University of Illinois at Urbana-Champaign

ECE 311: Digital Signal Processing Lab

LAB 4: SOLUTIONS

Chandra Radhakrishnan *Peter Kairouz*

Problem 1

(a)
```
>> n = (0:3)
>> x = ones(1,4);
>> y = filter([0.5 1 2], [1], x)
>> stem(n,y)
>>  title('Response y[n] = 0.5x[n]+x[n-1]+2x[n-2]'),xlabel('time index, n'),ylabel('y[n]')
```

The plot is shown in Fig. 1.

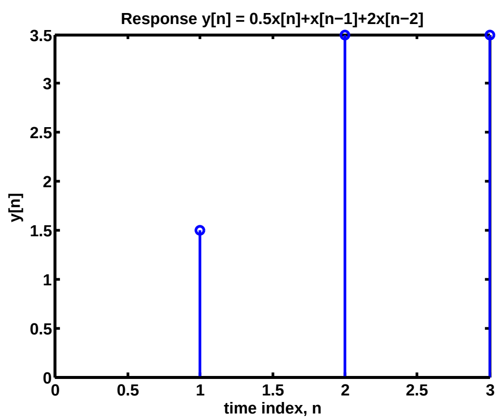

Figure 1: Response for system in Problem 1 (a)

(b)
```
>> y = filter([2], [1 -0.8], x)
>> stem(n,y)
>>  title('Response y[n] = 0.5x[n]+x[n-1]+2x[n-2]'),xlabel('time index, n'),ylabel('y[n]')
```

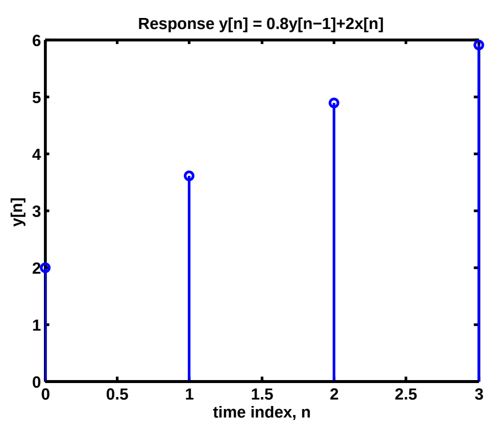

Figure 2: Response for system in Problem 1 (b)

The plot is shown in Fig. 2.

(c)
```
>> y = filter([0 2], [1 -0.8], x)
>> stem(n,y)
>>  title('Response y[n] = 0.8y[n-1]+2x[n-1]'),xlabel('time index, n'),ylabel('y[n]')
```
The plot is shown in Fig. 3.

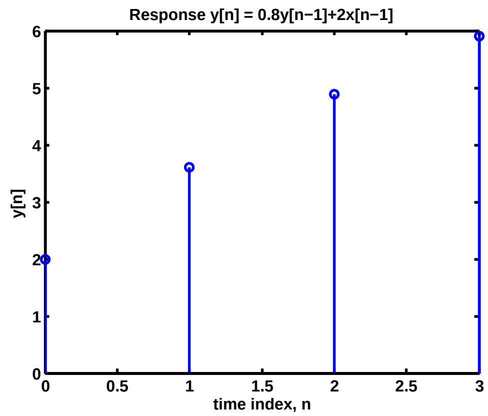

Figure 3: Response for system in Problem 1 (c)

Problem 2

(a) Refer to the function on the website.

(b) The plots of $y[n]$ for $x[n] = \delta[n]$ and $x[n] = u[n]$ are shown in Fig. 4.

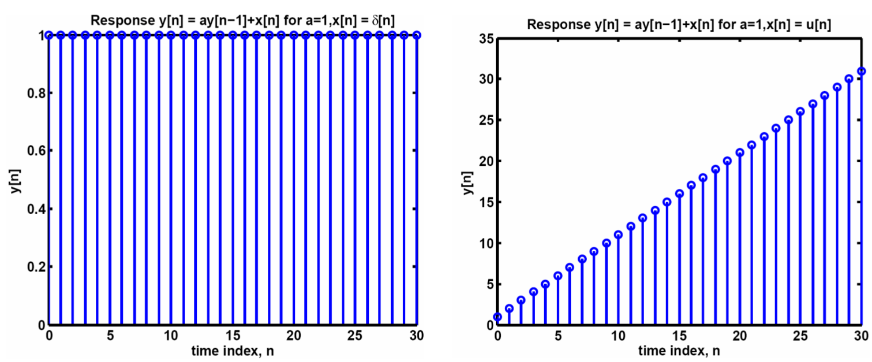

Figure 4: Response for system in Problem 2 (b)

(c) The plot for the difference $2y_1[n] - y_2[n]$ is shown in Fig. 5. The difference is not zero because the the initial conditions are not zero. The system is unstable and the effect of initial conditions persist.

