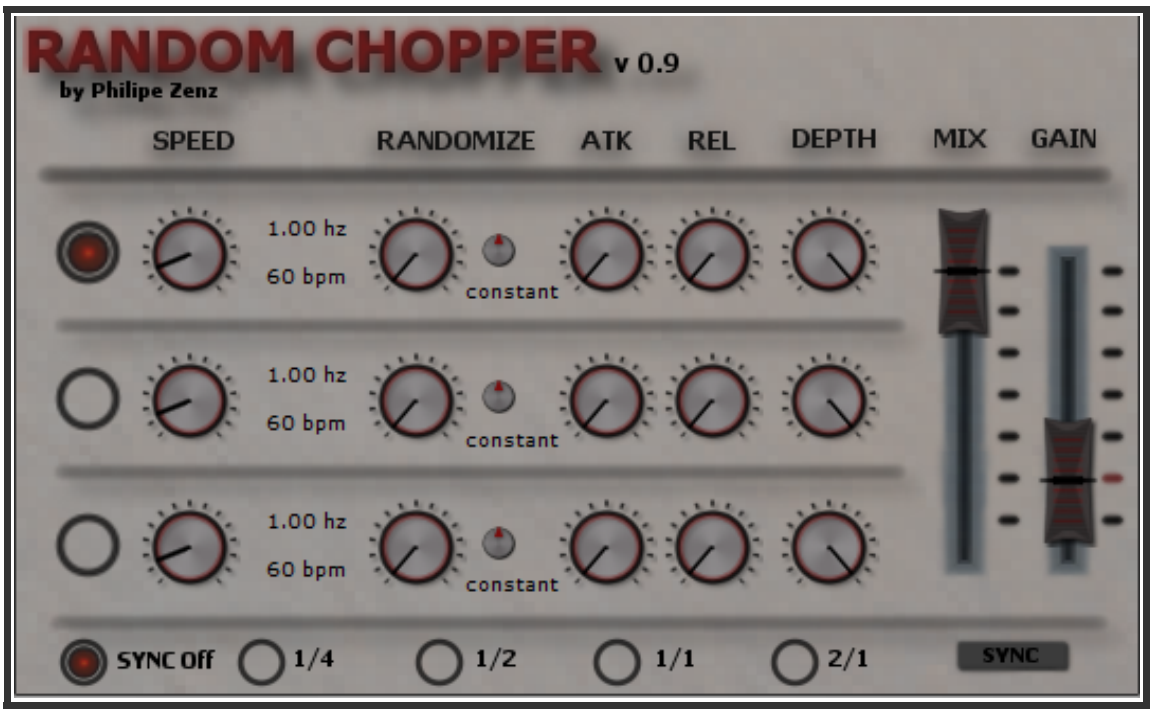
RANDOM CHOPPER v.0.9 by Philipe Zenz

Introduction

I'm a music creator in first place, not a programmer, and the concept for this plug-in emerged directly from my working process. I wanted this effect for my music but could not find it, so I decided to program it by myself. Without in-depth skills in C++, this was made possible by using Cockos' and Oli Larkin's WDL/IPlug library (http://www.cockos.com/wdl/ | https://github.com/olilarkin/wdl-ol), that has many basic functions already implemented, along with a series of tutorials by Martin Finke (http://www.martin-finke.de/blog/articles/audio-plugins-001-introduction/). Still, some features like manually typing in values are not available yet, but I hope to implement them when I find more time to engage myself in C++.

Random Chopper is basically a chopper/gate effect that mutes or reduces the output signal for a certain period of time repeatedly. It's core however feature is, that this period of time, the cycle length, can be randomized. By combining three individual modules, it's also possible to generate more complex modulation curves/patterns that apply to the signal. You may use this plugin-in to add some crackle or noise to your sounds, to rhythmize and de-rhythmize them, to add dynamics and make them sound richer, to destroy them, create glitches... there are lots of possibilities that you may try out.

