SOLUTIONS MANUAL

to accompany

Digital Signal Processing: A Computer-Based Approach

Fourth Edition

Sanjit K. Mitra

Prepared by

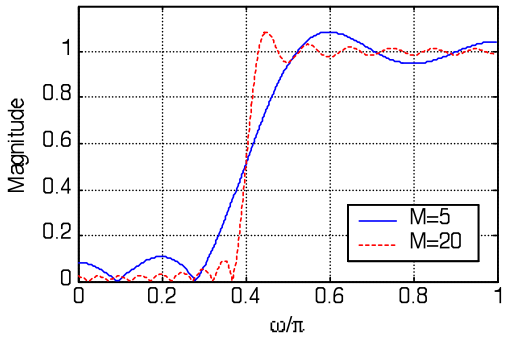
Chowdary Adsumilli, John Berger, Marco Carli,
Hsin-Han Ho, Rajeev Gandhi, Martin Gawecki, Chin Kaye Koh,
Luca Lucchese, Mylene Queiroz de Farias, and Travis Smith

Chapter 10 – Part 2

M10.1 The impulse response coefficients of the truncated FIR highpass filter with cutoff frequency at 0.4π can be generated using the following MATLAB statements:

```
n = -M:M;
num = -0.4*sinc(0.4*n);
num(M+1) = 0.6;
```

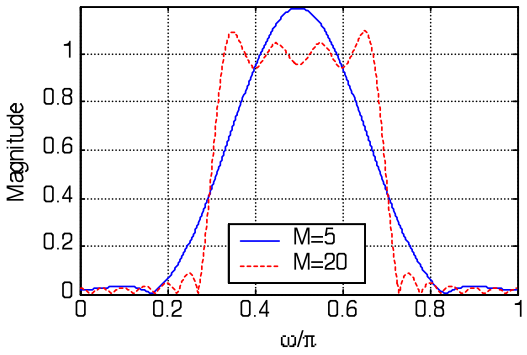
The magnitude responses of the truncated FIR highpass filter for two values of M are shown below:



M10.2 The impulse response coefficients of the truncated FIR bandpass filter with cutoff frequencies at 0.7π and 0.3π can be generated using the following MATLAB statements:

```
n = -M:M;
num = 0.7*sinc(0.7*n) - 0.3*sinc(0.3*n);
```

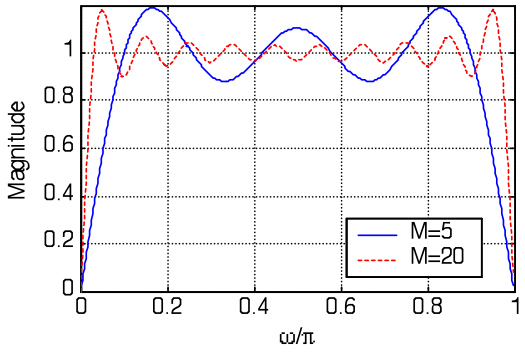
The magnitude responses of the truncated FIR bandpass filter for two values of M are shown below:



M10.3 The impulse response coefficients of the truncated Hilbert transformer can be generated using the following MATLAB statements:

```
n = 1:M;
c = 2*sin(pi*n/2).*sin(pi*n/2);b = c./(pi*n);
num = [-fliplr(b) 0 b];
```

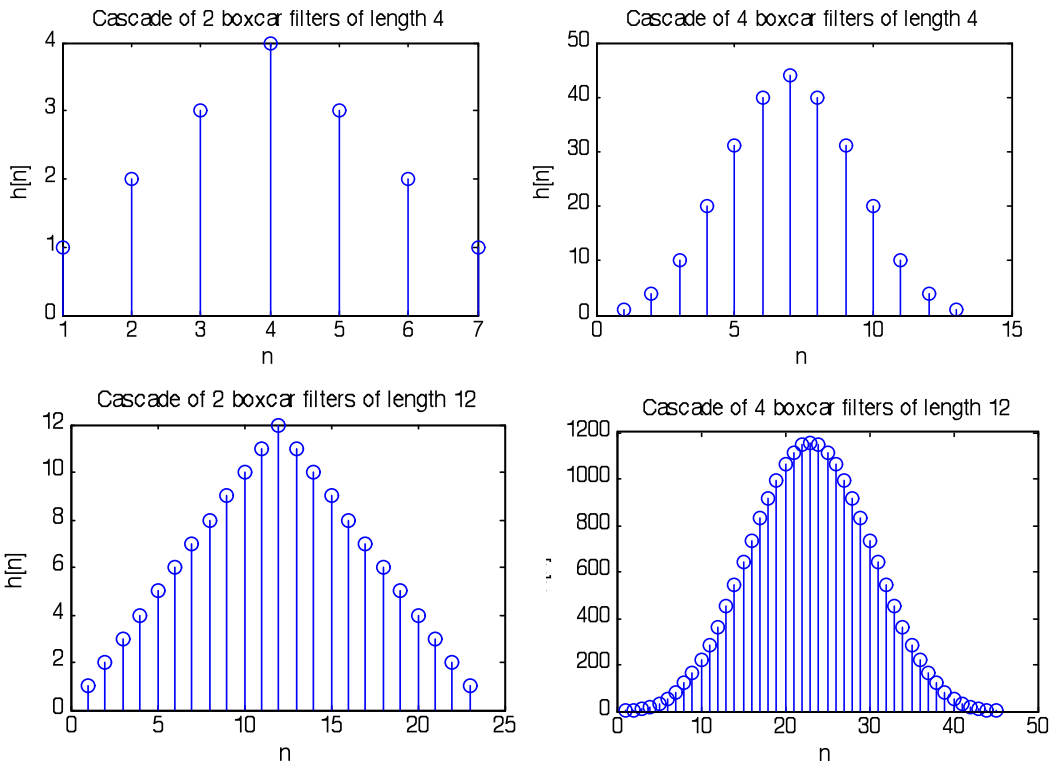
The magnitude responses of the truncated Hilbert transformer for two values of M are shown below:



M10.4 The MATLAB code for this problem is shown below:

```
% Problem M10.4
% Cascade of 2 boxcar filters of length 4
K = 2;N = 4;b = firgauss(K,N);
figure(1);
stem(b);xlabel('n');ylabel('h[n]');
title('Cascade of 2 boxcar filters of length 4');

% Cascade of 4 boxcar filters of length 4
K = 4;N = 4;b = firgauss(K,N);
figure(2);
stem(b);xlabel('n');ylabel('h[n]');
title('Cascade of 4 boxcar filters of length 4');

% Cascade of 2 boxcar filters of length 12
K = 2;N = 12;b = firgauss(K,N);
figure(3);
stem(b);xlabel('n');ylabel('h[n]');
title('Cascade of 2 boxcar filters of length 12');

% Cascade of 4 boxcar filters of length 12
K = 4;N = 12;b = firgauss(K,N);
figure(4);
stem(b);xlabel('n');ylabel('h[n]');
title('Cascade of 4 boxcar filters of length 12');
```

The resulting plots are as follows:

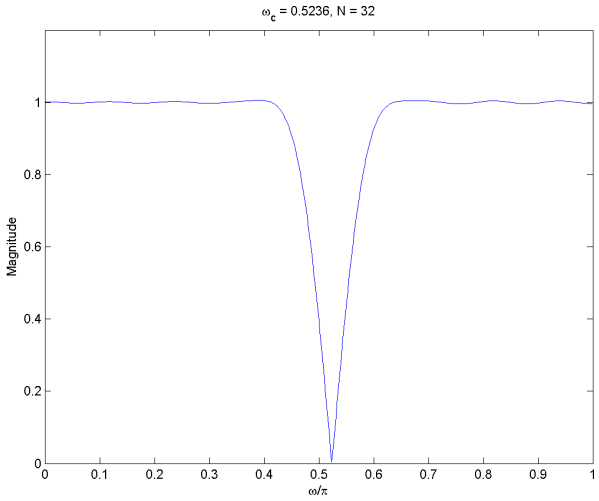


We can see that by increasing either K or N, the approximation to a Gaussian function gets better. It is noted that increasing the number of boxcar filters K to a number greater than 3 greatly affects the Gaussian shape of the impulse response.

M10.5 The MATLAB program for this problem is shown below:

```
% Problem M10.05
N = 32;
Fc = 450;
Fn = 75;
fc = (Fn/Fc)*pi;
M = N/2;
n = -M:1:M;
t = fc*n;
lp = fc*sinc(t);
b = 2*[lp(1:M) (lp(M+1) - 0.5) lp((M+2):N+1)];
bw = b.*hamming(N+1)';
[h2, w] = freqz(bw, 1, 512);
plot(w/pi, abs(h2));axis([0 1 0 1.2]);
xlabel('\omega/\pi');ylabel('Magnitude');
title(['\omega_c = ', num2str(fc), ', N = ', num2str(N)]);
```

The resulting frequency response is:



M10.6 Given $D(x) = 0.96x^2 - 0.96x - 0.76.$

Its approximation over the range $-2 \le x \le 3$ is given by $A(x) = a_0 + a_1 x$. We want to minimize the peak value of the absolute error, i.e., minimize:

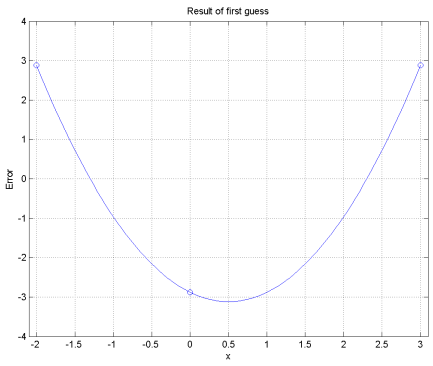
$$\max_{-2 \le x \le 3} \left| 0.96x^2 - 0.96x - 0.76 - a_0 - a_1 x \right|.$$

Since there are 3 unknowns, a_0, a_1, and ε, we need 3 extremal points on x, which we arbitrarily choose as $x_1 = -2, x_2 = 0,$ and $x_3 = 3$ We then solve the 3 linear equations: $a_0 + a_1 x - (-1)^\ell \varepsilon = D(x_\ell), \ell = 1,2,3.$ This leads to:

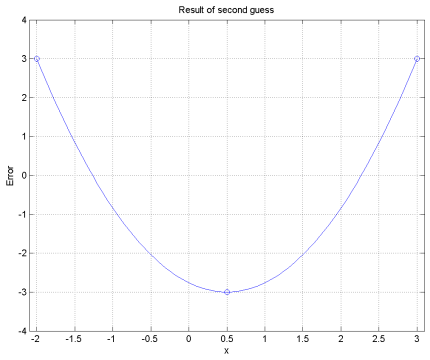
$$\begin{bmatrix} 1 & -2 & 1 \\ 1 & 0 & -1 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \varepsilon \end{bmatrix} = \begin{bmatrix} 5 \\ -0.76 \\ 5 \end{bmatrix},$$

whose solution yields $a_0 = 2.21,$ $a_1 = 0,$ and $\varepsilon = 2.88.$

A plot of the corresponding error: $\mathcal{E}_1(x) = 2.21x^2 - 2.88,$ is shown below in Figure (a):



(a)



(b)

After looking at $\mathcal{E}_1(x)$, we move the second extremal point x_2 to the location where $\mathcal{E}_1(x)$ is a minimum. The next extremal points are therefore given by $x_1 = -2$, $x_2 = 0.5$, and $x_3 = 3$. The new values of the unknowns are obtained by solving:

$$\begin{bmatrix} 1 & -3 & 1 \\ 1 & -0.5 & -1 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \varepsilon \end{bmatrix} = \begin{bmatrix} 5 \\ -1 \\ 5 \end{bmatrix},$$

which yields $a_0 = 2$, $a_1 = 0$, and $\varepsilon = 3$. A plot of the corresponding error $\mathcal{E}_2(x) = 2x^2 - 3$ is shown above in Figure (b).

