
Koby Miller Lab9

Table of Contents

Exercise 9.1	1
Exercise 9.2	6
My Functions	9

Exercise 9.1

```
w1 = -pi:pi/1000:pi;

%(a)
n = 0:1:99;
u = ones(1, 100);

% [u[n] - u[n-100]] = always 1 until n >= 100 then is 0
% our n only goes to 99

x = (0.5 + cos((pi/30).*n) + cos((pi/5).*n) + cos(pi.*n + 2*pi/3));

x_DTFT = dtft(x,w1);

figure(1);
subplot(2,1,1)
plot(w1/pi, abs(x_DTFT), 'b-');
title('Magnitude')
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude');
subplot(2,1,2)
plot(w1/pi, angle(x_DTFT)/pi, 'b-');
title('Phase')
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Phase(\times \pi rad)');

%(b)
M = 128;
w2 = -pi:2*pi/M:pi-2*pi/M; % freq vector
x_DFT = fft(x,M);

figure(2);
subplot(2,1,1)
plot(w2/pi, fftshift(abs(x_DFT)), 'rx');
title('Magnitude')
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude');
subplot(2,1,2)
plot(w2/pi, fftshift(angle(x_DFT)/pi), 'rx');
title('Phase')
```

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

%putting the graphs together now
figure(3)
subplot(2,1,1)
plot(w2/pi, fftshift(abs(x_DFT)), 'rx');
hold on;
plot(w1/pi, abs(x_DTFT), 'b-');
hold off;
grid on;
title('Magnitude part (b)')
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude');
legend('DFT', 'DTFT');
subplot(2,1,2)
plot(w2/pi, fftshift(angle(x_DFT)/pi), 'rx');
hold on;
plot(w1/pi, angle(x_DTFT)/pi, 'b-');
hold off;
grid on;
title('Phase part (b)')
xlabel('Normalized Radian Frequency (\times \pi rad/sample)') ;
ylabel('Phase(\times\pi rad)');
legend('DFT', 'DTFT');

% When comparing the DFT coefficients with the DTFT values for the
% Magnitude plot, you can see that the DFT coefficients amplitude is
% always larger than the DTFT amplitude by some multiple(around 2.6).
% The DFT coefficients are frequency samples of the DTFT.
% The sample frequency of the kth DFT coefficient would be:
% X_k = X(e^{j*(2\pi k/128)})
%
% When comparing the Phase plots, I can't exactly tell what is going
% on right now. There is definitely a pattern, but I can't tell how
% the coefficients relate to the DTFT just on looks of the plot alone.

%(c)
M = 512;
w2 = -pi:2*pi/M:pi-2*pi/M; % freq vector
x_DFT = fft(x,M);

