

Name: SOLUTION

Section:

Laboratory Exercise 7

DIGITAL FILTER DESIGN

7.1 DESIGN OF IIR FILTERS

Project 7.1 Estimation of IIR Filter Order

Answers:

From the problem statement of Q7.1, we have $F_T = 40$ kHz, $F_p = 4$ kHz, $F_s = 8$ kHz, $R_p = 0.5$ dB, and $R_s = 40$ dB.

Q7.1 The normalized passband edge angular frequency ω_p is –

$$\omega_p = \frac{2\pi F_p}{F_T} = \frac{2\pi(4 \times 10^3)}{40 \times 10^3} = \frac{\pi}{5} = 0.2\pi.$$
$$\omega_p = \frac{\omega_p}{\pi} = \frac{0.2\pi}{\pi} = 0.2$$

The normalized stopband edge angular frequency ω_s is –

$$\omega_s = \frac{2\pi F_s}{F_T} = \frac{2\pi(8 \times 10^3)}{40 \times 10^3} = \frac{2\pi}{5} = 0.4\pi.$$
$$\omega_s = \frac{\omega_s}{\pi} = \frac{0.4\pi}{\pi} = 0.4$$

The desired passband ripple R_p is - 0.5 dB

The desired stopband ripple R_s is – 40 dB

(1) Using these values and `buttord` we get the lowest order for a Butterworth lowpass filter to be – the correct call is `[N, Wn] = buttord(0.2, 0.4, 0.5, 40)`. This gives $N=8$.

The corresponding normalized passband edge frequency ω_n is - 0.2469, or $\omega_n = 0.2469\pi$.

(2) Using these values and `cheblord` we get the lowest order for a Type 1 Chebyshev lowpass filter to be - the correct call is

`[N, Wn] = cheblord(0.2, 0.4, 0.5, 40)`. This gives $N=5$.

The corresponding normalized passband edge frequency ω_n is - 0.2000, or $\omega_n = 0.2000\pi$.

(3) Using these values and `cheb2ord` we get the lowest order for a Type 2 Chebyshev lowpass filter to be - the correct call is

$$[N, Wn] = \text{cheb2ord}(0.2, 0.4, 0.5, 40). \text{ This gives } N=5.$$

The corresponding normalized passband edge frequency Wn is - 0.4000 , or $\omega_n = 0.4000\pi$.

(4) Using these values and `ellipord` we get the lowest order for an elliptic lowpass filter to be - the correct call is $[N, Wn] = \text{ellipord}(0.2, 0.4, 0.5, 40)$. This gives $N=4$ and $Wn = 0.2000$ or $\omega_n = 0.2000\pi$.

From the above results we observe that the Elliptic filter has the lowest order meeting the specifications.

Q7.2 The normalized passband edge angular frequency Wp is –

$$\omega_p = \frac{2\pi F_p}{F_T} = \frac{2\pi(1.050 \times 10^3)}{3.5 \times 10^3} = \frac{3\pi}{5} = 0.6\pi.$$

$$Wp = \frac{\omega_p}{\pi} = \frac{0.6\pi}{\pi} = 0.6$$

The normalized stopband edge angular frequency Ws is –

$$\omega_s = \frac{2\pi F_s}{F_T} = \frac{2\pi(0.6 \times 10^3)}{3.5 \times 10^3} = 0.3429\pi.$$

$$Ws = \frac{\omega_s}{\pi} = \frac{0.3429\pi}{\pi} = 0.3429$$

The desired passband ripple Rp is – 1.0 dB.

The desired stopband ripple Rs is – 50 dB.

(1) Using these values and `buttord` we get the lowest order for a Butterworth highpass filter to be – the correct call is $[N, Wn] = \text{buttord}(Wp, Ws, Rp, Rs)$. This gives $N=8$.

The corresponding normalized passband edge frequency Wn is – $Wn = 0.5646$, or $\omega_n = 0.5646\pi$.

(2) Using these values and `cheb1ord` we get the lowest order for a Type 1 Chebyshev highpass filter to be – the correct call is $[N, Wn] = \text{cheb1ord}(Wp, Ws, Rp, Rs)$. This gives $N=5$.

The corresponding normalized passband edge frequency W_n is – $W_n = 0.6000$, or
 $\omega_n = 0.6000\pi$.

(3) Using these values and `cheb2ord` we get the lowest order for a Type 2 Chebyshev highpass filter to be – the correct call is `[N, Wn] = cheb2ord(Wp, Ws, Rp, Rs)`. This gives $N=5$.

The corresponding normalized passband edge frequency W_n is – $W_n = 0.3429$, or
 $\omega_n = 0.3429\pi$.

(4) Using these values and `ellipord` we get the lowest order for an elliptic highpass filter to be – the correct call is `[N, Wn] = ellipord(Wp, Ws, Rp, Rs)`. This gives $N=4$.
The corresponding normalized passband edge frequency W_n is – $W_n = 0.6000$, or
 $\omega_n = 0.6000\pi$.

From the above results we observe that the Elliptic filter has the lowest order meeting the specifications.

Q7.3 The normalized passband edge angular frequency W_p is –

$$\begin{aligned}\omega_{p,1} &= \frac{2\pi F_{p,1}}{F_T} = \frac{2\pi(1.400 \times 10^3)}{7 \times 10^3} = \frac{2\pi}{5} = 0.4\pi. \\ W_{p1} &= \frac{\omega_{p,1}}{\pi} = \frac{0.4\pi}{\pi} = 0.4 \\ \omega_{p,2} &= \frac{2\pi F_{p,2}}{F_T} = \frac{2\pi(2.100 \times 10^3)}{7 \times 10^3} = \frac{3\pi}{5} = 0.6\pi. \\ W_{p2} &= \frac{\omega_{p,2}}{\pi} = \frac{0.6\pi}{\pi} = 0.6 \\ W_p &= [W_{p1} \ W_{p2}] = [0.4000 \ 0.6000]\end{aligned}$$

The normalized stopband edge angular frequency W_s is –

$$\begin{aligned}\omega_{s,1} &= \frac{2\pi F_{s,1}}{F_T} = \frac{2\pi(1.050 \times 10^3)}{7 \times 10^3} = \frac{3\pi}{10} = 0.3\pi. \\ W_{s1} &= \frac{\omega_{s,1}}{\pi} = \frac{0.3\pi}{\pi} = 0.3 \\ \omega_{s,2} &= \frac{2\pi F_{s,2}}{F_T} = \frac{2\pi(2.450 \times 10^3)}{7 \times 10^3} = \frac{7\pi}{10} = 0.7\pi.\end{aligned}$$

$$W_{s2} = \frac{\omega_{s,2}}{\pi} = \frac{0.7\pi}{\pi} = 0.7$$

$$W_s = [W_{s1} \ W_{s2}] = [0.3000 \ 0.7000]$$

The desired passband ripple R_p is -0.4 dB.

The desired stopband ripple R_s is -50 dB.

(1) Using these values and `buttord` we get the lowest order for a Butterworth bandpass filter to be – the correct call is `[N,Wn] = buttord(Wp,Ws,Rp,Rs) = buttord([0.4000 0.6000],[0.3000 0.7000],0.4,50)`, which gives `Order = 2N = 18`.

The corresponding normalized passband edge frequency W_n is –

$$W_n = [0.3835 \ 0.6165], \text{ or}$$

$$\omega_{n,1} = 0.3835\pi \text{ and } \omega_{n,2} = 0.6165\pi.$$

(2) Using these values and `cheblord` we get the lowest order for a Type 1 Chebyshev bandpass filter to be – the correct call is `[N,Wn] = cheblord(Wp,Ws,Rp,Rs) = cheblord([0.40000 0.6000],[0.3000 0.7000],0.4,50)`, which gives `Order = 2N = 12`.

The corresponding normalized passband edge frequency W_n is –

$$W_n = [0.4000 \ 0.6000], \text{ or}$$

$$\omega_{n,1} = 0.4000\pi \text{ and } \omega_{n,2} = 0.6000\pi.$$

(3) Using these values and `cheb2ord` we get the lowest order for a Type 2 Chebyshev bandpass filter to be – the correct call is `[N,Wn] = cheb2ord(Wp,Ws,Rp,Rs) = cheb2ord([0.4000 0.6000],[0.3000 0.7000],0.4,50)`, which gives `Order = 2N = 12`.

The corresponding normalized passband edge frequency W_n is –

$$W_n = [0.3000 \ 0.7000], \text{ or}$$

$$\omega_{n,1} = 0.3000\pi \text{ and } \omega_{n,2} = 0.7000\pi.$$

(4) Using these values and `ellipord` we get the lowest order for an elliptic bandpass filter to be – the correct call is `[N,Wn] = ellipord(Wp,Ws,Rp,Rs) = ellipord([0.4000 0.6000],[0.3000 0.7000],0.4,50)`, which gives `Order = 2N = 8`.

The corresponding normalized passband edge frequency `Wn` is –

$$Wn = [0.4000 \ 0.6000], \text{ or}$$

$$\omega_{n,1} = 0.4000\pi \text{ and } \omega_{n,2} = 0.6000\pi.$$

From the above results we observe that the Elliptic filter has the lowest order meeting the specifications.

Q7.4 The normalized passband edge angular frequency `Wp` is –

$$\omega_{p,1} = \frac{2\pi F_{p,1}}{F_T} = \frac{2\pi(2.100 \times 10^3)}{12 \times 10^3} = 0.35\pi.$$

$$Wp1 = \frac{\omega_{p,1}}{\pi} = \frac{0.35\pi}{\pi} = 0.35$$

$$\omega_{p,2} = \frac{2\pi F_{p,2}}{F_T} = \frac{2\pi(4.500 \times 10^3)}{12 \times 10^3} = 0.75\pi.$$

$$Wp2 = \frac{\omega_{p,2}}{\pi} = \frac{0.75\pi}{\pi} = 0.75$$

$$Wp = [Wp1 \ Wp2] = [0.3500 \ 0.7500]$$

The normalized stopband edge angular frequency `Ws` is –

$$\omega_{s,1} = \frac{2\pi F_{s,1}}{F_T} = \frac{2\pi(2.700 \times 10^3)}{12 \times 10^3} = 0.45\pi.$$

$$Ws1 = \frac{\omega_{s,1}}{\pi} = \frac{0.45\pi}{\pi} = 0.45$$

$$\omega_{s,2} = \frac{2\pi F_{s,2}}{F_T} = \frac{2\pi(3.900 \times 10^3)}{12 \times 10^3} = 0.65\pi.$$

$$Ws2 = \frac{\omega_{s,2}}{\pi} = \frac{0.65\pi}{\pi} = 0.65$$

$$Ws = [Ws1 \ Ws2] = [0.4500 \ 0.6500]$$

The desired passband ripple `Rp` is – `Rp = 0.6 dB`.

The desired stopband ripple `Rs` is – `Rs = 45 dB`.

(1) Using these values and `buttord` we get the lowest order for a Butterworth bandstop filter to be – the correct call is `[N,Wn] = buttord(Wp,Ws,Rp,Rs) = buttord([0.3500 0.7500],[0.4500 0.6500],0.6,45)`, which gives `Order = 2N = 18`.

The corresponding normalized passband edge frequent `Wn` is –

$$Wn = [0.3783 \ 0.7123], \text{ or } \omega_{n,1} = 0.3783\pi \text{ and } \omega_{n,2} = 0.7123\pi.$$

(2) Using these values and `cheblord` we get the lowest order for a Type 1 Chebyshev bandstop filter to be – the correct call is `[N,Wn] = cheblord(Wp,Ws,Rp,Rs) = cheblord([0.3500 0.7500],[0.4500 0.6500],0.6,45)`, which gives `Order = 2N = 10`.

The corresponding normalized passband edge frequency `Wn` is –

$$Wn = [0.3500 \ 0.7500], \text{ or } \omega_{n,1} = 0.3500\pi \text{ and } \omega_{n,2} = 0.7500\pi.$$

(3) Using these values and `cheb2ord` we get the lowest order for a Type 2 Chebyshev bandstop filter to be – the correct call is `[N,Wn] = cheb2ord(Wp,Ws,Rp,Rs) = cheb2ord([0.3500 0.7500],[0.4500 0.6500],0.6,45)`, which gives `Order = 2N = 10`.

The corresponding normalized passband edge frequency `Wn` is –

$$Wn = [0.4500 \ 0.6500], \text{ or } \omega_{n,1} = 0.4500\pi \text{ and } \omega_{n,2} = 0.6500\pi.$$

(4) Using these values and `ellipord` we get the lowest order for an elliptic bandstop filter to be – the correct call is `[N,Wn] = ellipord(Wp,Ws,Rp,Rs) = ellipord([0.3500 0.7500],[0.4500 0.6500],0.6,45)`, which gives `Order = 2N = 8`.

The corresponding normalized passband edge frequency `Wn` is –

$$Wn = [0.3500 \ 0.7500], \text{ or }$$

$$\omega_{n,1} = 0.3500\pi \text{ and } \omega_{n,2} = 0.7500\pi.$$

From the above results we observe that the Elliptic filter has the lowest order meeting the specifications.

Project 7.2 IIR Filter Design

A copy of Program P7_1 is given below:

```
% Program P7_1
% Design of a Butterworth Bandstop Digital Filter
Ws = [0.4 0.6]; Wp = [0.2 0.8]; Rp = 0.4; Rs = 50;
% Estimate the Filter Order
[N1, Wn1] = buttord(Wp, Ws, Rp, Rs);
% Design the Filter
[num,den] = butter(N1,Wn1,'stop');
% Display the transfer function
disp('Numerator Coefficients are ');disp(num);
disp('Denominator Coefficients are ');disp(den);
% Compute the gain response
[g, w] = gain(num,den);
% Plot the gain response
plot(w/pi,g);grid
axis([0 1 -60 5]);
xlabel('\omega /\pi'); ylabel('Gain in dB');
title('Gain Response of a Butterworth Bandstop Filter');
```

Answers:

Q7.5 The coefficients of the Butterworth bandstop transfer function generated by running Program P7_1 are as follows:

Numerator Coefficients are

Columns 1 through 9
0.0493 0.0000 0.2465 0.0000 0.4930 0.0000 0.4930 0.0000 0.2465

Columns 10 through 11

0.0 0.0493

Denominator Coefficients are

Columns 1 through 9

1.0000 0.0000 -0.0850 0.0000 0.6360 0.0000 -0.0288 0.0000 0.0561

Columns 10 through 11

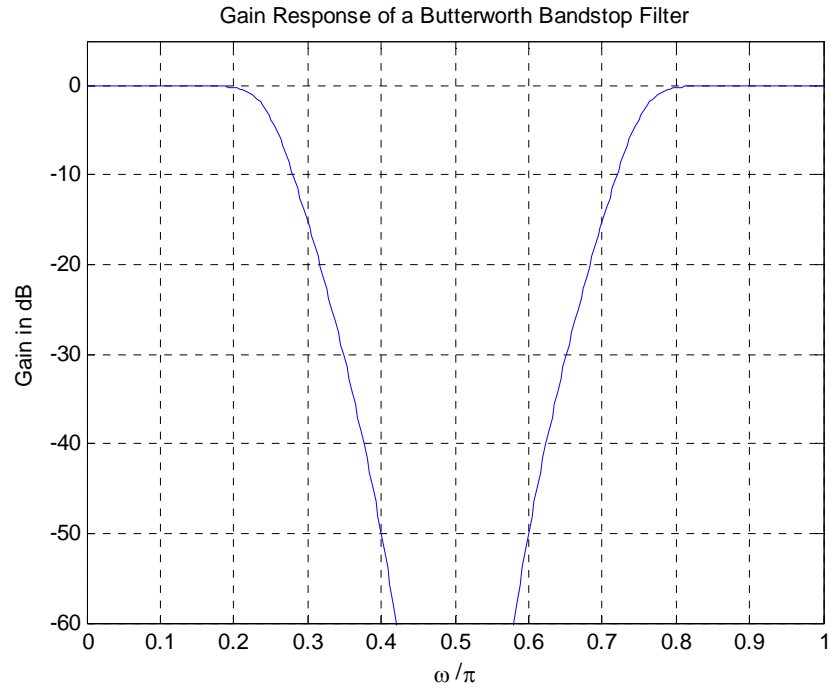
0.0000 -0.0008

The exact expression for the transfer function is –

$$H(z) = \frac{0.0493 + 0.2465z^{-2} + 0.4930z^{-4} + 0.4930z^{-6} + 0.2465z^{-8} + 0.0493z^{-10}}{1 - 0.0850z^{-2} + 0.6360z^{-4} - 0.0288z^{-6} + 0.0561z^{-8} - 0.0008z^{-10}}$$

The filter specifications are: $\omega_{p1} = 0.2\pi$, $\omega_{s1} = 0.4\pi$, $\omega_{s2} = 0.6\pi$, $\omega_{p2} = 0.8\pi$, $R_p = 0.4$ dB, and $R_s = 50$ dB.

The gain response of the filter as designed is given below:

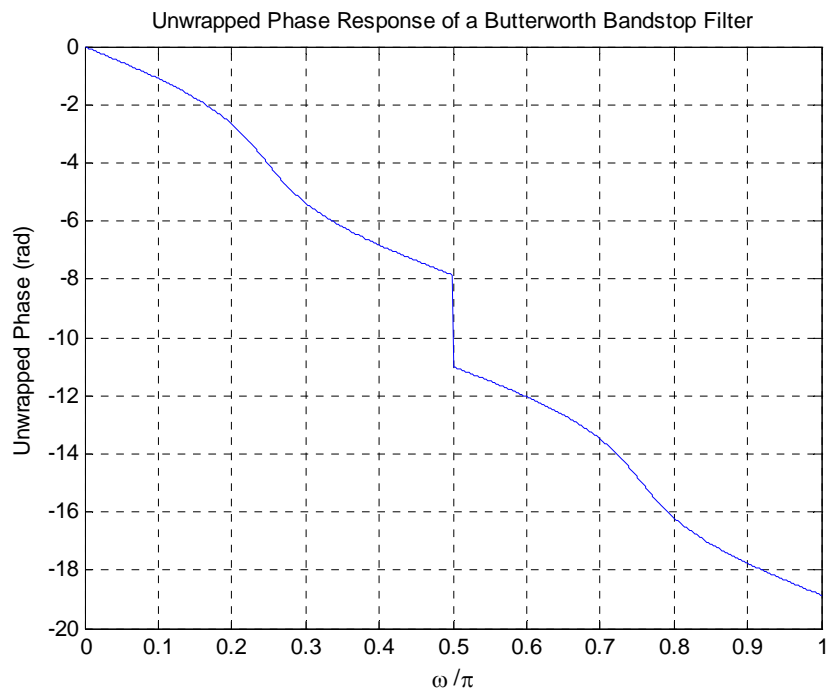


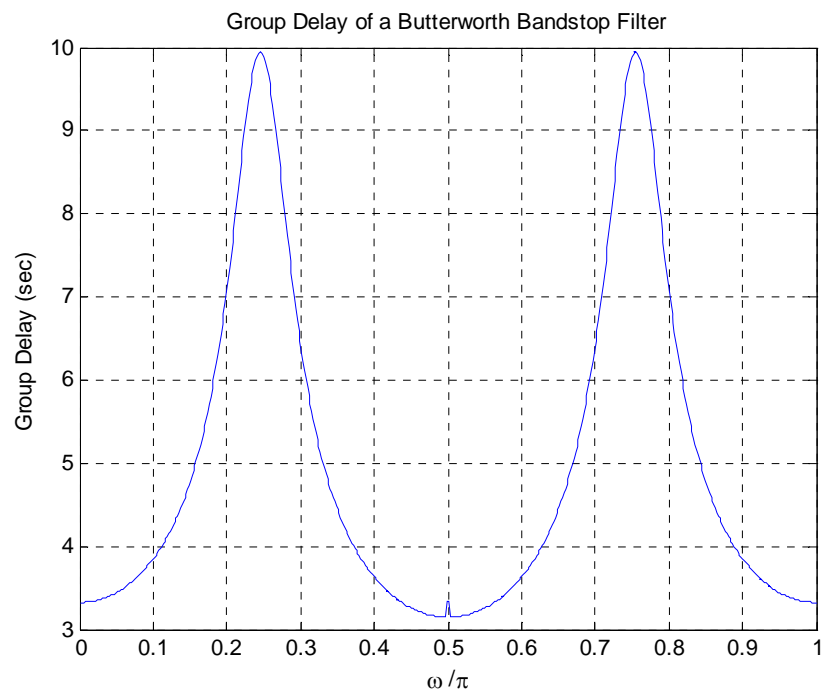
From the plot we conclude that the design MEETS the specifications.

The plot of the unwrapped phase response and the group delay response of this filter is given below:

Here is the program to find and plot the unwrapped phase response and group delay:

```
% Program Q7_5B
% Design of a Butterworth Bandstop Digital Filter for Q7.5.
% Plot the unwrapped phase and the group delay.
Ws = [0.4 0.6]; Wp = [0.2 0.8]; Rp = 0.4; Rs = 50;
% Estimate the Filter Order
[N1, Wn1] = buttord(Wp, Ws, Rp, Rs);
% Design the Filter
[num,den] = butter(N1,Wn1,'stop');
% Find the frequency response; find and plot unwrapped phase
wp = 0:pi/1023:pi;
wg = 0:pi/511:pi;
Hz = freqz(num,den,wp);
Phase = unwrap(angle(Hz));
figure(1);
plot(wp/pi,Phase);
grid;
% axis([0 1 a b]);
xlabel('\omega /\pi'); ylabel('Unwrapped Phase (rad)');
title('Unwrapped Phase Response of a Butterworth Bandstop Filter');
% Find and plot the group delay
GR = grpdelay(num,den,wg);
figure(2);
plot(wg/pi,GR);
grid;
%axis([0 1 a b]);
xlabel('\omega /\pi'); ylabel('Group Delay (sec)');
title('Group Delay of a Butterworth Bandstop Filter');
```





Here is the modified version of Program P7_1 to answer Q7.6:

```
% Program Q7_6
% Design of a Chebyshev Type 1 Lowpass Digital Filter
% meeting the design specification given in Q7.1.
% - Print out the numerator and denominator coefficients
%   for the transfer function.
% - Compute and plot the gain function.
% - Compute and plot the unwrapped phase response.
% - Compute and plot the group delay.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Design spec as given in Q7.1.
FT = 40*10^3;      % sampling freq
Fp = 4*10^3;       % analog passband edge freq
Fs = 8*10^3;       % analog stopband edge freq
Rp = 0.5;          % max passband ripple, dB
Rs = 40;           % min stopband attenuation, dB
% Convert spec to normalized digital frequencies
omega_p = 2*pi*Fp/FT;
Wp = 2*Fp/FT;      % omega_p/pi
omega_s = 2*pi*Fs/FT;
Ws = 2*Fs/FT;      % omega_s/pi
% Estimate the Filter Order
[N, Wn] = cheblord(Wp, Ws, Rp, Rs);
% Design the Filter
[num,den] = cheby1(N,Rp,Wn);
% Display the transfer function
disp('Numerator Coefficients are ');disp(num);
disp('Denominator Coefficients are ');disp(den);
% Compute the gain response
[g, w] = gain(num,den);
% Plot the gain response
figure(1);
plot(w/pi,g);grid;
axis([0 1 -60 5]);
xlabel('\omega /\pi'); ylabel('Gain in dB');
title('Gain Response of a Type 1 Chebyshev Lowpass Filter');
% Find and plot the phase
figure(2);
w2 = 0:pi/511:pi;
Hz = freqz(num,den,w2);
Phase = unwrap(angle(Hz));
plot(w2/pi,Phase);grid;
xlabel('\omega /\pi'); ylabel('Unwrapped Phase (rad)');
title('Unwrapped Phase Response of a Type 1 Chebyshev Lowpass Filter');
% Find and plot the group delay
figure(3);
GR = grpdelay(num,den,w2);
plot(w2/pi,GR);grid;
xlabel('\omega /\pi'); ylabel('Group Delay (sec)');
title('Group Delay of a Type 1 Chebyshev Lowpass Filter');
```

Q7.6 The coefficients of the Type 1 Chebyshev lowpass transfer function for the parameters given in Question 7.1 and generated by running modified Program P7_1 are as follows:

Numerator Coefficients are

0.0004 0.0020 0.0040 0.0040 0.0020 0.0004

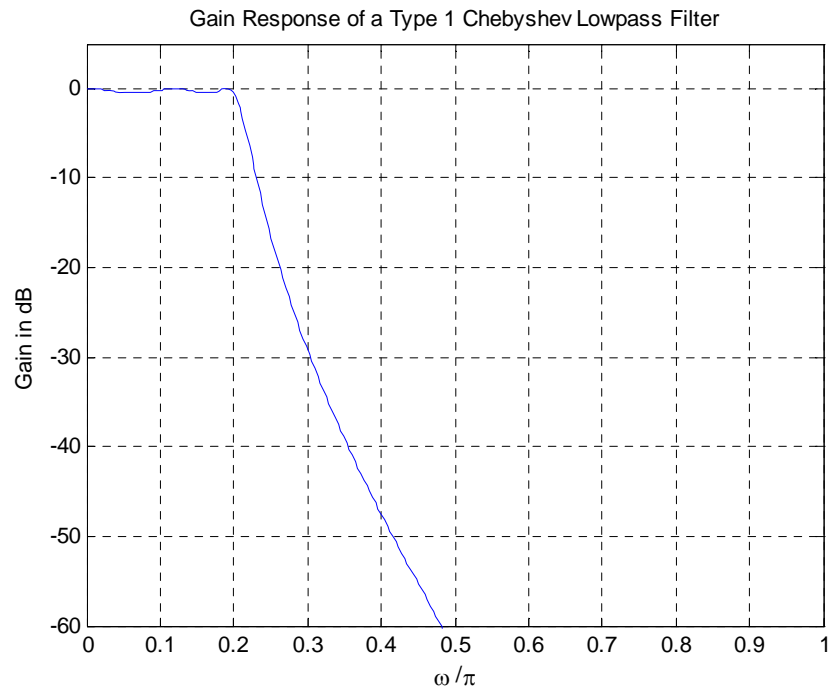
Denominator Coefficients are

1.0000 -3.8269 6.2742 -5.4464 2.4915 -0.4797

The exact expression for the transfer function is –

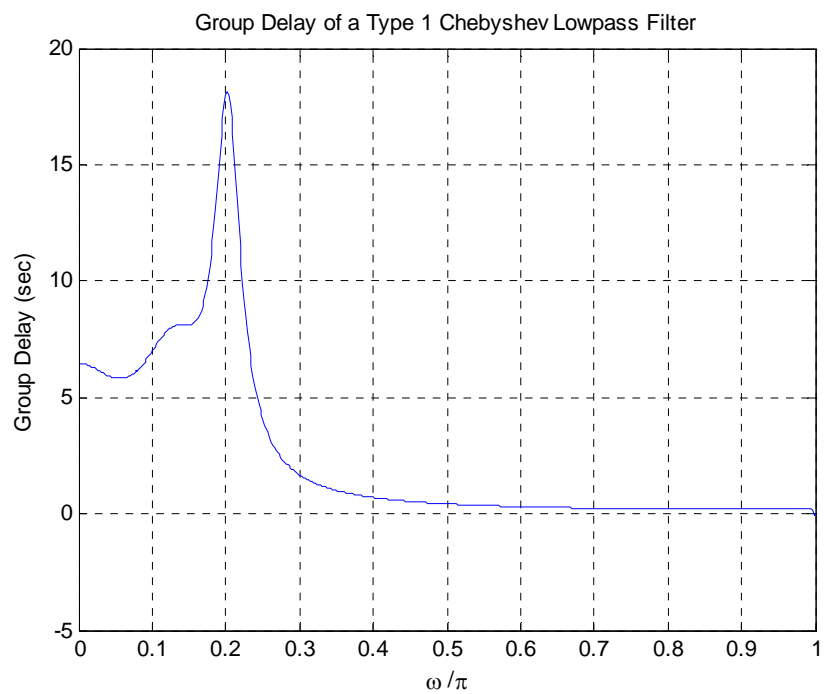
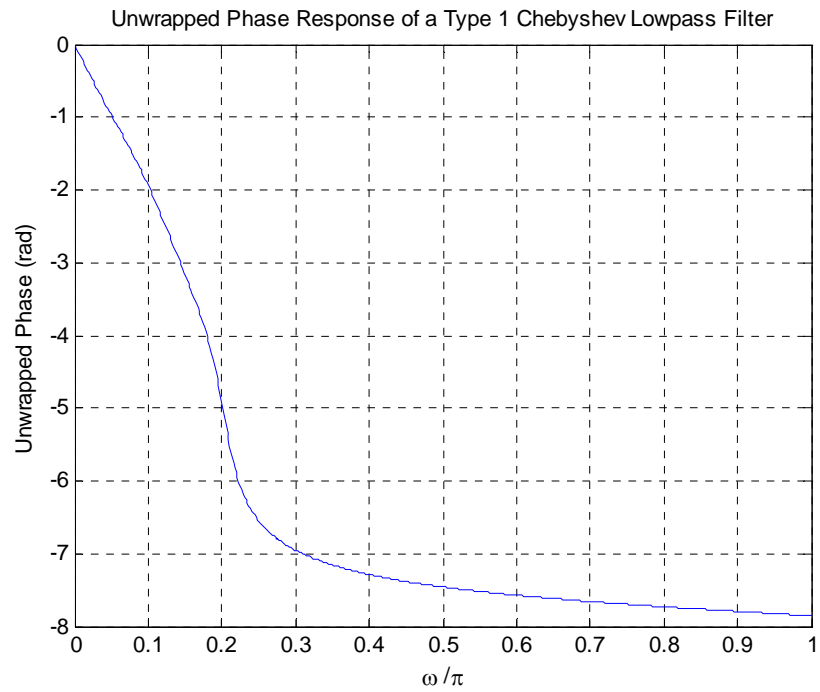
$$H(z) = \frac{0.0004 + 0.0020z^{-1} + 0.0040z^{-2} + 0.0040z^{-3} + 0.0020z^{-4} + 0.0004z^{-5}}{1 - 3.8269z^{-1} + 6.2742z^{-2} - 5.4464z^{-3} + 2.4915z^{-4} - 0.4797z^{-5}}$$

The gain response of the filter as designed is given below:



From the plot we conclude that the design MEETS the specifications.

The plot of the unwrapped phase response and the group delay response of this filter is given below:



Here is the modified version of Program P7_1 for answering question Q7.7:

```
% Program Q7_7
% Design of a Chebyshev Type 2 Highpass Digital Filter
% meeting the design specification given in Q7.2.
% - Print out the numerator and denominator coefficients
%   for the transfer function.
% - Compute and plot the gain function.
% - Compute and plot the unwrapped phase response.
% - Compute and plot the group delay.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Design spec as given in Q7.2.
FT = 3.50*10^3;      % sampling freq
Fp = 1.05*10^3;      % analog passband edge freq
Fs = 0.60*10^3;      % analog stopband edge freq
Rp = 1.0;            % max passband ripple, dB
Rs = 50;             % min stopband attenuation, dB
% Convert spec to normalized digital frequencies
omega_p = 2*pi*Fp/FT;
Wp = 2*Fp/FT;        % omega_p/pi
omega_s = 2*pi*Fs/FT;
Ws = 2*Fs/FT;        % omega_s/pi
% Estimate the Filter Order
[N, Wn] = cheb2ord(Wp, Ws, Rp, Rs);
% Design the Filter
[num,den] = cheby2(N,Rs,Wn,'high');
% Display the transfer function
disp('Numerator Coefficients are ');disp(num);
disp('Denominator Coefficients are ');disp(den);
% Compute the gain response
[g, w] = gain(num,den);
% Plot the gain response
figure(1);
axis([0 1 -60 5]);
% Add lines to the plot to help determine if the spec was met.
hold on;
tmpY = -60:65/511:5;
tmpX = ones(1,length(tmpY))*Wp;
plot(tmpX,tmpY,'r-'); % vertical line at passband edge freq
tmpX = ones(1,length(tmpY))*Ws;
plot(tmpX,tmpY,'g-'); % vertical line at stopband edge freq
tmpY = ones(1,length(w))*(-Rp);
plot(w/pi,tmpY,'r-'); % horizontal line at Rp
tmpY = ones(1,length(w))*(-Rs);
plot(w/pi,tmpY,'g-'); % horizontal line at Rs
% now plot the gain
plot(w/pi,g);grid;
xlabel('\omega /\pi'); ylabel('Gain in dB');
title('Gain Response of a Type 2 Chebyshev Highpass Filter');
hold off;
% Find and plot the phase
figure(2);
w2 = 0:pi/511:pi;
Hz = freqz(num,den,w2);
Phase = unwrap(angle(Hz));
plot(w2/pi,Phase);grid;
xlabel('\omega /\pi'); ylabel('Unwrapped Phase (rad)');
title('Unwrapped Phase Response of a Type 2 Chebyshev Highpass Filter');
```

```

% Find and plot the group delay
figure(3);
GR = grpdelay(num,den,w2);
plot(w2/pi,GR);grid;
xlabel('\omega /\pi'); ylabel('Group Delay (sec)');
title('Group Delay of a Type 2 Chebyshev Highpass Filter');

```

Q7.7 The coefficients of the Type 1 Chebyshev highpass transfer function for the parameters given in Question 7.2 and generated by running modified Program P7_1 are as follows:

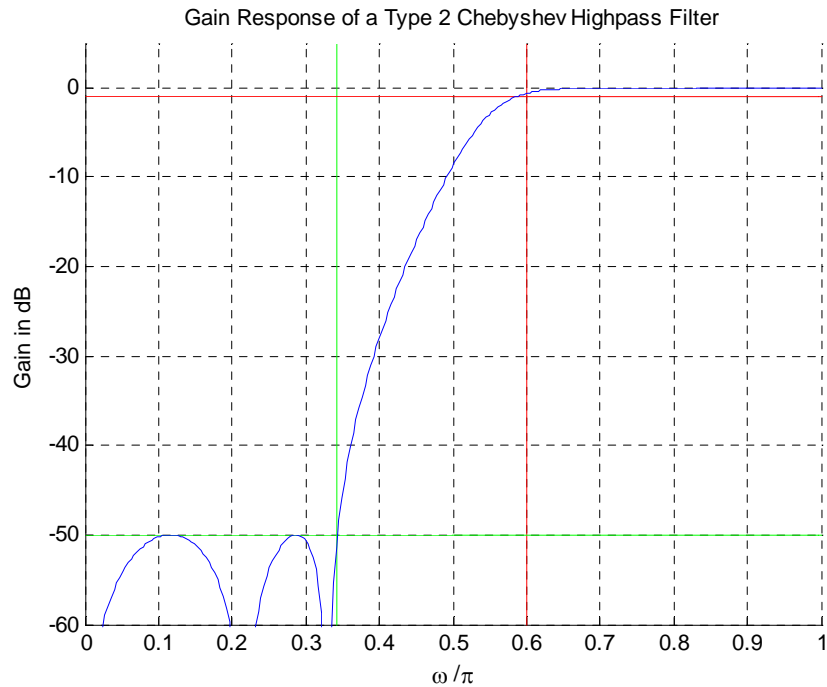
Numerator Coefficients are
0.0671 -0.2404 0.4146 -0.4146 0.2404 -0.0671

Denominator Coefficients are
1.0000 0.2933 0.7303 0.0711 0.0783 0.0001

The exact expression for the transfer function is –

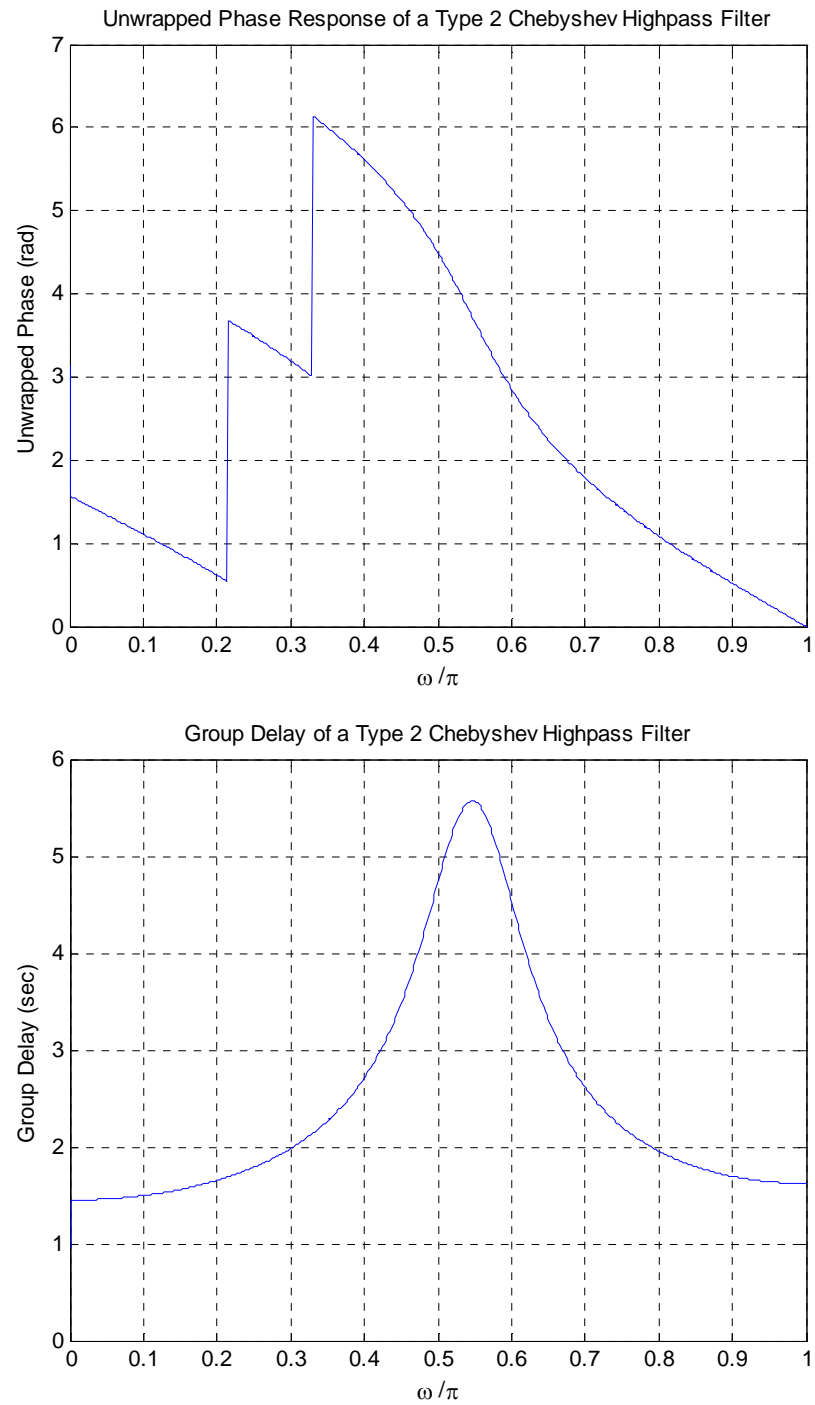
$$H(z) = \frac{0.0671 - 0.2404z^{-1} + 0.4146z^{-2} - 0.4146z^{-3} + 0.2404z^{-4} - 0.0671z^{-5}}{1 + 0.2933z^{-1} + 0.7303z^{-2} + 0.0711z^{-3} + 0.0783z^{-4} + 0.0001z^{-5}}$$

The gain response of the filter as designed is given below:



From the plot we conclude that the design MEETS the specifications.

The plot of the unwrapped phase response and the group delay response of this filter is given below:



Here is the modified version of Program P7_1 for answering question Q7.8:

```
% Program Q7_8
% Design of an Elliptic Bandpass Digital Filter
% meeting the design specification given in Q7.3.
% - Print out the numerator and denominator coefficients
%   for the transfer function.
% - Compute and plot the gain function.
% - Compute and plot the unwrapped phase response.
% - Compute and plot the group delay.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Design spec as given in Q7.3.
FT = 7.00*10^3;      % sampling freq
Fp1 = 1.4*10^3;      % analog lower passband edge freq
Fp2 = 2.1*10^3;      % analog upper passband edge freq
Fs1 = 1.05*10^3;     % analog lower stopband edge freq
Fs2 = 2.45*10^3;     % analog upper stopband edge freq
Rp = 0.4;            % max passband ripple, dB
Rs = 50;             % min stopband attenuation, dB
% Convert spec to normalized digital frequencies
omega_p1 = 2*pi*Fp1/FT;
Wp1 = 2*Fp1/FT;      % omega_p1/pi
omega_p2 = 2*pi*Fp2/FT;
Wp2 = 2*Fp2/FT;      % omega_p2/pi
Wp = [Wp1 Wp2];
omega_s1 = 2*pi*Fs1/FT;
Ws1 = 2*Fs1/FT;      % omega_s1/pi
omega_s2 = 2*pi*Fs2/FT;
Ws2 = 2*Fs2/FT;      % omega_s2/pi
Ws = [Ws1 Ws2];
% Estimate the Filter Order
[N, Wn] = ellipord(Wp, Ws, Rp, Rs);
% Design the Filter
[num,den] = ellip(N,Rp,Rs,Wn);
% Display the transfer function
disp('Numerator Coefficients are ');disp(num);
disp('Denominator Coefficients are ');disp(den);
% Compute the gain response
[g, w] = gain(num,den);
% Plot the gain response
figure(1);
axis([0 1 -60 5]);
% Add lines to the plot to help determine if the spec was met.
hold on;
tmpY = -60:65/511:5;
tmpX = ones(1,length(tmpY))*Wp1;
plot(tmpX,tmpY,'r-'); % vertical line at passband edge freq
tmpX = ones(1,length(tmpY))*Wp2;
plot(tmpX,tmpY,'r-'); % vertical line at passband edge freq
tmpX = ones(1,length(tmpY))*Ws1;
plot(tmpX,tmpY,'g-'); % vertical line at stopband edge freq
tmpX = ones(1,length(tmpY))*Ws2;
plot(tmpX,tmpY,'g-'); % vertical line at stopband edge freq
tmpY = ones(1,length(w))*(-Rp);
plot(w/pi,tmpY,'r-'); % horizontal line at Rp
tmpY = ones(1,length(w))*(-Rs);
plot(w/pi,tmpY,'g-'); % horizontal line at Rs
% now plot the gain
plot(w/pi,g);grid;
```

```

xlabel('\omega /\pi'); ylabel('Gain in dB');
title('Gain Response of an Elliptic Bandpass Filter');
hold off;
% Find and plot the phase
figure(2);
w2 = 0:pi/511:pi;
Hz = freqz(num,den,w2);
Phase = unwrap(angle(Hz));
plot(w2/pi,Phase);grid;
xlabel('\omega /\pi'); ylabel('Unwrapped Phase (rad)');
title('Unwrapped Phase Response of an Elliptic Bandpass Filter');
% Find and plot the group delay
figure(3);
GR = grpdelay(num,den,w2);
plot(w2/pi,GR);grid;
xlabel('\omega /\pi'); ylabel('Group Delay (sec)');
title('Group Delay of an Elliptic Bandpass Filter');

```

Q7.8 The coefficients of the elliptic bandpass transfer function for the parameters given in Question

7.3 and generated by running modified Program P7_1 are as follows:

Numerator Coefficients are

0.0116	-0.0000	-0.0046	-0.0000	0.0166	-0.0000	-0.0046
-0.0000	0.0116					

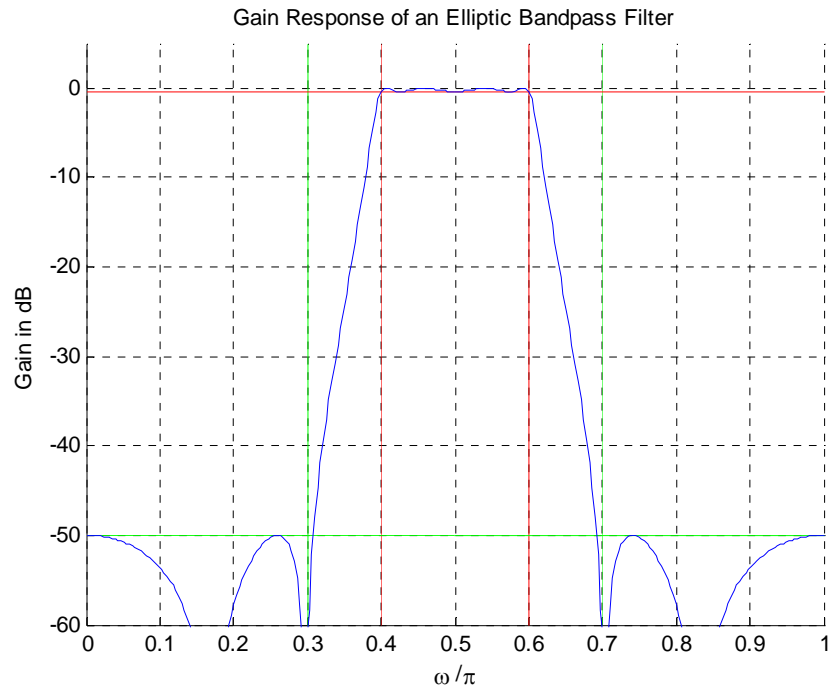
Denominator Coefficients are

1.0000	-0.0000	2.8611	-0.0000	3.4205	-0.0000	1.9609
-0.0000	0.4529					

The exact expression for the transfer function is –

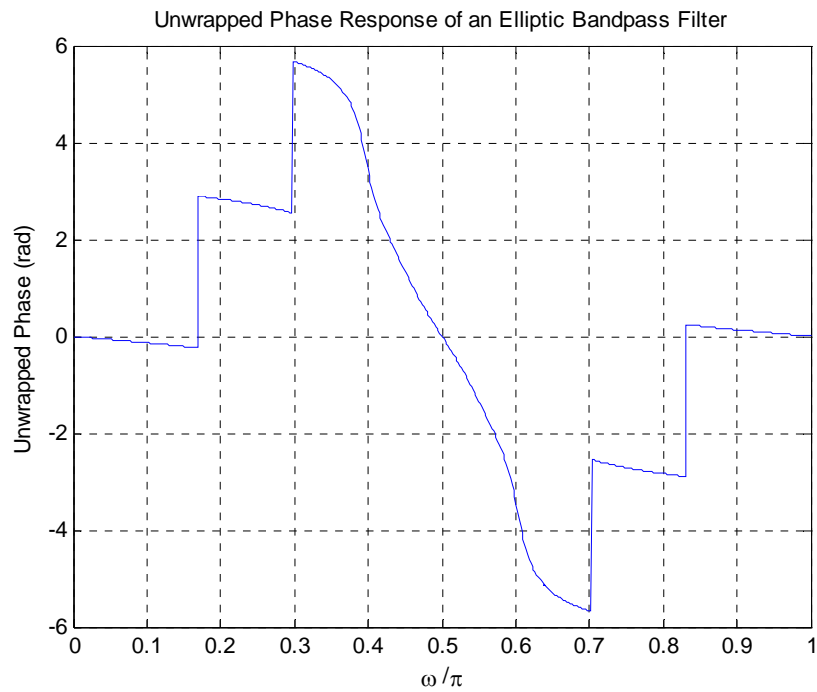
$$H(z) = \frac{0.0116 - 0.0046z^{-2} + 0.0166z^{-4} - 0.0046z^{-6} + 0.0116z^{-8}}{1 + 2.8611z^{-2} + 3.4205z^{-4} + 1.9609z^{-6} + 0.4529z^{-8}}$$

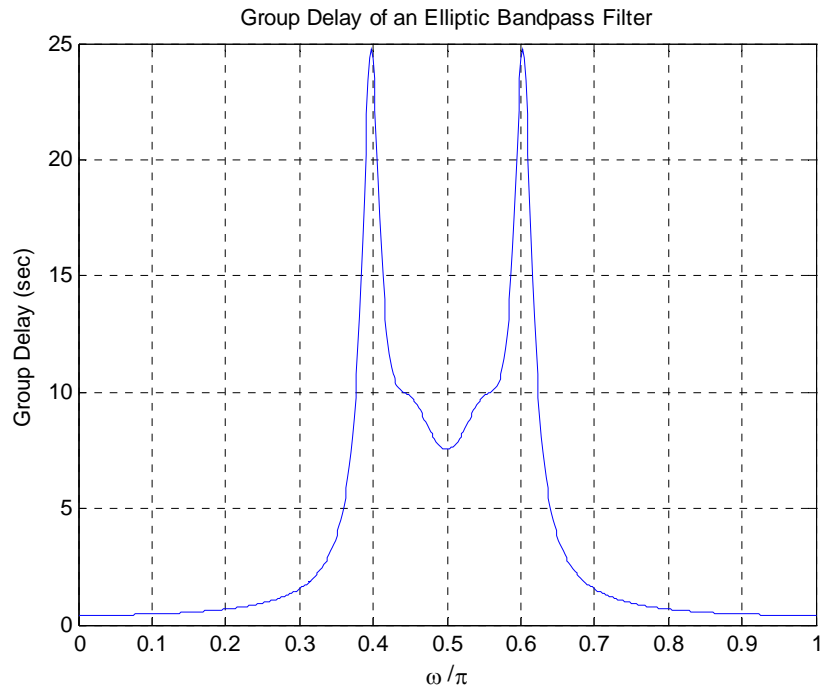
The gain response of the filter as designed is given below:



From the plot we conclude that the design MEETS the specifications.

The plot of the unwrapped phase response and the group delay response of this filter is given below:





7.2 DESIGN OF FIR FILTERS

Project 7.3 Gibb's Phenomenon

Answers:

Q7.9 The MATLAB program generating the impulse response, truncated to 81 samples, of a zero-phase ideal lowpass filter with a cutoff at $\omega_c = 0.4\pi$ and plotting its magnitude response is given below: NOTE: this program does all four lengths.

