Lab 4

IIR Filters

ECE 406: Real-Time Digital Signal Processing

March 31, 2015

Christopher Daffron

cdaffron@utk.edu

There were two objectives for this lab. The first one was to get hands-on experience with IIR filter design and the second one was to get practice with a circular buffering implementation of a direct form 1 IIR filter.

Task 0

Part A

All of the tasks that are part of this section of the assignment involve doing manual calculation of selected parameters using various methods of generating filter coefficients and then verifying those parameters using MATLAB functions. The first part was to determine the order of a given butterworth filter and the 3 dB cutoff frequency. To determine the order, the following equation was used:

There are two 3 db cutoff frequencies. They are determined using the following equation:

Next, MATLAB was used to verify the calculations. First, the buttord() function was used to find the order and 3 dB frequency of the filter. The code that was used to create this can be found in the appendix. The order of the filter from this calculation was 38. There are several reasons for the difference from the hand calculations. The algorithm used by MATLAB is more accurate than the hand equations and this order is based on an analog filter instead of a digital filter. The 3 dB cutoff frequencies that were found by MATLAB were 0.3476 and 0.6042. These values are pretty close to the calculated values. Next, the butter() function was used to create the analog filter coefficients and the impinvar() function was used to create the digital coefficients. The plot of the frequency response can be found below.

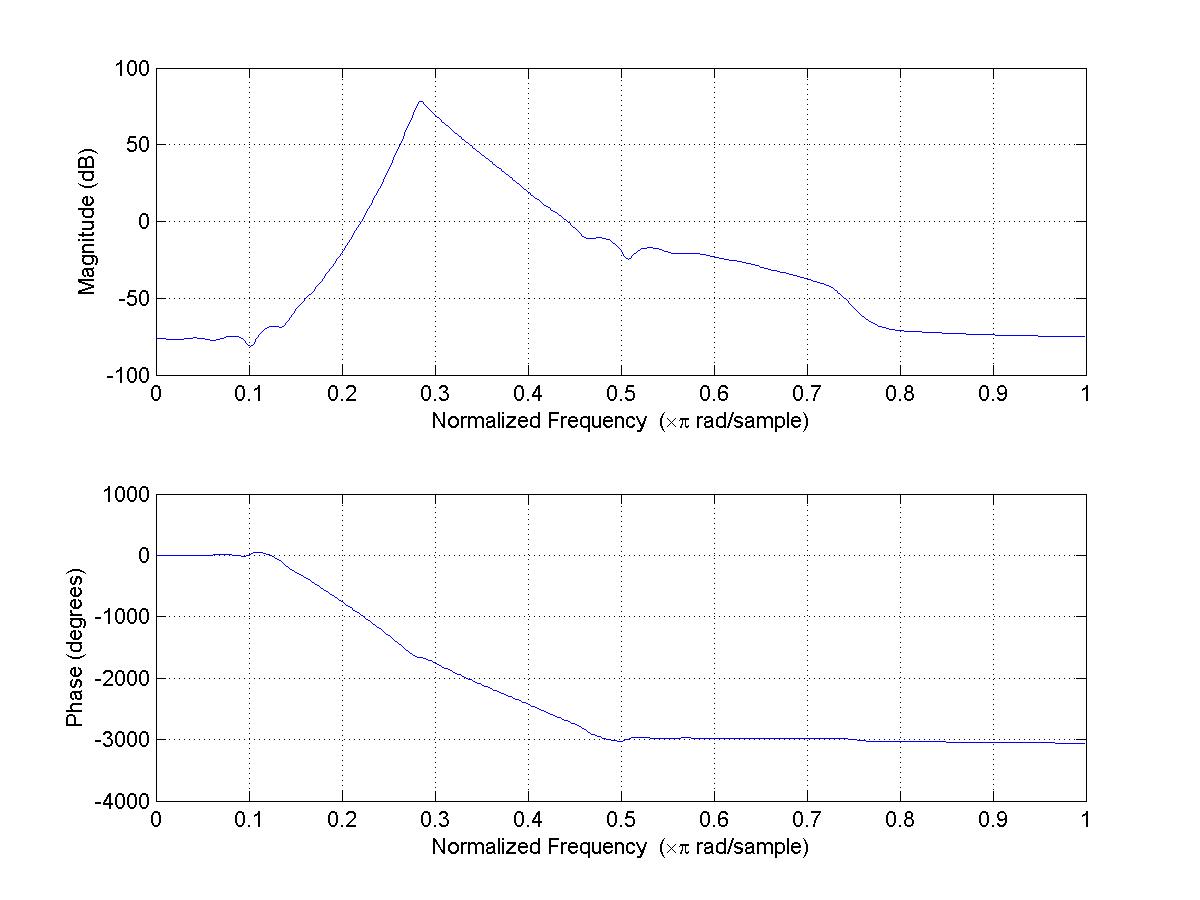


Figure 1: Impulse Invariance Frequency Response

The frequency response of the analog filter coefficients generated by the butter() function can be found below.

