Washington University in St. Louis

James McKelvey School of Engineering

Fall 2020 MEMS 4050 Vibrations Laboratory

Lab 1: Periodic Signals, Data Acquisition, and Fourier Analysis

Lab Instructor: Philip Bayly

Experiment Date: Friday, September 25, 2020

Report Submission Date: Friday, October 9, 2020

Group T (Friday 2PM)

We hereby certify that the lab report herein is our original academic work, completed in accordance to the McKelvey School of Engineering and Undergraduate Student academic integrity policies, and submitted to fulfill the requirements of this assignment:

Matthew Donaldson	*Team Leader*
Aidan Murphy	*Test Engineer I*
Sam Wille	*Data Acquisition Manager*
Andrew Brown	*Quality Control Engineer*
Mitry Anderson	*Test Engineer II*

ABSTRACT *This experiment aimed to understand the method by which signals are sensed, processed, and translated into digital data. This goal was accomplished by measuring triangular, square, and sine wave signals from a Beckman function generator using a Quattro Data Acquisition System. The collected data was rendered and plotted using SignalCalc software. The effects of altering SignalCalc processing parameters on the digital data proved that if a given wave (sine, square, or triangle) has multiple frequencies, a large FSpan setting encapsulates more resonant frequencies. This also showed that a larger lines value provides a better, more defined, frequency spectrum plot resolution. For a sine wave input signal at frequencies varying from 10Hz to 100Hz at 10Hz increments, it was found that the frequency spectrum voltage peaks have a maximum percent error of 14.2% when compared to the multimeter voltage (converted from RMS to peak voltage by multiplying RMS voltage by* $\sqrt{2}$*). The same comparison of multimeter peak voltage to the average time series peak voltage was done and yielded a maximum percent error of 19%. However the average percent errors for these comparisons are 2.1% and 5.2%, respectively. The comparison between recorded frequencies and readings from the multimeter showed that the maximum percent error between the multimeter and the recorded value is 4%. All other measurements are around 0.2%. When comparing experimental SignalCalc frequency peak values to theoretically calculated Fourier Series frequency peak values the percent error was found to be a maximum of 8.17% and the minimum of 4.21%. This was an acceptable range because although it is more than 5% it is less than 10% error*

INTRODUCTION

The purpose of this experiment is two-fold: first, to understand how to create digital data from continuous signals by taking samples of the continuous signals at discrete increments and second, to understanding the frequency content and frequency spectra from the digital data [1]. The hardware used to sample the continuous signal is the Quattro Data Acquisition System (Quattro), which takes an input signal and passes it through a series of filters to smooth the output data. The Quattro uses a Fourier transform to model the continuous signal created from the function generator [2]. However for the purposes of this lab, since the function is known to be a periodic function, a Fourier series is used in hand calculations.

In a periodic function, such as square or triangle wave form, more than one frequency occurs. These frequencies happen at integer values of its base frequency. The Quattro uses the fast Fourier transform, sampling the voltage at multiple frequencies to construct a frequency vs voltage representation of the input function. These resonant frequencies can be modeled using a Fourier series to capture the periodic function as discrete components using Sine and Cosine terms or the complex form of equation 1 below:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n cos(n\omega_0 t) + \sum_{n=1}^{\infty} b_n sin(nw_0 t) = \sum_{n=1}^{\infty} A_n exp(in\omega_0 t) \tag{1}$$

where a_o, a_n, b_n, A_n are defined below as constant terms, ω_0 is the fundamental frequency of the function, n is the term in the series and t is the time variable [3].

$$a_0 = \frac{2}{T} \int_{-T/2}^{T/2} f(t)\, dt \tag{2}$$

$$a_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) Cos(nw_0 t)\, dt \tag{3}$$

$$b_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) Sin(nw_0 t)\, dt \tag{4}$$

$$A_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) exp(-in\omega_0 t)\, dt \tag{5}$$

It is important to note here that a_0 represents the fundamental frequency and a_n and b_n, A_n are integer multiples of the fundamental frequency. These terms can be seen in the frequency domain graph. As you use more terms to model the continuous function as discrete their is less error between the continuous function and its discrete form [3]. In this experiment the waveform can be modeled as an finite series of Sine and Cosine. It is important to remember that no matter how many terms are used to model the function there will always be a jump or error near a discontinuity in the signal. This is known as Gibbs Phenomenon and can be seen in Fig. 1.

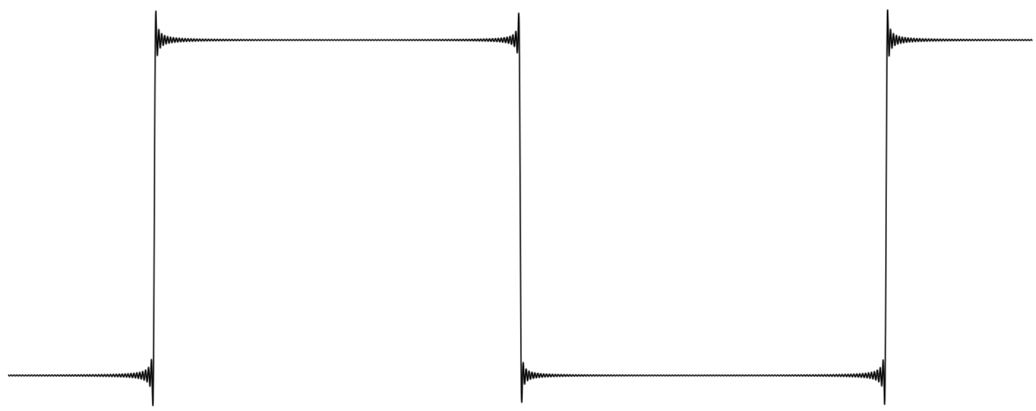


Figure 1 Shows Gibbs Phenomenon in the square wave. At the discontinuities the Fourier Series over jumps the square wave.[4]

Two parameters in SignalCalc help the quattro to convert the continuous function to a discrete form. They are the upper limit of the frequency range (Fspan) and the number of samples taken (Lines). Decreasing the Fspan cuts off the higher frequencies in the periodic signal [2]. This will limit the number of terms in the Fourier series because there will be fewer integer multiples of the fundamental frequency recorded. Also increasing the number of lines will allow for more samples to be taken in the domain.

It is important to realize here that when changing Fspan and Lines, it will change the nominal frequency resolution (dF) by,

$$dF = Fspan/Lines \tag{6}$$

From equation 6 dF is the difference between adjacent frequency points. So if Fspan is held constant and Lines are varied, dF is inversely proportional to Lines, so as the amount of Lines increases the difference between adjacent frequency points will decrease.[2] This would increase the resolution of the frequency domain [3]. Observe, dF is proportional to Fspan. If Fspan is varied and lines are held constant, there will be a decrease in the nominal frequency resolution as Fspan decreases. If Fspan increases, the opposite will be true. [2]

The change in dF results in a change in the time duration of the capture window (Tspan), with an inverse relation from the equation shown below,

$$Tspan = 1/dF \tag{7}$$

So as dF is increasing the time duration of the capture would decrease[3]. A smaller time frame would result in a lower resolution in the frequency domain. If equation 6 and equation 7 are combined to get ride of dF, it turns into equation 3 below:

$$Tspan = Lines/Fspan \tag{8}$$

From this a connection can be made between Lines and Tspan, as well as Fspan and Tspan. If Fspan is held constant then as the number of Lines increases then Tspan would also increase. Similarly, if Lines is held constant and Fspan is varied the time duration decreases.

