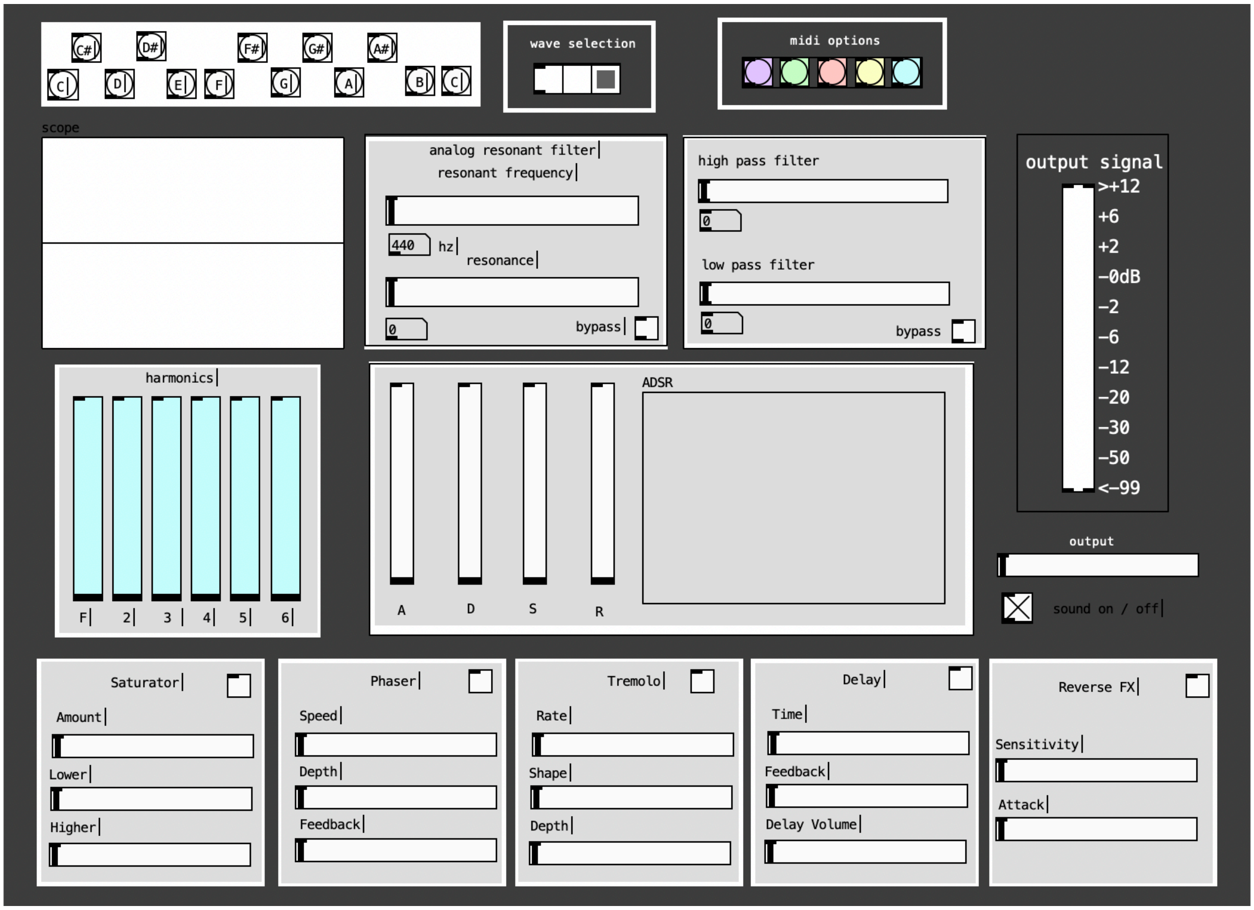
Synthesizer Report

Andrea Sofia Vallejo Budziszeswski
Computational Music Creativity
Jan, 2023

Main patch:



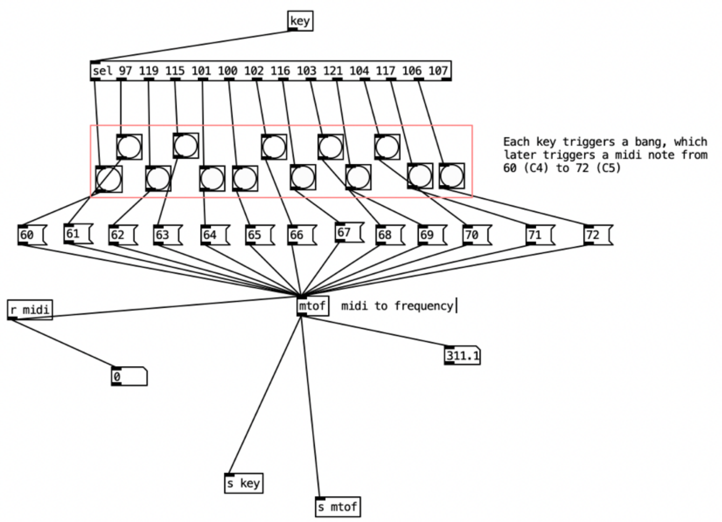
The synthesizer features a diverse selection of waveform options, including sine, triangular, and sawtooth waves. It also offers the ability to create additive synthesis by adding harmonics. Furthermore, it provides the user with a range of effects to enhance the audio output. The user can play notes with the use of their computer keyboard or the built-in virtual piano interface.

Signal flow:

Components:

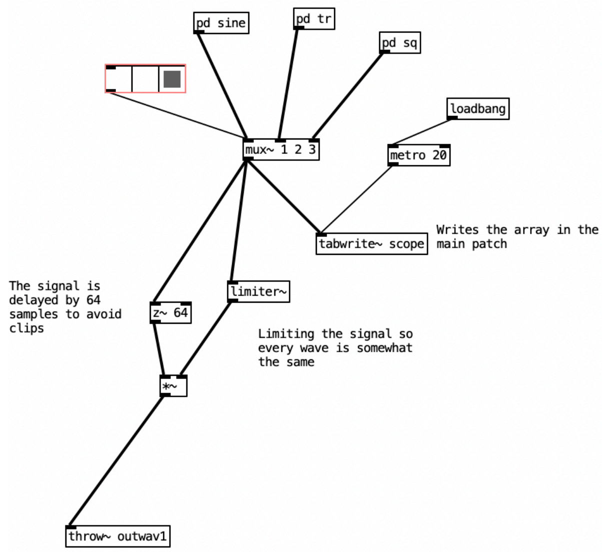
1. **Note selection, OSC, Harmonics and Wave selection**

   1.1 Note selection



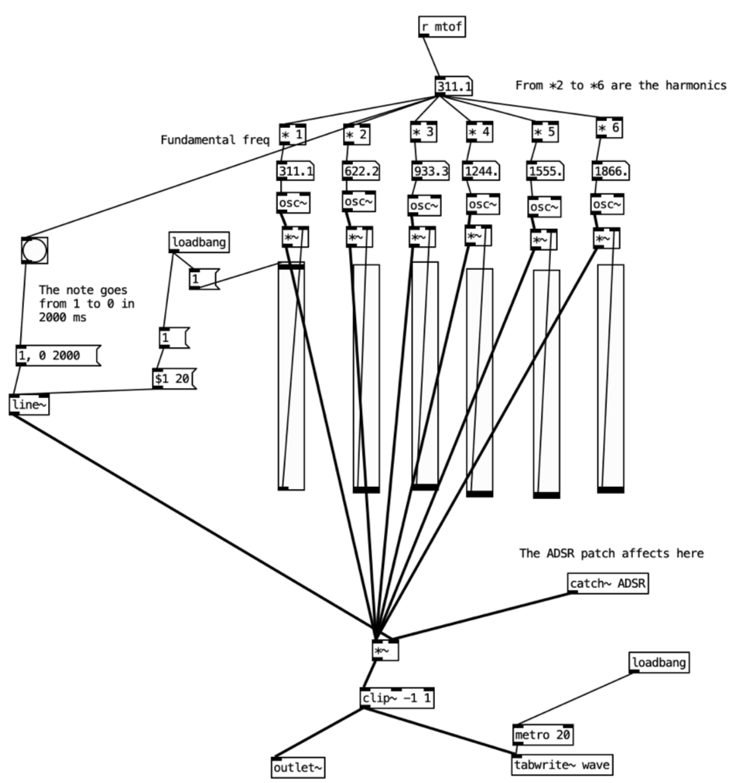
To enable the utilization of the computer's built-in keyboard, the key function was implemented. The designated keys (97, 119, 115, 101, etc.) were programmed to trigger a bang upon user input, which in turn activates a MIDI message containing the corresponding MIDI note number. The resulting MIDI number is then converted to a frequency using the mtof function before being routed to the output.

