

# Laboratory 02 - Introduction to Digital Data Acquisition, FFT, and Spectrum Analysis 1

MAE 3120, Spring 2020

Joe Marx & Christian Trummer

## Grading Rubric

Procedures, Results, Plots, Tables - 60%

Discussion Questions - 25%

Neatness - 15%

## Abstract

In this experiment, several methods of data analysis were explored and tested in order to further understand how these methods can improve data acquisition and analyzation. A waveform generator was used to create different signals while Python3 code was implemented to retrieve and then analyze the signal data. Once acquired, the data spectra was found; in order to minimize the leakage and energy loss, the Hanning Window method was applied to the data to smooth the signal and improve its continuity with a weighted cosine function. Conclusions about data acquisition methods were made as well; the effects of changing sampling frequency and/or the resolution were explored. Increasing sampling frequency will take data at a higher rate, however, this increases frequency resolution and can lead to leakage when plotting the signal's spectra. Increasing the sampling period will increase the total amount of data points taken; if the period is an integer multiple of the frequency, changing the period can decrease leakage as well. Using Hanning windowing and following the nyquist criterion, optimization of signal detection can be optimized in every stage to improve accuracy and minimize leakage: acquisition, analysis and application.

## Introduction and Background

Prior to the 1980s, the oscilloscope and strip-chart recorder represented the most common methods for measurement of time-varying signals. With time PC-based digital data acquisition became standard in most laboratories. By combining high-speed data acquisition cards with graphical software, it is now possible to design complex data acquisition systems with real-time data analysis and plotting features, with minimal programming. The data acquisition hardware converts analog inputs into the digital domain at the specified sampling rate, and the software manipulates and displays the desired output.

In this lab we use Python and the NI-DAQmx API for digital data acquisition. Using the **DAQ** Jupyter Notebook developed for this class, instructions are issued to the data acquisition hardware, either inside the PC or external to the PC (the hardware we use in our lab is connected through the USB port). The **DAQ** can be configured to record data to files, change sampling parameters, and display a live output of your sampled signal.

The goal of this tutorial is to provide you with your first experience using the **DAQ** notebook to perform data acquisition. You will use the **DAQ** to take samples and plot voltage data and to illustrate some limitations of digital data acquisition systems.

To help verify that you have configured the **DAQ** properly before performing trials, you will learn how to use **NI MAX** (a software provided by National Instruments).

Ultimately, you will experiment with digital data acquisition and some of its shortcomings. For your report you are expected to save all the data you will acquire in the lab to files and plot them in Python.

---

In spectral analysis the goal is to determine the frequency content of a signal. Aliasing can be a serious problem with digital data acquisition if the experimenter is not careful. Significant measurement errors called **aliasing** errors are possible if the waveform is not sampled at high enough frequency. To avoid aliasing, the **sampling rate** must be at least twice the maximum frequency of the measured signal. This restriction is called the **Nyquist criterion**. Signal aliasing occurs when waveforms are sampled at frequencies below the Nyquist frequency. Aliased signals appear to have frequencies (and possibly even waveform *shapes*) that differ from those of the actual signal. For adequate resolution of the waveform shape, data should be sampled at a much higher frequency – typically at least five times the Nyquist frequency, if possible.

Digital PC-based data acquisition will not totally replace oscilloscopes, at least not in the near future. The reason is sampling frequency. The maximum sampling frequency of modern PC A/D systems is typically less than a MHz (megahertz). By comparison, a good digital oscilloscope may sample as high as several GHz (gigahertz)!

The fast Fourier transform (FFT) is a computationally efficient form of the more general discrete Fourier transform (DFT), which is itself a discretized version of the even more general Fourier transform (FT). Like Fourier series analysis, FFT analysis enables us to calculate the frequency content of a signal. Fourier series analysis is useful for continuous, periodic, analog signals of known fundamental frequency. FFT analysis, on the other hand, is useful for discretely sampled (digital) data, and can be applied even if the signal is not periodic. With FFT analysis, the fundamental frequency of a periodic signal does not have to be known a priori. NumPy has built-in FFT features, which are utilized in this lab.

For  $N$  sampled data points at sampling frequency  $f_s$ , the most useful output of an FFT calculation is the frequency spectrum or amplitude spectrum, which is a plot of modified FFT amplitude versus frequency. The frequency spectrum shows the relative importance or contribution of discrete frequencies, which range from zero to  $f_s / 2$ . (The factor of two is a direct result of the Nyquist criterion.) The number of discrete frequencies on the frequency spectrum plot is  $N / 2 + 1$ . This is half of the number of discretely sampled data points in the original signal, plus one extra since we typically plot both extreme values – from zero Hz (DC component) to the folding frequency  $f_{folding}$ .

Here are some useful definitions for FFTs:

- $N$  is the **total number of discrete data points** taken.  $N$  is an input parameter, chosen by the user.
- $f_s$  is the **sampling frequency**, in Hz.  $f_s$  is an input parameter, chosen by the user. *All other properties of the FFT, including sampling time, maximum frequency, frequency resolution, etc., are determined solely from these two inputs,  $N$  and  $f_s$ .*
- $T$  is the **total sampling time**, and is calculated as  $T = N / f_s$ . To increase the sampling time, we must either *increase* the number of data points, or *decrease* the sampling frequency (or both).
- $f_{folding}$  is the **folding frequency**, also called  $f_{max}$ , the **maximum frequency**.  $f_{folding} = f_s / 2$ .  $f_{folding}$  is the maximum frequency plotted on the frequency spectrum plot, since  $f_{folding}$  is the maximum frequency at which reliable information about the signal can be calculated, due to the Nyquist criterion. The only way to increase  $f_{folding}$  is to increase the sampling frequency.
- $\Delta f$  is the **frequency resolution** or **frequency increment** of the frequency spectrum.  $\Delta f = 1 / T = f_s / N$ . On the frequency spectrum plot, amplitudes of the FFT are plotted at  $N / 2 + 1$  discrete frequencies, each separated by  $\Delta f$ . In other words, the discrete values of  $f$  are 0,  $\Delta f, 2\Delta f, 3\Delta f, \dots, [(N / 2 - 1)]\Delta f$ . (The amplitude at exactly  $f_{folding}$ , i.e., at  $(N / 2)\Delta f$ , is also plotted; this results in a total of  $(N / 2) + 1$  discrete frequencies, counting both  $f = 0$  and  $f = f_{folding}$ ). The *only* way to increase the frequency resolution is to increase sampling time.

Here is a summary of some useful techniques and rules to remember when calculating FFTs:

- To get better frequency resolution for a fixed sampling frequency, increase the number of data points.
- To get better frequency resolution for a fixed number of data points, decrease the sampling frequency. (But be careful here not to let  $f_s$  fall below the Nyquist criterion limit).
- To get frequency component information at higher frequencies, increase the sampling frequency.
- To reduce **leakage** in the frequency spectrum, do one or more of the following:
  - Increase the number of sampled data points  $N$  (at the cost of more computer time).
  - Decrease the sampling frequency  $f_s$  (but do not sample at such a low frequency that the Nyquist criterion is violated).
  - Multiply the time signal by a **windowing** function prior to taking the FFT (at the cost of throwing away a significant portion of the signal, in particular data points near the start and finish of the time trace).

## Precautions

Due to the nature of the A/D conversion process, it is necessary to take several precautions to ensure that the analog signal is adequately represented once it is converted to the digital domain. An experimenter who is not careful can encounter problems such as poor resolution, clipping, and aliasing. The most important factors when sampling a signal are dynamic range, input range, and sampling rate.

- The dynamic range of the data acquisition card is the number of discernible levels or bins that the A/D converter can assume. The number of bits outputted by the A/D converter determines the dynamic range; namely, there are  $2^N$  available levels or bins for an  $N$ -bit A/D converter. For example, a 12-bit A/D converter has a dynamic range of  $2^{12}$ , or 4096. It is important to utilize as much of the available dynamic range as possible.
- Another potential problem with digital data acquisition is clipping. If a voltage lies beyond the input range of the A/D converter, the signal is clipped.

## Equipment

- Computer
- Software: NI MAX, Jupyter
- Hardware: National Instrument CompactDAQ cDAQ-9174, NI-9201 C Series Voltage Input Module
- Function/waveform generator, along with appropriate cables
- Oscilloscope

## Procedure

## Part I - Introduction to NI MAX and Hardware Configuration

In this section, you will use the **NI MAX** software to verify that you have properly setup your hardware. The CompactDAQ Data Acquisition System (cDAQ) reads from four input modules. In this lab, we will be using the **NI 9201** module.

1. Verify that **NI MAX** and appropriate drivers are installed on your computer (see installation instructions if using personal computer).
2. Ensure the **NI 9201** card is in the furthest left spot on the cDAQ.
3. Connect the cDAQ to the computer using the USB cable and plug it in to a power source.
4. **NI MAX** may automatically launch after the cDAQ is connected and powered. Open **NI MAX** manually if this does not occur.
5. Select the *Devices and Interfaces* tab under *My System* in **NI MAX**. There, select the device with a name similar to **NI cDAQ-9174 "cDAQ1"**.
6. Select the first module (**1: NI 9201 "cDAQ1Mod1"**). If the **NI 9201** card is not the first module, change its location and then click the "Refresh" button at the top. Take note of the *Name* displayed on the right side of the screen (**cDAQ1Mod1**) as you may need this later when configuring the **DAQ** program.
7. Select *Test Panels* from the top right menu.
8. Set the Rate to **1000 Hz**, and the Mode to **Continuous**.
9. Click **Start** to observe a live graph of the analog input.
10. A low voltage signal should be observed around 0V. This is noise being measured by the device as no voltage source is connected to it. As long as there are no errors here, you can proceed with your data acquisition in the next part of the procedure.
11. Should you wish to view a live output of the signal from the module at any point, revisit the *Test Panels* screen in **NI MAX**.

## Part II - Digital Data Acquisition

In this section, you will use the **DAQ** Jupyter Notebook to test limitations of digital data acquisition systems.

1. Open the **DAQ** Jupyter Notebook located in the *Labs* folder. Read below for information on how to use the `acquire` function.
2. In the second cell, you will notice the `acquire` function. This function will be used throughout the course to acquire data using Python.
3. The docstring provides a simplified explanation of how to use certain variables in the function. A more detailed explanation is provided below.
4. `acquire` requires two variables to be passed, `N` and `fs`; all other arguments are optional and can be used as needed. To use an optional argument, include the arguments name and the value you wish to assign to it (e.g. `max_channel=7` ).
5. Arguments
  - `N` (int) : Number of samples to measure.
  - `fs` (float) : Sampling frequency in Hz (Hertz, cycles per second).
  - `max_channel` (int) : Maximum voltage channel to read from. If you look at the side of the **NI 9201** cards, you will notice that it lists channels from `ai0` to `ai7`. Setting this to `0` will result in data only being read from the `ai0` channel. Setting this to `7` will result in data being read from channels `ai0 - ai7`. The default value is `0`.
  - `file_out` (String) : Output directory and file name of initial output. This is the name and location of the output file. The default value is `' '`, which results in no file output. An example of a valid file name would be `'C:\\Users\\Josh\\Downloads\\Lab2_0.csv'`. All file names **must** end in an underscore followed by a one or two digit number, then the CSV file extension (e.g. `_0.csv`). Do not forget that Python requires the double backslash (`\\"`) in Strings.
  - `output` (String) : Type of output. `'X'` is the default value and outputs no file. `'N'` creates a new file every trial. `'R'` replaces data from the last trial. `'A'` appends new data to the last trial's file. If no file exists initially, a file will be created so long as `'X'` is not selected.
  - `time_sep` (float) : Time increment to update live plot in seconds. By default, the live plot updates every `1` second. It cannot update at any faster rate. This value does not need to be changed unless you are acquiring data for a long period of time and/or at a high sampling rate ( $> 15$  seconds). If you are acquiring data for more than 1 minute, this value should be changed to a number around `30`.
  - `zero_bound` (bool) : Determines whether live graph is bounded to 0. By default, this value is `True`, meaning that you will still be able to see the data recorded at 0 seconds while the live plot is updating. If you want to see the data from only the current `time_sep` value, set this to `False`.
  - `daq_name` (String) : Name of DAQ card/system. By default, this is `'cDAQ1Mod1'`. This is the String from *Part I - Step 6*. While you should not need to change this value, if the card you are using is in a different slot, the name may be different.
  - `mod_type` (String) : Type of DAQ card. By default, this is `'ai'` for an analog input. `'tc'` is used for a thermocouple. `'pr'` is used for an analog input with small differential voltage.

- `plt_sp_fft (bool)` : Plots frequency spectrum of data using Fast Fourier Transform (FFT). The default value is `False`. This argument only works with one channel input.
- `han_window (bool)` : Applies a Hanning window to the plot output. This does not change the saved data. The default value is `False`.

