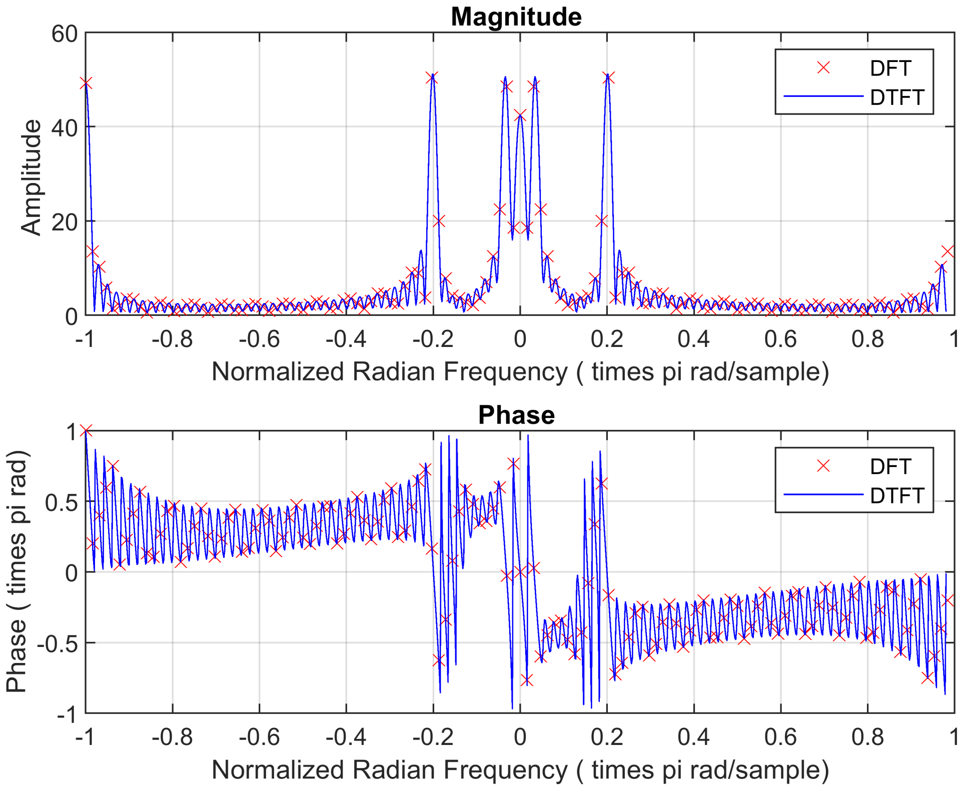
Exercise 9.1

Part A and B

```
N=100; % signal length
M=2^7; % FFT size (M >= N)
n=0:N-1; % discrete time vector
ustep=ones(1,100);
u=ustep>0;
u_minus100=ustep>100;
x = (0.5 + cos((pi/30).*n) + cos((pi/5).*n) + cos(pi.*n + (2*pi/3))).*(u-u_minus100);
% Calculate DFT and measure calculation time
w_DFT = -pi:2*pi/M:pi -2*pi/M; % for plotting only
X_DFT = fft(x,M);
% Calculate DTFT and measure calculation time
w_DTFT = -pi:2*pi/(N*10):pi -2*pi/N; % freq vector
X_DTFT = dtft(x,w_DTFT);
% Plot magnitude and phase of DTFT (bule solid line) and DFT (read x)
figure(1);
subplot(2, 1, 1)
plot(w_DFT/pi , fftshift(abs(X_DFT)), 'rx ');
hold on;
plot(w_DTFT/pi , abs(X_DTFT), 'b-');
hold off;
grid on;
title('Magnitude ')
xlabel('Normalized Radian Frequency ( times pi rad/sample)');
ylabel('Amplitude ');
legend('DFT ','DTFT ');
subplot(2, 1, 2)
plot(w_DFT/pi , fftshift(angle(X_DFT)/pi), 'rx ');
hold on;
plot(w_DTFT/pi , angle(X_DTFT)/pi , 'b-');
hold off;
grid on;
title('Phase ')
xlabel('Normalized Radian Frequency ( times pi rad/sample)');
ylabel('Phase ( times pi rad)');
legend('DFT ','DTFT ');
```

