**EECE5666 (DSP) : Homework-7**

**Due on April 19, 2022 by 11:59 pm via submission portal.**

**NAME**: McKean, Tyler



Table of Contents

[Instructions](#MW_H_D1DAE620)   
[Default Plot Parameters](#MW_H_DB1FB1CD)   
[Problem 7.1](#MW_H_2EF0AE58)   
[Problem 7.2](#MW_H_A8C68C7A)   
[Problem 7.3](#MW_H_02E0FFDA)   
[Problem 7.4](#MW_H_934D9E5F)   
[Problem 7.5](#MW_H_7DB25FA7)   
[Problem 7.6](#MW_H_9959998E)   
[Problem 7.7](#MW_H_80952945)   
[Problem 7.8](#MW_H_790820D1)

# Instructions

1. You are required to complete this assignment using Live Editor.
2. Enter your MATLAB script in the spaces provided. If it contains a plot, the plot will be displayed after the script.
3. All your plots must be properly labeled and should have appropriate titles to get full credit.
4. Use the equation editor to typeset mathematical material such as variables, equations, etc.
5. After completeing this assignment, export this Live script to PDF and submit the PDF file through the provided submission portal.
6. You will have only one attempt to submit your assignment. Make every effort to submit the correct and completed PDF file the first time.
7. Please submit your homework before the due date/time. A late submission after midnight of the due date will result in loss of points at a rate of 10% per hour until 8 am the following day, at which time the solutions will be published.

# Default Plot Parameters

set(0,'defaultfigurepaperunits','points','defaultfigureunits','points');

set(0,'defaultaxesfontsize',10); set(0,'defaultaxeslinewidth',1.5);

set(0,'defaultaxestitlefontsize',1.4,'defaultaxeslabelfontsize',1.2);



# Problem 7.1

We want to design a lowpass analog Chebyshev-I filter that has a 0.5 dB or better ripple at 10 Hz and at least 45 dB of attenuation at 20 Hz.

clc; close all; clear;

**(a)** Using the design procedure on Page 640 of the textbook (or that in Example 11.3) obtain the system function in a rational function form.

**Solution:** Follow the following steps. Perform numerical calculations using MATLAB below each step.

**Step-0**: Determine the analog passband ripple parameter  and stopband attenuation parameter 

We can determine the values of parameters  and  by





Ap = 0.5; As = 45; Omegap = 2\*pi\*10; Omegas = 2\*pi\*20;

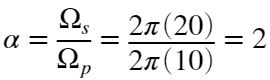
epsilon = sqrt(10^(Ap/10)-1), A = 10^(As/20)

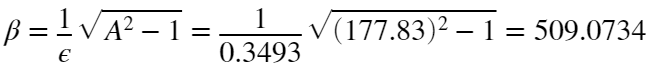
epsilon = 0.3493

A = 177.8279

**Step-1**: Compute the parameters  and  using (11.50):

Using (11.50), we calculate the values of  and  to be



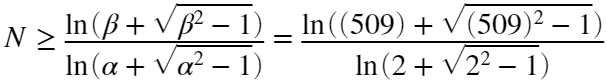


alpha = Omegas/Omegap, beta = sqrt((A^2 -1)/epsilon^2)

alpha = 2

beta = 509.0734

**Step-2:** Compute order  using (11.49) and round upwards to the nearest integer:

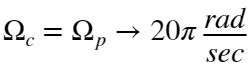


N = ceil(log(beta + sqrt(beta^2 - 1))/(log(alpha + sqrt(alpha^2 - 1))))

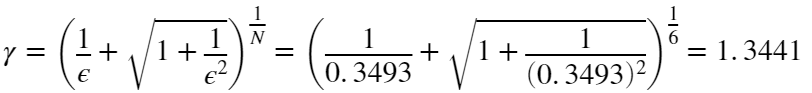
N = 6

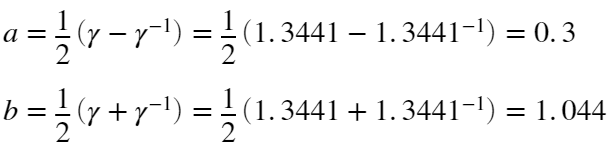
**Step-3:** Set  and compute  and  using (11.44) and (11.45):

We have a passband frequency requirement of 10Hz, thus



We must first calculate the value of  in order to compute values  and 





gamma = (1/epsilon + sqrt(1+1/(0.3493)^2))^(1/N)

gamma = 1.3441

a = 0.5\*(gamma - 1/gamma), b = 0.5\*(gamma + 1/gamma)

a = 0.3000

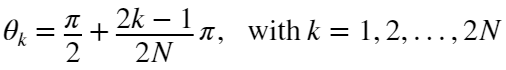
b = 1.0440

**Step-4:** Compute the pole locations using (11.41) and (11.42):

We compute the pole locations using the formulas

 with  

 with   with the angle being



Substituting :





Omegac = Omegap;

% Step-4: Calculations of Poles

k = 1:N; thetak = pi/2+(2\*k-1)\*pi/(2\*N);

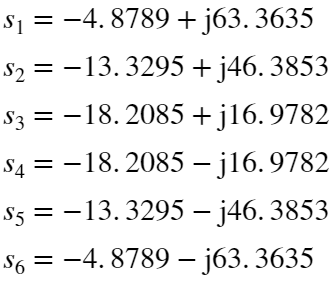
sigmak = (a\*Omegac)\*cos(thetak); Omegak = (b\*Omegac)\*sin(thetak);

sk = cplxpair(sigmak + 1j\*Omegak)

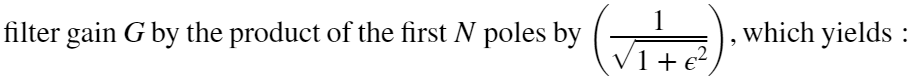
sk = 1×6 complex

-18.2085 -16.9782i -18.2085 +16.9782i -13.3295 -46.3853i -13.3295 +46.3853i ⋯

Taking only the first 6 poles calculated are the left-half elliptic shaped poles of the Chebyshev-I filter, which are



**Step-5:** Compute the filter gain  and the system function  from (11.43): Since  is even,

We can compute the 

Rp = 1/sqrt(1+epsilon^2);

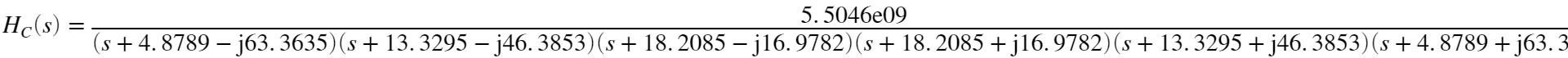
D = real(poly(sk));

G = D(end)\*Rp

G = 5.5046e+09

Thus, 

Finally, determine the system function:



s = tf('s');

Hc = G/((s + abs(sk(1)))\*(s + abs(sk(2)))\*(s + abs(sk(3)))\*(s + abs(sk(4)))\*(s + abs(sk(5)))\*(s + abs(sk(6))))

Hc =

5.505e09

------------------------------------------------------------------------------------

s^6 + 273.4 s^5 + 3.039e04 s^4 + 1.752e06 s^3 + 5.511e07 s^2 + 8.935e08 s + 5.831e09

Continuous-time transfer function.

Thus, the system function 



**(b)** Verify your design using the **cheb1ord** and **cheby1** functions.

**Solution:**

[N,Omegac] = cheb1ord(Omegap,Omegas,Ap,As,'s')

N = 6

Omegac = 62.8319

Fc = Omegac/(2\*pi)

Fc = 10

Thus, using the cheb1ord function yields the same result for  for the order of the filter with the cutoff frequency of

Hz

[C,D] = cheby1(N,Ap,Omegac,'s');



**(c)** Provide plots of impulse response, amplitude response, log-magnitude response in dB, and phase response in one figure using two rows and two columns.

**MATLAB script:**

% Impulse Response

trsys = tf(C,D);

[h t] = impulse(trsys);

% Frequency Response

Fmax = 40; F = linspace(0,Fmax,1001);

omega = 2\*pi\*F;

H = freqs(C,D,omega);

Hmag = abs(H);

Hdb = mag2db(Hmag);

% Phase Response

Hpha = angle(H);

Hgd = -diff(unwrap(Hpha))./diff(omega);

Hgd = [Hgd Hgd(end)];

% Amplitude Response

[Hr,wr] = zerophase(C,D,512,'whole');

% Plot Results

figure('Units','inches','Position',[0,0,12,4]);

% Impulse Response Plot

subplot(2,2,1),plot(t,h,'LineWidth',1.5),title('Impulse Response'),xlabel('Time Index t/s'),ylabel('\it{h(t)}')

xlim([0 t(end)])

% Magnitude Response Plot

subplot(2,2,2),plot(F,Hdb,'LineWidth',1.5),title('Log-Magnitude Response (dB)'), grid on

ylim([-80 10]), yticks([-45 -0.5]), xticks([10 20])

xlabel('Frequency (Hz)'),ylabel('Decibels')

