
Koby Miller Lab6

Table of Contents

Exercise 6.1	1
Exercise 6.2	8
Extra credit	11
My Functions	14

Exercise 6.1

```
w = -pi:pi/1000:pi;
n_ten = 0 : 9;      %length 10
n_hundred = 0 : 99; %length 100

%(a)
a_from_table = exp(-3*j*w);

temp = zeros(1,10);
temp(3) = 1;
a_ten = dtft(temp, w);

temp = zeros(1,100);
temp(3) = 1;
a_hundred = dtft(temp, w);

figure(1);
subplot(2,3,1);
plot(w/pi, 20*log10(abs(a_from_table)));
grid on;
title('Magnitude Response for a-table');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

subplot(2,3,4)
plot(w/pi, angle(a_from_table)/pi);
grid on;
title('Phase Response for a-table');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Phase (\times \pi rad)');

subplot(2,3,2);
plot(w/pi, 20*log10(abs(a_ten)));
grid on;
title('Magnitude Response for a-ten');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');
```

```
subplot(2,3,5)
plot(w/pi, angle(a_ten)/pi);
grid on;
title('Phase Response for a-ten');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Phase (\times \pi rad)');

subplot(2,3,3);
plot(w/pi, 20*log10(abs(a_hundred)));
grid on;
title('Magnitude Response for a-hundred');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

subplot(2,3,6)
plot(w/pi, angle(a_hundred)/pi);
grid on;
title('Phase Response for a-hundred');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Phase (\times \pi rad)');

%I really don't know why I cannot get the Magnitude Responses
%to look right.

%(b)
b_from_table = 1./(1-(8/9)*exp(-1*j*w));

b_ten = dtft((8/9).^n_ten, w);

b_hundred = dtft((8/9).^n_hundred, w);

figure(2);
subplot(2,3,1);
plot(w/pi, 20*log10(abs(b_from_table)));
grid on;
title('Magnitude Response for b-table');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

subplot(2,3,4)
plot(w/pi, angle(b_from_table)/pi);
grid on;
title('Phase Response for b-table');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Phase (\times \pi rad)');

subplot(2,3,2);
plot(w/pi, 20*log10(abs(b_ten)));
grid on;
title('Magnitude Response for b-ten');
```

```
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

subplot(2,3,5)
plot(w/pi, angle(b_ten)/pi);
grid on;
title('Phase Response for b-ten');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Phase (\times \pi rad)');

subplot(2,3,3);
plot(w/pi, 20*log10(abs(b_hundred)));
grid on;
title('Magnitude Response for b-hundred');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

subplot(2,3,6)
plot(w/pi, angle(b_hundred)/pi);
grid on;
title('Phase Response for b-hundred');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Phase (\times \pi rad)');

%For (b) you can definitielly see the similarities. Also, as expected,
%when using a larger length, you get a result that is closer to
%the expression obtained analytically. The values are a little off,
%but this is only a small length when you compare it to infinity.

%(c)
c_from_table = 1./(1+(8/9)*exp(-1*j*w));

c_ten = dtft((-8/9).^n_ten, w);

c_hundred = dtft((-8/9).^n_hundred, w);

figure(3);
subplot(2,3,1);
plot(w/pi, 20*log10(abs(c_from_table)));
grid on;
title('Magnitude Response for c-table');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

subplot(2,3,4)
plot(w/pi, angle(c_from_table)/pi);
grid on;
title('Phase Response for c-table');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Phase (\times \pi rad)');

subplot(2,3,2);
```

```
plot(w/pi, 20*log10(abs(c_ten)));
grid on;
title('Magnitude Response for c-ten');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

subplot(2,3,5)
plot(w/pi, angle(c_ten)/pi);
grid on;
title('Phase Response for c-ten');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Phase (\times \pi rad)');

subplot(2,3,3);
plot(w/pi, 20*log10(abs(c_hundred)));
grid on;
title('Magnitude Response for c-hundred');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

subplot(2,3,6)
plot(w/pi, angle(c_hundred)/pi);
grid on;
title('Phase Response for c-hundred');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Phase (\times \pi rad)');

%So the Magnitude Responses look similar, but the phase response
%seems backwards. I am not quite sure why this would be. However,
%at the ends of the graph, all of the phase responses go in the same
%direction. increasing on the left, decreasing on the right. Maybe
%with a larger n, this would end up being closer.

