Ping Pong Delay Plugin Report

Matthew Williams

21479304

University of West London

Lecturer: Peter Dowsett

MSc Digital Audio Engineering – Audio Programming 2

Abstract

Delay is one of the most widely used effects in modern music and some form of it can be found on almost track. It is therefore important to translate the effect into plugins to adapt to a modern music production workflow. This report details the methodology and testing of a tape style ping-pong delay effect plugin to work inside a digital audio workstation.

The report focuses on both the design of the plugin's DSP elements as well as the implementation with JUCE using C++. Through a thorough testing regime, the plugin's functions and parameters are all tested to ensure that a highly operational plugin has been designed. Future work involving the plugin is then commented on.

The repository for the project can be found in Appendix 1.

Table of Contents

List of Figures

1) Introduction

1.1) Overview

Delay is one of the most widely used effects in modern music. It consists of repetitions of the dry signal at certain set intervals. As with most musical effects, delay has its roots in analogue circuitry with the original effect being a result of a tape recorder's read head being placed just after the record head. Now that music production is mostly digitised, it is important to translate this effect for use with modern recording equipment in the form of plugins.

This report details the methodology and testing of a ping-pong style delay plugin for use in a DAW.

1.2) Report Structure

Firstly, this report will highlight some background theory on delay effects. Following from this are details of the methodology used to make the plugin using the JUCE framework with C++. Finally, the report shows the testing procedure to ensure that the plugin works as planned followed by some final conclusions from the project.

2) Background Theory

The delay effect consists of repetitions of the input signal at certain set intervals. To achieve this effect, the audio data from the past must be recovered to be played with the dry signal. In order to recover this past data, it must be stored for at least as long as the repeat interval.

To do this, a very common method is to use a circular buffer. The buffer stores audio data that is passed into it. The write index is the position within the buffer that the dry signal is being written to and this increments for each sample. When this buffer is full, the next data passed to it overwrites the data at the beginning of the buffer, hence the name circular buffer. The read index is then set to be a certain distance behind the write index. The distance between them is the interval between repeats. Feedback can then be applied, inputting the read index data as well as the dry data into the write index value.

In a ping pong delay, the signal from the left channel is written to the right channel's buffer. The feedback from the right channel is then passed to the left channel's buffer. The same happens for the right channel but instead its data is passed to the left channel. This creates a stereo effect where the signal bounces from ear to ear.

3) Methodology

3.1) Plugin Design Approach

The original design of this plugin can be seen in Appendix 2. This plugin includes multiple delay lines and a modulation effect delay, however, the GUI for the different chains that the signal went through seemed far too convoluted for the user to easily use. Therefore, a different approach was taken to ensure that the plugin was more intuitive for the user.

Having an appreciation for analogue electronics, the approach for this plugin was to create an effect that emulates some of the well-known qualities that tape echoes exhibit. Notably, tape runs using motors which are not always able to reliably stay at a constant speed. This means the recording has a tendency to speed up and slow down gradually which results in an organic sound and can give a chorus type effect. Another quality of tape is that it does not record the signal exactly as it comes in, adding some saturation as well as attenuation to certain frequencies depending on the speed that the tape is being recorded at.

Figure 1 - Korg SE-500 and Roland Space Echo RE-201 Tape Delays

An attempt to replicate some of these qualities in digital form has been made. The Korg SE-500 and Roland Space Echo RE-201 have been used as the main inspiration for the choice of plugin parameters. These can be seen in figure 1 and the include the use of multiple taps and a tone control, however, some variations have been made in the plugin.