It is also important to note here that the theoretical limit to avoid aliasing is the Nyquist value. To protect against this SignalCalc's limit is slightly below this at

$$Fspan = fs/2.56 \tag{9}$$

where fs is the sampling rate [3]. Anything above this would result in aliasing. However, the Quattro has an anti-aliasing filter, so there should be no aliasing frequencies in the measurements. SignalCalc also has a measurement range setting. If the range in the time or frequency domain is too small, clipping will occur, causing part of the signal and get cut off [2].

METHODS

Apparatus. Table 1 consists of the equipment used to generate and measure electrical signals. The function generator was the power source for the electrical signals and produced various AC functions depending on its settings. The function generator output its signal to the multimeter through a BNC cable attached to the Common and VΩ ports of the multimeter (Fig. 2). The multimeter measured the signal frequency and Root Mean Squared(RMS) voltage output by the function generator. The function generator also output to the Quattro via BNC cable (Fig. 2). In turn, the Quattro analyzed the multimeter signal in the time, frequency, and amplitude domains [2]. The SignalCalc software presented the Quattro analysis on the computer and controlled the Quattro's settings.

Table 1 Equipment Used to Generate and Measure Electrical Functions

Equipment	Make	Model	Serial #	Calibration Constant
Laptop	Lenovo	Thinkpad	N/A	N/A
SignalCalc 240 Dynamic Signal Analyzer	Data Physics Corporation	N/A	N/A	N/A
Quattro SignalCalc Ace	Data Physics Corporation	N/A	000002222	N/A
Multimeter	Fluke	223	N/A	N/A
Function Generator	Beckman	9010	525836	N/A

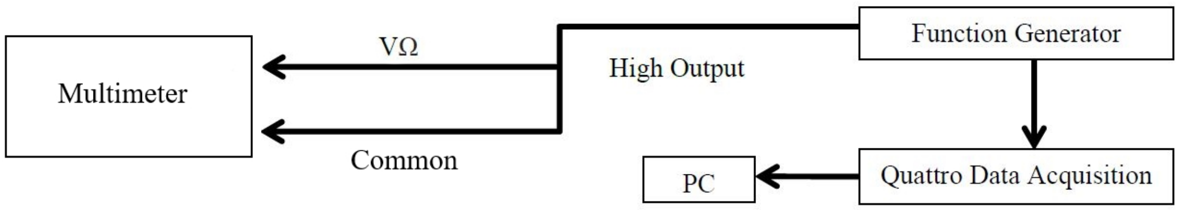


Figure 2 Set up of experimental apparatus [1, 5]

Procedure. To understand how the Quattro analyzed continuous voltage signals that were output by the function generator, three signals were analyzed: sine, triangle, an square waveforms. Each signal was analyzed at multiple frequencies and under multiple Quattro settings. However, the Quattro settings were always initialized to a Fspan of 80 Hz and lines equal to 200. All measurements were taken with "Trigger" set to "Free Run," "Auto End" to "1 Record," "Avg." to "Off," Range to 10 V, mV per EU to 1.000 k, and EU set to V. The Quattro analysis was saved for each run by the SignalCalc software. All functions at all frequencies had a RMS value of approximately 1 V. Before every run, the frequency output and RMS voltage were recorded from the multimeter readout.

First, a 25 Hz square wave was analyzed. The Fspan and Lines were varied from the initial settings across multiple runs until the effects of changing Fspan and Lines could be observed. The settings were initialized and then the same process was repeated with a 100 Hz sine wave until the effect of changing Fspan and Lines could be observed.

Next, the sine wave analyzed. The settings were initialized and a sine wave with frequency of 10 Hz was recorded. The Fspan and Lines were varied over multiple runs until the signal was captured with satisfactory resolution in both the time and frequency domains. This process was repeated, incrementing the frequency of the sine wave by 10 Hz until a 100 Hz wave had been analyzed.

Finally, analysis was performed on both a square wave and triangle wave for the frequencies 10 Hz, 15 Hz, 20 Hz, 35 Hz, 50 Hz. For each wave at each frequency, the settings were initialised and then the Fspan and lines were varied over multiple runs until the signal was captured with sufficient accuracy.

Analysis. After the data was collected, the peaks of the time series and frequency spectrum needed to be found for each trial. To this end, a MATLAB script was written to open the data file for a run, detect every peak above a certain threshold, record those peaks in an excel file (as well as the frequencies or times at which they occurred), and move on to the next trial. See Appendix D for the full code. In the frequency domain, the voltage data was in complex form, so in order to detect a real peak the real absolute value of the data was used. The program also had the option to generate graphs in both time and frequency domain for a particular run.

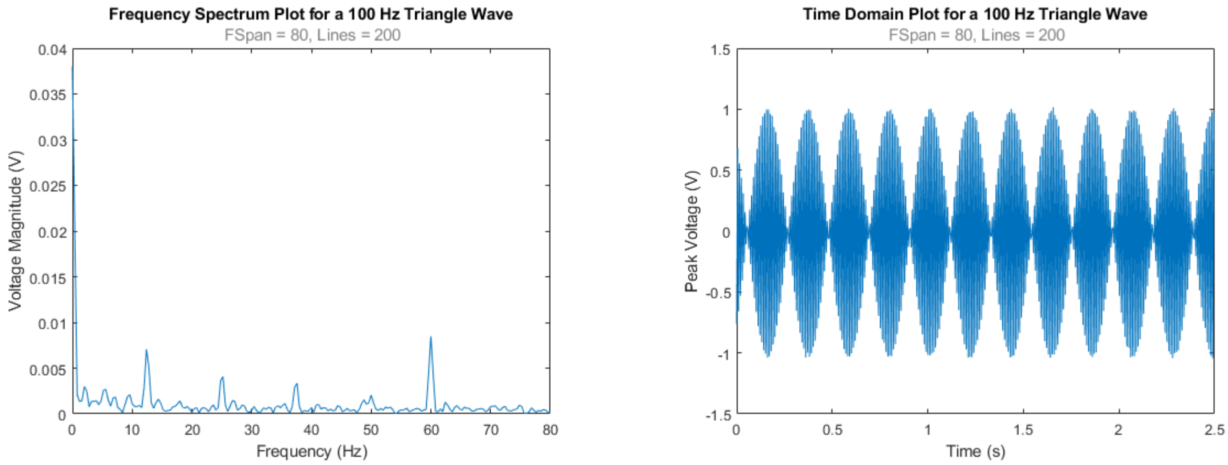
Then, Microsoft Excel was used to compare the measured peaks in frequency and voltage to the frequency and voltage recorded from the multi meter. This was achieved by simply subtracting the voltage peaks found from data from the voltage measured at the multi meter, and computing the percent error for each trial. The same process was used for the frequency peaks. The results of this analysis are discussed in the following section.