%(d)
d_from_table = (1-exp(5*j*w))./(1-exp(j*w));

u_ten = ones(1,10);
u_ten_shifted = u_ten;
u_ten_shifted(1:5) = 0;

u_hundred = ones(1,100);
u_hundred_shifted = u_hundred;
u_hundred_shifted(1:5) = 0;

d_ten = dtft((u_ten - u_ten_shifted), w);

d_hundred = dtft((u_hundred - u_hundred_shifted), w);

figure(4);
subplot(2,3,1);
plot(w/pi, 20*log10(abs(d_from_table)));
grid on;
title('Magnitude Response for d-table');
```

```
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

subplot(2,3,4)
plot(w/pi, angle(d_from_table)/pi);
grid on;
title('Phase Response for d-table');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Phase (\times \pi rad)');

subplot(2,3,2);
plot(w/pi, 20*log10(abs(d_ten)));
grid on;
title('Magnitude Response for d-ten');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

subplot(2,3,5)
plot(w/pi, angle(d_ten)/pi);
grid on;
title('Phase Response for d-ten');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Phase (\times \pi rad)');

subplot(2,3,3);
plot(w/pi, 20*log10(abs(d_hundred)));
grid on;
title('Magnitude Response for d-hundred');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

subplot(2,3,6)
plot(w/pi, angle(d_hundred)/pi);
grid on;
title('Phase Response for d-hundred');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Phase (\times \pi rad)');

%for (d) the Magnitude Responses all seem very similar. Both of the
%phase responses when using the dtft function look similar, but are
%different than the one obtained analytically

%(e)
% ##(###0)+##(#+#0)    when publishing this mainly changes to #s
% pi*delta(w - w0) + pi*delta(w + w0)
% I don't know how to do this in matlab....
% e_from_table_hundred = 0.5*(exp(j*(pi/4).*n)+exp(-1*j*(pi/4).*n));
e_ten = dtft((cos(pi/4).*n_ten), w);

e_hundred = dtft((cos(pi/4).*n_hundred), w);