1.2 Wave selection



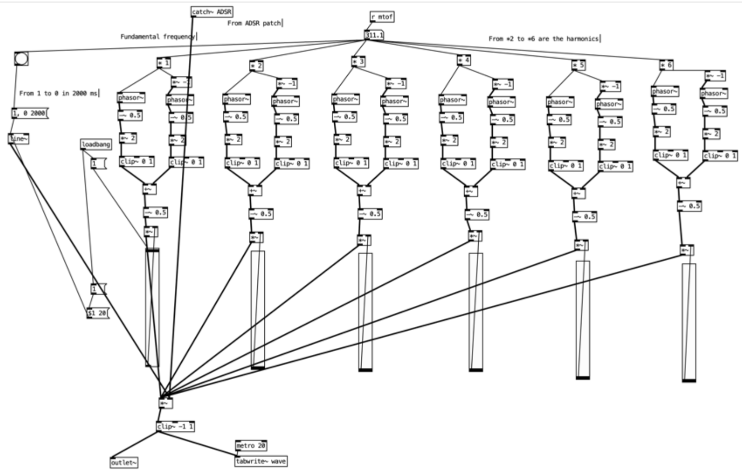
I used a multiplexer to be able to select the wave type. I also used a limiter in the output so I could normalize the signal.

1.3 Sine Wave



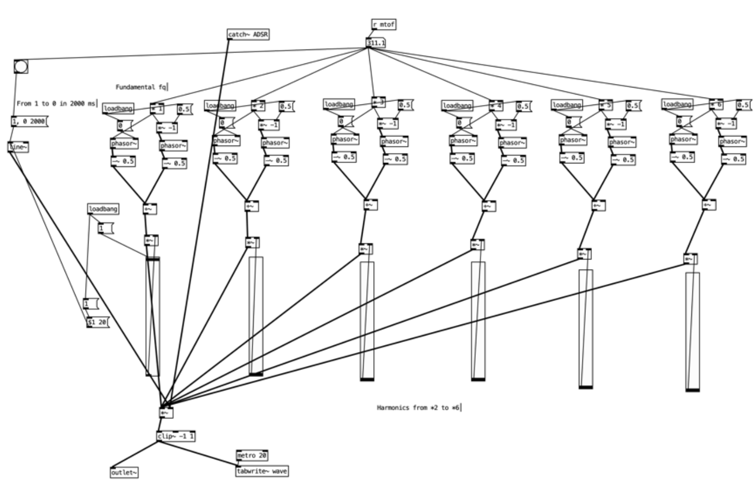
The sine wave is the easiest wave to generate, it is generated by an oscillator using the frequency selected from a note selection patch. The output is then multiplied and sent to an outlet. To create harmonics, the fundamental frequency is simply multiplied by a certain value. The amplitude or multiplication value can be adjusted using faders

