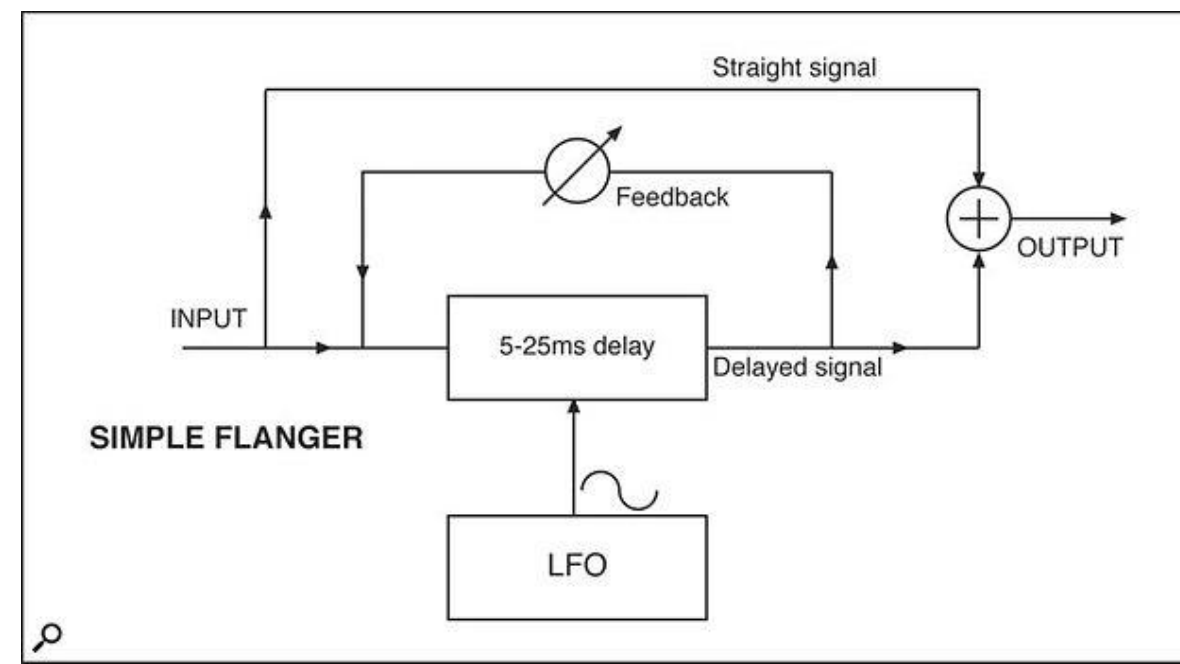


By: Will Cutter and Cal Goheen

University of Rochester

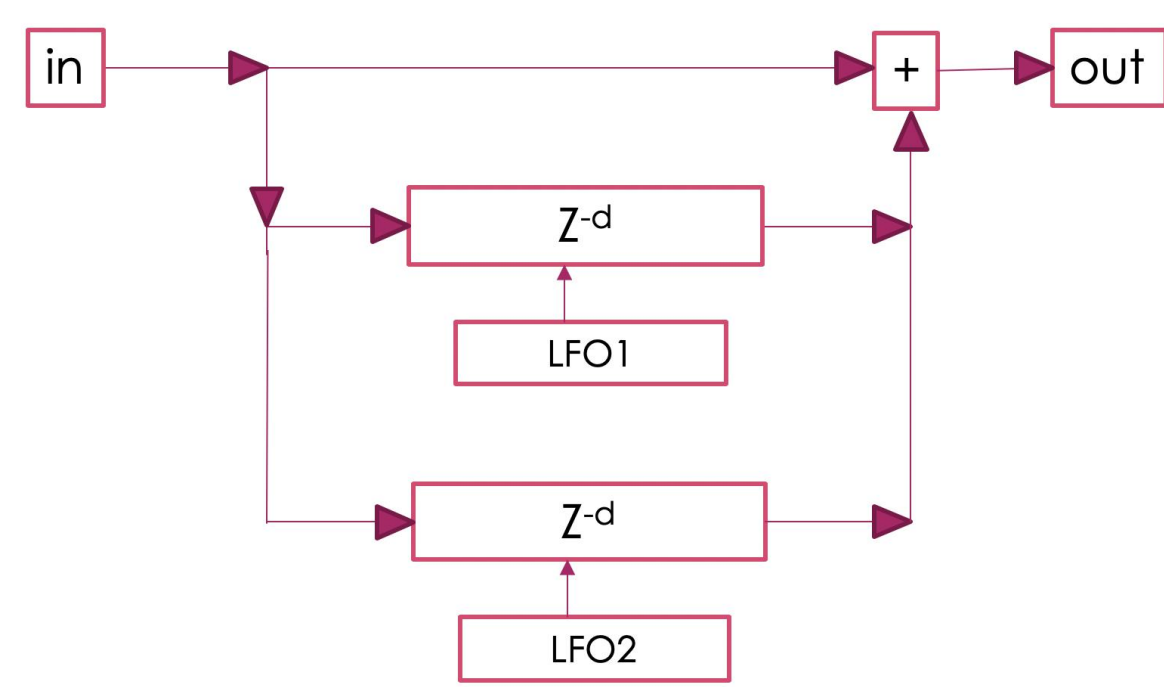
## Introduction

### Flanger



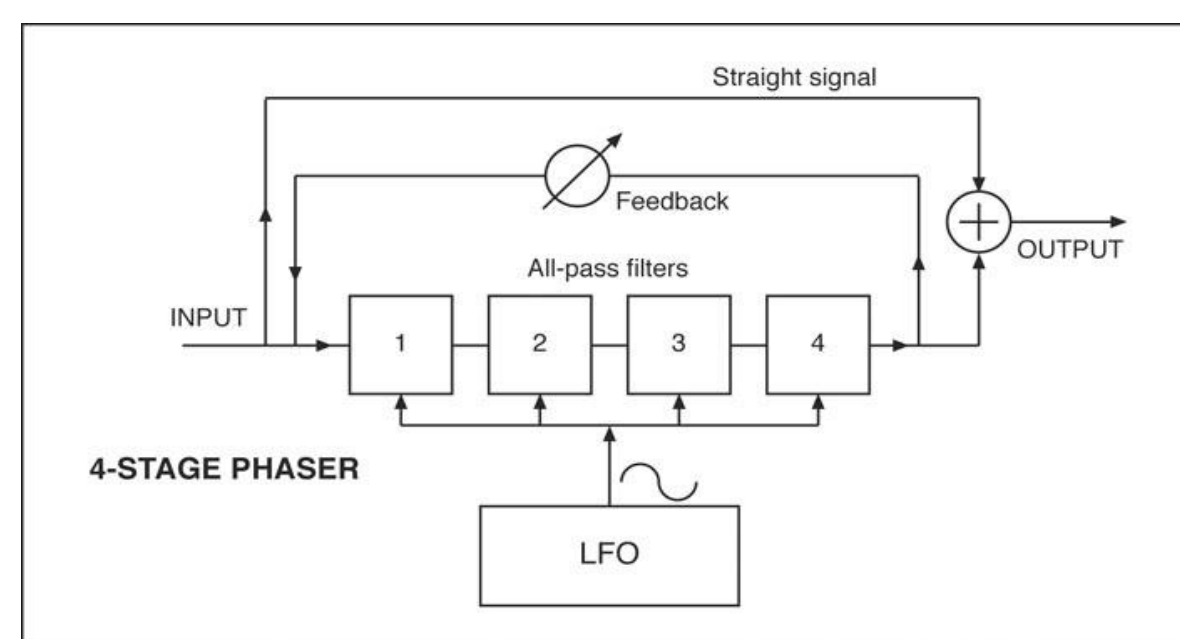
- Variable length fractional delay line modulated by an LFO
- Portion of delayed signal is fed back through delay line
- Delayed signal mixed with dry signal
- Output signal is a comb-filtered version of input signal, with peaks and troughs that are in a linear harmonic series

### Chorus



- Signal is split and sent through two variable length fractional delay lines modulated by two independent LFOs