There are multiple sources of error that affect the accuracy of these measurements. First, the function generator was set using analog potentiometer knobs, so there is error in the exact alignment of the knobs. Next, the multimeter, Quattro, and signal calc itself are all not perfectly sensitive, so there is error stack up in the measuring devices. Also, the multimeter never fully settled on a single number, so all multimeter measurements are best approximations. For the voltage measurements, it is assumed that the wires offer no electrical resistance, however in reality they have a slight resistance. The same goes for the frequency measurements, as the system was assumed to have no impedance between the function generator and the Quattro, but in reality there would be a slight impedance.

RESULTS & DISCUSSION

The collected signals are concisely displayed below using plots generated using MATLAB and Excel from data collected and exported from SignalCalc.

Triangle Waves. The initial capture of the 100Hz triangle wave is performed with a Fspan of 80 and Lines of 200 as seen in Fig. 3. This provides a shared point for qualitative comparison between the waveforms observed in the experiment.



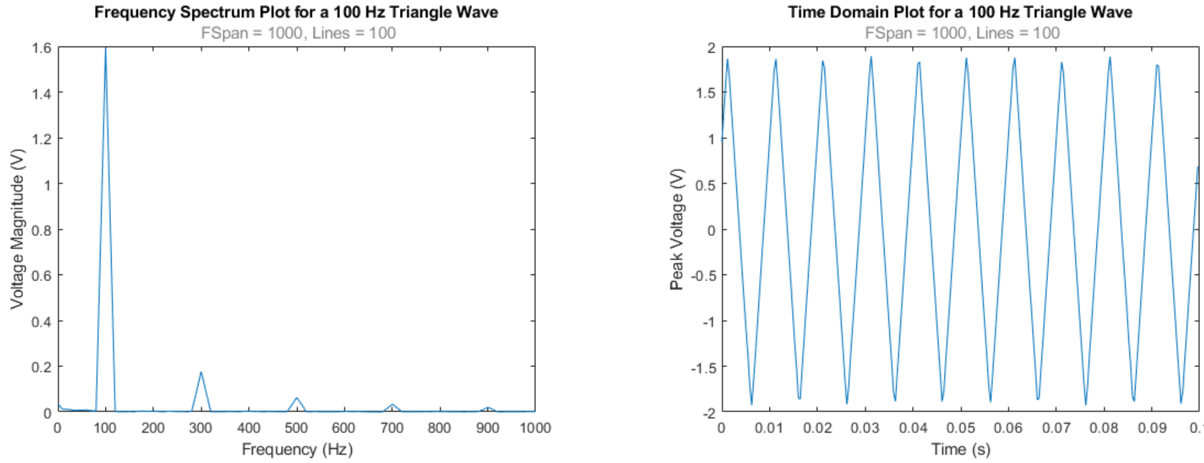
(a) Frequency spectrum plot. **(b) Time series plot.**

Figure 3 Frequency spectrum and time series plots for a 100 Hz triangle wave with SignalCalc settings of FSpan=80 and Lines=200. There figures are from the first lab of data collection.

The baseline test for the triangle wave form provides poor results in both frequency resolution and time series plot sufficient frequency data is not present, because the frequency is 100Hz, but Fspan is only 80Hz, so it will not measure the frequency supplied. In order to remedy this, the Fspan parameter is increased by a factor of 12.5 while the lines parameter is halved for Fig. 4 below.

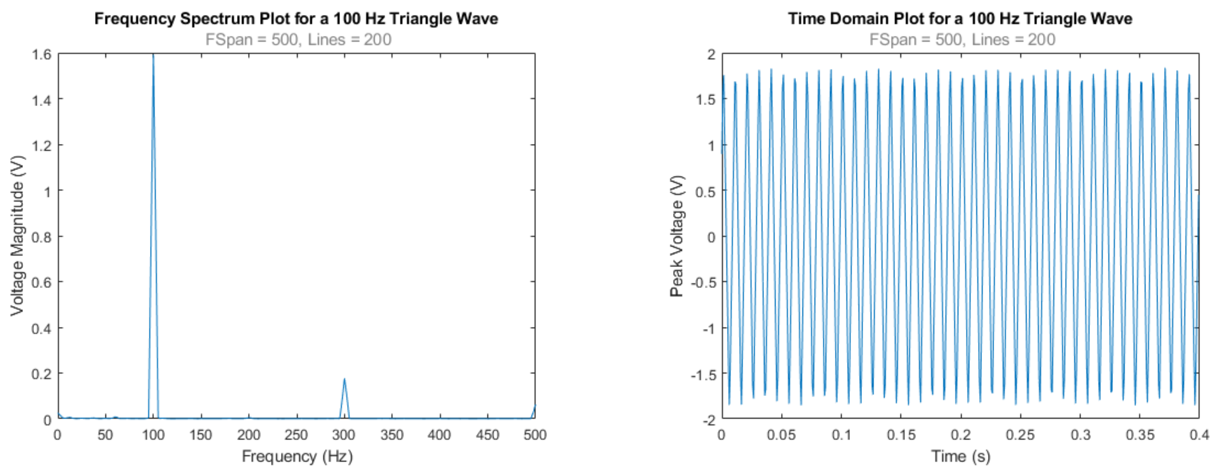
The harmonics occur every 200Hz, meaning they are odd harmonics. This is consistent with theoretical expectations. Additionally, they reduce the Tspan by a factor of 25, creating a much more clearly displayed triangle waveform on the time series plot. These changes should decrease frequency resolution by a factor of 25 based on equation 6. This expectation cannot be confirmed nor denied based on a single resolved spectrum.

Therefore, the waveform is analyzed with intermediate parameter settings, decreasing Fspan by a factor of two and increasing lines by the same. This overall increases the frequency resolution by a factor of four and increases the Tspan by a factor of four, as seen in Fig. 5. If the frequency resolution improves, then our expectations are met.



(a) Frequency spectrum plot. **(b) Time series plot.**

Figure 4 Frequency spectrum and time series plots for a 100 Hz triangle wave with SignalCalc settings of FSpan=1000 and Lines=100. There figures are from the first lab of data collection.

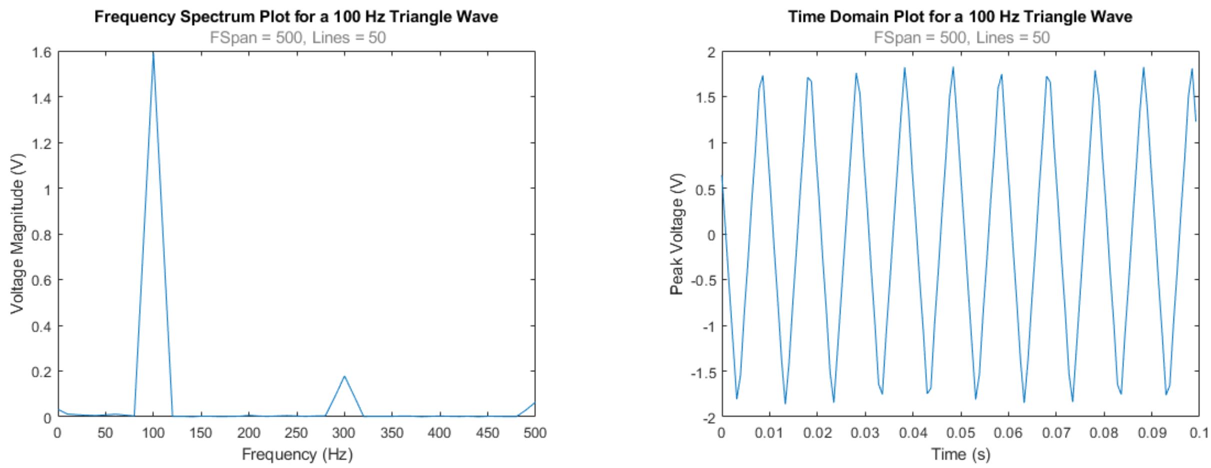


(a) Frequency spectrum plot. **(b) Time series plot.**

Figure 5 Frequency spectrum and time series plots for a 100 Hz triangle wave with SignalCalc settings of FSpan=500 and Lines=200. There figures are from the first lab of data collection.

