PiROMANCER

User's Guide

1 Introduction

PiROMANCER is a software implementation of a semi-modular monophonic synthesiser, built with a Raspberry Pi.

Monophony means that only one sound of one frequency (note) can be produced at any one time. This note can controlled using a MIDI keyboard.

Since monophonic synthesisers are often used as arpeggiators, it is recommended to play a series of notes (such as a chord) in quick succession (arpeggio).

This guide will focus on how to manipulate the synthesiser's sound, by explaining how each module works, followed by a short guide on how to create several sounds.

2 Patching

Modules can be connected together using patch cables. The signal processing modules (Oscillator, Filter, and Amplifier) must be patched together by connecting one module's *Signal Out* to the other's *Signal In*.

The LFO and Envelope outputs can be patched into the approprate inputs on the other modules (indicated by the corresponding arrow labels).

3 Signal Updating

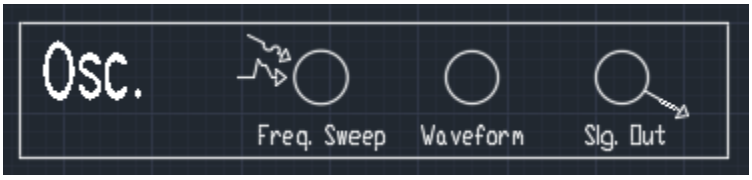
Due to the limitations of this software synthesis approach, the audio signal is not entirely realtime, and is instead updated depending on the state of the synthesiser.

If a patch is changed, added, or removed, the audio generator is updated. Also, if a button is held down, or a module's potentiometer (knob) is turned, the generator is updated.

This update has a small latency (of around 1 to 2 seconds), where changes are detected.

4 Modules

4.1 Oscillator



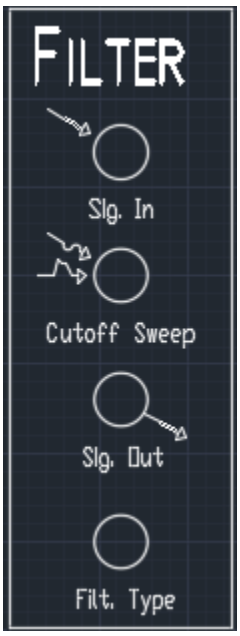
The oscillator converts the MIDI's note input to a waveform of the according frequency, creating an initial audio signal.

Frequency Sweep modifies the waveform's frequency based upon the amplitude of an LFO or envelope input.

The *Waveform* button switches between different waveform types in order whilst it is held down. These waveforms are:

- SuperSaw (Roland JP-8000 Emulation)
- Sine Wave
- Triangle Wave
- Square Wave
- Pink Noise

4.2 Filter



The filter module filters out a range of frequencies from an input signal, changing the characterstics of the sound output.

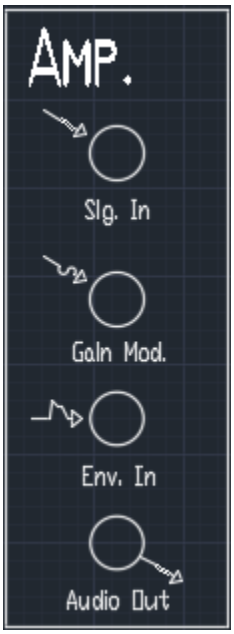
The filter's cutoff is approximately equal the note's frequency.

Cutoff Sweep modifies the cutoff frequency based upon the amplitude of the LFO or envelope input.

The *Filter Type* button switches between various filters whilst it is held down. These filters are:

- Moog Resonant Lowpass - excludes frequencies above cutoff, whilst producing resonance around the cutoff
- Resonant Bandpass - excludes frequencies outside the 'band' centered around the 'cutoff', whilst producing resonance.
- Lowpass - excludes frequencies above cutoff
- Highpass - excludes frequencies below cutoff
- Bandpass - excludes frequencies outside the 'band' centered around the 'cutoff'
- Band-Reject - excludes frequencies inside the 'band' centered around the 'cutoff'
- Notch - similar to Band-Reject, but with a narrow bandwidth

4.3 Amplifier



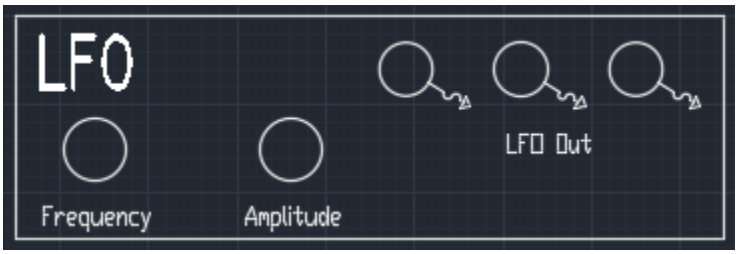
The amplifier effectively controls the output volume of the synthesiser, by changing the amplitude of the input signal, and sending it to the *audio output*.

