

ECE 4020  
Lab Part 1 Report

Ryan Colon

01.24.22

## Introduction

The purpose of this report is to document the design and evaluation of an FIR filter. This FIR filter is intended to replicate the types of low pass filters typically used in voice related applications, which industry standards provide the following specifications: passband frequency up to 3.7 kHz, passband gain of 1.0 to 1.5 dB, stop frequency of 4.3 kHz, and a stopband gain of -50 dB max. For the purpose of this lab the signal will be sampled at a frequency of 48 kHz and the maximum number of coefficients can be no more than 10% of the minimum possible. In the future, this filter will be implemented and tested on both a general-purpose computer with soundcard, and a dedicated digital signal processing chip.

## Design Procedure

This filter was designed in Matlab using the 'firlmord' and 'firpm' commands contained in the Signal Processing toolbox. The 'firlmord' command takes several inputs and using the Parks-McClellan method approximates the optimal parameters for an FIR filter. 'firlmord' takes the following inputs in order:

- Frequency band edges (real vector)
- Desired amplitudes (vector)
- Maximum allowable deviations (vector)
- Sampling frequency (real scalar)

And outputs the following in order:

- Order of filter (integer)
- Normalized frequency points (real vector)
- Amplitude response (real vector)
- Frequency band fit weights (real vector)

The output parameters of 'firlmord' are then used directly as input parameters to the 'firpm' command, which then returns a single real vector that contains all of the coefficients for the filter. The 'firpm' command also uses the Parks-McClellan method to design the filter.

It was mentioned earlier that one of the inputs for 'firlmord' is a maximum allowable deviations vector. For this program the vector is only two values, which represents the deviation for the pass and stop bands. The allowable deviations were calculated based upon the allowable ripple in dB using the following equations:

$$\text{Passband: } \frac{10^{\left\{\frac{R_p}{20}\right\}} - 1}{10^{\left\{\frac{R_p}{20}\right\}} + 1}$$

$$\text{Stopband: } 10^{\left\{-\frac{R_s}{20}\right\}}$$

where  $R_p$  and  $R_s$  represent the pass and stop band ripples respectively. The ripple values used in this program were  $R_p = 0.544$  dB (centered around 1.25 dB) and  $R_s = 52.4$  dB.

Applying all of this into Matlab results in the following code snippet:

```
passbandFreq = 3.7e3;  
stopbandFreq = 4.3e3;  
  
Rp = 0.544; %Passband ripple in dB  
Rs = 52.4; %Stopband ripple in dB (65 is current submitted)  
Fs = 48e3; %Sampling frequency  
F = [passbandFreq stopbandFreq]; %Frequency band edges in Hz  
A = [1.1548 0]; %Band amplitudes  
  
Dev = [(10^(Rp/20)-1)/(10^(Rp/20)+1) 10^(-Rs/20)]; %Calculate ripple  
  
[n,fo,ao,w] = firpmord(F,A,Dev,Fs); %Approximate filter parameters  
b = firpm(n,fo,ao,w); %Design filter
```