3.2) Signal Flow

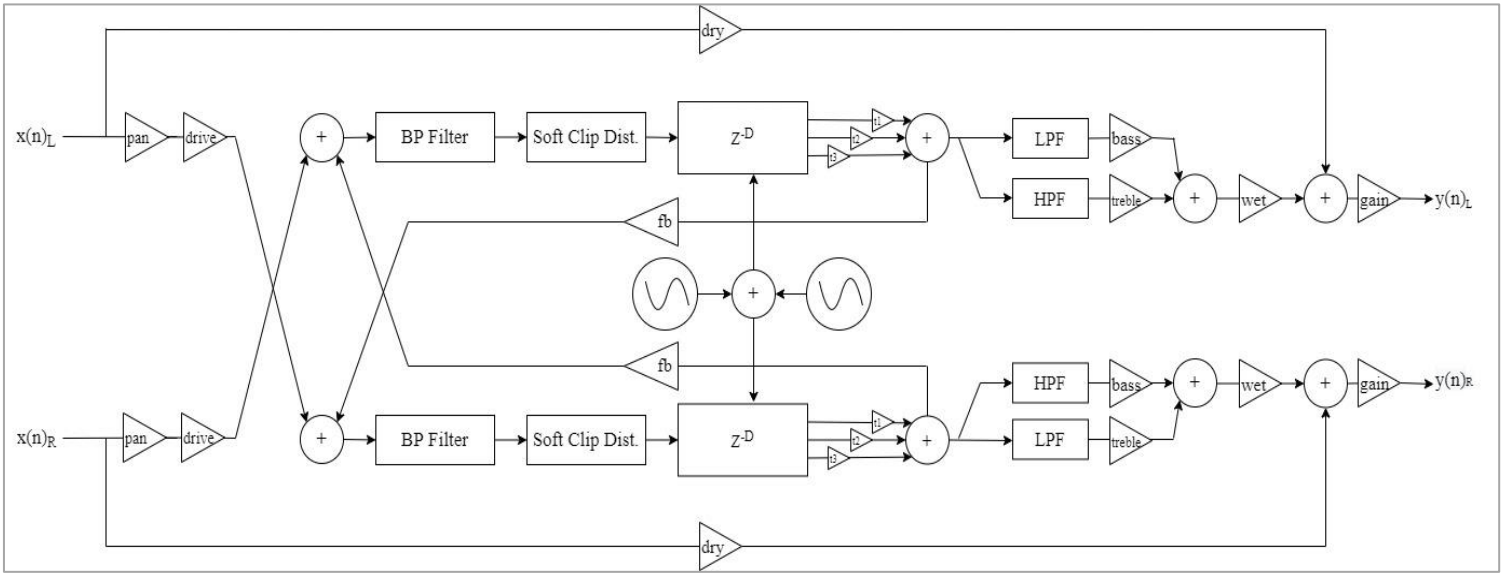


Figure 2 - Signal Flow of Delay Plugin

3.3) Delay Class

The delay class contains delay lines, functions to read and write to the buffers, and all of the parameter variables. Most of the class is kept private in order reduce the issues that can arise from user access of the variables and functions. The only parts of the class that are publicly accessible are the *getters* and *setters* for the variables and the main delay process.

3.4) Filter Class and LFO Class

Different filters are used within the signal flow. To make a filter, multiple different variables are required for storing previous values as well as the coefficients to carry out the filtering. To remove this from the delay class, a separate filter class is used. Inside the filter class, the previous values of the input signal are stored. Also, the processes to calculate and apply the coefficients in a biquad arrangement are present in the class. Within the class, the filters are made with arrays for each variable to implement them in stereo.

Similarly, an LFO class has been made. This can be set with a frequency and amplitude to then return the corresponding value from a sine wave.

3.5) Stereo Processing

The ping pong delay is a stereo effect based on previous memory from each channel. Hence, issues can arise from combining the two channels and the data from each channel must therefore be kept separate.

To ensure that the plugin operates smoothly for both channels, variables that change on every sample, such as LFO values, need an array to duplicate the processes. Otherwise, certain processes can cause audible artefacts as a result of data being processed in a non-interleaved fashion.

3.6) Fractional Delay Line Processing

When changing the delay interval, quantisation to the nearest sample causes audible distortion. To overcome this, a fractional delay line with interpolation has been used. Fractional delay lines are able to estimate the value between samples to remove the noise from quantisation error.

To estimate the value, linear interpolation has been used. This takes the sample either side of the fraction and applies equation 1 to them, (Tarr, 2018).

$$x_{m+f_{rac}} = (1 - f_{rac}) \cdot x_m + f_{rac} \cdot x_{m+1} \quad [1]$$

Cubic interpolation could be used to improve the accuracy of the estimation. However, this is far more computationally inefficient and linear interpolation is sufficient in removing the audible artefacts. This then makes it possible to have an LFO modulate the delay time and for the user to change the delay time parameter whilst the plugin is running. Smoothly operating modulation of the delay length without is essential to creating a tape-like effect as it makes it possible to emulate the changes in motor speed.

3.7) Multi-Tap Delay Processing

To make the plugin a multi-tap delay, as would often be found on tape echoes, there is more than one read index on the same delay line. Each read index is at a different division of the delay interval. The taps can then each be individually enabled or disabled from the delay. To emulate the behaviour of tape, when a tap is enabled, it reads from that position as well as applying that read position to the feedback loop. Similarly, the taps are all applied at the same amplitude as each other.

3.8) Tape Echo Effects

To give the impression of a tape echo from the plugin, the delay time is constantly changing in response to two LFOs. One of the LFOs runs at 0.1Hz and the other runs at 1Hz to give a wow and flutter effect respectively. The amplitudes are set low to keep the effect subtle but still

noticeable. The motors of tape machines are more likely to have inconsistencies when running slowly, therefore, the amplitudes of the wow LFO is mapped to the delay time controls to cause a more pronounced effect when the speed is set slower.

In order to saturate the signal, as a tape recorder would, a soft clipping algorithm from Pirkle (2013) is used on the input signal and the feedback loop. This creates a subtle distortion that can be made more prominent with increased drive.

To emulate the frequency response of tape recording, a varying bandpass filter has been used. This changes centre frequency dependant on the speed control. Figure 3, from Jack Endino (2006) shows the frequency response to higher speed recording in red and lower speed recording in blue.

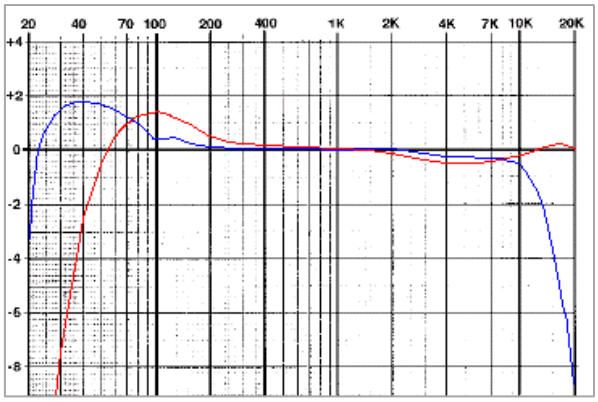


Figure 3 - Frequency Response to Different Tape Speeds

To emulate these curves the bandpass filter has a higher centre frequency when the speed is fast and a lower centre frequency when the speed is set to low.

3.9) Parameter Mapping

To map the parameters from the GUI to the variables in the delay class a ValueTreeState is used.

Parameters that are in the ValueTreeState can be attached to sliders that are in the PluginEditor. This means the slider values will automatically be sent to the ValueTree in the PluginProcessor and subsequently passed to *setters* in the delay class.

To ensure that the changes to the parameters are smoothly implemented, a simple first-order low-pass filter is used to removes any sonic artefacts from parameters' quickly changing value. The smoothing is done once during each buffer, apart from for the speed control which is very sensitive to changes and therefore is altered on each sample.

