DELAY-BASED AUDIO EFFECT SUITE

Cal Goheen and Will Cutter

University of Rochester

ABSTRACT

Audio effects are algorithms applied to an input signal which alter the sound in a desirable way. They are generally used in music production to achieve a desired sound, or help balance a mix. In this paper, three such effects were implemented in MATLAB as audio plugins: chorus, flanger, and phaser. These effects are all related in that their output is the sum of the original signal and a delayed copy of the input signal.

1. INTRODUCTION

1.1 Interference

The ‘sound’ of all three effects discussed in this paper is produced by interference, that is, when the delayed signal is mixed with the input, some frequencies will be boosted by constructive interference, and others will be cut by destructive interference. In the graph below, the input consists of a combination of the 500 Hz and 1000 Hz sine waves. After processing through a delay line with a length of 1 ms, and combining this signal with the input, the resulting output has no frequency component at 500 Hz, and twice the amplitude at 1000 Hz.

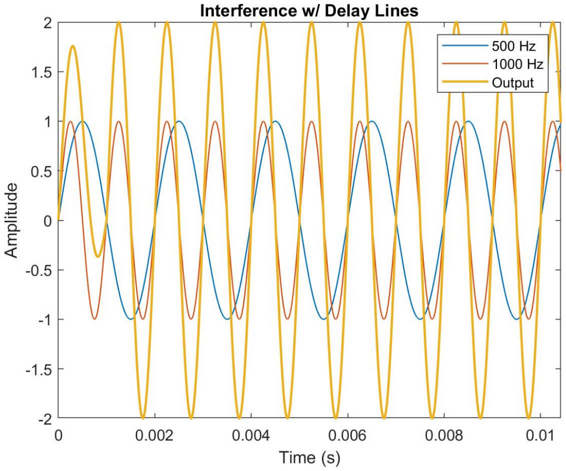

Figure 1: Interference with a delay line

1.2 Flanger Effect

A flanger effect consists of a fractional delay line with its delay length modulated over time. In addition, a portion of the delayed signal is fed back through the delay line. After summing the delayed signal with the input, the output is a comb-filtered version of the input, with peaks and troughs that are in a linear harmonic series. The process for a flanger is illustrated in the following figure:

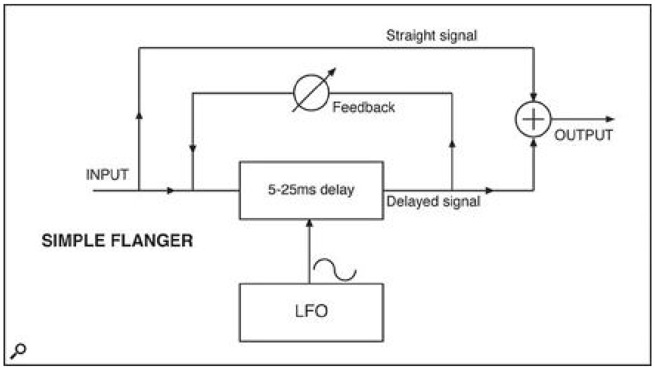

Figure 2: Flanger block diagram

1.3 Chorus Effect

A chorus effect aims to emulate real life chorus, where each voice is heard at slightly different times by the listener and there are minute variations in pitch from voice to voice within the chorus. This effect is achieved by creating multiple copies of the input signal which are then slightly delayed via fractional delay lines. This creates a phase difference between the dry signal and the wet signal. The length of the fractional delay line (i.e. the delay time) is then modulated over time, which slightly detunes the output. The resulting signal differs in both time and frequency from the original signal. Mixing the wet signal back with the dry signal creates a widened stereo image and makes the sound seem “thicker”.

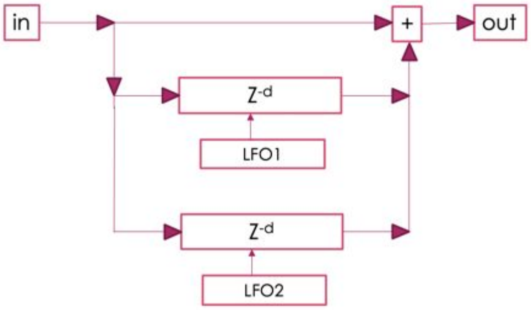

Figure 3: Chorus block diagram

1.4 Phaser Effect

A phaser effect is slightly different from a chorus or flanger in that instead of a delay line, a phaser uses a series of all-pass filters. The all-pass filters have a gain of unity, but alter the phase of the input depending on the signal's frequency. The frequency of the phase shift is modulated over time. After summing the output of the all-pass filters with the input signal, the overall output is a comb-filtered version of the input, but the peaks and troughs are not harmonically related. Feedback can also be included in a phaser, although it was not implemented in this project.