(d) The causal system given by the difference equation is BIBO stable when $|a| < 1$. The output signals for $y[-1] = 0$ and $y[-1] = \frac{1}{2}$ are shown in Fig. 6. Since the system is BIBO stable the effects of initial conditions reduce as times index increases and filter responses look similar.

Problem 3

(a) The plot of poles and zeros is shown in Fig. 7

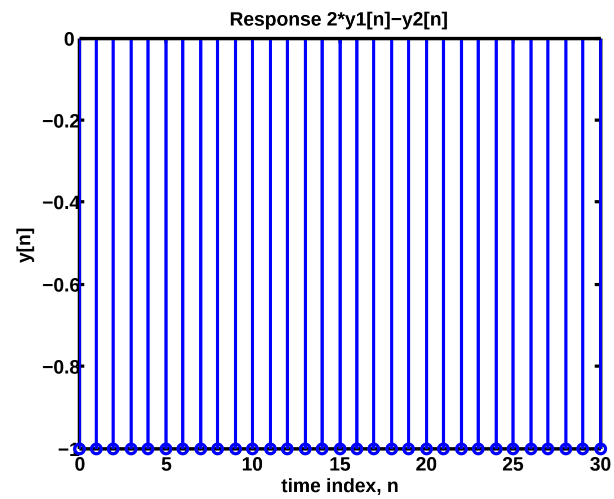

Figure 5: Response for system in Problem 2 (c)

```
>> dpzplot( [1 -1 0], [1 3 2] )
```

(b) $H(z)$ can be obtained as follows.

$$Y(z) + z^{-1}Y(z) + 0.5z^{-2}Y(z) = X(z)$$
$$\Rightarrow H(z) = \frac{z^2}{z^2 + z + 0.5}$$

The plot of poles and zeros is shown in Fig. 8

```
>> dpzplot( [1 0 0], [1 1 0.5] )
```

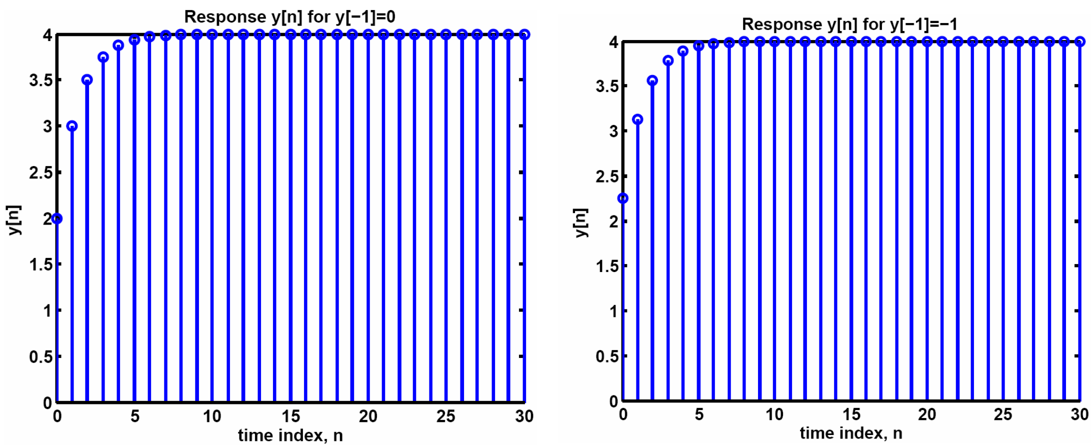

Figure 6: Response for system in Problem 2 (d)

