Assignment 8: Filter Structures and Design

Colt Thomas

July 21, 2020

Book Questions

Problem 1

Given the following LTI system withinput node $x(n)$ and output node $y(n)$:

$$y(n) = x(n) + 2x(n-1) + 3x(n-2)$$

Part A

Draw the Direct Form I block diagram structure.

This is an FIR filter since it doesn't require any feedback (eg. $y(n-1)$ ... $y(n-k)$). Below in figure **??** is a drawing of the block diagram. The equivalent z transformation of the system is $H(z) = 1 + 2z^{-1} + 3z^{-2}$

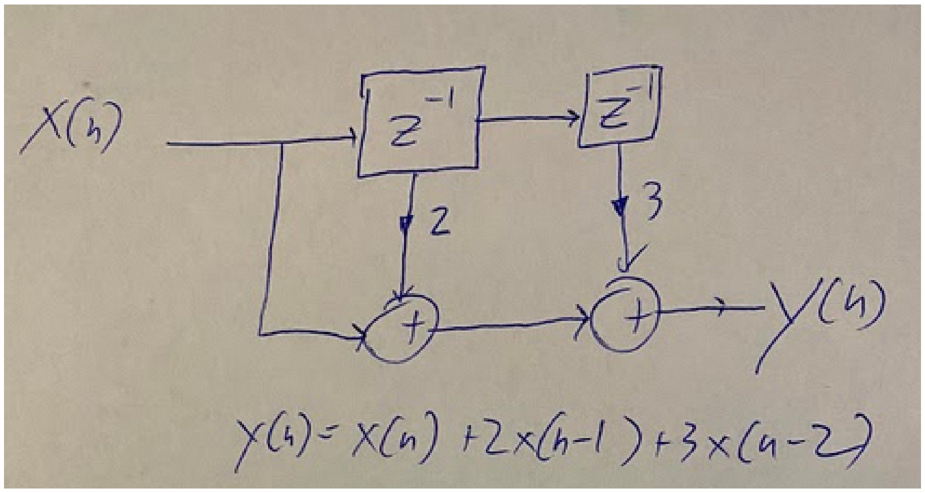


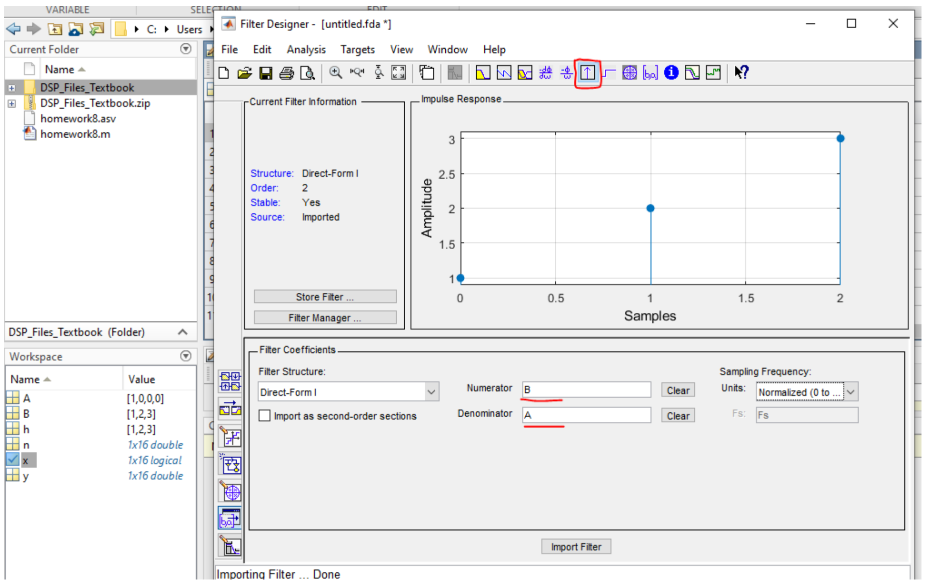
Figure 1: Direct Form I filter strucure of LTI system $y(n) = x(n) + 2x(n-1) + 3x(n-2)$

Part B

Use filterDesigner to analyze the structure in part (a) and document the impulse response.

When we pass in a unit impulse sequence we will get the impulse response $h(n)$. In the Matlab code section for this problem, I created two vectors A and B, which consist of coefficients a_k and b_k in the difference equation $y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$ (see book equation 9.1.1 for details). The Matlab *filter* command requires the A and B coefficients as definition of the system, in addition to the signal being passed into it. Our filter function on line 21 returns a vector y = [0,0,0,0,0,1,2,3,0,0,0,0,0,0,0,0]; notice that it is shifted because our unit impulse is $\delta(n-5)$.

The filterDesigner can import our A and B coefficients and perform an impulse response. In figure 2 the impulse response button is boxed in red, with the underlined filter coefficients underlined in red. Notice that filterDesigner can use any vector saved in the workspace.