An advantage of using a parameter tree is that the states of the parameters are able to be saved. This is done by using various JUCE functions to copy the states into an XML file upon closing of the DAW and then recall them states when the file is opened again.

4) Results

4.1) Panning and Ping Pong Delay

The ping pong effect and pan parameter can be tested by panning the input signal completely to one side. For the effect to be determined successful, the signal must repeat itself in differing channels for each repetition. Figure 4 shows the output of the plugin with the pan set to left and right respectively. By viewing both channels lined up with each other, it can be observed that each repetition changes the channel that it outputs from. Therefore, the ping pong effect and pan can be assessed to be working.

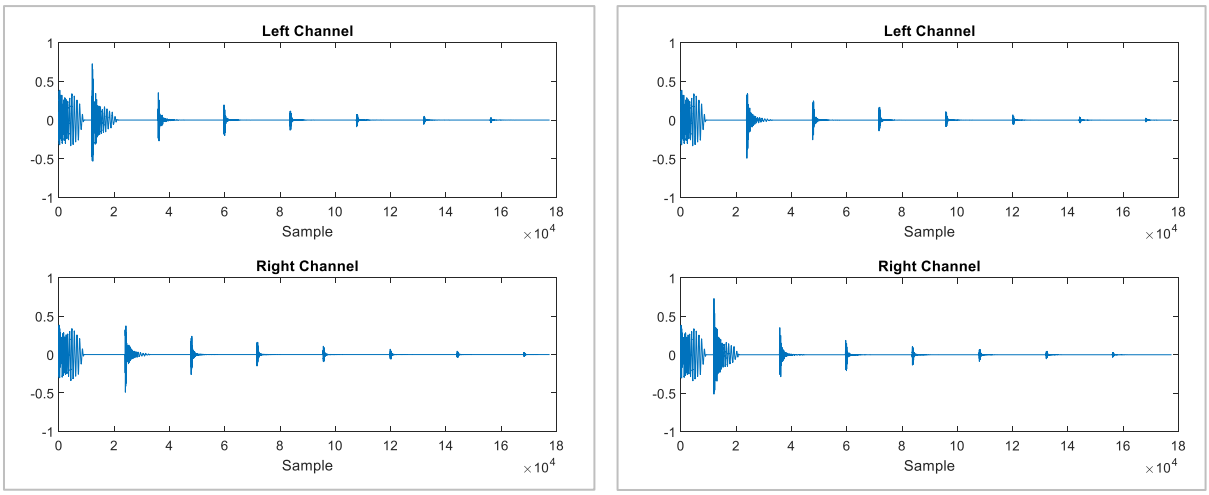


Figure 4 - Ping Pong Delay and Pan Test

4.2) Drive Parameter

The drive parameter changes the gain coefficient that is applied to the input value being written to the buffer. The drive parameter must therefore only change the output of the delayed wet signal and not the dry signal. The parameter has been tested at different values with no feedback.

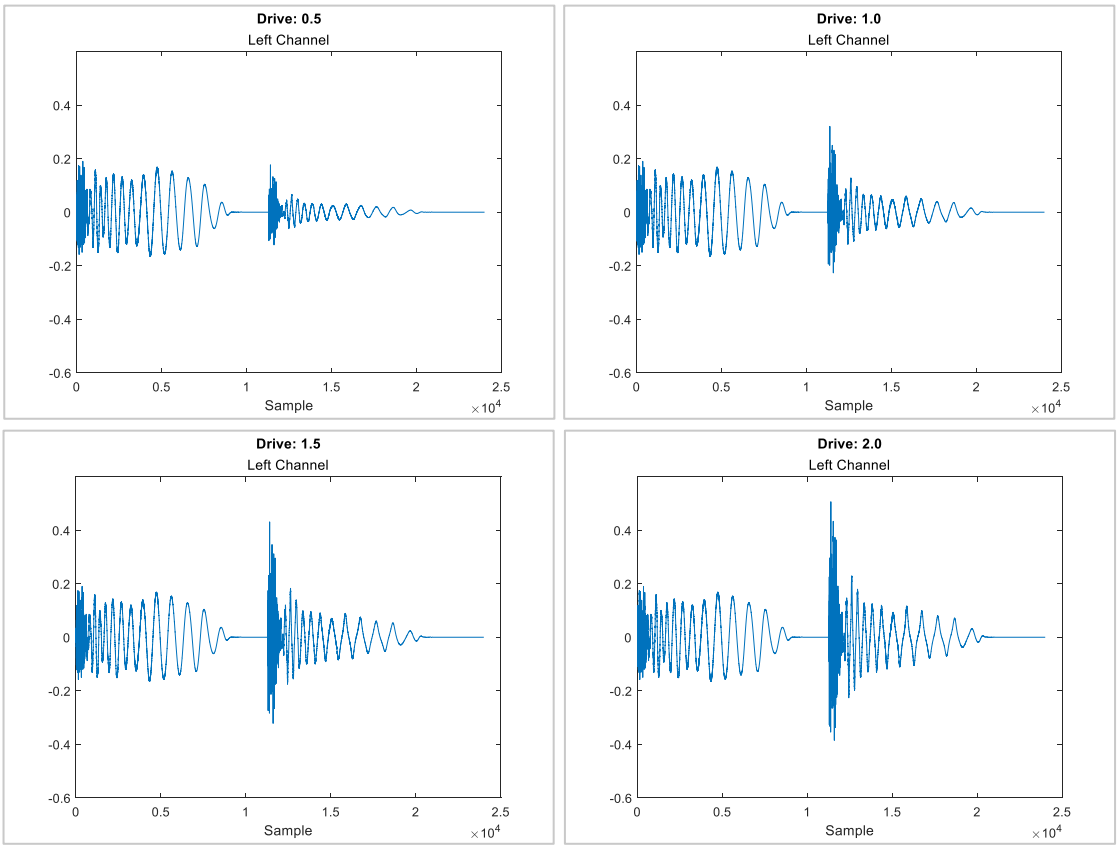


Figure 5 - Plugin Output to Different Drive Values

Figure 5 shows that the delayed signal successfully changes its amplitude depending on the value that the drive parameter has been set at. It can be noted that the delayed signal is not an accurate reproduction of the input signal, however, the delayed signal also has filtering and a distortion algorithm applied to it which alters some of the sonic qualities. With this in mind, and the observation that the drive parameter linearly alters the gain of the repetition, the parameter can be judged to successfully change the input drive to the plugin.

