Voxengo Tempo Delay User Guide

Version 2.8

https://www.voxengo.com/product/tempodelay/

Contents

Introduction

Tempo Delay is a multi-feature stereo delay plugin for professional music production applications. Tempo Delay is based on tempo, incorporating filter and tremolo sections with separate controls for each stereo channel. Instead of a single "delay length" control this plug-in features separate "delay", "repetition period", and "delay panning" controls which allow you to create an evenly sounding "ping-pong" stereo echoes.

Features

- Filter section
- Tremolo section
- Delay signal panning
- Stereo processing
- 64-bit floating point processing
- Preset manager
- Undo/redo history
- A/B comparisons
- Contextual hint messages
- All sample rates support
- Zero processing latency

Compatibility

This audio plug-in can be loaded into any audio host application that conforms to the AAX, AudioUnit, VST, or VST3 plug-in specification.

This plug-in is compatible with Windows (32- and 64-bit Windows XP, Vista, 7, 8, 10 and later versions, if not announced otherwise) and macOS (10.11 and later versions, if not announced otherwise, 64-bit Intel and Apple Silicon processor-based) computers (2.5 GHz dual-core or faster processor with at least 4 GB of system RAM required). A separate binary distribution file is available for each target computer platform and audio plug-in specification.

User Interface Elements

Note: All Voxengo plug-ins feature a highly consistent user interface. Most interface elements (buttons, labels) located at the top of the user interface are the same in all Voxengo plug-ins. For an in-depth description of these and other standard features, and user interface elements, please refer to the "Voxengo Primary User Guide".

The knobs at the left-hand and right-hand side can be adjusted in a linked manner by holding the "Ctrl" ("Cmd" on macOS) key; the same can be achieved by adjusting the knob with the right mouse button.

Tempo

The "BPM" parameter specifies the base (overall) tempo of the delay plugin, in beats per minute.

The "BPM Mult" parameter specifies the BPM value multiplier as numerator and denominator.

The "Host BPM" switch enables synchronization of plugin's BPM value with the host's master BPM value.

Delay & Gain

The "Delay" parameter specifies channel's initial delay, in percent of the base tempo.

The "Rep Period" parameter specifies channel's further repetition period, in percent of the base tempo.

The "Feedback" parameter specifies repetition's feedback amount, in decibel.

The "Panning" parameter specifies repetition's stereo panning, in percent left-right.

The "Gain" parameter specifies channel's overall loudness, in decibel.

Note that parameter values of the channels can be changed synchronously with the right mouse button.

Tremolo

This section controls the sound of the tremolo effect.

The "Pre/Post" switch selects the position of the tremolo effect: the "Pre" position applies the tremolo to the signal before the repetition is made, the "Post" position applies the tremolo to channel's overall output.

The "Period" parameter specifies tremolo's period, in percent of the base tempo.

The "Depth" parameter specifies tremolo's depth, in decibel.

Filter

This section controls the sound of the filter of the selected type. The filter is applied before the repetition is made.

The “Freq” parameter specifies filter’s center/corner frequency in Hertz.

The “Q” parameter specifies filter’s quality: higher quality value produces a steeper filter shape.

The “Drive” parameter specifies filter’s saturation amount (in percent).

Output

The “Wet Gain” parameter adjusts the loudness of the signal generated by the delay module, in decibel.

The “Dry Gain” parameter adjusts the loudness of the original unprocessed signal, in decibel.

The “Dry Mute” switch mutes the original signal completely.

Credits

DSP algorithms, internal signal routing code, user interface layout by Aleksey Vaneev.

Graphics user interface code by Vladimir Stolypko. Graphics elements by Vladimir Stolypko and Scott Kane.

This plug-in is implemented in multi-platform C++ code form and uses “zlib” compression library (written by Jean-loup Gailly and Mark Adler), “LZ4” compression library by Yann Collet, VST plug-in technology by Steinberg, AudioUnit plug-in SDK by Apple, Inc., AAX plug-in SDK by Avid Technology, Inc., Intel IPP and run-time library by Intel Corporation (used under the corresponding licenses granted by these parties).