DSP LAB 3 REPORT

Dekun Xie 210337152

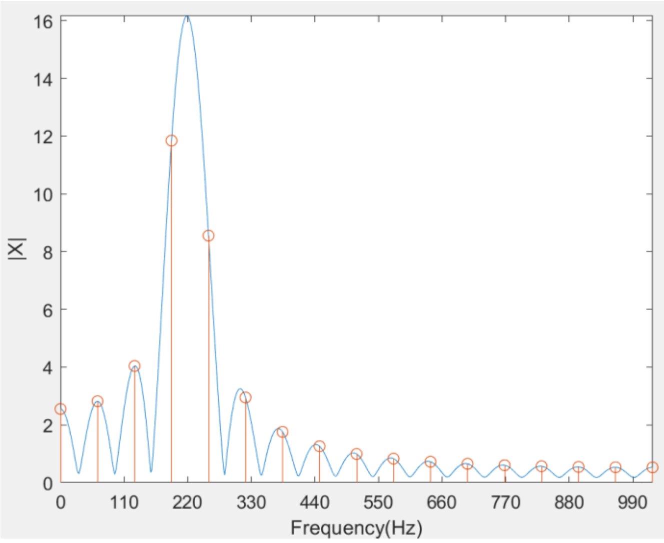
0.Introduction

In this lab, I will learn the spectral leakage and the function to solve this problem (zeros padding and windows). In addition, I will learn to plot the graph and the windows include Rectangular Window, Triangular Window, Sine Window, Hann Window, and the graph of the signal through the window. This lab will help me understand the window deeply.

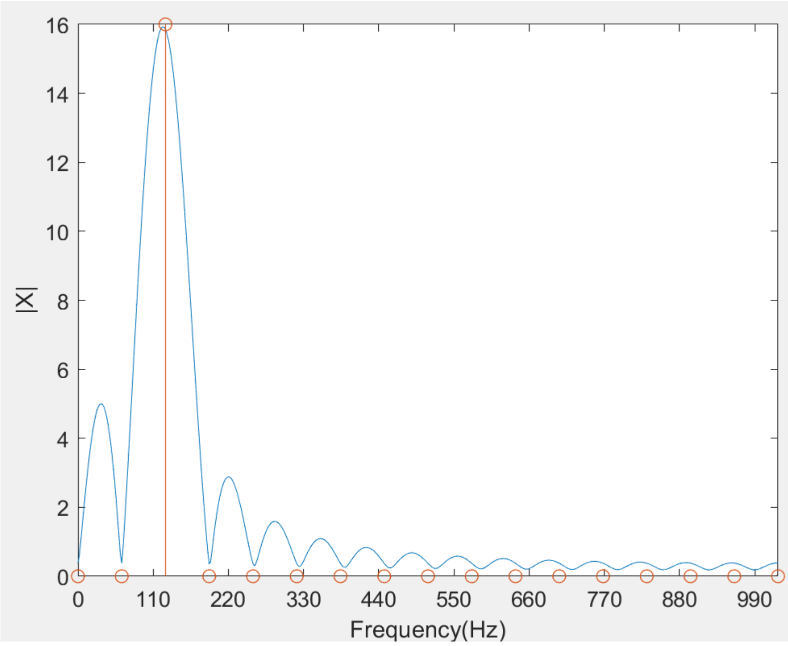
1.1 Using 32 samples of a sinusoid with frequency 220 Hz, sampled at a constant rate of 2,048 Hz, pad zeroes to it to a total length of 1024 samples. Calculate the DFT of the result. Plot this using a line plot, and overlay this with a stem plot of the DFT of the original 32 samples. See Figure 5 for guidance. Include your plot and code in your report.

```
f = 220;
fs = 2048;
dt = 1/fs;
N = 32;
M = 1024;
T = (0 : N-1) * dt;

x = sin(2* pi * f * T);

FFT_32 = fft(x);
magnitude_32 = abs(FFT_32(1: (N/2 + 1)));

```

```
x(N+1: M) = 0;
FFT_1024 = fft(x);
magnitude_1024 = abs(FFT_1024(1: (M/2 + 1)));

plot([0: M/2]*fs/M, magnitude_1024);
hold on
stem([0: N/2]*fs/N, magnitude_32);

axis tight;
xlabel('Frequency(Hz)')
ylabel('|X|')
set(gca, 'xtick', 0: 110: fs/2)
```



