Assignment 9: IIR Filter Design

Colt Thomas

July 27, 2020

Use the MATLAB filterDesigner toolto design a single section direct form digital low-pass filter to satisfy the specifications:

Pass-band edge: 0.45π,$Rp = 0.5$ *dB stop-band edge:* 0.5π, $As = 60dB$

In each design plot the filter response in dB, a pole-zero plot, and the "a"and "b"matrix coefficient values.Note any difficulties and provide likely explanations.

Problem 9.1

Design a Butterworth filter and determine N and the maximum stopband attenuation in dB

Below in Figure 1 is a screenshot of the Filter Designer window. We have a monotone passband with pretty good attenuation in the stopband region. The transition band is rather wide compared to other filters, which could be why we need such a high order filter. Our design as a Single Order section, Direct Form filter ended up having an order of 51. Figure 2 shows our pole-zero plot. I find it interesting that several zeros go off the unit circle. Figure 3 shows a plot of the coefficient values. Notice how high in magnitude the poles are. I think as we compare this filter to the other filters we will see some advantages and disadvantages of each of the filters.

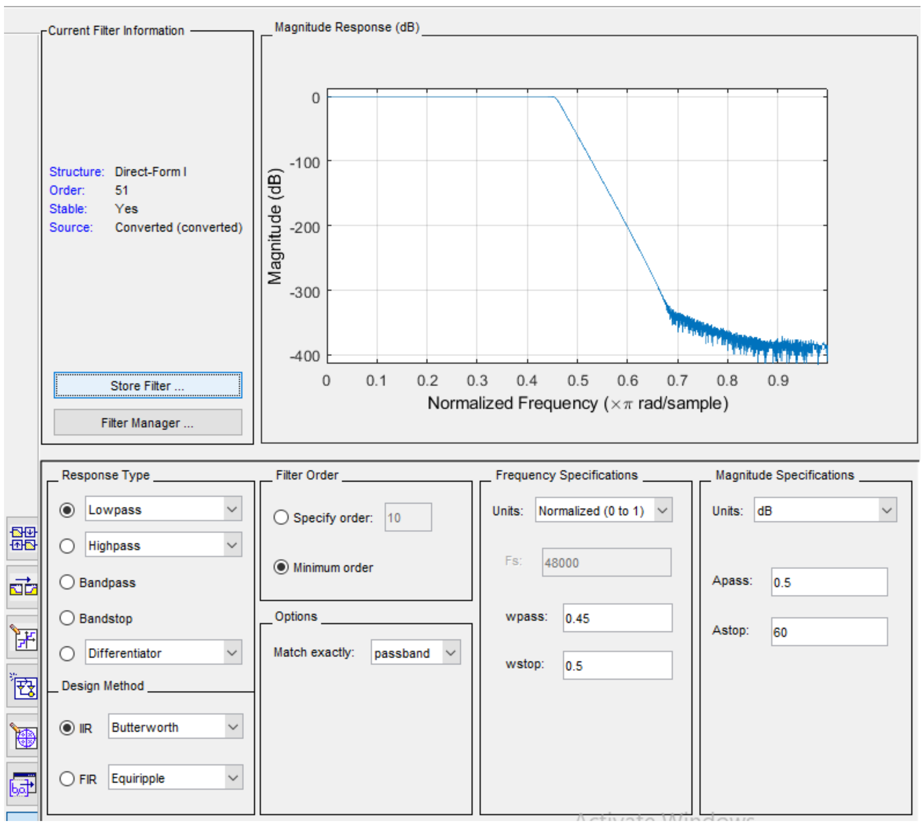


Figure 1: Spectrums of the input signal x compared to filtered output signal y

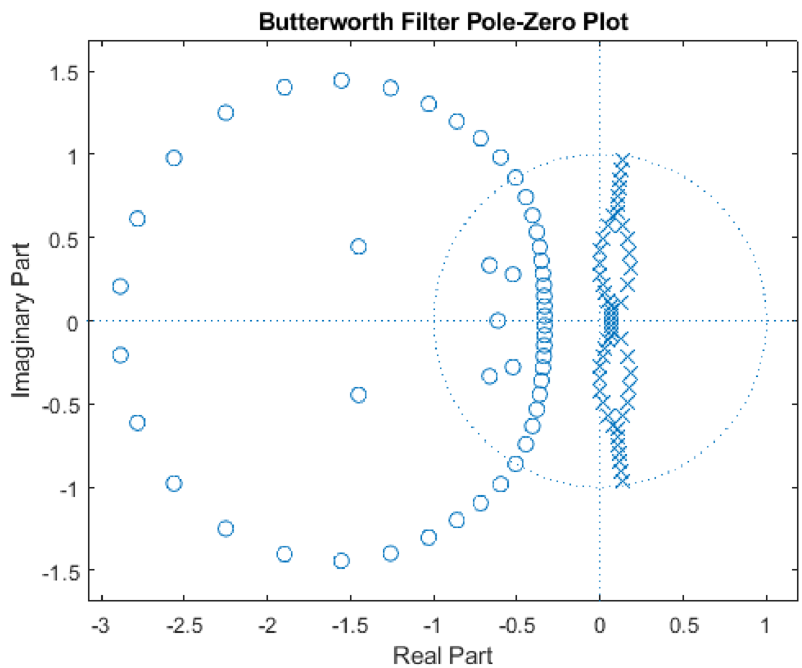


Figure 2: Screenshot of the Filter Designer window, with parameters required shown. The dB magnitude spectrum plot is also included.

