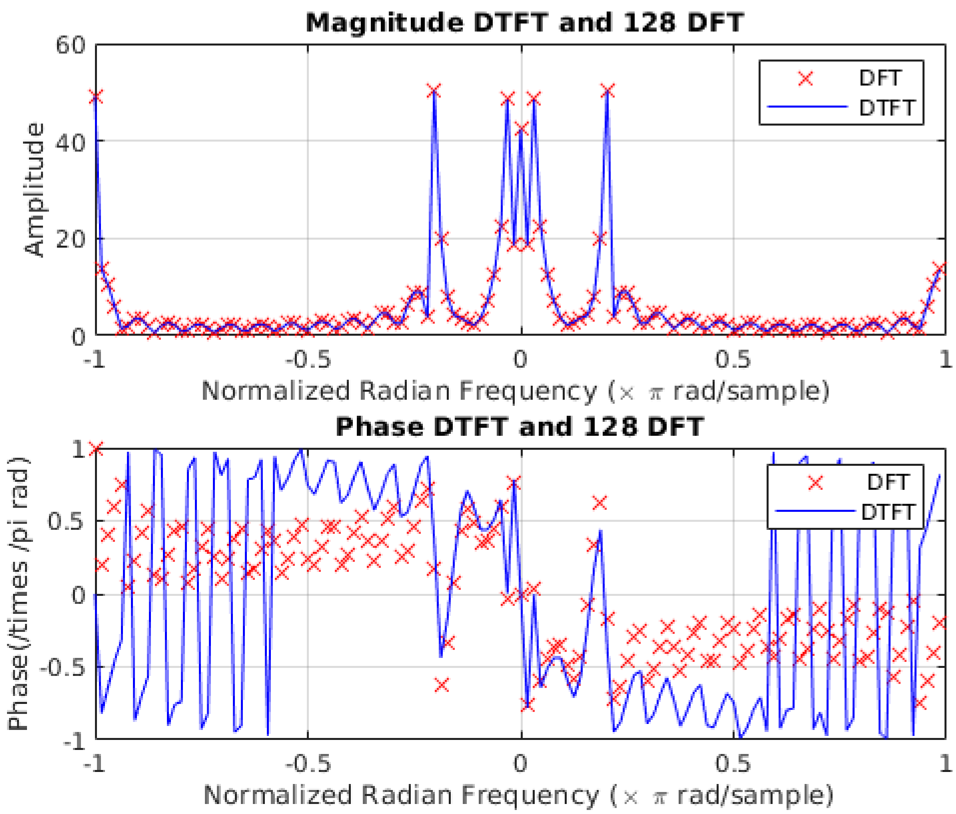
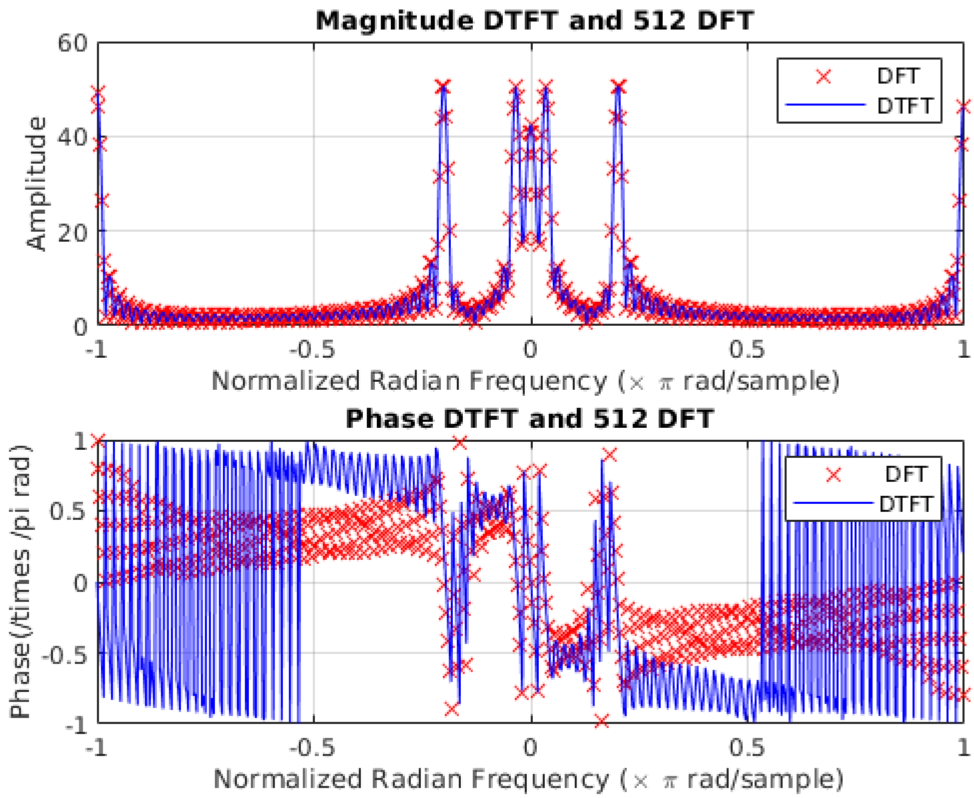
Exercise 9.1

