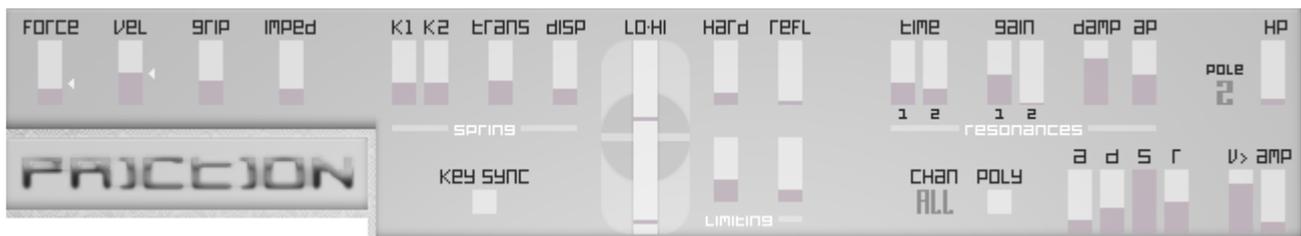


introduction

Pitched timbres created by friction have a period produced by the rate of catching and slippage between two contacting surfaces. Friction VST combines a friction algorithm with a 2nd order mass-spring to model surface responsivity, and two filtered waveguides to describe primary resonances. It is proposed as an elementary synthesizer modestly applicable to a range of emulative tasks.

The core algorithm uses 4x oversampling, however you will notice that increasing the sample rate improves emulative performance. For convenience, Arkecode's VST Oversampler is included in this distribution. The Oversampler is a wrapper that will double the sample rate of the host for the instance of the embedded plugin.

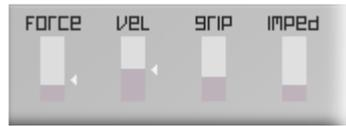
Instructions for using the VST Oversampler are included at the back of this manual.



Parameters for the synthesizer engine appear in the top panel and are arranged in four groups:

Friction coefficients are at the left. The mass-spring and mass-spring limiting parameters are in the center (if you have used the mass-spring percussion VST you will be familiar with these parameters). Coefficients for the two waveguide resonances are to the right.

the sound engine



friction

When you press your finger against a window, the sound produced is changed by the speed and pressure, and how clean, wet or oily your finger and the glass are. You can envision that if you maintained the same pressure, rate, and surface conditions, the pitch would remain constant.

The parameters work as a whole, so that changing one will change how the other settings effect the pitch and timbre. Following the example of a finger on glass, if your fingertip has a different texture, the force and speed that will produce oscillation may be different. No oscillation is produced if you move too slowly or too quickly.

Many combinations of settings do not oscillate. The most sensitive parameter is **velocity**, which generally needs to be near the median value, which is marked by an arrow on the gui. Velocity represents the difference in speed between the two surfaces. This slider is intended to be used as a center of applied modulations, as most positions generally do not produce sound.

The **force** slider also has a mark towards its lower range (representing unity in the algorithm). Note that this slider has a strong exponential curve, and also scales modulations assigned to force. In practical use, higher force settings produce louder transients and thus more aliasing, so lower settings achieve better results for emulations.

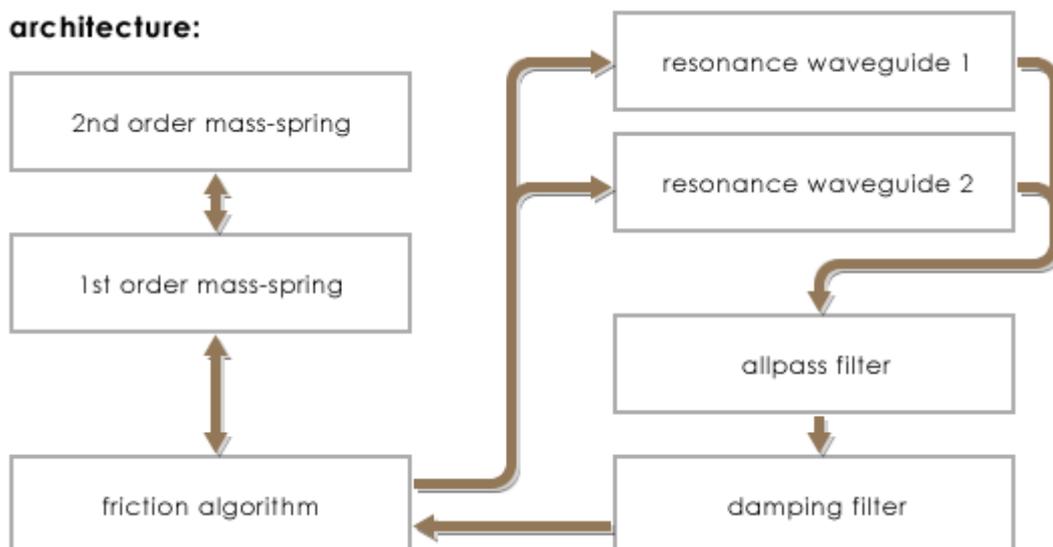
Grip represents the degree to which the surfaces catch on each other. Because the oscillation is produced by catching and slipping, too low or too high a setting may inhibit oscillation. Generally, smoother, squeakier tones are produced with low grip values, and more caustic timbres with higher settings.

