

MAGALOV AUDIO GRAIN-SKAN

QUICK START GUIDE:

1. Extract Grainskan folder to your vst directory.
2. Initiate an instance of Grainskan in your host.
3. Select 'load all programs' from the 'file' menu in the master controls.
4. Select 'GSPresetBank.txt' from the 'Grainskan/samples' directory.
5. Play away!

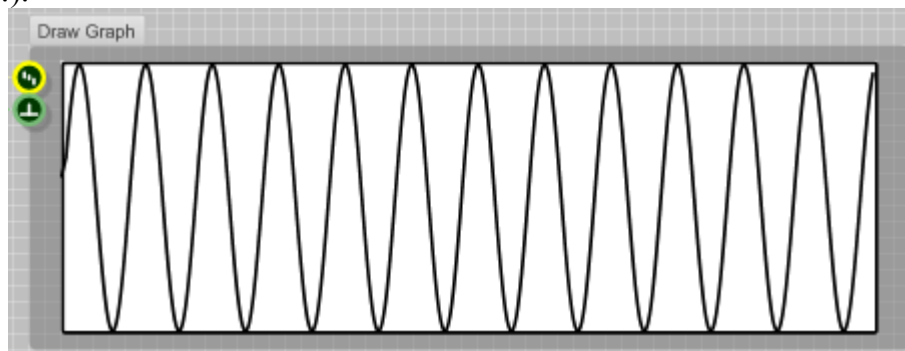
THE GUIDE

The Grainscan synth is perhaps best described as a hybrid sampler-wavetable-granular synthesizer. It can be used to create twisted techno funk loops, luscious evolving pads, glitchy blips and much much more. Its engine is based upon a crude time-domain pitchshifter, harnessing the colouration introduced in the process to generate rich harmonic tones, and parameterized in such a way as to allow the user to explore the transition between effect and instrument.

A Crude Time-Domain Pitchshifter (CTDP)

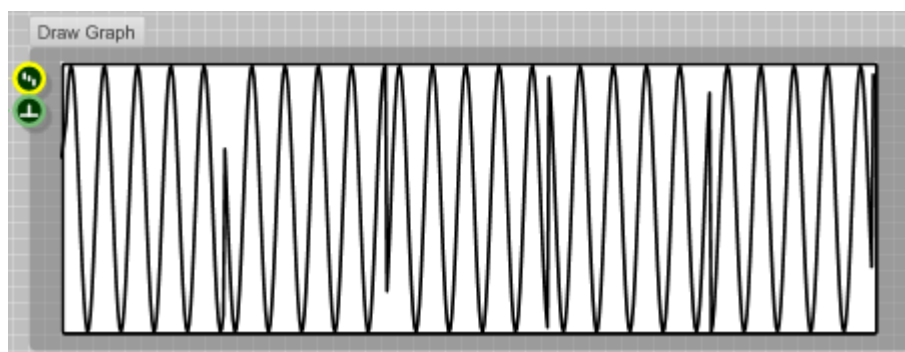
To understand how Grainscan works it's perhaps best to cover the mechanics of a CTDP. You will probably be familiar with such an effect, familiar with the low-fidelity, ringing, distorted, inharmonic sound they produce.

They work by taking short samples from an audio stream, and playing these back at a faster or slower speed. Each sample captures the timbre of the audio at that instant, and when played back will sound like the original audio but shifted in frequency proportional to the ratio of the speeds (In theory at least!).