*Figure 1. Code snippet to produce filter*

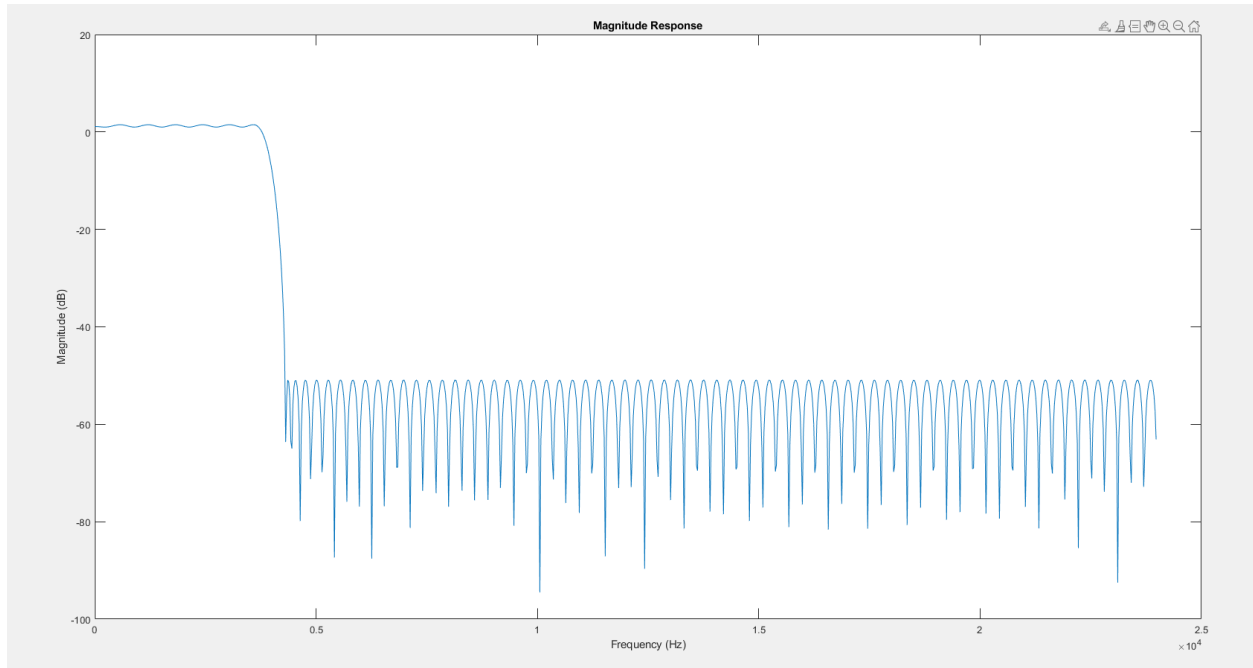
This code is based upon an example provided in the Matlab documentation for the 'firpmord' command (see appendix A).

The values that can be seen in figure 1 will produce the most optimal filter that could be designed given the coefficient limit and filter parameters, but these values were not discovered on the first attempt. The filter started out being way better than the specifications required, and as a result produced a large number of coefficients. Parameters were then tweaked until the number of coefficients couldn't be reduced any further while meeting specifications. This took 4 revisions.

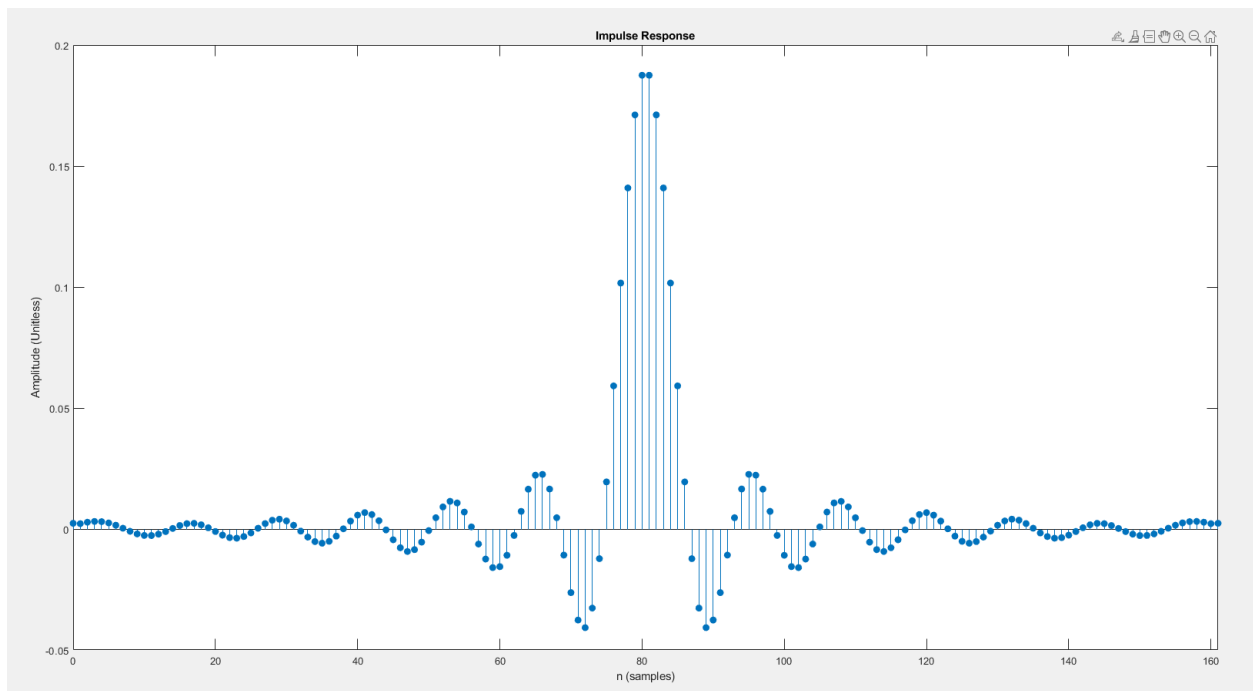
The generated coefficients were then saved to a text file as newline separated values where the first line is text describing the contents of the file. For the complete code implementation used to design the filter, see appendix B.