The code for this problem is given below:

```
% Problem M10.06
clear;

polyD = [0.96 -0.96 -0.76];
x = [-2 0 3];
D = (polyD(1).*x.^2 + polyD(2).*x + polyD(3))'
A = [1 x(1) 1;1 x(2) -1;1 x(3) 1];
C = inv(A)*D

y = -2:0.05:3;
E = polyD(1).*y.^2 + polyD(2).*y + polyD(3) - C(1) -
C(2).*y;

% Results of first guess
figure(1);
plot(y,E);
axis([-2.1 3.1 -4 4]);
xlabel('x');
ylabel('Error');
title('Result of first guess');
```

```
hold on;
plot([x(1) x(1)], [E(1) E(1)], 'o');
plot([x(2) x(2)], [E(61) E(61)], 'o');
plot([x(3) x(3)], [E(end) E(end)], 'o');

grid;
hold off;

x = [-2 0.5 3];
D = (polyD(1).*x.^2 + polyD(2).*x + polyD(3))'
A = [1 x(1) 1;1 x(2) -1;1 x(3) 1];
C = inv(A)*D

y = -2:0.05:3;
E = polyD(1).*y.^2 + polyD(2).*y + polyD(3) - C(1) -
C(2).*y;

% Results of second guess
figure(2);
plot(y,E);
axis([-2.1 3.1 -4 4]);
xlabel('x');
ylabel('Error');
title('Result of second guess');
hold on;
plot([x(1) x(1)], [E(1) E(1)], 'o');
plot([x(2) x(2)], [E(51) E(51)], 'o');
plot([x(3) x(3)], [E(end) E(end)], 'o');
grid;
hold off;
```

M10.7 Given: $D(x) = 0.5x^3 - 2.25x^2 + 3x + 3$..

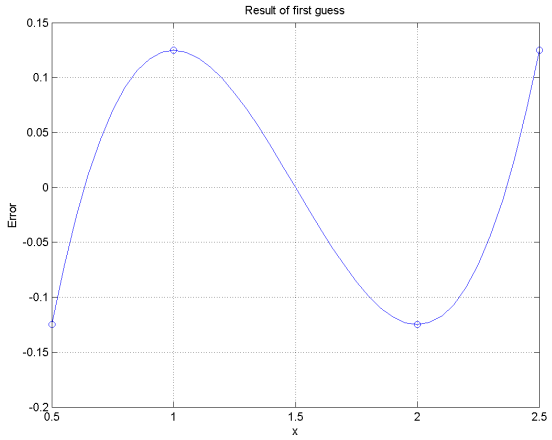
Its approximation over the range $0.5 \le x \le 2.5$ is given as $A(x) = a_0 + a_1 x + a_2 x^2$. We want to minimize the peak value of the absolute error, or minimize:

$$\max_{-2 \le x \le 2} \left| 0.5x^3 - 2.25x^2 + 3x + 3 - a_0 - a_1 x - a_2 x^2 \right|.$$

Since there are 4 unknowns, a_0, a_1, a_2, and ε, we need 4 extremal points on x, which we arbitrarily choose as $x_1 = 0.5, x_2 = 1$, $x_3 = 2$, and $x_4 = 2.5$ We then solve the 4 linear equations: $a_0 + a_1 x + a_2 x^2 - (-1)^\ell \varepsilon = D(x_\ell)$, $\ell = 1,\ 2,\ 3,\ 4$. This leads to:

$$\begin{bmatrix} 1 & -2 & 4 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & -1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \varepsilon \end{bmatrix} = \begin{bmatrix} 4 \\ 4.25 \\ 4 \\ 4.25 \end{bmatrix},$$

whose solution yields $a_0 = 4.125$, $a_1 = 0$, $a_2 = 0$, and $\varepsilon = -0.125$. A plot of the corresponding error is shown below in figure below:



The code for generating this plot is as follows:

```
% Problem M10.07
clear;

polyD = [1/2 -9/4 3 3];
x = [0.5 1 2 2.5];
d = polyD(1).*x.^3 + polyD(2).*x.^2 + polyD(3).*x +
polyD(4);
D = d';
A = [   1 x(1) x(1)^2 1;
        1 x(2) x(2)^2 -1;
        1 x(3) x(3)^2 1;
        1 x(4) x(4)^2 -1];
C = inv(A)*D;

y = 0.5:0.05:2.5;
E = polyD(1).*y.^3 + polyD(2).*y.^2 + polyD(3).*y...
    + polyD(4) - C(1) - C(2).*y - C(3).*y.^2;

% Results of first guess
figure(1);
plot(y,E);
% axis([0.4 2.6 -3.9 2.35]);
xlabel('x');
ylabel('Error');
title('Result of first guess');
hold on;
plot([x(1) x(1)], [E(1) E(1)], 'o');
plot([x(2) x(2)], [E(end) E(end)], 'o');
```

```
plot([x(3) x(3)], [E(1) E(1)], 'o');
plot([x(4) x(4)], [E(end) E(end)], 'o');
grid;
hold off;
```

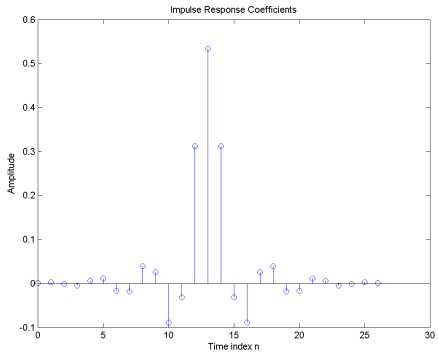
M10.8 Given: $\omega_p = \frac{6\pi}{15}$, $\omega_s = \frac{10\pi}{15} \Rightarrow \omega_T = \frac{8\pi}{15}$.

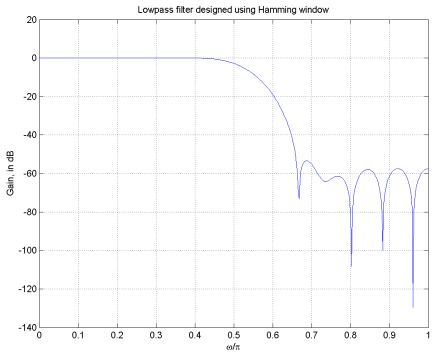
From Table 10.2, we can verify that the minimum stopband attenuation requirement is satisfied by the Hann, Hamming, and Blackman windows, so all three implementations are computed. The code for this problem is shown below:

```
% Problem M10.8
fp = 3;
fs = 5;
Fs = 15;

wp = (fp*pi)/(Fs/2);
ws = (fs*pi)/(Fs/2);
wc = (wp + ws)/2;
dw = ws - wp;

% Hamming Window
M = ceil(3.32*pi/dw);
N = 2*M+1;
n = -M:M;
num = (wc/pi)*sinc(n*(wc/pi));
wh = hamming(N)';
b = num.*wh;

figure(1);
k = 0:2*M;
stem(k,b);
title('Impulse Response Coefficients');
xlabel('Time index n'); ylabel('Amplitude');

figure(2);
[h, w] = freqz(b,1,512);
plot(w/pi, 20*log10(abs(h))); grid;
xlabel('\omega/\pi'); ylabel('Gain, in dB');
title('Lowpass filter designed using Hamming window');

% Hann Window
M = ceil(3.11*pi/dw);N = 2*M+1;n = -M:M;
num = (wc/pi)*sinc(n*(wc/pi));
wh = hann(N)';b = num.*wh;

figure(3);
```

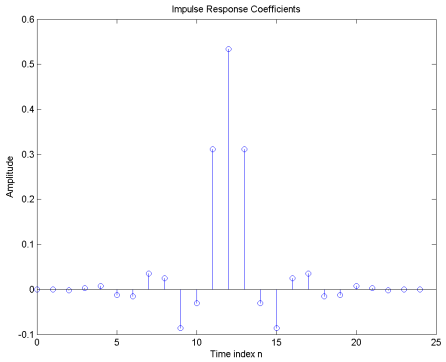
```
k = 0:2*M;
stem(k,b);
title('Impulse Response Coefficients');
xlabel('Time index n'); ylabel('Amplitude');

figure(4);
[h, w] = freqz(b,1,512);
plot(w/pi, 20*log10(abs(h)));grid;
xlabel('\omega/\pi');ylabel('Gain, in dB');
title('Lowpass filter designed using Hann window');

% Blackman Window
M = ceil(5.56*pi/dw);N = 2*M+1;n = -M:M;
num = (wc/pi)*sinc(n*(wc/pi));
wh = blackman(N)';b = num.*wh;

figure(5);
k = 0:2*M;
stem(k,b);
title('Impulse Response Coefficients');
xlabel('Time index n'); ylabel('Amplitude');

figure(6);
[h, w] = freqz(b,1,512);
plot(w/pi, 20*log10(abs(h)));grid;
xlabel('\omega/\pi');ylabel('Gain, in dB');
title('Lowpass filter designed using Blackman window');
```

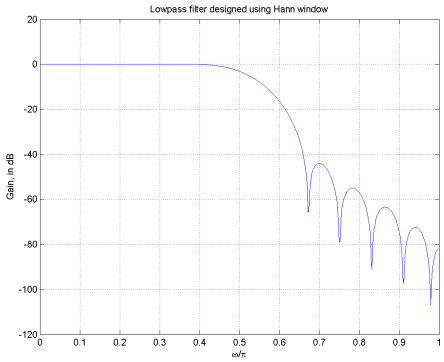
Lowpass filter design using the Hamming window:



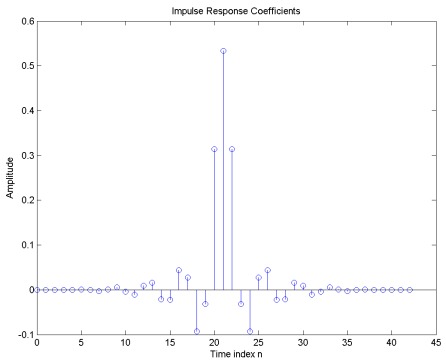


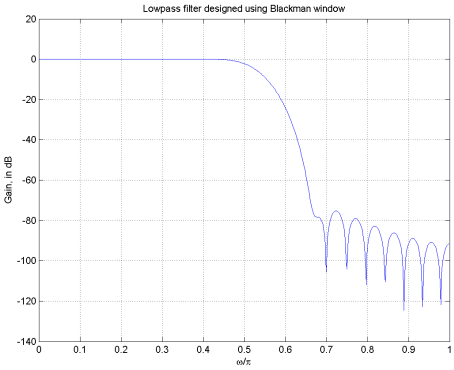
Lowpass filter design using the Hann window:





Lowpass filter design using the Blackman window:





Note that the Hann window method results in using the lowest filter order. All filters meet the requirements of the specifications.

M10.9 For this problem, we have $\alpha_s = 40$ so we can use Eq. (10.44):

$\beta = 0.5842(40-21)^{0.4} + 0.07886(40-21) = 3.3953.$

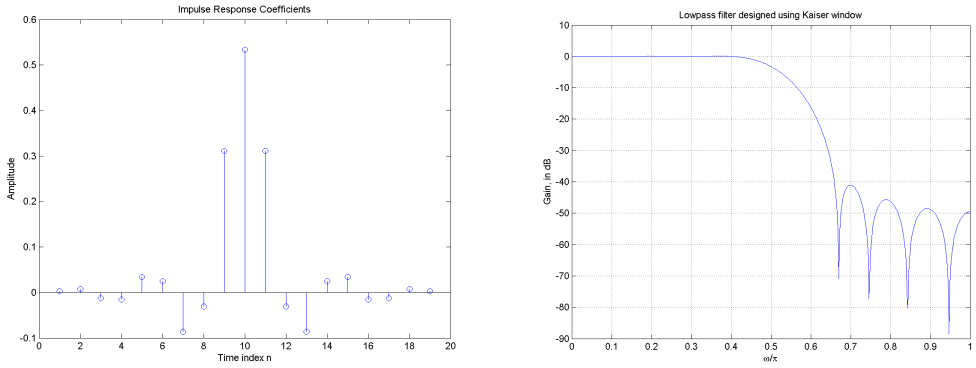
And using Eq. (10.45), we can calculate N: $N = \dfrac{40-8}{2.285(4\pi/15)} = 16.7165 \approx 17.$

We choose 18 since N must be even. The code for plotting these responses is shown below:

```
% Problem M10.9
fp = 3;
fs = 5;
Fs = 15;
as = 40;

wp = (fp*pi)/(Fs/2);
ws = (fs*pi)/(Fs/2);
wc = (wp + ws)/2;
dw = ws - wp;

beta = 0.5842*(as-21)^0.4 + 0.07886*(as-21);
N = (as - 8)/(2.285*dw);
```

```
N = round(N) + mod(round(N),2);
n = -N/2:N/2;
num = (wc/pi)*sinc(n*(wc/pi));
wh = kaiser(N+1,beta)';
b = num.*wh;
disp([beta N])

figure(1);
stem(b);
title('Impulse Response Coefficients');
xlabel('Time index n');
ylabel('Amplitude')

figure(2);
[h, w] = freqz(b,1,512);
plot(w/pi, 20*log10(abs(h)));grid;
xlabel('\omega/\pi');
ylabel('Gain, in dB');
title('Lowpass filter designed using Kaiser window');
```

The resulting figures are:



M10.10 Given: $\omega_p = 0.3\pi, \omega_s = 0.4\pi, \alpha_s = 42\text{dB}, \omega_c = 0.35\pi, \Delta\omega = 0.1\pi$.

