Digital Signal Processing HW11 MATLAB Part

Name: Jingyang Zhang
Net ID: jz2807

1. **Speech filtering exercise, same as in HW 7, but this time design IIR filters: Butterworth, Chebyshev-1, Chebyshev-2, and Elliptic. Compare to your solutions for HW 7- HW 10. Try different levels of stop-band attenuation. Plot the frequency response, pole-zero diagram, and impulse response for each filter. (Use subplot to put the 3 figures for each filter on the same page.) What stop-band attenuation and corresponding filter order are needed to satisfactorily suppress the high frequency noise?**

Solution:

.m file(s): jyz_HW11.m
make_plot.m

Code(make_plot.m):

```
function[] = make_plot(om, H, A, B, i)
As = [20,40,60,20,40,60,20,40,60,20,40,60];
subplot(3,1,1)
plot(om/pi, abs(H))
if 1<=i && i <=3
   title(['Frequency Response of The Butter Filter, As =
   ',num2str(As(i)),'dB']);
elseif 4<=i && i <=6
   title(['Frequency Response of The Chebyshev I Filter, As =
   ',num2str(As(i)),'dB']);
elseif 7<=i && i <=9
   title(['Frequency Response of The Chebyshev II Filter, As =
   ',num2str(As(i)),'dB']);
elseif 10<=i && i <=12
   title(['Frequency Response of The Elliptic Filter, As =
   ',num2str(As(i)),'dB']);
end
grid on;ylim([-0.1 1.1])
subplot(3,1,2)
zplane(B, A)
title('Pole-Zero Diagram');
subplot(3,1,3)
stem(0:19, abs(filter(B, A,[1,zeros(1,19)])), '.')
title('Impulse Response')
xlabel('n');ylabel('h(n)');xlim([-0.5, 19.5]);
```

Code(jyz_HW11.m):

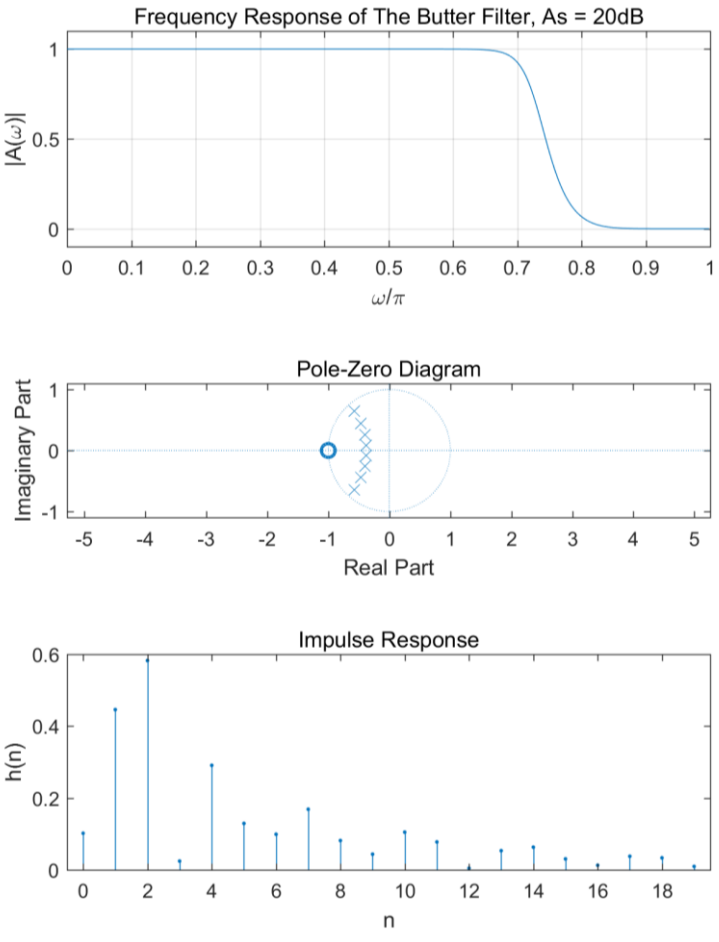
```
close all
clear

%Load signals and basic processing ↓
load NoisySpeech.txt
load mtlb
Noi = NoisySpeech;
Noif = fft(Noi, 4096);
mtlbf = fft(mtlb, 4096);
wp = (2500/(Fs/2))*pi; fp = wp/pi;%pass_band
ws = (2900/(Fs/2))*pi; fs = ws/pi;%stop_band
wc = (wp + ws) / 2; fc = wc/pi;   %cutoff_band
Rp = 0.5;  %no more than 0.5 dB of ripple in a passband
As1 = 20;%at least 40 dB of attenuation in the stopband
As2 = 40;%at least 50 dB of attenuation in the stopband
As3 = 60;%at least 60 dB of attenuation in the stopband

%Determine the length of the filters↓
Nbutt1 = buttord(fp, fs, Rp, As1);
Nbutt2 = buttord(fp, fs, Rp, As2);
Nbutt3 = buttord(fp, fs, Rp, As3);
Ncheb11 = cheb1ord(fp, fs, Rp, As1);
Ncheb12 = cheb1ord(fp, fs, Rp, As2);
Ncheb13 = cheb1ord(fp, fs, Rp, As3);
Ncheb21 = cheb2ord(fp, fs, Rp, As1);
Ncheb22 = cheb2ord(fp, fs, Rp, As2);
Ncheb23 = cheb2ord(fp, fs, Rp, As3);
Nellip1 = ellipord(fp, fs, Rp, As1);
Nellip2 = ellipord(fp, fs, Rp, As2);
Nellip3 = ellipord(fp, fs, Rp, As3);
%Produce the filters↓
[Bbutt1, Abutt1] = butter(Nbutt1, fc);
[Bbutt2, Abutt2] = butter(Nbutt2, fc);
[Bbutt3, Abutt3] = butter(Nbutt3, fc);
[Bcheb11, Acheb11] = cheby1(Ncheb11, Rp, fc);
[Bcheb12, Acheb12] = cheby1(Ncheb12, Rp, fc);
[Bcheb13, Acheb13] = cheby1(Ncheb13, Rp, fc);
[Bcheb21, Acheb21] = cheby2(Ncheb21, As1, fc);
[Bcheb22, Acheb22] = cheby2(Ncheb22, As2, fc);
[Bcheb23, Acheb23] = cheby2(Ncheb23, As3, fc);
[Bellip1, Aellip1] = ellip(Nellip1, Rp, As1, fc);
[Bellip2, Aellip2] = ellip(Nellip2, Rp, As2, fc);
[Bellip3, Aellip3] = ellip(Nellip3, Rp, As3, fc);
%Frequency Response↓
[Hbutt1, ombutt1] = freqz(Bbutt1, Abutt1);
[Hbutt2, ombutt2] = freqz(Bbutt2, Abutt2);
[Hbutt3, ombutt3] = freqz(Bbutt3, Abutt3);
[Hcheb11, omcheb11] = freqz(Bcheb11, Acheb11);
```