4.3) Tap Selection

The plugin allows the use of multiple taps on the same delay line. The taps read from the buffer at their index and write that data back into the feedback loop. Tap 1 is set from the speed parameter and is then used to set the other taps. Tap 2 is half the length of tap 1 and tap 3 is a third the length of tap 1.

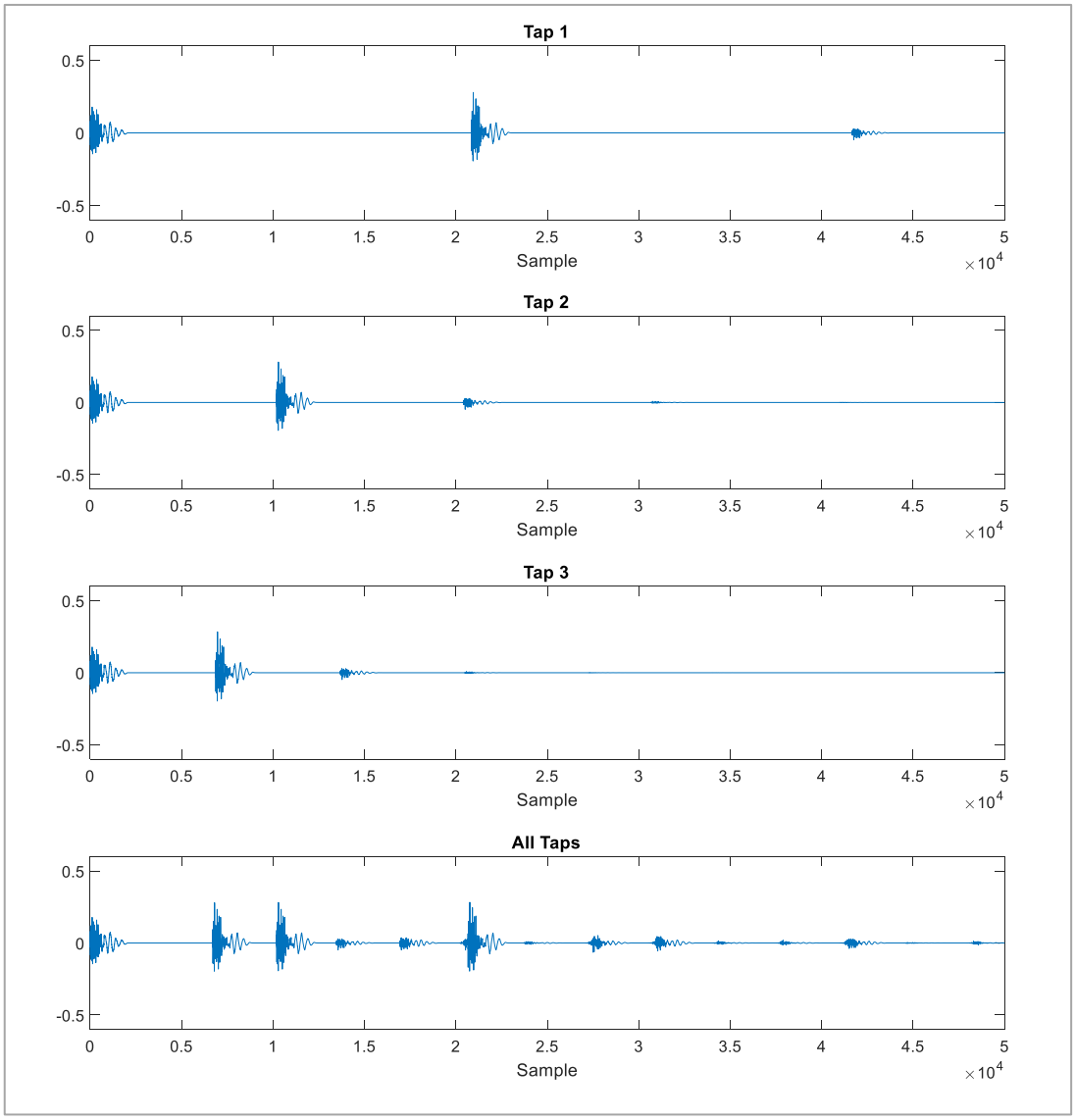


Figure 6 - Output with Different Taps Enabled

Figure 6 shows the output with the same speed parameter value but different taps selected, as well as the combination of all taps. It can be seen that the taps are set at the correct division of the tap 1 speed. It can also be seen that the combination successful applies all the taps concurrently. Furthermore, the use of the division of tap 1 to set the delay intervals creates a pleasing rhythmic effect when the taps are all applied in combination.

4.4) Feedback Parameter

The feedback parameter alters the gain of the signal being sent from the read index to be written back into the write index. To be assessed as successful, a higher feedback coefficient causes more repetitions of the delayed signal.