#### 6. Example Usage

- 100 samples of data, sampling rate of 1000 Hz, no file output: `acquire(100, 1000)`
- 100 samples of data, sampling rate of 1000 Hz, create a new file after every trial, file location location is 'C:\Users\Josh\Downloads':  
`acquire(100, 1000, file_out='C:\\Users\\Josh\\Downloads\\Lab2_0.csv', output='N')`
- 300e3 samples of data, sampling rate of 1000 Hz, no file output, refresh the plot every 30 seconds, view only the last 30 seconds of data:  
`acquire(300e3, 100, time_sep=30, zero_bound=False)`

## Part III - Data Acquisition Board Accuracy

1. Before using the *NI 9201* module, you need to test it to verify it is behaving within its specifications (given in Appendix B). Discuss with your lab partners what you need to measure and how you would proceed. Perform these measurements and report their results as well as how you performed them in your report.
2. While performing trials, make sure you save the measured data and can access the data after the lab section. You will need it to generate plots in your lab report.
3. Your lab report must include a table with the following columns:
  - Run number / filename
  - Lab section (which step in the procedures)
  - Signal
    - Frequency (Hz) and Waveform Type (e.g. *Sine 20*)
    - Amplitude (Vpp)
    - DC Offset (VDC)
  - Acquisition
    - Sampling Rate (Hz)
    - Number of Samples
  - Comments (e.g. *Curve looks continuous*)

## Part IV - Discrete Data Acquisition

Here you will demonstrate the digital data acquisition system acquire data at discrete times. We transform the original continuous (analog) signal into a discrete (digital) signal. There is a significant amount of theory regarding digital data acquisition and signal processing that will be introduced in the coming lectures and labs.

1. Using a BNC T-Adapter, connect the waveform/function generator to the first channel of the oscilloscope and a pair of output wires.
2. Connect the wire output to the *NI 9201* module. The positive output should be connected to `AI0` and the negative to `COM`.
3. Power on the oscilloscope. Configure the function generator to produce a 20 Hz sine wave, 0V DC Offset, 1V peak-to-peak amplitude.
4. Using the `acquire` function in the third cell of the **DAQ** notebook, set the sampling rate to 100,000 Hz and the number of samples to 10,000. You are acquiring 100 ms of data which corresponds to two waveforms and about 50,000 points per waveform.
5. Verify the signal using the oscilloscope. Run the `acquire` function with no file output to test that the function is working properly.
  - To have a 'live' output in Jupyter, use this line of code: `acquire(120 * fs, fs, time_sep=1, zero_bound=False)`. When you are done observing the output, click the *Stop* button in Jupyter.
6. Once you are comfortable with the acquisition, set a file output and save one run of the data for your report.
7. Decrease the data acquisition rate until the signal displayed on the graph starts to look "pixelated". To save the same duration of data, additionally adjust the number of samples so that you acquire 100 ms (two waveforms). Use the following frequencies and save each case to a file for use in your report:
  - 10,000 Hz
  - 1,000 Hz

- 100 Hz

8. In your report discuss the appearance of your recorded waveforms.

## Part V - Leakage effect

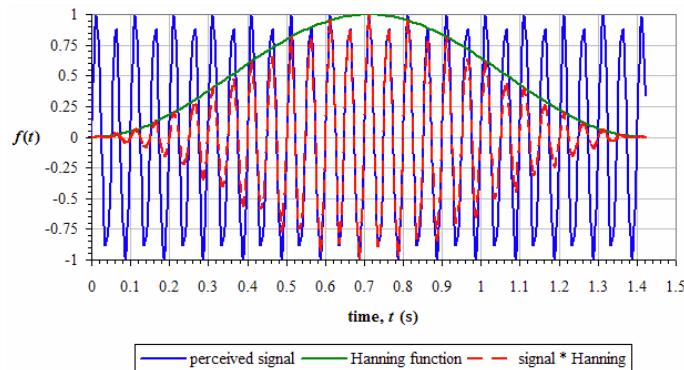
1. Remember to be organized when acquiring data. Generate a table that includes all the runs and associated parameters you will acquire in this lab and report this matrix of experiments in your lab report.
2. Using the `acquire` function, set `plt_sp_fft` to `True`. This will display the frequency plot when a FFT is applied to your data. Additionally, set the `time_sep` to 120 to prevent the graph from updating.
3. Set a 10 Hz sine wave, 0V DC Offset, and 2V peak-to-peak amplitude on the waveform generator. Keep monitoring your signal on the oscilloscope. For each of the steps below, save the time history signal to file. Use Python to recreate each spectrum. In your report you are expected to report both the time series and frequency spectra.
4. Set  $f_s = 200$  Hz,  $N = 256$ .
5. To try to reduce the leakage, try first to increase the sampling rate. Set  $f_s = 1000$  Hz,  $N = 256$ .
6. Try the following settings. Set  $f_s = 25$  Hz,  $N = 256$ .
7. Try the following settings. Set  $f_s = 25$  Hz,  $N = 512$ . What can you conclude about the spectral accuracy of our system?
8. Finally, try the following settings. Set  $f_s = 25.6$  Hz,  $N = 512$ . This corresponds to a "perfect FFT", can you think why?

## Part VI - Windowing of FFT

Keep the same parameters as above for the signal generator. Keep monitoring your signal on the oscilloscope.

1. Using the `acquire` function, set `han_window` to `True`. This will apply a Hanning windowing function. It has the following formula and appearance:

$$u_{\text{Hanning}}(t) = \frac{1}{2} \left( 1 - \cos \frac{2\pi t}{T} \right)$$



2. Redo the measurements from Part V and save data for each condition. Do you observe any improvement?
3. Add a 1V DC Offset and redo the measurements. What do you observe?
4. By now you should know how to optimize the spectral response of a system. Now create triangular waves of similar frequency and select the proper sampling rate, period, and windowing. How many harmonics do you observe?

# Results

## Figures

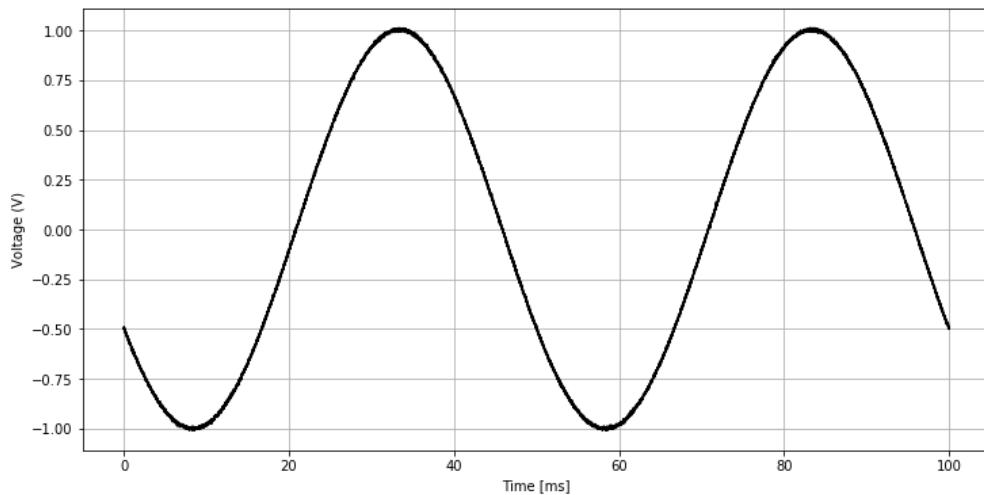


figure 1. Signal of a 20 Hz sine wave, 0V DC Offset, and 1V peak-to-peak amplitude with  $f_s = 100,000 \text{ Hz}$ ,  $N = 10,000$

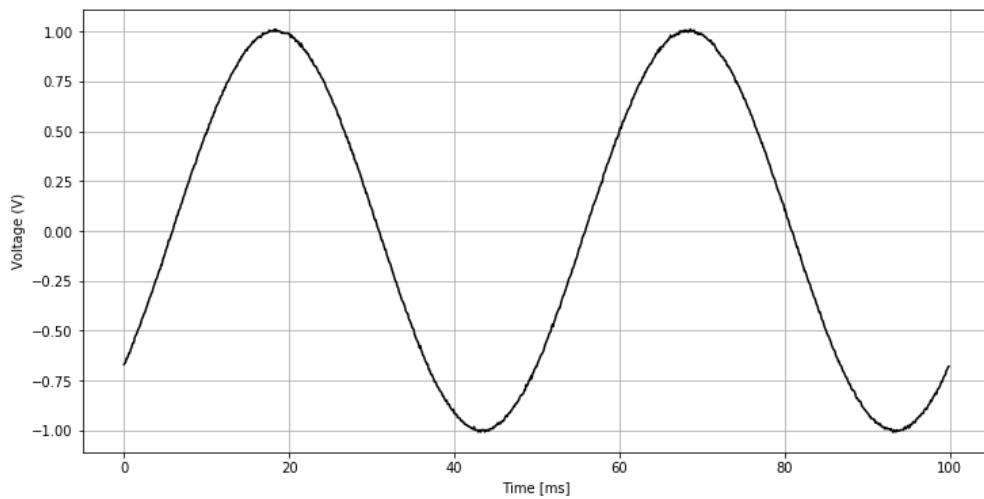


figure 2. Signal of a 20 Hz sine wave, 0V DC Offset, and 1V peak-to-peak amplitude with  $f_s = 10,000 \text{ Hz}$ ,  $N = 10,000$

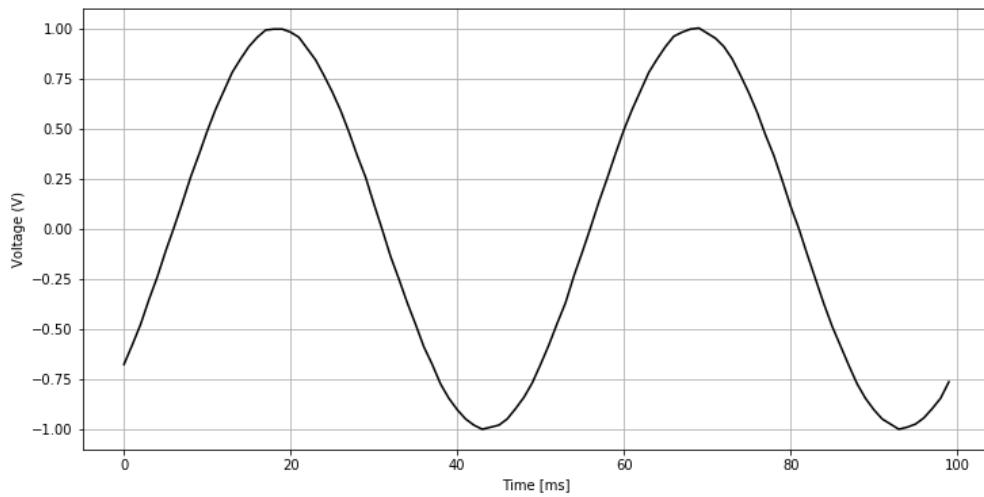


figure 3. Signal of a 20 Hz sine wave, 0V DC Offset, and 1V peak-to-peak amplitude with  $f_s = 1,000 \text{ Hz}$ ,  $N = 10,000$

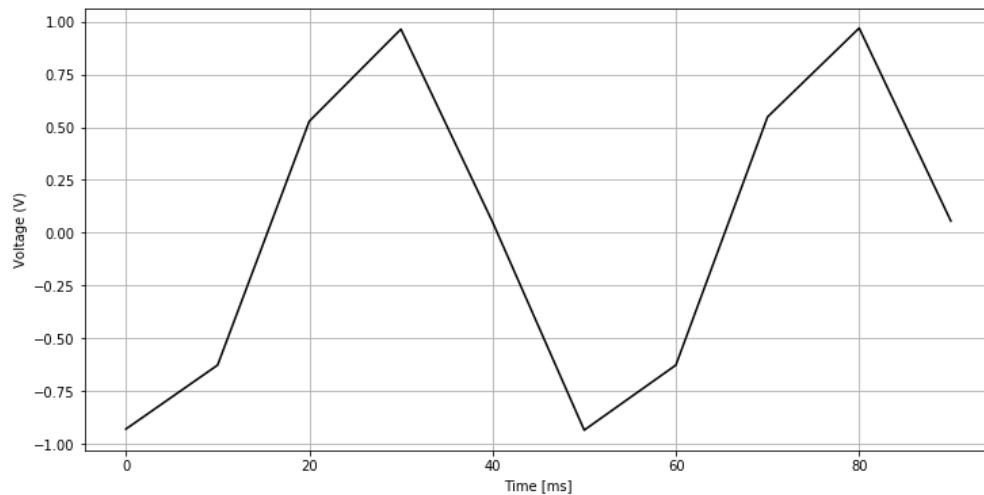


figure 4. Signal of a 20 Hz sine wave, 0V DC Offset, and 1V peak-to-peak amplitude with  $f_s = 100 \text{ Hz}$ ,  $N = 10,000$