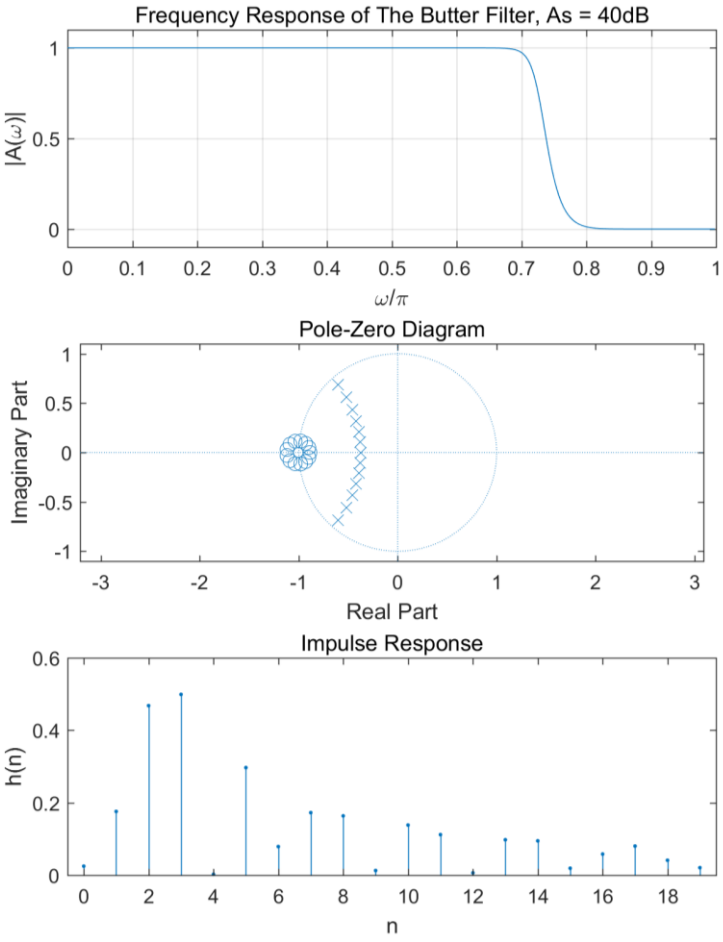
```
[Hcheb12, omcheb12] = freqz(Bcheb12, Acheb12);
[Hcheb13, omcheb13] = freqz(Bcheb13, Acheb13);
[Hcheb21, omcheb21] = freqz(Bcheb21, Acheb21);
[Hcheb22, omcheb22] = freqz(Bcheb22, Acheb22);
[Hcheb23, omcheb23] = freqz(Bcheb23, Acheb23);
[Hellip1, omellip1] = freqz(Bellip1, Aellip1);
[Hellip2, omellip2] = freqz(Bellip2, Aellip2);
[Hellip3, omellip3] = freqz(Bellip3, Aellip3);
%Filtering↓
ybutt1 = filter(Bbutt1, Abutt1, Noi);
ybutt2 = filter(Bbutt2, Abutt2, Noi);
ybutt3 = filter(Bbutt3, Abutt3, Noi);
ycheb11 = filter(Bcheb11, Acheb11, Noi);
ycheb12 = filter(Bcheb12, Acheb12, Noi);
ycheb13 = filter(Bcheb13, Acheb13, Noi);
ycheb21 = filter(Bcheb21, Acheb21, Noi);
ycheb22 = filter(Bcheb22, Acheb22, Noi);
ycheb23 = filter(Bcheb23, Acheb23, Noi);
yellip1 = filter(Bellip1, Aellip1, Noi);
yellip2 = filter(Bellip2, Aellip2, Noi);
yellip3 = filter(Bellip3, Aellip3, Noi);

make_plot(ombutt1, Hbutt1, Abutt1, Bbutt1, 1);
disp('Check The Title To See The Property of Filter')
disp('Press Enter To See Next Plot, And Listen To Signal
Filtered by It')
disp(' ')
soundsc(ybutt1, Fs)
pause
clf
make_plot(ombutt2, Hbutt2, Abutt2, Bbutt2, 2)
disp('Check The Title To See The Property of Filter')
disp('Press Enter To See Next Plot, And Listen To Signal
Filtered by It')
disp(' ')
soundsc(ybutt2, Fs)
pause
clf
make_plot(ombutt3, Hbutt3, Abutt3, Bbutt3, 3)
disp('Check The Title To See The Property of Filter')
disp('Press Enter To See Next Plot, And Listen To Signal
Filtered by It')
disp(' ')
soundsc(ybutt3, Fs)
pause
clf
make_plot(omcheb11, Hcheb11, Acheb11, Bcheb11, 4)
disp('Check The Title To See The Property of Filter')
disp('Press Enter To See Next Plot, And Listen To Signal
```