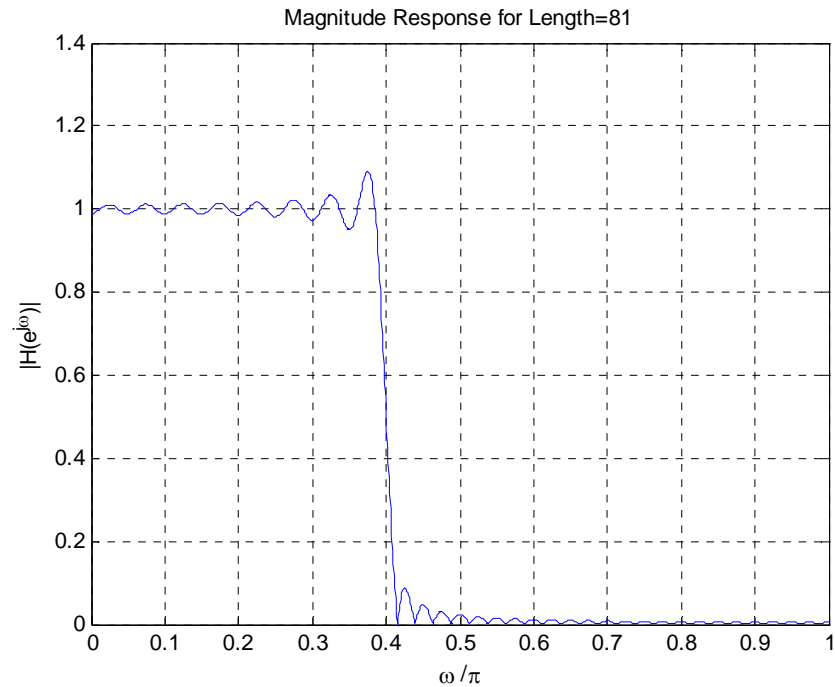
```
% Program Q7_9
% Investigate Gibbs phenomena for a FIR lowpass filter as
% asked for in Q7.9.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n = -40:40; % this gives us a length of 81
hn_81 = 0.4 * sinc(0.4*n); % the length-81 impulse response
omega = 0:pi/1023:pi; % radian frequency vector
W = omega/pi; % Matlab normalized freq vector
Hz_81 = abs(freqz(hn_81,1,omega)); % 1024 samples of |H(e^jw)|
figure(1);
plot(W,HZ_81); grid;
xlabel('\omega / \pi'); ylabel('|H(e^{j\omega})|');
title('Magnitude Response for Length=81');
% Reduce length to 61 and repeat
hn_61 = hn_81(11:71);
```

```

Hz_61 = abs(freqz(hn_61,1,omega));
figure(2);
plot(W,HZ_61); grid;
xlabel('\omega /\pi'); ylabel('|H(e^{j\omega})|');
title('Magnitude Response for Length=61');
% Reduce length to 41 and repeat
hn_41 = hn_61(11:51);
Hz_41 = abs(freqz(hn_41,1,omega));
figure(3);
plot(W,HZ_41); grid;
xlabel('\omega /\pi'); ylabel('|H(e^{j\omega})|');
title('Magnitude Response for Length=41');
% Reduce length to 21 and repeat
hn_21 = hn_41(11:31);
Hz_21 = abs(freqz(hn_21,1,omega));
figure(4);
plot(W,HZ_21); grid;
xlabel('\omega /\pi'); ylabel('|H(e^{j\omega})|');
title('Magnitude Response for Length=21');

```

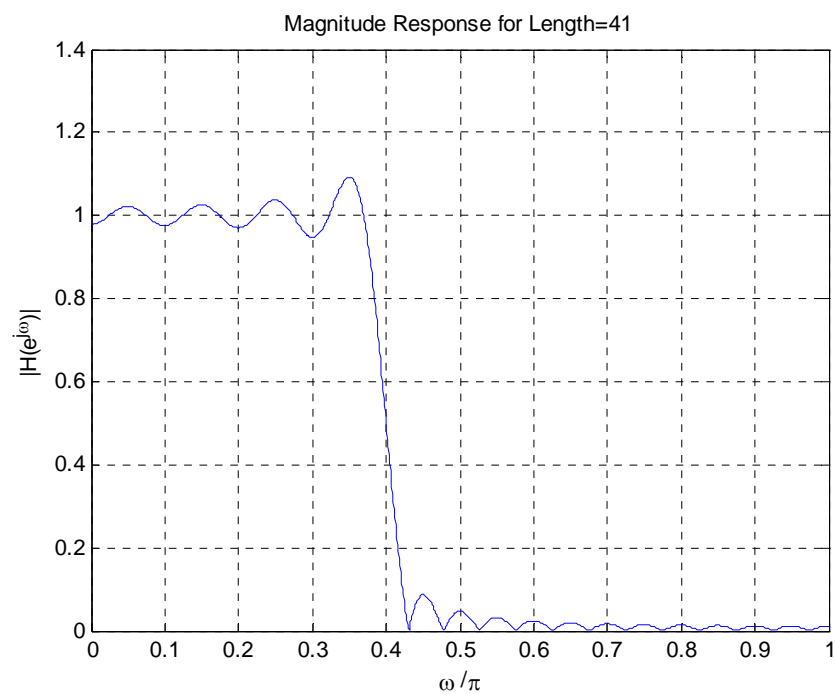
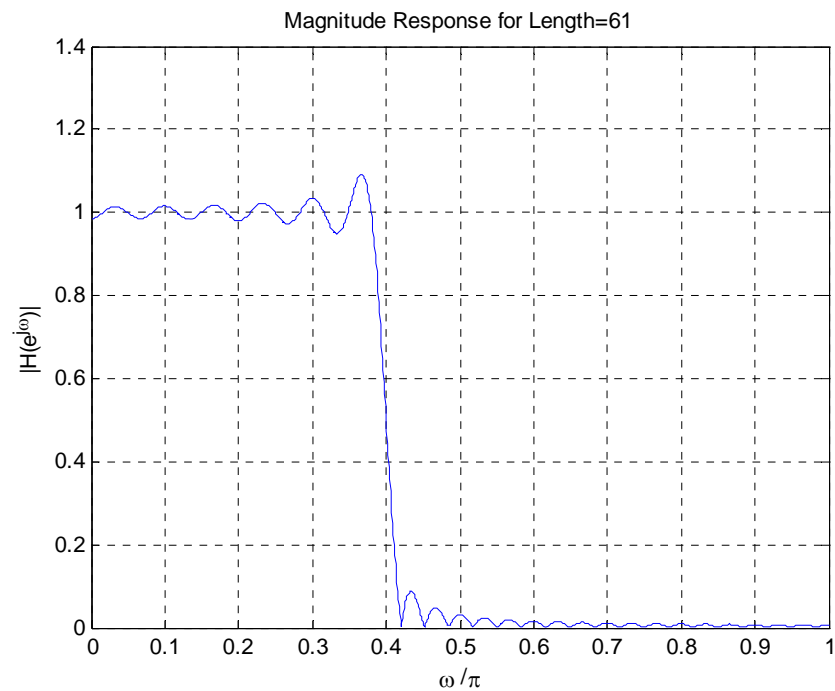
The plot of the magnitude response generated by running this program is as shown below:

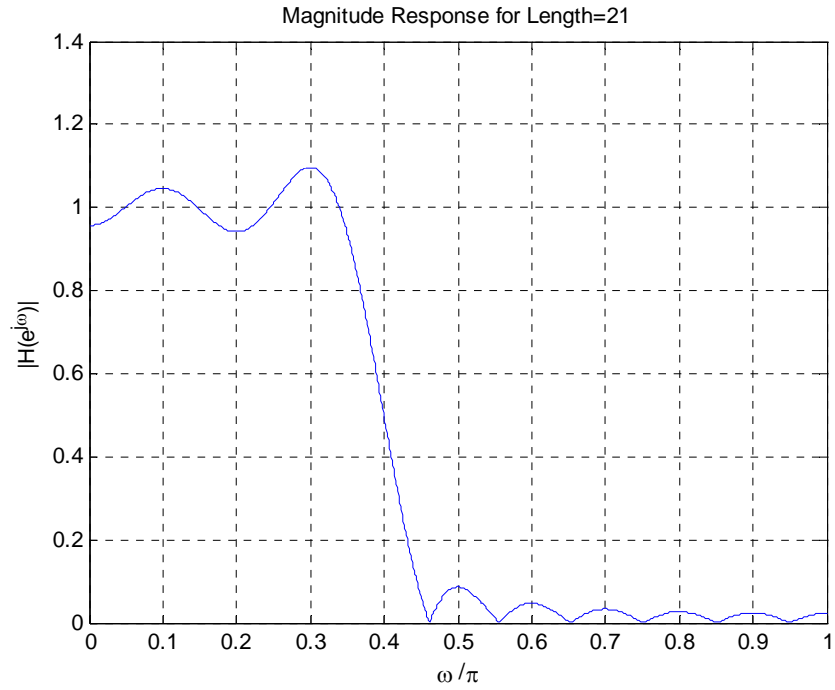


The program was modified as indicated below to extract the coefficients of a shorter length filter using the colon operator:

The code shown above does this already.

The magnitude response plots generated by running the modified program for the following lengths, 61, 41, and 21, are given below:





From these plots we observe the oscillatory behavior of the magnitude responses in each case due to the Gibb's phenomenon. The relation between the number of ripples and the length of the filter is – The number of ripples decreases in direct proportion to the length. The relation between the heights of the largest ripples and the length of the filter is – the peak ripple height is not affected by the length: it stays the same no matter what the length.

The modified program to generate impulse response coefficients for an even-length filter is given below –

```
% Program Q7_9B
% Investigate Gibbs phenomena for a FIR lowpass filter as
% asked for in Q7.9, with an even filter length.
% The "trick" to make the length even is to offset the array "n" by 0.5
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n = -39.5:39.5; % this gives us a length of 80
hn_80 = 0.4 * sinc(0.4*n); % the length-80 impulse response
omega = 0:pi/1023:pi; % radian frequency vector
W = omega/pi; % Matlab normalized freq vector
Hz_80 = abs(freqz(hn_80,1,omega)); % 1024 samples of |H(e^jw)|
figure(1);
plot(W,HZ_80); grid;
xlabel('\omega /\pi'); ylabel('|H(e^{j\omega})|');
title('Magnitude Response for Length=80');
% Reduce length to 60 and repeat
hn_60 = hn_80(11:70);
Hz_60 = abs(freqz(hn_60,1,omega));
figure(2);
plot(W,HZ_60); grid;
xlabel('\omega /\pi'); ylabel('|H(e^{j\omega})|');
title('Magnitude Response for Length=60');
% Reduce length to 40 and repeat
hn_40 = hn_60(11:50);
Hz_40 = abs(freqz(hn_40,1,omega));
figure(3);
plot(W,HZ_40); grid;
xlabel('\omega /\pi'); ylabel('|H(e^{j\omega})|');
title('Magnitude Response for Length=40');
% Reduce length to 20 and repeat
hn_20 = hn_40(11:30);
Hz_20 = abs(freqz(hn_20,1,omega));
figure(4);
plot(W,HZ_20); grid;
xlabel('\omega /\pi'); ylabel('|H(e^{j\omega})|');
title('Magnitude Response for Length=20');
```

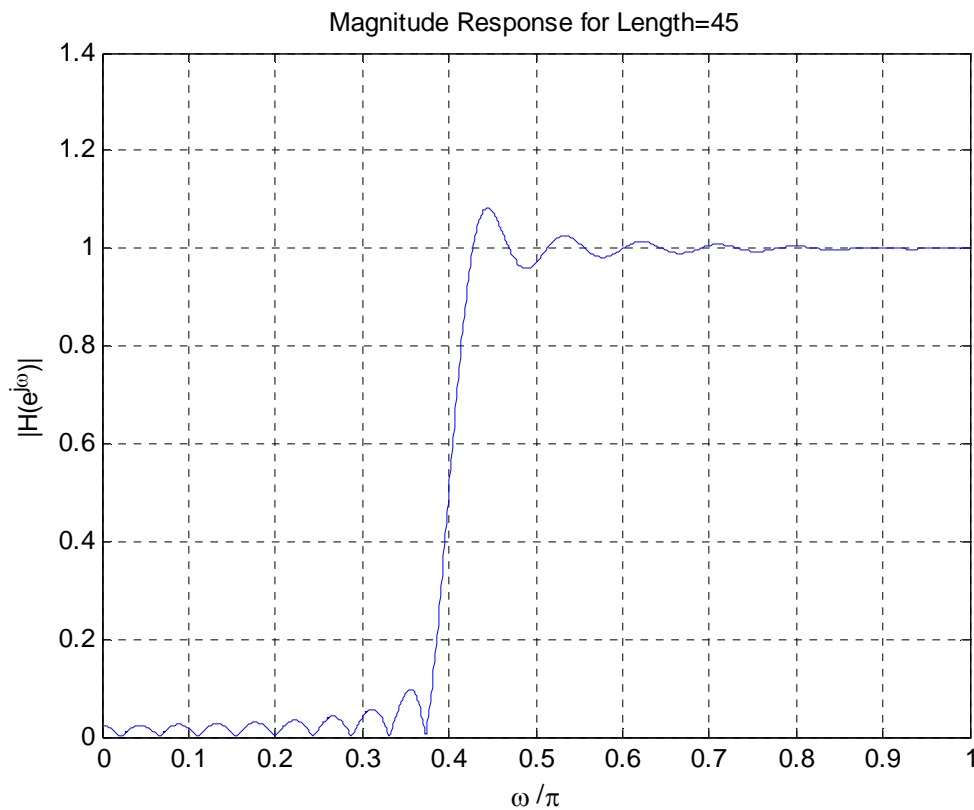
Q7.10 The MATLAB program generating the impulse response, truncated to 45 samples, of a zero-phase ideal highpass filter with a cutoff at $\omega_c = 0.4\pi$ and plotting its magnitude response is given below:


```

% Program Q7_10
% Investigate Gibbs phenomena for a FIR highpass filter.
% The desired impulse response is given by Eq. (7.18) in the
% Lab book.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n = -22:22; % this gives us a length of 45
hn = -0.4 * sinc(0.4*n);
hn(23) = 0.6; % 0.6 = 1 - w_c/pi; see (7.18)
omega = 0:pi/1023:pi; % radian frequency vector
W = omega/pi; % Matlab normalized freq vector
Hz = abs(freqz(hn,1,omega)); % 1024 samples of |H(e^jw)|
figure(1);
plot(W,Hz); grid;
xlabel('\omega / \pi'); ylabel('|H(e^{j\omega})|');
title('Magnitude Response for Length=45');

```

The plot of the magnitude response generated by running this program is as shown below:



From these plots we observe the oscillatory behavior of the magnitude responses due to the Gibbs's phenomenon.

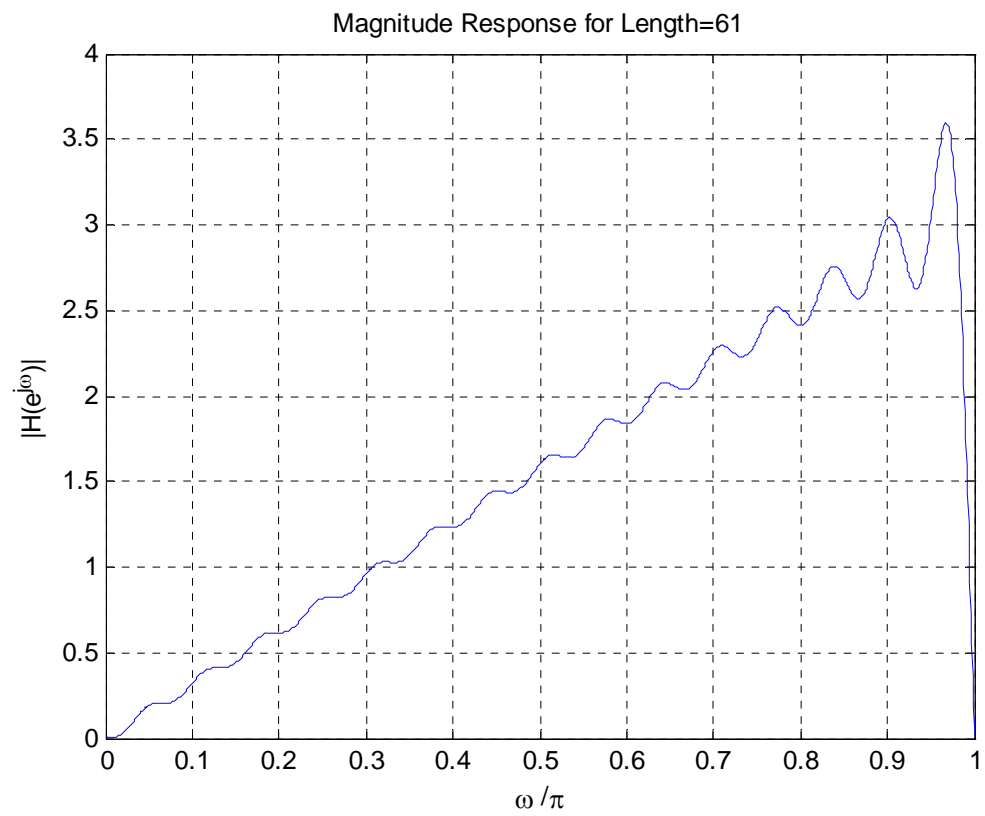
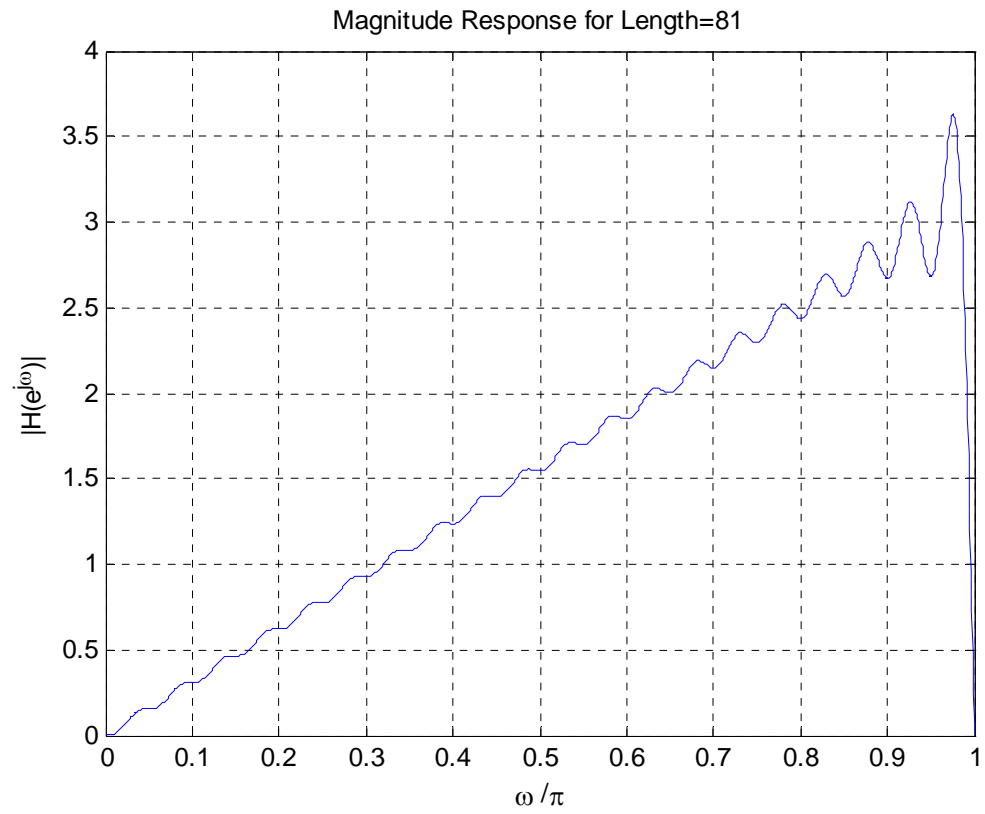
The modified program to generate impulse response coefficients for an even-length filter is given below -

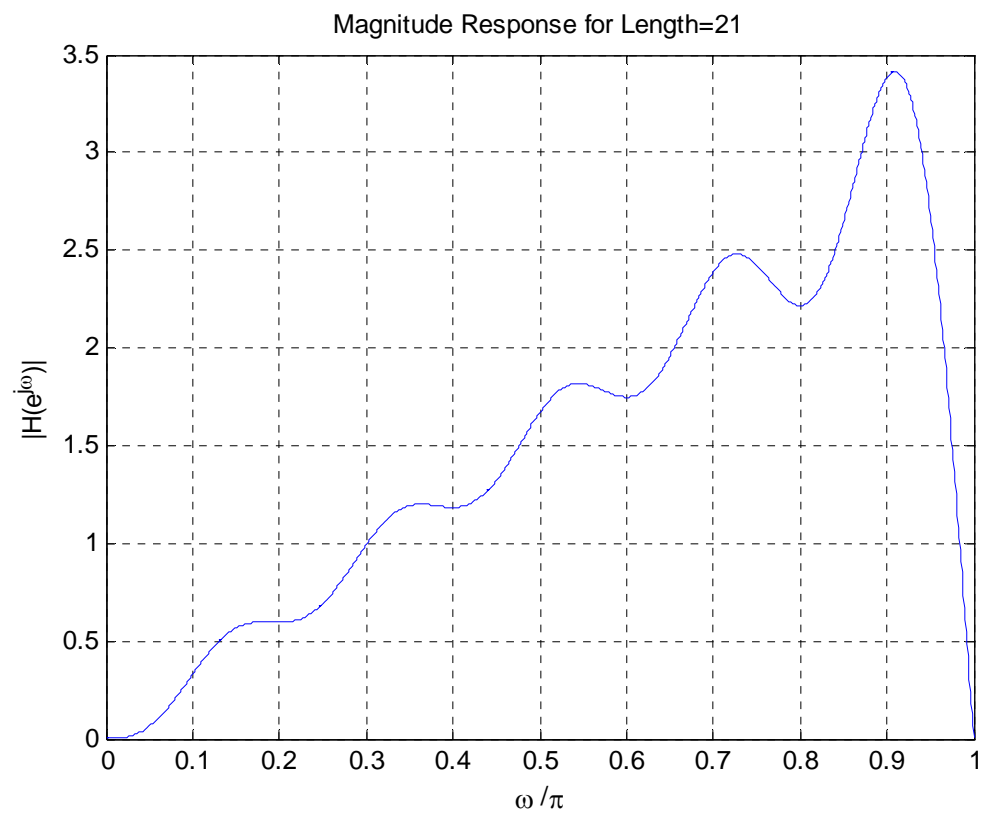
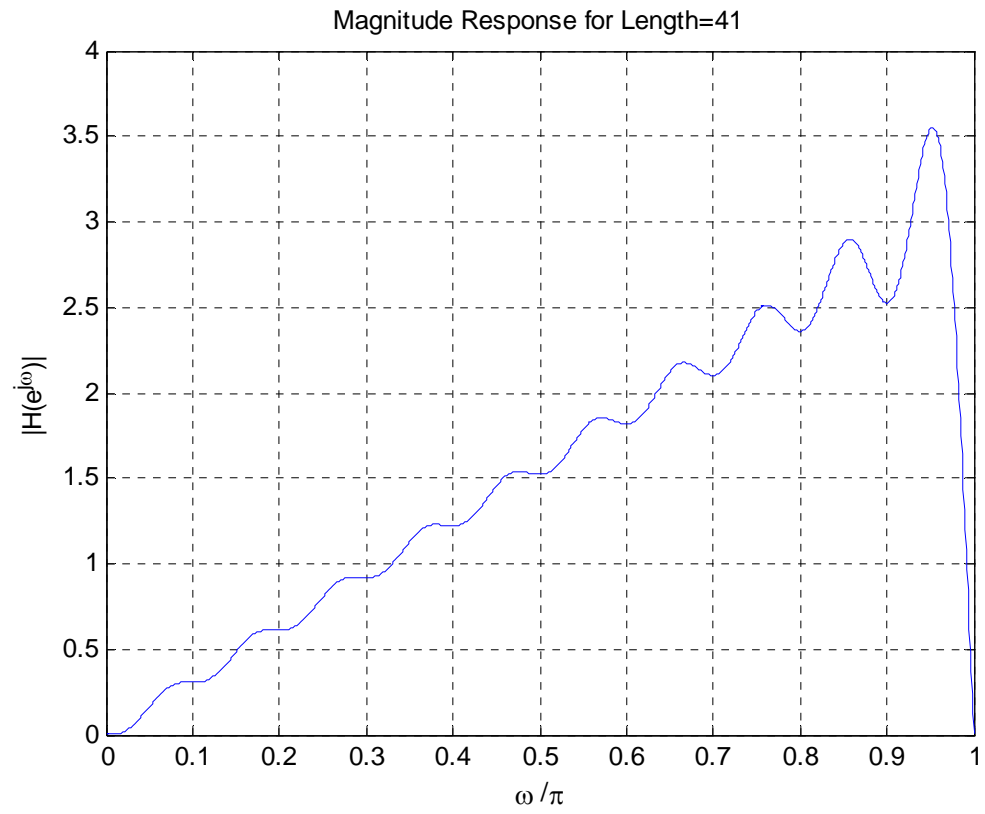
This is a TRICK QUESTION! The answer is: you can't make the length even in this case. Here's why: it's a zero phase FIR filter. That means it's also a linear phase FIR filter, because zero phase is a special subset of linear phase. Now, the desired finite length impulse response has even symmetry about the midpoint of $h[n]$. That makes this a TYPE-II linear phase FIR filter. TYPE-II filters CAN'T BE HIGHPASS, because they have to have a zero at $z = -1$, which is the same as $\omega = \pm\pi$. See pages 13-21 of module 7 of the course notes for more details about this.

Q7.11 The MATLAB program generating the impulse response samples of a zero-phase differentiator of length $2M + 1$ and plotting its magnitude response is given below:

```
% Program Q7_11
% Investigate Gibbs phenomena for FIR differentiator
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M = [40 30 20 10];          % vector of filter half-lengths
omega = 0:pi/1023:pi;       % radian frequency vector
W = omega/pi;               % Matlab normalized freq vector
for TRIAL=1:4,               % loop on lengths
    n = 1:M(TRIAL);
    b = cos(pi*n)./n;
    hn = [-fliplr(b) 0 b];
    Hz = abs(freqz(hn,1,omega)); % 1024 samples of |H(e^jw)|
    figure(TRIAL);
    plot(W,Hz);grid;
    xlabel('\omega /\pi'); ylabel('|H(e^{j\omega})|');
    title(['Magnitude Response for Length=',int2str(2*M(TRIAL)+1)]);
end
```

The program was run for the following different values of length 81, 61, 41, and 21. From the plots generated we observe the oscillatory behavior of the magnitude responses in each case due to the Gibb's phenomenon.