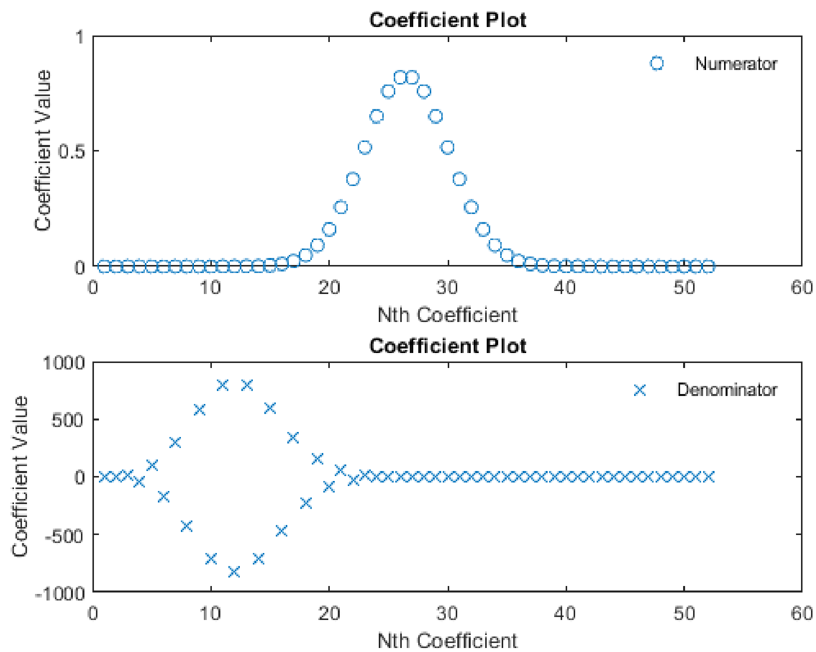


Figure 3: Pole-zero plot of the Butterworth filter design.

Problem 9.2

Design a Chebychev 1 filter and determine N and the maximum stopband attenuation in dB.

Figure 4 has the Filter Designer window with the dB magnitude plot included. This filter has a significantly lower order than the Butterworth filter above (N=16), but it doesn't attenuate as much in the stopband. It is still has a monotone passband though. In Figure 5 we see that the zeros are more tightly packed together than the Butterworth filter, and the poles are spaced out more. This filter would be better against quantization error because of that (but it won't be as great as others unless cascaded). In Figure 6 we notice that the numerator coefficients have a significantly smaller magnitude, while the denominator coefficients have a slightly smaller magnitude.

