ECE 311

Lab2 report

Name: Yang Shi

NetID: yangshi5

Report Item 1:

N = 8000/8 = 1000. So the oscilloscope can contain 1000 samples in the same time. $f_s = 1000$ Hz. According to Shannon-Nyquist sampling theory, $f_s > 2*f_{max}$. Therefore, $f_{max} < 1000/2 = 500$ Hz in order to avoid aliasing.

$\Delta t = 1/f_s = 0.001$. According to Rayleigh limit, $N*\Delta t > 1/f_{min}$, $1000*0.001 = 1 > 1/f_{min}$. So $f_{min} > 1$ Hz.

Therefore, the maximum frequency component I can capture with that oscilloscope without aliasing is 500 Hz and the minimum frequency component is 1 Hz.

In a single acquisition memory, there is only a snapshot of the signal. Since the waveforms with different frequencies will add up with each other, it is difficult to actually separate them apart using a single acquisition memory.

Report Item 2:

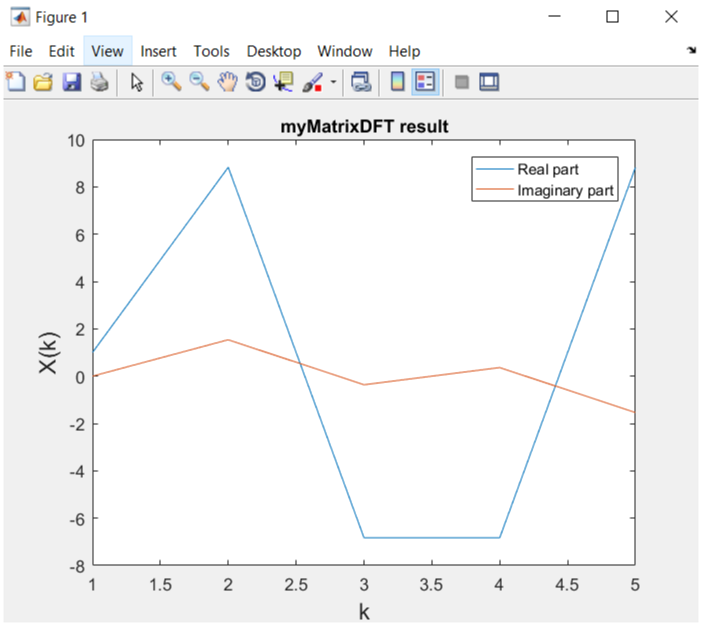
$X_d(\omega)$ = sum of $x(t)*e^{-i\omega n}$ for n = 0 to N-1. And $e^{-i\omega n} = \cos(\omega n) – i*\sin(\omega n)$. Since n is an integer, when 2π is added on ω, the result will be exactly the same. Therefore, there will be a lot of spectral copies. The distance between two copies will be 2π normally. The range of ω having the physical significance is from -2π to 2π.

Report Item 3:

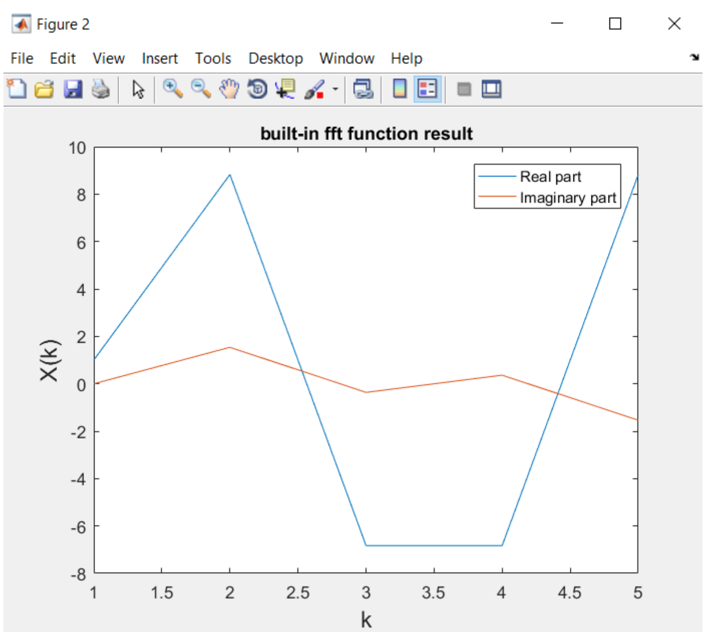
Code part:

```
function [Xk] = myMatrixDFT(x)
[N, M] = size(x);
Xk = zeros(N, 1);
Z = zeros(N); % DFT matrix initialization

for k = 0 : (N-1)
    for j = 0 : (N-1)
        Z(k+1, j+1) = exp(-1i*2*pi*k*j/N);
    end
end

Xk = Z*x;

Y = fft(x);
Xreal = real(Xk);
Ximag = imag(Xk);
Yreal = real(Y);
Yimag = imag(Y);

figure(1);
plot(Xreal);
hold on
plot(Ximag);
xlabel('k','fontsize',14);
ylabel('X(k)','fontsize',14);
legend('Real part','Imaginary part');
title('myMatrixDFT result');

figure(2);
plot(Yreal);
hold on;
plot(Yimag);
xlabel('k','fontsize',14);
ylabel('X(k)','fontsize',14);
title('built-in fft function result');
legend('Real part','Imaginary part');

end
```

Explanation:
The figure 1 below is the plot for the DFT of x = [1; 3; -4; -3; 4] using my own implemented function. And it contains the output using plot function.