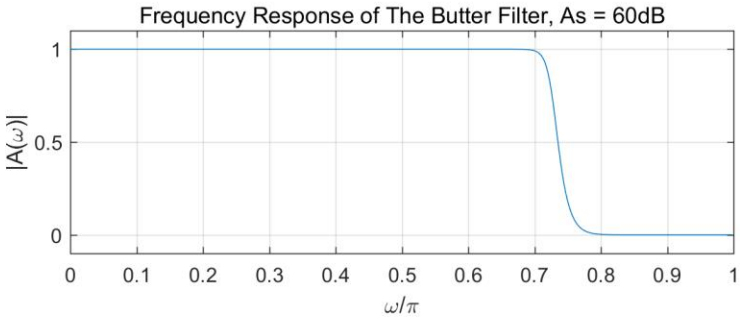
```
Filtered by It')
disp(' ')
soundsc(ycheb11, Fs)
pause
clf
make_plot(omcheb12, Hcheb12, Acheb12, Bcheb12, 5)
disp('Check The Title To See The Property of Filter')
disp('Press Enter To See Next Plot, And Listen To Signal
Filtered by It')
disp(' ')
soundsc(ycheb12, Fs)
pause
clf
make_plot(omcheb13, Hcheb13, Acheb13, Bcheb13, 6)
disp('Check The Title To See The Property of Filter')
disp('Press Enter To See Next Plot, And Listen To Signal
Filtered by It')
disp(' ')
soundsc(ycheb13, Fs)
pause
clf
make_plot(omcheb21, Hcheb21, Acheb21, Bcheb11, 7)
disp('Check The Title To See The Property of Filter')
disp('Press Enter To See Next Plot, And Listen To Signal
Filtered by It')
disp(' ')
soundsc(ycheb21, Fs)
pause
clf
make_plot(omcheb22, Hcheb22, Acheb22, Bcheb12, 8)
disp('Check The Title To See The Property of Filter')
disp('Press Enter To See Next Plot, And Listen To Signal
Filtered by It')
disp(' ')
soundsc(ycheb22, Fs)
pause
clf
make_plot(omcheb23, Hcheb23, Acheb23, Bcheb23, 9)
disp('Check The Title To See The Property of Filter')
disp('Press Enter To See Next Plot, And Listen To Signal
Filtered by It')
disp(' ')
soundsc(ycheb23, Fs)
pause
clf
make_plot(omellip1, Hellip1, Aellip1, Bellip1, 10)
disp('Check The Title To See The Property of Filter')
disp('Press Enter To See Next Plot, And Listen To Signal
Filtered by It')
```

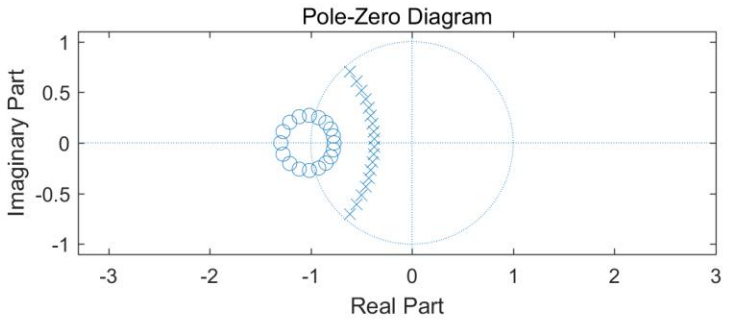
```
disp(' ')
soundsc(yellip1, Fs)
pause
clf
make_plot(omellip2, Hellip2, Aellip2, Bellip2, 11)
disp('Check The Title To See The Property of Filter')
disp('Press Enter To See Next Plot, And Listen To Signal
Filtered by It')
disp(' ')
soundsc(yellip2, Fs)
pause
clf
make_plot(omellip3, Hellip3, Aellip3, Bellip3, 12)
soundsc(yellip3, Fs)
disp('Check The Title To See The Property of Filter')
pause
disp('The Program Is Over!')
```