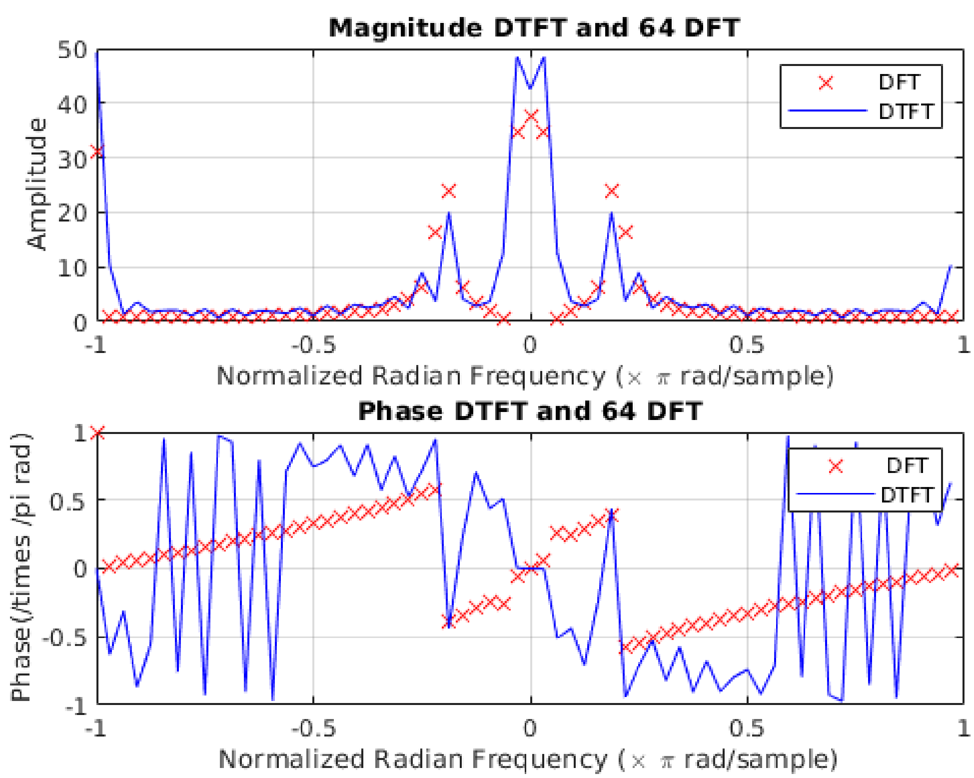
```
%a

X = zeros(1,100);
for n = 1:100
    X(n) = (0.5 + cos((pi/30)*(n-1)) + cos((pi/5)*(n-1)) +
 cos(pi*(n-1) + ...
    2*pi/3))* (un_dsp((n-1),0) - un_dsp((n-1),100));
end

M = 128;
w = - pi :2* pi / M : pi -2* pi / M ;
X_DTFT = dtft(X,w);

%b

X_DFT = fft(X,128);
% The DFT coefficients are frequency samples of the DTFT values. THe
 kth
% DFT coefficient is sampled at the frequency (2*pi*k)/128

% a and b plot




figure ;
subplot (2 , 1 , 1)
plot ( w / pi , fftshift ( abs ( X_DFT ) ) , 'rx ') ;
hold on ;
plot ( w / pi , abs ( X_DTFT ) , 'b - ') ;
hold off ;
grid on ;
title ( ' Magnitude DTFT and 128 DFT')
xlabel ( 'Normalized Radian Frequency (\times \pi rad/sample) ') ;
ylabel ( ' Amplitude ') ;
legend ( ' DFT ' , ' DTFT ') ;
subplot (2 , 1 , 2)
plot ( w / pi , fftshift ( angle ( X_DFT ) / pi ) , 'rx ') ;
hold on ;
plot ( w / pi , angle ( X_DTFT ) / pi , 'b - ') ;
hold off ;
grid on ;
title ( 'Phase DTFT and 128 DFT')
xlabel ( ' Normalized Radian Frequency (\times \pi rad/sample) ') ;
ylabel ( ' Phase(/times /pi rad) ') ;
legend ( ' DFT ' , 'DTFT') ;

% c
```

```
M = 512;
w = - pi :2* pi / M : pi -2* pi / M ;
X_DTFT = dtft(X,w);
X_DFT = fft(X,512);


figure ;
subplot (2 , 1 , 1)
plot ( w / pi , fftshift ( abs ( X_DFT ) ) , 'rx ') ;
hold on ;
plot ( w / pi , abs ( X_DTFT ) , 'b - ') ;
hold off ;
grid on ;
title ( ' Magnitude DTFT and 512 DFT')
