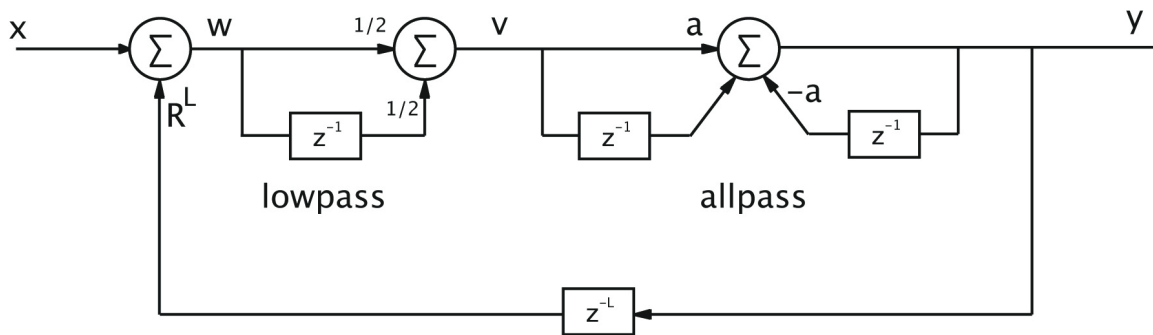


Lab 6

Plucked String Model (Karplus–Strong)

In this lab you are going to create a filter that implements the plucked string model. The signal flow diagram below describes a plucked string model for a fundamental period L , and an allpass coefficient a .



Functions in Matlab

You may have noticed from using functions such as `magPlot`, that it is easy to write our own functions to help organize and modularize our scripts. Until now, we have been running our commands one after another. By writing our own functions we can make scripts that are much more powerful and do a lot less copy and paste while we're at it. The main thing to remember is that functions get their own `.m` file, and the `.m` file needs to have the **same name** as the function. For example, `avg.m` would contain:

```
function m = avg(x,n)
m = sum(x)/n;
end
```

See <http://www.mathworks.com/help/matlab/ref/function.html> if you need more help with functions. If you want to get fancy, it is also possible to nest functions inside of other functions. See me if you want help doing this.

Assignment

Implement the filter using your pro DSP skills in MATLAB. You will have to choose values for R and a . Play around with different values until you find something that sounds good.

1. Derive the difference equation for y_t in terms of R , L and a . **You must submit a full (can be hand written) derivation to me before the due date. Only include the final difference equation in your write up.**
2. Choose a particular frequency and plot the magnitude and phase response of this filter. Make sure to label your plots and axes appropriately (including your selected frequency).
3. Write a function that takes 3 parameters: frequency, sample-rate and an input sound

- and filters the sound through this filter for the selected frequency.
4. Filter an impulse. Write the resulting sound to a .wav file. Plot the full waveform of the resulting sound.
 5. Filter a short noise burst (in the range of 10–20 ms) through the filter at the same frequency. Save the resulting sound to a .wav file. Plot the full waveform of the resulting sound.

Submission Requirements

As you may have noticed, this lab will be slightly different in that the first problem must be completed and handed in **on paper**. This can either be done during one of the lab help sessions, or you can arrange a time to give it to me.

Your write-up should contain:

- (Only) the difference equation for Part 1
- Descriptions of what you had difficulty with, or anything you observed/learned while doing this lab.
- All of the plots asked for.

Hints:

- Be careful with your difference equation. Take the time to work it out and make sure it's right.
- In parts 4 and 5, the actual signal part of the input will be very short, and you will need to concatenate enough zeros to allow the sound to decay naturally.
- Alternately, you could write your function in such a way that the function automatically generates an output signal of the appropriate length. Be sure to document how the function should be used.
- Remember that the `zeros()` command can generate a vector of zeros of any size. If you use the `filter()` command as part of your function for part 3, remember that our filter will be of a rather high order because of the z^{-1} term. Remember to account for this properly.