We will use the Hann window since it meets the requirements and has the lowest order from Table 10.2.

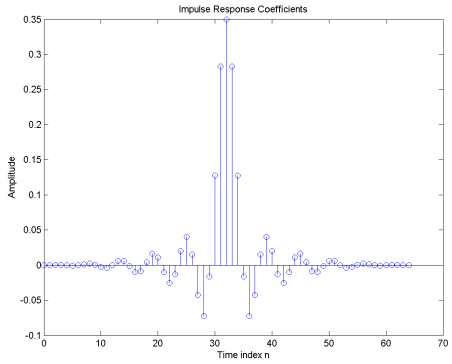
$$M = \frac{3.11\pi}{\Delta\omega} \approx 32 \qquad \Rightarrow \qquad N = 65$$

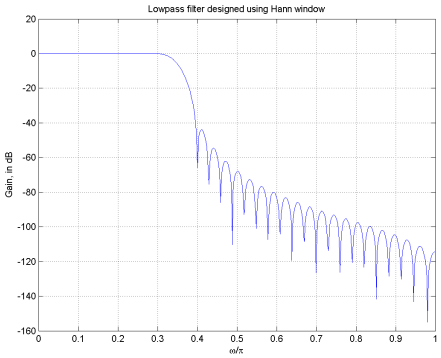
The code for this problem is shown below:

```
% Problem M10.10
wp = 0.3*pi;
ws = 0.4*pi;
wc = (wp + ws)/2;
dw = ws - wp;
```

```
M = ceil(3.11*pi/dw);
N = 2*M + 1;
disp(M)

n = -M:M;
lp = (wc/pi)*sinc((wc/pi)*n);
wh = hanning(N);
b = lp.*wh';

figure(1);
k = 0:2*M;
stem(k,b);
title('Impulse Response Coefficients');
xlabel('Time index n');ylabel('Amplitude');

figure(2);
[h, w] = freqz(b,1,512);
plot(w/pi, 20*log10(abs(h)));grid;
xlabel('\omega/\pi');ylabel('Gain, in dB');
title('Lowpass filter designed using Kaiser window');
```

The generated plots are as follows:





M10.11 We use the same specifications from Problem M10.10 but we use the Dolph-Chebyshev window, with length derived using Eq. (10.41):

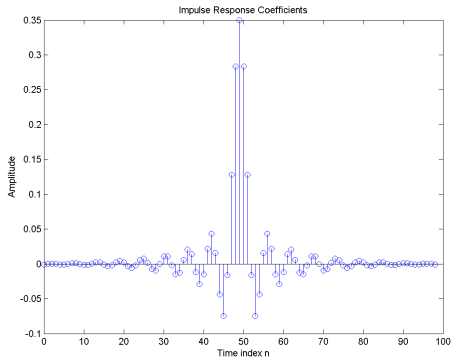
$$N = \frac{2.056(40) - 16.4}{2.285(0.1\pi)} \approx 99..$$

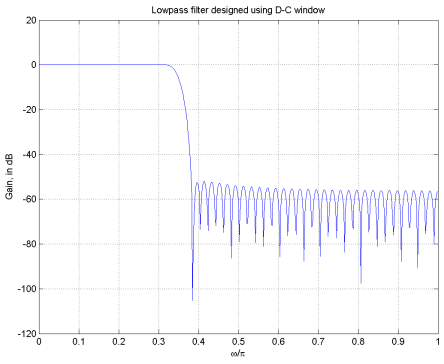
We use $N = 99$, which is a much higher order than in Problem 10.10. The code for generating the plots is shown below:

```
% Problem M10.11
wp = 0.3*pi;
ws = 0.4*pi;
wc = (wp + ws)/2;
dw = ws - wp;
```

```
N = ceil((2.056*42 - 16.4)/(2.285*dw));
N = N + (1 - mod(N,2));
M = (N-1)/2;
disp(N)

n = -M:M;
lp = (wc/pi)*sinc((wc/pi)*n);
wh = chebwin(N,42);
b = lp.*wh';

figure(1);
k = 0:2*M;
stem(k,b);
title('Impulse Response Coefficients');
xlabel('Time index n');ylabel('Amplitude');

figure(2);
[h, w] = freqz(b,1,512);
plot(w/pi, 20*log10(abs(h)));grid;
xlabel('\omega/\pi');ylabel('Gain, in dB');
title('Lowpass filter designed using D-C window');
```

The resulting responses are as follows:



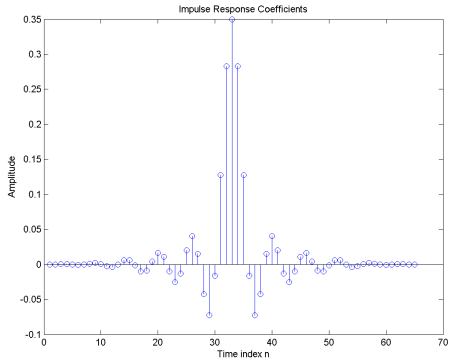


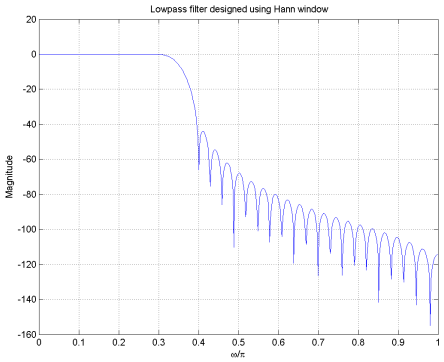
M10.12 Again, the parameters are the same as in Problem M10.10, but we use the built-in `fir1` function:

```
% Problem M10.12
wp = 0.3*pi;
ws = 0.4*pi;
wc = (wp + ws)/2;
dw = ws - wp;

M = ceil(3.11*pi/dw);
N = 2*M + 1;
```

```
disp(M)

n = -M:M;
b = fir1(2*M, wc/pi, hanning(N));

figure(1);
stem(b);
title('Impulse Response Coefficients');
xlabel('Time index n');ylabel('Amplitude');

figure(2);
[h, w] = freqz(b,1,512);
plot(w/pi, 20*log10(abs(h)));
grid;
xlabel('\omega/\pi');ylabel('Magnitude');
title(' Lowpass filter designed using fir1.m');
```

The generated plots are shown below:



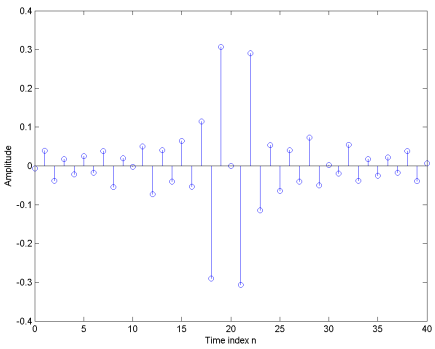


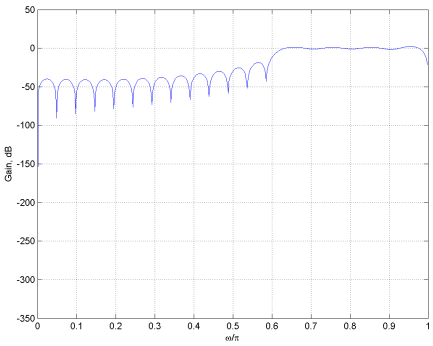
M10.13 The code for this problem is shown below:

```
% Problem M10.13
clear;
N = 40;
wp = 0.6*pi;

for k = 1:N+1,
    w(k) = 2*pi*(k-1)/(N+1);

    if(w(k) >= wp && w(k) <= (pi+wp))
        H(k) = 1;
    else
        H(k) = 0;
    end

    if (w(k) <= pi)
```

```
        phase(k) = 1i*exp(-1i*w(k)*N/2);
    else
        phase(k) = -1i*exp(1i*(2*pi-w(k))*N/2);
    end
end
H = H.*phase;
f = ifft(H);
[FF, w] = freqz(f, 1, 512);
k = 0:N;

figure(1);
stem(k, real(f));
xlabel('Time index n');ylabel('Amplitude');

figure(2);
plot(w/pi,20*log10(abs(FF)));grid
xlabel('\omega/\pi');ylabel('Gain, dB');
```

The corresonding plots are shown below:



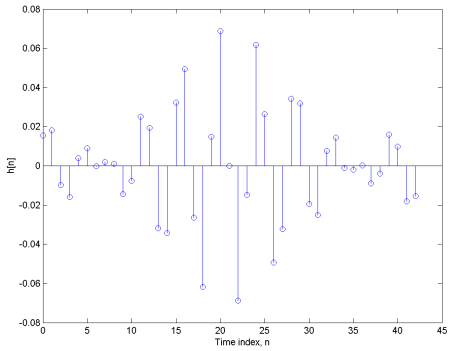


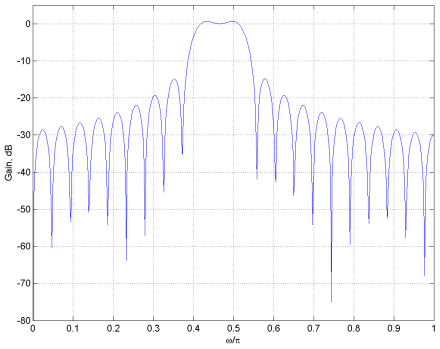
M10.14 The code for this problem is shown below:

```
% Problem M10.14
clear;
N = 42;
wp1 = 0.4*pi;
wp2 = 0.55*pi;

L = N + 1;
for k = 1:L,
    w = 2*pi*(k-1)/L;
    if (w >= wp1 && w <= wp2)
        H(k) = 1i*exp(-1i*w*N/2);
    elseif (w >= (2*pi - wp2) && w <= (2*pi - wp2))
        H(k) = -1i*exp(1i*(2*pi-w)*N/2);
    else
        H(k) = 0;
```

```
    end
end
f = ifft(H);
[FF, w] = freqz(f, 1, 512);
k = 0:N;

figure(1);
stem(k, real(f));
xlabel('Time index, n');ylabel('h[n]');

figure(2);
plot(w/pi, 20*log10(abs(FF)));grid;
ylabel('Gain, dB');xlabel('\omega/\pi');
axis([0 1 -80 5]);
```

The corresonding plots are shown below:



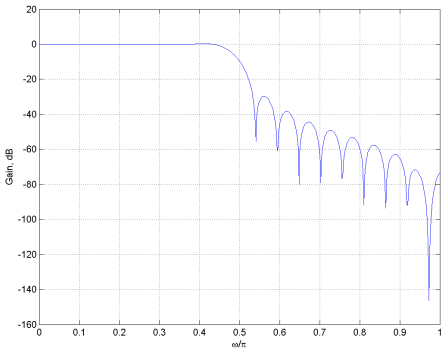


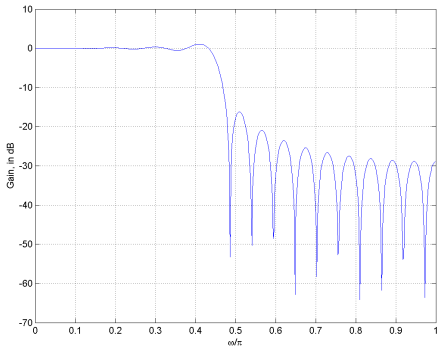
M10.15 The code for this problem is shown below:

```
% Problem M10.15
clear;
N = 37;
wc = 0.45*pi;
M = ((N-1)/2);
ind = 1;
for k = 0:floor(M*wc/pi)
    H(ind) = exp(-1i*2*pi*M*k/N);
    ind = ind + 1;
end
k = ceil(M*wc/pi);
H(ind) = 0.5*exp(-1i*2*pi*M*k/N);
ind = ind + 1;
for k = ceil(M*wc/pi)+1:N-1-ceil(M*wc/pi)-1
    H(ind) = 0;
    ind = ind + 1;
end
k = N-1-ceil(M*wc/pi);
```

```
H(ind) = 0.5*exp(-1i*2*pi*M*k/N);
ind = ind + 1;
for k = N-1-floor(M*wc/pi):N-1
    H(ind) = exp(-1i*2*pi*M*k/N);
    ind = ind + 1;
end

h = ifft(H);
[FF, w] = freqz(h, 1, 512);
plot(w/pi, 20*log10(abs(FF)));grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
```