The figure 2 is the plot for the DFT of x = [1; 3; -4; -3; 4] using MATLAB's build-in fft function. And it contains the output using plot function.



Since these two plots are exactly the same, the result of my Matrix DFT function ought to be correct. DFT function will do a multiplication and addition operation for every entre in the DFT matrix and the size of matrix is N*N. Therefore, the DFT is O(N^2).

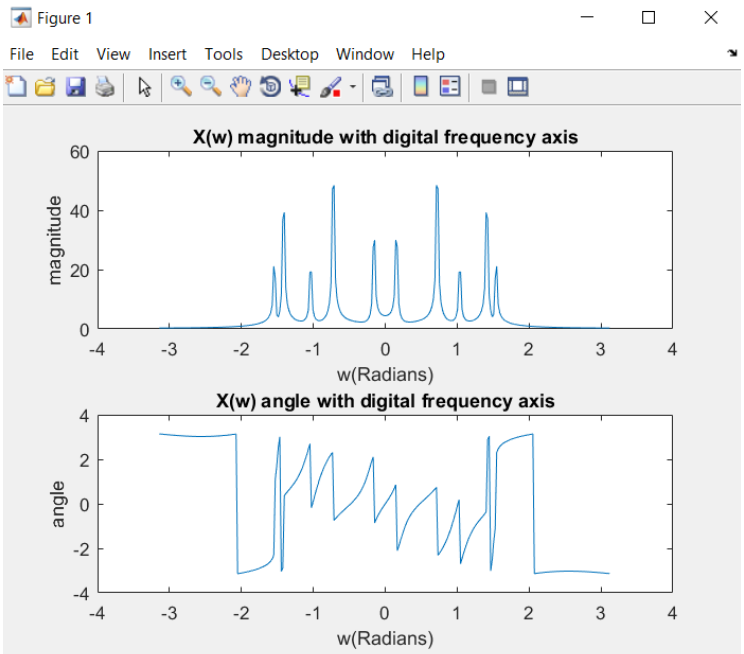
Report Item 4:

Code part:

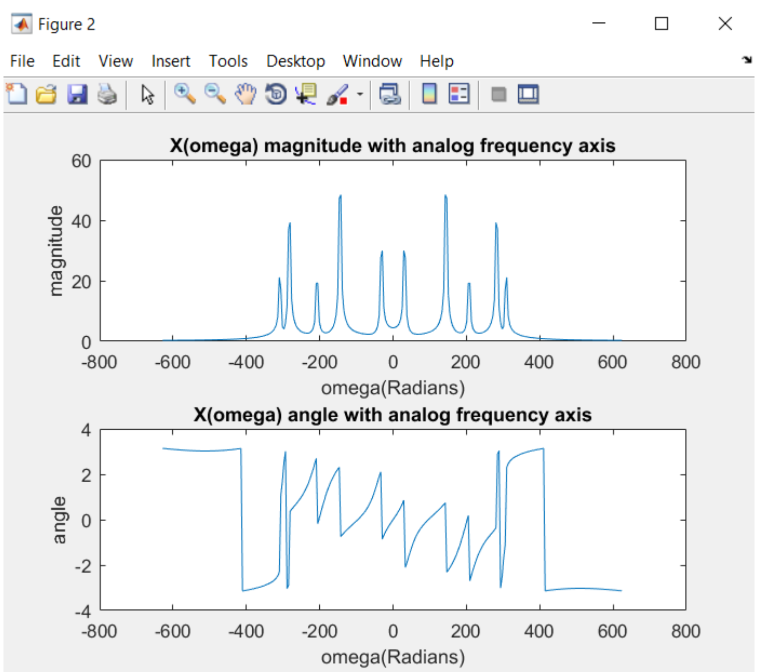
```
%Report_Item_4
load('signal.mat');
Xw = fft(x); %calculate fft
[M, N] = size(x); % N is the size of x, M = 1
fs = 200; %sampling rate

shift_Xw = fftshift(Xw); %shift xw
w = fftshift((0:N-1)/N*2*pi);
w(1:N/2) = w(1:N/2)-2*pi; %shift w

omega = fs*w; %get corrsponding omega

figure(1);
subplot(2, 1, 1);
plot(w, abs(shift_Xw));
xlabel('w(Radians)');
ylabel('magnitude');
title('X(w) magnitude with digital frequency axis');
subplot(2, 1, 2);
plot(w, angle(shift_Xw));
xlabel('w(Radians)');
ylabel('angle');
title('X(w) angle with digital frequency axis');

figure(2);
subplot(2, 1, 1);
plot(omega, abs(shift_Xw));
xlabel('omega(Radians)');
ylabel('magnitude');
title('X(omega) magnitude with analog frequency axis');
subplot(2, 1, 2);
plot(omega, angle(shift_Xw));
xlabel('omega(Radians)');
ylabel('angle');
title('X(omega) angle with analog frequency axis');
```

Explanation part:

The figure 1 is the plot for the amplitude and angle of X(ω) vs. ω.