figure(5);
```

```

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

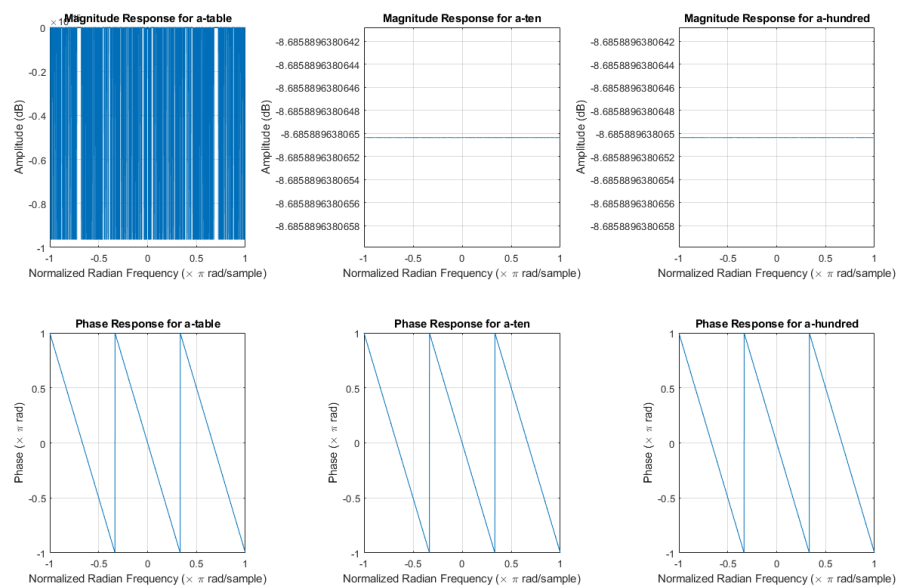
subplot(2,2,3)
plot(w/pi, angle(e_ten)/pi);
grid on;
title('Phase Response for e-ten');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Phase (\times \pi rad)');

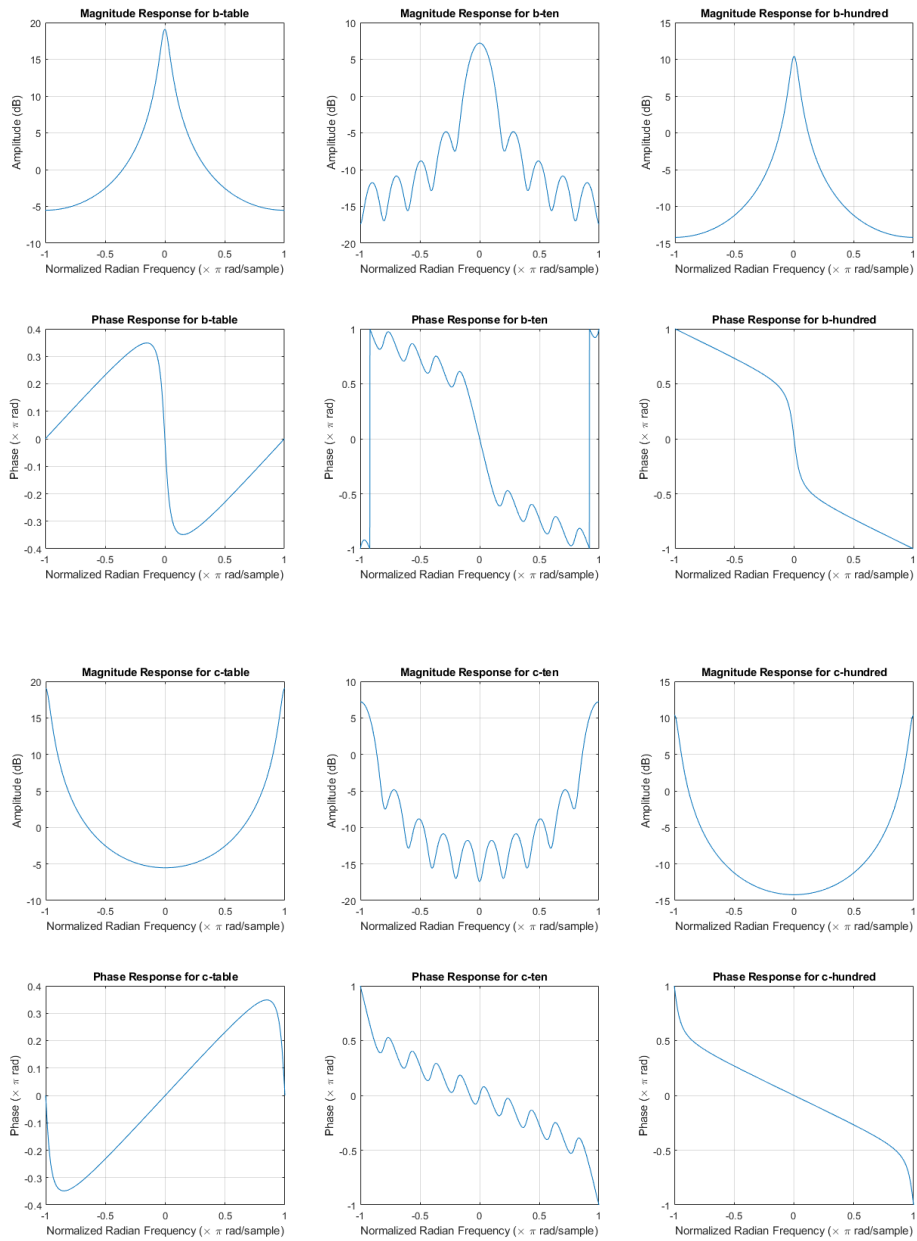
subplot(2,2,2);
plot(w/pi, 20*log10(abs(e_hundred)));
grid on;
title('Magnitude Response for e-hundred');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

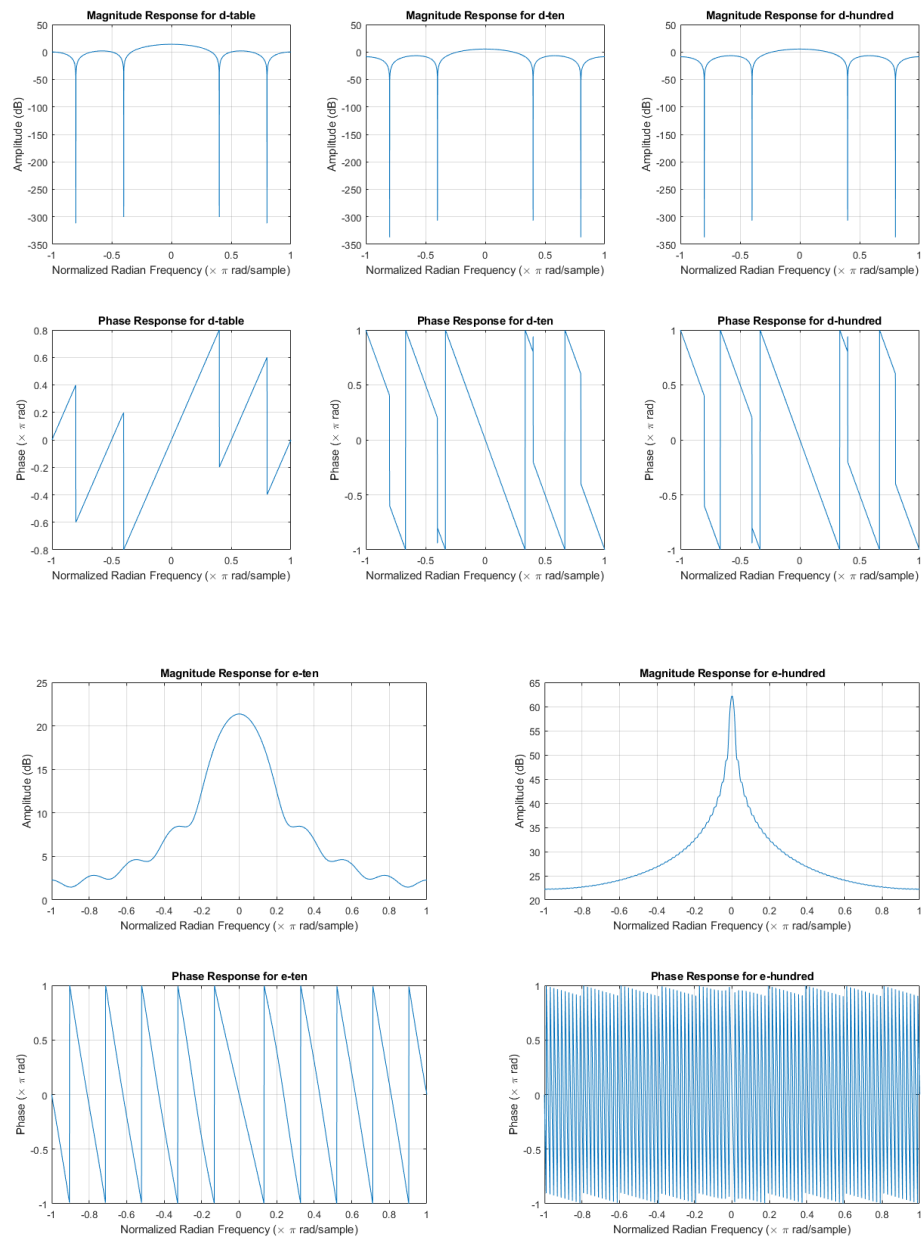
subplot(2,2,4)
plot(w/pi, angle(e_hundred)/pi);
grid on;
title('Phase Response for e-hundred');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Phase (\times \pi rad)');

```

%So I didn't know exactly how to input the expression obtained
 %analytically for (e). However, you can see the relation between
 %the two different lengths of n.







