Noise Engineering
Cursus Iteritas Magnus

Dynamically generated wavetable oscillator
using orthogonal functions

Overview

Cursus Iteritas Magnus is an oscillator that works from a dynamically generated wavetable. It gives the user spectral-like controls over three different modes based on different conceptualizations of frequency: Fourier, which uses sine waves; Daubechies, using wavelets; and Walsh mode, using the Walsh transform. Cursus Iteritas Magnus parametrizes a wide variety of sounds, but because the sounds are all based off of orthogonal functions, it has a musical tone structure and can produce an extremely wide variety of harmonic sounds.

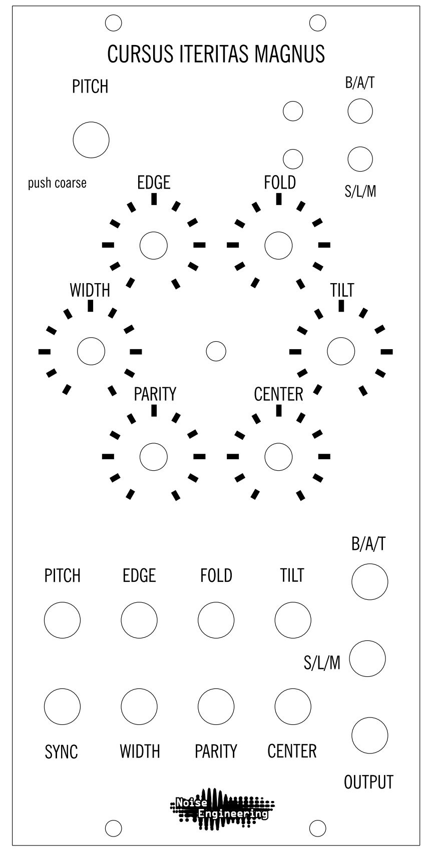
Cursus Iteritas Magnus is the 5U version of Cursus Iteritas.

Patch Tutorial

The easiest way to get to know Cursus is to just plug in the output and play with the knobs to get a feel for the sounds it can make.
Cursus works very well with any gate, modulation or envelope source -- the more the better! Patching the outputs of a trigger or gate sequencer directly into the parameter inputs on CIM is a great way to generate complex rhythmic tonality variation.
Every modulation source is fun with Cursus! Try patching in cycling envelopes, CV sequences, and whatever else you have in your system.

Interface

All knobs on Cursus Iteritas Magnus function as offsets for the input jacks. The controls function similar to a bandpass filter; center, width, and tilt allow the filter to be asymmetric.
Pitch: 1v/8va pitch control. Turn the encoder for fine tuning, press and turn the enco--der for coarse tuning. For even more control, see Range below.
Center: selects the center harmonic used to build the wavetable.
Structure: allows selection of harmonics included in the output. In the center position, all harmonics are included. Fully left only even harmonics; fully right, only odd.
Edge: controls the oversampling filter of the wavetable. As this is turned to the right, it will add musical overtones.
Fold: wavefolder. Enough said.
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.
Sync: triggers edge-based oscillator reset.
Mode: selects which orthogonal function set is used to produce the wavetable.
Range: two-octave offset pitch ranges.
Out: audio output.



Tone Generation

Cursus Iteritas Magnus generates a spectral description based on knob 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). As the period of the full length of the wavetable always evenly divides the sample rate, the additional aliasing is largely harmonic in nature. Fold controls the signal wavefolding.

In many places in the signal path, there are analog-style soft clipping stages to give more warmth and complexity to the sounds generated.

Variable Sample Rate

Cursus Iteritas Magnus uses a sample rate that is a multiple of the fundamental (lowest) oscillator frequency. This moves alias power that is a multiple of the fundamental to be mapped to a multiple of this tone, therefore making the aliasing align with the harmonics of the tone. This works well for settings with a strong harmonic structure (spread fully CW or fully CCW) and adds unique aliasing character for other tones.

Calibration of Tuning

Cursus Iteritas Magnus comes pre-calibrated but over time it may need a touch up. Pitch calibration is controlled by a linear resistor-divider network. To calibrate the tuning, attach a voltmeter (preferably 4+ digit) to the test points TPCV and TPGND on the rear panel and adjust the trim pot. The voltage measured should be 5/16 (.3125) times the input voltage applied to the CV input. A reasonable way to tune the scale is to use an adjustable voltage source to generate 4 volts then adjust the tuning trim until the test points read 1.2500V. Cursus Iteritas Magnus can also be tuned using a reference supply capable of generating a 1 volt difference and using a stroboscope such as the Peterson 490 to tune to an octave interval. This is method is preferred to the meter-only method.

Design Notes

This module started many years ago when Scott Jager and Yasi Perera turned Stephen onto Walsh Functions. The big question was how to reduce the large number of variables (32 harmonic volumes for a 32-band Walsh synthesizer) into a reasonable control set. Bandpass filter-like controls seemed to be a good solution and there already exist similar controls in the various existing Harmonic Oscillators. A software prototype was written that proved that a sequency bandpass control scheme was usable. The then project went to sleep for a couple years as other modules took priority. When we started working on it again, we wanted it to have three modes much like our other current modules so we went searching for other orthogonal function sets that could fit in the same control scheme.

The Fourier Series was an obvious second set of orthogonal functions to use which perfectly mapped to the bandpass-like controls. Modern mathematics has given us an ocean of orthogonal function sets in wavelets, so that seemed another good place to look. The Daubechies 4 wavelet fit the bill: it’s easy to compute and has an interesting—and somewhat sawtooth-like—waveform. The controls were a little less natural since this wavelet has more time precision and more frequency redundancy. With some work however it worked out quite naturally.

CIM is part of our second release into large-format modules. It’s by far the most melodic oscillator we make, and we’ve found it to be remarkably versatile. Like the BIM, it can be a lead, a bassline, or even percussive (though BIM is more suitable for percussion straight out of the box).

Special Thanks

Kris Kaiser
Scott Jager
Yasi Perera
Shawn Jimmerson
The team at Noisebug in Pomona, CA

References

Hutchins Jr, Bernard A. "Experimental electronic music devices employing Walsh functions." Journal of the Audio Engineering Society 21.8 (1973): 640-645.

Brown, Owen. A Digital Waveform Synthesizer Using Walsh Functions. Diss. 1971.

Rozenberg, Maurice. "Microcomputer-controlled sound processing using Walsh Functions." Computer Music Journal (1979): 42-47.