Mattin Mir-Tahmasebi (mm5213) and Ahmed Ibrahim (aai13)

February 2016

RTDSP Lab 4 Report

*Real-Time Implementation of FIR Filters*

RTDSP Lab 4 Report

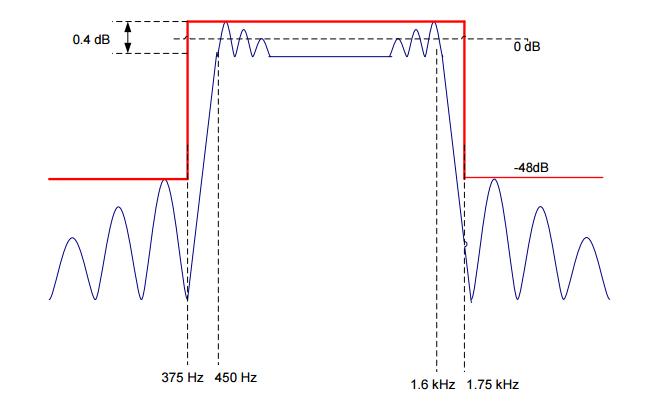
Real-Time Implementation of FIR Filters

# Matlab Filter Design

An FIR filter must be designed using MATLAB. The filter has the following specifications:

* Stop band attenuation of -48dB
* Maximum passband ripple of 0.4dB
* Passband frequencies between 450 and 1750Hz

*Figure 1* below highlights the full specifications for the pass band filter.



*Figure 1 - The full specifications for the pass band filter [1]*

To implement this filter, the Parks-MClellen algorithms is used. This algorithm uses iterative techniques (Remez algorithm) to design the filter to the specifications. Using a simple rectangular filter will produce large errors at the discontinuities due to Gibbs phenomenon. The Parks-MClellen algorithm has the advantage that the errors between the ideal and actual magnitude response can be controlled and minimised in each of the frequency bands.

The algorithm is implemented in MATLAB via the firpmord and firpm functions. firpmord takes parameters that describe the desired cutoff frequencies and amplitudes of the stopbands/passbands of the filter, the allowed deviation from the desired amplitudes, and the sampling frequency. It returns parameters that can be passed to firpm to produce an array of frequency coefficients.

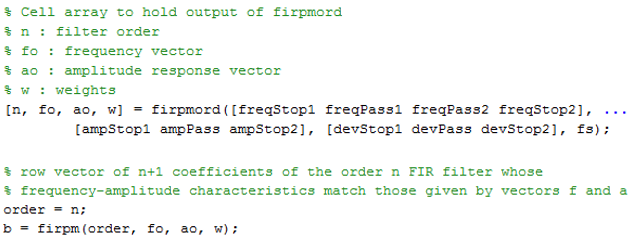
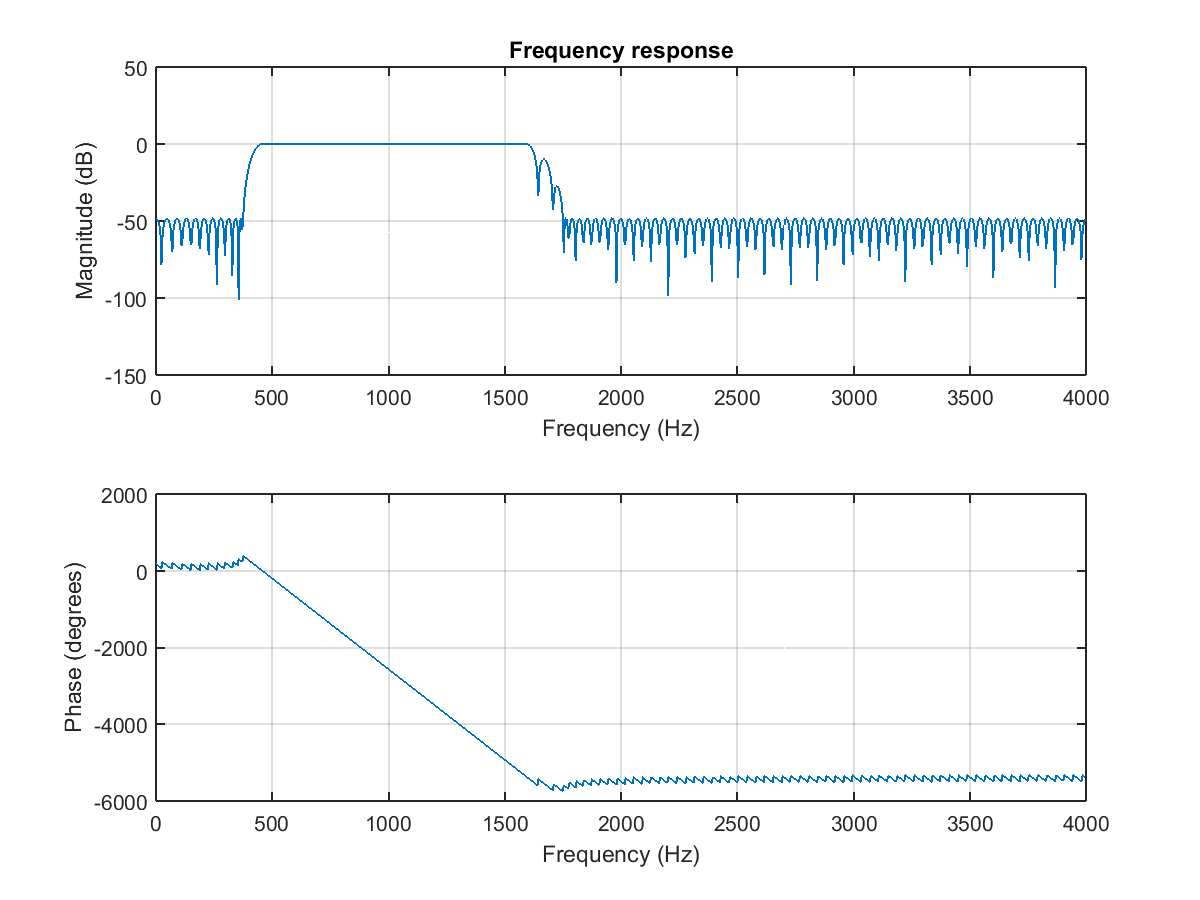