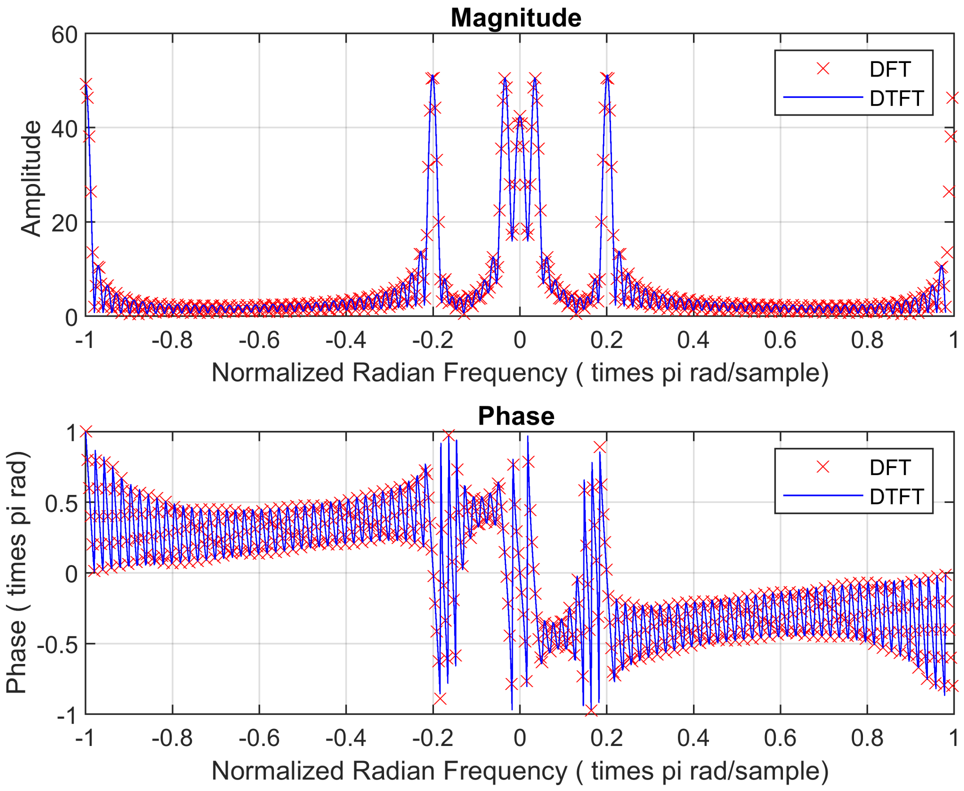


The DFT coefficients are the frequency samples of the DTFT. They overlay nice and snug. The sample frequency of the kth coefficient is (pi*k)/64

Part C

```
N=100; % signal length
M=2^9; % FFT size (M >= N)
n=0:N-1; % discrete time vector
ustep=ones(1,100);
u=ustep>0;
u_minus100=ustep>100;
x = (0.5 + cos((pi/30).*n) + cos((pi/5).*n) + cos(pi.*n + (2*pi/3))).*(u-u_minus100);
% Calculate DFT and measure calculation time
w_DFT = -pi:2*pi/M:pi -2*pi/M; % for plotting only
X_DFT = fft(x,M);
% Calculate DTFT and measure calculation time
w_DTFT = -pi:2*pi/(N*10):pi -2*pi/N; % freq vector
X_DTFT = dtft(x,w_DTFT);
% Plot magnitude and phase of DTFT (bule solid line) and DFT (read x)
figure(2);
subplot(2, 1, 1)
plot(w_DFT/pi , fftshift(abs(X_DFT)), 'rx ');
hold on;
plot(w_DTFT/pi , abs(X_DTFT), 'b-');
hold off;
grid on;
title('Magnitude ')
```

