

MIT OpenCourseWare
<http://ocw.mit.edu>

2.161 Signal Processing: Continuous and Discrete
Fall 2008

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
DEPARTMENT OF MECHANICAL ENGINEERING

2.161 Signal Processing - Continuous and Discrete
Fall Term 2008

Solution of Problem Set 7

Assigned: October 30, 2008

Due: November 6, 2008

Problem 1:

There are a number of ways to prove this – here are two:

(a) Start with an acausal filter of the same length

$$H_a(z) = \sum_{n=-(N-1)/2}^{(N-1)/2} h_n z^{-n}$$

with N odd, and with real, odd symmetric coefficients $h_n = -h_{-n}$ so that the impulse response is a real, odd function. Since

$$\{h_n\} \xleftrightarrow{\text{DTFT}} H(e^{j\Omega}),$$

from the properties of the DTFT, namely that a real, odd function has a imaginary, odd DTFT, then $H_a(e^{j\Omega})$ is imaginary and odd. $H_a(z)$ may be made causal by adding a delay of $(N-1)/2$, that is

$$H(z) = z^{-(N-1)/2} H_a(z)$$

so that

$$H(e^{j\Omega}) = e^{-j\Omega(N-1)/2} H_a(e^{j\Omega})$$

Since $H_a(e^{j\Omega})$ is imaginary

$$\angle H(e^{j\Omega}) = \angle j e^{-j\Omega(N-1)/2} = -\frac{(N-1)}{2}\Omega \pm \pi/2$$

which is linear with frequency $\Omega = \omega\Delta T$ (apart from possible jump discontinuities where $H_a(e^{j\Omega})$ changes sign. Note that in practical filters, any jump in the phase response will occur outside the pass-band, and is generally ignored.) The equivalent delay is found from the slope, and represents a delay of $(N-1)/2$ samples.

(b) Assume the causal filter and let

$$\begin{aligned} H(z) &= \sum_{n=0}^{N-1} h_n z^{-n} \\ &= h_0 z^0 + h_1 z^{-1} + \dots + h_{N-1} z^{-(N-1)} \end{aligned}$$

with N odd, and with odd symmetric coefficients about the mid-point, $h_n = -h_{N-1-n}$. Group the symmetric components together

$$H(z) = h_0 (z^0 - z^{-(N-1)}) + h_1 (z^{-1} - z^{-(N-2)}) + h_2 (z^{-2} - z^{-(N-2)}) + \dots + h_{(N-1)/2} z^{-(N-1)/2}$$

$$\begin{aligned}
&= \sum_{n=0}^{(N-1)/2-1} h_n \left(z^{-n} - z^{-(N-1-n)} \right) + h_{(N-1)/2} z^{-(N-1)/2} \\
&= z^{-(N-1)/2} \left[\sum_{n=0}^{(N-1)/2-1} h_n \left(z^{(N-1)/2-n} - z^{-((N-1)/2-n)} \right) \right]
\end{aligned}$$

since, as noted in the problem statement, for an odd-symmetric function about the mid-point $h_{(N-1)/2} = 0$. The frequency response is $H(e^{j\Omega}) = H(z)|_{z=e^{j\Omega}}$

$$\begin{aligned}
H(e^{j\Omega}) &= e^{-j\Omega(N-1)/2} \left[\sum_{n=0}^{(N-1)/2-1} h_n \left(e^{j\Omega((N-1)/2-n)} - e^{-j\Omega((N-1)/2-n)} \right) \right] \\
&= j e^{-j\Omega(N-1)/2} \left[\sum_{n=0}^{(N-1)/2-1} 2h_n \sin(\Omega((N-1)/2+n)) \right] \\
&= j e^{-j\Omega(N-1)/2} H_a(e^{j\Omega}),
\end{aligned}$$

The function $H_a(e^{j\Omega})$ is purely real, therefore

$$\angle H(e^{j\Omega}) = \angle j e^{-j\Omega(N-1)/2} = -\frac{(N-1)}{2}\Omega \pm \pi/2$$

which is linear with frequency $\Omega = \omega\Delta T$ (apart from possible jump discontinuities of π at sign changes in $H_a(e^{j\Omega})$, which as noted above occur outside the pass-band) and represents a delay of $(N-1)/2$ samples.