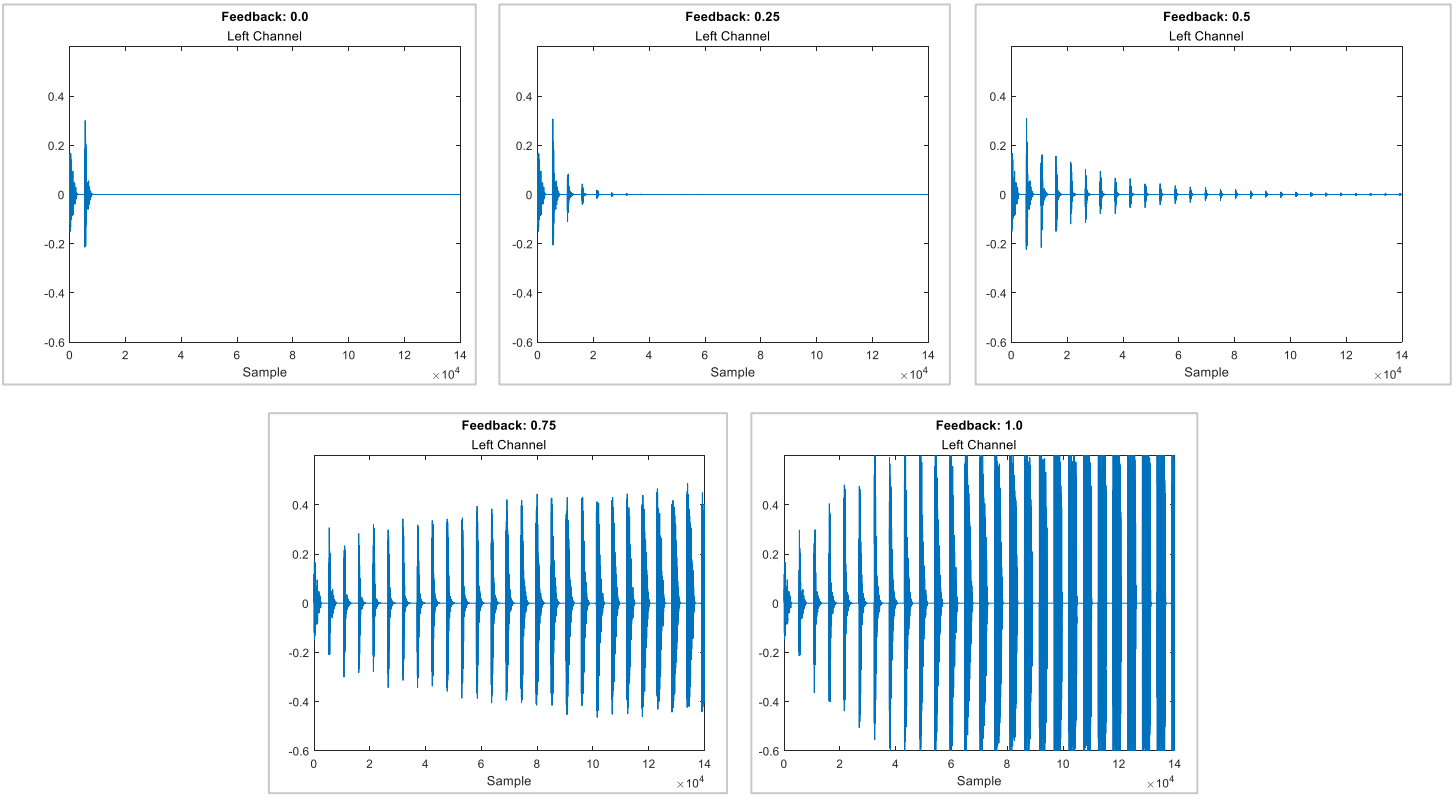


Figure 7 - Plugin Output with Different Feedback Values

Figure 7 shows the output of the plugin to different values of the feedback parameter. With minimum feedback, there are no repetitions after the initial delay. Following from this, as the feedback increases, the amplitude of the further repetitions increases. The feedback can be seen to be stable for values below 0.75 where it tails off to decrease the output eventually to zero. The tests of 0.75 and 1.0 feedback values show an unstable output where the repetitions increase in amplitude each time.

For a feedback value below 1.0 with such a long delay as shown in figure 7 it would conventionally be expected that the output would not become unstable. However, it is assumed that the non-linear gain curve that results from the distortion algorithm causes the feedback to become unstable. This is not considered an issue as delay effects are famous for their build-up of feedback and it is often used as a deliberate effect.

The tests show that the feedback parameter successfully controls the gain of the signal being sent back into the delay line from the read index and can therefore be judged as successful for its purpose.

4.5) Speed Parameter

The speed parameter refers to the speed of the motor on a tape delay. This controls the time interval between each delayed signal. When the speed is at maximum, the time interval is at its shortest and when set at minimum, the time delay is at its longest.