The corresonding plots are shown below, with the plot on the right being a reference from the solution of Problem 10.33:



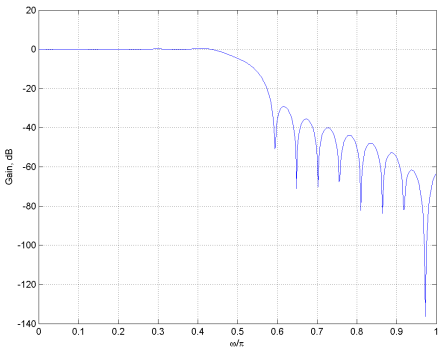


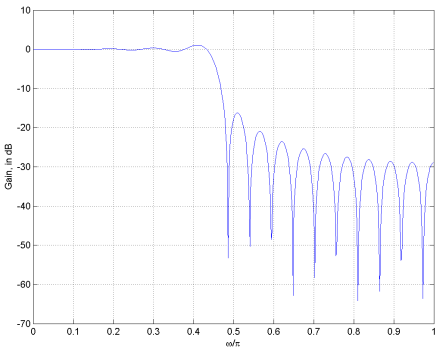
M10.16 The code for this problem is shown below:

```
% Problem M10.15
clear;
N = 37;
wc = 0.45*pi;
M = ((N-1)/2);
ind = 1;
for k = 0:floor(M*wc/pi)
    H(ind) = exp(-1i*2*pi*M*k/N);
    ind = ind + 1;
end
k = ceil(M*wc/pi);
H(ind) = (2/3)*exp(-1i*2*pi*M*k/N);
ind = ind + 1;
k = ceil(M*wc/pi)+1;
H(ind) = (1/3)*exp(-1i*2*pi*M*k/N);
ind = ind + 1;
for k = ceil(M*wc/pi)+2:N-1-ceil(M*wc/pi)-2
    H(ind) = 0;
    ind = ind + 1;
end
```

```
k = N-1-ceil(M*wc/pi)-1;
H(ind) = (1/3)*exp(-1i*2*pi*M*k/N);
ind = ind + 1;
k = N-1-ceil(M*wc/pi);
H(ind) = (2/3)*exp(-1i*2*pi*M*k/N);
ind = ind + 1;

for k = N-1-floor(M*wc/pi):N-1
    H(ind) = exp(-1i*2*pi*M*k/N);
    ind = ind + 1;
end

h = ifft(H);
[FF, w] = freqz(h, 1, 512);
plot(w/pi, 20*log10(abs(FF)));grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
```

The corresponding plots are shown below, with the plot on the right being a reference from the solution of Problem 10.33:





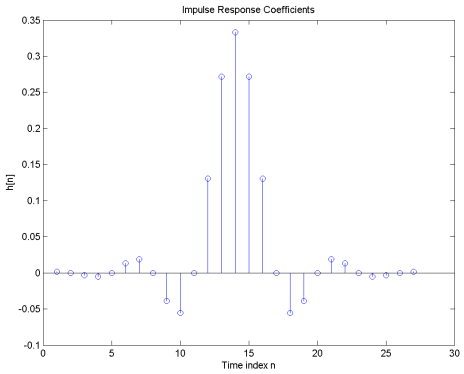
M10.17 The code for this problem is shown below:

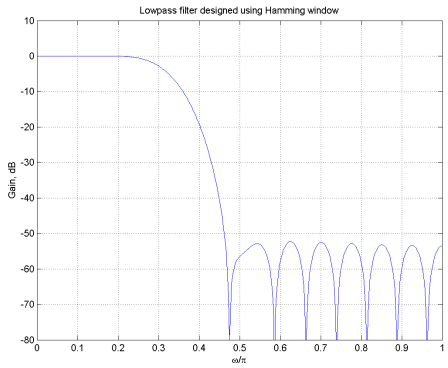
```
% Problem M10.17
clear;
fp = 3;
fs = 5;
Fs = 15;
as = 40;

wp = (fp*pi)/(Fs/2);
ws = (fs*pi)/(Fs/2);
wc = (wp + ws)/2;
dw = ws - wp;

% Hamming
M = ceil(3.32*pi/dw);
N = 2*M;
```

```
b = fir1(N, ws/(2*pi));
[H, w] = freqz(b,1,512);
figure(1);
stem(b);
title('Impulse Response Coefficients');
xlabel('Time index n');ylabel('h[n]');
figure(2);
plot(w/pi, 20*log10(abs(H)));grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
title('Lowpass filter designed using Hamming window');
axis([0 1 -80 10]);

% Hann
M = ceil(3.11*pi/dw);
N = 2*M;
b = fir1(N, ws/(2*pi), hanning(N+1));
[H, w] = freqz(b,1,512);
figure(3);
stem(b);
title('Impulse Response Coefficients');
xlabel('Time index n');ylabel('h[n]');
figure(4);
plot(w/pi, 20*log10(abs(H)));grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
title('Lowpass filter designed using Hann window');
axis([0 1 -80 10]);

% Blackman
M = ceil(5.56*pi/dw);
N = 2*M;
b = fir1(N, ws/(2*pi), blackman(N+1));
[H, w] = freqz(b,1,512);
figure(5);
stem(b);
title('Impulse Response Coefficients');
xlabel('Time index n');ylabel('h[n]');
figure(6);
plot(w/pi, 20*log10(abs(H)));
grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
title('Lowpass filter designed using Blackman window');
axis([0 1 -80 10]);

% Kaiser
beta = 0.5842*(as-21)^0.4 + 0.07886*(as-21);
N = (as - 8)/(2.285*dw);
N = round(N) + mod(round(N),2);
b = fir1(N, ws/(2*pi), kaiser(N+1, beta));
```

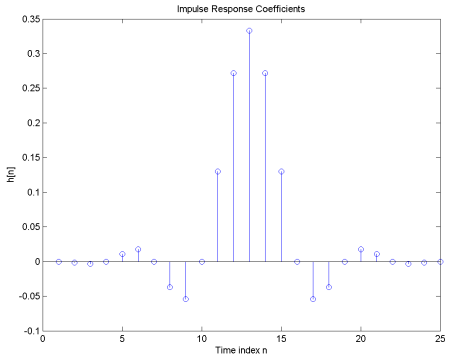
```
[H, w] = freqz(b,1,512);
figure(7);
stem(b);
title('Impulse Response Coefficients');
xlabel('Time index n');ylabel('h[n]');
figure(8);
plot(w/pi, 20*log10(abs(H)));grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
title('Lowpass filter designed using Kaiser window');
axis([0 1 -80 10]);
```

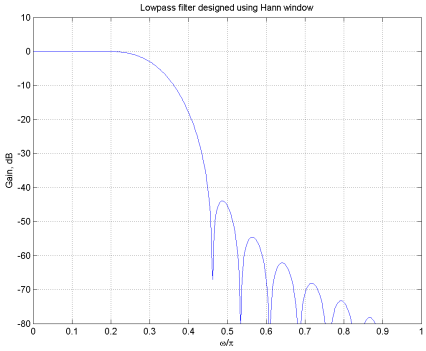
The corresponding plots are shown below:



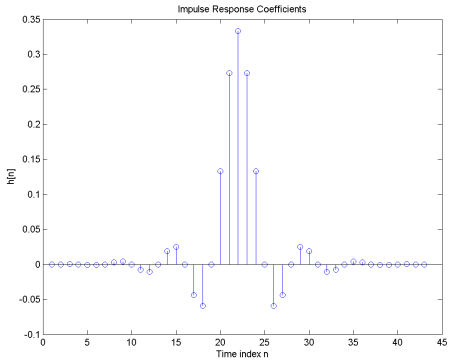


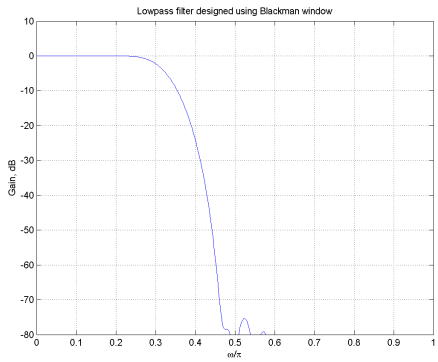
Hamming Window Responses



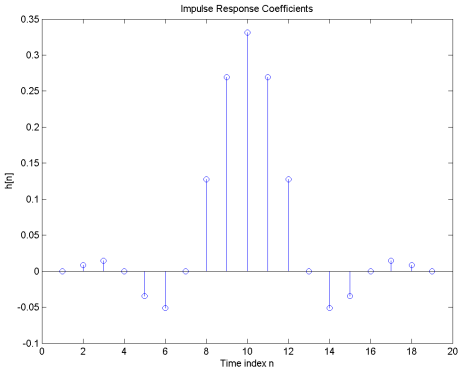


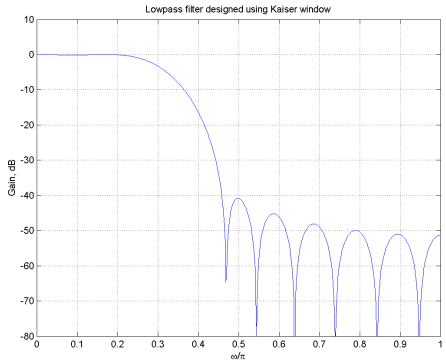
Hann Window Responses





Blackman Window Responses





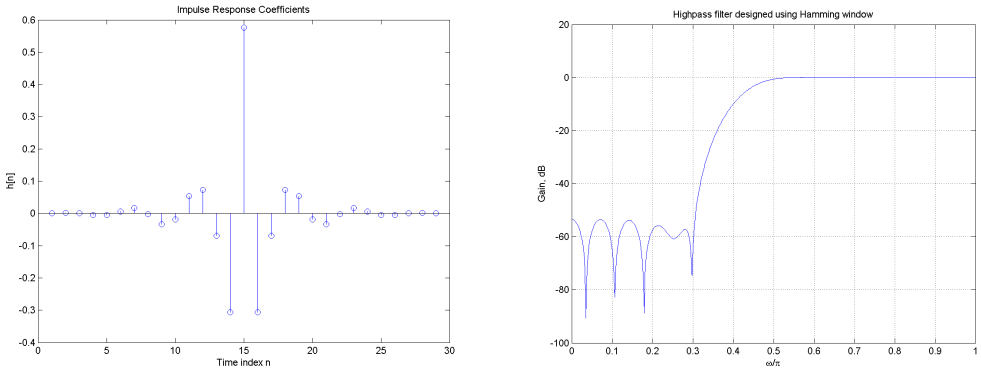
Kaiser Window Responses

M10.18 The code for this problem is shown below:

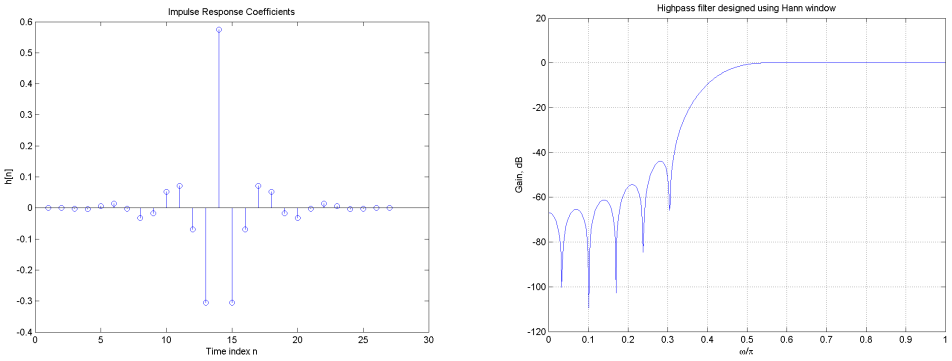
```
% Problem M10.18
clear;
wp = 0.55*pi;
ws = 0.3*pi;
wc = (wp + ws)/2;
dw = wp - ws;
as = 42;

% Hamming
M = ceil(3.32*pi/dw);
N = 2*M;
b = fir1(N,(wc/pi),'high',hamming(N+1));
[H, w] = freqz(b,1,512);
figure(1);
stem(b);
title('Impulse Response Coefficients');
xlabel('Time index n');ylabel('h[n]');
figure(2);
plot(w/pi, 20*log10(abs(H)));grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
```

```
title('Highpass filter designed using Hamming window');

% Hann
M = ceil(3.11*pi/dw);
N = 2*M;
b = fir1(N,(wc/pi),'high', hanning(N+1));
[H, w] = freqz(b,1,512);
figure(3);
stem(b);
title('Impulse Response Coefficients');
xlabel('Time index n');ylabel('h[n]');
figure(4);
plot(w/pi, 20*log10(abs(H)));grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
title('Highpass filter designed using Hann window');

% Blackman
M = ceil(5.56*pi/dw);
N = 2*M;
b = fir1(N,(wc/pi),'high', blackman(N+1));
[H, w] = freqz(b,1,512);
figure(5);
stem(b);
title('Impulse Response Coefficients');
xlabel('Time index n');ylabel('h[n]');
figure(6);
plot(w/pi, 20*log10(abs(H)));
grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
title('Highpass filter designed using Blackman window');

Kaiser
ds = 10^(-as/20);
[N,Wn,beta,type] = kaiserord([(ws/pi) (wp/pi)],[1 0],[ds
ds]);
N = N + 1;
b = fir1(N,(wc/pi),'high',kaiser(N+1,beta)');
[H,w] = freqz(b,1,512);
figure(7);
stem(b);
title('Impulse Response Coefficients');
xlabel('Time index n');ylabel('h[n]');
figure(8);
plot(w/pi, 20*log10(abs(H)));grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
title('Highpass filter designed using Kaiser window');
```

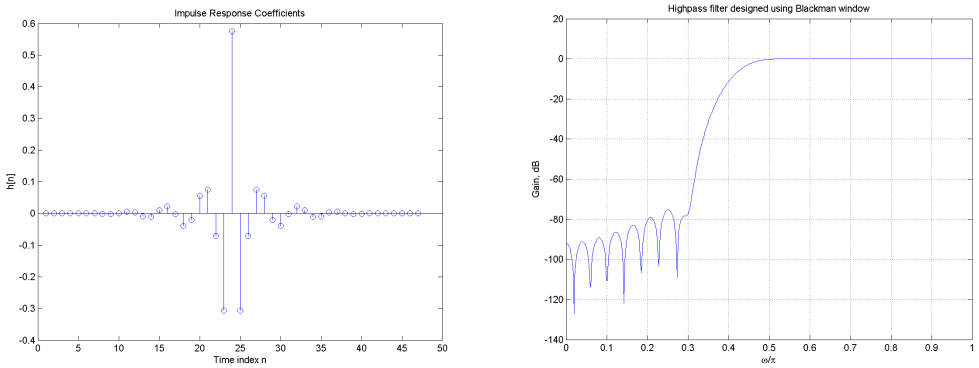
The corresponding plots are shown below:



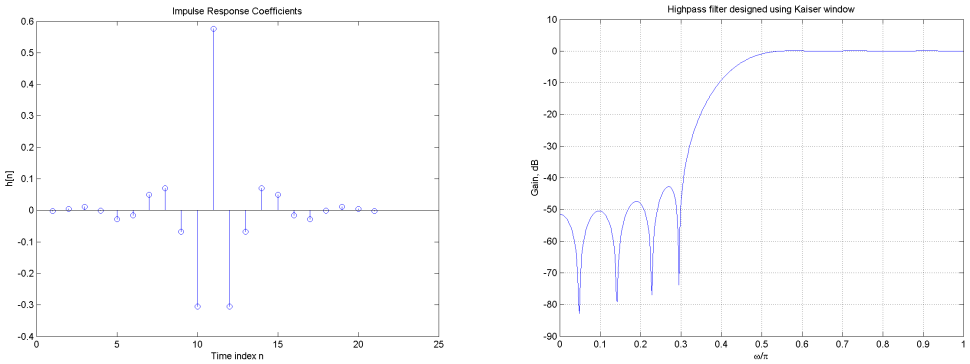
Hamming Window Responses



Hann Window Responses



Blackman Window Responses



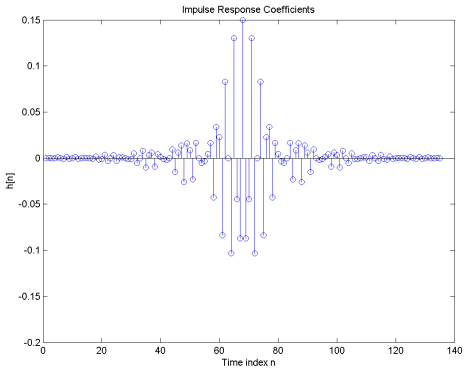
Kaiser Window Responses

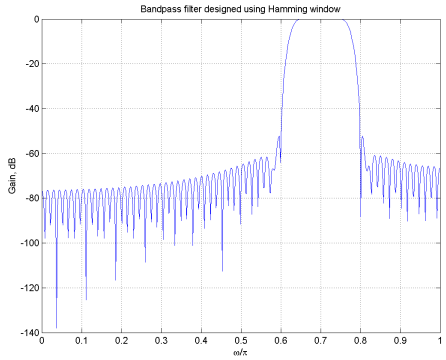
M10.19 The code for this problem is shown below:

```
% Problem M10.19
wp1 = 0.65*pi;
wp2 = 0.75*pi;
ws1 = 0.6*pi;
ws2 = 0.8*pi;
ap = 0.2;
as = 42;
dw1 = wp1 - ws1;
dw2 = ws2 - wp2;
dw = min(dw1,dw2);
wc1 = mean([ws1 wp1]);
wc2 = mean([ws2 wp2]);

% Hamming
N = 2*ceil((3.32*pi)/dw);
b = fir1(N, [wc1/pi wc2/pi]);
[H, w] = freqz(b,1,512);
figure(1);
stem(b);
title('Impulse Response Coefficients');
xlabel('Time index n');ylabel('h[n]');
figure(2);
plot(w/pi, 20*log10(abs(H)));grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
title('Bandpass filter designed using Hamming window');

% Hann
N = 2*ceil((3.11*pi)/dw);
b = fir1(N, [wc1/pi wc2/pi],hanning(N+1));
[H, w] = freqz(b,1,512);
figure(3);
stem(b);
title('Impulse Response Coefficients');
```

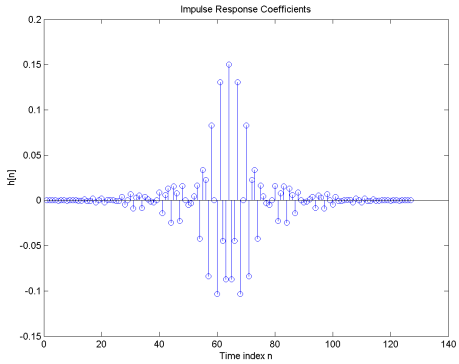
```
xlabel('Time index n');ylabel('h[n]');
figure(4);
plot(w/pi, 20*log10(abs(H)));grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
title('Bandpass filter designed using Hann window');

% Blackman
N = 2*ceil((5.56*pi)/dw);
b = fir1(N, [wc1/pi wc2/pi],blackman(N+1));
[H, w] = freqz(b,1,512);
figure(5);
stem(b);
title('Impulse Response Coefficients');
xlabel('Time index n');ylabel('h[n]');
figure(6);
plot(w/pi, 20*log10(abs(H)));grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
title('Bandpass filter designed using Blackman window');

% Kaiser
ds = 10^(-as/20);
dp = 10^(-ap/20);
[N, Wn, beta, type] = kaiserord([wc1/pi wc2/pi], [1 0], [dp
ds]);
b = fir1(2*N, [wc1/pi wc2/pi], kaiser(2*N+1, beta));
[H, w] = freqz(b,1,512);
figure(7);
stem(b);
title('Impulse Response Coefficients');
xlabel('Time index n');ylabel('h[n]');

figure(8);
plot(w/pi, 20*log10(abs(H)));grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
title('Bandpass filter designed using Kaiser window');
```

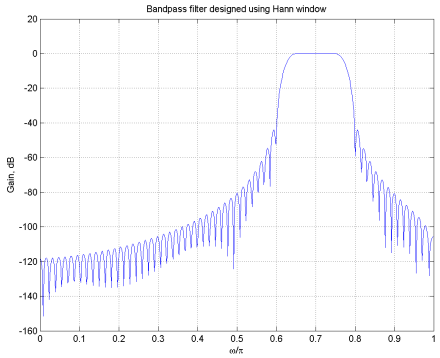
The corresponding plots are shown below:



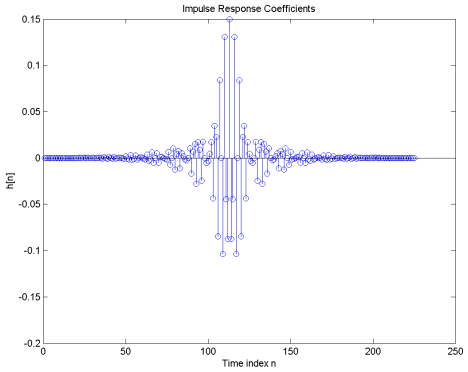


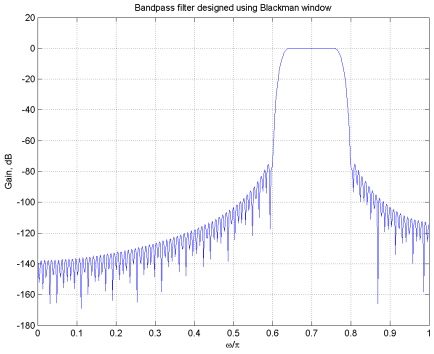
Hamming Window Responses



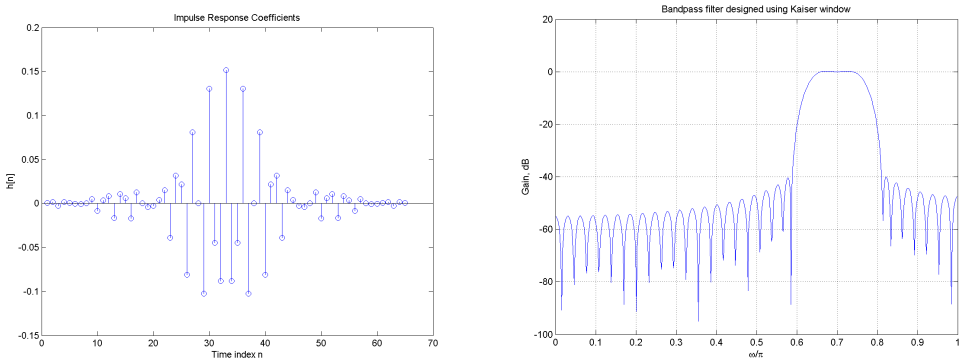


Hann Window Responses





Blackman Window Responses

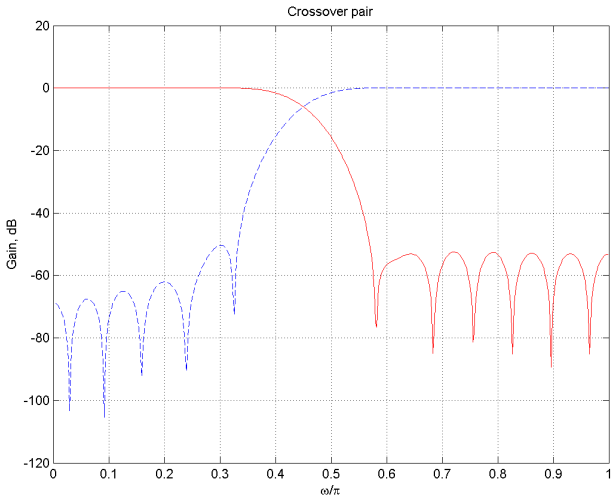


Kaiser Window Responses

M10.20 The code for this problem is shown below:

```
% Problem M10.20
Fc = 18;
Fs = 80;
N = 29;
wc = pi*(Fc/Fs);
d1 = fir1(N-1,wc);
d2 = fir1(N-1,wc,'high');
[h1,w] = freqz(d1,1,512);
h2 = freqz(d2,1,w);
plot(w/pi,20*log10(abs(h1)),'-r',...
    w/pi,20*log10(abs(h2)),'--b');
grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
title('Crossover pair');
```