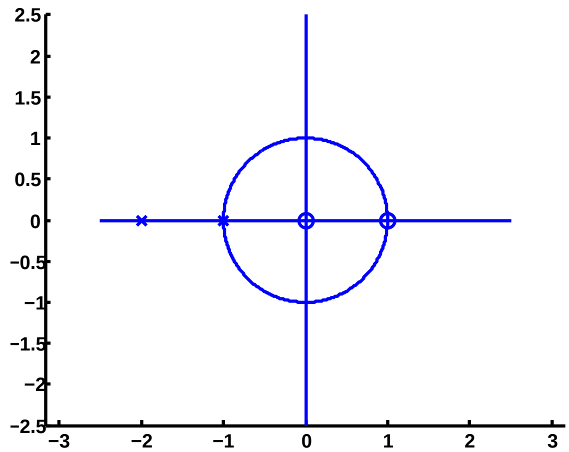

Figure 7: Response for system in Problem 3 (a)

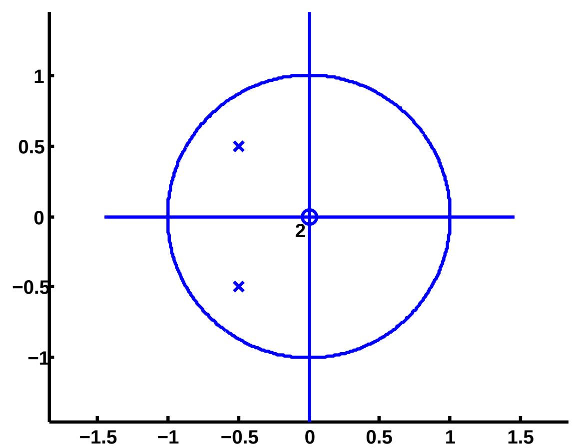

Figure 8: Response for system in Problem 3 (b)

(c) $H(z)$ can be obtained as follows.

$$Y(z)1.25z^{-1}Y(z) + 0.75z^{-2}Y(z)0.125z^{-3}Y(z) = X(z) + 0.5z^{-1}X(z)$$
$$\Rightarrow H(z) = \frac{z^3 + 0.5z^2)}{(z^3 1.25z^2 + 0.75z 0.125}$$

The plot of poles and zeros is shown in Fig. 9

```
>> dpzplot( [1 0 0], [1 1 0.5] )
```

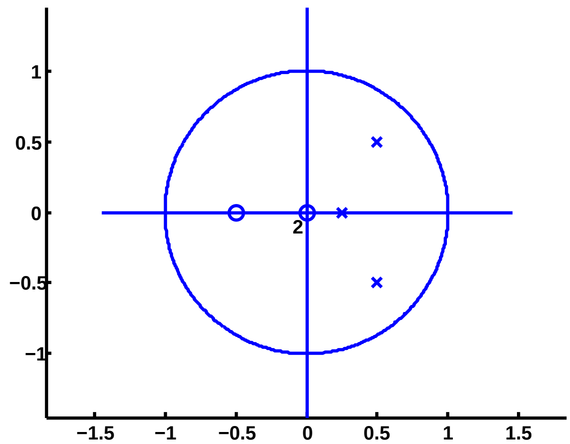

Figure 9: Response for system in Problem 3 (c)

Problem 4

The filter can be designed using `[b a] = ellip(4,0.2,40,[0.41 0.47]);`. The filter coefficients are

```
>> b =
    0.0104   -0.0142    0.0417   -0.0394    0.0627   -0.0394    0.0417   -0.0142    0.0104
>> a =
    1.0000   -1.4456    4.4656   -4.1966    6.5971   -3.9092    3.8755   -1.1678    0.7525
```

The magnitude and phase plot of the designed filter is shown in Fig. 10. The following MATLAB code is used for generating it.

```
>> [H,w]=freqz(b,a,4096);
>> plot(w/pi,20*log(abs(H)))
>> title('Response y[n] for y[-1]=-1'),xlabel('frequency \times \pi radians, n'),ylabel('|H_d(\omega)|')
```

(a) The first 200 samples of the impulse response are shown in Fig. 11. The MATLAB code is shown below,

```
>> x = [1 zeros(1,4095)];
>> h = filter(b,a,x);
>> n=(0:199);
>> plot(n,h(1:200))
>> title('Impulse Response of designed Elliptic filter'),xlabel('time index, n'),ylabel('h[n]')
```

(b) The 16-bit quantized coefficients and the the difference between the original and quantized coefficients can be generated using the following code.

```
>> M = max(abs([b a]));
>> a16 = quant(a,16,M);
>> errA = abs(a - a16);
```