Problem 2:

- (a) The system has 7 zeros and 7 poles at the origin. It is therefore a non-recursive FIR system.
(b) The zeros are at:

$$1, \quad -\frac{4}{3}, \quad -\frac{3}{4}, \quad \frac{4}{3} \left(\cos\left(\frac{\pi}{3}\right) \pm j \sin\left(\frac{\pi}{3}\right) \right), \quad \frac{3}{4} \left(\cos\left(\frac{\pi}{3}\right) \pm j \sin\left(\frac{\pi}{3}\right) \right)$$

With the help of MATLAB:

```

z1 = 1;
z2 = -4/3;
z3 = -3/4;
z4 = (4/3)*(cos(pi/3) + i*sin(pi/3));
z5 = (4/3)*(cos(pi/3) - i*sin(pi/3));
z6 = (3/4)*(cos(pi/3) + i*sin(pi/3));
z7 = (3/4)*(cos(pi/3) - i*sin(pi/3));
z = [z1 z2 z3 z4 z5 z6 z7];
p = [0 0 0 0 0 0 0];
K=1;
H = tf(zpk(z, p, K, -1))

```

The transfer function is

$$\begin{aligned}
H(z) &= \frac{z^7 - z^6 + 2.79z^4 - 2.79z^3 + z^1 - z^0}{z^7} \\
&= z^0 - z^{-1} + 2.79z^{-3} - 2.79z^{-4} + z^{-6} - z^{-7}
\end{aligned}$$

so that this is an odd symmetric filter with $N = 8$ and

$$\{h_n\} = \{1, -1, 0, 2.79, -2.79, 0, 1, -1\}$$

Then $H(z^{j\Omega})$ has a linear phase shift with frequency

$$\angle H(e^{j\Omega}) = \angle \left(j e^{-j\Omega(N-1)/2} \right) = -\frac{7}{2}\Omega + \pi/2$$

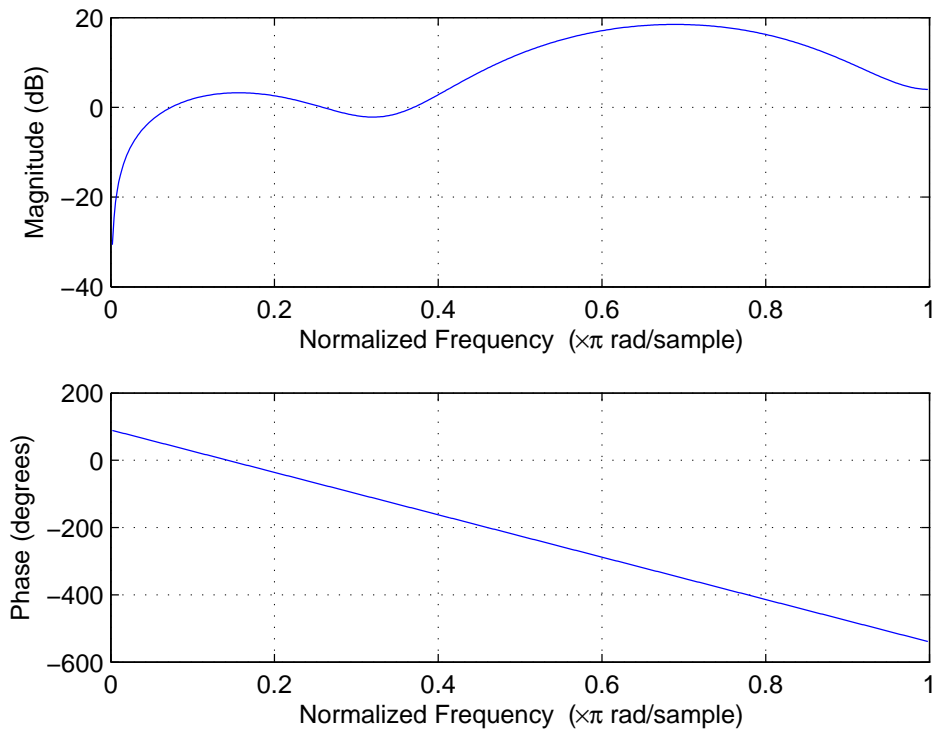
and the delay Δ is the slope, that is

$$\Delta = -\frac{d}{d\Omega} \angle H(e^{j\Omega}) = 7/2 \text{ steps.}$$

(c) The MATLAB command

```
freqz([1 -1 0 2.792 -2.792 0 1 -1], [1])
```

generates the following plot



which shows a crude “band-pass” characteristic. The phase plot shows a change in phase of $-7\pi/2$ over a range of π in frequency, corresponding to a delay of $7/2$ steps.

Problem 3: The following MATLAB script was used for this problem:

```
% Problem Set 6 -- Problem 3
% Enter the signal and contaminate it:
t = 0:.1:100;
signal = sin(4*pi*t);
noise = 2*(rand(size(signal))-0.5);
```