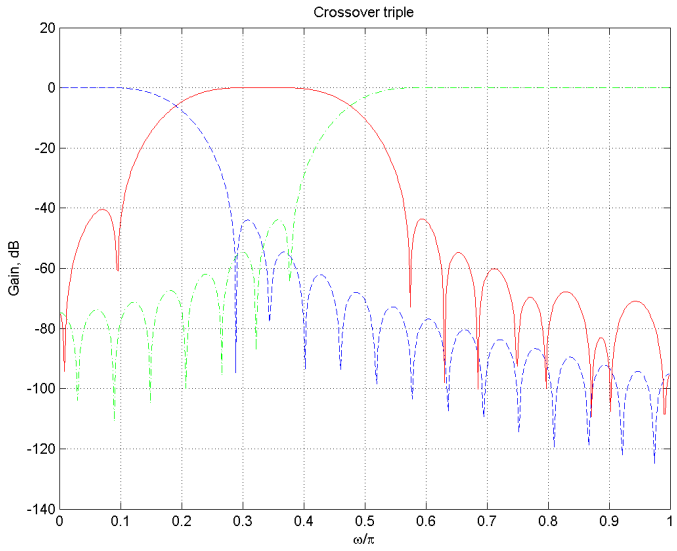
The corresponding plot is shown below:



M10.21 The code for this problem is shown below:

```
% Problem M10.21
Fc1 = 4.2;
Fc2 = 10.5;
Fs = 44.1;
N = 32;

wc1 = 2*pi*(Fc1/Fs);
wc2 = 2*pi*(Fc2/Fs);
d1 = fir1(N, (wc1/pi), hanning(N+1));
d2 = fir1(N, (wc2/pi), 'high', hanning(N+1));
d3 = -d1-d2;
d3(17) = 1-d1(17)-d2(17);
[h1,w] = freqz(d1,1,512);
h2 = freqz(d2,1,w);
h3 = freqz(d3,1,w);
g1 = 20*log10(abs(h1));
g2 = 20*log10(abs(h2));
g3 = 20*log10(abs(h3));
plot(w/pi, g1,'--b',w/pi,g2,'-.g',w/pi,g3,'-r');grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
title('Crossover triple');
```

The corresponding plot is shown below:

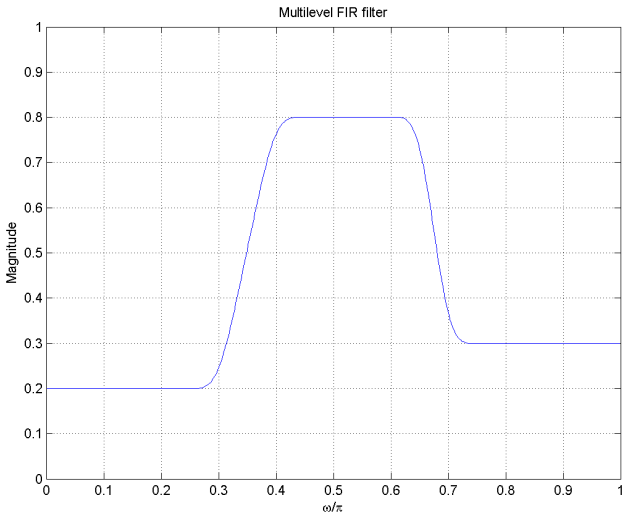


M10.22 The code for this problem is shown below:

```
% Problem M10.22
fpts = [0 0.3 0.4 0.65 0.7 1];
mval = [0.2 0.2 0.8 0.8 0.3 0.3];
```

```
b = fir2(80, fpts, mval);
[H,w] = freqz(b,1,512);
figure(1);
plot(w/pi,abs(H));grid;
xlabel('\omega/\pi');ylabel('Magnitude');
title('Multilevel FIR filter');
axis([0 1 0 1]);
```

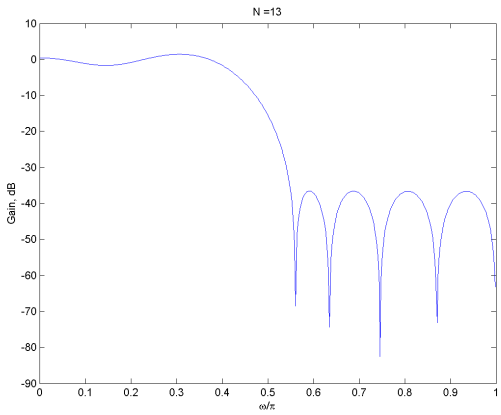
The corresponding plot is shown below:



M10.23 The code for this problem is shown below:

```
% Program M10.23
Ft = 2;
Fp = 0.4;
Fs = 0.55;
ds = 0.015;
dp = 0.18;
F = [Fp Fs];
A = [1 0];
DEV = [dp ds];
[N,Fo,Ao,W] = remezord(F,A,DEV,Ft);
b = remez(N,Fo,Ao,W);
[H,w] = freqz(b,1,512);
figure(1);
plot(w/pi, 20*log10(abs(H)));
xlabel('\omega/\pi');ylabel('Gain, dB');
title(strcat('N = ',num2str(N)));
```

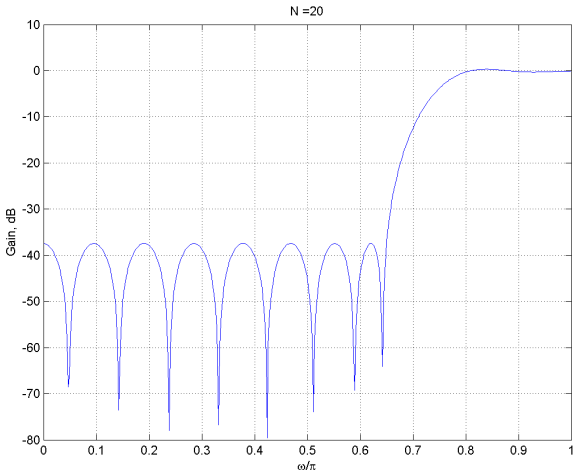
The corresonding plot is shown below:



M10.24 The code for this problem is shown below:

```
% Program M10.24
Ft = 2;
Fp = 0.8;
Fs = 0.65;
ds = 0.015;
as = -20*log10(ds)
dp = 0.04;
F = [Fs Fp];
A = [0 1];
DEV = [ds dp];
[N,Fo,Ao,W] = remezord(F,A,DEV,Ft);
b = remez(N,Fo,Ao,W);
[H,w] = freqz(b,1,512);
figure(1);
plot(w/pi, 20*log10(abs(H)));grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
title(strcat('N = ',num2str(N)));
```

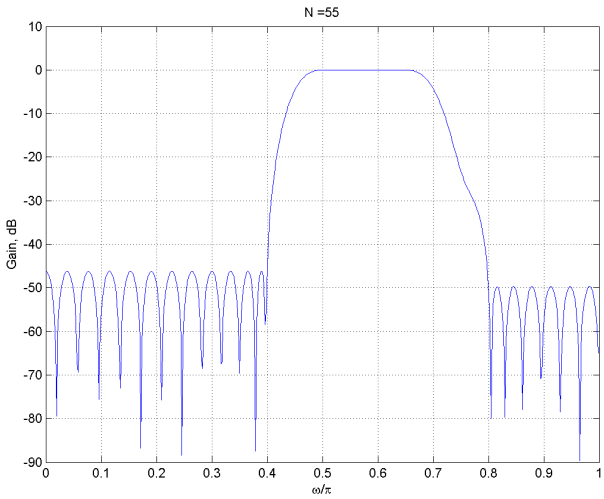
The corresponding plot is shown below. From the problem specification and plot, we can see that the automatically generated design satisfies the stopband attenuation requirement of 36.4782 dB.



M10.25 The code for this problem is shown below:

```
% Program M10.25
Ft = 2;
Fp1 = 0.5;
Fp2 = 0.65;
Fs1 = 0.4;
Fs2 = 0.8;
ds1 = 0.006;
as1 = -20*log10(ds1);
ds2 = 0.004;
as2 = -20*log10(ds2);
dp = 0.001;
disp([as1 as2])
F = [Fs1 Fp1 Fp2 Fs2];
A = [0 1 0];
DEV = [ds1 dp ds2];
[N,Fo,Ao,W] = remezord(F,A,DEV,Ft);
b = remez(N,Fo,Ao,W);
[H, w] = freqz(b, 1, 512);
figure(1);
plot(w/pi,20*log10(abs(H)));grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
title(strcat('N = ',num2str(N)));
```

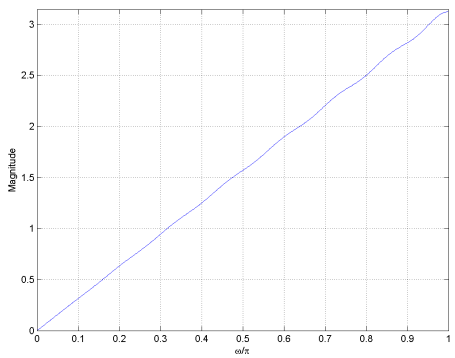
The corresponding plot is shown below. Again, from the problem specification and plot, we can see that the automatically generated design satisfies the stopband attenuation requirements of 44.437 dB and 47.9588 dB.

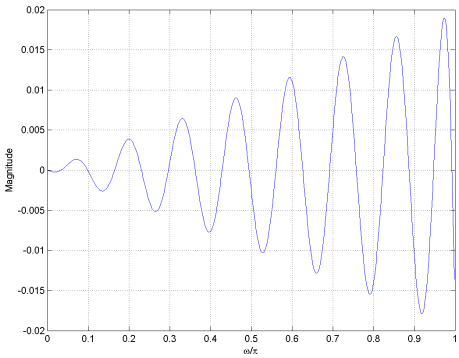


M10.26 The code for this problem is shown below:

```
% Program M10.26
N = 32;
b = remez(N-1, [0 1], [0 pi], 'differentiator');
[H, w] = freqz(b, 1, 512);
figure(1)
plot(w/pi,abs(H)); grid;
xlabel('\omega/\pi');ylabel('Magnitude');
axis([0 1 0 pi]);

figure(2)
plot(w/pi,abs(H)-w); grid;
xlabel('\omega/\pi');ylabel('Magnitude');
```

The corresponding plots are shown below, with the second plot showing the absolute error:



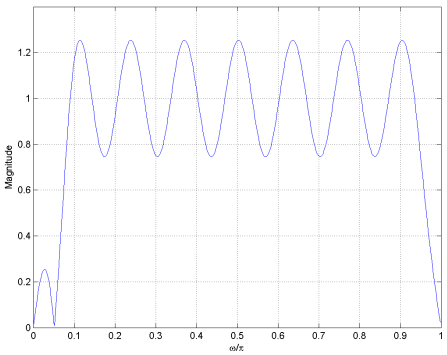


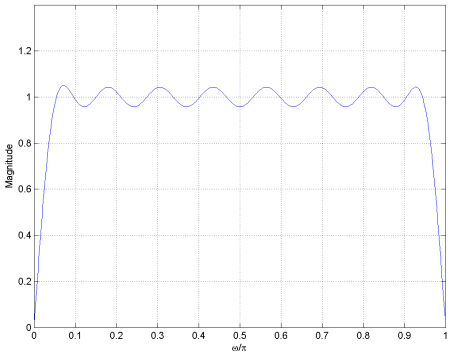
M10.27 The code for this problem is shown below:

```
% Program M10.27
N = 28;
```

```
f = [0.02 0.06 0.08 0.95 0.98 1];
m = [0 0 1 1 0 0];

figure(1)
b = firpm(30, f, m, 'hilbert');
[H,w] = freqz(b,1,512);
plot(w/pi,abs(H));grid
xlabel('\omega/\pi');ylabel('Magnitude');
axis([0 1 0 1.4]);

figure(2)
wt = [1 2*N 1];
b = firpm(30, f, m, wt, 'hilbert');
[H,w] = freqz(b,1,512);
plot(w/pi,abs(H));grid
xlabel('\omega/\pi');ylabel('Magnitude');
axis([0 1 0 1.4]);
```

The corresponding plots are shown below. The original filter design leaves significant passband ripples due to the low order of the filter. This can be remedied by including weighting functions for the passband coefficients, as shown in the second plot.