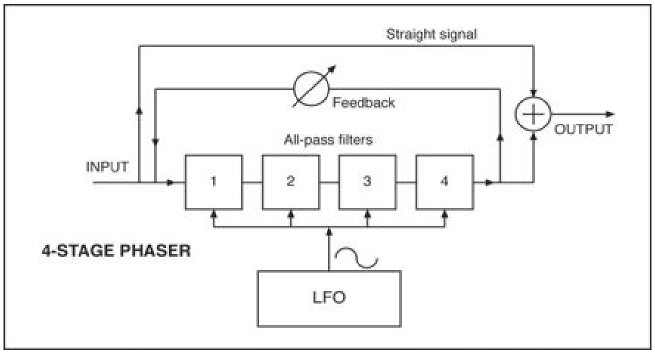

Figure 4: Phaser block diagram

2. METHODS

2.1 Low-Frequency Oscillator

A low-frequency oscillator (LFO) is necessary in all three audio effects to achieve modulation over time. In the flanger and chorus, an LFO modulates the length of the variable fractional delay lines. In the phaser, the LFO modulates the value of the feedback and feedforward coefficient, *g*.

In this project, the LFO was implemented as a class in MATLAB, allowing all three audio effect classes to reference it. The LFO class has the following user-controlled parameters:

freq - frequency of the LFO signal
base - DC offset of LFO signal
depth - amplitude of the LFO signal
phi - initial phase offset of the LFO signal
waveform - waveform of LFO signal
framesize - frame size of LFO signal

When the process function of the LFO is called, a frame is filled with samples of the LFO signal at the given sample rate, which can be accessed by parent classes during processing.

2.2 Variable Fractional Delay Line

A fractional delay line is a system component which produces a delay of *L* samples. If *L* is not an integer, its value is linearly interpolated from the values of the two nearest samples. For a variable fractional delay line, the value of *L* can be altered each time a sample is processed through the delay line.

Originally, this component was implemented in MATLAB for use in each effect. However, upon testing in real-time, it was found that there were inefficiencies in the class that caused it to be too computationally expensive. For this reason, the MATLAB system object *audioexample.DelayFilter* was used instead, which provides the same functionality. An example output is shown below which depicts the modulated delay length:

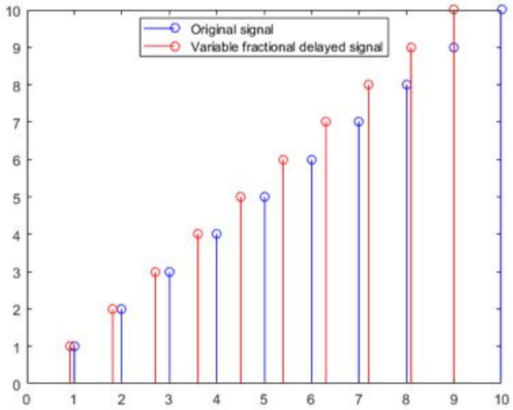

Figure 5: Variable fractional delay

2.3 All-Pass Filter

The figure below depicts the implementation of a first order all-pass filter:

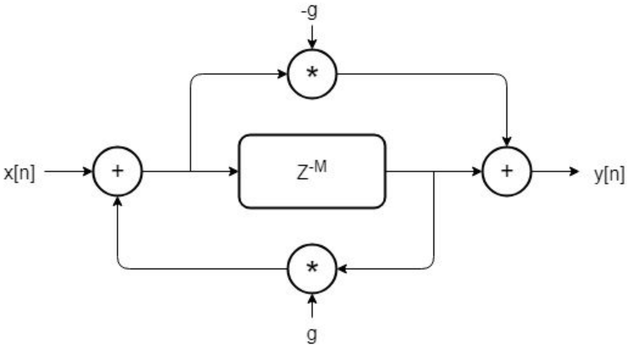

Figure 6: All-pass filter block diagram

In order to speed up processing, the all-pass filters in our project were implemented by calculating the biquad filter coefficients, shown below:

$$\begin{aligned} b_0 &= 1-\alpha \\ b_1 &= -2\cdot\cos\left(\omega_0\right) \\ b_2 &= 1+\alpha \\ a_0 &= 1+\alpha \\ a_1 &= -2\cdot\cos\left(\omega_0\right) \\ a_2 &= 1-\alpha \end{aligned}$$

Source: 'EQ Cookbook', Robert Bristow Johnson

These coefficients are for a second order all-pass filter, or two all-pass filters in series. The number of notches in the frequency spectrum of the phaser is denoted as the *poles* of the phaser, and is equal to the number of second-order all-pass filters.

2.4 Testing the audio effects

Each audio effect was tested using Audio Test Bench in MATLAB. The input signal to each effect was a 1kHz sine wave with an amplitude of 0.5. The input and output signal were then compared in both the time and frequency domain using the scope function from Audio Test Bench.