The figure 2 is the plot for the amplitude and angle of X(Ω) vs. Ω.



From the plots, we can see that there are 5 frequency tones in the signal.

And their values in radians are: 33.51, 142.4, 209.4, 280.6, 310. (finding these values using data cursor in the Matlab plot)

Therefore, the values of these frequencies in Hz are the values above divided by 2π: 5.33Hz, 22.66Hz, 33.33Hz, 44.66Hz, 49.34Hz.

Report Item 5:

Code part:

```
%Report_Item_5
load('NMRSpec.mat');

%calculate the fft and then shift it
origin = fft(st);
st_length = length(st);
shift_origin = fftshift(origin);
w = fftshift((0:st_length-1)/st_length*2*pi);
w(1:st_length/2) = w(1:st_length/2)-2*pi;
%plot the origin spectrum
figure(1);
subplot(2, 1, 1);
plot(BW*w/(2*pi), abs(shift_origin));
xlabel('frequency (Hz)');
ylabel('amplitude');
title('original signal amplitude vs. frequency');
subplot(2, 1, 2);
plot(BW*w/(2*pi), angle(shift_origin));
xlabel('frequency (Hz)');
ylabel('angle');
title('original signal angle vs. frequency');

%calculate the 32-point DFT
shorter = st(1:32);
shorterfft = fft(shorter);
shorter_shift = fftshift(shorterfft);

ww = fftshift((0:32-1)/32*2*pi);
ww(1:32/2) = ww(1:32/2)-2*pi;
%plot the 32-point spectrum
figure(2);
subplot(2, 1, 1);
plot(BW*ww/(2*pi), abs(shorter_shift));
xlabel('frequency (Hz)');
ylabel('amplitude');
title('32-point DFT spectrum amplitude vs. frequency');
subplot(2, 1, 2);
plot(BW*ww/(2*pi), angle(shorter_shift));
xlabel('frequency (Hz)');
```

```
ylabel('angle');
title('32-point DFT spectrum angle vs. frequency');

%zero-pad the 32-point DFT to 512 points
longer = zeros(512, 1);
longer(1:32) = shorter;
longerfft = fft(longer);
longer_shift = fftshift(longerfft);

www = fftshift((0:512-1)/512*2*pi);
www(1:512/2) = www(1:512/2)-2*pi;
%plot 512-points DFT spectrum
figure(3);
subplot(2, 1, 1);
plot(BW*www/(2*pi), abs(longer_shift));
xlabel('frequency (Hz)');
ylabel('amplitude');
title('512-point DFT spectrum amplitude vs. frequency');
subplot(2, 1, 2);
plot(BW*www/(2*pi), angle(longer_shift));
xlabel('frequency (Hz)');
ylabel('angle');
title('512-point DFT spectrum angle vs. frequency');
```

Explanation part:

The figure 1 below is the plot of magnitude and phase spectrum of the original signal.

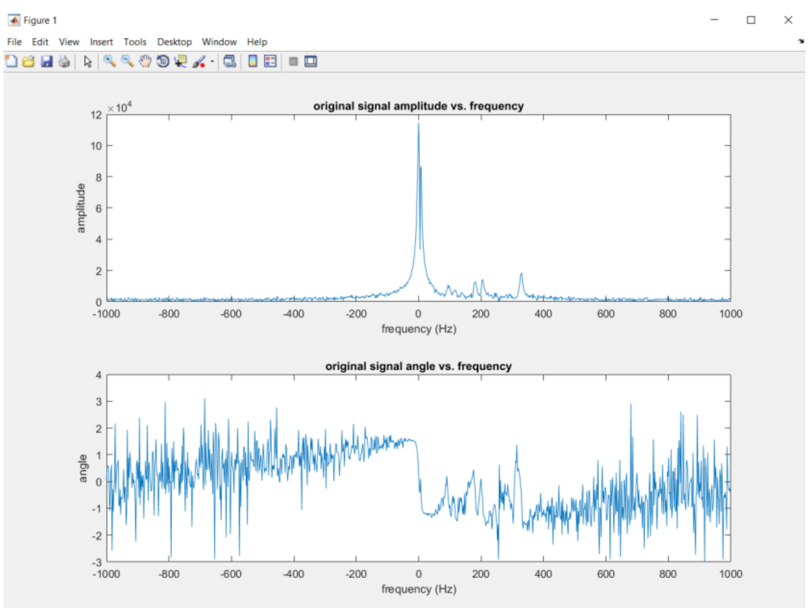


Figure 2 is the plot of magnitude and phase spectrum of the 32-point signal.