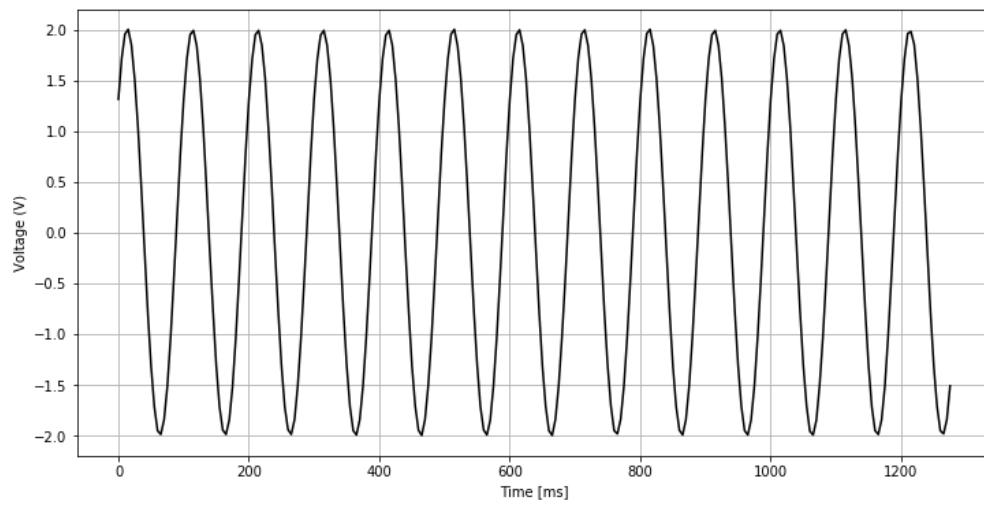


figure 5a. Signal of a 10 Hz sine wave, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 200 \text{ Hz}$ ,  $N = 256$

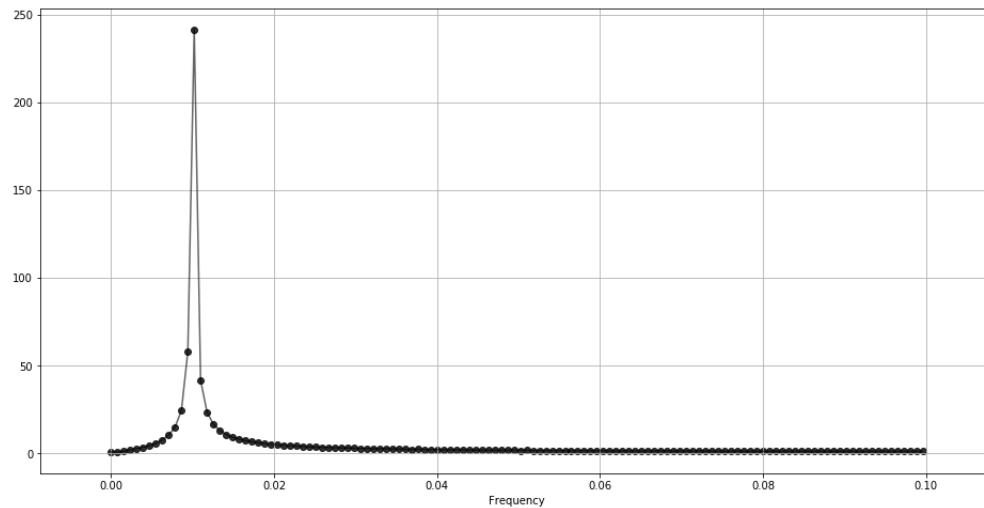


figure 5b. Frequency spectrum of a 10 Hz sine wave, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 200 \text{ Hz}$ ,  $N = 256$

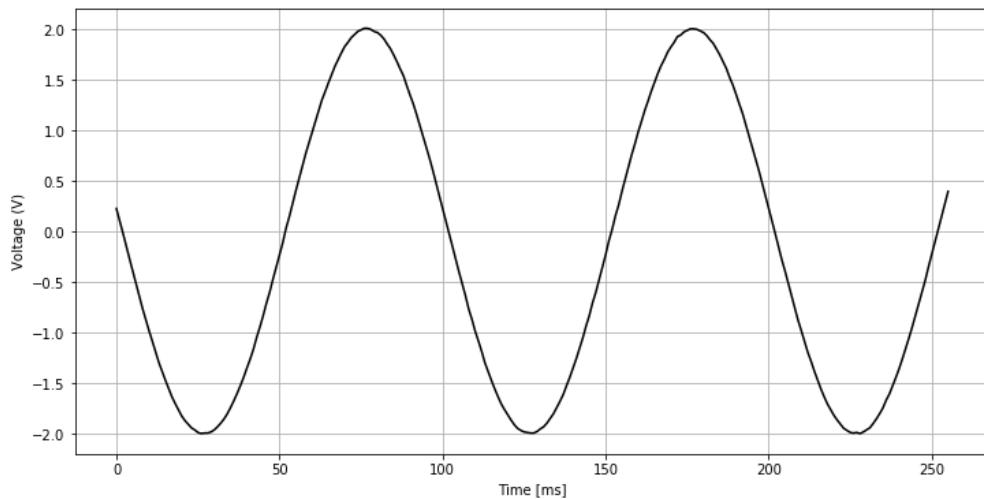


figure 6a. Signal of a 10 Hz sine wave, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 1000 \text{ Hz}$ ,  $N = 256$

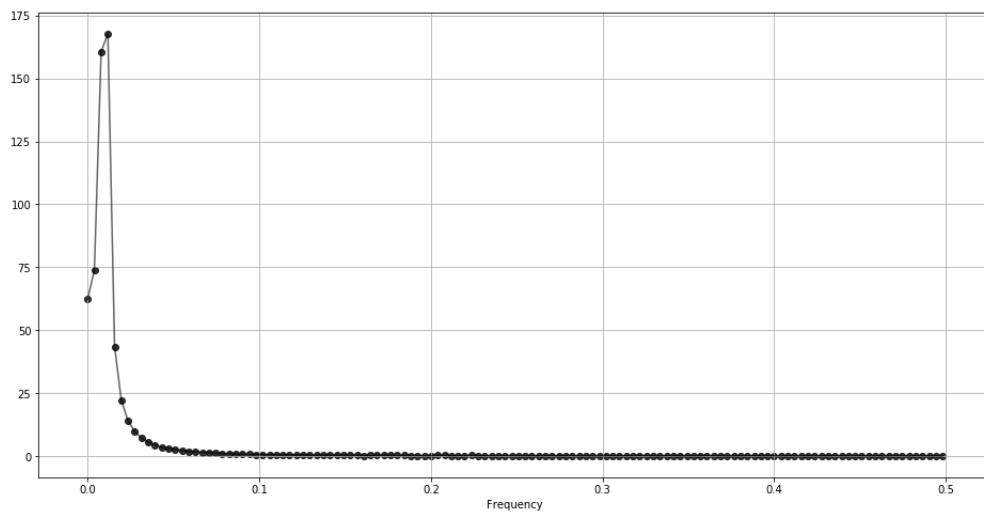


figure 6b. Frequency spectrum of a 10 Hz sine wave, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 1000 \text{ Hz}$ ,  $N = 256$

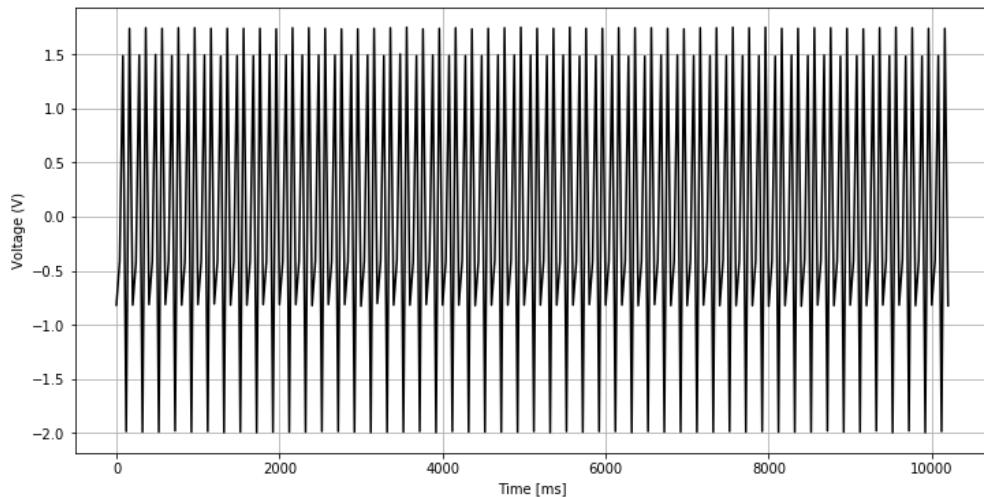
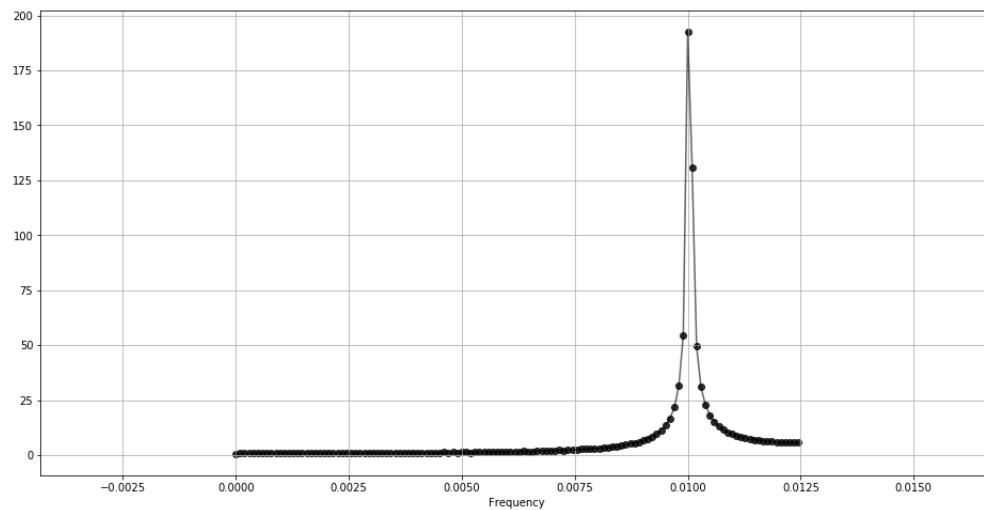
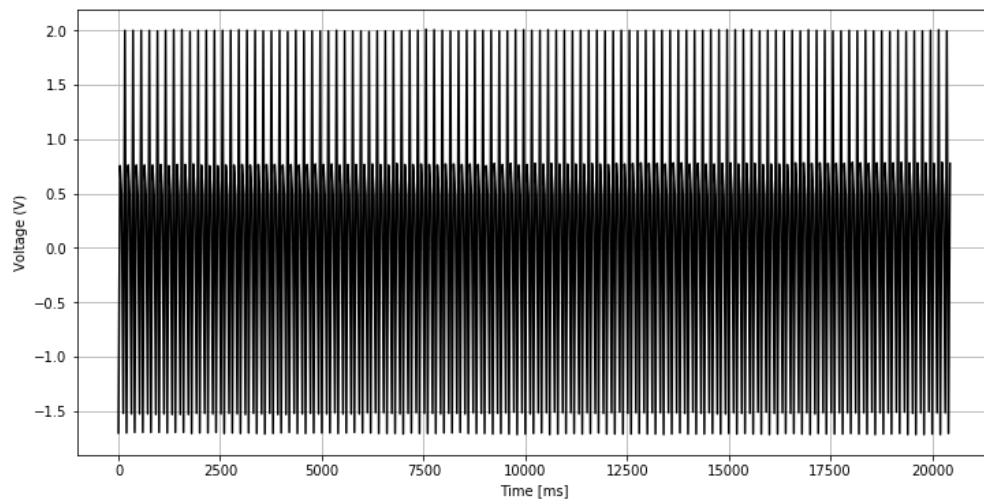
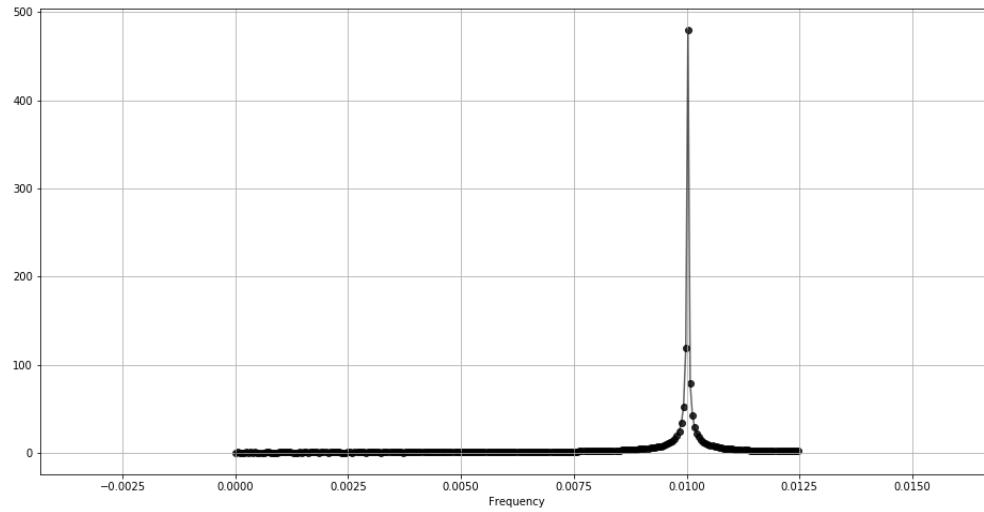
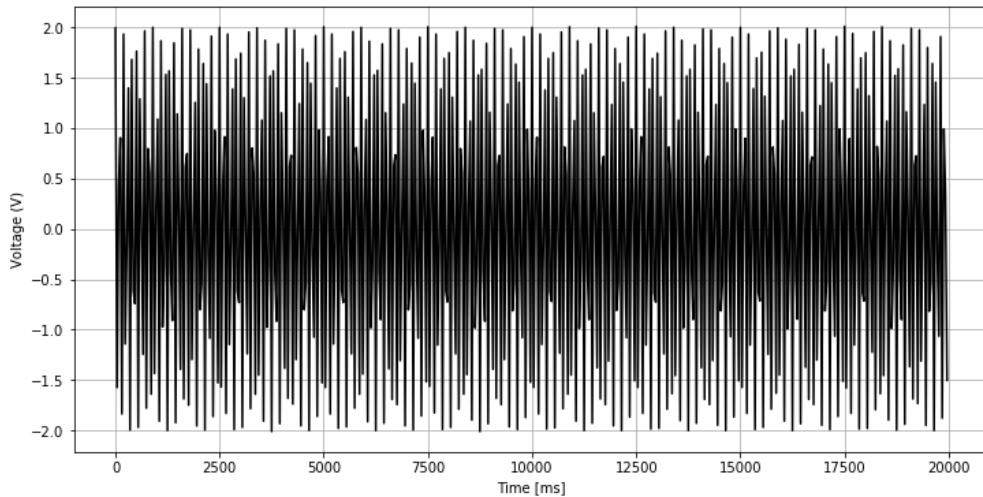
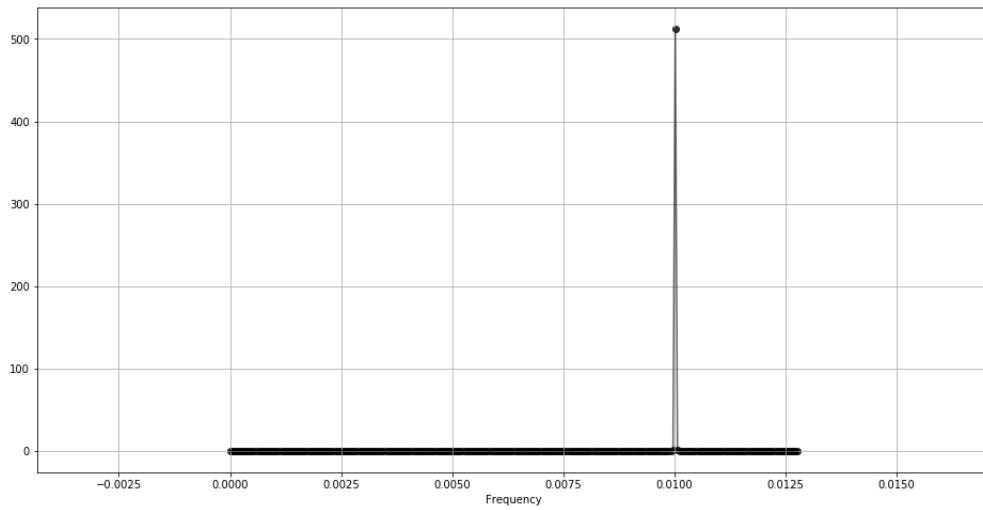
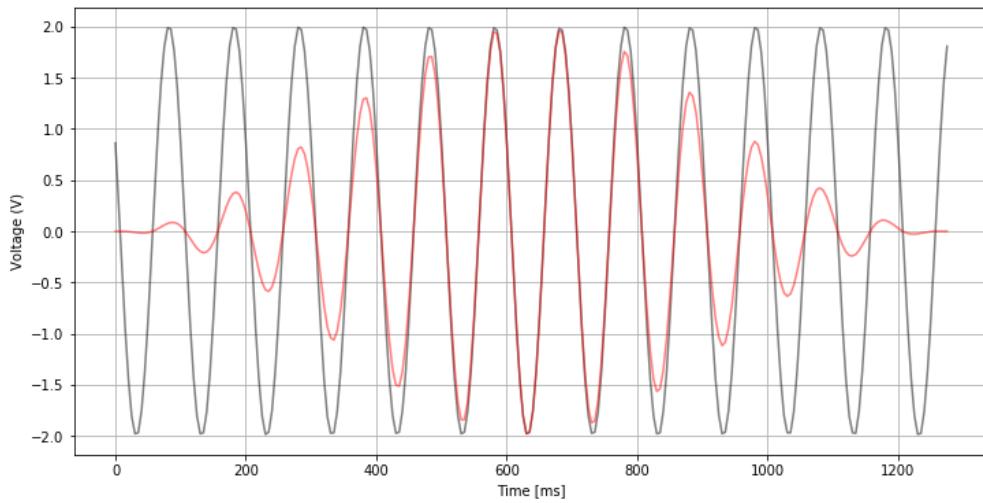