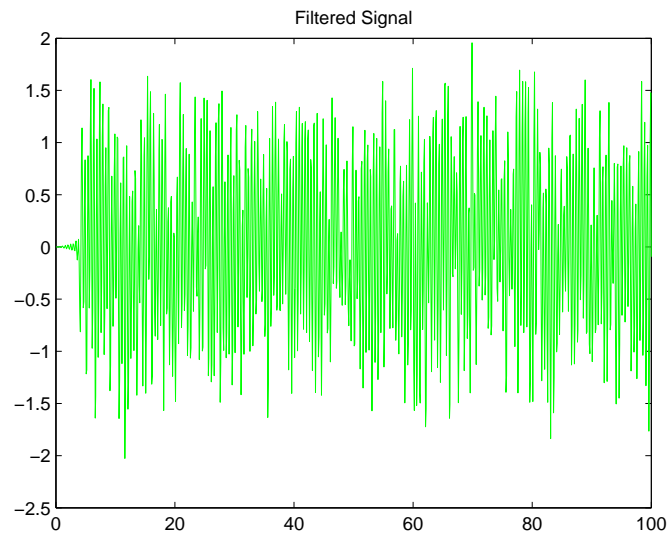
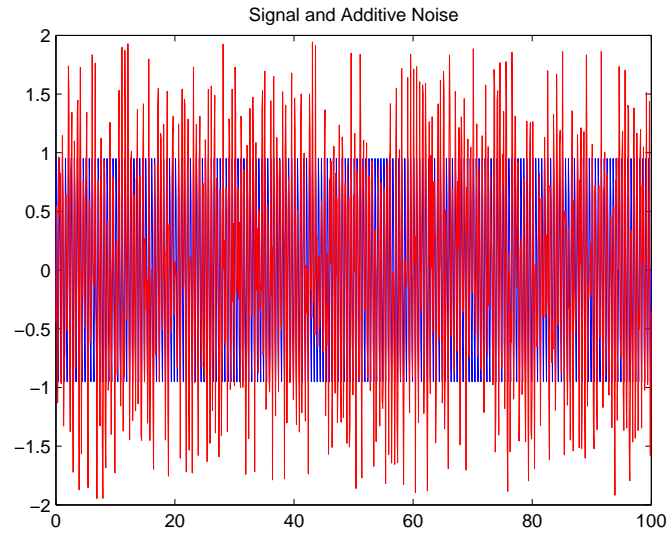
```

noisy_signal = signal + noise;
% Plot the waveforms
figure(1)
plot(t,signal,'b')
hold on
plot(t,noisy_signal,'r')
title('Signal and Additive Noise')
hold off
%
% Design the filter
%
w_c = 0.42;
w_stop = 0.48;
passband_ripple = 0.1;
stopband_ripple = 0.01
[N,Wn,BETA,FILTYPE] = kaiserord([w_c w_stop],[1 0], [passband_ripple stopband_ripple],2)
B = fir1(N,Wn,FILTYPE,kaiser(N+1,BETA));
A = 1;
%
% Filter the noisy data and plot the output
%
filtered_signal = filter(B,A,noisy_signal);
figure(2)
plot(t,filtered_signal,'g')
title('Filtered Signal')
%
% Plot the frequency response and pole-zero plots
%
figure(3)
freqz(B,A);
figure(4)
zplane(B,A)
df = 1/100.1;
f = -5:df:-5+1000*df;
figure(5)
plot(f,fftshift(abs(fft(noisy_signal))))
xlabel('Frequency (Hz)')
ylabel('Magnitude |F|')
title('Magnitude Spectrum of Noisy Signal')
figure(6)
plot(f,fftshift(abs(fft(filtered_signal))))
xlabel('Frequency (Hz)')
ylabel('Magnitude |F|')
title('Magnitude Spectrum of Low-Pass Filtered Signal')

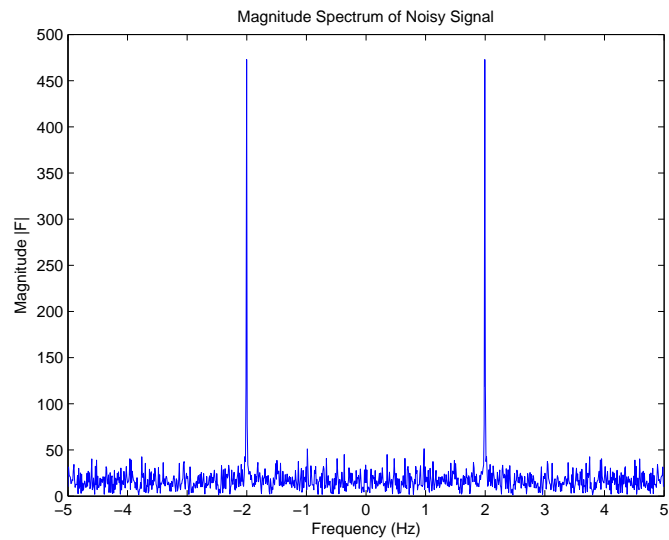
```