% Amplitude Response Plot

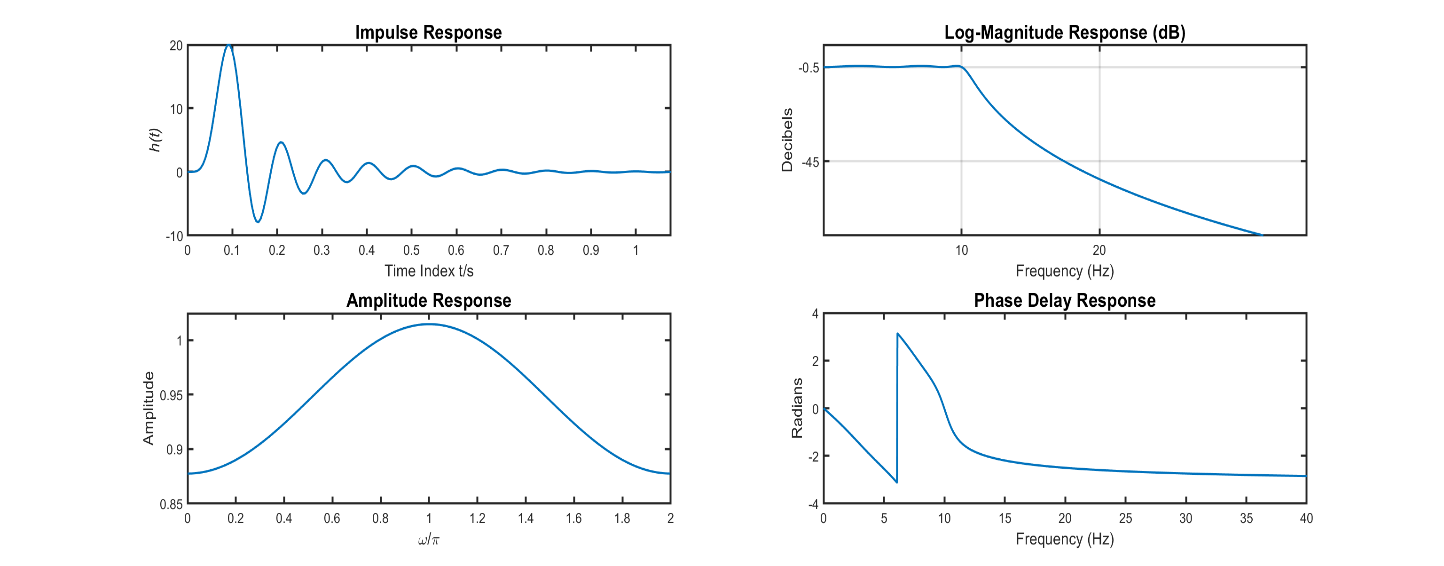
subplot(2,2,3), plot(wr/pi,Hr,'LineWidth',1.5), title('Amplitude Response')

xlabel('\omega/\pi'), ylabel('Amplitude'), ylim([0.85 1.025])

% Phase Response Plot

subplot(2,2,4), plot(F,Hpha,'LineWidth',1.5), title('Phase Delay Response'), xlabel('Frequency (Hz)')

ylabel('Radians')





# Problem 7.2

**Text Problem 11.21 (Page 694)**

Consider a -order analog Butterworth lowpass filter  with -dB cutoff frequency of 10 Hz.



**(a)** Determine and graph pole locations of 

**Solution:** .

clc; close all; clear;

N = 9; Fc = 10; Omegac = 2\*pi\*Fc;

[C,D] = butter(N,Omegac,'s');

sk = roots(D)

sk = 9×1 complex

-10.9106 +61.8773i

-10.9106 -61.8773i

-31.4159 +54.4140i

-31.4159 -54.4140i

-48.1320 +40.3875i

-48.1320 -40.3875i

-62.8319 + 0.0000i

-59.0426 +21.4898i

-59.0426 -21.4898i

Thus, the 9 poles of the Butterworth lowpass filter are:



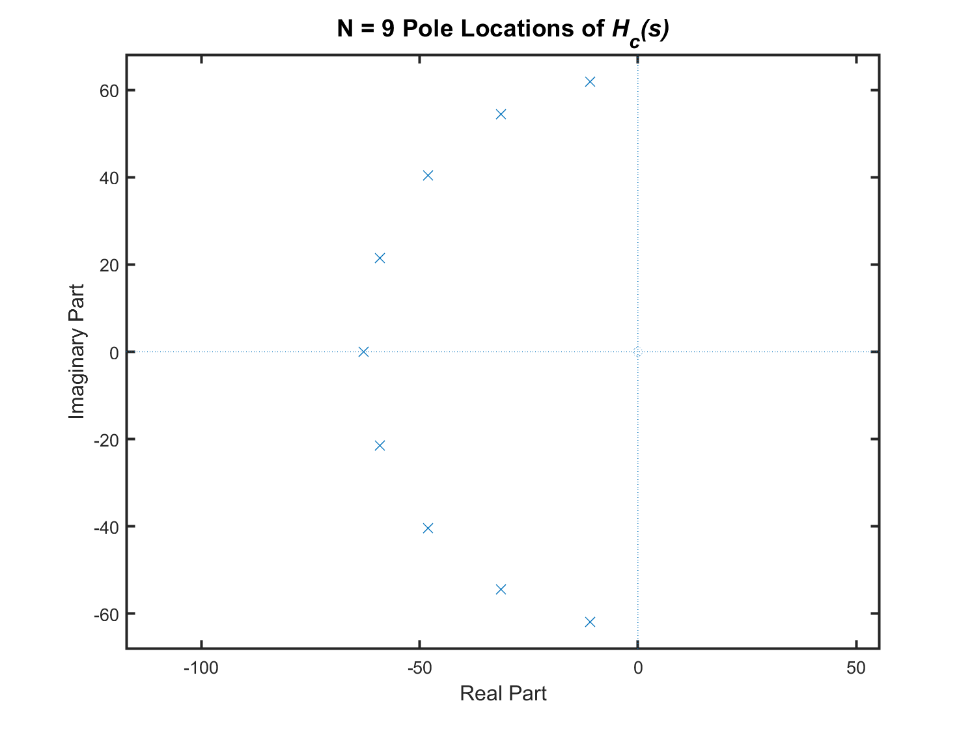








figure, zplane(C,D), title('N = 9 Pole Locations of \it{H\_c(s)}')





**(b)** Plot the magnitude and log-magnitude responses over  Hz range.

**MATLAB script:**

% Frequency Response

Fmax = 100; F = linspace(0,Fmax,2001);

omega = 2\*pi\*F;

H = freqs(C,D,omega);

Hmag = abs(H);

Hdb = mag2db(Hmag);

% Plot Results

figure('Units','inches','Position',[0,0,12,4]);

% Magnitude Response Plot

subplot(1,2,1),plot(F,Hmag,'LineWidth',1.5),title('Magnitude Response'), grid on

xlim([0 100]), ylim([0 1.1]), yticks([0 0.707 1])

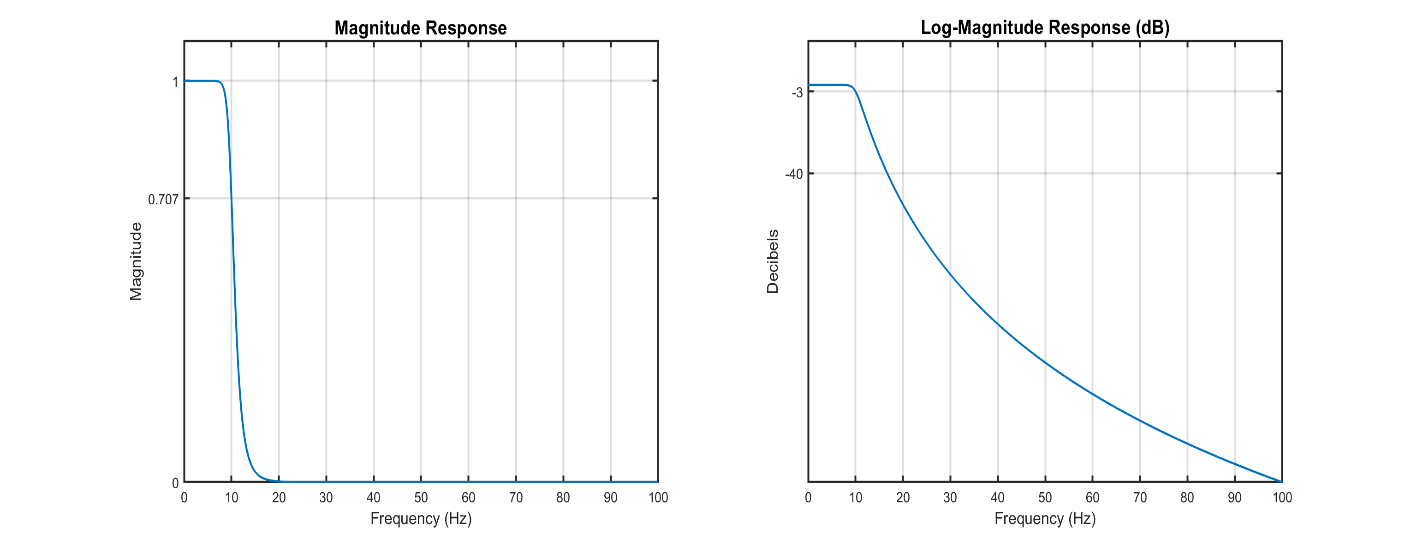
xlabel('Frequency (Hz)'),ylabel('Magnitude')

% Log-Magnitude Response Plot

subplot(1,2,2),plot(F,Hdb,'LineWidth',1.5),title('Log-Magnitude Response (dB)'), grid on

xlabel('Frequency (Hz)'),ylabel('Decibels'), xlim([0 100])

yticks([-40 -3])





**(c)** Determine frequencies at which the attenuation is  db,  db, and  db.

**Solution:**

figure

% Log-Magnitude Response Plot

plot(F,Hdb),title('Log-Magnitude Response (dB)'), grid on

xlabel('Frequency (Hz)'),ylabel('Decibels')

ylim([-55 -25]), yticks([-50 -40 -30]), xlim([11 22])