Note: *if the amplifier has no input signal, no audio is produced.*

Gain Modulation changes the signal's amplitude based upon the amplitude of the LFO.

Envelope In applies the envelope to control the amplitude of the signal.

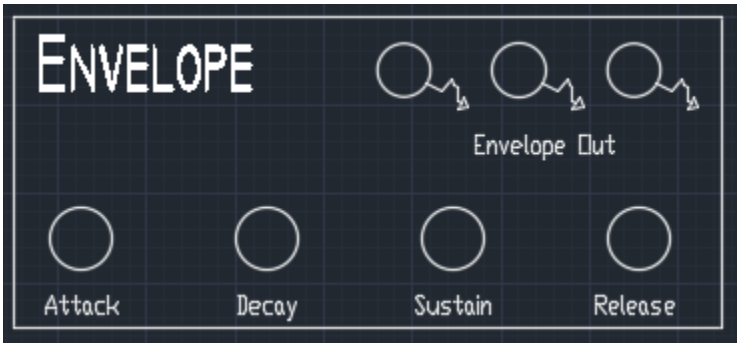
4.4 LFO (Low Frequency Oscillator)



The LFO produces a low frequency sine wave, for use as a control signal. Note that it can only be patched into connections with the "LFO input" arrow (an arrow with a sine wave).

The *Frequency* dial changes the sine wave's frequency, whilst the *Amplitude* changes its amplitude. These values are represented between 0 and 1, but are multiplied when patched to modules, depending on the module. In general, the frequency ranges from 0.1 to 10.1 Hz. However, in the filter for example, the amplitude is multiplied by 400, to change the cutoff in this range.

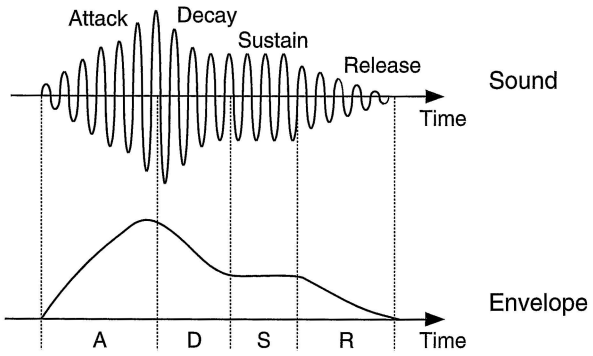
4.5 ADSR Envelope



This module outputs a MIDI controlled ADSR (Attack, Decay, Sustain, Release) envelope, outputting its amplitude.

Note that this can only be patched into connections with the "envelope input" arrow (an arrow with an envelope amplitude outline).

Figure 1: An ADSR Envelope [1, p. 39]



When a note is pressed, the envelope is triggered. Whilst the note is held, the envelope is kept at sustain. When the note is released, the release period begins.

The *Attack, Decay, and Release* dials control the length of each of these periods, from 0 to 1 second.

The *Sustain* dial controls the percentage of the input note's velocity (how hard the key is pressed), that the sustain period's amplitude is. This ranges from 0 to 100 percent.

5 Tutorial

To create a simple sound, patch the Oscillator's *Signal Out* to the Amplifiers *Signal In*. Now press a key on the MIDI controller to play a note.

Patch the Envelope's *Output* to the Amplifier's *Envelope In*. Now, when a key is pressed, the note is played, but there is silence otherwise. This note can be held down to keep the envelope in "sustain".

Try adjusting the Envelope's controls to adjust it's behaviour. Increasing the *Attack* makes it take longer to rise, for example. Decreasing the *Sustain* reduces the amplitude whilst the note is held.

Hold down the Oscillator's *Waveform* button to change between waveforms. Release this as soon as you hear "static" (whilst holding a key on the keyboard). This is pink noise.

Next, disconnect the Oscillator to Amplifier patch. Now connect a patch from the Oscillator to the Filter, and from the Filter to the Amplifier.

The default filter is a Moog-style resonant lowpass, which surpresses frequencies above the cutoff, whilst creating resonance.

This will create a sound similar to that of wind.

Now patch the LFO's *Output* to the Filter's *Cutoff Sweep*. This will add a subtle "pulsing" effect to the sound.

By adjusting the LFO's controls, this effect's behaviour can be changed. Using a high LFO *Amplitude* will make this effect more prevalent.

References

[1] Martin Russ. *Sound Synthesis and Sampling, 2nd Edition*. Focal Press, Linacre House, Jordan Hill, Oxford, OX2 8DP, 2004.