figure(4)
subplot(2,1,1)
plot(w2/pi, fftshift(abs(x_DFT)), 'rx');
hold on;
plot(w1/pi, abs(x_DTFT), 'b-');
hold off;
grid on;
title('Magnitude part (c)')
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude');
legend('DFT', 'DTFT');
subplot(2,1,2)
```

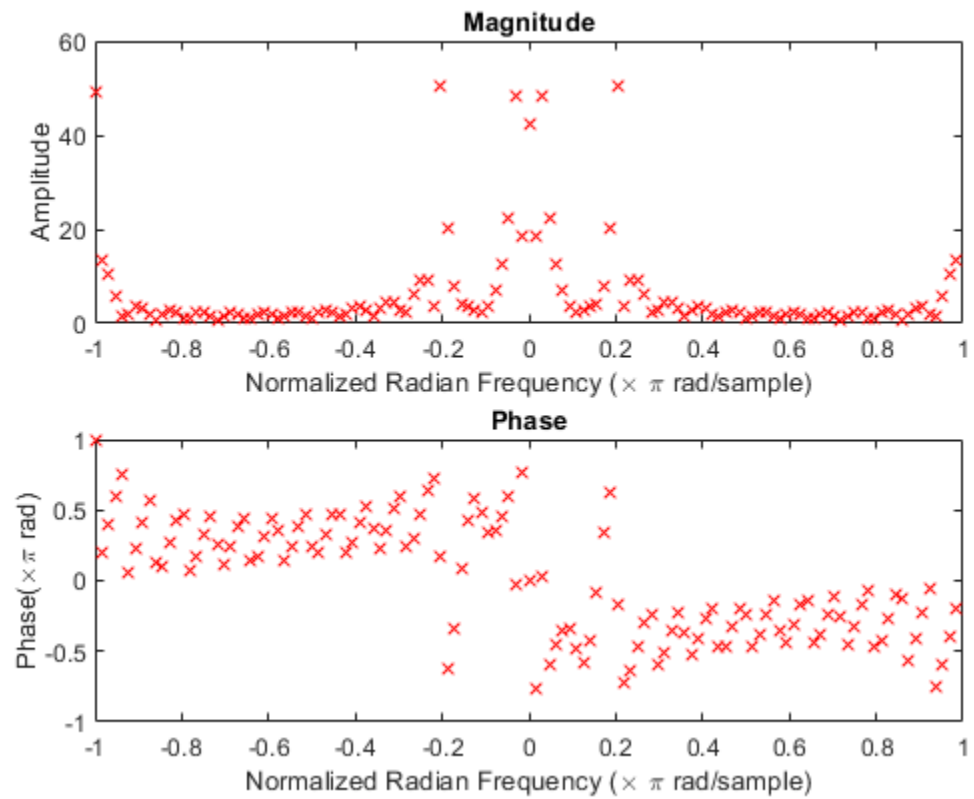
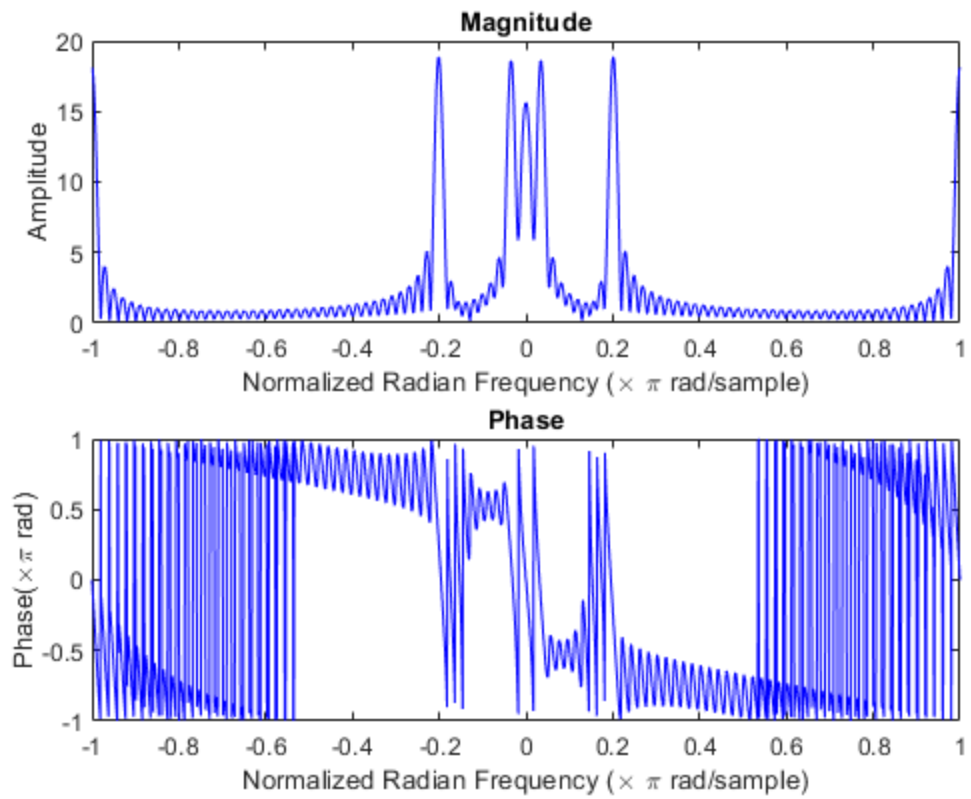
```
plot(w2/pi, fftshift(angle(x_DFT)/pi), 'rx');
hold on;
plot(w1/pi, angle(x_DTFT)/pi, 'b-');
hold off;
grid on;
title('Phase part (c)')
xlabel('Normalized Radian Frequency (\times \pi rad/sample)') ;
ylabel('Phase(\times \pi rad)');
legend('DFT', 'DTFT');

