

# **The Definitive Guide To Evolver**

By Anu Kirk

# The Definitive Guide to Evolver

## Table of Contents

|  |     |
|--|-----|
| Introduction.....  | 3   |
| Before We Start.....   | 5   |
| A Brief Overview.....  | 6   |
| The Basic Patch.....   | 7   |
| The Oscillators.....   | 9   |
| Analog Oscillators.....                                      | 9   |
| Frequency.....   | 10  |
| Fine.....  | 11  |
| Shape/PW.....  | 11  |
| Level.....   | 16  |
| Glide.....   | 16  |
| Special Analog Oscillator Features.....                      | 17  |
| Digital Oscillators.....                                     | 22  |
| Shape.....   | 22  |
| Exploring the Digital Difference.....                        | 24  |
| Frequency Modulation.....                                    | 26  |
| Ring Modulation.....   | 33  |
| Shape Sequence.....  | 36  |
| Some Final Digital Oscillator Tips.....                      | 36  |
| Noise.....   | 37  |
| The VCA and Envelope 2.....                                  | 38  |
| ADSR Explained.....  | 38  |
| The Filters.....   | 42  |
| Lowpass Filter.....  | 42  |
| Env1 – The Filter Envelope.....                              | 46  |
| Highpass Filter.....   | 50  |
| Let it Rip!.....   | 54  |
| Distortion.....  | 54  |
| Feedback.....  | 55  |
| Grunge.....  | 56  |
| Output Hack.....   | 56  |
| The Digital Delay.....                                       | 58  |
| Basic Delay.....   | 59  |
| Advanced Delay: Making Effects.....                          | 61  |
| Master class: Karplus-Strong synthesis.....                  | 66  |
| Modulation.....  | 68  |
| ENV 3 – The Mod Envelope.....                                | 68  |
| Low Frequency Oscillators 1-4.....                           | 70  |
| Modulation Slots 1-4.....                                    | 72  |
| The Sequencer.....   | 75  |
| What is it?.....   | 75  |
| Parameters.....  | 77  |
| Using Evolver’s Sequencer.....                               | 79  |
| Traditional “Step Sequencer”.....                            | 80  |
| “Motif/Arpeggiator” live playback function.....              | 87  |
| Modulation Source for Evolver Patches.....                   | 91  |
| Using Evolver’s Sequencer to Control Other MIDI Devices..... | 93  |
| Making Sounds.....   | 95  |
| Drum Sounds.....   | 95  |
| Drum Beats.....  | 98  |
| Using Evolver Effectively (As an Effects Unit).....          | 99  |
| External Input Mode.....                                     | 100 |
| External Input Volume.....                                   | 100 |
| Input Hack.....  | 100 |
| Processing Audio.....  | 100 |
| External Modulation.....                                     | 101 |
| Guitar Amp Simulator.....                                    | 101 |
| Conclusion.....  | 102 |
| Appendix A: MIDI notes to Sequencer values table.....        | 103 |
| Appendix B: Tempo Clock Multiplier Note Lengths.....         | 104 |
| Appendix C: Default Waveshapes.....                          | 105 |
| Appendix D: Tuned Delay Values, Notes, and Frequencies.....  | 106 |
| Acknowledgements.....  | 107 |

## **Introduction**

---

Writing manuals for products is difficult, thankless work. A good manual clearly explains every single function of the product in question, and does so in terms the product's intended audience can clearly understand. For complicated products aimed at a wide range of users, it is often tough to decide what level of detail to include. Too much, and your advanced users stop reading. Too little and beginners are left confused.

It is thankless, because many users don't even bother to open the manual! Especially in the music world. Users leave the manual in the box or throw it in a corner. That is, until they get frustrated with an interface or feature they can't figure out. Then they go after the manual with a vengeance, and woe to the company that can't answer their questions within the first few paragraphs!

Sadly, the industry doesn't help much, either. For too many products, especially software, manuals have become an afterthought, or nonexistent. Printed manuals are an endangered species, hunted almost to extinction by the dreaded PDF.

Including a printed manual adds substantial cost – even “cheap” manuals are pricey to make and bind, and they can add substantially to a product's shipping weight. And why bother going to all that trouble when, as noted above, your research (both empirical and anecdotal) shows that users aren't going to read it? Market research and consumer behavior tends to show that price trumps all. If you can skip on the manual and drop your street price by \$10-\$50, well, you'd be crazy not to.

All these factors contribute to the dearth of good user manuals out there.

The best user manuals educate as well as elucidate. They inspire as much as they inform. They even provide knowledge useful beyond their own product. If you've ever read a truly excellent manual, you know what I mean. If you haven't, go find a manual for a Mackie mixer. The PDFs are online. Read one. They write great manuals.

As a product designer, I always feel it is important to include a manual appropriate to the depth of the product. I have been irritated far too many times by excellent, deep, complex products that not only shipped without a substantial printed manual, but often failed to come with a detailed manual at all! (Native Instruments, this would be the part where you slink away in shame.)

I decided to write this guide because several users had expressed dissatisfaction with Dave Smith's manual for Evolver. Personally, I found Dave's manual to be good for what it was – an engineer's detailed breakdown of his product's features. But I also felt a little hungry after reading it, and a little frustrated.

## The Definitive Guide to Evolver

The more I dug into and used Evolver, the more I really appreciated the quality of the product's design. Having been messing around with synthesizers for 20 years, I immediately found myself being able to apply all my old tricks and get great results, and continued to be surprised as new depths were revealed.

Evolver is one of the most flexible and versatile synthesizers ever made, and given its price and value I have no doubt it will, in time, become truly revered by all the hardcore synth geeks out there.

But I felt the existing manual does Evolver a disservice. I was concerned that people would "give up" on this great little box simply because there was no one out there to show them how to make the most of it.

So here you go. I am certainly not going to claim this guide is definitive. I think of it as a companion piece to Dave's manual. I hope you enjoy reading it as much as I enjoyed writing it.

And I hope it inspires you to do two things. One is, of course, to make some interesting racket with your Evolver.

The other is to make sure that when you buy products, you read the manual! And if it's good, let the manufacturer know. There's likely some poor tech writer responsible for it, and your note will make their day. And if it is terrible, let them know that as well.

Now get your Evolver fired up, and let's get started!

## Before We Start

---

This guide makes a few assumptions:

**You have a Dave Smith Instruments Evolver.** I wrote this thinking about a “classic” mono Evolver with the 2.0 operating system upgrade, but I believe pretty much everything applies to the Poly Evolver as well.

**Evolver is properly connected to power.** Radio Shack and other stores sell these neat little switches that go between a cord and a wall socket that let you turn stuff on and off. I use one between my Evolver’s power adaptor and the extension cord it is plugged into. Voila! Instant power switch.

**You can hear the output.** Yeah. If you haven’t got that going on, you’re going to need a lot more help than I can provide.

**You have somehow connected some sort of MIDI controller** to drive the thing. Whether it’s plugged directly into the back or through some sort of MIDI interface doesn’t matter. You should be able to press a key and have the Evolver make some noise.

**You are vaguely familiar with synthesizers and terminology.** Presumably nobody just picked up an Evolver “for the hell of it,” without ever having owned or read about synthesizers before. But you never know. I will try to explain all jargon and terms the first time they are mentioned, but I am inherently assuming some knowledge.

**You should also read Dave’s Evolver Manual.** You may want to keep it nearby for reference so you can match parameter names and values to what Evolver’s LED display shows.

I have tried to make this guide relevant whether or not you have the Soundtower Evolver Editor software, and have included screenshots where it is helpful.

### A Brief Overview

---

Evolver is a powerful, true-stereo subtractive synthesizer with extensive modulation routing capabilities and an integrated digital delay and a 16-step sequencer.

This guide will walk through all of the major elements of Evolver. Let's start with a brief overview of what they are:

**Oscillators** – Oscillators generate the basic sound (timbre and pitch) for Evolver. Evolver has 4 of them – 2 digital, 2 analog. One digital oscillator and one analog oscillator are assigned to the left channel, the other ones are assigned to the right channel.

**Filters** – Filters help shape the timbre and volume by removing frequencies from the “raw” sound of the oscillators. Evolver has 2 independent analog lowpass filters (one for the left channel, one for the right) and a stereo digital highpass filter.

**Envelopes** – Envelopes shape all sorts of parameters over time. Evolver has 3 envelopes, all of the same basic ADSR design. One envelope is dedicated to patch volume, one envelope is dedicated to the lowpass filter, and the remaining one is a modulation source.

**Modulation** – Modulation gives synthesizer sounds character, motion, and interest. Evolver provides many modulation sources, including 4 LFOs, the above-mentioned envelope, and the sequencer.

In addition, Evolver provides 4 flexible modulation slots, which allow almost any parameter to behave as a modulation source and be routed to almost any other parameter as a destination.

**Effects** – Effects are another component in determining a synthesizer's overall sound. Evolver's effects include a few flavors of distortion and a digital delay.

**Sequencer** – Sequencers provide a means for generating a set of values over time. Evolver's sequencer is a “classic” 4-track 16-step design.

**External Inputs** – Evolver also provides audio inputs, allowing it to act as an effects unit for incoming audio.

Don't worry if you don't know what some of the above terms mean. I hope to explain them all as we go along, and the included “exercises” should help you gain a deep understanding of what's really going on.

## The Basic Patch

---

For many of the explanations and exercises, I will refer to “the basic patch”. This is a sort of minimal, vanilla sound that will act as a kind of common starting point. It will allow us to isolate specific features and clearly hear what happens when parameters are varied or changed.

Our first task is to program this in. If you have the included ZIP file, you may be able to load the patch through a MIDI SysEx program or other editor, such as SoundTower’s Evolver Editor.

Fire up your Evolver. Find a memory location you can dedicate to this basic patch. I like to use #128 in bank A, because it’s easy to get to.

Put in the following values. They’re more or less in order from the Evolver front panel, if you’re reading from left to right, top to bottom:

When you’re done, play a few notes. Beep! Bleah. It isn’t very interesting, but you will find it very useful.

Once it’s done, save it. Then save it again, in a different memory location. Setting this up is a bit tedious, and you only want to have to do it once.

Dave Smith wrote the first version of the MIDI specification and was instrumental in its becoming adopted by synthesizer manufacturers.

## The Definitive Guide to Evolver

| Family                | Parameter    | Value          | Family           | Parameter         | Value |
|-----------------------|--------------|----------------|------------------|-------------------|-------|
| Evolver-wide          | Main Volume  | 75             | VCA              | VCA Level         | 0     |
|                       | Use Program  | On             |                  | Env Amt           | 100   |
|                       | Tempo        |                |                  | Attack            | 0     |
|                       | Lock Seq     | OFF            |                  | Decay             | 0     |
| Overall               | Poly Chain   | OFF            |                  | Sustain           | 100   |
|                       | Transpose    | -24            |                  | Release           | 0     |
|                       | BPM          | 120            |                  | Output Pan        | St1   |
| Oscillator 1, 2, 3, 4 | Clock Div    | 8n             |                  | Volume            | 100   |
|                       | Frequency    | C0             | Velocity         | 0                 |       |
|                       | Fine         | 0              | Exp/Lin          | Epn               |       |
|                       | Level        | 50             | Env              |                   |       |
| Oscillator 1 & 2      | Glide        | 0              | Modulation       | Mod X Source      | OFF   |
|                       | Shape        | Saw            | Slots (all)      | Mod X Amt         | 0     |
|                       | Sync 2→1     | OFF            | Mod X            | OFF               |       |
| Oscillator 3 & 4      | Osc Slop     | 0 (OFF)        | Destinaton       |                   |       |
|                       | Shape        | 1              | Delay (1, 2, 3)  | Delay Time X      | 0     |
|                       | FM 4→3       | 0              | Delay Level X    | 0                 |       |
| Noise                 | Ring Mod 4→3 | 0              | Delay Feedback 1 | 0                 |       |
|                       | Ring Mod 3→4 | 0              | Delay Feedback 2 | 0                 |       |
|                       | Ring Mod 3→4 | 0              | Output Hack      | 0                 |       |
|                       | Shape Seq    | OFF            | Distortion       | 0 0               |       |
|                       | Noise Vol    | 0              | LFO (all)        | Frequency         | 0     |
| Filter                | Shape Seq    | OFF            | Frequency        | 0                 |       |
|                       | Shape        | 1              | Shape            | Tri               |       |
|                       | FM 4→3       | 0              | Amount           | 0                 |       |
|                       | Ring Mod 4→3 | 0              | Destination      | OFF               |       |
|                       | Ring Mod 3→4 | 0              | ENV 3            | Amount            | 0     |
|                       | Shape Seq    | OFF            | Destination      | OFF               |       |
|                       | Noise Vol    | 0              | Attack           | 0                 |       |
|                       | Frequency    | 164            | Decay            | 0                 |       |
|                       | Env Amt      | 0              | Sustain          | 0                 |       |
|                       | Attack       | 0              | Release          | 0                 |       |
|                       | Decay        | 0              | Delay            | 0                 |       |
|                       | Sustain      | 100            | Velocity         | 0                 |       |
|                       | Release      | 0              | Sequencer        | (All step values) | 0     |
| Resonance             | 0            | Seq [All] Dest | OFF              |                   |       |
| Keyboard Amt          | 0            | MIDI           |                  |                   |       |
| 2/4 pole              | 2 P          | Controllers    |                  |                   |       |
| Velocity              | 0            | Velocity       | 0                |                   |       |
| Audio Mod             | 0            | Destination    | OFF              |                   |       |
| Split                 | 0            | Mod Wheel Amt  | 0                |                   |       |
| Highpass              | 0 0          | Destination    | OFF              |                   |       |
| (Hi-Pass Filter)      |              | Pressure Amt   | 0                |                   |       |
| External Input        |              | Destination    | OFF              |                   |       |
| Ext In Volume         | 0            | Breath Amt     | 0                |                   |       |
| Ext In Mode           | St           | Destination    | OFF              |                   |       |
| Input Hack Amt        | 0            | Foot Con Amt   | 0                |                   |       |
| Input Peak Amt        | 0            | Destination    | OFF              |                   |       |
| Input Peak            | OFF          |                |                  |                   |       |
| Destination           |              |                |                  |                   |       |
| Env Follow Amt        | 0            |                |                  |                   |       |
| Env Follow            | OFF          |                |                  |                   |       |
| Destination           |              |                |                  |                   |       |



## The Oscillators

---

Oscillators are the basic building blocks of a synthesizer's sound. The sound the oscillators produce is modified by the synthesizer's other components: filters, envelopes, modulators, effects, and so on – all of which we will learn about. A synthesizer's oscillators create both the pitches/frequencies and the basic timbres.

Evolver offers four oscillators. Four oscillator architectures are extremely powerful and flexible. They're also becoming more popular.

Evolver is also special in that it has both analog and digital oscillators.

### Analog Oscillators

---

Back in the early days of synthesizer history (the 1960's), analog oscillators were the only technology available. Tonewheels, such as those used in the Hammond B3 and the Telharmonium, were too big, too mechanical, and too limited.

While the sound of analog oscillators was warm and fat, early analog oscillators were also notoriously unstable. They would respond to temperature changes by going out of tune, and not in any predictable fashion.

While this was tedious when trying to record (re-tuning after every take, or stopping a take in the middle if things got too bad), it was particularly unbearable for live performance. Many performers carried screwdrivers and left the tops or backs off of their synths so they could get at the tuning controls during the show.

The other problem was that you needed a lot of analog oscillators if you wanted to make a polyphonic synthesizer. If you had a two-oscillator per voice design, and wanted a reasonable (for the day) 5-note polyphonic synthesizer, you had to include 10 oscillators! The extra circuitry not only made polyphonic synthesizers expensive, it also generated a

#### Why Is It Called An "Oscillator?"

Because the signal that comes out is a wave that "oscillates" – it swings from low to high and back again, over and over.

#### The First Synthesizer

Believe it or not, the world's first synthesizer appeared in 1906!

It was called the Telharmonium, and was developed by an American named Thaddeus Cahill.

The Telharmonium was an additive synthesizer that used giant rotating tonewheels (an idea later employed by the Hammond organ). It weighed around 200 tons and required a 60-foot frame built from 18-inch steel girders set onto a brick foundations.

In his patent documentation, Cahill repeatedly called the Telharmonium's sound generation "synthesis".

The Telharmonium was also tied to a telephone network for distributing electronic classical music to subscribers.

For more information, read Reynold Weidenaar's book "Magic Music from the Telharmonium"

lot of extra heat, which exacerbated the tuning problems.

Fortunately for us, Evolver features modern analog oscillators. They deliver a warm, fat, analog sound, but are rock-solid and don't go out of tune (unless you want them to!)

Evolver's analog oscillators offer the following control parameters:

- Frequency
- Fine tuning ("Fine")
- Shape/PW (PW means "Pulse Width")
- Level
- Glide
- Oscillator slop ("Osc slop")
- Oscillator hard sync ("Sync 2→1")

### Frequency

Each of Evolver's oscillators can be tuned over a huge range of notes. The "Frequency" parameter determines what pitch the oscillator will play when middle C (C3, or MIDI note number 60) is played. Technically, since Evolver's values are notes, the parameter should be called "pitch" – if it were truly frequency, there'd be a number indicating Hertz (Hz). But I digress.

Each of Evolver's oscillators has its own separate frequency parameter. This enables a couple of interesting effects. For most basic sounds, you will want to make sure all of the active oscillators are set to the same frequency.

You can set a single oscillator down an octave or two and use it as a "sub-oscillator", which is common in some classic synthesizer architectures. Expanding on that idea, you can tune each oscillator to separate octaves (C0, C2, C3, C4) for big stacked sounds.

Taking this to the next step, you can set each oscillator to a different frequency and play a "one-finger chord". Granted, the chord position is fixed (i.e. it's always going to be a Major 7<sup>th</sup> chord with the root being the key you play), so it won't stay "in key" on all but a

#### **How Many Oscillators?**

Some of the world's best-known synthesizers only had a single oscillator per voice (and sometimes only a single voice!). Roland's classic TB-303 "Bassline" is an example.

Perhaps most popular were two-oscillator synthesizers. Giants of the two-oscillator world include the Roland Jupiter-6, the Oberheim Matrix-12, and the Dave Smith-designed Sequential Circuits Prophet-5.

But why have two when you can have three? Some other popular synthesizers had three oscillators, sometimes using one as a "sub-oscillator". Three oscillator synthesizers of note include the legendary Minimoog, and even the PPG Wave.

Native Instruments includes "Carbon", a 4-oscillator subtractive synth in Reaktor 4.0, and "Photone", another 4-oscillator synth in "Electronic Instruments, Volume 2".

handful of notes. This is not necessarily a bad thing – Debussy wrote many pieces where he moved a “fixed chord shape” around, regardless of key signature. We will revisit these “one-finger” chords later.

Note that for MIDI keyboard control, the frequency parameter interacts with the Key Transpose (“Key Off/Xpose”) parameter. If you’re adjusting frequency and it doesn’t seem to be working right, check and make sure Key Transpose isn’t set to something weird.

### **Fine**

---

For all sorts of reasons, you may want to tune Evolver’s oscillators in degrees smaller than individual notes. The “Fine” parameter allows each oscillator to be tuned from -50 cents to +50 cents in single-cent steps. When the “Fine” parameter is set to 0, the note frequency is unchanged.

Small values (+/- 2-3 cents) are useful for fattening up sounds or creating a kind of chorusing effect known as “detuning” (because the oscillators are slightly out-of-tune relative to each other).

Larger values can make some nice obnoxious noises. Some very popular rave sounds were simple sawtooth oscillators with more detuning than was previously considered “tasteful”.

Using an LFO to modulate the value of the “Fine” parameter in the range of a few cents can also help produce a very rich chorus effect when coupled with delay modulation.

The “Fine” parameter will also come in handy when we start exploring the digital oscillators and Frequency Modulation.

### **Shape/PW**

---

Evolver’s analog oscillators can produce many of the traditional analog waveforms: sawtooth, triangle, sawtooth-triangle mix, and a full range of pulse waves.

There are 12 notes or semi-tones in an octave.

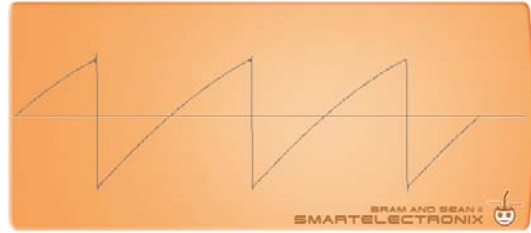
There are 100 cents (centavos) in a semi-tone.

The relationship between harmonics and waveshapes is too complex to address here.

All you really need to know is that different waveshapes produce different sets and ratios of harmonics, which means a different timbre or basic sound.

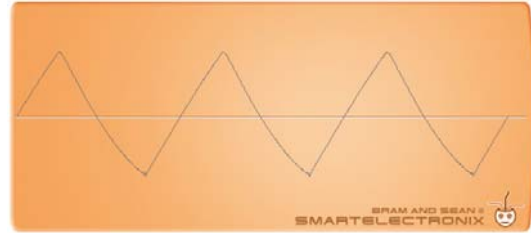
## Sawtooth

The sawtooth wave is the “generic synthesizer” sound. Sawtooth waves contain all harmonics, and are bright and buzzy in their native state. By producing all harmonics, they produce a wide range of frequencies, they respond very well to being filtered. This is one reason they have proven so popular over the years with synthesizer makers and programmers.



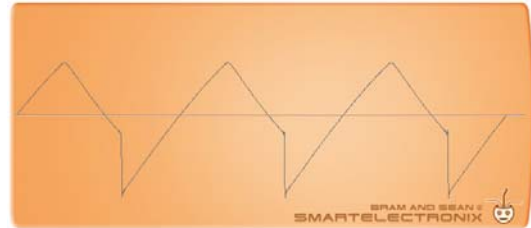
## Triangle

Triangle waves have very few harmonics, and thus produce muted, flute-like sounds. Triangle waves are very similar to sine waves, but have a bit more character and harmonic content.



## Mix

This waveform is a mix of sawtooth and triangle, and produces a sound somewhere between the two.



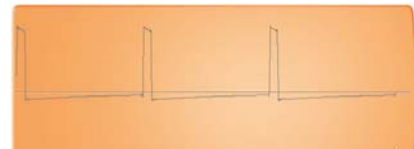
## Pulse Waves

One of Evolver's analog oscillator shapes is a "pulse wave". A pulse wave goes immediately and abruptly from zero to full, stays there, then abruptly drops back to zero.

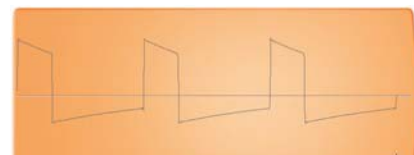
Evolver offers a range of pulse waves, with increasingly longer "pulse" periods. At P-01, the pulse period is so short it's basically nothing, and at P-99, the pulse period is so long, the "off" period is basically nothing. This is why at the extremes you don't hear any sound...it's like the oscillator isn't making a wave at all.

At P-50, the “on” period and “off” period are of equal duration. If this sounds like a square wave, that's because it is. A square wave is a pulse wave where the "on" and "off" portions are of equal size/duration. In other words, all square waves are pulse waves, but not all pulse waves are square waves.

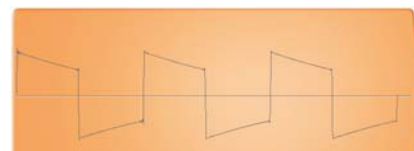
Square waves have a particular bright and hollow



**Pulse Wave - P10**



**Pulse Wave – P30**



**Pulse Wave – P50**



**Pulse Wave – P90**

sound (they only contain odd harmonics). They can be used to create a wide range of sounds.

### Exercise 1: Exploring Analog Waves

Bring up our basic patch. Set the OSC levels to 0 for everything except OSC 1. Set OSC 1's Shape to P-50. That's basically a square wave. Now set it to P-25. Sounds a bit different, right? Now try P-75. Hear a difference between that and P-25? Think about why.

Compare those Pulse waves with the other waveshapes available for OSC 1 – Triangle, Saw, and Mix. Can you tell them apart easily? Compare how they sound at very low and very high pitches.

Step through all of the different pulse width ranges.

### Exercise 2: PWM Strings

The Evolver, like many classic analog synths, allows you to modulate (i.e. change/control/affect) the width of your pulse waves (also known as “pulse width”) from a variety of sources. Pulse Width Modulation (PWM for short) is a classic analog synthesizer effect.

The most common method of modulation on many synthesizers is via a simple LFO.

Since manually changing pulse width results in a change in timbre, one would expect an LFO-modulated pulse width to produce a regularly changing/varying timbre, right?

In fact, a "classic" analog string sound comes from modulating the pulse width.

Start with our basic patch.

- Set OSC 3 and 4 Level to 0
- OSC 1 Shape to P-52
- OSC 2 Shape to P-48
- LFO 1 Shape to “tri” (triangle wave)
- LFO 1 Frequency = 42
- LFO 1 Amount = 30
- LFO 1 Destination = O1P (OSC 1 Pulse Width)
- LFO 2 Shape = “tri” (triangle wave)

The dictionary defines modulation as:  
“The variation of a property of an electromagnetic wave or signal, such as its amplitude (volume), frequency (pitch), or phase.”

LFO stands for “Low Frequency Oscillator”.

By “low frequency”, we mean oscillation of fewer than 20 cycles per second (or 20 Hertz, abbreviated Hz)..

20 Hz, or cycles per second, is the lowest frequency humans can perceive as pitch. If the LFO was faster than 20 Hz, it'd just be a regular oscillator.

20 Hz is still pretty fast, though, and most LFOs can be set to run far below 1 Hz.

## The Definitive Guide to Evolver

- LFO 2 Frequency = 32
- LFO 2 Amount = 23
- LFO 2 Destination = O2P (OSC 2 Pulse Width)
- VCA Attack = 61
- VCA Release = 34

Leave the filters off/wide open - they should be already set that way from the basic patch.

Check it out! Add some echo and chorus, and boom, you're Gary Numan!

Next, if your keyboard or controller has a Modulation Wheel, try assigning it to control "**OSC ALL Pulse Width**". Set it to a value of +52. Hold down a key and start moving the Mod wheel up. What is happening?

Experiment with increasing **LFO Amount** dramatically for one, then both oscillators. LFO amount controls the “depth” of the LFO’s oscillation or swing – how high and how low it goes. Find some values that you think sound good.

Then try increasing **LFO Frequency** the same way. The LFO frequency is the speed at which the LFO changes or modulates the signal. Higher frequencies mean faster oscillations up and down.

What would slower sound like? Try decreasing the LFO frequency or “speed” and find out.

Once you have values you like for LFO frequency and LFO amount, set

- Mod Wheel Destination = “FiL” (Filter Freq)
- Mod Wheel Amount = -78

This will let you use your Mod Wheel to “roll off” the high frequencies of the patch. You may also want to play with the “Output Pan” setting, too – see if you like your sound better “wide stereo” or “mono”.

Finally, **save the patch** to an empty memory location! Evolver has plenty of memory locations, so use them, especially if you come across a sound you

From here on out, rather than call out LCD read-outs (“FiL”) and whatever Soundtower or other editor software might call a parameter, I will simply refer to what the parameter should be, such as “LPF Frequency”

like. If you're using editor software, you might want to give it a name like "PWM Strings".

The key to making the "string sound" rich is to make the variation of the pulse widths irregular, but not too extreme.

Add a third LFO, set to:

- Shape = Triangle
- Frequency = 5 (very slow)
- Amount = 14 (shallow depth)
- Destination = OSC ALL Pulse Width (OAP)

I set the LFO to use a triangle wave because it is the smoothest - the others all provide abrupt changes which make the sound less "swirly", and will introduce some abrupt stops.

Then again, that may be an effect you like! Try substituting very slow, synced sawtooth LFOs for your triangle LFOs. By adjusting the frequencies (speeds) and the amounts (depths) of the LFOs, you can get some nice subtle rhythmic throbbing in your basic PWM string sound.

### **Exercise 3: PWM Horn**

Go back to the basic patch (which we will continue to use as a common starting point)

- OSC 1 Shape = P-01
- OSC 2 Shape = P-03
- ENV 3 Attack = 26
- ENV 3 Decay = 78
- ENV 3 Sustain = 7
- ENV 3 Release = 83
- ENV 3 Amount = 61
- ENV 3 Destination = OSC ALL Pulse Width
- Output Pan = St1 or RS3 (use either extreme, this patch sounds much better in stereo)

So now instead of using LFOs to modulate the sound, we are simply using the Modulation Envelope (ENV 3) to modulate pulse width in a different fashion.

It produces a very different sound – a classic "synth horn". Try some different values for Envelope 3, and

once you've found something you like, go ahead and save it.

Note that we have now created two fairly different patches using almost exactly the same settings! The main difference was the modulation.

### Level

---

Level is simply the loudness (amplitude or volume) of a given oscillator. The level parameter ranges from 0 (off) to 100 (full volume). This parameter is handy for balancing the relative volumes of different oscillators.

Note that if you turn up all of Evolver's oscillators to maximum and play a note, the clip light will come on and Evolver will distort. I find a good rule of thumb is to make sure that the sum of all oscillator levels in Evolver should be less than 125 to prevent this from happening. Ultimately the sound you want will determine your oscillator levels, including whether or not they distort.

### Glide

---

Normally, when you switch from one note to another, the pitches change immediately, as soon as the new note is pressed. The Glide parameter changes this behavior. Rather than instantaneous pitch changes, Glide causes the pitches to change over time.

Low glide values mean the pitches change fast, and high values mean they take a long time.

Glide also has two modes, **normal** and **fingered**. In normal mode, glides occur whenever new notes are played. "Fingered" mode only glides when you keep the previous key held down and then press a new key.

Let's try it. Bring up the Basic Patch.

- OSC 3 and 4 Level = 0
- OSC 1 Glide = F96
- OSC 2 Glide = F91

Press a key, let it go. Press another. Nothing is different, right?

#### Anti Distortion

##### Formula:

$$\begin{aligned} & \text{OSC 1 Level} + \\ & \text{OSC 2 Level} + \\ & \text{OSC 3 Level} + \\ & \text{OSC 4 Level} = \\ & < 125 \end{aligned}$$

#### Other Names:

On some synths...

Glide is also known as "portamento"

"Fingered mode" is also called "legato"



## The Definitive Guide to Evolver

Now press a key. Hold it down. Press the key an octave higher. You should hear the pitch “zoom” up to the new note. Keep holding the first key down and play several random keys in succession. The pitch will slur up and around until you finally let go of the other keys, when it will dive back down to the original pitch.

Glide also includes a setting for “OFF”, which actually prevents oscillators from tracking any MIDI pitches at all – regardless of what key is played, the oscillators will only play the frequency to which they are set. This could be useful if you want to keep some oscillators at a fixed pitch while the others change – if you’re using them to modulate other parameters, for example, or if you want them to provide drones.

Using our above example, try setting

- OSC 2 Glide = OFF

Play a few notes.

Finally, Glide only affects notes triggered by MIDI – a keyboard or external sequencer. Glide cannot affect notes triggered from Evolver’s internal sequencer.

## **Special Analog Oscillator Features**

---

### **Oscillator Slop**

Remember how old analog oscillators used to drift out of tune all the time? Well, small variations in tuning can help give synthesizer sounds an organic, interesting character. In their native state, Evolver’s oscillators stay in tune.

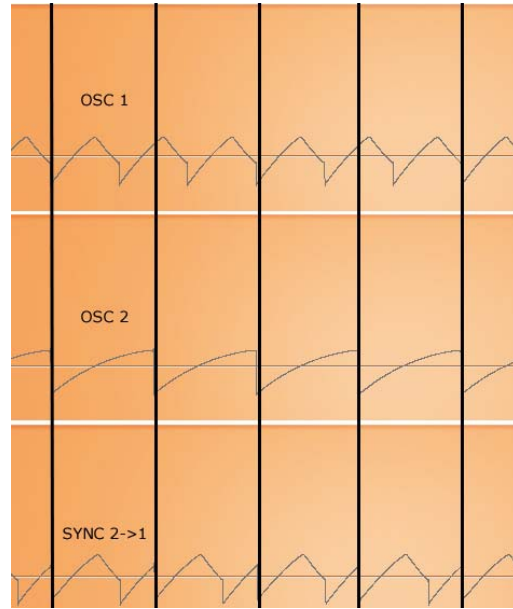
But Dave Smith added the ability to simulate some of this analog character via the “Oscillator Slop” (Osc slop) parameter. It ranges from 0 (off) to 5 (maximum slop). This parameter allows the oscillators (and may even encourage them!) to drift out of tune a bit from time to time.

It is useful if you want to add some unpredictability or a vintage quality to your sounds. It is subtle, but tasty. Personally I leave it set to 5 on any sound that uses the analog oscillators.

## Hard Sync

The basic analog waveshapes can seem somewhat limiting. One way synthesizer designers attempted to create more variation from this limited palette was through synchronizing the oscillators, so that when one of them started its waveform at zero, the other would be forced to “restart”, regardless of wherever it was in its waveform. This behavior is called “hard sync”.

Evolver includes the ability to sync oscillator 1 to oscillator 2, so that whenever oscillator 2’s wave re-starts, oscillator 1’s wave will re-start as well. That’s why Evolver’s label says “Sync 2→1” – it means “Sync, where 2 controls 1”



Hard Sync: OSC2→OSC1

Now, right away, one thing should be obvious: If oscillator 2 and oscillator 1 are at exactly the same pitch/frequency, turning this parameter on has no effect. Why? Because when the two oscillators are running at the same pitch, their waveforms are already starting at the same place every time!

### Exercise 1: Exploring Hard Sync

Recall our basic patch.

Set OSC 3 and 4 Level to 0, because we only want to hear the analog oscillators.

Set OSC 2 Level to 0. For now, we are only interested in hearing oscillator 2’s effect on oscillator 1, not oscillator 2’s actual sound.

Turn “Sync 2→1” on. Play a few notes. Turn “Sync” off. Play a few more notes. No difference, right? As we noted above, since both oscillators are set to “C1”, they are already re-setting at the same time.

Turn “Sync” back on. Now let’s see what happens when we change the pitch of the two oscillators relative to each other. Since we are using oscillator 2 to “control” oscillator 1, let’s try that first. Change

oscillator 2's frequency from C0 to C1, by gradually incrementing the value.

You should hear the pitch of oscillator 1 change. In fact, it steps up and “tracks” the pitch value you are setting with oscillator 2! Why? Well, pitch is determined by a wave's periodicity (how fast it repeats). Since oscillator 2 is controlling oscillator 1's periodicity (every time oscillator 2 re-starts, it makes oscillator 1 re-start), it is, in effect, controlling oscillator 1's pitch.

The notion of periodicity will come in handy again later when we get to the digital delay.

This works in both directions – go ahead and try decreasing the pitch of oscillator 2 down below oscillator 1's frequency of C1.

Not only that, but changing the pitch of oscillator 1 seems to have no effect! Set oscillator 2 back to C1. Hold down a note and adjust the frequency of oscillator 1 in half steps up to E1, then down past C1 to A0. The pitch doesn't change – oscillator 2 is in control.

While changing oscillator 1's frequency by changing oscillator 2's frequency is a neat party trick, it's not exactly interesting. So let's move on.

### **Exercise 2: Hard Sync for New Timbres**

We just learned that when hard sync is enabled, oscillator 2 determines the pitch for oscillator 1, and further, changing oscillator 1's pitch doesn't seem to do much at all. But that's not quite true.

Recall our basic patch.

- Turn “Sync” ON
- Set OSC 2, 3, and 4 Level = 0 (just like we did previously)

Now start decreasing OSC 1 Frequency.

Keep turning it down, as low as it can go. What happens?

The sound gets quieter the lower you go, until finally no sound is heard. Why? Well, oscillator 2 is in charge. Every time it resets, it resets oscillator 1. As you decrease oscillator 1's pitch, it takes longer and

## The Definitive Guide to Evolver

longer for oscillator 1's waveform to rise. Since oscillator 2 keeps resetting oscillator 1's wave, the lower oscillator 1's pitch goes, the less of a rise oscillator 1 can make (and thus less volume) before oscillator 2 forces it back to 0.

Clearly lowering the pitch of oscillator 1 doesn't do much. Let's try raising it quite a bit. Bring oscillator 1 back to C0. Now start increasing it. Raise it all the way up to C4, going slowly. Hmm. This is a bit more interesting, isn't it?

As you pass C1 and go higher and higher, the timbre begins to change fairly dramatically, and varies quite a bit with each change in oscillator 1's pitch.

Set oscillator 1 back to C0. Change oscillator 1's shape to "Tri" (triangle) and go back up again. Hear how the smooth triangle wave gets rough and buzzy in spots? Try it with a few different flavors of pulse waves. Clearly, oscillator 1's waveshape makes a big difference in the resulting sound.

Try it again, only this time, Change oscillator 2's shape to "P-50". Does it sound different? Nope. It doesn't. In fact, oscillator 2's shape or timbre is irrelevant. All that matters is the frequency/pitch, because that's what causes oscillator 1 to re-start.

### **Exercise 3: A Classic Prophet-5/Dave Smith sound**

So now we know the "rules" for getting interesting "hard sync" sounds:

- "Sync 2→1" must be ON
- Oscillator 2 controls the pitch
- When oscillator 1's pitch is higher than oscillator 2's pitch, interesting things happen
- Oscillator 2's waveshape doesn't matter, but oscillator 1's waveshape does

Let's put all this together. Since distinct changes in oscillator 1's pitch produce distinct changes in the timbre, it follows that smooth, extreme changes in oscillator 1's pitch should produce some smooth, extreme changes in the timbre, right?

## The Definitive Guide to Evolver

Recall our basic patch.

- Turn “Sync 2→1” on. We have to do this because that’s the effect we’re exploring.
- Set OSC 2, 3, and 4 Level = 0. We only want to hear OSC 1.

We know that oscillator 1’s pitch needs to be higher than oscillator 2’s for interesting things to happen. So set OSC 1 Frequency = C2, and make sure OSC 2’s is at C1.

Set oscillator 1’s waveshape to “P-50” – it will sound better than “SAW” for this (in my opinion).

Now what we need is a way to make oscillator 1’s pitch change in some interesting way. How about Envelope 3 (a.k.a. “The Mod Envelope”)?

Set ENV3 as follows:

- Attack = 15 (a short attack gives a nice bite)
- Decay = 87 (longer decays emphasize the effect)
- Sustain = 0
- Release = 0
- Amount = 67
- Delay = 0
- Velocity = 0
- Destination = O1F (OSC 1 Frequency)

The “destination” parameter is the crucial element that makes the whole thing work.

Turn Sync “OFF” for a second, and you can hear what Envelope 3 is doing – it is making the pitch of oscillator 1 shoot up fast and come down slow, sort of like an electronic drum.

But when you turn hard sync “ON”, the sound comes alive and becomes a classic, aggressive synth tone.

Play with the values for ENV 3. What happens when you give it a longer attack? What happens when you drastically change amount?

Also try different wave shapes for OSC 1.

This was such a distinctive and popular sound made by the Prophet-5 that Native Instruments uses it as the default patch when you load up their popular emulation, the Pro-53.

Congratulations! You have now mastered hard sync! Go ahead and save this sound somewhere.

You should now have a basic understanding of all of the parameters for the analog oscillators, and a few decent patches to boot!

Look at your Evolver or editor and review the parameters for OSC 1 and 2. For each parameter, you should be able to remember and understand what changing it will do. If not, bring up the basic patch and twiddle away.

Once you're confident you know what all those parameters do, you are ready for the digital oscillators.

## Digital Oscillators

---

Evolver has two digital oscillators, referred to as OSC 3 and OSC 4. OSC 3 is hard-wired to the left channel, and OSC 4 is hard-wired to the right. Evolver's digital oscillators offer the following control parameters:

- Frequency
- Fine Tuning ("Fine")
- Shape
- Level
- Glide
- Frequency Modulation
- Ring Modulation
- Shape Sequence

**Frequency**, **Fine**, **Level**, and **Glide** parameters work the same for digital oscillators as they do for analog oscillators, so I won't repeat them here. Refer to the analog oscillator section if you need more information.

## Shape

---

Evolver's digital oscillators allow the user to select from a variety of different waveshapes. The "Shape" parameter determines which waveshape is used for each oscillator. This is similar to the "Shape/PW" parameter for the analog oscillators, only you can select from a large range of digital waveshapes.

Synthesizer sound "programs" or configurations are frequently called "patches" because the first large modular synthesizers used a series of cables to "patch" one component's output to another's input.

These complex patch cable configurations became known as "patches" and the term stuck.

## What Are Waveshapes?

There's nothing really mysterious about waveshapes. You can think of waveshapes as being very short samples – in fact, they are only a single cycle in length.

Waveshapes can produce almost any type of wave that can be drawn or created. So instead of a basic sawtooth or triangle (like the analog oscillators produce), you get a “squiggle”. The different digital waveshapes provide different (and usually much more harmonically rich or interesting) timbres than traditional analog oscillator shapes.

Of course, you can also use digital waveshapes to do replicas of the analog waveshapes as well. Evolver includes digital waveshapes for sawtooth and square waves (which its analog oscillators can produce), as well as sine waves (which the Evolver's analog oscillators can't produce).

## Where Do Waveshapes Come From?

Back in the early days of synthesis, people were frustrated by the perceived limitations of analog oscillators. Aside from the stability and cost drawbacks of analog oscillators mentioned earlier, you could only easily make a handful of very simple oscillator waves: sine, square, triangle, sawtooth, and the like.

Even with tricks like pulse-width modulation and hard sync, people felt these sounds were too limited and too limiting. The basic analog shapes just weren't versatile enough for advanced synthesis. And while they could do serviceable emulations of certain instruments, analog oscillators did not sound “real” enough.

Eventually, someone hit on the idea of using very short digital recordings of real instruments. After all, if you could zoom in on a sustained trumpet note and see what the “wave shape” was, couldn't you just make an oscillator that made that shape and get a trumpet noise?

Well, no. It's not that simple at all, because most real instruments have waveshapes that change over time

A number of successful synthesizers utilized waveshapes, including the Korg DW-8000 and the Prophet-VS.

The path that digital wave shape synthesis pioneered also led to “wave table” instruments like the PPG Wave, “ROMplers” like the popular Alesis QS-8, and all digital samplers.

## The Definitive Guide to Evolver

as they play, and that vary considerably depending on the register or octave in which the instrument is playing, and that are also affected by how loud the instrument is playing.

The notion that truly interesting sounds are continuously changing is very important. We have already heard what simple pulse-width modulation and analog slop or “drift” can do to make sounds more interesting. We will revisit this idea several times. But for now, let’s get back to waveshapes.

Dropping in a 1-cycle digital recording simply replaces the boring, static, artificial-sounding basic wave shape generated by an analog oscillator with a boring, static, artificial-sounding complex wave shape played by a digital oscillator.

However, using wave shapes did allow for some interesting new sounds and broadened the sonic palette considerably from what the old analog standbys provided.

### Default and User Waveshapes

Evolver has 2 different kinds – **default** waveshapes and **user** waveshapes. The default waveshapes are stored in ROM, are 12-bit resolution, and 128 samples in length. The default waveshapes are in fact identical to those found in the Prophet-VS synthesizer. Appendix C includes a widely-circulated table with some names of the different default waveshapes and some of my notes about them.

Ironically, several of the Prophet-VS’ waveshapes (and thus Evolver’s) were themselves created by copying the front-panel drawings on the Korg DW-8000 by hand!

Evolver also lets you load your own waveshapes to RAM which are 16-bit resolution and 128 samples in length. In order to load in your own, you will need some sort of computer software and MIDI interface. As of this writing, there are two methods that I have used. One is the Soundtower Evolver editing software. The other is a freeware application called “Evolver Wave Dump” by Darren Richards.

### Exploring the Digital Difference

---

Evolver’s digital oscillators sound very different than the analog oscillators for several reasons.



## The Definitive Guide to Evolver

One reason is the waveshape itself. In addition to being able to emulate many of the analog oscillator waveshapes, the digital oscillators offer dozens more different shapes.

Waveshapes determine the basic harmonic structure of a sound. It is true that this harmonic structure is further shaped by envelopes, filters, and modulation, but what you start with (i.e. the waveshape) has a major effect on the outcome.

Bring up our basic patch. Make the following changes:

- OSC1, 2, and 4 level = 0
- OSC3 Shape = 4

Play a few notes and start changing the values for OSC3 shape.

Play the notes higher and lower.

Try a variety of different shapes in different note ranges.

You should hear sounds that are clearly very different than what the analog oscillators produce. The shape of the oscillator's wave makes a big difference in the basic sound.

Another reason the digital oscillators sound different is that the digital oscillators are digital. They produce their sound in a completely different fashion. And because they aren't analog oscillators, there is no "drift" parameter which can be applied, nor can the oscillators be hard synced the way the analog oscillators can.

Additionally, due to the peculiarities of the way Evolver's digital oscillators work, as you play higher and higher pitches with the digital oscillators, you will hear a particular kind of digital "grit" or distortion appear called "aliasing".

Let's take a listen. Bring up our "basic patch"

- OSC 1, 2, and 4 Level = 0
- OSC 3 = shape 3 (Square Wave)
- Verify key transpose = 0

- pitch bend range = 7

Start playing at middle C. Grab the pitch bender on your keyboard controller and slowly bend the note up. As you move the pitch bender, you'll hear a kind of "swirly" digital noise. That's aliasing.

Explaining why aliasing occurs (and why it's called "aliasing") is beyond the scope of this guide. For now, just accept aliasing as part of the sonic character of Evolver's digital oscillators.

Evolver's digital oscillators also have a few special effects available which the analog oscillators do not: **Frequency Modulation (FM)** and **Ring Modulation**.

### Frequency Modulation

---

Evolver's digital oscillators provide the most basic implementation of FM: One oscillator (the carrier) has its frequency varied (modulated) by the other oscillator (the modulator).

In other words, you apply one signal (the modulator, which you can't hear) to another (the carrier, which you can). The result? The modulator makes the carrier sound different.

Conceptually, this is somewhat similar to both hard sync and pulse width modulation. In fact, it is really no different from using an LFO as a modulation source, except instead of modulating a signal for effect, you are modulating an oscillator to produce a new timbre.

Before your head starts spinning around about this, let's do a quick exercise which should illustrate the concept quite clearly.

#### Exercise 1: A Simple FM Sound

Bring up the basic patch.

- OSC 1, 2, and 4 Level = 0
- OSC 3 shape= 01 (sine wave)
- OSC3 level = 50

Frequency modulation synthesis was invented by John Chowning in the late 1960s.

FM is a very simple, clever idea that can produce an incredible range of timbres.

Yamaha's DX-7, one of the best-selling synthesizers of all time, utilized the principles of FM synthesis for its sound generation.

## The Definitive Guide to Evolver

Play a few notes so you can hear what the sine wave sounds like. Got it?

- Verify that OSC 4 is set to the same frequency as oscillator 3 (C1) and that OSC 4 shape = 01 (sine wave)
- Set “FM 4→3” to 12 (Like hard sync, the label means “FM, where oscillator 4 is affecting oscillator 3”)

Now play a few notes. That is not a sine wave anymore, is it?

Experiment with incrementally increasing the **FM 4→3** parameter. You will notice that very small changes in value can produce very big effects on the timbre.

### Exercise 2: New Waves

We are currently modulating a sine wave with a sine wave. This most basic form of FM is exactly how Yamaha’s “DX” line of synths works, only those synthesizers provide varying numbers of modulator sine waves and carrier sine waves that you can organize in different ways.

Unlike the DX-7 and other FM synths which only use sine waves as carriers and modulators, Evolver lets you use any of the digital waveshapes for carriers and modulators.

This feature allows you to get much more out of the simple “1 modulator, 1 carrier” FM architecture that Evolver provides.

Take the sound we currently have and change oscillator 4’s waveshape from 01 (sine) to 02 (sawtooth). Now listen to the sound. How about that!

To compare how much of a difference the FM is making, turn FM 4→3 back to 0. That’s a sine wave! Now bring FM 4→3 back to 12.

Experiment with several other waveshapes for OSC 4 (the modulator). Waves 09, 36, and 70 will give you some idea of the range.

## The Definitive Guide to Evolver

Remember that you can select shapes for the carrier (OSC 3 in this case) as well as the modulator (OSC 4):

- Go back to sine waves (01) for both
- Set the “FM 4→3” parameter to 12 (if it’s not already there)
- Set OSC 3 shape = 02 and play a few notes.  
Fuzzy!
- Now set OSC 3 = shape 15

Try a few more, and then experiment with different shapes for both OSC 3 and 4.

### Exercise 3: A Certain Ratio

Now we’ve experimented a bit different waveshapes for carriers and modulators, but always tuned to the same frequencies.

What if we changed the frequencies of the oscillators? That made a big difference when doing hard sync, right?

Yes! Actually, the difference between the frequencies is crucial for good FM sounds. This difference between frequencies is referred to as “ratio”, since you will want both oscillators to track pitch changes by the same amount.

Since both of Evolver’s digital oscillators follow the keyboard’s pitches (unless you tell them not to, by setting “Glide” to “OFF”) at the same rate, maintaining a ratio is easy: just tune the oscillators differently.

## The Definitive Guide to Evolver

This chart gives you settings for most possible whole-number ratios. Pitch/note values (C1, C2, etc.) assume your carrier is set to C1.

Bring up the basic patch.

- OSC 1, 2, and 4 Level = 0
- Verify OSC 3 and 4 shapes = 01
- Verify OSC 3 frequency = C1
- FM 4→3 = 12
- Using the chart to the right, set OSC 4's values to each of the ratios in turn, and play a few notes up and down the keyboard

Use the “Freq” parameter for the basic semi-tones tuning. Adjust the “Fine” parameter for cents, and note that all cents values had to be rounded up or down a little to the nearest cent, since Evolver doesn't do fractional cents.

Keeping the ratios as whole numbers between 1 and 4 means you will be able to hear/perceive the frequency of the modulator. This in turn means the resulting timbre always has a distinctive pitch.

If you use whole-number ratios greater than 4, you probably just hear the modulator as harmonics (unless you are playing very low notes).

If your ratios aren't whole numbers, you're going to get “non-harmonic partials” – stuff that sounds like bells, or noise to some degree. It may not be as “musically useful” as whole-number ratios, but it can be very cool.

The amount of FM you apply makes a big difference in the sound as well. The more FM applied, the heavier the “warping” of the basic waveform. The settings for this exercise are pretty tame but do a good job of showing how changing the ratio alters the harmonic structure.

Pick a couple of ratios and try stepping up the amount of FM a bit while still using sine waves. You can hear

| Modulator is...   | Ratio |
|---|-------|
| The same note (C1)  | 1:1   |
| One octave higher (C2, 12 semi-tones)   | 2:1   |
| An octave and a 5 <sup>th</sup> (G2, 19 semi-tones),<br>+2 cents                  | 3:1   |
| Two octaves higher (C3, 24 semi-tones)  | 4:1   |
| Two octaves higher plus a major 3 <sup>rd</sup> (E3, 28 semi-tones),<br>-14 cents | 5:1   |
| Two octaves and a 5 <sup>th</sup> (G3, 31 semi-tones),<br>+2 cents                | 6:1   |
| Two octaves and a 7 <sup>th</sup> (B3, 34 semi-tones),<br>-31 cents               | 7:1   |
| Three octaves higher (C4, 36 semi-tones)  | 8:1   |
| Three octaves and a 2 <sup>nd</sup> (B5, 38 semi-tones), +4 cents                 | 9:1   |
| Three octaves and a major 3 <sup>rd</sup> , (C#5, 40 semi-tones) -14 cents        | 10:1  |

how even small increases have a huge impact on the sound.

Often your best results will come from holding down a note and spinning through parameter values until you find something suitable.

Remember that the waveshapes you use have a huge effect as well. Try this same exercise using a sawtooth (shape 02) as the carrier and a sine (shape 01) as the modulator. You can hear how the sine wave is modifying the timbre slightly.

Using other waveshapes as carriers and modulators coupled with the FM amount provides a whole world of sounds. And that's without even adding modulation...

### **Exercise 4: Modulating FM!**

Of course, you can apply modulation to FM parameters and get some very cool results. This exercise will incorporate several of the techniques we've studied to produce something fun.

Bring up the basic patch

- Turn OSC 1, 2, and 4 levels to 0
- Set OSC 3's shape to 02 (sawtooth)
- Verify OSC 3's frequency is C1
- Set OSC 4's shape to 07
- OSC 4 frequency = C#-1
- FM 4→3 = 15
- ENV 3 Decay = 46
- ENV 3 Amount = 29
- ENV 3 Destination = FM OSC 4→3

Before you play a note, think about what this patch is set up to do.

We have a simple FM patch between two complex waves (from Exercise 2). We have set an interesting ratio between carrier and modulator (in this case, the modulator frequency is actually lower than the carrier). We learned this in Exercise 3. Finally, ENV 3 is modulating the amount of FM – starting it a little

bit higher than it is by default and dropping it fairly quickly.

Now play some notes. You should be hearing a nice, gritty FM sound somewhat reminiscent of our first hard sync patch!

### **One Small Problem**

Before you go thinking you can make perfect, pristine DX-7 sounds with Evolver, it must be pointed out that the aliasing inherent in Evolver's digital oscillators really "trashes up" your FM sounds when you are using high ratios and/or playing high notes. You may have already noticed that lower notes produce "cleaner" FM sounds.

Another thing is that Evolver's tuning is limited to cents. In the FM world, small tuning increments (smaller than a cent) do make a big difference.

Like it or not, there isn't anything that can be done about this, and it will result in some FM sounds producing odd timbres on some notes and not others. So you may occasionally find that the exact sound you want is just out of reach.

But the Evolver FM sound is unique, and can produce some very cool sounds. Rather than clean and sterile, Evolver's FM sounds have a surprisingly organic quality.

And if you want true DX-7 sounds, used DX-7s are relatively inexpensive these days, as are software versions/emulations.

### **Out of Tune?**

One other thing you may have noticed is that applying FM can make your sound "out of tune". Because of the nature of FM, even small amounts will change the basic intonation of your patch.

You can correct this by either adjusting the main "Transpose" (or Key Off/Tran) parameter (which is fast, but only gives you half-step increments) or by adjusting the Frequency and Fine values of both of the digital oscillators. Be patientt may not be easy to get it just where you want.

### FM Summary

There are 4 basic components to Evolver's FM sounds:

- The waveshape used as a **carrier**
- The waveshape used as a **modulator**
- The **ratio** between the carrier and modulator
- The **amount** of FM applied

### Advanced Tips

Entire careers have been spent studying FM synthesis. There's plenty of additional stuff available to read. Since FM is only one component of Evolver's sound, I will conclude this section with a few quick tips:

#### More off!

You can use the "Fine" parameter to deliberately make the ratios slightly off for a rougher sound.

#### Other Ratios

You also don't have to use fixed ratios – you can assign a modulation source to vary the ratio by (such as an LFO, Envelope 3, or the sequencer) changing the frequency of the modulator. And remember you can disconnect the oscillators from MIDI if you want to keep the modulator a constant frequency.

#### Listen to the Modulator

Note that the "FM" signal replaces the carrier but not the modulator. For all of our exercises, we turned the volume off the modulator (OSC 4) to 0 to keep from hearing it. But you don't have to - if you want to hear the FM sound and the sound of the modulator, leave the volume for your modulator (OSC 4 in our case) something other than 0.

#### Mix and Match

Remember that you can use the analog oscillators at the same time! You can use them to subtly fatten up a primarily digital/FM sound. Conversely, you can use FM to provide some secret digital grit to a plain analog sound.

#### Double the Fun

All of our FM exercises involved using OSC 4 to modulate the OSC 3 carrier. Obviously, you can just



reverse those and use OSC 4 as the carrier and modulate it with OSC 3.

But we didn't even address one very important feature – you can do both at the same time. Starting with a basic sine-on-sine FM sound, increasing the amount of FM “the other way” will immediately make the modulator a more complex wave, which in turn makes the carrier more complex, which in turn makes the modulator more complex.

You should not be surprised to learn that small changes have big results when doing this. Don't forget to try it with both oscillators audible (if listening to just one isn't wild enough)!

### **Ring Modulation**

---

Ring modulation is created by multiplying two audio signals together ( $A \times B$ ). This multiplication results in the sum and difference frequencies of the audio signals appearing.

If you take a 440 Hz (“standard” A4) sine wave and a 2093 Hz (C7) sine wave and ring modulate them (multiply them) you get:

$440 + 2093 = 2533$  Hz (gratingly between Eb7 and E7)

$2093 - 440 = 1653$  Hz (a noticeably flat G#6)

### **Exercise 1: Basic Ring Mod**

Let's try it out. Bring up the basic patch.

- Set ALL OSC Level = 0
- Find the **Ring Mod 4→3** parameter and increase slowly to 100. Just like with Hard Sync and FM, this parameter means “Use OSC 4 to modulate OSC 3”

Start playing notes up and down the keyboard.

A few things you should notice immediately:

- You're hearing sound even though all oscillators are at 0

## The Definitive Guide to Evolver

- Changing the value of “Ring Mod” doesn’t do anything beyond changing the volume of the Ring Mod signal

Now adjust the frequency of the modulator (in this case, OSC 4). Start by changing it to G1. Then try D#0. Finally, try something higher, like C#3.

Also try adjusting the “Fine” parameter for the modulator. Notice how detuning even a little bit can introduce some very interesting variation to the sound?

Listen to how different modulator pitch values change the sound. You may have already noticed that many of the timbres have a metallic or bell-like quality to them – that’s where “ring modulation” gets its name.

### **Exercise 2: Lord of the Ring Mod**

If waveforms other than sine waves are used, the ring modulator’s output is more complex, including the sum and difference frequencies of all of the waveform’s harmonics, too.

Go back to the basic patch.

- All OSC Levels = 0
- Ring Mod 4→3 = 100
- OSC 4 shape = 04

Now play a few notes. Kinda funky!

Now set the carrier (OSC 3) to shape 08 and the modulator (OSC 4) to shape 52. Play a few notes. Down in the lower octaves, this sound has an almost “vocal” quality.

### **Practical Applications and Advanced Tips**

One nice thing about Evolver’s ring modulation is that both the carrier and the modulator track the keyboard.

Traditional ring modulators frequently used a fixed frequency for their modulation which produced rather dissonant results as different notes were played.

## The Definitive Guide to Evolver

You can re-create more traditional sounds by setting the modulator (OSC 4 in this case) to have a glide setting of “OFF”, which will force it to always play a constant pitch.

### **Both Sides Now**

Ring modulation is multiplication, right? Since it doesn't matter whether you multiply  $A \times B$  or  $B \times A$ , it follows that **Ring Mod 4→3** should sound the same as **Ring Mod 3→4** for a given set of settings. Verify this is true. This means you can create nice stereo sounds using Ring Mod and various other parameters (like Filter Split).

### **Oscillators too**

Remember, the “ring mod amount” parameter is simply a level control, determining how much of the ring modulated signal is mixed in with your original oscillators.

I find Evolver's ring modulation to be surprisingly handy, and able to produce some very crisp and clean tones. In fact, I am often able to get more traditionally “FM-sounding” timbres using Ring Mod than from using Evolver's FM!

Remember to save anything that sounds good.

## Shape Sequence

---

Evolver also allows you to designate a “shape sequence” for each digital oscillator. The “Shape seq” parameter allows you to select 0 (no shape sequence) or select sequence 1, 2, 3, or 4 to be the “shape sequence”. At each step of the sequence, Evolver will change the waveshape for the selected oscillator to the value indicated by the sequence.

The easiest way to understand this is to simply program in a bunch of random numbers between 1 and 96 for Sequence 1. Set oscillator 3’s “Shape seq” parameter to 1 (so it uses Sequence 1). Press “Start” on the sequencer and enjoy the results.

This can also be applied to make each note you play sound different.

One obvious and fun thing you can do is combine shape sequences with FM and/or Ring Mod to produce sounds that change dramatically from note to note.

Finding exactly the right waveshapes is often a matter of trial and error. Eventually, you may find you have a few favorites. And don’t forget that you can create your own and load them into the user slots.

## Some Final Digital Oscillator Tips

---

### FM + Ring Mod

Remember that you can use FM and Ring mod on both oscillators, and that you can use both at the same time.

### Use ‘em All

You can mix the digital oscillators with the analog oscillators – each can supplement the other.

### A Little Bit Goes A Long Way

Small changes can have big effects.

The “Shape Sequence” effect is very similar to “wavesequencing” on the Korg Wavestation.

Dave Smith led the design team for the Korg Wavestation!

## Noise

---

Evolver has a “5<sup>th</sup> oscillator” – a white noise generator. Since it isn’t pitched, it is not technically an oscillator. But it is another sound generation (and modulation) component, and one that will be used for many great effects and sounds.

Like the other oscillators, Evolver’s noise generator sits before the filter and before the amplifier, so it will be affected by both of those things as well.

Noise appears equally in both the left and right outputs.

Its only parameter is “volume”.

### Practical Applications

The obvious uses are things like “wind”, “waves”, “explosions”, “static”, and other white noise sounds.

With the addition of short envelopes in the VCA and filter, you can make some interesting drum sounds – we will study drum sounds later on.

You can use noise to add perceived brightness to a sound – mix a little in there quietly. It’s also great for sequences if you want them to be more percussive. For these kinds of applications, a little bit of noise goes a long way – you don’t have to overdo it!

The classic 80s “Simmons” toms, kick, and snare were not much more than plain old sine or triangle waves whose pitches were controlled by envelopes with a little white noise mixed in.

## The VCA and Envelope 2

---

VCA stands for “Voltage Controlled Amplifier”. It’s a fancy way of saying “volume control”. The VCA is like a volume control for each sound.

Evolver’s VCA is “hard-wired” to Envelope 2 – so ENV 2 always controls the VCA. The hardwired envelope provides a way to adjust that volume over time, every time a key is pressed. Using one of the modulation slots, you can assign ENV 2 to other things as well, but it will always control volume as well.

The VCA, when coupled with ENV 2, determines how loud the individual patch gets over time after the envelopes are triggered (usually by pressing a key or the sequencer).

## ADSR Explained

---

Evolver has 3 envelopes. Envelope 1 (ENV 1) controls the lowpass filter, and is sometimes called “the filter envelope”. Envelope 2 (ENV 2) controls the VCA. Envelope 3 doesn’t directly control anything – it is available as a modulation source (and will be discussed more in-depth later).

All three of Evolver’s envelopes include the same basic functions and employ them to similar effect. All 3 envelopes are so-called “ADSR” envelopes – the letters stand for “Attack”, “Decay”, “Sustain”, and “Release”. Here’s what each parameter does:

- **Attack** – A time/duration measurement. How long it takes for the controlled parameter to reach a maximum value
- **Decay** – A time/duration measurement. How long it takes for the controlled parameter to decrease from maximum to its “sustain” value
- **Sustain** – Unlike the others, sustain isn’t a time or duration. It is the level at which a given parameter remains when the key is held down. The sustain period only happens after the attack and decay phases have been completed.
- **Release** – A time/duration measurement. How long it takes for the parameter to return to zero after the key is released

## The Definitive Guide to Evolver

In the case of the VCA, imagine you have your hand on the volume knob of your stereo. It starts at 0 (no volume). When I tap you on the shoulder, you're going to turn the volume up to maximum over a period of time I've already given you – that's the "A". So I tap you on the shoulder and you turn the knob over the period of time.

The "D" is what comes next – right after you hit the maximum, you immediately begin turning the volume down. You're going to stop at the value I gave you for "S". How fast or slow you go is the "D" parameter.

You stay at "S" until take my hand off your shoulder, at which point you start turning the knob back down to zero.

The length of time it takes you to go from wherever "S" is to zero volume is the "R" parameter.

The ADSR envelope has proven a popular and enduring design because it combines simplicity with great flexibility. For example, the volume or amplitude characteristics of many sounds can be approximated using an ADSR envelope.

### Exercise 1: Some Basic VCA Envelopes

Here are some examples of envelope settings for typical applications:

- **Strings** – Slow attack (A=50), no decay (D=0), full sustain (S=100), and long release (R=50)
- **Woodwind** – short attack (A=20), short decay (D=20), moderate sustain (S=75), and no release (R=0)
- **Drums/percussion** – Always have an immediate attack (A=0), a short decay (D=25-50), no sustain (S=0), and no release (R=0, since you can't typically stop a drum once it's hit)
- **Piano** - Immediate attack (A=0), long decay, no sustain, and no release)

Try them on the basic patch and hear how they change the sound. Try a few of your own settings, too.

Once you understand how ADSR envelopes work, it is pretty easy to apply the knowledge to any sort of sound. Think about how different envelope settings would affect the sounds we've made previously.

There are other parameters which relate to the VCA and its envelope: "VCA Level", "Envelope Amount" (Env Amt), "Velocity", and "Exponential/Linear Envelope" (Exp/Lin Env).

### **VCA Level**

The VCA level is the VCA's "baseline". If VCA level is set higher than 0, the oscillators will be heard constantly, whether the envelope is going on or not. This is handy if you want to make "drones" that don't require keys to be pressed to be played.

The VCA envelope's values "start" from this level (and return to it when finished). This means you can make a sound that is always droning quietly until a key is pressed, at which point it follows a volume contour determined by the envelope.

This also means that if you set the VCA level to be 100 (the maximum), no matter what you set the envelope to, you won't hear any difference in volume – it will be at the maximum the entire time, since the volume envelope only adds to the VCA level and it is already "maxed out".

### **Envelope Amount (Env Amt)**

Envelope Amount determines how much the envelope affects the overall VCA level.

If set to 0, Envelope 2 has no effect on the VCA at all – it might as well be set to all zeros. As you increase the Env Amt parameter, it effectively increases the maximum available to the VCA. That means it determines the level the VCA reaches at the end of the Attack stage, as well as the level that will be reached when Sustain is set to 100.



### **Velocity**

This parameter affects how much MIDI velocity affects VCA envelope level. It makes higher velocity notes create louder “maximums” and sustains.

At 0, velocity has no effect whatsoever – the value is ignored. As you increase the velocity parameter, the effect that velocity has on the VCA envelope is increased.

The higher the value, the more your patch will respond to MIDI velocity by changing volume. Best settings depend heavily on your playing style, the controller you use, and of course, the effect you desire and the sound you are creating.

### **Exponential/Linear Envelope**

This parameter affects whether the A, D, and R phases of all 3 envelopes are linear or exponential.

Linear envelopes change the same amount over a given time interval – if you drew it out on graph paper, it would be a straight line. For example, an Attack stage setting of 99 might correspond to a 3 second envelope. This would mean that over each second, the numerical value increased by 33.

Exponential envelopes do not change the same amount over time – they have a “curved” response.

Because of the way our hearing works, humans tend to perceive exponential changes in sound as being smoother or more regular than linear changes.

### **Not Master Volume**

Note that all of these VCA parameters are different from “Master Volume”. The Master Volume control is after the VCA (and everything else, in fact). So no matter how “loud” the VCA gets, if the Master Volume is set low, the patch won’t be very loud.

## The Filters

---

We have managed to create a wide variety of sounds using just oscillators and envelopes. But filters are important too.

Filters are the primary way the “raw” sound of a synthesizer’s oscillators is shaped into something interesting and useful.

Most synthesizers (including Evolver) use a “subtractive synthesis” method, where the basic oscillator sound is filtered, to remove (or subtract) frequencies from the basic sound to produce new timbres.

There are 4 basic types of filters – lowpass, highpass, bandpass, and band-reject. Evolver features lowpass and highpass filters. Bandpass filters can be created by placing a highpass and a lowpass filter one after the other, so it is possible to use both of Evolver’s filters to emulate a bandpass filter. “Band-reject” filters (also called notch filters) are like the reverse of bandpass filters – they remove a specific set of frequencies.

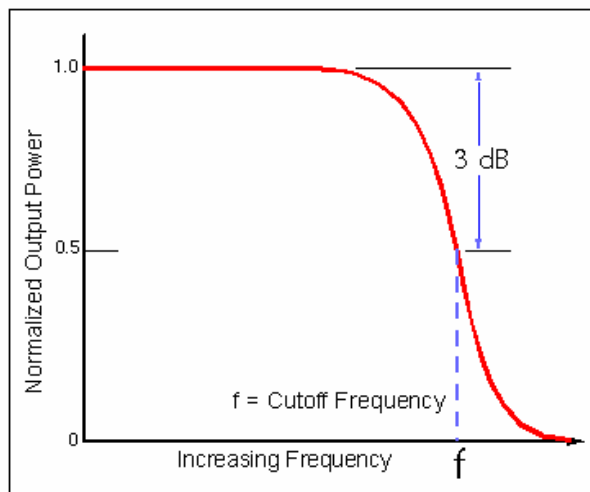
You may have heard about highpass and lowpass filters on other synthesizers, and how they really made a given synthesizer sound great. Does that mean Evolver’s lowpass and highpass filters sound just like those of synthesizer “X”? No. Every synthesizer’s filters are a little bit different. It’s like chocolate cake – everyone has their own “secret recipe” which provides a variation in flavor.

### Lowpass Filter

---

A lowpass filter (LPF) blocks high frequencies and allows low frequencies to pass through.

A LPF doesn’t just stop at an exact frequency, however – it reduces some frequencies above the “center” cutoff frequency as well. As you get farther from the center, the amount of reduction of frequencies decreases.



low pass filter

Evolver includes analog low-pass filters for the left and right channels. The fact the filters are analog and separate means they will always respond slightly differently, adding to Evolver's sonic character. However, the filters are identical in terms of features and functions, and are generally treated as a single filter.

### **2-pole vs. 4-pole**

Evolver's filters can be operated in 2-pole or 4-pole mode. "2-pole" filters have a 12 dB per octave roll-off, which just means that frequencies below the cutoff are reduced at that rate (12 dB per octave). It is a "milder" reduction or less-extreme sounding filter relative to the 4-pole filter.

4-pole filters have a 24 dB per octave roll-off. 4-pole filters have a more aggressive reduction and more extreme character relative to the 2-pole filter.

### **Frequency**

The frequency parameter simply determines the center of where the filter starts to work – this is referred to as the "cutoff frequency".

It is also the point from which the filter starts to move when controlled by the filter envelope.

The term "pole" comes from digital signal processing theory. It'd take too long to explain more.

On other synths, LPF cutoff frequency may be labeled simply "cutoff" or "frequency".

### **But What's It For?**

As previously mentioned, filters take the basic, raw sound of the oscillators and subtract frequencies, thus changing the harmonic structure. This change in harmonic structure change helps create a more interesting sound.

### **Exercise 1: Filter Frequency**

Bring up the basic patch. Set the LPF to "wide open" – a setting of 164. Assign the mod wheel to Filter Frequency, with an amount of -99. Hold down a note and move the mod wheel around. Hear how it damps all the high frequencies out as the wheel is increased?

## The Definitive Guide to Evolver

If you don't have a mod wheel, just access the Filter Frequency parameter on your Evolver and turn the knob.

Now try it with the filter set to 4-pole mode instead of 2-pole mode. Hear a difference?

Try different notes, too.

### Exercise 2: Resonance

Evolver's LPF has another important parameter called "resonance". Resonance emphasizes the filter's character by creating a feedback loop – the output of the filter is fed back into its input. Resonance is so named because the feedback is similar to the filter "resonating" at its cutoff frequency.

The higher the value of the resonance parameter, the more signal is fed back to the filter.

Adding resonance emphasizes the filter's character. Lower levels can provide a very funky "honky" or "quacky" sound.

Let's try it out. Bring up the basic patch (I hope you're not sick of it yet!).

Make the following changes:

- OSC 3 and 4 Level = 0
- Mod Wheel Destination = rES (resonance)
- Mod Wheel Amt = 99
- Filter Frequency = 95
- 2/4 pole = 2 P (2-pole)

Play a note – middle C or lower. Hold it down, and start moving the mod wheel around. Try some very low and very high notes.

Do it slowly, so you can hear the filter's effect on the timbre as the value changes.

Now, let's see what happens with 4 pole mode. Change the 2/4 pole setting to 4 P (4-pole). Turn the mod wheel all the way down.

On other synths, including the legendary MiniMoog, LPF resonance may be called "emphasis" because it emphasizes the filter frequency and sound.

## The Definitive Guide to Evolver

Hold down a note and start turning up the mod wheel. What happens as you increase the mod wheel to maximum?

You should hear a sort of feedback or “howling”. This is because when Evolver’s lowpass filter is in 4-pole mode, high resonance values cause the filter to feedback so much it “self-oscillates” at the filter frequency.

This isn’t surprising, given that it is a feedback loop. There’s so much filtered signal being fed back and filtered again that the filter frequency becomes prominent enough that it is heard as a kind of “howling” sound.

Remember the filter’s resonant frequency is determined by the cutoff parameter – as you change the cutoff frequency, the resonant frequency changes, too.

You’ll notice that as you play notes up and down the keyboard with high resonance, certain notes seem to “jump out” more, depending on the relationship of the pitch/frequency of the note to the filter cutoff frequency. Some notes “beat” horribly against the filter’s resonance and some don’t.

This makes sense if you think about it – the note frequencies are changing as you play up and down the keyboard, but the filter cutoff (and thus resonant frequency) isn’t. This results in the uneven filter response and varying sound across the keyboard.

It sure would be handy if there was a way to adjust the filter’s cutoff frequency automatically with the notes, wouldn’t it?

### **Exercise 3: Key Amount**

Key Amount (“Key Amt” or “Kbd Amt”) to the rescue! This parameter simply adjusts the filter cutoff depending on the MIDI note number.

At a value of 72, the filter cutoff moves exactly the same amount as the keys change, resulting in “balanced” response.

Key Amount values lower than 72 means the filter cutoff goes up slower than the note frequency. Values higher than 72 mean the filter cutoff goes up higher faster than the note frequency.

Let's try it out!

Adjust the filter Key Amt setting to 72 and play up and down the keyboard again. It's very different, isn't it?

Leaving Key Amt at 72, change the LPF Frequency to 72. Now play up and down the keyboard. You will hear the filter oscillating in tune with the notes.

Experiment with different LPF frequency settings in conjunction with very high resonance values.

You will notice that the filter's frequency can change and go slightly sharp or flat depending on how much resonance is being applied to it.

### **Modulation is Key**

The previous examples allowed you to explore the basic filter parameters.

By modulating the filter's parameters, the sound can be made still more interesting. Let's explore.

### **Env1 – The Filter Envelope**

---

This envelope is assigned to control the cutoff frequency of the low-pass filter. It works just like the VCA envelope except it controls the filter cutoff frequency instead of volume.

One useful difference from the VCA envelope is that the "Env Amt" parameter is bi-directional for the filter, which means the envelope can decrease the value of the filter rather than increasing it.

### **Practical Applications**

There's a reason Evolver has an envelope dedicated to filter cutoff. The powerful, flexible, time-specific control of the ADSR envelope, coupled with the

sound-shaping filter provides one of the best ways to create an interesting synthesizer sound.

### Exercise 1: Filter Sweep

This is one of the most basic and well-known synth sounds. Start with the basic patch.

- OSC 3 and 4 = shape 02 (sawtooth)
- Give each oscillator a Fine setting between -5 and 5 – try to balance them out
- Filter Frequency (LPF Frequency) = 0
- Filter Resonance = 62
- Filter 2/4 Pole = 4P (4-pole)
- ENV 1 Attack = 98
- ENV 1 Sustain = 100
- ENV 1 Amount = 99

Now press a low key and hold it down. *Bwaaaaah!*  
Entire musical careers have been built on that sound.

All we're doing is having the Filter Envelope modulate (control) the filter frequency!

### Exercise 2: Filter Pop

OK, without changing anything else, let's just change the settings of ENV 1:

- Filter Attack = 10
- Filter Decay = 29
- Filter Sustain = 10
- Filter Env Amount = 99
- Filter Key Amt = 72

Now play a few notes. Nice and punchy.

This is somewhat similar to the short, poppy envelope we used back on the PWM horn sound, only now it is adjusting the filter frequency instead of pulse width.

### Exercise 3: Extreme Resonance

When set to 4-pole mode, the LPF's resonance can produce some interesting results at extreme settings.

## The Definitive Guide to Evolver

Bring up the basic patch.

- All OSC Level = 0
- Noise Volume = 50
- LPF to 4 pole mode
- LPF resonance = 100.
- Mod 1 Source = MIDI note number (Nno)
- Mod 1 Amount = 99
- Mod 1 Destination = LPF Frequency (FiL)

Now play a few notes. This is a recreation of a popular sound. What you're hearing is the Evolver's noise generator being "pitched" by extreme filter resonance. It has an ethereal, mysterious, almost watery quality.

Go ahead and turn the noise volume to 0. Now play a few notes. You can still hear sound, even though none of Evolver's sound generators are being used. That sweet sound is just the filters resonating. If it sounds a little like guitar feedback, it's because it is very similar in concept.

You may also notice the sound has a bit of "beating" or chorusing – that is evidence that the analog filters are operating independently and aren't behaving identically.

But wait, there's more! Turn the distortion on and crank it up to around 80. Make sure you back your master volume down, because this can get loud. Now you're getting something that almost sounds like a square wave.

You can also play around with the VCA envelope. Try a few long, slow attacks and decays to make some nice ethereal filter feedback tones. Then do a few short, snappy envelopes. Save anything you like.

### **Exercise 4: Let's Split!**

Remember, Evolver actually has 2 separate lowpass filters – one for the left channel, and one for the right.

Because they're both analog filters, they always behave a little differently, which helps Evolver sound rich, especially in stereo.



Using the “split” parameter, you can increase the difference between the two filters. “Split” is an offset value – it increases the frequency value of the left filter while decreasing the frequency value of the right filter by the same amount.

Even small amounts of filter split can have big effects on your sound. At extreme values, split forces the left filter wide open while keeping the right completely damped down.

You can use an LFO to modulate split. Depending on the degree and speed, this can provide some subtle variation or heavy throbbing.

Note that split will produce 2 different “notes” for extreme resonance patches and it can be very tricky to tune them.

### **Exercise 5: Independence**

Left and Right Filter Cutoff and Resonance are also modulation destinations. So rather than having their changes mirrored, you could use different modulation sources to produce totally independent filter effects.

Bring up the basic patch

- OSC 3 and 4 level = 0
- LPF = 4 pole
- LPF Frequency = 76
- LPF Cutoff = 52
- LFO 1 Destination = Filter Cutoff 1 (Left Filter) – F1F
- LFO 1 frequency = ST2 (2 cycles per step)
- LFO 1 amount = 39
- LFO 1 shape = Reverse Saw
- LFO 2 destination = Filter Cutoff 2 (Right Filter) – F2F
- LFO 2 frequency = St8 (8 cycles per step)
- LFO 2 shape = Reverse Saw

Play a few notes. You can hear the two LFOs addressing each filter independently.

LFOs are just one way you can control each filter. Any of the other modulation sources are great for this- ENV 3, the sequencer – get creative and try a few others before moving on.

### **Velocity**

This parameter determines how much the filter envelope is affected by MIDI note velocity. At 0, velocity has no effect. As you increase this value, louder notes will force the filter to open more.

### **Audio Mod**

The “Audio Mod” allows the sound of the analog oscillators themselves to affect the frequency of the lowpass filter. If the analog oscillators are being used, audio mod can add a “sizzly” quality to your sound.

### **Summary**

The lowpass filter’s frequency and resonance interact to shape the “raw” sound of Evolver’s oscillators.

At its most simple, the filter removes high frequencies (and thus emphasizes the low frequencies). But the filter really comes alive when modulated.

The lowpass filter frequency and resonance values can be modulated a variety of ways – using the filter envelope is one. Using LFOs is another.

Skill and creativity with filter control and modulation will get you a long way in mastering subtractive synthesis, and Evolver provides many neat things you can do.

### **Highpass Filter**

---

Evolver features 2 identical 4-pole digital highpass filters (HPF). Highpass filters are the reverse of lowpass filters: they block low frequencies and let high frequencies pass through.

Evolver’s HPFs are always treated as a single filter, and will be referred to as a single filter henceforth.

## The Definitive Guide to Evolver

The HPF can be placed in one of two locations in Evolver's signal path. It can be placed either after the low-pass filter, before the delay (for synth sounds), or following the external input (for processing external audio).

Evolver's HPF is digital, unlike the LPF, which is analog. Because the HPF is digital, it shouldn't exhibit any of the variation of the dual analog low-pass filters, so treating it as a single stereo filter is fine.

Evolver's HPF is only available in 4-pole configuration and has no resonance control.

The HPF has only a single parameter: Highpass ("Hi-pass Filter"). The single parameter controls both cutoff frequency and where in the audio path the filter is placed:

- 0 = HPF is completely bypassed and is not in the circuit.
- o 1 - o99 = **HPF on the oscillators**. As the value rises to 99, the cutoff frequency rises. At 99 only the highest frequencies are let through.
- i 0 - i99 – **HPF on the input**. As the value rises, the cutoff frequency rises. At 99, only the highest frequencies from the input are let through.

The highpass filter has an undocumented feature: When set to "0", it is removed completely from the circuit.

So if you map the highpass filter to a controller, such as the mod wheel, you must set the highpass filter's frequency parameter to 1 or higher (whether it is being used for the oscillators or the external input) to get any effect at all. If you leave it at zero, all the modulation in the world won't make it do anything.

### Practical Applications

The highpass filter cuts out low frequencies. The obvious effect is that if your sound is too bassy or muddy, you can reduce the low end by using the HPF.

## The Definitive Guide to Evolver

A not-so-obvious corollary is that the HPF effectively emphasizes the higher frequencies – “cutting low frequencies” isn’t really different from “boosting high frequencies”. So if you want to make very buzzy or bright sounds, bringing in a little (or a lot) of HPF will get you there.

Try monkeying around with the HPF on the basic patch, or any other patch you can find. Its effect is pretty easy to understand, and since there’s only one parameter that can be adjusted, it is easy to explore.

Note that many of the same things you tried for the LPF can be applied to the HPF. Try doing a HPF filter sweep. Load the Basic Patch.

- ENV 3 Attack = 78
- ENV 3 Decay = 98
- ENV 3 Amount = 99
- ENV 3 Destination = HPF (Highpass Frequency/HiP)
- Highpass = 0.5 (so it can be modulated)

Play a few notes.

Try setting up an LFO to modulate the HPF parameter. Experiment with different waves, frequencies, and amounts – a little modulation of the HPF (slow rate, shallow depth) can make a dull sound a little more interesting

Or try using ENV 3 to modulate the HPF– this can provide some interesting sonic contours to a sound

### **Creating a Bandpass Filter**

Earlier I mentioned you can create a bandpass filter by placing a lowpass and highpass filter “in series”, as Evolver does.

This is logical – if an LPF removes low frequencies and leaves some high frequencies, and an HPF removes high frequencies and leaves low frequencies, putting them together should remove most of the lows and highs and leave some “middle frequencies”.

How many of those frequencies are left depends on

## The Definitive Guide to Evolver

how far apart the LPF and HPF filters are – the closer in frequency the LPF and HPF, the fewer frequencies are let through. [INSERT CHART].

Exactly which frequencies are let through depends on where exactly the LPF and HPF cutoff frequencies are.

Let's build a sample bandpass filter and play with it a bit. Bring up the basic patch.

- LPF Frequency = 76
- LPF Resonance = 0
- LPF mode = 4 pole
- Mod Wheel Dest = LPF Frequency
- Mod Wheel Amount = 99
- Mod Slot 1 = Mod Wheel -> HPF Frequency
- Mod Slot Amount = 99

Move your Mod wheel around and you'll hear the two filters working in tandem, sliding around to whatever frequency the Mod Wheel specifies.

We have the LPF mode set to 4-pole and resonance set to 0 so that it matches the HPF (which, remember, only operates in 4-pole mode and has no resonance).

We also are making sure to apply the same amount of modulation to both filters at the same time.

Doing the above keeps the bandpass filter response symmetrical (meaning it has the same shape on both sides). What happens if we change these settings?

Try:

- Switching the LPF to 2 pole mode.
- Adding resonance to the LPF
- Altering modulation amounts of each filter
- Using LFOs to control one or more of the filters

## Let it Rip!

Just writing out the title and section headings of this part of the guide makes me smile. **Distortion. Feedback. Grunge. Output Hack.** Sounds brutal, nasty, and raw, right? So will your Evolver, because it includes these features. Any time guitarists start giving you lip about how synths are wimpy, you can bust out some Evolver tones and make them go cover in fear behind their archaic Marshall stacks.

Though synth players have swiped guitar fuzzboxes for years, the addition of these sorts of effects to synthesizers is a relatively new phenomenon. Evolver includes several different flavors that can help you create some true sonic mayhem.

### Distortion

Evolver features a nice digital distortion circuit. In the musical world and Evolver in particular, distortion results from extreme increases in gain that result in a sort of “overload”.

Distortion is also known as “clipping” because as the gain of a signal is increased to the point of distortion, the tops of the waveform are “clipped off”. This results in the squaring off the tops of waves. As you know by now, changing the shape of a waveform changes its timbre.

You have probably also figured out that adding distortion dramatically increases volume. You’re right, so make sure to back down your master volume for this section to avoid waking the neighbors.

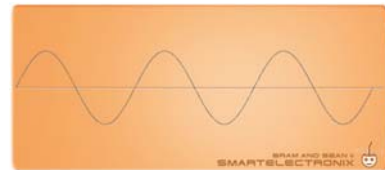
Evolver’s digital distortion can be placed to affect its internal oscillators or placed to affect the external input.

If used on the internal oscillators, the distortion is placed after the LPF and VCA, in front of the delay.

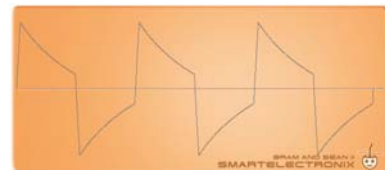
If used on the external input, the distortion is placed before the LPF. This lets you use the LPF to shape the raw distorted sound of the external signal – applying



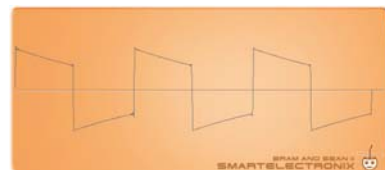
High gain causes wave to clip



Sine wave



Sine wave, distortion = 10



Square wave – similar to distorted sine!

distortion typically exaggerates the high frequency content, and the LPF helps compensate for that..

In either case, the distortion parameter ranges from 0-99, where higher settings provide more distortion.

Whenever the distortion is on (i.e. not 0), Evolver's noise gate is activated. The noise gate is controlled by the left channel only, so if the left channel level is turned down close enough to zero, it will shut off the noise gate on both channels.

This is not a problem for levels above around 10, but it can cause some strange behavior at low levels, where even though oscillator 2 is set to a high value and is playing, it won't be heard because oscillator 1 is too low to trigger the noise gate.

### Isolating the Noise Gate

The noise gate can be used alone (without distortion), by setting distortion to "1". A setting of "o01" applies the noise gate to the internal oscillators, a setting of "i01" applies the noise gate to the external input.

### Feedback

---

Evolver includes 2 special "feedback" lines which take the output signal from the VCA and "feed it back" to the input. The left and right channels each have separate feedback lines which are governed by one set of controls, similar to the LPF.

Note these feedback lines are not to be confused with the feedback parameter of the digital delay!

These feedback lines are tunable – the "Feedback Frequency" parameter controls their pitch, from C0 to C4. The actual frequency can be affected by LPF settings as well.

By default, the feedback does not track incoming pitches. It is static, at the value of "Feedback Frequency". It can be controlled by MIDI note, just like we did earlier with "extreme resonance":

## Exercise 1: Tuning Feedback

Bring up the basic patch

- Modulation Slot 1 and 2 source = “MIDI Note Number” (Nno)
- Modulation Slot 1 and 2 target = “Feedback Frequency”
- Modulation Slot 1 and 2 amount = +99

Leave the base Feedback Frequency set to “0”, or it will “add” to the note number and play transposed.

Feedback also has a “level” control, which determines how much signal gets fed back. Feedback Level ranges from 0-100. Crank it up to 100 for now.

## Grunge

According to the Evolver manual, the “Grunge” parameter emphasizes the feedback effect, particularly when feedback is high. Grunge is either “on” or “off”.

While the inner workings of the Grunge circuit remain a secret, activating “Grunge” imparts a harsher quality to the distortion sounds. It definitely makes the sound “ruder”.

## Output Hack

The Evolver manual states that Output Hack “Trashes the output signal, quite rudely.”

Output Hack provides only a single parameter (amount) which ranges from 0-14. As you increase this parameter, the sounds gets more “hacked”.

To my ears, it sounds like some sort of hard bit rate reduction (also known as “bit crushing” or “truncation”). This also makes sense given the range of the parameter – at 0 you are getting the full 16 bits (subtracting 0) and at 14 you are only getting 2 bits (subtracting 14)! As Output Hack increased, the sound becomes increasingly nasty, gnarly, and low-fidelity. Wicked!

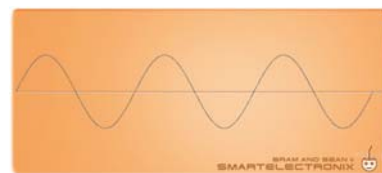
Bring up the basic patch. Play a few notes, then increase the Output Hack amount by 1. Repeat until you’ve gone up to 14 and back down.

Evolver enables MIDI notes to produce quarter-tone differences when MIDI Note IN is used as a modulation source and the destination amount is set at 99.

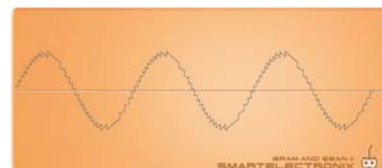
This means you have to use a second slot to get true semi-tone differences.

Evolver’s modulation is additive – each additional modulation source “adds” its amounts to the destination, so 2 quarter-tone slots produce the same effect as 1 half-tone slot would!

This is important to remember any time you want to use MIDI Note IN to provide modulation that exactly tracks the keyboard pitches.



Sine wave



Sine wave, output hack=12



Output hack can help you produce some nice “lo-fi” textures and sounds.

### **Practical Applications**

These various forms of distortion allow you to change the basic sounds your Evolver produces. Lower levels can add a nice patina and some warmth to your sounds. Higher levels can really help you make something nasty, rude, and brutish. Industrial. Punk. You name it.

But you can also make some very nice ethereal textures with distortion as well.

Feedback has some other neat uses. We’ll explore this a bit later when we talk about Karplus-Strong synthesis.

## The Digital Delay

---

A digital delay is a very simple thing – it repeats signals after a pre-defined time.

Evolver's digital delay is the same quality as Evolver's A/D and D/A converters (16-bit, 48 KHz) and can provide up to 1 second of delay time.

Evolver's delay sums both Left and Right channels to mono (L+R) for its input. Evolver's delay output is fed back to both Left and Right channels equally, which places it in the center of the stereo field at all times.

The delay offers 3 "taps". In this case, "tap" is a digital signal processing (DSP) term. Imagine the digital delay as a pipeline that is 1 second long. Evolver's signal goes in one side of the pipe, and 1 second later, comes out the other side.

We can "tap into" the "pipeline" of delayed signal at 3 points. Each tap can be at the same place on the pipeline or the taps can be placed at different points on the pipeline.

Tap placement corresponds to a delay time – placing a tap near the start of the pipe produces a very short delay, while placing a tap at the very end would produce a delay of 1 second.

We can also control the amount of tapped signal.

Each of the 3 taps has a **time** parameter and a **level** parameter.

**Time** determines where the delay pipeline is "tapped". The range of time values is divided into 3 sets of ranges (more about that in a moment).

**Level** simply determines how loud delayed signal is played – 0 is "not at all" and 100 is "as loud as the original".

## Basic Delay

---

A basic audio delay is the simplest effect there is – you hear the original sound, and then some period of time later, you hear it repeated again. Let’s build a simple 1-tap delay.

### Exercise 1: 1-Tap Delay

This couldn’t be easier! Bring up the basic patch.

- Delay Level 1 = 75
- Delay Time 1 = 144

Play a short note. Hear it “echo”? Congratulations! You’ve made a 1-tap delay.

You are using 1 delay, or “tap” into our pipeline of signal, and the output of that “tap” is what you’re hearing as the echo. Simple and useful, but not terribly exciting.

Let’s explore that “time” parameter.

In Exercise 1, the delay time parameter was set to 144. Delay time’s parameter ranges from 0-150. However, unlike most other Evolver parameters which provide continuous adjustment, the “delay time” parameter actually provides 3 different kinds of time measurement: **plain**, **tuned**, and **synchronized**.

“**Plain**” delay times run from 1-21 and 95-150. These are just different time durations. Higher numbers mean longer delays. At 150, you’re using the full 1 second of delay time. At 1, you’re using such a short delay you almost can’t hear it. Very short delay times (1-21) are the best when you want to create non-rhythmic ambience. Delay times from 95-110 are great for “slapback” echo effects. Delay times from 110-150 create perceptibly discrete delay repeats.

The “**tuned**” delay times are specified in the middle of the parameter range, from steps 22-94. Each “tuned” delay value corresponds to a musical frequency. From the Evolver manual, “Since delay is a time measurement, higher delay numbers are lower frequencies.”

| Value  | Delay type   |
|--------|--------------|
| 0-21   | Plain/short  |
| 22-94  | Tuned        |
| 95-150 | Plain/long   |
| 151+   | Synchronized |

## The Definitive Guide to Evolver

For example, the “lowest” (i.e. shortest) tuned delay value of 22 corresponds to 2,093 Hz, or note C7, which is a fairly high pitch. The “highest” (longest) tuned delay value of 94 corresponds to 32.7 Hz, or note C1.

A table showing all tuned delay values and their musical notes is included in Appendix D.

Uses of the “tuned” delay times will be explored later.

**“Synchronized” delay** times appear when you increase the parameter past 150. Evolver provides different note values relative to the sequencer clock and tempo for your patch.

These range from as long as “all 32 steps of the sequence” to as short as “1/6<sup>th</sup> the length of a sequence step”.

These delay times are best-suited for making rhythmic delays, since they’ll automatically lock to incoming MIDI clock when you want them to.

### **Exercise 2: Multi-Tap Delay**

Why just use 1 tap when you have 3?

- Delay level 2 = 50
- Delay level 3 = 30
- Delay time 1 = 110
- Delay time 2 = 125
- Delay time 3 = 150

Play some notes. So now you have a little 3-echo delay. Each “tap” is occurring at a different time. You have just made a “multi-tap delay”.

Try varying delay levels and times a bit until you understand how these things work. For example, you could make delay 1 very quiet and delay 3 very loud, so the delays fade in instead of fading out.

You can also set your delay times close together for interesting effects.

### Exercise 3: Delay Feedback

Our multi-tap delay is fun, but limited: It only creates 3 repeats. That's not very flexible or interesting. Fortunately, Evolver has 2 other delay parameters which will help out: **(Delay) Feedback 1** and **(Delay) Feedback 2** (appearing as "Dly > Dly Input" and "Dly > Flt Input" in the Soundtower Editor, respectively). I add the "Delay" on the front of the parameter name to avoid confusion with the "distortion" feedback parameters – they are very different!

Feedback 1 takes the output of our delay "pipeline" and mixes it back into the input. The higher the value for Feedback 1, the higher the level of signal mixed back into the input.

Let's hear what that does. Bring up the basic patch.

- Set up at least one delay time to 125
- Set the level of the delay to 70
- Set Feedback 1 to 50

Play a few notes. Hear how the repeats keep going? Try different settings of Feedback 1. The higher Feedback 1 is, the more "repeats" you get.

Feedback 2 does same thing – it feeds the delay output back into the input, except it goes into the LPF before the delay. This allows the filter to affect the delayed signal. For example, you can apply some filter to have the delay repeats get progressively darker.

Turn Feedback 1 to 0, turn Feedback 2 up to 75 and hit a few notes.

### **Advanced Delay: Making Effects**

---

It may seem difficult to believe this, but a simple multi-tap delay can be used to make all sorts of great audio effects, provided it allows for modulation of delay time. Fortunately, each of Evolver's 3 taps can have their time and levels modulated.

Chorusing and flanging are closely related. Both are "time delay" effects, and are fairly simple to create.

Both effects are created by mixing delay taps of varying time with the original signal.

### Chorus

Start with our basic patch.

- Delay level 1 = 75
- Delay time 1 = 85
- LFO 1 destination = delay time 1
- LFO 1 frequency = 25
- LFO 1 wave = triangle
- LFO 1 modulation amount = 15

Now play a few notes. You have just made a basic chorus!

Try some other values:

- Delay Time 2 = 40
- LFO 1 Frequency = 41
- LFO 1 Amount = 60

Also very nice!

Let's break down what's happening here. We are starting with a basic, fairly short delay time, at a volume level close to the original signal.

We are then using LFO 1 to modulate the delay time by +/-15 (so it runs from 60-100) at a rate of 25, using a triangle wave.

Because this is just a simple 1-tap chorus, it is probably not as rich sounding as other chorus effects you may have heard.

We can improve the chorus by adding more taps. The best results come from applying different modulation rates and depths to each delay tap. Many popular chorus pedals achieve their "smooth" and "creamy" sounds through this same mechanism. The goal is to introduce as much variation as possible.

As you increase the base delay times from 80 to 100, you will hear the chorus get more and more "detuned".

Since the actual delay time can never be less than 0, the LFO's modulation effect on the delay will always "bottom out" at 0 regardless of the LFO amount.

You can use this to intentionally create modulation effects that have different "shapes".

## The Definitive Guide to Evolver

As you increase the LFO modulation amount from 5 to 25, you will hear the chorus' "pitch wobble" increase.

Generally, I find good chorus results come from keeping the total delay range (the base Delay 1 Time +/- the LFO amount) between 75 and 95. It is a subjective thing, however, and your results may vary.

There's also some interaction between the LFO frequency and the LFO amount, in that faster frequencies will sound better with lower amounts – you can get some "classic" chorus sounds by keeping the LFO frequency up around 65-85 and the LFO amount around 5-7.

### Flanger

The only substantial differences between a chorus and a flanger are the delay times. Flanging occurs with short delay times, chorusing happens with longer delay times.

To create a flanger, we will adjust our controls to provide a shorter delay time and adjust the modulation to emphasize the flanging effect.

Start with the chorus we just built in the previous exercise. Now make the following changes:

- Delay 1 Level = 100 (to make the effect as obvious and loud as possible)
- Delay 1 time = 45. For flanging, you'll get best results staying between 20 and 80 – the lower the number, the higher the "resonant frequency" of the flanging effect.
- LFO 1 destination = delay time 1
- LFO 1 amount = 45. Delay range also needs to be broader than you'd use for a chorus. Too little and you won't hear an effect. Too much and it won't sound like flanging. Now you have a delay time that will swing from 0-90 (45 +/- 45)
- LFO 1 frequency = 20. Again, different rates change the way the effect sounds.

"Flanging" got its name because the effect was originally generated by recording the same sound on 2 different tape decks, then playing them back at the same time and slowing one of them down slightly by resting your hand on the "flange" of the tape reel!

## The Definitive Guide to Evolver

- Delay Feedback 1 (a.k.a. Dly > Dly Input) = 90 (to emphasize the flanging effect)

You have just made a flanger!

### More Taps?

You can make your choruses sound “richer” by using 2 or even 3 taps. Flanging usually works best with a single tap.

If you really want to get fancy, you can use some modulation slots to route some LFOs to slightly vary the delay levels for each of the taps in your chorus. Try using a different LFO for level than you’re using to vary the rate for a given tap (i.e. use Tap 1’s rate LFO to control Tap 2’s level and vice versa).

Regardless of the number of taps, you can always emphasize the chorusing or flanging effect by increasing Feedback 1, Feedback 2, or both.

### Summary

- To create either effect, modulate a delay time using a slow LFO
- For flanging, you want to keep your delay times ranging from 0-85
- For chorusing, your delay time should range from 80-100
- Make sure your Delay Level is pretty high – 75-100
- Increase either or both Delay Feedback channels to emphasize the effect
- More taps means a richer sound

### Resonator (Tuned Delay)

Remember how Evolver’s delay taps can be “tuned” to different pitches? You can use this to make a “resonator” that will apply an additional pitch to whatever material you play into it, whether from Evolver’s oscillators or the external input.

The simplest way to achieve this is to just set a single delay tap to a particular pitch value (you can use Appendix D to find one you like). Unfortunately, you can only use one tap for this effect to work – you can’t make a resonant “chord”.



## The Definitive Guide to Evolver

Either activate your external input and run something through it (like a drum loop or a speaking voice) or set up a little noise patch on your Evolver to “feed” the resonator.

- Turn Delay Feedback 1 up high (60-100) to get this effect to work, but not so high that it starts “ringing”
- If you’re at 100 for Delay Feedback, make sure the Delay Level is less than 100.
- Set Delay 1 Time to 35. The “pitch” of your resonator is controlled by Delay 1 Time. Remember how we noted earlier that the steps between 22 and 94 were pitched? Well, whichever step you select determines the pitch of your resonator.
- You can use the delay level controls to adjust the amount of “Resonator” sound that you hear. Anything between 50 and 90 works well.

### **MIDI Control**

You can even use MIDI notes to determine your resonator’s pitch. To do this and have the resonator pitch track your keyboard chromatically, you need to use 2 modulation slots (using just 1 slot will give you quarter tones). Set both modulation slots as follows:

- Source: MIDI note number
- Destination: Delay time
- Amount: +99

### **Why This Works**

Remember much earlier when we were discussing periodicity? A delay is really just a periodically repeating chunk of sound...just like an oscillator is a periodically repeating chunk of sound.

Since Evolver’s delay can be set to time periods which correspond to musical frequencies, and those times can “track” keyboard pitch changes, it is possible to create a delay which can be “played” as though it were an oscillator.

## **Master class: Karplus-Strong synthesis**

The Evolver manual has a brief and cryptic note about “Karplus-Strong synthesis” in the manual.

Karplus-Strong synthesis (named for its 2 “inventors”) is a method used to simulate plucked strings involving a simple feedback loop with a lowpass filter.

The Karplus-Strong model is functionally the same thing as the Resonator model we just made, with one key difference. In Karplus-Strong synthesis, we only want to hear the sound of the delay feeding back on itself. The sound that feeds (or “excites”) the delay is unimportant.

There are 2 ways to implement it using Evolver, each providing different timbres. One uses the digital delay, the other uses the feedback channels.

In both cases, you need to use MIDI note number to modulate either the delay time (if using the digital delay) or the feedback pitch (if using the feedback channels). As we have noted in a few exercises, this requires the use of 2 modulation slots to enable keyboard tracking of pitches.

Let’s start with the MIDI-tracking feedback resonator we made earlier:

- OSC 1, 2,3, 4 level = 0
- Feedback Level= 60
- Modulation Slot 1 Source = MIDI note number
- Modulation Slot 1 Destination = Feedback Frequency
- Modulation Slot 1 Amount = +99
- Modulation Slot 2 Source = MIDI note number
- Modulation Slot 2 Destination = Feedback Frequency
- Modulation Slot 2 Amount = +99

We’ve cranked feedback up, and have set things up so that MIDI notes determine the delay line’s pitch.

Now all we need is an “impulse” to get the delay going.

## The Definitive Guide to Evolver

That impulse is going to be a little bit of noise. Set up a very short VCA envelope for noise and open the filter up as follows:

- ENV 3 Attack = 0
- ENV 3 Decay = 17
- ENV 3 Amount = 99
- ENV 3 Destination = Noise level

All this little thing is going to do is provide some “excitement” (i.e. signal) for our delay/feedback chain.

Now play a few notes. You should be hearing a buzzy, twangy sound.

You’ve done it! You can vary the sound by adjusting filter parameters. The impulse you use has surprisingly little effect.

A little bit of 4-pole LPF will tighten things up a bit. Try it with LPF Frequency at about 64.

You can also use the digital delay instead of the feedback lines:

- OSC 1, 2,3, 4 level = 0
- Delay Level 1 = 100 (you want it loud since it’s the sound generator)
- Delay Time 1 = 94 (needed as a “base pitch”)
- Delay Feedback 1 = 79 (too much and it will ring endlessly, too little and it won’t ring enough)
- Modulation Slot 1 Source = MIDI note number
- Modulation Slot 1 Destination = Delay Time 1
- Modulation Slot 1 Amount = +99
- Modulation Slot 2 Source = MIDI note number
- Modulation Slot 2 Destination = Delay Time 1
- Modulation Slot 2 Amount = +99
- ENV 3 Attack = 0
- ENV 3 Decay = 12
- ENV 3 Amount = 99
- ENV 3 Destination = Noise level

In either configuration, you can use the other delays, but you need to keep their levels low and note they may respond in an unusual fashion. Try it and see!

## Modulation

---

While a synthesizer's basic sonic character comes from its oscillators and filters, its flexibility, power and ability to make truly interesting sounds comes from its modulation possibilities. We have tasted this power through our earlier exercises. Now we will explore it further.

Modulation is another area where Evolver truly shines. Evolver offers a wide range of modulation sources and provides a huge range of destination options. Evolver is so flexible it is arguably on par with classic "patch cable" modular synth systems.

Evolver offers the following dedicated modulation sources:

- Envelope 3 (a general purpose envelope)
- 4 identical low frequency oscillators (LFOs)
- 4 Modulation Slots for user-defined modulation routings
- External MIDI controllers
- Evolver's Sequencer (which is really four modulation sources)

Because Evolver's Sequencer is so flexible and rich, it deserves its own dedicated section, which follows after this one.

### ENV 3 – The Mod Envelope

---

Envelope 3 (ENV 3) is a general-purpose user-defined modulation source. That means ENV 3 doesn't control anything by default. Evolver gives you the ability to assign it to a number of destinations.

ENV 3 is best suited for modulation you want to happen every time a key is pressed.

ENV 3 is very similar to the Filter and VCA envelopes – it has ADSR parameters and so forth.

Unlike the Filter and VCA envelopes, Envelope 3 includes a **delay** parameter. Normally, Envelope 3 triggers simultaneously with the Filter envelope and the VCA Envelope. As the delay parameter is increased, the start time of ENV 3 occurs later. This

means the modulation effect controlled by ENV 3 is delayed.

### Practical Applications

One obvious use is to add a **delayed vibrato** to a sound:

- Give ENV 3 an attack of 25 or so
- Give ENV 3 a long decay
- Set ENV 3 delay to about 30
- Assign ENV 3 to LFO1 Amount

Now make LFO 1 provide some vibrato:

- Assign LFO 1 to OSC Pitch ALL
- Give it a rate of 50
- Give it a depth of 30

Now when you press a key, the sound has no vibrato, but if you hold it down, it will start to fade in

You can use the same approach for any modulation, or even for bringing in extra oscillators or **guitar-style feedback**:

- OSC 2 Freq = G0
- OSC 2 Shape = MIX
- OSC 3 Freq = C2
- OSC 3 Shape = 13
- OSC 3 Level = 0
- OSC 4 Freq = G1
- OSC 4 Shape = 16
- OSC 4 Level = 0
- Distortion = 80
- ENV 3 Attack = 88
- ENV 3 Sustain = 100
- ENV 3 Release = 17
- ENV 3 Amount = 99
- ENV 4 Delay = 44
- Modulation Slot 1 Source = ENV 3
- Modulation Slot 1 Dest = OSC 1 Level
- Modulation Slot 1 Amount = -32
- Modulation Slot 2 Source = ENV 3
- Modulation Slot 2 Dest = OSC 2 Level
- Modulation Slot 2 Amount = -32
- Volume = 46
- Pan = MONO

## The Definitive Guide to Evolver

Press a key and hold it down until you hear the “feedback”- the lower notes provide truer pitches but the upper ones are wonderfully nasty.

Dissecting this, you can see we’re simply setting up a big distorted 5<sup>th</sup> interval using the analog oscillators.

ENV 3 waits a bit before quickly fading in the digital oscillators higher with some nice piercing tones (my wife just begged me to turn it down!). We are then using 2 modulation slots to use the inverted signal from ENV 3 to turn down the analog oscillators.

You can tart this sound up a bit more by adding some filter magic via the filter envelope if you like.

We have already used ENV 3 in a few other exercises, so I won’t elaborate on it further.

### **Low Frequency Oscillators 1-4**

---

Evolver offers four separate (but otherwise identical) Low Frequency Oscillators (LFOs).

LFOs are special types of oscillators that produce frequencies below the range of human hearing, or 20 Hz. 20 Hertz is 20 cycles per second, which is still fairly fast.

LFOs are typically used to produce periodic modulation or changes in sound – vibrato (periodic pitch modulation) and tremolo (periodic volume modulation) are some examples.

Although they are nominally designated as “low frequency” modulation devices, many synthesizers offer LFOs with speeds that extend into the audible range. Evolver’s LFOs can go as slow/low as 1 cycle every 30 seconds (0.033 Hz!) or as fast/high as 261 Hz, which is middle C.

Evolver has 4 LFOs. They each have the same available parameters: **Frequency, Shape, Amount, and Destination**

## Frequency

Frequency is how fast the LFO cycles. The higher the value, the faster the cycling.

LFO frequency actually has 2 different sets of frequency settings. The first set runs from 0-150 and corresponds to absolute, “unsynchronized” rates – no matter what your program or MIDI tempo is, a rate of 50 sounds the same every time.

Turning the value past 150, Evolver displays a set of synchronized LFO frequencies. When you want your LFO modulation to “beat match” or sync to the tempo of your music, these are the values you want.

The actual LFO cycle rate depends on the program clock and/or MIDI clock data as follows:

| Display | LFO cycles...             | Speed   |
|---------|---------------------------|---------|
| S32     | Once per 2 sequences      | Slowest |
| S16     | Once per sequence         |         |
| S8      | Over 8 steps (2x per seq) | Slower  |
| S4      | Over 4 steps (4x per seq) |         |
| S2      | Over 2 steps (8x per seq) |         |
| S1      | With each step            | Matched |
| St2     | 2 times per step          | Faster  |
| St4     | 4 times per step          |         |
| St8     | 8 times per step          |         |
| S16     | 16 times per step         |         |

## Shape

Shape determines how the values output by the LFO change. Evolver offers 4 LFO shapes:

- **Triangle** – provides continuous changes, rising and falling equally around a center value. Best for smooth, continuous modulation.
- **Sawtooth** – rising up to a peak and then immediately dropping to a minimum value before starting to rise again
- **Reverse Sawtooth** – Declines quickly from maximum to minimum, then immediately jumps to maximum and repeats
- **Square** – Alternates between 0 and maximum (note this is the only LFO shape that does not go below 0)

- **Random** – abruptly jumps to a new value each cycle

### **Amount**

Amount determines how much the LFO varies from 0.

For all waves except Square, this amount includes positive and negative deflection. This means if you use a triangle wave and set the amount to 15, the LFO will output values from -15 to +15 during each cycle.

A square wave LFO with an amount of 15 would alternate between output values of 0 and 15, changing each cycle.

### **Destination**

Destination simply determines what the LFO is modulating. We have already used LFOs to modulate things like OSC pulse width and delay time – the destination parameter is how we got the LFOs to control them.

## **Modulation Slots 1-4**

---

The modulation slots are one of Evolver's most powerful features, and are also one of the simplest to understand.

The modulation slots are like virtual patch cables – they let you connect any modulation source to any modulation destination. They also let you control how much the source affects the destination.

Each slot has the same 3 parameters:

### **Source**

Modulation Slot Source determines where your modulation comes from – there are 24 available sources, including sequences, LFOs, Envelopes, External audio, MIDI controllers, Noise, and Oscillators 3 and 4



### Destination

Modulation Slot Destination determines where your modulation is going – Evolver offers a whopping 74 destinations, including various and grouped parameters for Oscillators, Filters, VCA, Delay, LFOs, Envelopes, Distortion, and Sequences

### Amount

Modulation Slot Amount specifies how much the source affects the destination. This parameter ranges from -99 to +99, which means you can “invert” the modulation source.

### Practical Applications

Evolver’s modulation slots are like the icing on a cake or the spices in food – they can be viewed as simple “goodies” but are, to connoisseurs, the critical element that separates the “good” from the “great”.

Because modulation slots are so conceptually simple, rather than offer a set of exercises I am just going to provide a few suggestions:

- **Map LFOs to additional locations** – Each LFO has, by default, a single “destination”. You can use modulation slots to assign a single LFO to multiple destinations.
- **Reverse Modulation Polarity** – The modulation slots’ “Amount” parameter is bi-directional, so the modulation slots can also be used to “invert” a modulation if so desired.
- **High Frequency LFOs** – It may seem like an oxymoron, but there may be times you want to use an LFO with a very high frequency. Evolver’s LFOs “top out” at 261 Hz. However, you could use Oscillator 3 or 4, assign it to a fixed pitch, and use it as a “high frequency” LFO source mapped to whatever destination you like.
- **Be Subtle** – Use modulation slots to add tiny amounts of LFO variation to parameters like pitch, filter split, feedback, FM, or pan to give your sounds some motion and “evolution”. This makes them more interesting

## The Definitive Guide to Evolver

- **Be Radical** – Use modulation slots to add extreme amounts of variation to any parameter. As Dave Smith is fond of saying in the manual, “things can get pretty wild”.

If you have a sound that doesn't seem unique or interesting enough, try using a few modulation slots. Point them at FM level or Ring Mod and mix the results in a little bit. Route a few more parameters around.

I use the modulation slots for almost every sound I create. Though simple, I find them to be Evolver's most useful and unique feature.

## The Sequencer

---

### What is it?

---

In the last few years, the term “sequencer” has come to mean “fancy MIDI recording and editing”, such as you might find in Cubase, Logic, or any one of a number of programs.

For musicians familiar with “groove boxes” or drum machines, “sequencer” usually means something that lets you create a series of short patterns and then string them together to create a song.

But back in days of yore, a sequencer was a box with a bunch of knobs in a row that sent out CV (control voltage) values one after another. This is what old-school synth people think of when they talk about a sequencer, and that is how Evolver’s sequencer works.



Moog 960 Sequencer

Evolver’s sequencer is not going to let you record or edit MIDI data in any extensive fashion – though it can, in a very limited way, behave like this. And Evolver’s sequencer cannot be used to create songs from individual sequences, though there is a way to “fake it” during live playback.

So what is Evolver’s sequencer? And what use is it?

Evolver’s sequencer is really a table of values, which is read, um “sequentially” (hence the name) at a rate determined by the sequence tempo and the clock divide setting.

































































The table’s values can be assigned to control a variety of Evolver’s parameters. The sequencer’s rhythmic, repeating changes are another key component in Evolver’s unique character.

Back in the old days, it was much more difficult to insert rests or vary clock speed and note length, which is why so many synthesizer recordings from the 60s-80s have sequences consisting of nothing but endlessly repeating 8<sup>th</sup> notes or 16<sup>th</sup> notes.

It was too hard to do anything else!

## The Definitive Guide to Evolver

Dave Smith describes the sequencer as “4 rows of 16 knobs”. This is a helpful way to visualize what is going on, so I’m actually going to draw you a picture!

|            | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  | 13  | 14  | 15  | 16  |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sequence 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|            | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  |
| Sequence 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|            | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  |
| Sequence 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|            | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  |
| Sequence 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|            | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  | 00  |

You can see that there are 4 rows of knobs. Each row is a sequence. Each sequence has 16 steps. Each column is a step. They are currently all set to a value of “00”.

Programming the sequencer consists of setting each one of these virtual “knobs” to a particular value.

When the sequencer is started, Evolver...

- Reads the values in the first column for all 4 sequences
- Sends those values to the destinations assigned to each sequence
- Triggers the envelopes (like pressing a key)

...it then advances to the next column/step for each sequence and does the same thing, and repeats this process until it reaches the end of the sequence, at which point it restarts.

Evolver uses the **sequence tempo** and **clock divide** to decide how fast to move from one column to the next.

**Tempo** is set in beats per minute (BPM) and determines the base playback speed. You can set the tempo to be between 30 and 250 BPM for each program using the BPM parameter.

**Clock divide** determines what kind of musical subdivision is played:

- half-notes
- quarter notes

It triggers the envelopes as long as you haven’t told it not to. For this section, please make sure the TRIGGER parameter is set to “ALL”

## The Definitive Guide to Evolver

- 8<sup>th</sup> notes – can be half-swing, swing, or triplet
- 16<sup>th</sup> notes – can be half-swing, swing, or triplet
- 32<sup>nd</sup> notes – can also be triplets
- 64<sup>th</sup> note triplets

No matter what you set things to, Evolver cannot play faster than 250 notes per minute or slower than 30 notes per minute

When Evolver triggers a note, it “holds the key down” for a time period equal to half of the step length. If the notes seem to be too long or too short for your taste, tweak the VCA envelope (ENV 2) until it sounds right.

### How do you get to the sequencer?

- Press the “Sequencer” button
- Select a row by pressing one of the selector buttons – the first one selects the first 8 positions for sequence 1, the second button selects the second 8 positions for sequence 1, and so on.

## Parameters

---

Each sequencer step ranges from 0-100 to rST (102 possible values). The steps for Sequence 1 go beyond rST and include “oFF” – a total of 103 possible values. Here’s what those values mean:

### 0-100:

This is pretty straightforward. It is the “value” for that knob or step. This value is sent to the destination you set for that sequence. So if you had sequence 1 set to Filter Cutoff, and step 1 was set to a value of “00”, the sequencer would send the value of “00” to the Filter Cutoff, and damp the sound way down. If step 2 of sequence 1 was set to “100”, it would open the filter all the way back up.

### rST:

This is an abbreviation for “Reset”. This is basically a marker that says “start the sequence over.” It is essential if you do not want your sequence to run the full 16 steps.

## The Definitive Guide to Evolver

Each sequence only pays attention to one “rST” set for it: the first one.

What isn't quite so obvious is what happens if the sequences are different lengths. Evolver is “monophonic”, and as we've noted, that means it can only fire a single “note” at a time.

With the sequencer, it just means the loops start “shifting” around a bit

So if you have a sequence that looks like this:

|        | 1  | 2  | 3  | 4   | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16  |
|--------|----|----|----|-----|----|----|----|----|----|----|----|----|----|----|----|-----|
| Seq. 1 | 52 | 42 | 46 | 36  | 38 | 28 | 38 | 42 | 52 | 42 | 46 | 36 | 38 | 28 | 38 | rST |
| Seq. 2 | 24 | 42 | 52 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00  |
| Seq. 3 | 00 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00  |
| Seq. 4 | 00 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00  |

...it's going to play like it was programmed like this:

|        | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16  |
|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|
| Seq. 1 | 52 | 42 | 46 | 36 | 38 | 28 | 38 | 42 | 52 | 42 | 46 | 36 | 38 | 28 | 38 | rST |
| Seq. 2 | 24 | 42 | 52 | 24 | 42 | 52 | 24 | 42 | 52 | 24 | 42 | 52 | 24 | 42 | 52 | rST |
| Seq. 3 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00  |
| Seq. 4 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00  |

(I removed the “knobs” to make it easier to read):

This can get very cool if you use different lengths for each of your sequences. We'll come back to this a little later.

### oFF:

This can only be used on Sequence 1 (the value isn't available for the other sequences). It sets that step to be silent, meaning Evolver just “skips” that step -- the envelopes do not fire, and values are not transmitted. This is useful for creating rhythmic patterns that are not constant streams of notes.

For the rest of this guide, I am going to refer to an Evolver silent step as an “oFF”. Since Evolver uses the abbreviation “rST” for “reset”, it is easy to confuse it with “rest”. They're very different!

Note that Sequence 1 is the “master” for oFFs. Whenever they occur in Sequence 1, they'll occur in all

## The Definitive Guide to Evolver

the other sequences. This has some interesting not-so-obvious effects, too. Let's look back at our demo sequence:

|        | 1         | 2         | 3         | 4          | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16         |
|--------|-----------|-----------|-----------|------------|----|----|----|----|----|----|----|----|----|----|----|------------|
| Seq. 1 | 52        | 42        | 46        | 36         | 38 | 28 | 38 | 42 | 52 | 42 | 46 | 36 | 38 | 28 | 38 | <b>rST</b> |
| Seq. 2 | <b>24</b> | <b>42</b> | <b>52</b> | <b>rST</b> | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00         |
| Seq. 3 | 00        | 00        | 00        | 00         | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00         |
| Seq. 4 | 00        | 00        | 00        | 00         | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00         |

Now let's add some oFFs to Sequence 1:

|        | 1         | 2         | 3         | 4          | 5  | 6   | 7  | 8  | 9  | 10  | 11 | 12 | 13  | 14 | 15 | 16         |
|--------|-----------|-----------|-----------|------------|----|-----|----|----|----|-----|----|----|-----|----|----|------------|
| Seq. 1 | 52        | 42        | 46        | oFF        | 38 | oFF | 38 | 42 | 52 | oFF | 46 | 36 | oFF | 28 | 38 | <b>rST</b> |
| Seq. 2 | <b>24</b> | <b>42</b> | <b>52</b> | <b>rST</b> | 00 | 00  | 00 | 00 | 00 | 00  | 00 | 00 | 00  | 00 | 00 | 00         |
| Seq. 3 | 00        | 00        | 00        | 00         | 00 | 00  | 00 | 00 | 00 | 00  | 00 | 00 | 00  | 00 | 00 | 00         |
| Seq. 4 | 00        | 00        | 00        | 00         | 00 | 00  | 00 | 00 | 00 | 00  | 00 | 00 | 00  | 00 | 00 | 00         |

How will it play back? Like this:

|        | 1         | 2         | 3         | 4   | 5         | 6   | 7         | 8         | 9         | 10  | 11        | 12        | 13  | 14        | 15        | 16         |
|--------|-----------|-----------|-----------|-----|-----------|-----|-----------|-----------|-----------|-----|-----------|-----------|-----|-----------|-----------|------------|
| Seq. 1 | 50        | 66        | 77        | oFF | 99        | oFF | 111       | 44        | 33        | oFF | 11        | 79        | oFF | 22        | 50        | <b>rST</b> |
| Seq. 2 | <b>24</b> | <b>42</b> | <b>52</b> | oFF | <b>42</b> | oFF | <b>24</b> | <b>42</b> | <b>52</b> | oFF | <b>42</b> | <b>52</b> | oFF | <b>42</b> | <b>52</b> | <b>rST</b> |
| Seq. 3 | 00        | 00        | 00        | 00  | 00        | 00  | 00        | 00        | 00        | 00  | 00        | 00        | 00  | 00        | 00        | 00         |
| Seq. 4 | 00        | 00        | 00        | 00  | 00        | 00  | 00        | 00        | 00        | 00  | 00        | 00        | 00  | 00        | 00        | 00         |

See how the oFFs are falling? They are in different places in Sequence 2 (due to Sequence 2's short length), but always the same relative to Sequence 1.

## Using Evolver's Sequencer

Evolver's sequencer has 3 main types of uses:

1. Traditional "Step Sequencer"
2. "Motif/Arpeggiator" live playback function
3. Modulation source for your sounds

Intrigued? Good! Let's take a more in-depth look at each of these uses.

## Traditional “Step Sequencer”

Even though it is only 16 steps long, Evolver’s sequencer can be used as a traditional “step sequencer” which triggers specific notes in a predetermined pattern.

For Evolver to function as a step sequencer, we’re going to have to assign at least one sequence to control the pitch of at least one oscillator.

Before we get started, we need to check a few global parameters.

MIDI TRIGGER should be set to “ALL”  
MIDI/KEY TRANSPOSE should be set to “off”

### Exercise 1: A basic sequence

Bring up our “basic patch”.

- Sequence 1 Destination = Oscillator Pitch All
- Make sure the other sequences are not assigned to any destinations

Enter the following sequence values:

|    | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | Destination   |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------|
| S1 | 52 | 42 | 46 | 36 | 38 | 28 | 38 | 42 | 52 | 42 | 46 | 36 | 38 | 28 | 38 | 42 | OSC ALL PITCH |
| S2 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -             |
| S3 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -             |
| S4 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -             |

You can also enter notes using an attached MIDI keyboard. This is very handy if you know the notes you want to play. Evolver won’t “record” notes C2 or lower – it will insert oFFs instead.

Be careful, because in this mode it will “wrap around” after the 16<sup>th</sup> step and start back on step 1, “erasing” what you have previously played. You cannot enter “rST” (reset) from a MIDI keyboard.

Once you have the values in there, go ahead and press “Start/Stop”. Sit back for a bit and listen to your first sequence!

OK, that’s probably starting to drive you nuts. Press “Start/Stop” again to stop the playback.



### Exercise 1A: Different Length

Let's try changing the length of our sequence.

Go to step 13 and set it to "rST" (remember, this means "RESET"). Press "Start/Stop" and notice how the sequence restarts in a different place.

Try putting your "rST" step in different places until you are confident you understand how it affects your sequence playback.

### Exercise 2: Faking Polyphony

It's time to get a little fancier. We're now going to make our sequence "polyphonic". I put this in quotes because we're sort of faking it – Evolver can only play one envelope at a time, but using the sequencer, we can actually play chords!

Remember that modulation sources are like invisible hands, and when routed to destinations, it's like having those invisible hands turning those knobs the way you want. The sequencer is really just another modulation source, and we're going to use the hands to vary the different oscillator pitches for us, as though before every key press we were reconfiguring the oscillators to make a new "1-finger chord".

Unlike the oscillator frequency parameters and the MIDI tune numbers, the sequencer allows you to enter "quarter tones". This is pretty cool, as not a lot of hardware synths support any sort of alternative tunings. For now it means that 2 sequencer steps are required to change the pitch one half-step.

This is really important, so I'm going to say it again: **The sequencer's oscillator Frequency values are in quarter-tone increments. So if you want step 2 to be an octave higher than step 1, you need to add 24 to step 1's value, not 12.**

Appendix A has a table of MIDI note values converted to Evolver Sequencer oscillator values, assuming you have MIDI Transpose set to +24.

## The Definitive Guide to Evolver

Bring back our basic patch, and enter the following sequence:

|    | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9   | 10 | 11 | 12 | 13 | 14 | 15 | 16 | Destination |
|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|-------------|
| S1 | 52 | 42 | 46 | 36 | 38 | 28 | 38 | 42 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | OSC 1 PITCH |
| S2 | 60 | 56 | 52 | 50 | 52 | 42 | 52 | 56 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | OSC 2 PITCH |
| S3 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -           |
| S4 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -           |

- Set OSC 3 and 4 Level = 0
- Sequence 1 Destination = OSC 1 Pitch
- Sequence 2 Destination = OSC 2 Pitch

After you've entered all the values and set the sequencer destinations appropriately, hit play. How about that!

The key here is that each sequence is assigned to a different oscillator. This means on each step, the sequencer is sending out values to modify the oscillator frequency before pressing the key for you.

It is just like you pressed a key, played a note, let the key go, edited the frequency for each of the oscillators, and then pressed a key again. Except it's much faster, and doesn't make mistakes.

### Exercise 3: oFFs

So far, our sequences have been unending, relentless streams of notes. If you're doing techno or you're Giorgio Moroder, that's probably fine. The rest of the world will want to know how to insert some silence.

Bring up our basic patch and enter our favorite sequence:

|    | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | Destination   |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------|
| S1 | 52 | 42 | 46 | 36 | 38 | 28 | 38 | 42 | 52 | 42 | 46 | 36 | 38 | 28 | 38 | 42 | OSC ALL PITCH |
| S2 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -             |
| S3 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -             |
| S4 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -             |

Now go set steps 4, 9, and 14 to "oFF":

|    | 1  | 2  | 3  | 4   | 5  | 6  | 7  | 8  | 9   | 10 | 11 | 12  | 13 | 14  | 15 | 16 | Destination   |
|----|----|----|----|-----|----|----|----|----|-----|----|----|-----|----|-----|----|----|---------------|
| S1 | 52 | 42 | 46 | oFF | 38 | 28 | 38 | 42 | oFF | 42 | 46 | oFF | 38 | oFF | 38 | 42 | OSC ALL PITCH |
| S2 | 00 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00  | 00 | 00 | 00  | 00 | 00  | 00 | 00 | -             |
| S3 | 00 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00  | 00 | 00 | 00  | 00 | 00  | 00 | 00 | -             |
| S4 | 00 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00  | 00 | 00 | 00  | 00 | 00  | 00 | 00 | -             |

## The Definitive Guide to Evolver

Listen to the sequence and note what happens – Evolver doesn't play on those beats. Also, Evolver will not transmit values if you have them aimed at other destinations.

Play around with putting oFFs in various places until you are confident you understand how it works.

### Exercise 3A: Faking Independent oFFs

*“But I want to put rests in places aside from where Sequence 1 plays them”*

The good news? As usual, Evolver's flexibility provides a way to do it. The bad news is that it requires you to sacrifice another sequence to make it work.

Assume you have a two-oscillator sequence set up.

Sequence 1 is assigned to oscillator 1 pitch and

Sequence 2 is assigned to oscillator 2 pitch:

|      | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9   | 10 | 11 | 12 | 13 | 14 | 15 | 16 | Destination |
|------|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|-------------|
| Seq1 | 52 | 42 | 46 | 36 | 38 | 28 | 38 | 42 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | OSC 1 PITCH |
| Seq2 | 60 | 56 | 52 | 50 | 52 | 42 | 52 | 56 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | OSC 2 PITCH |
| Seq3 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -           |
| Seq4 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -           |

### More oFFs than just Sequence 1...

The simplest way is to dedicate a sequence to oscillator 2's volume. Set the volume to zero on the steps you don't want it to play. Set it to something higher where you do want it to play. Note that with this method, you cannot make Osc 2 play when Sequence 1 has an oFF. Remember that a sequence 1 rest means the envelopes do not fire.

This sample sequence will demonstrate...

|      | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9   | 10 | 11 | 12 | 13 | 14 | 15 | 16 | Destination |
|------|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|-------------|
| Seq1 | 52 | 42 | 46 | 36 | 38 | 28 | 38 | 42 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | OSC 1 PITCH |
| Seq2 | 60 | 56 | 52 | 50 | 52 | 42 | 52 | 56 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | OSC 2 PITCH |
| Seq3 | 99 | 00 | 99 | 99 | 00 | 99 | 00 | 99 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | OSC 1 LEVEL |
| Seq4 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -           |

### Truly Independent oFFs

## The Definitive Guide to Evolver

So what if you did want to make oscillators 1 and 2 play at different but exclusive times? You could get around this by assigning the sequences as follows:

|      | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9   | 10 | 11 | 12 | 13 | 14 | 15 | 16 | Destination |
|------|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|-------------|
| Seq1 | 52 | 42 | 46 | 36 | 38 | 28 | 38 | 42 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | OSC 1 PITCH |
| Seq2 | 60 | 56 | 52 | 50 | 52 | 42 | 52 | 56 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | OSC 2 PITCH |
| Seq3 | 99 | 00 | 99 | 99 | 00 | 99 | 00 | 99 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | OSC 1 LEVEL |
| Seq4 | 99 | 99 | 00 | 99 | 00 | 99 | 99 | 00 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | OSC 2 LEVEL |

So here we're simply doing the same thing, only we're using sequence 3 to create our silence for sequence 1 by setting the volume to zero, and using sequence 4 to create silence for sequence 2.

### Digital oscillator oFFs

If the oscillator you want to “rest” is one of the digital oscillators (3 or 4), another option is to use one of the sequences to change the waveshape to 95 on the steps you don't want that oscillator to play. This is the best method if you're already using a sequence to change the waveshapes.

|      | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9   | 10 | 11 | 12 | 13 | 14 | 15 | 16 | Destination |
|------|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|-------------|
| Seq1 | 52 | 42 | 46 | 36 | 38 | 28 | 38 | 42 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | OSC 3PITCH  |
| Seq2 | 60 | 56 | 52 | 50 | 52 | 42 | 52 | 56 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | OSC 4 PITCH |
| Seq3 | 02 | 95 | 03 | 95 | 04 | 05 | 95 | 06 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -           |
| Seq4 | 11 | 12 | 05 | 13 | 95 | 14 | 15 | 95 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -           |

Note that you also should set:

- OSC 1 and 2 Level = 0
- OSC 3 and 4 Level = 50
- OSC 3 Shape Seq = SE3
- OSC 4 Shape Seq = SE4

### Last Resorts: LFO and Mod Wheel

And finally, you could set one of the LFOs to control the volume for the oscillator you want off. This will work best with the square wave LFO synced to tempo, and will really only work best if you want every N steps silenced. It is of limited flexibility, but you may find a case where it is useful

...And you could assign oscillator volume to the Mod wheel and ride that manually (or have MIDI software do it).

### Exercise 4: Tempo Clock Mod

Tempo Clock Mod is a special modulation destination. It allows you to get the sequencer to play notes of different lengths by messing with the tempo.

Remember there are 2 components to the speed of a sequence's playback:

- **Tempo** - how fast the sequencer ticks in beats per minute (BPM).
- **Clock Divider** – what type of note is played on each sequencer tick (quarter note, 8<sup>th</sup> note, 16<sup>th</sup> triplet, etc.)

You can read the sections on those specific parameters for more information.

I found Dave's explanation of the Tempo Clock Mod parameter in the manual to be a little confusing. The "Tempo Clock Mod" parameter modifies the overall tempo, and ranges from 0-100. That much is simple.

But the range setup is a bit unusual: Tempo Clock Mod is a ratio, and it is centered at 40. As the ratio number increases (i.e. becomes larger than 40), the tempo slows down. As the ratio number decreases (i.e. becomes smaller than 40), it speeds up.

Since this value is "centered" at 40, if you set Tempo Clock Mod to 40 for a given sequencer step, you won't hear any change in what's going on.

If you set Tempo Clock Mod to 80 (which is  $40 \times 2$ ) for a given sequencer step, the note played will be 2 times as long, since the clock is running half as fast on that step.

If you set Tempo Clock mod to 20 (which is  $40 \times \frac{1}{2}$ ) for a given sequencer step, the note played will be  $\frac{1}{2}$  as long, since the clock is running twice as fast.

This table will help explain a bit more:

| Tempo Clock Mod Value           | 10       | 20           | 30       | 40     | 60        | 80         | 100 |
|---------------------------------|----------|--------------|----------|--------|-----------|------------|-----|
| Note length is multiplied by... | .25      | .5           | .75      | 1      | 1.5       | 2          | 2.5 |
| Tempo effect                    | 4x speed | Double speed | 1½ speed | Normal | 2/3 speed | Half speed |     |

Faster ← → Slower

## The Definitive Guide to Evolver

Appendix B includes a chart where I did all the math (well, Excel did) to show exactly which settings create which values.

So let's put this into practice.

|      | 1  | 2  | 3   | 4  | 5  | 6  | 7  | 8   | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | Destination   |
|------|----|----|-----|----|----|----|----|-----|----|----|----|----|----|----|----|----|---------------|
| Seq1 | 20 | 20 | 60  | 60 | 80 | 80 | 40 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | CLO (Seq Clk) |
| Seq2 | 60 | 84 | rST | 00 | 00 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | OSC ALL PITCH |
| Seq3 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -             |
| Seq4 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -             |

In our simple sequence, we have all of our “music” events on Sequence 2, and Sequence 1 controlling Tempo Clock Mod. It is important to note that Tempo Clock Mod controls Sequence 1, and all of the other sequences follow Sequence 1's note values, much in the same way that all sequences follow Sequence 1's oFFs.

Take the demo sequence and monkey around with different Tempo Clock Mod values until you get the hang of it. The chart in Appendix B should come in handy, especially if you realize that you can convert from whatever base clock division you set with the Clock Divider parameter to any other note lengths using different Tempo Clock Mod values:

| Base                   | half note | qtr note | 8 <sup>th</sup> | 8 <sup>th</sup> trip. | 16 <sup>th</sup> | 16 <sup>th</sup> trip | 32 <sup>nd</sup> | 64 <sup>th</sup> |
|------------------------|-----------|----------|-----------------|-----------------------|------------------|-----------------------|------------------|------------------|
| Half note              | 40        | 20       | 10              |                       | 5                |                       |                  |                  |
| Qtr note               | 80        | 40       | 20              |                       | 10               |                       | 5                |                  |
| 8 <sup>th</sup>        |           | 80       | 40              |                       | 20               |                       | 10               | 5                |
| 8 <sup>th</sup> trip.  |           |          |                 | 40                    |                  | 20                    |                  |                  |
| 16 <sup>th</sup>       |           |          | 80              |                       | 40               |                       | 20               | 10               |
| 16 <sup>th</sup> trip. |           |          |                 | 80                    |                  | 40                    |                  |                  |
| 32 <sup>nd</sup>       |           |          |                 |                       | 80               |                       | 40               | 20               |
| 64 <sup>th</sup>       |           |          |                 |                       |                  |                       | 80               | 40               |

Blank means exact value not possible

## “Motif/Arpeggiator” live playback function

---

This is really just an extension of Evolver as a step sequencer, but a few clever features were added to Evolver’s design that allow it to do much more.

Most of these ideas are based around manipulating the **Envelope Trigger Select** parameter.

Envelope Trigger Select affects both the control of the sequencer and how the sequencer responds to that control.

The sequencer can be triggered by:

- The start/stop button
- A keypress on your MIDI keyboard
- A signal at the external input

When triggered, the sequencer can:

- Play through a sequence once and stop, uninterrupted
- Play through the sequence continuously, uninterrupted
- Play through the sequence once and stop, re-starting every time it is triggered
- Play through the sequence continuously, re-starting every time it is triggered
- “Gate” the sequence – the sequence will play silently, looping. When a trigger is received, it will play the notes it is currently on.
- “Gate” the sequence and restart – It will cycle silently, until a trigger is received, at which point it will re-start from the beginning of the sequence and be audible.
- Play the first step when the first trigger event is received. Play the second step when the next trigger event is received, etc. Sequence will repeat at end.

### Exercise 1: Changing Pitch

You can use your MIDI keyboard to shift all the pitches of your Evolver sequence, transposing it on-the-fly based on your keyboard inputs. When you press a key, Evolver adds that key’s value to everything in your sequence.

## The Definitive Guide to Evolver

Set up the following sequence with the basic patch:

|      | 1  | 2  | 3   | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | Destination   |
|------|----|----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------|
| Seq1 | 60 | 84 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | OSC ALL PITCH |
| Seq2 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -             |
| Seq3 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -             |
| Seq4 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -             |

By default, Evolver assumes the base key is MIDI note number 0, which is very, very low. So if you press “C3”, the sequence will transpose up 3 octaves. Since we want to be able to play up and down the keyboard with a decent range, we need to do a few things.

One is to adjust the key transpose value. Set it to -36. This has the effect of making your middle keys lower. Since Evolver starts from 0, we want our keys to be close to that, too.

Now set the MIDI trigger value to “ALL”.

Press “Start/Stop”. After the sequence cycles all the way through once, play the G above middle C. You’ll hear the whole sequence play back again, transposed up a 5<sup>th</sup>.

As the sequence plays, you can transpose it by pressing different keys. You can do this at any time during the sequence – in the middle of a note, or partway through the sequence – and Evolver will immediately respond to the change.

Transpose your sequence up, down, and all around until you either “get it” (or can’t take listening to the sequence anymore!)

Note that the sequence keeps cycling synchronized to the Evolver’s clock regardless of which keys you press or when, and it won’t stop until you press the “Start/Stop” button.

This may not seem all that exciting, but it’s all relative. Let’s move on to...



### Exercise 1A: Fake Arpeggiator

So, Evolver can repeat a sequence over and over, and transpose it up or down? Sounds like an arpeggiator!

Program in the following sequence:

|      | 1  | 2  | 3  | 4  | 5  | 6  | 7   | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | Destination   |
|------|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|----|----|---------------|
| Seq1 | 60 | 68 | 74 | 84 | 74 | 68 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | OSC ALL PITCH |
| Seq2 | 00 | 00 | 00 | 00 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -             |
| Seq3 | 00 | 00 | 00 | 00 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -             |
| Seq4 | 00 | 00 | 00 | 00 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -             |

Hit Start/Stop on the Evolver, and play some keys on your MIDI controller.

Yeah, it's not a "real arpeggiator", but it is pretty close. And if you leave out the 3rds, and just use octaves and 5ths, you don't have to worry about key signature.

Even better, you can "program" it to use whatever pattern or motif you want.

If you play in a band where they want you to play crazy 16<sup>th</sup> notes live, this may come in handy.

This sequence is one I keep saved and available in my Evolver at all times – inevitably, someone will want a bass part playing one of the patterns in here:

|      | 1  | 2  | 3  | 4  | 5   | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | Destination   |
|------|----|----|----|----|-----|----|----|----|----|----|----|----|----|----|----|----|---------------|
| Seq1 | 60 | 84 | 60 | 84 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | OSC ALL PITCH |
| Seq2 | 60 | 60 | 84 | 84 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -             |
| Seq3 | 60 | 60 | 60 | 84 | rST | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -             |
| Seq4 | 00 | 00 | 00 | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | -             |

### Exercise 2: Re-triggering a sequence

Of course, you don't have to limit yourself to octaves and 5ths, you can put in a fancier line if you want (perhaps even one that uses rST and oFF).

One thing that is kind of annoying, though, is the sequence just keeps looping. It doesn't have to. You can set it to restart every time you press a key.

Set the Trigger Select value to MIDI RESET (ndr):

Now press Start/Stop. Nothing's happening, right?  
Now hit a key.

Take your same previous sequence and play it now, taking care to try and "interrupt" it at different times. Every time you press a key, the sequence will start over.

This adds another level of flexibility to live playing – you can make your sequence "stutter" by hammering away on the key. It's sort of like re-starting a sample.

### **Exercise 3: One-shot sequences**

Re-triggering is handy, but the sequence still keeps infinitely looping once it starts. However, there is a way to set the sequencer to play back once and stop.

You have a sequence you just need to trigger once live as an emphasis in the song.

Set Trigger Select to MIDI Sequence 1 (NS1).

You hit the key once, the sequence plays, it stops. The next time you need it to play, you hit it again.

If you set it to MIDI Sequence 1 Reset (N1r), the sequence will re-start every time you press a key, transposed appropriately, play through once, and stop. I find this setting to be the most useful for this sort of thing...it's almost like playing a sample or scratching a record.

Try all the different variations on this parameter until you get a feel for it.

Remember also that you don't have to use a MIDI keyboard for this – you can use a signal at the external input or even the "Start/Stop" button.

## **Modulation Source for Evolver Patches**

Just like you use an LFO to modulate parameters, you can use Evolver's sequencer to modulate parameters. The sequencer is literally a table of values, and those values can be read out to just about any modulation destination.

Depending on whether or not you sync Evolver's clock to MIDI, you can make effects that are tightly synchronized with the beat or completely independent of it. Since the sequencer changes immediately and abruptly from one value to the next, I typically only use the sequencer as a modulation destination when I want beat-synced effects.

It is also usually a good idea to switch MIDI triggering to "MIDI trigger reset" (ndr) – this way, Evolver is just going to read out the values to their destinations without triggering the envelopes. This allows you to get some nice burbling stuff going on within a sustained note. And because it resets the sequence every time you hit a new key, every "note" will have the same pattern.

Opting for "MIDI trigger" (nId) without the reset means your "beats" may not match up with what you're playing. Depending on what you want, this could be OK.

### **Exercise 1: Rhythmic Filter**

In the previous section, we explored how to create "arpeggiator"-style effects. This time, we're going to do something similar, only instead of using the sequencer to change pitches, we're going to make it apply rhythmic effects to the lowpass filter.

Bring up the basic patch.

- LPF mode = 4-pole
- LPF Frequency = 50
- LPF Resonance = 82
- Seq 1 Destination = Filter Frequency
- Env Trig = NS (MIDI Gate)
- Clock Divide = 8n

## The Definitive Guide to Evolver

Input some random values into Sequence 1 (87, 29, 73, 51, rST). Press the “Start/Stop” button. Hit a low key, hold it down, and listen to the sequencer modulating the filter.

Try making a new sequence (2) and setting its destination to LPF resonance. Make it a different length than the frequency sequence.

Also remember to back down the resonance parameter a bit so the sequence has some room to move – your sequence values get added to existing parameter values, they don’t replace them.

Now take a slow triangle LFO (rate 2) and have it adjust LPF Filter frequency by 31. Maybe take another LFO and assign it to control Filter Split with a slow frequency (7) and small amount (6). This is starting to sound pretty cool.

Add a little synchronized delay (Delay 1 Time = S15, Delay 1 Level = 92) and a bit of distortion to taste.

Hold down some bass notes. Mmm.

### Advanced Techniques

You can use Evolver’s sequences to modulate almost any parameters. Obviously sequences are best suited when you want to issue specific values at specific times, and want the changes to be abrupt.

Use appropriate modulation sources – If you want continuous, repeating changes, use an LFO. If you want a specific series of values in a particular order, use a sequence. If you want something you can freely or manually control, use an external MIDI controller.

Remember that you can use any of the 4 modulation slots to route the sequencer to additional destinations beyond the 4 sequencer assignments. I often get some interesting variations by assigning a sequence designed for one thing to a relatively unrelated destination, or one that makes subtle changes.

## The Definitive Guide to Evolver

Here are a few ideas to get you started. After you “max out” your sequences (all 4 set up and assigned to something), use an extra modulation slot to...

- Subtly change your sound by assigning a pitch sequence to attack time
- Add some percussion: Assign a sequence to noise volume
- Make a sequence vary the rate of one or all LFOs on each step
- Bring out some more texture by making a sequence also control ring modulation for the digital oscillators

Those modulation slots are there for a reason! Use ‘em!

### Using Evolver’s Sequencer to Control Other MIDI Devices

---

There are two things that are important to note if you want to use Evolver’s sequencer to control other MIDI devices, like a synth or sampler.

The first is MIDI notes Evolver sends out when you assign MIDI NOTE OUT (MnO) as the destination. Evolver sends the step value **plus one**. So even though Evolver’s sequencer data ranges from 0-100, the notes sent range from 1-101. This is because MIDI note values start at 1 - a MIDI note of “0” corresponds to “note off”.

The second is that Evolver is actually sending MIDI note numbers based on the step value. This may not be a big surprise, but remember that normally when you use an Evolver sequence to drive Evolver’s internal oscillators, each step value represents a quarter-tone, so you have to program step values in increments of 2 for each half step.

This means if you just take a sequence that was driving Evolver’s oscillators and send it out via MIDI, it isn’t going to sound the same.

Also, you can only send velocity outputs using Sequence 2, and that’s only if you have Sequence 1 set to send MIDI note number. Same goes for Sequence 4

## **The Definitive Guide to Evolver**

– It can only send velocity when Sequence 3 is set to note number.

## **Making Sounds**

---

Good sounds change a lot. Generally, the longer the sound, the more variation required to keep it interesting.

In this guide so far, we've explored each individual facet of Evolver. Truly interesting sounds use many of these elements at once, tastefully. Just because you have 4 oscillators and 4 LFOs doesn't mean you have to use them all on every sound.

A handy way to learn some sounds is to try to replicate them on your Evolver. Bring up your favorite soft synth or program something on a hardware synth. Then try to copy the parameters on Evolver, doing a section at a time and noting how each one changes the sound. Soon you'll be adding Evolver-specific tweaks.

Take apart the factory presets in Evolver and see how they are put together as well.

Be creative. Try anything and everything, and save anything that seems the least bit promising.

## **Drum Sounds**

---

As a monophonic, non-multi-timbral synth, you'd think that Evolver was really only suited for creating single drum sounds.

That's more or less correct. You can either sample the resulting sounds and then play the samples or lay down one track for kick, one for snare, one for hi-hat, etc.

With some Herculean efforts you can get more than one sound at a time – well explore that after some individual sounds.

All drum sounds have a few things in common – They have 0 for attack, snappy, short decays, no sustain, and no release. They also have no real “pitch”, and frequently include some noise.

Making interesting drum sounds on Evolver is an art in itself, and unfortunately, one that I have yet to master.

All of the following examples start from the basic patch.

### **Kick Drums**

The classic 808 kick drum is just a sine wave with some pitch modulation. Here's a quick and dirty approximation:

- OSC 1, 2, and 4 level = 0
- OSC 3 level = 100
- OSC 3 freq = C-2
- VCA Decay = 96
- VCA Amount = 100
- ENV 3 Decay = 60
- ENV 3 Amount = 24
- ENV 3 Destination = OSC 3 Freq

Play some notes below middle C. Boomy. The key to this sound is the slight bit of pitch variation (the “fall”) being added by ENV 3.

You can “tighten up” the decay by bringing down the VCA decay. You can change the character of the “kick” by adjusting the ENV 3 parameters.

I like to use a mod slot to map ENV 3 to noise, set the amount to 1, and then clamp down the 2-pole LPF to add just a bit of “grit” to the attack.

You can add distortion to make the kick “blow up”, and dial in a tiny bit of the HPF if it's too bassy for you (as if there is such a thing!)

### **Snare Drums**

Here's how to make the worst synth snare in the world. Start with the basic patch:

- ALL OSC Level = 0
- VCA Decay = 35
- VCA Amount = 100
- Noise Volume = 100



## The Definitive Guide to Evolver

Snap! A nice awful white noise snare, as simple as can be. A snappy VCA is applied to the noise source.

We have to do better than that. Going from where we were...

- LPF Freq = 0 (close it down)
- LPF Resonance = 52
- 4-pole mode
- Filter Env Decay = 65
- Filter Env Amount = 80

Ok, now we have something Kraftwerk might have used on one of their 70s albums.

By adding the filter, we're able to shape the sound a bit more. Play with the resonance value and the decays of both envelopes.

Bring in a bit of HPF (30) to get rid of some of the low end and make it snappier, if you like.

Adjusting the Filter Envelope Amount will adjust the "damping" of the snare.

### Hi-Hats

You can use something very similar for hi-hat cymbals. Take our same horrible snare sound:

- ALL OSC Level = 0
- VCA Decay = 35
- VCA Amount = 100
- Noise Volume = 100

Change the VCA Decay to 15. Hats are nice and tight. Now turn the HPF up to about 88. Not fantastic, but a passable techno hi-hat. If you kick the VCA decay out to about 55, you can get an "open hi-hat", too.

This means you could sequence a closed and open hi-hat sequence by assigning one sequence to control VCA decay.

Rather than using noise, you could also dial up some high-pitched digital oscillators and use some aggressive FM and/or Ring Mod to produce some metallic clanging instead. Give it a try. Don't forget

to clear ENV 3's "OSC Freq" assignment, or your OSC hats won't sound right.

### **Simmons Toms**

A distinctive 80s sound. Love it or hate it, it's easy:

- OSC 3 Freq = C-2
- OSC 3 Level = 100
- VCA Decay = 68
- VCA Amount = 100
- ENV 3 Decay = 83
- ENV 3 Amount = 51
- Mod Slot Source = ENV 3
- Mod Slot Destination = Noise Level
- Mod Slot Amount = 24
- Pan = MONO

The VCA is providing the overall drum "shape" – shorter decay values will give a tighter drum, longer values a looser, longer drum.

ENV 3 is providing the tom "pitch". Increase the ENV 3 amount to make the "fall" more pronounced, but observe that it gets pretty cartoony at higher values. Lower values will give you more of a flat kick drum-type sound.

ENV 3 is also adding in some noise (via modulation slot) which is a key element of the Simmons Tom sound. Use the Mod Slot Amount to adjust how much noise gets added. Me, I like a lot of noise in my Simmons Toms.

If you want to make this sound fancier, you can do some stuff with the Filter and Filter envelope to shape the noise a bit. Since you're working with a sine wave for the pitch stuff, the filter won't do much to it.

### **Drum Beats**

---

Well, we also have the sequencer, which can modify a whole bunch of parameters for us. This lets us get some unique drum beats by using a single patch and some programming.

Try taking the "Kraftwerk Snare" sound as a patch.

## The Definitive Guide to Evolver

Assign your 4 sequences as follows:

- Seq 1 = HPF
- Seq 2 = Filter Resonance
- Seq 3 = Resonance
- Seq 4 = Env All Decay

Seq 1 determines basically whether you have a hat or not – for hat steps, set the value (controlling the HPF) to something high, like 90-100.

Seq 2 helps define your drum. Use a value of 0 for where you want a kick. Use 30-40 for “snares”. Use 90-100 for hats.

Seq 3 also helps define the drum by adding resonance. Use nothing (0) for kicks, use middle values (61) for snares, and use high values (90+) for hats.

Seq 4 adjusts the decays of the drums. Kicks and snares should be long, hats should be short.

Sometimes turning the VCA Envelope Level up a bit helps things sound better.

By adjusting values on each step, you can generate some beats. In my experience, it's pretty tough to get both good hats and good kicks at the same time. You might have to do multiple passes.

Some of the Evolver User Groups on the web have much better drum loops you can dissect for tips than my horrible attempts.

## Using Evolver Effectively (As an Effects Unit)

---

The easiest way to understand Evolver as an external effects unit is to simply imagine using any non-Evolver audio source as an oscillator.

So now you can plug any audio source – a guitar, a vocal, even an entire mix – into Evolver and process it using any or all of Evolver's features:

- Lowpass filter
- Highpass filter
- VCA
- Pan

The ability to process external audio was a hallmark of many great analog synths.

For various reasons, this feature fell out of favor in the 80s and has only recently begun to return in synthesizer design.

- Delay
- Distortion
- Output/Input Hack

You also get a few special new parameters to play with:

- Envelope Follower
- Peak Hold

And you can use Evolver's internal oscillators in addition to your external audio source.

### External Input Mode

---

There are 4 settings:

- ST – Stereo. Uses both inputs
- L – Left. Uses left input
- R – Right. Uses right input
- SPL – Split. Uses left input for audio signals and right input for control signals

### External Input Volume

---

This parameter is a simple gain for the external input.

Use it to achieve the desired level for your source – you can overdrive this input with high levels if you want to add distortion, however, I'd recommend using the "distortion" effect instead.

Also, when adjusting this parameter, the LEDs on the Evolver will act as a sort of "level meter". The "Row" LEDs on the left are for the left channel. The top LEDs are for the right channel.

### Input Hack

---

Input Hack is just like Output Hack ("trashes the signal, quite rudely") only it is applied to the input signal.

It also sounds like bit reduction, and ranges from 0-14.

### Processing Audio

---

Any of the tricks we've learned so far can be applied to audio signals at Evolver's input. Try taking a drum

loop and feeding it to Evolver. Bring up the Ext In Volume and run it through the LPF.

Then try it with the sequenced LPF we did earlier. If you're using a MIDI sequencer to control everything, you can get the LPF pattern to sync up with the drums.

### **External Modulation**

---

You can also use your external signals as modulation sources. Take the same drum loop.

Locate the "In Peak Amt" parameter (Input Peak Amount). Assign its destination to Noise. Now the noise signal will "track" the drum loop.

Once you get the hang of that, zero it out and try using Env Fol Amt as the source and Noise as the destination again.

Evolver has a whole new personality as an effects unit. I have barely scratched the surface myself, as I've been so busy learning how to use its synth side.

### **Guitar Amp Simulator**

---

Plug a guitar or bass in to one of Evolver's inputs. Crank up the input gain and the Ext In Vol so you can hear it.

Now try putting some distortion on it and bring down the LPF a bit. Try adding some resonance. You can get a nice rude guitar sound going. It's not going to displace a real amp or Pod or anything, but I guarantee you will find some inspiring guitar and bass tones.

Especially when you start using Peak and Env Follow as modulation sources while you're playing.

## **Conclusion**

---

Well, that about wraps it up. You should now have a better understanding of how to use your Dave Smith Instruments Evolver, some new patches, and some new ideas.

I hope you had as much fun using this guide as I had writing it.

## Appendix A: MIDI notes to Sequencer values table

---

This chart assumes your oscillators are set to +24.

Note that odd numbers always produce quarter-tones.

| Seq. | MIDI  | Seq. | MIDI |
|------|-------|------|------|
| 0    | C -1  | 52   | D1   |
| 2    | C# -1 | 54   | D#1  |
| 4    | D -1  | 56   | E1   |
| 6    | D# -1 | 58   | F1   |
| 8    | E -1  | 60   | F#1  |
| 10   | F -1  | 62   | G1   |
| 12   | F# -1 | 64   | G#1  |
| 14   | G -1  | 66   | A1   |
| 16   | G# -1 | 68   | A#1  |
| 18   | A -1  | 70   | B1   |
| 20   | A# -1 | 72   | C2   |
| 22   | B -1  | 74   | C#2  |
| 24   | C0    | 76   | D2   |
| 26   | C#0   | 78   | D#2  |
| 28   | D0    | 80   | E2   |
| 30   | D#0   | 82   | F2   |
| 32   | E0    | 84   | F#2  |
| 34   | F0    | 86   | G2   |
| 36   | F#0   | 88   | G#2  |
| 38   | G0    | 90   | A2   |
| 40   | G#0   | 92   | A#2  |
| 42   | A0    | 94   | B2   |
| 44   | A#0   | 96   | C3   |
| 46   | B0    | 98   | C#3  |
| 48   | C1    | 100  | D3   |
| 50   | C#1   |      |      |

## Appendix B: Tempo Clock Multiplier Note Lengths

| Clock Mod Value | Note Length Multiplier | Tempo Effect | Clock Mod Value | Note Length Multiplier | Tempo Effect | Clock Mod Value | Note Length Multiplier | Tempo Effect |
|-----------------|------------------------|--------------|-----------------|------------------------|--------------|-----------------|------------------------|--------------|
| 0               | 0.000                  | -            | 35              | 0.875                  | 114%         | 70              | 1.750                  | 57%          |
| 1               | 0.025                  | 4000%        | 36              | 0.900                  | 111%         | 71              | 1.775                  | 56%          |
| 2               | 0.050                  | 2000%        | 37              | 0.925                  | 108%         | 72              | 1.800                  | 56%          |
| 3               | 0.075                  | 1333%        | 38              | 0.950                  | 105%         | 73              | 1.825                  | 55%          |
| 4               | 0.100                  | 1000%        | 39              | 0.975                  | 103%         | 74              | 1.850                  | 54%          |
| <b>5</b>        | <b>0.125</b>           | <b>800%</b>  | <b>40</b>       | <b>1.000</b>           | <b>100%</b>  | 75              | 1.875                  | 53%          |
| 6               | 0.150                  | 667%         | 41              | 1.025                  | 98%          | 76              | 1.900                  | 53%          |
| 7               | 0.175                  | 571%         | 42              | 1.050                  | 95%          | 77              | 1.925                  | 52%          |
| 8               | 0.200                  | 500%         | 43              | 1.075                  | 93%          | 78              | 1.950                  | 51%          |
| 9               | 0.225                  | 444%         | 44              | 1.100                  | 91%          | 79              | 1.975                  | 51%          |
| <b>10</b>       | <b>0.250</b>           | <b>400%</b>  | 45              | 1.125                  | 89%          | <b>80</b>       | <b>2.000</b>           | <b>50%</b>   |
| 11              | 0.275                  | 364%         | 46              | 1.150                  | 87%          | 81              | 2.025                  | 49%          |
| 12              | 0.300                  | 333%         | 47              | 1.175                  | 85%          | 82              | 2.050                  | 48.8%        |
| 13              | 0.325                  | 308%         | 48              | 1.200                  | 83%          | 83              | 2.075                  | 48.2%        |
| 14              | 0.350                  | 286%         | 49              | 1.225                  | 82%          | 84              | 2.100                  | 47.6%        |
| 15              | 0.375                  | 267%         | 50              | 1.250                  | 80%          | 85              | 2.125                  | 47.1%        |
| 16              | 0.400                  | 250%         | 51              | 1.275                  | 78%          | 86              | 2.150                  | 46.5%        |
| 17              | 0.425                  | 235%         | 52              | 1.300                  | 77%          | 87              | 2.175                  | 46.0%        |
| 18              | 0.450                  | 222%         | 53              | 1.325                  | 75%          | 88              | 2.200                  | 45.5%        |
| 19              | 0.475                  | 211%         | 54              | 1.350                  | 74%          | 89              | 2.225                  | 44.9%        |
| <b>20</b>       | <b>0.500</b>           | <b>200%</b>  | 55              | 1.375                  | 73%          | 90              | 2.250                  | 44.4%        |
| 21              | 0.525                  | 190%         | 56              | 1.400                  | 71%          | 91              | 2.275                  | 44.0%        |
| 22              | 0.550                  | 182%         | 57              | 1.425                  | 70%          | 92              | 2.300                  | 43.5%        |
| 23              | 0.575                  | 174%         | 58              | 1.450                  | 69%          | 93              | 2.325                  | 43.0%        |
| 24              | 0.600                  | 167%         | 59              | 1.475                  | 68%          | 94              | 2.350                  | 42.6%        |
| 25              | 0.625                  | 160%         | 60              | 1.500                  | 67%          | 95              | 2.375                  | 42.1%        |
| 26              | 0.650                  | 154%         | 61              | 1.525                  | 66%          | 96              | 2.400                  | 41.7%        |
| 27              | 0.675                  | 148%         | 62              | 1.550                  | 65%          | 97              | 2.425                  | 41.2%        |
| 28              | 0.700                  | 143%         | 63              | 1.575                  | 63%          | 98              | 2.450                  | 40.8%        |
| 29              | 0.725                  | 138%         | 64              | 1.600                  | 63%          | 99              | 2.475                  | 40.4%        |
| 30              | 0.750                  | 133%         | 65              | 1.625                  | 62%          | <b>100</b>      | <b>2.500</b>           | <b>40.0%</b> |
| 31              | 0.775                  | 129%         | 66              | 1.650                  | 61%          |                 |                        |              |
| 32              | 0.800                  | 125%         | 67              | 1.675                  | 60%          |                 |                        |              |
| 33              | 0.825                  | 121%         | 68              | 1.700                  | 59%          |                 |                        |              |
| 34              | 0.850                  | 118%         | 69              | 1.725                  | 58%          |                 |                        |              |

Note: These are “theoretical” values – Evolver’s tempo can never go slower than 30 BPM or faster than 250 BPM, and it will “clamp” at these values if either extreme is exceeded. Additionally, some tempo-synced effects may not properly track at very high tempos.



## Appendix C: Default Waveshapes

| Wave # | Prophet-VS name | 48 | Guitar   |
|--------|-----------------|----|----------|
| 01     | Sine            | 49 | Nice     |
| 02     | Saw             | 50 | WWind    |
| 03     | Sqr             | 51 | Oboe     |
| 04     | WmBell          | 52 | Harp     |
| 05     | RdBell          | 53 | Pipe     |
| 06     | R2Bell          | 54 | Hack1    |
| 07     | W2Bell          | 55 | Hack2    |
| 08     | FmtBell         | 56 | Hack3    |
| 09     | FzReed          | 57 | Pinch    |
| 10     | FmtAOh          | 58 | BellHrm  |
| 11     | FmtAhh          | 59 | BellVox  |
| 12     | TriPlus         | 60 | Hi Harm  |
| 13     | DisBel          | 61 | Hi Reed  |
| 14     | Pulse1          | 62 | BellReed |
| 15     | Pulse2          | 63 | WmWhstl  |
| 16     | SqrReed         | 64 | Wood     |
| 17     | Oohh            | 65 | Pure     |
| 18     | Eehh            | 66 | Med Pure |
| 19     | FeedBack        | 67 | HiHarm   |
| 20     | Piano1          | 68 | FullBell |
| 21     | E.Pno           | 69 | Bell     |
| 22     | M.Harm          | 70 | Pinch    |
| 23     | HiTop           | 71 | Clustr   |
| 24     | WmReed          | 72 | M. Pinch |
| 25     | 3 & 5           | 73 | VoxPnch  |
| 26     | Hollow          | 74 | OrgPnch  |
| 27     | Hvy7            | 75 | AhhPnch  |
| 28     | BelOrg          | 76 | PnoOrg   |
| 29     | BassBel         | 77 | BrReed   |
| 30     | Tine1           | 78 | NoFund   |
| 31     | PhSQR           | 79 | ReedHrm  |
| 32     | Orient          | 80 | LiteFund |
| 33     | HiPipe          | 81 | MelOrg   |
| 34     | Mass            | 82 | Bell     |
| 35     | ReedOrg         | 83 | Bell     |
| 36     | OrgAhh          | 84 | 3&5Saw   |
| 37     | MelOrg          | 85 | 5thSin   |
| 38     | FmtOrg          | 86 | Sin2Oct  |
| 39     | Clar            | 87 | Sin4Oct  |
| 40     | AhhFem          | 88 | Saw5th   |
| 41     | AhhHom          | 89 | Saw2Oct  |
| 42     | Ahhbass         | 90 | Sqr5th   |
| 43     | RegVox          | 91 | Sqr5Oct  |
| 44     | Vocal           | 92 | Sqr2Oct  |
| 45     | Homme           | 93 | WarmLo   |
| 46     | HiAhh           | 94 | Bells    |
| 47     | Bass            | 95 | (null)   |

## Appendix D: Tuned Delay Values, Notes, and Frequencies

| Delay Value | Note | Frequency (Hz) | Delay Value | Note | Frequency (Hz) |
|-------------|------|----------------|-------------|------|----------------|
| 22          | C7   | 2093           | 59          | B3   | 246.94         |
| 23          | B6   | 1975.53        | 60          | A#3  | 233.08         |
| 24          | A#6  | 1864.66        | 61          | A3   | 220            |
| 25          | A6   | 1760           | 62          | G#3  | 207.65         |
| 26          | G#6  | 1661.22        | 63          | G3   | 196            |
| 27          | G6   | 1567.98        | 64          | F#3  | 185            |
| 28          | F#6  | 1479.98        | 65          | F3   | 174.61         |
| 29          | F6   | 1396.91        | 66          | E3   | 164.81         |
| 30          | E6   | 1318.51        | 67          | D#3  | 155.56         |
| 31          | D#6  | 1244.51        | 68          | D3   | 146.83         |
| 32          | D6   | 1174.66        | 69          | C#3  | 138.59         |
| 33          | C#6  | 1108.73        | 70          | C3   | 130.81         |
| 34          | C6   | 1046.5         | 71          | B2   | 123.47         |
| 35          | B5   | 987.77         | 72          | A#2  | 116.54         |
| 36          | A#5  | 932.33         | 73          | A2   | 110            |
| 37          | A5   | 880            | 74          | G#2  | 103.83         |
| 38          | G#5  | 830.61         | 75          | G2   | 98             |
| 39          | G5   | 783.99         | 76          | F#2  | 92.5           |
| 40          | F#5  | 739.99         | 77          | F2   | 87.31          |
| 41          | F5   | 698.46         | 78          | E2   | 82.41          |
| 42          | E5   | 659.26         | 79          | D#2  | 77.78          |
| 43          | D#5  | 622.25         | 80          | D2   | 73.42          |
| 44          | D5   | 587.33         | 81          | C#2  | 69.3           |
| 45          | C#5  | 554.37         | 82          | C2   | 65.41          |
| 46          | C5   | 523.25         | 83          | B1   | 61.74          |
| 47          | B4   | 493.88         | 84          | A#1  | 58.27          |
| 48          | A#4  | 466.16         | 85          | A1   | 55             |
| 49          | A4   | 440            | 86          | G#1  | 51.91          |
| 50          | G#4  | 415.3          | 87          | G1   | 49             |
| 51          | G4   | 392            | 88          | F#1  | 46.25          |
| 52          | F#4  | 369.99         | 89          | F1   | 43.65          |
| 53          | F4   | 349.23         | 90          | E1   | 41.2           |
| 54          | E4   | 329.63         | 91          | D#1  | 38.89          |
| 55          | D#4  | 311.13         | 92          | D1   | 36.71          |
| 56          | D4   | 293.66         | 93          | C#1  | 34.65          |
| 57          | C#4  | 277.18         | 94          | C1   | 32.7           |
| 58          | C4   | 261.63         |             |      |                |

## Acknowledgements

---

**Moog 960 Sequencer image** taken from [www.bigbluewave.co.uk](http://www.bigbluewave.co.uk).

**All waveform screenshots** taken using SmartElectronix s(M)exoscope VST plug-in. Visit <http://www.smartelectronix.com>. Not super-accurate, but free, handy, and pretty.

Thanks to the Internet Evolver community for their encouragement and patience.

Special thanks to **GregAE** for his detailed edits and suggestions.

Finally, this guide would have no reason to exist without **Dave Smith**. Thanks for the tools and the inspiration.