

Voxengo TransGainer User Guide



Version 1.7

<http://www.voxengo.com/product/transgainer/>

Contents

Introduction 3

 Features 3

 Compatibility 3

User Interface Elements 4

 Envelope Control 4

 “Legacy” Algorithm 4

 “Precise” Algorithm 5

 Output 5

Credits 6

 Beta-Testers 6

Questions and Answers 7

Introduction

TransGainer, an audio plug-in suitable for a wide range of professional music production uses, implements an audio signal envelope adjustment algorithm that reacts on transients rather than on a signal's loudness level. This algorithm allows you to adjust volume of attack and sustain stages of any sounds you use it on. TransGainer was designed in a way to be suitable for all possible sound sources – be it individual tracks or full mixes.

In many cases TransGainer can be used in place of gate and expander plug-ins while delivering a better sound and offering an easier control. Beside that, TransGainer can be used for audio recording restoration/re-mastering purposes and reverb tail modifications with excellent results.

TransGainer allows you to specify positive or negative gain adjustment for a sound's transient stage. You can also choose an expected average time between transients and transient threshold level so that the algorithm delivers you the most precise results possible. With the help of TransGainer you can pass the border from good to excellent audio recordings.

Features

- Transient stage loudness adjustments
- Four processing algorithms
- 5 processing modes in the “legacy” algorithm
- Transient-to-transient time control
- Stereo and multi-channel processing
- Internal channel routing
- Channel grouping
- Mid/side processing
- Up to 8x oversampling
- 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) and Mac OS X (10.6 and later versions, 32- and 64-bit, Intel processor-based) computers (2 GHz dual-core or faster processor with at least 2 GB of system RAM required). A separate binary distribution file is available for each target computer platform for each audio plug-in specification.

User Interface Elements

Note: Most interface elements (buttons, labels) located on the top of the user interface and on the bottom are standard among all Voxengo plug-ins and do not require much learning effort. For an in-depth description of these and other standard user interface elements and features please refer to the “Voxengo Primary User Guide”. Learned once it will allow you to feel comfortable with all pro audio plug-ins from Voxengo.

Envelope Control

This panel allows you to control input signal’s envelope (dynamics). Here you can see the transient detection meter which displays “power” of detected transients. With sound material that has weak transients the meter will mostly display values around zero while with a highly-dynamical signal the meter will read around 1.0.

TransGainer features two distinctive processing algorithms: “Legacy” which is the algorithm introduced in the initial plug-in’s release; and “Precise”/“Precise U” which are the latest reworked algorithms.

“Legacy” Algorithm

The “Mode” selector chooses processing mode of the plug-in (note that the selected mode also affects transient detection meter):

- Mellow: a “standard” mode offering a smooth overall sound.
- LF Focus: a mode with a focus on the lower frequencies.
- HF Focus: a mode with a focus on the higher frequencies.
- Round: mode with a “rounder” sound overall.
- Sharp: a generally “sharper” sounding mode – allows you to adjust transients even if the input sound’s transients are weakly-defined. However, this mode may sound less “controlled”.

The “Auto” switch enables automatic selection of the “Detect Delay” and “Contour” parameters based on the values of the “Trans Gain” and “Sustain Gain” parameters. Note that the “Auto” mode does not engage any audio signal-dependent parameter adjustment.

The “Det Delay” parameter specifies transient detection guideline – an expected average delay (in milliseconds) between two adjacent transients. Lower “Detect Delay” values force plug-in to react to most transients. Note that overly low values may produce a bit “unstable” or “dirty” sound. Best set to match song’s tempo. This parameter partially affects duration and envelope form of transient and sustain stages. Any unwanted “unstable” fluctuations at lower “Detect Delay” settings can be reduced by increasing the “Contour” parameter. However, when you are using high “Detect Delay” values (thus reducing the number of detected intermediate transients and producing an overall “stable” sound) the “Contour” parameter can be set lower.

The “Contour” parameter specifies duration (in milliseconds) of additional transient-to-transient release stage (this parameter can be also called “Decay” or “Release”). Higher “Contour” values produce smoother sound, but at the expense of reduced

“punch” effect. Mellow instrumental and orchestral recordings may require higher “Contour” values while contemporary club music will sound better with low “Contour” values. Note that high “Contour” parameter may hinder the effect of the “Detect Delay” parameter, so it is suggested to adjust the “Detect Delay” parameter with “Contour” set to a lower value. When you are using extreme “Trans Gain” and “Sustain Gain” settings you should probably use a higher “Contour” value if you want to get a natural sound decay.

The “Trans Gain” parameter specifies maximal gain adjustment (in decibels) any transient may cause. Real gain adjustment is usually lower than this value. Only the fastest transients (that drive the detection meter to 1.0) may force algorithm to reach the specified “Trans Gain” value.

The “Sustain Gain” parameter specifies maximal gain adjustment (in decibels) that may happen during signal’s sustain stage.

“Precise” Algorithm

This algorithm is based on the same concepts as the “Legacy” algorithm, but further refines the control over results you get. The “Precise” algorithm has no “Sustain Gain” parameter, because the “Trans Gain” parameter coupled with the “Trans Decay” (see below) parameter can be used in tandem thus effectively redefining the sustain part of the sound.

The “Precise U” algorithm is one of the latest editions of the “Precise” algorithm which offers an even greater precision of transient selectivity. The “Precise B” algorithm is better suited for mix and bus (broadband) processing.

The “Det Thresh” parameter adjusts transient detector’s threshold level that is used to skip weak transients which may produce “dirty” sound. You may use higher threshold value if the sound becomes too “wobbly” or “dirty” meaning a lot of weak (unpronounced) transients are being detected. When you are using low “Detect Delay” values you usually need to use high “Detect Thresh” values; and vice versa. The meaning of the “threshold” is quite straight-forward: it is the power level of “wanted” transients; transients that are weaker will be removed.

The “X.Attack” parameter specifies transient attack or hold time in milliseconds. This parameter controls the time a transient signal stays at the maximal level. Lower values provide selectivity to pronounced transients. Higher values can be used to increase selectivity of short, high-frequency, transients. If you need “snappy” fast sound you have to use lower attack values, but if you need “big” sound you should use higher attack values.

The “X.Decay” parameter specifies duration of detected transients/events. This parameter is similar to the “Contour” parameter of the “Legacy” algorithm.

Output

The “Out Gain” parameter controls the overall output gain (in decibels) of the plug-in.

Credits

DSP algorithms and internal signal routing code were created by Aleksey Vaneev.

Graphics user interface code and the “standard” graphics design were created by Vladimir Stolypko.

This plug-in is implemented in multi-platform C++ code form and uses “zlib” compression library (written by Jean-loup Gailly and Mark Adler), LibLZF by Marc Alexander Lehmann, filter design equations by Robert Bristow-Johnson, 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).

Voxengo TransGainer Copyright © 2009-2016 Aleksey Vaneev.

VST is a trademark and software of Steinberg Media Technologies GmbH.

Beta-Testers

Alan Willey

gl.tter

Jay Key

Matthew Fagg

Murray McDowall

Niklas Silen

Steffen Bluemm

Questions and Answers

Q. I'm unable to obtain an even dynamical enhancement with this plugin, because "punchy" sounds in the original track become much louder than less "punchy" sounds in the same track. How can this problem be resolved?

A. Such situation can be handled pretty well by inserting a compressor (saturator) before TransGainer. By means of compressor or saturator original track's dynamics can be equalized so that TransGainer's algorithm produces an equal loudness increase on every transient.

Happy Mixing and Mastering!