Exercise 6.2

```
%(a)
[wannabe_orig, fs] = audioread('corrupted_wannabe.wav');
wannabe_orig = wannabe_orig.';
w = -pi:pi/2000:pi;
wannabe_X = dtft(wannabe_orig, w);

figure(6);
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)');

%(b)
%0.4535 is the Normalized Radian Frequency of the interference
% w = Omega/F
% 0.4535 = Omega/2000
% Omega = 907 rad/sec
%To convert normalized frequency to angular frequency around the unit
%circle, multiply by pi. To convert normalized frequency back to
%hertz, multiply by half the sample frequency.

%0.4545*pi = 1.4247

%(c)
%y[n] = x[n] - 2cos(0.4535*pi)x[n - 1] + x[n - 2]
%Impulse response:
%h[n] = delta[n] - 2cos(0.4535*pi)delta[n - 1] + delta[n - 2]

x = wannabe_orig;
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(0.4535*pi)*x_delayed) + x_delayed_two;
%Don't need what is above, made for testing stuff though

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(0.4535*pi)*delta_two;

h = delta - delta_two + delta_three;
h_dtft = dtft(h, w);

figure(7)
subplot(2,1,1)
plot(w/pi, angle(h_dtft)/pi);
grid on;
title('Phase Response for h');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Phase (\times \pi rad)');

subplot(2,1,2);
plot(w/pi, 20*log10(abs(h_dtft)));
grid on;
```

```

title('Magnitude Response for h');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

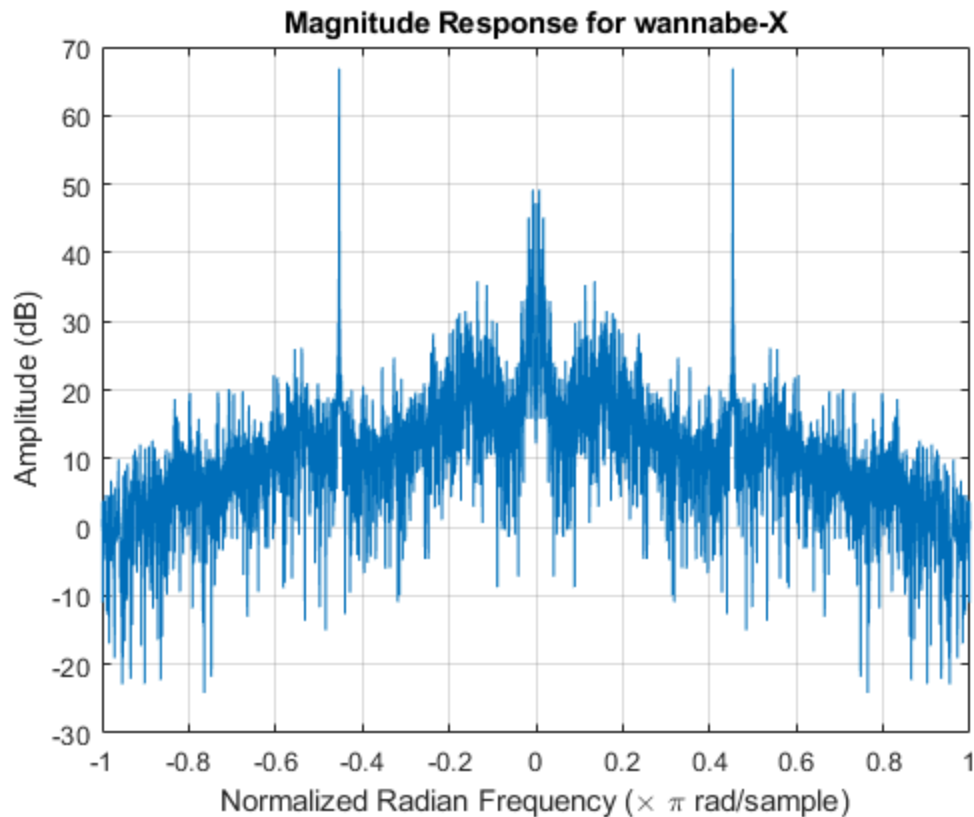
%(d)
filtered_wannabe = filter(h, 1, wannabe_orig);
%soundsc(filtered_wannabe, fs)
audiowrite('filtered_wannabe.wav', filtered_wannabe, fs);

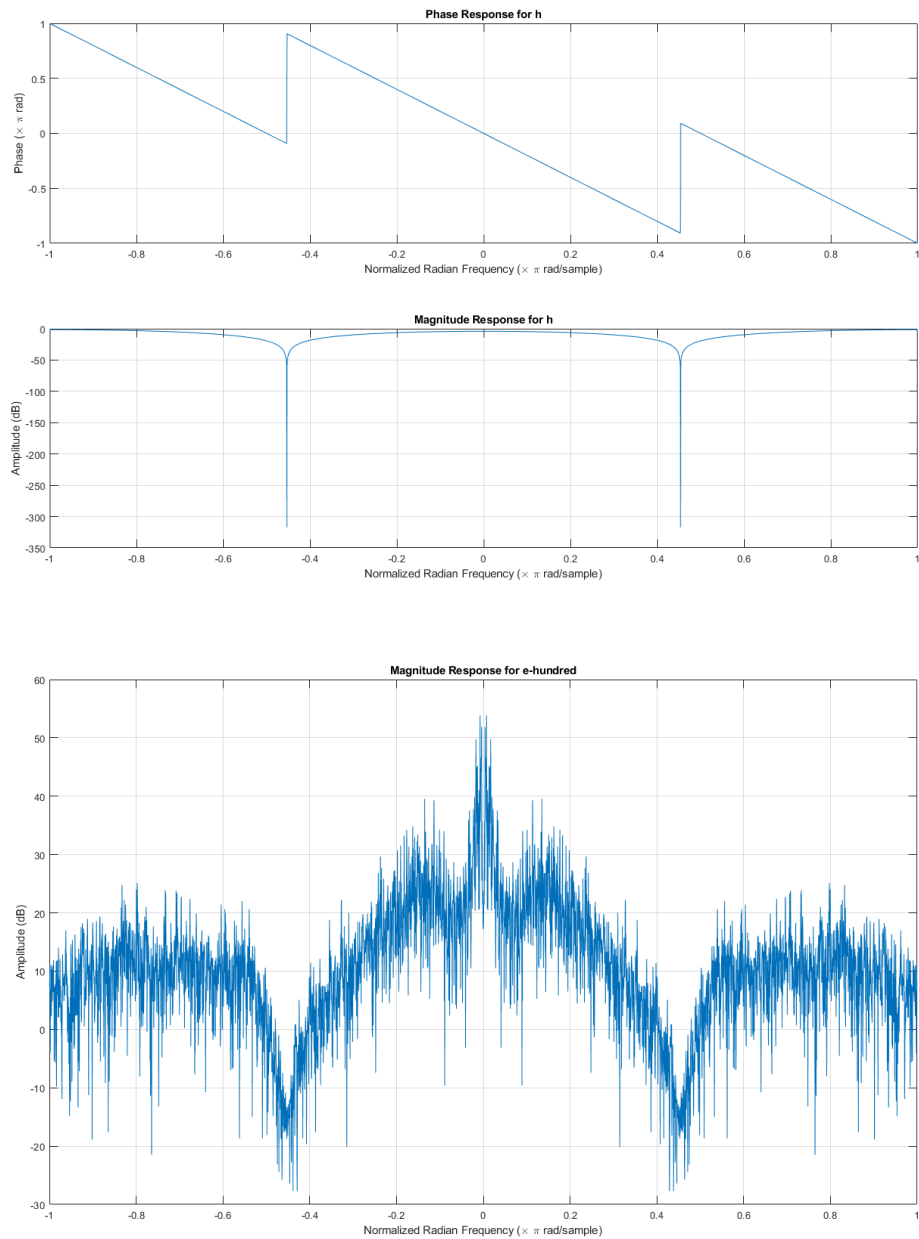
filtered_wannabe_X = dtft(filtered_wannabe, w);

figure(8)
plot(w/pi, 20*log10(abs(filtered_wannabe_X)));
grid on;
title('Magnitude Response for e-hundred');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

%So when I compare figure 6 and figure 8, you can clearly see that
%the spikes in figure 6 are now gone. In fact, there is now sort of a
%v shape at those points. If you listen to the audio after it is