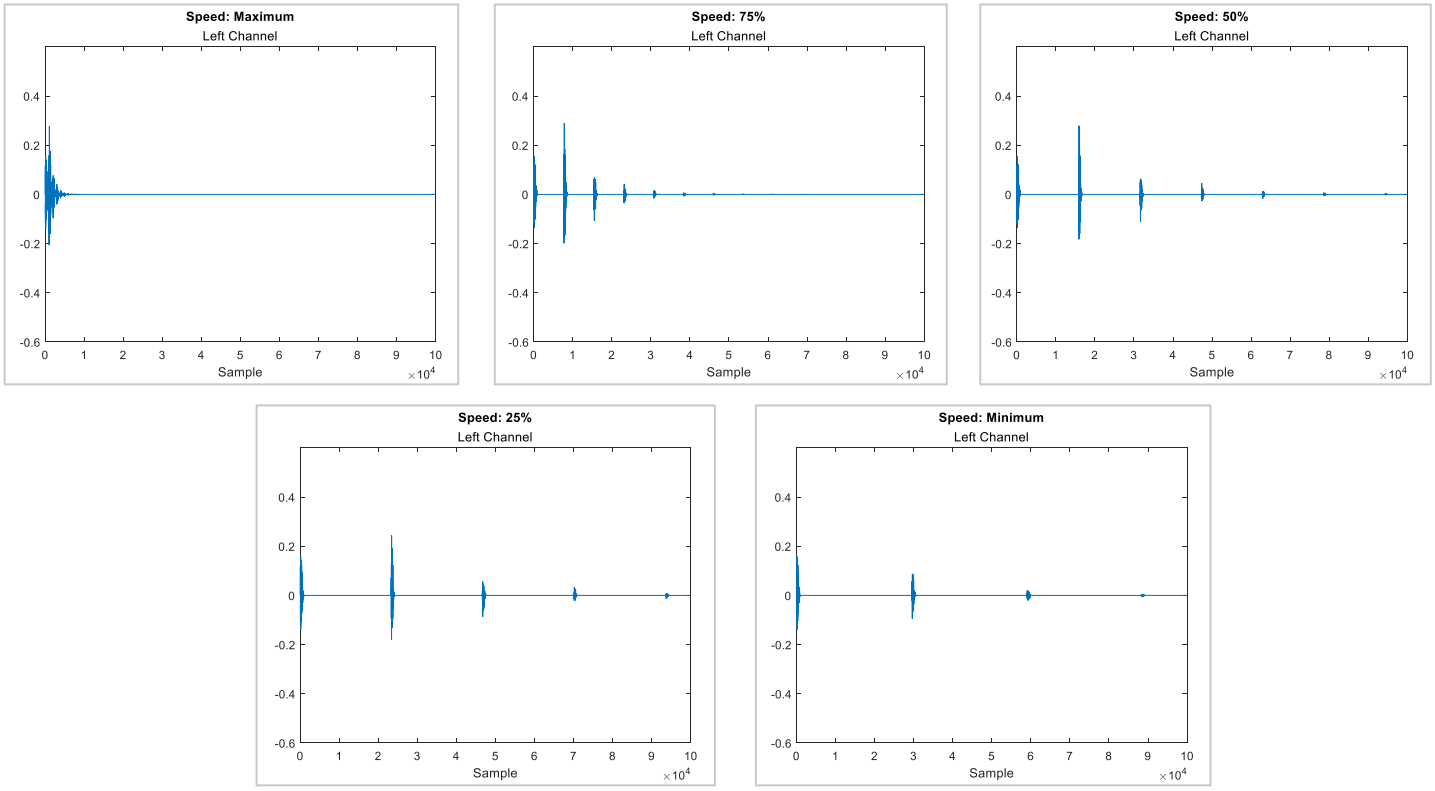


Figure 8 - Plugin Output with Different Speed Controls

Figure 8 displays the effect of the speed set at multiple different values. It can be seen that when the speed is set to maximum the repetitions occur with the shortest interval between them. The interval length then increases linearly as the speed parameter decreases its value.

It can be noted that when the speed is at minimum and interval is at its greatest, the output amplitude of the repetitions is lower than the other speeds. This indicates that the signal has successfully been filtered differently for different speeds. The speed parameter does not only change the delay interval but its also changes the centre frequency of the band pass filter that is applied to the signal. This is to replicate the change in frequency of recording on tape at different speeds whereby tape recorded with a higher speed has more high frequency content than lower speeds and vice versa.

The frequency response of the delayed signal to a delta function with different speed values can be seen in figure 9. The blue signal shows the response at maximum speed, the purple signal is at 50% speed, and orange is at minimum speed.

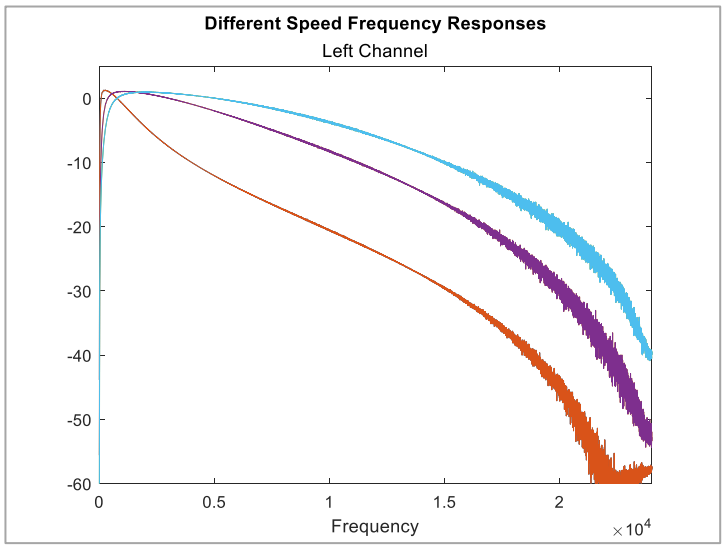


Figure 9 - Frequency Response of Delay at Different Speed Values

It can be seen that the delayed signal has been through a bandpass filter with a low Q value. The filter successfully changes its centre frequency depending on the speed of the delay resulting in a different frequency response. For the lowest speed, it can be seen that the high frequencies are attenuated quite rapidly to give a higher bass response and more noticeable effect than at higher speeds. When the speed is set to maximum, the bandpass filter has a far less steep attenuation and therefore the effect is less noticeable, however, the attenuation of the lowest frequencies can be observed.

Furthermore, the phase response of the filter may also be commented on. It was noticed that the band pass filter caused a 180° phase shift at the output which can cause phase cancellation with the dry signal. The output of this filter was therefore inverted to ensure it was the same phase as the input signal.

