Aidan Baker

MUMT307 Final Project

Physical Modelling String (pluck) instrument in Max MSP / Gen

Objective / Outline:

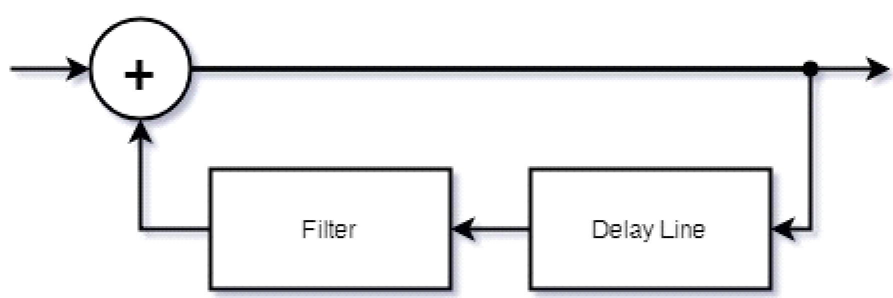
After a few failed endeavours into other potential fields of study for the final project, I finally settled on doing a project related to physical modelling and simulation.

The main "scope" of the project that I decided upon developing was a somewhat beginner look into the world of physical modelling systems. I intended to have two systems; one for modelling of a string, and one for modelling of the resonating body which the string vibrates over (similarly to how most acoustic stringed instruments work, basically). Initially, I designed a streamlined version of these two systems (see the *development* section), and then looked further into the algorithms that I was utilising to find the maxmum potential of implementing adjustable parameters within the algorithms. Secondly, I implemented MIDI functionality with the delay times syncronizing with the corresponding pitches. Afterwards, I made a nice little GUI with some additional features to wrap things up.

Research:

String synthesis:

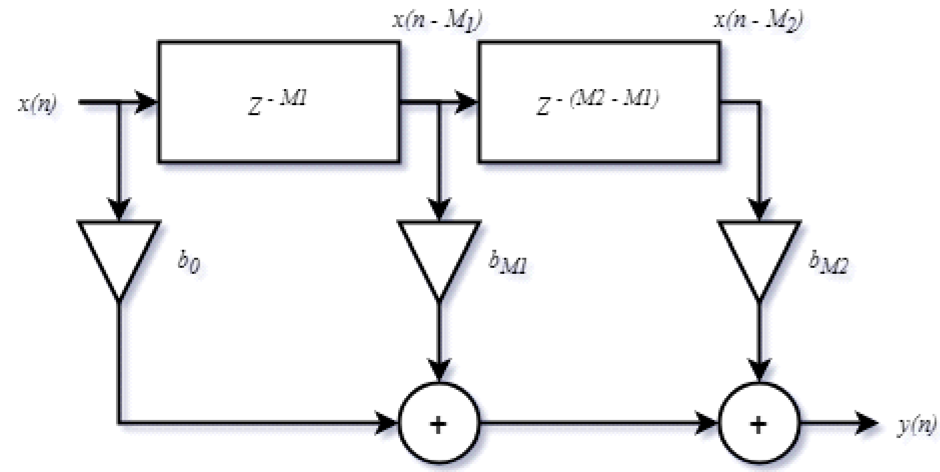
For the backbone of the project, I needed some kind of system for generating somewhat realistic sounding plucks for the first phase of processing. The first algorithm I looked into was the karplus-strong string synthesis algorithm, where short noise pulses are fed through a system of feedback delays which cause comb filtering. A low-pass filter is applied to the signal upon each feedback loop to help with dampening[1]. When used correctly, the times of these delays are able to be modulated to a degree of accuracy where the comb filtering has a sense of pitch. The following diagram represents a simple implementation; lacking mathematical notation, for simplicity's sake.



[1] Burk, Phil, et al. “Chapter 4: The Synthesis of Sound by Computer: Section 4.9: Physical Modeling.” *Music and Computers*, Columbia University, 2011, sites.music.columbia.edu/cmc/MusicAndComputers/.