- Delayed signals mixed with dry signal
- Output signal sounds like multiple "voices" of the input signal, resulting in a wider and thicker sound

### Phaser



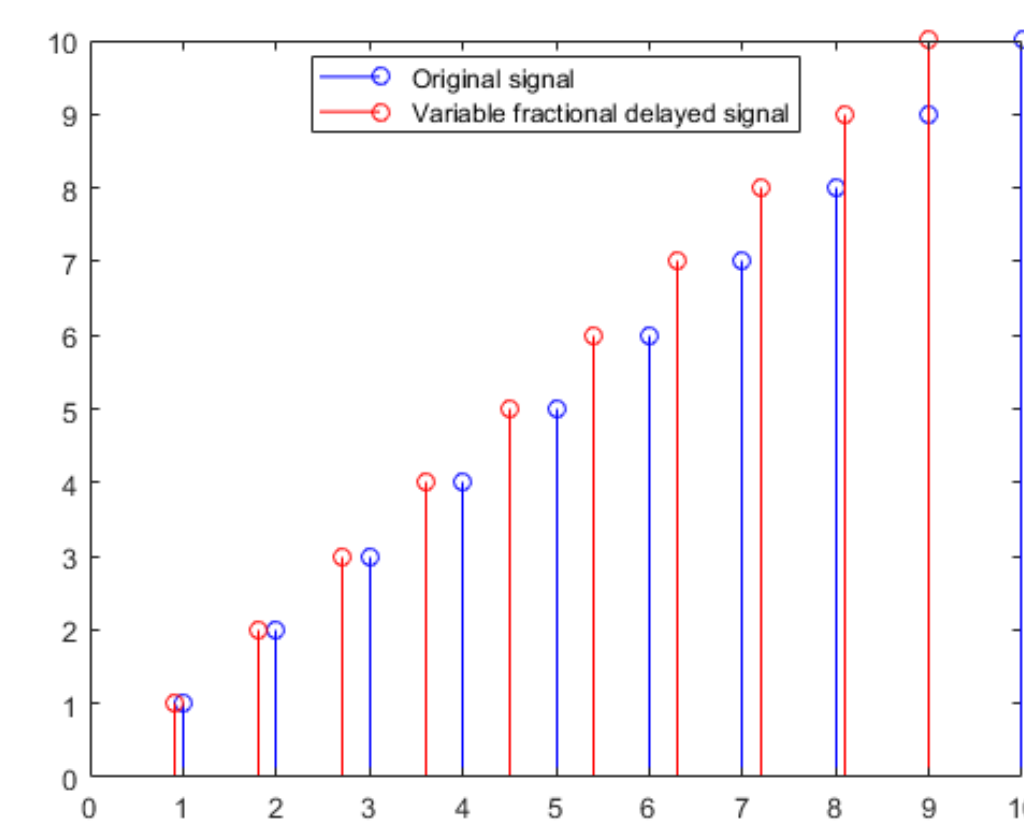
- Signal sent through cascaded allpass filters,
- Phase shift produced by each allpass filter is modulated by an LFO
- Portion of phase-shifted signal is fed back through cascaded allpass filters
- Phase-shifted signal mixed with dry signal
- Output signal is a comb-filtered version of input signal, with peaks and troughs that are not harmonically related