figure 7a. Signal of a 10 Hz sine wave, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25 \text{ Hz}$ ,  $N = 256$

figure 7b. Frequency spectrum of a 10 Hz sine wave, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25 \text{ Hz}$ ,  $N = 256$ figure 8a. Signal of a 10 Hz sine wave, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25 \text{ Hz}$ ,  $N = 512$ figure 8b. Frequency spectrum of a 10 Hz sine wave, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25 \text{ Hz}$ ,  $N = 512$

figure 9a. Signal of a 10 Hz sine wave, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25.6 \text{ Hz}$ ,  $N = 512$ figure 9b. Frequency spectrum of a 10 Hz sine wave, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25.6 \text{ Hz}$ ,  $N = 512$ figure 10a. Hanning Window Signal of a 10 Hz sine wave, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 200 \text{ Hz}$ ,  $N = 256$

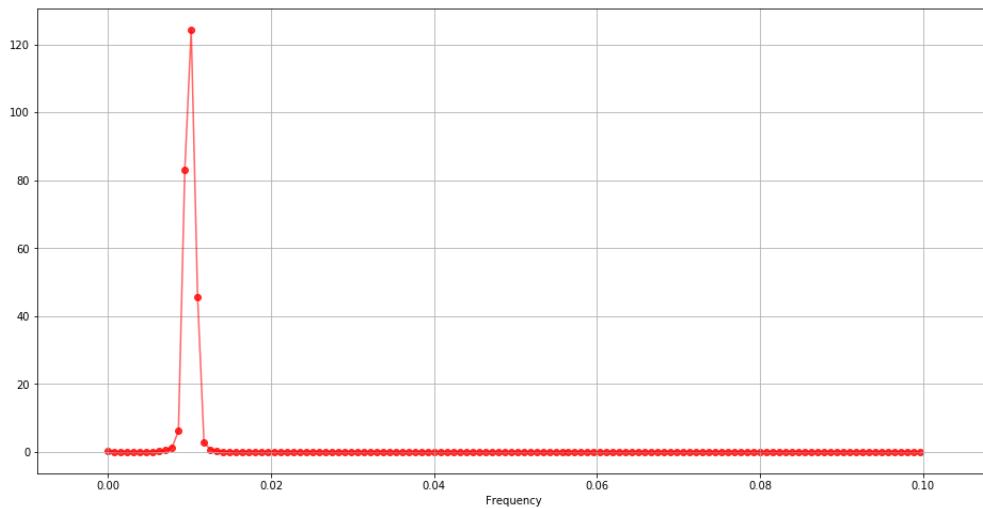


figure 10b. Frequency spectrum of a 10 Hz sine wave after Hanning Windowing, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 200 \text{ Hz}$ ,  $N = 256$

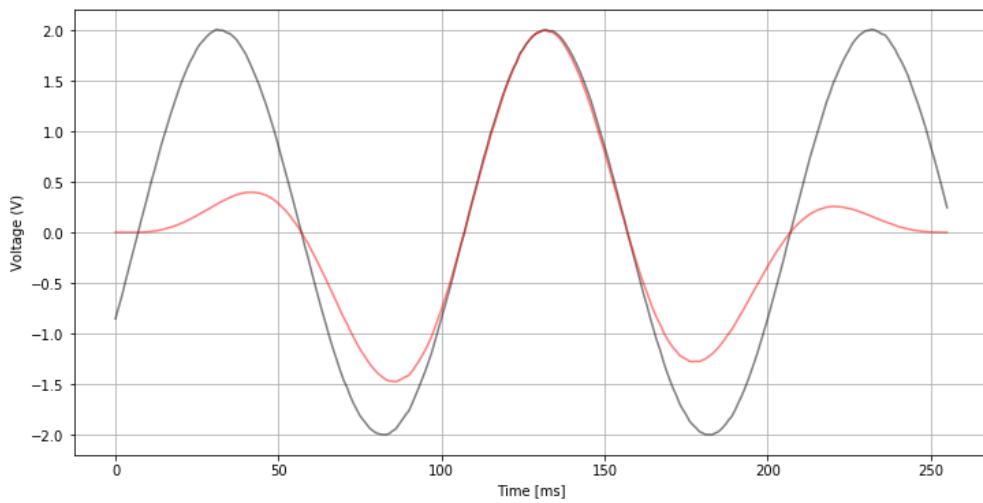


figure 11a. Hanning Window Signal of a 10 Hz sine wave, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 1000 \text{ Hz}$ ,  $N = 256$

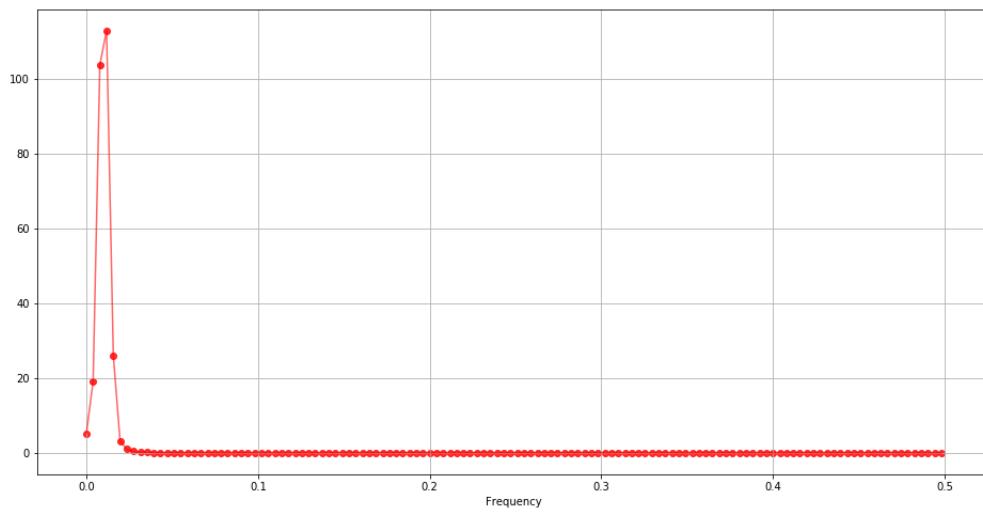


figure 11b. Frequency spectrum of a 10 Hz sine wave after Hanning Windowing, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 1000 \text{ Hz}$ ,  $N = 256$

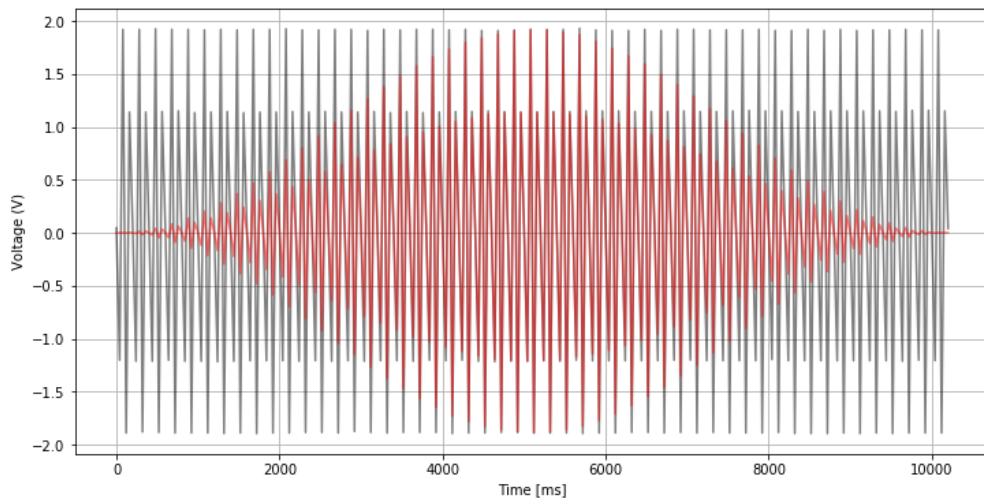


figure 12a. Hanning Window Signal of a 10 Hz sine wave, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25 \text{ Hz}$ ,  $N = 256$