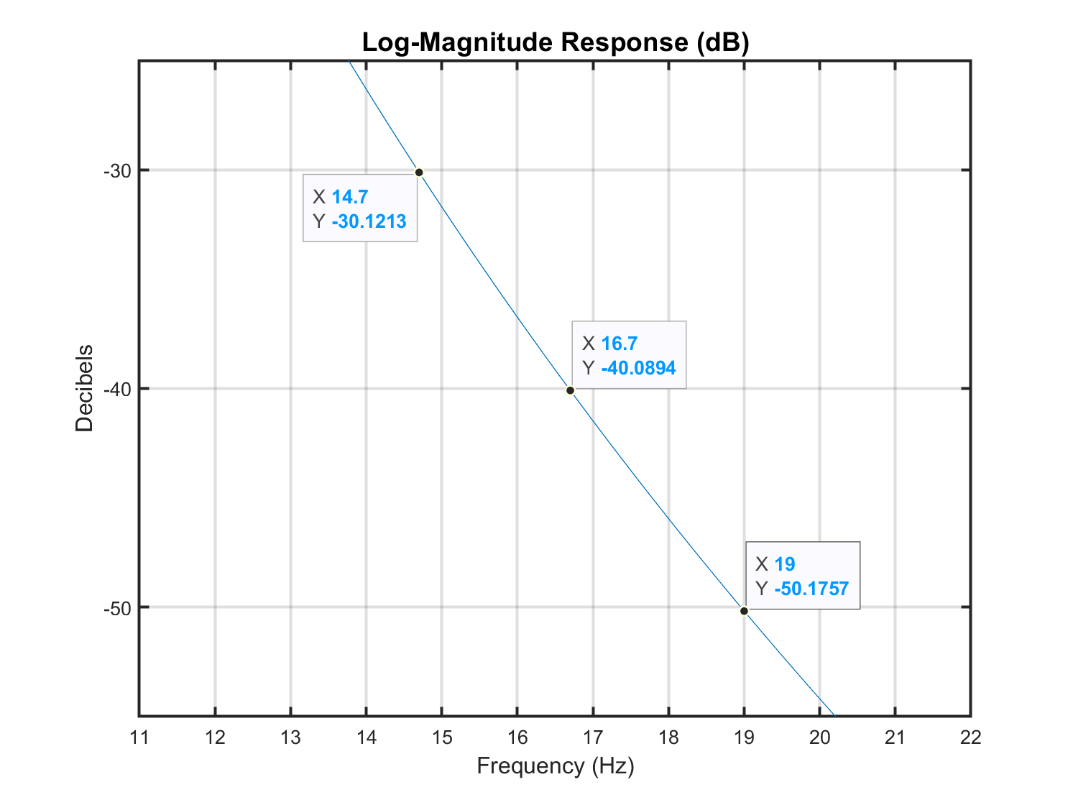
ax = gca;

chart = ax.Children(1);

datatip(chart,14.7,-30.1213,'Location','southwest');

datatip(chart,16.7,-40.0894,'Location','northeast');

datatip(chart,19,-50.1757,'Location','northeast');



Observing the data points of the Log-Magnitude Response, we can see that the corresponding frequencies are:









# Problem 7.3

**Text Problem 11.31 (Page 695)**

A lowpass digital filter's specifications are given by:



**(a)** Using bilinear transformation and Chebyshev-I approximation approach obtain a system function  in the rational function form that satisfies the above specifications.

**Solution:**

clc; close all; clear;

Omegap = 0.4\*pi; Omegas = 0.55\*pi;

Ap = 0.5; As = 50; Td = 2;

Omegap = (2/Td)\*tan(Omegap/2); Omegas = (2/Td)\*tan(Omegas/2);

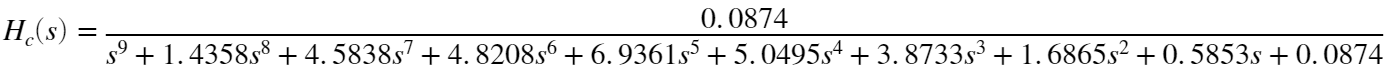
[N,Omegac] = cheb1ord(Omegap,Omegas,Ap,As,'s')

N = 8

Omegac = 0.7265

[C,D] = cheby1(N,Ap,Omegac,'s');

After using the Chebyshev-I approximation approach, we find that the CT system function  is:



Then using the bilinear transformation function along with an arbitrary factor of , we derive the system function  to be:

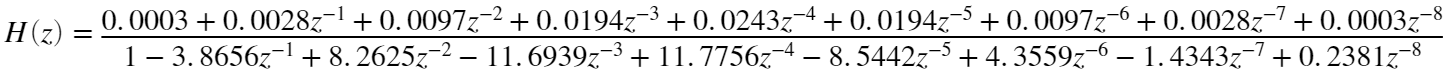
[B,A] = bilinear(C,D,1/Td)

B = 1×9

0.0003 0.0028 0.0097 0.0194 0.0243 0.0194 0.0097 ⋯

A = 1×9

1.0000 -3.8656 8.2625 -11.6939 11.7756 -8.5442 4.3559 ⋯





**(b)** Provide design plots in the form of log-magnitude, phase delay, group-delay, and impulse responses.

**MATLAB script:**

% Frequency Response

omega = linspace(0,pi,1001);

H = freqz(B,A,omega);

Hmag = abs(H);

Hdb = mag2db(Hmag);

% Impulse Response

N = 50; n = 0:N; x = (n==0); h = filter(B,A,x);

% Phase Response

Hpha = angle(H);

Hgd = -diff(unwrap(Hpha))./diff(omega);

Hgd = [Hgd Hgd(end)];

%[sos G] = tf2sos(B,A);

%Hgd = grpdelay(sos,1001)';

Hgd = medfilt1(Hgd,3);

% Plot Results

figure('Units','inches','Position',[0,0,12,4]);

% Magnitude Response Plot

subplot(2,2,1),plot(omega/pi,Hdb,'LineWidth',1.5),title('Log-Magnitude Response (dB)'), grid on

xticks([0 0.4 0.55 1]), xticklabels({'0','0.4','0.55','1'}), yticks([-50 -0.5]), ylim([-80 10]),

xlabel('\omega/\pi'),ylabel('Magnitude')

% Impulse Response Plot

subplot(2,2,2),stem(n,h,'filled'),title('Impulse Response'),xlabel('Time Index t/s'),ylabel('\it{h(t)}')

% Group-Delay Plot

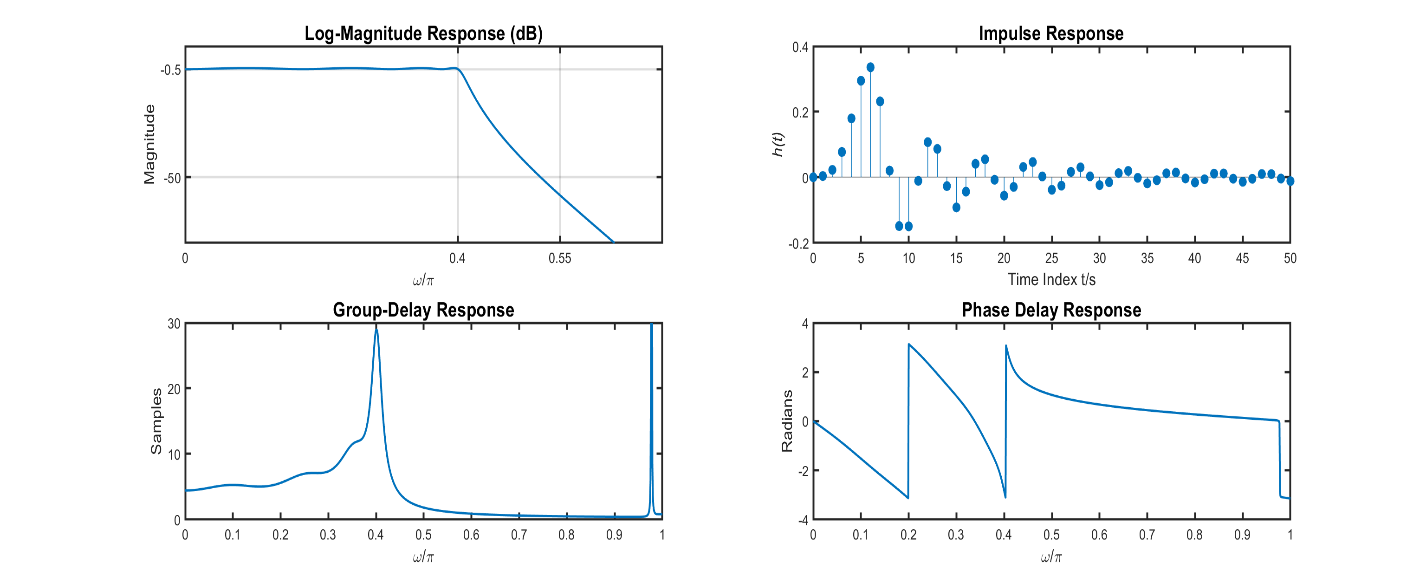
subplot(2,2,3), plot(omega/pi,Hgd,'LineWidth',1.5), title('Group-Delay Response'), xlabel('\omega/\pi')

ylabel('Samples'), ylim([0 30])

% Phase Delay Plot

subplot(2,2,4), plot(omega/pi,Hpha,'LineWidth',1.5), title('Phase Delay Response'), xlabel('\omega/\pi')

ylabel('Radians')





**(c)** Determine the exact band-edge frequencies for the given specifications.

**Solution:**

% Exact Band-Edge Frequencies

ind = find(Hdb > -Ap,1,'last'); w1 = omega(ind)/pi % Exact Passband Edge

w1 = 0.3990

ind = find(Hdb < -As,1,'first'); w2 = omega(ind)/pi % Exact Stopband Edge

w2 = 0.5220

Thus, the exact band-edge frequencies are

 and , which satisfy the design requirements.



# Problem 7.4

**Text Problem 11.38**

A highpass filter specifications are given by:





**(a)** Using the Butterworth approximation obtain a system function  in the cascade function form that satisfies the above specifications.