Current Filter Information

Structure: Direct-Form I
Order: 2
Stable: Yes
Source: Imported

Workspace

Name	Value
A	[1,0,0,0]
B	[1,2,3]
h	[1,2,3]
n	1x16 double
x	1x16 logical
y	1x16 double

Filter Coefficients

Filter Structure: Direct-Form I

☐ Import as second-order sections

Numerator: B

Denominator: A

Sampling Frequency:
Units: Normalized (0 to ...
Fs: Fs

Importing Filter ... Done

Figure 2: Use of Filter Designer to perform an impulse response given our coefficients. Note that the filter structure is set to *Direct Form I*

Problem 2

Given the following LTI system withinput node x(n) and output node y(n)

$$H(z) = \frac{1}{1 - 1.7z^{-1} + 1.53z^{-2} - 0.648z^{-3}}$$

Part A

Draw the Direct Form I block diagram structure. Notice that this is the z-transform of the impulse response $h(n)$. From the book equation 9.1.2, we can derive our A and B coefficients from the system above as A = [1,-1.7,1.53,-0.648] and B=[1]. The impulse response can be found by:

$$H(z) = X(z)/Y(z)$$

$$H(z) = \frac{1}{1 - 1.7z^{-1} + 1.53z^{-2} - 0.648z^{-3}}$$

$$Y(z) = X(z)$$

$$y(n) - 1.7y(n-1) + 1.53y(n-2) - 0.648y(n-3) = x(n)$$

$$y(n) = 1.7y(n-1) - 1.53y(n-2) + 0.648y(n-3) + x(n)$$

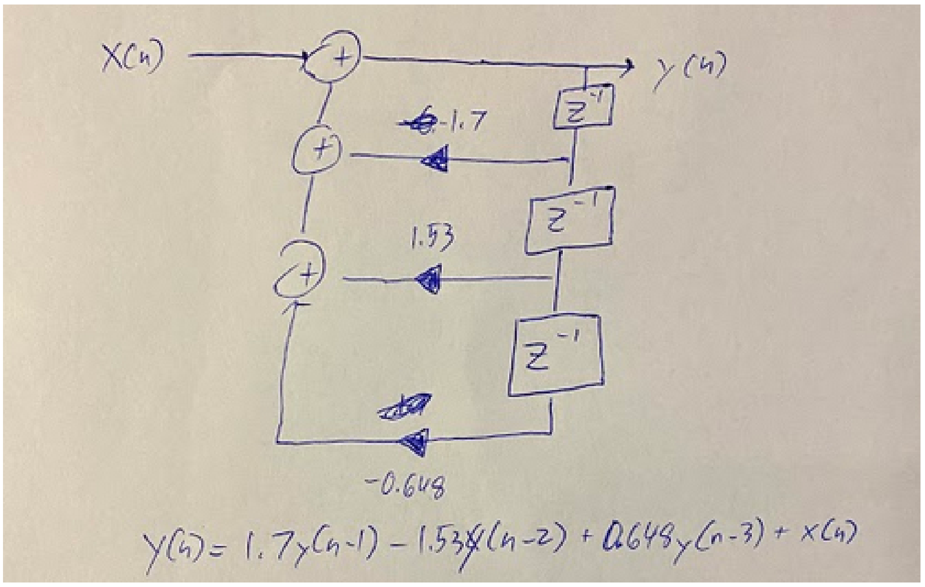
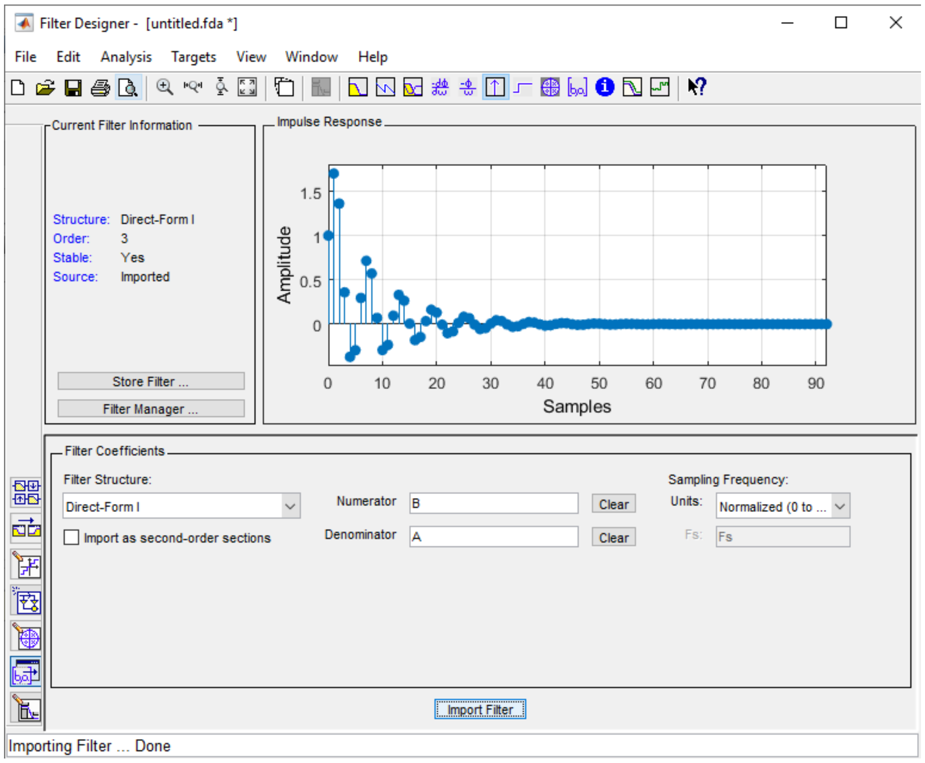


Figure 3: Sketch of system $y(n) = 1.7y(n-1) - 1.53y(n-2) + 0.648y(n-3) + x(n)$

Part B

Use filterDesigner to analyze the structure in part (a) and document the impulse response.