1.4 Triangle Wave



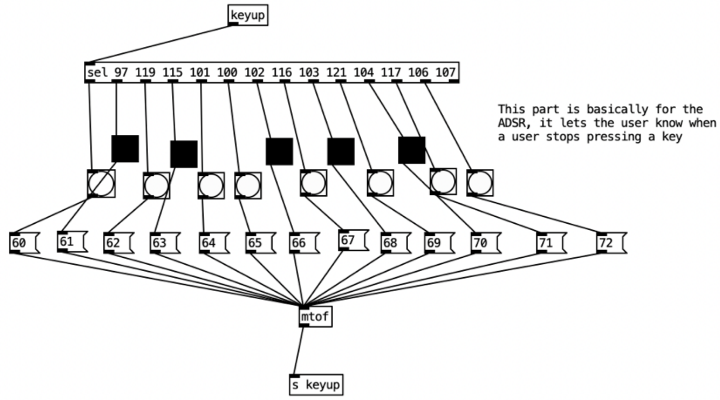
The input and output for the triangular wave is the same as the one. The main difference is in the method of generation. Instead of an oscillator, a phasor is used to create ramps whose values range from 0 to 1. These ramps repeat in cycles, depending on the input frequency. A positive frequency value will generate upward ramps, while a negative value will generate downward ramps, which gives the signal the form of a triangle.

1.5 Saw tooth

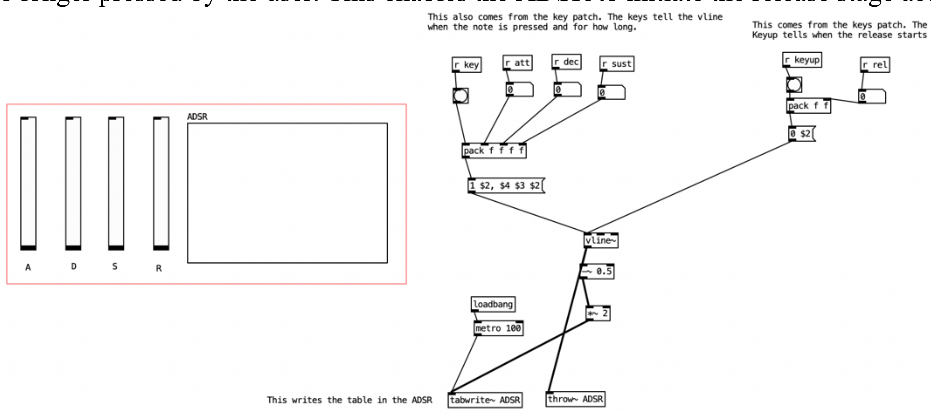


The inputs and outputs are the same as above. The difference is, the sawtooth wave, on the other hand, is an asymmetric waveform with a sharp transition from the peak to the trough. It has both an infinite number of odd harmonics and a fundamental frequency.

2. ADSR

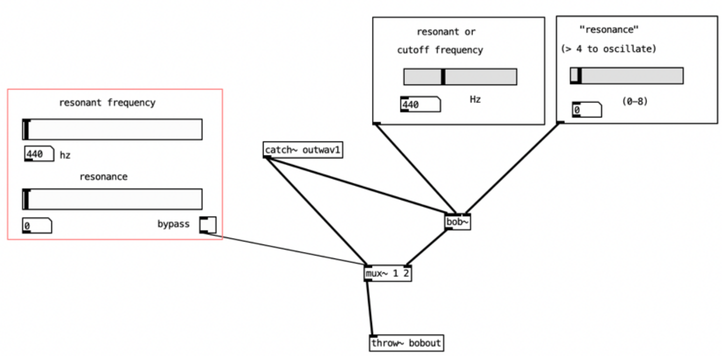


In order to achieve the release functionality of the ADSR, the keyup event was employed to detect when a key is no longer pressed by the user. This enables the ADSR to initiate the release stage accordingly.



Subsequently, the audio signal is routed to a high-precision ramp generator, known as the vline. The vline applies a target value, a time interval in milliseconds, and an initial delay (also in milliseconds) to the audio signal. By utilizing the vline in conjunction with the faders, the user can precisely control the attack, decay, sustain, and release characteristics of the audio signal.