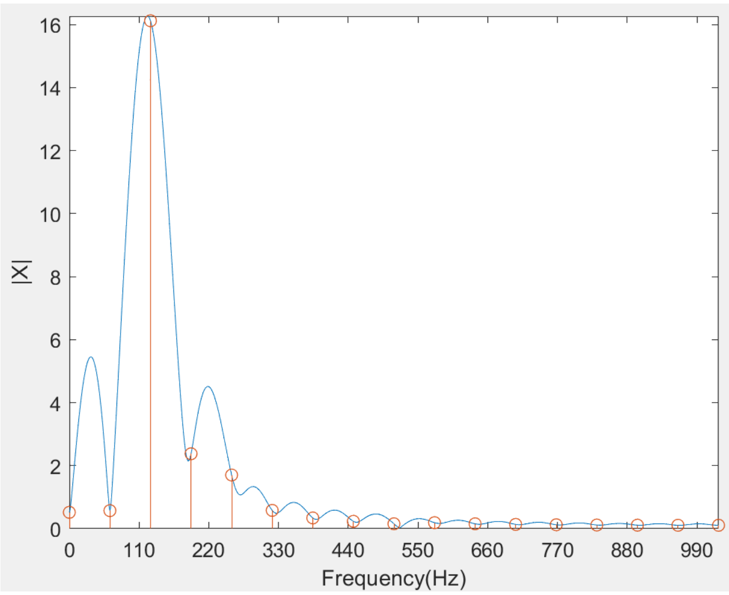
1.2 Using your code from 1.1, reduce the frequency of the sinusoid to 128 Hz and run the code again. Include only the plot and comment on what has happened.



There is no spectral leakage when sample number is 32 but there is a spectral leakage when we pad zeros to it to 1024.

1.3 Find the DFT of the signal zero-padded to a length 2048. Overlay the DFT spectrum of length-32 using stem, with the zero-padded DFT spectrum using plot, an example of which is given in Figure 5. Include just your plot.

$$x[n] = \sin(2\pi 128n/2048) + 0.2\sin(2\pi 220n/2048) + 0.01\cos(2\pi 525n/2048)$$



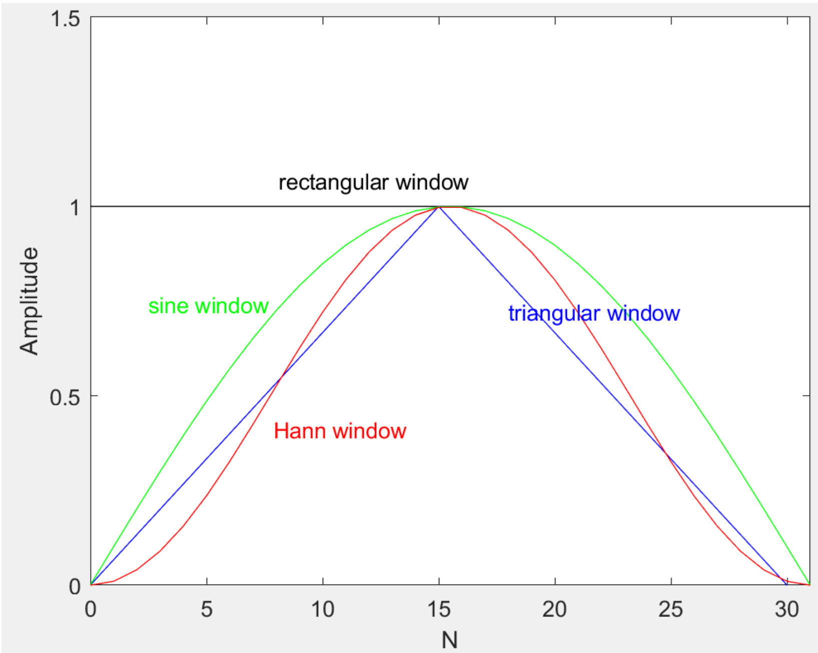
1.4 Without the knowledge that there are multiple pure tones in this sequence, can you pick out the right frequency components in the magnitude spectrum?

No, I can't. I can only recognize the frequency of 128 but ignore frequency of 220 and 525.