This time we are working with a system that has feedback (IIR filter). The impulse response takes a bit longer than the filter length to settle back to zero due to the internal state registers.



Structure: Direct-Form I
Order: 3
Stable: Yes
Source: Imported

Filter Structure: Direct-Form I
Import as second-order sections
Numerator: B
Denominator: A
Sampling Frequency:
Units: Normalized (0 to ...
Fs: Fs

Importing Filter ... Done

Figure 4: Use of Filter Designer to perform an impulse response given our coefficients. Note that the filter structure is set to *Direct Form I*

Problem 3

Given the following LTI system with input node $x(n)$ and output node $y(n)$:

$$H(z) = \frac{1 - 3z^{-1} + 3z^{-2} + 1z^{-3}}{1 + 0.2z^{-1} - 0.14z^{-2} + 0.44z^{-3}}$$

Part A

Draw the Direct Form I block diagram structure.

Convert from the z-transform to a difference equation:

$$\frac{X(z)}{Y(z)} = \frac{1 - 3z^{-1} + 3z^{-2} + 1z^{-3}}{1 + 0.2z^{-1} - 0.14z^{-2} + 0.44z^{-3}}$$

$y(n) = -0.2y(n-1) + 0.14y(n-2) - 0.44y(n-3) + x(n) - 3x(n-1) + 3x(n-2) + x(n-3)$

Just like the previous problems, we can convert this easily to Direct Form I. See Figure 5.

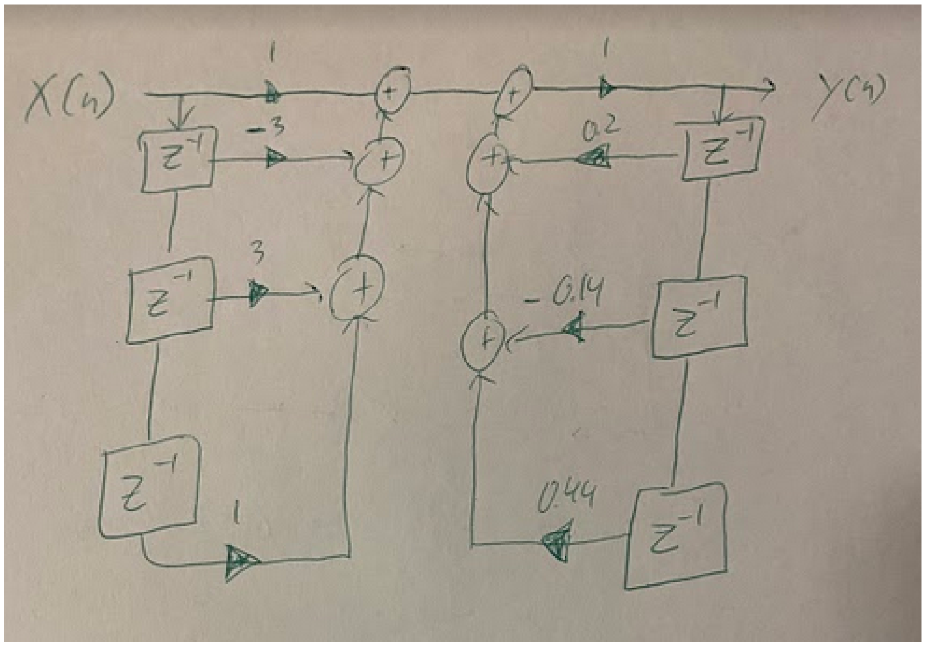
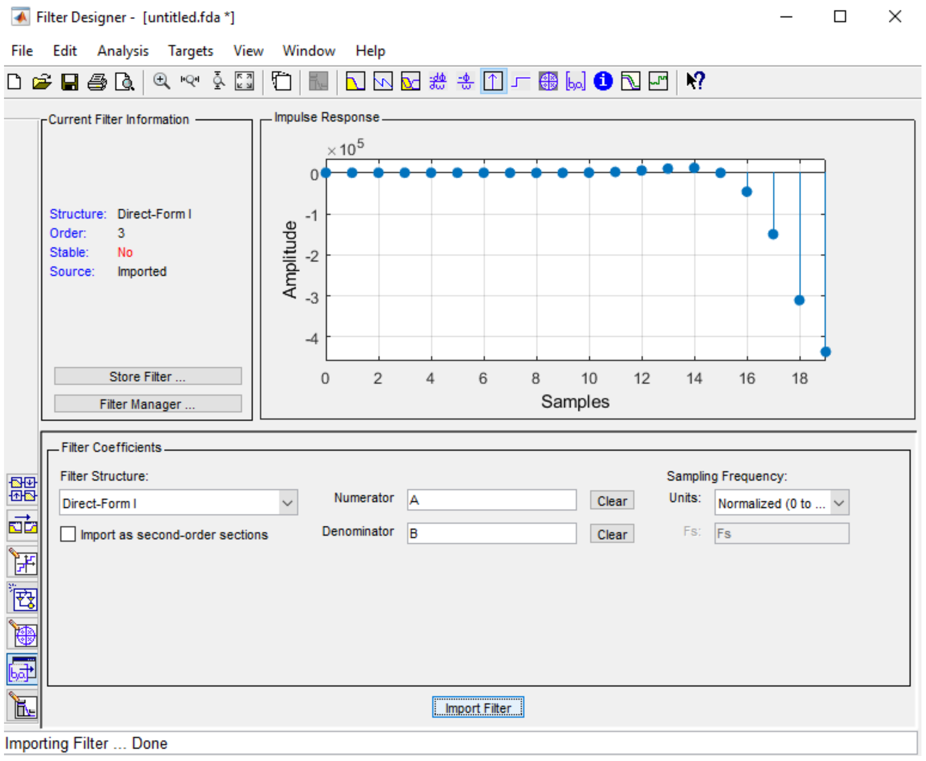