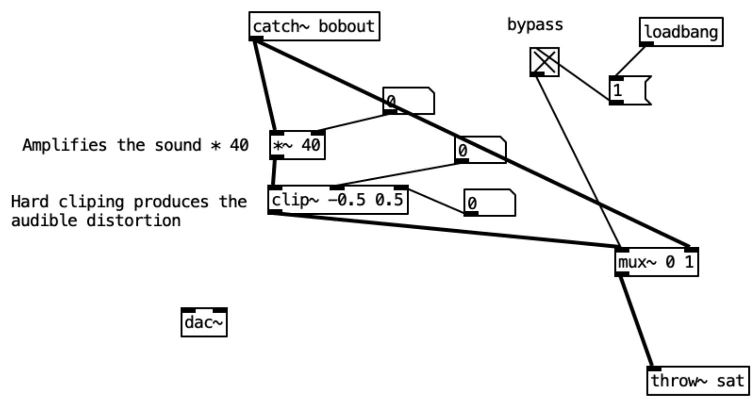
3. Analog Resonant Filter



Analog resonant filters are commonly used in electronic music synthesizers and other audio processing equipment to create a variety of tonal and timbral effects. They can be used to shape the frequency content of a sound, to create a sense of movement or animation in a sound, or to create a sense of tension or excitement.

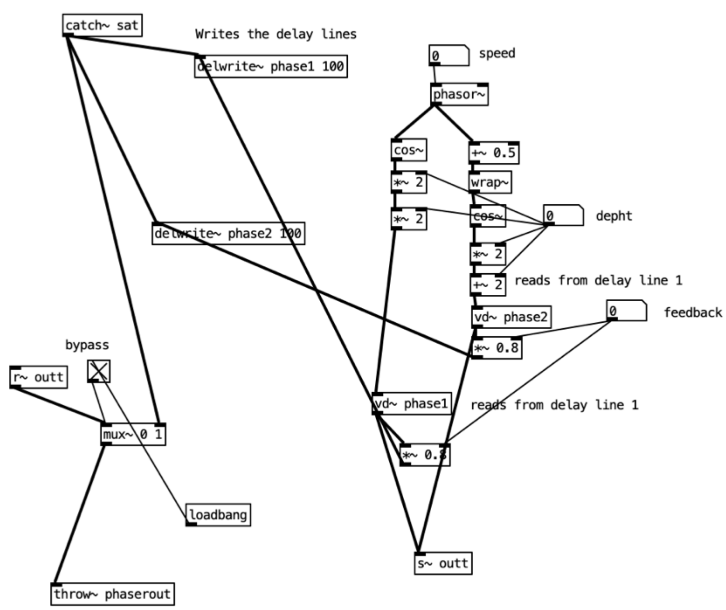
An analog resonant filter is a type of audio filter that uses analog circuits to create a resonant peak at a specific frequency, also known as the cutoff frequency or the resonance frequency. This peak can be adjusted in terms of Q or bandwidth, which refers to the width of the peak.
For this patch, I used “bob”, which is a numerical simulation of the Moog analog resonant filter. By default, it does one step of 4th-order Runge-Kutte integration per audio sample. This works OK for resonant/cutoff frequencies up to about 1/2 Nyquist. To improve accuracy and/or to extend the range of the filter to higher cutoff frequencies you can oversample by any factor - but note that computation time rises accordingly. At high cutoff frequencies/resonance values the RK approximation can go unstable.