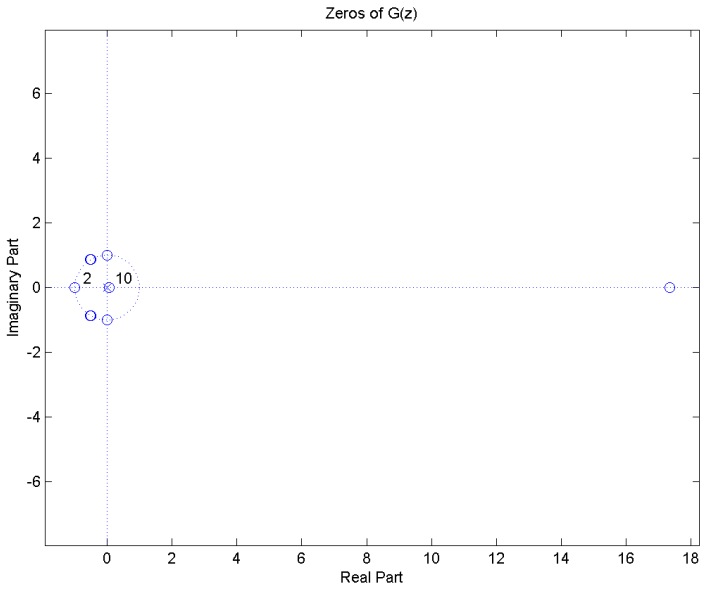


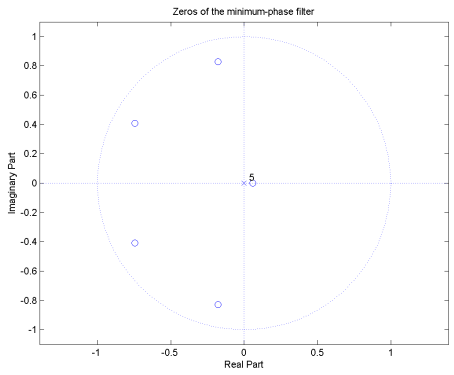
M10.28 The code for this problem is shown below (Note: this code requires the use of the `minphase` function, which is included in the CD):

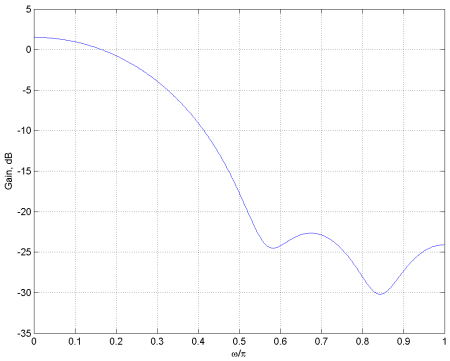
```
% Program M10_28

% Design of a minimum-phase lowpass FIR filter
Wp = 0.25; Ws = 0.5; Rp = 1; Rs = 25;
% Desired ripple values of minimum-phase filter
dp = 1- 10^(-Rp/20); ds = 10^(-Rs/20);

% Compute ripple values of prototype linear-phase
filter
Ds = (ds*ds)/(2 - ds*ds);
Dp = (1 + Ds)*((dp + 1)*(dp + 1) - 1);
```

```
% Estimate filter order
[N,fpts,mag,wt] = remezord([Wp Ws], [1 0], [Dp Ds]);

% Design the prototype linear-phase filter H(z)
[b,err,res] = remez(N,fpts,mag,wt);
K = ceil(N/2);
b1 = b(1:K);

% Design the linear-phase filter G(z)
lenerr = res.error(length(res.error));
c = [b1 (b(K+1) + lenerr) fliplr(b1)]/(1+Ds);

% Plot All Zeroes
figure(1)
zplane(c);title('Zeros of G(z)')
c1 = c(K+1:N+1);
[y, ssp, iter] = minphase(c1);

% Plot Minimum Phase Filter Zeroes
figure(2)
zplane(y);title('Zeros of the minimum-phase filter');
[hh,w] = freqz(y,1, 512);

% Plot the gain response of the minimum-phase filter
figure(3)
plot(w/pi, 20*log10(abs(hh)));grid
xlabel('\omega/\pi');ylabel('Gain, dB');
```

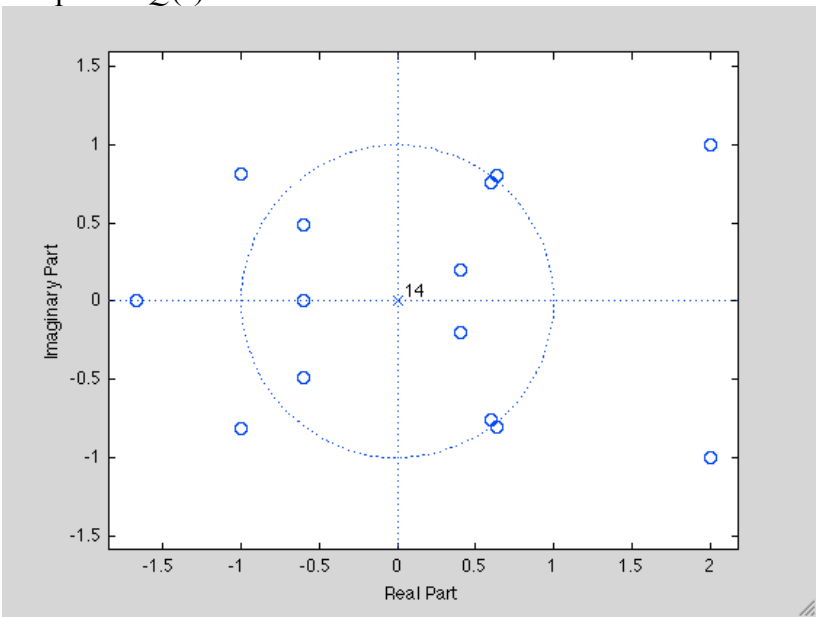
The corresponding plots are shown below:







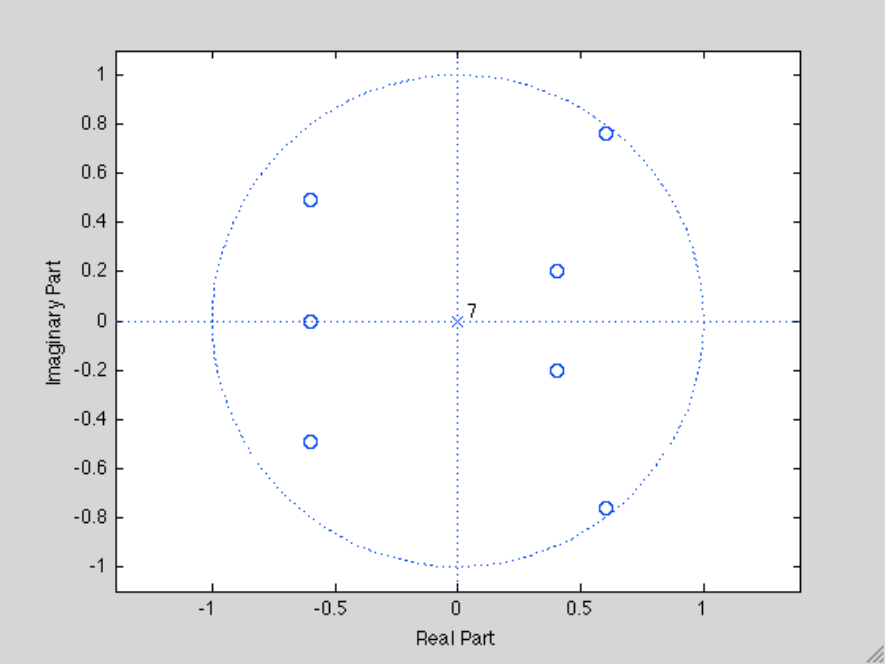
M10.29 The zero-plot of $Q(z)$ is shown below:



The minimum-phase factor obtained using the function `firminphase.m` is given by

$$Q_{\min}(z) = 1.0322 - 0.215z^{-1} - 0.1881z^{-2} + 0.5261z^{-3} + 0.4691z^{-4} - 0.2222z^{-5} - 0.1124z^{-6} + 0.0698z^{-7}.$$

The zero-plot of $Q_{\min}(z)$ is shown below:



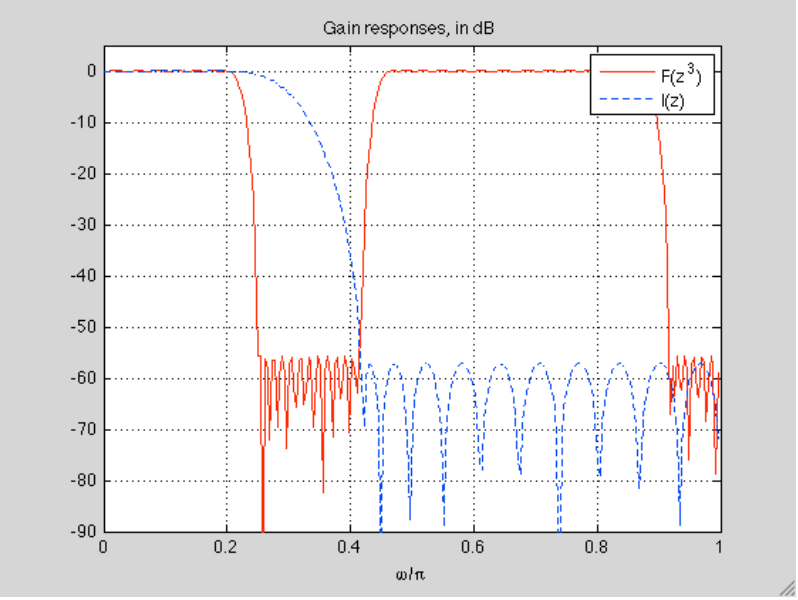
M10.30 The specifications given are: $\omega_p = 0.2\pi$, $\omega_s = 0.25\pi$, $\delta_p = 0.002$, $\delta_s = 0.002$.

We first determine the value of the sparsity factor L, which is determined satisfying the relation $\omega_s < \dfrac{2\pi}{L} - \omega_s$, or equivalently, $L < \dfrac{\pi}{\omega_s}$ or $L < 4$. We choose $L = 3$.

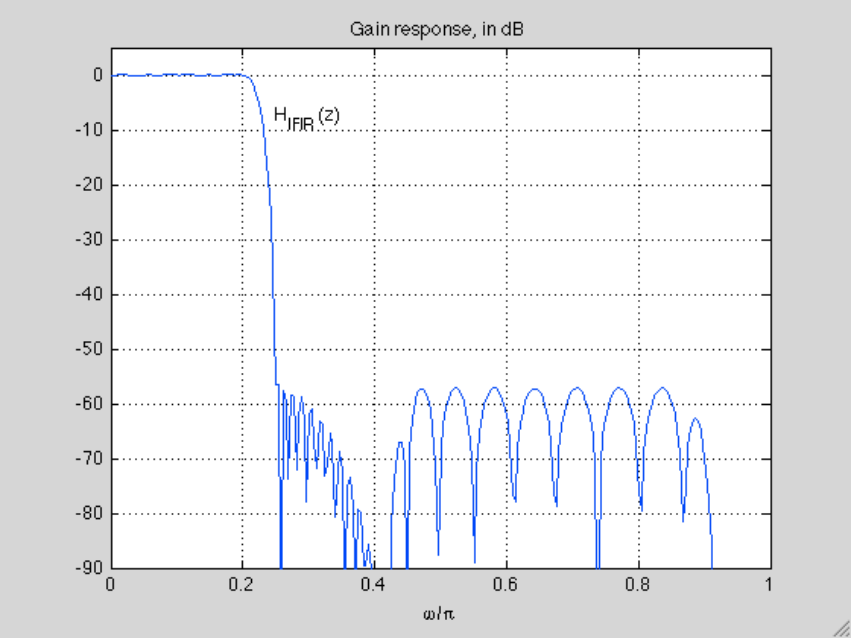
The MATLAB code for designing the IFIR filter $H_{IFIR}(z) = F(z^3)I(z)$ is given below:

```
% Program #M10.30
[h,g]=ifir(3,'low',[.2 .25],[.002 .002]);
[hh,w]=freqz(h,1,256); hg=freqz(g,1,256);
h = hh.*hg; % Compounded response
Fg = 20*log10(abs(hh)); Ig = 20*log10(abs(hg));
plot(w/pi,Fg,'-r',w/pi,Ig,'--b'); grid;
axis([0 1 -90 5]);
legend('F(z^3)','I(z)');
xlabel('\omega/\pi');title('Gain responses, in dB');
pause;
plot(w/pi,20*log10(abs(h))); grid;
axis([0 1 -90 5]);
xlabel('\omega/\pi');title('Gain response, in dB');
gtext('H_{IFIR}(z)');
```