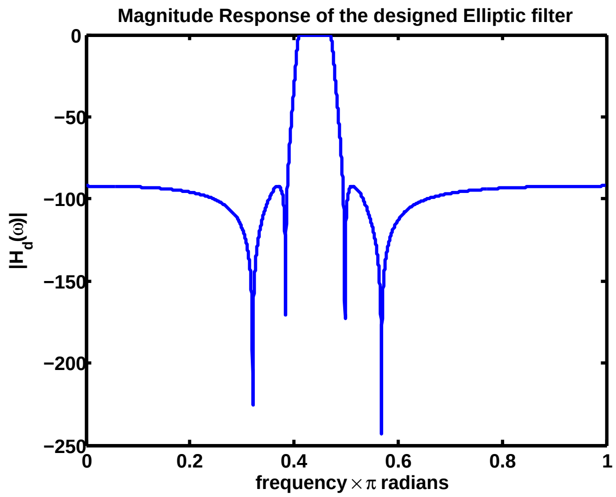

Figure 10: Magnitude Response of the Elliptic filter in Problem 4

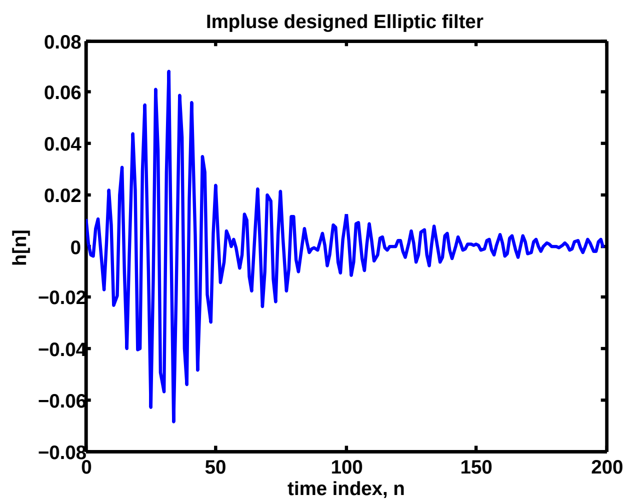

Figure 11: Impulse Response of the Elliptic filter in Problem 4(a)

The above code can be repeated for the coefficient `b`. The 16-bit quantized filter coefficients are

```
>> a16 =
    1.0000   -1.4457    4.4654   -4.1967    6.5969   -3.9094    3.8753   -1.1679    0.7524
>>  b16 =
    0.0103   -0.0143    0.0417   -0.0395    0.0626   -0.0395    0.0417   -0.0143    0.0103
```

The magnitude response of the quantized filter is shown in Fig. 12. The quantization affects the passband. The filter no longer has a quantized gain in the passband. The sidelobes also change but the change in the passband is a bigger concern.

(c) The pole-zero plot for the 16-bit quantized coefficients is shown in Fig. 13. Note that the poles are very close to the unit circle. Quantization errors can make the filter unstable.

(d) The 12-bit quantized coefficients and the the difference between the original and quantized coefficients can be generated using the following code.

```
>> M = max(abs([b a]));
>> a16 = quant(a,12,M);
>> errA = abs(a - a12);
```

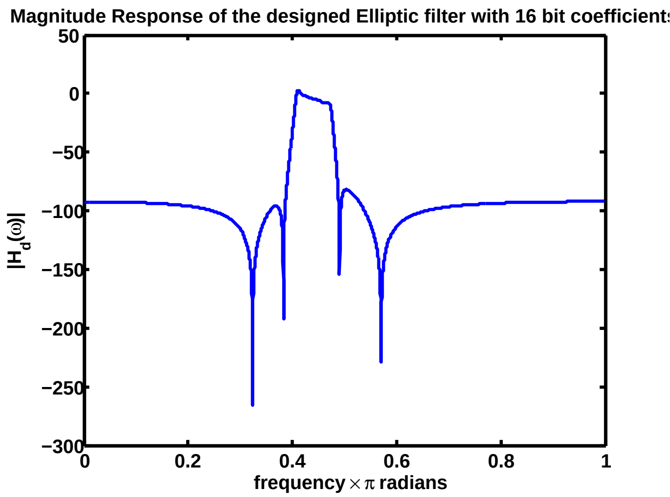

Figure 12: Magnitude Response of the Elliptic filter in Problem 4(b) - 16 bits quantized coefficients

The above code can be repeated for the coefficient `b`. The 12-bit quantized filter coefficients are

```
>> a12 =
    0.9986   -1.4463    4.4646   -4.1973    6.5939   -3.9106    3.8751   -1.1693    0.7505
>>  b12 =
     0.0097   -0.0161    0.0387   -0.0419    0.0612   -0.0419    0.0387   -0.0161    0.0097
```

The magnitude response of the quantized filter is shown in Fig. 14. The passband gets further affected due to the quantization.