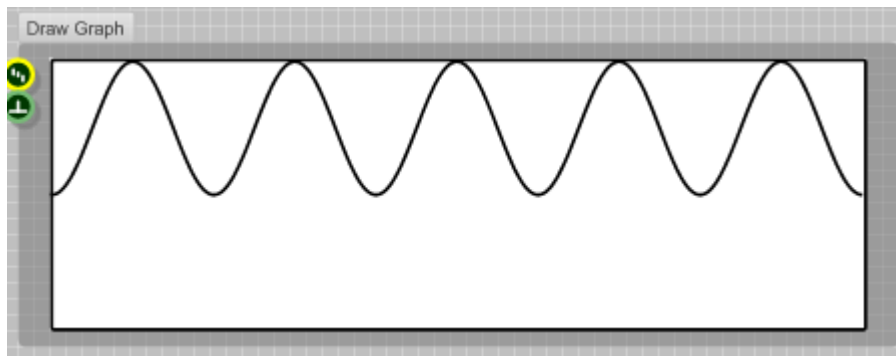


Original Sample

The first problem arises at the boundaries between the samples when you arrange them together at the output; as they are not necessarily continuous you have to ensure that there is a smooth transition from one sample to the next. This is achieved by enveloping the sample, reshaping it so it starts at zero, rises to full amplitude, and slopes back down to zero at the end of the sample. By doing this we know the transition point will always be at zero and one sample will pass smoothly onto the next. This is performed using a 'Raised cosine window'.

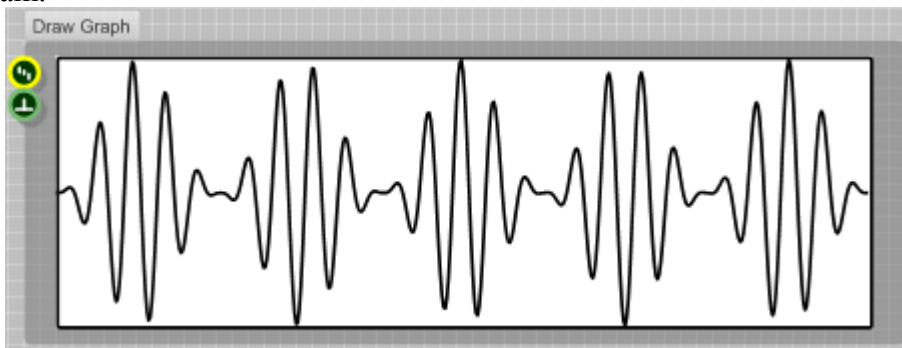


Samples Doubled in Pitch



Raised Cosine Window

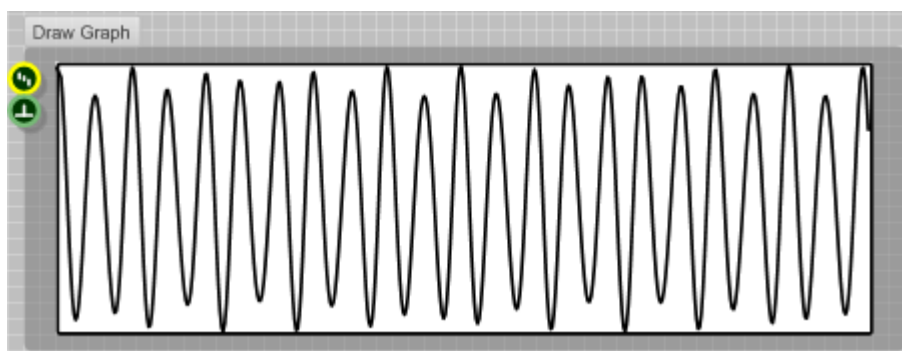
Solving the first problem brings us to the second. Now at the transition points between the reconstructed samples we have a gap in the audio, the stream is now being modulated by the cosine causing it to pulse. The solution to this is to perform the process twice, but to keep the second process out of phase with the first, so we sum the two streams together we will have an even amplitude stream.