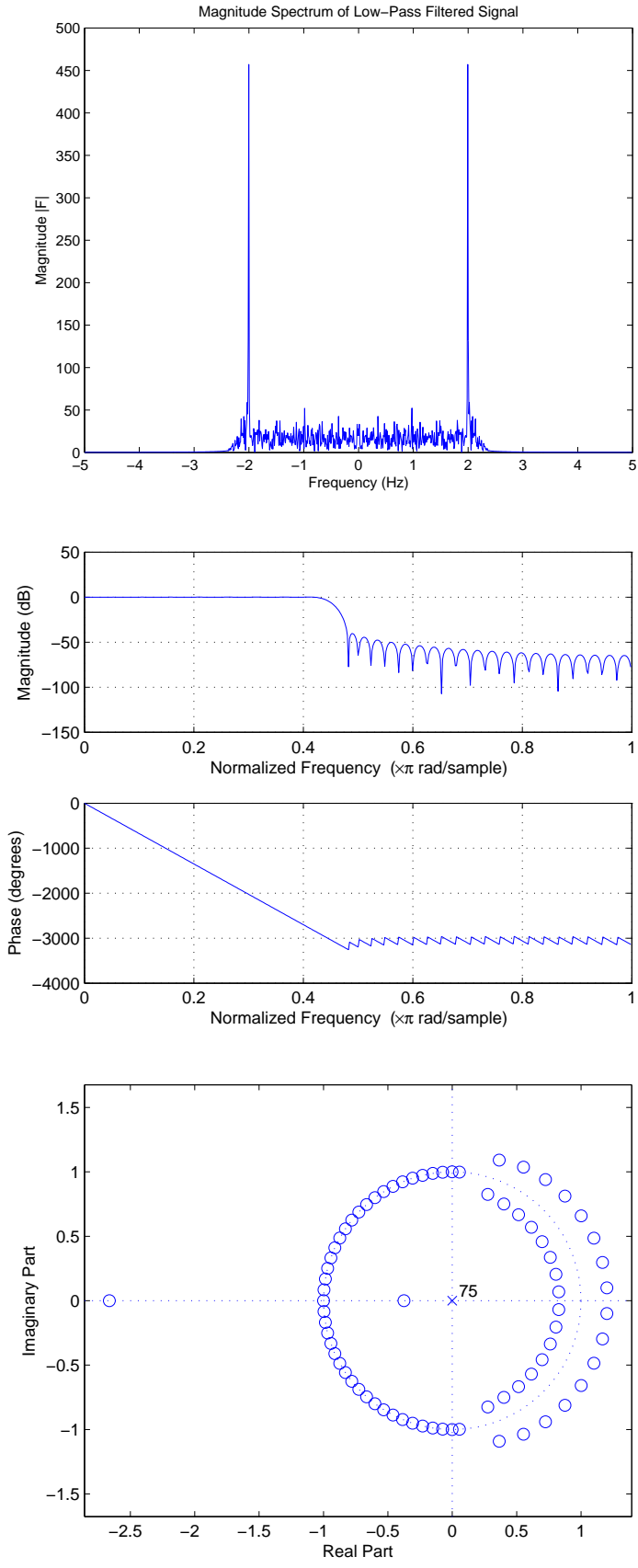
This script gave the filter parameters from `kaiserord()` as: $N = 75$, $\beta = 3.395$, $\omega_n = 0.45$.

The following plots were produced:



The following two spectra have been prepared using `fftshift()` to center them on 0 Hz,





Note that the pole-zero plot is composed with zeros on the unit-circle at angles Ω corresponding to frequencies above the cut-off frequency – thus forcing the magnitude response to zero at those

frequencies - this is therefore a low-pass filter. It is interesting to note that below the cut-off frequency, the zeros appear to be in pairs where their magnitudes are reciprocals - one inside and one outside the unit circle (in addition to be in complex conjugate pairs as well). Think about what this means (it also is evident in the pole-zero plot given in Problem 2, and we will see the same thing in the results of Problems 4 and 5).

Problem 4: The MATLAB script from Problem 3 was modified slightly for this problem to design a band-pass filter:

```
% Problem Set 6 -- Problem 4
% Enter the signal and contaminate it:
t = 0:.1:100;
signal = sin(4*pi*t);
noise = 2*(rand(size(signal))-0.5);
noisy_signal = signal + noise;
% Plot the waveforms
figure(1)
plot(t,signal,'b')
hold on
plot(t,noisy_signal,'r')
title('Signal and Additive Noise')
hold off
%
% Design the filter
%
% Frequency specs - normalized to the Nyquist frequency
w_sl=0.35;
w_cl=0.39;
w_cu = 0.41;
w_su = 0.45;
% Pass and stop band ripple
p_rip = 0.1;
s_rip = 0.01;
[N,Wn,BETA,TYPE] = kaiserord([w_sl w_cl w_cu w_su],[0 1 0], [s_rip p_rip s_rip],2)
B = fir1(N,Wn,TYPE, kaiser(N+1,BETA));
A = 1;
%
% Filter the noisy data and plot the output
%
filtered_signal = filter(B,A,noisy_signal);
figure(2)
plot(t,filtered_signal,'g')
title('Filtered Signal')
%
% Plot the frequency response and pole-zero plots
%
figure(3)
freqz(B,A);
figure(4)
zplane(B,A)
df = 1/100.1;
```