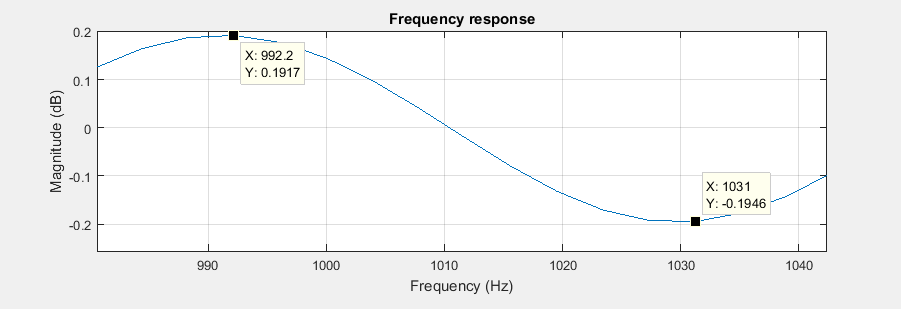
Figure - The firpmord function passes the passband specification as arguments to firpm, which returns an array of frequency coefficients, b.

Our first attempt used the given specifications for the ideal magnitude response of the filter as the parameters to firpmord and produced the magnitude and phase response shown in *Figure 3*. The filter order, however, needed to be increased to n+4 due to this filter giving insufficient attenuation on the stop band (-47.54dB).



*Figure 3 Magnitude and phase responses of the filter using the given specifications*

The new filter now gives good attenuation (-48.27dB) and pass band ripple (found to be 0.386dB) as shown in *Figure 4*. Show diagram of new filter freq response.



*Figure 4 Pass band ripple*

# Non-Circular FIR Filter

An N-tap FIR filter algorithm must be designed. The output is given by:

The Z-transform can be found:

Giving



*Figure 5 Direct form of an FIR filter*

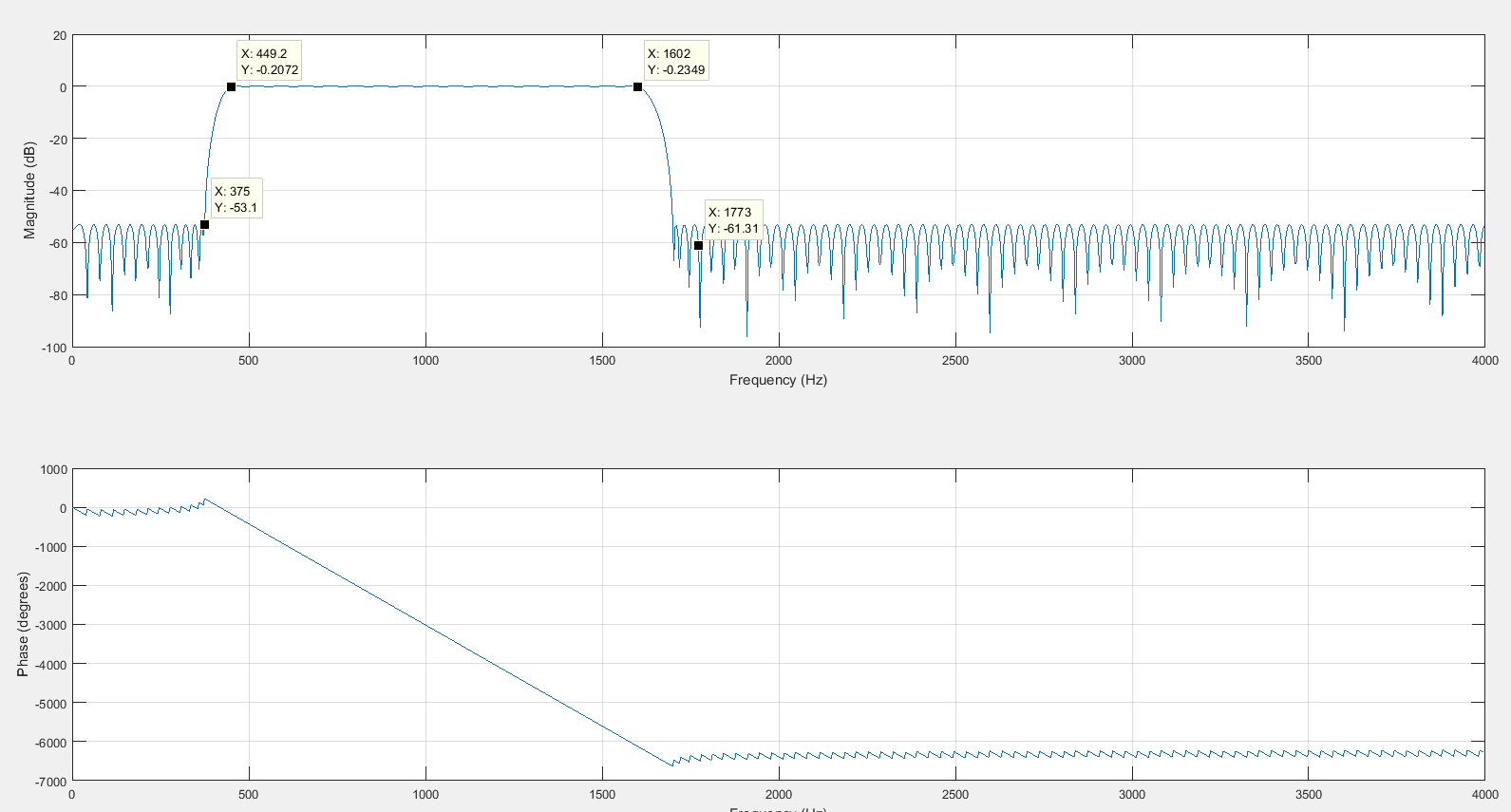
Essentially the Direct Form as shown in *Figure 5* must be implemented. To implement the delay line, an array buffer will be used.