Enveloped Signal



Out-of-Phase Enveloped Signal

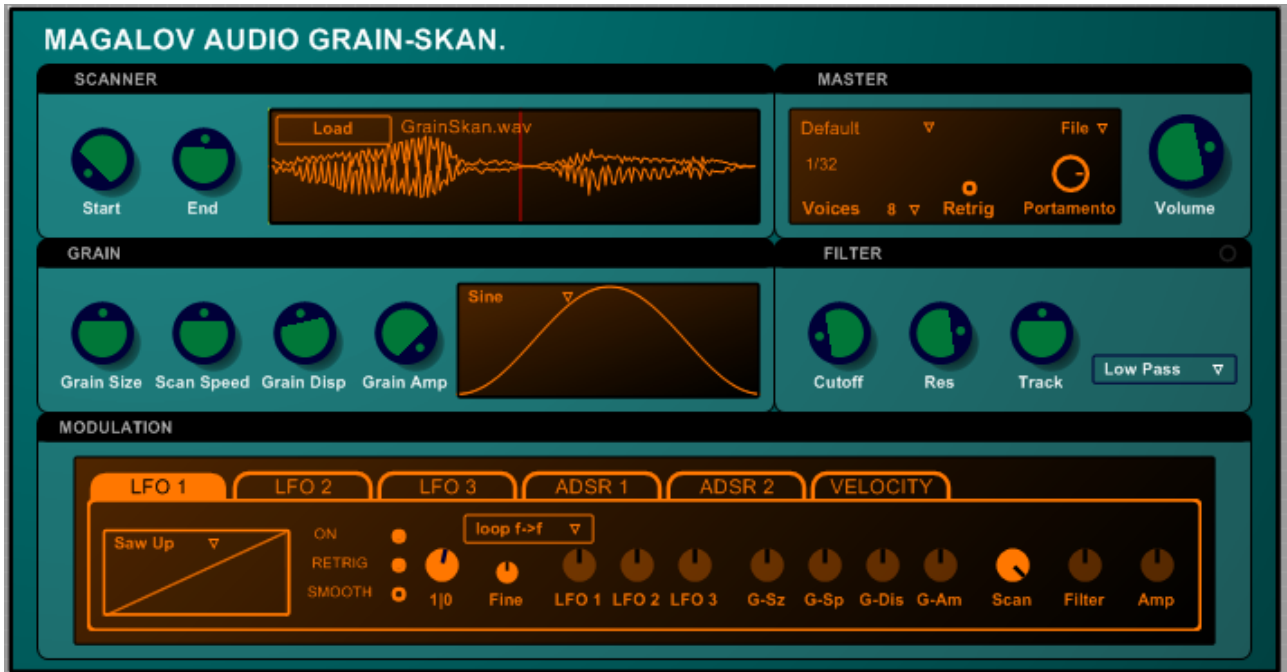


Sum of Signals

When it comes to parameterize the CTDP, apart from the shape of the envelope, we need to define the size of the envelope. This is our next problem. If we choose the window to be too short i.e. high frequency it will start to introduce colouration to the sound at the frequency of the envelope, the windowing is after all just ring modulation. If it is too large we will lose time resolution, the original sample will become smeared out, and phasing between the two envelope streams will occur. As this is only a 'Crude' TDP being described there is no solution to this problem, without greatly increasing the complexity of the model, without introducing synchronization, without overlapping and adding there is nothing we can do, unless we give up on making a CTDP and instead use this model to build a synthesizer.

The Grainscan

What has just been described is in fact the engine for the Grainscan synth. It generates two envelope signals which are used to shape an audio sample. The difference being the envelopes are called grains and the process of enveloping is called granulation and you the user has control over all the workings of this engine.



The Keyboard:

To begin at the beginning, arguably the most fundamental part of a synthesizer is the keyboard, it controls the period of the sound being generated giving the sound its pitch. With the Grainscan the keyboard input controls the period of the envelopes/grains, this implies the colouration we were talking about earlier is harmonically related. If you are using the Grainscan as a synth the timbre of the audio that is granulated defines the timbre of the output and the frequency of the grains controls the frequency of the output. We will cover this further when we get to the scanner.

The Grains:

In this section you have 5 controls which define the shape of the grains:

Grain size: This controls the size and hence the period of the grains, another way to look at it is to consider it a 'Detune' control.

Grain speed: If this was a CTDTP, this would be our pitch-shift control. It controls the rate at which the sampled audio is played back. If we have high frequency granulation and hence strong colouration, this controls the timbre of the audio, a higher grain speed will result in a richer sound. And if we have lower frequency granulation this can be used to pitch-shift the audio.

Grain displacement: Even though our grains are always out of phase with each other when the audio is reconstructed, we can control the point at which they sample the audio. The sampling points are relative so this controls only one of the grains. This can be used to control the even/odd harmonic distribution.

Grain Amp: This controls the gain of one of the grains, you can use this to attenuate or turn off the grain, or in fact reverse the amplitude, again this is useful for adjusting the harmonic distribution.

