example, you can specify the amount that an oscillator will follow the envelope. As the envelope changes, the frequency of the oscillator will change with it. It is most common to have the amplitude and filter frequency follow the envelope. (Having the oscillator frequency follow the envelope is normally reserved for unusual effects).

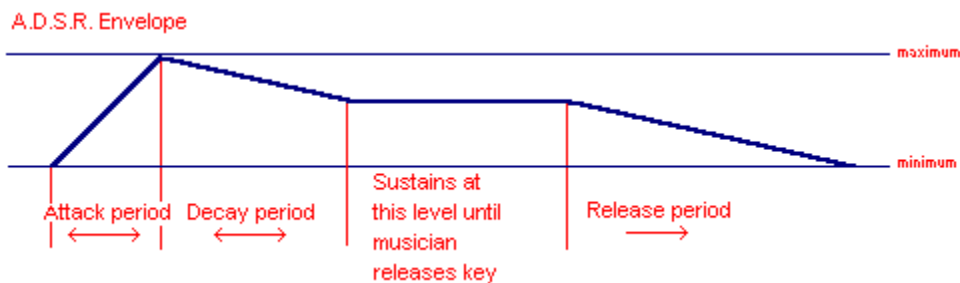
The traditional method for configuring the envelope, (though SimSynth doesn't use it), is the ADSR envelope, which stands for attack-decay-sustain-release. In older synthesizers there were typically four knobs used to set these four values.

Synthesizers have been mostly used for musical applications, and the ADSR envelope is linked heavily to the activity of a musician at a music keyboard. When the key is pressed, the attack specifies the amount of time it takes for the envelope to rise from zero to full value. The envelope then falls to a level specified by the sustain value, over an amount of time specified by the decay value. As long as the musician holds the key, the envelope remains at the sustain value. When the musician releases the key, the envelope falls to zero in the amount of time specified by the release value.

We talk about the envelope reaching full value, but this doesn't mean that if the

envelope is, (for example), changing a filter frequency, that it is changing the frequency over its full range. Instead of using a simple switch to indicate whether or not the filter frequency should be changed by the envelope, you use a slider to specify exactly how much you wish for the filter frequency to track. (You can even set it to a minus value, and as the envelope goes up, the filter level goes down!)

For example: the amplitude of an organ sound should be controlled by an envelope with no attack, no decay, full sustain, and no release. A piano sound should have no attack, but a fairly lengthy decay down to a sustain of zero, (since a piano note eventually fades to nothing, even if you hold the key forever). For all effective purposes there would be no release time, since a piano note cuts off when you release the key, (though you may want a subtle smidgen of release because a real piano can't completely stop the sound as quickly as a synthesizer can). If you set the release time as long as the decay time, it would sound like a piano with the sustain pedal pressed. Finally, a weepy string sound is a typical use of a long attack, a moderate decay to a moderate sustain level, then a long release time.



Many later synthesizers, (Roland™ synthesizers for example), departed from the ADSR model in order to achieve more flexibility.

Note: SimSynth departs from the ADSR, mostly because it is not controlled by a musical keyboard. SimSynth's envelope simply has five stages. For each stage you can specify an amount of time, and an envelope level to be reached in the specified amount of time.

LFO stands for Low Frequency Oscillation. LFOs operate much like the sound emitting oscillators, except they are extremely slow by comparison, and instead of listening to them, they are used to control the same things controlled by the envelopes. For example, an LFO controlling the frequency of an oscillator, very subtly, provides tremolo. LFO subtly controlling the amplitude of the sound gives a vibrato effect. Great amounts of LFO are useful for special effects.

Effects...

SimSynth contains some built in effects. Older analog synthesizers, (after which SimSynth is modeled), usually did not have built in effects, (though newer digital synthesizers usually do).

The effects are just that, effects. It is a very simple matter to try each of them, and hear what they sound like.

Tips and Conclusions...

- The other manuals show some differences between SimSynth and the traditional analog synth which has been described.

- Try loading the presets which came with SimSynth to examine how they were done.

- When making your own presets, be careful. Every parameter setting matters. (Envelope control over oscillator frequency is a good example. It can completely alter a sound, even at a minimum setting).
- If you have read this, you really do know as much as you need to get started. There are some sliders and buttons on SimSynth that have not been discussed, but you now have enough overview to consult the reference manual. To gain a real working knowledge, a lot of experimenting is required.
- Some synthesizer programmers never create a sound from scratch. They always load a similar sound, and then customize it.