Figure 5: Sketch of system $y(n) = -0.2y(n-1) + 0.14y(n-2) - 0.44y(n-3) + x(n) - 3x(n-1) + 3x(n-2) + x(n-3)$

Part B

Use filterDesigner to analyze the structure in part(a)and document the impulse response.

Did the same thing as we did for problems 1 and 2. This isn't the greatest filter, but it's still a filter. Notice that this is unstable, since the impulse response does not converge.



Structure: Direct-Form I
Order: 3
Stable: No
Source: Imported

Filter Structure: Direct-Form I
Import as second-order sections
Numerator: A
Denominator: B
Sampling Frequency:
Units: Normalized (0 to ...
Fs: Fs

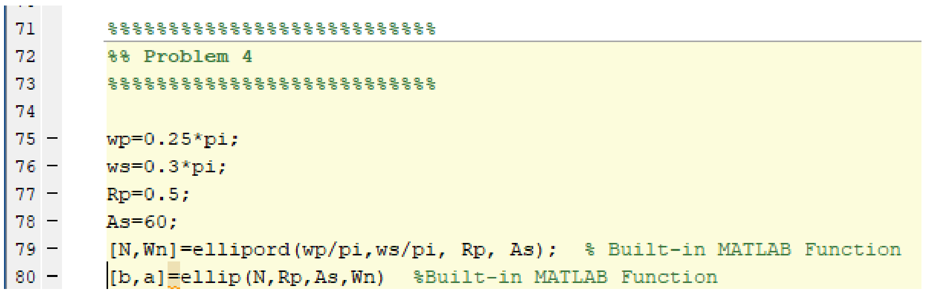
Importing Filter ... Done

Figure 6: Use of Filter Designer to perform an impulse response given our coefficients. Note that the filter structure is set to *Direct Form I*

Problem 4

Part A

An IIR lowpass filter designed to meet the specifications of 0.5 dB ripple in the passband, 60 dB ripple in the stopband, a passband edge frequency $\omega_p = 0.25\pi$*, and a stopband edge frequency* $\omega_s = 0.3\pi$ *is obtained using the following MATLAB script in Figure 7*



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Problem 4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