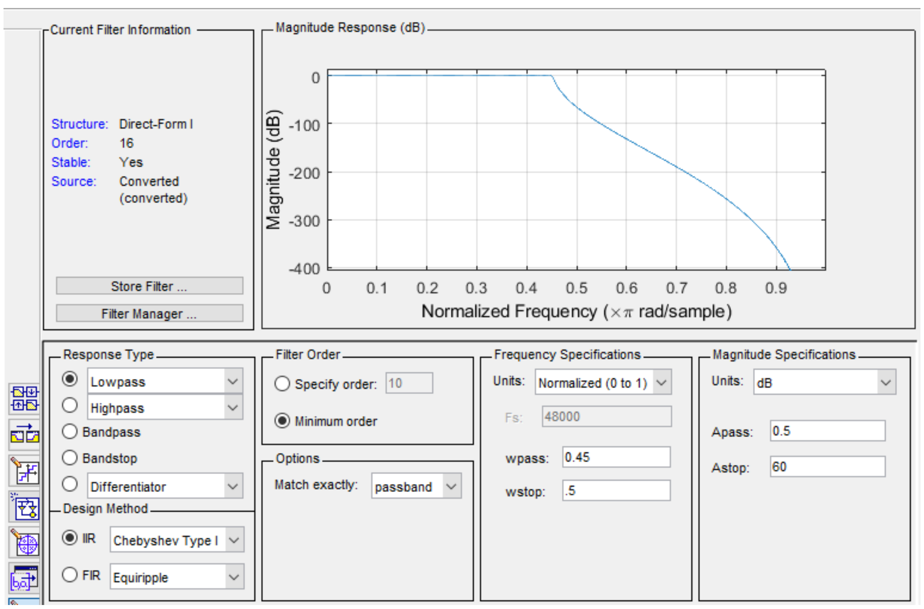


Figure 4: Spectrum of the input signal x compared to filtered output signal y

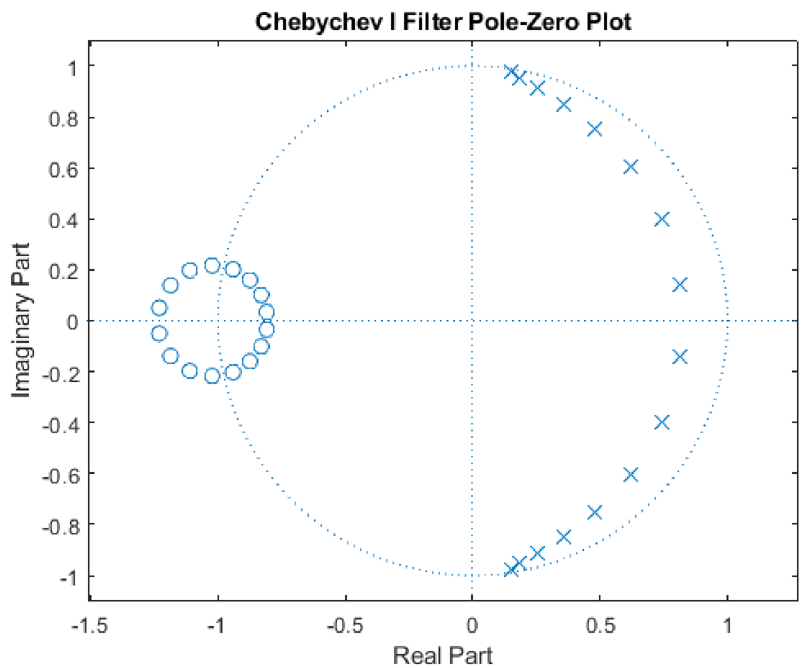


Figure 5: Screenshot of the Filter Designer window, with parameters required shown. The dB magnitude spectrum plot is also included.

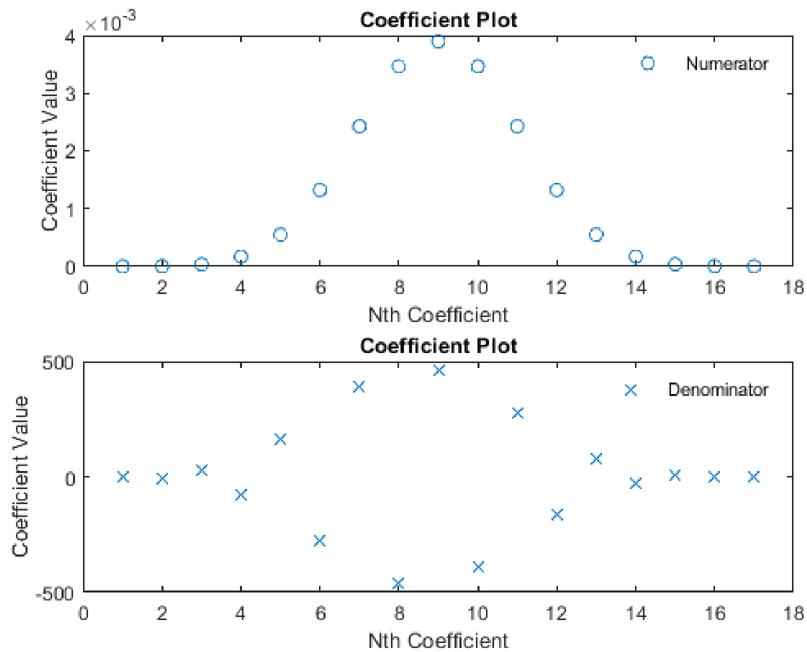


Figure 6: Pole-zero plot of the Chebychev 1 filter design.

Problem 9.3

Design a Chebycheb 2 filter and determine N and the maximum stopband attenuation in dB.

In figure 7 we notice that the filter is quite different from a Butterworth and Chebychev I filter. This time we have an equiripple stopband in addition to a monotone passband. The other stopbands would be significantly attenuated whereas this filter does not attenuate as much, however it does have some nulls. The order is the same as the Chebychev I filter, with an order of N=16. The pole-zero plot is different. Notice that the zeros on the unit circle create those nullspaces. I like to imaging the magnitude spectrum plot as being wrapped around the unit circle starting from $\theta = 0$ to π, being mirrored across the x-axis. The poles serve as "stakes" that attenuate the left hand size of the plane (or right hand size of the spectrum). Figure 9 shows the coefficients, which are very low in magnitude, and all positive in value.