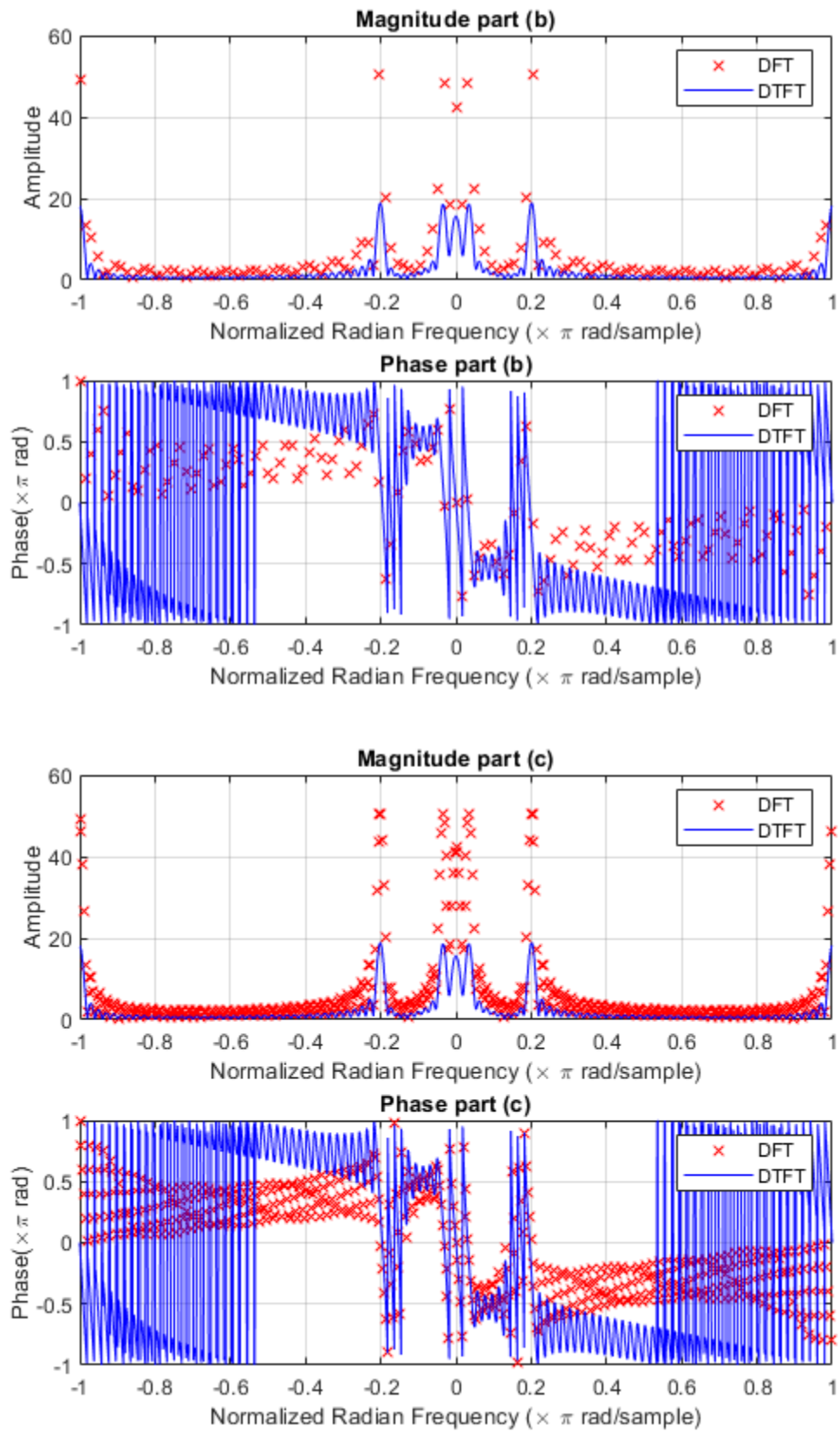
% Comparing these plots to the 128-point DFT from (b), you can
% clearly see that this is way 'clearer'. There are more samples
% so the result is a finer plot that easily shows how the DFT
% coefficients follow the DTFT values. If one uses an even larger
% size DFT the result would be even clearer. The one used in the
% example is way larger and there are so many points that it looks
% like a line.