Lab 6 – The Karplus-Strong String Model Nate Paternoster

Part 1 – Defining the Karplus-Strong filter

The difference equation:

$$y_n = 0.5ax_t + (0.5a + 0.5)x_{t-1} + 0.5x_{t-2} - 0.5y_{t-1} + 0.5aR^L y_{t-L} \\ + (0.5aR^L + 0.5R^L)y_{t-L-1} + 0.5R^L y_{t-L-2}$$

The transfer function:

$$H(z) = \frac{0.5a + (0.5a + 0.5)z^{-1} + 0.5z^{-2}}{1 + az^{-1} + 0.5aR^L z^{-L} - (0.5aR^L + 0.5R^L)z^{-L-1} - 0.5R^L z^{-L-2}}$$

Part 2 – Plotting the magnitude and phase response of the Karplus-Strong filter

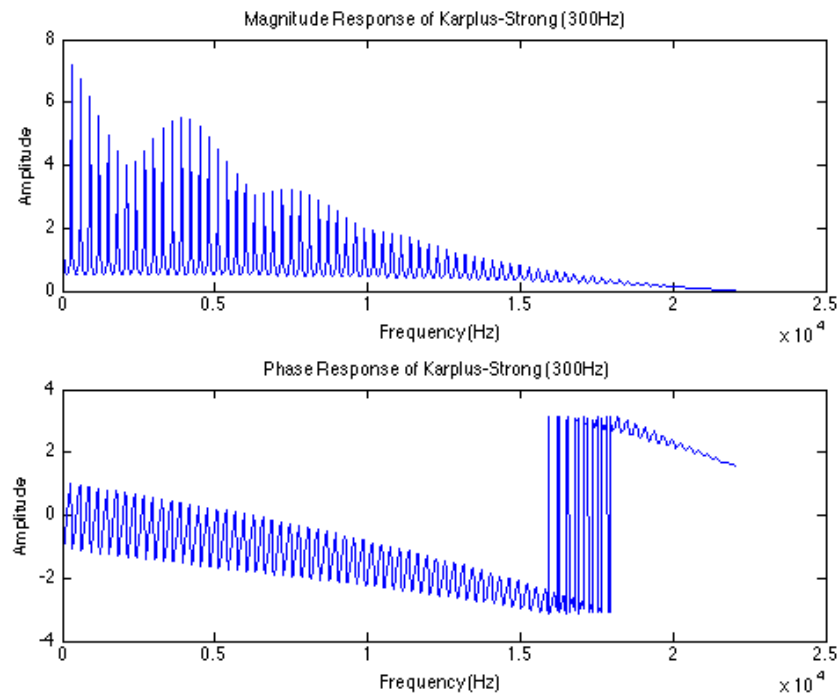


Figure 2a: The magnitude and phase response of the Karplus-Strong filter using a 300Hz input.

Part 3 – Filtering a unit impulse through the Karplus-Strong filter

A unit impulse is filtered through the Karplus-Strong filter and played back. The frequency again chosen for the filter was 300Hz. The filtered signal is played back and plotted.