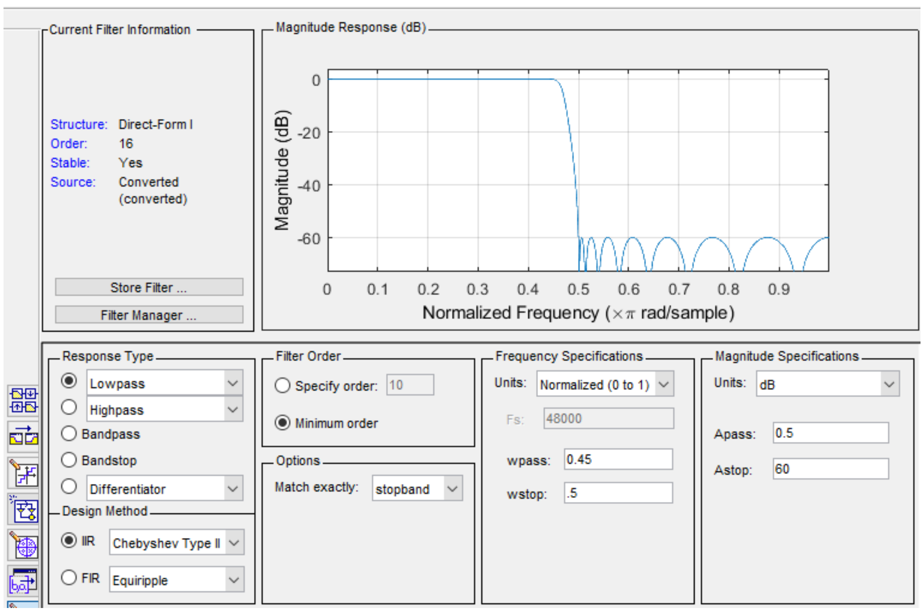


Figure 7: Spectrum of the input signal x compared to filtered output signal y

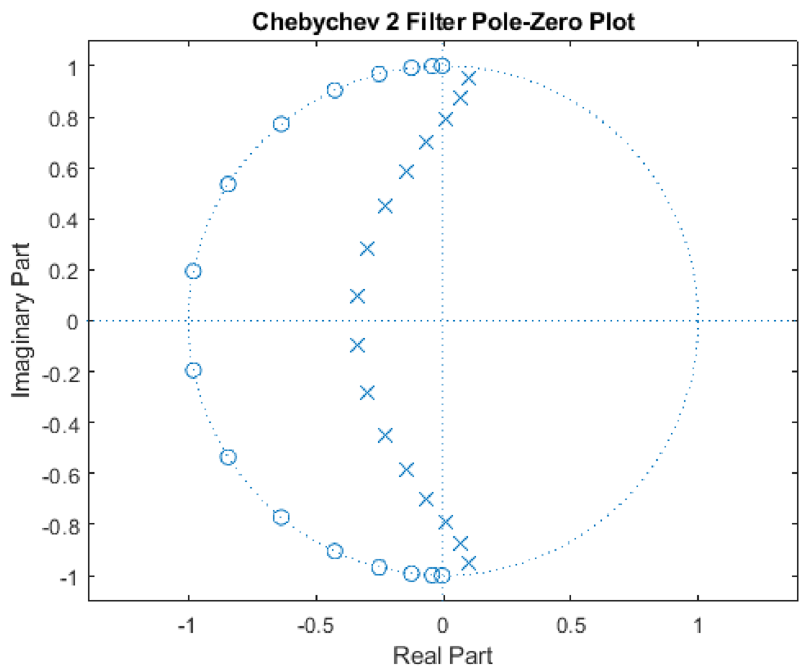


Figure 8: Screenshot of the Filter Designer window, with parameters required shown. The dB magnitude spectrum plot is also included.