## Results

Several plots were generated to visually evaluate the filter and ensure it meets passband and stopband specs. All the plots generated are shown below:



*Figure 2. Magnitude-Frequency Response Plot*



*Figure 3. Impulse Response Plot*

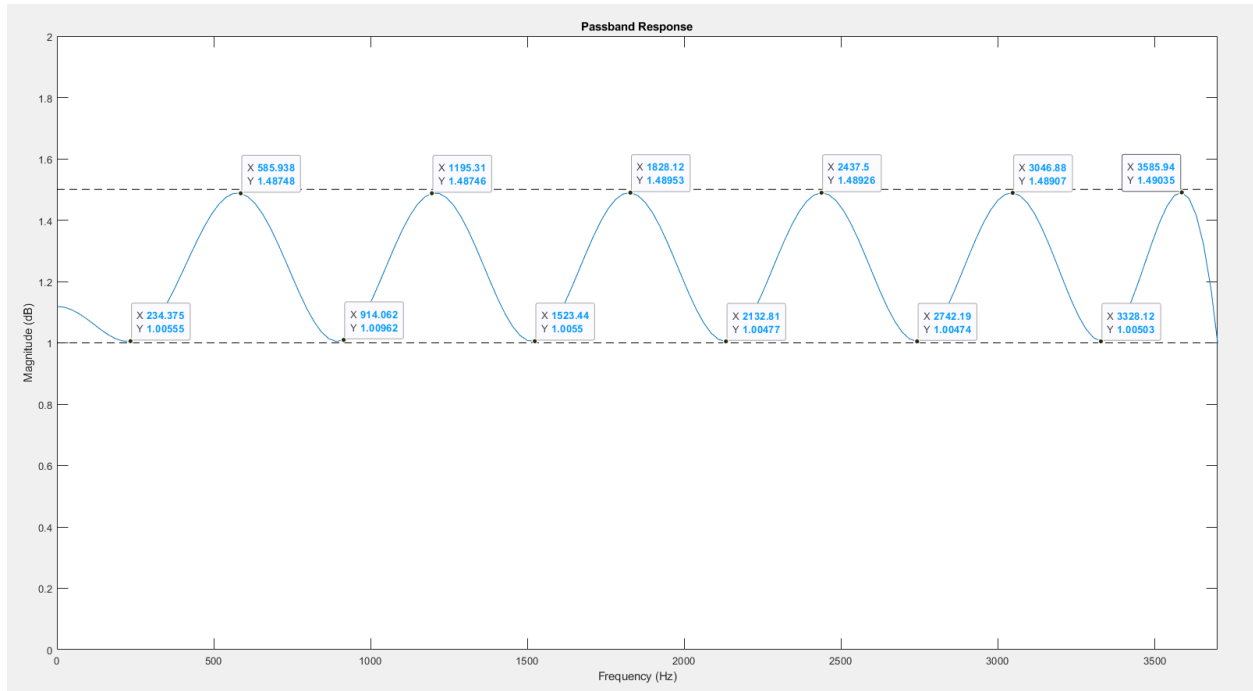


Figure 4. Passband Magnitude-Frequency Response Plot

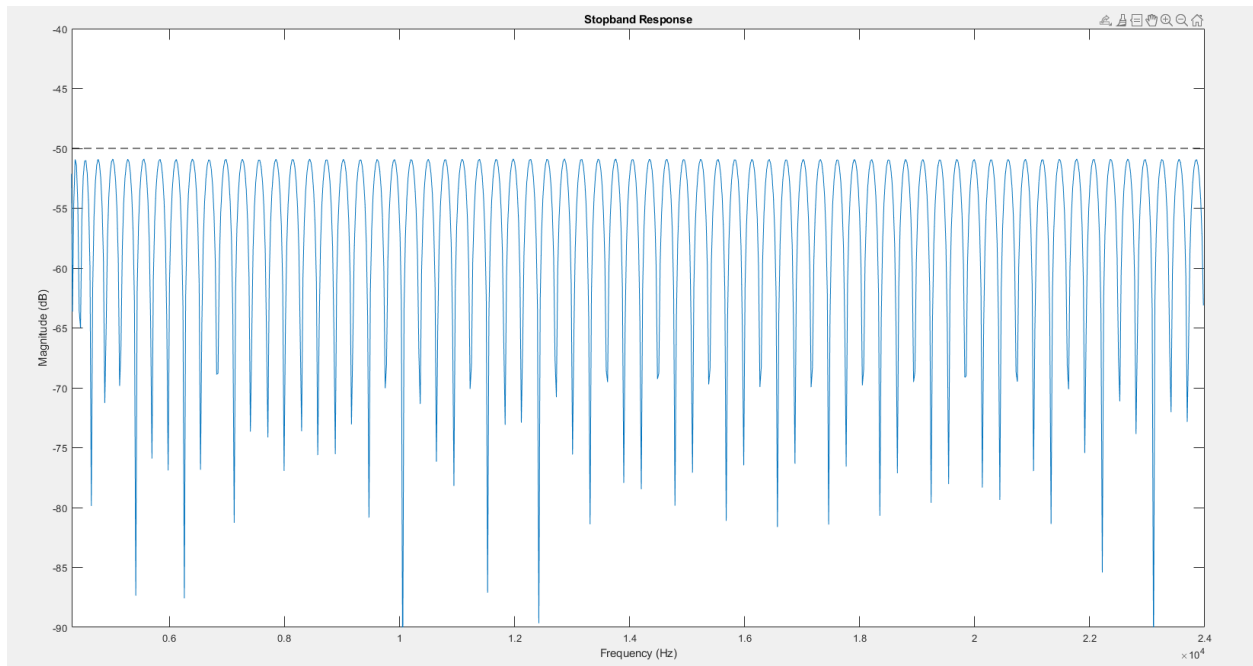


Figure 5. Stopband Magnitude-Frequency Response Plot

Figures 4 and 5 both contain dashed lines which denote the filter specifications. As can be seen in both plots, the filter is within the specified range, and thus meets the desired specs. The only way to verify the minimum number of coefficients spec was to discuss with other engineers and try different combinations of parameters. This yielded

that the minimum number of coefficients that could be derived was 161, which as can be seen from the impulse response plot was the number of coefficients for this filter.

