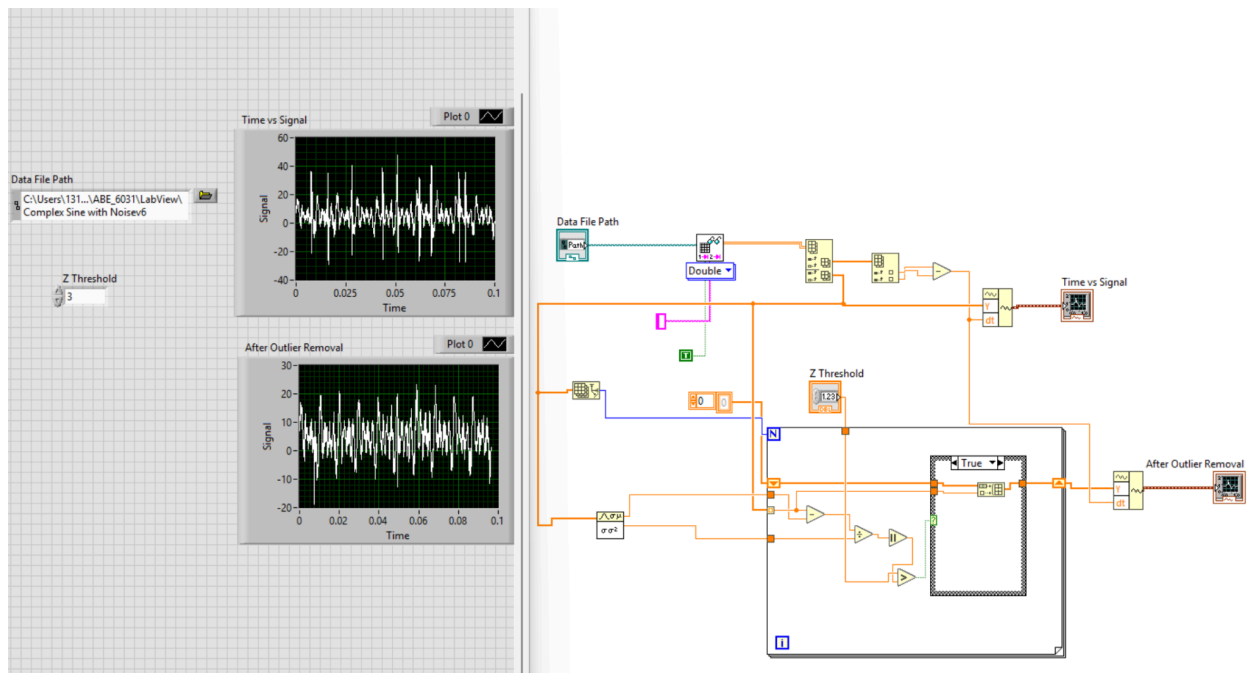
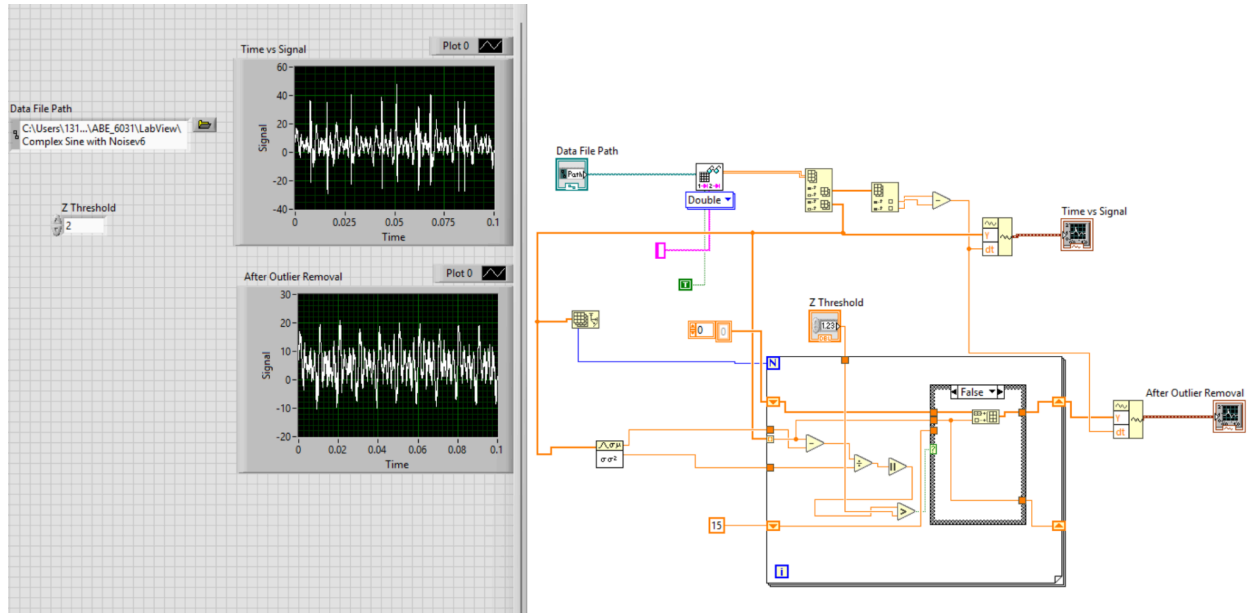