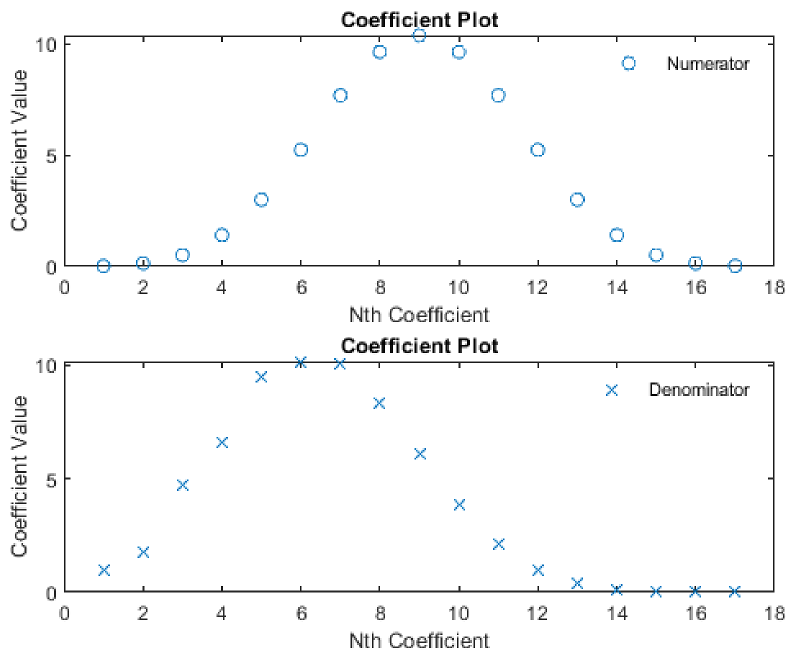


Figure 9: Pole-zero plot of the Chebychev 2 filter design.

Problem 9.4

Design an Elliptic filter and determine N and the maximum stopband attenuation in dB.

So far this is the lowest filter order out of the four, with an order of 8 as seen in Figure 10. Notice how this filter has equiripple in the stopband and the passband (instead of being monotone). The pole-zero plot also shows zeros on the unit circle in Figure 11, which creates those nulls on the spectrum. The filter coefficients in Figure 12 also are small in magnitude.

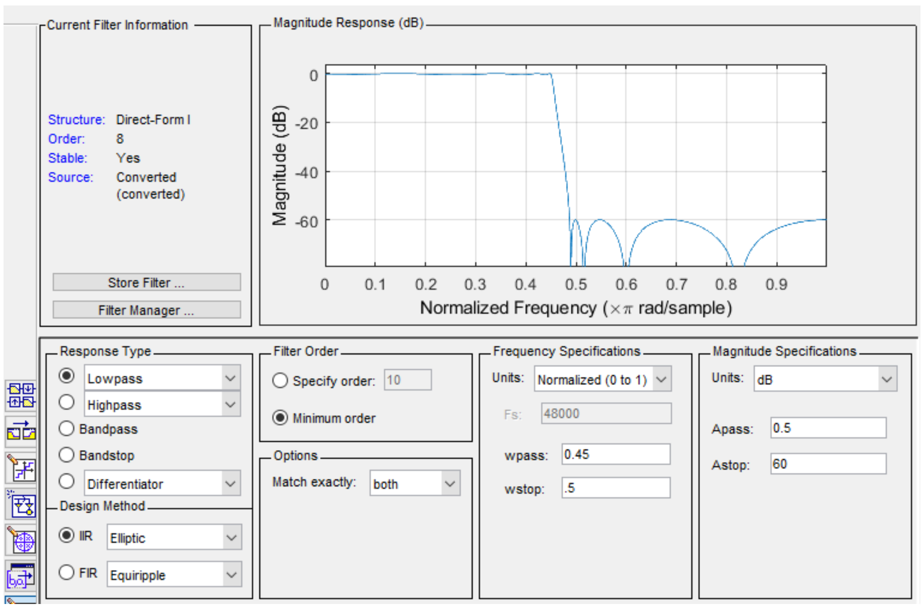


Figure 10: Spectrum of the input signal x compared to filtered output signal y

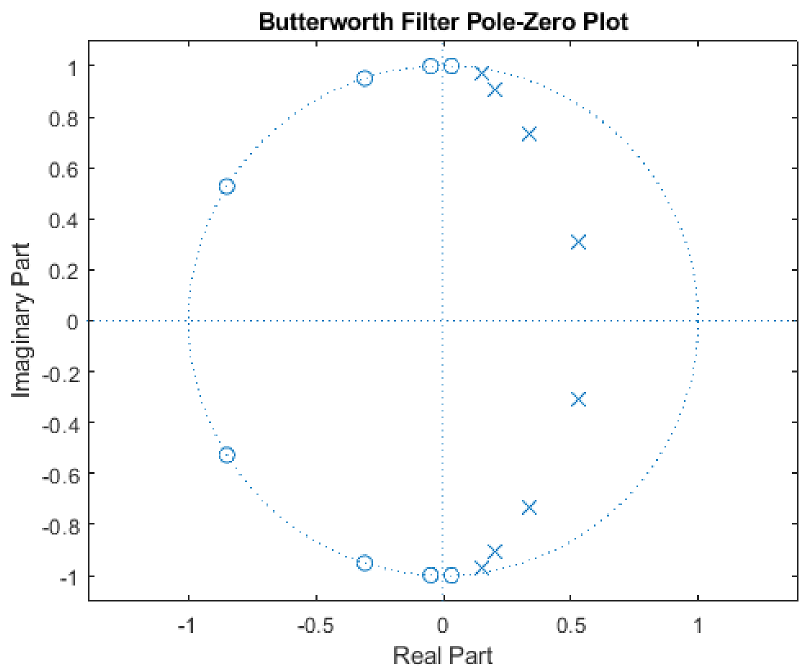


Figure 11: Screenshot of the Filter Designer window, with parameters required shown. The dB magnitude spectrum plot is also included.