4. Saturator:



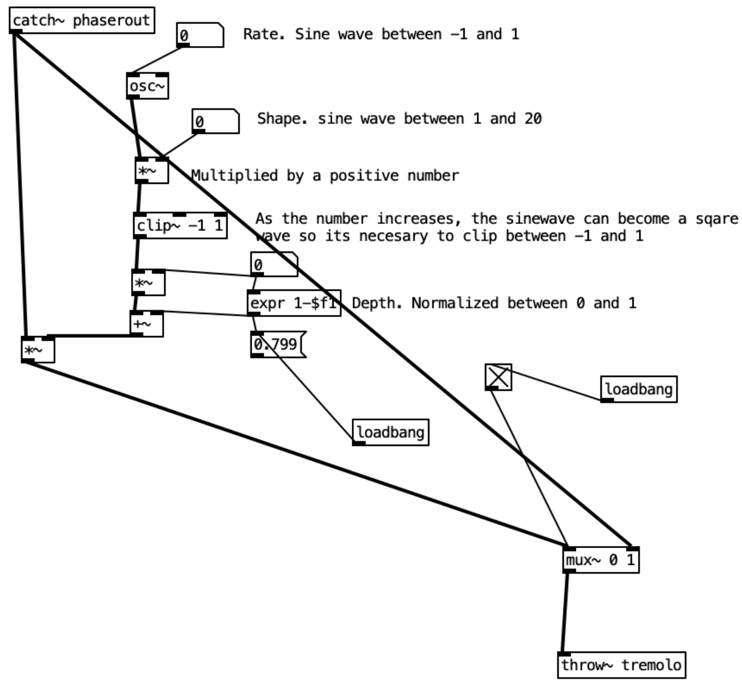
A saturator is an audio processing tool that is used to add harmonic distortion to a sound signal. This can be used to add warmth and depth to a sound, or to make it sound more intense or "overdriven." In this synthesizer, the sound is distorted by amplifying it before applying hard-clipping. The result sounds close to what you’d get if you raised the gain of the input of your soundcard to make it clip.

5. Phaser



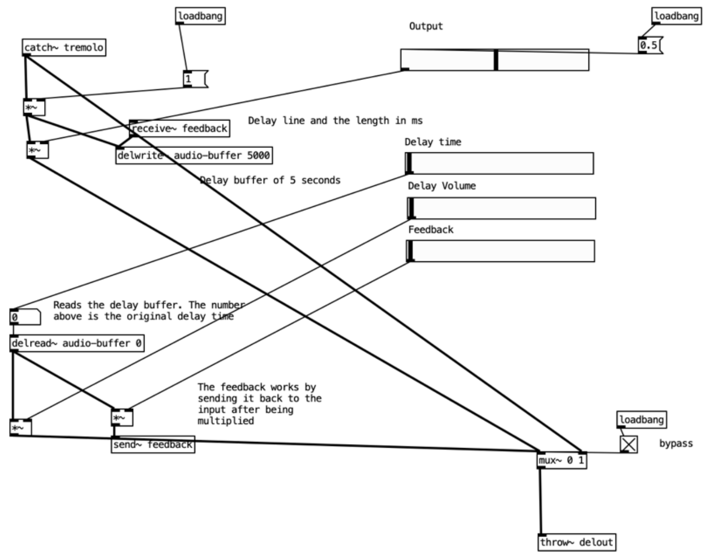
A phaser is a type of audio effects processor that creates a sweeping or "whooshing" sound by applying a series of filters to the audio signal. The filters are typically modulated (often by an LFO) which creates the sweeping effect. The phaser effect can be used to create a sense of movement or depth in a sound, or to add a unique character to a sound. In this patch the 3 sliders to the right control the speed of the oscillation, it’s depth, and the amount of feedback in the delay lines. Please not that these sliders are not supposed to be moved while playing (you’ll hear some unwanted noise)