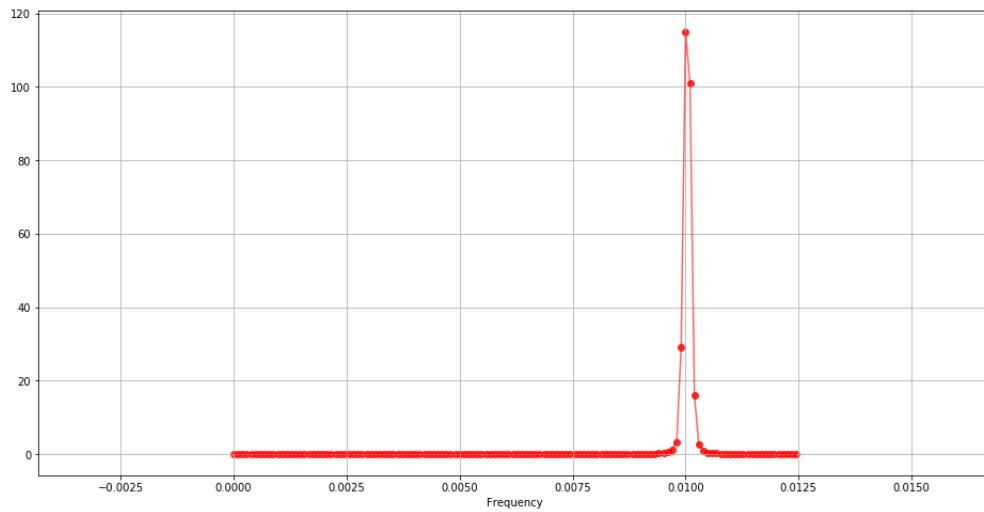


figure 12b. Frequency spectrum of a 10 Hz sine wave after Hanning Windowing, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25 \text{ Hz}$ ,  $N = 256$

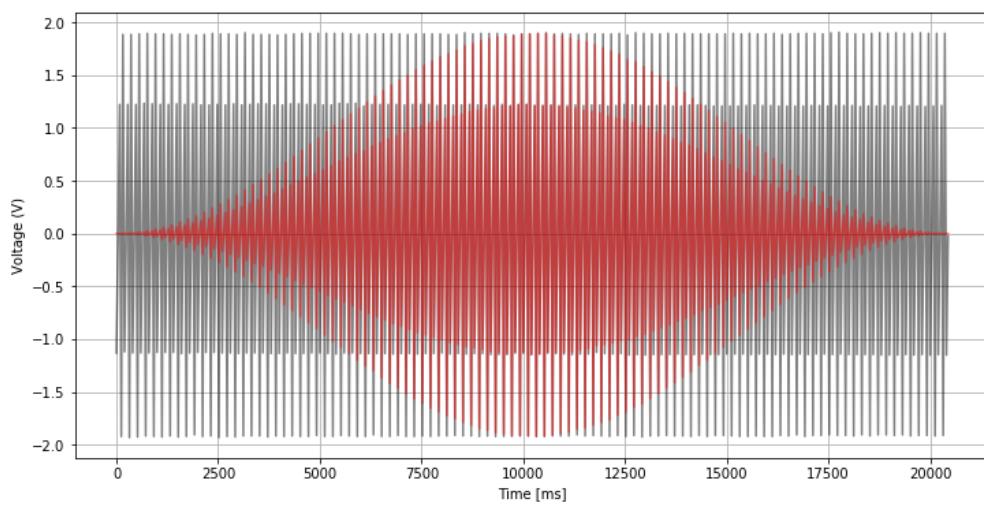


figure 13a. Hanning Window Signal of a 10 Hz sine wave, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25 \text{ Hz}$ ,  $N = 512$

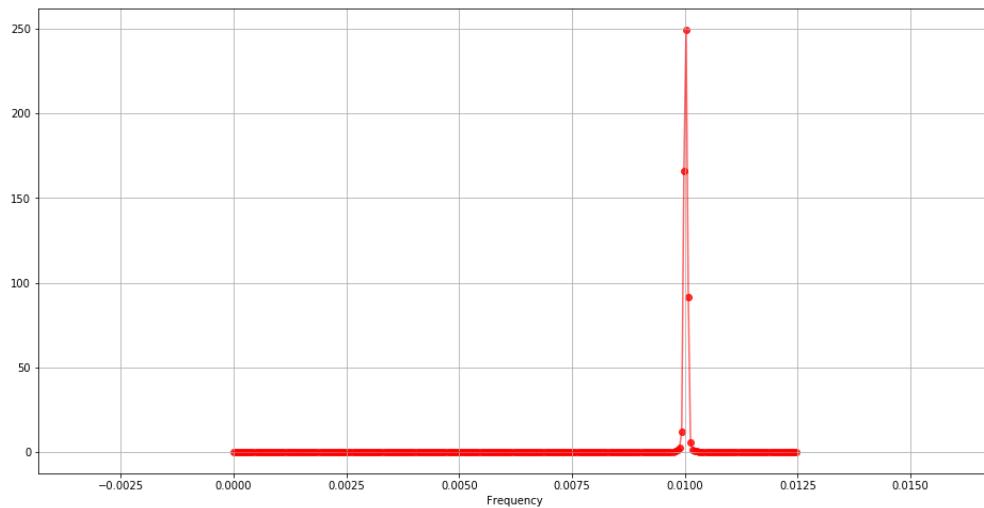


figure 13b. Frequency spectrum of a 10 Hz sine wave after Hanning Windowing, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25 \text{ Hz}$ ,  $N = 512$

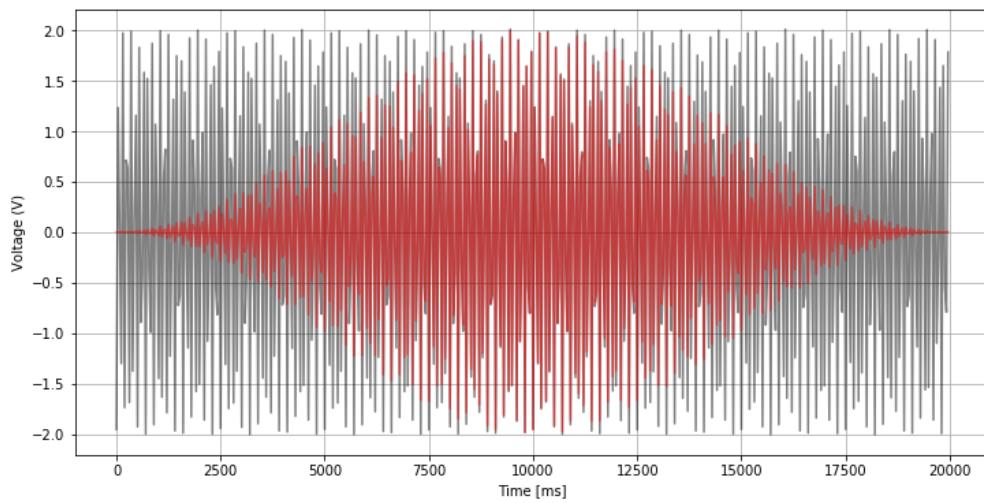


figure 14a. Hanning Window Signal of a 10 Hz sine wave, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25.6 \text{ Hz}$ ,  $N = 512$

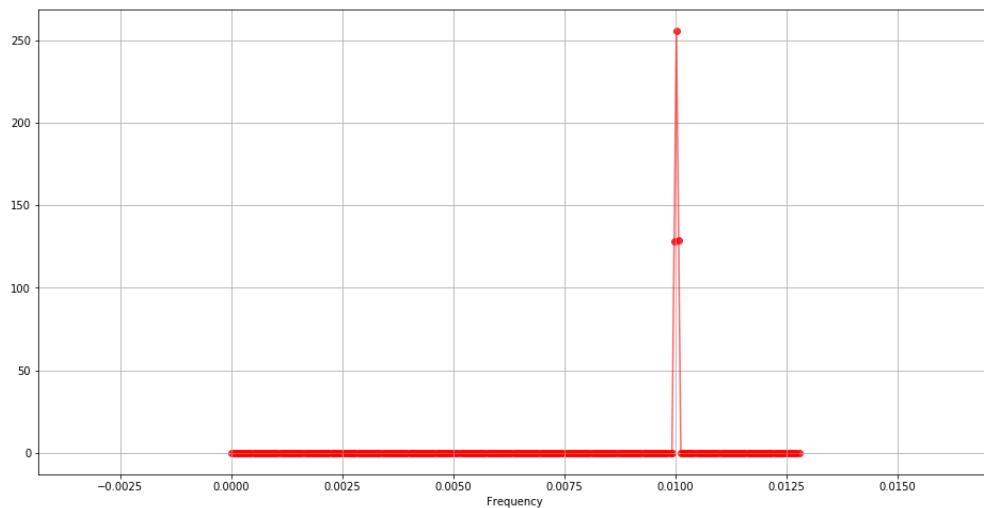


figure 14b. Frequency spectrum of a 10 Hz sine wave after Hanning Windowing, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25.6 \text{ Hz}$ ,  $N = 512$

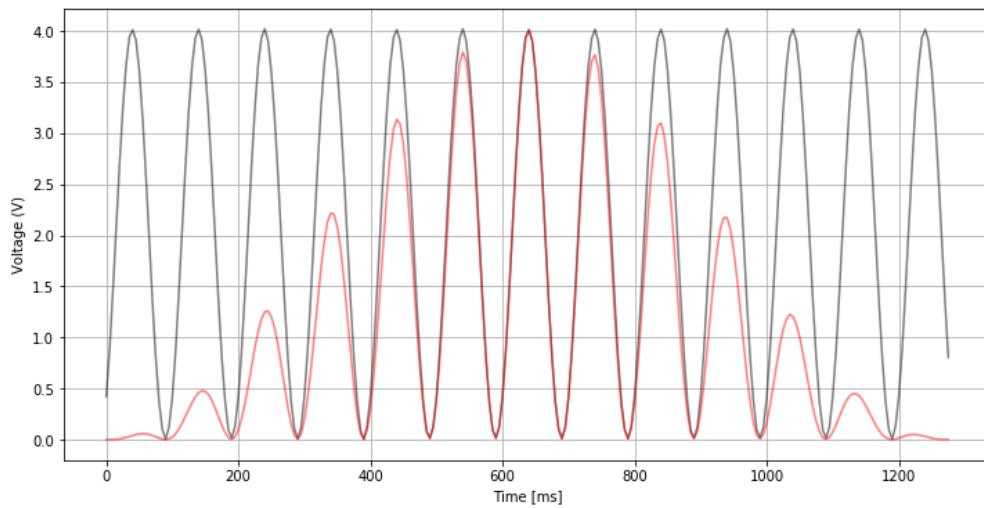


figure 15a. Hanning Window Signal of a 10 Hz sine wave, 1V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 200 \text{ Hz}$ ,  $N = 256$

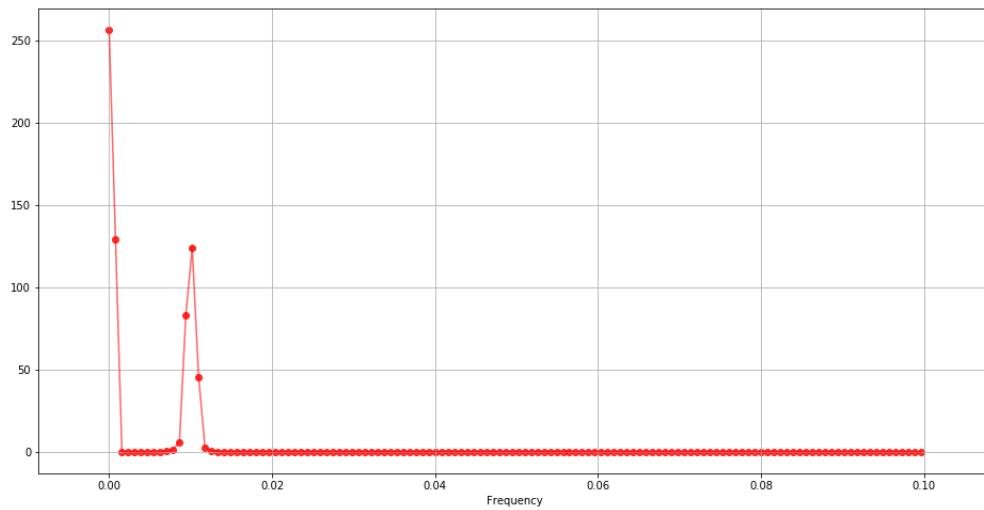


figure 15b. Frequency spectrum of a 10 Hz sine wave after Hanning Windowing, 1V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 200 \text{ Hz}$ ,  $N = 256$