2.1 Program the four windows above. Using N=31 for the triangular window, and N=32 for the other windows, plot each window on the same graph (using plot instead of stem). Label each plot using gtext. Include this plot in your report.

```
N = 32;
N_tri = 31;

rectangular(1:N) = 1;

triangular = zeros(1,N_tri);
for n = 0 : N_tri -1
    triangular(n+1) = ((N_tri-1)/2 - abs(n-(N_tri-1)/2))*2/(N_tri-1);
end

sine = zeros(1,N);
for n = 0 : N-1
    sine(n+1) = sin(pi * n / (N-1));
end

Hann = zeros(1,N);
for n = 0 : N-1
    Hann(n+1) = 1/2 * (1-cos(2 * pi * n/(N-1)));
end

plot(0: N-1, rectangular,'black');
hold on;
plot(0:N_tri-1, triangular,'blue');
hold on;
```

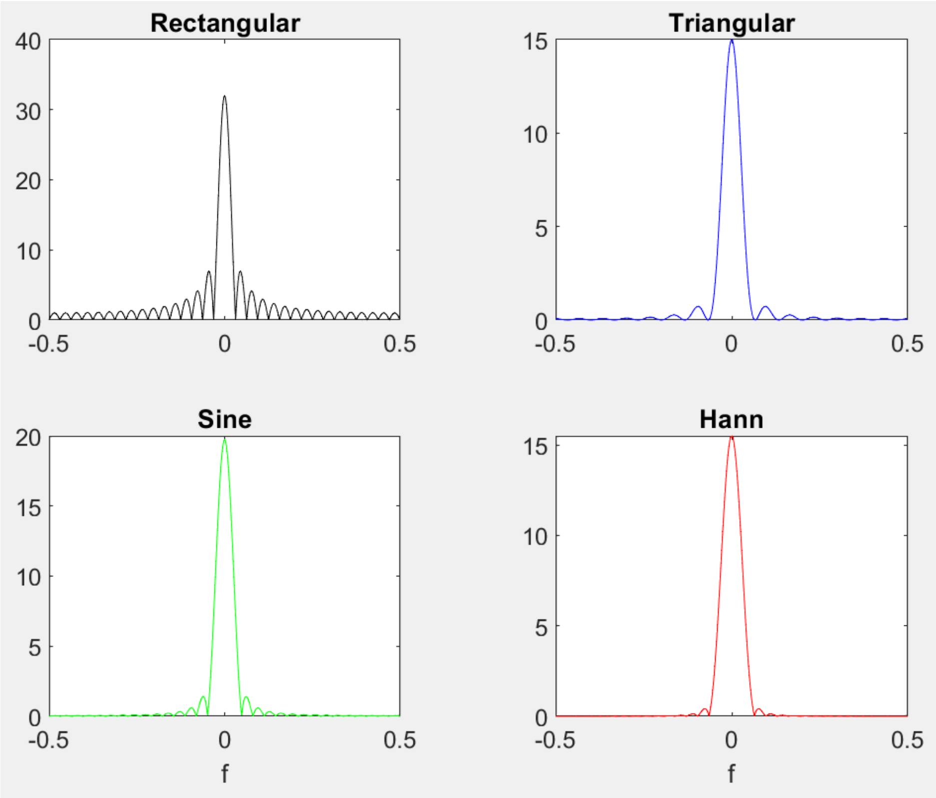
```
plot(0: N-1, sine,'green');
hold on;
plot(0: N-1, Hann,'red');

axis tight;
xlabel('N')
ylabel('Amplitude')
ylim([0,1.5]);

gtext('rectangular window','Color','black');
gtext('triangular window','Color','blue');
gtext('sine window','Color','green');
gtext('Hann window','Color','red');
```



2.2 Now evaluate the DFT of each window at 1,024 points (fft(w,1024)). In four different diagrams, plot the normalized magnitude spectrum (not dB) as a function of normalized frequency for each window, but before doing so you will want to use fftshift to move the zero frequency to the center. The normalized frequency interval is. Your figures should look like those in Figure 7. Include your figure in your report.

```matlab
N = 32;
N_tri = 31;
sample_number = 1024;

rectangular(1:N) = 1;

triangular = zeros(1,N_tri);
for n = 0 : N_tri -1
    triangular(n+1) = ((N_tri-1)/2 - abs(n-(N_tri-1)/2))*2/(N_tri-1);
end

sine = zeros(1,N);
for n = 0 : N-1
    sine(n+1) = sin(pi * n / (N-1));
end

Hann = zeros(1,N);
for n = 0 : N-1
    Hann(n+1) = 1/2 * (1-cos(2 * pi * n/(N-1)));
end

y_rec = fftshift(fft(rectangular,sample_number));
y_tri = fftshift(fft(triangular,sample_number));
y_sine = fftshift(fft(sine,sample_number));
y_Hann = fftshift(fft(Hann,sample_number));

t = tiledlayout(2,2);
ax1 = nexttile;
plot((-sample_number/2:sample_number/2 - 1)/sample_number, abs(y_rec),'black');
title('Rectangular')

ax2 = nexttile;
plot((-sample_number/2:sample_number/2 - 1)/sample_number, abs(y_tri),'blue');
title('Triangular')

ax3 = nexttile;
plot((-sample_number/2:sample_number/2 - 1)/sample_number, abs(y_sine),'green');
title('Sine')

ax4 = nexttile;
plot((-sample_number/2:sample_number/2 - 1)/sample_number, abs(y_Hann),'red');
title('Hann')

xlabel(ax3,'f')
```