xlabel ( 'Normalized Radian Frequency (\times \pi rad/sample) ') ;
ylabel ( ' Amplitude ') ;
legend ( ' DFT ' , ' DTFT ') ;
subplot (2 , 1 , 2)
plot ( w / pi , fftshift ( angle ( X_DFT ) / pi ) , 'rx ') ;
hold on ;
plot ( w / pi , angle ( X_DTFT ) / pi , 'b - ') ;
hold off ;
grid on ;
title ( 'Phase DTFT and 512 DFT')
xlabel ( ' Normalized Radian Frequency (\times \pi rad/sample) ') ;
ylabel ( ' Phase(/times /pi rad) ') ;
legend ( ' DFT ' , 'DTFT') ;

% The 512 point DFT has more samples, giving a more accurate
 transform.
% Using more points would give more accurate transforms as well.

%d
M = 64;
w = - pi :2* pi / M : pi -2* pi / M ;
X_DTFT = dtft(X,w);
X_DFT = fft(X,64);


figure ;
subplot (2 , 1 , 1)
plot ( w / pi , fftshift ( abs ( X_DFT ) ) , 'rx ') ;
hold on ;
plot ( w / pi , abs ( X_DTFT ) , 'b - ') ;
hold off ;
grid on ;
title ( ' Magnitude DTFT and 64 DFT')
xlabel ( 'Normalized Radian Frequency (\times \pi rad/sample) ') ;
ylabel ( ' Amplitude ') ;
legend ( ' DFT ' , ' DTFT ') ;
subplot (2 , 1 , 2)
plot ( w / pi , fftshift ( angle ( X_DFT ) / pi ) , 'rx ') ;
hold on ;
plot ( w / pi , angle ( X_DTFT ) / pi , 'b - ') ;
```

```
hold off ;
grid on ;
title ( 'Phase DTFT and 64 DFT')
xlabel ( ' Normalized Radian Frequency (\times \pi rad/sample) ') ;
ylabel ( ' Phase(/times /pi rad) ') ;
legend ( ' DFT ' , 'DTFT') ;

% The 64 point dft is less accurate than the 128 dft as it has less
 samples
```