wp=0.25*pi;
ws=0.3*pi;
Rp=0.5;
As=60;
[N,Wn]=ellipord(wp/pi,ws/pi, Rp, As);  % Built-in MATLAB Function
[b,a]=ellip(N,Rp,As,Wn)  %Built-in MATLAB Function
```

Figure 7: Given MATLAB code.

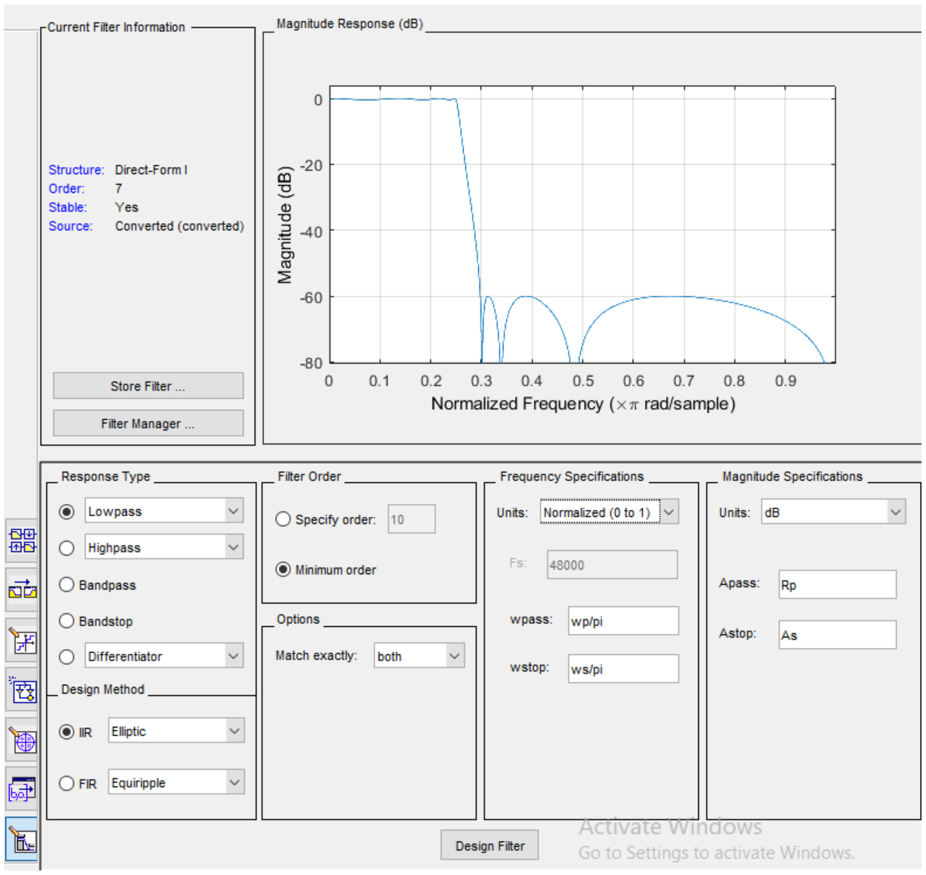
This should result in a single section structure with filter coefficients b and a, respectively, and MATLAB double precision accuracy which we consider to be an acceptable approximation to infinite precision.Document the coefficients plot the log magnitude using of the filter response using the freqz MATLAB command.

This section was implemented in part A of problem 4 below in the Matlab code. Comments are provided for additional details. Note that the freqz() command returns the frequency response given filter coefficients.

Part B

Use the filterDesigner inputs wp, ws, Rp, As to design a single section Direct Form elliptic IIR filter with a "minimum order" assuming double precision (infinite approximation). Observe and document the log-magnitude plot of the filter response. Compare the results with that of part A. What do you conclude regarding MATLAB and filterDesigner results?

This is a more visual way to produce filter coefficients, and it allows flexibility in determining what type of filter structure to use. I noticed that with SOS, the filter designer will return a matrix of non-complex valued coefficients. For single section, I was able to get the numerator and denominator coefficients and plot the freqz results to give a comparison in Figure 9. Notice that the filter has a tighter band transition between the passband and stopband frequencies, at the cost of more ripple in the stopband.



Current Filter Information

Structure:	Direct-Form I
Order:	7
Stable:	Yes
Source:	Converted (converted)

Response Type: [✓ Lowpass] Highpass Bandpass Bandstop Differentiator

Design Method: [✓ IIR] Elliptic; FIR Equiripple

Filter Order: Specify order: 10; [✓ Minimum order]

Options: Match exactly: both

Frequency Specifications: Units: Normalized (0 to 1); Fs: 48000; wpass: wp/pi; wstop: ws/pi

Magnitude Specifications: Units: dB; Apass: Rp; Astop: As

Figure 8: Filter Designer results for our given parameters.

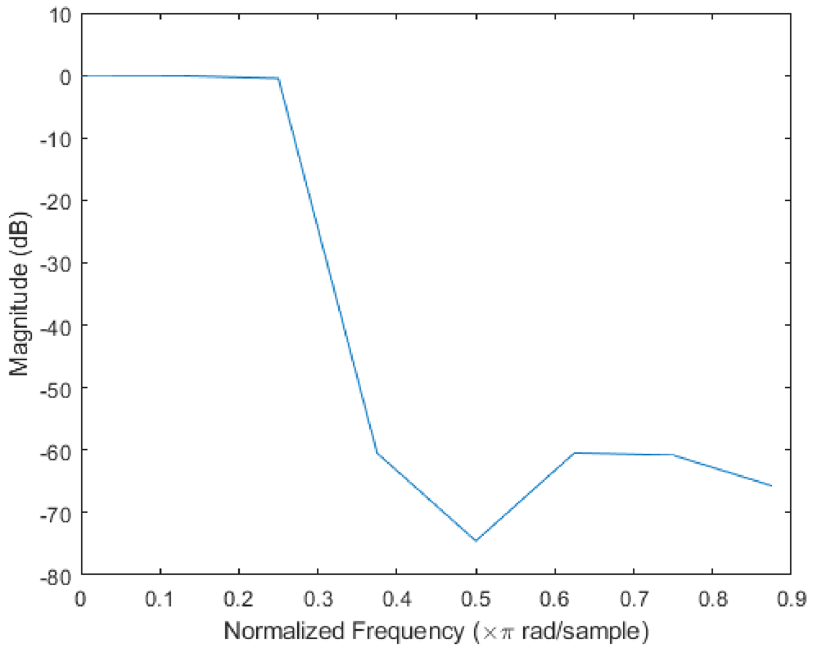
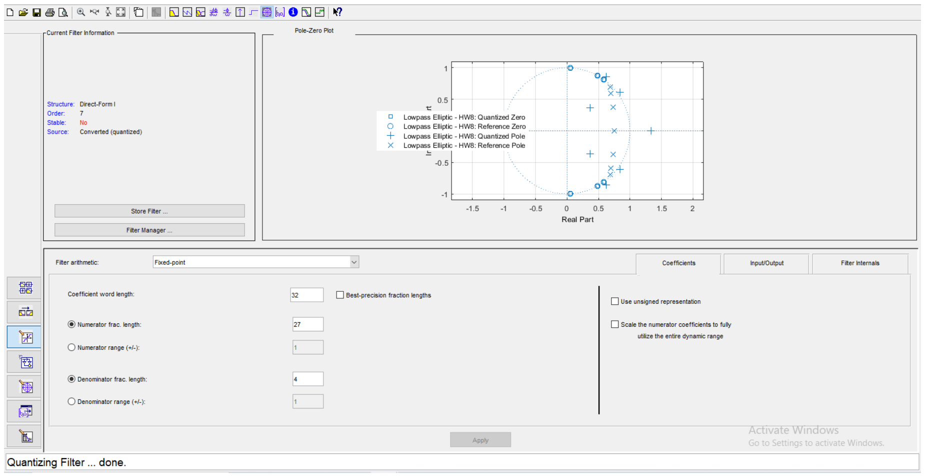


Figure 9: Plot of the freqz() output, the frequency response, of the output from Filter Designer.

Part C

With the single-section Direct Form filter from part B,quantize the coefficients to four decimal precision by determining the equivalent binary precision. Use this as input to the filterDesigner“word length”parameter. Remember that a “sign bit”must be included as part of the“word length”. Now generate the log-magnitude plot of the resulting filter and compare it with the infinite precision design.

There are a few ways to do this: first method is to use Filter Designer and navigate to the quantization tab. In figure 10, notice the parameters that are set cause the filter to become unstable. To make things worse, the quantization effects cause more attenuation and a worse cut-off frequency for out filter as shown in Figure 11



Current Filter Information

Structure: Direct-Form I
Order: 7
Stable: No
Source: Converted (quantized)

Filter arithmetic: Fixed-point

Coefficient word length: 32
☐ Best-precision fraction lengths
[✓ Numerator frac. length]: 27
Numerator range (+/-): 1
[✓ Denominator frac. length]: 4
Denominator range (+/-): 1
☐ Use unsigned representation
☐ Scale the numerator coefficients to fully utilize the entire dynamic range

Quantizing Filter ... done.

Pole-Zero Plot
□ Lowpass Elliptic - HW8: Quantized Zero
○ Lowpass Elliptic - HW8: Reference Zero
+ Lowpass Elliptic - HW8: Quantized Pole
× Lowpass Elliptic - HW8: Reference Pole
Real Part
Store Filter ...
Filter Manager ...
Coefficients
Input/Output
Filter Internals
Apply
Activate Windows
Go to Settings to activate Windows.

Figure 10: Setting Quantization in Filter Designer.

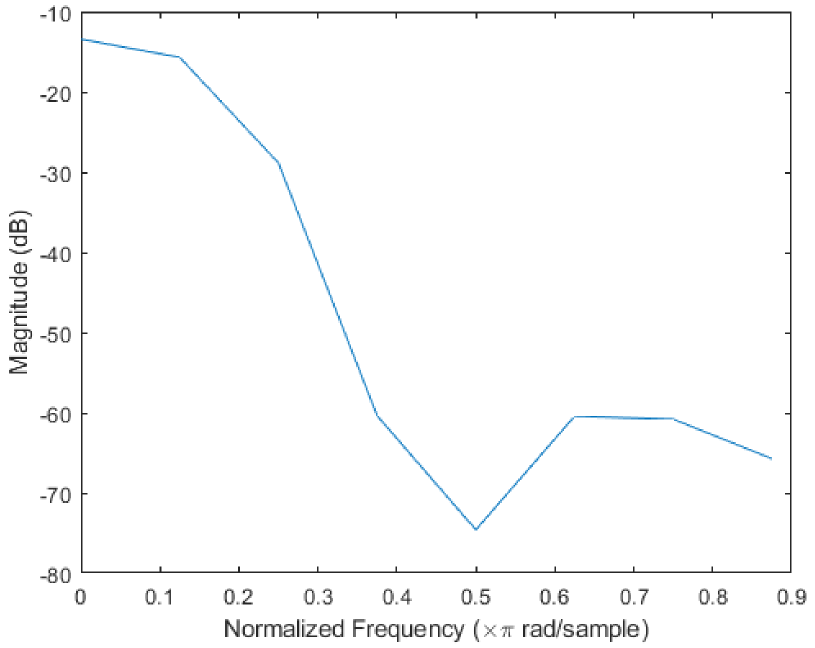
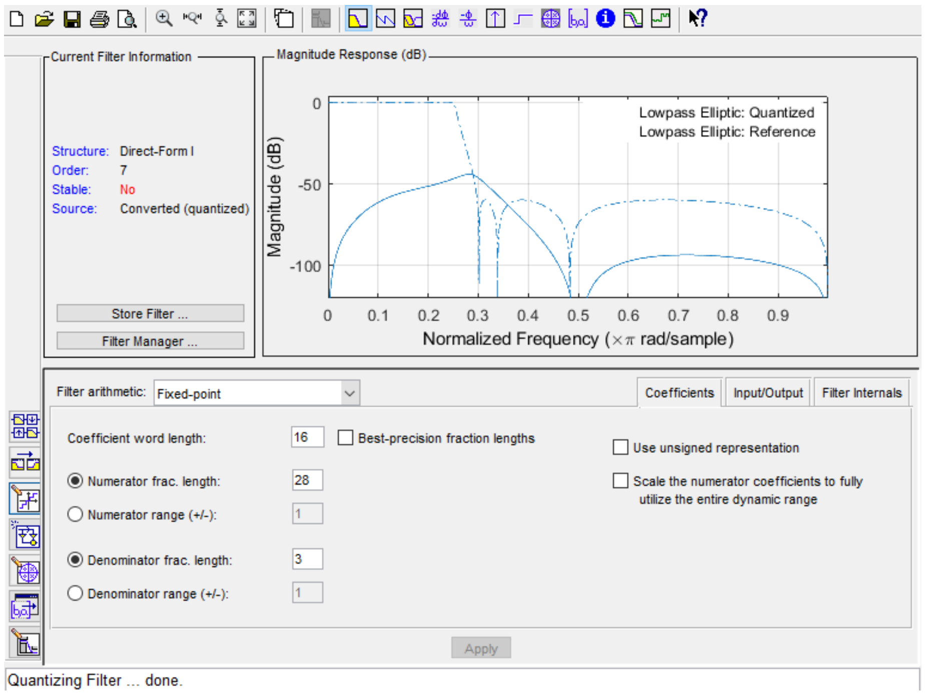


Figure 11: Quantization effects illustrated. Notice the attenuation and the transition band.

Part D

Repeat part C Quantizing the Direct Form coefficients to three decimals places(by rounding)by determining the equivalent binary precision. Compare the resulting filter's then generate the log-magnitude plot with the infinite precision filter's plot

Our filter just went from bad to worse with this one. In fact, it is a bit of a bandpass filter now with the nullspace at the zero, and the targeted frequency range for the lowpass filter is severely attenuated. The filter is also unstable.