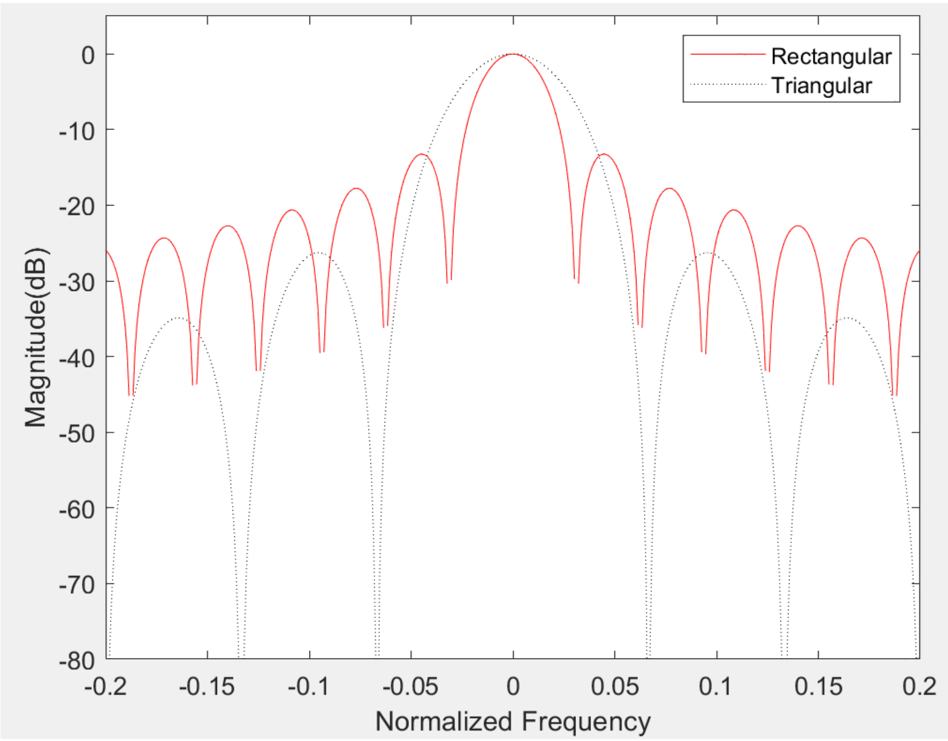
```
xlabel(ax4,'f')
```



2.3 For the windows in part 2.2, plot the normalized dB spectral function of the rectangular and triangular windows, eqs. (1)-(2), on the same graph. Limit the magnitude axis to [-80:5] dB, and the frequency axis to [-0.2:0.2]. Include your plot with a legend, as well as your code. Your figure should look similar to that in Figure 8.

