

Assignment #2: Digital Polyphonic Synthesizer

Luís Guimarães up202204188@up.pt

Introduction

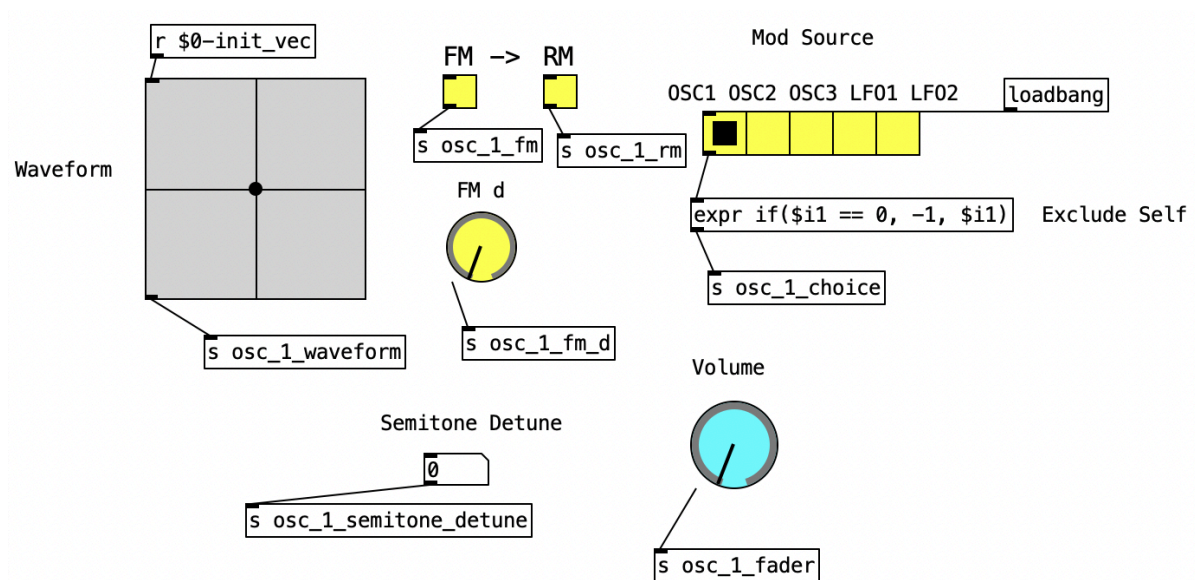
The goal of this project was to create a polyphonic synthesizer as a Pure Data patch, using a set of concepts mentioned during the classes.

The developed synthesizer includes:

- **3 oscillators**, each featuring **vector synthesis** (for the definition of the waveform);
- An **ADSR** envelope;
- **16 voices** for the 3 oscillators;
- **2 LFOs**;
- **FM and RM modulation** (processed sequentially: FM → RM) for each oscillator;
- An effects section which includes **spectral gating** and **reverb**.

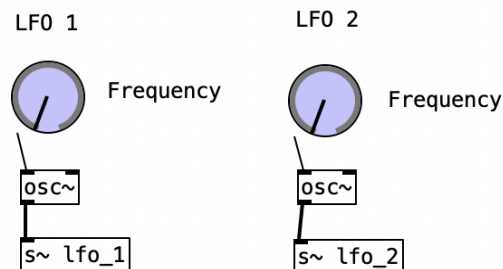
Implementation

- Each of the 3 oscillators looks like this:

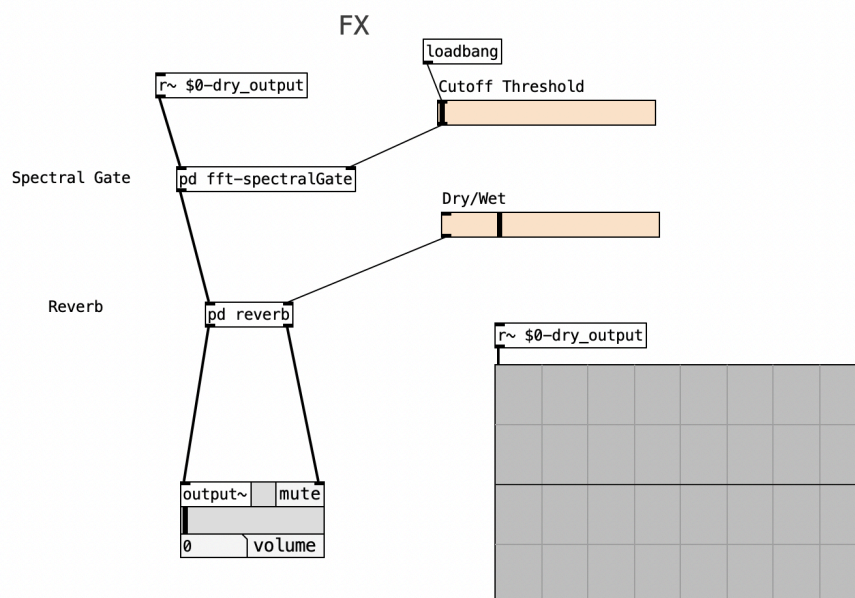


- The 16-voice polyphony happens to an abstraction named *oscillators*. This abstraction generates the signal for each of the 3 oscillators and applies the ADSR envelope before sending the output to its [outlet~]. Their parameters are global.
- The 2 LFOs are simply sine oscillators with a frequency knob:

LF0s



- Then, generating the signal for each oscillator, if the FM, RM or both toggles are enabled, the respective modulation is computed based on the selected source. The modulation source can be one of the following: the other 2 oscillators (pre-fader) or the 2 LFOs. If both FM and RM are enabled, they are computed sequentially (FM → RM).
- Finally, an effects section receives the multiple voices as a single input. The effects included are spectral gating - frequency components above a certain threshold are removed - and reverb - a mixture between dry and wet signals based on a value provided by the user:



Conclusions/Drawbacks

- The synthesizer can generate a plethora of sounds with different levels of complexity and “aggressiveness”. Because of how accessible modulation (FM, RM or both) is, it is easier to achieve more “aggressive” and dissonant sounds.
- Regarding the development, implementing polyphony took a lot of work. By trying to fix a given problem, another one would arise. The synthesizer started as monophonic, and after most features were implemented, it was turned into polyphonic. In retrospect, starting with implementing a polyphonic synth and developing more features from there, could potentially result in a smoother development experience.
- Performance-wise, the patch can be further optimised:
 - Even when modulation (FM, RM) is disabled, they are still computed and later discarded. This also happens with the oscillators when their faders are set to 0. Ideally, they should not be processed at all.
 - The latency is too high and not ideal for real-time performances. The issue mentioned above may be the culprit.