Tapped Delay Lines

For improving the sound of realism in the latter half of the processing chain (specifically the resonating body model) I looked into working with tapped delay lines. Utilising tapped delay lines, one can simulate multiple sources of reflection from within the processing chain, where a system would extract the signal from within the processing chain, performs scaling, and mixes it with other tap delays to create a new signal[2]. The following diagram represents a possible feedforward combination of tapped delay lines, as well as this delay line's associated difference equation.



$$y(n) = b_0 x(n) + b_{M1} x(n - M_1) + b_{M2} x(n - M_2)$$

This system was ultimately utilised in the internal components of the resonant body model, where tapped delay lines are used shortly before a feedback element to further the sense of internal reflection.

Diffusors:

To actually have some sort of aspect of reverb within the project (my favorite kind of effect to make in Max), I decided to look into ways to implement a somewhat simple sound diffusion system to help make the sound a tad bit more realistic (in any "real" room that this "real" instrument would be played in, there would almost always be at least a bit of reverberation within the space). To look into developing one, I worked out how diffusion systems generally work; a series of tapped delays in series with feedback elements implemented in[3]. With even a few different delay times being fed through one another, the signal created through the resulting feedback would give the feeling of having an

---

[2] Smith, Julius O. "Acoustic Modeling with Digital Delay." *Acoustic Modeling with Delay*, CCRMA, 2010, ccrma.stanford.edu/~jos/pasp/Acoustic_Modeling_Digital_Delay.html.

[3] Costello, Sean. "ValhallaDelay: The Diffusion Section." *Valhalla DSP*, 13 June 2019, valhalladsp.com/2019/06/13/valhalladelay-the-diffusion-section/.

impercievable amount of reflections (which is ultimately what the diffusors are for at the latter ends of reverb impulses)[4].

Development:

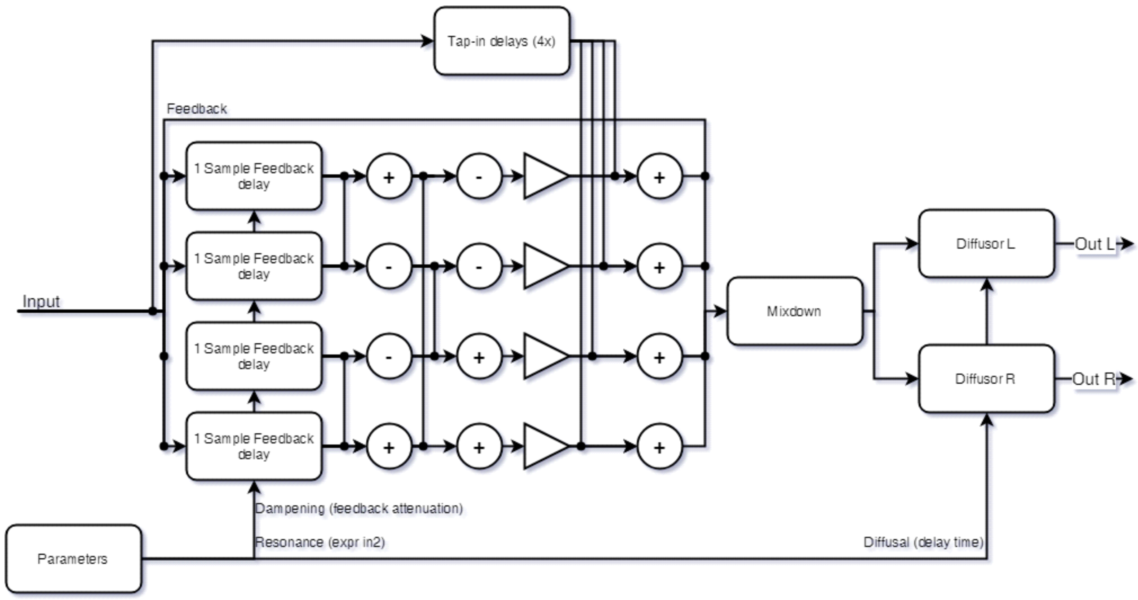
Modelling Components:

The main component of what makes this instrument sound the way it does is the string waveguide system. The string waveguide system in this case is quite similar to the way that the karplus-strong string synthesis algorithm works, which was explained above, the difference here is that the use of multiple “coloration” parameters are taken into acount (pluck and pickup, specifically), where these two parameters are mixed together with the incoming MIDI pitch value to create minor discrepencies with the initial “karplus-strong-esque" sound. Another difference is the amount of delays utilised in the waveguide. Where a basic karplus algorithm would utilise a singular delay, this waveguide uses five, where two of these delays have short feedback elements, and the entirety of the system feeds back into itself (with dampening filters implemented as well).