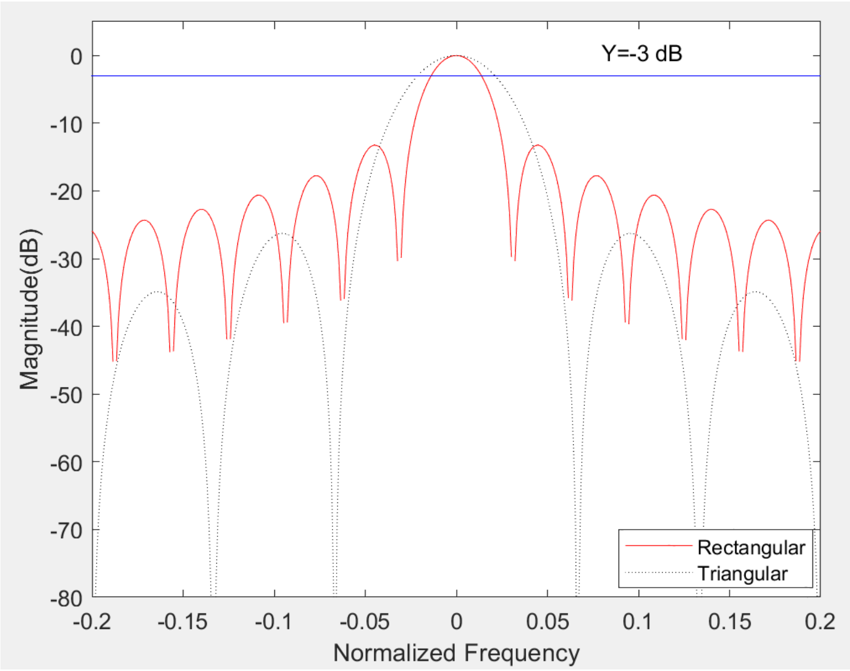
```
N = 32;
N_tri = 31;
sample_number = 1024;

rectangular(1:N) = 1;

triangular = zeros(1,N_tri);
for n = 0 : N_tri -1
    triangular(n+1) = ((N_tri-1)/2 - abs(n-(N_tri-1)/2))*2/(N_tri-1);
```

```
end

y_rec = abs(fftshift(fft(rectangular,sample_number)));
y_tri = abs(fftshift(fft(triangular,sample_number)));

mag_rec = 20*log10(y_rec / max(y_rec));
mag_tri = 20*log10(y_tri / max(y_tri));

plot((-sample_number/2:sample_number/2 - 1)/sample_number, mag_rec,'red');

hold on;
plot((-sample_number/2:sample_number/2 - 1)/sample_number, mag_tri,'black:');

legend('Rectangular','Triangular')
xlabel('Normalized Frequency');
ylabel('Magnitude(dB)')
axis([-0.2,0.2,-80,5])
```



2.4 Find the width of the main lobe for these two windows. What is the width of the main lobe for the triangular window in terms of the width for the rectangular window? Do you have any intuition why this is so?

The width of main lobe is the width when magnitude = -3 dB.



Find the point of intersection in the plot

$X_{Rectangular}$ = (-0.0137, -3) and (0.0137, -3)

$X_{Triangular}$ = (-0.0215, -3) and (0.0215, -3)

So,

Width$_{\textbf{Rectangular}}$ = 0.0137 * 2 = 0.0274

Width$_{\textbf{Triangular}}$ = 0.0215 * 2 = 0.0430

Intuition:

The width of triangular window is around double of the width of rectangular window.

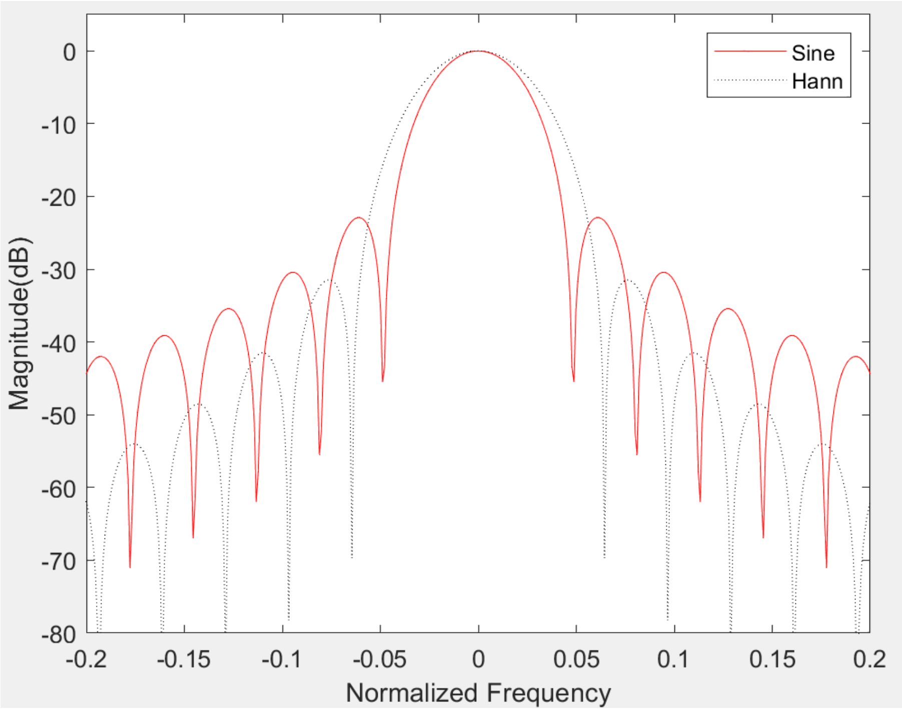
2.5 At what magnitude (in dB) is the height of the first side-lobe

for the rectangular and triangular windows?