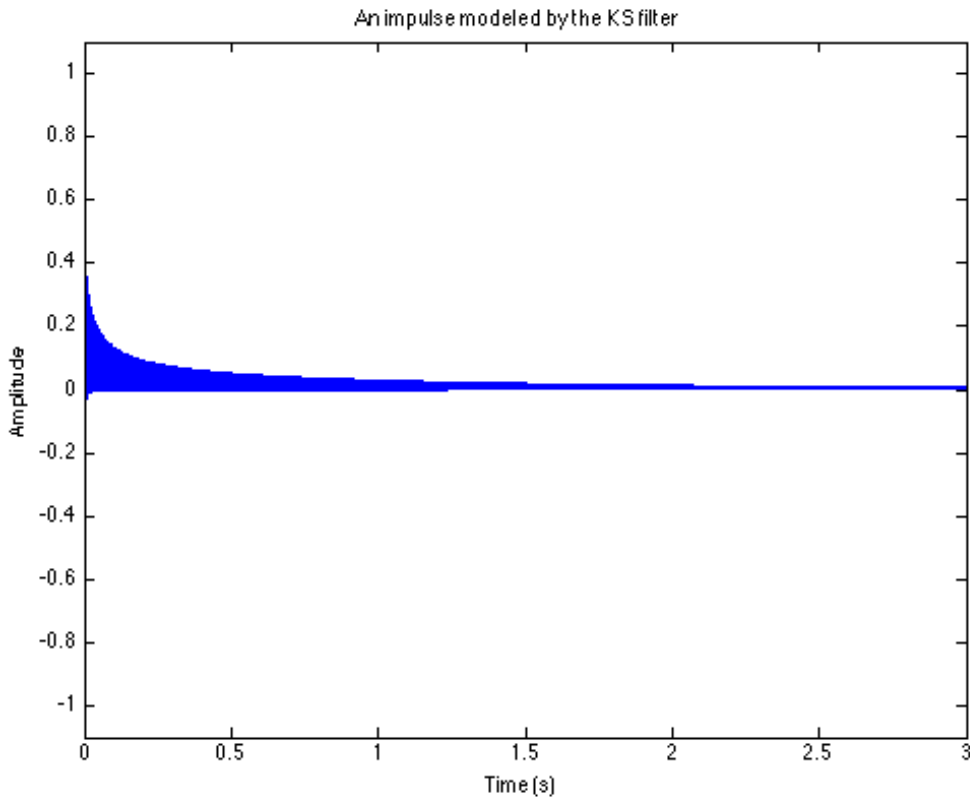


Figure 3a: The impulse response of the Karplus-Strong filter.

Part 4 – Filtering a noise burst through the Karplus-Strong filter

A short noise burst of roughly 10ms is filtered through the Karplus-Strong filter and played back.

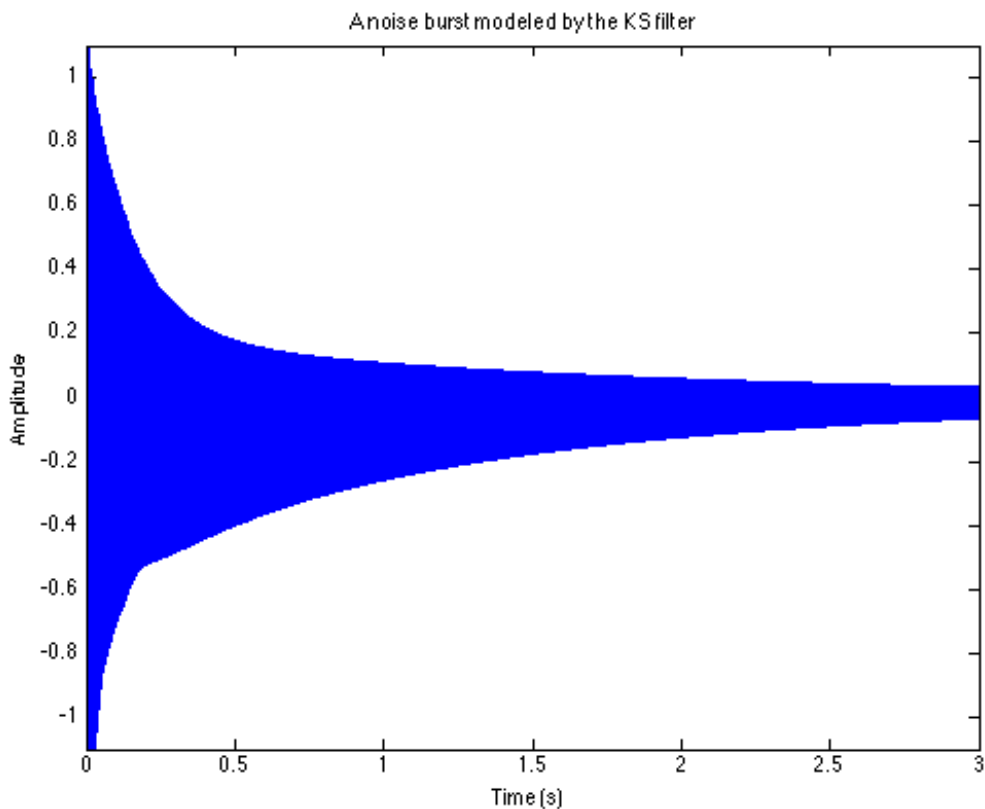


Figure 4a: A short noise burst filtered through the Karplus-Strong filter.

Conclusion

This lab was a good introduction to physical modeling synthesis. It was challenging to implement the correct coefficients when factoring in the variable L delay and to match dimensions when plotting the impulse and noise graphs. However, I discovered that you could use the zeros function to fill in all the coefficients before the L delay terms efficiently and that you could use the sample rate and period to specify the length of time of the signals.