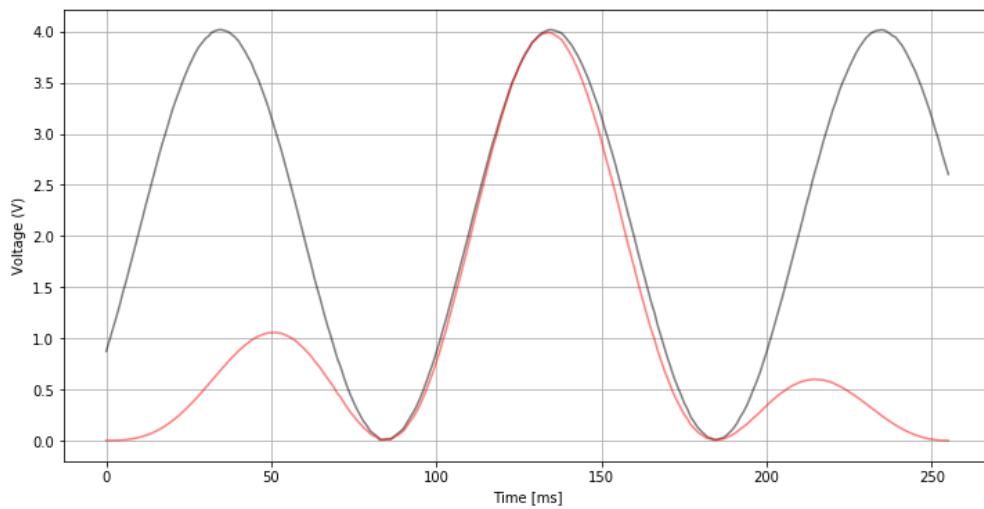


figure 16a. Hanning Window Signal of a 10 Hz sine wave, 1V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 1000 \text{ Hz}$ ,  $N = 256$

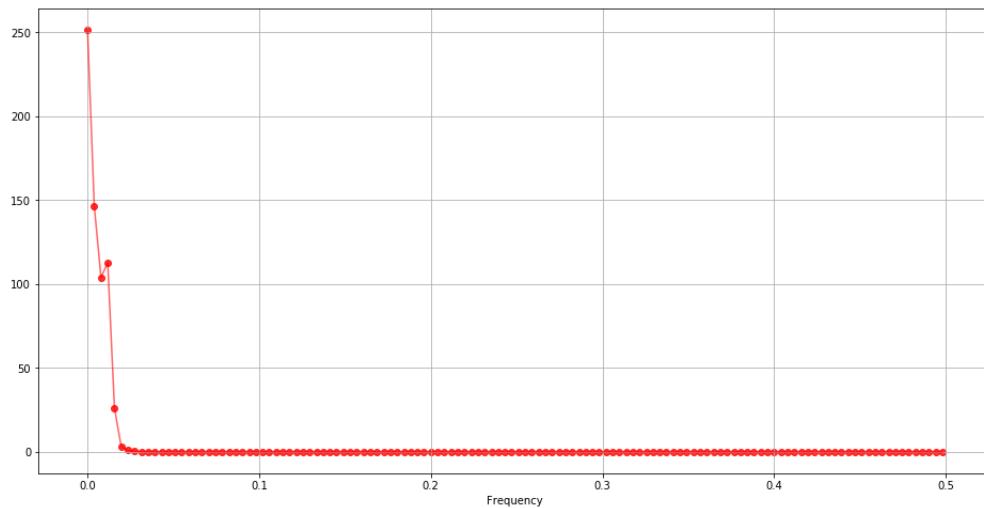


figure 16b. Frequency spectrum of a 10 Hz sine wave after Hanning Windowing, 1V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 1000 \text{ Hz}$ ,  $N = 256$

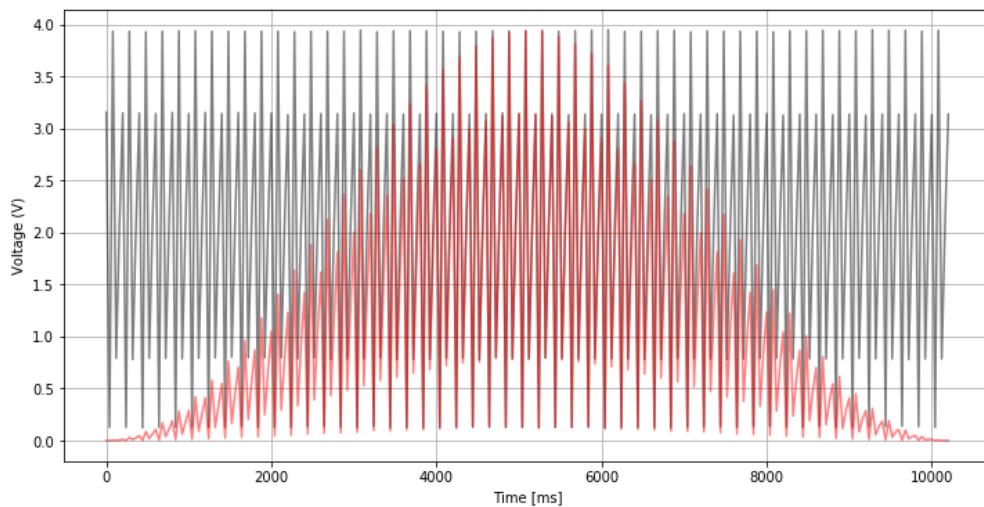


figure 17a. Hanning Window Signal of a 10 Hz sine wave, 1V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25 \text{ Hz}$ ,  $N = 256$

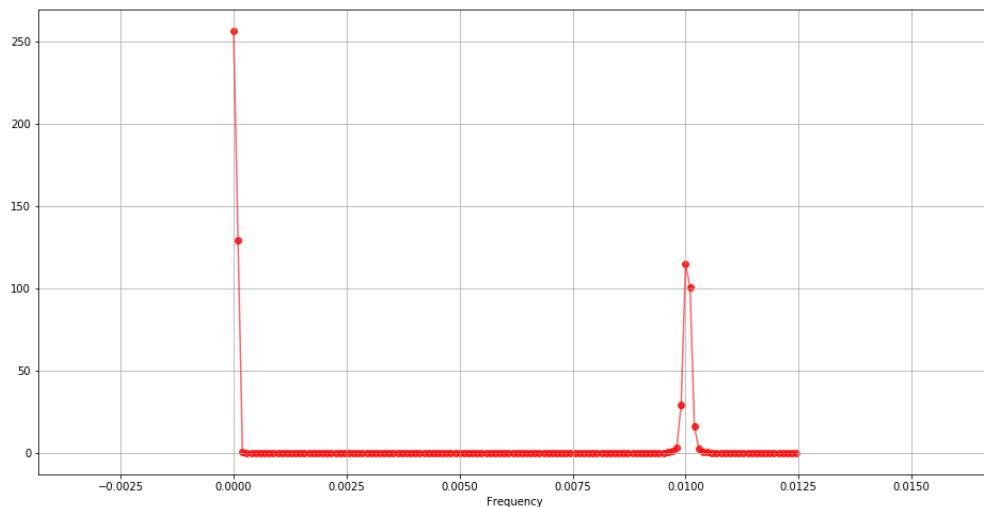


figure 17b. Frequency spectrum of a 10 Hz sine wave after Hanning Windowing, 1V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25 \text{ Hz}$ ,  $N = 256$

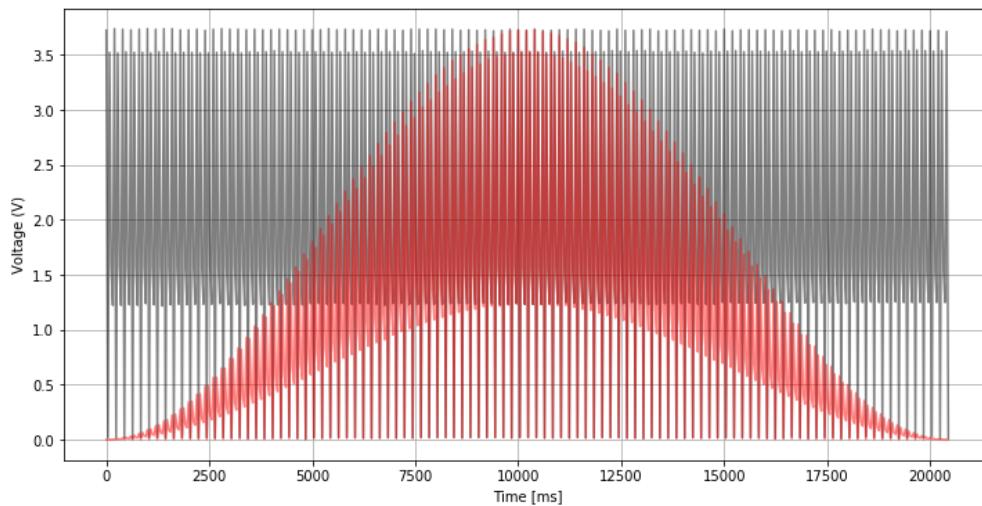


figure 18a. Hanning Window Signal of a 10 Hz sine wave, 1V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25 \text{ Hz}$ ,  $N = 512$

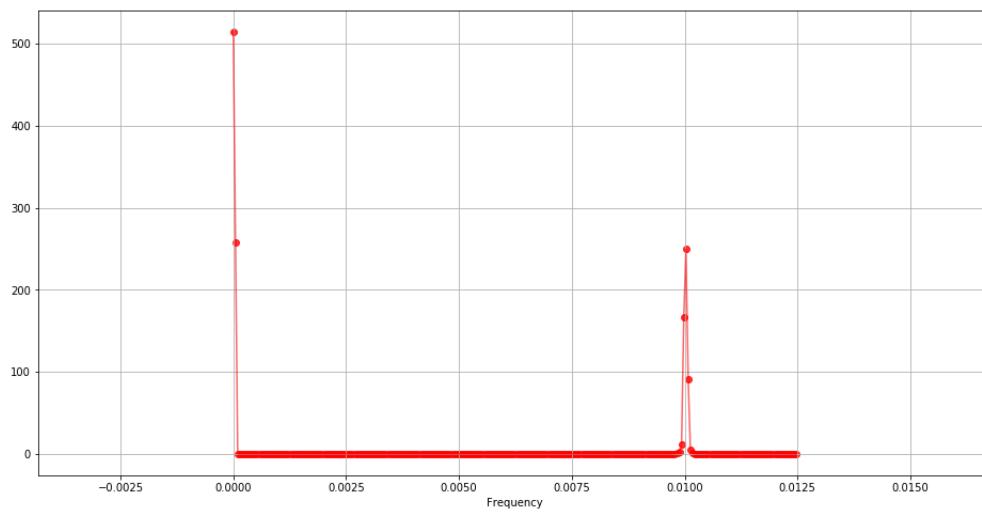


figure 18b. Frequency spectrum of a 10 Hz sine wave after Hanning Windowing, 1V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25 \text{ Hz}$ ,  $N = 512$

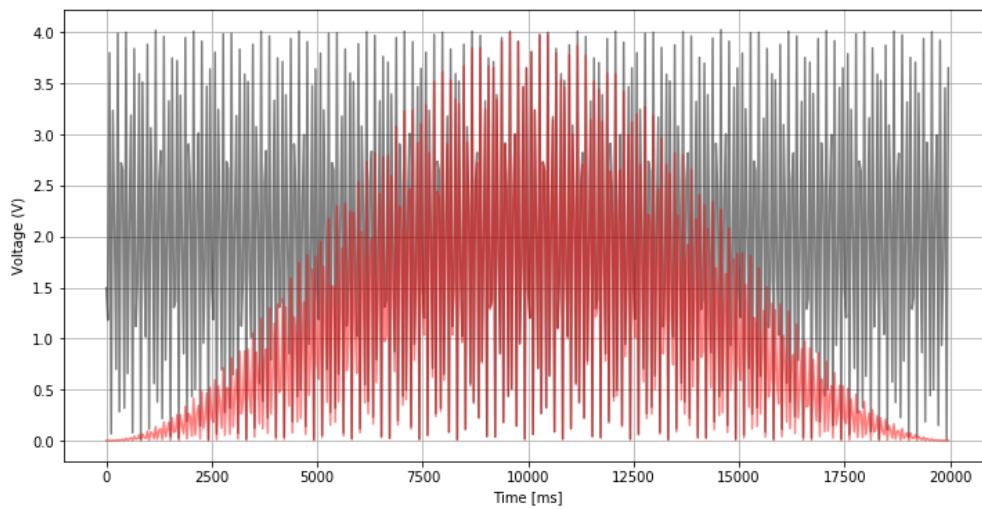


figure 19a. Hanning Window Signal of a 10 Hz sine wave, 1V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25.6 \text{ Hz}$ ,  $N = 512$