This system is "excited" by the use of a single-sample impulse which is fed through a filter, this filter turns the single impulse into a more realistic "click" (this filter's cutoff is controlled by the system's "sharpness" parameter, and as it decreases the sound becomes more muted) which is then fed into the string waveguide and subsequently filtered.

After the string waveguide, a model of a resonating body is applied upon the signal next up in the chain. Most of the purpose of the resonating body is to add a small bit of sound diffusal (i.e a somewhat basic, short RT60 stereo reverb system) but also helps to improve realism of the sound with the resonation, with multiple parameters controlling internal delay times and mix values, such as the dampening and "material" parameters. The resonating body flowchart can be seen below:

[4] Berardi, Umberto, and Higini Arau. *Increasing Reverberation Time with Diffusers: a New Acoustic Design for More Sustainable Halls*. International Symposium on Room Acoustics, 9 June 2013.



Misc:

Near the end of the signal processing chain, there are two state-variable filters which are simply controlled by a cutoff knob as one of the parameters for some additional control on the sound. The gain, and resonance values however, are constants, and are set as thier defaults within the internals of the system.

I have known Max MSP to occasionally cause issue in relation to DC offset of signals when Gen is utilised (likely more of an issue on my side than anything else, as Gen is a more "true to life" representation of doing DSP than anything else in max MSP, and doing that generally requires a large amount of precision). I was sure to try and account for DC offset issues within the Gen patches themselves, but ultimately was still met with DC offset issues. To correct this, biquad filters at the end of the processing chains for L / R were utilised with prespecified coefficients to create a DC offset filter (although not an *ideal* DC offset filter; the filter I got working reliably enough begins cutting off frequencies at around 70 Hz, and an *idealized* one utilizing the biquad filter model was unstable, likely due to the reason that for a steeper slope using biquad filters, a high Q value needs to be implemented which can create more problems than it fixes)[5].

Usage:

Parameters:

Pluck / Pickup: The pluck and pickup parameters control general coloration of the sound. The values generated by the control knobs feed into the waveguide system, and are closely related to eachother.

[5] “DC Offset and Audio Filtering.” *SoliCall*, SoliCall Ltd., 2019, solicall.com/dc-offset-and-audio-filtering/.

Sharp: The sharpness parameter controls the filtering of the incoming impulse excitation, where a lower value creates a more low-end sound, as the waveguide isn't able to produce frequencies beyond that of the incoming impulse.

Damp: The dampening parameter controls some of the feedback attenuators of the string waveguide, as well as feedback attenuators within resonating body model. the dampening parameter does not, however, affect the feedback levels for the late diffusion network.

Material: The material parameter controls how the resonaitng body model effects the sound, and is arguably the most noticable parameter for the resonating body side of the plugin, as adjusting it controls incoming signal values within the feedback delay network subsystem. Adjusting it affects comb filtering slightly, and can give it a more "metallic" sound.

Diffusal: The diffusion parameter controls the delay times within the stereo diffusion system applied upon the signal *after* the feedback delay network (the layout of how this works is visible above). The parameter does not adjust the delay times *directly,* but instead feeds into multipliers which are then fed into 6 different delay times (three on each channel, with all three adjusted slightly for a stereo effect).

Filter: The filtering parameter is the simplest to explain parameter, where the value corresponds to the cutoff frequencies of two filters at the end of each of the stereo signal chains. The gain and resonance input values are set to constants (a low resonance, a gain identical to the incoming signal), for a neutral sounding filter.

Other Controls:

Keyboard: On-screen keyboard as an alternative to incoming MIDI signals (however, this system *is* midi capable, and will accept incoming midi signals).

Voices: Controls the internal [poly~] object's voice amount. Default is mono.

Octave: Octave-level note offset.

Detune: Semitone-level note offset.