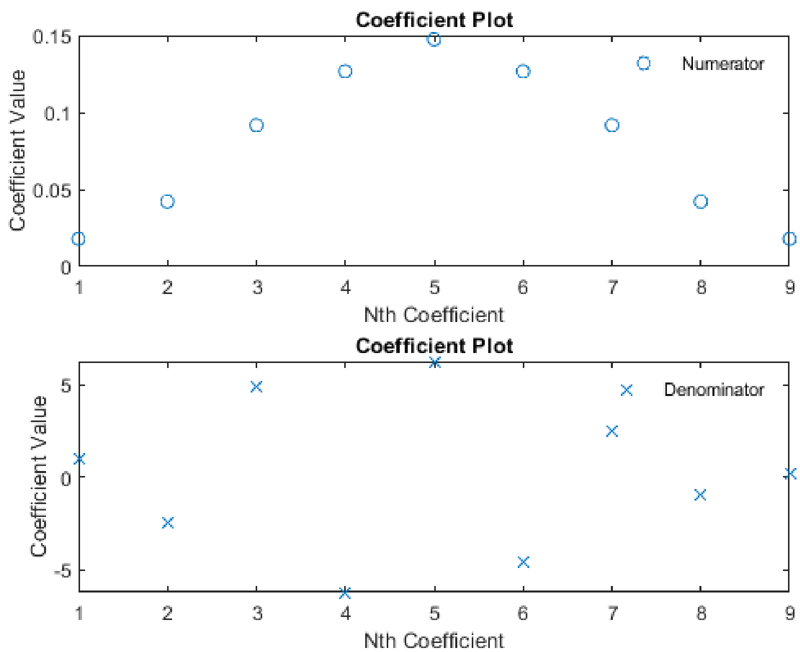


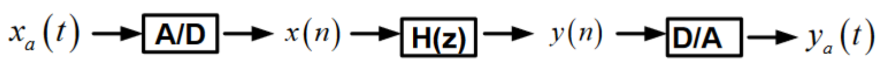
Figure 12: Pole-zero plot of the Elliptical filter design.

Problem 9.5

Create a MATLAB m-file that models this problem:

$$x_a(t) = 3sin(40\pi t) + 3cos(50\pi t)$$

The above analog signal isto be processed by the following system in which the sampling frequency is 100 Samples/sec:



Design a minimum-order IIR digital filter that will pass the first component of x(n) with attenuation of less than 1 dB and suppress the second component to at least 50 dB. The filter should have a monotone pass-band and an equal ripple stop-band. Determine the system function in rational function form and plot the log-magnitude response.

Below is the Matlab code that solves the problem, but I'll make some comments about the figures listed below. Our filter successfully attenuates the 25Hz component, and maintains the magnitude of our desired filter. This is maintained due to the monotone passband. Also note that a lot of the harmonic components were attenuated by about 10 dB. Our final figure is the settings of our filter.

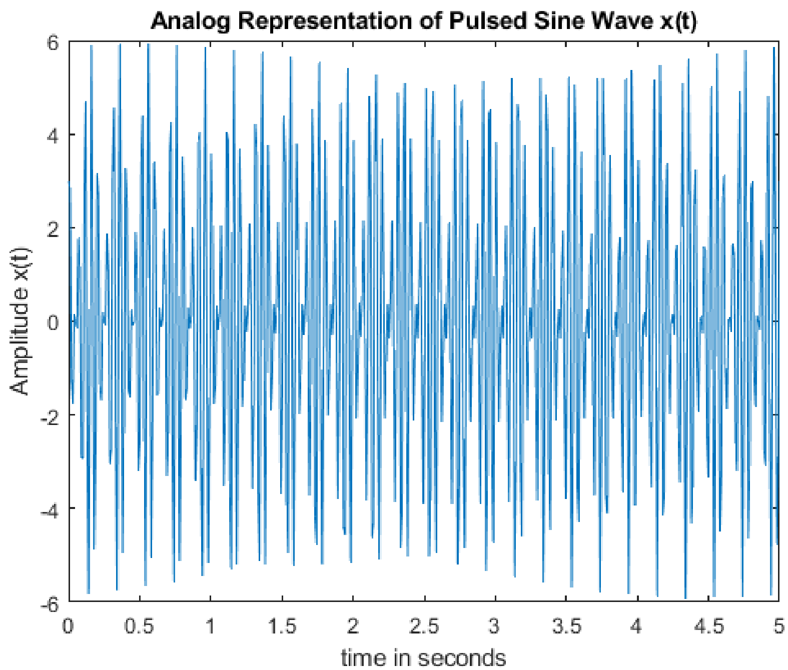


Figure 13: Our initial analog signal x_a. Notice that 20 and 25 Hz have a common multiple, so we see parts of the signal reaching twice the amplitude due to harmonics.

