

Vulpus Labs

Curvaceous

| | |
|------------------------------|---|
| Introduction | 2 |
| Theory of Operation | 3 |
| Usage | 5 |
| Credits and Acknowledgements | 7 |

Introduction

Curvaceous is a polyphonic oscillator which plays a waveform formed of up to seven curve segments. It uses the same controls as **Curvature** to define the waveform, whose endpoints can be modulated during playback by a control voltage or audio rate signal.



VULPUS LABS
KNOW MANY THINGS

Curvaceous exists because a common use case for **Curvature** is as an oscillator, but building a polyphonic oscillator out of multiple instances would be unduly cumbersome.

While **Curvature** is a voltage-to-voltage converter which just happens to be usable as an oscillator by feeding it a pure sawtooth wave input signal, **Curvaceous** generates an input sawtooth wave of the desired frequency itself for each of its polyphonic channels, and outputs the waveform that **Curvature** would output given that input.

The conventional controls for a polyphonic oscillator are provided, with facilities for octave selection, frequency modulation and so on.

Theory of Operation

As with **Curvature**, **Curvaceous**'s waveform definition controls define *points* on a curve, and *segment curves* for the values between those points. Each point has an "x" and a "y" value.

Where **Curvature** converts an input "x" signal into an output "y" signal, **Curvaceous** generates its output signal by sending a sawtooth wave signal at the desired frequency into the "x" input. This means that at the start of an oscillator cycle, the value of "x" will be -5v, and at the end it will be +5v, with the value changing in a straight line from low to high over the course of the cycle.

There is a start point, whose "x" value is fixed at -5v, and an end point, whose "x" value is fixed at +5v. The "y" values for these points define the start and end of the curve. For a smooth oscillator waveform without noisy discontinuity, you will usually want them to be the same, since the input sawtooth wave resets instantaneously at the end of each oscillator cycle from +5v to -5v.

Up to six intermediate points can be added. Each intermediate point must have an "x" value greater than that of the point before it (otherwise it is simply ignored). The "y" value ranges between -5v and +5v.

For input "x" values between these points, intermediate "y" values are generated, based on the shape of the segment curve defined for each point.


It is helpful to be able to visualise this, which is why **Curvaceous** incorporates a handy oscilloscope showing the values of "y" as "x" sweeps from -5v to +5v.

The available segment curve shapes are:

- "Step" - the segment is a flat line at its start point's "y" value, jumping instantaneously to the next point's start "y" value at the end.
- "Linear" - the segment is a straight line from its start point's "y" value to its end point's "y" value.
- "Quarter sine" - the segment is shaped like the first quarter of a sine wave, i.e. the rising curve from $y=0$ to $y=1$ between $x=0$ and $x=\pi/2$ for $y=\sin(x)$. The rate of change is faster at the beginning of the segment and slower at the end.
- "Inverse quarter cosine" - the segment is shaped like the first quarter of a cosine curve, turned upside down. The rate of change is slower at the beginning of the segment and faster at the end.

- “Sinusoid” - the rate of change is slower at the beginning and end of the segment, and faster in the middle.

The following example curves (screenshots are from **Curvature** - the controls are substantially the same) show how to configure a sine wave oscillator, plus two variations.

| | | |
|--|--|--|
|  |  |  |
| <p>A “sine wave” curve:</p> <ul style="list-style-type: none"> • Quarter sine (-5v, 0v) to (-2.5v, 5v) • Inverse quarter cosine to (0v, 0v) • Quarter sine to (2.5v, -5v) • Inverse quarter cosine to (5v, 0v) | <p>The “sine wave” curve only with the segment types set to “sinusoid”.</p> | <p>A more irregular curve with seven segments defined.</p> |

Usage



Pitch controls for the oscillator are on the left; waveform curve controls are on the right.

You will want to set up a waveform of some sort using the controls on the right before connecting anything to the controls on the left, or the output is likely to be disappointing.

The **PITCH/SYNC** inputs accept a polyphonic V/OCT signal and audio signal respectively. The **PITCH** signal follows the usual convention for frequency control: every volt above 0v doubles the frequency, raising the pitch by an octave; every volt below 0v halves the frequency, lowering the pitch by an octave. The **SYNC** signal resets the oscillator each time it crosses from a negative (or zero) to a positive voltage - try it with the output of another polyphonic oscillator and then move the big pitch dial around, if you're not

certain why you might want to do that.

The **FREQ MOD** section contains polyphonic and monophonic frequency modulation controls. The red knobs on the right are attenuvertors - turn them right or left from the centre to increase the amount of modulation, based on the input signal. The **LIN/EXP** switches switch between linear and exponential scales.

The **PITCH** section contains a big knob for dialling in a pitch offset between -7 and +7 semitones, and a set of range selector buttons that pick the base octave, following organ pipe convention for historical reasons.

The **AUDIO OUT** jack carries the polyphonic output of the oscillator.

All of the above should be reasonably familiar if you've used any other polyphonic oscillator (and if you haven't, I suggest you play with a slightly simpler one first then come back...)

Controls for the waveform curve are arranged in rows, with one row for each point in the curve. Each point has a knob and CV input jack controlling its “y” value, a knob and CV input jack controlling its “x” value if it is not the start or end point, and a “curve” knob controlling the shape of the curve between that point and the next.

If a CV input is connected, the value sent to the jack entirely overrides the value set by the knob.

Credits and Acknowledgements

Curvaceous was written by Dominic Fox in May 2023.

Thanks to Stephen Sauvé for beta-testing, and to the developers at Cherry Audio for their great products, especially Voltage Modular.