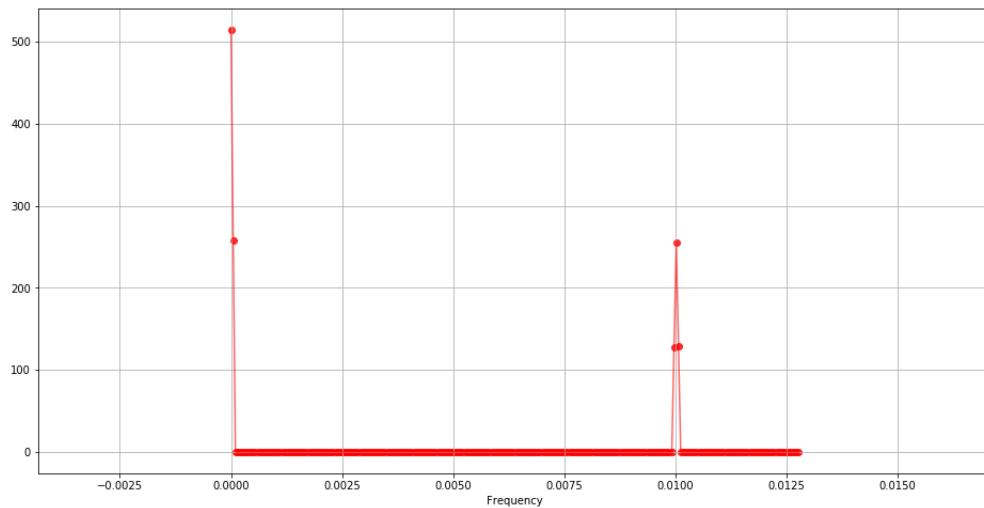


figure 19b. Frequency spectrum of a 10 Hz sine wave after Hanning Windowing, 1V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25.6 \text{ Hz}$ ,  $N = 512$

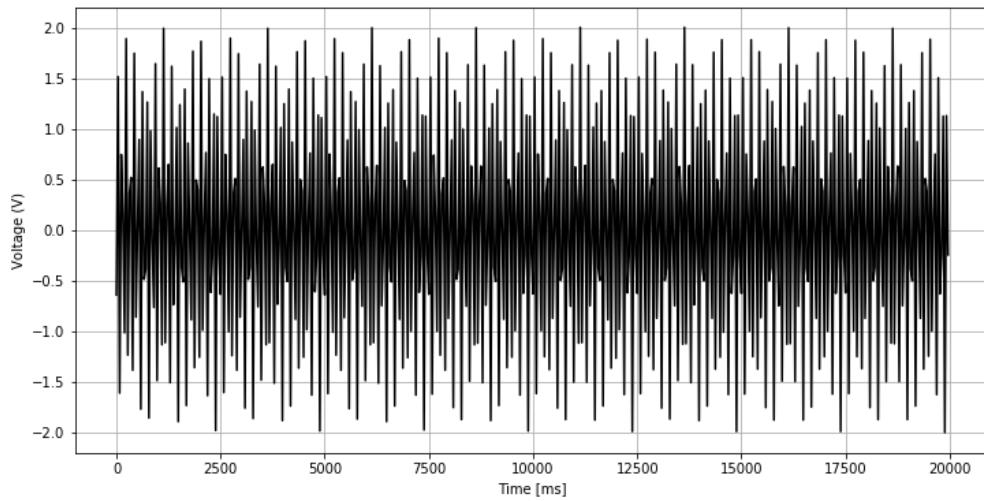


figure 20a. Signal of a 10 Hz triangle sine wave, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25.6 \text{ Hz}$ ,  $N = 256$

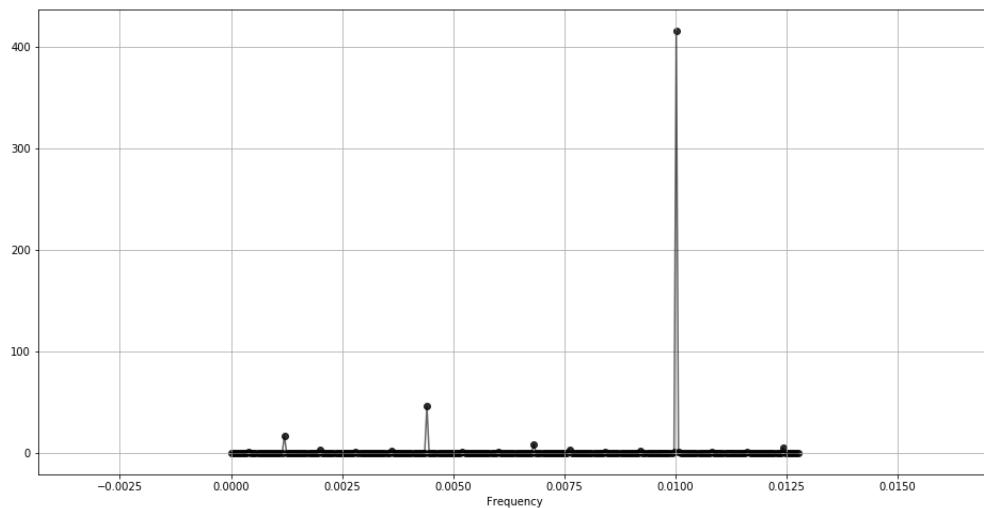


figure 20b. Frequency spectrum of a 10 Hz triangle sine wave, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25.6 \text{ Hz}$ ,  $N = 256$

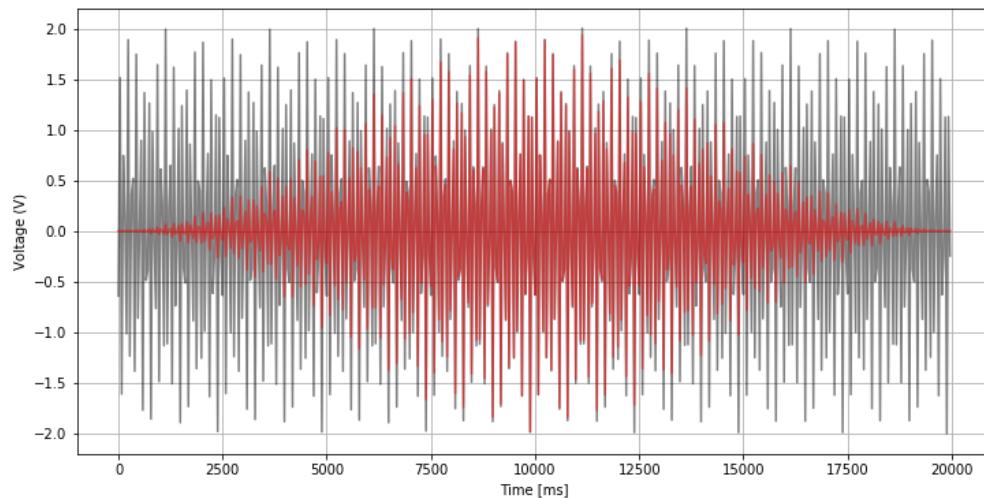


figure 20c. Hanning Window Signal of a 10 Hz triangle sine wave, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25.6 \text{ Hz}$ ,  $N = 512$

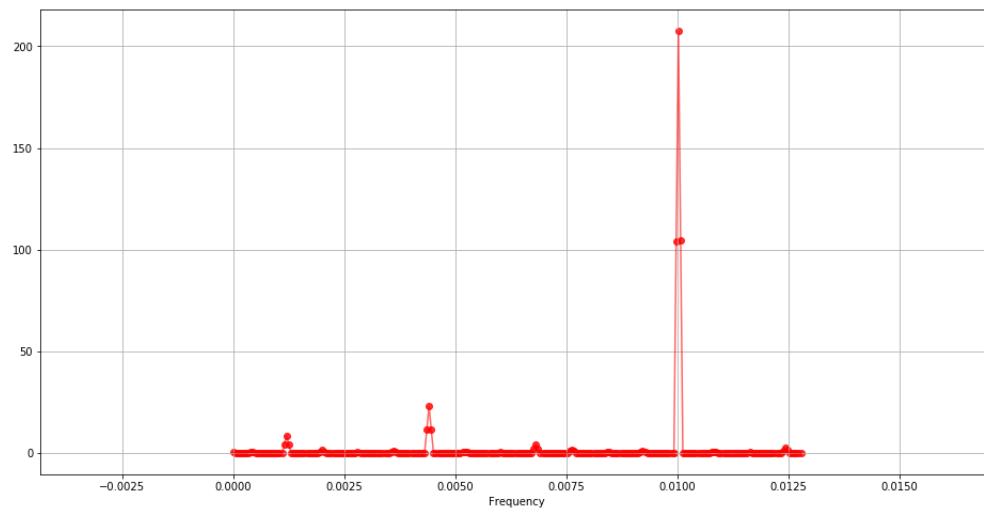


figure 20d. Frequency spectrum of a 10 Hz triangle sine wave after Hanning Windowing, 0V DC Offset, and 2V peak-to-peak amplitude with  $f_s = 25.6 \text{ Hz}$ ,  $N = 512$

## Tables

Table 1

Figure	File Name	Lab Section	Frequency and Waveform	Amplitude	DC Offset	Sampling Rate [Hz]	Number of Samples	Comments
1	Lab2_0.png	IV	20 Hz Sine Wave	1V	0V	100,000	10000 Resolution great; smooth curve	
2	Lab2_1.png	IV	20 Hz Sine Wave	1V	0V	10,000	10000 Resolution great; smooth curve	
3	Lab2_2.png	IV	20 Hz Sine Wave	1V	0V	1,000	10000 Resolution good; becoming grainy	
4	Lab2_3.png	IV	20 Hz Sine Wave	1V	0V	100	10000 Resolution very poor; doesn't appear continuous	
5a	Lab2_4.png	V	10 Hz Sine Wave	2V	0V	200	256 Resolution good	
5b	Lab2_4fft.png	V	10 Hz Sine Wave	2V	0V	200	256 mild leakage	
6a	Lab2_5.png	V	10 Hz Sine Wave	2V	0V	1000	256 Resolution good	
6b	Lab2_5fft.png	V	10 Hz Sine Wave	2V	0V	1000	256 much leakage	
7a	Lab2_6.png	V	10 Hz Sine Wave	2V	0V	25	256 Difficult to see individual waves	
7b	Lab2_6fft.png	V	10 Hz Sine Wave	2V	0V	25	256 moderate leakage	
8a	Lab2_7.png	V	10 Hz Sine Wave	2V	0V	25	512 Difficult to see individual waves	
8b	Lab2_7fft.png	V	10 Hz Sine Wave	2V	0V	25	512 Very little leakage	
9a	Lab2_8.png	V	10 Hz Sine Wave	2V	0V	25.6	512 Difficult to see individual waves	
9b	Lab2_8fft.png	V	10 Hz Sine Wave	2V	0V	25.6	512 No leakage; perfect FFT	
10a	Lab2_9window.png	VI	10 Hz Sine Wave	2V	0V	200	256 Smooth curve	
10b	Lab2_9windowfft.png	VI	10 Hz Sine Wave	2V	0V	200	256 moderate leakage	
11a	Lab2_10window.png	VI	10 Hz Sine Wave	2V	0V	1000	256 Smooth curve	
11b	Lab2_10windowfft.png	VI	10 Hz Sine Wave	2V	0V	1000	256 much leakage	
12a	Lab2_11window.png	VI	10 Hz Sine Wave	2V	0V	25	256 Difficult to see individual waves	
12b	Lab2_11windowfft.png	VI	10 Hz Sine Wave	2V	0V	25	256 moderate leakage	
13a	Lab2_12window.png	VI	10 Hz Sine Wave	2V	0V	25	512 Difficult to see individual waves	
13b	Lab2_12windowfft.png	VI	10 Hz Sine Wave	2V	0V	25	512 moderate leakage	
14a	Lab2_13window.png	VI	10 Hz Sine Wave	2V	0V	25.6	512 Difficult to see individual waves	
14b	Lab2_13windowfft.png	VI	10 Hz Sine Wave	2V	0V	25.6	512 Very little leakage	
15a	Lab2_14window.png	VI	10 Hz Sine Wave	2V	1V	200	256 Smooth curve	
15b	Lab2_14windowfft.png	VI	10 Hz Sine Wave	2V	1V	200	256 much leakage	
16a	Lab2_15window.png	VI	10 Hz Sine Wave	2V	1V	1000	256 smooth curve	
16b	Lab2_15windowfft.png	VI	10 Hz Sine Wave	2V	1V	1000	256 much leakage	
17a	Lab2_16window.png	VI	10 Hz Sine Wave	2V	1V	25	256 Difficult to see individual waves	
17b	Lab2_16windowfft.png	VI	10 Hz Sine Wave	2V	1V	25	256 much leakage	
18a	Lab2_17window.png	VI	10 Hz Sine Wave	2V	1V	25	512 difficult to see individual waves	
18b	Lab2_17windowfft.png	VI	10 Hz Sine Wave	2V	1V	25	512 mild leakage	
19a	Lab2_18window.png	VI	10 Hz Sine Wave	2V	1V	25.6	512 difficult to see individual waves	
19b	Lab2_18windowfft.png	VI	10 Hz Sine Wave	2V	1V	25.6	512 very little leakage	
20a	Lab2_19.png	VI	10 Hz Triangle Sine Wave	2V	0V	25.6	512 difficult to see individual waves	
20b	Lab2_19fft.png	VI	10 Hz Triangle Sine Wave	2V	0V	25.6	512 No leakage; perfect FFT	
20c	Lab2_19window.png	VI	10 Hz Triangle Sine Wave	2V	0V	25.6	512 difficult to see individual waves	
20d	Lab2_19windowfft.png	VI	10 Hz Triangle Sine Wave	2V	0V	25.6	512 very little leakage	

## Discussion Questions

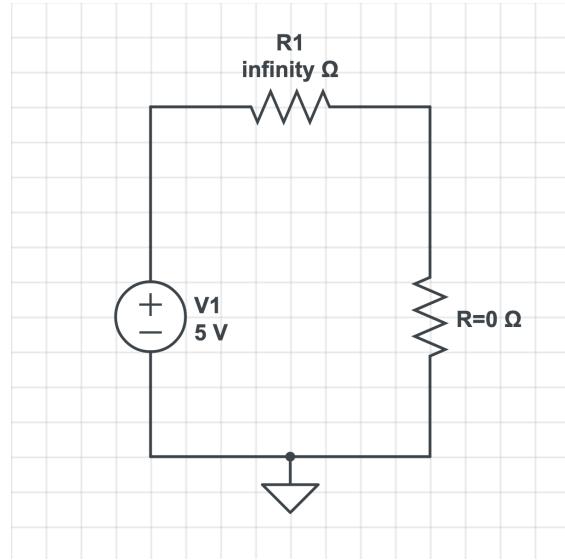
Note: all red text represents information written by Joe and Christian

1. What is the resistance of your coaxial cable? Does it depend on length of the cable? Explain why coaxial cables are often used in communication between instruments and in telecommunication.

