
Koby Miller Lab7

Table of Contents

Exercise 7.1	1
Exercise 7.2	21
My Functions	24
Pictures of handwritten work	24

Exercise 7.1

```
figure_counter = 1;

%(a)
%(i)  $y[n] = 0.8y[n-1] + x[n]$ 
a = [1 -0.8];
b = [1];

delta = 0;
delta(1) = 1;
delta(2:20)=0;

figure(figure_counter);
figure_counter = figure_counter + 1;
i_filter = filter(b,a,delta);
stem(i_filter);
grid on;
title('Filter generated impulse of (a) i');
xlabel('n');

n = 0:19;
u=0;
u(1:20) = 1;
i_y = (0.8).^n;
i_y = i_y.*u;
figure(figure_counter);
figure_counter = figure_counter + 1;
stem(i_y);
grid on;
title('Analytically generated impulse of (a) i');
xlabel('n');

figure(figure_counter);
figure_counter = figure_counter + 1;
zplane(0.8)
grid on;
title('pole-zero plot of (a) i');
```

```
%(ii) y[n] = -0.8y[n # 1] + x[n]
a = [1 0.8];
b = [1];

delta = 0;
delta(1) = 1;
delta(2:20)=0;

figure.figure_counter;
figure_counter = figure_counter + 1;
ii_filter = filter(b,a,delta);
stem(ii_filter);
grid on;
title('Filter generated impulse of (a) ii');
xlabel('n');

n = 0:19;
u=0;
u(1:20) = 1;
ii_y = (-0.8).^n;
ii_y = ii_y.*u;
figure.figure_counter;
figure_counter = figure_counter + 1;
stem(ii_y);
grid on;
title('Analytically generated impulse of (a) ii');
xlabel('n');

figure.figure_counter;
figure_counter = figure_counter + 1;
zplane(-0.8)
grid on;
title('pole-zero plot of (a) ii');

%(iii)
a = [1 0.8];
b = [1];

delta = 0;
delta(1:30)=0;
delta(10) = 1;

figure.figure_counter;
figure_counter = figure_counter + 1;
iii_filter = filter(b,a,delta);
stem(iii_filter);
grid on;
title('Filter generated impulse of (a) iii');
xlabel('n');

n = 0:19;
u=0;
```

```
u(1:20) = 1;
n_shifted = [-10:-1, n];
u_shifted = [zeros(1,10), u];

iii_y = (-0.8).^n_shifted;
iii_y = iii_y.*u_shifted
figure.figure_counter;
figure_counter = figure_counter + 1;
stem(iii_y);
grid on;
title('Analytically generated impulse of (a) iii');
xlabel('n');

figure.figure_counter;
figure_counter = figure_counter + 1;
zplane(0,-0.8)
grid on;
title('pole-zero plot of (a) iii');

%(iv) y[n] = 2y[n # 1] + 2x[n]
a = [1 -2];
b = [2];

delta = 0;
delta(1) = 1;
delta(2:10)=0;

figure.figure_counter;
figure_counter = figure_counter + 1;
iv_filter = filter(b,a,delta);
stem(iv_filter);
grid on;
title('Filter generated impulse of (a) iv');
xlabel('n');

n = 0:9;
u=0;
u(1:10) = 1;
iv_y = 2*(2.^n);
iv_y = iv_y.*u;
figure.figure_counter;
figure_counter = figure_counter + 1;
stem(iv_y);
grid on;
title('Analytically generated impulse of (a) iv');
xlabel('n');

figure.figure_counter;
figure_counter = figure_counter + 1;
zplane(2)
grid on;
title('pole-zero plot of (a) iv');

%(b)
```

```
%(i) y[n] = 1.0 cos(0.2#)y[n # 1] # 0.25y[n # 2] + x[n]
a = [1 -cos(0.2*pi) 0.25];
b = [1];

delta = 0;
delta(1) = 1;
delta(2:20)=0;

figure.figure_counter;
figure_counter = figure_counter + 1;
i_filter = filter(b,a,delta);
stem(i_filter);
grid on;
title('Filter generated impulse of (b) i');
xlabel('n');

n = 0:19;
i_y = 0;
i_y(1) = 1;
i_y(2) = cos(0.2*pi);
for k = 3:length(n)
    i_y(k) = (cos(0.2*pi)*i_y(k-1)) - 0.25*i_y(k-2);
end

figure.figure_counter;
figure_counter = figure_counter + 1;
stem(ii_y);
grid on;
title('Analytically generated impulse of (b) i');
xlabel('n');

temp = [1, cos(0.2*pi), 0.25];
temp = roots(temp);
figure.figure_counter;
figure_counter = figure_counter + 1;
zplane(temp);
grid on;
title('pole-zero plot of (b) i');

%it is stable so we need to find the freq response
w = -pi:pi/1000:pi;
freq_response = 1./(1-(cos(0.2*pi)./exp(j.*w))+(0.25./
(exp(j.*w)).^2));
figure.figure_counter;
figure_counter = figure_counter + 1;
stem(freq_response);
grid on;
title('Frequency Response of (b) i');

figure.figure_counter;
figure_counter = figure_counter + 1;
plot(w/pi, 20*log10(abs(freq_response)));
grid on;
```

```
title('Magnitude Response of (b) i)');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

%(ii) y[n] = 1.8 cos(0.2#)y[n # 1] # 0.81y[n # 2] + x[n]
a = [1 -1.8*cos(0.2*pi) 0.81];
b = [1];

delta = 0;
delta(1) = 1;
delta(2:50)=0;

figure.figure_counter);
figure_counter = figure_counter + 1;
ii_filter = filter(b,a,delta);
stem(ii_filter);
grid on;
title('Filter generated impulse of (b) ii)');
xlabel('n');

n = 0:49;
ii_y = 0;
ii_y(1) = 1;
ii_y(2) = 1.8*cos(0.2*pi);
for k = 3:length(n)
    ii_y(k) = (1.8*cos(0.2*pi)*ii_y(k-1)) - 0.81*ii_y(k-2);
end

figure.figure_counter);
figure_counter = figure_counter + 1;
stem(ii_y);
grid on;
title('Analytically generated impulse of (b) ii)');
xlabel('n');

temp = [1, -1.8*cos(0.2*pi), 0.81];
temp = roots(temp);
figure.figure_counter);
figure_counter = figure_counter + 1;
zplane(temp);
grid on;
title('pole-zero plot of (b) ii)');

%it is stable so we need to find the freq response
w = -pi:pi/1000:pi;
freq_response = 1./(1-(1.8*cos(0.2*pi)./exp(j.*w))+(0.81./
(exp(j.*w)).^2));
figure.figure_counter);
figure_counter = figure_counter + 1;
stem(freq_response);
grid on;
title('Frequency Response of (b) ii)');
```

```
figure.figure_counter);
figure_counter = figure_counter + 1;
plot(w/pi, 20*log10(abs(freq_response)));
grid on;
title('Magnitude Response of (b) ii');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

%(iii) y[n] = 1.98 cos(0.2#)y[n # 1] # 0.9801y[n # 2] + x[n]
a = [1 -1.98*cos(0.2*pi) 0.9801];
b = [1];

delta = 0;
delta(1) = 1;
delta(2:500)=0;

figure.figure_counter);
figure_counter = figure_counter + 1;
iii_filter = filter(b,a,delta);
stem(iii_filter);
grid on;
title('Filter generated impulse of (b) iii');
xlabel('n');

n = 0:499;
iii_y = 0;
iii_y(1) = 1;
iii_y(2) = 1.98*cos(0.2*pi);
for k = 3:length(n)
    iii_y(k) = (1.98*cos(0.2*pi)*iii_y(k-1)) - 0.9801*iii_y(k-2);
end

figure.figure_counter);
figure_counter = figure_counter + 1;
stem(iii_y);
grid on;
title('Analytically generated impulse of (b) iii');
xlabel('n');

temp = [1, -1.98*cos(0.2*pi), 0.9801];
temp = roots(temp);
figure.figure_counter);
figure_counter = figure_counter + 1;
zplane(temp);
grid on;
title('pole-zero plot of (b) iii');

%it is stable so we need to find the freq response
w = -pi:pi/1000:pi;
freq_response = 1./(1-(1.98*cos(0.2*pi)./exp(j.*w))+(0.9801./
(exp(j.*w)).^2));
figure.figure_counter);
```

```
figure_counter = figure_counter + 1;
stem(freq_response);
grid on;
title('Frequency Response of (b) iii');

figure_counter = figure_counter + 1;
plot(w/pi, 20*log10(abs(freq_response)));
grid on;
title('Magnitude Response of (b) iii');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');
```

```
iii_y =
```

Columns 1 through 7

0	0	0	0	0	0	0
---	---	---	---	---	---	---

Columns 8 through 14

0	0	0	1.0000	-0.8000	0.6400	-0.5120
---	---	---	--------	---------	--------	---------

Columns 15 through 21

0.4096	-0.3277	0.2621	-0.2097	0.1678	-0.1342	0.1074
--------	---------	--------	---------	--------	---------	--------

Columns 22 through 28

-0.0859	0.0687	-0.0550	0.0440	-0.0352	0.0281	-0.0225
---------	--------	---------	--------	---------	--------	---------

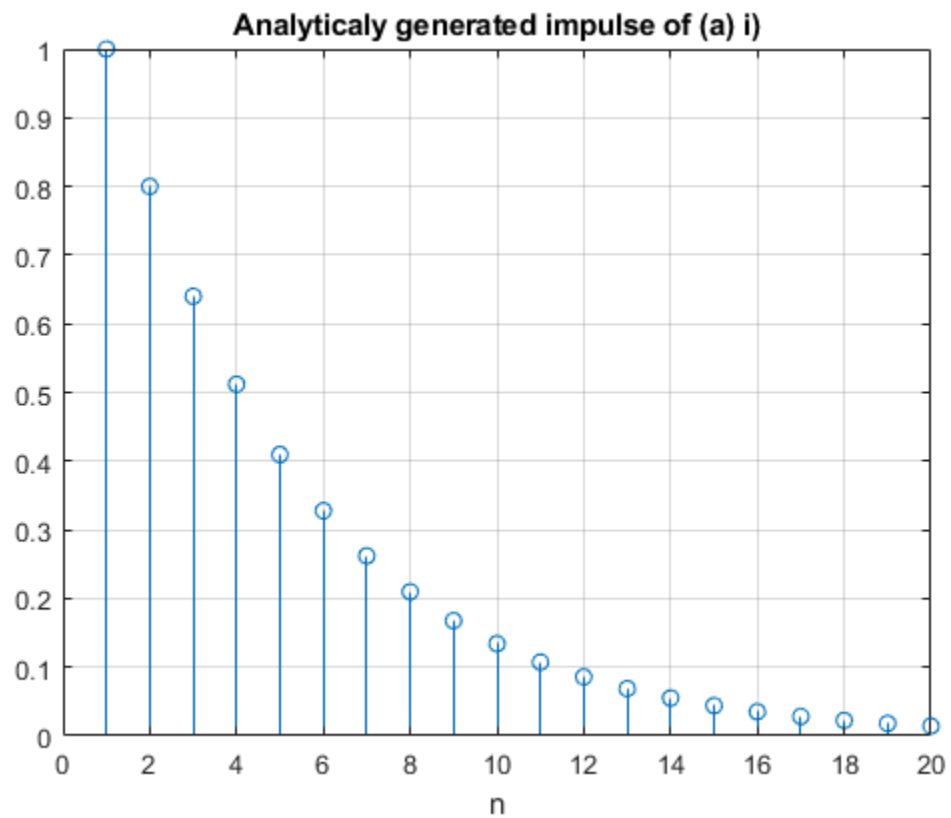
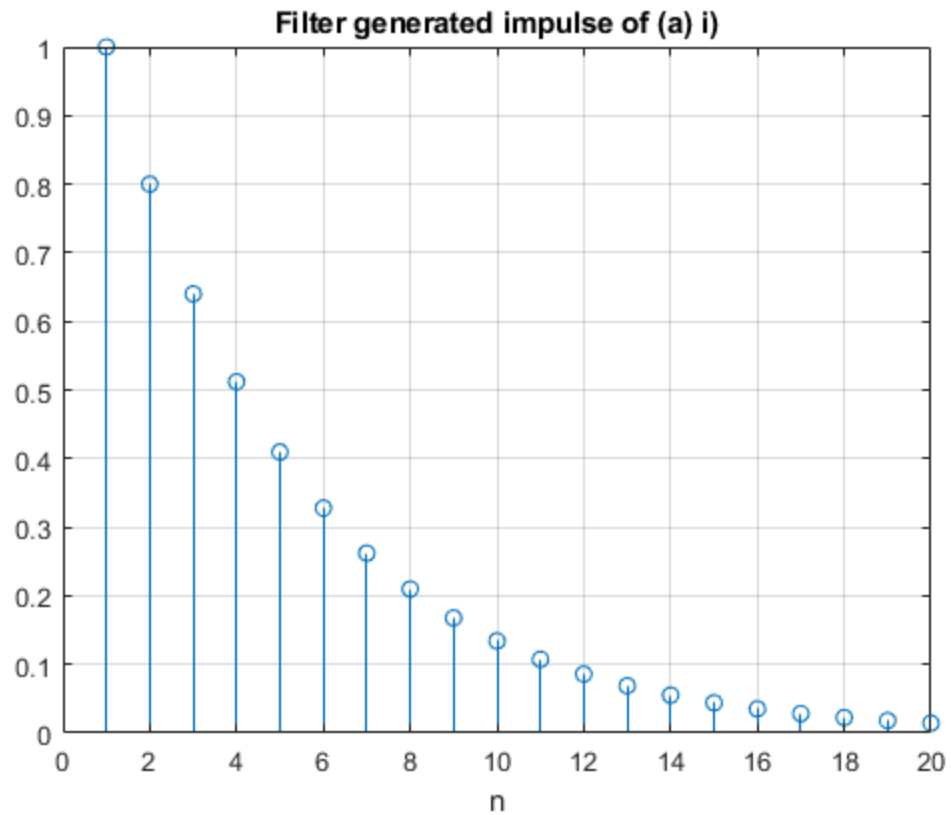
Columns 29 through 30

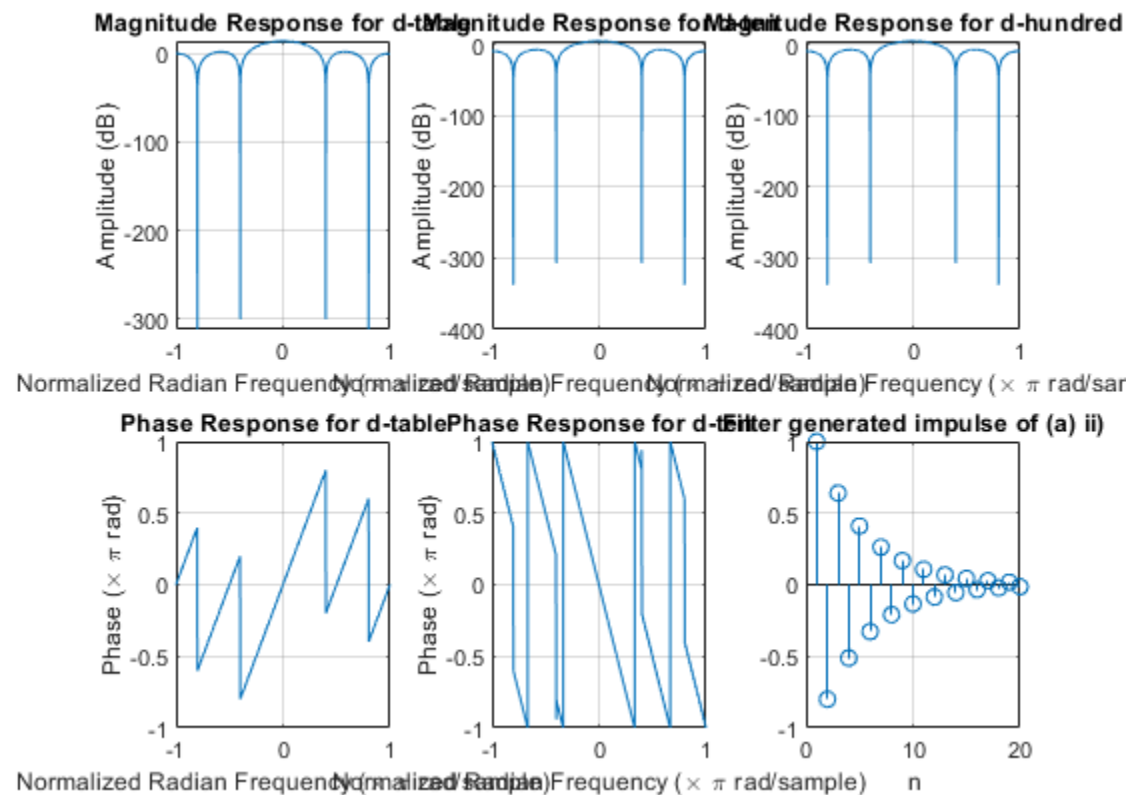
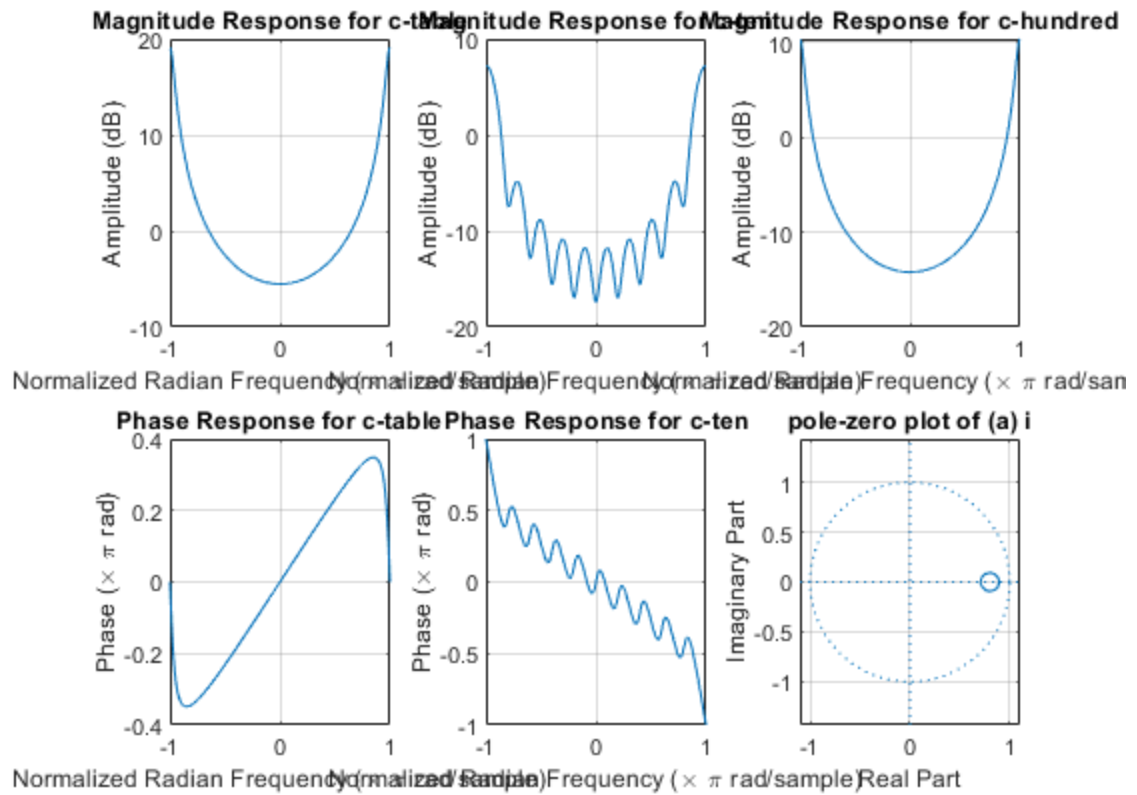
0.0180	-0.0144
--------	---------

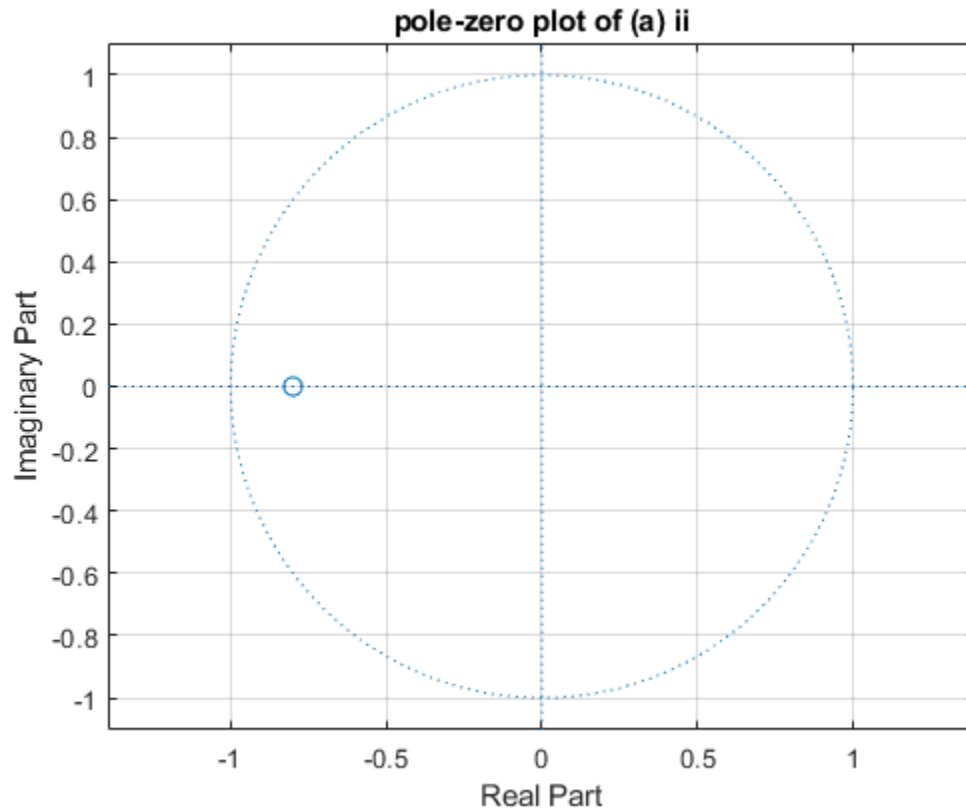
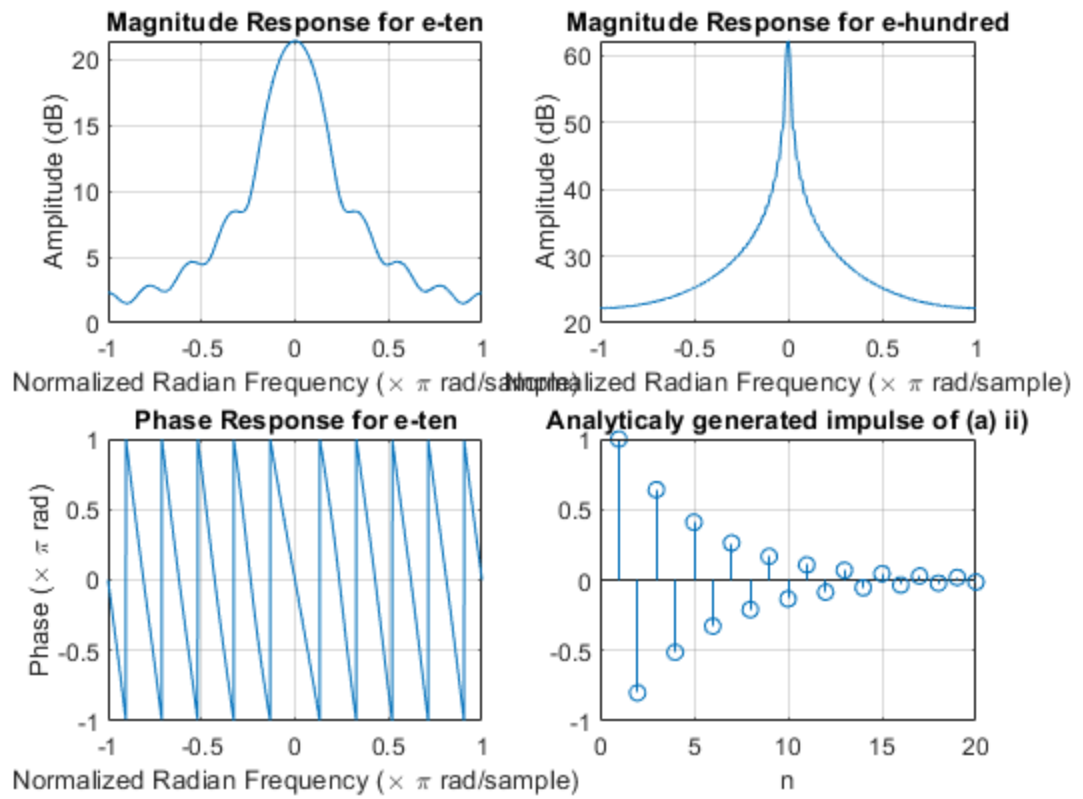
Warning: Using only the real component of complex data.

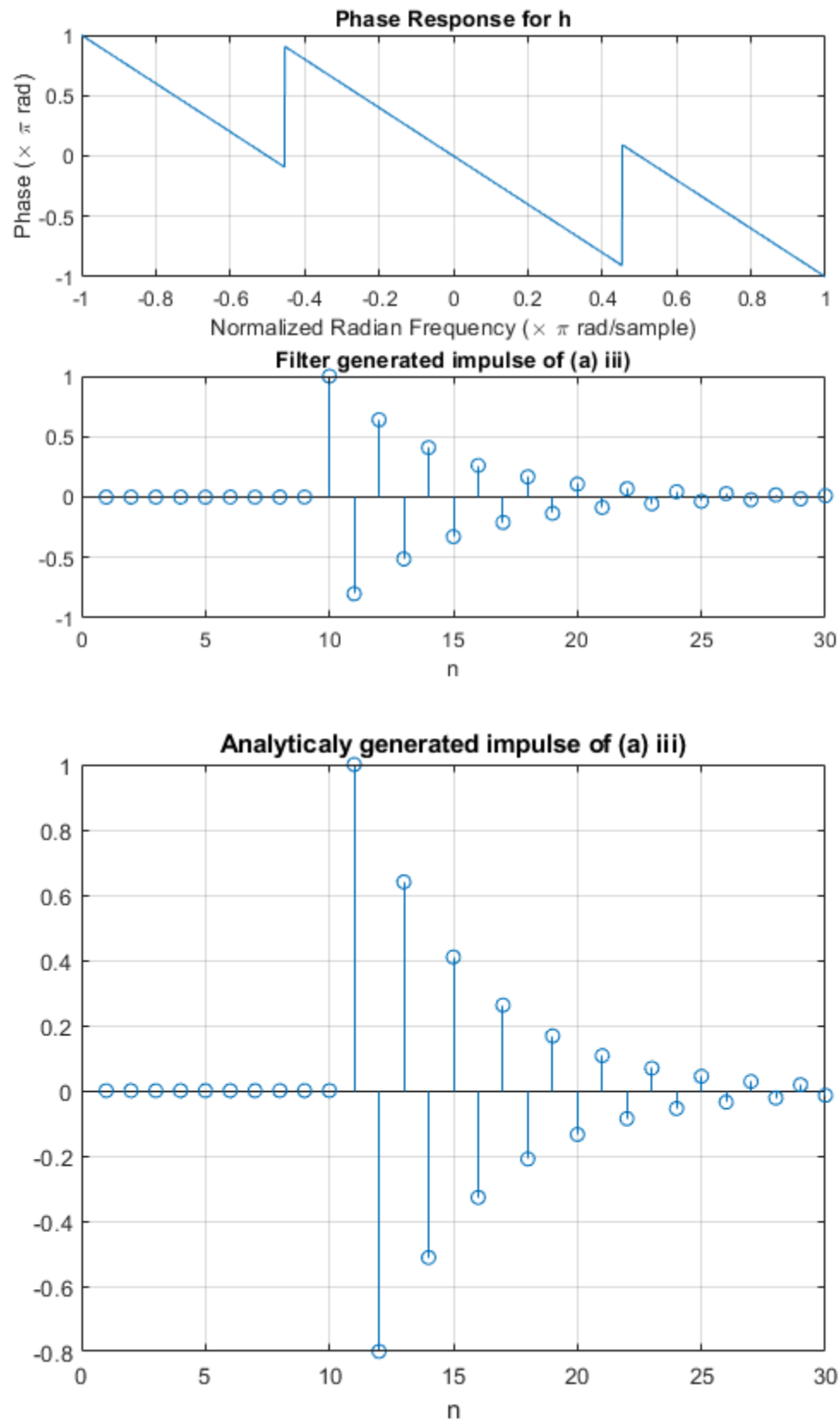
Warning: Using only the real component of complex data.

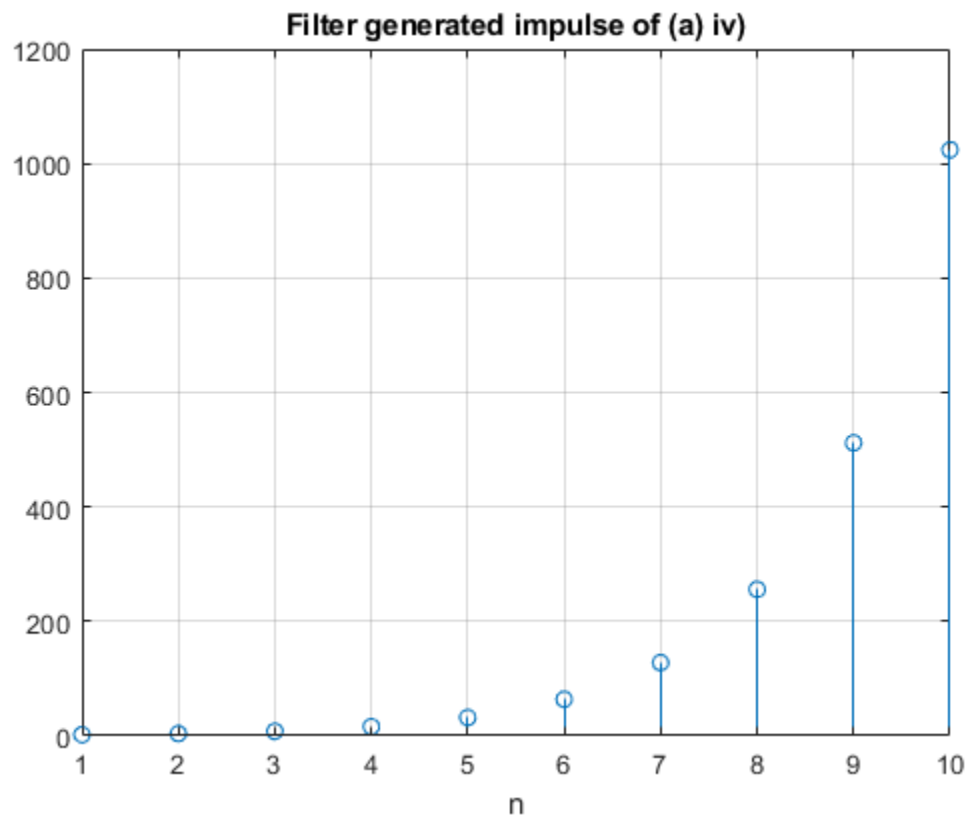
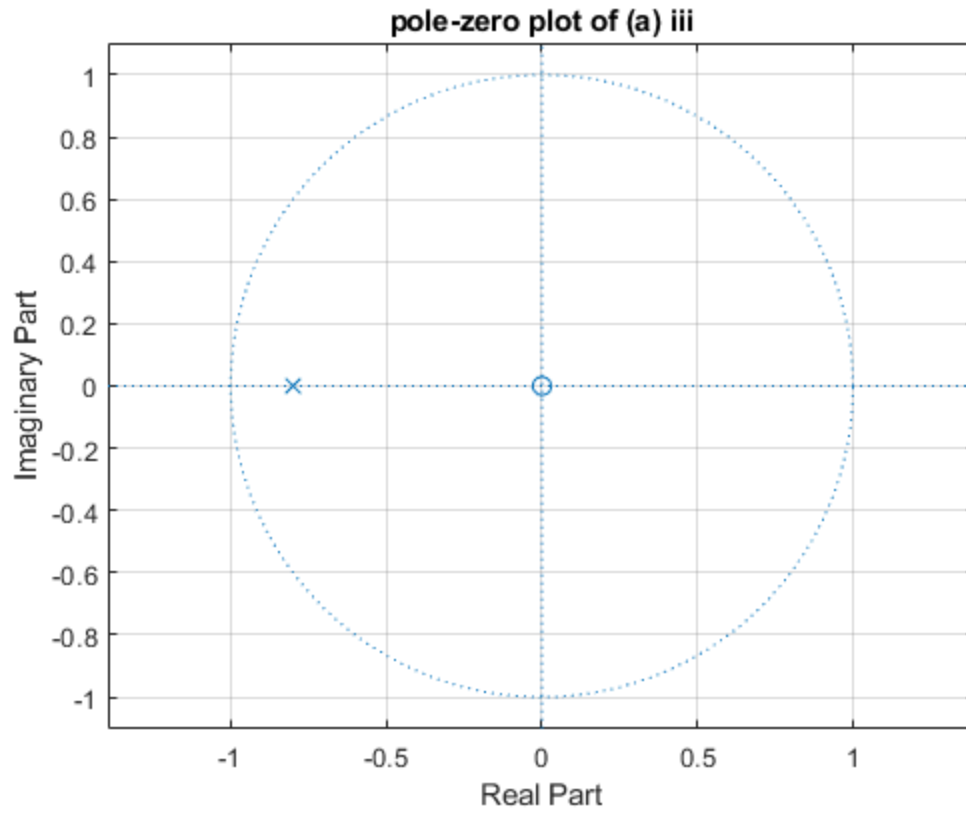
Warning: Using only the real component of complex data.

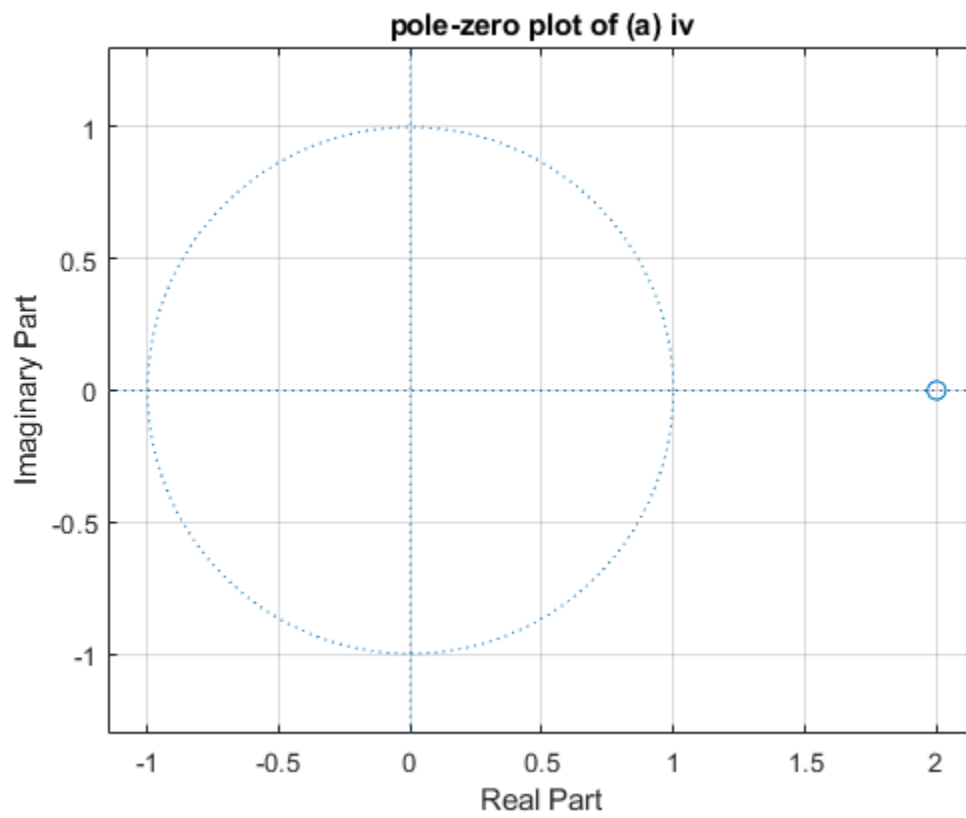
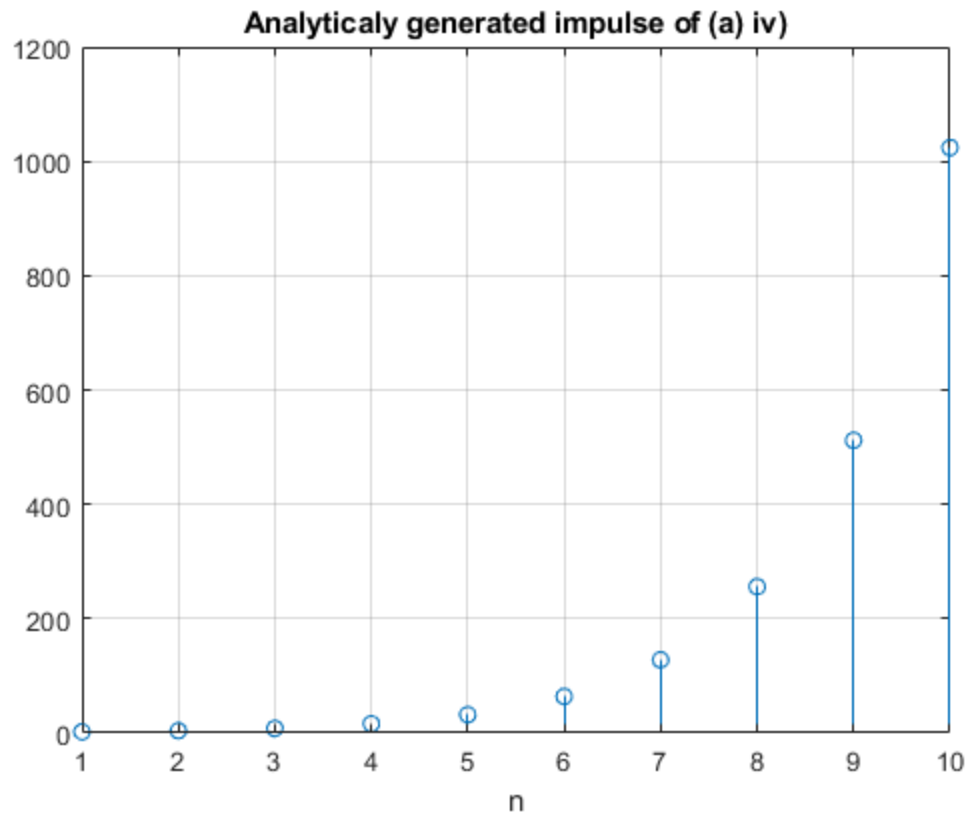


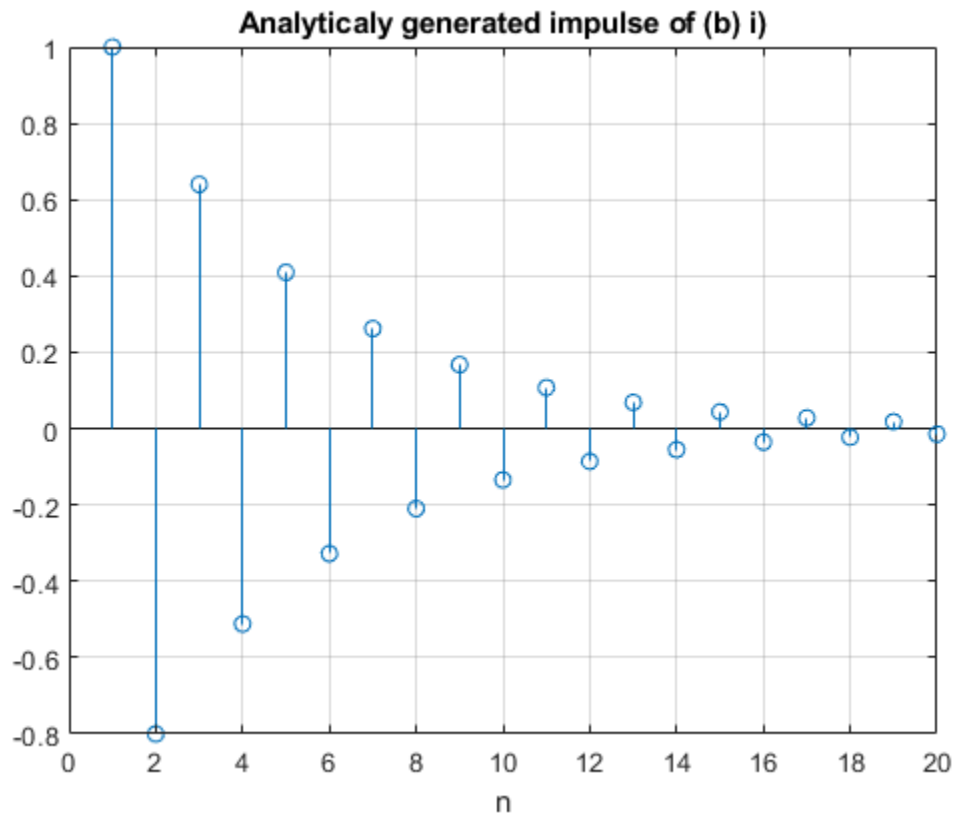
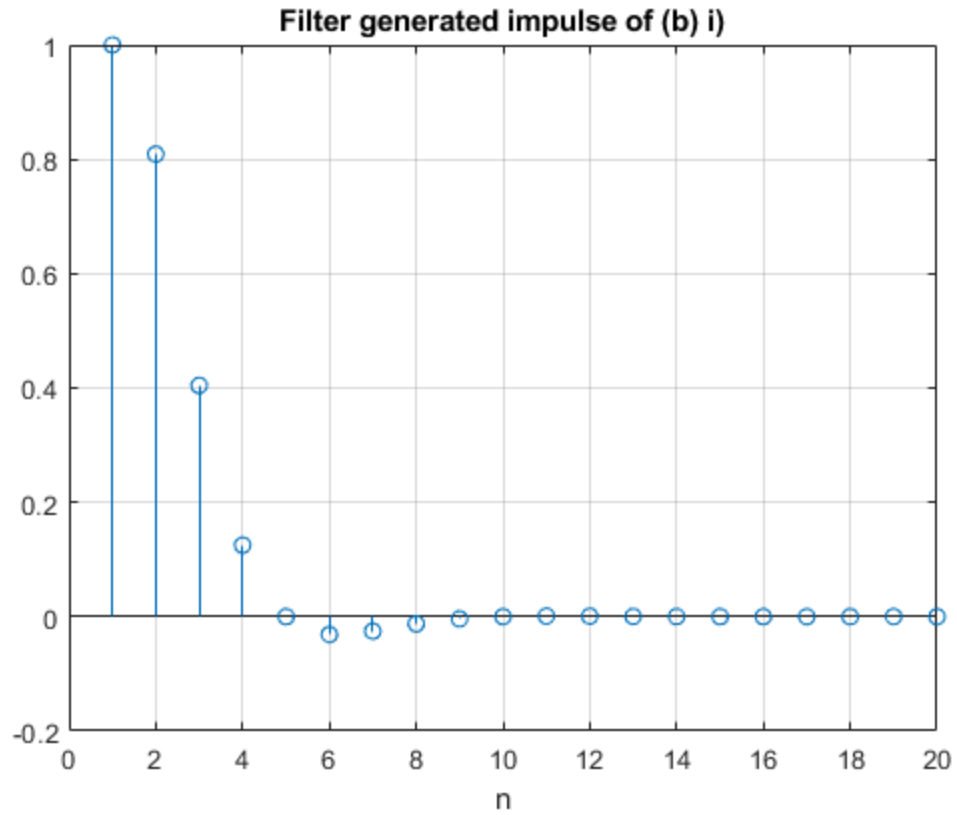


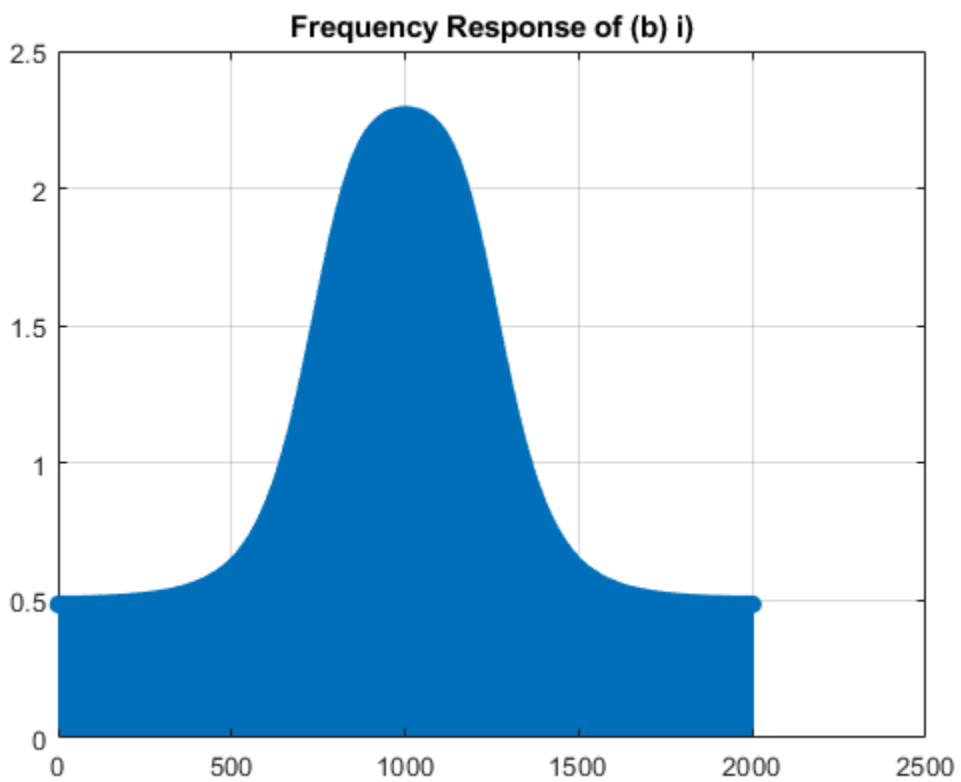
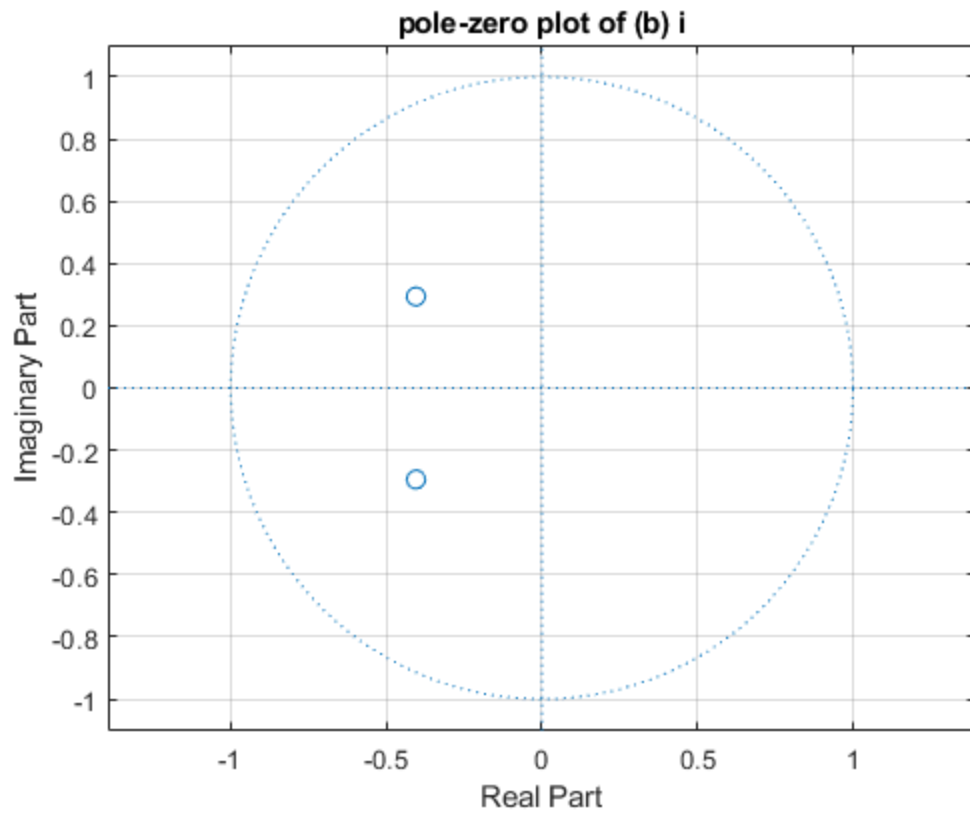


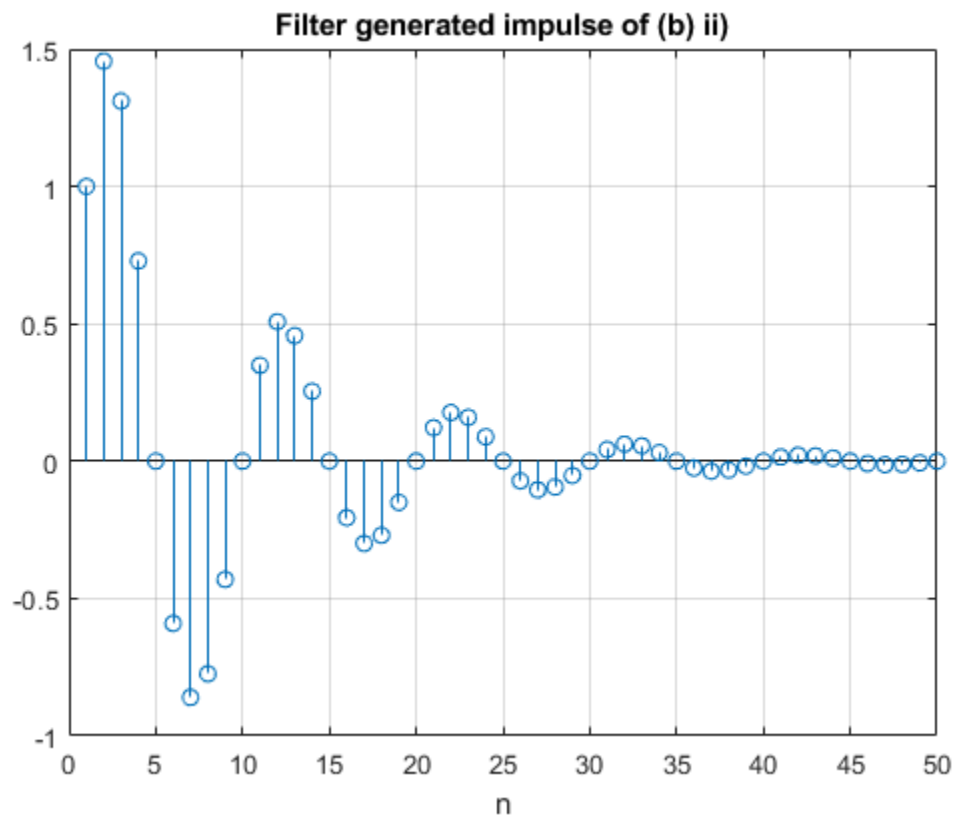
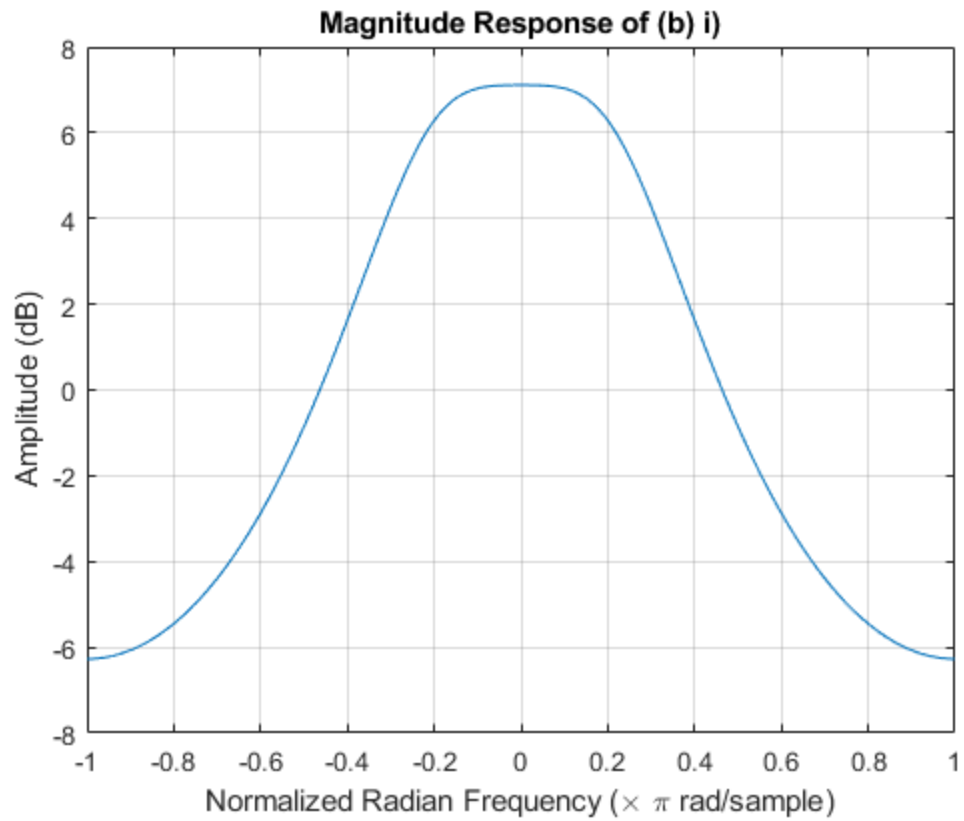


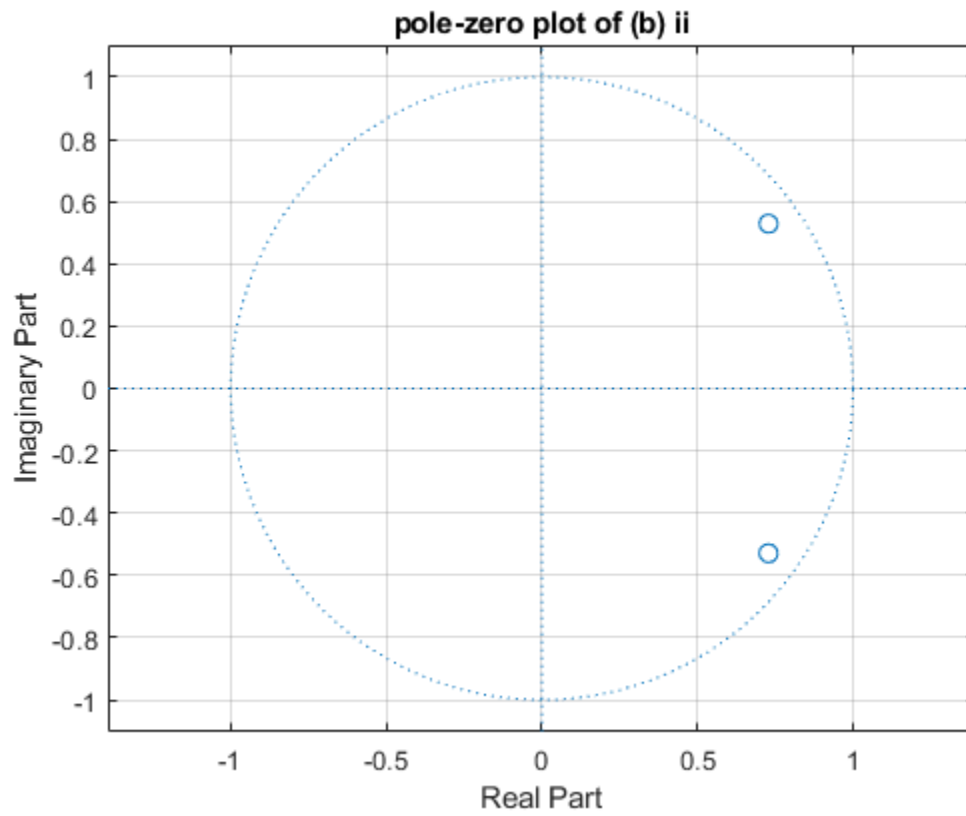
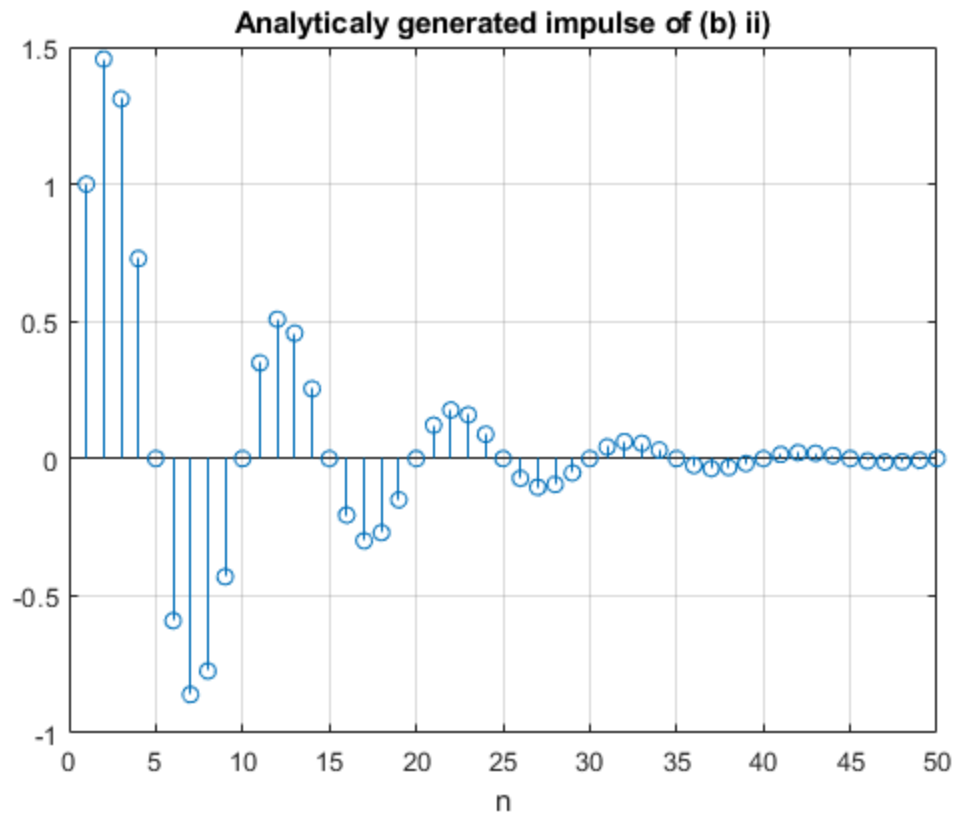


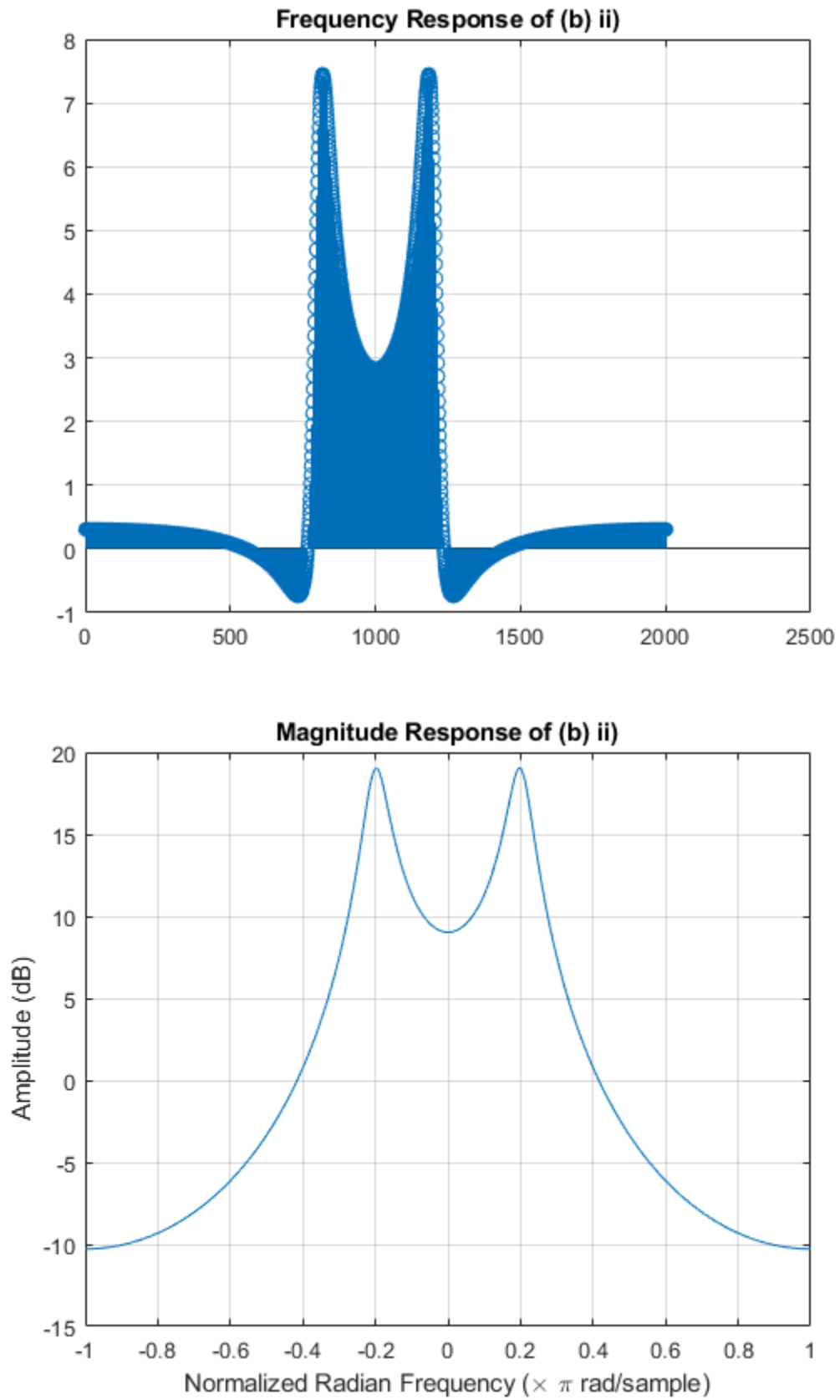


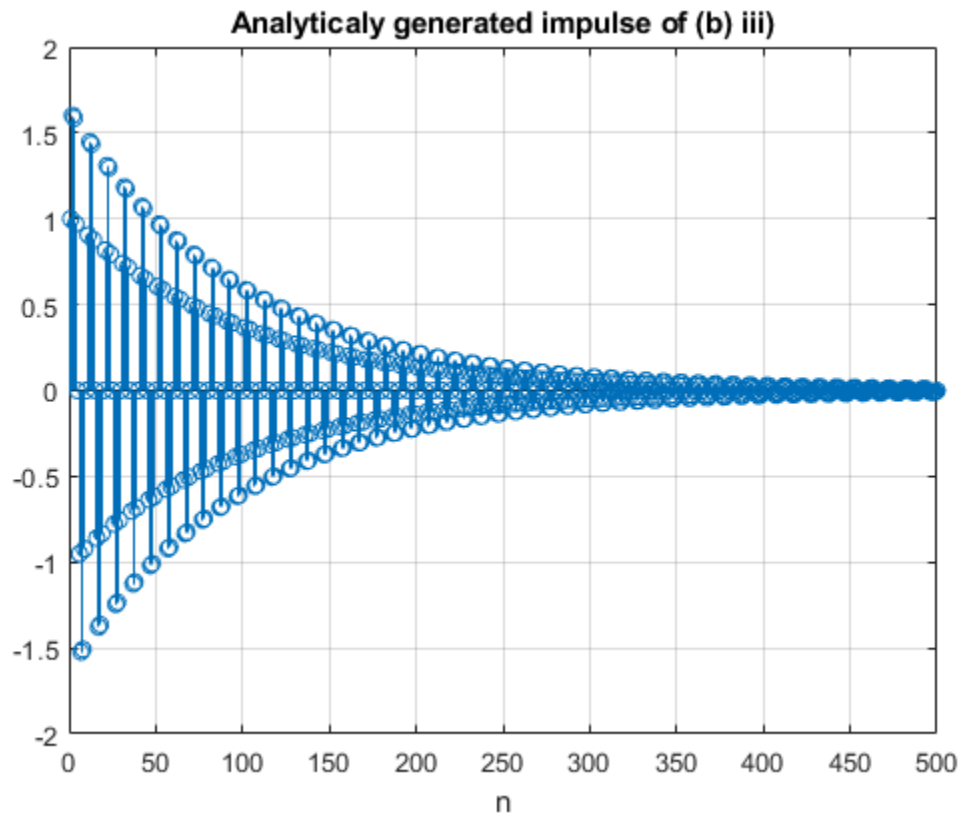
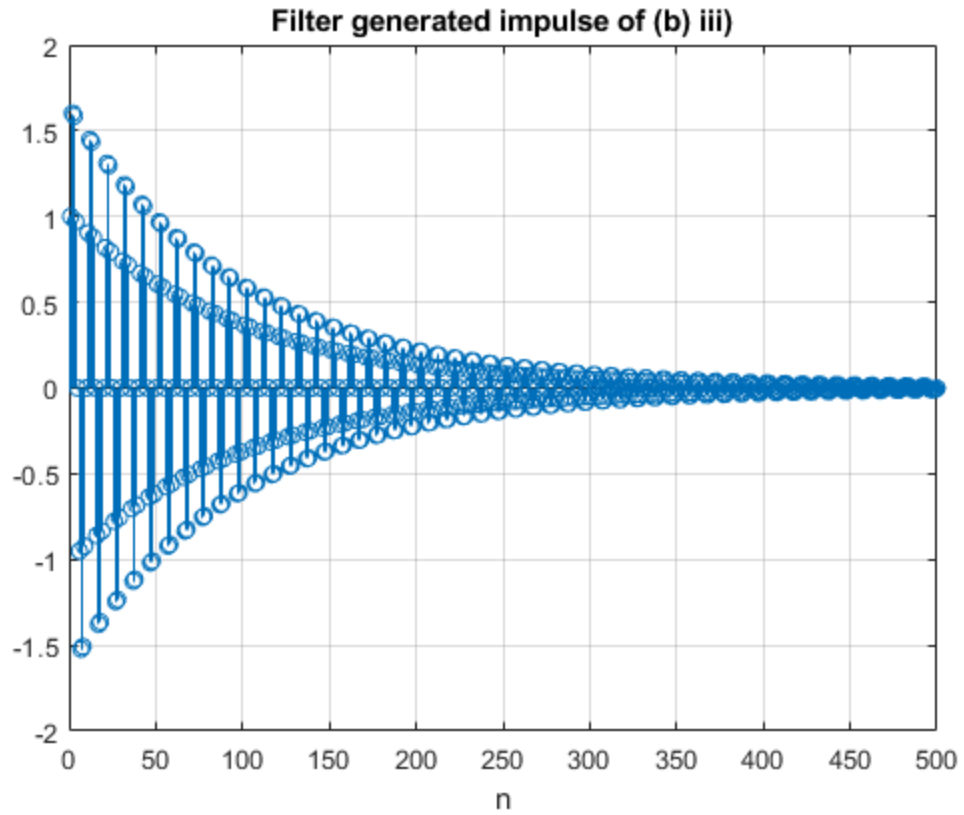


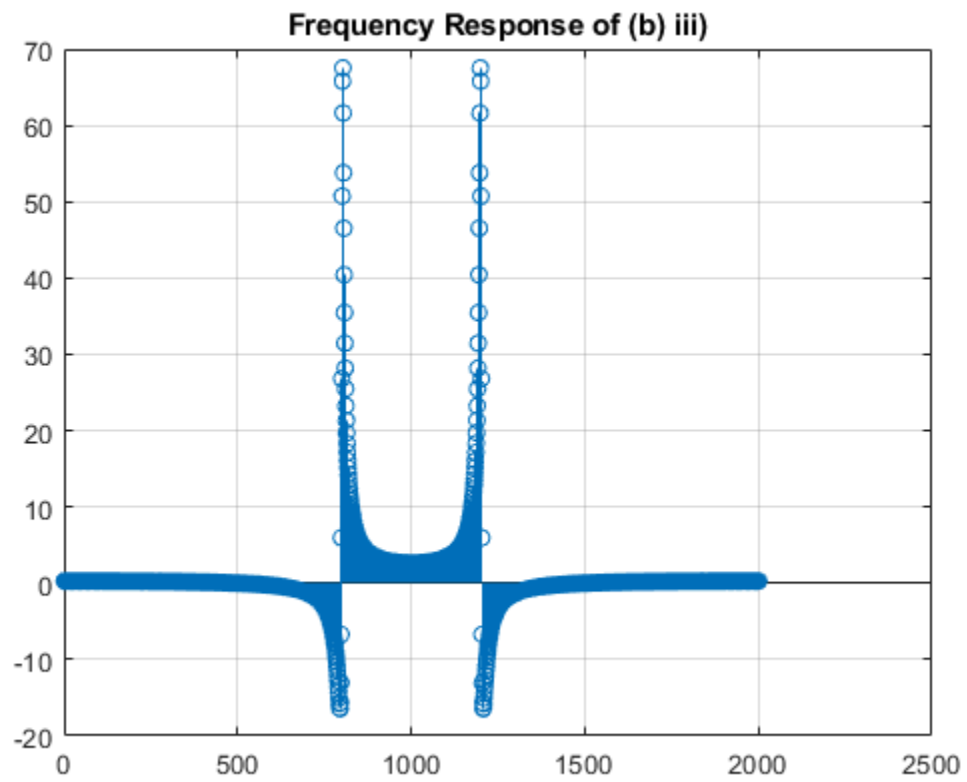
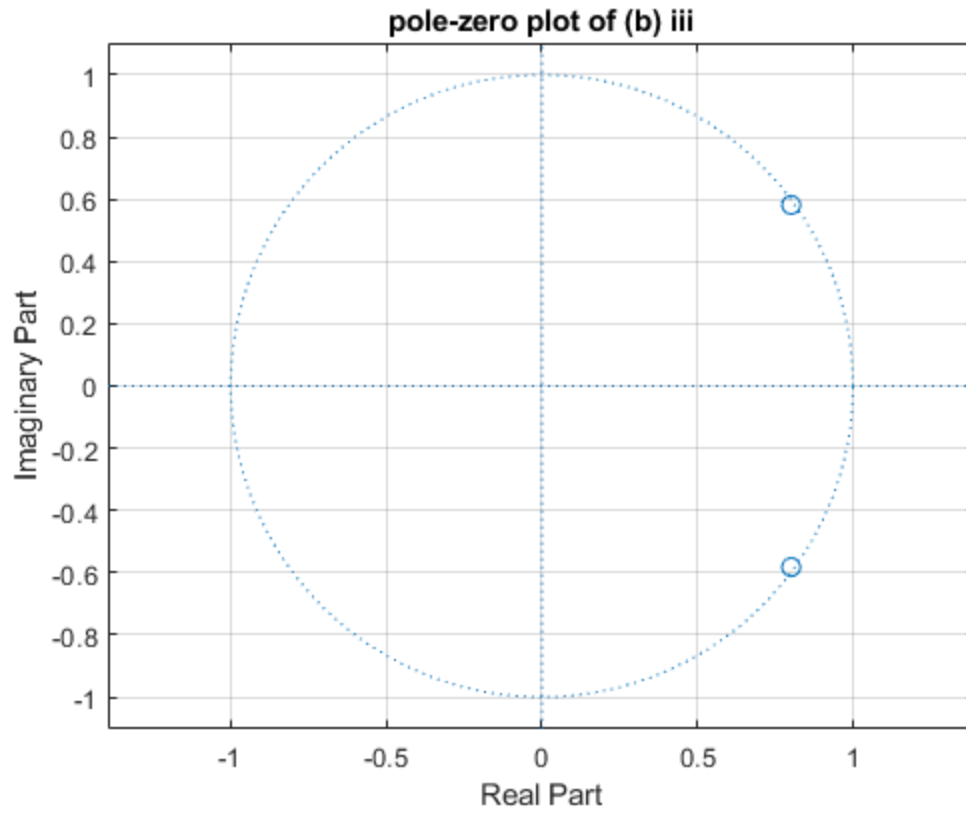


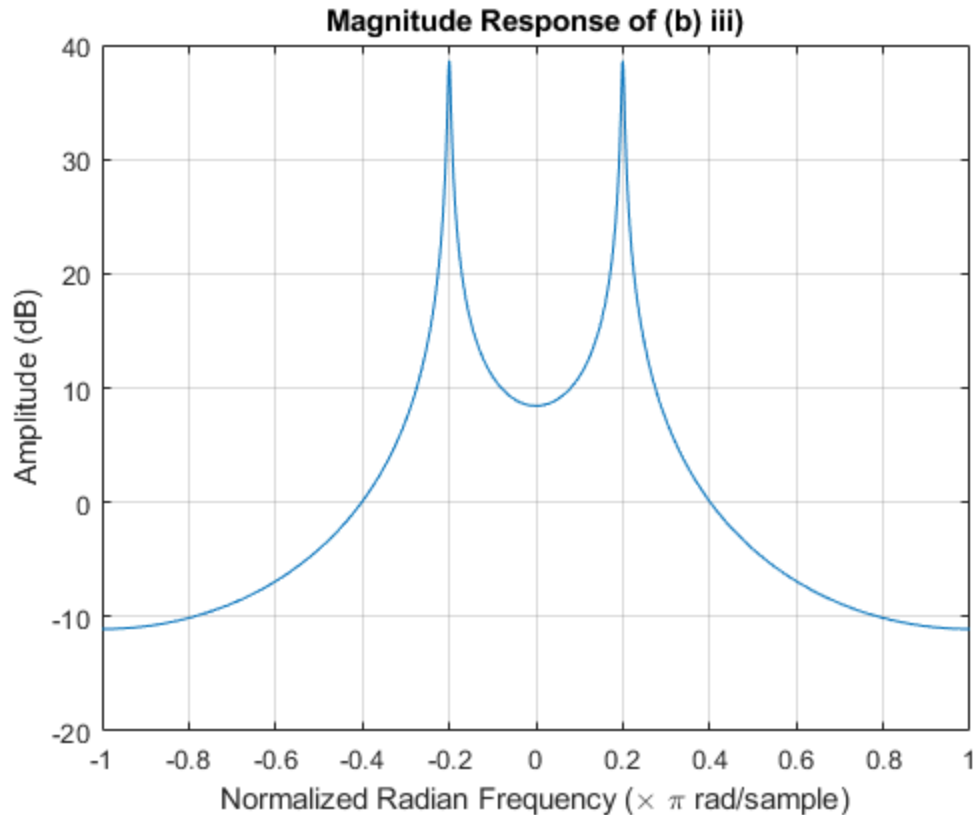












Exercise 7.2

```

%(a)
[bad_wannabe, fs] = audioread('bad_wannabe.wav');
bad_wannabe = bad_wannabe.';
w = -pi:pi/2000:pi;

wannabe_X = dtft(bad_wannabe, w);
figure.figure_counter;
figure_counter = figure_counter + 1;
plot(w/pi, 20*log10(abs(wannabe_X)));
grid on;
title('Magnitude Response for wannabe-X');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');
%interference at around 0.05

x = bad_wannabe;
x_delayed = [0 x 0]; % x[n-1]
x_delayed_two = [0 0 x]; %x[n-2]
x_longer = [x 0 0]; % x[n]

%y = x_longer - (2*cos(0.05*pi)*x_delayed) + x_delayed_two;

```

```
delta = zeros(1,length(x_longer));
delta_two = delta;
delta_three = delta;

delta(1) = 1;
delta_two(2) = 1;
delta_three(3) = 1;

delta_two = 2*cos(0.05*pi)*delta_two;

h = delta - delta_two + delta_three;

filtered_wannabe = filter(h, 1, bad_wannabe);

filtered_wannabe_X = dtft(filtered_wannabe, w);
figure.figure_counter;
figure_counter = figure_counter + 1;
plot(w/pi, 20*log10(abs(filtered_wannabe_X)));
grid on;
title('Magnitude Response for filtered-wannabe-X');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

filtered_wannabe = filtered_wannabe/max(abs(filtered_wannabe));
audiowrite('filtered_wannabe.wav', filtered_wannabe, fs);

%(b)
%Pic of alpha
ejw = exp(j*0.05);
ejw_negative = exp(j*-0.05);

figure.figure_counter;
figure_counter = figure_counter + 1;
zplane(0.5*ejw, 0.5/ejw_negative)
grid on;
title('alpha = 0.5');

figure.figure_counter;
figure_counter = figure_counter + 1;
zplane(0.9988*ejw, 0.9988/ejw_negative)
grid on;
title('alpha = 0.9988');

figure.figure_counter;
figure_counter = figure_counter + 1;
zplane(1.2*ejw, 1.2/ejw_negative)
grid on;
title('alpha = 1.2');

%find freq response to get mag response. plug e^jw in for z
z = exp(j.*w);

%alpha = 0.5;
```

```
freq_response = ((1-ejw./z).*(1-ejw_negative./z))./((1-(0.5*ejw./z)).*(1-(0.5*ejw_negative./z)));

figure.figure_counter;
figure_counter = figure_counter + 1;
plot(w/pi, 20*log10(abs(freq_response)));
grid on;
title('Magnitude Response when alpha = 0.5');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

%alpha = 0.9988;
freq_response = ((1-ejw./z).*(1-ejw_negative./z))./((1-(0.9988*ejw./z)).*(1-(0.9988*ejw_negative./z)));

figure.figure_counter;
figure_counter = figure_counter + 1;
plot(w/pi, 20*log10(abs(freq_response)));
grid on;
title('Magnitude Response when alpha = 0.9988');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

%alpha = 1.2;
freq_response = ((1-ejw./z).*(1-ejw_negative./z))./((1-(1.2*ejw./z)).*(1-(1.2*ejw_negative./z)));

figure.figure_counter;
figure_counter = figure_counter + 1;
plot(w/pi, 20*log10(abs(freq_response)));
grid on;
title('Magnitude Response when alpha = 1.2');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

%alpha = 0.1;
freq_response = ((1-ejw./z).*(1-ejw_negative./z))./((1-(0.1*ejw./z)).*(1-(0.1*ejw_negative./z)));

figure.figure_counter;
figure_counter = figure_counter + 1;
plot(w/pi, 20*log10(abs(freq_response)));
grid on;
title('Magnitude Response when alpha = 0.1');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

%alpha = 0.75;
freq_response = ((1-ejw./z).*(1-ejw_negative./z))./((1-(0.75*ejw./z)).*(1-(0.75*ejw_negative./z)));

figure.figure_counter;
figure_counter = figure_counter + 1;
plot(w/pi, 20*log10(abs(freq_response)));
```

```
grid on;
title('Magnitude Response when alpha = 0.75');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

%After looking at the Magnitude Responses that I plotted with varying
%alphas, I see that the closer alpha gets to the value e^jw, the
%"skinnier" the plot seems. I am assuming "skinnier" is what is meant
%by "sharpness". Another interesting thing I notice is that the
%closer alpha gets to 0, the wider the range in dB.

%(c)
%function made is printed at the end

%(d)
[b a] = notch(0.05, 0.99);
iired_wannabe = filter(b, a, bad_wannabe);

iired_wannabe = iired_wannabe/max(abs(iired_wannabe));
audiowrite('iired_wannabe.wav', iired_wannabe, fs);
%soundsc(iired_wannabe,fs);

iir_wannabe_X = dtft(iired_wannabe, w);
figure.figure_counter;
figure_counter = figure_counter + 1;
plot(w/pi, 20*log10(abs(iir_wannabe_X)));
grid on;
title('Magnitude Response for iir-wannabe-X');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

%So I can't figure out what I am doing wrong here, but my
%iir_wannabe sounds the same as the bad_wannabe. The Magnitude
%Response also looks similar to part (a) (but if you look at specific
%points, they have moved very slightly). My guess is that this filter
%is supposed to decrease a smaller section along the x-axis. This
%would make it so you could pinpoint the exact interference and only
%get rid of that. I must not have the right w (mine is 0.05). This is
%just my guess though. I do not think my notch function is wrong
%and after calling that function you just need to filter the original
%song. So if my w is wrong, then my output to my notch function
%will be wrong
```

My Functions

[include>dtft.m</include](#) [include>notch.m</include](#)

Pictures of handwritten work

[include>7.1.jpg</include](#) [include>7.1plots.jpg</include](#) [include>7.1bgeneral.jpg</include](#) [include>7.1b-work.jpg</include](#) [include>7.1plots.jpg</include](#) [include>7.2.jpg</include](#)

Published with MATLAB® R2020a