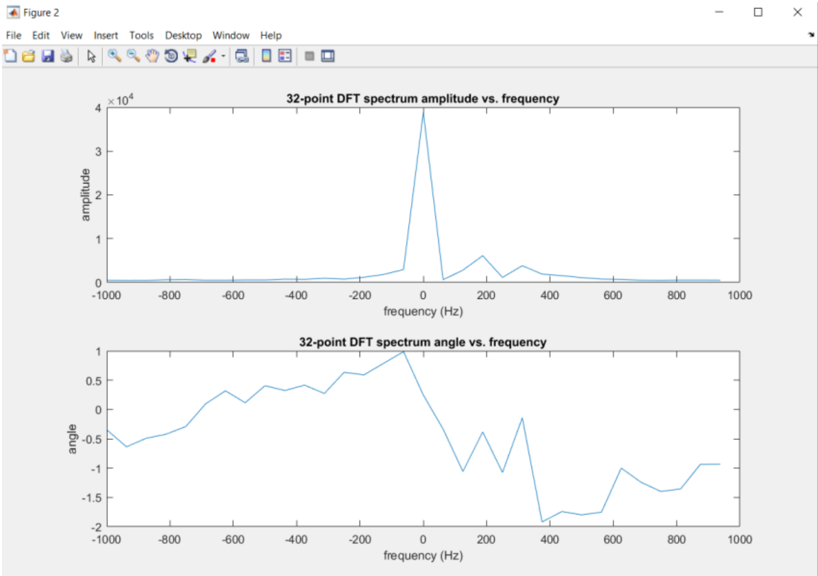
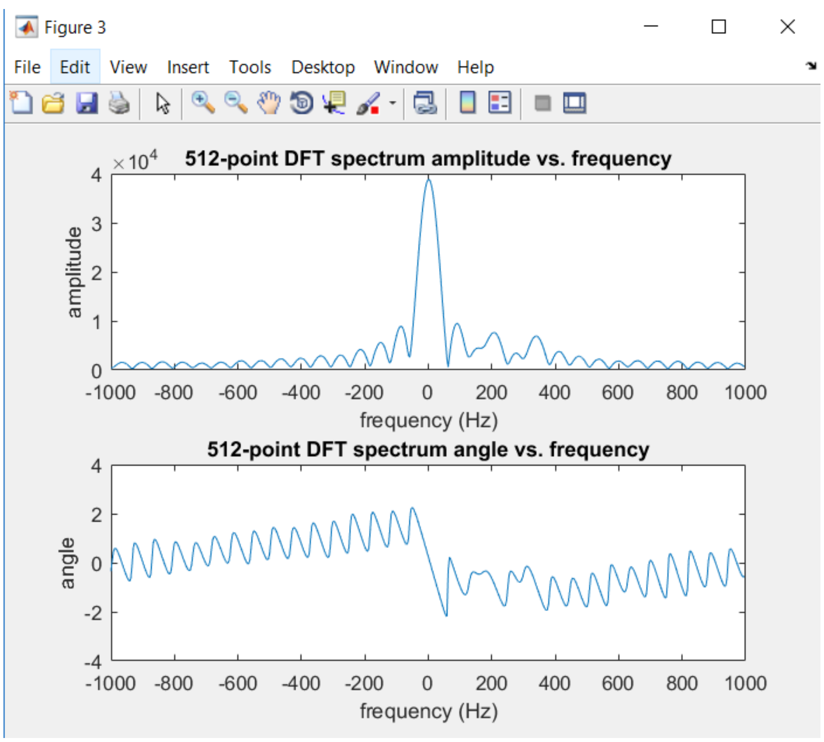


Figure 3 is the plot of magnitude and phase spectrum of the zero-padding 512-point signal.



From the original spectrum, we can find two peaks at around 185 Hz and 209 Hz. But in the spectrum of 32-point DFT, we can't distinguish them. Zero-padding method make the spectrum resolution clearer. And we can almost see the peak at around 185 Hz. Zero-padding method does help us to improve the visual quality because zero-padding makes a longer FFT. Therefore, there will be more frequency bins available. In that case, more frequency peaks will be shown in the figure.

Report Item 6:

Code part:

```
%Report_Item_6
N = 20;
M = 512;
origin = zeros(1, M);
%get frequency response of rectangular window
rect = origin;
rect(1:N) = 1;
fftrect = fftshift(fft(rect));
w = fftshift((0:M-1)/M*2*pi);
w(1:M/2) = w(1:M/2)-2*pi;
%plot rectangular window
figure(1);
subplot(2, 1, 1);
stem(rect(1:N));
xlabel('n');
ylabel('w(n)');
title('rectangular window')

subplot(2, 1, 2);
plot(w, mag2db(abs(fftrect)));
xlabel('w');
ylabel('dB');
title('magnitude responce of zero-padding (512 points)
rectangular window');

%get frequency response of triangular window
tri = origin;
for k = 1 : N/2+1
    tri(1, k) = 2*(k-1)/N;
end
for j = (N/2+2) : N
    tri(1, j) = tri(1, N+2-j);
end
ffttri = fftshift(fft(tri));

%plot triangular window
figure(2);
subplot(2, 1, 1);
```

```
stem(tri(1:N));
xlabel('n');
ylabel('w(n)');
title('triangular window')

subplot(2, 1, 2);
plot(w, mag2db(abs(ffttri)));
xlabel('w');
ylabel('dB');
title('magnitude responce of zero-padding (512 points)
triangular window');

%get frequency response of hamming window
hamm = origin;
for j = 1 : N
    hamm(j) = 0.54 - 0.46*cos(2*pi*(j-1)/N);
end

ffthamm = fftshift(fft(hamm));
%plot hamming window
figure(3);
subplot(2, 1, 1);
stem(hamm(1:N));
xlabel('n');
ylabel('w(n)');
title('hamming window');

subplot(2, 1, 2);
plot(w, mag2db(abs(ffthamm)));
xlabel('w');
ylabel('dB');
title('magnitude responce of zero-padding (512 points) hamming
window');

%get frequency response of hanning window
hann = origin;
for j = 1 : N
    hann(j) = 0.5 - 0.5*cos(2*pi*(j-1)/N);
end

ffthann = fftshift(fft(hann));
%plot hanning window
figure(4);
subplot(2, 1, 1);
stem(hann(1:N));
xlabel('n');
ylabel('w(n)');
```

```
title('hanning window');

subplot(2, 1, 2);
plot(w, mag2db(abs(ffthann)));
xlabel('w');
ylabel('dB');
title('magnitude responce of zero-padding (512 points) hanning
window');

%get frequency response of kaiser window
kaiser = origin;
beta = 0.1;
Z = 0 : (N-1)
%calculate x in the bessel function
for k = 1 : N
    Z(k) = beta*sqrt(1-((Z(k)-N/2)/(N/2))*((Z(k)-N/2)/(N/2)));
end
J = besselj(0, Z);%get bessel function

kaiser(1:N) = J;

fftkaiser = fftshift(fft(kaiser));
%plot kaiser window
figure(5);
subplot(2, 1, 1);
stem(kaiser(1:N));
xlabel('n');
ylabel('w(n)');
title('kaiser window');

subplot(2, 1, 2);
plot(w, mag2db(abs(fftkaiser)));
xlabel('w');
ylabel('dB');
title('magnitude responce of zero-padding (512 points) kaiser
window');
```

Explanation part:

Figure 1 is the plot of rectangular window and its magnitude response zero-padded 512 points in decibels:

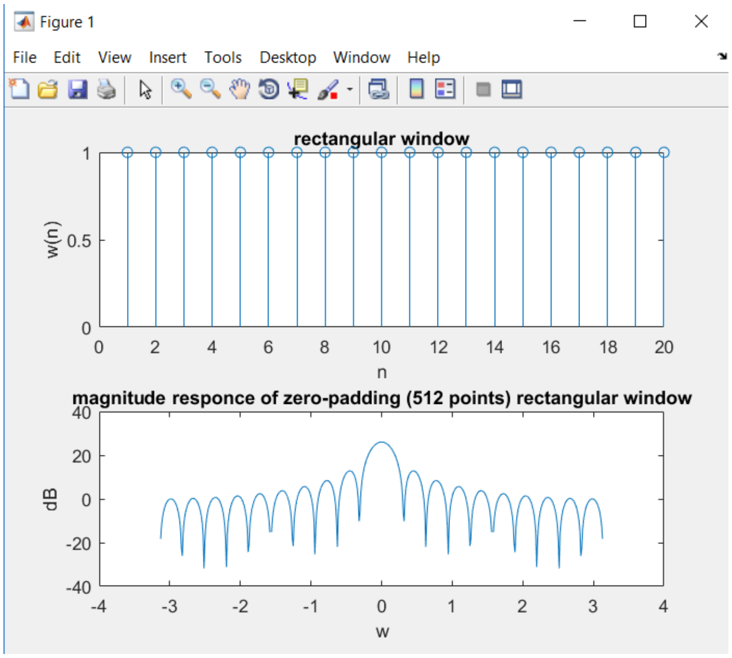


Figure 2 is the plot of triangular window and its magnitude response zero-padded 512 points in decibels:

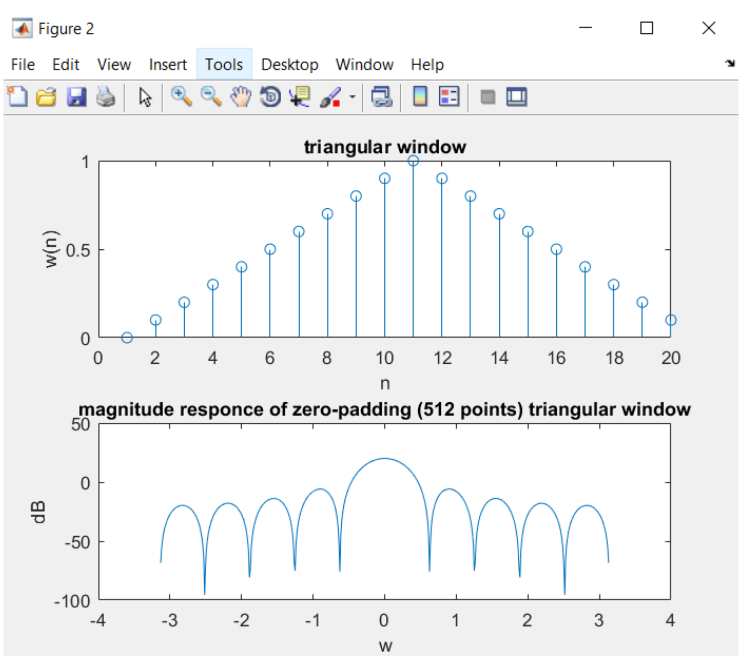


Figure 3 is the plot of hamming window and its magnitude response zero-padded 512 points in decibels:

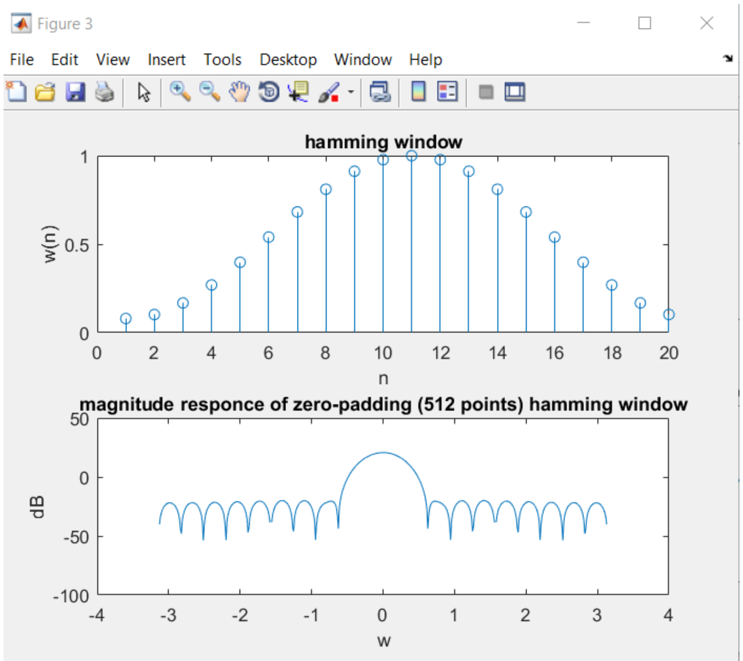


Figure 4 is the plot of hanning window and its magnitude response zero-padded 512 points in decibels:

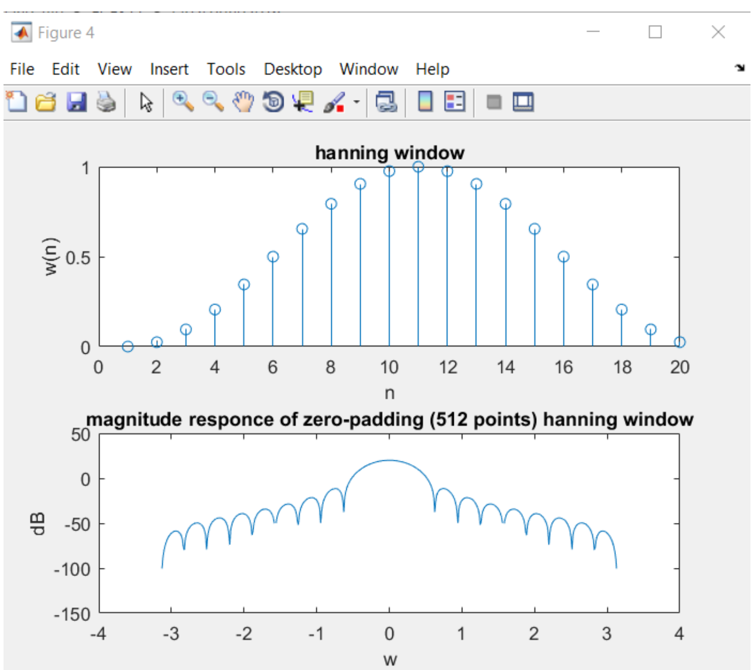
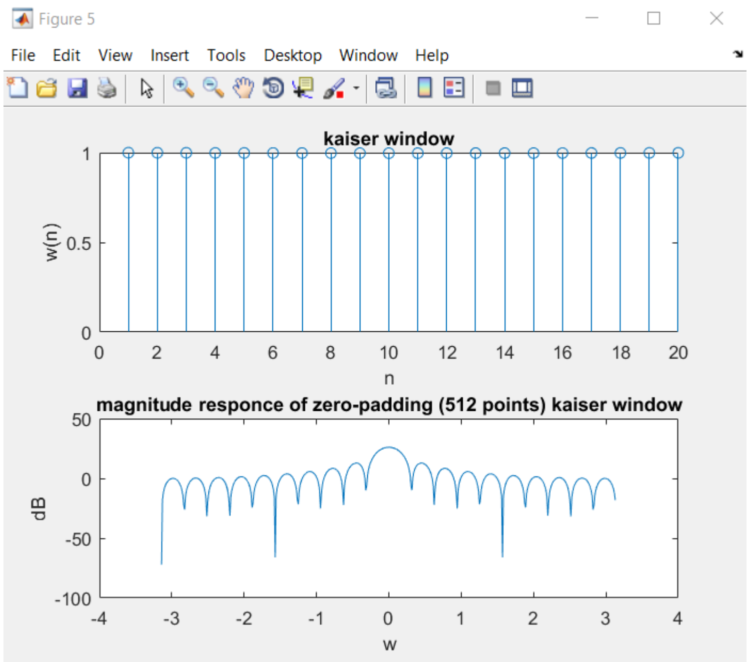