Take Home Final Exam

The figure displays a LabVIEW block diagram for outlier removal using Z-score. On the left, two plots are shown: 'Time vs Signal' and 'After Removal'. The 'Time vs Signal' plot shows a noisy signal with a range of approximately -40 to 60. The 'After Removal' plot shows the same signal with outliers removed, with a range of approximately -5 to 15. The block diagram on the right shows the signal being read from a file, converted to a double, and then processed by a Z-score calculation block. The Z-score is compared to a threshold (set to 3) to identify outliers. Outliers are removed from the signal, and the resulting signal is plotted.





For this step I sorted for outliers that were beyond a standard deviation of 2 sigmas for outliers, and then replaced said value with the previous value, otherwise keeping the value in the array. I tried deleting the value outright but that changed the distribution of the curve slightly, and replacing said value with the mean would create noise in the graph, and interpolating between the previous and next value could be problematic if the next value was also an outlier, so I chose to use just the previous value as a compromise.

Step 2: removing noise

▶

Configure Filter [Filter]

✕

Filtering Type

Lowpass

Filter Specifications

Cutoff Frequency (Hz)

400

High cutoff frequency (Hz)

400

○

Finite impulse response (FIR) filter

Taps

29

●

Infinite impulse response (IIR) filter

Topology

Butterworth

Order

3

Input Signal

Amplitude

Time

Result Preview

Amplitude

Time

View Mode

●

Signals

○

Transfer function

○ Show as spectrum

Scale Mode

○

Magnitude in dB

○

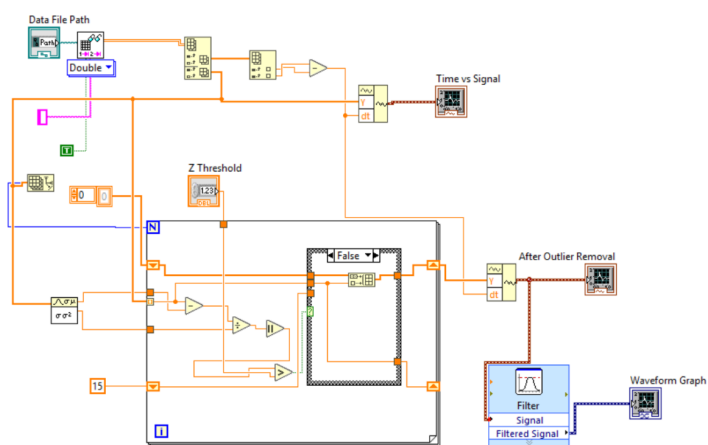
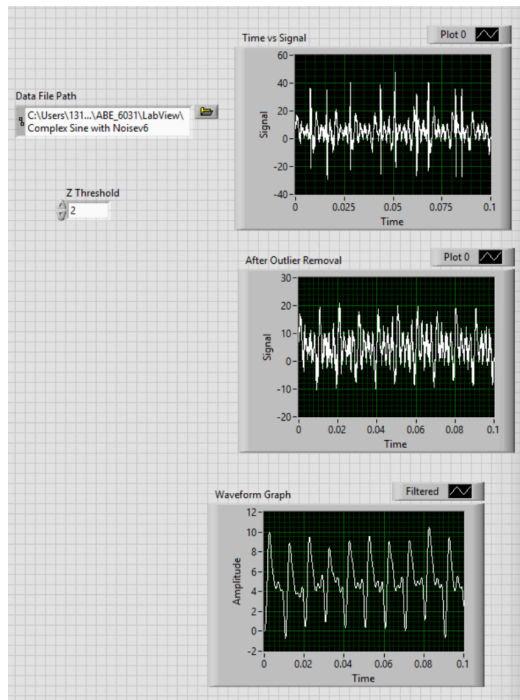
Frequency in log