**Solution:**

clc; close all; clear;

omegas = 0.6\*pi; omegap = 0.8\*pi;

As = 40; Ap = 1;

[N,Omegac] = buttord(omegap/pi,omegas/pi,Ap,As)

N = 7

Omegac = 0.7709

[bhp,ahp] = butter(N,Omegac,'high')

bhp = 1×8

0.0002 -0.0014 0.0042 -0.0071 0.0071 -0.0042 0.0014 ⋯

ahp = 1×8

1.0000 3.7738 6.5614 6.6518 4.2030 1.6437 0.3666 ⋯

[b0,B,A] = dir2cas(bhp,ahp)

b0 = 2.0235e-04

B = 4×3

1.0000 -1.9907 0.9908

1.0000 -2.0031 1.0031

1.0000 -2.0135 1.0135

1.0000 -0.9927 0

A = 4×3

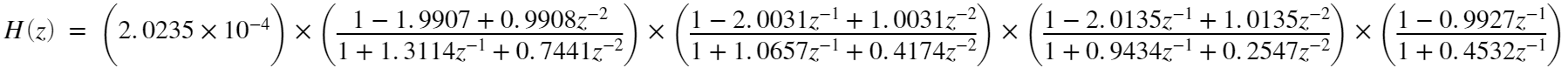
1.0000 1.3114 0.7441

1.0000 1.0657 0.4174

1.0000 0.9434 0.2547

1.0000 0.4532 0

Thus, the rational system function  in its cascade form is:



[sos G] = tf2sos(bhp,ahp);



**(b)** Provide design plots in the form of log-magnitude, phase delay, group-delay, and impulse responses.

**Solution:**

% Frequency Response

omega = linspace(0,pi,2001);

H = freqz(bhp,ahp,omega);

Hmag = abs(H);

Hdb = mag2db(Hmag);

% Impulse Response

N = 50; n = 0:N; x = (n==0); h = filter(bhp,ahp,x);

% Phase Response

Hpha = angle(H);

Hgd = grpdelay(sos,2001)';

% Plot Results

figure('Units','inches','Position',[0,0,12,4]);

% Log-Magnitude Response Plot

subplot(2,2,1), plot(omega/pi,Hdb,'LineWidth',1.5),title('Log-Magnitude Response (dB)'), grid on

xticks([0 0.6 0.8 1]),xticklabels({'0','0.6','0.8','1'}), yticks([-40 1]), ylim([-80 10]),

xlabel('\omega/\pi'),ylabel('Magnitude')

% Impulse Response Plot

subplot(2,2,2),stem(n,h,'filled'),title('Impulse Response'),xlabel('Time Index t/s'),ylabel('\it{h(t)}')

ylim([-0.25 0.25])

% Group-Delay Plot

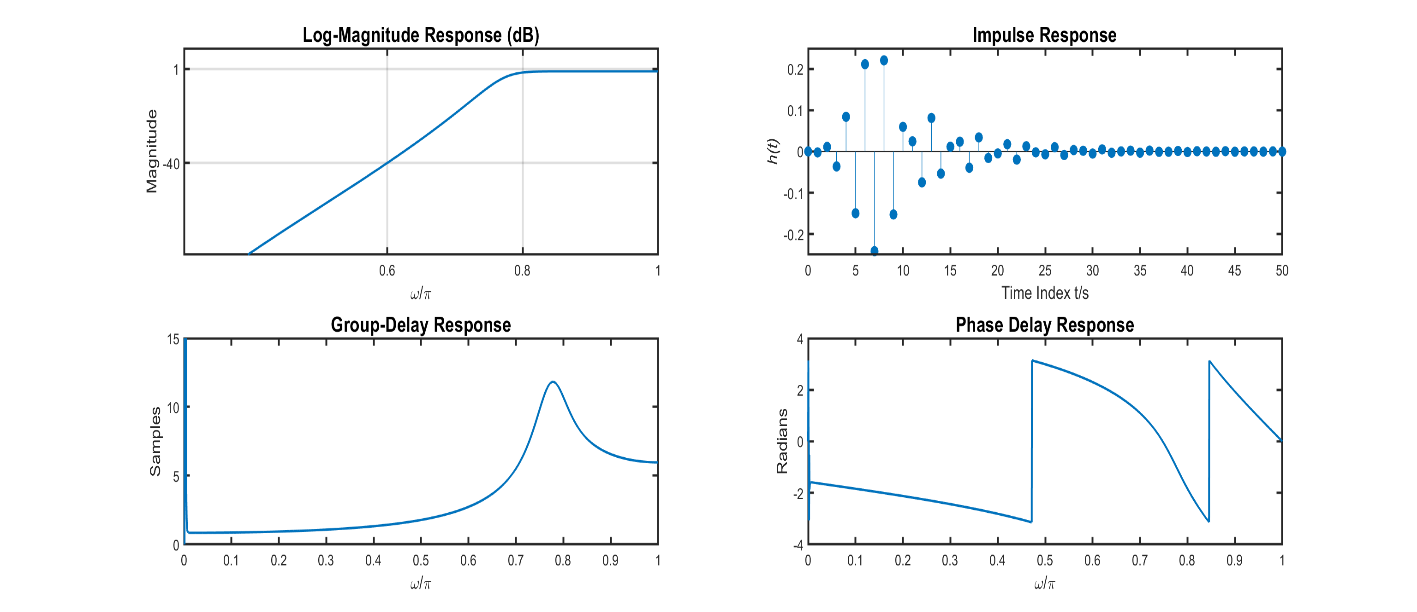
subplot(2,2,3), plot(omega/pi,Hgd,'LineWidth',1.5), title('Group-Delay Response'), xlabel('\omega/\pi')

ylabel('Samples'), ylim([0 15])

% Phase Delay Plot

subplot(2,2,4), plot(omega/pi,Hpha,'LineWidth',1.5), title('Phase Delay Response'), xlabel('\omega/\pi')

ylabel('Radians')





**(c)** Determine the exact band-edge frequencies for the given specifications.

**Solution:**

% Exact Band-Edge Frequencies

ind = find(Hdb > -Ap,1,'first'); w1 = omega(ind)/pi % Exact Passband Edge

w1 = 0.7905

ind = find(Hdb < -As,1,'last'); w2 = omega(ind)/pi % Exact Stopband Edge

w2 = 0.5995

Thus, the exact band-edge frequencies are

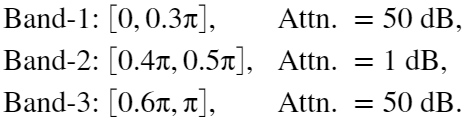
 and , which meets the design requirements for the highpass filter specifications.



# Problem 7.5

**Text Problem 11.43 (Page 698)**

A digital filter is specified by the following band parameters:





**(a)** Using Chebyshev II approximation, obtain a system function  in the rational function form that satisfies the above specifications.

**Solution:**

clc; close all; clear;

omegas1 = 0.3\*pi; omegap1 = 0.4\*pi;

omegas2 = 0.6\*pi; omegap2 = 0.5\*pi;

As1 = 50; Ap = 1; As2 = 50; As = max(As1,As2);

omegas = [omegas1 omegas2];

omegap = [omegap1 omegap2];

[N,Omegac] = cheb2ord(omegap/pi,omegas/pi,Ap,As)

N = 4

Omegac = 1×2

0.3000 0.6000

[b,a] = cheby2(N,As,Omegac,'bandpass')

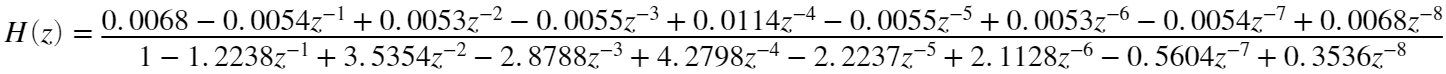
b = 1×9

0.0068 -0.0054 0.0053 -0.0055 0.0114 -0.0055 0.0053 ⋯

a = 1×9

1.0000 -1.2238 3.5354 -2.8788 4.2798 -2.2237 2.1128 ⋯

With the numerator and denominator coefficients now calculated, the system function can be described as





**(b)** Provide design plots in the form of magnitude, log-magnitude, group-delay, and impulse responses.

**MATLAB script:**

% Frequency Response

omega = linspace(0,pi,1001);

H = freqz(b,a,omega);

Hmag = abs(H);

Hdb = mag2db(Hmag);

% Impulse Response

N = 50; n = 0:N; x = (n==0); h = filter(b,a,x);

% Phase Response

Hpha = angle(H);

Hgd = -diff(unwrap(Hpha))./diff(omega);

Hgd = [Hgd Hgd(end)];

Hgd = medfilt1(Hgd,3);

% Plot Results

figure('Units','inches','Position',[0,0,12,4]);

% Magnitude Response Plot

subplot(2,2,1),plot(omega/pi,Hmag,'LineWidth',1.5),title('Magnitude Response'), grid on

xticks([0 0.3 0.4 0.5 0.6 1]), yticks([0 0.707 1]), ylim([0 1.1]),

xlabel('\omega/\pi'),ylabel('Magnitude')

% Log-Magnitude Response Plot

subplot(2,2,2),plot(omega/pi,Hdb,'LineWidth',1.5),title('Magnitude Response (dB)'), grid on

xticks([0 0.3 0.4 0.5 0.6 1]), yticks([-50 -1]), ylim([-100 10]),

xlabel('\omega/\pi'),ylabel('Magnitude')

% Group-Delay Plot

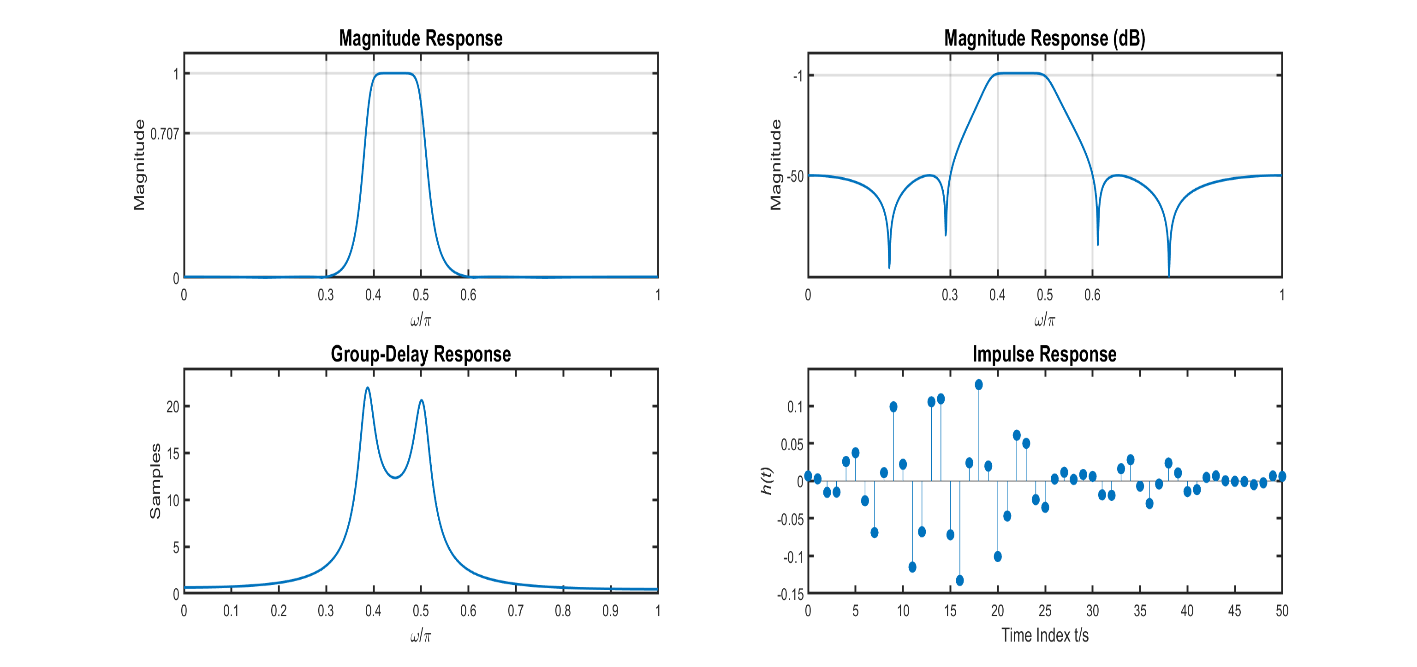
subplot(2,2,3), plot(omega/pi,Hgd,'LineWidth',1.5), title('Group-Delay Response'), xlabel('\omega/\pi')

ylabel('Samples'), ylim([0 24])

% Impulse Response Plot

subplot(2,2,4),stem(n,h,'filled'),title('Impulse Response'),xlabel('Time Index t/s'),ylabel('\it{h(t)}')

ylim([-0.15 0.15])





**(c)** Determine the exact band-edge frequencies for the given attenuation.

**Solution:**

% Exact Band-Edge Frequencies

ind = find(Hdb > -Ap); wp1 = omega(ind)/pi; % Exact Passband Edges

LowerPassEdge = wp1(1), UpperPassEdge = wp1(end)

LowerPassEdge = 0.3920

UpperPassEdge = 0.4970

inds1 = find(Hdb(1:401) < -As,1,'last'); % Exact Lower Stopband Edge

LowerStopEdge = omega(inds1)/pi

LowerStopEdge = 0.3000

inds2 = find(Hdb < -As); % Exact Upper Stopband Edge

LowerStopEdge = omega(602)/pi

LowerStopEdge = 0.6010

Thus the exact band-edge frequencies are:

, 

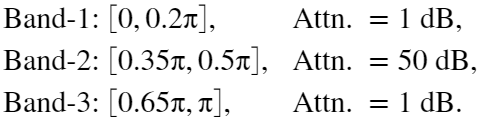
, 



# Problem 7.6

**Text Problem 11.66 (Page 701)**

A digital filter is specified by the following band parameters:



**(a)** Using Butterworth approximation, obtain a system function  in the cascade form that satisfies the above specifications.

**Solution:**

clc; close all; clear;

wp1 = 0.2; ws1 = 0.35; Ap1 = 1;

wp2 = 0.65; ws2 = 0.5; As = 50;

omegap = [wp1 wp2];

omegas = [ws1 ws2];

[N,Omegac] = buttord(omegap,omegas,Ap1,As)

N = 6

Omegac = 1×2

0.2566 0.6130

[b,a] = butter(N,Omegac,'stop');

[bo,B,A] = dir2cas(b,a)

bo = 0.0982

B = 6×3

1.0000 -0.4752 1.0001

1.0000 -0.4766 0.9956

1.0000 -0.4788 1.0045

1.0000 -0.4815 0.9956

1.0000 -0.4838 1.0044

1.0000 -0.4851 0.9999

A = 6×3

1.0000 0.5928 0.7650

1.0000 0.3322 0.4315

1.0000 -0.0441 0.2424

1.0000 -0.5481 0.2885

1.0000 -0.9404 0.5150

1.0000 -1.2428 0.8133

Here the coefficents for the cascade form of  can be written as:



**(b)** Provide design plots in the form of magnitude, log-magnitude, group-delay, and impulse responses.

**Solution:**

% Frequency Response

omega = linspace(0,pi,1001);

H = freqz(b,a,omega);

Hmag = abs(H);

Hdb = mag2db(Hmag);

% Impulse Response

L = 50; n = 0:L; x = (n==0); h = filter(b,a,x);

% Phase Response

Hpha = angle(H);

Hgd = -diff(unwrap(Hpha))./diff(omega);

Hgd = [Hgd Hgd(end)];

Hgd = medfilt1(Hgd,3);

% Plot Results

figure('Units','inches','Position',[0,0,12,4]);

% Magnitude Response Plot

subplot(2,2,1),plot(omega/pi,Hmag,'LineWidth',1.5),title('Magnitude Response'), grid on

xticks([0 0.2 0.35 0.5 0.65 1]), yticks([0 0.707 1]), ylim([0 1.1]),

xlabel('\omega/\pi'),ylabel('Magnitude')

% Log-Magnitude Response Plot

subplot(2,2,2),plot(omega/pi,Hdb,'LineWidth',1.5),title('Magnitude Response (dB)'), grid on

xticks([0 0.2 0.35 0.5 0.65 1]), yticks([-50 -1]), ylim([-100 10]),

xlabel('\omega/\pi'),ylabel('Magnitude')

% Group-Delay Plot

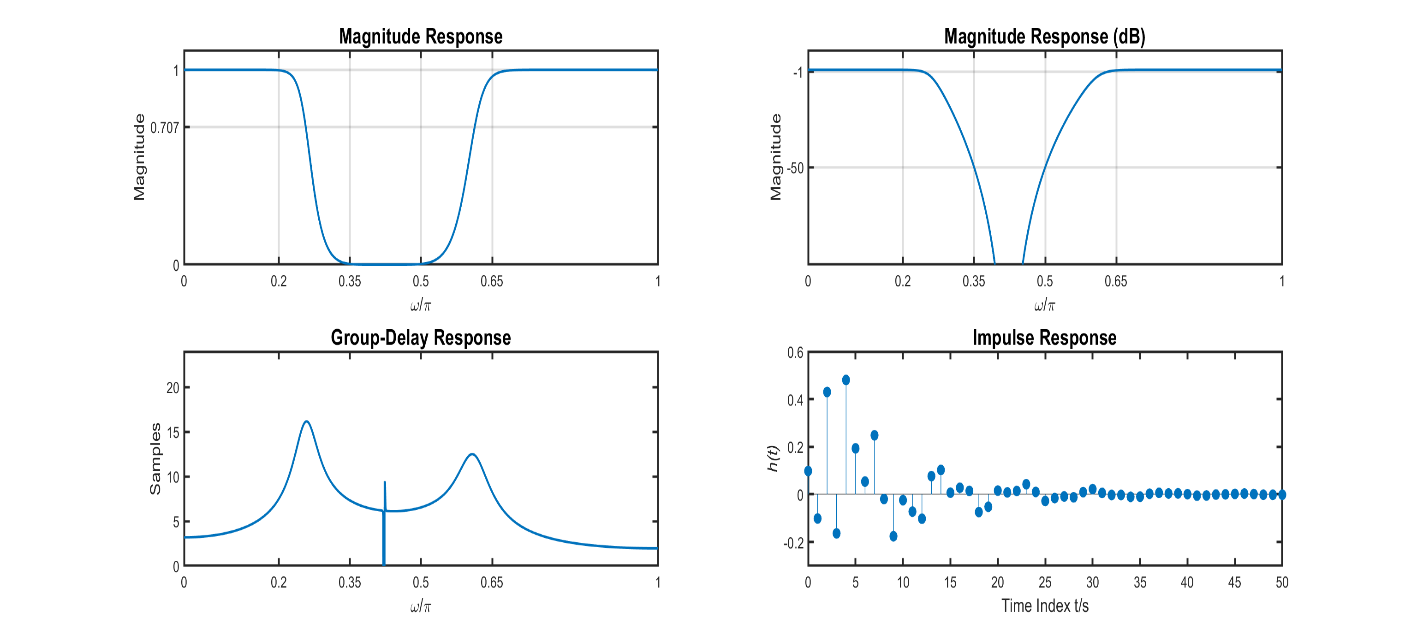
subplot(2,2,3), plot(omega/pi,Hgd,'LineWidth',1.5), title('Group-Delay Response'), xlabel('\omega/\pi')

ylabel('Samples'),xticks([0 0.2 0.35 0.5 0.65 1]), ylim([0 24])

% Impulse Response Plot

subplot(2,2,4),stem(n,h,'filled'),title('Impulse Response'),xlabel('Time Index t/s'),ylabel('\it{h(t)}')

ylim([-0.3 0.6])





**(c)** Determine the exact band-edge frequencies for the given attenuation.

**Solution:**

ind = find(Hdb(1:401) < -Ap1, 1,'first'); wplow = omega(ind)/pi;

ind = find(Hdb(1:401) > -As, 1,'last'); wslow = omega(ind)/pi;

ind = Hdb(500:502); wsupper = omega(502)/pi;

ind = find(Hdb(1:701) < -Ap1, 1,'last'); wpupper = omega(ind)/pi;

wplow,wslow,wsupper,wpupper

wplow = 0.2430

wslow = 0.3490

wsupper = 0.5010

wpupper = 0.6310

Thus, the exact band-edge frequencies for this Butterworth Bandstop IIR Filter are:

, , , and 



# Problem 7.7

**Text Problem 11.70 (Page 702)**

An analog signal is to be processed using the effective continuous-time system of Figure 6.18 in which the sampling frequency is 1 kHz.



**(a)** Design a minimum-order IIR digital filter that will suppress the  Hz component down to  dB while pass the  Hz component with attenuation of less than  dB. The digital filter should have an equiripple passband and stopband. Determine the system function of the filter and plot its log-magnitude response in dB.

**Solution:**

Since the requirements call for a filter with an equiripple passband and stopband, the only filter that can achieve the design specs are an analog Elliptic filter

clc; close all; clear;

% Design Requirements

Fs = 1000; Fs2 = Fs/2;

fs = 300/Fs; As = 50;

fp = 250/Fs; Ap = 1;

omegas = 2\*fs;

omegap = 2\*fp;

[N,omegac] = ellipord(omegap,omegas,Ap,As)

N = 6

omegac = 0.5000

[b1,a1] = ellip(N,Ap,As,omegac)

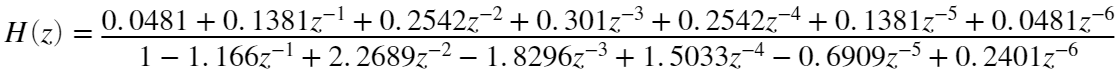
b1 = 1×7

0.0481 0.1381 0.2542 0.3010 0.2542 0.1381 0.0481

a1 = 1×7

1.0000 -1.1660 2.2689 -1.8296 1.5033 -0.6909 0.2401

Deriving the coefficients of the Digital Elliptic Lowpass IIR Filter shows that the system function is:



% Frequency Response

omega = linspace(0,pi,2001);

H = freqz(b1,a1,omega);

Hmag = abs(H);

Hdb = mag2db(Hmag);

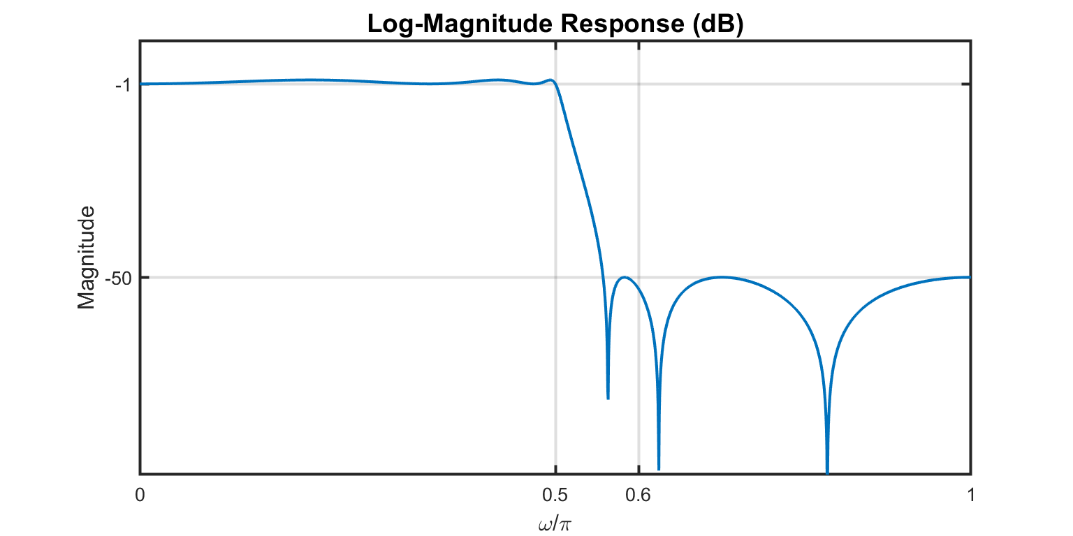
% Log-Magnitude Response Plot

figure('Units','inches','Position',[0,0,6,3]);

plot(omega/pi,Hdb,'LineWidth',1.5),title('Log-Magnitude Response (dB)'), grid on

xticks([0 0.5 0.6 1]), yticks([-50 -1]), ylim([-100 10]),

xlabel('\omega/\pi'),ylabel('Magnitude')





**(b)** Process the signal  through the effective analog system. Generate sufficient samples so that the output response  goes into steady-state. Plot the steady-state  and comment on the filtering result.

**Solution:**

Ts = 1/Fs; t = 0:0.0001:.5;

% Continuous-Time Signal

xc = 5\*sin(2\*pi\*250.\*t)+10\*sin(2\*pi\*300.\*t);

% Discrete-Time Signal

n = 0:500; nTs = n\*Ts;

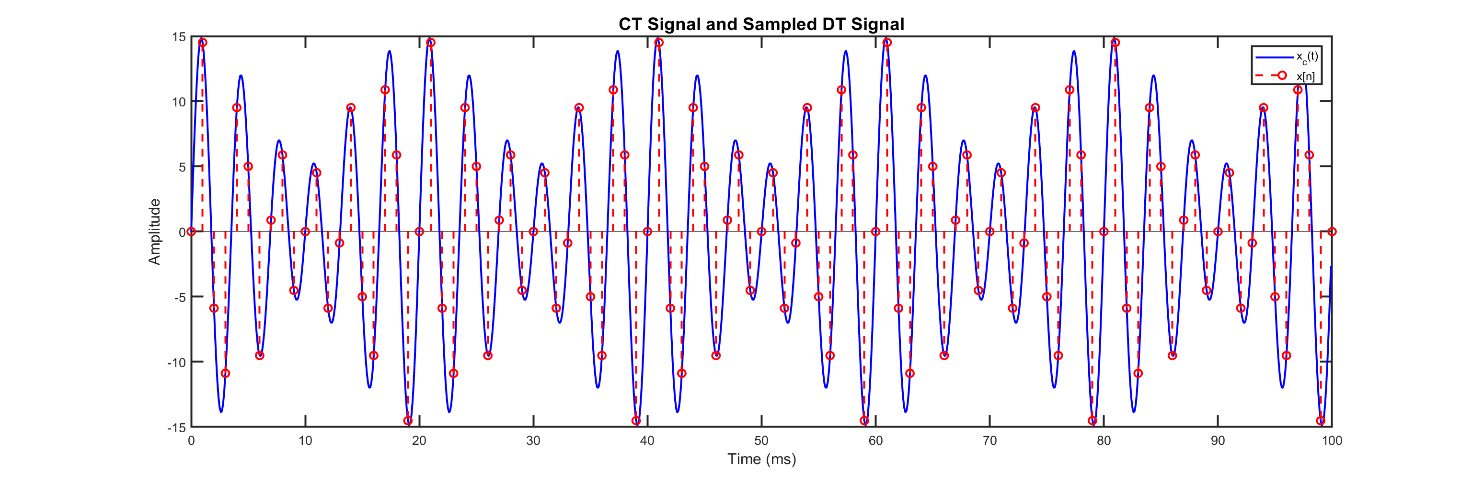
xn = 5\*sin(2\*pi\*250.\*nTs)+10\*sin(2\*pi\*300.\*nTs);

% Plot the CT signal and Sampled DT signal

figure('Units','inches','Position',[0,0,12,4]);

plot(t(1:1000)\*1000,xc(1:1000),'b','LineWidth',1.5), hold on, stem(nTs(1:101)\*1000,xn(1:101),'--r','LineWidth',1.5), xlabel('Time (ms)'),ylabel('Amplitude')

title('CT Signal and Sampled DT Signal'), legend('x\_c(t)','x[n]')



% Filter the Discrete-Signal x[n] through Elliptic Lowpass IIR Filter

yn = filter(b1,a1,xn);

% Reconstruct the analog signal through interpolation and analyze results

yt = yn \* sinc(Fs\*(ones(length(n),1)\*t-nTs'\*ones(1,length(t))));

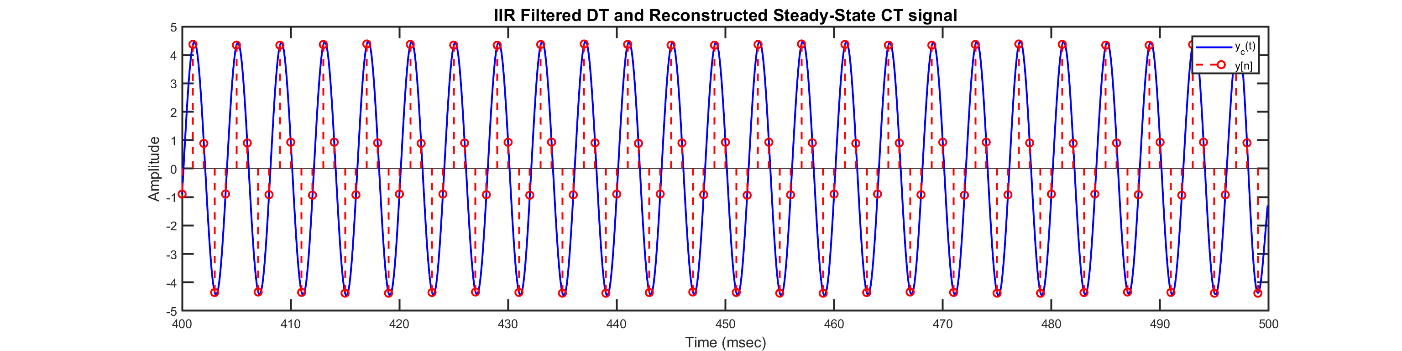
figure('Units','inches','Position',[0,0,12,3]);

plot(t(4001:5000)\*1000,yt(4001:5000),'b','LineWidth',1.5), hold on, stem(n(401:500),yn(401:500),'--r','LineWidth',1.5)

xlabel('Time (msec)')

ylabel('Amplitude')

title('IIR Filtered DT and Reconstructed Steady-State CT signal'), legend('y\_c(t)','y[n]')



**Comment:**

After analyzing the the reconstructed signal, , we see that the resulting signal is a pure sinusoid with only one fundamental frequency. The higher frequency of 300 Hz has been filtered out by the IIR filter and only the 250 Hz component remains. Once the signal reaches a steady-state, there is a small difference in magnitude as the reconstructed signal has a peak amplitude of about 4.5, compared to the original sinusoidal component .



**(c)** Repear parts (a) and (b) by designing an equiripple FIR filter. Compare the orders of the two filters and their filtering results.

**Solution:**

% Compute Absolute passband and stopband ripple values

[dp,ds] = spec\_convert(Ap,As,'rel','abs')

dp = 0.0575

ds = 0.0033

% Estimate Filter using FIRPMORD function

f = [0.5 0.6]; % Band-edge array

a = [1 0]; % Band-edge desired gain

dev = [dp,ds]; % Band tolerance

[M,fo,ao,W] = firpmord(f,a,dev); M

M = 34

% Filter Design using FIRPM function

[h,delta] = firpm(M,fo,ao,W);

% Frequency Response

omega = linspace(0,pi,2001);

H = freqz(h,1,omega);

Hmag = abs(H);

Hdb = mag2db(Hmag);

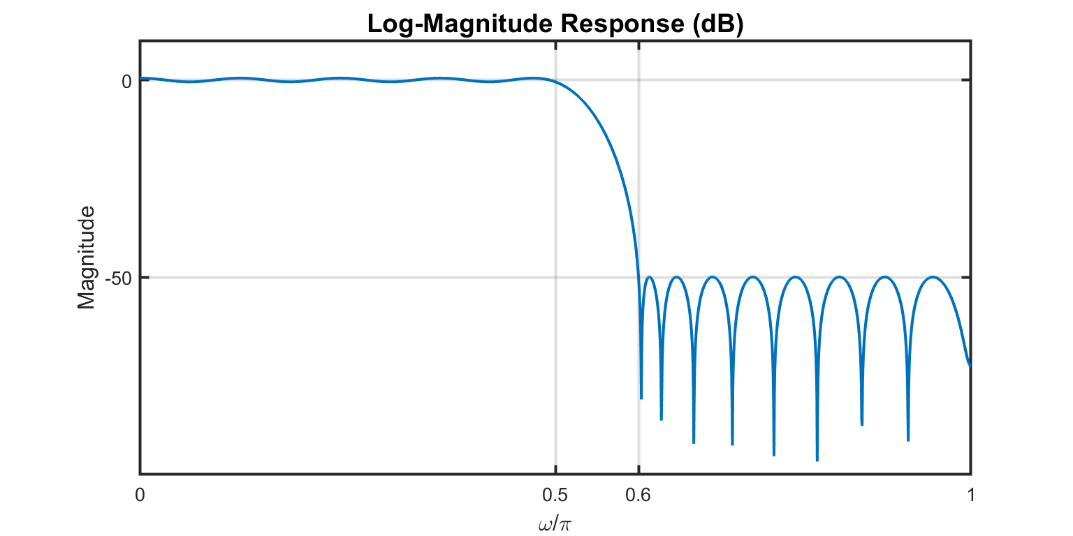
% Log-Magnitude Response Plot

figure('Units','inches','Position',[0,0,6,3]);

plot(omega/pi,Hdb,'LineWidth',1.5),title('Log-Magnitude Response (dB)'), grid on

xticks([0 0.5 0.6 1]), yticks([-50 0]), ylim([-100 10]),

xlabel('\omega/\pi'),ylabel('Magnitude')



% Filter the Discrete-Signal x[n] through Equiripple Lowpass FIR Filter

yn = filter(h,1,xn);

% Reconstruct the analog signal through interpolation and analyze results

yt = yn \* sinc(Fs\*(ones(length(n),1)\*t-nTs'\*ones(1,length(t))));

figure('Units','inches','Position',[0,0,12,3]);

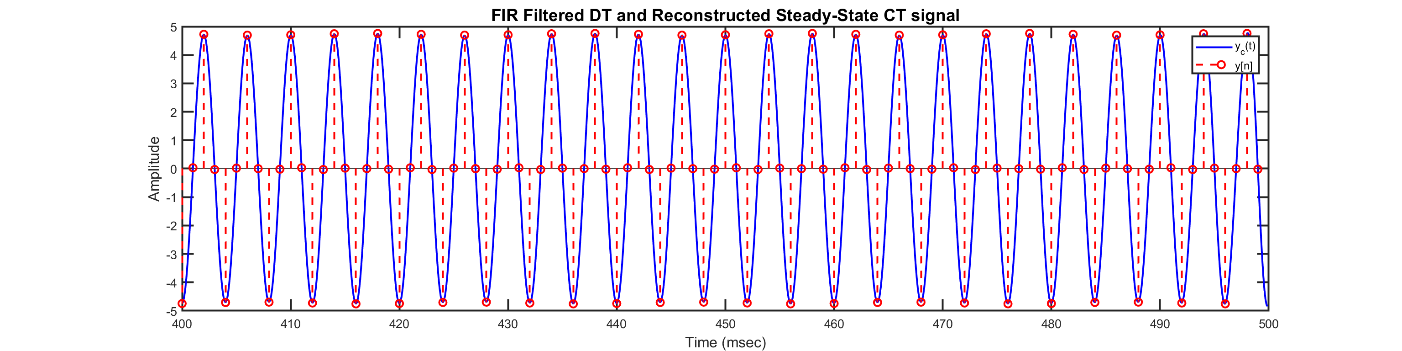
plot(t(4001:5000)\*1000,yt(4001:5000),'b','LineWidth',1.5), hold on

stem(n(401:500),yn(401:500),'--r','LineWidth',1.5)

xlabel('Time (msec)')

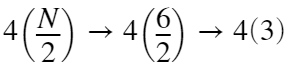
ylabel('Amplitude')

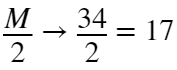
title('FIR Filtered DT and Reconstructed Steady-State CT signal'), legend('y\_c(t)','y[n]')



**Comment:**

After drawing comparisons between the two filters, the Elliptic IIR filter had an order of , while the FIR filter had order , which is almost 6 times the order of the IIR Filter. When comparing the number of multiplications per output sample:

The IIR Filter had order , thus this filter had = 12 multiplications per output sample

The FIR Filter had order , thus the number of multiplications per output sample was .

When comparing the exact band-edge frequencies, the IIR filter had a passband edge at  and a sharper transition band with a stopband edge at 

Whereas the FIR filter had a narrower stopband width with a passband edge at  and stopband edge at 

The resulting reconstructed signals had a difference in amplitude, as the FIR filter produced a signal that was closer to the sinusoidal component of , whereas the IIR had a smaller peak amplitude due to its sharper transition band cutting off frequencies before the desired stopband-edge frequency.

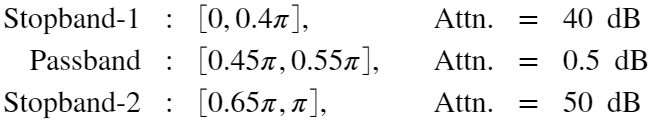
Overall, the IIR filter achieved the design requirements using about 6 times less coefficients and 5 multiplcations/output sample less than the FIR filter while also providing a shorter transition band that provided the necessary attenuations in the corresponding frequency bands.



# Problem 7.8

**Text Problem 11.71 (Page 703)**

Consider the following bandpass digital filter specifications:





**(a)** Design a minimum order FIR filter to satisfy the above specifications. Plot its magnitude, log-magnitude (dB), and group-delay responses in one figure using 3 rows and 1 column.

**MATLAB script**:

clc; close all; clear;

omegas1 = 0.4\*pi; omegap1 = 0.45\*pi;

omegas2 = 0.65\*pi; omegap2 = 0.55\*pi;

As1 = 40; As2 = 50; As = max(As1,As2);

Ap = 0.5;

[dp,ds1] = spec\_convert(Ap,As1,'rel','abs')

dp = 0.0288

ds1 = 0.0103

[dp,ds2] = spec\_convert(Ap,As2,'rel','abs')

dp = 0.0288

ds2 = 0.0033

% Estimate Filter using FIRPMORD function

f = [omegas1,omegap1,omegap2,omegas2]/pi; % Band-edge array

a = [0 1 0]; % Band-edge desired gain

dev = [ds1 dp ds2]; % Band tolerance

[M,fo,ao,W] = firpmord(f,a,dev); M

M = 65

% Filter Design using FIRPM function

[h,delta] = firpm(M,fo,ao,W);

n = 0:M; omega = linspace(0,pi,1001);

% Magnitude Response

H = freqz(h,1,omega); Hmag = abs(H);

Hdb = mag2db(Hmag);

% Phase and Group-Delay

Hpha = angle(H);

Hgd = -diff(unwrap(Hpha))./diff(omega);

Hgd = [Hgd Hgd(end)];

Hgd = medfilt1(Hgd,3);

% Plot Results

figure

% Magnitude Response Plot

subplot(3,1,1),plot(omega/pi,Hmag,'LineWidth',1.5),title('Magnitude Response'), grid on

xticks([0 0.4 0.45 0.55 0.65 1]), yticks([0 0.707 1]), ylim([0 1.1]),

xlabel('\omega/\pi'),ylabel('Magnitude')

% Log-Magnitude Response Plot

subplot(3,1,2),plot(omega/pi,Hdb,'LineWidth',1.5),title('Magnitude Response (dB)'), grid on

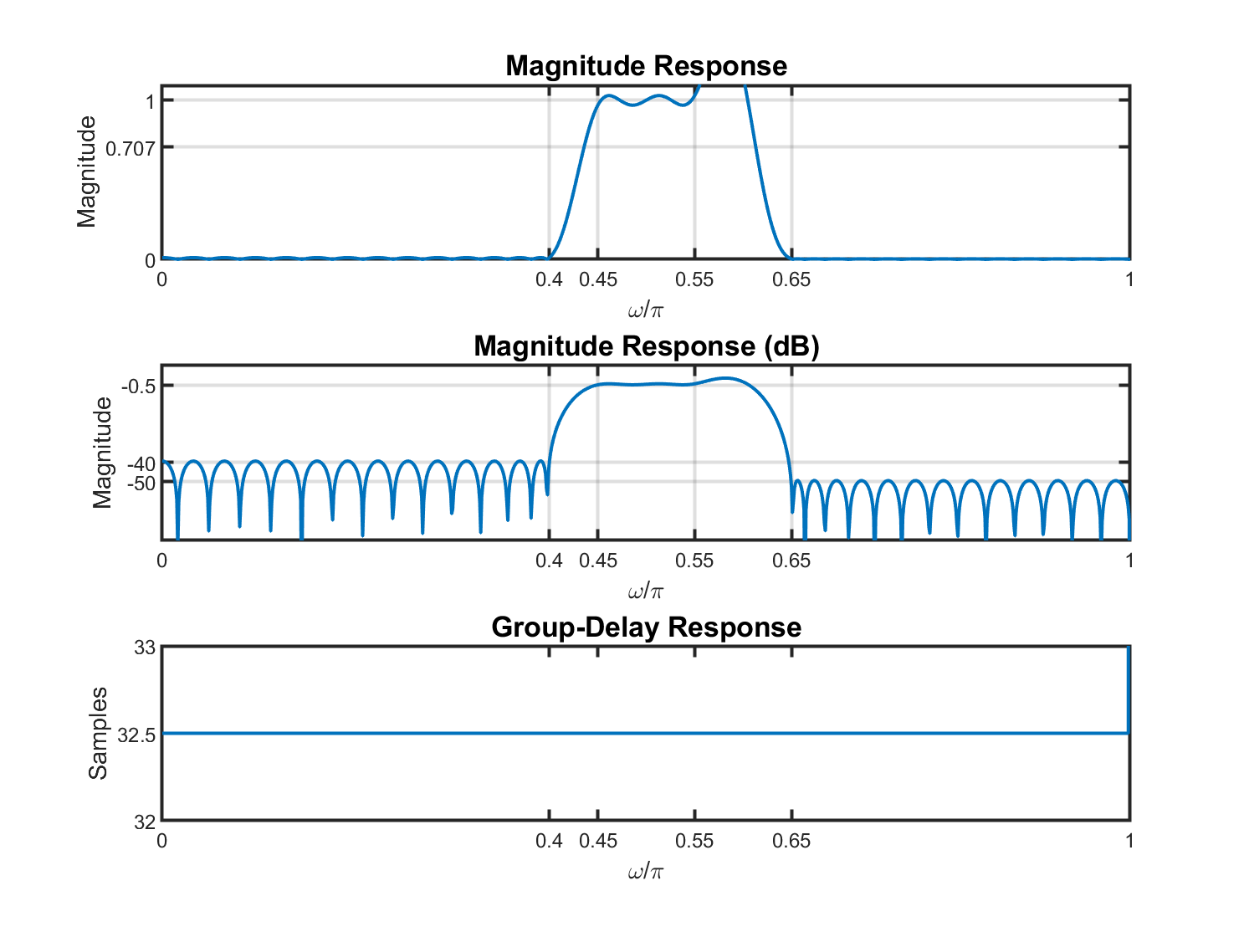
xticks([0 0.4 0.45 0.55 0.65 1]), yticks([-50 -40 -0.5]), ylim([-80 10]),

xlabel('\omega/\pi'),ylabel('Magnitude')

% Group-Delay Plot

subplot(3,1,3), plot(omega/pi,Hgd,'LineWidth',1.5), title('Group-Delay Response'), xlabel('\omega/\pi')

ylabel('Samples'),xticks([0 0.4 0.45 0.55 0.65 1]), ylim([32 33])





**(b)** Design a minimum order IIR filter to satisfy the above specifications. Plot its magnitude, log-magnitude (dB), and group-delay responses in one figure using 3 rows and 1 column. From your plots determine the exact band-edge frequencies.

**MATLAB script**:

Omegas = [omegas1 omegas2];

Omegap = [omegap1 omegap2];

[N,Omegac] = ellipord(Omegap/pi,Omegas/pi,Ap,As); N

N = 5

[b,a] = ellip(N,Ap,As,Omegac,'bandpass');

omega = linspace(0,pi,1001);

% Magnitude Response

H = freqz(b,a,omega); Hmag = abs(H);

Hdb = mag2db(Hmag);

% Phase and Group-Delay

Hpha = angle(H);

Hgd = -diff(unwrap(Hpha))./diff(omega);

Hgd = [Hgd Hgd(end)];

Hgd = medfilt1(Hgd,3);

% Plot Results

figure

% Magnitude Response Plot

subplot(3,1,1),plot(omega/pi,Hmag,'LineWidth',1.5),title('Magnitude Response'), grid on

xticks([0 0.4 0.45 0.55 0.65 1]), yticks([0 0.707 1]), ylim([0 1.1]),

xlabel('\omega/\pi'),ylabel('Magnitude')

% Log-Magnitude Response Plot

subplot(3,1,2),plot(omega/pi,Hdb,'LineWidth',1.5),title('Magnitude Response (dB)'), grid on

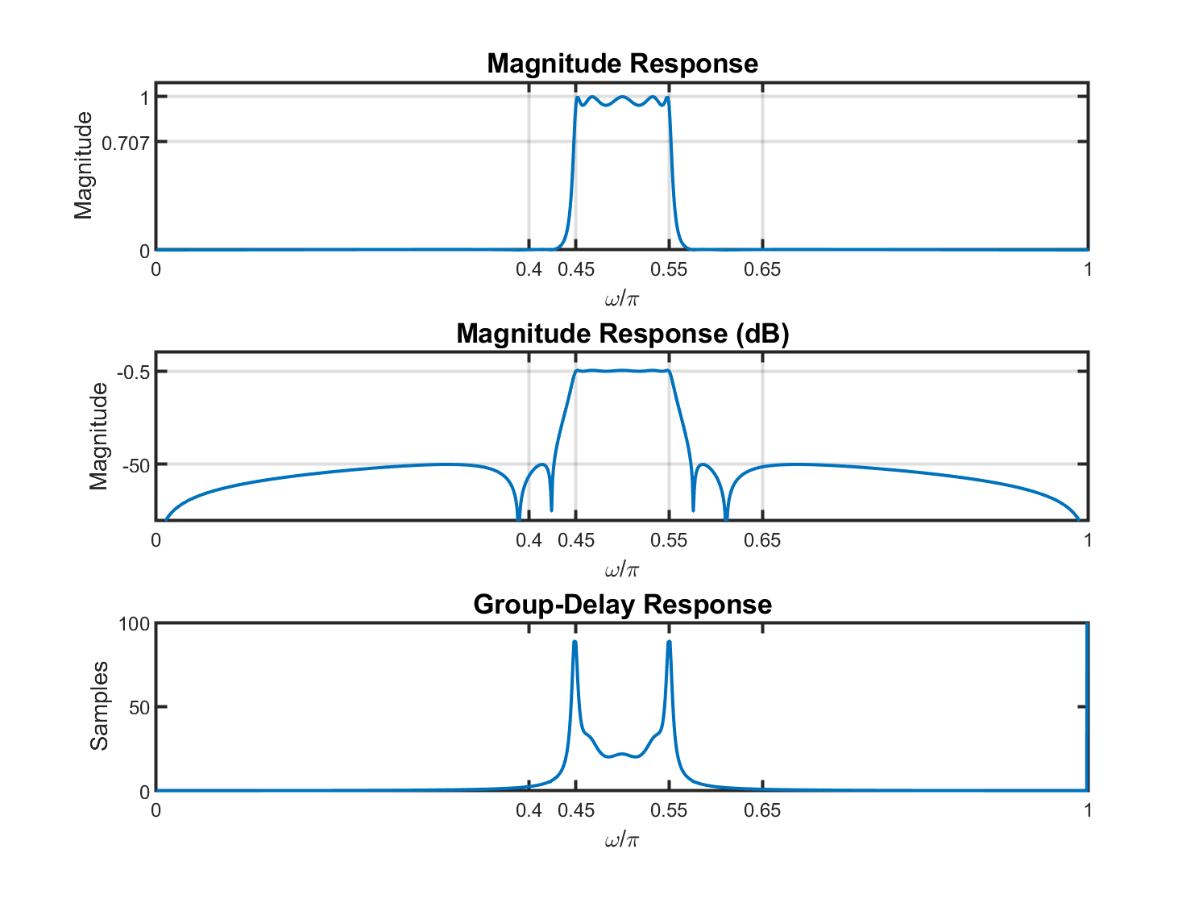
xticks([0 0.4 0.45 0.55 0.65 1]), yticks([-50 -0.5]), ylim([-80 10]),

xlabel('\omega/\pi'),ylabel('Magnitude')

% Group-Delay Plot

subplot(3,1,3), plot(omega/pi,Hgd,'LineWidth',1.5), title('Group-Delay Response'), xlabel('\omega/\pi')

ylabel('Samples'),xticks([0 0.4 0.45 0.55 0.65 1]),ylim([0 100])

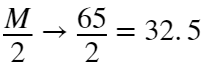


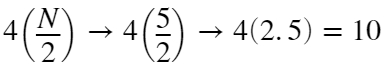


**(c)** Compare the two filter designs in terms of their responses and orders.

**Comparison**:

The FIR filter design was designed using the Parks-McClellan algorithm which produced an equiripple FIR filter with

minimum order . The design produced a linear phase which resulted in a constant group-delay. The magnitude response was unsatisfactory towards the design specifications, as the first stopband did not meet the attenuation requirement of 40dB, due to difference between the lower and upper transition bands. The number of multiplications per output sample for the FIR filter was 

The IIR filter was designed using a digital Elliptic filter and achieved a minimum order of  This filter was able to achieve attenuation requirements in both stopbands and the passband, but resulted in a nonlinear phase and group-delay response. The number of multiplications per output sample for the IIR filter was 

Thus, the IIR filter was able to satisfy the design requirements with order , which resulted in only 10 multiplications per output sample. This is about 3 times less the number of multiplications, so the IIR filter not only satisfied the magnitude requirements, but is computationally quicker also.



function [A,B] = spec\_convert(C,D,typein,typeout)

% typein: 'abs' or 'rel' or 'ana'

% typeout: 'abs' or 'rel' or 'ana'

% C,D: input specifications

% A,B: output specifications

% Enter your function code below

if nargin > 4

error('too many input arguments')

end

switch typein

case 'abs'

switch typeout

case 'rel'

Ap = 20\*log10((1+C)/(1-C)); % Relative Output: Passband ripple

As = floor(20\*log10((1+C)/D)); % Relative Output: Stopband Attenuation

A = Ap; B = As;

case 'ana'

Ap = 20\*log10((1+C)/(1-C)); % Relative Output: Passband ripple

As = floor(20\*log10((1+C)/D)); % Relative Output: Stopband Attenuation

A = sqrt(10^(Ap/10)-1); % Analog Output: Passband

B = floor(10^(As/20)); % Analog Output: Stopband

end

case 'rel'

switch typeout

case 'abs'

dp = (10^(C/20)-1)/(1+10^(C/20)); % Absolute Output: Passband Error

ds = (1+dp)/(10^(D/20)); % Absolute Output: Stopband Error

A = dp; B = ds;

case 'ana'

epsilon = sqrt(10^(C/10)-1); % Analog Output: Passband

B = floor(10^(D/20)); % Analog Output: Stopband

A = epsilon;

end

case 'ana'

switch typeout

case 'rel'

Ap = round(10\*log10(C^(2)+1),2); % Relative Output: Passband ripple (in dB)

As = 20\*log10(D); % Relative Output: Stopband Attenuation (in dB)

A = Ap; B = As;

case 'abs'

Ap = round(10\*log10(C^(2)+1),2); % Relative Output: Passband ripple (in dB)

As = 20\*log10(D); % Relative Output: Stopband Attenuation (in dB)

dp = (10^(Ap/20)-1)/(1+10^(Ap/20)); % Absolute Output: Passband Error

ds = (1+dp)/(10^(Ap/20)); % Absolute Output: Stopband Error

A = dp; B = ds;

end

end

end