6. Tremolo



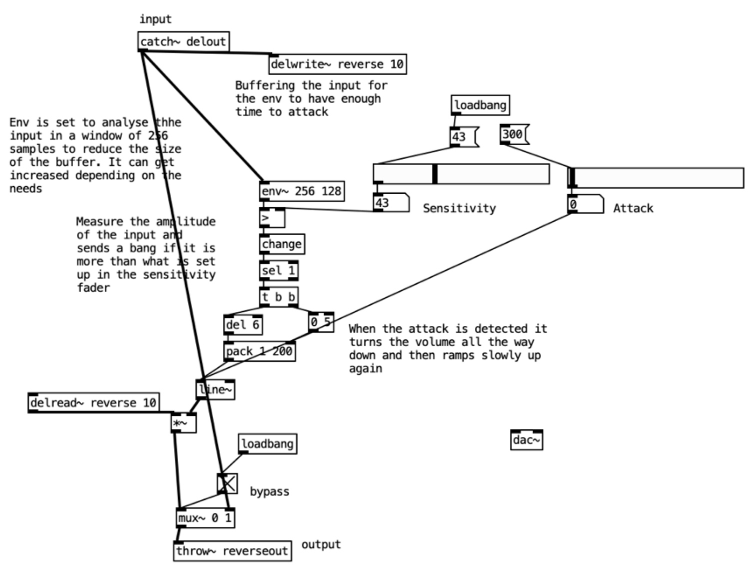
A tremolo effect is an audio effect that modulates the amplitude (volume) of a sound signal. It creates a periodic variation in the volume of the sound, giving the impression of a pulsing or "trembling" effect. The modulation is often achieved by using an LFO (low frequency oscillator) to vary the gain of the sound signal. This effect can be used to add movement and interest to a sound or to create a sense of urgency or tension. Some examples of how tremolo could be used are in guitar amplifiers, some electronic organs or in some software effects. In this patch, it gets the signal, and it gets multiplied by the output of the oscillator. Then the output is tuned, and the input signal is multiplied by a number between 0 and 1 (depth). As the value of the oscillator approaches the minimum value, the signal gets multiplied by a smaller value and it diminishes the volume. And as the value increases, the volume increases.

7. Delay



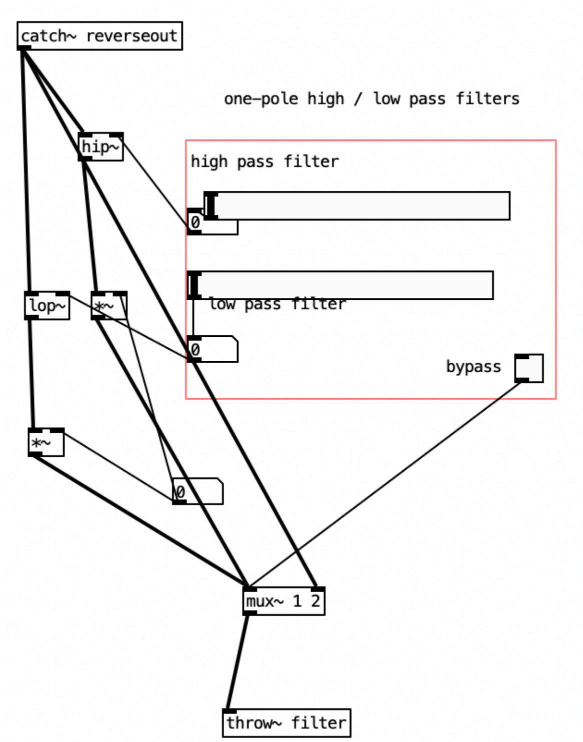
A delay effect is an audio processing tool that creates an echo-like effect by repeating a sound signal at a later time. The delay effect is achieved by recording the original sound and then playing it back at a later time, creating the illusion of multiple sounds happening at once. The delay time, or the time between the original sound and the repeated sound, can be adjusted to create different effects. Short delay times can be used to create a sense of space and depth in a sound, while longer delay times can be used to create a sense of movement or to create a distinct echo effect. In this synth, the input signal goes to the delay buffer with a delay of 5 seconds, which at the same time is receiving the feedback from the output signal, after it is multiplied.

8. Reverse



The reversed tape effect is an audio processing technique that simulates the sound of a recording being played backwards on a tape machine. The effect is typically achieved by recording a sound and then reversing the audio file so that it plays backwards. The reversed sound is then mixed with the original sound, creating a unique and distinct effect. The reversed tape effect can be used to create a variety of interesting and unusual sounds and is often used in music production and sound design to add a sense of mystery, confusion, or surrealism to a sound. The effect can also be used to create a sense of backwards motion, as if time is running in reverse. There are different ways to achieve the reversed tape effect, it can be done with software effect, by physically reversing a tape or by using a hardware effect unit. The reverse effect can be applied in a subtle way to add a sense of depth to a sound, or it can be used in a more pronounced way to create a striking and unusual sound. This reversed tape effect is based on an envelope follower. The value of the threshold and the duration of the attack can be set up (as ADSR). Longer attacks and long-lasting notes will make it sound like some sort of violin, and shorter attacks with fast notes can make it sound like a reversed tape.

