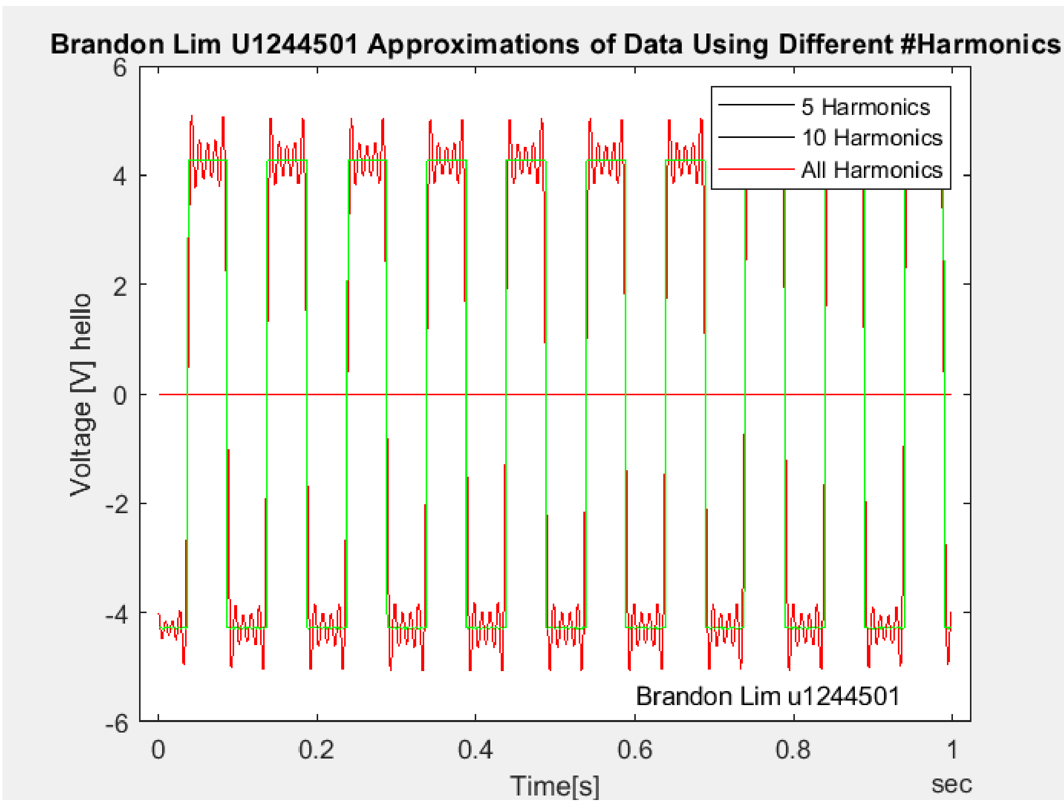
Brandon Lim
u1244501

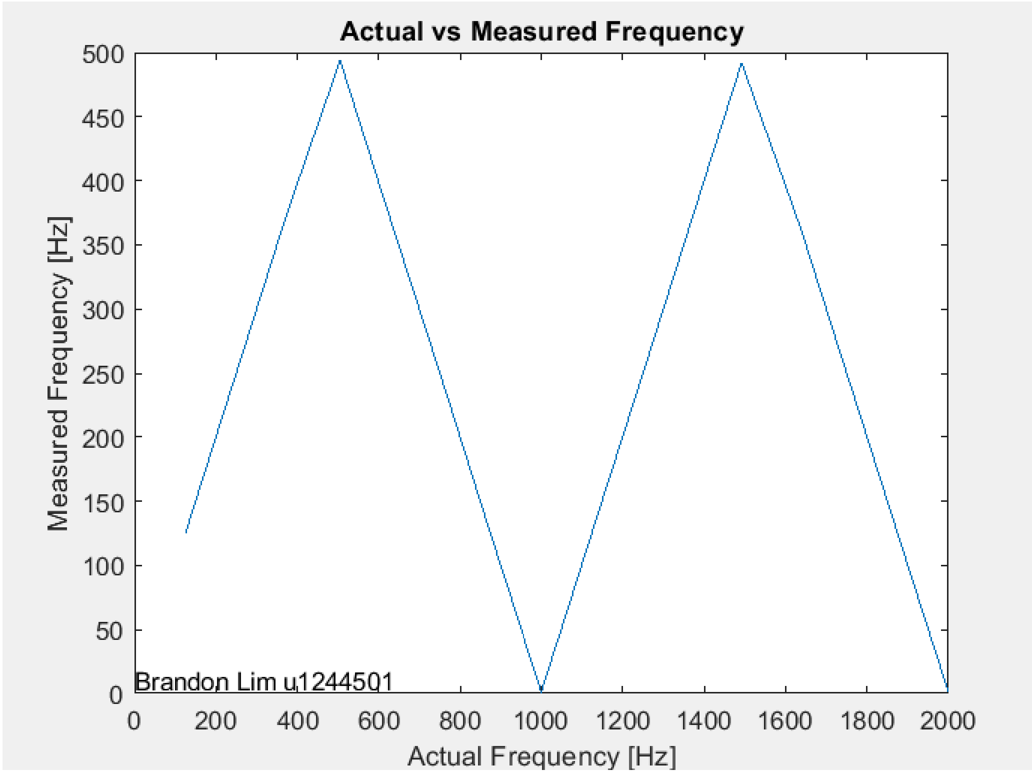
Post-Lab Exercises

1. Given your understanding of aliasing, how could you be sure to correctly determine the frequency of an unknown sinusoidal signal using digital sampling?
    a. Increase sampling frequency until measured frequency stops changing. When it stops changing we know that the frequency is no longer aliasing and the measured frequency is our actual frequency.
2. Using the square wave saved data from 4.1.5, plot the original signal (single channel) and plot the spectral analysis. From the spectral analysis you should see harmonic peaks. Using the ifft() function, reconstruct the signal with data up through 5 harmonics, then 10 harmonics, and then all harmonics. Plot the three reconstructed signals on the same axes. Include plots in the Post-lab submission.
   HINTS:
   • think about how the magnitude is calculated from the raw data.
   • Be careful of what data you include and what data you set to zero
   • If G = fft(V) then V = ifft(G)



3. Using the data from 4.3.7, create a plot of Measured Frequency (y-axis) vs Actual Frequency (x-axis). Attach plots to the Post-lab submission.

Brandon Lim
u1244501




```
clear, clc, close all
load('Lab3SquareWave.mat'); %Loading Data

subplot(2,1,1)
plot(timeData,rawData(:,1)) %Plotting the voltage vs the time
title("Voltage vs Time")
ylabel("Magnitude [V]")
xlabel("Time [s]")

sample_rate = 500; %initializing a sample rate
[ Mag, Phase, freq ] = fft_sample(rawData, sample_rate ); %doing a fast fourier transform to find magnitude,