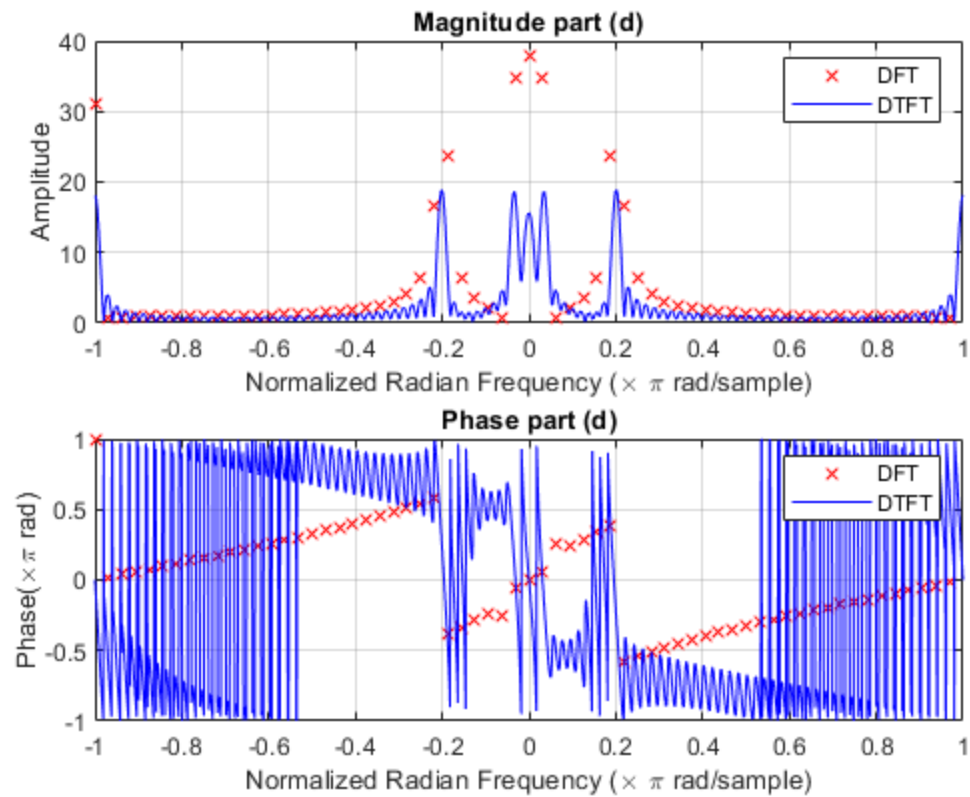
%(d)
M = 64;
w2 = -pi:2*pi/M:pi-2*pi/M; % freq vector
x_DFT = fft(x,M);

figure(5)
subplot(2,1,1)
plot(w2/pi, fftshift(abs(x_DFT)), 'rx');
hold on;
plot(w1/pi, abs(x_DTFT), 'b-');
hold off;
grid on;
title('Magnitude part (d)')
xlabel('Normalized Radian Frequency (\times \pi rad/sample)');
ylabel('Amplitude');
legend('DFT', 'DTFT');
subplot(2,1,2)
plot(w2/pi, fftshift(angle(x_DFT)/pi), 'rx');
hold on;
plot(w1/pi, angle(x_DTFT)/pi, 'b-');
hold off;
grid on;
title('Phase part (d)')
xlabel('Normalized Radian Frequency (\times \pi rad/sample)') ;
ylabel('Phase(\times \pi rad)');
legend('DFT', 'DTFT');

% As expected, there are less points for the 64-point DFT. This means
% that the results are not as fine as the 128-point DFT. You can
% still see the overall patterns which can be helpful in its own ways.
% For example, the phase plot has a way clearer increasing slope
% pattern that wasn't seen before because of all the clutter.
```







Exercise 9.2

```
%(a)
```

```
yconv = conv(x,h);
figure(6);
stem(yconv);
```

```
% Since we are filtering using a highpass filter, many of the lower
% many of the lower frequencies are filtered out of our resulting
% signal.
```

```
%(b)
```

```
%(1)
M = 256;
w2 = -pi:2*pi/M:pi-2*pi/M; % freq vector
x_DFT = fft(x,M);
h_DFT = fft(h,M);
```

```
%(2)
Y256 = x_DFT.*h_DFT;
```

```
%(3)
```

```
y256 = ifft(Y256);

figure(7);
stem(y256);

% From what I can see, the frequency-domain analysis did give the
% same output vector as the time-domain analysis except that the
% frequency-domain analysis has more data towards the right side.
% There is extra information appended to the end of the first
% plot in order to reach 256 points. Most of the values are very very
% close to 0 though.