OK

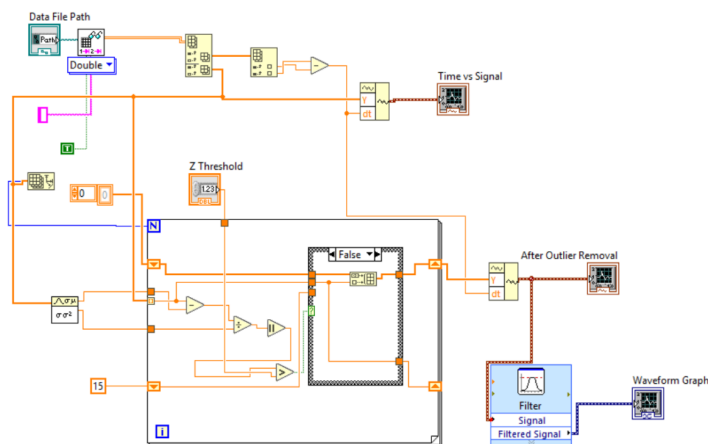
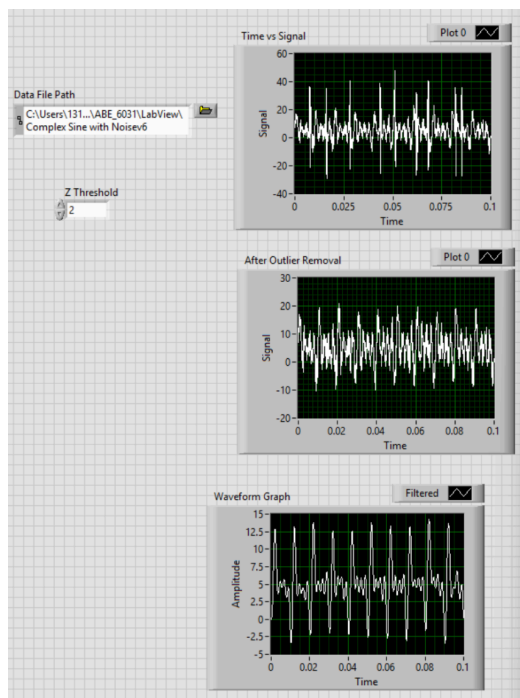
Cancel

Help

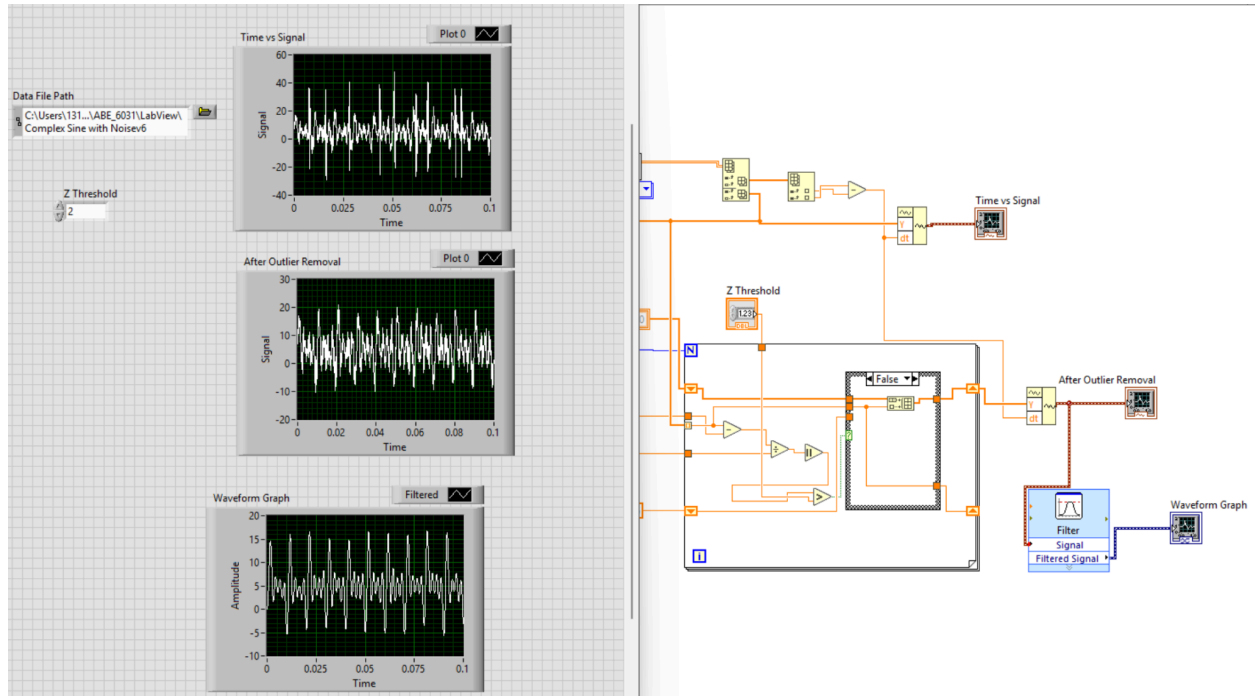
200 Hz:



300 Hz:

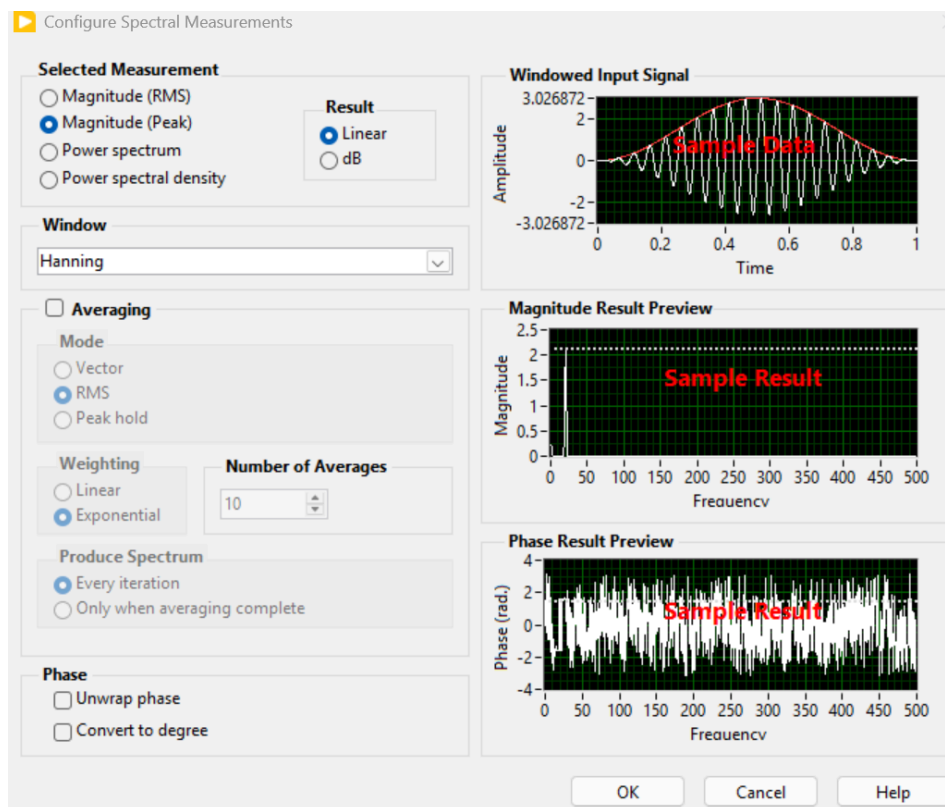
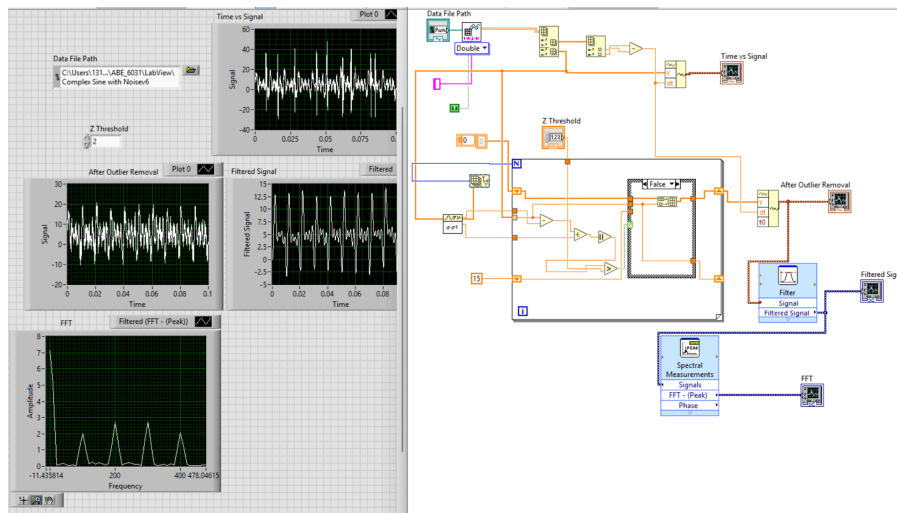