-24.4 dB for rectangular window

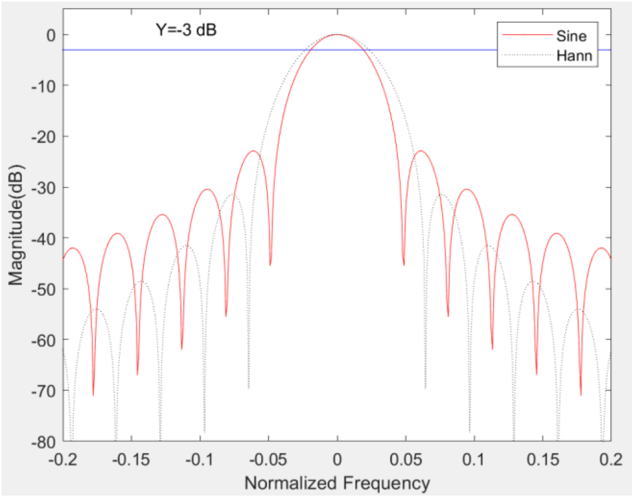
-34.9 dB for triangular window

2.6 Repeat 2.3 for the sine and Hann window functions, eqs. (3)-(4). Include your plot with a legend. No need to include your code.



2.7 What is the width of the main lobe for the sine and Hann windows?

The width of main lobe is the width when magnitude = -3 dB.



Find the point of intersection in the plot

X_{Sine} = (-0.0195, -3) and (0.0195, -3)

X_{Hann} = (-0.0234, -3) and (0.0234, -3)

So,

Width$_{Sine}$ = 0.0195 * 2 = 0.0390

Width$_{Hann}$ = 0.0234 * 2 = 0.0468

2.8 At what magnitude (in dB) is the height of the first side-lobe for the sine and Hann window?

-42.0 dB for Sine window

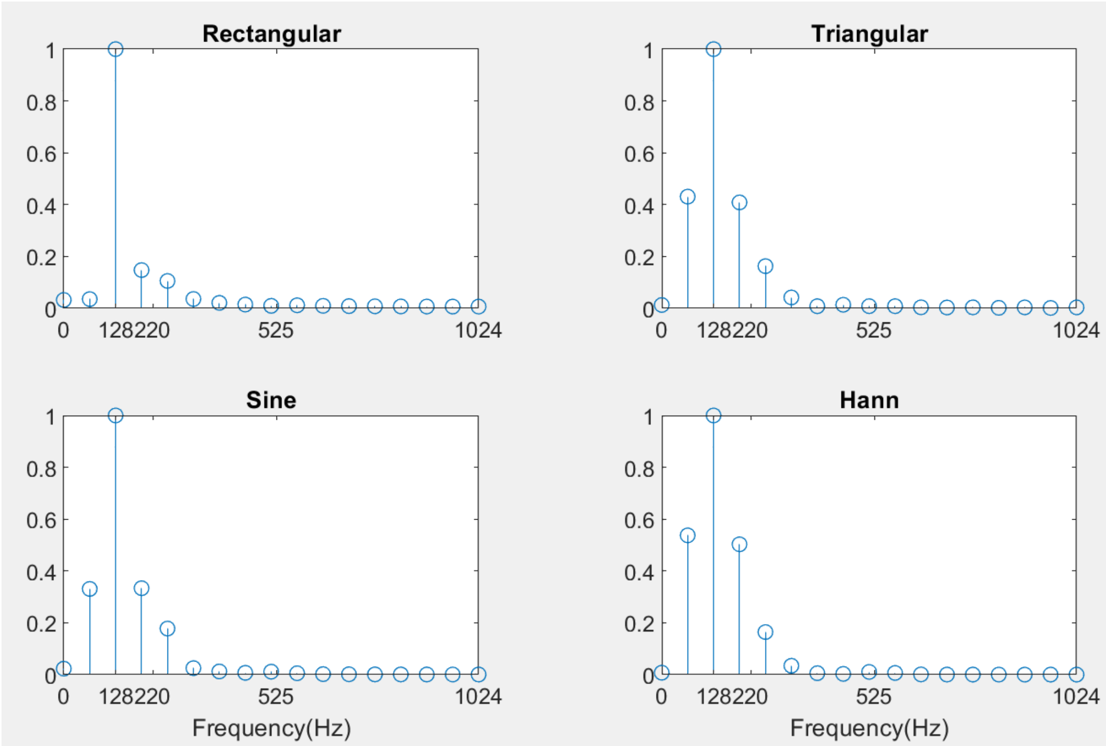
-54.0 dB for Hann window

3.1 Apply windowing to the 32-sample sequence you created in 1.3 using the rectangular, triangular, sine, and Hann windows. Plot each spectrum (four plots) using a stem plot (not in dB). Plot the normalized magnitude for each case (abs(X)./max(abs(X))).