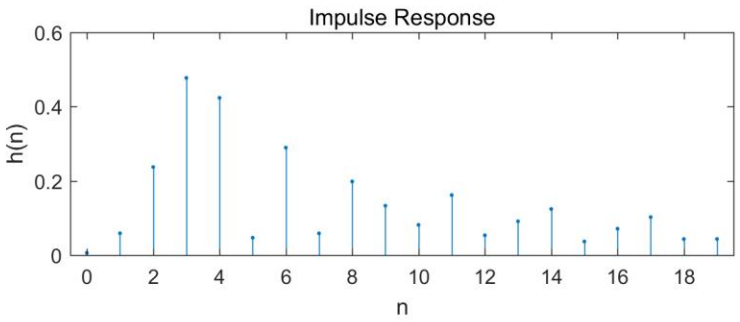
Result(plots):

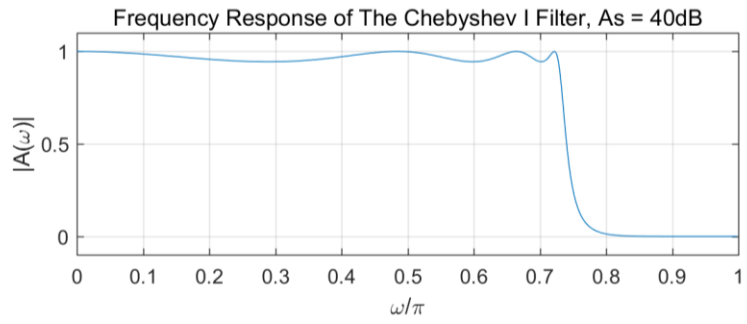


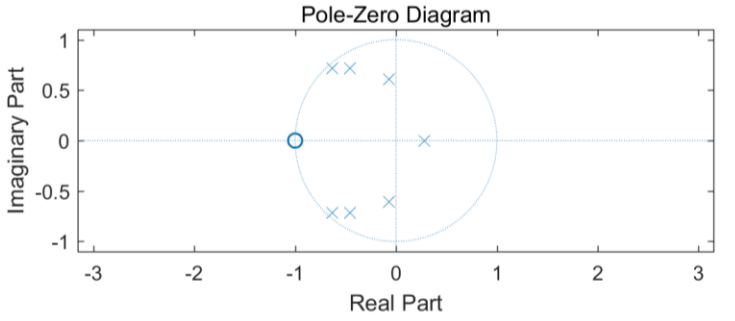


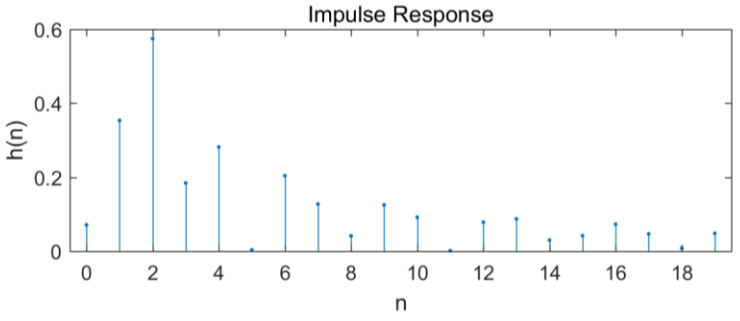

Frequency Response of The Butter Filter, As = 60dB
|A(ω)|
ω/π

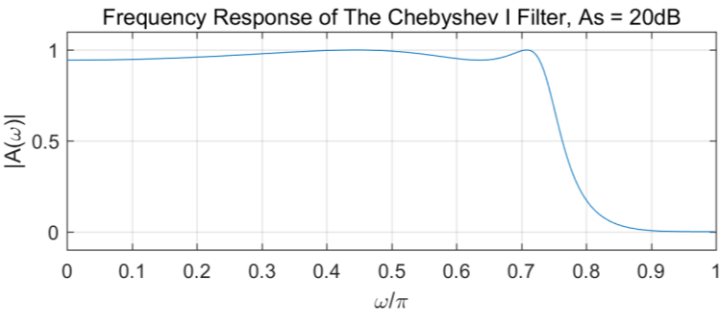

Pole-Zero Diagram
Imaginary Part
Real Part

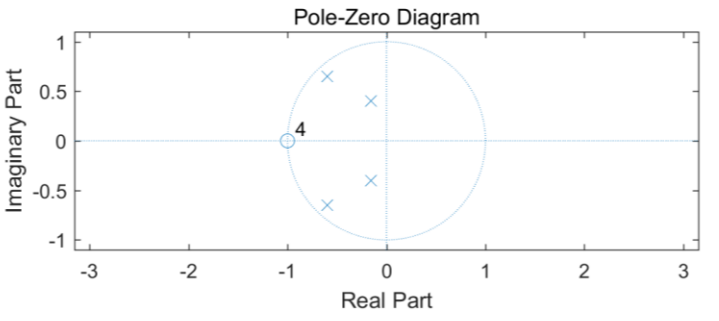

Impulse Response
h(n)
n

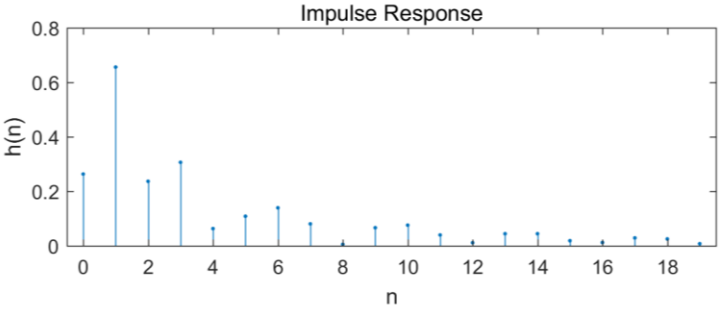

Frequency Response of The Chebyshev I Filter, As = 40dB
|A(ω)|
ω/π

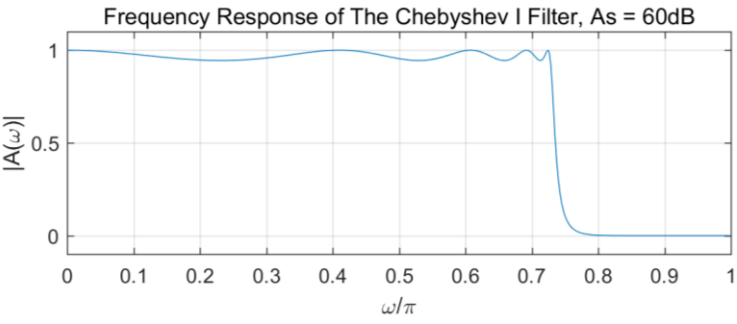

Pole-Zero Diagram
Imaginary Part
Real Part