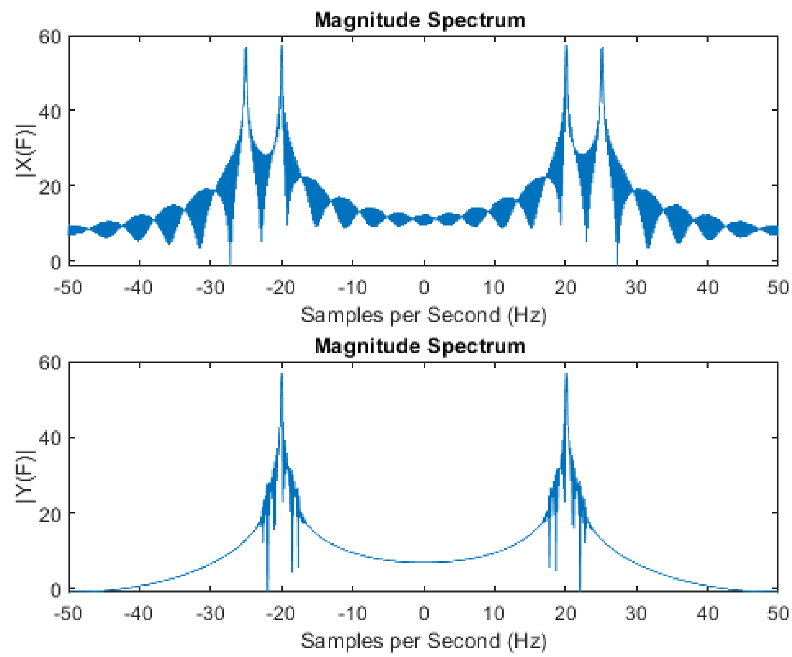


Figure 14: Spectrums of the input signal x compared to filtered output signal y

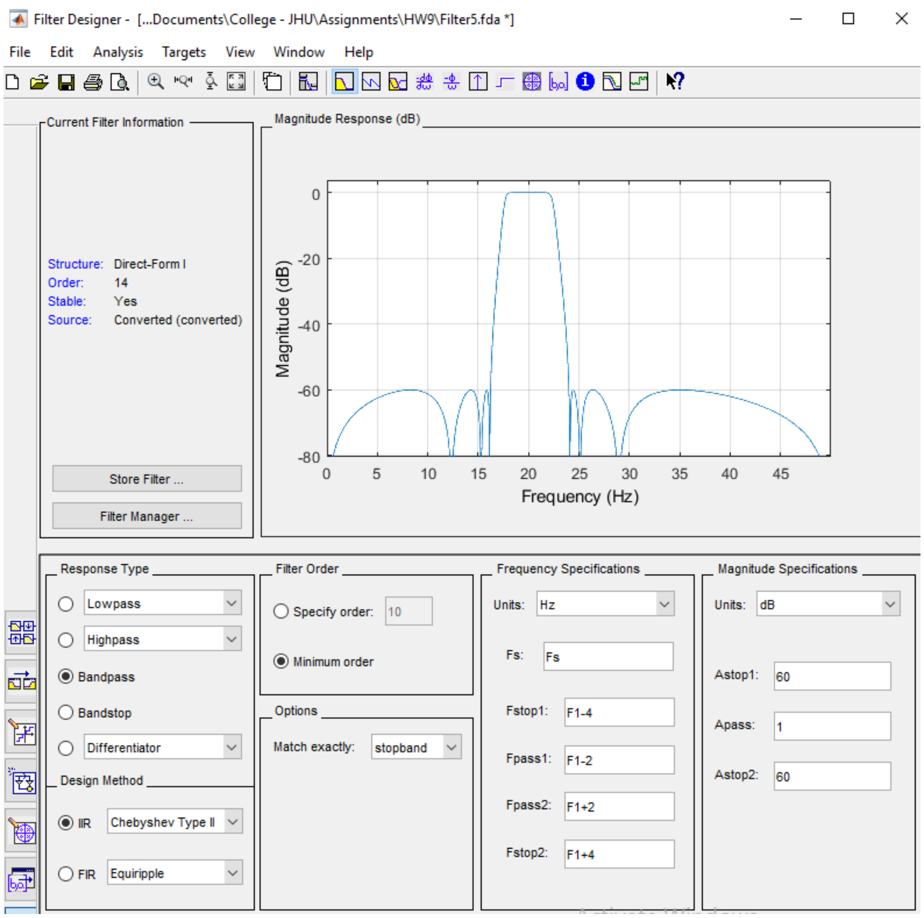


Figure 15: Filter Designer parameters and spectrum of our chosen filter. See above for requirements.