400Hz:



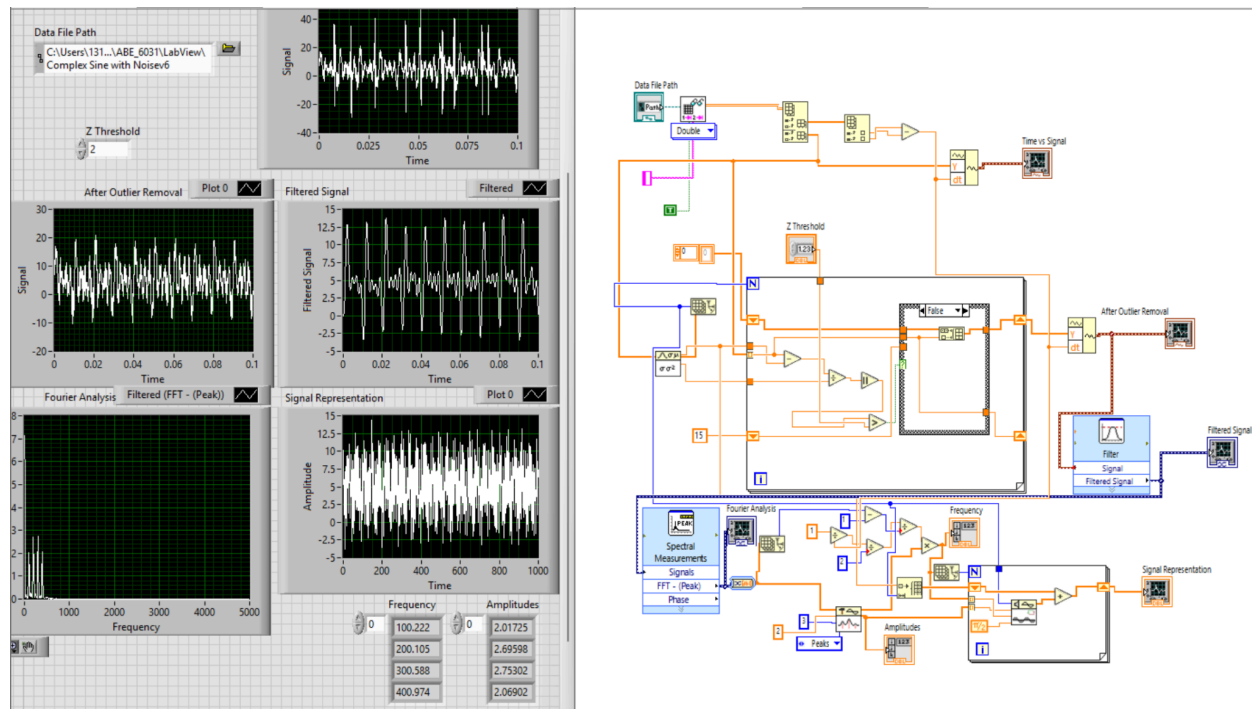
For step 2 I used a lowpass filter with the standard IIR filter and butterworth topology since the 10000Hz frequency usually means that the peaks are every 100-200Hz, so limiting below 500Hz is the best strategy since 500Hz+ is where most of the noise is. 400Hz seemed like the best strategy. I also tested out 300Hz but it didn't change things that significantly. 20Hz significantly altered the graph. This was a relatively simple step.