## Methods

### LFO

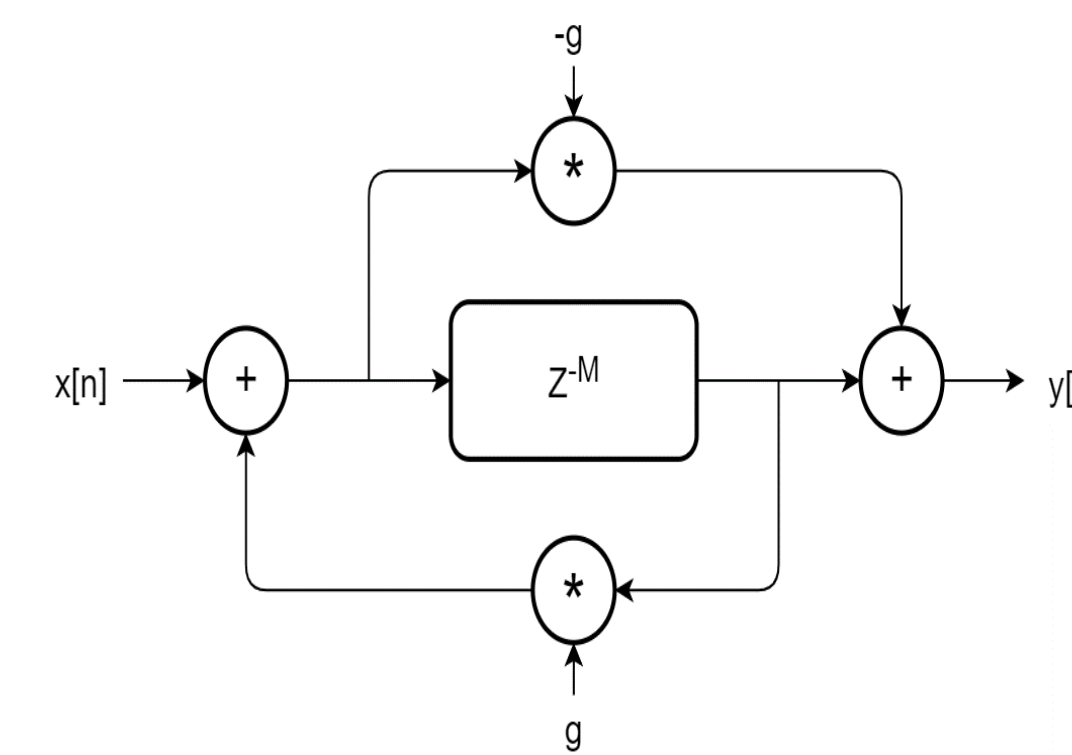
- Used to modulate the length of the variable fractional delay lines (i.e. delay time) and the all-pass filter 'cutoff' frequency
- Implemented as a class in MATLAB so the audioPlugin class for each audio effect can easily reference it
- LFO frequency can be varied from 0 to 10 Hz

### Variable Fractional Delay Line



- The basis for all three effects
- Implemented using the MATLAB system object *dsp.VariableFractionalDelay* and *audioexample.DelayFilter*
- Delay value can vary from sample to sample within a frame
- Uses linear interpolation to calculate new samples at non-integer sampling intervals

### Allpass Filter



- Implemented by calculating the bi-quad coefficients of the filter
- Passes all frequencies at unity gain (i.e. neither increases or decreases the level of any frequency)
- Alters the phase response of the input signal
- Output signal is a phase-shifted version of the input signal