9. Filters

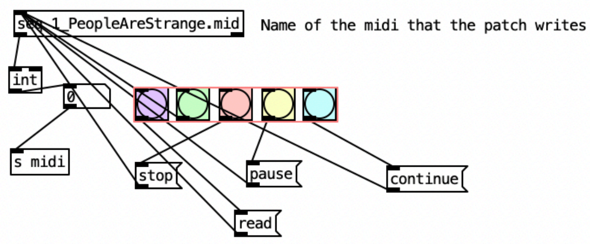


For the HPF and LPF I used “hip” and “lop”, which are a one-pole high/low pass filter with a specified cutoff frequency. The fader, which is connected to the right inlet, lets you control the cutoff frequency.
A one-pole high pass filter is a type of high pass filter that uses a single pole in its transfer function. It is a simple and basic form of high pass filter that is often used in audio processing applications. A one-pole high pass filter uses a single pole, which is a mathematical term that represents the slope of the filter's frequency response. The slope is the rate at which the filter reduces or attenuates the frequencies below the cutoff frequency. The steeper the slope, the more aggressive the filter is in removing low frequencies. One-pole high pass filters have a relatively gentle slope, meaning that they will not remove low frequencies as aggressively as a multi-pole high pass filter would. This can be desirable in situations where a more subtle or gradual reduction of low frequencies is needed. A one-pole high pass filter can be implemented using an active or passive circuit, or with digital signal processing techniques. These filters are common in audio processing applications such as equalization, noise reduction, and audio effects.

Future plans

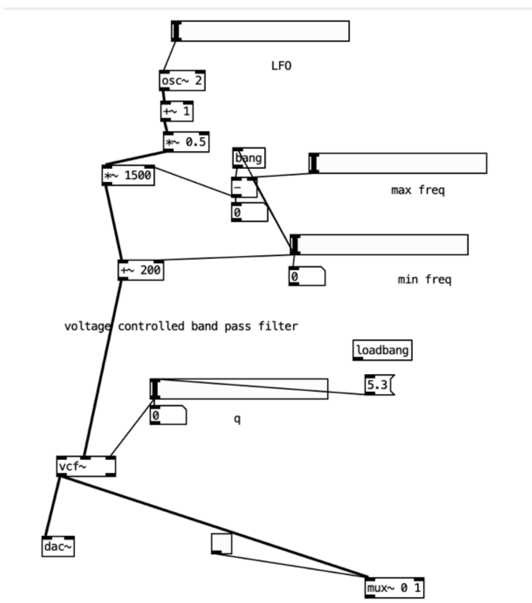
1. MIDI Options

As part of my implementation, I have developed several subpatches, one of which is the MIDI options section. I have successfully implemented the majority of its functionality, allowing the user to load, play, pause, stop and resume MIDI files, while simultaneously adjusting the sound parameters within the synthesizer. However, I have encountered an issue with the object used for reading MIDI files, which interprets the velocity of the notes as the MIDI note number. This results in an abnormal sound output. The current workaround is manually editing the MIDI file within the patch, however, this is a time-consuming and inefficient solution. I am currently seeking a more appropriate solution to rectify this issue."



2. Automatic Wha Wha

I have attempted to create a patch that simulates a wha-wha pedal effect by utilizing a low-frequency oscillator (LFO) to continuously sweep the frequency range set by the user. Unfortunately, the current implementation is not functioning as intended and the reason for this is currently undetermined. I will continue to investigate and make efforts to rectify the issue and implement it in the next deliverable.



3. Sequencer and Looper

I plan to integrate a sequencer within the synthesizer, in order to provide more advanced composition capabilities. Additionally, I am also considering the implementation of a looper function. This feature would be particularly beneficial for live performance scenarios, allowing for the creation of dynamic and evolving soundscapes.