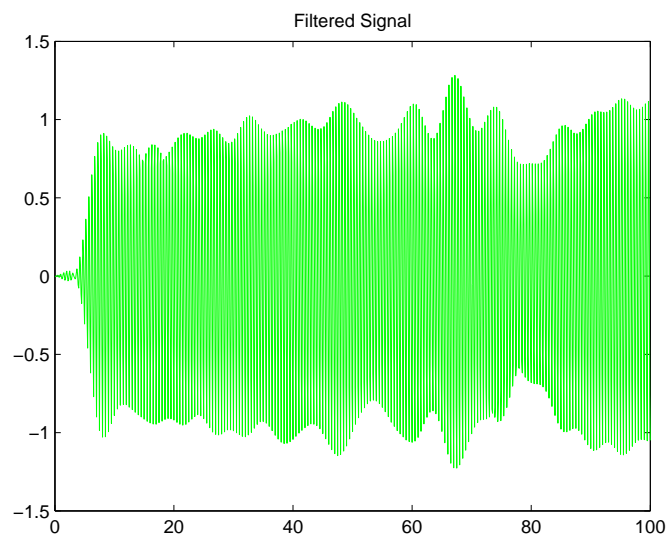
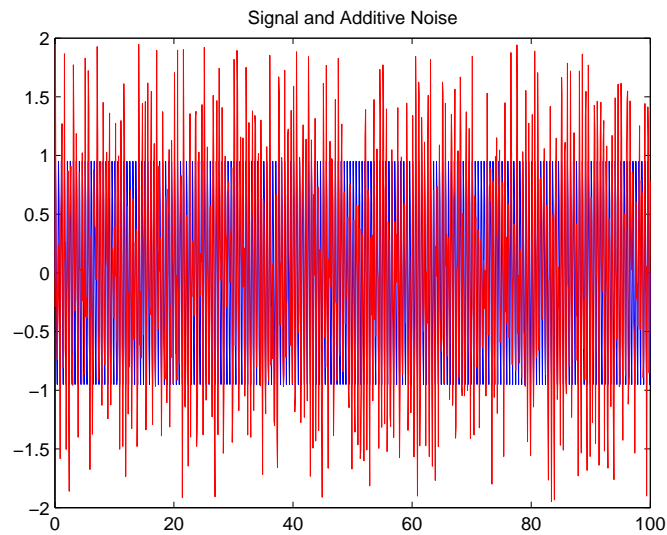


```

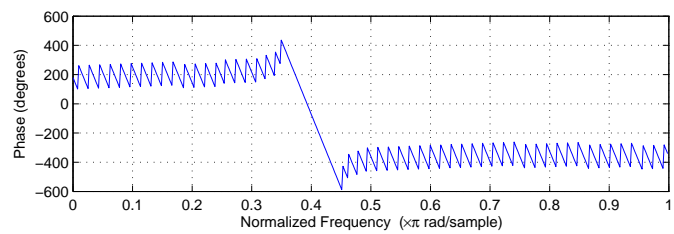
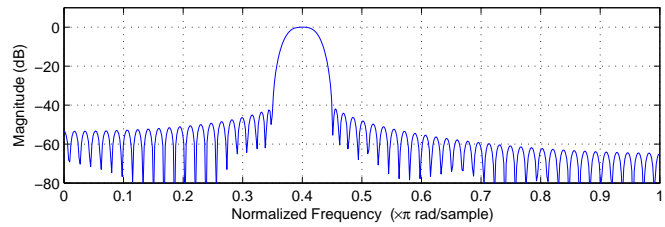
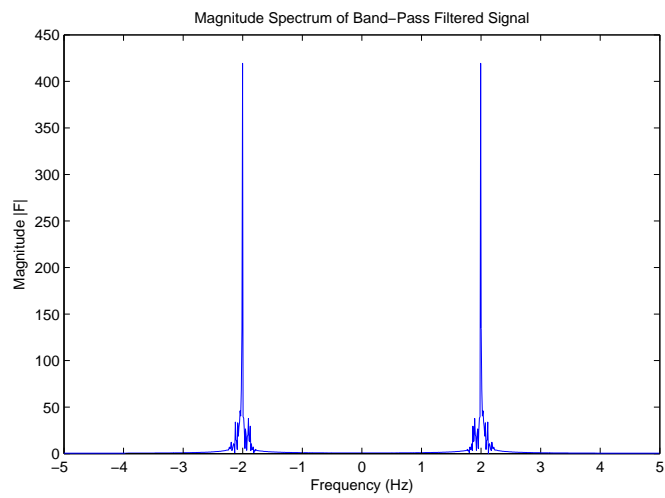
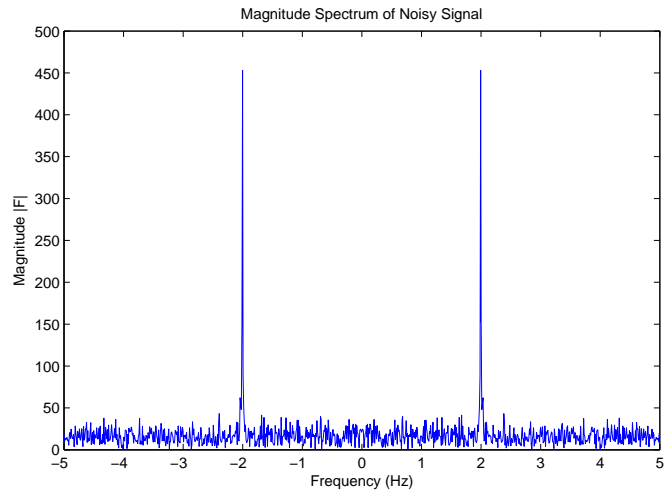
f = -5:df:-5+1000*df;
figure(5)
plot(f,fftshift(abs(fft(noisy_signal))))
