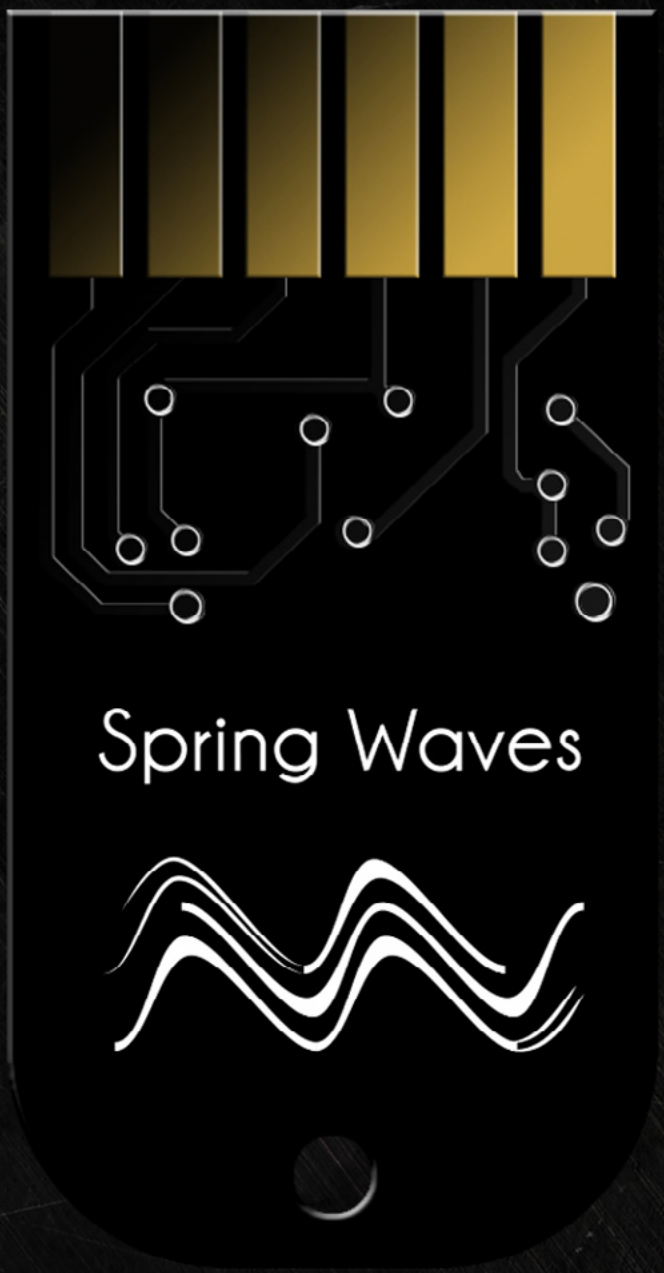
Spring Waves

Physical Modeling Effects
for Tiptop Audio Z-DSP



TIPTOPaudio

Spring Waves - Physical Modeling Effects for Tiptop Audio Z-DSP

Background:

Physical modeling

Physical modeling is the principle of processing a sound through a set of equations and algorithms simulating the behaviour of a real world physical object (string, blown pipe, drum skin, spring, metal bar, piece of wood, etc.). Several techniques have been developed for physical modeling, and two variants of them are available in this cartridge: waveguide (or Karplus- Strong based) modeling, and a spring model that could be considered as a "state space model" as defined in the **Julius O. Smith III** classification.

Waveguide

The term "digital waveguide synthesis" was coined by **Julius O. Smith III**, one of the great contributors to this type of synthesis, and describes a set of different techniques utilizing the **Karplus–Strong algorithm**. Invented by **Alexander Strong** and **Kevin Karplus**, this algorithm is based on a small delay line where an impulse (or short sound impact) is provided as input and the output is fed back to the input after being filtered by a low pass filter. The circulation of the impulse in the delay line is very similar to the circulation of a wave on a string. The pitch is then controlled by the length of the delay line. The "digital waveguides" are based on a set of such algorithms connected together in different manners.

Spring model

The Spring Waves cartridge also features two unique Spring models. This model calculates at the sampling rate all of the Newtonian forces controlling the behaviour of a mass-spring system, and takes into account the following factors: the spring itself and its return force (with a potentially non linear behaviour), a point where one side of the spring is attached, a mass attached on the other side, fluid friction (potentially non linear) depending on the speed of the mobile mass, mechanical limits (a physical spring can't stretch out infinitely) and static and dynamic frictions (friction due to the contact with a rough surface).
The audio input is used either to "move" the point where one side of the spring is attached, or as force directly applied to the moving mass on the other side. The audio output is directly provided by the motion of the moving mass.
More information about his algorithm can be found here: **http://www.anckorage.com/wp-content/uploads/2014/09/Anckorage-SpringSound-v0.pdf**

How it works:

In the context of the Z-DSP, the FV1 chip (its brain) computes these physical modeling algorithms. Physical modeling is known to have heavy CPU consumption, but, despite its limited power, the nice design of the chip allows running up to four waveguide lines or one spring model, with optimised code.