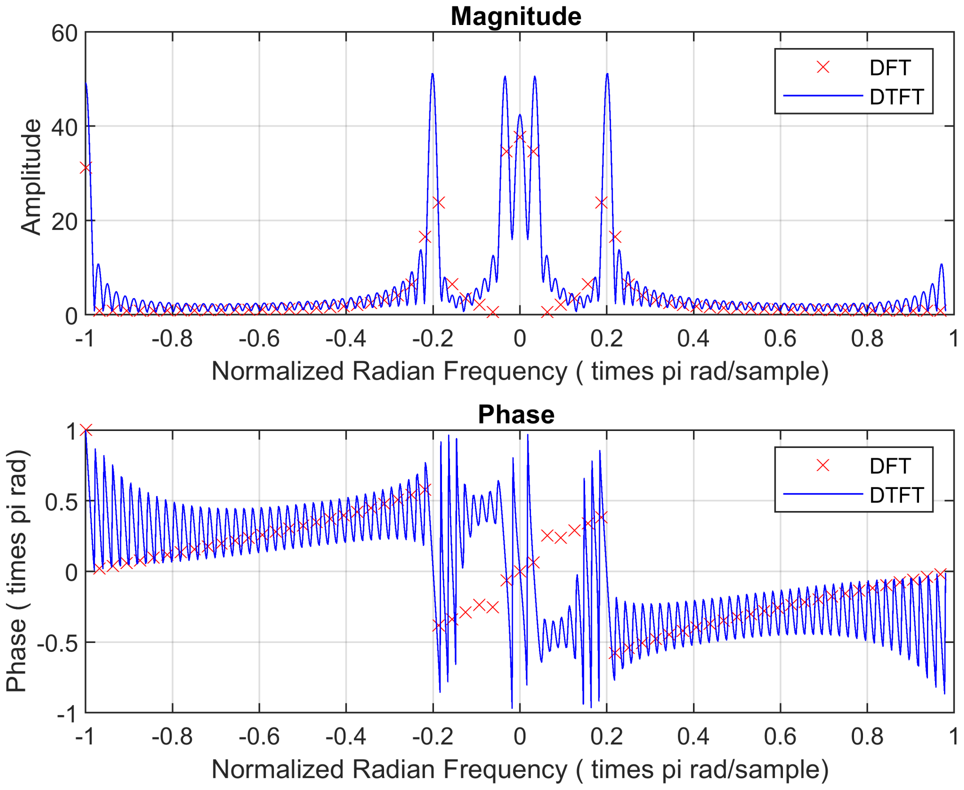
```
xlabel('Normalized Radian Frequency ( times pi rad/sample)');
ylabel('Amplitude ');
legend('DFT ','DTFT ');
subplot(2, 1, 2)
plot(w_DFT/pi , fftshift(angle(X_DFT)/pi), 'rx ');
hold on;
plot(w_DTFT/pi , angle(X_DTFT)/pi , 'b-');
hold off;
grid on;
title('Phase ')
xlabel('Normalized Radian Frequency ( times pi rad/sample)');
ylabel('Phase ( times pi rad)');
legend('DFT ','DTFT ');
```



The sample frequency of the kth coefficient is (pi*k)/256. Compared to the 128 point FFT, this transformation has more detail (more coefficients). Using an even larger DFT will result in an even more accurate coefficient vector.

```
N=100; % signal length
M=2^6; % FFT size (M >= N)
n=0:N-1; % discrete time vector
ustep=ones(1,100);
u=ustep>0;
u_minus100=ustep>100;
x = (0.5 + cos((pi/30).*n) + cos((pi/5).*n) + cos(pi.*n + (2*pi/3))).*(u-u_minus100);
% Calculate DFT and measure calculation time
w_DFT = -pi:2*pi/M:pi -2*pi/M; % for plotting only
```

```
X_DFT = fft(x,M);
% Calculate DTFT and measure calculation time
w_DTFT = -pi:2*pi/(N*10):pi -2*pi/N; % freq vector
X_DTFT = dtft(x,w_DTFT);
% Plot magnitude and phase of DTFT (bule solid line) and DFT (read x)
figure(2);
subplot(2, 1, 1)
plot(w_DFT/pi , fftshift(abs(X_DFT)), 'rx ');
hold on;
plot(w_DTFT/pi , abs(X_DTFT), 'b-');
hold off;
grid on;
title('Magnitude ')
xlabel('Normalized Radian Frequency ( times pi rad/sample)');
ylabel('Amplitude ');
legend('DFT ','DTFT ');
subplot(2, 1, 2)
plot(w_DFT/pi , fftshift(angle(X_DFT)/pi), 'rx ');
hold on;
plot(w_DTFT/pi , angle(X_DTFT)/pi , 'b-');
hold off;
grid on;
title('Phase ')
xlabel('Normalized Radian Frequency ( times pi rad/sample)');
ylabel('Phase ( times pi rad)');
legend('DFT ','DTFT ');
```