%filtered, you won't hear that high pitch noise anymore. So the
%interference signal is now gone

```





Extra credit

```
[what_orig, fs] = audioread('what_did_he_say.wav');
what_orig = what_orig.';

what_X = dtft(what_orig, w); %w is still 2000

figure(9);
plot(w/pi, 20*log10(abs(what_X)));
grid on;
title('Magnitude Response for what-X');
```

```
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

% so from 0 to about 0.25*pi it is around -20dB
%   from 0.25*pi to pi, it is mainly between 10 to 30dB
% you need a lowpass filter

filtered_what = filter(my_filter, 1, what_orig);
soundsc(filtered_what, fs)

filtered_what_X = dtft(filtered_what, w);

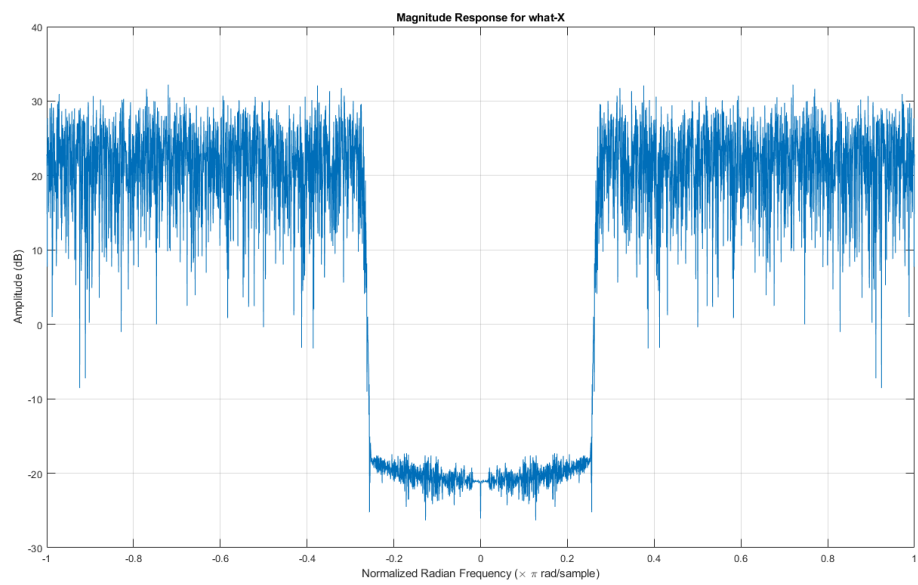
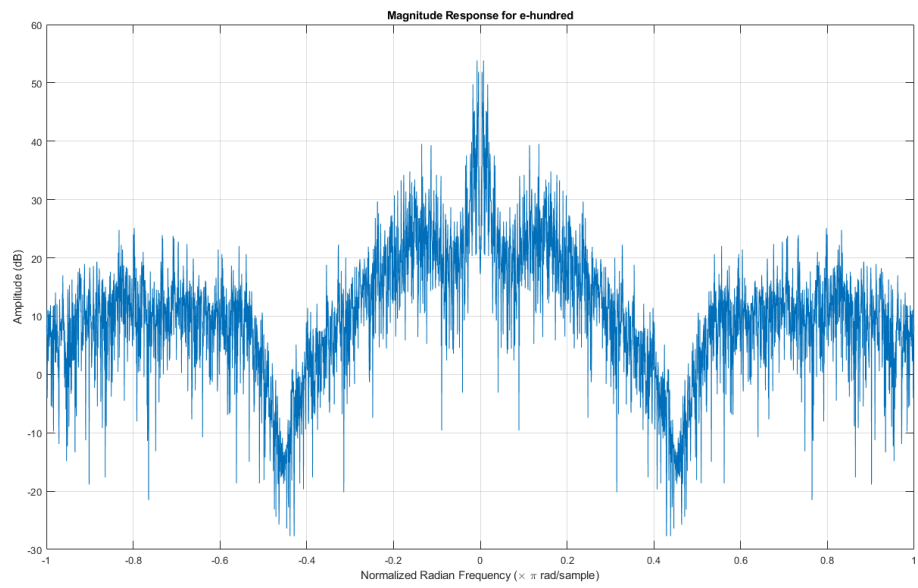
figure(10);
plot(w/pi, 20*log10(abs(filtered_what_X)));
grid on;
title('Magnitude Response for filtered-what-X');
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude (dB)');

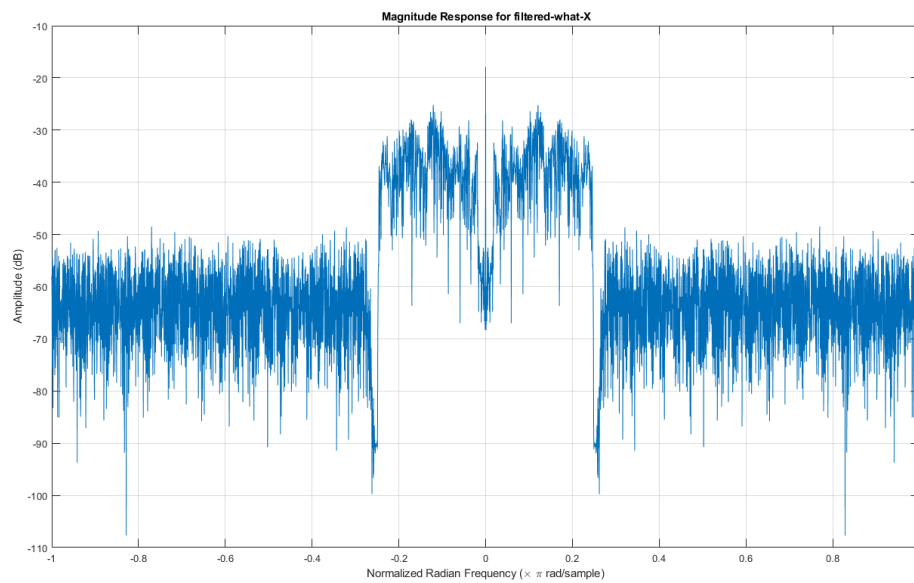
filtered_what_scaled = filtered_what/max(abs(filtered_what));
audiowrite('filtered_what.wav', filtered_what_scaled, fs);

%Filter parameters
%Response Type = Lowpass
% Frequency Specifications:
%   Units = Hz, Fs = 8000, Fpass = 980, Fstop = 1000
% Magnitude Specifications:
%   Units = dB, Apass = 1, Astop = 80
%The rest was given to us

% I believe the sentence is:
% Oak is strong and also gives shade.

% The overall shape between the original DTFT and the filtered are
% pretty similar, however, when you look at their Amplitudes, they
% are quite different. The filtered sort of squished the original and
% try to get the same range of amplitude at every frequency (around
% -60dB) It moved these high amplitudes down 80dB (which is what my
% Astop was) I could have made all of the amplitudes in the same
% region, but then there was a little more noise.
```





My Functions

```
function X = dtft(x, w)
    X = 0;

    for a = 1 : length(x)
        X = X + (x(1,a) * exp(-1j*w*a-1));
    end

end
```

Published with MATLAB® R2020a