xlabel('Frequency (Hz)')
ylabel('Magnitude |F|')
title('Magnitude Spectrum of Noisy Signal')
figure(6)
plot(f,fftshift(abs(fft(filtered_signal))))
xlabel('Frequency (Hz)')
ylabel('Magnitude |F|')
title('Magnitude Spectrum of Band-Pass Filtered Signal')

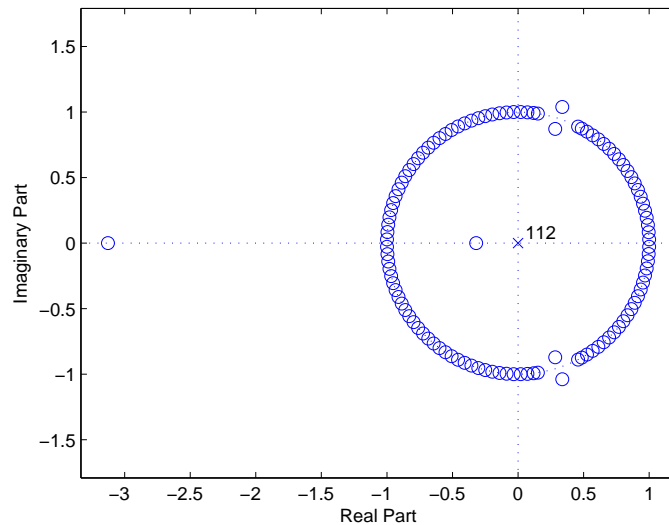
```