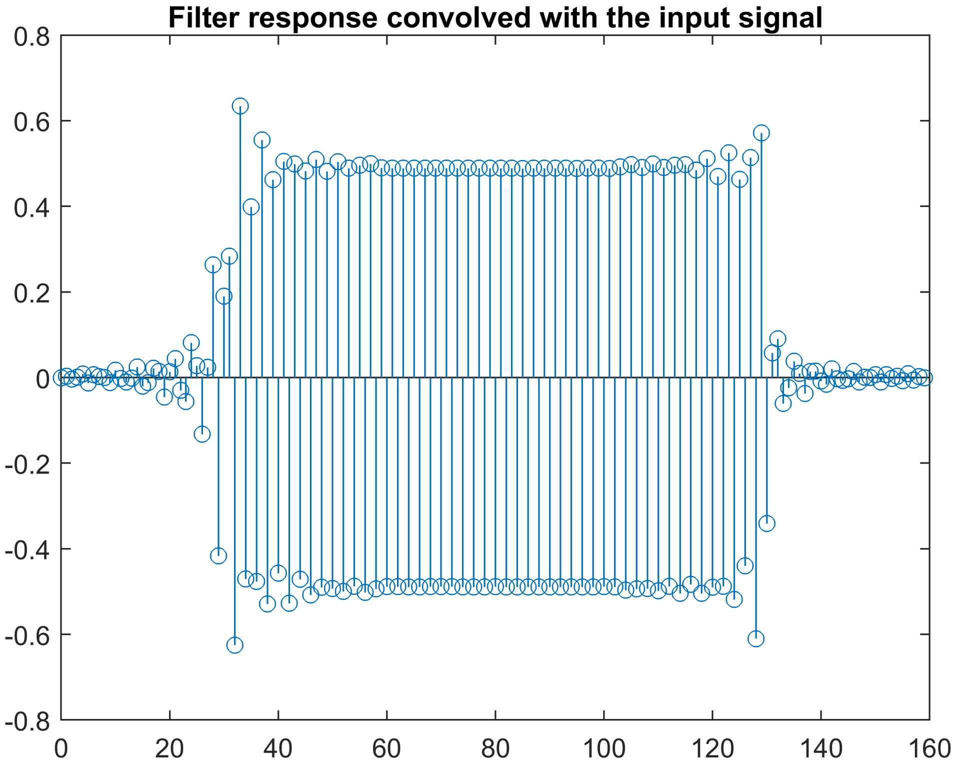
The sample frequency of the kth coefficient is (pi*k)/32. Compared to the 128 point FFT, this transformation has less detail (less coefficients). It does not perfectly resemble the DTFT, and there are instances where the DFT is not near the value of the DTFT (between -0.2 and 0.2pi)

Exercise 9.2

Part A

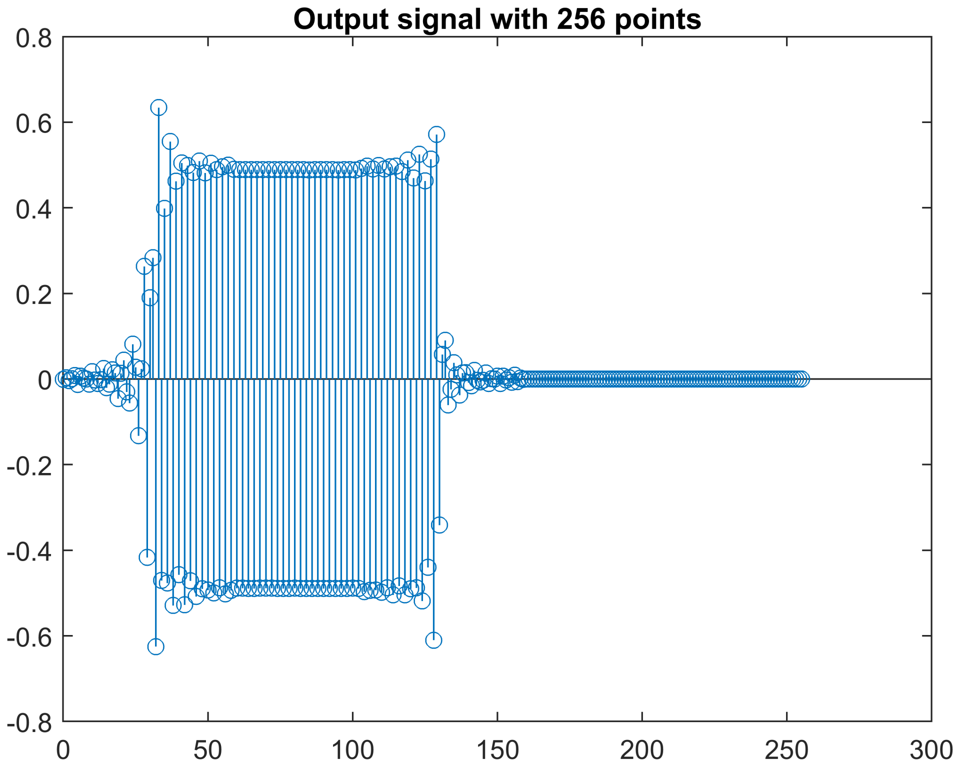
Filter coefficients saved in **b**

```
yconv = conv(x,b);
figure(3)
stem(0:159,yconv);
title('Filter response convolved with the input signal');
```