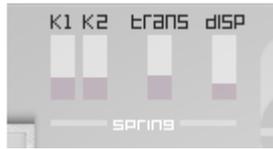
Impedance indicates the degree to which a surface resists conducting vibration, as a combination of factors such as mass.

The friction parameters are applied to and respond to the first segment of the mass-springs.

All SynthEdit VST parameters can be fine-tuned by holding [ctrl] when adjusting them. This may be necessary for achieving a satisfactory setting on some parameters, such as velocity.

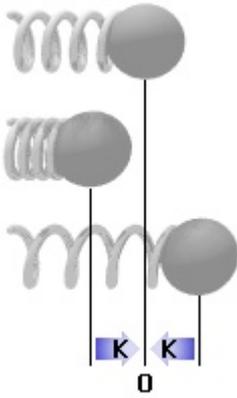
architecture:





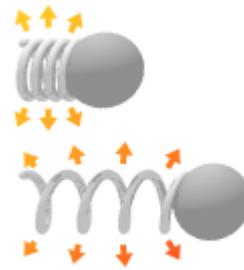
mass-springs

A spring is a system that potentiates force due to displacement. A first-order mass-spring model describes the motion of one point. Friction VST uses two mass-springs running in series, or a 2nd order mass-spring.

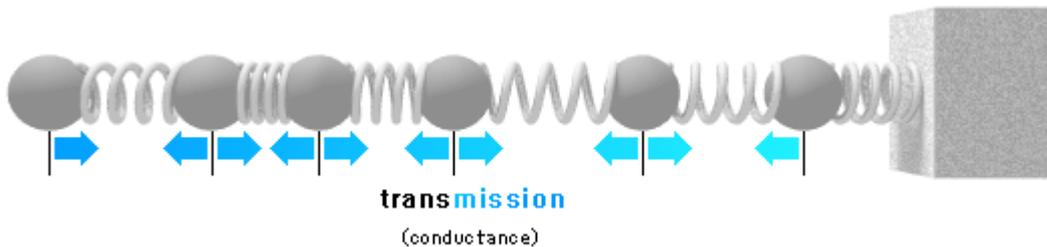


The variable **K** is used to indicate the spring constant, or how the spring reacts to displacement. **K1** indicates the mass-spring making contact.

The variable **disp** indicates dispersion of energy, or how rapidly energy decays from the mass-spring. Low settings retain energy and produce more tonal signals.



dispersion
(energy loss)



transmission
(conductance)

The third variable is **trans** indicating transmission of the signal between cascaded mass-springs.

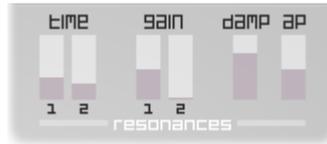
boundaries

Boundary conditions are applied to limit and reflect the first order. Separate upper and lower thresholds limit the signal like a compressor. The **hard** sliders set the amount of dynamic reduction applied beyond the thresholds.

Limiting the mass-spring oscillation simulates the vibrational medium and introduces timbral caustics. Increasing **reflection** of energy on each axis adds to environmental definition.