However, this time do not use fftshift. Use the proper frequency indices, i.e., f = n*Fs/N. Plot only the frequencies between 0 and the Nyquist frequency. Include your code and plot. Your plot should look like Figure 9: The normalized magnitude DFT of the length-32 windowed sum of three sinusoids with frequencies 128, 220, and 525 Hz.

```
f1 = 128;
f2 = 220;
f3 = 525;
fs = 2048;
dt = 1/fs;
N = 32;
N_tri = 31;
T = (0 : N-1) * dt;

x = sin(2* pi * f1 * T) + 0.2*sin(2 * pi * f2 * T) + 0.01 * cos(2 * pi * f3 * T);

rectangular(1:N) = 1; %rectangular window

triangular = zeros(1,N_tri);%triangular window
for n = 0 : N_tri -1
    triangular(n+1) = ((N_tri-1)/2 - abs(n-(N_tri-1)/2))*2/(N_tri-1);
end
triangular = [triangular 0];%padding zeron to 32 samples

sine = zeros(1,N); %sine window
for n = 0 : N-1
    sine(n+1) = sin(pi * n / (N-1));
end

Hann = zeros(1,N); %Hann window
for n = 0 : N-1
    Hann(n+1) = 1/2 * (1-cos(2 * pi * n/(N-1)));
end

y_rec = fft(x .* rectangular);
y_tri = fft(x .* triangular);
y_sine = fft(x .* sine);
```

```
y_Hann = fft(x .* Hann);

mag_rec = abs(y_rec(1: (N/2 + 1)))/max(abs(y_rec(1: (N/2 + 1))));
mag_tri = abs(y_tri(1: (N/2 + 1)))/max(abs(y_tri(1: (N/2 + 1))));
mag_sine = abs(y_sine(1: (N/2 + 1)))/max(abs(y_sine(1: (N/2 + 1))));
mag_Hann = abs(y_Hann(1: (N/2 + 1)))/max(abs(y_Hann(1: (N/2 + 1))));

t = tiledlayout(2,2);
ax1 = nexttile();
stem([0: N/2]*fs/N, mag_rec);
title('Rectangular');
axis tight;
xticks([0 128 220 525 1024]);

ax2 = nexttile();
stem([0: N/2]*fs/N, mag_tri);
title('Triangular');
axis tight;
xticks([0 128 220 525 1024]);

ax3 = nexttile();
stem([0: N/2]*fs/N, mag_sine);
xlabel(ax3,'Frequency(Hz)');
title('Sine');
axis tight;
xticks([0 128 220 525 1024]);

ax4 = nexttile();
stem([0: N/2]*fs/N, mag_Hann);
xlabel(ax4,'Frequency(Hz)');
title('Hann');
axis tight;
xticks([0 128 220 525 1024]);
```