The plots from this run clearly show a fundamental frequency at 100Hz, but due to lower Fspan, shows fewer harmonics. However, the frequency resolution is clearly higher as shown by the sharper peaks. Additionally, the increase in Tspan can be seen by the more crowded time series plot.

As a final experiment with triangle waves, the Fspan is maintained, but the lines parameter is lowered by a factor of four in Fig. 6.

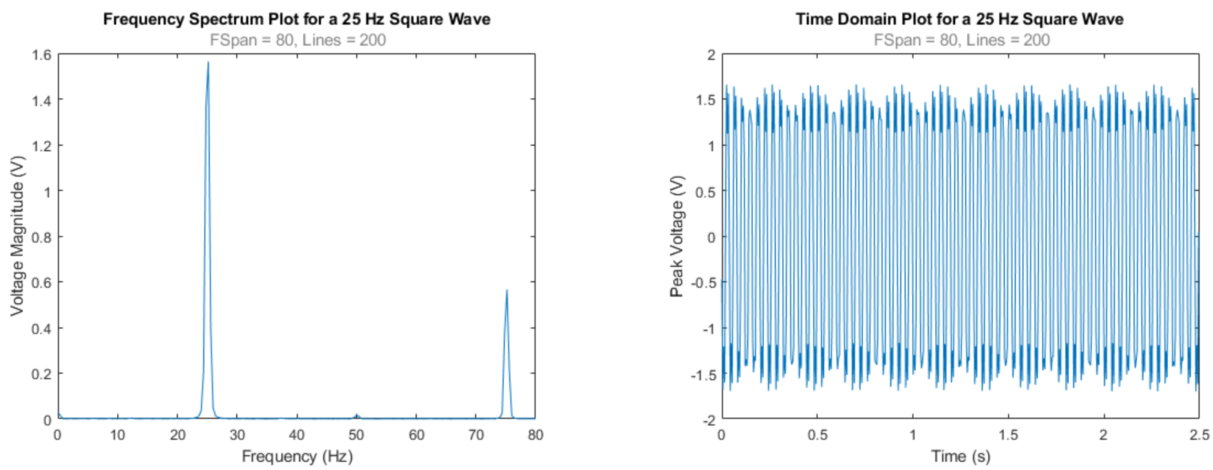


(a) Frequency spectrum plot. **(b) Time series plot.**

Figure 6 Frequency spectrum and time series plots for a 100 Hz triangle wave with SignalCalc settings of FSpan=500 and Lines=50. There figures are from the first lab of data collection.

This signal capture maintains a clear spectrum plot and has a clearly defined time series plot with easy to observe triangle waves.

Square Waves. The initial comparative test for the 25Hz square wave is performed at an Fspan of 80 and a Lines of 200 in Fig. 7 below.

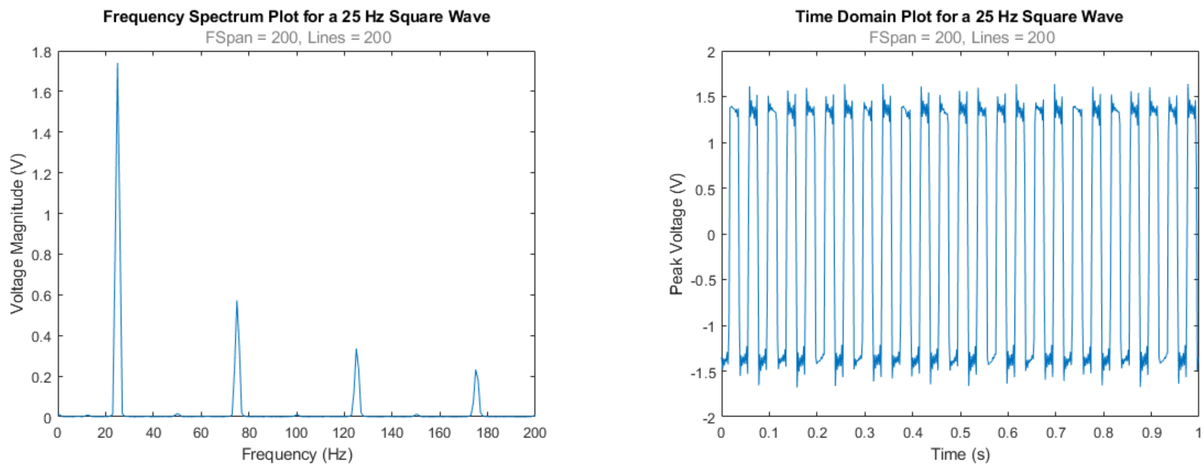


(a) Frequency spectrum plot. **(b) Time series plot.**

Figure 7 Frequency spectrum and time series plots for a 25 Hz square wave with SignalCalc settings of FSpan=80 and Lines=200. There figures are from the second lab of data collection.

As with the triangle wave, these settings produce an unclear time series plot. However, the frequency spectrum produces a clear fundamental frequency and harmonic. The harmonic occurs at 75Hz, three times the fundamental frequency, meaning it is an odd harmonic. This is consistent with theoretical expectations.

For the next test, the Fspan is increased by a factor of 2.5 and the Lines parameter is maintained. This means that the frequency range should capture more harmonics and the Tspan is decreased by a factor of 2.5, leading to a clearer time series plot in Fig. 8 below.



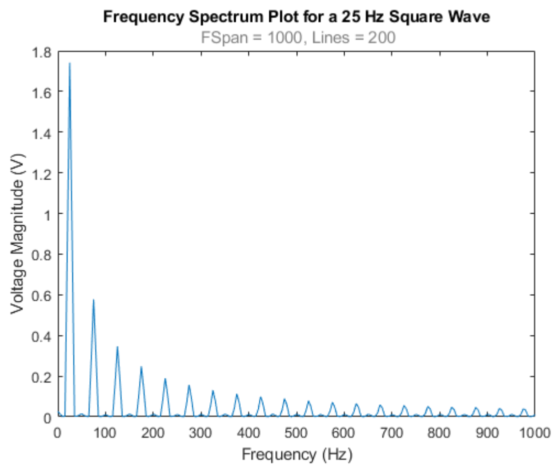
(a) Frequency spectrum plot. **(b) Time series plot.**

Figure 8 Frequency spectrum and time series plots for a 25 Hz square wave with SignalCalc settings of FSpan=200 and Lines=200. There figures are from the second lab of data collection.

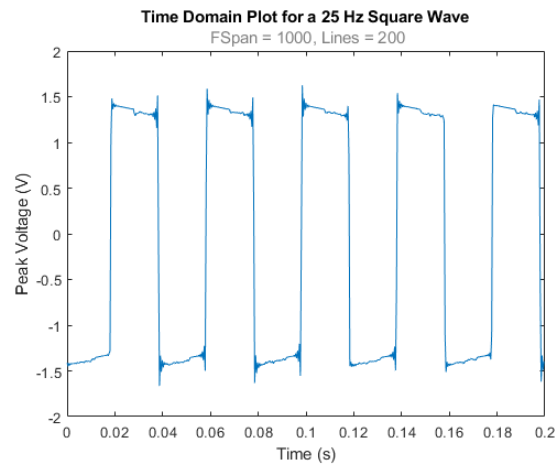
The theoretical expectations for the change are met in Fig. 8, but the clarity of the time series plot leaves much to be desired. Therefore, in Fig. 9, the Fspan is further increased by a factor of five, theoretically leading to many more harmonics and a much clearer square wave in the time series plot.

The change in the plots due to the altered Fspan meet expectations, and produce a particularly clear square wave with a clearly visible demonstration of the Gibbs Phenomenon.

Finally, the lines parameter was increased by a factor of four in Fig. 10 to improve the frequency resolution. This change was successful in creating higher resolution frequency peaks, but muddied the time series plot.

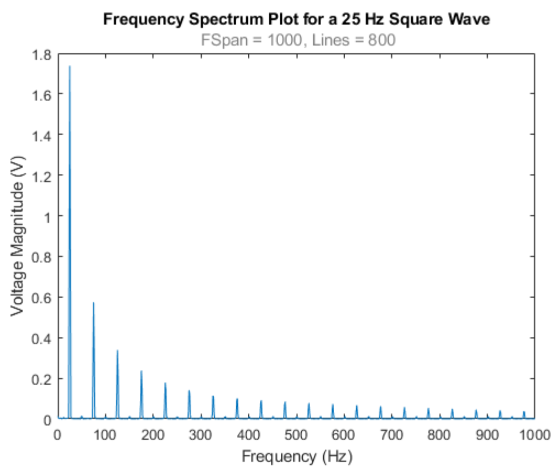


(a) Frequency spectrum plot.

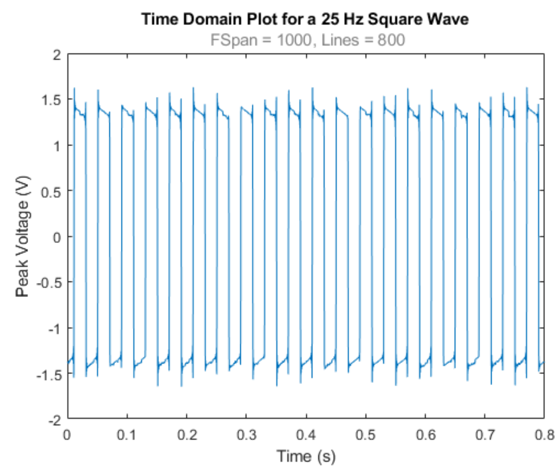


(b) Time series plot.

Figure 9 Frequency spectrum and time series plots for a 25 Hz square wave with SignalCalc settings of FSpan=1000 and Lines=200. There figures are from the second lab of data collection.



(a) Frequency spectrum plot.



(b) Time series plot.

Figure 10 Frequency spectrum and time series plots for a 25 Hz square wave with SignalCalc settings of FSpan=1000 and Lines=800. There figures are from the second lab of data collection.

Sine Waves. The table below has the measured and recorded voltage values from SignalCalc and the multimeter (Table 2). The voltage values from the frequency spectrum and time series plots are the peak voltages as measured in SignalCalc whereas the multimeter voltage is the RMS voltage of the sine wave. To compare these voltages, the multimeter RMS volatage was multiplied by $\sqrt{2}$ to convert RMS sine voltage to peak sine voltage, which is the voltage that SignalCalc measures. The run number relates to the output data run numbers that hold the data for the proper sine function at different frequencies. Table 3 relates the frequency spectrum peak and the corresponding multimeter readings to the run number. This allows for comparison of which frequency the sine wave was set to for a specific run number. Also notice that some runs are excluded from these plots. This is due to the FSpan SignalCalc setting being initially set to 80, causing the spectral range to stop at 80 Hz, cutting off the expected frequency spectrum peaks for the 80, 90, and 100 Hz sine waves (runs 38, 40, and 42 respectively).

Table 2 Comparison of recorded voltage peaks of a sine wave at varying frequencies from the frequency spectrum plot, the time series plot, and the multimeter reading. These values are from the first lab of data collection.

Run number	Frequency Spectrum Voltage Peak (V)	Time Series Average Peak Voltage (V)	Multimeter RMS Voltage (V)	Multimeter Converted Peak Voltage (V)
23	1.529	1.507	1.069	1.512
24	1.536	1.513	1.069	1.512
25	1.536	1.509	1.069	1.512
26	1.539	1.498	1.081	1.529
27	1.318	1.488	1.081	1.529
28	1.533	1.470	1.085	1.534
29	1.542	1.485	1.085	1.534
30	1.542	1.424	1.087	1.537
31	1.545	1.487	1.087	1.537
32	1.534	1.368	1.088	1.539
33	1.320	1.494	1.088	1.539
34	1.539	1.310	1.093	1.546
35	1.543	1.482	1.093	1.546
36	1.539	1.242	1.089	1.540
37	1.544	1.470	1.089	1.540
39	1.541	1.438	1.088	1.539
41	1.497	1.485	1.088	1.539
43	1.539	1.498	1.088	1.539

Table 3 Comparison of recorded frequency peaks of a sine wave at varying frequencies from the frequency spectrum plot and the multimeter reading. These values are from the first lab of data collection.

Run number	Frequency Spectrum Peaks (Hz)	Multimeter Frequency (Hz)
23	10	10.03
24	10	10.03
25	10	10.03
26	20	20.02
27	20.8	20.02
28	30	29.96
29	30	29.96
30	40	39.99
31	40	39.99
32	50	50.04
33	52	50.04
34	60	59.98
35	60	59.98
36	70	70.02
37	70	70.02
39	80	80.01
41	90.625	89.97
43	100	100.2

The frequency spectrum plot of a 35Hz square wave was analyzed to find the voltage peaks and their corresponding frequencies. These peaks are compared to the values predicted by hand calculating the constants of a Fourier series for a 35Hz square wave with the same amplitude seen on the voltmeter. The results of this analysis are shown in Table 4.

Table 4 Comparison of Experimental and Theoretical Frequency Peak Data from the second lab of data collection.

Run Number	Time Series Average Peak Voltage (V)	Frequency Spectrum Frequency Peaks (Hz)	Frequency Spectrum Voltage Peaks (V)	Fourier Series Frequency Spectrum Voltage (V)	Percent Error of Fourier to Frequency Spectrum Peaks (%)
72	1.425	35	1.773	1.851	4.21
72	1.425	105	0.583	0.617	5.49
72	1.425	175	0.347	0.370	6.17
72	1.425	245	0.249	0.264	5.85
72	1.425	315	0.194	0.206	5.55
72	1.425	385	0.157	0.168	6.70
72	1.425	455	0.131	0.142	8.17

CONCLUSION

This experiment changed the participants' perspective on the process of signal capture, and showed clearly that the theory behind input parameters is highly accurate to the digital data output. The analysis of triangle, square, and sine waves led to understanding of the effects of combinations of the Fspan and Lines parameters, as well as the relationship between periodic signals and their frequency spectra. More specifically, increasing lines with all else constant will give more precise results, due to a better resolution. Increasing Fspan with all else constant will allow measurement of a wider range of frequencies, but decrease the resolution. All of the waveforms were able to be captured without aliasing, with sufficient frequency resolution, and with clear time series plots. Additionally, the triangle and square waves were confirmed to have a finite number of odd harmonics, and the sine wave was confirmed to only consist of a fundamental frequency. Harmonic frequencies can accurately be estimated by assuming theoretical expectations are accurate. For example, a 100Hz triangular wave's first odd harmonic can be accurately assumed to be 300Hz.

Some of the results of the experiment were limited in accuracy due to the inability to find exact peak voltage values due to the prevalence of the Gibbs Phenomenon. The rapid oscillatory behavior of the signal at the corners of the square wave made determination of peak voltages difficult to discern. The experimental procedure could possibly be improved by the implementation of more complex signals, and the need to develop a filter to isolate the clean waveforms found above. This change would allow for deeper immersion and understanding of signal structure and the characteristics upon which to filter the data.

References

[1] 2020, "LAB 1: PERIODIC SIGNALS, DATA ACQUISITION AND FOURIER ANALYSIS," FL2020.E37.MEMS.4050.01 - Vibrations Lab.

[2] n.d., "SignalCalc Summary Guide," Tech. rep., Data Physics Corporation | Automotive Testing Technology International.

[3] L., M., 2001, *Fundamentals of Vibrations*, McGraw-Hill.

[4] 2020, "Functional approximation of square wave using 125 harmonics," Wikipedia.

[5] 2018, "How To Use A Klein Multimeter," Cool Circuit, accessed Oct. 4, 2020, https://www.coolcircuit.com/how-to-use-a-klein-multimeter/

[6] "Anti-Aliasing Filters and Their Usage Explained," National Instruments, Austin TX, updated March 14, 2019, accessed September 17, 2020, https://www.ni.com/en-us/innovations/white-papers/18/anti-aliasing-filters-and-their-usage-explained.html

[7] "Nyquist Frequency," Gatan, Pleasanton CA, accessed September 17, 2020, https://www.gatan.com/nyquist-frequency

[8] "RMS Voltage Tutorial," Electronics Tutorials, accessed September 17, 2020, https://www.electronics-tutorials.ws/accircuits/rms-voltage.html

[9] Esqueda, F., Bilbao, S., and Välimäki, V., 2016, "Aliasing Reduction in Clipped Signals," IEEE Transactions on Signal Processing, **64**(20), pp. 5255–5267.

[10] "Gibbs' Phenomenon," MIT OpenCourseWare, accessed Fall 2011, https://ocw.mit.edu/courses/mathematics/18-03sc-differential-equations-fall-2011/unit-iii-fourier-series-and-laplace-transform/operations-on-fourier-series/MIT18_03SCF11_s22_7text.pdf

A Glossary

- **Anti-aliasing filter:** A filter which analyzes input frequencies, and selectively excludes outlier frequencies above the Nyquist frequency. This maintains the integrity of measurements by excluding frequencies for which the sampling rate is less than twice the input frequency. [6].
- **Nyquist Frequency:** The Nyquist Frequency is half of the sampling rate. This frequency is the maximum input frequency for which a sampling rate can accurately capture a signal. [7].
- **Root Mean Square Voltage:** The Root Mean Square Voltage is the representation of the average magnitude of a sinusoidal signal. It represents the magnitude of an AC voltage as if it were DC, hence it is often referred to as the DC equivalent voltage. [8].
- **Clipping (in signal processing):** Clipping is a signal distortion process by which upper and lower bounds are implemented on an oscillating signal. Any values above or below those respective bounds are said to be the value of the limit. [9].
- **Gibbs phenomenon:** States that no matter the amount of terms that are included in the Fourier series for a signal, there will always be an error at or near a jump discontinuity in the signal in the form of an overshoot. This overshoot is always around 9% of the size of the jump. [10].

B Sample Calculations

The Fourier Series for a function f(t) is given in general form by the following equation, with the constants a_0, a_n, and b_n defined below.

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(n\omega_0 t) + \sum_{n=1}^{\infty} b_n \sin(n\omega_0 t)$$

$$a_0 = \frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)\,dt$$

$$a_n = \frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)\cos(n\omega_0 t)dt$$

$$b_n = \frac{2}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)\sin(n\omega_0 t)dt$$

In order to find the Fourier Series for the square wave used in the experiment, the following parameters were assumed:

$$f = 35.00\ Hz \quad A = 1.527V \quad T = \frac{1}{f} = 0.02857s \quad \omega_0 = \frac{2\pi}{T} = 219.9115\ rad/s$$

Figure B.1 shows one period of an ideal square wave as a function of time. The equation for this wave can be defined by the piece-wise function:

$$f(t) = \begin{cases} A & 0 < t < \frac{T}{2} \\ -A & -\frac{T}{2} < t < 0 \end{cases}$$

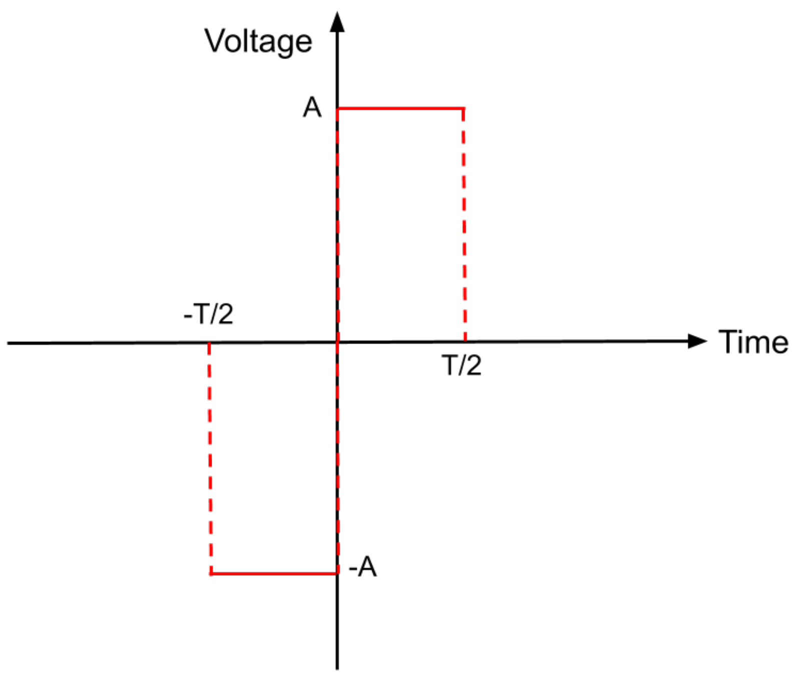


Figure B.1 The square wave shown above was used as a basis for the Fourier series derived below.