Step 3: Fourier Analysis



For step 3 I used the FFT Spectral Measurements to conduct the Fourier Analysis, using Peak Magnitude, Linear and Hanning Distance. One strange aberration I found was the DC spike at -11, but this is expected for the data. This was a relatively simple step.

Step 4:



Step 4 was relatively complicated. I used the Peak VI to extract the highest peaks in the fourier series, limiting to only peaks above 1.5 in amplitude (I could have also limited to 2 and gotten the same result) and bin width of 3, which gave the locations and the amplitudes. However, in order to get the exact frequency, I had to multiply all the values in the locations array to get the proper frequencies. $\Delta f \approx 10$ just for reference.

$Dt = .0001$ (passed in)

$F_s = 1/dt$

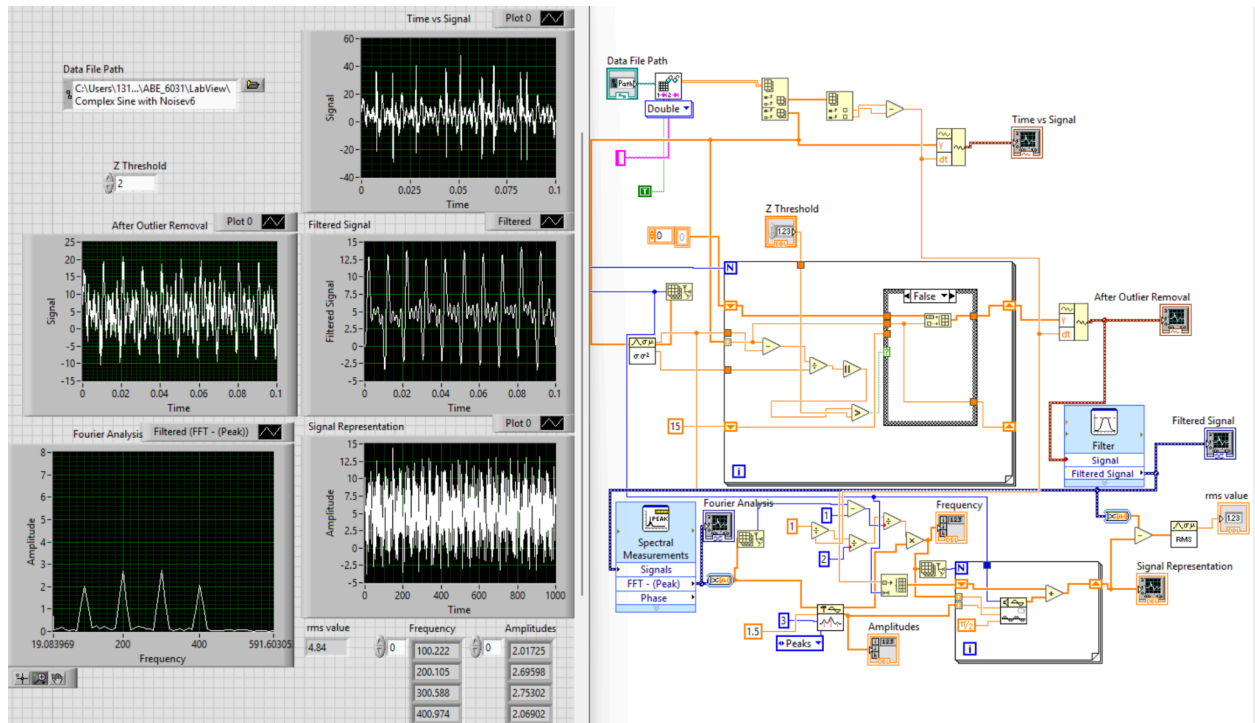
$f_{\text{max}} \approx F_s / 2$

$\Delta f = f_{\text{max}} / (\text{Array_Size} - 1)$

After this, I computed the summation $y_{\text{model}}(t) = \sum_{k=1}^4 A_k \cos(2\pi f_k t)$ using a for loop and a sine

wave offset by $\pi/2$ to get the resultant signal representation of the wave, which was rather inaccurate.

Step 5



For step 5, I subtracted the waveform result from the signal and then calculated the RMS value from said error to get the RMS error value or 4.84, which is pretty significant.