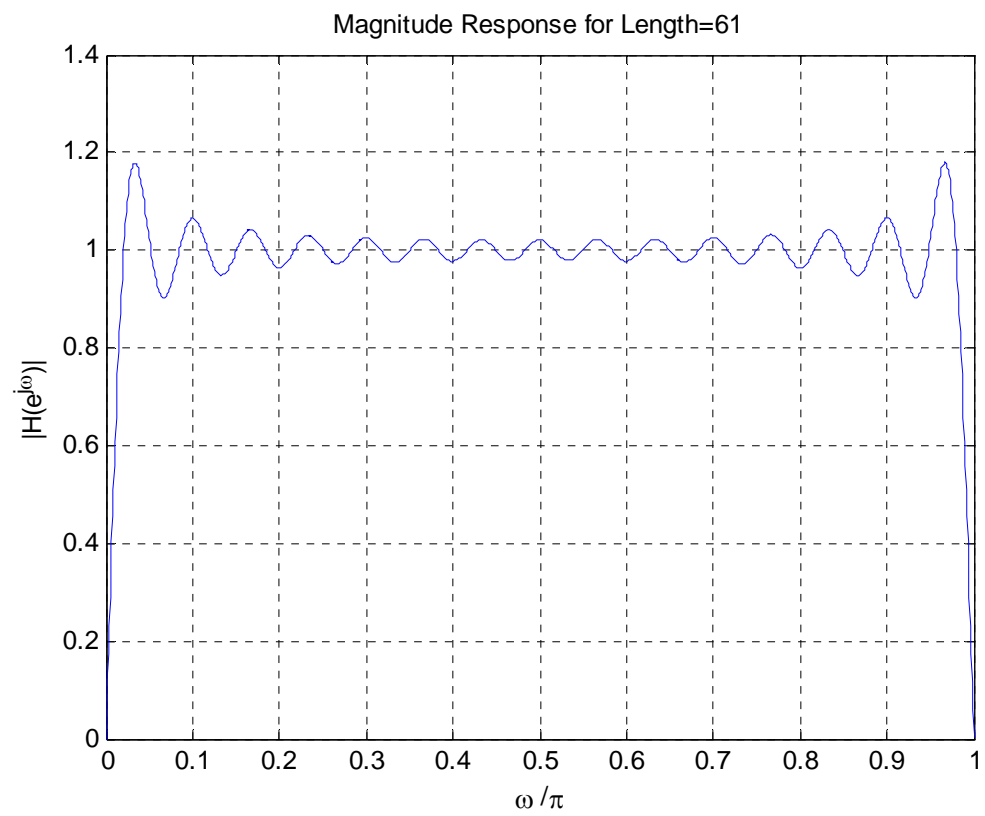
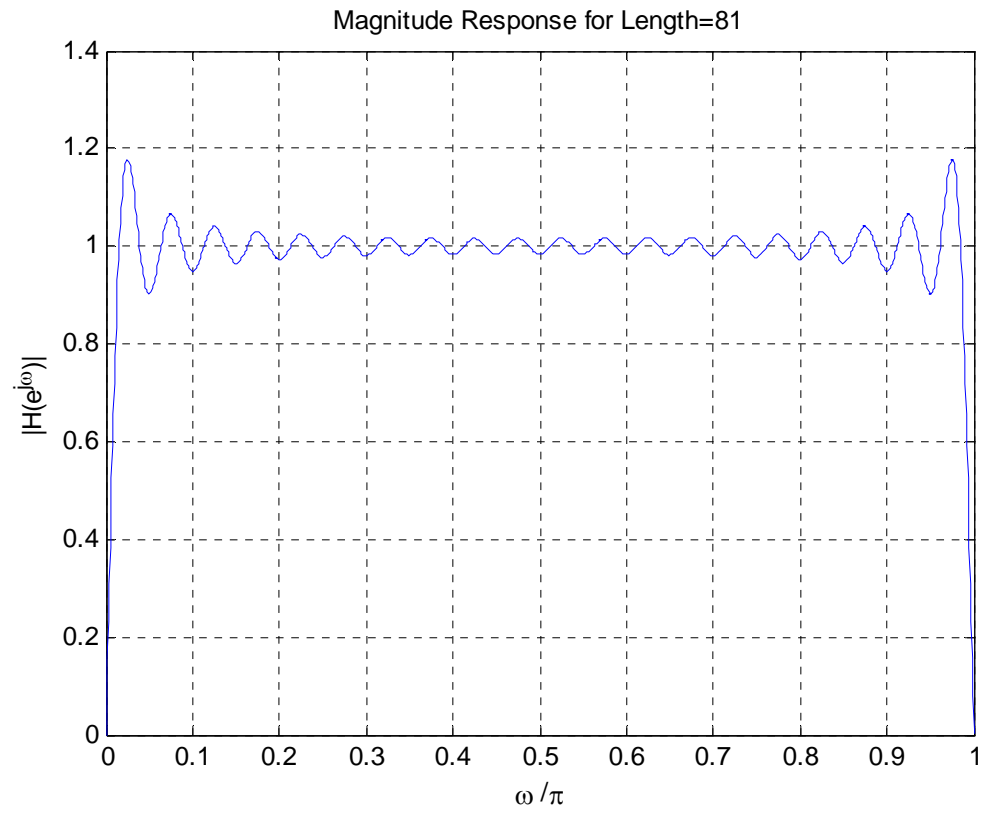
The relation between the number of ripples and the length of the filter is – The number of ripples is decreases in direct proportion as the length is decreased.

The relation between the heights of the largest ripples and the length of the filter is – The peak ripple does not change with length.

Q7.12 The MATLAB program generating the impulse response samples of a zero-phase Hilbert transformer of length $2M + 1$ and plotting its magnitude response is given below

```
% Program Q7_12
% Investigate Gibbs phenomena for FIR Hilbert xformer
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M = [40 30 20 10];           % vector of filter half-lengths
omega = 0:pi/1023:pi;        % radian frequency vector
W = omega/pi;                % Matlab normalized freq vector
for TRIAL=1:4,                % loop on lengths
    n = 1:M(TRIAL);
    c = sin(pi*n*0.5);
    b = 2*(c.*c)./(pi*n);
    hn = [-fliplr(b) 0 b];
    Hz = abs(freqz(hn,1,omega)); % 1024 samples of |H(e^jw)|
    figure(TRIAL);
    plot(W,Hz);grid;
    xlabel('\omega /\pi'); ylabel('|H(e^{j\omega})|');
    title(['Magnitude Response for Length=',int2str(2*M(TRIAL)+1)]);
end
```

The program was run for the following different values of length 81, 61, 41, and 21. From the plots generated we observe the oscillatory behavior of the magnitude responses in each case due to the Gibb's phenomenon.



The relation between the number of ripples and the length of the filter is – The number of ripples decreases in direct proportion to the decrease in the length of the impulse response.

The relation between the heights of the largest ripples and the length of the filter is – The peak ripple is not affected by the length.

Project 7.4 Estimation of FIR Filter Order

Answers:

Q7.13 The estimated order of a linear-phase lowpass FIR filter with the following specifications: $\omega_p = 2$ kHz, $\omega_s = 2.5$ kHz, $\delta_p = 0.005$, $\delta_s = 0.005$, and $F_T = 10$ kHz obtained using `kaiord` is – $N = 46$. The correct call is `kaiord(2000,2500,0.005,0.005,10000)`.

The purpose of the command `ceil` is – To round the estimated order up to the next largest integer; the order has to be integer, so if the formula returns a fraction it needs to be rounded up to the next whole number.

The purpose of the command `nargin` is – To detect if `kaiord` has been called with four arguments or with five. If five, it's assumed that all the frequencies are analog and that the last argument is the sampling frequency. If four, then the sampling frequency defaults to 2, implying that the other frequency arguments are in units of cycles per sample.

- Q7.14** (a) The estimated order of the linear-phase FIR filter with sampling frequency changed to $F_T = 20$ kHz is – $N=91$.
- (b) The estimated order of the linear-phase FIR filter with ripples changed to $\delta_p = 0.002$ and $\delta_s = 0.002$ is – $N=57$.
- (c) The estimated order of the linear-phase FIR filter with stopband edge changed to $\omega_s = 2.3$ kHz is – $N=76$.

From the above results and that obtained in Question Q7.13 we observe that:

The relation between the filter order and sampling frequency is as follows - for a given analog transition bandwidth, an increase in sampling frequency results in a proportional increase in the estimated order, up to rounding up to the next integer. This may be seen clearly from (7.7) on p. 110 of the Lab manual. The analog transition bandwidth in Hz is given by $|F_p - F_s|$. However, the transition bandwidth in the denominator of (7.7) is digital and has units of radians per sample. The relationship between $|F_p - F_s|$ and $\Delta\omega$ is given by

$$\Delta\omega = 2\pi |F_p - F_s| / FT .$$

Thus, increasing FT by a factor of 2 will decrease $\Delta\omega$ by a factor of two, which doubles the order estimated in (7.7) up to the rounding to the next larger integer.

The relation between the filter order and ripples is as follows - The estimated order is approximately proportional to the log (base 10) of the ripples.

The relation between the filter order and the transition band is as follows – To within rounding, the order changes in proportion to the transition bandwidth. Increasing the transition bandwidth by a factor of two tends to halve the order. Similarly, decreasing the transition bandwidth by a factor of two tends to double the order. This may again be understood by the appearance of $\Delta\omega$ in the denominator of (7.7).

Q7.15 The estimated order of a linear-phase lowpass FIR filter with the specifications as given in Question Q7.13 and obtained using `kaiserord` is – $N=54$. The correct call is:

```
kaiserord([2000 2500],[1 0],[0.005 0.005],10000)
```

Comparing the above value of the order with that obtained in Question Q7.13 we observe – the order estimated by `kaiserord` is lower. This is because `kaiserord` uses a different approximation for the order. Specifically, `kaiserord` uses the approximation given in (7.37) on p. 115 of the Lab manual; this approximation is normally associated with a windowed FIR design where the window will specifically be a Kaiser window.

Q7.16 The estimated order of a linear-phase lowpass FIR filter with the specifications as given in Question Q7.13 and obtained using `firpmord` is – $N=47$. The correct call is

```
firpmord([2000 2500],[1 0],[0.005 0.005],10000)
```

Comparing the above value of the order with that obtained in Questions Q7.13 and Q7.15 we observe - In this case, `firpmord` delivers a result that is one larger than `kaiord` and still quite a bit less than `kaiserord`. Again, it should be kept in mind that the approximation used by `kaiord` is quite general; it's intended to be used for any FIR filter design technique. These other formulas used by `kaiserord` and `firpmord` are more specialized. The approximation used by `firpmord` is specifically designed to give a good order estimate for a design by the Parks-McClellan algorithm.

Q7.17 The estimated order of a linear-phase bandpass FIR filter with the following specifications: passband edges at 1.8 and 3.6 kHz, stopband edges at 1.2 and 4.2 kHz, $\delta_p = 0.01$, $\delta_s = 0.02$, and $F_T = 12$ kHz, obtained using `kaiord` is – Here there is yet another discrepancy. In the Lab manual, it states for Q7.17 that $\delta_p = 0.1$. If you use that figure, then your call looks like this:

```
kaiord([1800 3600],[1200 4200],0.1,0.02,12000)
```

and you get $N=20$.

But here in the Lab report file it says $\delta_p = 0.01$. Using instead that figure, your call is

```
kaiord([1800 3600],[1200 4200],0.01,0.02,12000)
```

and you get $N=33$. **Since it's unclear, I'll take either answer.**

Q7.18 The estimated order of a linear-phase bandpass FIR filter with the specifications as given in Question Q7.17 and obtained using `kaiserord` is – Again there is discrepancy. If you use $\delta_p = 0.1$, then your call looks like

```
kaiserord([1200 1800 3600 4200],[0 1 0],[0.02 0.1 0.02],12000)
```

and you get $N=37$. If instead you take $\delta_p = 0.01$, then your call is

```
kaiserord([1200 1800 3600 4200],[0 1 0],[0.02 0.01 0.02],12000)
```

and you get $N=45$. **I'll accept either answer.**

Comparing the above value of the order with that obtained in Question Q7.17 we observe – the orders estimated by `kaiserord` are a lot higher, but probably more accurate if you are going to do a Kaiser window design.

Q7.19 The estimated order of a linear-phase bandpass FIR filter with the specifications as given in Question Q7.17 and obtained using `firpmord` is – if you take $\delta_p = 0.01$, then your call is

```
firpmord([1200 1800 3600 4200],[0 1 0],[0.02 0.1 0.02],12000)
```

and your answer is $N=22$. If instead you take $\delta_p = 0.01$, then your call is

```
firpmord([1200 1800 3600 4200],[0 1 0],[0.02 0.01 0.02],12000)
```

and your answer is $N=35$. **I'll accept either answer.**

Comparing the above value of the order with that obtained in Questions Q7.17 and Q7.18 we observe – The order estimated by `firpmord` is again between the other two, which is probably accurate for a Parks-McClellan design.

Project 7.5 FIR Filter Design

Answers:

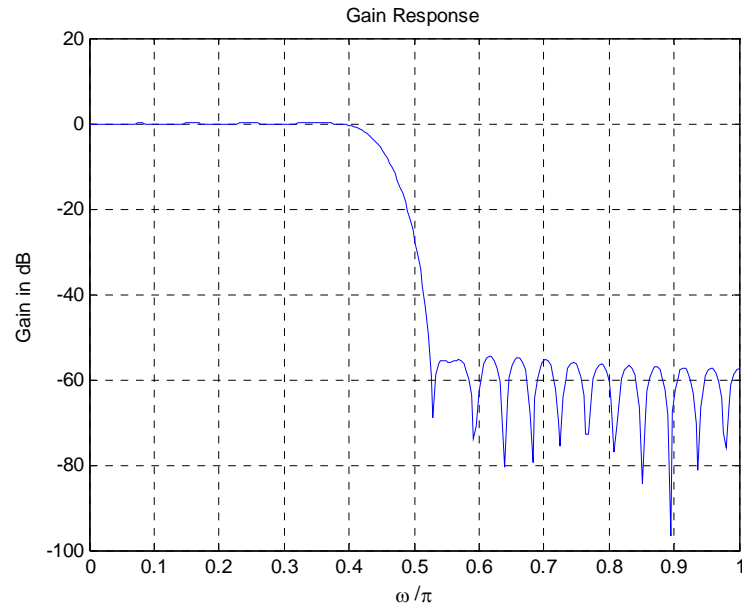
Q7.20 The MATLAB program to design and plot the gain and phase responses of a linear-phase FIR filter using `fir1` is shown below. The filter order is estimated using `kaiord`. The output data are the filter coefficients.

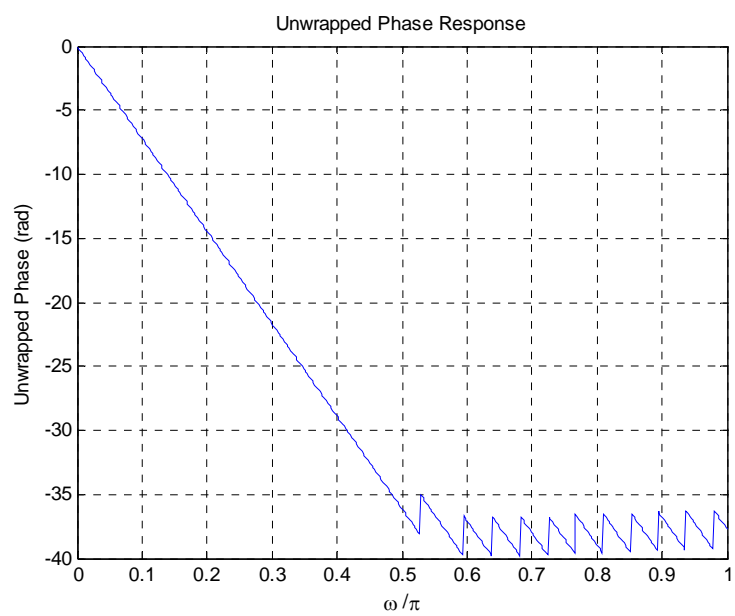
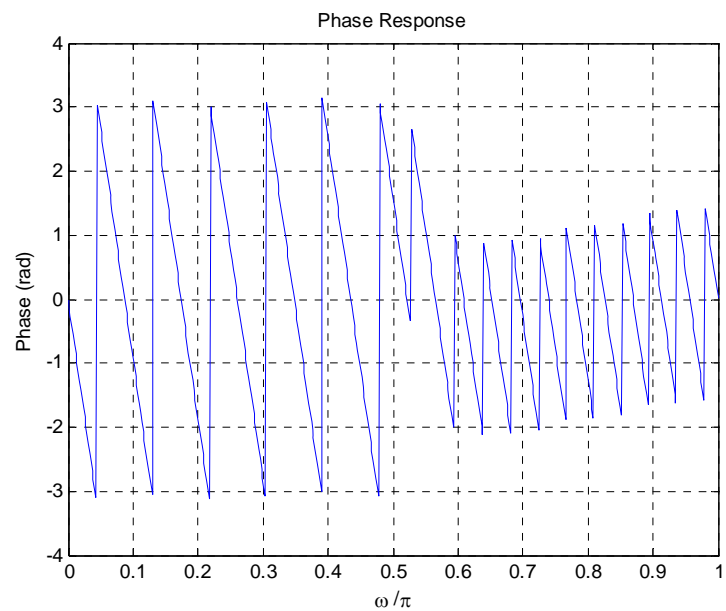
```
% Program Q7_20
% Design a linear phase Lowpass FIR Digital Filter
% meeting the design specification given in Q7.13.
% - Print out the numerator coefficients
%   for the transfer function.
% - Compute and plot the gain function.
% - Compute and plot the phase response.
% - Compute and plot the unwrapped phase response.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;
% Design spec as given in Q7.13.
Fp = 2*10^3;
Fs = 2.5*10^3;
FT = 10*10^3;
Rp = 0.005;
Rs = 0.005;
% Estimate the filter order and print to console
N = kaiord(Fp,Fs,Rp,Rs,FT)
% Design the filter; Hamming window by default
Wp = 2*Fp/FT; % These freqs are normalized: they go
Ws = 2*Fs/FT; % zero to one, not zero to pi.
Wn = Wp + (Ws - Wp)/2;
h = fir1(N,Wn);
% Show the Numerator Coefficients
disp('Numerator Coefficients are ');disp(h);
% Compute and plot the gain response
[g, w] = gain(h,[1]); % same "gain" fcn as in Lab 4
figure(1);
plot(w/pi,g);grid;
%axis([0 1 -60 5]);
xlabel('\omega /\pi'); ylabel('Gain in dB');
title('Gain Response');
% Compute the frequency response
w2 = 0:pi/511:pi;
Hz = freqz(h,[1],w2);
% TEST: did we meet the spec?
MagH = abs(Hz);
T1 = 1.005*ones(1,length(w2));
T2 = 0.995*ones(1,length(w2));
T3 = 0.005*ones(1,length(w2));
figure(4);
plot(w2/pi,MagH,w2/pi,T1,w2/pi,T2,w2/pi,T3);grid;
% Find and plot the phase
figure(2);
Phase = angle(Hz);
plot(w2/pi,Phase);grid;
xlabel('\omega /\pi'); ylabel('Phase (rad)');
title('Phase Response');
figure(3);
UPhase = unwrap(Phase);
plot(w2/pi,UPhase);grid;
xlabel('\omega /\pi'); ylabel('Unwrapped Phase (rad)');
title('Unwrapped Phase Response');
```

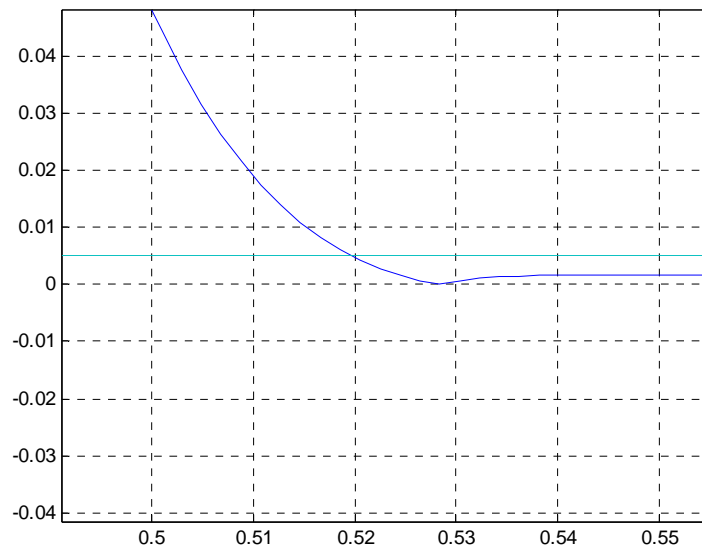
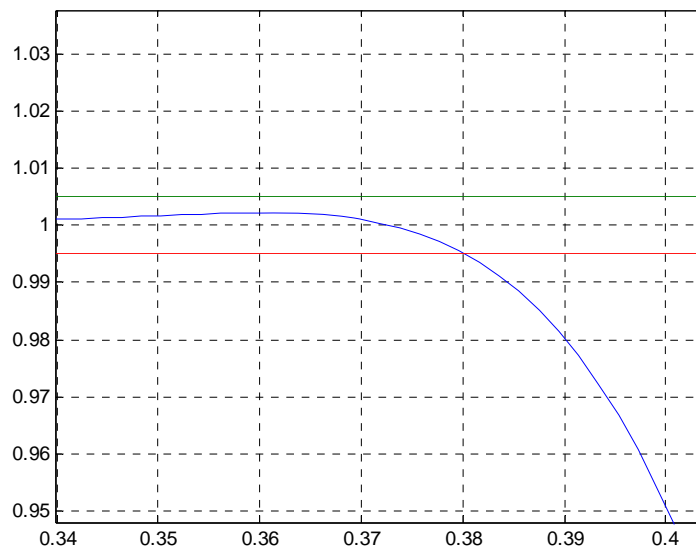
The coefficients of the lowpass filter corresponding to the specifications given in Question 7.20 are as shown below –

0.0010	-0.0004	-0.0015	0.0000	0.0024	0.0010	-0.0038	-0.0032	0.0049
0.0071	-0.0050	-0.0128	0.0026	0.0202	0.0038	-0.0284	-0.0166	0.0366
0.0404	-0.0436	-0.0909	0.0483	0.3129	0.4498	0.3129	0.0483	-0.0909
-0.0436	0.0404	0.0366	-0.0166	-0.0284	0.0038	0.0202	0.0026	-0.0128
-0.0050	0.0071	0.0049	-0.0032	-0.0038	0.0010	0.0024	0.0000	-0.0015
-0.0004	0.0010							

The generated gain and phase responses are given below:



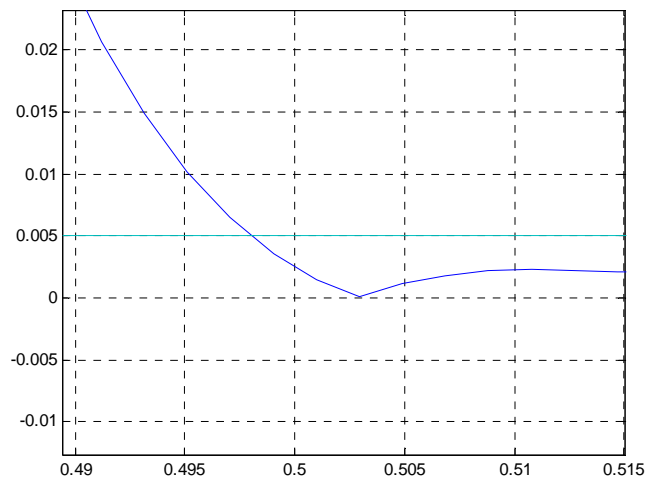
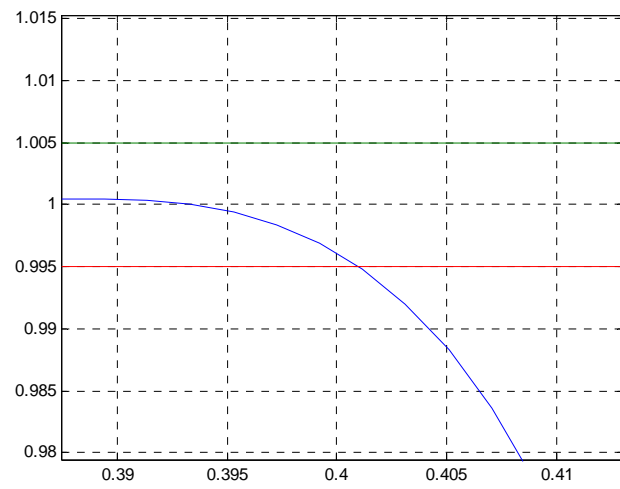
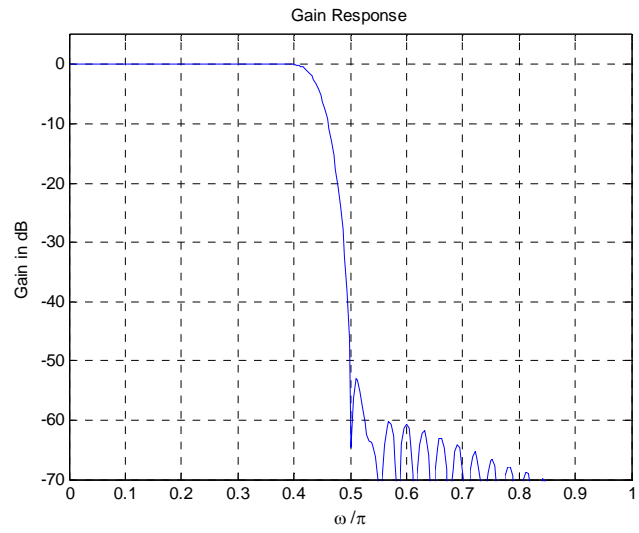


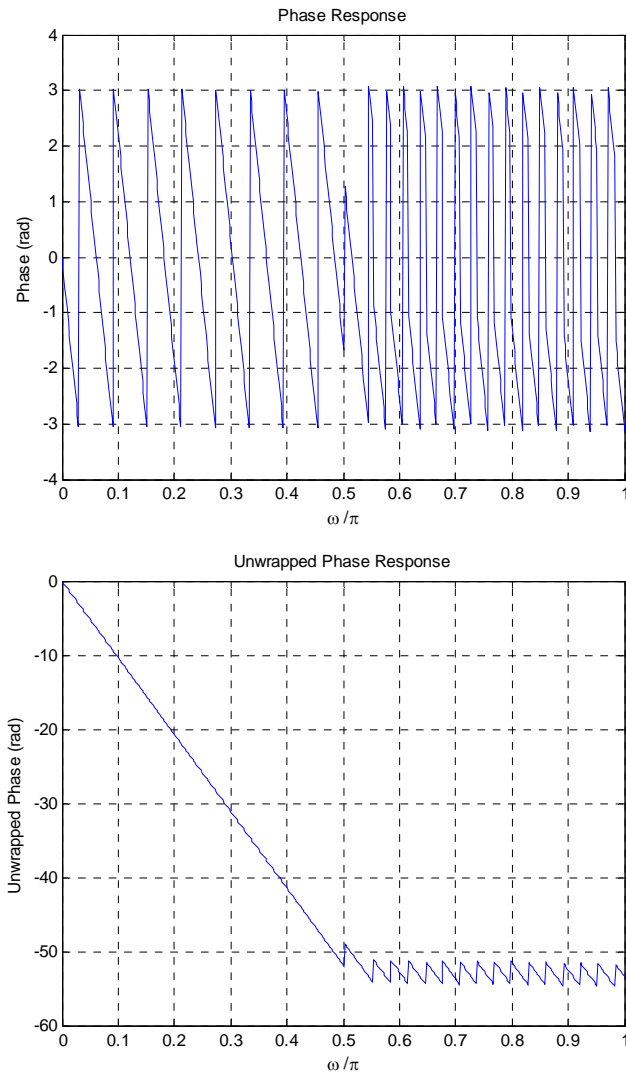


From the gain plot we observe that the filter as designed DOES NOT meet the specifications. AS SHOWN in the two detail plots above, with $N=46$, neither the passband spec at $w_p = 0.4$ (normalized frequency) nor the stopband spec at $w_s = 0.5$ (normalized frequency) are met. So this design DOES NOT meet the spec.