A plot of the gain response of $F(z^3)$ and that of $I(z)$ are shown below:



Finally, the plot of the gain response of $H_{IFIR}(z) = H(z^3)I(z)$ is shown below:

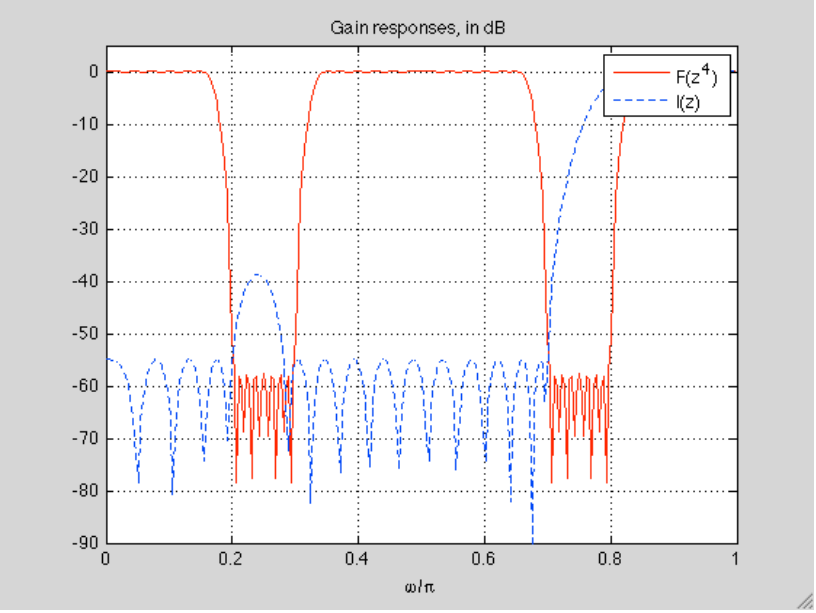


M10.31 The code for this problem is shown below:

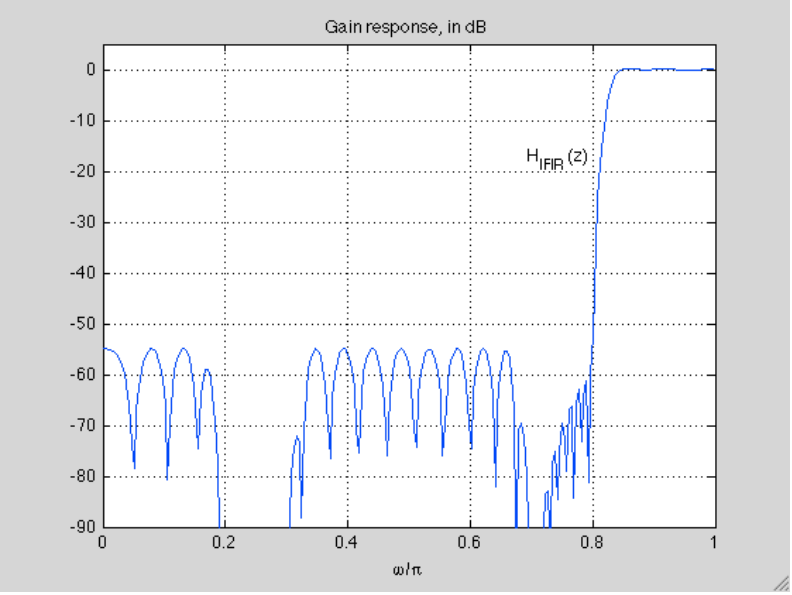
```
% Program #M10.31
[h,g]=ifir(4,'high',[.8 .85],[.002 .002]);
[hh,w]=freqz(h,1,256); hg=freqz(g,1,256);
h = hh.*hg; % Compounded response
Fg = 20*log10(abs(hh)); Ig = 20*log10(abs(hg));
```

```
plot(w/pi,Fg,'-r',w/pi,Ig,'--b'); grid;
axis([0 1 -90 5]);
legend('F(z^4)','I(z)');
xlabel('\omega/\pi');title('Gain responses, in dB');
pause;
plot(w/pi,20*log10(abs(h))); grid;
axis([0 1 -90 5]);
xlabel('\omega/\pi');title('Gain response, in dB');
gtext('H_{IFIR}(z)');
```

A plot of the gain response of $F(z^4)$ and that of $I(z)$ are shown below:



Finally, the plot of the gain response of $H_{IFIR}(z) = H(z^4)I(z)$ is shown below:



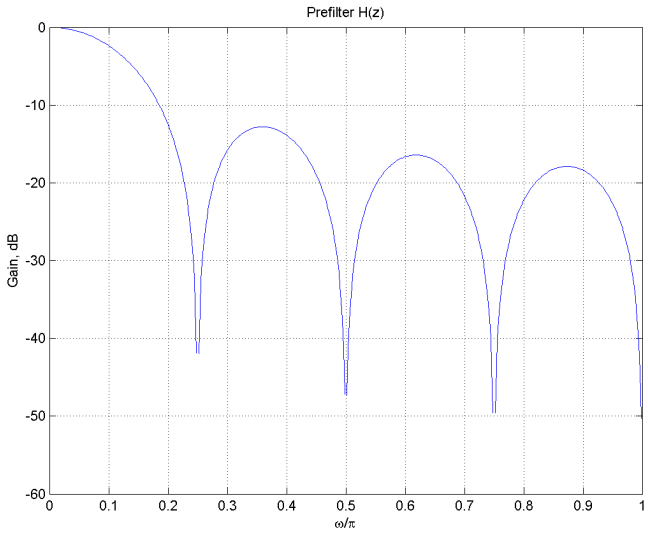
M10.32 The code for this problem is shown below, and is written in function format so as to reuse necessary portions of the subroutines:

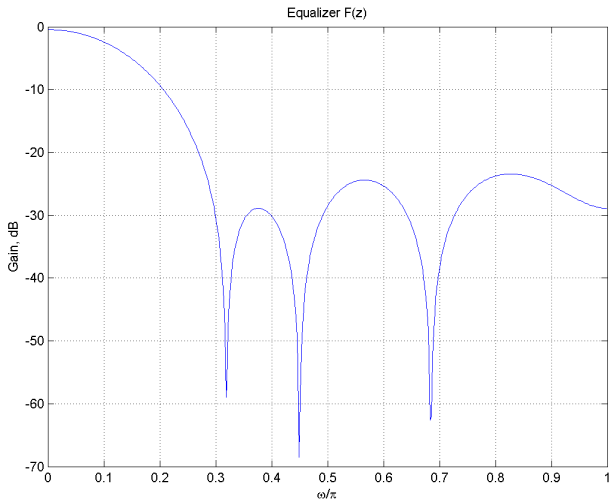
```
function [] = probm10_32()
% Problem M10.32
wp = 0.1*pi;
ws = 0.25*pi;
ap = 0.15;
as = 45;

% Plot the result of using an equalizer of length InpN
% Creating filters
InpN = ceil(2*pi/ws);
Wfilt = ones(1, InpN);
Efilt = remezfunc(InpN, Wfilt);

% Plot running sum filter response
figure(1);
Wfilt = Wfilt/sum(Wfilt);
[hh, w] = freqz(Wfilt, 1, 512);
plot(w/pi, 20*log10(abs(hh))); grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
title('Prefilter H(z)');

% Plot equalizer filter response
figure(2);
[hw, w] = freqz(Efilt, 1, 512);
plot(w/pi, 20*log10(abs(hw))); grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
title('Equalizer F(z)');

% Plot cascade filter response
figure(3);
Cfilt = conv(Wfilt, Efilt);
[hc, w] = freqz(Cfilt, 1, 512);
plot(w/pi, 20*log10(abs(hc))); grid;
xlabel('\omega/\pi');ylabel('Gain, dB');
title('Cascade filter H(z)F(z)');

end

% function
% Remez function using 1/P(z) as desired amplitude
% and P(z) as weighting
function [N] = remezfunc(Nin, Wfilt);
% Nin : number of tuples in the remez equalizer filter
```

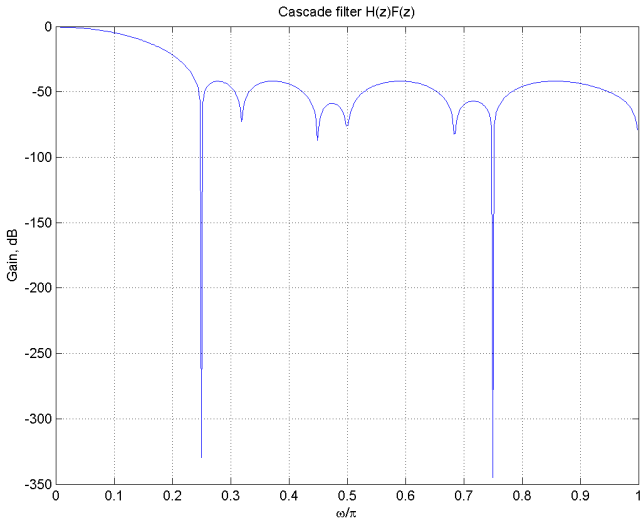
```
% Wfilt : the prefilter
a = [0:0.001:0.999]; % The accuracy of the computation
w = a.*pi;
wp = 0.1*pi;
ws = 0.25*pi;
i = 1;n = 1;
for t = 1:(length(a)/2),
    if w(2*t) < wp
    pas(i) = w(2*t - 1);
    pas(i+1) = w(2*t);
    i = i+2;
    end
    if w(2*t-1) > ws
    sto(n) = w(2*t - 1);
    sto(n+1) = w(2*t);
    n = n+2;
    end
end
    w = cat(2, pas, sto);
    bi = length(w)/2;
    for t1 = 1:bi,
    bw(t1) = (w(2*t1) + w(2*t1-1))/2;
    W(t1) = Weight(bw(t1), Wfilt, ws);
    end
    W = W/max(W);
    for t2 = 1:length(w),
    G(t2) = Hdr(w(t2), Wfilt, wp);
    end
    G = G/max(G);
    N = remez(Nin, w/pi, G, W);

end %function

% Weighting function
function[Wout] = Weight(w, Wfilt, ws);
    K = 22.8;
    L = length(Wfilt);
    Wtemp = 0;
    Wsum = 0;
    for k = 1:L,
    Wtemp = Wfilt(k)*exp((k-1)*i*w);
    Wsum = Wsum + Wtemp;
    end
    Wout = abs(Wsum);
    if w > ws,
    Wout = K*max(Wout);
    end
end %function
```

```
% Desired function
function [Wout] = Hdr(w, Wfilt, ws);
    if w <= ws,
    L = length(Wfilt);
    Wtemp = 0;
    Wsum = 0;
    for k = 1:L,
    Wtemp = Wfilt(k)*exp(i*(k-1)*w);
    Wsum = Wsum + Wtemp;
    end
    Wsum = abs(Wsum);
    Wout = 1/Wsum;
    else
    Wout = 0;
    end
end %function
```

The corresponding plots are shown below:







M10.33 The MATLAB code for this problem is shown below:

```
% Program_10_33.m
w = 0:pi/255:pi;
alpha = -0.5;
[NUM,DEN] = iirftransf(ones(1,10)/10,1,[alpha  1],[1  alpha]);
h1 = freqz(NUM,DEN,w);
alpha = 0;
[NUM,DEN] = iirftransf(ones(1,10)/10,1,[alpha  1],[1  alpha]);
h2 = freqz(NUM,DEN,w);
alpha = 0.5;
[NUM,DEN] = iirftransf(ones(1,10)/10,1,[alpha  1],[1  alpha]);
h3 = freqz(NUM,DEN,w);
plot(w/pi,abs(h1),'-',w/pi,abs(h2),'--',w/pi,abs(h3),'-.');
xlabel('\omega/\pi');ylabel('Magnitude');
axis([0 1 0 1.1]);
```

```
gtext('\alpha = -0.5');gtext('\alpha = ');gtext('\alpha = 0.5');
```

A plot of the magnitude responses of the transformed 10-point moving average filter for the three values of is shown below: