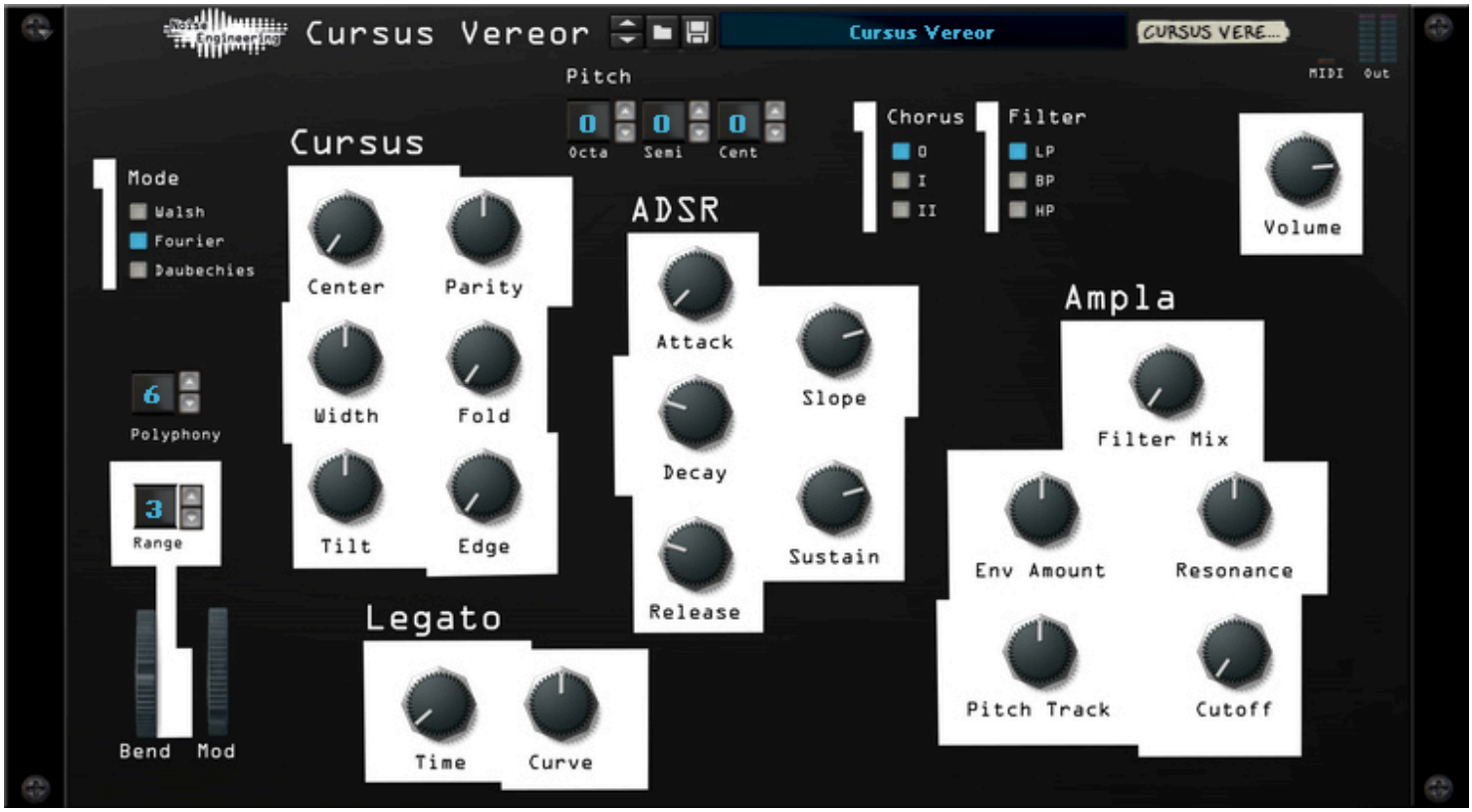
Noise Engineering

Cursus Vereor

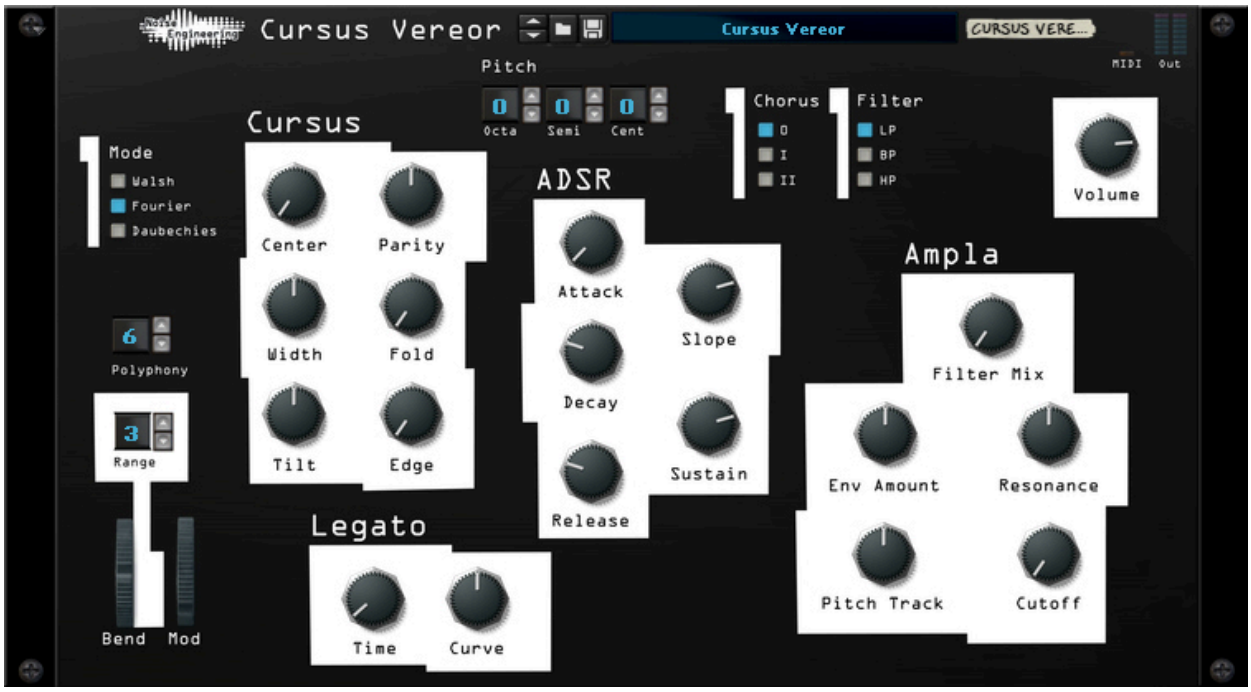
Dynamically-generated wavetable synth for unique tones



User Guide

Welcome to Cursus Vereor.

Cursus Vereor is a synthesizer that produces unique timbres from a dynamically generated wavetable. It gives the user spectral-like controls over three different modes based on different orthogonal conceptualizations of frequency: Fourier, which uses sine waves; Daubechies, using wavelets, and Walsh mode, using square waves. It has a musical tone structure and can produce an extremely wide variety of harmonic sounds.



Tone parameters

Mode: Selects which function set is used to produce the wavetable: Fourier, Daubechies, or Walsh. More information about the various modes can be found in the Function Descriptions section.

Center: Sets the center harmonic used to build the wavetable.

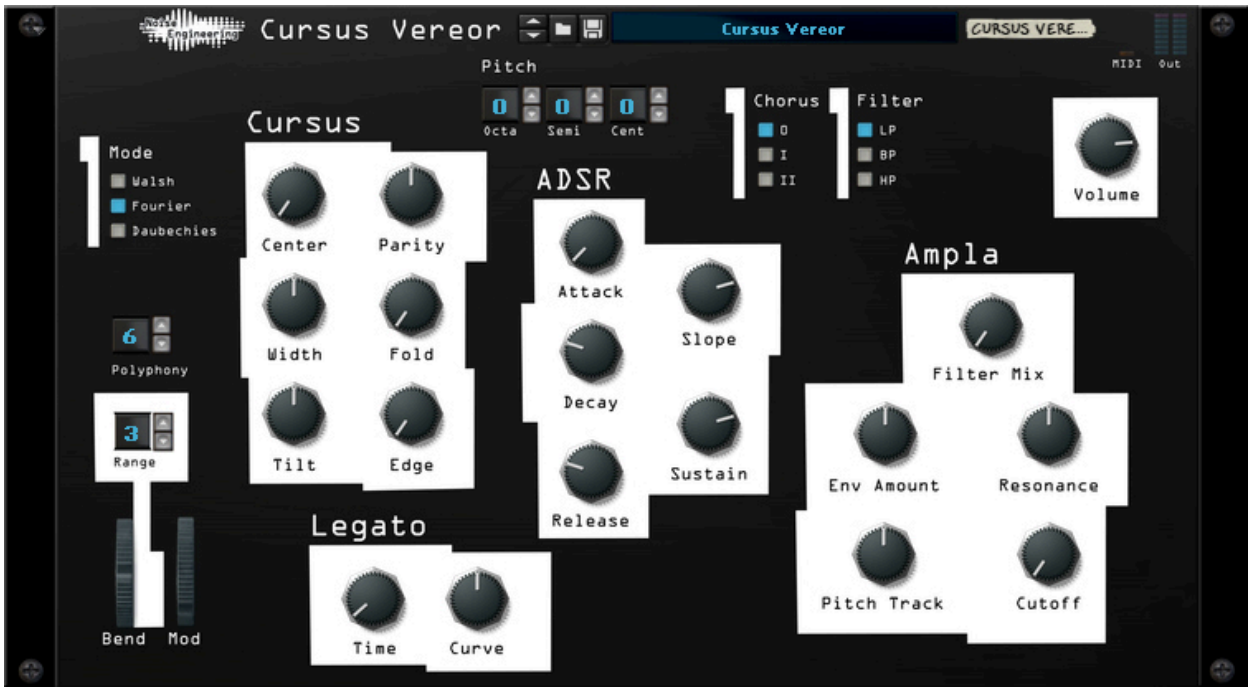
Width: Controls how many different harmonics are used to create the wavetable.

Tilt: Weights the spread of harmonics. In the middle it is symmetric; at left, lower harmonics are louder while at right, higher harmonics get more volume.

Parity: Controls what harmonics are included in the sound. In the center position, all harmonics are included. Fully left, only even harmonics; fully right, only odd.

Fold: Controls the infinifold section. For the first 3/4 of the range, this sets the threshold of the folder. CV will dynamically add multiple stages to maximize the amount of folding based on the threshold and signal amplitude. When the control is in the top quarter of its range, a pulse train based on the signal is mixed in to give even more harmonic content.

Edge: Controls the oversampling filter of the wavetable. As it is turned to the right, it will add musical overtones. Edge sounds something like a traditional “bitcrusher” effect.



ADSR

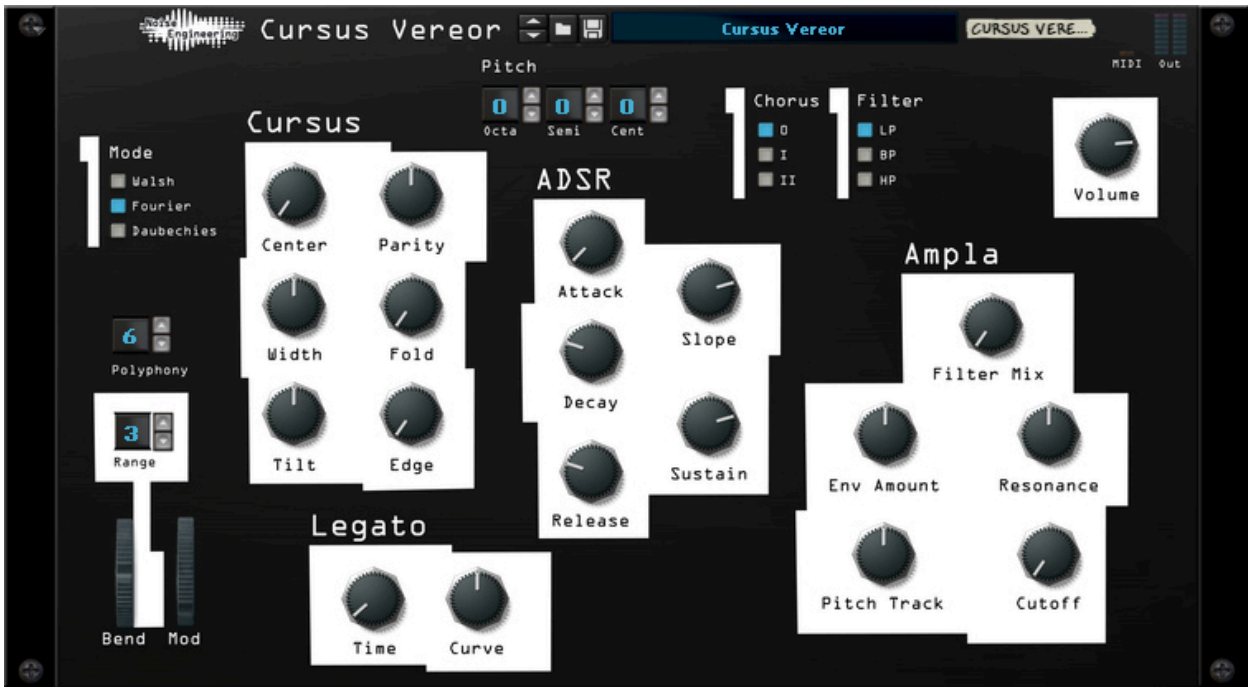
Attack: Controls the attack time for the envelope: this sets the amount of time it takes the envelope to go from minimum to maximum.

Decay: Controls the decay time for the envelope: this sets the amount of time it takes the envelope to go from the peak reached in the Attack stage to the level set in the Sustain stage.

Sustain: Sets the sustain level of the envelope: this is the level the envelope holds at after the Attack and Decay stages while a note is held down.

Release: Sets the release time for the envelope: this is the amount of time it takes the envelope to go from the Sustain level to minimum.

Slope: Changes the curve of the Attack, Decay, and Release stages of the envelope.



Ampla (dynamics+filter)

Volume: Sets the output level of the plugin.

Filter Mix: Controls the mix of unfiltered and filtered signals. To the left, no filter is heard. To the right, only the filtered signal is heard.

Resonance: Resonance control for the filter. At high values, the Resonance modulates the filter cutoff frequency for added harmonic content. This parameter will only be audible if the Filter Mix parameter is set higher than minimum.

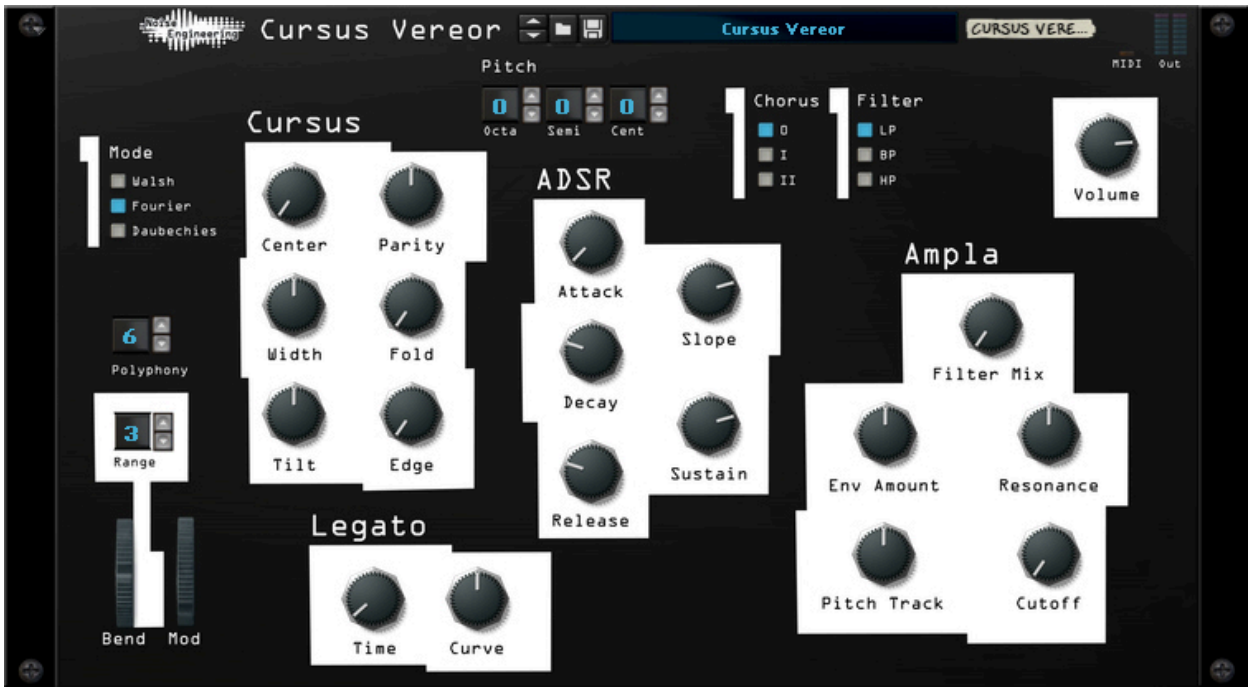
Env Amount: Controls how much the envelope opens the filter.

Cutoff: Sets the minimum frequency for the filter.

Pitch Track: Controls how much the filter's frequency tracks the notes being played.

LP/BP/HP: Sets the filter type: lowpass, bandpass, or highpass. The Filter will only be audible if the Filter Mix parameter is set higher than minimum.

Chorus: A vintage-inspired chorus: 0 is off, I is some, and II is a lot.



Note control

Polyphony: Sets the maximum number of simultaneous voices the plugin can play. The number to the right in parentheses indicates the number of voices currently playing.

Legato Time (1 Polyphony only): If two notes overlap, this sets the amount of time it takes one note's pitch to slide to the next.

Legato Curve (1 Polyphony only): Sets the curve of the pitch slide when two notes overlap.

Range: Sets the pitch bend range in semitones.

Pitch: The pitch selector adjusts the pitch of the fundamental oscillator by an octave, semitone and/or cent.

Back Panel

Back-panel knobs act as attenuators for all inputs.

Gates

Gate: input to trigger the module.

Note: CV input to specify note.

Output

Envelope: A CV output that tracks the current envelope level.

Left/Right: Stereo audio outputs.

Function Descriptions

This section is for those who want to know a bit of the scientific backstory and under-the-hood workings of the Cursus Vereor. Don't worry: if all you want is to use the plugins and make some awesome sounds, you don't need to read all of this technical mumbojumbo.

Fourier: The Fourier series/transform was discovered by Jean-Baptiste Joseph Fourier in the early 19th century to help solve differential equations used in the physics of heat. Of the three function sets in Cursus, it is the most known to musicians as it is just a harmonic series of sine waves. It is the basis (pun intended) of a tremendous amount of mathematics, physics and engineering that make our modern world. Cursus uses a 16-frequency series to form a final wavetable of 32 samples. Learn more here.

Daubechies: The Daubechies wavelet was described by Baroness Ingrid Daubechies. It is in the family of Orthogonal functions called wavelets. Wavelets are a late 20th century discovery that parameterized a huge set of orthogonal functions. In many cases where a Fourier transform has been used in the past, a wavelet can do the same job but better and faster.

For example, JPEG2000 moves the jpeg format to one using wavelets rather than a cosine transform and removes the blocking artifacts often seen in JPEG compression. The basis functions of the Daubechies wavelet look and sound a lot like low-passed sawtooth waves. Learn more here.

Walsh: The functions in the Walsh transform (named for Joseph Walsh) are similar to those in the Fourier series in that they generally increase in frequency content, but instead of using sine waves, the functions are composed of square waves. This is called "Sequency." The property that these functions are all square waves (bi-valued) was core to their adoption in a lot of technologies in the mid 20th century as this transform can be computed extremely efficiently on primitive computers or even on dedicated hardware.