(e) The impulse response for the 12-bit quantized coefficients using the filter command is computed as follows,

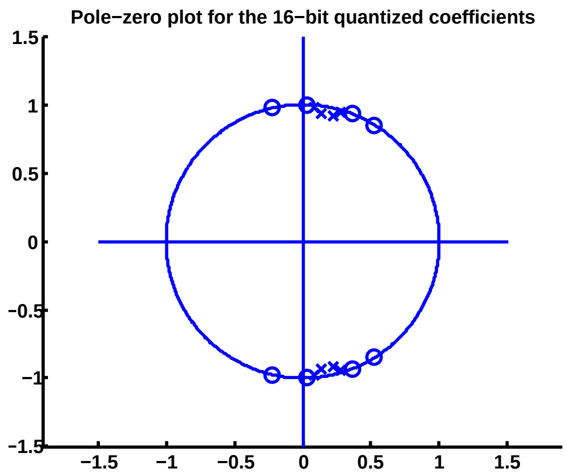

Figure 13: Pole - zero plot for Part (c)- 16 bits quantized coefficients

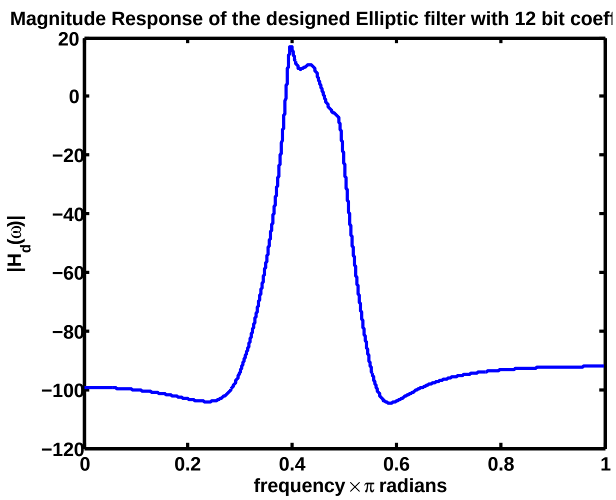

Figure 14: Magnitude Response of the Elliptic filter in Problem 4(d) - 12 bits quantized coefficients

```
>> h = filter(b12,a12,x);
>> n=(0:199);
>> plot(n,h(1:200))
>> title('Impulse response for 12-bit coefficients'),xlabel('time index, n'),ylabel('h[n]')
```

Figure 15 shows the impulse response. The filter is causal but not stable. The quantization causes 2 poles to move out of the unit circle (can be seen using dpzplot).

Problem 5

(a) The coefficients cascade form of the filter can be obtained by the following code,

```
>> [bc ac] = df2cf(b,a);
```

The coefficients of the cascade are shown below,

```
>> bc =
    0.0104   -0.0111    0.0104
```

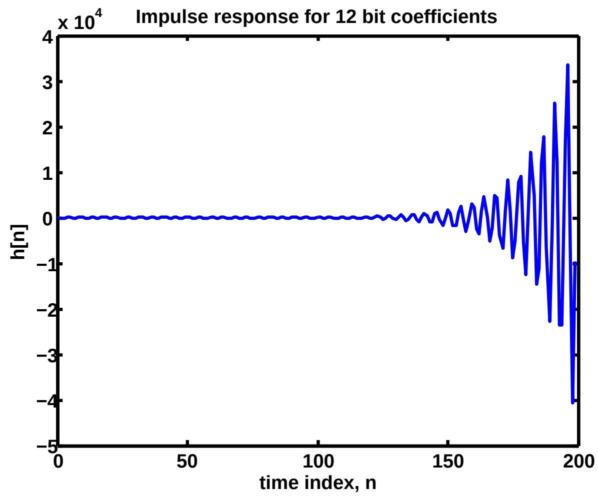

Figure 15: Impulse response for 12-bit coefficients 4(e)

```
    1.0000   -0.7146    1.0000
    1.0000    0.4245    1.0000
    1.0000   -0.0144    1.0000
>> ac =
    1.0000   -0.1720    0.9665
    1.0000   -0.5611    0.9678
    1.0000   -0.2649    0.8960
    1.0000   -0.4476    0.8979
```

