CELLA

User Manual

Version 2.0.1

LICENSE

Source code for this plugin can be found on GitHub:
https://github.com/victorkashirin/CellaVCV

Code is licensed under GPL v3.0.

Bezier

Bezier is a smooth random voltage generator that uses Bezier curves for interpolation between consecutive values. These curves allow the generation of lines with various characteristics, from smooth to spiky, using a limited set of parameters. The functionality is heavily inspired by the module *Random Bezier Waves / ADDAC507*, which **ADDAC** designed in collaboration with Rijnder Kamerbeek aka **Monotrail**.

General Algorithm

1. **Random Value Generation**: At a frequency defined by the **FREQ** parameter, a new random value between -5V and 5V is generated. Each time this value is generated, a trigger signal is sent to the **TRIG** port. By default, the random value is drawn from a uniform distribution, meaning there is no bias in the distribution of the values. Alternatively, it can be drawn from a normal distribution, which tends to generate values closer to the midpoint rather than the extremes. This behavior can be adjusted via the context menu.
2. **Interpolation**: The module interpolates between the previous value and the newly generated one according to the curve parameterized by the **CURVE** knob. When the knob is set to the 12 o'clock position, the interpolation is linear. Turning it fully clockwise results in a rounded curve, while turning it fully counterclockwise produces a spiked shape. The result of the interpolation is output through the **OUT** port, while an inverted signal (relative to the **OFFSET**) is sent to the **OUT–** port. Additionally, the **GATE** output continuously sends a signal equal to the maximum value between 0 and the generated curve.
3. **Repetition**: The interpolation completes by the time the next random value is generated, and the process repeats from step 1.

Additional Details:

- The output signal is always constrained to the -5V to 5V range.
- The generated signal can be scaled using the **LEVEL** knob, which attenuates the signal from 0% to 100%.
- The frequency cannot be synchronized with the outside clock.

Modulation

Both the frequency and level can be modulated with an external signal. By default, modulation is sampled and applied only when a new random value is generated. However, this behavior can be modified via context menu switches, allowing continuous modulation of frequency and/or level.

Frequency modulation is unrestricted, while level modulation, when applied externally, is typically clipped to the 0% to 100% range. This clipping behavior can also be adjusted in the context menu.

Clipping

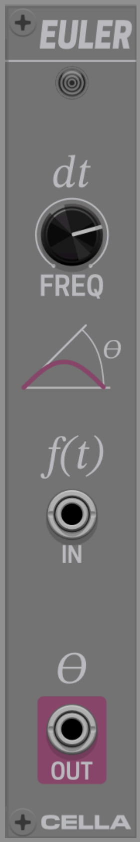
Since the output is limited to the -5V to 5V range, applying an offset may cause the resulting curve to clip. This clipping is handled differently based on the selected mode:

- **CLIP**: The curve is simply clipped between -5V and 5V.
- **FOLD**: The curve folds back from the clipping point.
- **WRAP**: The curve jumps to the opposite limit and continues from there.

Context Menu Options

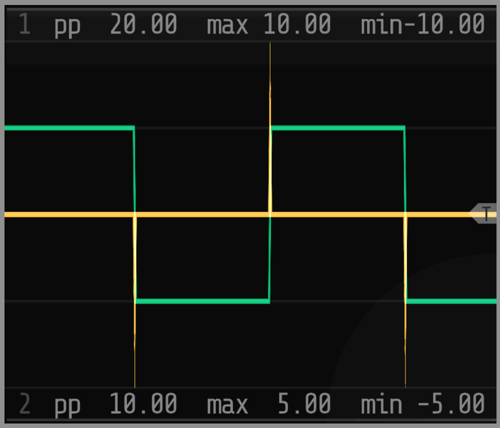
- **Continuous Level Modulation / Continuous Frequency Modulation**: When off, modulation signal is sampled only when a new random value is drawn. When on, modulation is applied continuously.
- **Asymmetric Curve**: When enabled, the curve will have asymmetry, starting smoothly and ending spiky, or vice-versa, depending on the **CURVE** parameter.
- **Distribution**: Choose between **Uniform** for equal probability of any random value, or **Normal** for values more likely to be closer to the midpoint (0 or the offset value).
- **Post-modulation Level Clip**: Sets the clipping for the level after modulation but before the offset is applied.

Euler



Euler is a simple module designed to measure the rate of change of an incoming signal. Mathematically, it calculates the angle of the slope Θ of the function f(t) at moment t. The slope angle in this context is always between -90° and 90°, making it easy to normalize the output between -10V and 10V.

Example: If you feed a sine wave from an LFO into the module and set both the LFO frequency and Euler's **FREQ** parameter to 1Hz, the scope (see left image below) will show two lines: the original sine wave (green) and the resulting signal from **Euler** (yellow), which will resemble a cosine wave. When the sine wave crosses 0, its slope angle is either -45° or 45°. Normalized, the output will be -5V or 5V, respectively. When the sine wave reaches its minimum or maximum value, the slope is 0, resulting in a 0V output from the module.



If you feed square wave (see right image above), the resulting output would be short triggers of 10V and -10V, which correspond to slope rising or falling vertically at 90° and -90° angles, respectively.

Sensitivity and FREQ parameter

Let's clarify the function of the **FREQ** parameter in the Euler module:

The angle of a slope is typically calculated using the formula `arctan(rise/run)`, where *rise* represents the change in voltage and *run* would normally represent time (seconds). However, because voltage and time are in different units, the angle produced by this formula wouldn't have a useful value range.

To make the output meaningful, we introduce the **FREQ** parameter, which represents the frequency of the periodic process being analyzed. This parameter effectively scales the time component (**dt** or "run"), bringing it into a functional relationship with the voltage change (or "rise").

For example, when analyzing a sine wave with a frequency of 1Hz and setting the **FREQ** parameter to 1Hz, the Euler module outputs a signal that perfectly corresponds to the cosine of the input sine wave. This is because the module now is calibrated to interpret the slope changes at that specific frequency.

In practice, when dealing with arbitrary signals, this exact relationship might not hold perfectly. However, you can adjust the **FREQ** parameter to scale the output to a level that is useful or meaningful for your specific signal. In essence, you can think of the **FREQ** parameter as controlling the sensitivity of the module—adjusting how the module interprets the rate of change of the input signal.

Another use for the **FREQ** parameter is boosting the output signal, so that output value is a magnified representation of incoming signal change rate.