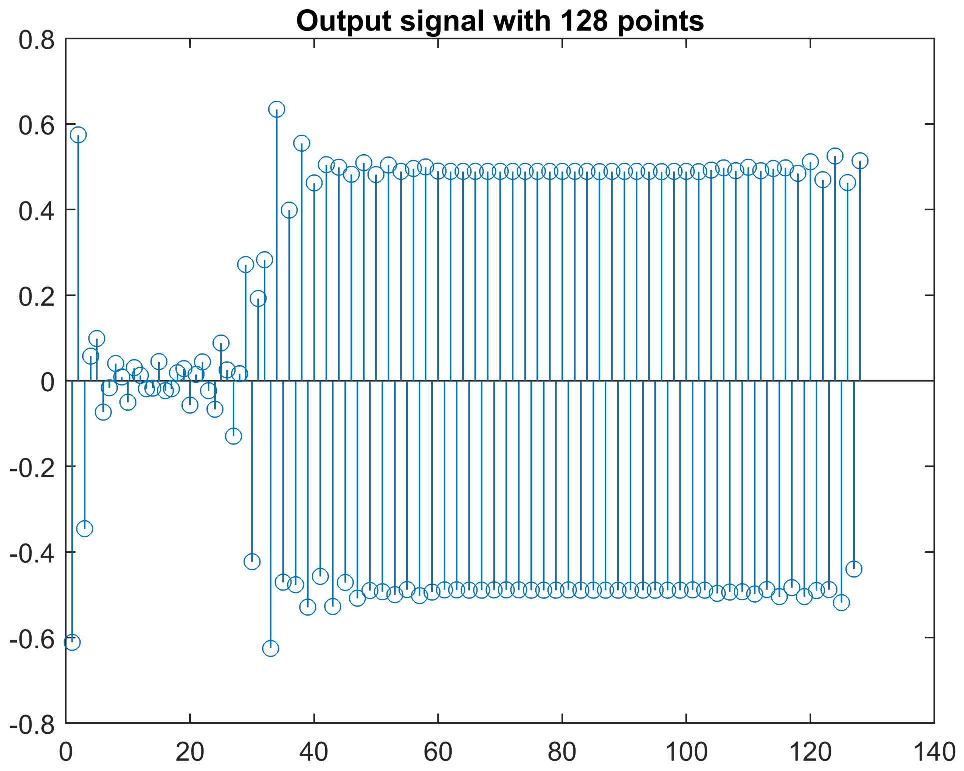
Part B

```
M=2^8; % FFT size (M >= N)
X_DFT = fft(x,M);
H_DFT = fft(b,M);
Y256 = X_DFT.*H_DFT;
y256 = ifft(Y256);
stem(0:255,y256);
title('Output signal with 256 points')
```



The frequency domain analysis plot gives the same appearance as yconv. However, the frequency domain analysis returns a longer output.

```
M=2^7; % FFT size (M >= N)
X_DFT = fft(x,M);
H_DFT = fft(b,M);
Y128 = X_DFT.*H_DFT;
y128 = ifft(Y128);
stem(1:128,y128);
title('Output signal with 128 points')
```


Output signal with 128 points

The frequency domain analysis plot gives the same appearance as yconv. However, the frequency domain analysis returns a shorter output. This is because the length M is limited to 128, so it does not compute past that and its accuracy is therfore lower.

```
function H = dtft(b, w)
    H = 0;
    for k = 1:length(b)
        H = H + b(k) * exp(-1j*w*(k-1));
    end
end
```