Current Filter Information
Magnitude Response (dB)
Lowpass Elliptic: Quantized
Lowpass Elliptic: Reference
Magnitude (dB)
Normalized Frequency (×π rad/sample)
Store Filter ...
Filter Manager ...
Coefficients
Input/Output
Filter Internals
Apply

Structure: Direct-Form I
Order: 7
Stable: No
Source: Converted (quantized)

Filter arithmetic: Fixed-point

Coefficient word length: 16
[ ] Best-precision fraction lengths
[✓ Numerator frac. length:] 28
Numerator range (+/-): 1
[✓ Denominator frac. length:] 3
Denominator range (+/-): 1
[ ] Use unsigned representation
[ ] Scale the numerator coefficients to fully utilize the entire dynamic range

Quantizing Filter ... done.

Figure 12: Quantization effects illustrated via Filter Designer. Notice the attenuation and the transition band.

Problem 5

Use filterDesigner to do the following problem: Start with the infinite precision digital lowpass filter used in Problem 4.(Don't forget to change quantization setting to "double precision". Convert the structure into a cascaded SOS structure.

Part A

Using infinite precision and the SOS form realization, plot the log-magnitude of the cascaded SOS filter. How does it compare to infinite precision single-section direct form in Problem 4part B?

One of the nice things about Filter Designer and the Freqz command is that you can get a SOS object. This object has N amount of rows for each section of filter (cascaded filter). Freqz can also take the SOS object as a parameter and give you your log magnitude frequency response. See the matlab code. This multi-order filter is significantly more "smooth" compared to the filter in problem 4.

Part B

Quantize the cascaded SOS structure coefficients to four decimal susing the filterDesigner "quantization"feature with the appropriate word length in the filterDesigner. Now generate the log-magnitude of the filter response and compare it with the results of part A above.

In figure 13 we see that there is no significant difference. This is because the filter generated was actually a set of cascaded elliptical filters, which reduces the effects of quantization.

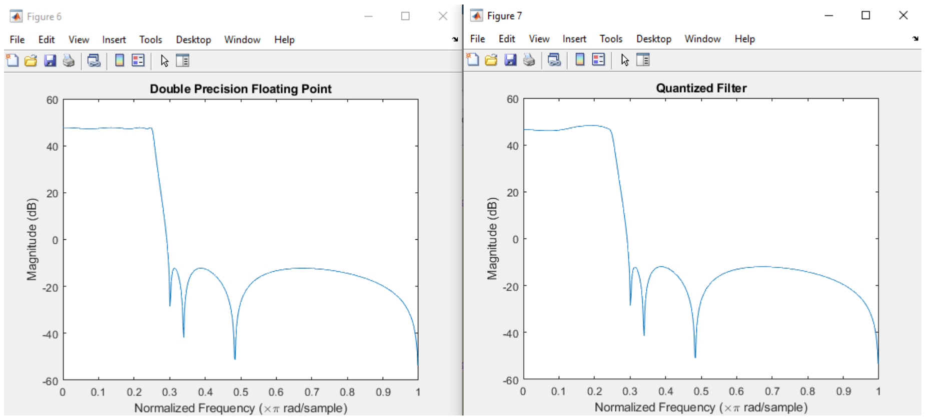


Figure 13: Quantization effects illustrated via Filter Designer. Notice how similar the filters are; this is the result of cascaded elliptical filters.

Part C

Similar to part B, quantize the cascaded SOS structure coefficients to three decimals and generate the log-magnitude plot for the resulting filter response. Compare results with part A.

This time we see more effects, but it is mainly with the passband ripple. It still isn't nearly as bad as our filters in problem 4.

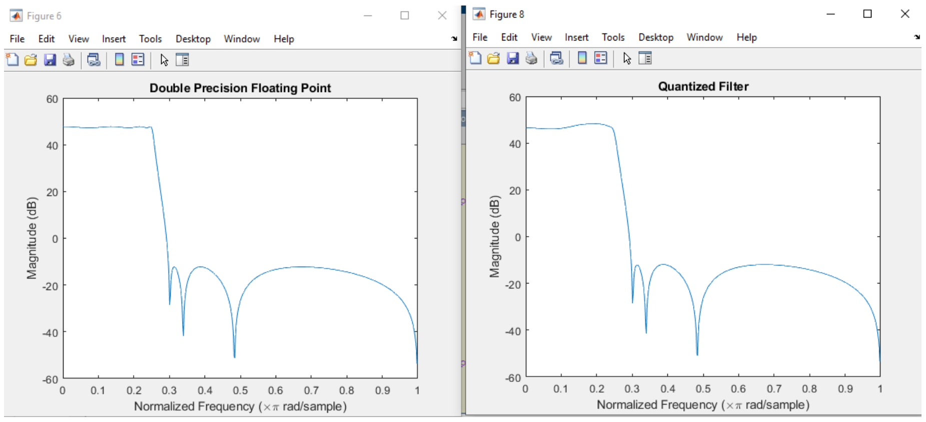


Figure 14: Quantization effects illustrated via Filter Designer. Notice how similar the filters are; this is the result of cascaded elliptical filters.

Part D

Comment on the plots in parts A, B, C and compare them with the similar plots in Problem 4. In this case, which structure is the most robust with respect to the coefficient quantitation?

Hands down, the cascaded second-order-sections elliptical filter in problem 5. Cascading the filters separates the poles from each other, and reduces the quantization effects of using fixed point arithmetic.

Matlab Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Problem 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%
% Part B
%%%%%%%%%%%
```

```
% Use filterDesigner to analyze the structure in part (a)
    and
% document the impulse response


[x, n] = impseq(0,-5,10);
% Remember: your B coefficients relate to your input x
   and your A
% coefficients relate to your output y. See eq 9.1.1 pg
   564

% y(n) = x(n)+2x(n-1)+3x(n-2)
% H(z) = (1+2z^-1+3z^-2)/(1)
B = [1,2,3]; % (1+2z^-1+3z^-2)
A = [1,0,0,0]; % (1) this works since there is the 1+sum(
   A coeff) in the H(z) denominator

y = filter(B,A,x);


%%%%%%%%%%%%%%%%%%%%%%%%%
%% Problem 2
%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%
% Part B
%%%%%%%%%%
% Use filterDesigner to analyze the structure in part (a)
    and
% document the impulse response

[x, n] = impseq(0,-5,10);
% Remember: your B coefficients relate to your input x
   and your A
% coefficients relate to your output y. See eq 9.1.1 pg
   564

% y(n)=  1.7y(n-1)-1.53y(n-2)+0.648y(n-3)+x(n)
% H(z) = \frac{1}{1-1.7z^{-1}+1.53z^{-2}-0.648z^{-3}}
B = [1];
A = [1,-1.7,1.53,-0.648]; % note the polarity of the
   coefficients! They are
                            % opposite of what is seen in
                                the impulse
                            % response, but same as H(z)