4.6) Output Parameters

In the output section of the plugin, there are parameters to control the bass and treble. This is applied to only the wet signal and not the dry. To test this, a delta function has been used and figure 10 shows the frequency response at different levels of bass and treble. This was conducted with the feedback and dry signal set to 0%.

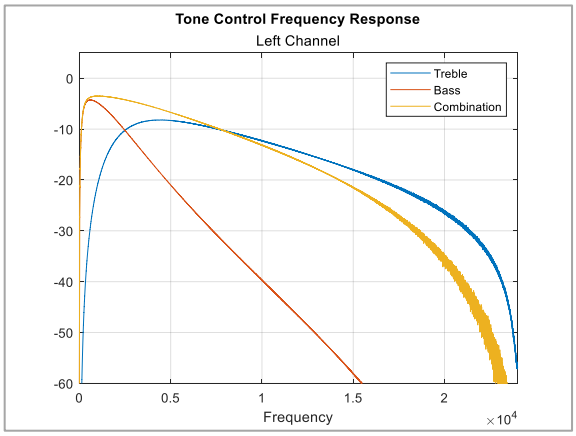


Figure 10 - Output of Plugin with Different Values of Bass and Treble

With only the bass parameter enabled, it can be observed that high frequencies from the signal are steeply attenuated. Similarly, with only the treble parameter enabled, the low frequency signals are steeply attenuated.

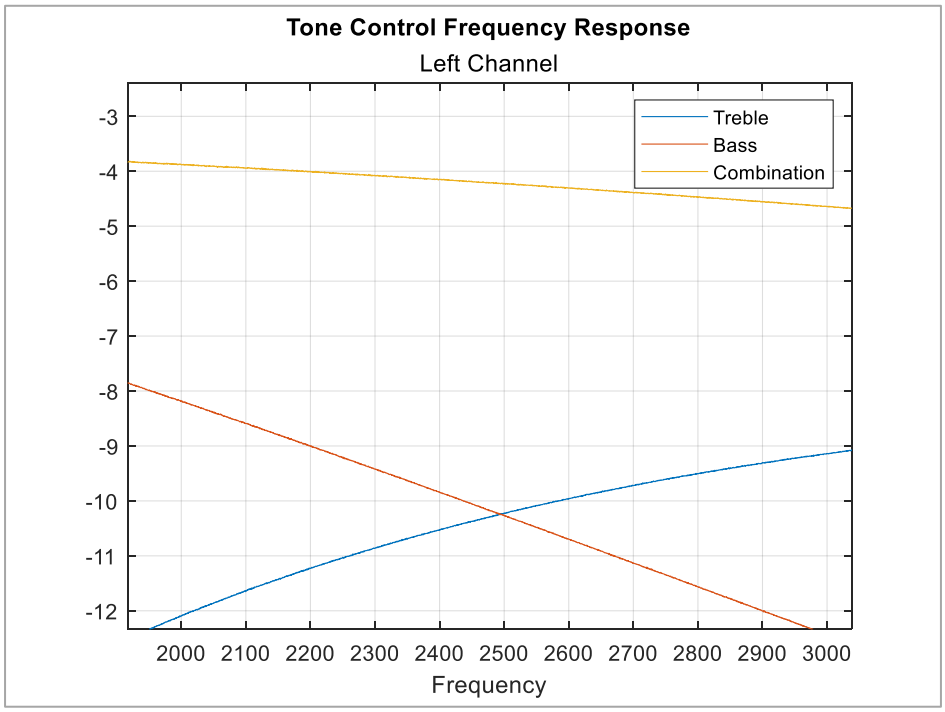


Figure 11 – Linkwitz-Riley Crossover at 2.5kHz

A Linkwitz-Riley crossover filter combination was used meaning that the filters are attenuated to -6dB at the cutoff frequency. This means that the summed response is flat. Due to the bandpass filter described in a previous section, a completely flat response cannot be tested, however, from figure 11 it can be observed that the crossover occurs at the set frequency of 2.5kHz and is at 6dB less than the combined signals. It can therefore be determined that the

bass and treble parameters work as planned with the combination output allowing signals to pass without attenuation.

The wet and dry amount can be controlled with the balance parameter which blends the amount of wet and dry mix in the output. Figure 12a shows the output with the balance set to 100% dry and figure 12b shows the output with the balance at 100% wet. It can be seen that figure 12a contains only the original input and figure 12b shows only the delayed output.

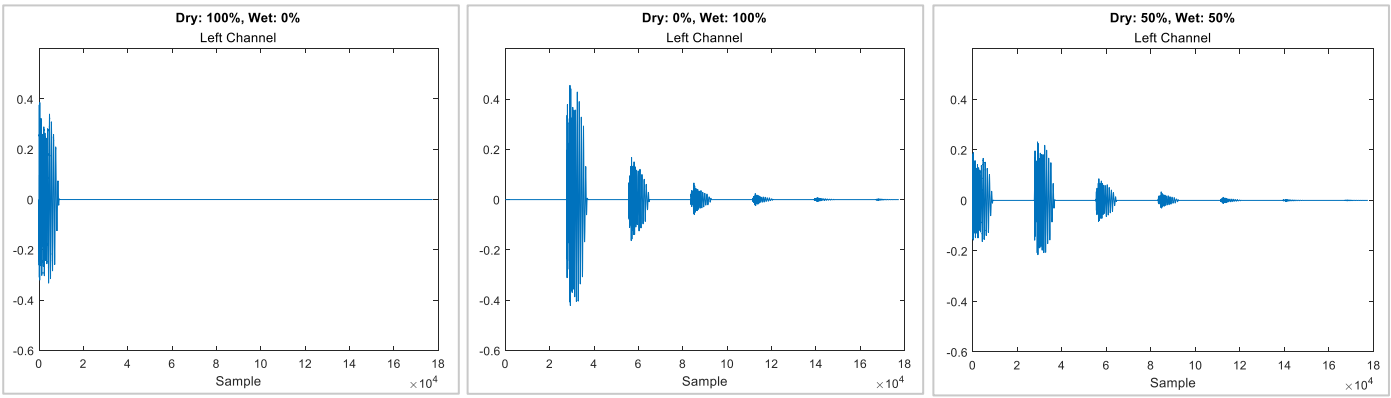


Figure 12 - Wet/Dry Mix Output

In figure 12c, the balance is set to its middle setting. By the nature of a wet/dry mix this means that both outputs are set to 50%. As a result, when the balance is not set to either maximum wet or dry, the output signal is attenuated by half. To account for this, a make-up gain control has been added with a maximum linear value of 2 which allows for the user to normalise the gain to match the input.