The filter order that meets the specifications is – $N=66$

For the filter that DOES MEET THE SPEC, here are the plots:





Q7.21 The MATLAB program of Question Q7.20 was modified as indicated below for using different windows other than default Hamming window.

```
% Program Q7_21
% Design a linear phase Lowpass FIR Digital Filter
% meeting the design specification given in Q7.13. Use other
% than Hamming window.
% - Print out the numerator coefficients
% for the transfer function.
% - Compute and plot the gain function.
% - Compute and plot the phase response.
% - Compute and plot the unwrapped phase response.
```

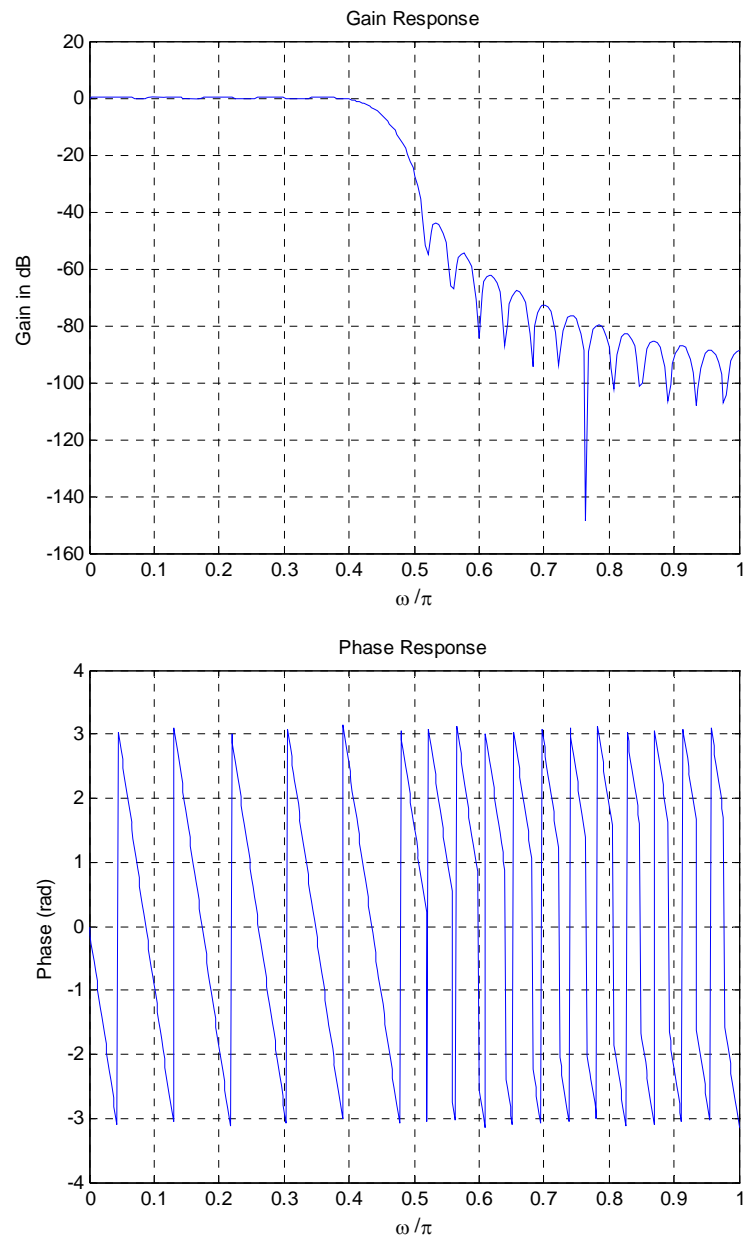


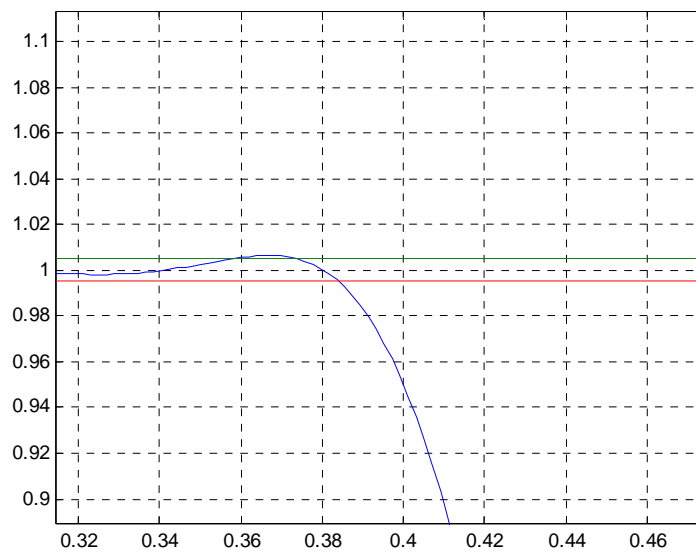
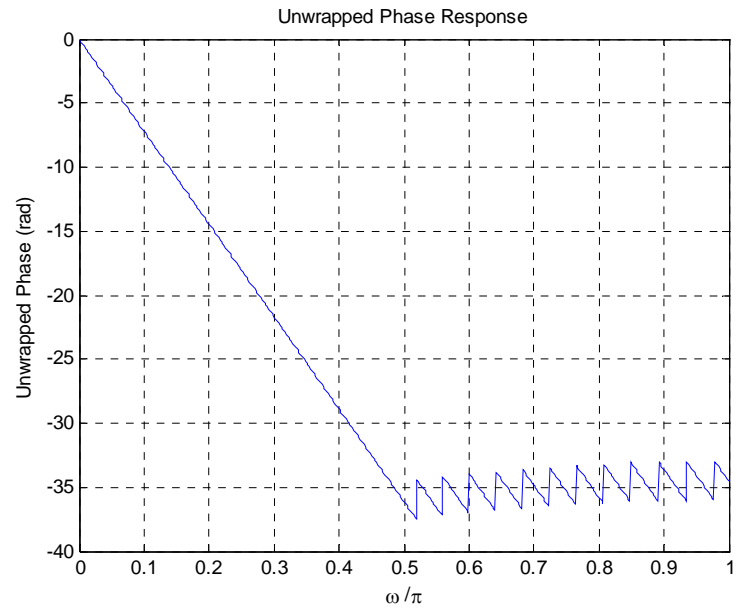
```

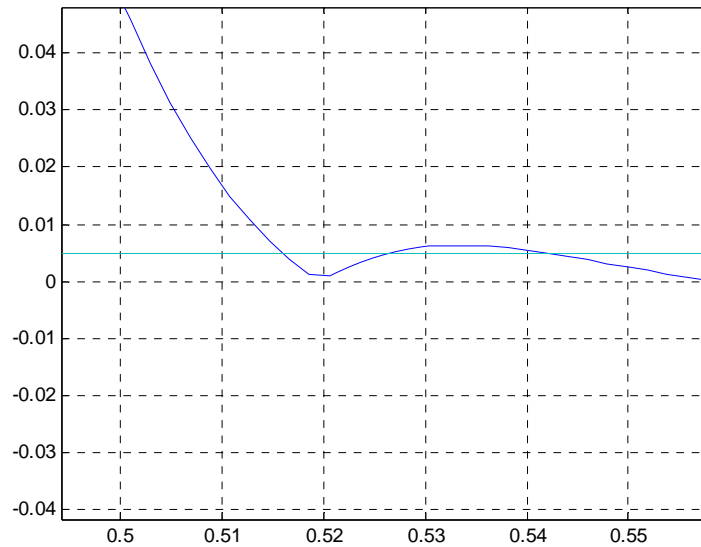
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;
% Design spec as given in Q7.13.
Fp = 2*10^3;
Fs = 2.5*10^3;
FT = 10*10^3;
Rp = 0.005;
Rs = 0.005;
% Estimate the filter order and print to console
N = kaiserord(Fp,Fs,Rp,Rs,FT);
N = N+13
% Design the filter
Wp = 2*Fp/FT; % These freqs are normalized: they go
Ws = 2*Fs/FT; % zero to one, not zero to pi.
Wn = Wp + (Ws - Wp)/2;
%h = fir1(N,Wn); % Default Hamming window
%Wdw = hann(N+3);Wdw=Wdw(2:N+2); % Hann; see footnote on p. 563 of text.
%Wdw = blackman(N+3);Wdw=Wdw(2:N+2); % Blackman; see text p. 563.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Dolph-Chebyshev Section %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Wdw = chebwin(N+1,38);
figure(5);
plot([0:1/511:1],abs(freqz(Wdw,[1],0:pi/511:pi)));grid; % plot window
xlabel('\omega /\pi');ylabel('|Window|');title('Window');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% end Dolph Cheby window %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
h = fir1(N,Wn,Wdw);
% Show the Numerator Coefficients
disp('Numerator Coefficients are ');disp(h);
% Compute and plot the gain response
[g, w] = gain(h,[1]);
figure(1);
plot(w/pi,g);grid;
%axis([0 1 -60 5]);
xlabel('\omega /\pi'); ylabel('Gain in dB');
title('Gain Response');
% Compute the frequency response
w2 = 0:pi/511:pi;
Hz = freqz(h,[1],w2);
% TEST: did we meet the spec?
MagH = abs(Hz);
T1 = 1.005*ones(1,length(w2));
T2 = 0.995*ones(1,length(w2));
T3 = 0.005*ones(1,length(w2));
figure(4);
plot(w2/pi,MagH,w2/pi,T1,w2/pi,T2,w2/pi,T3);grid;
% Find and plot the phase
figure(2);
Phase = angle(Hz);
plot(w2/pi,Phase);grid;
xlabel('\omega /\pi'); ylabel('Phase (rad)');
title('Phase Response');
figure(3);
UPhase = unwrap(Phase);
plot(w2/pi,UPhase);grid;
xlabel('\omega /\pi'); ylabel('Unwrapped Phase (rad)');
title('Unwrapped Phase Response');

```

(a) Use of Hanning window – The generated gain and phase responses are given below:



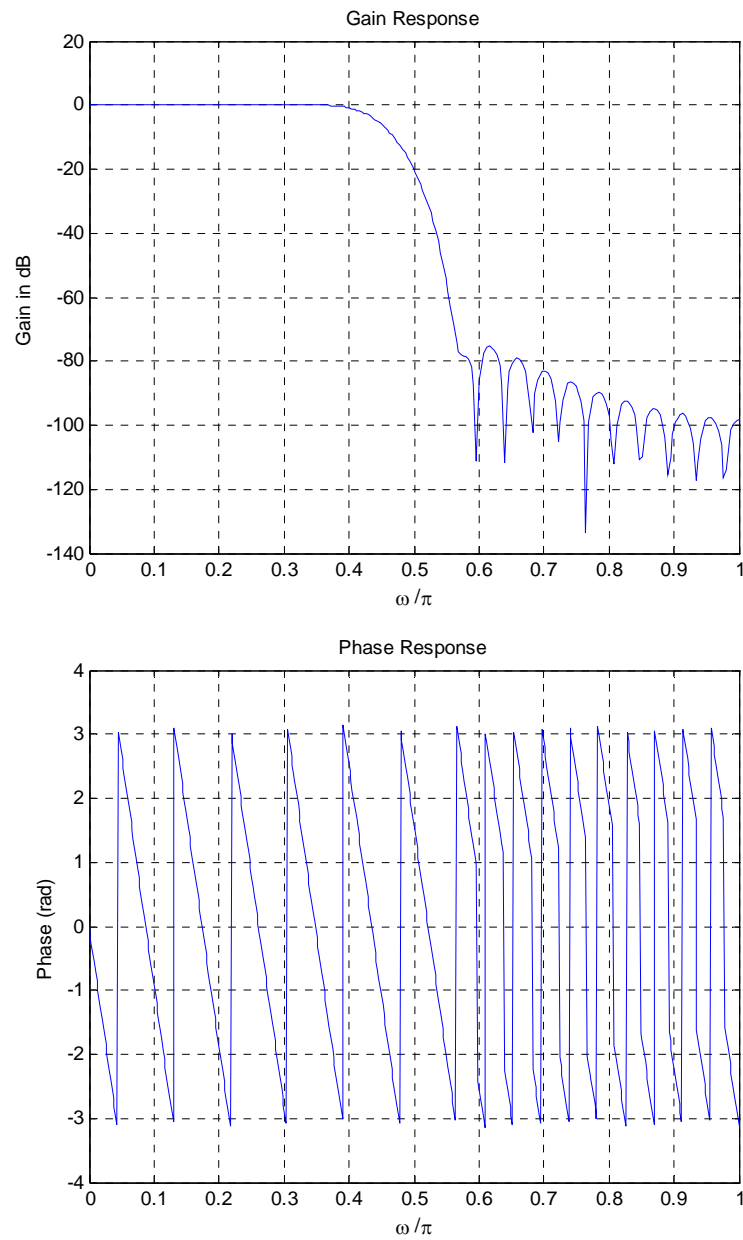


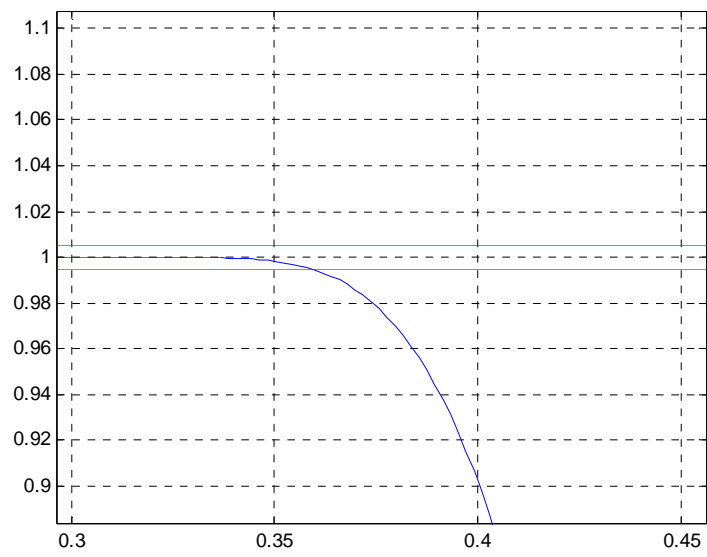
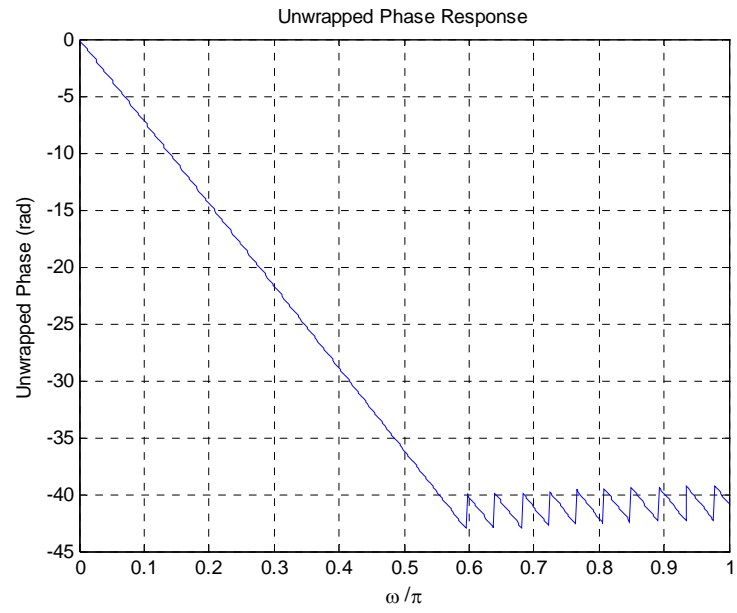


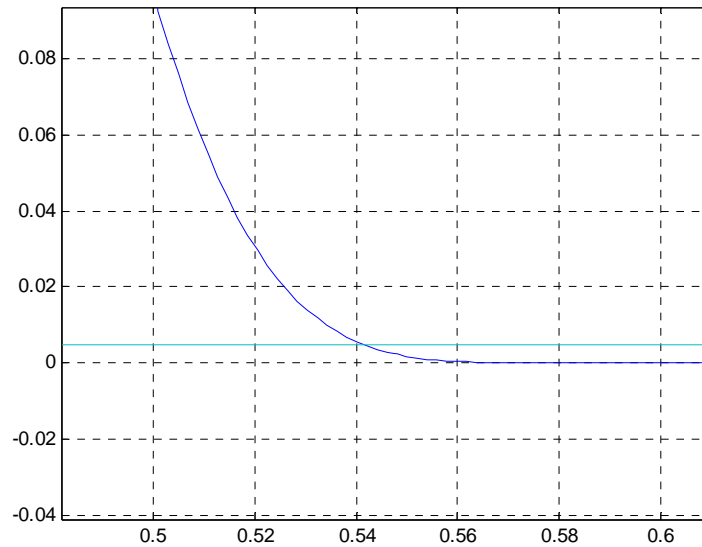
From the gain plot we observe that the filter as designed DOES NOT meet the specifications. AS SHOWN in the two detail plots above, with $N=46$, neither the passband spec at $w_p = 0.4$ (normalized frequency) nor the stopband spec at $w_s = 0.5$ (normalized frequency) are met. So this design DOES NOT meet the spec.

The filter order that meets the specifications is – ANOTHER TRICK QUESTION!! The Hann window CANNOT MEET THIS SPEC. Here's why. The minimum stopband attenuation is given by $\alpha_s = -20\log \delta_p = 46.0206$ dB. However, if you look in Table 10.2 on page 535 of the text, you will see that the Hann window is capable of providing a stopband attenuation of only 43.9 dB.... e.g., not enough!

(b) Use of Blackman window – The generated gain and phase responses are given below:





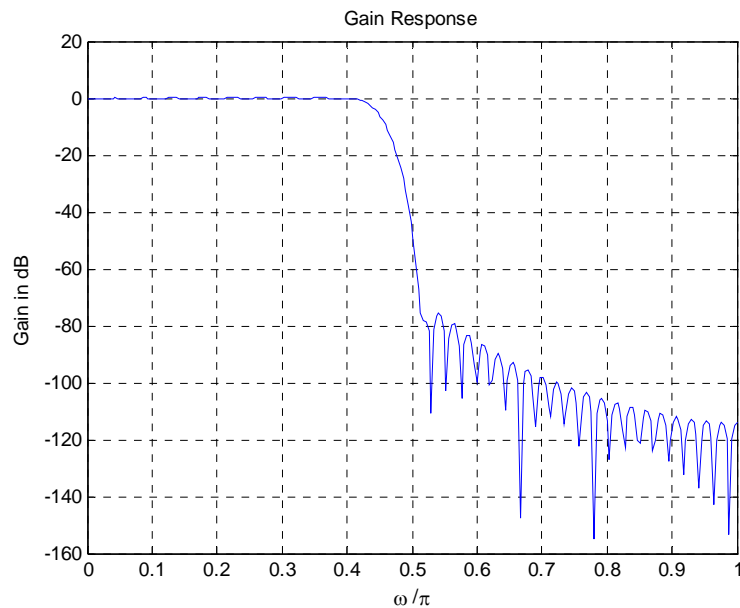


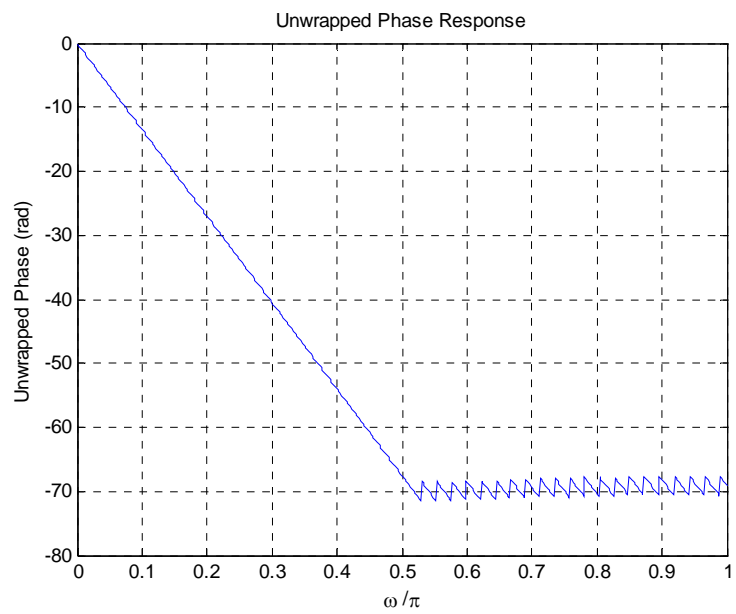
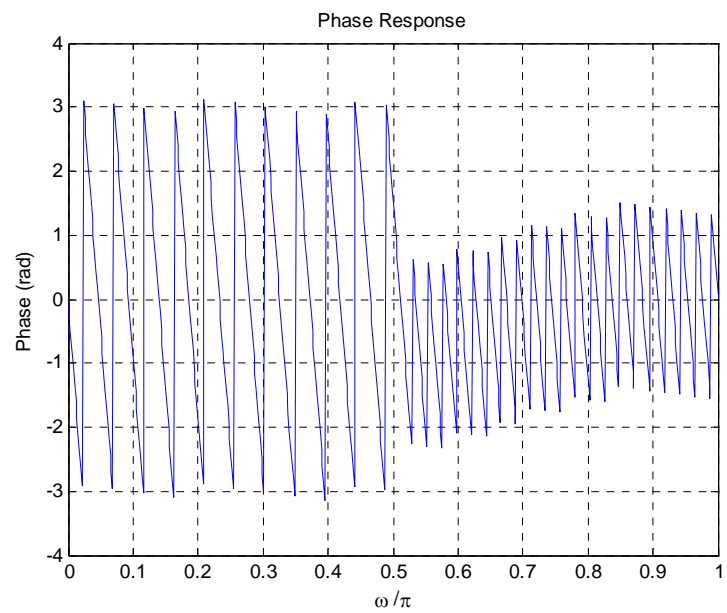
From the gain plot we observe that the filter as designed DOES NOT meet the specifications.

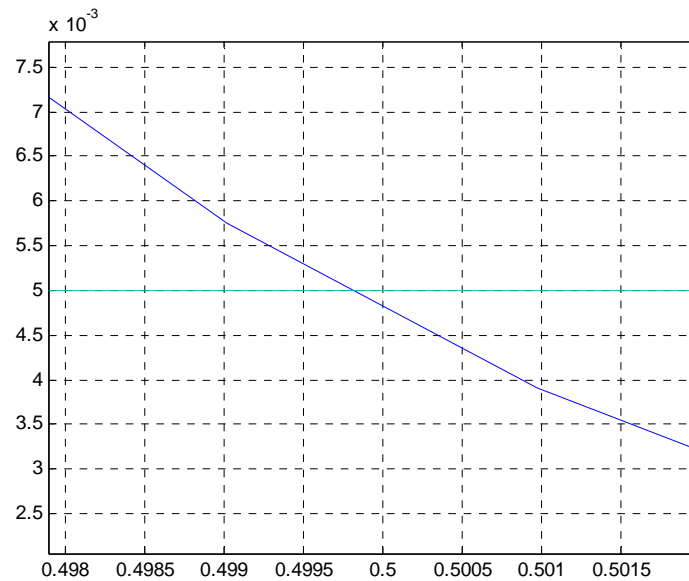
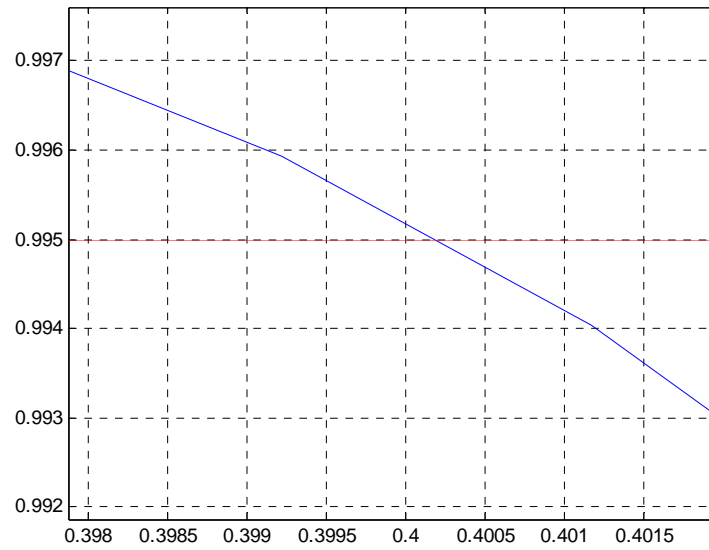
AS SHOWN in the two detail plots above, with $N=46$, neither the passband spec at $\omega_p = 0.4$ (normalized frequency) nor the stopband spec at $\omega_s = 0.5$ (normalized frequency) are met. So this design DOES NOT meet the spec.