The constants a_0 and a_n both will equal zero for all values of n, because the square wave is an odd function, and the integral of an odd function is zero if the bounds of integration are equal and opposite. All that remains is to solve for b_n. Plugging in the expression for f(t) into the equation for b_n yields:

$$b_n = \frac{2}{T}\left[\int_0^{\frac{T}{2}} Asin\left(n\omega_0 t\right) dt + \int_{-\frac{T}{2}}^{0} -Asin\left(n\omega_0 t\right) dt\right]$$

This expression can be simplified as follows:

$$= \frac{2}{T}\left[-\frac{Acos\left(n\omega_0 t\right)}{n\omega_0}\Bigg|_0^{\frac{T}{2}} + \frac{Acos\left(n\omega_0 t\right)}{n\omega_0}\Bigg|_{-\frac{T}{2}}^{0}\right]$$

$$= \frac{2}{Tn\omega_0}\left[-Acos\left(\frac{n2\pi T}{2T}\right) + A + A - Acos\left(-\frac{n2\pi T}{2T}\right)\right]$$

$$\frac{A}{n\pi}\left[2 - \cos\left(\pi n\right) - \cos\left(-\pi n\right)\right]$$

$$b_n = \frac{A}{\pi}\begin{cases} \frac{4}{n} & n = 1, 3, 5, 7, \ldots \\ 0 & n = 2, 4, 6, 8, \ldots \end{cases}$$

With an expression for b_n, now a general expression for the Fourier series of f(t) can be found as follows:

$$f\left(t\right) = \sum_{n=1}^{\infty} \frac{A}{n\pi}\left[2 - \cos\left(\pi n\right) - \cos\left(-\pi n\right)\right] \sin(n\omega_0 t)$$

The first four coefficients can be calculated from the equation for b_n above, and were found to be:

$$b_1 = 1.8510 \quad b_3 = 0.6170 \quad b_5 = 0.3702 \quad b_7 = 0.2644$$

Plugging these into the general form gives the following four term approximation of a square wave:

$$f\left(t\right) = 1.8510\sin(219.6601t) + 0.6170\sin(658.9803t) + 0.3702\sin(1098.3005t) + 0.2644\sin(1537.62t)$$

The output of this function, as well as that of a 100 term approximation is shown in Fig B.2.

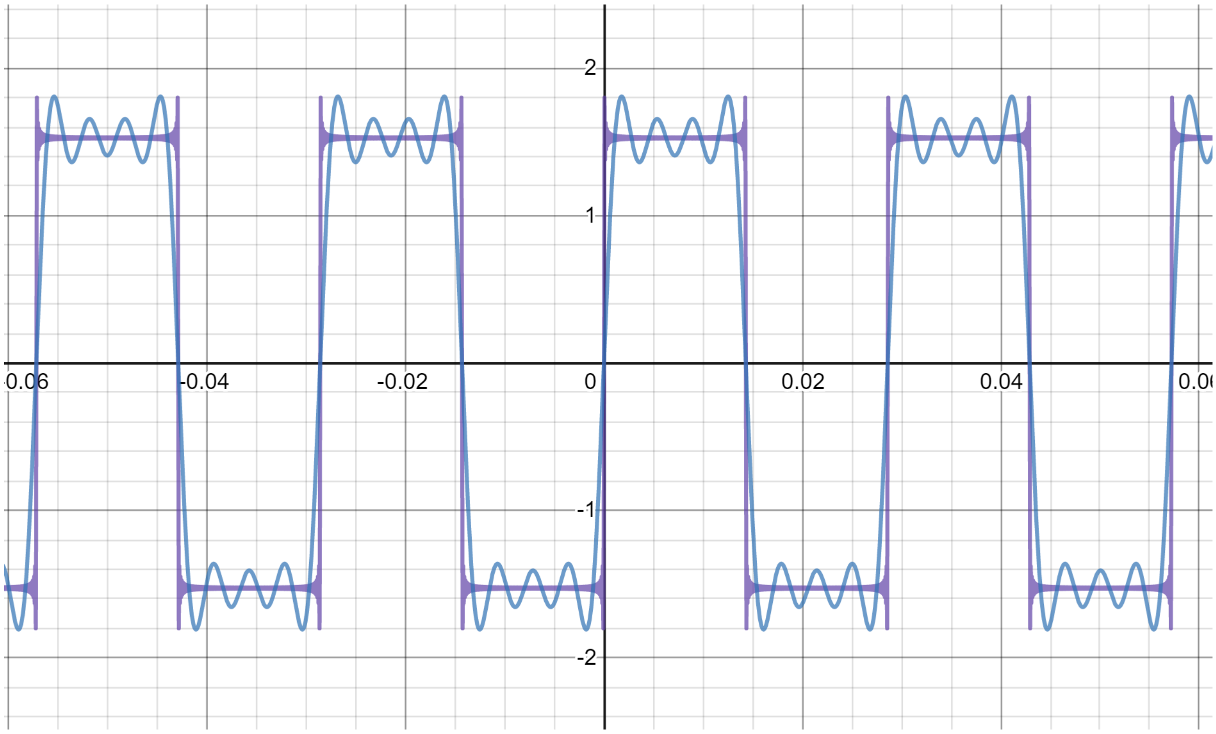


Figure B.2 The 100 term approximation is shown in purple, and the 4 term approximation is shown in blue. Note the Gibbs phenomenon occurring in both graphs near the jumps.

C Raw Data

In Table C.1 below, the frequency spectrum voltage peaks (and the frequencies at which they occur) as calculated in the MATLAB code are shown for a square wave and a triangle wave, with various sampling parameters. The data from trials with 80 for Fspan and 200 for Lines were done with the settings specified in the procedure, while the other trials were done with more optimized settings.

Table C.1 Sample Data for Various Waveforms, with Different Sampling Parameters

Wave Input	Frequency Spectrum Voltage Peak (V)	Frequency Spectrum Peaks (Hz)	Fspan	Lines
10 Hz Square	1.7501	10	80	200
10 Hz Square	0.5602	30	80	200
10 Hz Square	0.3145	50	80	200
10 Hz Square	0.2251	70.4	80	200
25 Hz Square	1.7401	25	200	200
25 Hz Square	0.5724	75	200	200
25 Hz Square	0.3360	125	200	200
25 Hz Square	0.2329	175	200	200
10 Hz Triangle	1.4349	10	80	200
10 Hz Triangle	0.1545	30	80	200
10 Hz Triangle	0.0506	50	80	200
10 Hz Triangle	0.0263	70.4	80	200
100Hz Triangle	1.4205	100	1000	800
100Hz Triangle	0.1534	300	1000	800
100Hz Triangle	0.0524	500	1000	800
100Hz Triangle	0.0249	700	1000	800
100Hz Triangle	0.0158	901.25	1000	800

Table C.2 below depicts the raw data from the exported MATLAB file for 10 sine waves at frequencies ranging from 10Hz to 100Hz at increments of 10Hz. The exported MATLAB data for the voltage and frequency peaks from the frequency spectrum are compared to the raw multimeter readout of the RMS voltage and corresponding frequency. For each of the 10 sine waves, data is tabulated for the the specified procedural settings of 80 FSpan and 200 Lines, and altered settings of 1000 FSpan and varying Lines (either 800 or 1600).