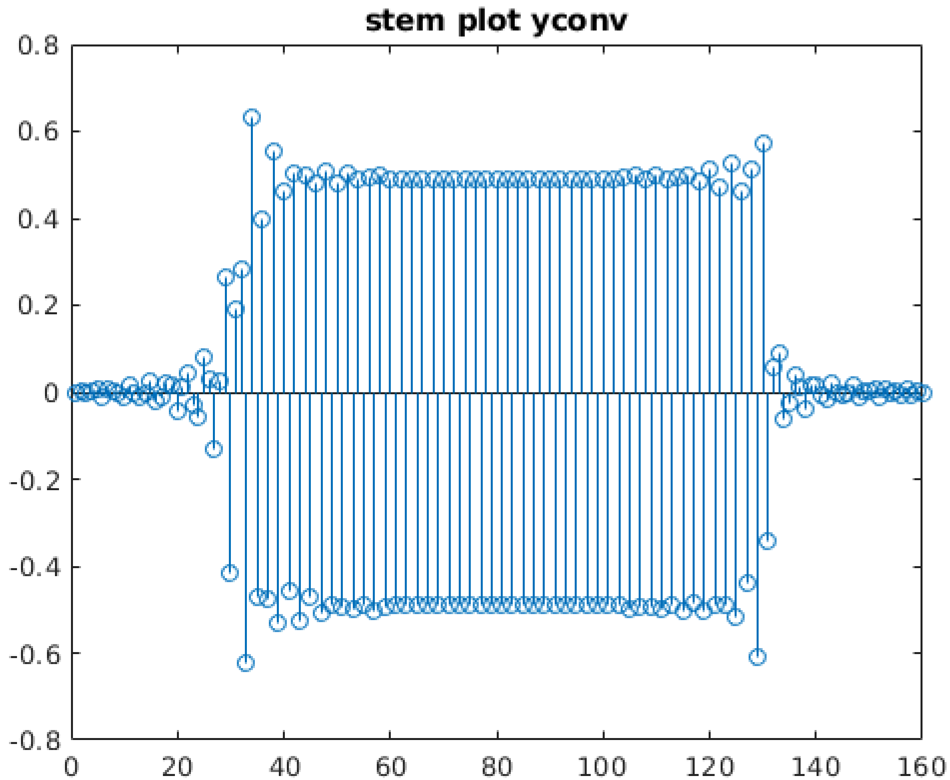


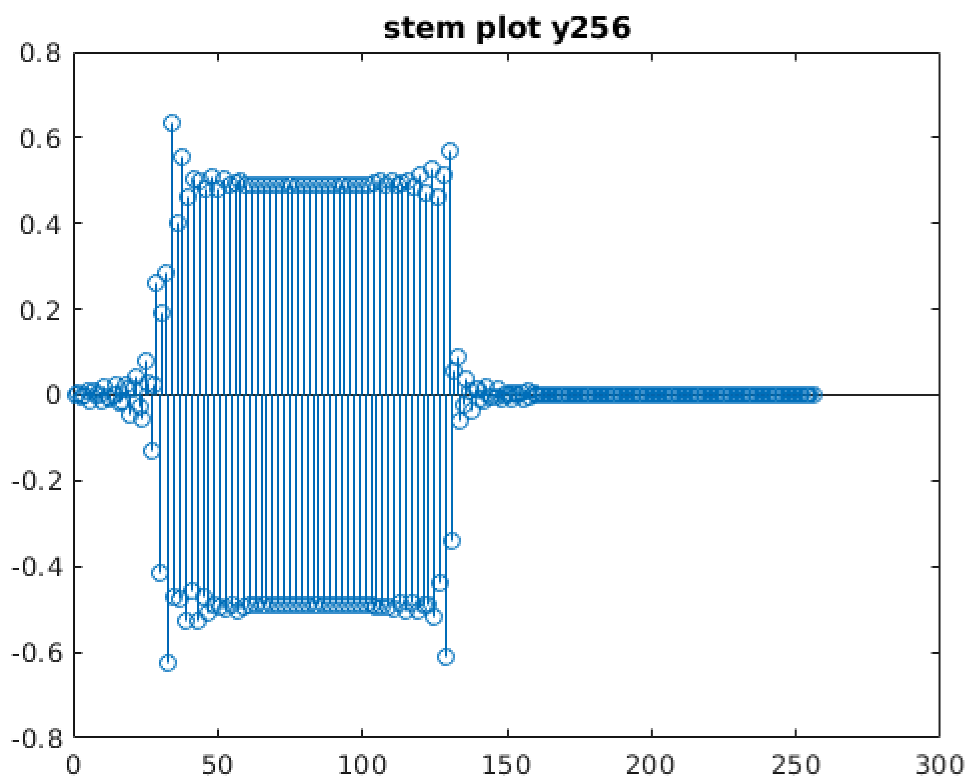

Magnitude DTFT and 512 DFT
Amplitude
Normalized Radian Frequency (× π rad/sample)
DFT
DTFT
Phase DTFT and 512 DFT
Phase(/times /pi rad)
Normalized Radian Frequency (× π rad/sample)
DFT
DTFT