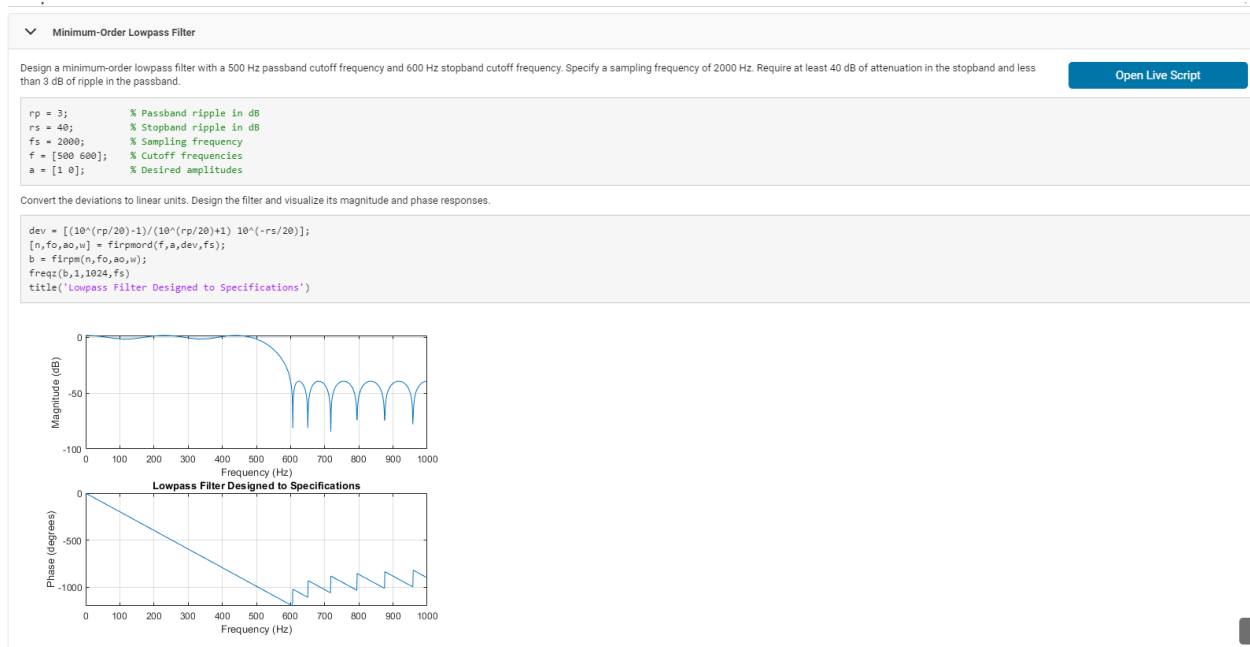
## Conclusion

In this report an FIR filter intended for voice applications was designed and evaluated using tools available in the Matlab signal processing toolbox. It was found that the designed filter meets specifications and can continue to further implementation.

As a concluding note, it may be that for the passband specifications, this filter was over optimized to meet a minimum number of coefficients. While the filter meets specifications, it's allowed ripple allows it to get quite close to the edges of the allowed region. This may mean that in physical implementation, due to a variety of factors that contribute to error, the filter's passband ripple may need to be reduced to allow a greater tolerance for error in implementation.

## Appendix

### A. Firlmord documentation example 1



### B. Complete Matlab code implementation for filter design

```
%DSP_FIR_LPF_Design_Part1
%Ryan Colon
%01.15.22
%Purpose is to create an LPF meeting following specifications
%using firlmord and firlm to do the calculation

%Specifications:
%Passband:
%   Frq: 0-3.7 kHz
%   Gain: +1 - +1.5 dB
%Stopband:
%   Frq: 4.3 kHz
%   Gain: <-50 dB
%Sample Rate: 48 kHz
%Number of coefficients: no more than 10%

%Following code was heavily inspired by an example provided in the
%matlab documentation for firlmord

clear
clc

passbandFreq = 3.7e3;
stopbandFreq = 4.3e3;

Rp = 0.544; %Passband ripple in dB
Rs = 52.4; %Stopband ripple in dB (65 is current submitted)
```

```

Fs = 48e3; %Sampling frequency
F = [passbandFreq stopbandFreq]; %Frequency band edges in Hz
A = [1.1548 0]; %Band amplitudes

Dev = [(10^(Rp/20)-1)/(10^(Rp/20)+1) 10^(-Rs/20)]; %Calculate ripple

[n,fo,ao,w] = firlmord(F,A,Dev,Fs); %Approximate filter parameters
b = firlm(n,fo,ao,w); %Design filter
[h,w2] = freqz(b,1,1024,Fs);
hval = 20*log10(abs(h));
plot(w2, hval)
title('Magnitude Response')
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')

for i = 1:length(hval)
    %check and make sure passband meets spec
    if w2(i) <= passbandFreq && (hval(i) >= 1.5 || hval(i) <= 1.0)
        fprintf('We dont meet passband spec: %f at %d\n', hval(i), w2(i));
        return
    end

    %check and make sure stopband meets spec
    if w2(i) >= stopbandFreq && hval(i) > -50
        fprintf('We dont meet stopband spec: %f at %d\n', hval(i), w2(i));
        return
    end
end

impz(b) %Plot Impulse Response
ylabel('Amplitude (Unitless)')

fprintf("Filter coefficients calculated\n")
fprintf("Number of coefficients: %d", n)
fprintf("\nWriting coefficients to file...\n")

%Write the coefficients to a file for submission
FID = fopen('RyanColon_FIR_Coefficients_Rev4.txt','w');

if FID > 0
    fprintf(FID, "Ryan Colon, FIR Filter, Attempt 4 (Optimizing)\n");
    for i = 1:(length(b)-1)
        fprintf(FID, '%.14f\n', b(i));
    end
    fprintf(FID, '%.14f', b(length(b)));
    fclose(FID);
end

disp("Done")

```