Matlab Code

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Problem 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N = 1024;
[H,W] = freqz(Num1,Den1,N);

% Pole-Zero Plot

figure(1)
zplane(Num1,Den1);
title('Butterworth Filter Pole-Zero Plot')
figure(2)
subplot(2,1,1)
plot(Num1,'o')
title('Coefficient Plot')
ylabel('Coefficient Value')
xlabel('Nth Coefficient')
legend('Numerator')

subplot(2,1,2)
plot(Den1,'x')
title('Coefficient Plot')
ylabel('Coefficient Value')
xlabel('Nth Coefficient')

legend('Denominator')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Problem 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N = 1024;
[H,W] = freqz(Num2,Den2,N);

% Pole-Zero Plot

figure(3)
zplane(Num2,Den2);
title('Chebychev I Filter Pole-Zero Plot')
figure(4)
subplot(2,1,1)
plot(Num2,'o')
title('Coefficient Plot')
ylabel('Coefficient Value')
```

```
xlabel('Nth Coefficient')
legend('Numerator')

subplot(2,1,2)
plot(Den2,'x')
title('Coefficient Plot')
ylabel('Coefficient Value')
xlabel('Nth Coefficient')

legend('Denominator')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Problem 3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N = 1024;
[H,W] = freqz(Num3,Den3,N);

% Pole-Zero Plot

figure(5)
zplane(Num3,Den3);
title('Chebychev 2 Filter Pole-Zero Plot')
figure(6)
subplot(2,1,1)
plot(Num3,'o')
title('Coefficient Plot')
ylabel('Coefficient Value')
xlabel('Nth Coefficient')
legend('Numerator')

subplot(2,1,2)
plot(Den3,'x')
title('Coefficient Plot')
ylabel('Coefficient Value')
xlabel('Nth Coefficient')

legend('Denominator')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Problem 4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N = 1024;
[H,W] = freqz(Num4,Den4,N);

% Pole-Zero Plot

```

```
figure(7)
zplane(Num4,Den4);
title('Butterworth Filter Pole-Zero Plot')
figure(8)
subplot(2,1,1)
plot(Num4,'o')
title('Coefficient Plot')
ylabel('Coefficient Value')
xlabel('Nth Coefficient')
legend('Numerator')

subplot(2,1,2)
plot(Den4,'x')
title('Coefficient Plot')
ylabel('Coefficient Value')
xlabel('Nth Coefficient')

legend('Denominator')




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Problem 5
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%1.Define Fs=100, Ts=1/Fs.
Fs = 100;    % samples/second
Ts = 1/Fs;   % sample period

%2. Sum the two signals and sample the sum using asample time of Ts.
duration = 5; % in seconds
t = linspace(0,duration,duration*Fs); % time vector
F1 = 20;
F2 = 25;
x_a = 3*sin(2*pi*F1*t)+3*cos(2*pi*F2*t);

figure(1);
plot(t,x_a);
title('Analog Representation of Pulsed Sine Wave x(t)')
xlabel('time in seconds');
ylabel('Amplitude x(t)')
%3. Calculate and display the magnitude in dB of the signal x(n) with a
%   N=1024 fft.
```

```
N = 1024;
H_x = fft(x_a,N); % DFT
W = linspace(0,pi,length(H_x));
Xmag = abs(H_x);

figure(2)

subplot(2,1,1)
% plot(W,Xmag);
FF = linspace(-50,50,N); % corresponding freq. axis
plot(FF,20*log10(abs(fftshift(H_x))));
xlabel('Samples per Second (Hz)')
title('Magnitude Spectrum ')
ylabel('|X(F)|')
% set(gca,'XTick',-pi:pi/2:pi)
% set(gca,'XTickLabel',{'-pi','-pi/2','0','pi/2','pi'})


%4. Design a digital filter as a single section direct form structure
%   according to the specification of the problem using the MATLAB
%   filterDesigner tool. Submit the filter design window showing all of the
%   design setting. Export the filter as an object handle into the MATLAB
%   workspace

B = Hd5.Numerator;
A = Hd5.Denominator;

%5. Compute the discrete signal output y(n) and calculate and display the
%   fft(1024)magnitude  in dB6. Comment on the effects of your filtering.

y = filter(B,A,x_a);
H_y = fft(y,N); % DFT
W = linspace(0,pi,length(H_y));
Ymag = abs(H_y);

subplot(2,1,2)
% plot(W,Ymag);

plot(FF,20*log10(abs(fftshift(H_y))));
xlabel('Samples per Second (Hz)')
```

```
title('Magnitude Spectrum ')
ylabel('|Y(F)|')
% set(gca,'XTick',-pi:pi/2:pi)
% set(gca,'XTickLabel',{'-pi','-pi/2','0','pi/2','pi'})
```