subplot(2,1,2)
plot(freq,Mag) %Plotting freq vs magnitude for spectral analysis
title("Frequency vs Magnitude")
ylabel("Magnitude [Volts]")
xlabel("Frequency [Hz]")
text(150,5.5,"Brandon Lim u1244501")
```

Brandon Lim
u1244501

```
%%
clear, clc, close all
load('Lab3SquareWave.mat');
%Reconstructing with harmonics
G = fft(rawData); %Fast foward Transform of the data

%Only indexing first five harmonics of data
Gfive = G;
Gfive(100:end) = 0;

%Indexing first ten harmonics of data
Gten = G;
Gten(200:end) = 0;

%inversing the fast fourier transform of our individual harmonic data sets
%to get
yfiveH = ifft(Gfive);
ytenH = ifft(Gten);
y = ifft(G)
x = timeData

figure
plot(x,real(yfiveH) .* 2,"k")
hold on
plot(x,real(yfiveH) .* 2,"r")
hold on
plot(x,real(y),"g")
legend("5 Harmonics","10 Harmonics","All Harmonics")
title("Brandon Lim U1244501 Approximations of Data Using Different #Harmonics")
```


```
plot(x,real(yfiveH) .* 2,"k")
hold on
plot(x,real(yfiveH) .* 2,"r")
hold on
plot(x,real(y),"g")
legend("5 Harmonics","10 Harmonics","All Harmonics")
title("Brandon Lim U1244501 Approximations of Data Using Different #Harmonics")
```


```
xlabel("Time[s]")
ylabel("Voltage [V] hello")
text(0.6,-5.5,"Brandon Lim u1244501")
%%
clear, clc
Measured = [125 250 374 494 388 259 133 1 136 222 370 492 362 239 135 1];
Actual = [125 250 374 505 610 741 866 1000 1135 1222 1370 1492 1638 1761 1865 2000];
plot(Actual, Measured)
xlabel("Actual Frequency [Hz]")
ylabel("Measured Frequency [Hz]")
title("Actual vs Measured Frequency")
text(0,10,"Brandon Lim u1244501")
```