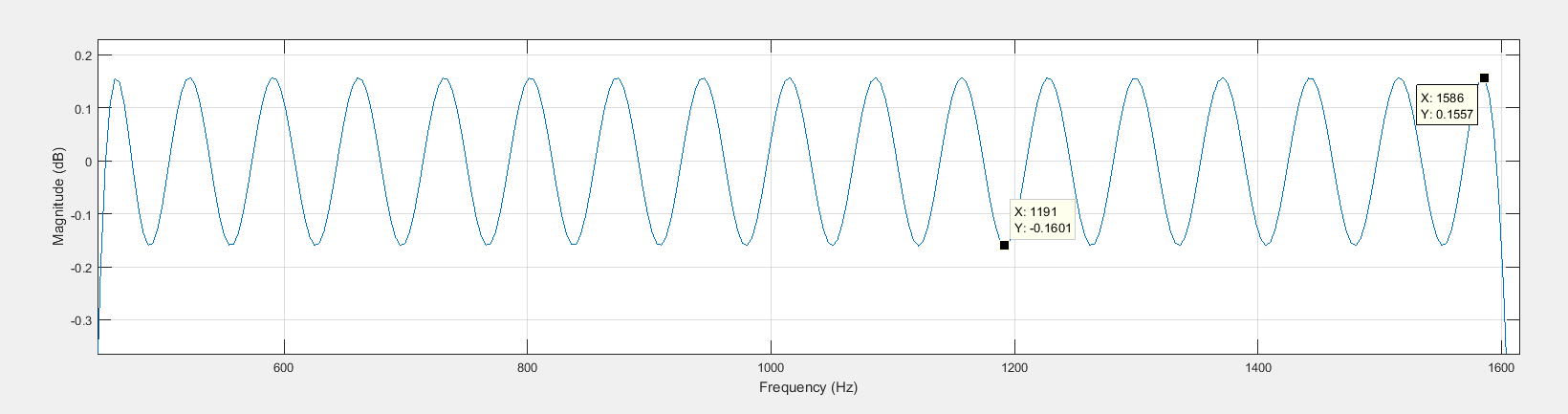
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |

The parameters are slightly altered from the specification, as with the filter that was previously being produced, around the stop band edge of the first transition band, one of the lobes was not matching the specification according to the network analyser, even though it was correct in Matlab, so using some trial and error we reached values that create a more correct result.

This may be due to delay in the ADC/DAC of the DSP board.

Figure 1 below shows the response of the filter adheres to the amplitude specification and Figure 2 demonstrates that the passband ripple is within the 0.4dB required, by showing the amplitude of the highest/lowest points.



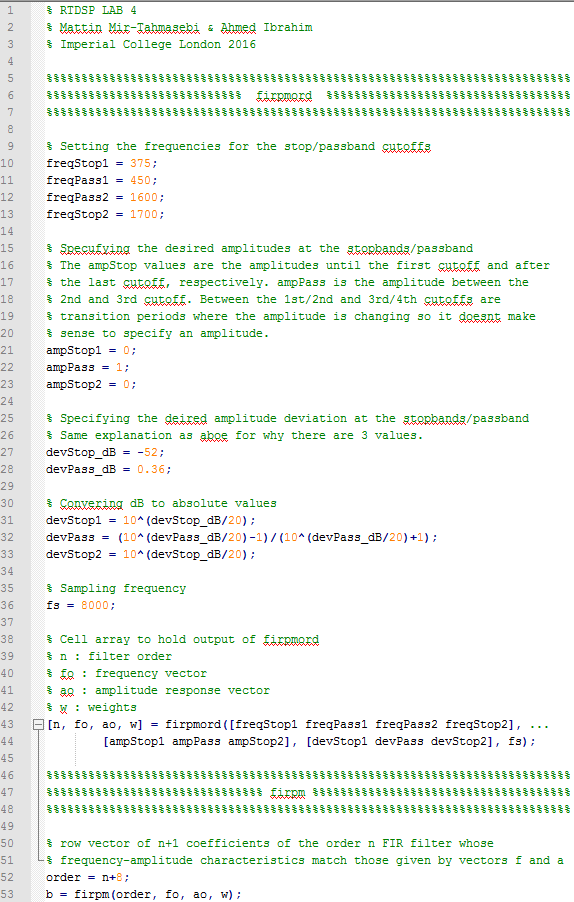


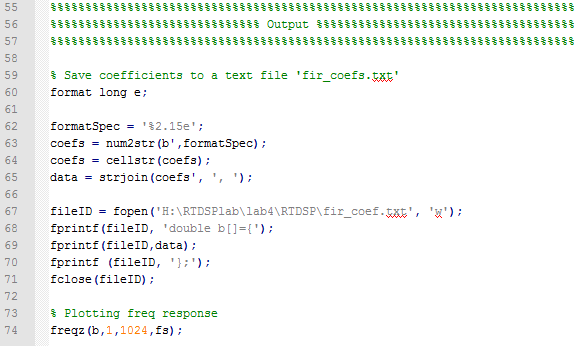
# Circular-Buffer Filters

# Network Analyser Results

Gain is reduced by ~12dB due to : -6 from averaging of left and right channels, -6 from potential divider, -.8 from gain of dsk

### APPENDIX A: filter\_genterator.m





### APPENDIX B: initio.c