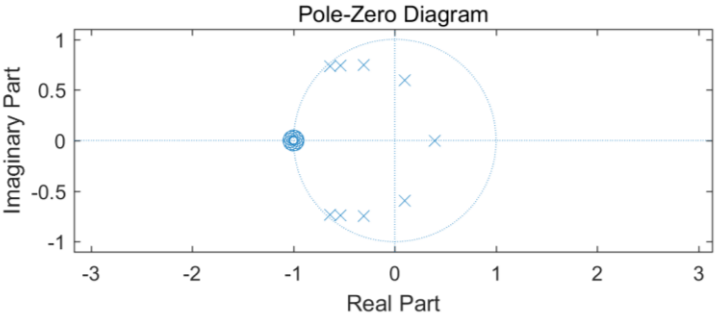

Impulse Response
h(n)
n

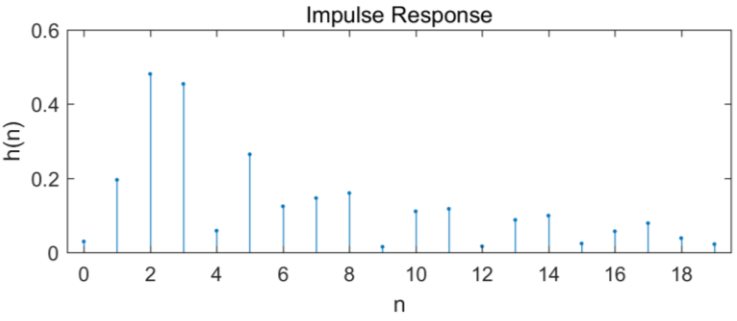

Frequency Response of The Chebyshev I Filter, As = 20dB
|A(ω)|
ω/π

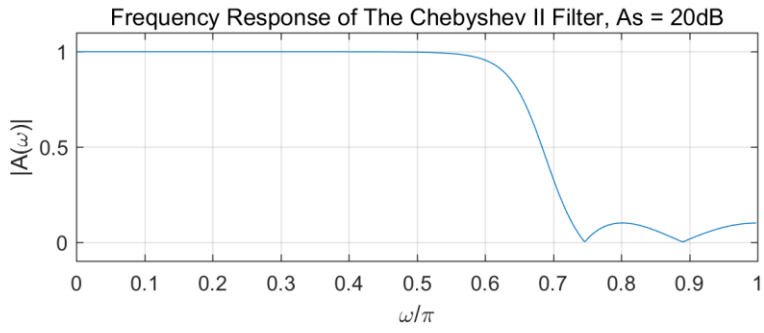

Pole-Zero Diagram
Imaginary Part
Real Part

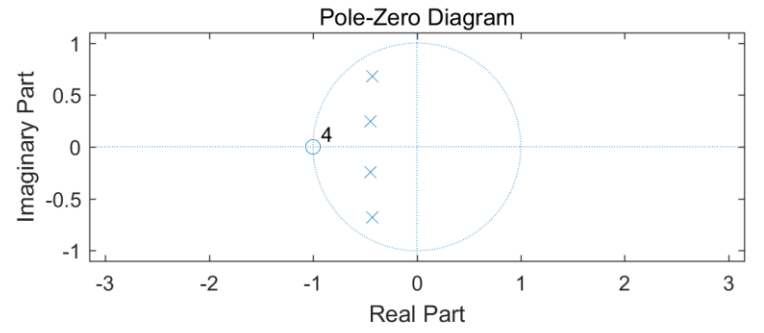

Impulse Response
h(n)
n

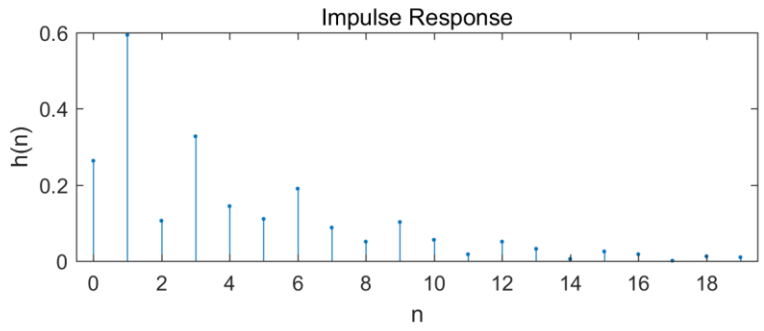

Frequency Response of The Chebyshev I Filter, As = 60dB
|A(ω)|
ω/π

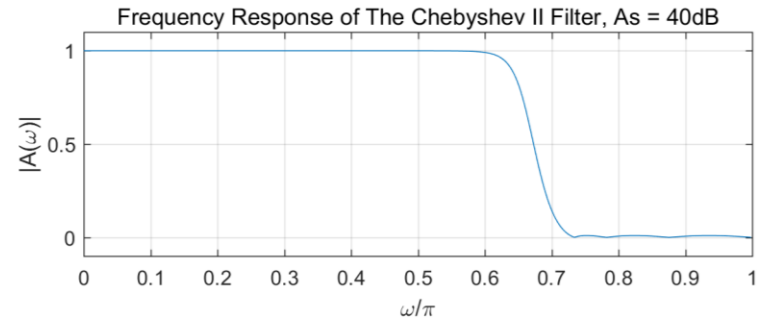

Pole-Zero Diagram
Imaginary Part
Real Part

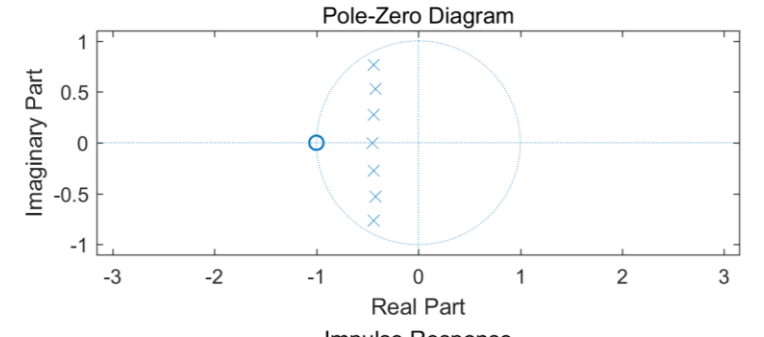

Impulse Response
h(n)
n

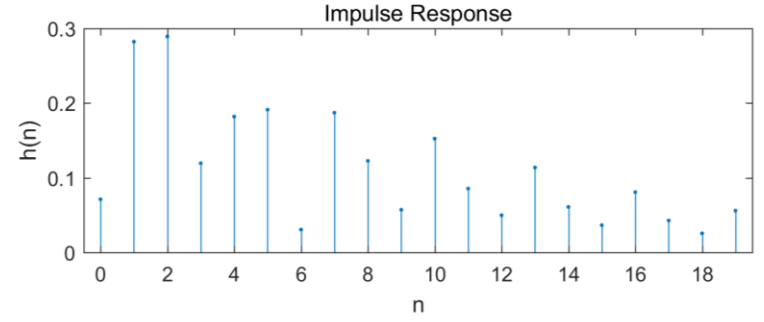

Frequency Response of The Chebyshev II Filter, As = 20dB
|A(ω)|
ω/π

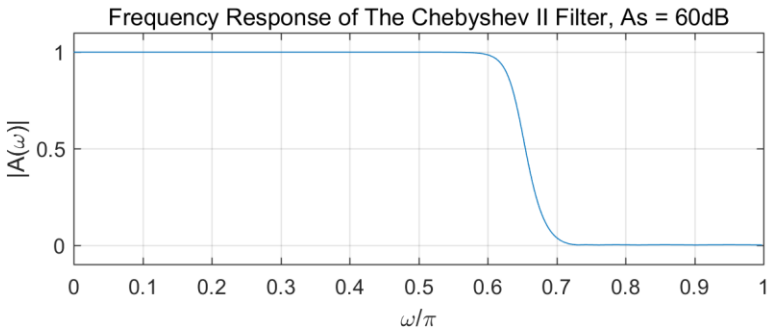

Pole-Zero Diagram
Imaginary Part
Real Part
4

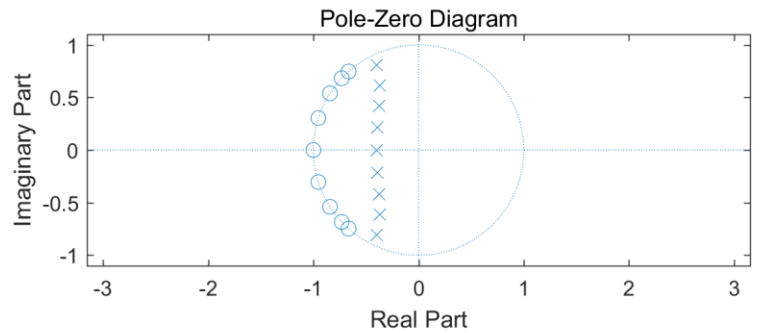

Impulse Response
h(n)
n

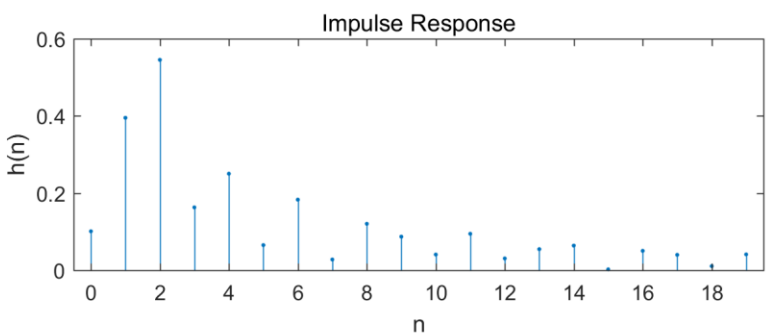

Frequency Response of The Chebyshev II Filter, As = 40dB
|A(ω)|
ω/π

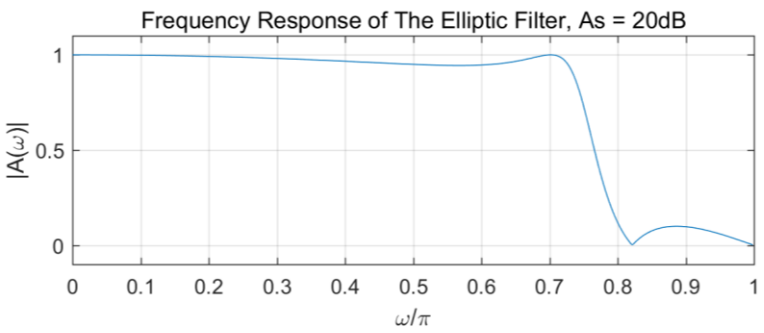

Pole-Zero Diagram
Imaginary Part
Real Part

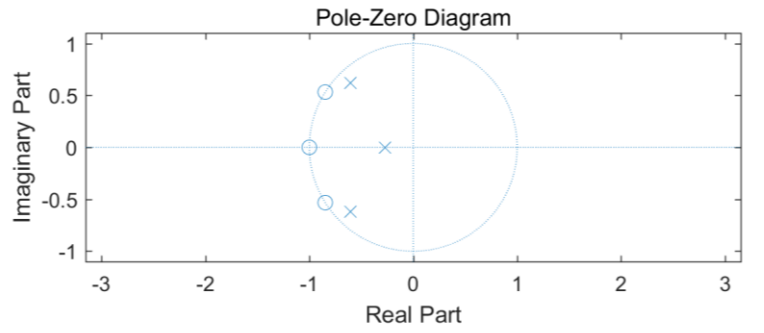

Impulse Response
h(n)
n

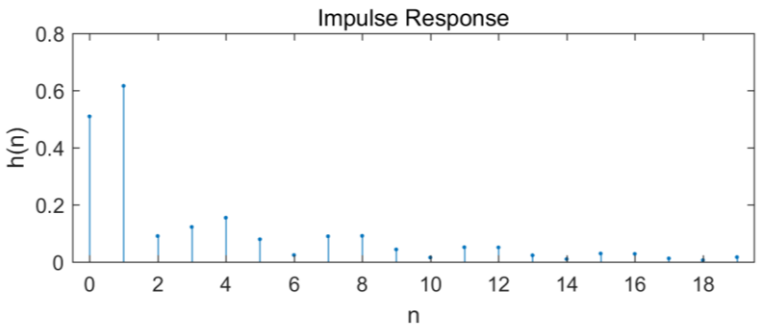

Frequency Response of The Chebyshev II Filter, As = 60dB
|A(ω)|
ω/π

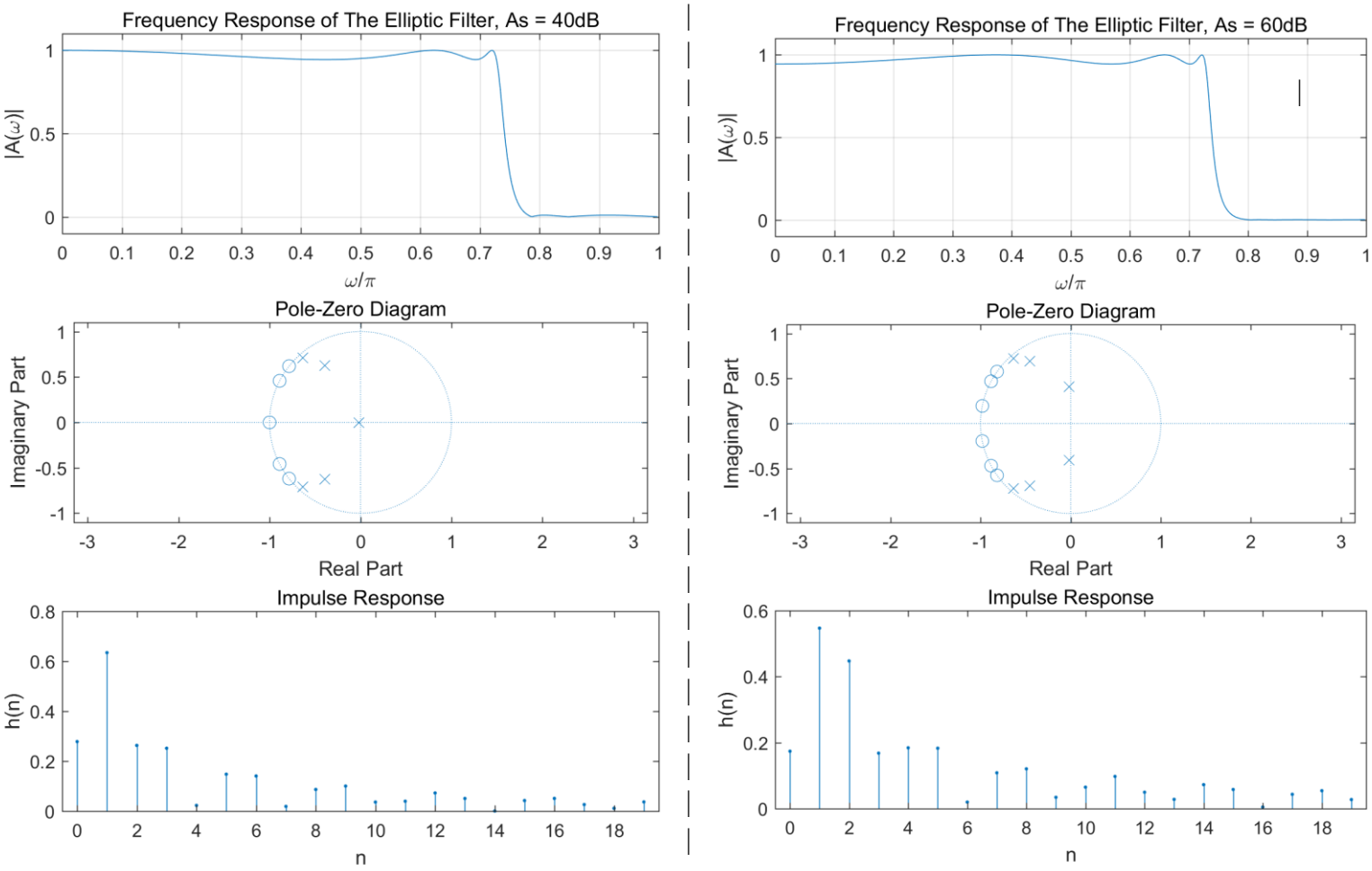

Pole-Zero Diagram
Imaginary Part
Real Part


Impulse Response
h(n)
n


Frequency Response of The Elliptic Filter, As = 20dB
|A(ω)|
ω/π


Pole-Zero Diagram
Imaginary Part
Real Part


Impulse Response
h(n)
n



Comments: In HW7, we just plot the spectrum of the signal; In HW8, we designed a FIR filter by putting some zeros in the frequency band where the noise lies; In HW9, we designed 4 filters, long weighted filter, short weighted filter, long unweighted filter and short unweighted filter to remove the noise; In HW10, we designed design 4 FIR filters using Hamming, Hann, Blackman, and Kaiser Window.

In this task, 4 kinds of filters (Butter, Chebyshev I, Chebyshev II, Elliptic) were designed to implement noise removing. There are 3 different values of stop-band attenuation, so 12 corresponding filters were produced. The orders are needed to satisfactorily suppress the high frequency noise are shown in the table below. (No more than 0.5 dB of ripple in a passband)

Basically, for the same stop-band attenuation, $L_{Butter} > L_{Cheb\ I} = L_{Cheb\ II} > L_{Elliptic}$.

Filter Type / Attenuation	Butter	Chebyshev I	Chebyshev II	Elliptic
20dB	8	4	4	3
40dB	13	7	7	5
60dB	18	9	9	6