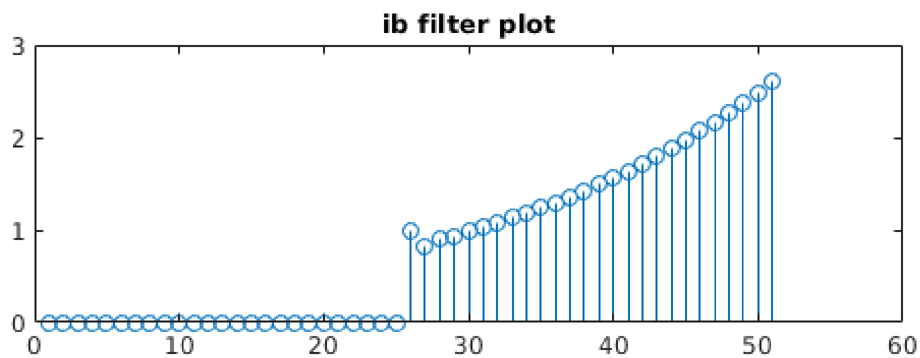
Exercise 7.1

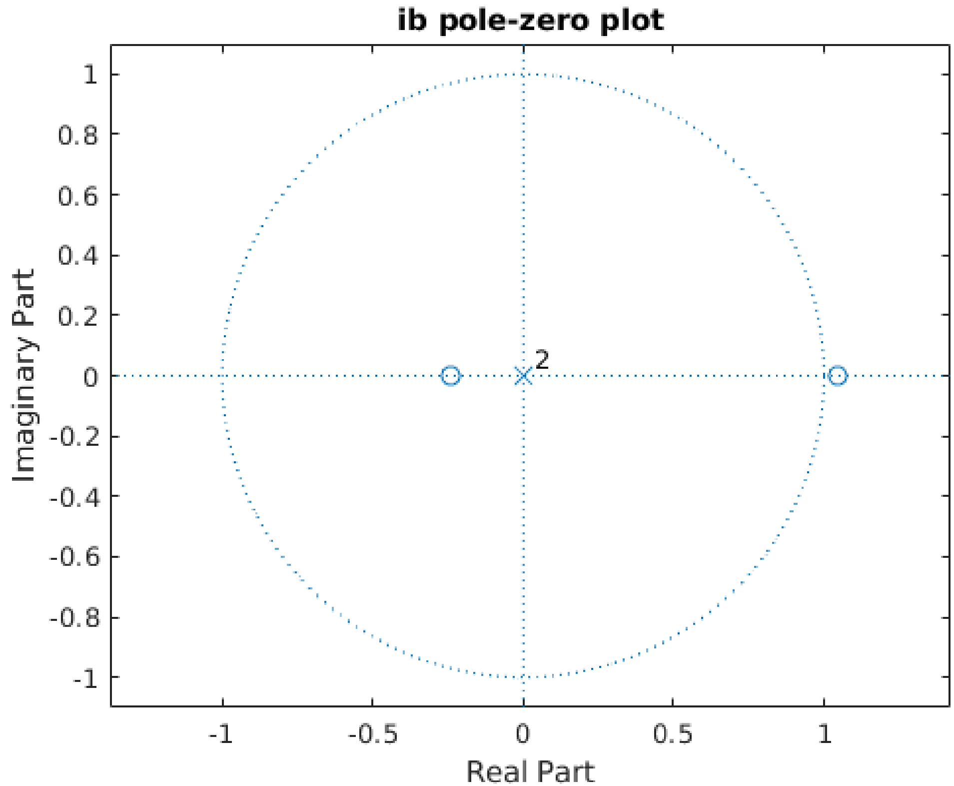
```
%a)

% i) h[n] = #[n] - 0.8#[n-1]
% H[z] = 1 - 0.8z^-1
%ROC: C - {#}
%Poles:0
%zeros:0.8
% stable
figure;
zplane(0.8,0);
title('i pole-zero plot');

%The filter matches the analytic equation, the pole corresponds to the
%decay towards zero

impulse = [ zeros(1,25) 1 zeros(1,25)];
a = [1 -0.8];
b = [1];
y = filter(b,a,impulse);

figure;
stem(y);
title('i filter plot')

% ii) h[n] = #[n] + 0.8#[n-1]
% H[z] = 1 + 0.8z^-1
%ROC: C - {#}
%zeros:-0.8
%poles:0
%stable
figure;
zplane(-0.8,0);
title('ii pole-zero plot');

a = [1 +0.8];
b = [1];
y = filter(b,a,impulse);

figure;
stem(y);
title('ii filter plot')

%The filter matches the analytic equation, the pole corresponds to the
%decay and growth towards zero

% iii) h[n] = #[n + 10] - 0.8#[n + 9]
%H[z] = z^10 - 0.8z^9
%ROC: C - {0}
%zeros:0.8, 0 mult. 9
%poles: none
```

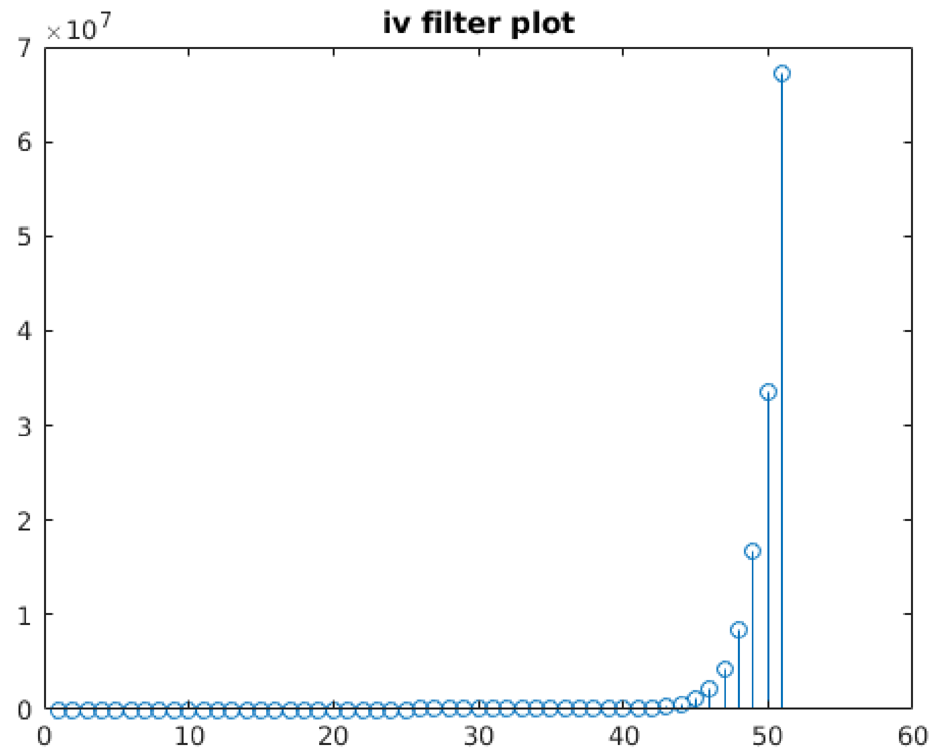
```
%stable
figure;
zplane([-0.8;0;0;0;0;0;0;0;0;0]);
title('iii pole-zero plot');

a = [1 +0.8];
b = [0 0 0 0 0 0 0 0 0 0 1];
y = filter(b,a,impulse);

figure;
stem(y);
title('iii filter plot')

%The filter matches the analytic equation, the pole corresponds to the
%decay and growth towards zero

% iv) h[n] = (1/2)#[n] - #[n-1]
%H[z] = 1/2 - z^-1
%ROC: C - {#}
%zeros: 2
%poles: 0
%stable
figure;
zplane(2,0);
title('iv pole-zero plot');

a = [1 -2];
b = [2];
y = filter(b,a,impulse);

figure;
stem(y);
title('iv filter plot')

%The filter matches the analytic equation, the pole corresponds to the
%growth

%b)

%i) h[n] = #[n] - cos(0.2pi)#[n -1] - 0.25#[n-2]
%freq resp: H(e^jw) = 1 - cos(0.2pi)*e^-jw - 0.25e^-2jw
% H[z] = 1 - cos(0.2pi)z^-1 - 0.25z^-2
%ROC: C - {#}
%stable
zer = [-0.23863; 1.04765];
poles = [0;0];



figure;
zplane(zer,poles);
title('ib pole-zero plot');

a = [1 -cos(0.2*pi) -0.25];
```

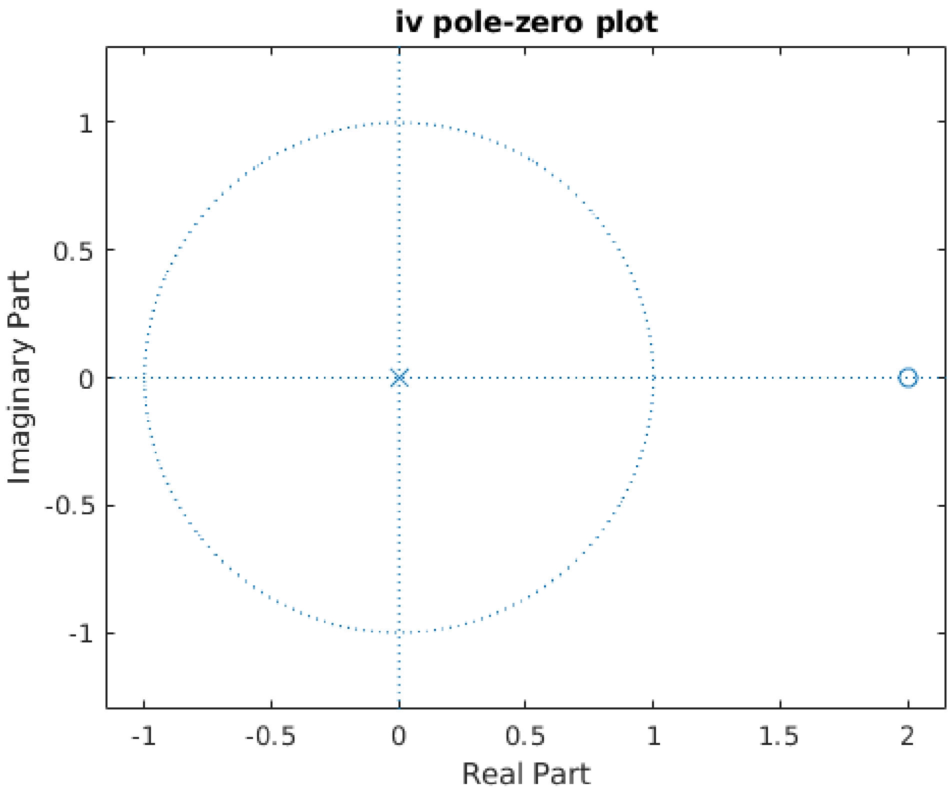
```
b = [1];
y = filter(b,a,impulse);

figure;
subplot(2,1,1)
stem(y);
title('ib filter plot')


[b,a] = zp2tf(zer,poles,1);
fvtool(b,a);
title ( 'Magnitude Response ib')


% The poles at zero correspond with the magnitude response


%ii) h[n] = #[n] - 1.8cos(0.2pi)#[n -1] - 0.81#[n-2]
% H[z] = 1 - 1.8cos(0.2pi)z^-1 - 0.81z^-2
%freq resp: H(e^jw) = 1 - 1.8cos(0.2pi)*e^-jw - 0.81e^-2jw
%ROC: C - {#}
%stable
zer = [-0.429534; 1.88576];
poles = [0;0];

figure;
zplane(zer,poles);
title('iib pole-zero plot');

a = [1 -1.8*cos(0.2*pi) -0.81];
b = [1];
y = filter(b,a,impulse);

figure;
subplot(2,1,1)
stem(y);
title('iib filter plot')

[b,a] = zp2tf(zer,poles,1);
fvtool(b,a);
title ( 'Magnitude Response iib')

% The poles at zero correspond with the magnitude response

%iii) h[n] = #[n] - 1.98cos(0.2pi)#[n -1] - 0.9801#[n-2]
% H[z] = 1 - 1.98cos(0.2pi)z^-1 - 0.9801z^-2
%freq resp: H(e^jw) = 1 - 1.98cos(0.2pi)*e^-jw - 0.9801e^-2jw
%ROC: C - {#}
%stable
zer = [-0.42487; 2.07434];
poles = [0;0];

figure;
zplane(zer,poles);
```

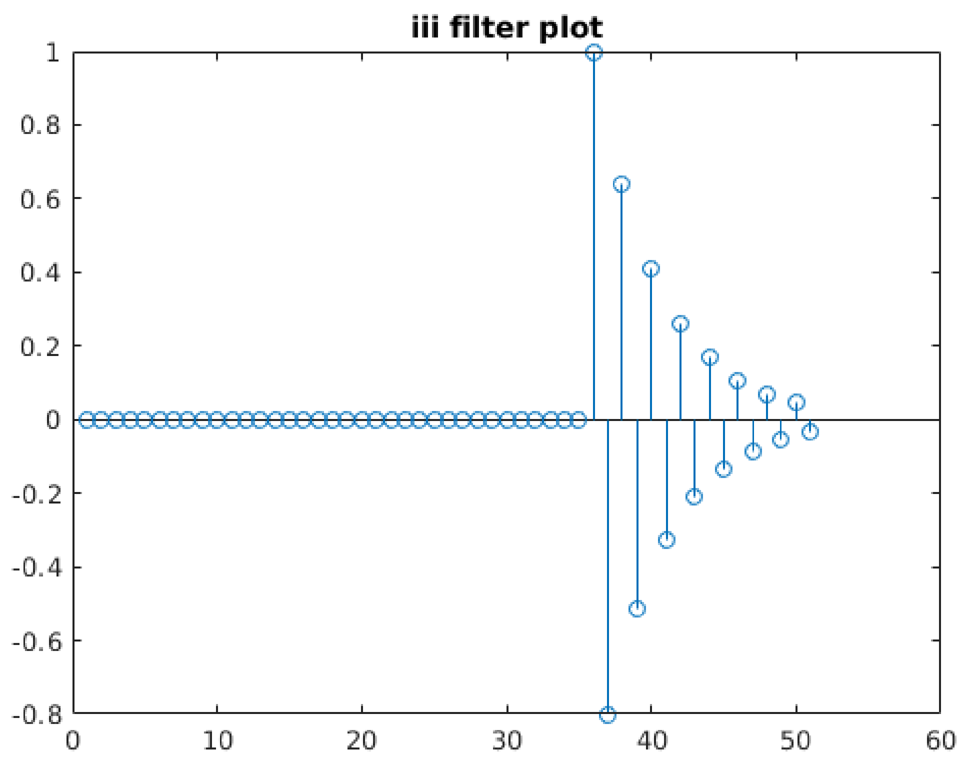
```
title('iiib pole-zero plot');


a = [1 -1.98*cos(0.2*pi) -0.9801];
b = [1];
y = filter(b,a,impulse);


figure;
subplot(2,1,1)
stem(y);
title('iiib filter plot')

[b,a] = zp2tf(zer,poles,1);
fvtool(b,a);
title ( 'Magnitude Response iiib')

% The poles at zero correspond with the magnitude response
```

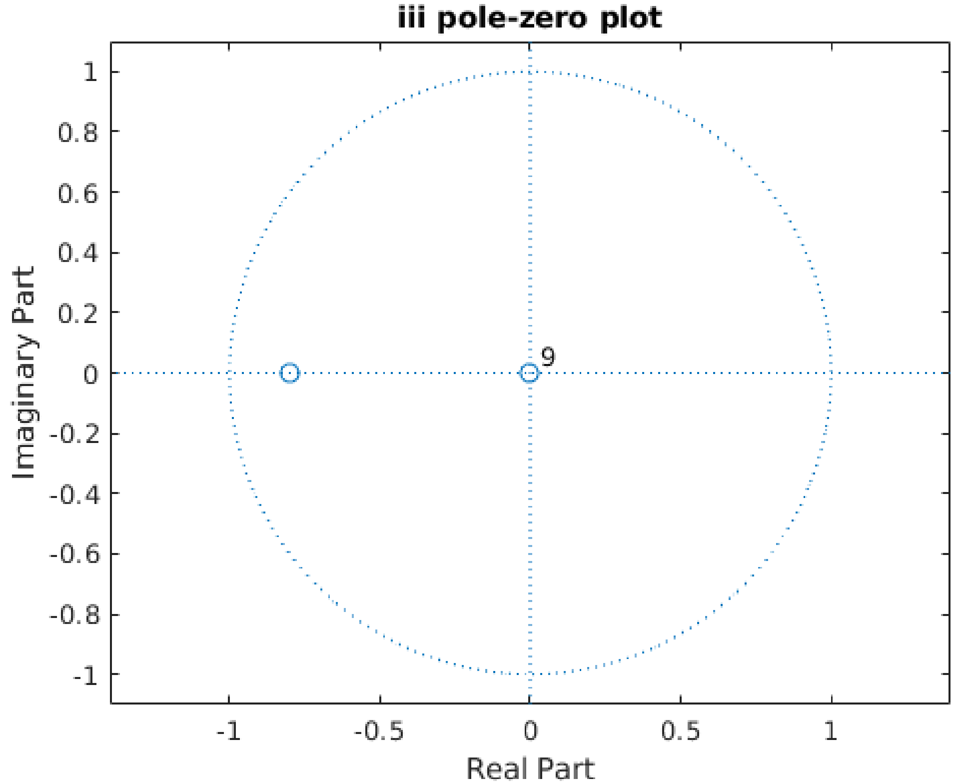


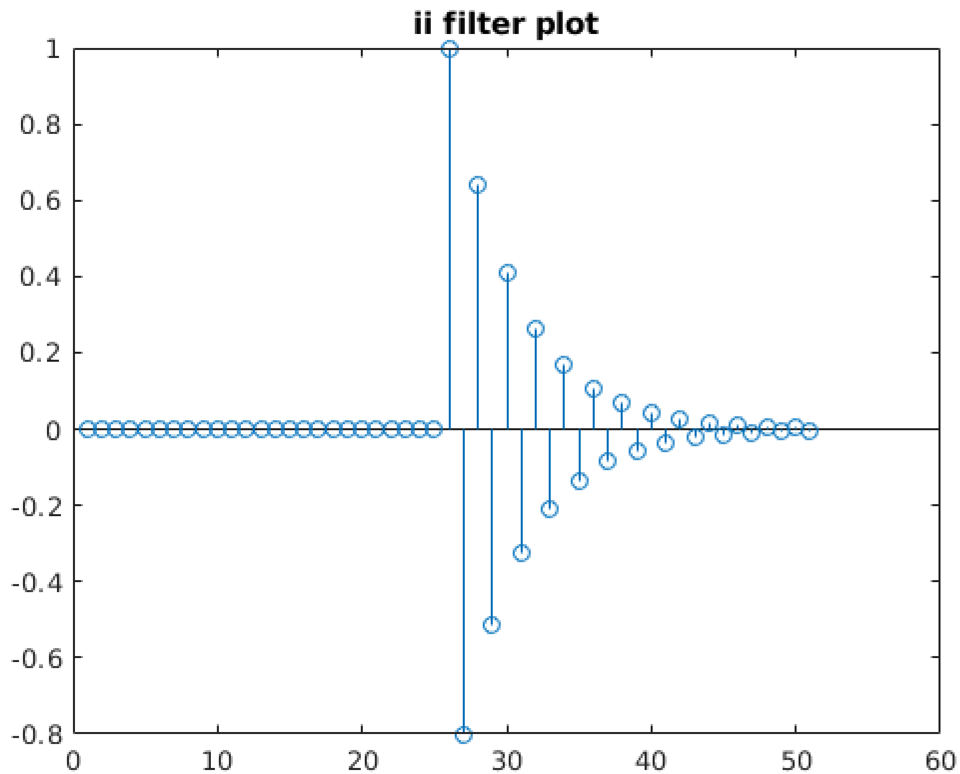

ib pole-zero plot
Imaginary Part
Real Part
2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1

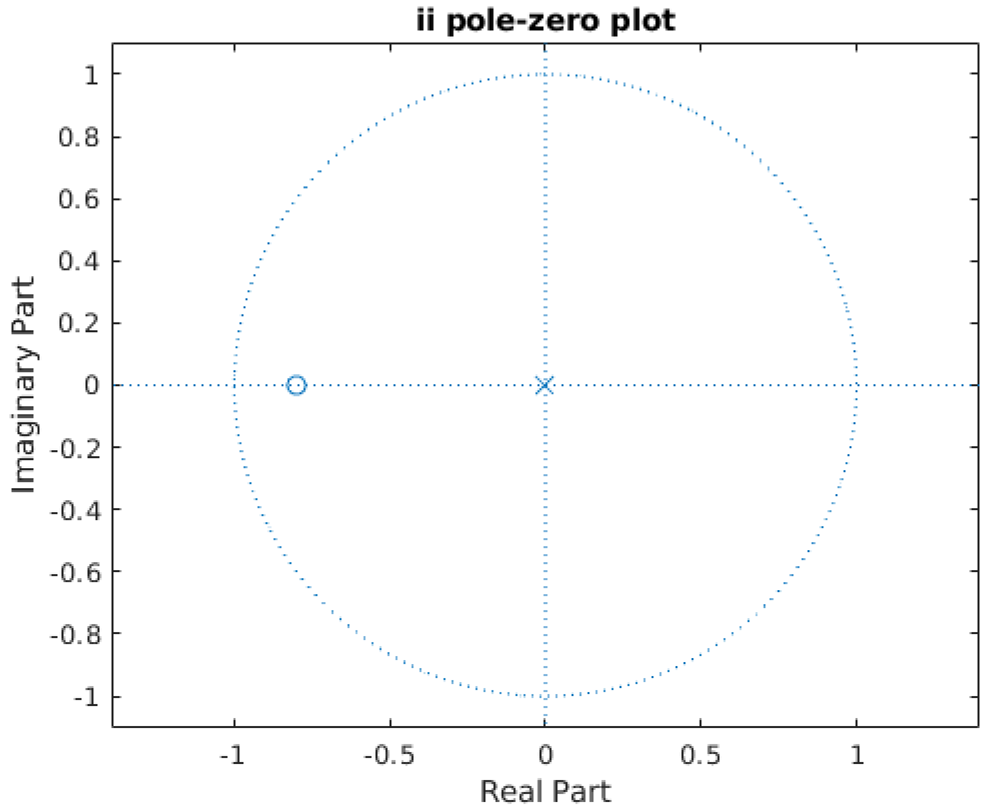

iv filter plot
×10^7
7
6
5
4
3
2
1
0
0
10
20
30
40
50
60

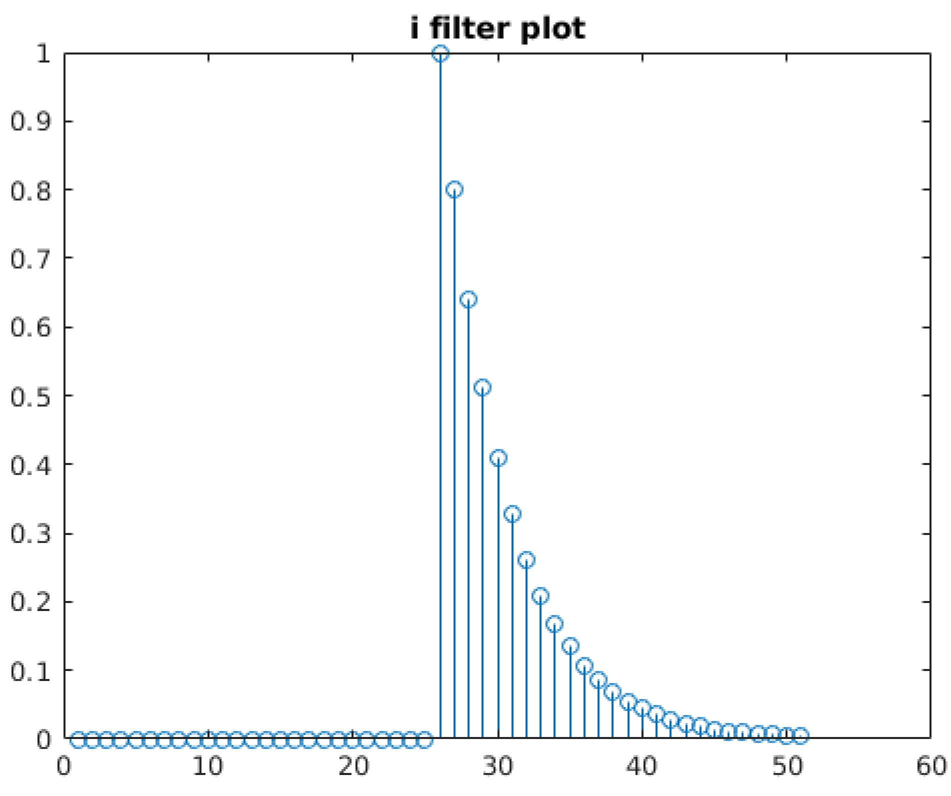

iv pole-zero plot
Imaginary Part
Real Part
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
1.5
2

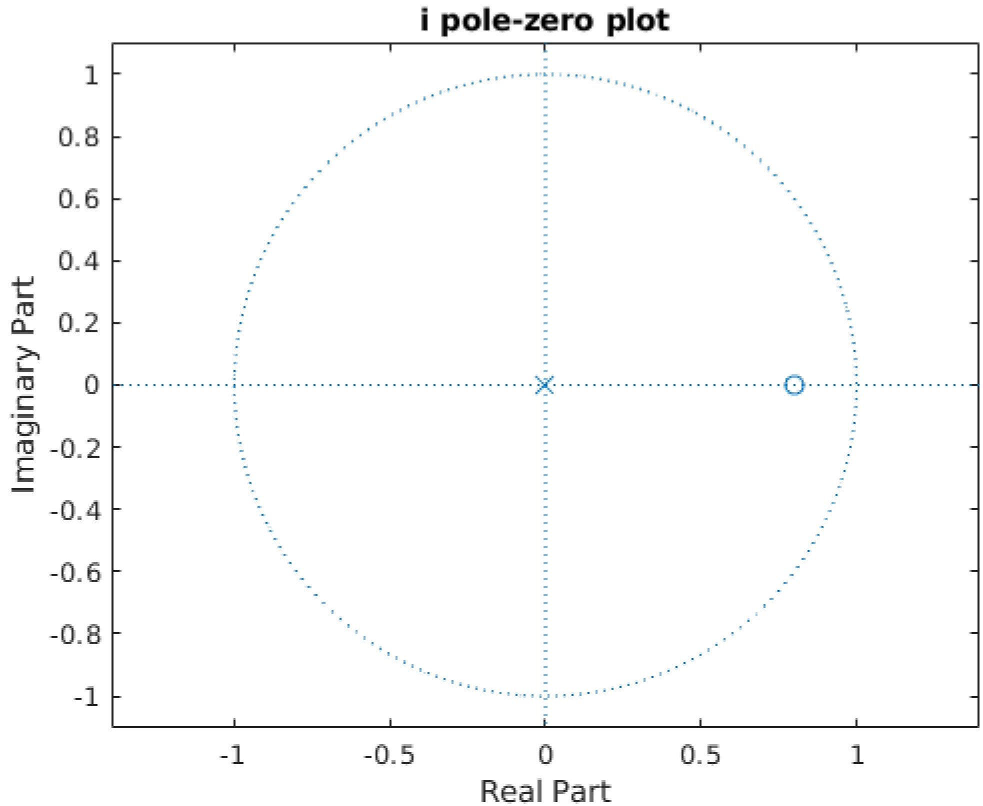

iii filter plot
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
10
20
30
40
50
60

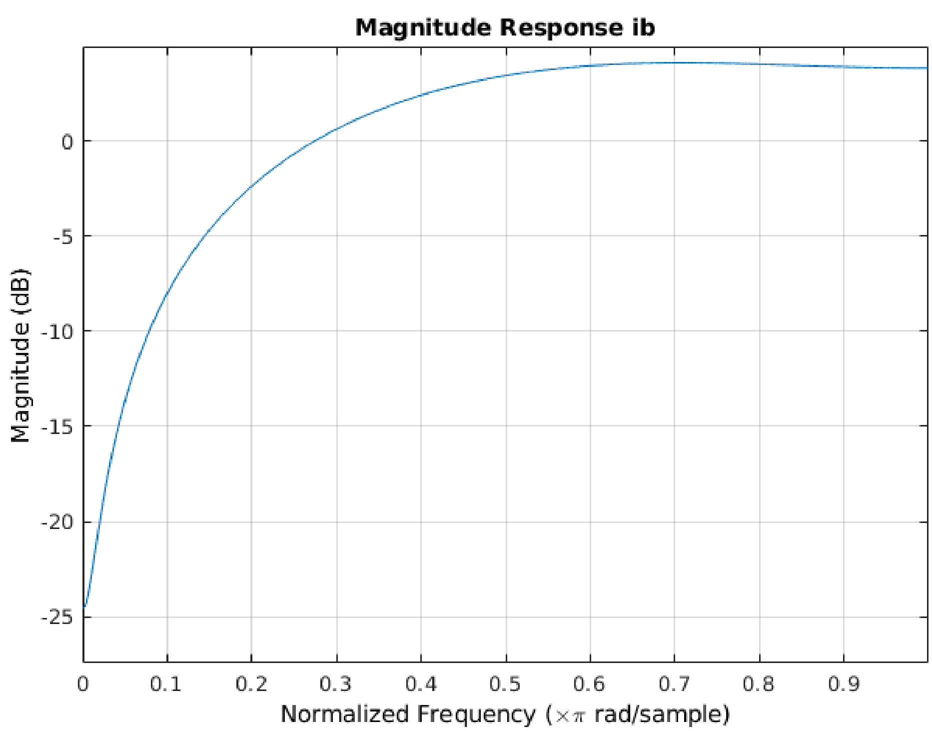

iii pole-zero plot
Imaginary Part
Real Part
9
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1

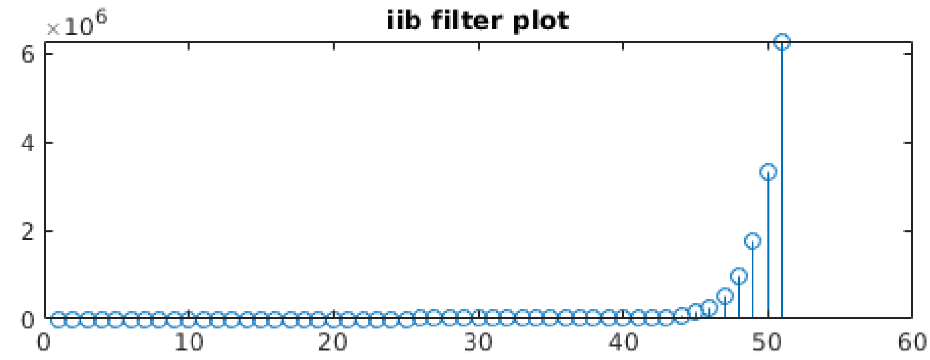

ii filter plot
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
10
20
30
40
50
60

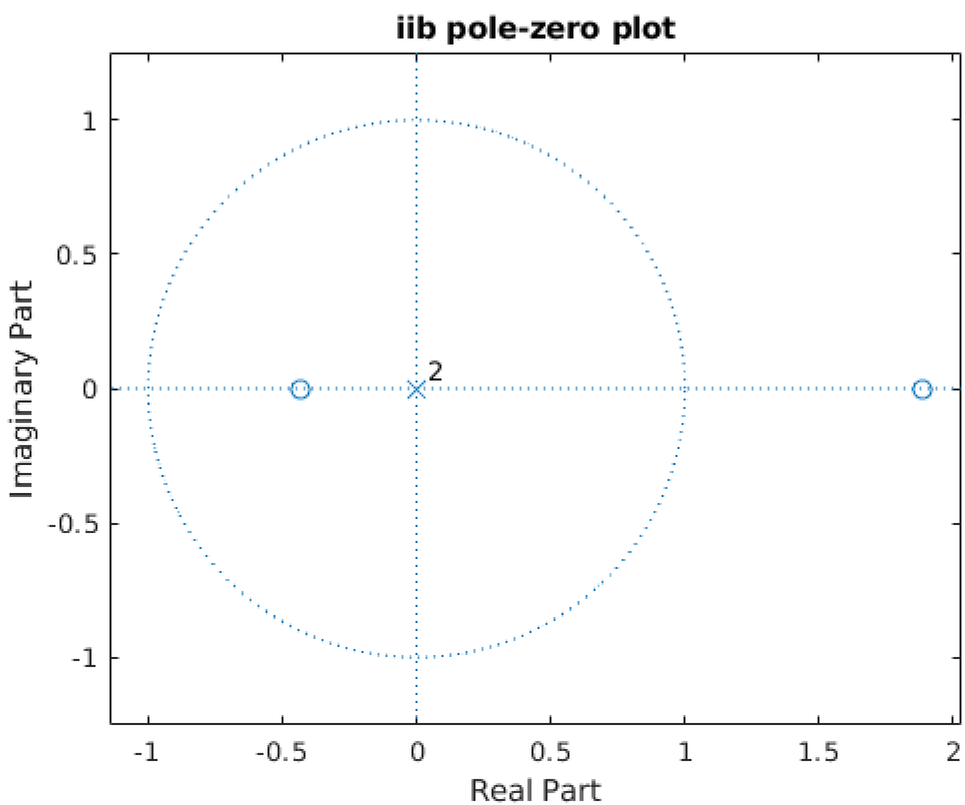

ii pole-zero plot
Imaginary Part
Real Part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1

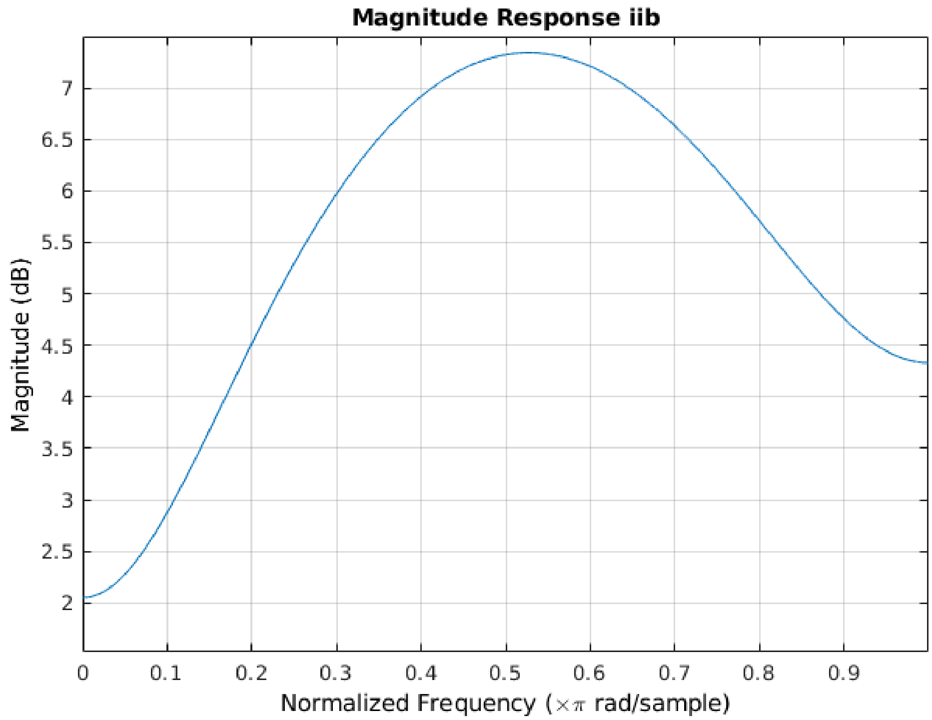

i filter plot
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
60

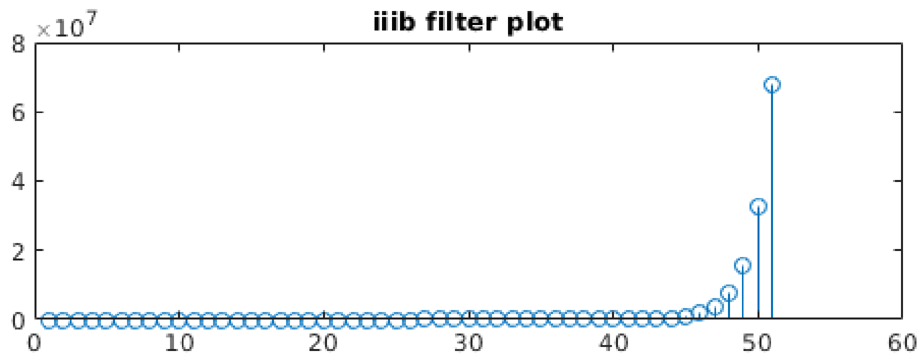

i pole-zero plot
Imaginary Part
Real Part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1

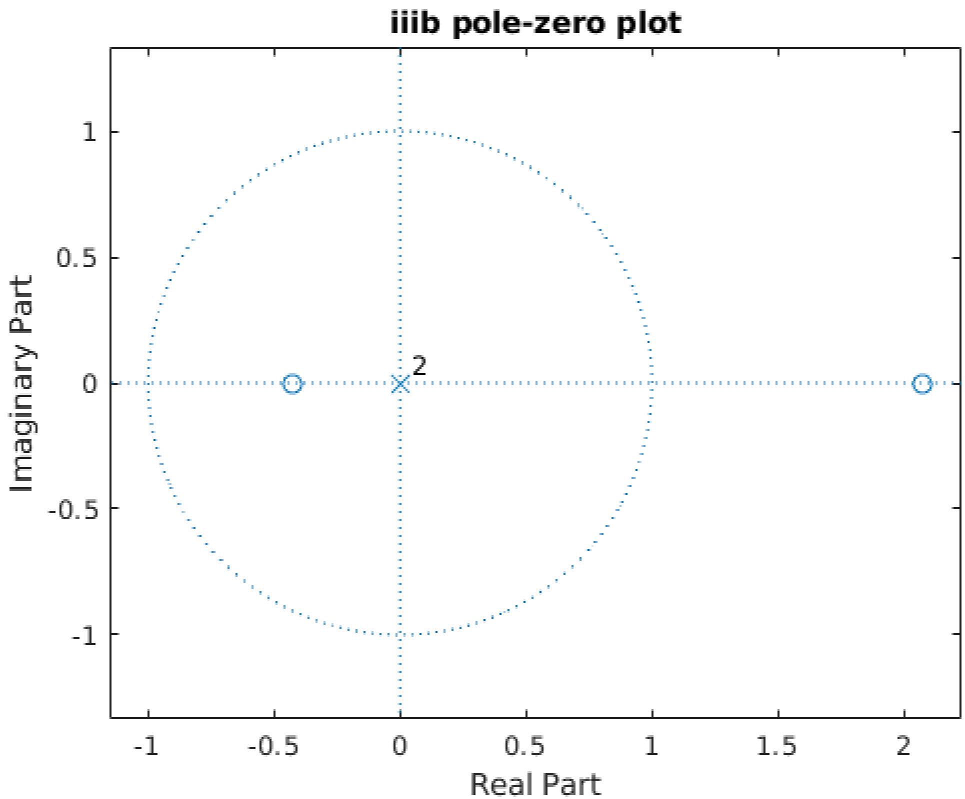

Magnitude Response ib
Magnitude (dB)
Normalized Frequency (×π rad/sample)
0
-5
-10
-15
-20
-25
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

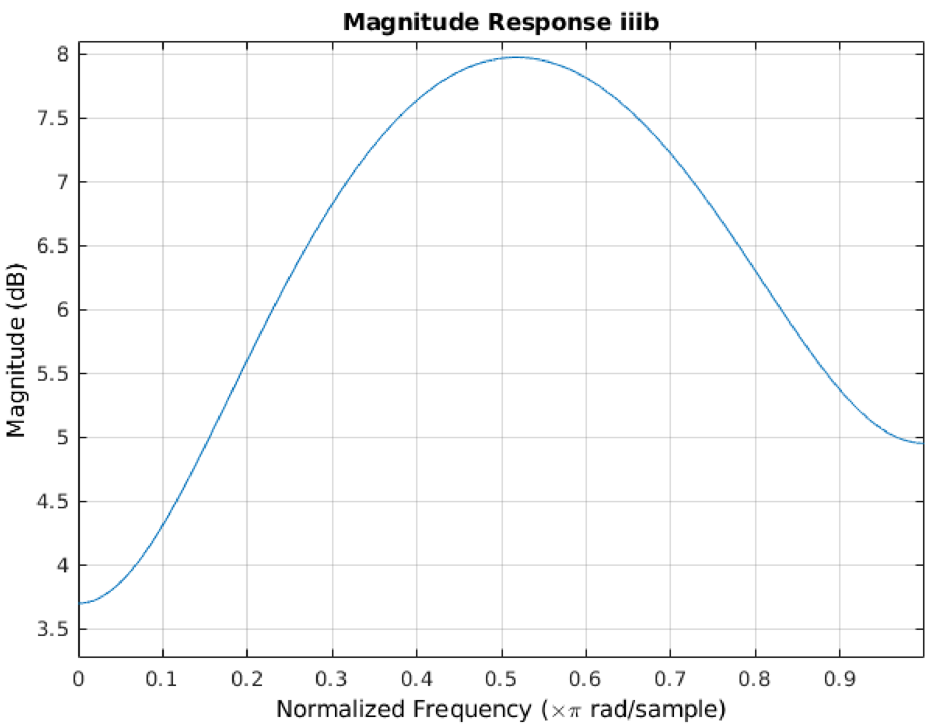

iib filter plot
×10^6
6
4
2
0
0
10
20
30
40
50
60


iib pole-zero plot
Imaginary Part
Real Part
1
0.5
0
-0.5
-1
2
-1
-0.5
0
0.5
1
1.5
2


Magnitude Response iib
Magnitude (dB)
Normalized Frequency (×π rad/sample)
7
6.5
6
5.5
5
4.5
4
3.5
3
2.5
2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9


iiib filter plot
×10^7
8
6
4
2
0
0
10
20
30
40
50
60


iiib pole-zero plot
Imaginary Part
Real Part
2
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
1.5
2

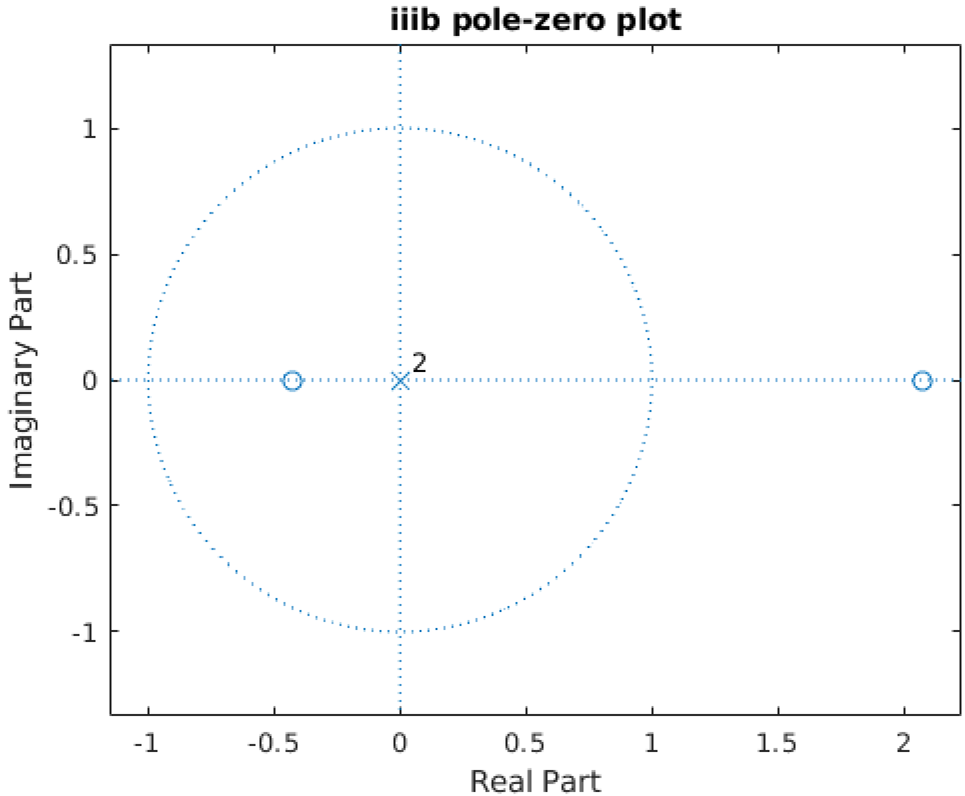

Magnitude Response iiib
Magnitude (dB)
Normalized Frequency (×π rad/sample)
8
7.5
7
6.5
6
5.5
5
4.5
4
3.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

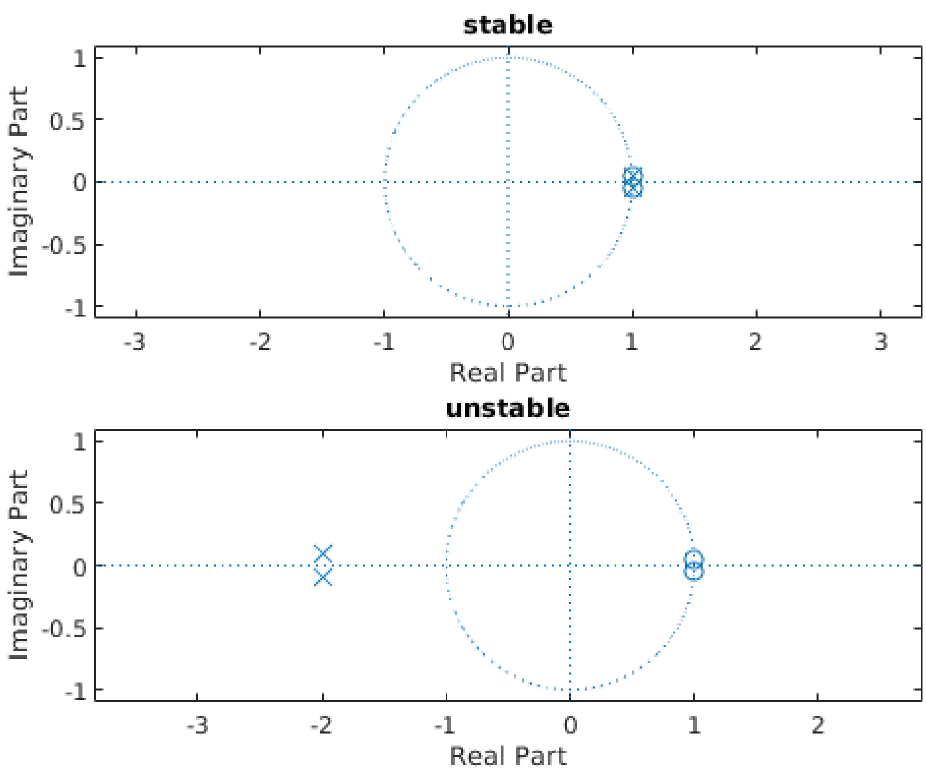
Exercise 7.2

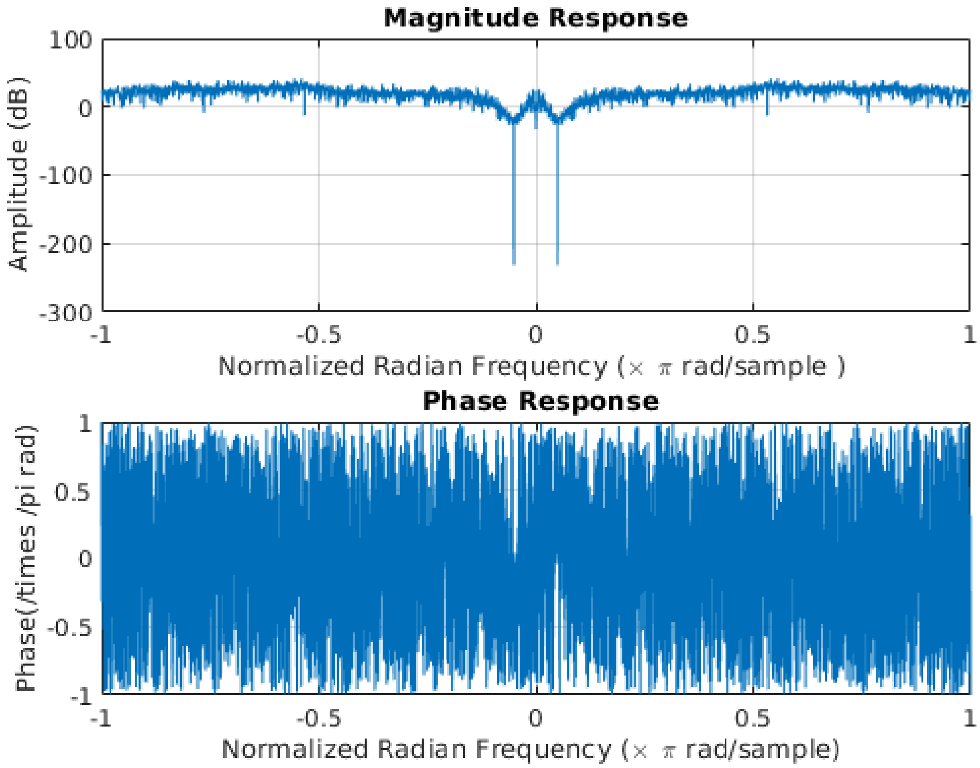
```
%a

[corrupt,Fs] = audioread('bad_wannabe.wav');

w = -pi:pi/2000:pi;
corrupt_dtft =dtft(corrupt,w);

figure;
subplot (2 , 1 , 1)
plot (w/pi,20*log10(abs(corrupt_dtft))) ;
grid on ;
title ( 'Magnitude Response Corrupt')
xlabel ( 'Normalized Radian Frequency (\times \pi rad/sample ) ') ;
ylabel ( ' Amplitude (dB) ') ;


h = [1 -2*cos(0.05*pi) 1];
not_corrupt = conv(corrupt,h);
audiowrite('fired_wannabe.wav',not_corrupt,Fs)
not_corrupt = dtft(not_corrupt,w);

figure;
subplot (2 , 1 , 1)
plot (w/pi,20*log10(abs(not_corrupt))) ;
grid on ;
title ( 'Magnitude Response')
xlabel ( 'Normalized Radian Frequency (\times \pi rad/sample ) ') ;
ylabel ( ' Amplitude (dB) ') ;
subplot (2 , 1 , 2)
plot ( w / pi , angle ( not_corrupt) / pi ) ;
grid on ;
title ( ' Phase Response')
xlabel ( 'Normalized Radian Frequency (\times \pi rad/sample) ');
ylabel('Phase(/times /pi rad)');


%The filtered audio quality dropped dramatically.


%b)
alp = 1;
zer = [exp(j*0.05);exp(-j*0.05)];
poles = [alp*exp(j*0.05);alp*exp(-j*0.05)];

% The filter is stable with -1 <= alpha <= 1

figure;
subplot(2,1,1);
zplane(zer,poles);
title('stable')
```

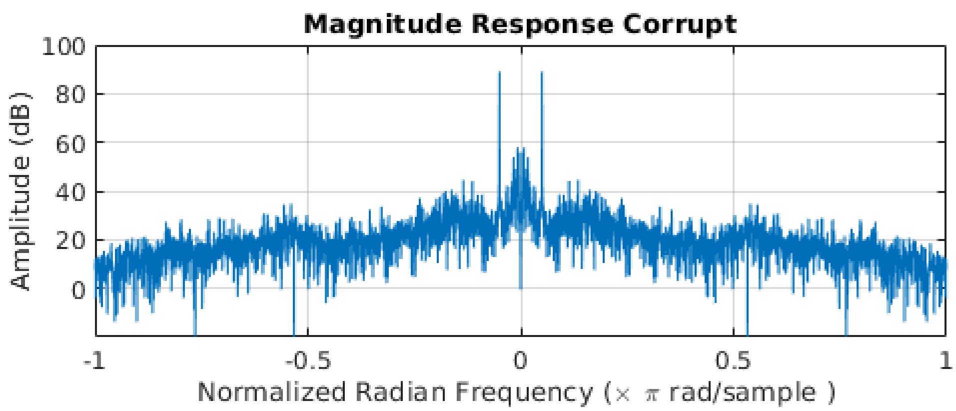
```
alp = -2;
zer = [exp(j*0.05);exp(-j*0.05)];
poles = [alp*exp(j*0.05);alp*exp(-j*0.05)];

subplot(2,1,2)
zplane(zer,poles);
title('unstable')

alp = 0;
zer = [exp(j*0.05);exp(-j*0.05)];
poles = [alp*exp(j*0.05);alp*exp(-j*0.05)];

[b,a] = zp2tf(zer,poles,1);
fvtool(b,a);
title('alpha = 0');

alp = 0.25;
zer = [exp(j*0.05);exp(-j*0.05)];
poles = [alp*exp(j*0.05);alp*exp(-j*0.05)];


[b,a] = zp2tf(zer,poles,1);
fvtool(b,a);
title('alpha = 0.25');

alp = 0.5;
zer = [exp(j*0.05);exp(-j*0.05)];
poles = [alp*exp(j*0.05);alp*exp(-j*0.05)];

[b,a] = zp2tf(zer,poles,1);
fvtool(b,a);
title('alpha = 0.5');

alp = 0.75;
zer = [exp(j*0.05);exp(-j*0.05)];
poles = [alp*exp(j*0.05);alp*exp(-j*0.05)];

[b,a] = zp2tf(zer,poles,1);
fvtool(b,a);
title('alpha = 0.75');

alp = 1;
zer = [exp(j*0.05);exp(-j*0.05)];
poles = [alp*exp(j*0.05);alp*exp(-j*0.05)];

[b,a] = zp2tf(zer,poles,1);
fvtool(b,a);
title('alpha = 1');


% The notch seems to be exponentially decreasing as the alpha value is
% increased, with 0 notch at 1.

%c)
```

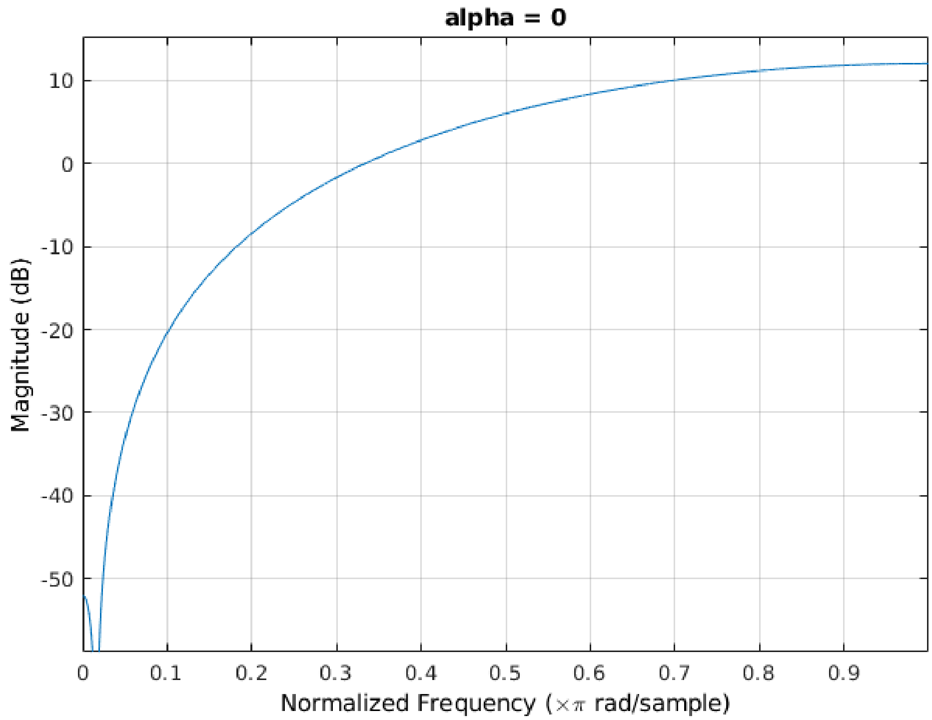
```
% difference equation
% y[n] - 2cos(w0))y[n-1] + y[n-2] = x[n] - 2*alpha*cos(w0))x[n-1] +
% alpha^2 x[n-2]
type notch

%d)
[b,a] = notch(0.05*pi, 0.99);

[bad,Fs] = audioread('bad_wannabe.wav');

good = filter(b,a,bad);

audiowrite('iired_wannabe.wav',good,Fs)


% The filtered audio is much more clear than when using then other
 nulling
% filter


w = -pi:pi/2000:pi;
good =dtft(good,w);

figure;
plot (w/pi,20*log10(abs(good))) ;
grid on ;
title ( 'Magnitude Response Corrupt')
xlabel ( 'Normalized Radian Frequency (\times \pi rad/sample ) ') ;
ylabel ( ' Amplitude (dB) ') ;


% The magnitude response shows that the notch filter did not
 completely
% eliminate the frequency at the where there was interference, but
 reduced
% it, leading to better audio quality


function [b,a] = notch(w0 , alpha)
b = [1 -2*cos(w0) 1];
a = [1 -2*alpha*cos(w0) alpha*alpha];
```

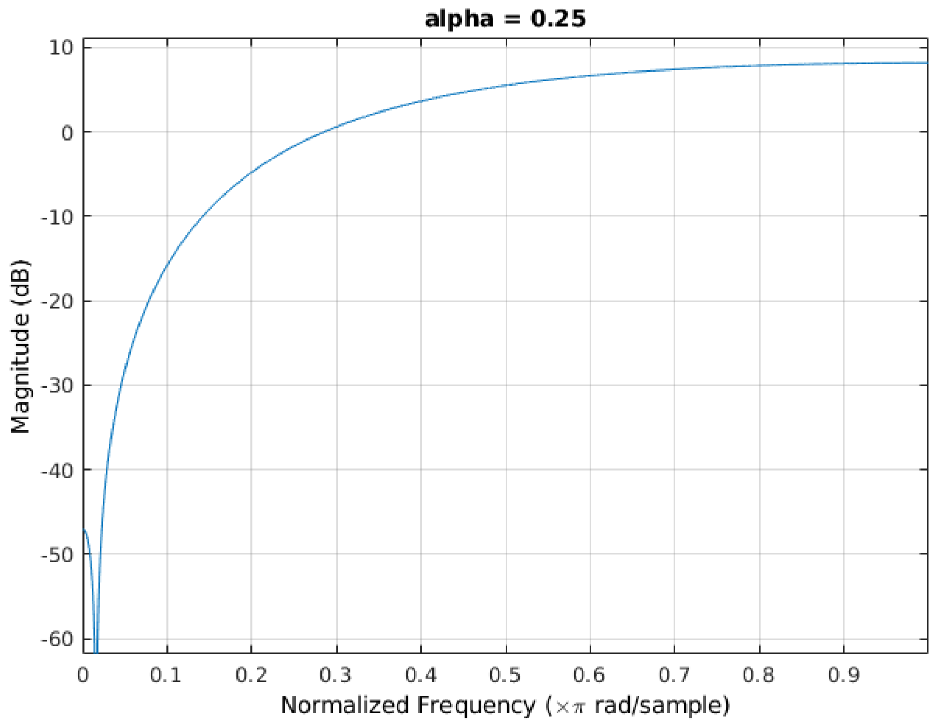

iiib pole-zero plot
Imaginary Part
Real Part
2
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
1.5
2

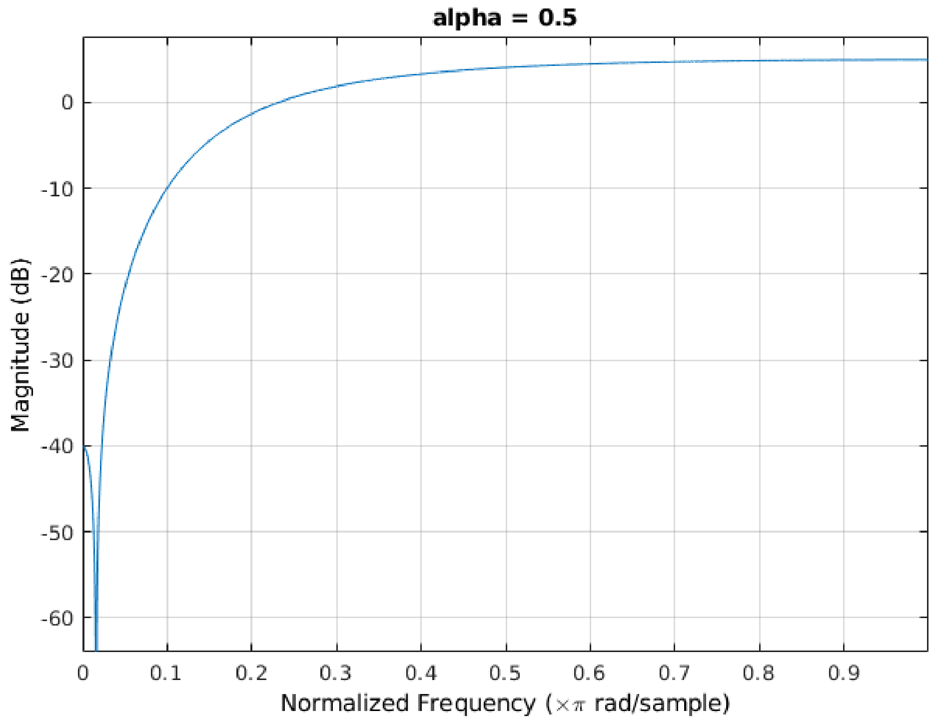

stable
unstable
Imaginary Part
Real Part
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3

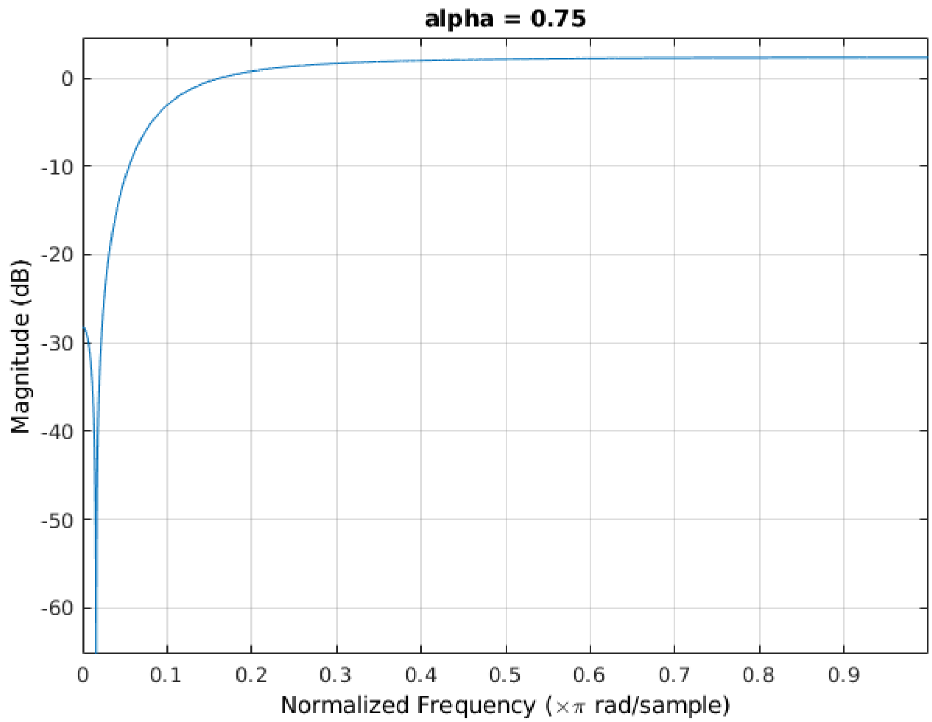

Magnitude Response
Amplitude (dB)
Normalized Radian Frequency (× π rad/sample)
Phase Response
Phase(/times /pi rad)
Normalized Radian Frequency (× π rad/sample)

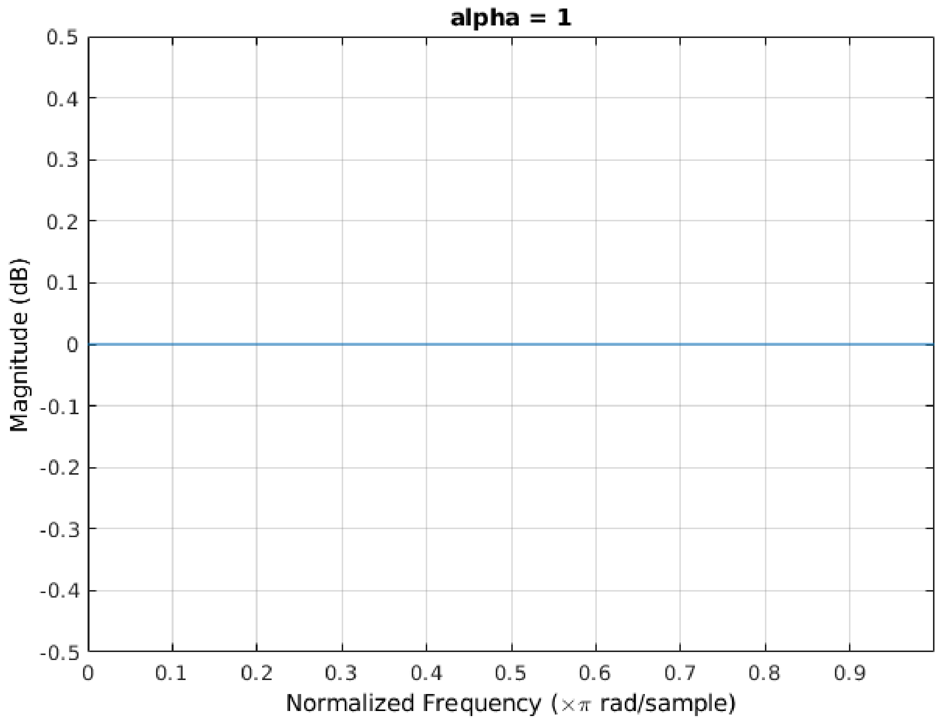

Magnitude Response Corrupt
Amplitude (dB)
Normalized Radian Frequency (× π rad/sample)

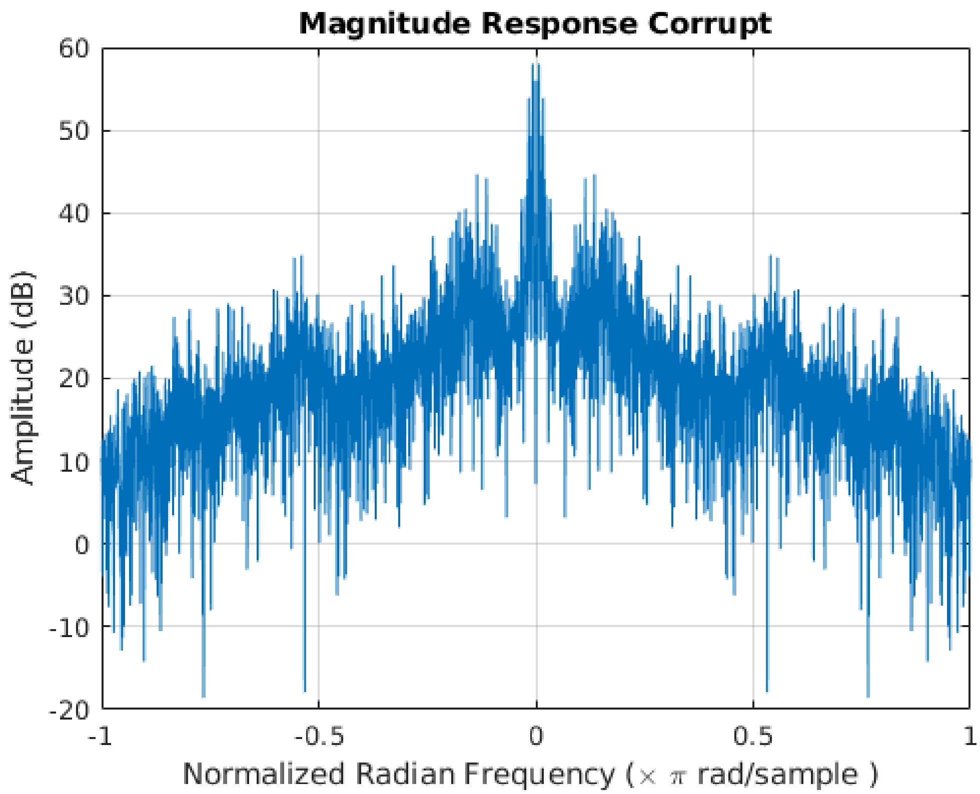

alpha = 0
Magnitude (dB)
10
0
-10
-20
-30
-40
-50
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Normalized Frequency (×π rad/sample)


alpha = 0.25
Magnitude (dB)
10
0
-10
-20
-30
-40
-50
-60
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Normalized Frequency (×π rad/sample)


alpha = 0.5
0
-10
-20
-30
-40
-50
-60
Magnitude (dB)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Normalized Frequency (×π rad/sample)


alpha = 0.75
0
-10
-20
-30
-40
-50
-60
Magnitude (dB)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Normalized Frequency (×π rad/sample)





Published with MATLAB® R2020a