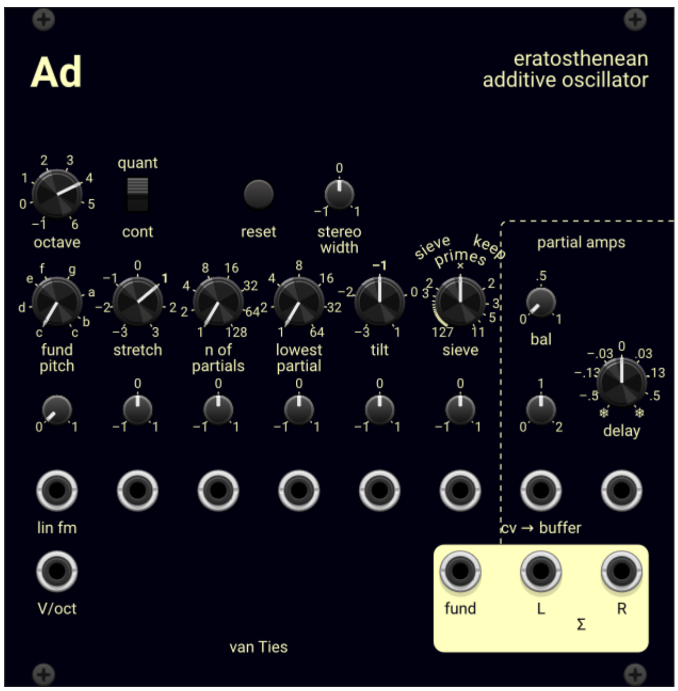
Ad van Ties
an Erastothenean Additive Oscillator
Manual
v2.3.0

Matthias Sars



'Ad' in the 'van Ties' plugin is a module for VCV Rack. It's a sound source based on the concept of additive synthesis, so I figured Ad is a good name.

Additive synthesis is based on the idea of Fourier decomposition.[1] Ad works by adding up to 128 sine waves (partials), more or less like as follows:

$$\sum_{i=n}^{n+N-1} A_i i^t \sin\Big(2\pi\big(1+(i-1)s\big)ft\Big).$$

[1] https://en.wikipedia.org/wiki/Fourier_series
https://youtu.be/spUNpyF58BY
https://youtu.be/nmgFG7PUHfo
https://youtu.be/SCujIf5eJ2w

n and N correspond to respectively the '**lowest partial**' and '**number of partials**' knobs on the panel. t is an exponent which is labelled '**tilt**', s the '**stretch**' parameter, f the fundamental frequency, which can be set with the '**fundamental pitch**' and the '**octave**' knobs. A_i are coefficients in which the '**sieve**' parameter and the '**partial amps**' section are hidden. We'll get to that.

Before we continue: almost all of Ad's parameters act continuously on the output wave. That is, there is a cross-fading between parameters that actually only make sense as integers: the number of partials, the lowest partial and sieve. (Exceptions are the octave knob and, when in quantized mode, the stretch parameter.)

The '**lowest partial**' and '**the number of partials**' parameters have the effect of respectively a high-/band-pass and a low-pass filter.

For negative values of the **tilt** parameter the lower partials are emphasized, for positive values the higher ones. In the latter case i^t can become large numbers. That's why inside Ad the amplitudes are normalized, such that the output voltages don't exceed ± 5 virtual volts.

The **stretch** parameter is the distance between partials in units of the fundamental frequency, so if it is set to 1, we have a harmonic spectrum. If it is greater than 1, the spectrum gets stretched out, if it is between 0 and 1 the spectrum gets squeezed and if it is 0 the spectrum collapses into a single frequency. If it is negative, you get partials sounding lower than the fundamental, but the spectrum 'folds' around 0. It's easier to understand through experiment: play with the number of partials and the stretch parameter while monitoring the Σ outputs in a spectrum analyser (for example the Bogaudio Analyzer-XL[2]).

The stretch parameter can be **quantized**, such that the second partial is a consonant with respect to the first. With consonant I mean here a perfect prime, a minor or major third, a perfect fourth or fifth or a minor or major sixth, plus or minus octaves, all in just intonation.

(Being nitpicky: note that a stretch value of -1 means a second partial with 0 frequency, which implies that the quantization steps are infinitely dense around there. That's why between $-1\frac{1}{2}$ and $-\frac{1}{2}$ the number of quantization steps is reduced.)

The term 'erastosthenean' refers to the sieve of Eratosthenes in mathematics, an algorithm for finding prime numbers.[3] Ad's '**sieve**' parameter is based on this. If it's set to '$\times$', it does nothing. If it's set to 2 at the right-hand '**keep primes**' side, all partials that are proper multiples of 2 (i.e. not 2 itself, but 4, 6, 8, ...) are sieved out. If it's set to 3, all proper multiples of 3 are filtered out, etcetera. If it's set fully clockwise, the primes and the fundamental are left over.

A similar thing is going on counter-clockwise ('**sieve primes**'), except that in this case the primes themselves are sieved out as well. If it's set fully counter-clockwise, only the fundamental is left. This means that (assuming that the other parameters are set to their initial values) we get a continuous transition

[2] `https://library.vcvrack.com/Bogaudio/Bogaudio-AnalyzerXL`

[3] `https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes`

between a saw, a square and a sine wave when we sweep the sieve parameter from $\times$, via 2 to 127 at the left side.