Figure 4 is the plot of kaiser window and its magnitude response zero-padded 512 points in decibels:



Mainlobe width of the rectangular window is smaller than the one of triangular window. The sidelobe height of the rectangular window is higher than the one of triangular window. Hamming window has lower side lobes than the ones of rectangular window.

Report Item 7:

Code part:

```
%Report_Item_7
N1 = 20;
w = linspace(0, 1, N1);
Xw = exp(-1i*w*N1/2)./exp(-1i*w/2)*N1.*diric(w, N1);
%plot the magnitude and phase response when N = 20;
figure(1);
subplot(2, 1, 1);
plot(w, abs(Xw));
xlabel('radians');
ylabel('amplitude');
title('magnitude response of rectangular window when N = 20');
subplot(2, 1, 2);
plot(w, angle(Xw));
xlabel('radians');
ylabel('angle');
```

```
title('phase response of rectangular window when N = 20');

%plot the magnitude and phase response when N = 40
N2 = 40;
ww = linspace(0, 1, N2);
Xww = exp(-1i*ww*N2/2)./exp(-1i*ww/2)*N2.*diric(ww, N2);
%plot the magnitude and phase response when N = 40;
figure(2);
subplot(2, 1, 1);
plot(ww, abs(Xww));
xlabel('radians');
ylabel('amplitude');
title('magnitude response of rectangular window when N = 40');
subplot(2, 1, 2);
plot(ww, angle(Xww));
xlabel('radians');
ylabel('angle');
title('phase response of rectangular window when N = 40');
```

Explanation part:

Figure 1 below is the magnitude and phase response of rectangular window when N = 20:

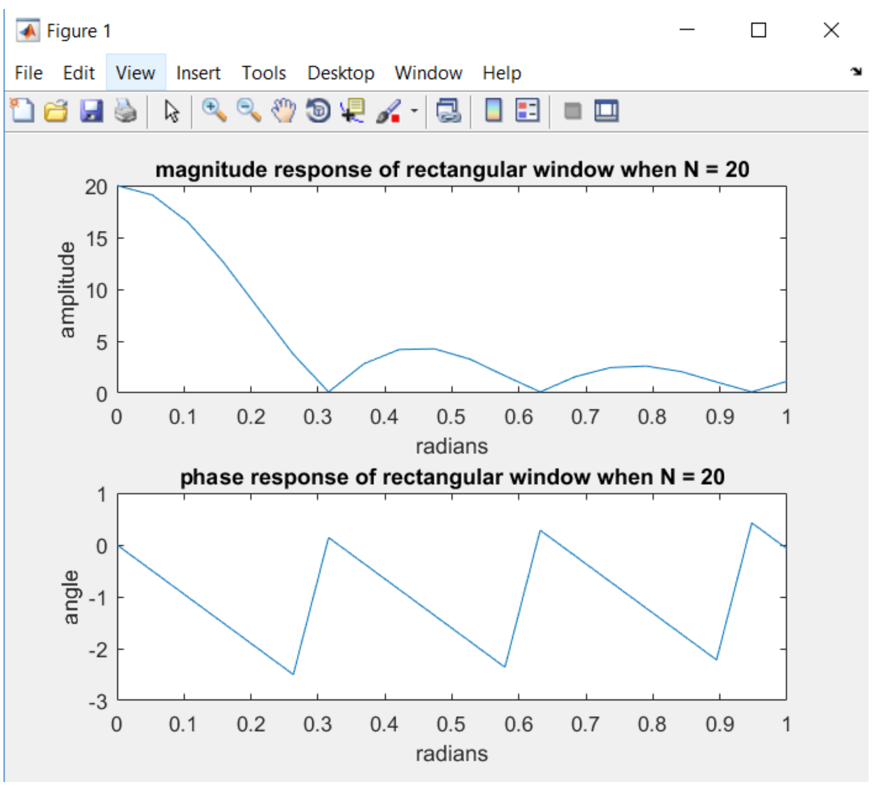
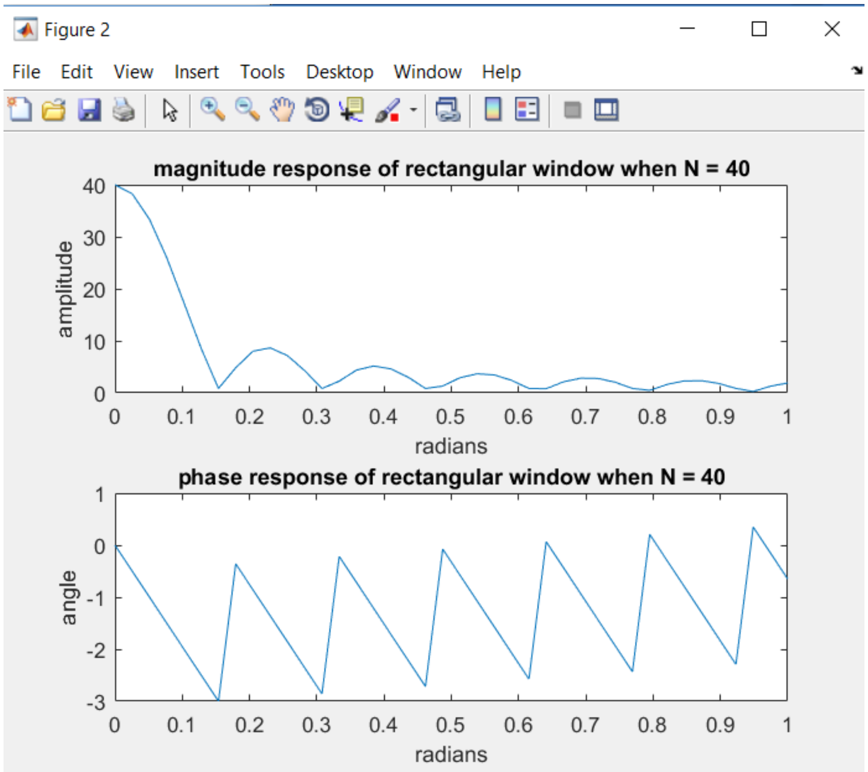


Figure 2 below is the magnitude and phase response of rectangular window when N = 40:



For N = 20, the width of the main lobe is around 0.3*2 = 0.6 radians.

For N = 40, the width of the main lobe is around 0.15*2 = 0.3 radians.

Report Item 8:

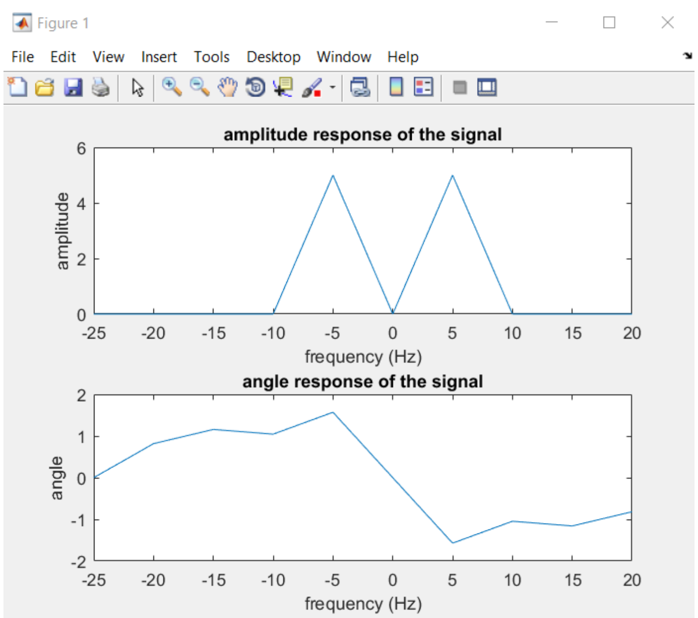
Code part:

```
%Report_Item_8
f0 = 5;
dt = 0.02;
T = 1/f0;
N = T/dt; %which is 10 in this situation

n = 0 : (N-1);
xn = sin(2*pi*f0*n*dt); %signal function
%get the response
fftx = fftshift(fft(xn));
w = fftshift((0:N-1)/N*2*pi);
w(1:N/2) = w(1:N/2)-2*pi;
%plotting
```

```
figure(1);
subplot(2, 1, 1);
plot(1/dt*w/2/pi, abs(fftx));
xlabel('frequency (Hz)');
ylabel('amplitude');
title('amplitude response of the signal');
subplot(2,1,2);
plot(1/dt*w/2/pi, angle(fftx));
xlabel('frequency (Hz)');
ylabel('angle');
title('angle response of the signal');
```

Explanation part:

The figure 1 below is the amplitude and phase response of the signal:



The minimum value of N to minimize the spectral leakage is $N = 1/f_0 /dt = 1/5/0.02 = 10$