Grain Shape: This drop-down control allows you to specify the shape of the grain, the smoother the shape, the smoother the sound, the converse is of course true, so you VA freaks better watch out: absolutely no effort has been made to reduce aliasing.

The Scanner:

The scanner only has 3 controls, start, end and load. The start and end controls are only guidelines for where the audio will be scanned from. The scanning itself is controlled from the modulation section and so the limits may be broken if more than one modulator is controlling the scan position.

How does it work?

Each 'scan' modulation source outputs, at most, a value between -1 and 1, in the scanner this corresponds to a position between the start point and the end point, where it is pointing to is called the scanning point. Meanwhile, the grains will granulating the audio sample starting from the scanning point. Every time a grain is completed the grains section will sample a new point from the scanner and start granulating from there.

If there is no modulation the scanning point is static, the grains will be sampling the same section of the audio over and over, hence the output wave itself will be static and periodic in the true sense.

If there is very slow modulation, the scanning point will vary only slightly between each sample, hence each grain will be only slightly different to the previous, this implies the output sound will be a slowly evolving timbre.

If the modulation rate matches the original playback rate of the sample the Grainscan will be acting as if it is a CTDP, resampling and pitchshifting the audio.

For larger modulation rates, the sound will neither be harmonic, nor match the original sample, this is useful for generating noisy belch-like sounds.

If you have multiple modulators controlling the scan position their output is summed to get the scanning point, allowing complex dynamic sounds.

Loading Wav Files

Wav files can only be loaded from a folder adjacent to the dll called 'samples' you should make sure such a folder exists and any files you wish to load into Grainskan are located there. If Grainskan is unable to load a file, you should get a 'No Sample' message.

The filter:

I will take this opportunity to catch my breath, this need no further explanation.

The Master Controls:

Again this is fairly self-explanatory, with the exception of the preset file options. These use the Synthmaker preset standard which saves as a text file, this can be imported to a spreadsheet application such as Open Office Calc and manually edited.

The Modulation:

There are six sources of modulation, and each of these has 10 destinations:

The LFOs:

There are 3 identical LFOs, these are frequency synced to the host application. There are various waveforms which can be selected, some standard, some not.

The standard ones include: Sine, Half Cos, Saw Up, Pulse, Sinc and Noise. Inverses of these waveforms can be selected by using a negative modulation amount. Sinc is similar to an exponentially decreasing oscillation, and noise is in fact 'Red noise' or 'Brownian motion' in other words like a drunken man staggering aimlessly from left to right trying to get home.

The non-standard ones are Random Pulse and Random Sample. Random pulse is a pulse wave whose amplitude at each period is randomly either up or down. Think of a random gate effect. Random Sample is similar with the exception at each period the output is a random value between -1 and 1.

The other controls with the LFOS are:

On/Off: turn off unused LFOs to save cpu.

Retrig: This retriggers the LFO when a key is pressed, otherwise the LFO is retrigged after the ADSR release has settled to zero. Having retrigger on restarts the LFO but may result in some Glitches in the sound.

Smooth: This smooths out any hard edges of the LFO.

Rate: The rate of the LFO, the format is 'Bars|Quarter notes'

Fine: Fine tune the rate, the range in 0.5->2 times.

The ADSRs:

Pretty standard stuff here, exp make the ADSR envelop exponential.

Velocity:

This takes the velocity of the key pressed and sends it to the modulation destination, the signal can be smoothed to reduce clicks.

The Destinations:

LFO1, LFO2, LFO3: the LFOs can be modulated with feedback thanks to Synthmakers per-sample feedback abilities. By feeding back you can get some interesting modulation shapes.

Grain Size, Grain Speed, Grain Displacement, Grain Amp: The parameters talked about earlier can all be modulated from here.

Scan: As mentioned in the scanner section, this controls the scan position.

Filter: this controls the filter frequency

Amp: Modulates the current notes amplitude. **Note: At least one ADSR to Amp is needed to get sound.**

Well, that about covers it, I could probably go into more detail but I'll leave the rest up to you..

Enjoy.

Samuel Magalov.

Thanks to all the Devs at Synthmaker HQ who made this madness possible and everybody at the Synthmaker forum for their encouragement and inspiration.