Table C.2 Frequency Spectrum Data for 10 Different Frequency Sine Waves, with Different Sampling Parameters and their corresponding multimeter readouts

Wave Input	Frequency Spectrum Voltage Peak (V)	Frequency Spectrum Peaks (Hz)	Multimeter RMS Voltage (V)	Multimeter Frequency (Hz)	Fspan	Lines
10 Hz Sine	1.3787	10.0	0.954	10.04	80	200
20 Hz Sine	1.3739	20.0	0.964	20.07	80	200
30 Hz Sine	1.3816	30.0	0.973	29.98	80	200
40 Hz Sine	1.3834	40.0	0.975	40.01	80	200
50 Hz Sine	1.3890	50.0	0.976	50.02	80	200
60 Hz Sine	1.3936	60.0	0.981	60.01	80	200
70 Hz Sine	1.3925	70.0	0.981	69.99	80	200
80 Hz Sine	1.3906	80.0	0.983	79.98	80	200
90 Hz Sine	N/A	N/A	0.982	90.0	80	200
100 Hz Sine	N/A	N/A	0.982	100.0	80	200
10 Hz Sine	1.3791	10.0	0.954	10.04	1000	800
20 Hz Sine	1.3769	20.0	0.964	20.07	1000	800
30 Hz Sine	1.3850	30.0	0.973	29.98	1000	800
40 Hz Sine	1.3839	40.0	0.975	40.01	1000	800
50 Hz Sine	1.3891	50.0	0.976	50.02	1000	800
60 Hz Sine	1.3931	60.0	0.981	60.01	1000	1600
70 Hz Sine	1.3951	70.0	0.981	69.99	1000	1600
80 Hz Sine	1.3932	80.0	0.983	79.98	1000	1600
90 Hz Sine	1.3933	90.0	0.982	90.0	1000	1600
100 Hz Sine	1.3913	100.0	0.982	100.0	1000	1600

D MATLAB Code

The following MATLAB script was used to load and read in each run file from a local file path. For each loaded MATLAB file, the corresponding run numbers, calculated voltage peaks and frequency peaks from the frequency spectrum data, and calculated average voltage peaks from the times series plot were exported to an excel file.

```
%%
clear all;
clc;
close all;

%%
% mainFilepath = 'D:\WashU\Classes\Fall 2020\Vibrations LAB\Lab 1\Run Files';
mainFilepath = 'D:\WashU\Classes\Fall 2020\Vibrations LAB\Lab 1\Run ...
    Files\second run files';

% excellFilename = '/first_runs.xlsx';
excellFilename = '/first_runs.xlsx';


cd(mainFilepath)
A = {'Run number','S-plot peak (V)','S-plot frequency(Hz)', ...
     'X-plot avg peak (V)'};
xlswrite(strcat(mainFilepath,excellFilename),A)

numFolders = 72;
% numFolders = 73;
height = 2;

for currentRun = 1:numFolders
    localPath = strcat(mainFilepath,'/Run');
    if currentRun < 10
        localPath = strcat(localPath,'0000');
    end
    if currentRun ≥ 10 && currentRun < 100
```

```
        localPath = strcat(localPath,'000');
    end
    if currentRun ≥ 100 && currentRun < 1000
        localPath = strcat(localPath,'00');
    end
    localPath = strcat(localPath,int2str(currentRun));
    localPath = strcat(localPath,'/MATLAB/DPsv00001.mat');
    load(localPath);

    figure(1)
    plot(abs(S1(:,1)),abs(S1(:,2)));
    figure(2)
    plot(X1(:,1),X1(:,2));

    [Vpk_Splot,fpk_Splot] = findpeaks(abs(S1(:,2)));
    [voltage_Splot, frequency_Splot]=findPeaksGreaterThan(Vpk_Splot,S1, ...
                                                           fpk_Splot,0.1);

    [Vpk_Xplot,fpk_Xplot] = findpeaks(X1(:,2));
    [voltage_Xplot, frequency_Xplot]= findPeaksGreaterThan(Vpk_Xplot,X1,...
                                                           fpk_Xplot,0.1);

    size_Splot = size(voltage_Splot);
    testNumberArr_Splot = zeros(size_Splot(1),size_Splot(2));
    mean_voltage_xplot = zeros(size_Splot(1),size_Splot(2));
    for i = 1:size(voltage_Splot)
        testNumberArr_Splot(i) = currentRun;
        mean_voltage_xplot(i) = mean(voltage_Xplot);
    end

    writematrix([testNumberArr_Splot, voltage_Splot, frequency_Splot, ...
                 mean_voltage_xplot],strcat(mainFilepath, ...
                 excellFilename), 'Range',strcat('A',int2str(height)));

    height = height + size_Splot(1);

end

```

```matlab
function [peaks, locations]= findPeaksGreaterThan(pks,arr,locs,val)
    currentindex = 1;
    arrSizea = size(pks);
    arrSize = arrSizea(1);
    peaks = zeros(arrSize,1);
    locations = zeros(arrSize,1);
    for i=1:arrSize
        if(pks(i) ≥ val)
            peaks(currentindex) = pks(i);
            locations(currentindex) = arr(locs(i),1);
            currentindex = currentindex + 1;
        end
    end
    peaks = peaks(peaks ≠ 0);
    locations =  locations(locations ≠ 0);
end
```

The following MATLAB script was used to generate plots based on one section of data (either the first or second data collection date) for a specific run number. The title of the plots was manually changed based on the corresponding run number SignalCalc settings.

```matlab
%%
clear all; close all; clc;
%% Graphing run function for either file path

%First File path (STRICTLY for local file path of stored files)
% mainFilepath = 'D:\WashU\Classes\Fall 2020\Vibrations LAB\Lab 1\Run Files';
%Second File path (STRICTLY for local file path of stored files)
mainFilepath = 'D:\WashU\Classes\Fall 2020\Vibrations LAB\Lab 1\Run ...
    Files\second run files';

% Input for the run number that is inteded to be graphed
runNumber = 72;
for currentRun = runNumber
    localPath = strcat(mainFilepath,'/Run');
    if currentRun < 10
        localPath = strcat(localPath,'0000');
```

```
    end
    if currentRun ≥ 10 && currentRun < 100
        localPath = strcat(localPath,'000');
    end
    if currentRun ≥ 100 && currentRun < 1000
        localPath = strcat(localPath,'00');
    end
    localPath = strcat(localPath,int2str(currentRun));
    localPath = strcat(localPath,'/MATLAB/DPsv00001.mat');
    load(localPath);

    % Creates Frequency spectrum plot
    figure(1)
    plot(abs(S1(:,1)),abs(S1(:,2)));
    title_1 = {'Frequency Spectrum Plot for a 25 Hz Square Wave';
               '\color{gray}\rm FSpan = 1000, Lines = 800'};
    title(title_1);
    xlabel('Frequency (Hz)');
    ylabel('Voltage Magnitude (V)');

    % Creates Time domain plot
    figure(2)
    plot(X1(:,1),X1(:,2));
    title_2 = {'Time Domain Plot for a 25 Hz Square Wave';
               '\color{gray}\rm FSpan = 1000, Lines = 800'};
    title(title_2);
    xlabel('Time (s)');
    ylabel('Peak Voltage (V)');
    xlim([0 0.2]);
end
```