There are two sum ('**Σ**') output jacks, labeled **L**eft and **R**ight. If the **stereo width** parameter is set to 0, both outputs are the same. If the parameter is set to 1, the partials are distributed over the two channels, except for the fundamental, which goes to both channels. This is done in such a way that for any value of the sieve parameter, those two channels are pretty much in balance:

	left	right
mult. of 2:	4, 8, 12, 16, 20, 24, 28, ...	6, 10, 14, 18, 22, 26, 30, ...
mult. of 3:	15, 27, 39, 51, 63, 75, 87, ...	9, 21, 33, 45, 57, 69, 81, ...
mult. of 5:	35, 65, 95, 125	25, 55, 85, 115
mult. of 7:	77, 119	49, 91
mult. of 11:		121
primes:	2, 5, 11, 17, 23, 31, 41, ...	3, 7, 13, 19, 29, 37, 43, ...

If the width parameter is set to -1, the distribution is flipped. If there's more than one polyphony channel, the odd channels are flipped.

There's also a **fundamental** output which outputs the first partial, i.e. a sine with the fundamental frequency.

There's a **linear FM** input, which respects negative voltages (i.e. through-0 FM). This only affects the main Σ outputs, not the fundamental output, so you can self-patch the fundamental output into the FM input, for example.

The **partial amplitudes** section works as follows: the CV coming in at **CV→buffer** is recorded into a buffer. Assume the **delay** time is set to a positive value and both the balance knob and attenuator / amplifier knob are set to 1. Then the lowest partial is attenuated by the current CV value (10 V corresponds to unity), $1\times$ the delay time later the next partial is attenuated by this value, $2\times$ the delay time later the next partial and so on. In other words: the incoming CV travels from the lowest partial upwards. If you set the delay time to a negative value, it travels downwards from the highest partial. If it is set to 0 all the partials are affected simultaneously, in other words: it works then as an overall VCA. If you turn the knob all the way (counter-)clockwise to ❄, the buffer freezes and no CV is recorded.

Maybe it's easier to understand it by trying it yourself: send CV (for example an LFO or an envelope) into the CV input and set the number of partials to maximum. Experiment then with the rate of your CV source and the drawer rate. It is insightful to monitor the Σ outputs using a spectrum analyser. (I recommend the Bogaudio Analyzer-XL again.) If the LFO is a sine, one can get comb filter-like effects. Or when you send in an envelope, the individual partials each get an envelope with an additional delay stage.

(Being nitpicky again: the buffer contains 1048576 slots. With a sample rate of 48 kHz, this corresponds to 22 seconds. The thing will freeze also when the required buffer size exceeds the actual buffer size. For example: at a sample rate of 48 kHz and having all the 128 partials running, the maximum delay time is $22\,\mathrm{s}/128 = 0.17\,\mathrm{s}$.)

The **balance** parameter works as a cross-fader between unity (at 0) and full action of attenuation by the buffered CV (at 1). (Nitpick alert: the little trimmer knob above the CV in jack can attenuate, but also also boost the incoming CV up to a factor 2. This can come in handy for signals with a maximum value of 5 V: this way 5 V correspondends to unity.)

The phasors inside Ad can drift apart, especially when one plays with the stretch parameter. Usually this doesn't cause any audible difference, as long as the output doesn't go through a wave shaper. Audible drifting can also occur though, when the module runs for a while. The reason for that is that the sines are not all computed brute force, but recursively from each other. Some rounding artefacts can occur there.

There are a few ways to reset the phasors: pressing the **reset** button does this, and also empties the buffer. If the amplitudes of all the partials are 0, the phasors will be reset, as well as if both Σ outputs are disconnected.

The pitch, stretch and tilt parameters can be pushed beyond the knob ranges using CV.

Ad works with polyphony. It can get CPU-heavy, though.

Ad has a huge pitch compass, 9 octaves with the knobs only, especially towards the lower side. The idea behind that is to make it also possible to generate chords, rather than timbres. You can do this by selecting only a few partials using the lowest partial and number of partials knobs. It could be interesting to play with this transition zone of harmony and timbre.

Demos can be found here:
`https://youtube.com/playlist?list=PLTg6VAqMki3XDdgPw0jcTMevmnDNcnVxR`.

What's new…

… in v2.1.0?

- the CV drawer
- the "sieve primes" mode
- bug fix
- some knob mappings

… in v2.2.0?

- a different, generally more efficient algorithm computing the sines. This algorithm can cause some phase shifting after running a while, though.
- The CV drawer works differently.
- Some parameters are renamed.
- Some of the knob mappings have changed.
- Got rid of a few attenuverters.

… in v2.2.1?

- The phasors can't drift anymore in quantized stretch mode.

… in v2.3.0?

- The CV buffer section is reworked. The most significant bit is that there is now a large buffer which runs at audio rate. Now one can send snappy envelopes into it, which are replayed without any smoothening.
- More compact front panel,
  - by replacing the auxiliary oscillator section with only the fundamental output,
  - getting rid of the stereo width and balance CV inputs, because I consider those 'set and forget' parameters,
  - gettig rid of the amp parameter, since the new CV buffer section can take this role when the delay time is set to 0.
- Got rid of the change of v2.2.1. I didn't like the clicky sound when turning the stretch knob.
- Added the reset button.

To do:

- still a nicer front panel
- including a little screen showing the spectrum
- clock in for the delay time
- optimization