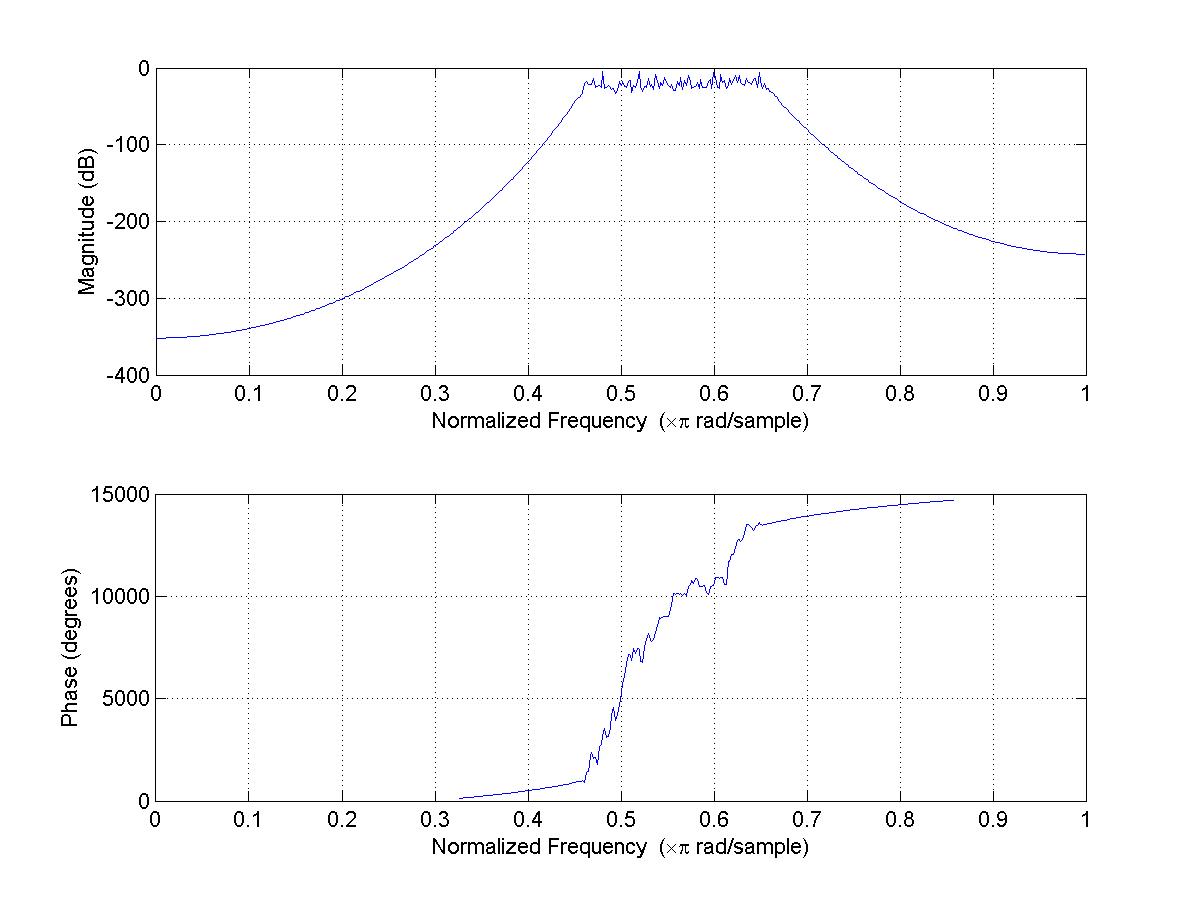


Figure 2: Analog Butterworth Response

This frequency response is much worse than the response of a digital filter created using the butter() function directly, as shown below.