(b) The filter output can be computed by making successive calls to the filter function where the input to the next filter is the output from the previous filtering operation. The impulse response is shown in Fig. 16. The code is shown below. The error between the two outputs `h` and `hc` is, 2.94e-013.

```
>> y1 = filter(bc(1,:),ac(1,:),x);
>> y2 = filter(bc(2,:),ac(2,:),y1);
>> y3 = filter(bc(3,:),ac(3,:),y2);
>> hc = filter(bc(4,:),ac(4,:),y3);
>> errC = abs(hc-h)
```

(c) The quant function can be used to quantize the coefficients. The quantized coefficients of the cascaded filter are,

```
>> [bc ac] = df2cf(b,a);
>> M=max(abs([bc(1,:) bc(2,:) bc(3,:) bc(4,:) ac(1,:) ac(2,:) ac(3,:) ac(4,:)]));
>> bc16=quant(bc,16,M);
>> ac16=quant(ac,16,M);

>> bc16 =
    0.0103    -0.0111    0.0103
    1.0000   -0.7146    1.0000
    1.0000    0.4245    1.0000
    1.0000   -0.0144    1.0000


ac16 =
1.0000   -0.1720    0.9665
```

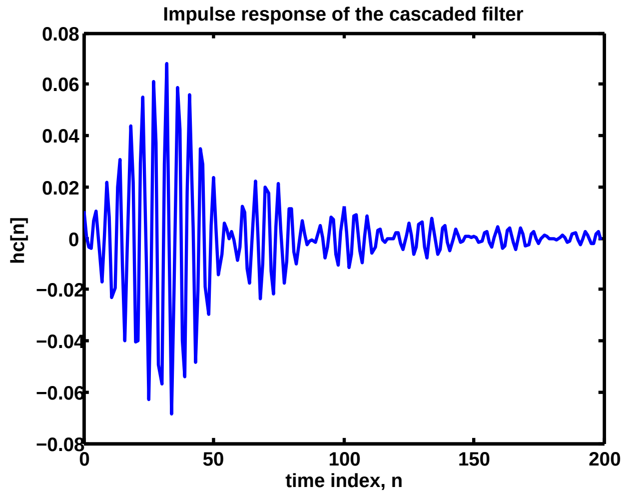

Figure 16: Impulse response for the cascaded structure of Problem 5(b)

```
    1.0000   -0.5612    0.9677
    1.0000   -0.2649    0.8960
    1.0000   -0.4476    0.8979
```

The impulse response can be generated as in part (a).

```
>> y1 = filter(bc16(1,:),ac16(1,:),x);
>> y2 = filter(bc16(2,:),ac16(2,:),y1);
>> y3 = filter(bc16(3,:),ac16(3,:),y2);
>> hc16 = filter(bc16(4,:),ac16(4,:),y3);
```

(d) The Magnitude response is shown in Fig. 17. As can be seen the passband is now flat. The pole-zero plots can be generated

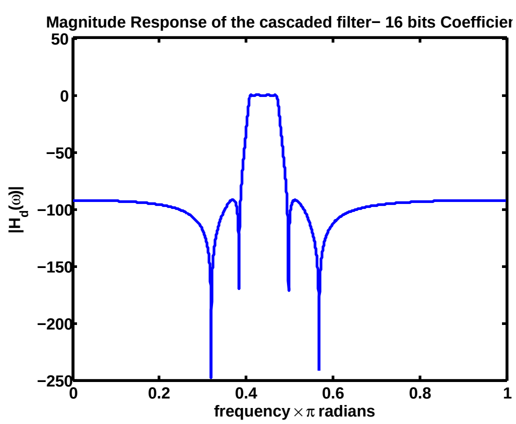

Figure 17: Magnitude response for the cascaded structure of Problem 5(d) - 16 bits coefficients

using the `dpzplot` function. The pole-zero plots are shown in Fig. 18. We can see that the cascaded sections are stable.

```
>> dpzplot(bc16(1,:),ac16(1,:));
>> title('1^{st} Subsection');
>> dpzplot(bc16(2,:),ac16(2,:));
>> title('2^{nd} Subsection');
>> dpzplot(bc16(3,:),ac16(3,:));
```

```
>> title('3^{rd} Subsection');
>> dpzplot(bc16(4,:),ac16(4,:));
>> title('4^{th} Subsection');
```

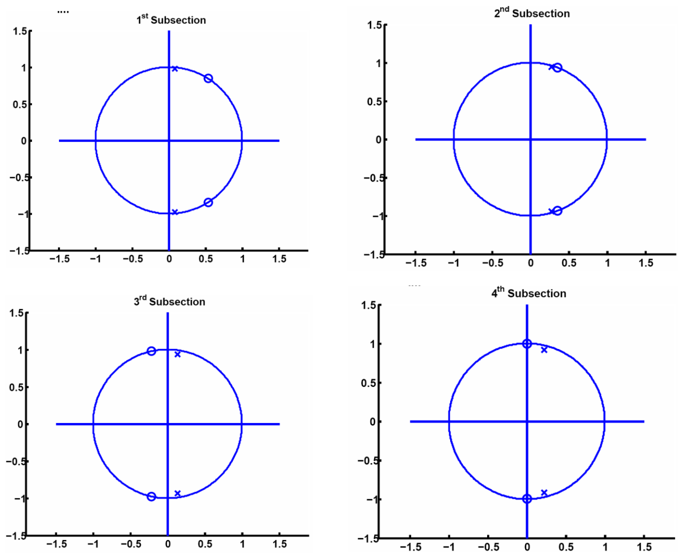

Figure 18: Pole-zero plot for the cascaded structure of Problem 5(d) - 16 bits coefficients

(e) The procedure in part (d) can be repeated for part (e). The magnitude response and the pole-zero plots are shown in Fig .19 and Fig. 20 respectively. The cascaded sections are stable and the magnitude response has a flat passband. Note a cascaded structure is more robust to quantization errors.

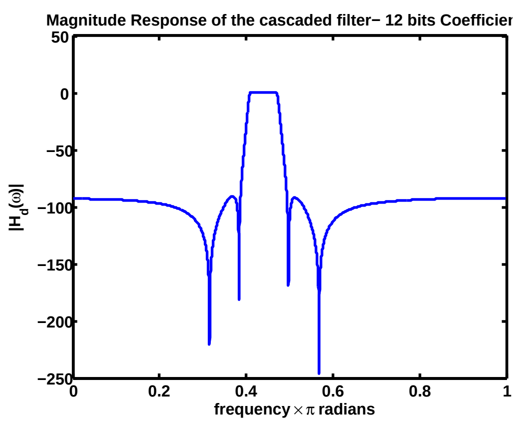

Figure 19: Pole-zero plot for the cascaded structure of Problem 5(e) - 12 bits coefficients

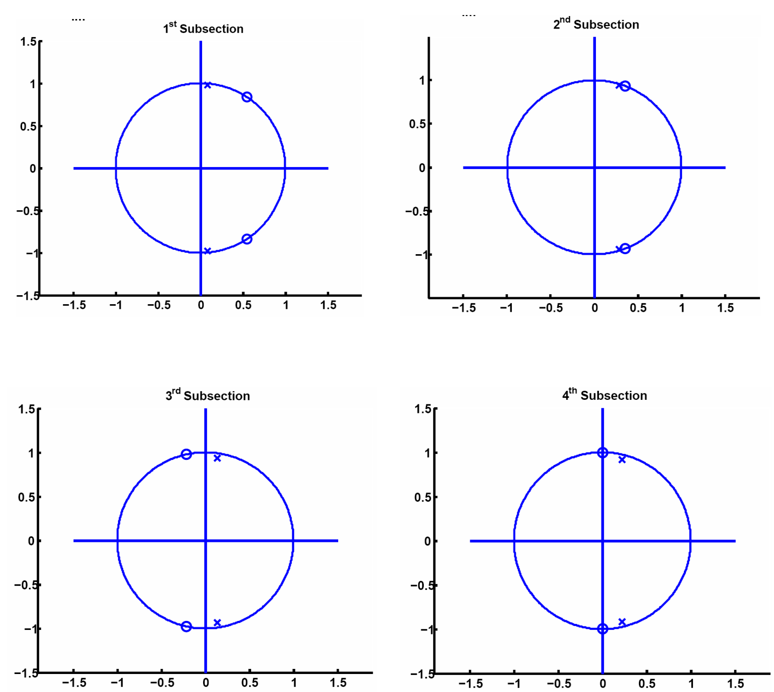

Figure 20: Pole-zero plot for the cascaded structure of Problem 5(e) - 12 bits coefficients