The filter order that meets the specifications is – $N=86$

For the filter that DOES MEET THE SPEC, here are the plots:



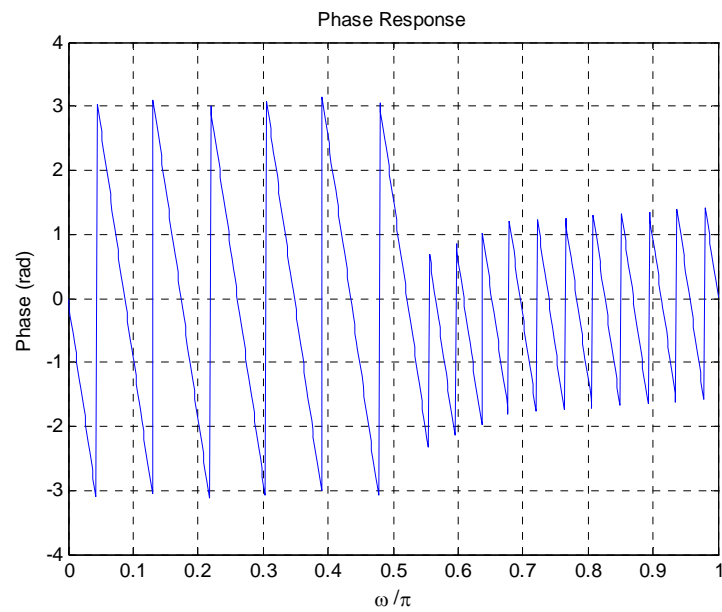
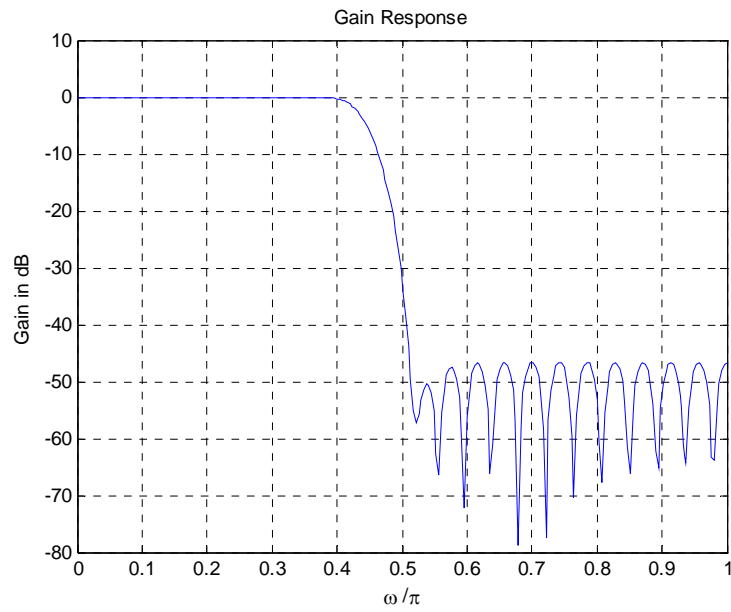


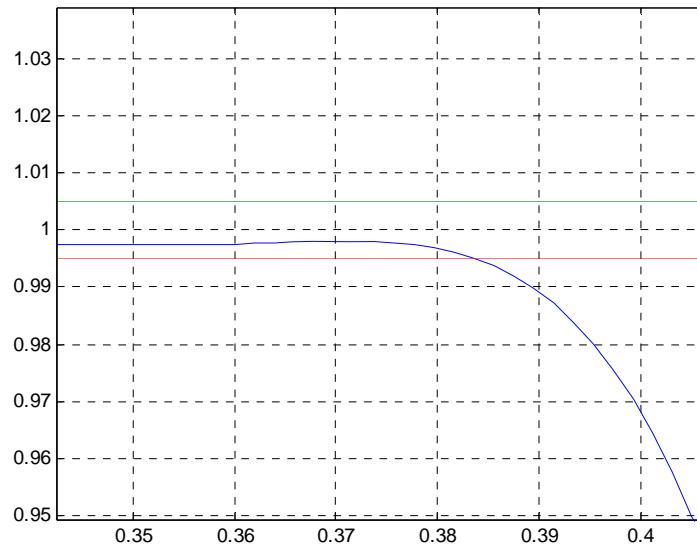
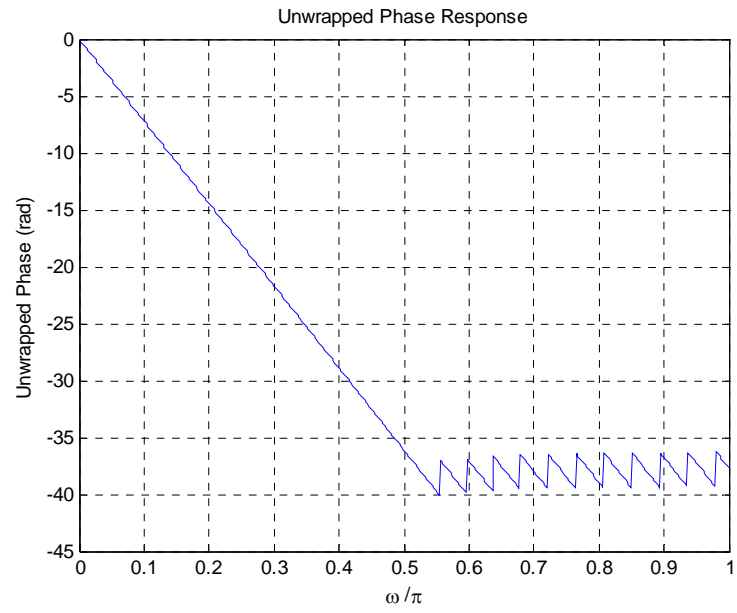


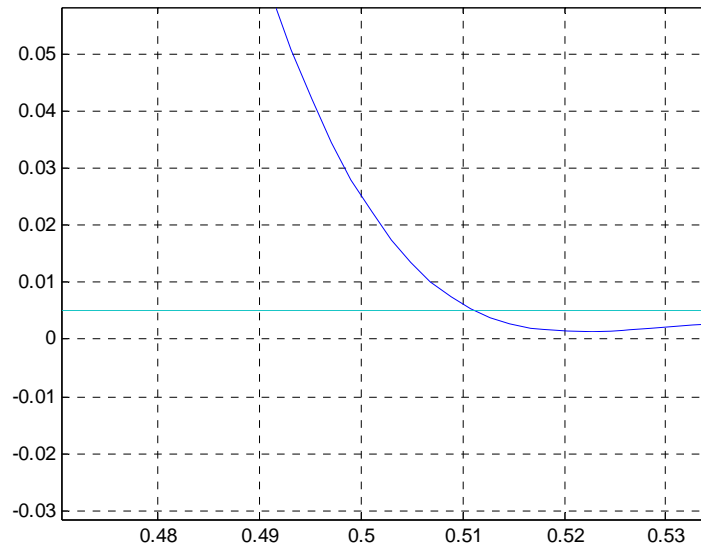
(c) Use of Dolph-Chebyshev window – The generated gain and phase responses are given below:

I have empirically adjusted the “chebwin” sidelobe attenuation parameter R until the passband and stopband ripple just meet the spec. This resulted in $R = 38$ dB.

With this value R, I then call chebwin and run the program with the order N=46 designed by kaiord.





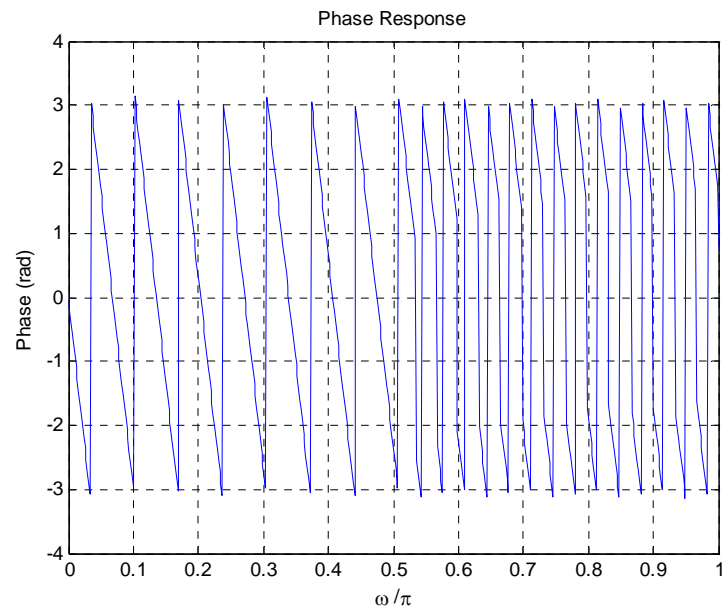
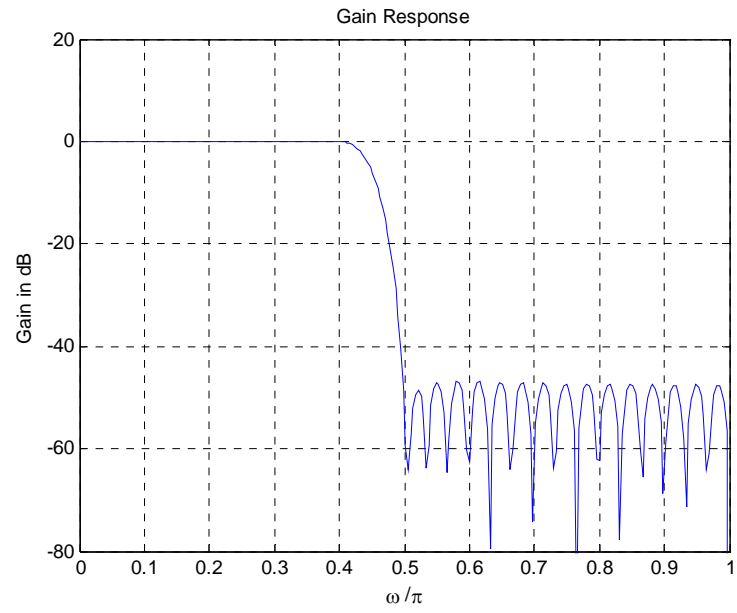


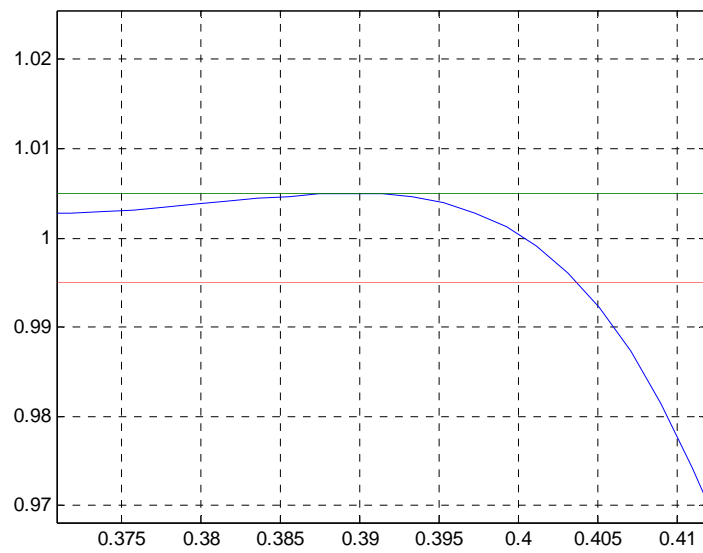
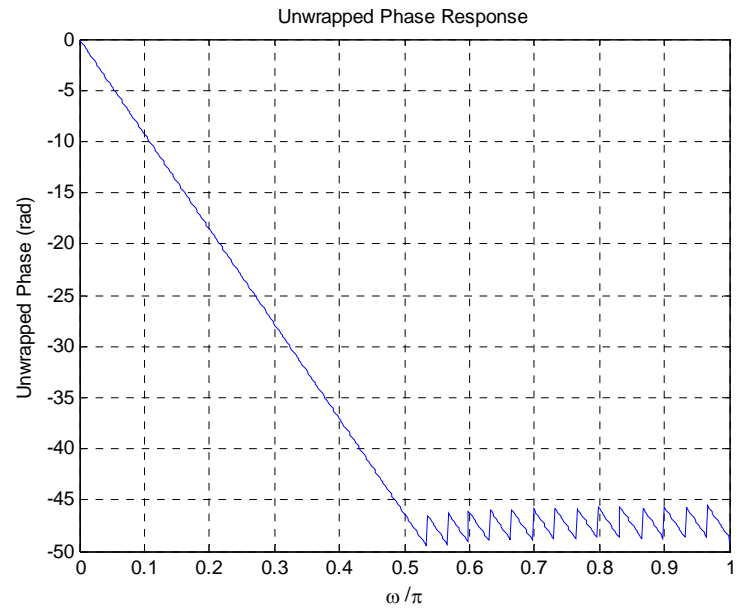
The parameters γ and β used in the design are – We have $-20\log \gamma = 38$, which implies that $\gamma = 10^{-38/20} = 12.59 \times 10^{-3}$. Thus, we have also from (7.32) on p. 115 of the Lab manual with $2M=46$: $\beta = 1.00608$.

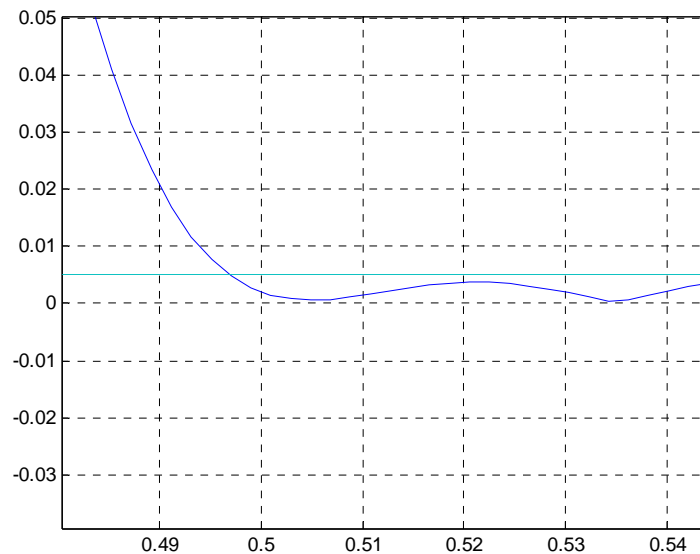
From the gain plot we observe that the filter as designed DOES NOT meet the specifications. AS SHOWN in the two detail plots above, with $N=46$, neither the passband spec at $w_p = 0.4$ (normalized frequency) nor the stopband spec at $w_s = 0.5$ (normalized frequency) are met. So this design DOES NOT meet the spec.

The filter order that meets the specifications is – $N=61$.

For the filter that DOES MEET THE SPEC, here are the plots:







Q7.22 The MATLAB program to design and plot the gain and phase responses of a linear-phase FIR filter using `firpm` is shown below. The filter order is estimated using `kaiord`. The output data are the filter coefficients.

```
% Program Q7_22
% Use Parks-McClellan to design a linear phase Lowpass
% FIR Digital Filter meeting the design specification given
% in Q7.13.
% - Print out the numerator coefficients
%   for the transfer function.
% - Compute and plot the gain function.
% - Compute and plot the phase response.
% - Compute and plot the unwrapped phase response.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;
% Design spec as given in Q7.13.
Fp = 2*10^3;
Fs = 2.5*10^3;
FT = 10*10^3;
Rp = 0.005;
Rs = 0.005;
% Estimate the filter order and print to console
N = kaiord(Fp,Fs,Rp,Rs,FT)
% Design the filter using Parks-McClellan
Wp = 2*Fp/FT;    % These freqs are normalized: they go
Ws = 2*Fs/FT;    % zero to one, not zero to pi.
F = [0 Wp Ws 1];
A = [1 1 0 0];
h = firpm(N,F,A);
```

```

% Show the Numerator Coefficients
disp('Numerator Coefficients are ');disp(h);
% Compute and plot the gain response
[g, w] = gain(h,[1]);
figure(1);
plot(w/pi,g);grid;
%axis([0 1 -60 5]);
xlabel('\omega /\pi'); ylabel('Gain in dB');
title('Gain Response');
% Compute the frequency response
w2 = 0:pi/511:pi;
Hz = freqz(h,[1],w2);
% TEST: did we meet the spec?
MagH = abs(Hz);
T1 = 1.005*ones(1,length(w2));
T2 = 0.995*ones(1,length(w2));
T3 = 0.005*ones(1,length(w2));
figure(4);
plot(w2/pi,MagH,w2/pi,T1,w2/pi,T2,w2/pi,T3);grid;
% Find and plot the phase
figure(2);
Phase = angle(Hz);
plot(w2/pi,Phase);grid;
xlabel('\omega /\pi'); ylabel('Phase (rad)');
title('Phase Response');
figure(3);
UPhase = unwrap(Phase);
plot(w2/pi,UPhase);grid;
xlabel('\omega /\pi'); ylabel('Unwrapped Phase (rad)');
title('Unwrapped Phase Response');

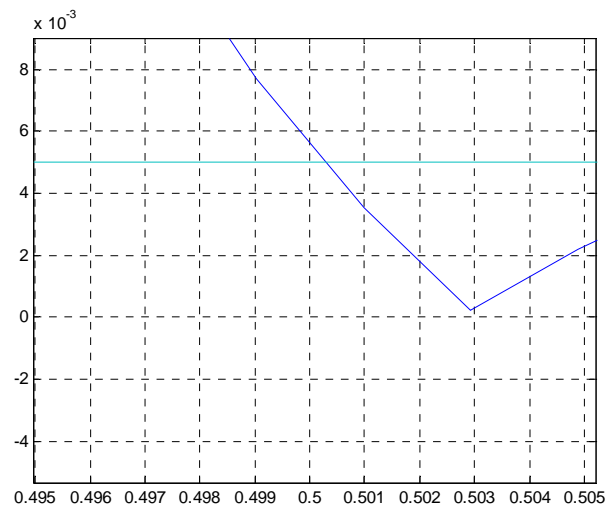
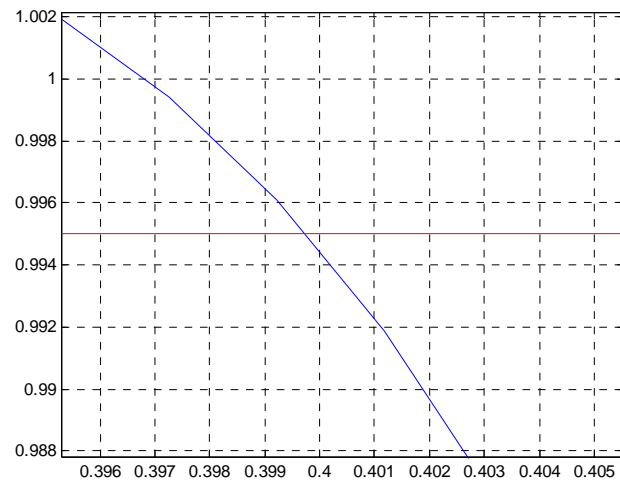
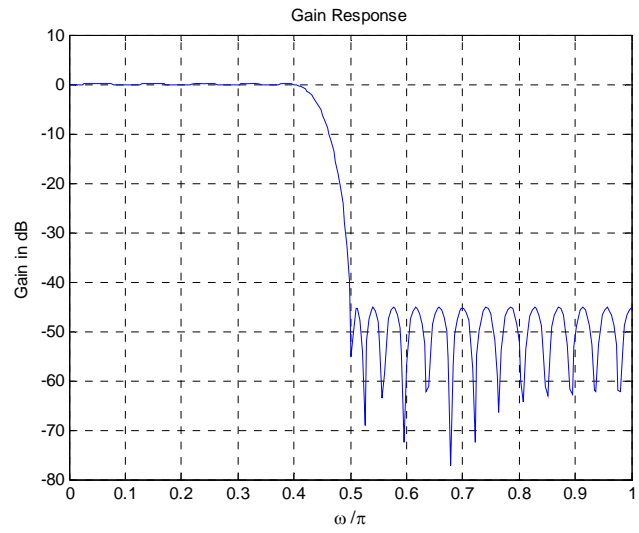
```

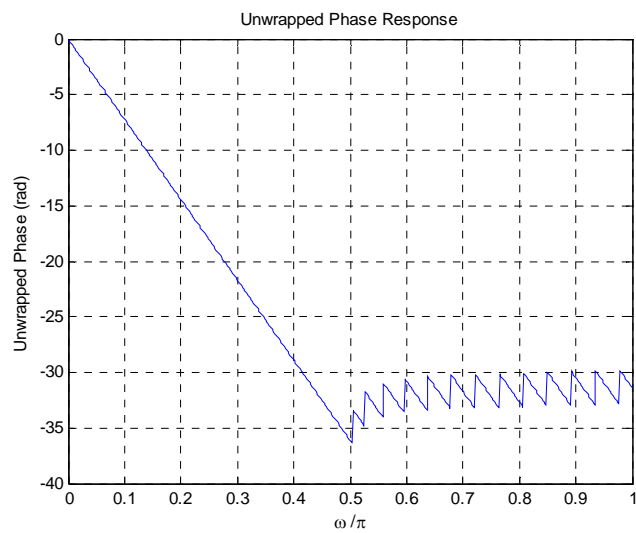
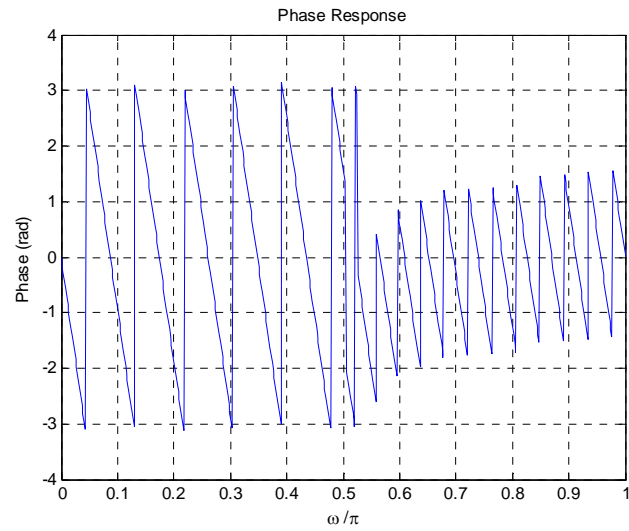
The coefficients of the lowpass filter corresponding to the specifications given in Question 7.20

are as shown below –

0.0028	-0.0022	-0.0046	-0.0006	0.0053	0.0019	-0.0073	-0.0058	0.0079
0.0106	-0.0069	-0.0170	0.0032	0.0243	0.0045	-0.0319	-0.0182	0.0390
0.0422	-0.0448	-0.0924	0.0486	0.3136	0.4501	0.3136	0.0486	-0.0924
-0.0448	0.0422	0.0390	-0.0182	-0.0319	0.0045	0.0243	0.0032	-0.0170
-0.0069	0.0106	0.0079	-0.0058	-0.0073	0.0019	0.0053	-0.0006	-0.0046
-0.0022	0.0028							

The generated gain and phase responses are given below:

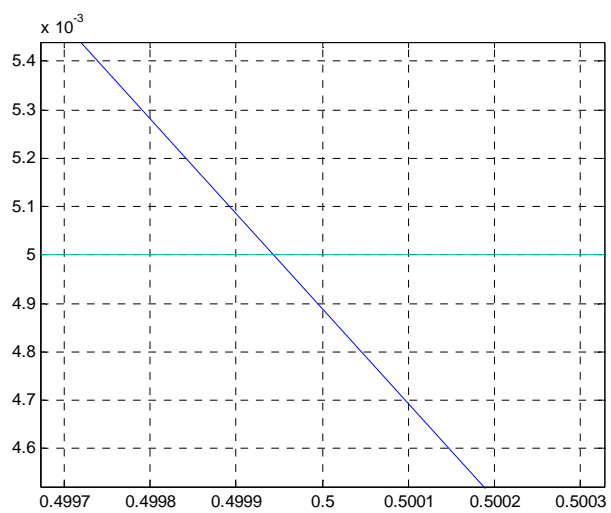
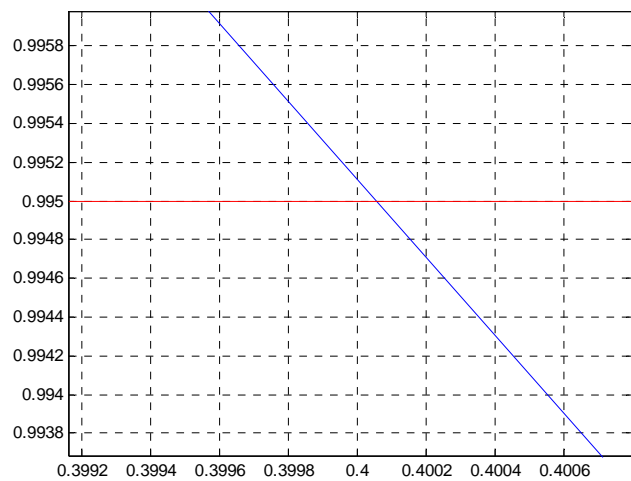
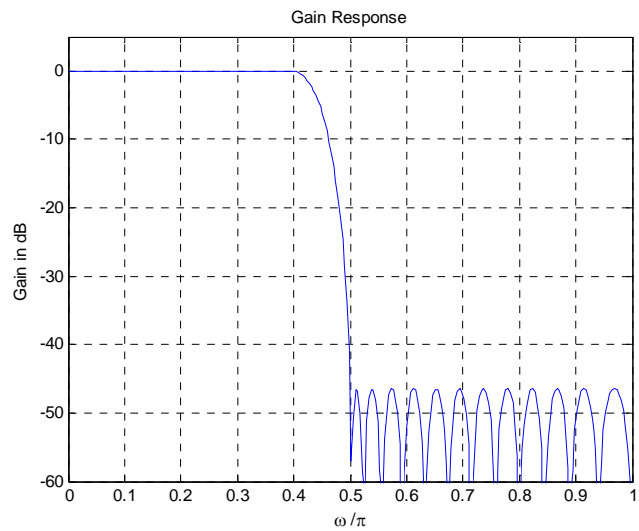


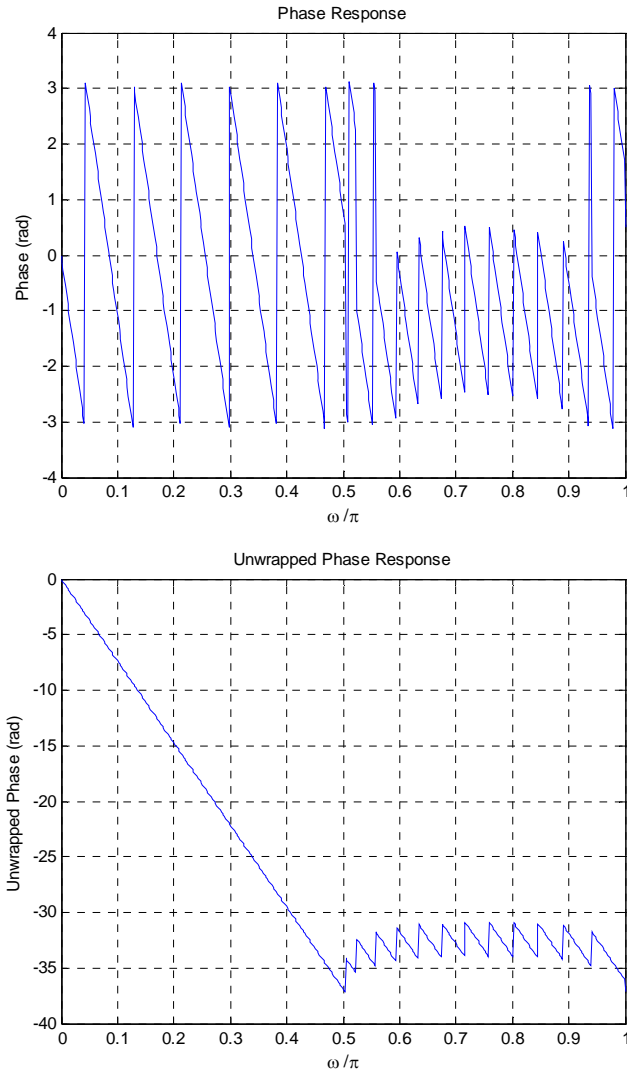


From the gain plot we observe that the filter as designed DOES NOT meet the specifications.