This script gave the filter parameters from `kaiserord()` as: $N = 112$, $\beta = 3.395$, $\omega_n = [0.37 \ 0.43]$.
The following plots were produced:



The following two spectra have been prepared using `fftshift()` to center them on 0 Hz,





Note that the pole-zero plot is composed with zeros on the unit-circle at angles Ω corresponding to frequencies above and below the cut-off frequency – thus forcing the magnitude response to zero at those frequencies - this is therefore a band-pass filter. As in Problem 3, it is interesting to note that within the pass-band the four zeros appear to be in pairs where their magnitudes are reciprocals - one inside and one outside the unit circle (in addition to be in complex conjugate pairs as well).

Problem 5: The following MATLAB script was used to design and test a band-stop filter:

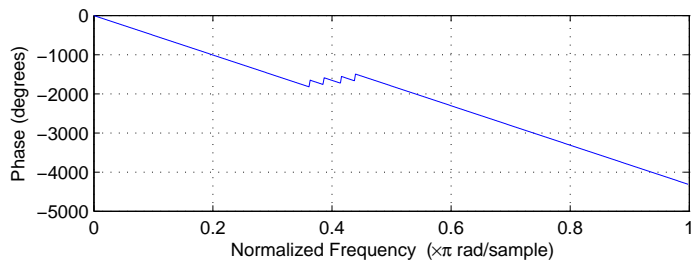
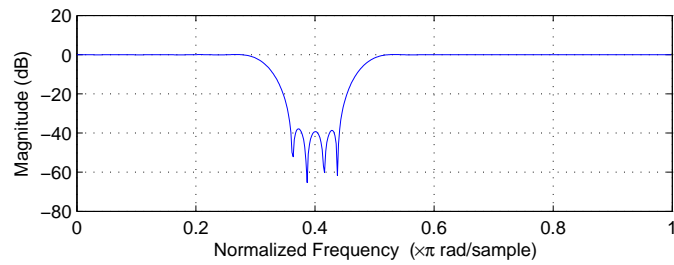
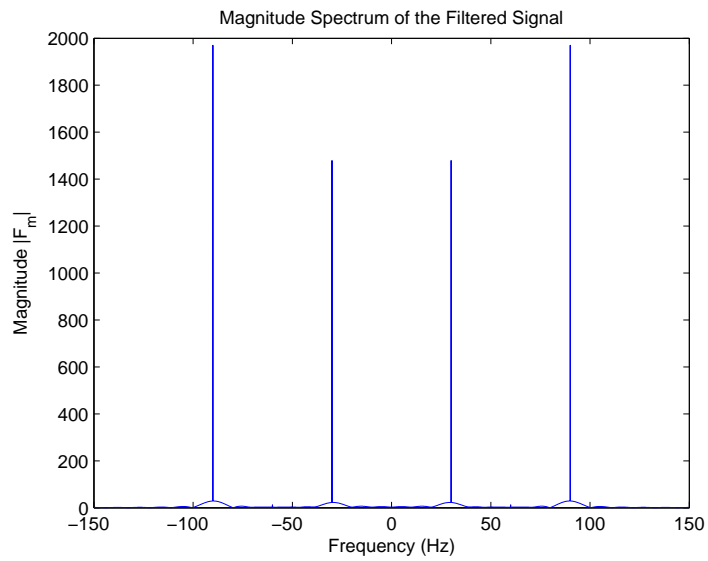
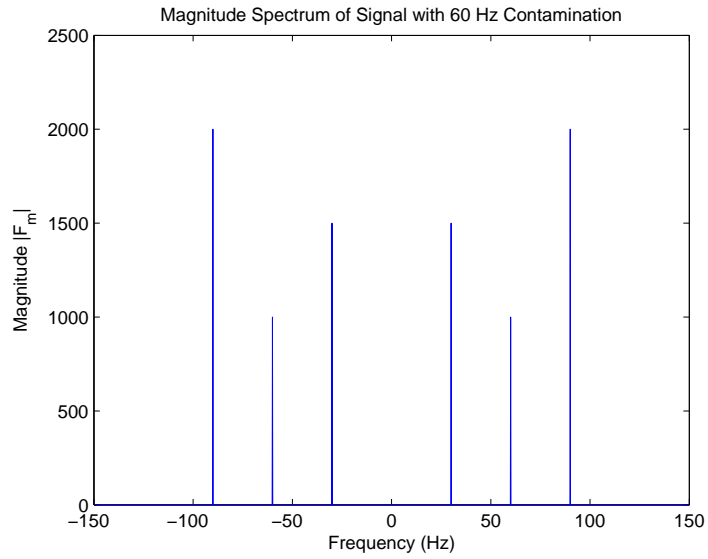
```
% Problem Set 6 -- Problem 5
% Enter the signal and contaminate it:
Dt = 1/300;
t = 0:Dt:1999*Dt;
f_sample = 1/Dt;
f_nyquist = f_sample/2;
%
signal = 1.5*sin(2*pi*30*t) + 2*cos(2*pi*90*t);
noise = sin(2*pi*60*t);
noisy_signal = signal + noise;
% Plot the magnitude spectrum of the
% contaminated signal
f = -150:1/(2000*Dt):-150+1999/(2000*Dt);
figure(1)
plot(f,fftshift(abs(fft(noisy_signal))))
title('Magnitude Spectrum of Signal with 60 Hz Contamination')
xlabel('Frequency (Hz)')
ylabel('Magnitude |F_m|')
f_norm = 60/f_nyquist;
%
% Design the filter
%
w_sl = f_norm*0.9;
w_cl = f_norm*0.7;
w_cu = f_norm*1.3;
w_su = f_norm*1.1;
```