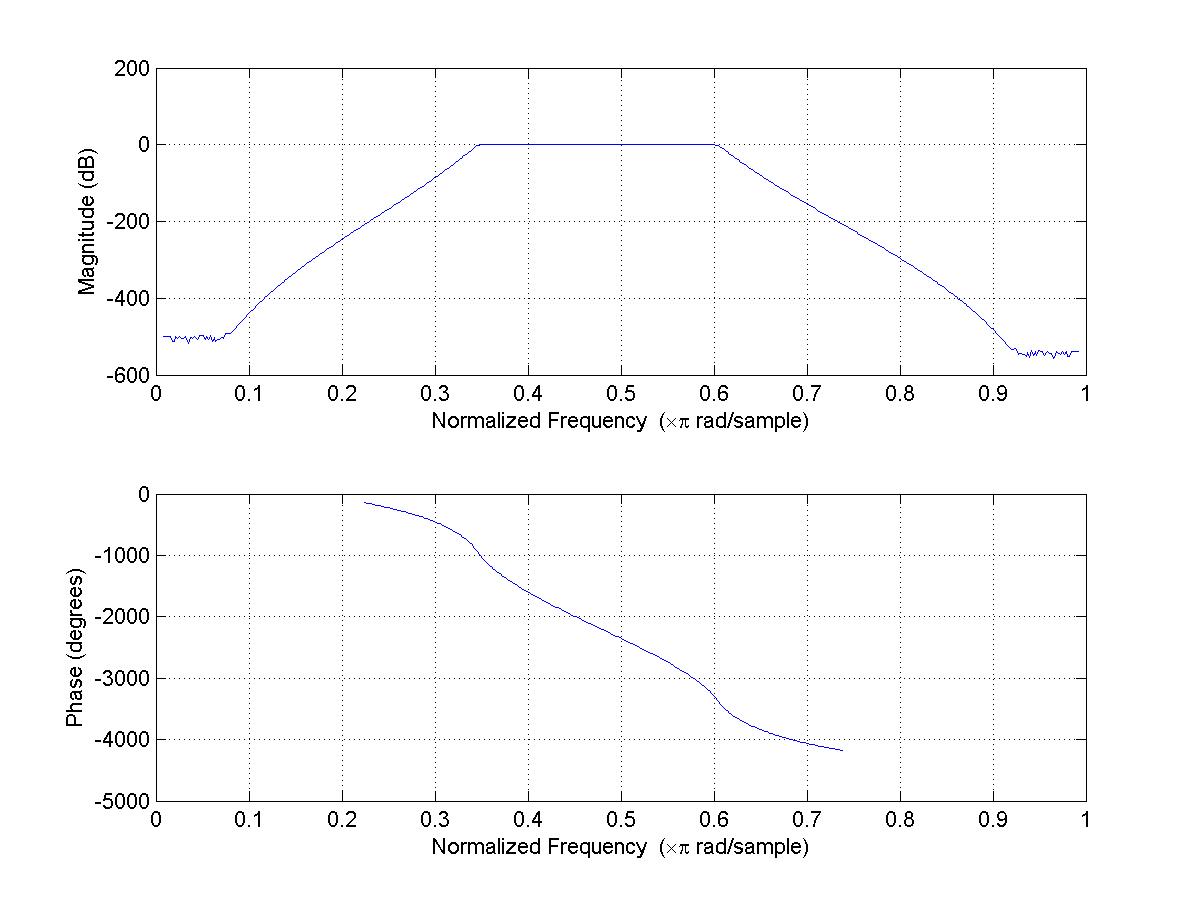


Figure 3: Digital Butterworth Response

After finding the frequency response and filter coefficients manually using the individual MATLAB functions, the fdatool was used to find the coefficients. A screenshot of the tool is shown below.

Figure 4: FDA Tool

Part B

The next part was to use the bilinear transformation. The first task is to determine the order and cutoff frequency of the butterworth filter analytically using the bilinear transformation. The equation that is used to determine the order and cutoff frequency of the filter is independent of whether impulse invariance or bilinear transformation is used, so the equations from the previous part can be used. Next, the MATLAB functions were used. The code for this can be found in the appendix.

[INSERT SOME STUFF HERE]

Finally, MATLAB’s fdatool was used to determine the coefficients. A screenshot of the fdatool can be found below and the coefficients generated can be found in the appendix.

Figure 5: FDA Tool

Next, the appropriate MATLAB functions were used to design an elliptic filter. The code used to generate it can be found in the appendix.

[INSERT SOME STUFF HERE]

Finally, the fdatool was used to design the filter and obtain the coefficients. A screenshot of the fdatool can be found below.

Figure 6: FDA Tool

Task 1

The first task was to measure and graph the theoretical frequency response of all of the generated filters. Those graphs can be found below.

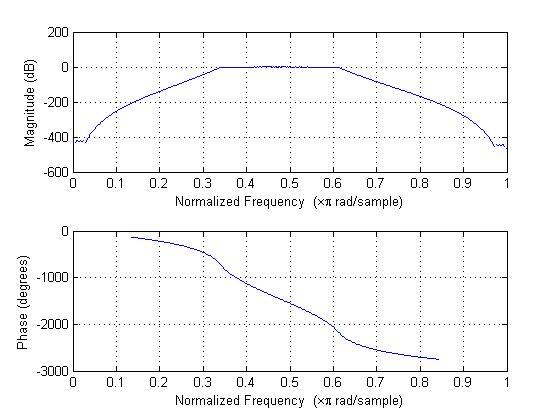


Figure 7: Theoretical Frequency Response – Butterworth Bilinear

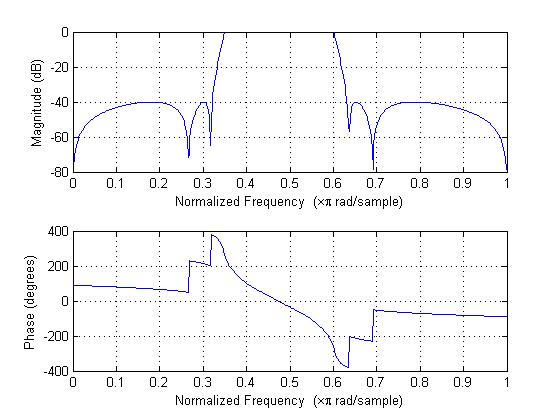


Figure 8: Theoretical Frequency Response – Elliptic Bilinear