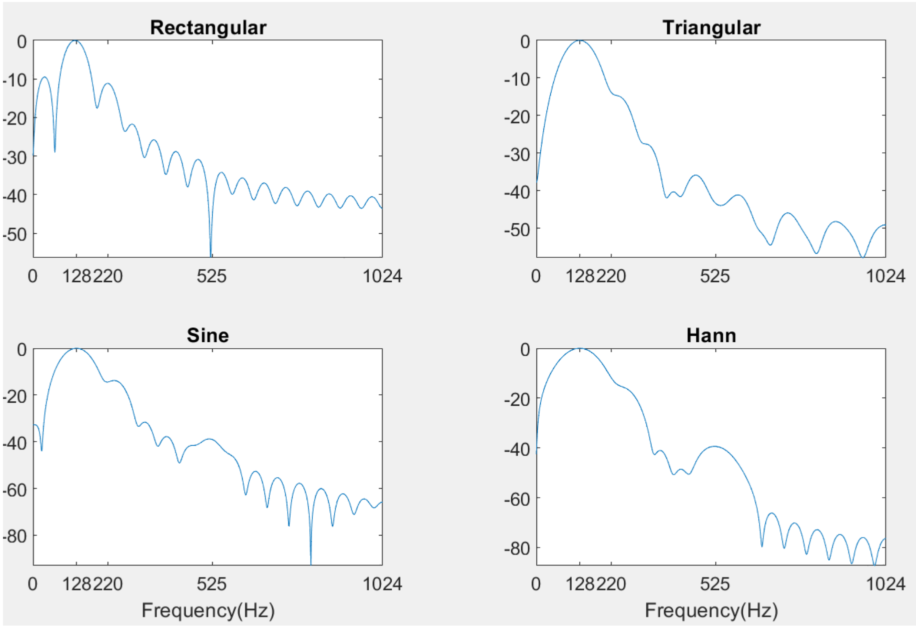


3.2 For each of the windowed signals in 3.1, zero-pad each sequence to a length of 2,048 and evaluate the DFT (you can do both at the same time by fft(x,2048)). Plot each magnitude spectrum (four plots) using plot on a normalized dB magnitude scale. Plot only the frequencies between 0 and the Nyquist frequency. Include your code and plot.

```
f1 = 128;
f2 = 220;
f3 = 525;
fs = 2048;
dt = 1/fs;
N = 32;
N_tri = N -1;
T = (0 : N-1) * dt;
M = 2048;

x = sin(2* pi * f1 * T) + 0.2*sin(2 * pi * f2 * T) + 0.01 * cos(2 * pi * f3 * T);

```

```
rectangular(1:N) = 1; %rectangular window

triangular = zeros(1,N_tri);%triangular window
for n = 0 : N_tri -1
    triangular(n+1) = ((N_tri-1)/2 - abs(n-(N_tri-1)/2))*2/(N_tri-1);
end
triangular = [triangular 0];

sine = zeros(1,N); %sine window
for n = 0 : N-1
    sine(n+1) = sin(pi * n / (N-1));
end

Hann = zeros(1,N); %Hann window
for n = 0 : N-1
    Hann(n+1) = 1/2 * (1-cos(2 * pi * n/(N-1)));
end

y_rec = fft(x .* rectangular, M);
y_tri = fft(x .* triangular, M);
y_sine = fft(x .* sine, M);
y_Hann = fft(x .* Hann, M);

mag_rec = 20*log10(abs(y_rec(1: (M/2 + 1)))/max(abs(y_rec(1: (M/2 + 1)))));
mag_tri = 20*log10(abs(y_tri(1: (M/2 + 1)))/max(abs(y_tri(1: (M/2 + 1)))));
mag_sine = 20*log10(abs(y_sine(1: (M/2 + 1)))/max(abs(y_sine(1: (M/2 + 1)))));
mag_Hann = 20*log10(abs(y_Hann(1: (M/2 + 1)))/max(abs(y_Hann(1: (M/2 + 1)))));

t = tiledlayout(2,2);
ax1 = nexttile();
plot([0: M/2]*fs/M, mag_rec);
title('Rectangular');
axis tight;
xticks([0 128 220 525 1024]);

ax2 = nexttile();
plot([0: M/2]*fs/M, mag_tri);
title('Triangular');
axis tight;
xticks([0 128 220 525 1024]);

ax3 = nexttile();
plot([0: M/2]*fs/M, mag_sine);
xlabel(ax3,'Frequency(Hz)');
```

```
title('Sine');
axis tight;
xticks([0 128 220 525 1024]);

ax4 = nexttile();
plot([0: M/2]*fs/M, mag_Hann);
xlabel(ax4,'Frequency(Hz)');
title('Hann');
axis tight;
xticks([0 128 220 525 1024]);
```



3.3 Answer the following questions:

3.3.1 Knowing how the main lobes depend on the type of window, and how the sidelobes decay, for which one(s) of these windows can you say with certainty that this signal has three components?

Sine and Hann windows according to the wide lobe.

3.3.2 For which window(s) can you accurately determine the frequency of the middle amplitude sinusoid?

Rectangular window

3.3.3 Why are the sine and Hann windows good at revealing the lowest amplitude component, but not the middle amplitude component?

As Hann and Sine windows get low and narrow side lobe which is good for lowest amplitude component, but get wide main lobe which is not good for middle amplitude one.

3.3.4 Why should the lowest amplitude component have a normalized dB level of –40 dB?

As the amplitude is 0.01. Decibel = 20log10(0.01) = -40 dB. Therefore, the lowest amplitude component have a normalized dB level of -40 dB.

4.Conclution

After finishing the lab, I know that spectral leakage is caused by unproperly sampling and it will always happen is the real world. One way to solve it is zero padding. It can increase the revolution of the plot while no information will be added. And another way is using window. From the lab, I know the certain shape of the windows and the feature of them. The Rectangular Window gets narrow lobes while others get wider ones. Furthermore, Sine and Hann Window can reveal the low amplitude component when Rectangular can reveal the middle one.