User Interface



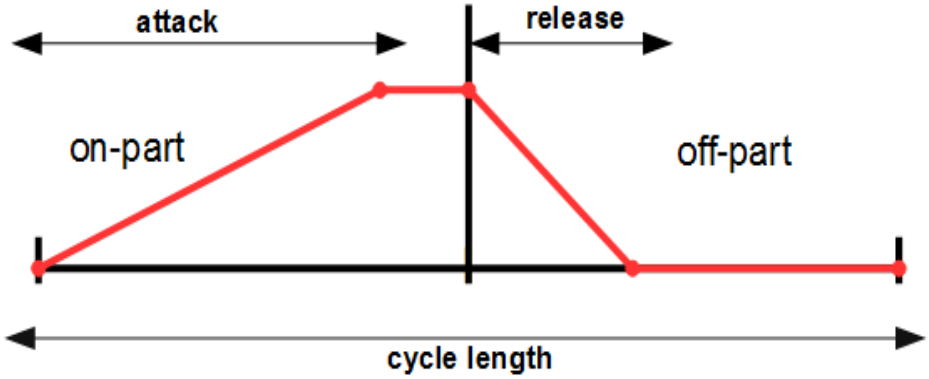
Overview

The user interface is made up of three sections: in the middle the three chopper modules that are switched on and off by the checkboxes on the left; at the bottom the sync section that will be explained later; right-hand the mix and gain faders that apply to output of all three modules together. As common, **mix** allows you to mix the audio processed by the plug-in with the original input signal (Dry/Wet). Because the output signal often is inherently lowered by the plug-in, **gain** allows you to (re-)raise the output level up to 500%. Note that the gain fader works as a distortion effect when overdriven. Customisable gain fader behaviour may be implemented in coming releases.

Basic Chopper Parameters (Speed, Attack, Release, Depth)

The **speed** parameter defines the time by which the signal-on-off-cycle is repeated and needs no further explanation. It allows you to choose values from 0.2hz (every 5 seconds) and 60hz (~A#1). Since the cycle length is designed to be randomized, there is no sync-to-host function for the speed parameter. However, you can sync it by setting the correct bpm value and using the sync section (discussed later). Still, this means in case of a tempo change, you have to adjust the speed using automation.

Attack and **release** allow to modify the square-shaped (On/Off) gain modulation curve. By using small values, you can get rid of the clicks between the on and off-parts of each cycle. Greater values will turn the square-shaped curve into a sawtooth-like or triangle curve. The following outline illustrates one cycle.



Depth defines how much the output signal is actually modified by the corresponding module. For example, a depth of 25% will mean that the the gain is only lowered by 25% during the off-part. Depth basically offers the same functionality as the mix fader, but affects a single module's output instead of the main output. Therefore, it can be used to mix one module with each other: for example, if you have a module running a really slow speed and one at fast speed, by setting the slow module's depth to something lower than 100%, the fast module's effect will also be percieved during the off-parts of the slow module, at lower volume however. Note that all modules apply at the same time, i.e. they are not processed in a chain. For that reason, it does not matter which module you chose for which set of preferences.

Randomization

Random chopper uses a pseudo-random algorithm that picks values from the input signal's amplitude as seed and transforms them into pseudo-random values.

The **randomize** knob represents the key feature of this plug-in and allows you to set the amount of randomization. More precisely, it defines the range of possible derivation in length that each cycle may have from it's usual duration defined by the speed value, while the actual derivation is a random value. You will find that the knob gradually changes it's behaviour and tends to lengthen the cycle at higher values. This is because the function can never shorten the cycle length by more than 100%, but can lengthen it by any percent value.

By using the small knob next to the randomize knob, you can chose one **randomization mode** from *constant*,

stabilize and *disintegrate*. These setting defines how the amount of randomization is affected by the input signal's volume. Stabilize will decrease the randomization amount and therefore create an evener sound at lower volumes, disintegrate does the contrary and creates a more ragged sound at lower volumes. Constant means no affection by volume at all.

Note that this feature is not so well balanced in the current version and that its noticeability highly depends on the properties of the input signal. You may want to try out the different modes with decaying guitar chords or something similar, that shows their functionality quite well. This feature may be reworked for future releases.

Synchronization

As mentioned before, due to the randomization of the cycle length, there is no real sync function that would make a rythm sync to host. However, the **sync-section** allows you to reset all modules to the beginning of their cycle together and thereby to ensure that, depending on the actual cycle length, either an attack or a continued signal is audible at these reset-points. Reset-points can sync to host (Quarter, Half, 1 Bar, 2 Bars), and can be set manually using the **sync-button** below the gain and mix faders along with automation. You may also want to use the sync-button when trying to create some non-randomized rhythmic parts and the modules got out of sync while experimenting speed values.

Changelog

v. 0.9 – First Release

Known Issues

-

Contact Information

If you are missing a feature or got ideas how Random Chopper could be improved, feel free to contact me and help to develop this plug-in further. If you find a bug, please report it to me.

music@philipezenz.com

Also check out: http://www.philipezenz.com/plugins

License