The following parameter values were used for each audio effect during testing:

Flanger:

Phase Offset = 0.42857
Feedback = 0.20942
LFO Frequency = 1.0063
LFO Center = 0.01
LFO Depth = 0.003022
HPF Cutoff = 150
LFO Waveform = Sine
Dry/Wet = 0.75

Chorus:

LFO Frequency 1 = 5.0501
LFO Frequency 2 = 0.93278
LFO Center 1 = 0.011923
LFO Center 2 = 0.020275
LFO Depth 1 = 0.00046703
LFO Depth 1 = 0.0014011
HPF Cutoff = 277.06
Dry/Wet = 0.66484

Phaser:

Phase Offset = 0.60714
Poles = 8
LFO Frequency = 1611.8
LFO Center = 849.2
LFO Depth = 249.8
LFO Waveform = Sine
APF Cutoff = 924.41
HPF Cutoff = 202.55
Dry/Wet = 0.85714

3. RESULTS

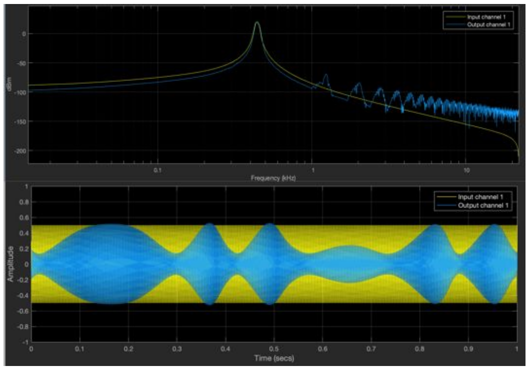

Figure 7: Flanger time and frequency display

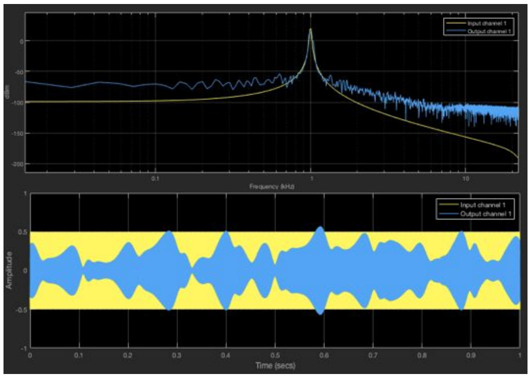

Figure 8: Flanger time and frequency display

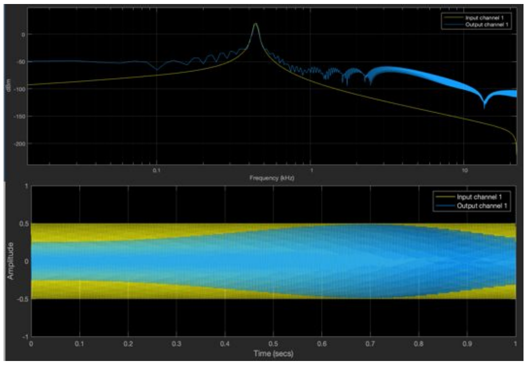

Figure 9: Phaser time and frequency display

4. CONCLUSIONS

The time and frequency displays shown above plot the input signal in yellow and the output signal in blue. The frequency display in Figure 7 shows a comb-filtered output spectrum with a linear harmonic series of peaks and troughs, the expected output spectrum of a signal that has been processed through a flanger. The output signal sounds similar to the output signal from other flanger plugins, with the characteristic “swoosh” sweeping effect.

The waveform display in Figure 8 illustrates the randomly modulated delay introduced by the chorus effect. In contrast to the waveform display of the flanger effect, which appears to have a periodic cycle of constructive and destructive interference, there appears to be no cyclic interference produced by the chorus effect. The output signal sounds “thicker”, as if there are multiple voices of the input audio signal.

The frequency display in Figure 9 shows a comb-filtered output spectrum, although the peaks and troughs produced by the phaser effect are not harmonically related. The output of the phaser developed in this project is comparable to the output of other phaser plugins, with a characteristic “swirling”, other-worldly sound.

After analyzing the output from each effect visually, using the scope function in Audio Test Bench, and auditorily, it can be concluded that the effects designed in this project achieved their desired results. Further development could include modifying the existing code to produce VST plugins, allowing these effects to be used in a digital audio workstation.

5. REFERENCES

[1]Bristow-Johnson, Robert. “Cookbook Formulae for Audio Equalizer Biquad Filter Coefficients.” Cookbook Formulae for Audio EQ Biquad Filter Coefficients, www.w3.org/2011/audio/audio-eq-cookbook.html

[2]Brown, Griffin. “Understanding Chorus, Flangers, and Phasers in Audio Production.” *Izotope.com*, IZotope, Inc., 6 Sept. 2018, www.izotope.com/en/blog/music-production/understanding-chorus-flangers-and-phasers-in-audio-production.html.

[3]“Design an Audio Plugin.” Mathworks.com, The MathWorks, Inc., www.mathworks.com/help/audio/gs/design-an-audio-plugin.html.

[4]“Dsp.VariableFractionalDelay.” Mathworks.com, TheMathWorks, Inc., www.mathworks.com/help/dsp/ref/dsp.variablefractionaldelay-system-object.html.

[5]Howell, Steve. “Q. What's the Difference between Phasing and Flanging?” Soundonsound.com, SOS Publications Group, Mar. 2006, www.soundonsound.com/sound-advice/q-whats-difference-between-phasing-and-flanging