```

```
y = filter(B,A,x);

%%%%%%%%%%%%%%%%%%%%%%%%%
%% Problem 3
%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%
% Part B
%%%%%%%%%%%
% Use filterDesigner to analyze the structure in part (a)
    and
% document the impulse response

[x, n] = impseq(0,-5,10);
% Remember: your B coefficients relate to your input x
    and your A
% coefficients relate to your output y. See eq 9.1.1 pg
    564


% H(z) = \frac{1-3z^{-1}+3z^{-2}+1z^{-3}}{1+0.2z
    ^{-1}-0.14z^{-2}+0.44z^{-3}}
B = [1,-3,3,1];
A = [1,0.2,-0.14,0.44]; % note the polarity of the
    coefficients! They are
                            % opposite of what is seen in
                                the impulse
                            % response

y = filter(B,A,x);


%%%%%%%%%%%%%%%%%%%%%%%%%
%% Problem 4
%%%%%%%%%%%%%%%%%%%%%%%%%

wp=0.25*pi;
ws=0.3*pi;
Rp=0.5;
As=60;
% Eliptical filter order selection
[N,Wn]=ellipord(wp/pi,ws/pi, Rp, As);  % Built-in MATLAB
    Function
% elliptic or Cauer digital/analog filter design
[b,a]=ellip(N,Rp,As,Wn);  %Built-in MATLAB Function

```

```
% -- Part A
% the freqz command returns an N point complex frequency
    response vector H
% given filter coefficients a and b
[H,W] = freqz(b,a,N);

figure(4)
plot(W/pi,20*log10(abs(H)))
ax = gca;
% ax.YLim = [-100 20];
% ax.XTick = 0:.5:2;
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude (dB)')

% -- Part B
N=8
[H,W] = freqz(Num,Den,N);
figure(5)
plot(W/pi,20*log10(abs(H)))
ax = gca;
% ax.YLim = [-100 20];
% ax.XTick = 0:.5:2;
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude (dB)')


% quantizenumeric(pi,1,32,4,'fix')

%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Problem 5
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Note: filterDesigner was used to generate SOS objects.
    See HW8.
% -- Part A
[H,W] = freqz(SOS_a);
figure(6)
plot(W/pi,20*log10(abs(H)))
ax = gca;
% ax.YLim = [-100 20];
% ax.XTick = 0:.5:2;
title('Double Precision Floating Point')
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude (dB)')

% -- Part B (Quantized)
[H,W] = freqz(SOS_b);
```

```
figure(7)
plot(W/pi,20*log10(abs(H)))
ax = gca;
% ax.YLim = [-100 20];
% ax.XTick = 0:.5:2;
title('Quantized Filter')
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude (dB)')

% -- Part C (3 decimal)
[H,W] = freqz(SOS_b);
figure(8)
plot(W/pi,20*log10(abs(H)))
ax = gca;
% ax.YLim = [-100 20];
% ax.XTick = 0:.5:2;
title('More Quantized Filter')
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude (dB)')
```