The **LO** and **HI** thresholds indicate the level where the softlimiting starts to take effect, and not the limit of the signal.





resonances

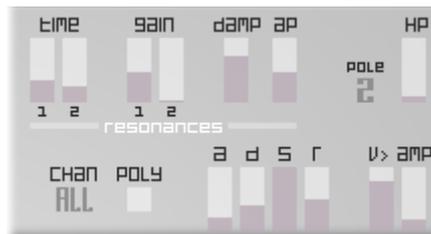
Two delays are applied where the mass-spring and friction algorithms meet. These are intended for the emulation of resonant channels, such as a bar on an iron gate, the length of a swinging cable, or the volume of a chamber.

Each delay has separate time and gain coefficients. The output is summed and filtered by 1st order allpass and damping filters before being returned.

The maximum **time** of the delays when modulation is applied is 131072 samples. The highest slider setting corresponds to 0.256 seconds. The sliders are exponentially scaled and delay lengths are rounded to the nearest sample.

The **gain** sliders are also scaled exponentially with the highest setting corresponding to -0.125 of unity. Negative gain values are used to indicate the reflection of acoustic vibrations.

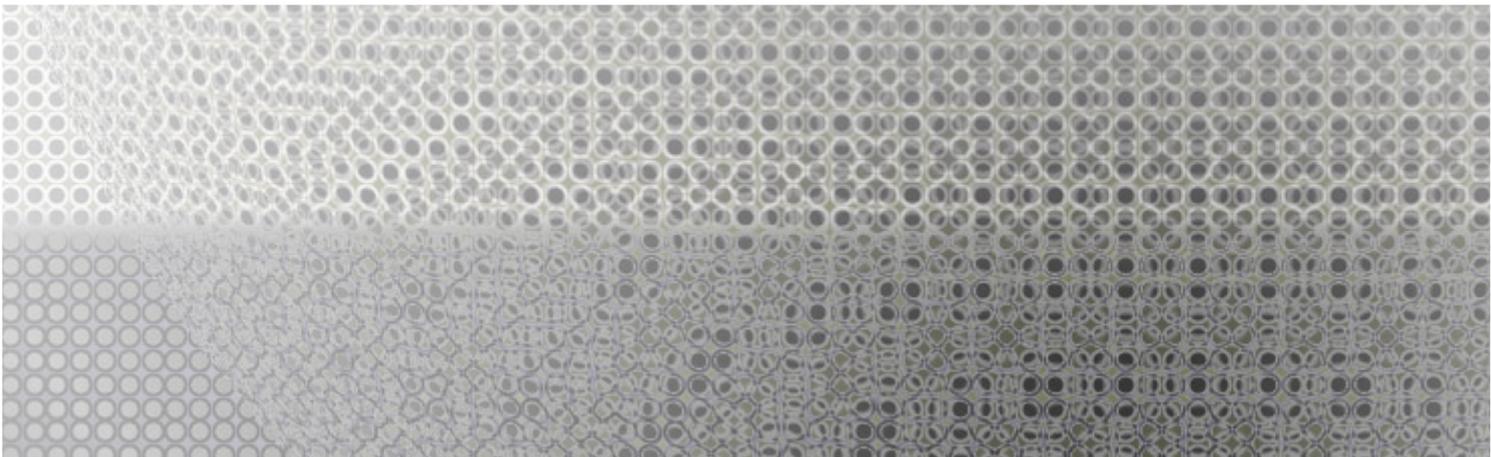
The **allpass** filter has a frequency dependent phase response, higher frequencies being delayed less than lower frequencies. Dense materials such as metal or water behave similarly, and metallic emulations will benefit from very high settings. Note that the highest setting indicates a 360° delay for the fundamental frequency, which can have a sudden increase on the gain.



output

Signals generated by the synthesis engine are passed through a 1- to 4-pole highpass filter positioned before the amplification envelope.

A 4 voice polyphony mode is available should your cpu support it :)



modulation

sends and modulation heirarchy

Sends are used to assign controllers and to route modulation sources to multiple destinations.

So that modulators can be assigned to other modulation sources, a heirarchy is used, and the sends are split into groups. LFOs 3 and 4 can be assigned to the envelopes, and the envelopes can be assigned to LFOs 1 and 2.

The heirarchy labels indicate that higher sends can use that modulator as a source, and that lower sends can be assigned to it.