For example, it was used for image compression in early satellites, similar to the way the discrete cosine transform (a close relative to the Fourier series!) is used for JPEG compression. Not surprisingly, the square nature of these functions comes through in the way they sound. Walsh functions were presented in the 70s by Bernie Hutchins (of the obscure but incredibly informative series Electronotes) as an interesting synthesis technology, but they were not used much outside of hobbyist circles. Learn more here.

Tone Generation

Cursus Vereor generates a spectral description based on slider positions. Center, Width, Tilt, Structure determine amplitudes for each harmonic. This description is fed into the inverse transform for the current function set to produce the time-domain wavetable.

The wavetable is normalized to reduce amplitude variations across spectral changes. Oversampling of the wavetable depends on pitch: lower octaves have higher oversampling since the sample rate only varies by a factor of two. The Edge control interpolates the oversampling from point sampling to a cubic-spline interpolation (NURBS).

Because the period of the full length of the wavetable always evenly divides the sample rate, the additional aliasing is largely harmonic in nature.In many places in the signal path, there are soft-clipping stages that mimic analog-style clipping to give more warmth and complexity to the sounds generated.

About the Preset Names

Our names are a bit unusual. It’s true. Product names, preset names... Let us explain.

At Noise Engineering, we think it's our job to make the tools, but not our job to tell you how to use them. Often, when products are described by a specific function (e.g., "drum module"), people grab the product for that function...and then don't explore what it can do beyond that space. Our synths are designed to be versatile and not serve a single function, and our effects are generally non-standard.

So you'll find that our product names are deliberately created to not tell you what to do with them. You decide how they best fit your workflow. Is this one for percussion? Is it smooth? Is it harsh? Is it for all your pads?

We give each Rack Extension a load of presets meant to hit a wide range of sounds so that you can have a quick taste. We started out with descriptive names like everyone else uses...and then realized that even within the team, people had different perceptions of sounds and how we would name them. And so we went back to our core practice of making the tool and not telling you how to use it: we chose not to be prescriptive.

So, about those preset names.

We are a small team of nerds. And faced with a daunting task like naming 1,000 presets for a single device, we do what we do best: we automate. We briefly considered using a dictionary, but if you’ve ever read a dictionary (at least one of us has), you’ll know there are some words in there that at least one of our users is bound to not want popping up in their session. So we did a workaround. Stephen, our chief noisemaker and also head engineer, went to the nerdiest resource he could find: the IETF, or the Internet Engineering Task Force. They produce documents for voluntary Internet standards. They are technical and cover things like Network File Systems, MD5, ISCSI, Secure Shell-2, and others. Want a nerdy list? Check it out here.

The Requests for Comments series contains technical and organizational notes about the Internet. So we grabbed some of those and made our own dictionary. If some of the presets have very weird terms -- there is probably an esoteric technical meaning to it. If Joseph or some other name pops up, you can thank them for their contribution to trying to make the Internet a slightly more sane place. Of course there was still the occasional questionable word here or there, so we went in and made a few adjustments. You may one day find a preset with the name Puppies_rainbows or with Unicorn in the name. You can thank Kris for that.

We randomly selected names from this list. These presets were then organized into categories. Each Rack Extension has its own theme, including articles of clothing, keyboard keys, and tea. Have fun with them and explore. We hope that our products will help unleash your creativity and help inspire you to think outside the box...and then get back in.

About NE

Noise Engineering is located in Los Angeles, California. We started around 2014 when Chief Noisemaker Stephen McCaul wanted a hobby for his off time from his day job and started making Eurorack modules in a spare bedroom at home. One thing led to another and a couple of years later, he and wife Kris Kaiser quit their day jobs and took the company full time. Noise Engineering has since grown in size and has established itself as a well-regarded and innovative synthesizer brand, with products in Eurorack, 5U, and multiple software platforms.

Special Thanks

Mattias Häggström Gerdt

Scott Jager

Yasi Perera

Shawn Jimmerson

Eric Cheslak

Bana Haffar

William Mathewson

Mickey Bakas

Tyler Thompson

Alex Anderson

Beta Testers

joeyluck

dioxide

Loque

aeox

tl3ss

NaviRetlav

Skullture

NisseJ

EpiGenetik

Lincolnjet

saibotsemaj