## Generating Real-Time Audio Effects in MATLAB

```
classdef myBasicPlugin < audioPlugin
% myBasicPlugin is a template basic plugin, use this template to create
% your own basic plugin.
properties
% Use this section to initialize properties that the end-user interacts
% with.
end
properties (Access = private)
% Use this section to initialize properties that the end-user does not
% interact with directly.
end
properties (Constant)
% This section contains instructions to build your audio plugin
% interface. The end-user uses the interface to adjust tunable
% parameters. Use audioPluginParameter to associate a public property
% with a tunable parameter.
end
methods
function out = process(plugin, in)
% This section contains instructions to process the input audio
% signal. Use plugin.MyProperty to access a property of your
% plugin.
end
function reset(plugin)
% This section contains instructions to reset the plugin between
% uses or if the environment sample rate changes.
end
function setMyProperty(plugin, val)
% This section contains instructions to execute if the
% specified property is modified. Properties associated with
% parameters are updated automatically. Use the set method to
% execute more complicated instructions.
end
end
end
```

### Results

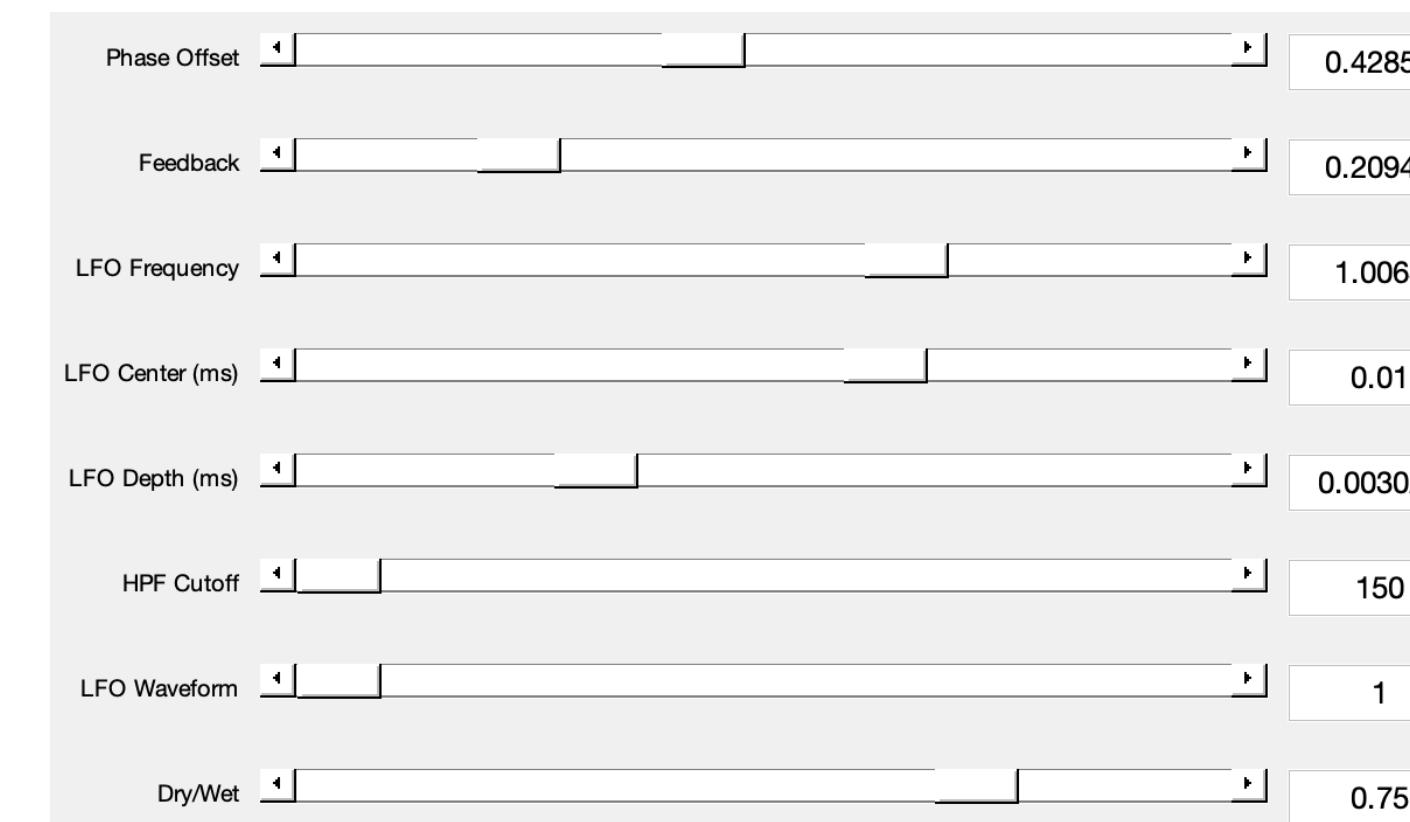


Figure A: Flanger settings

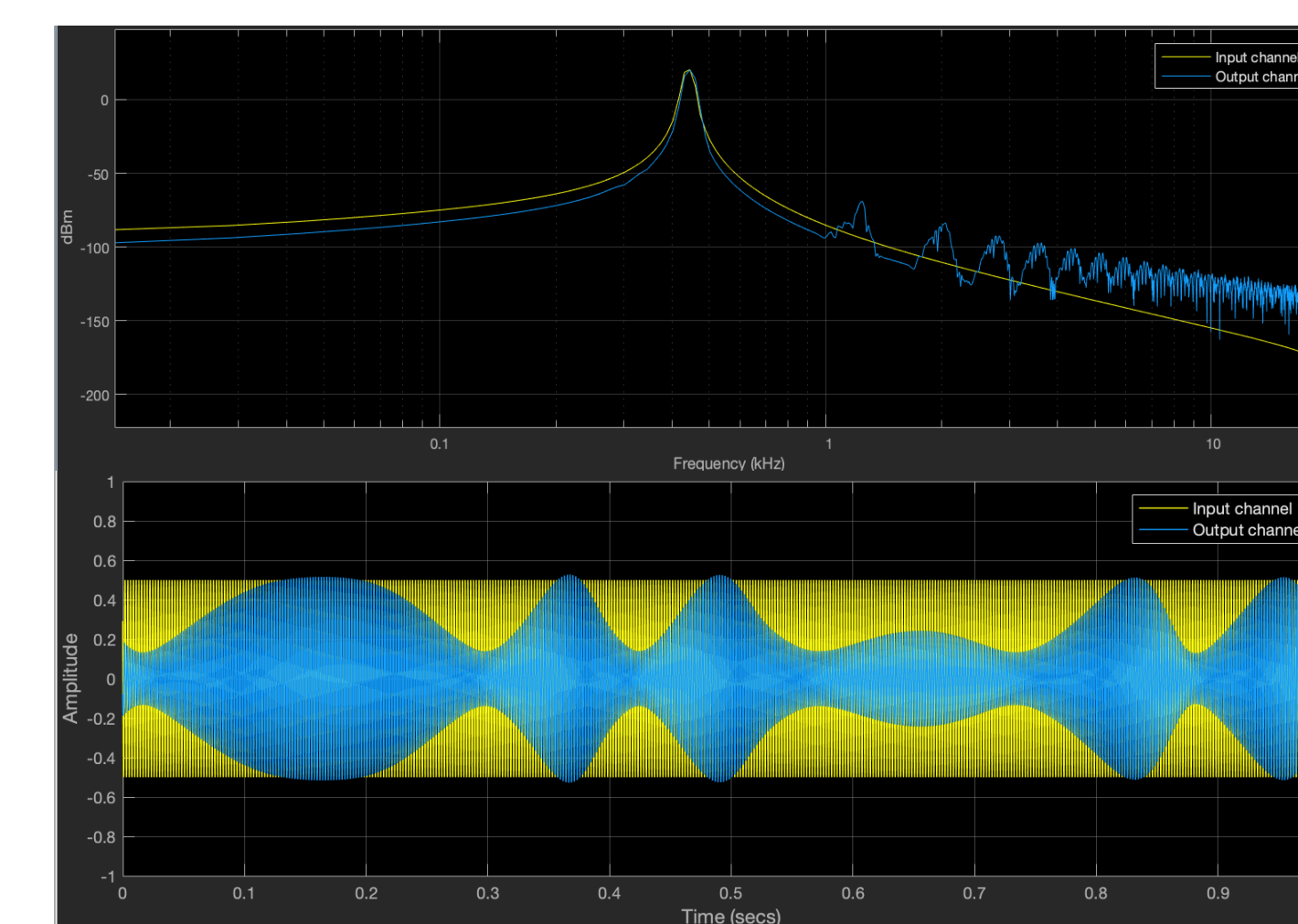


Figure B: Flanger input and output frequency and waveform display

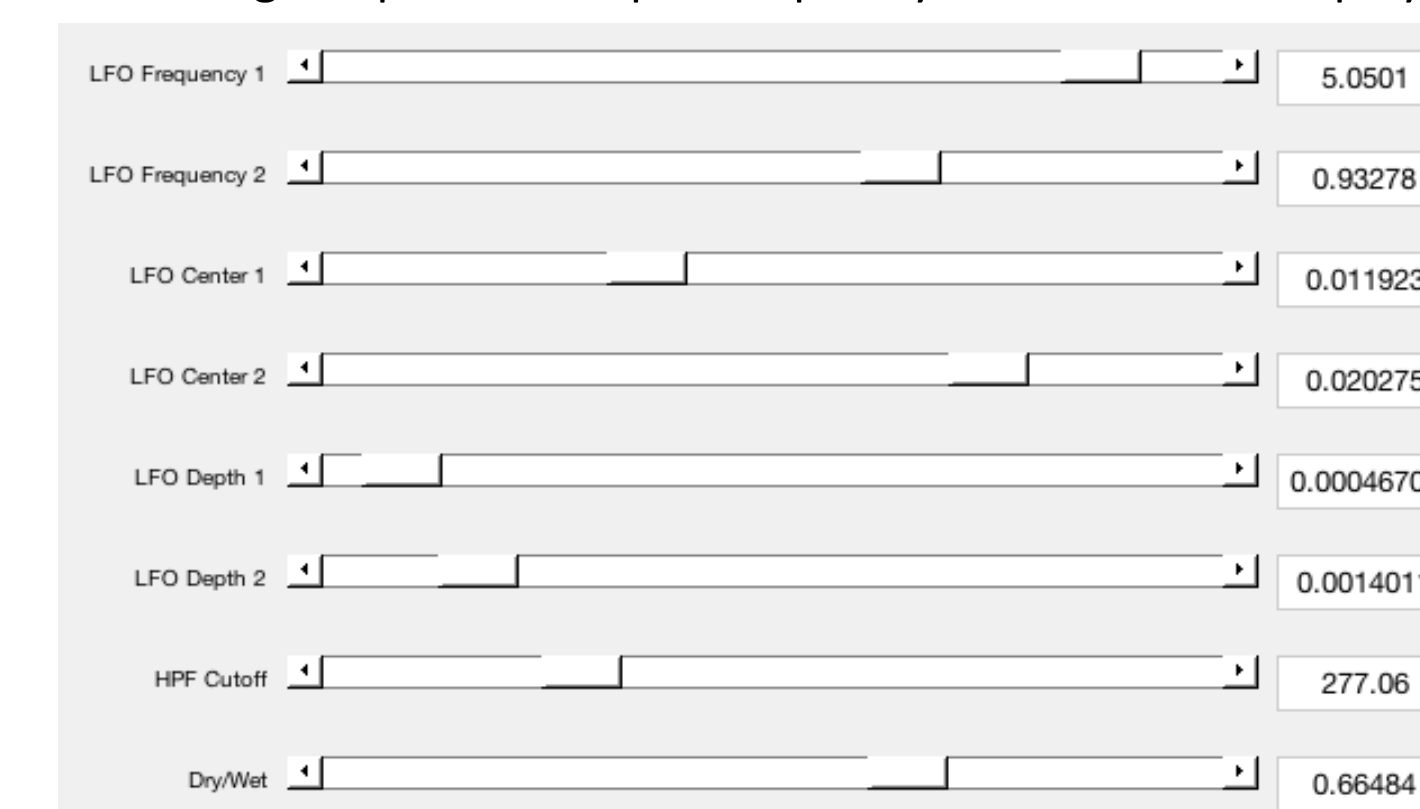


Figure C: Chorus settings

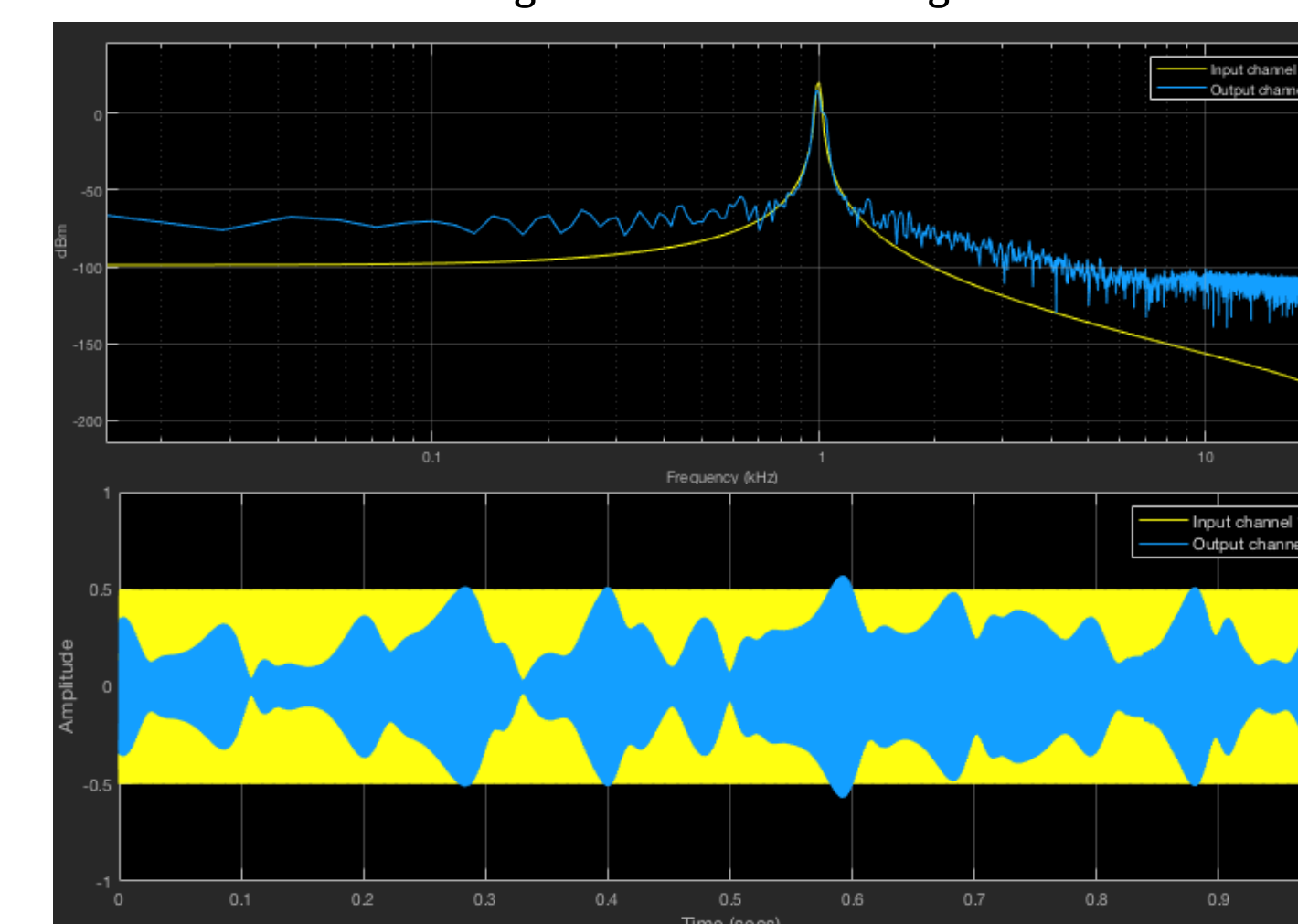


Figure D: Chorus input and output frequency and waveform display

## Results (continued)

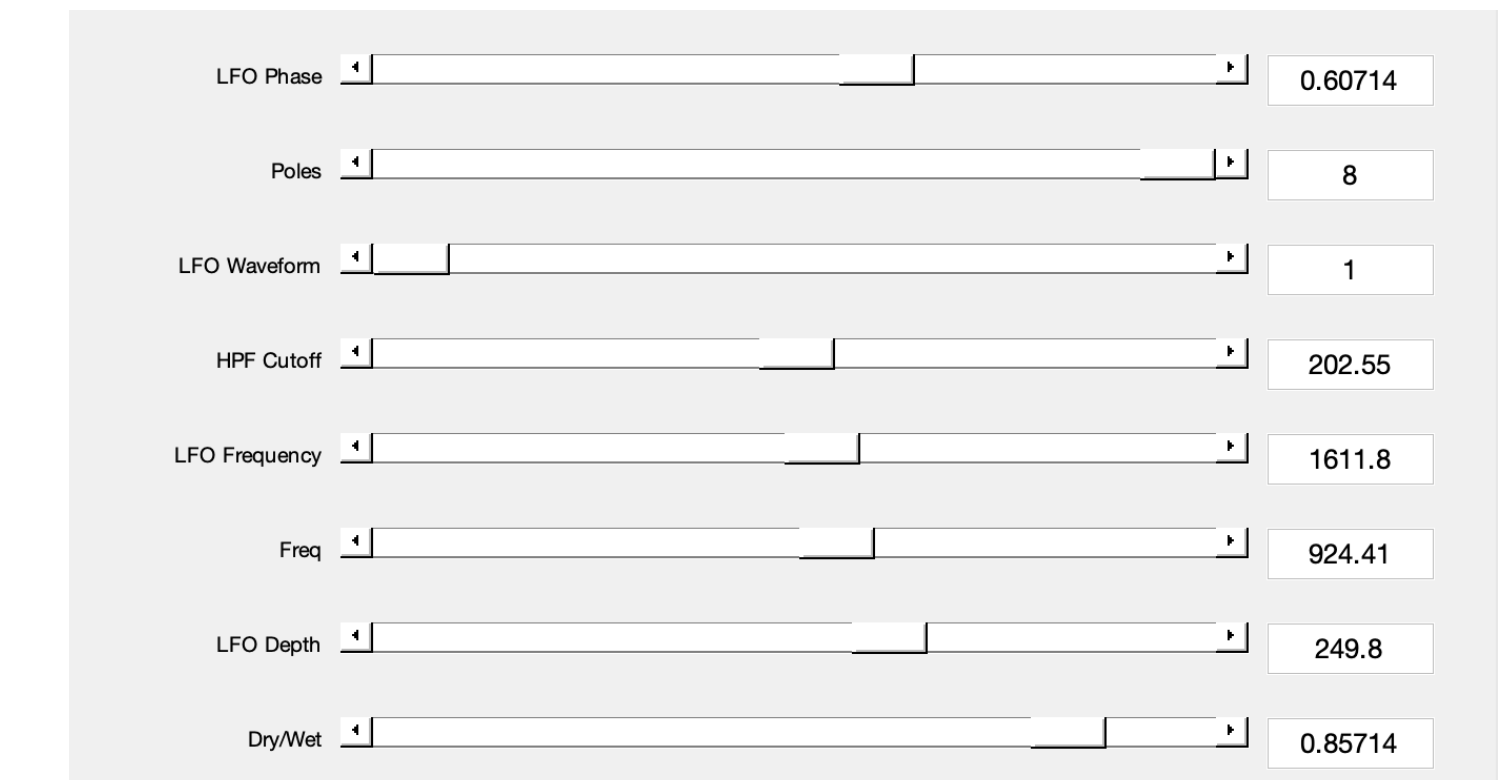


Figure E: Phaser settings

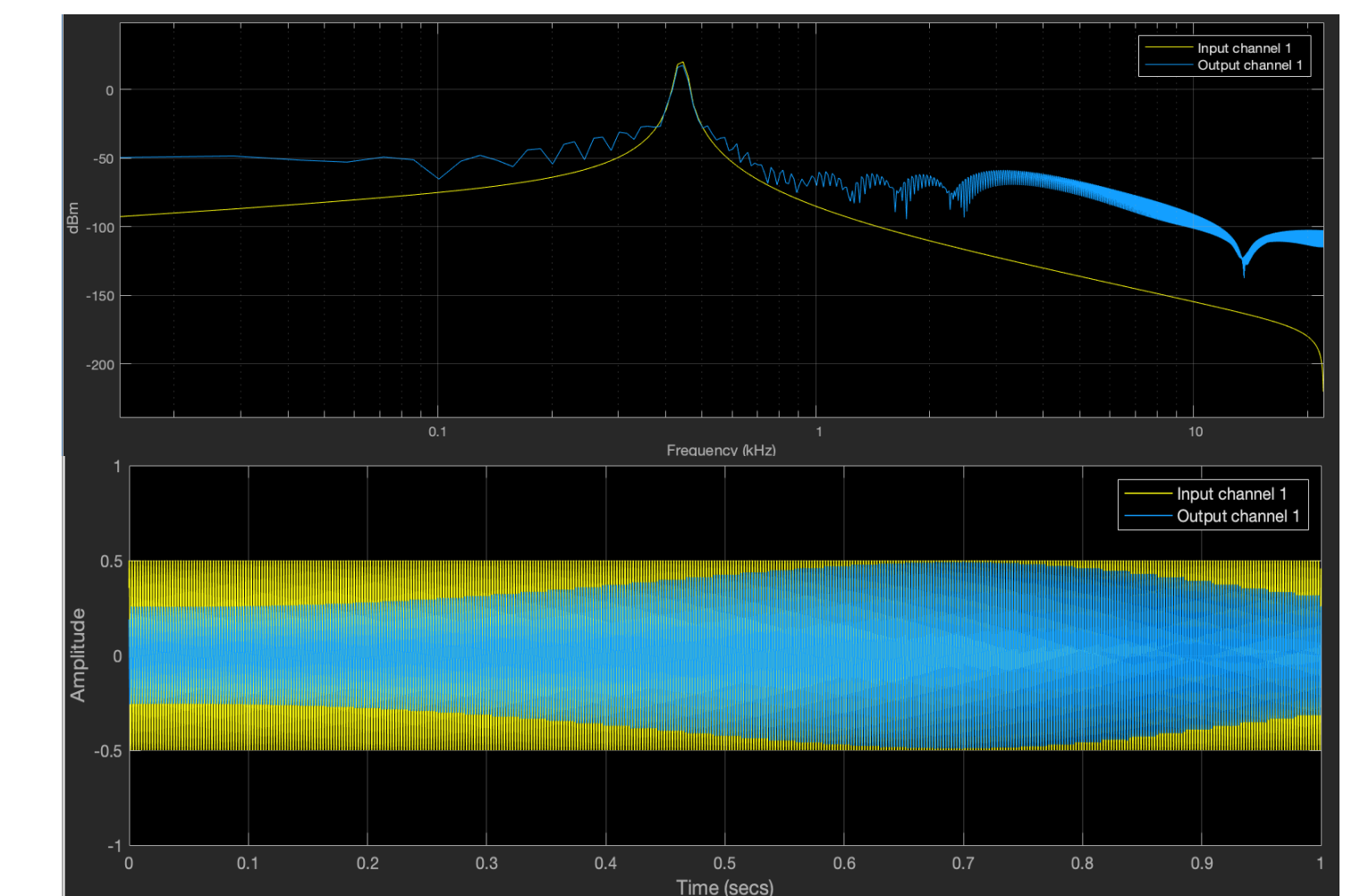


Figure F: Phaser input and output frequency and waveform display

## Discussion

- Each audio effect was tested by inputting a 1kHz sine wave with an amplitude of 0.5
- The frequency spectrum in Figure B shows that the output signal is a comb-filtered version of the input signal, with peaks and troughs that are harmonically related
- The waveform display in Figure D illustrates the randomly modulated delay produced by the chorus effect
- The frequency spectrum in Figure F shows that the output signal is a comb-filtered version of the input signal, though the peaks and troughs are not harmonically related
- Next step in project is to implement functions as Virtual Software Instruments (VST) so they can be used in a DAW

## References

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](http://www.w3.org/2011/audio/audio-eq-cookbook.html)

"Design an Audio Plugin." *Mathworks.com*, The MathWorks, Inc., [www.mathworks.com/help/audio/guide/design-an-audio-plugin.html](http://www.mathworks.com/help/audio/guide/design-an-audio-plugin.html).

"Dsp.VariableFractionalDelay." *Mathworks.com*, The MathWorks, Inc., [www.mathworks.com/help/dsp/ref/dsp.variablefractionaldelay-system-object.html](http://www.mathworks.com/help/dsp/ref/dsp.variablefractionaldelay-system-object.html).

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](http://www.soundonsound.com/sound-advice/q-whats-difference-between-phasing-and-flanging).