Sends can also be assigned to higher sends, which can be used for assigning velocity to modulation depth. To reduce the amount of clicking necessary to select a destination, the selection of sends is limited to the current group and the group directly above it.



envelopes

Chris Kerry's graphic envelopes should be familiar to users of Fauna or Oscine Tract. Up to 32 sections can be used, with a variety of contours selectable by clicking on the envelope section you wish to edit.

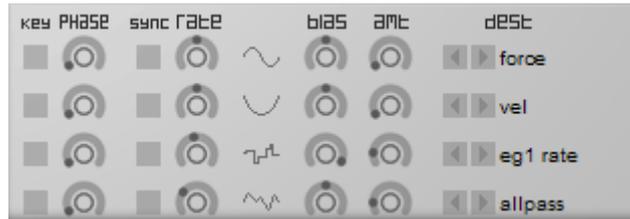
The **rpt** and **sus** sliders ("repeat from" and "sustain") are the loop start and end points respectively. The lowest position on both sliders turns the option off.

Selecting the **lock** icon allows you to adjust the height of an envelope segment without changing it's time. The lock option must be selected in the current session and will not work if the patch is saved with this button in the on position.

LIR stands for 'loop in release.' When this button is selected (light shade) the loop will continue to play when the key is released, otherwise the envelope will progress past the sustain segment.

Invert options are provided for the amplitude of the envelope, and for the way velocity effects the envelope rate. When inverted, higher velocity will shorten the envelope time.

It is recommended that you edit the graphic envelopes in mono mode for correct performance.



LFOs

The LFOs have 13 waveshapes, including a stepped and smooth noise option. The parameters should be straightforward - **key** reinitialises the LFO to phase on a gate event, and **sync** locks the rate to the host tempo.

Selectable sync rates in measures are 64, 32, 16, 8, 4, 2, 1, 1/2, 1/4, 1/6, 1/8, 1/12, 1/16 and 1/32.

Bias shift the polarity of the LFO. In the center, the lfo output is normal (-5 to +5). To the far right, the LFO is only positive (0 to 10) and to the far left, only negative (-10 to 0).



x-y pad

The x-y pad can be assigned using the send section. The rate of movement is also available as a modulation source. The **rate** slider can be used to adjust the response to movement, lower settings having a slow release.

USE

patching techniques

At the present time I have spent a few months developing mass-springs and my ears are 'pretty good' on scraping, shrieking, ringing, squeaky and screeching sounds.. most of the presets are thus elementary and demonstrate timbre only. There are some foley applications and I'm sure noise aficionados will enjoy it, ultimately I'm not sure what this VST will be used for, so I've created the presets as starting points.

The architecture is abstract, and some creativity is useful in emulative patching. An analysis of a physical form may translate well to the parameter set, or may be improved by fortuitous settings. For instance, an actual object with low springiness may be more successfully emulated with high k values if the desired sound is high pitched. The parameters might be set to describe a systemic dynamic rather than an element with known qualities.

If you have the mass-spring VSTi, you have a better opportunity to observe the timbral effects of the limiter on the mass-spring oscillations. High reflections usually create more caustics and high 'hard' values create higher pitched transients but may reduce the amplitude of oscillation.

Use of modulators assigned to the gain parameters allows selective emphasis of timbres produced by resonance.

Lower dispersion values may create an unnatural sustain for an emulation, but may boost timbral richness given the elementary architecture in comparison to acoustic forms.

Arkecode VST Oversampler

The VST Oversampler is a wrapper that doubles the sample rate of the enclosed plugin. This allows you to improve the timbre of emulations without running all plugins at a higher sample rate.

Further information on VST Oversampler can be found at <http://chris.r.walton.googlepages.com/oversampler>

A distribution of the oversampler is included in the zip file, entitled friction_os.dll. To use the oversampling, copy this file to the same location as friction.dll and select 'friction os' from the VST list instead of 'friction.'

Thanks to Chris Walton for his kind contribution and permission.

license

Friction VST may not be redistributed without permission. There is no expression of guarantee due to the use of third-party resources.

Friction VST is made with SynthEdit - <http://www.synthedit.com>

© 2008 Rurik Leffanta · All Rights Reserved

<http://www.xoxos.net>

VST is a trademark of Steinberg Soft und Hardware GmbH