The resistance of the coaxial cable used was  $50 \Omega$ . The resistance does depend on the length of the cable  $R = \frac{\rho L}{A}$ . Coaxial cables are commonly used between instruments and in telecommunications because they have low resistance due to the coaxial layering that includes multiple insulating layers.

2. Explain why you need to select the output of your waveform generator to infinite impedance. What would happen if you had it selected for  $50 \Omega$  impedance? Explain this with a simple circuit.

The waveform generator must be set to infinite impedance in order to produce the largest signal output to be interpreted. If the impedance was  $50 \Omega$ , the output signal would be too small to interpret. Consider a simple circuit with one resistor and a voltage source. In order to ensure the waveform generator receives the maximum possible voltage signal, it must have a very high resistance compared to that of the wires. When in series, current through resistors is the same. However, if one resistor is extremely large, the voltage drop across it will be very large and the voltage loss due to the wires will be negligible. Setting the waveform generator's impedance to infinity  $RWG >> RWire$ , this means that  $VWG \approx VSignal$  and  $VWire \approx 0$ . In the photo below, R1 will have a voltage drop of almost the entire source voltage of 5V while the second resistor (representing the wires) will have a voltage drop near 0 V.



**3. Part V, for each of the test cases, calculate the frequency resolution and the energy contained at the signal frequency (10 Hz).**

Comment on:

**A. The resolution of the sine wave.**

The resolution of the sine waves in figures 7a, 8a, and 9a is very poor - likely due to the fact that the sampling frequencies and number of data points yield a large sampling period relative to the number of waveforms. This makes a pattern that looks like a series of closely grouped vertical lines of different heights. The sine waves shown in figures 5a and 6a have high resolutions due to the small sampling period relative to the number of waveforms - due to the sampling frequencies and number of data points.

**B. The width of the spike on the frequency spectrum and the energy contained at 10 Hz vs what you would expect. Explain how the width changes for each condition and what is the source of the observed phenomenon and how it can be corrected.**

The energy contained at the signal frequency is greatest at the lowest frequency resolution. On the frequency spectrum, the width of the spike decreases when the frequency resolution decreases. The extra width of this spike is caused by leakage, and it can be minimized by adjusting the number of samples taken or adjusting the sampling frequency so that the number of samples is an integer multiple of the sampling frequency.

for a sampling frequency of 200 Hz and 256 data points, the frequency resolution = 0.78125 Hz

for a sampling frequency of 1000 Hz and 256 data points, the frequency resolution = 3.90625 Hz

for a sampling frequency of 25 Hz and 256 data points, the frequency resolution = 0.09765625 Hz

for a sampling frequency of 25 Hz and 512 data points, the frequency resolution = 0.048828125 Hz

for a sampling frequency of 25.6 Hz and 512 data points, the frequency resolution = 0.0 Hz

**4. Part V conclusions:**

**A. What is the benefit and drawbacks of increasing the sampling frequency?**

Increasing the sampling frequency literally means increasing the number of data points per second. This means the resolution of your signal will be very good; the frequency resolution will also be increased, however, meaning that there will be leakage on the frequency spectrum.

**B. What is the benefit and drawbacks of increasing the sampling period?**

Increasing the sampling period is beneficial because it allows for more raw data collection. However, displaying all of that data in one graph (in a useful manner) can be quite difficult. Additionally, increasing the sampling period can be an effective way to eliminate leakage if the sampling period is increased to become an integer multiple of the period in the signal.

**C. What is a "perfect FFT"?**

A perfect FFT is a Fast Fourier Transform where the resulting frequency spectrum contains no leakage.

**5. What is the effect of the Hanning windowing on your signal? Does it totally eliminate leakage?**

The Hanning window forces the signal to start and finish at the same point in one measured period. It doesn't completely eliminate leakage, but it allows it to be reduced by having the signal be continuous on multiple periods. Additionally, the Hanning window imposes a cosine wave to the signal causing lower apparent voltages throughout the signal.

**6. What is the effect of windowing when there is a DC offset in addition of the sinusoidal signal? What can you conclude about the mean of a signal on which windowing can be applied? Propose a procedure to apply windowing when the signal has non-zero mean.**

Windowing while there is a DC offset and sinusoidal signal causes a false DC offset at  $f = 0$  and a false amplitude at  $f = \Delta f$ . When windowing is applied, the mean of the signal is incorrect - (from pt. V) it appears to be around 2V while the true mean is 1V (the prescribed offset). In order to

reduce this effect (when a signal has a nonzero mean), the offset in the signal should be removed and then the windowing should be applied, and subsequently the offset should be reapplied to the windowed data to display the true mean.

## Conclusions

Many unique waveforms were produced under the various conditions proposed in this experiment. Figures 1-4 show plain signal data and the following figures are couplets of plain signal (or Hanning signal) data and the corresponding frequency spectrum as a result of performing a Fast Fourier Transform, or FFT, to the signal. The last four figures, 20a-d, show the result of a triangle sine wave being produced by the waveform generator and collecting the resulting signal, Hanning signal, and the frequency spectra for each type of signal. Table 1 describes each wave form in each figure, accompanied by all associated parameters.

In figures 1-4 (Part IV), the signal data from the 100,000 Hz, 10,000 Hz, 1,000 Hz, and 100 Hz sampling frequencies is shown. As one can see, the signal resolution is great for figures 1 & 2, but then appears to get grainy in figure 3, and subsequently figure 4 is quite choppy and doesn't even appear continuous. This is due to the decreasing sampling rate which essentially causes a much lower fidelity, or data points per timestep, making the signal more grainy.

In figures 5-9 (Part V), the spectral accuracy of the system increases from figures 5b - 9b. The defining factors for spectral accuracy are aliasing and leakage. Aliasing does not occur for any of the waveforms because in all cases  $f_s > 2f$ . The spectral accuracy increases over Part V because better combinations of  $f_s$  and  $N$  are being selected. The best way to reduce leakage is to select a sampling frequency and number of data points where  $N$ , the number of data points, is an integer multiple of  $f_s$  (sampling frequency). This is why 25.6 Hz and 512 yield a perfect FFT - 512 is an integer multiple of 25.6  $\Rightarrow 25.6 \cdot 20 = 512$ . As one can see, figure 9b shows the frequency spectrum with no leakage - a perfect FFT.

In Part VI of the experiment, a Hanning Window was applied to the signal. There was essentially no improvement in the signal itself, resolution was the same for each  $N - f_s$  combination. For the frequency spectra, however, the Hanning Window had a slight effect on the leakage. While it doesn't completely eliminate leakage, it allows it to be reduced by having the signal be continuous on multiple periods. Additionally, adding the 1V offset had an unintended effect on the signal: Windowing while there is a DC offset and sinusoidal signal causes a false DC offset at  $f = 0$  and a false amplitude at  $f = \Delta f$ . When windowing is applied, the mean of the signal is incorrect - (from pt. V) it appears to be around 2V while the true mean is 1V (the prescribed offset). In order to reduce this effect (when a signal has a nonzero mean), the offset in the signal should be removed and then the windowing should be applied, and subsequently the offset should be reapplied to the windowed data to display the true mean.

When a triangle waveform was generated and its signal collected,  $f_s$  was chosen to be 25.6 Hz and  $N$  was chosen as 512 as this combination yielded a perfect FFT in previous runs. From the signal data, one can count at least 24 harmonics. Additionally, the Hanning Windowing produced slightly more leakage in the frequency spectrum than did the frequency spectrum with no Hanning Window applied, this comparison can be observed between figures 20b and d.

## Appendices

### Appendix A - NI cDAQ-9174



[Online Manual](https://www.ni.com/documentation/en/compactdaq-chassis/latest/cdaq-9174/overview/)

[User Manual](https://www.ni.com/pdf/manuals/372838e.pdf)

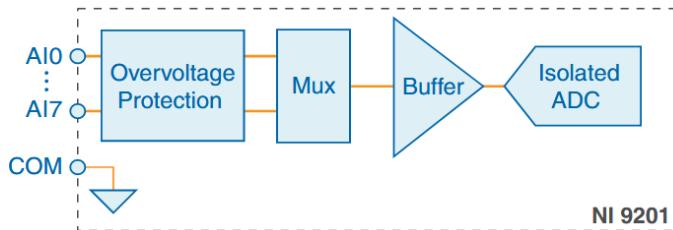
[Specification Sheet](https://www.ni.com/pdf/manuals/374045a.pdf)

### Appendix B - NI 9201



[HTML Manual](https://www.ni.com/documentation/en/c-series-voltage-input-module/latest/9201/overview/)

[Datasheet](https://www.ni.com/pdf/manuals/373783a_02.pdf)

**Signal Level:**  $\pm 10V$ **Channels:** 8 Single-Ended**Max Sample Rate (Single Channel):** 800 kS/s**Max Sample Rate (Scanning):** 500 kS/s**Simultaneous No****ADC Resolution:** 12-Bit**Type of ADC:** Successive approximation register (SAR)**Table 1.** Sample Rate (Aggregate)

Mode	Maximum Sample Rate (R Series Expansion Chassis)	Maximum Sample Rate (All Other Chassis)
Single Channel	475 kS/s	800 kS/s
Scanning	475 kS/s	500 kS/s

Input range  $\pm 10 V$ 

Measurement voltage, channel-to-COM (V)

Minimum  $\pm 10.3$ Typical  $\pm 10.53$ Maximum  $\pm 10.8$ Overvoltage protection, channel-to-COM  $\pm 100 V$ **Table 2.** NI 9201 Accuracy (Excludes Noise)

Measurement Conditions		Percent of Reading (Gain Error)	Percent of Range <sup>1</sup> (Offset Error)
Calibrated	Typical (25 °C, $\pm 5$ °C)	$\pm 0.04\%$	$\pm 0.07\%$
	Maximum (-40 °C to 70 °C)	$\pm 0.25\%$	$\pm 0.25\%$

Stability

Gain drift  $\pm 34 \text{ ppm}/^\circ\text{C}$ Offset drift  $\pm 100 \mu\text{V}/^\circ\text{C}$ 

Input bandwidth (-3 dB) 690 kHz min

Input impedance

Resistance  $1 \text{ M}\Omega$ Capacitance  $5 \text{ pF}$ 

Input noise, code-centered

RMS 0.7 LSBrms

Peak-to-peak 5 LSB

No missing codes 12 bits

DNL -0.9 to 1.5 LSB

INL  $\pm 1.5 \text{ LSB}$ 

Crosstalk, at 10 kHz -75 dB

Settling time, to 1 LSB  $2 \mu\text{s}$ 

MTBF 1,092,512 hours at 25 °C; Bellcore Issue 2, Method 1, Case 3, Limited Part Stress Method

```
In [1]: #Used to answer Discussion question 3
fs = 200
N = 256
fr = fs/N
print('for a sampling frequency of', fs, 'Hz and', N, 'data points, the frequency resolution = ',fr, 'Hz')

fs = 1000
N = 256
fr = fs/N
print('for a sampling frequency of', fs, 'Hz and', N, 'data points, the frequency resolution = ',fr, 'Hz')

fs = 25
N = 256
fr = fs/N
print('for a sampling frequency of', fs, 'Hz and', N, 'data points, the frequency resolution = ',fr, 'Hz')

fs = 25
N = 512
fr = fs/N
print('for a sampling frequency of', fs, 'Hz and', N, 'data points, the frequency resolution = ',fr, 'Hz')

fs = 25.6
N = 512
fr = fs/N
print('for a sampling frequency of', fs, 'Hz and', N, 'data points, the frequency resolution = ',fr, 'Hz')
```

for a sampling frequency of 200 Hz and 256 data points, the frequency resolution = 0.78125 Hz  
for a sampling frequency of 1000 Hz and 256 data points, the frequency resolution = 3.90625 Hz  
for a sampling frequency of 25 Hz and 256 data points, the frequency resolution = 0.09765625 Hz  
for a sampling frequency of 25 Hz and 512 data points, the frequency resolution = 0.048828125 Hz  
for a sampling frequency of 25.6 Hz and 512 data points, the frequency resolution = 0.05 Hz