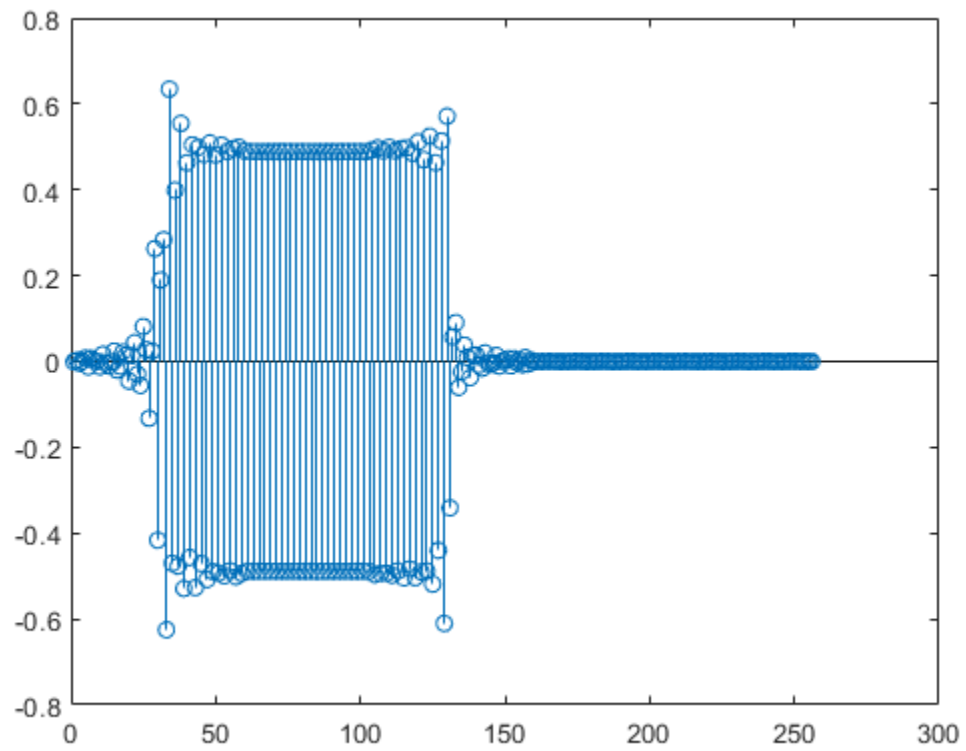
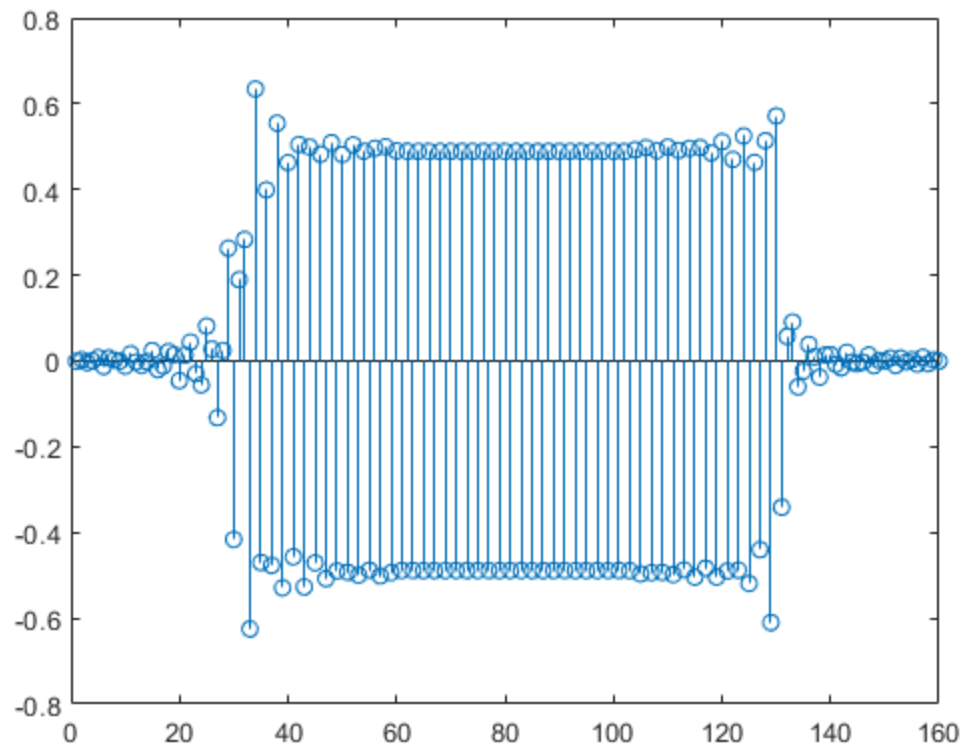
%(c)
M = 128;
w2 = -pi:2*pi/M:pi-2*pi/M; % freq vector
x_DFT = fft(x,M);
h_DFT = fft(h,M);