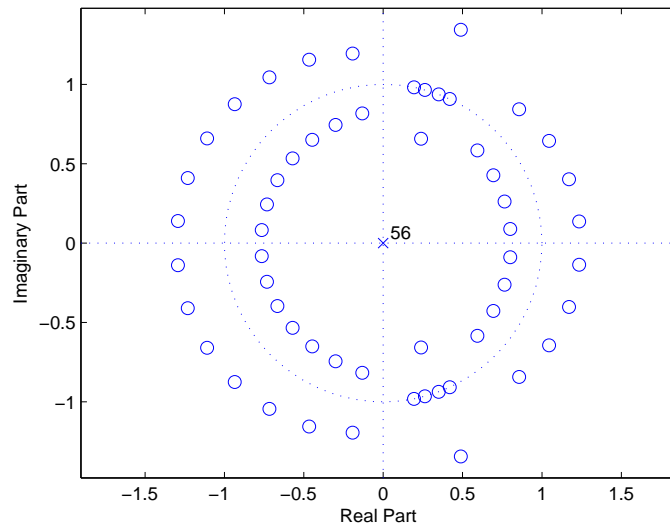
```

p_ripple = 0.1;
s_ripple = 0.01;
[N,Wn,BETA,TYPE] = kaiserord([w_cl w_sl w_su w_cu],[1 0 1], [p_rip s_rip p_rip],2)
B = fir1(N,Wn,TYPE, kaiser(N+1,BETA));
A = 1;
% Filter the noisy data and plot the output magnitude spectrum
%
figure(2)
filtered_signal = filter(B,A,noisy_signal);
plot(f, fftshift(abs(fft(filtered_signal))))
title('Magnitude Spectrum of the Filtered Signal')
xlabel('Frequency (Hz)')
ylabel('Magnitude |F_m|')
%
% Plot the frequency response and pole-zero plots
%
figure(3)
freqz(B,A);
figure(4)
zplane(B,A)
% Extra plots - not asked for!
% Plot segments of the input and filtered signals
% First interpolate the signals ant time by a factor of eight to make the
% plots more readable.
s_8 = resample(signal,8,1);
ns_8 = resample(noisy_signal,8,1);
fs_8 = resample(filtered_signal,8,1);
t_8 = resample(t,8,1);
% Plot the original signal and the contaminated signal
figure(5)
plot(t_8(300:500), s_8(300:500),'b--',t_8(300:500), ns_8(300:500),'r')
xlabel('Time (sec)')
ylabel('Signal')
title('Segment of Signal and Contaminated Signal')
figure(6)
plot(t_8(300:500), s_8(300:500),'b--',t_8(300:500), fs_8(300:500),'r')
xlabel('Time (sec)')
ylabel('Signal')
title('Segment of Uncontaminated Signal and Filter Output')
figure(7)
plot(t_8(1:500), s_8(1:500),'b--',t_8(1:500), fs_8(1:500),'r')
xlabel('Time (sec)')
ylabel('Signal')
title('Initial Segment of Uncontaminated Signal and Filter Output')

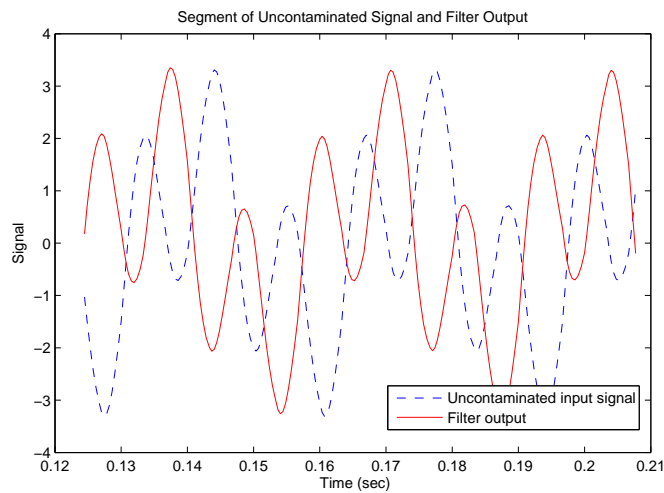
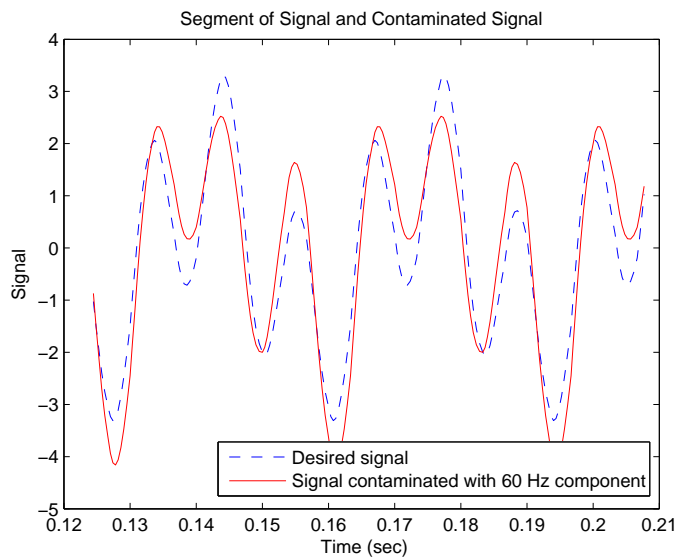
```

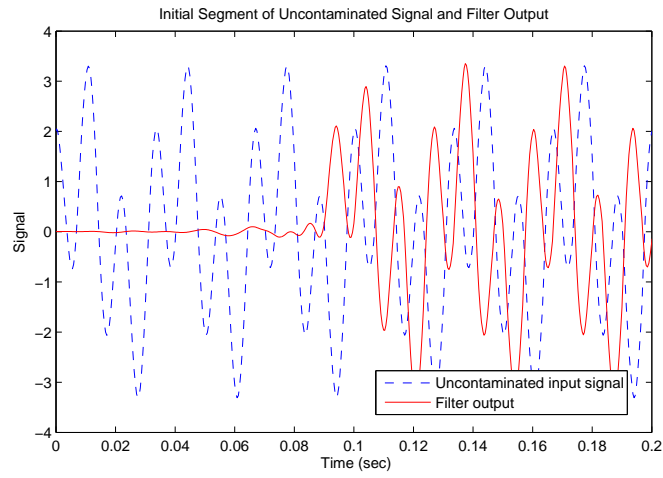
This script gave the filter parameters from `kaiserord()` as: $N = 56$, $\beta = 3.395$, $\omega_{cl} = 360$ rad/s, $\omega_{cu} = 480$ rad/s. The following plots were produced





The following plots were not asked for in the problem. To look at the waveforms it was necessary to interpolate the data records to show details of the waveforms, using `resample()`. See the script for details.





Note the delay in the filter output.