The filter order that meets the specifications is - $N=47$.

For the filter with $N=47$ that DID MEET THE SPEC, the plots are shown below:





Q7.23 The MATLAB program to design and plot the gain and phase responses of a linear-phase FIR filter using `fir1` and `kaiser` is shown below. The filter order N is estimated using Eq. (7.37) and the parameter β is computed using Eq. (7.36). The output data are the filter coefficients.

```
% Program Q7_23
% Use Kaiser window to design a linear phase Lowpass
% FIR Digital Filter meeting the design specification given
% in Q7.23.
%
% It is clear from the statement of the question that Mitra
% wants us to use (7.36) and (7.37) for this problem. That
% isn't the greatest thing to try because kaiserord already does
% exactly what we need.... but that's Q7_24! So here goes!
% - Print out the numerator coefficients
% for the transfer function.
```

```

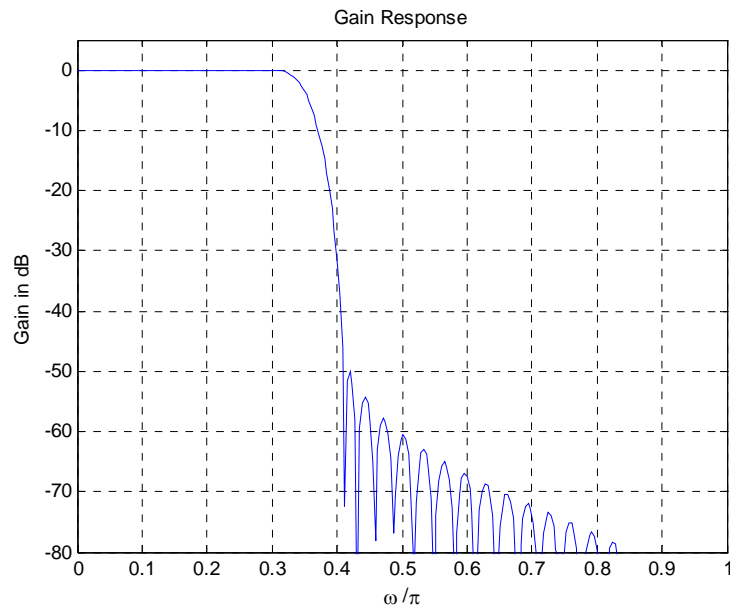
% - Compute and plot the gain function.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;
% Design spec as given in Q7.23.
Wp = 0.31;
Ws = 0.41;
Wn = Wp + (Ws-Wp)/2;
As = 50;
Ds = 10^(-As/20);
Dp = Ds;           %Kaiser window design has equal ripple in
                   % passband and stopband.
% estimate order using (7.37)
if As > 21
    N = ceil((As-7.95)*2/(14.36*(abs(Wp-Ws)))+1)
else
    N = ceil(0.9222*2/abs(Wp-Ws)+1)
end
% Use (7.36) to get Beta
if As > 50
    BTA = 0.1102*(As-8.7);
elseif As >= 21
    BTA = 0.5842*(As-21)^0.4+0.07886*(As-21);
else
    BTA = 0;
end
Win = kaiser(N+1,BTA);
h = fir1(N,Wn,Win);
% Show the Numerator Coefficients
disp('Numerator Coefficients are ');disp(h);
% Compute and plot the gain response
[g, w] = gain(h,[1]);
figure(1);
plot(w/pi,g);grid;
axis([0 1 -80 5]);
xlabel('\omega /\pi'); ylabel('Gain in dB');
title('Gain Response');
% Compute the frequency response
w2 = 0:pi/511:pi;
Hz = freqz(h,[1],w2);
% Find and plot the phase
figure(2);
Phase = angle(Hz);
plot(w2/pi,Phase);grid;
xlabel('\omega /\pi'); ylabel('Phase (rad)');
title('Phase Response');
figure(3);
UPhase = unwrap(Phase);
plot(w2/pi,UPhase);grid;
xlabel('\omega /\pi'); ylabel('Unwrapped Phase (rad)');
title('Unwrapped Phase Response');

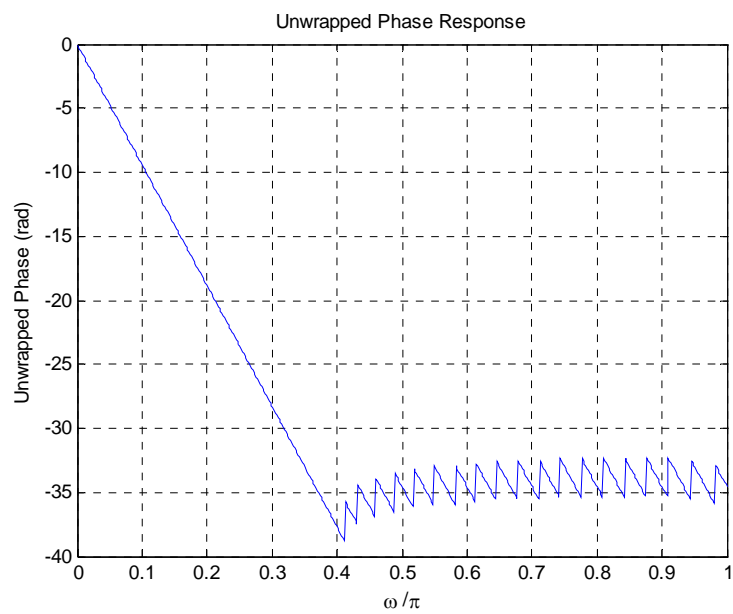
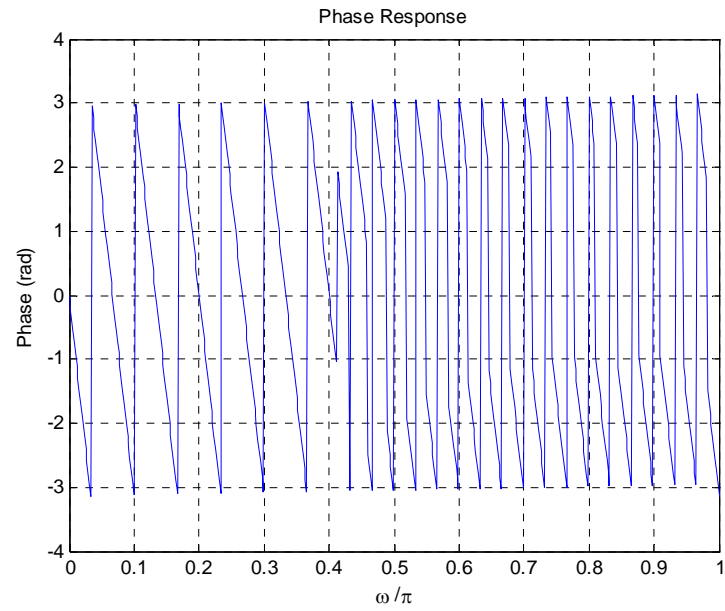
```

The coefficients of the lowpass filter corresponding to the specifications given in Question 7.23 are as shown below –

0.0003	0.0008	0.0003	-0.0011	-0.0017	0.0000	0.0026	0.0027	-0.0010	-0.0049
-0.0035	0.0033	0.0080	0.0034	-0.0074	-0.0119	-0.0018	0.0140	0.0161	-0.0027
-0.0241	-0.0201	0.0127	0.0406	0.0236	-0.0354	-0.0754	-0.0258	0.1214	0.2871
0.3597	0.2871	0.1214	-0.0258	-0.0754	-0.0354	0.0236	0.0406	0.0127	-0.0201
-0.0241	-0.0027	0.0161	0.0140	-0.0018	-0.0119	-0.0074	0.0034	0.0080	0.0033
-0.0035	-0.0049	-0.0010	0.0027	0.0026	0.0000	-0.0017	-0.0011	0.0003	0.0008
0.0003									

The generated gain and phase responses are given below:





From the gain plot we observe that the filter as designed DOES meet the specifications.

The filter order that meets the specifications is - $N=60$.

Q7.24 The MATLAB program to design and plot the gain and phase responses of a linear-phase FIR filter using `fir1` and `kaiser` is shown below. The filter order N and the parameter β are evaluated using `kaiserord`. The output data are the filter coefficients.

```

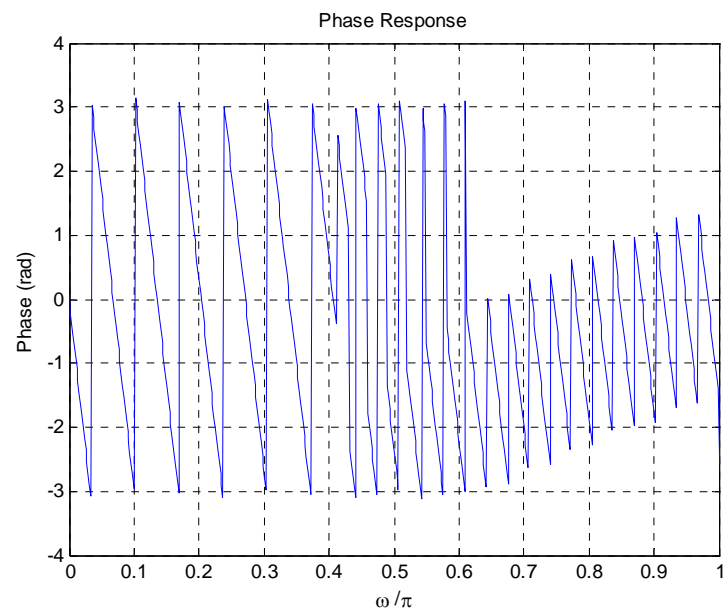
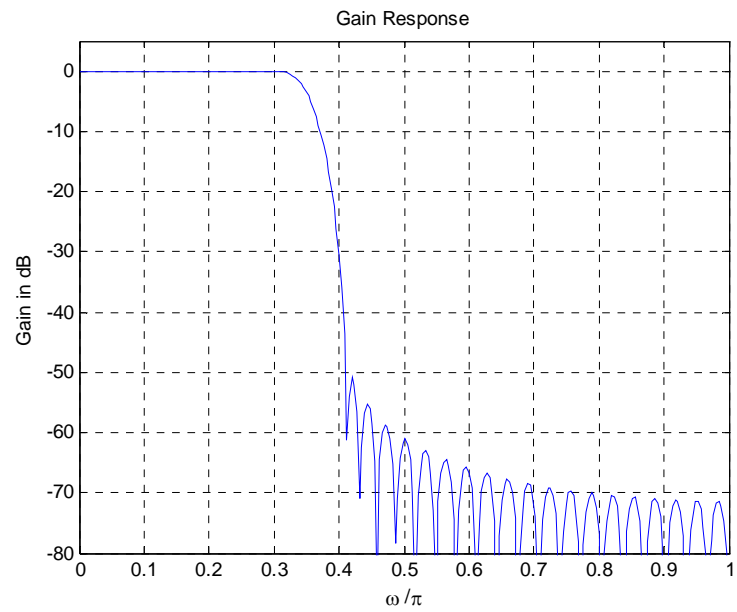
The coefficients of the lowpass filter corresponding to % Program Q7_24
% Use Kaiser window to design a linear phase Lowpass
% FIR Digital Filter meeting the design specification given
% in Q7.23. Use kaiserord and fir1.
% - Print out the numerator coefficients
%   for the transfer function.
% - Compute and plot the gain function.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;
% Design spec as given in Q7.23.
Wp = 0.31;
Ws = 0.41;
As = 50;
Ds = 10^(-As/20);
% Design the Filter
F = [Wp Ws];
A = [1 0];
DEV = [Ds Ds];
[N,Wn,BTA,Ftype] = kaiserord(F,A,DEV);
Win = kaiser(N+1,BTA);
h = fir1(N,Wn,Ftype,Win);
% Show the Numerator Coefficients
disp('Numerator Coefficients are ');disp(h);
% Compute and plot the gain response
[g, w] = gain(h,[1]);
figure(1);
plot(w/pi,g);grid;
axis([0 1 -80 5]);
xlabel('\omega /\pi'); ylabel('Gain in dB');
title('Gain Response');
% Compute the frequency response
w2 = 0:pi/511:pi;
Hz = freqz(h,[1],w2);
% Find and plot the phase
figure(2);
Phase = angle(Hz);
plot(w2/pi,Phase);grid;
xlabel('\omega /\pi'); ylabel('Phase (rad)');
title('Phase Response');
figure(3);
UPhase = unwrap(Phase);
plot(w2/pi,UPhase);grid;
xlabel('\omega /\pi'); ylabel('Unwrapped Phase (rad)');
title('Unwrapped Phase Response');

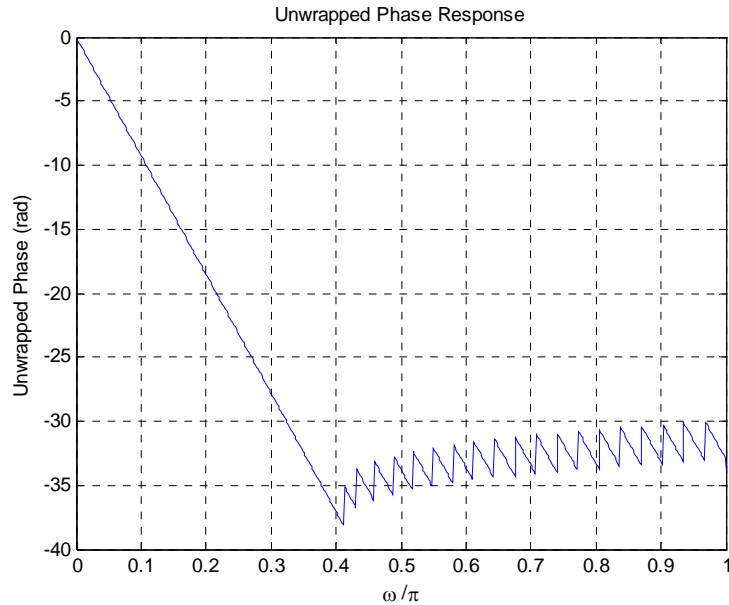
```

the specifications given in Question 7.23 are as shown below –

$W_p = 0.31$; $W_s = 0.41$; $A_s = 50$ dB.

The generated gain and phase responses are given below:





From the gain plot we observe that the filter as designed DOES meet the specifications.

The filter order that meets the specifications is – N=59.

Q7.25 The MATLAB program to design and plot the magnitude response of a linear-phase multiband FIR filter using `fir2` is shown below:

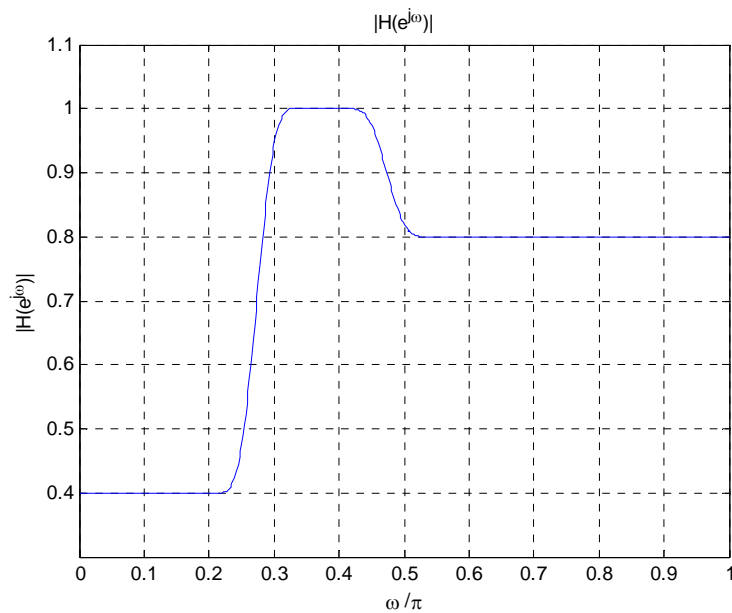
```
% Program Q7_25
% Use fir2 to design a linear phase Lowpass
% FIR Digital Filter meeting the design specification given
% in Q7.23.
% - Compute and plot the gain function.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;
% Design spec as given in Q7.25.
N = 95;
A = [0.4 0.4 1.0 1.0 0.8 0.8];
F = [0 0.25 0.3 0.45 0.5 1.0];
% Design the Filter
h = fir2(N,F,A);
% Compute and plot the gain response
[g, w] = gain(h,[1]);
figure(1);
plot(w/pi,g);grid;
%axis([0 1 -80 5]);
xlabel('\omega /\pi'); ylabel('Gain in dB');
title('Gain Response');
% Compute the frequency response
w2 = 0:pi/511:pi;
Hz = freqz(h,[1],w2);
% Plot
figure(2);
plot(w2/pi,abs(Hz));grid;
xlabel('\omega /\pi'); ylabel('|H(e^{j\omega})|');
```

```

title('|H(e^{j\omega})|');
% Find and plot the phase
figure(3);
Phase = angle(Hz);
plot(w2/pi,Phase);grid;
xlabel('\omega /\pi'); ylabel('Phase (rad)');
title('Phase Response');
figure(4);
UPhase = unwrap(Phase);
plot(w2/pi,UPhase);grid;
xlabel('\omega /\pi'); ylabel('Unwrapped Phase (rad)');
title('Unwrapped Phase Response');

```

The magnitude response of the filter designed for the specifications given in Question Q7.25 is shown below:



From the magnitude response plot we observe that the filter as designed DOES NOT meet the specifications.

Q7.26 The MATLAB program to design and plot the gain response of a linear-phase bandpass FIR filter using `firpm` and `kaiserord` is shown below:

Again there is ambiguity depending on whether δ_p is supposed to be 0.1 or 0.01. If you make it 0.01 you get this:

```

% Program Q7_26
% Use kaiserord and firpm to design the linear phase bandpass
% FIR Digital Filter specified in Q7.17.

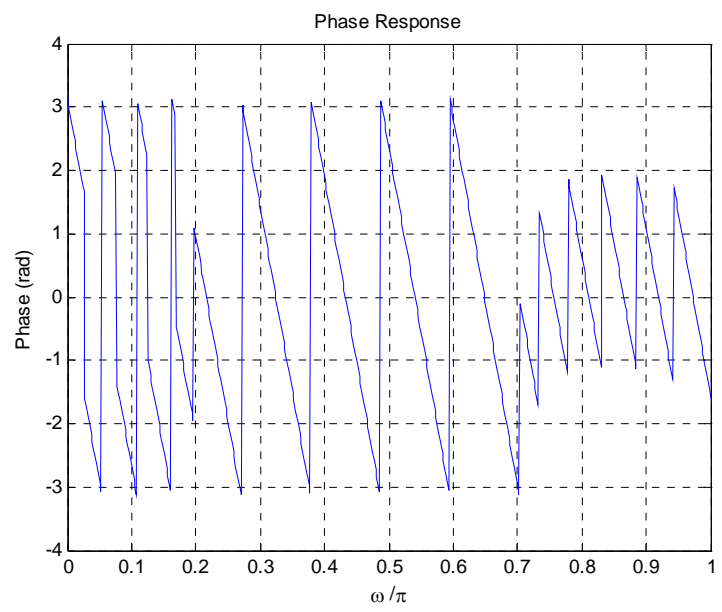
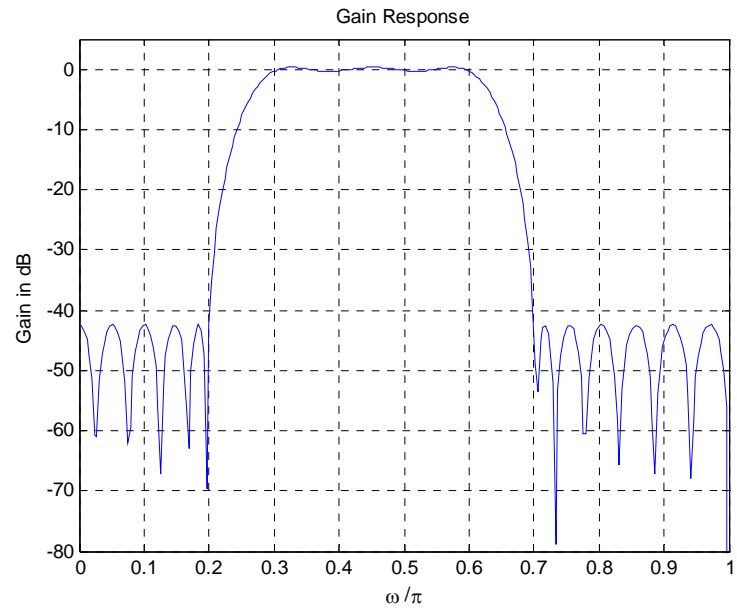
```