4.7) Wow and Flutter

The wow and flutter that is applied to the delay uses two LFOs at different frequencies. The wow LFO is at a frequency of 0.1Hz and the flutter LFO has a frequency of 1Hz. The testing of the LFOs is difficult to show in images so a video of testing can be seen in the link in Appendix 3.

In the video, a sine wave can be seen. When the plugin is applied to it, the wave's frequency becomes unstable. The frequency has small fast fluctuations, which are a result of the flutter LFO, and longer larger changes in frequency which are from the wow LFO. This shows the wow and flutter to be working.

Furthermore, the wow LFO's amplitude is controlled by the speed parameter. From changing the speed, it can be seen that the slower speeds results in a slightly larger frequency change from the wow LFO. It can therefore be noted that the wow and flutter effects work as intended.

When the plugin is applied to a guitar sample, the wow LFO gives the signal as chorus type effect when mixed with the dry signal.

5) Conclusions

In conclusion, this report detailed the methodology and testing of a tape-echo style delay plugin made using JUCE with C++. The plugin architecture allows for a simplistic and intuitive GUI for the user with the DSP of the plugin allowing more complex systems to be explored.

The testing of the plugin showed that each parameter and function works as intended. The testing with different signals and viewing of the signal in both the time and frequency domain allowed for a rigorous testing procedure that ensures beyond reasonable doubt that the plugin works as intended.

In terms of the tape style delay, the implemented effects give tape delay behaviour and are implemented as intended, however, they could not be described as an accurate emulation. To improve the accuracy of the tape qualities of this plugin, closer analysis of a real tape delay would need to be conducted. This could become more accurate by, for example, closely inspecting the frequency response at each speed and applying that to an equaliser on the signal. Similarly, work could be done to accurately track the rates and amplitude of the wow to precisely control the speed fluctuations.

6) References

Bristow-Johnson, R., n.d. Cookbook formulae for audio equalizer biquad filter coefficients. [ebook] Available at: (Accessed 20 Jan 2022).

Endino, J. (2006) *The Unpredictable Joys of Analog Recording.* Available at: http://www.endino.com/graphs/index.html (Accessed: 01/02/ 2022).

Pirkle, W.C. (2013) *Designing audio effect plug-ins in C++: with digital audio signal processing theory.* Focal Press.

Sakai, H. (1970) 'Perceptibility of Wow and Flutter', *Journal of the Audio Engineering Society,* 18(3), pp. 290-298.

Tarr, E. (2018) *Hack Audio.* Taylor and Francis.

Zölzer, U. and Zlzer, U. (2011) *DAFX.* 2. ed. edn. Chichester: Wiley.

Videos:

Korg se-300 stage echo (10/06/2010) YouTube video, added by hiwattbob [Online]. Available at: https://www.youtube.com/watch?v=J3omn-7WccA&t=155s [Accessed 23/01/2022].

Masterclass w/ Prince Fatty (J Dilla, Mad Professor): Creativity, Mixing & Dub FX Tips (15/03/2017) YouTube video, added by Point Blank Music School [Online]. Available at: https://www.youtube.com/watch?v=AmEwSLuUbsI&t=1946s [Accessed 23/01/2022].

Images:

Korg SE-500 (n.d.) [Online]. Available at: https://www.ocsidance.com/product/korg-se500-stage-echo/ (Accessed 01/02/2022).

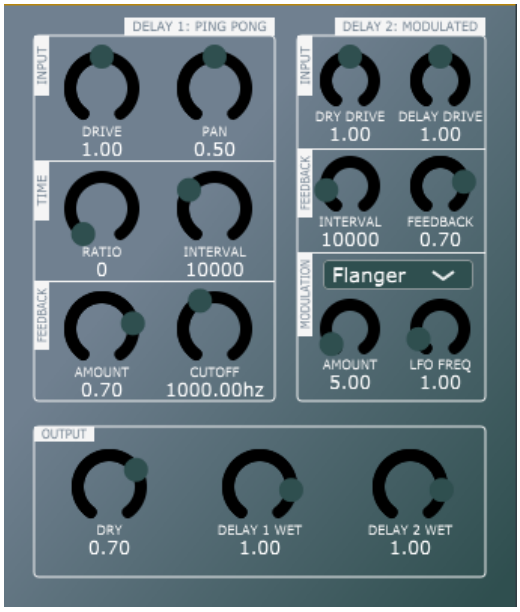
Roland Space Echo (n.d.) [Online]. Available at: https://theprodigy.info/equipment/roland-space-echo-re-201.html (Accessed 01/02/2022).

7) Appendices

Appendix 1
Project Repository:

https://github.com/mattwilliams2099/Delay_Assessment_Repo.git

Appendix 2
Original GUI for project:



Appendix 3
Link to wow-flutter video:

https://drive.google.com/file/d/18lUtsXvwYCv_f1d8Gq_lwO5fn5SeNxCJ/view?usp=sharing