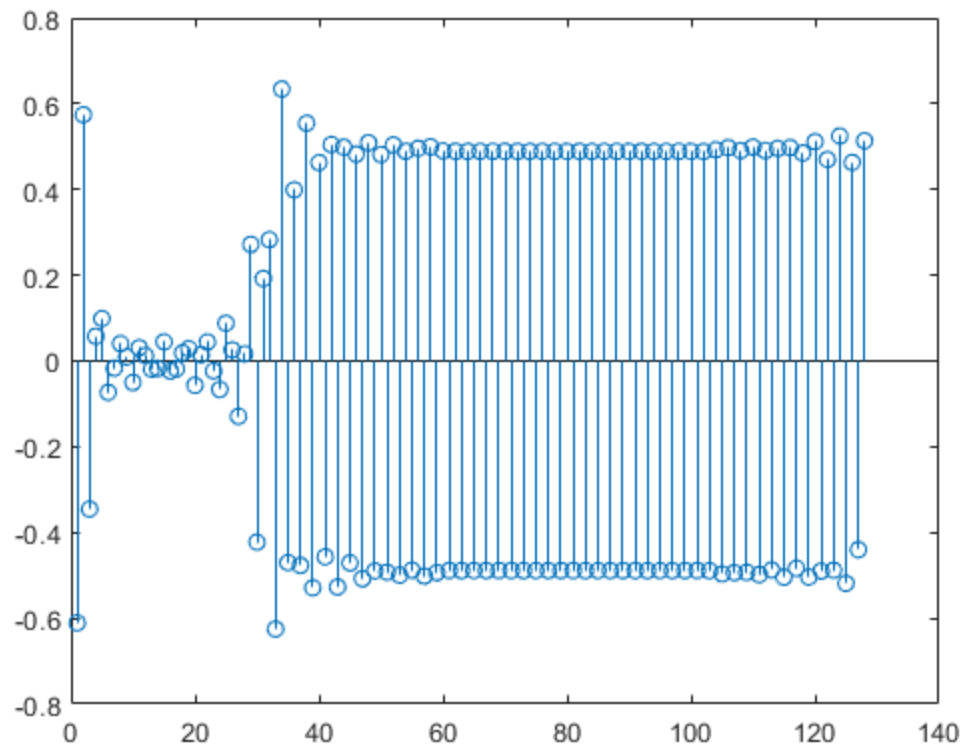
%(2)
Y128 = x_DFT.*h_DFT;

%(3)
y128 = ifft(Y128);

figure(8);
stem(y128);

% y128 is very similar to yconv, but not exactly the same. The first
% things is that the data is cut off at 128 points where the original
% had 160 points. Also some of the begining values see to be scaled
% up a little, but not too much. Once you get to around point 25, the
% values go back to normal. The reason they are different is because
% M (128) is less than the length of the convolution of x and h (160)
% so the DFT cannot be accurately constructed.
```





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