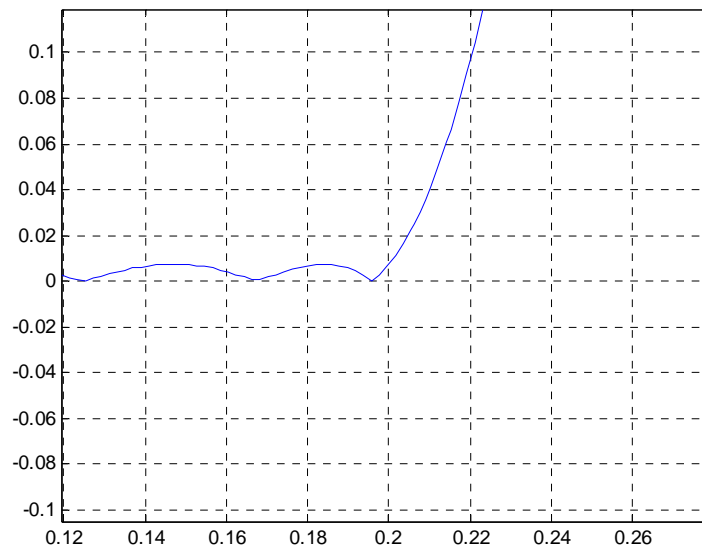
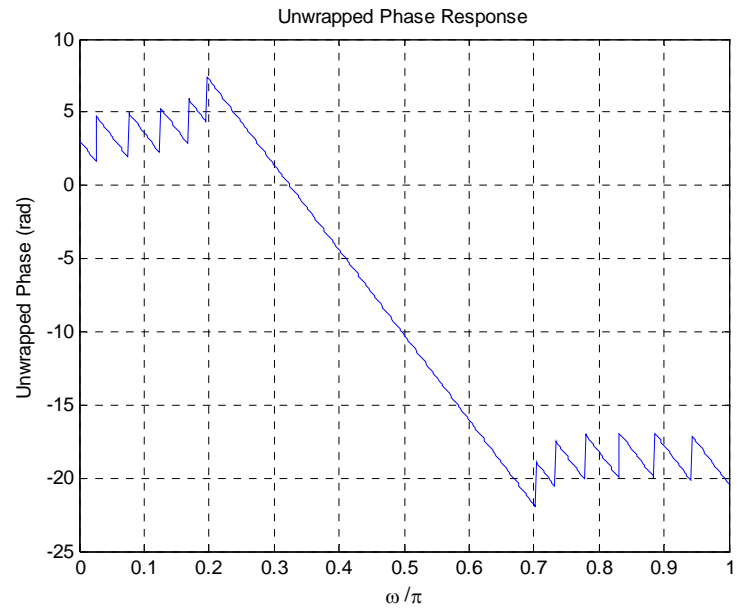
```

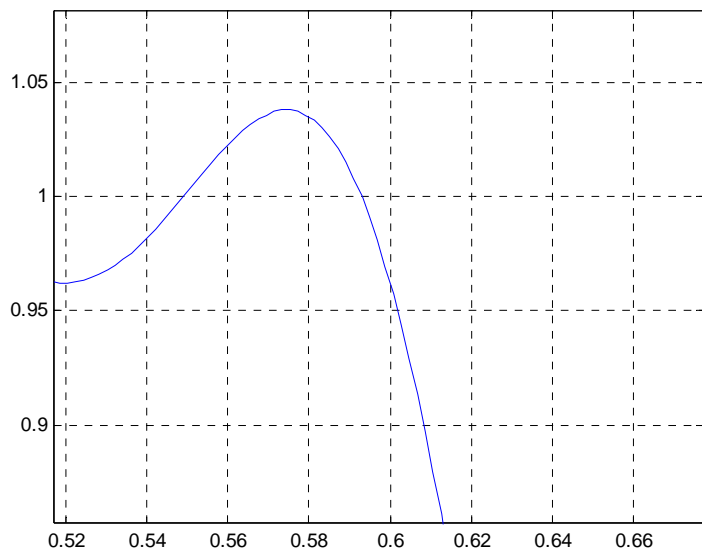
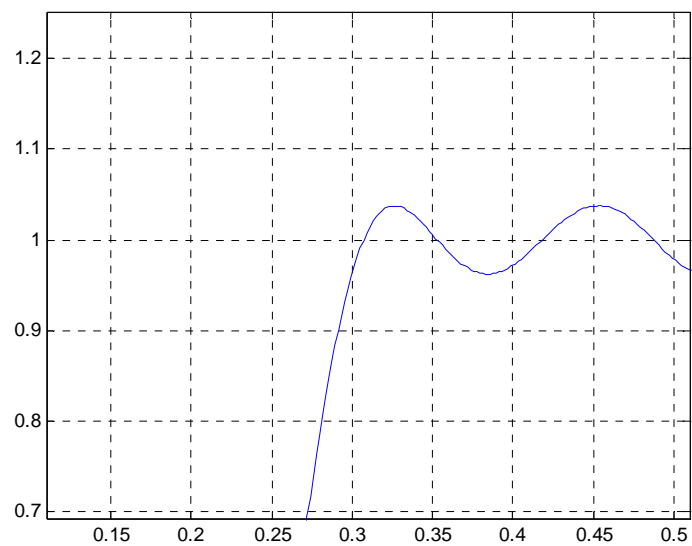
% - Print out the numerator coefficients
%   for the transfer function.
% - Compute and plot the gain function.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;
% Design spec as given in Q7.17.
F = [1200 1800 3600 4200];
A = [0 1 0];
DEV = [0.02 0.1 0.02];
Fs = 12000;
Dp = 0.1;
Ds = 0.02;
[N,Wn,BTA,FILTYPE] = kaiserord(F,A,DEV,Fs);
N
% firpm setup
F2 = 2*[0 1200 1800 3600 4200 6000]/Fs;
A2 = [0 0 1 1 0 0];
wgts = max(Dp,Ds)*[1/Ds 1/Dp 1/Ds];
h = firpm(N,F2,A2,wgts);
% Show the Numerator Coefficients
disp('Numerator Coefficients are ');disp(h);
% Compute and plot the gain response
[g, w] = gain(h,[1]);
figure(1);
plot(w/pi,g);grid;
axis([0 1 -80 5]);
xlabel('\omega /\pi'); ylabel('Gain in dB');
title('Gain Response');
% Compute the frequency response
w2 = 0:pi/511:pi;
Hz = freqz(h,[1],w2);
% Find and plot the phase
figure(2);
Phase = angle(Hz);
plot(w2/pi,Phase);grid;
xlabel('\omega /\pi'); ylabel('Phase (rad)');
title('Phase Response');
figure(3);
UPhase = unwrap(Phase);
plot(w2/pi,UPhase);grid;
xlabel('\omega /\pi'); ylabel('Unwrapped Phase (rad)');
title('Unwrapped Phase Response');

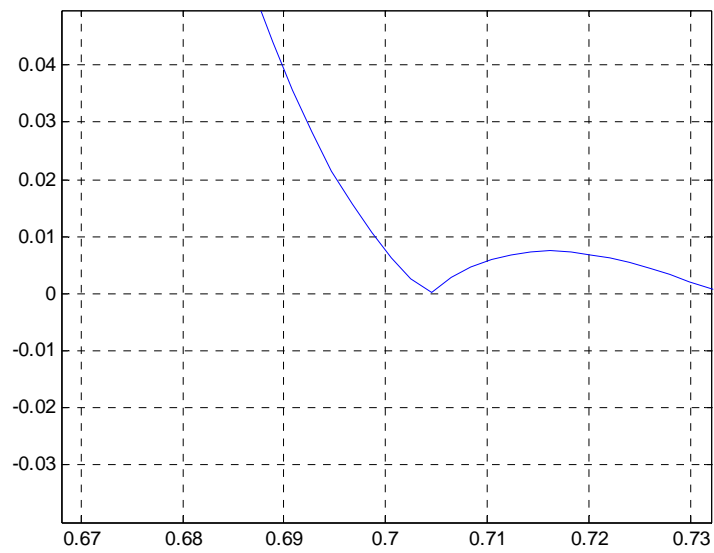
```

The gain response of the filter designed for the specifications given in Question Q7.17 is shown below:





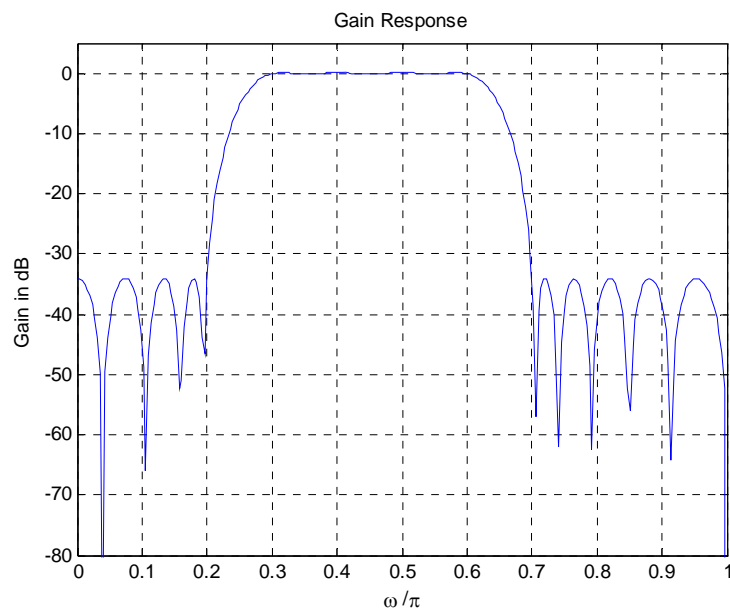


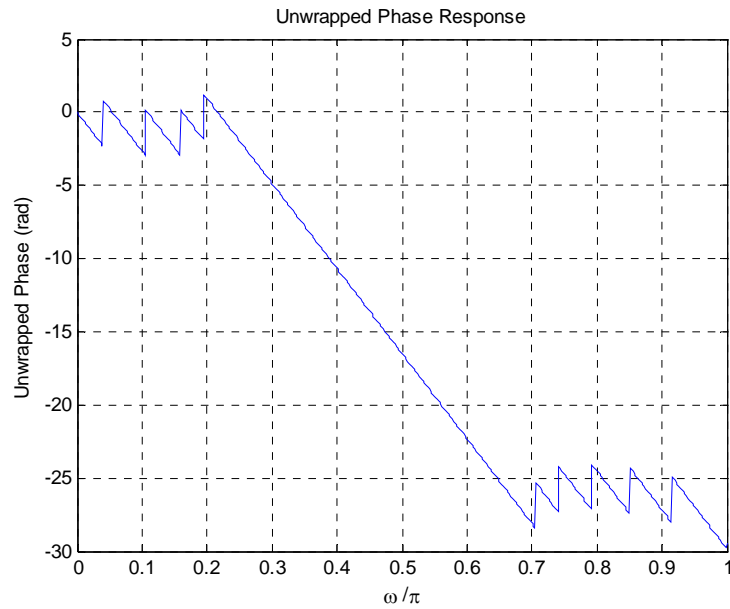


From the gain response plot we observe that the filter as designed DOES meet the specifications.

The filter order that meets the specifications is – $N=37$,

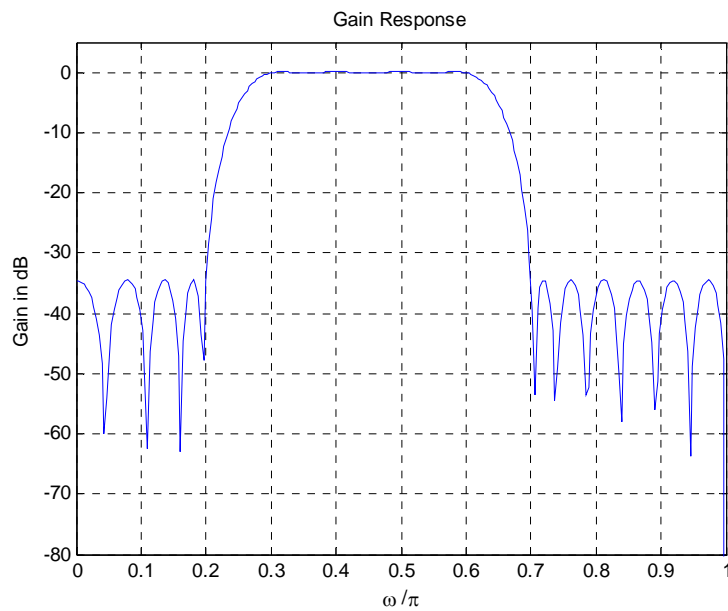
TAKING $\delta_p = 0.01$ instead, you get this:

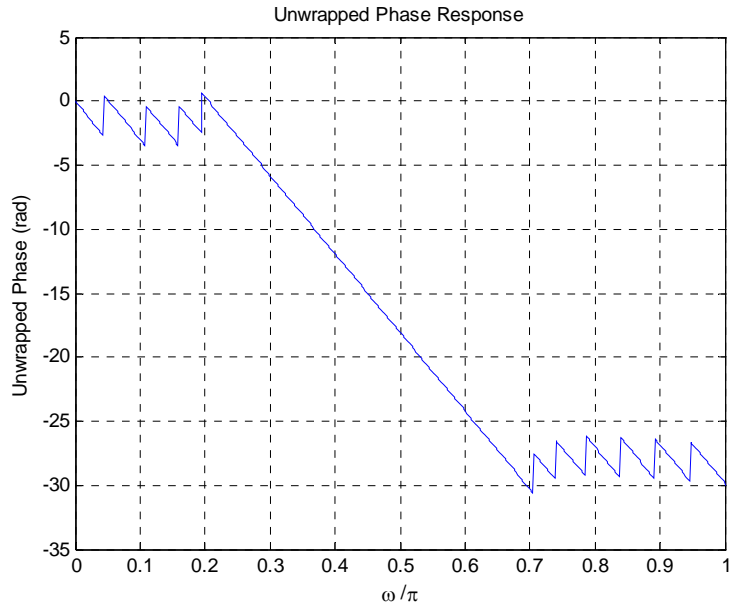




If you examine this one closely, it does not quite meet the spec at the passband edge frequencies. The order is $N=37$.

Increasing the order slightly to $N=39$, we obtain the following filter which does meet the spec:





Q7.27 Using the MATLAB program developed in Question Q7.26 the linear-phase FIR bandpass filter for the specifications of Question Q7.27 is designed.

Here's the new code:

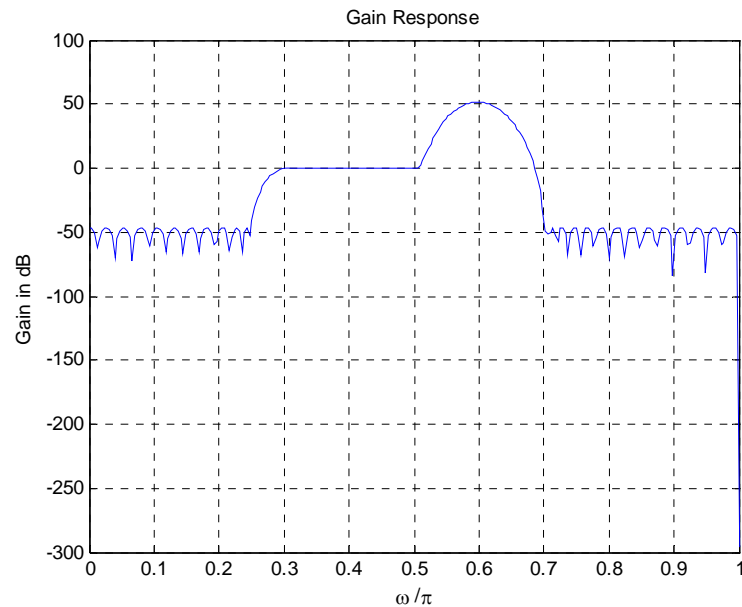
```
% Program Q7_27
% Use kaiserord and firpm to design the linear phase bandpass
% FIR Digital Filter specified in Q7.27.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;
% Design spec as given in Q7.27.
Fs1 = 1500;
Fp1 = 1800;
Fp2 = 3000;
Fs2 = 4200;
Fs = 12000;
Dp = 0.1;
Ds = 0.02;
F = [Fs1 Fp1 Fp2 Fs2];
A = [0 1 0];
DEV = [Ds Dp Ds];
[N,Wn,BTA,FILTYPE] = kaiserord(F,A,DEV,Fs);
% firpm setup
ws1 = 2*Fs1/Fs;
wp1 = 2*Fp1/Fs;
wp2 = 2*Fp2/Fs;
ws2 = 2*Fs2/Fs;
F2 = [0 ws1 wp1 wp2 ws2 1];
A2 = [0 0 1 1 0 0];
wgts = max(Dp,Ds)*[1/Ds 1/Dp 1/Ds];
h = firpm(N,F2,A2,wgts);
% Show the Numerator Coefficients
```

```

disp('Numerator Coefficients are ');disp(h);
% Compute and plot the gain response
[g, w] = gain(h,[1]);
figure(1);
plot(w/pi,g);grid;
axis([0 1 -80 5]);
xlabel('\omega /\pi'); ylabel('Gain in dB');
title('Gain Response');
% Compute the frequency response
w2 = 0:pi/511:pi;
Hz = freqz(h,[1],w2);
% Find and plot the phase
figure(2);
Phase = angle(Hz);
plot(w2/pi,Phase);grid;
xlabel('\omega /\pi'); ylabel('Phase (rad)');
title('Phase Response');
figure(3);
UPhase = unwrap(Phase);
plot(w2/pi,UPhase);grid;
xlabel('\omega /\pi'); ylabel('Unwrapped Phase (rad)');
title('Unwrapped Phase Response');
figure(4);
% Add lines to the plot to help determine if the spec was met.
hold on;
tmpY = 0:1.4/4:1.4;
tmpX = ones(1,length(tmpY))*wp1;
plot(tmpX,tmpY,'r-'); % vertical line at passband edge freq
tmpX = ones(1,length(tmpY))*wp2;
plot(tmpX,tmpY,'r-'); % vertical line at passband edge freq
tmpX = ones(1,length(tmpY))*ws1;
plot(tmpX,tmpY,'g-'); % vertical line at stopband edge freq
tmpX = ones(1,length(tmpY))*ws2;
plot(tmpX,tmpY,'g-'); % vertical line at stopband edge freq
tmpY = ones(1,length(w))*(Dp);
plot(w/pi,tmpY,'r-'); % horizontal line at Dp
tmpY = ones(1,length(w))*(Ds);
plot(w/pi,tmpY,'g-'); % horizontal line at Ds
% now plot the Frequency Response
plot(w2/pi,abs(Hz));grid;
hold off;

```

The gain response of the filter is shown below:

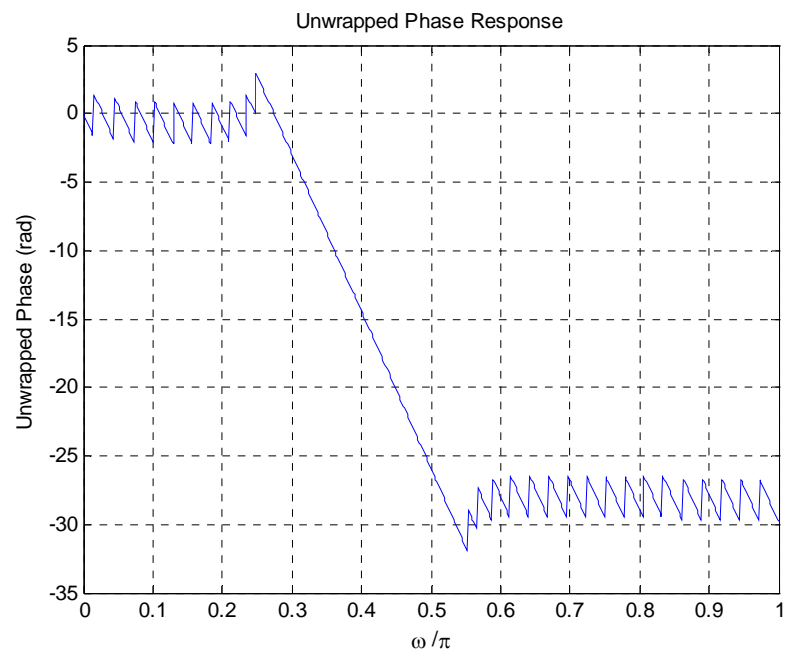
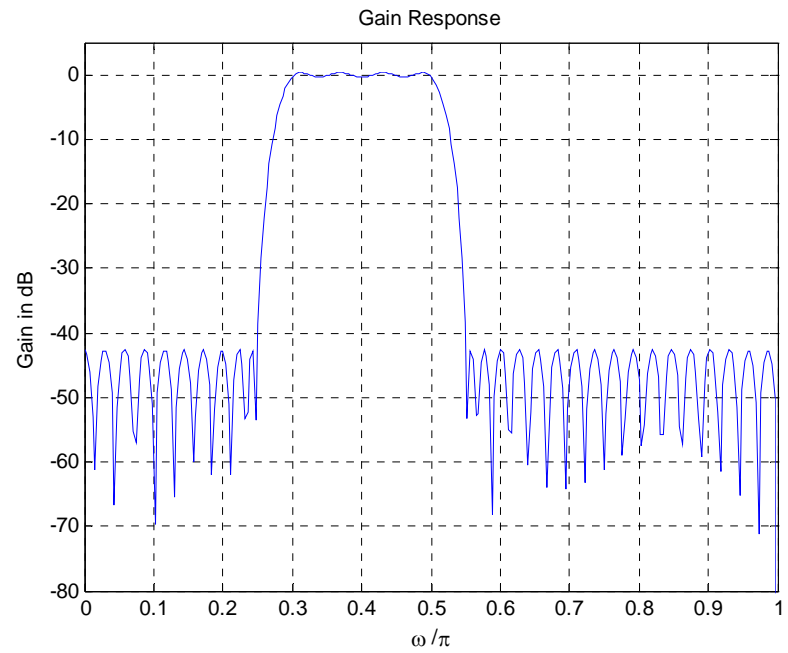


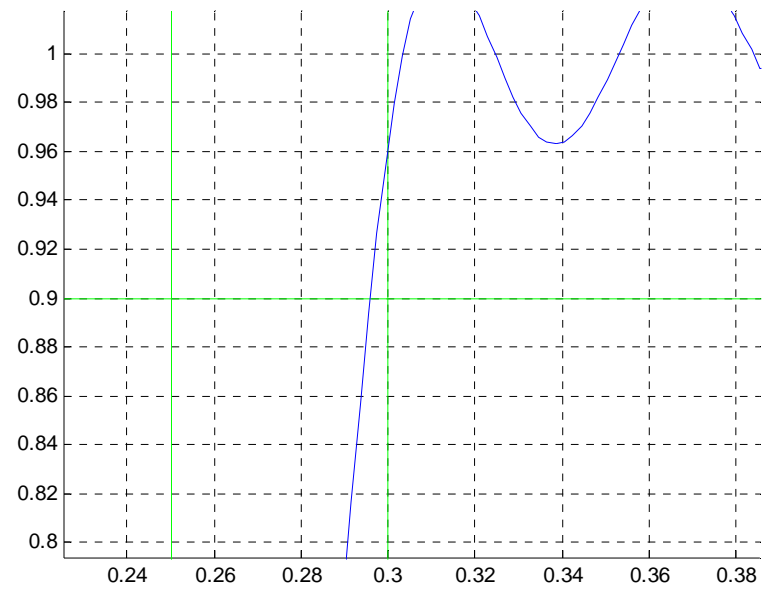
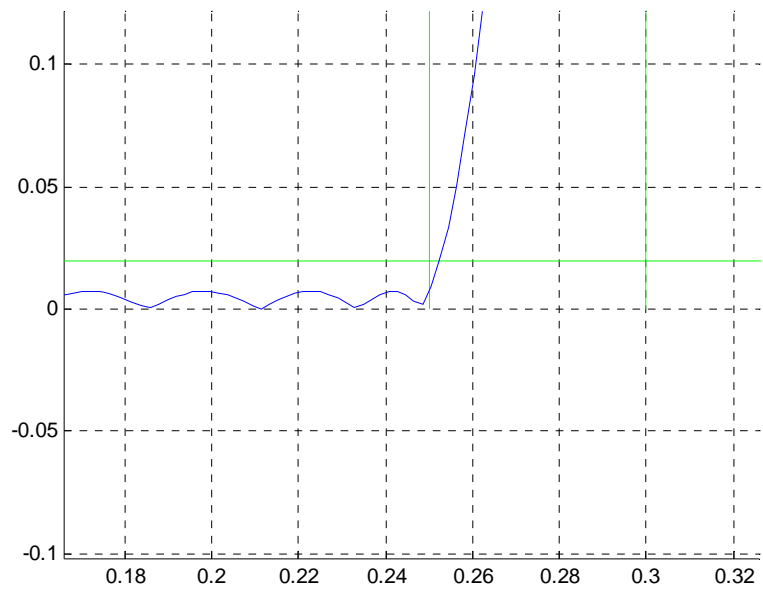
From the gain response plot we observe that the filter as designed DOES NOT meet the specifications: THE PASSBAND RIPPLE IS EXCEEDED IN THE UPPER TRANSITION BAND. The filter is optimal in the minimax sense, as are all Parks-McClellan designs. The reason this occurred is because: 1) the transition bands are don't care regions for firpm, 2) erratic behavior of this type in the transition bands is often observed if the transition bandwidths are not equal. The usual solution is to make all transition bands have the width of the smallest transition band; e.g. to overdesign the other transition bands in order to make firpm "play nice."

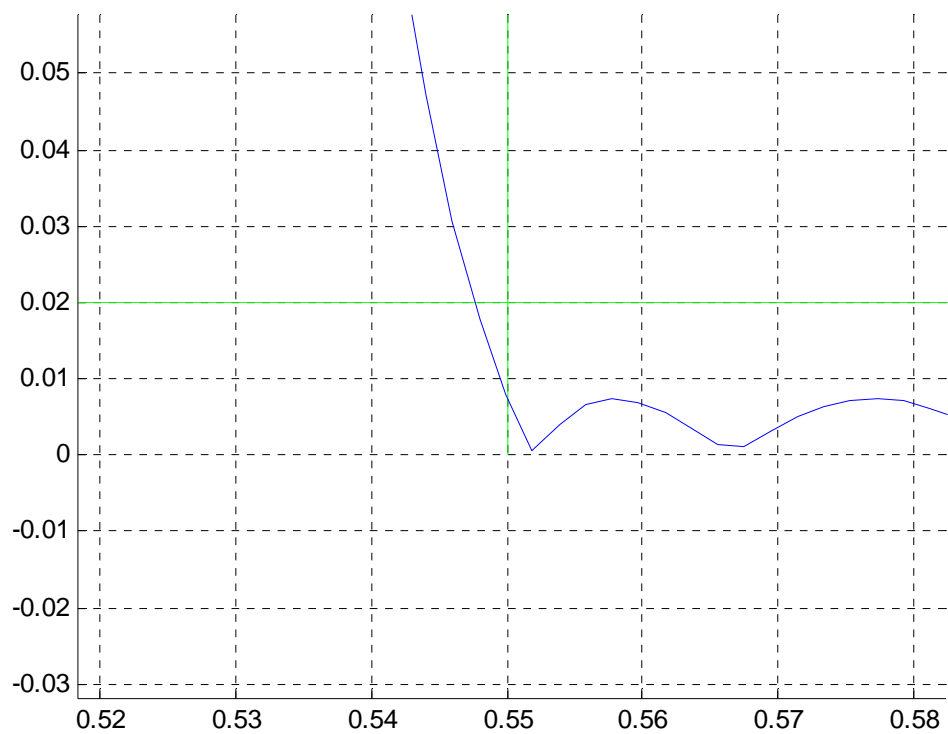
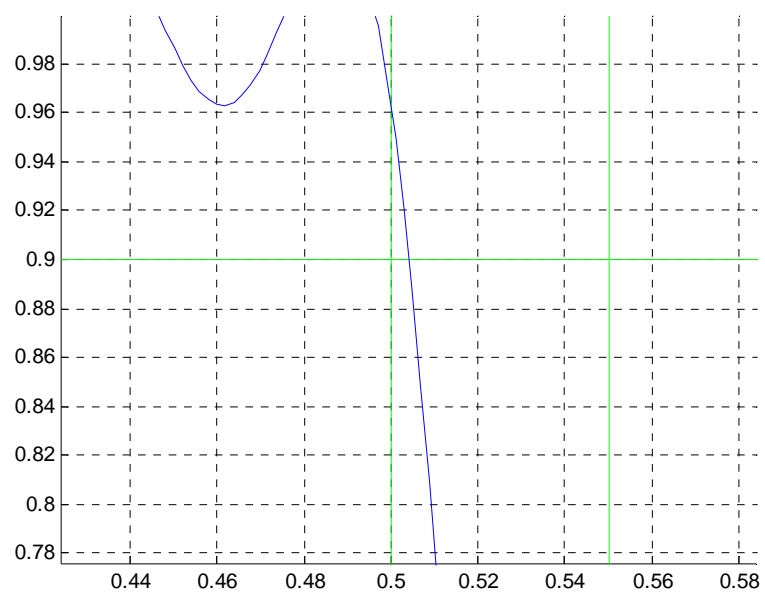
The filter order that meets the specifications is – $N = 73$ (as delivered by kaiserord).

The new specifications for smooth roll-off in the transition bands are –

```
Fs1 = 1500;
Fp1 = 1800;
Fp2 = 3000;
Fs2 = 3300;
Fs = 12000;
```







Date: 4 December 2007

Signature: Havlicek