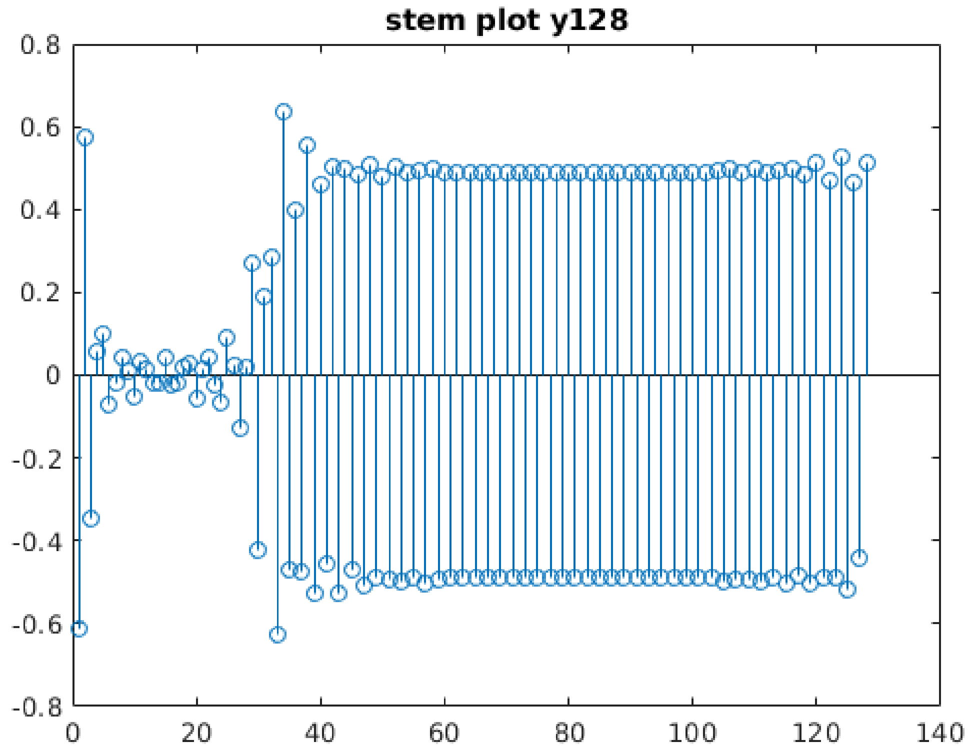

Magnitude DTFT and 64 DFT
Amplitude
Normalized Radian Frequency (× π rad/sample)
DFT
DTFT
Phase DTFT and 64 DFT
Phase(/times /pi rad)
Normalized Radian Frequency (× π rad/sample)
DFT
DTFT

Exercise 9.2

```
yconv = conv(h, X);

figure;
stem(yconv);
title('stem plot yconv');

% based on the characteristics of the filter, the plot can be
 explained
% as the majority of points are those which passed through with an
% absolute normalized frequency is at or above 0.5. As specidied by
 the
% filter.

%b
M = 256;

X_DFT = fft(X,256);
h_DFT = fft(h,256);

DFT_Y256 = X_DFT .* h_DFT;

y256 = ifft(DFT_Y256,256);

figure;
stem(y256);
title('stem plot y256');

% Yes the ifft gives a very similar output as the time domain
 analysis, at
% least as far as can be observed by viewing the plots

%c
M = 128;

X_DFT = fft(X,128);
h_DFT = fft(h,128);

DFT_Y128 = X_DFT .* h_DFT;

y128 = ifft(DFT_Y128,128);

figure;
stem(y128);
title('stem plot y128');

% The y128 does not seem the same by plot inspection, as it does not
 have
% as many samples, making the result less accurate;
```


stem plot yconv
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
20
40
60
80
100
120
140
160


stem plot y256
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
50
100
150
200
250
300



Published with MATLAB® R2020a