For waveguides, the one second delay line is split in 2 to 4 parts and necessary lowpass and DC-cut filters are inserted in each feedback loop. The rest of the code manages the connections between the waveguide lines and the controls. When used as effects, as it is the case for Z-DSP, waveguides can be used in different ways:

- input very short sound burst or impulse and let the waveguide ring, with low damping (controlling the tuning and additional parameters). A variant is of course to use drum sounds as input. Very interesting results can also be obtained using sequences. For drums and sequences, the damping needs to be lowered to fit the BPM.
- input a continuous sound (with a not too high level to avoid clipping) and use the wave guide as a resonator. This can even give very nice results processing an instrument track or even a voice. In this case the damping needs to be quite high.

For the spring model, the whole program is dedicated to arithmetics to calculate all the newtonian forces and their consequences on the motion (and resulting sound). As only one spring can be run by the FV-1, the more spontaneous way to use it as a filter, but, like the waveguide, you can also input a short audio burst to make it ring or use it as a resonator.

Controls:

For the waveguide models, the first six programs have common controls:

- Left and right inputs can be used the same way
- Control 1 is defining the pitch of the waveguide (low frequencies on the right). Note that for programs 1 and 2, control 3 is also for pitch since control 1 sets the pitch for one half of the waveguide lines and control 3 the other half.
- Control 2 is damping (low damping, meaning longer ringing, is clockwise).

For the spring models, the last two programs have common controls:

- The Left input is the "regular" input and is moving the point the spring is attached to (the output will be the resulting end position of the spring where le "Mass" is attached)
- The Right input is an "alternate" input that is going to be used as a force directly applied to the moving mass (be careful with low frequencies ....). This input needs to remain at a very low level.
- Control 1 is always the hardness of the spring (thus determining the force that will bring the mass to the center point). You can compare its behaviour a bit to the frequency control of a regular filter.
- Control 3 is always the Fluid Friction, that will slow the moving mass (the friction depends on speed). You can compare its behaviour a bit to the the resonance control of a regular filter, but here low friction means high resonance.

Programs:

1> Double Waveguide Mixed

Two waveguide lines, with their output merged and fed into feedback (non conventional waveguide).

Tune 1 - Pitch of the first waveguide line. See the Control section above
Damp - Damp of both waveguide lines. See the Control section above
Tune2 - Pitch of the second waveguide line.

2> String Waveguide

Waveguide simulating a plucked string.

Tune 1 - Pitch of the first waveguide line. See the Control section above
Damp - Damp of both waveguide lines. See the Control section above
Tune 2 - Pitch of the second waveguide line.

3> Bass Waveguide

A waveguide dedicated to bass processing.

Tune 1 - Pitch. See the Control section above
Damp - Damp. See the Control section above
PitchEnv - Depth of the input envelope follower. The resulting envelope is connected to the pitch of the waveguide. The center position is "neutral", left is lower pitch and right higher pitch.

4> Minor Chords

Four waveguides tuned into a minor chord. The last note being a fourth, a minor seventh, a seventh or an octave.

Tune 1 - Fundamental Pitch. See the Control section above
Damp - See the Control section above
Chord - Upper note pitch: fourth, minor seventh, seventh or octave.

5> Waveguide Mesh 1

Four independent waveguides with spread pitch.

Tune - See Pitch the Control section above
Damp - See Damping the Control section above
Spread - DIstance between the different waveguide pitches.

6> Waveguide Mesh 4

Four interconnected waveguides (the connection makes the sound different from the previous program) with spread pitch.

Tune - See Pitch the Control section above
Damp - See Damping the Control section above
Spread - DIstance between the different waveguide pitches.

7> Spring Filter 3 (LP)

Spring physical model that is a bit similar to a lowpass lowpass filter

Force - Spring force ("similar" to the frequency control of a filter).
See the Control section above
Lin-3 - Linearity of the Spring Hardness and the Fluid Friction together
Friction - Similar to the inverse of the resonance control of a filter).
See the Control section above

8> Static Friction 2

Spring physical with Static Friction: this is very nice to destroy a sound and make it very crunchy!

Force - Spring force ("similar" to the frequency control of a filter).
See the Control section above
Static - Static friction level (the higher it is the more it will scratch..)
Friction - Similar to the inverse of the resonance control of a filter).
See the Control section above

Algorithm Design
Christophe Duquesne-Aurige

Graphic Design
Piero Fragola

Manual
Chris Clepper
Christophe Duquesne-Aurige
Piero Fragola

Beta Testing
Dr Richard Boulanger
Chris Carter
Chris Clepper
Richard Devine
Joseph Fraioli
Gur Milstein
Neil Parfitt
Robin Rimbaud
BT