University of Colorado Boulder
ECEE Department

ECEN 2250 - Introduction to circuits and electronics - Fall 2023
Location: Engineering Center, ECCR 1B40, MWF 2:30PM - 3:20PM
Instructor: Professor Eric Bogatin, Dr. Mona ElHelbawy
Lab #2
Lab Title: Changing voltages: Signals
Date of Experiments: September 20th, 2023
Names: Connor Sorrell

Experiment 1

Introduction:

The overarching purpose of this lab is to set up and get comfortable using the Analog Discovery 3 scope and the waveform generator. This lab is important and useful because it introduces the workhouse instrument that will be used going forward in every lab.

Procedure:

The equipment needed for this lab is a computer with a microUSB port, the Analog Discovery 3 scope, a microUSB cable, and the diligent waveform software. The cable connects the scope to the computer via the microUSB cord. Then, hook up the 1+ connector pin to the waveform connector pin, and the 1- pin to ground. After this, the waves can be seen on the Waveform software.

Experimental Analysis & Discussion:

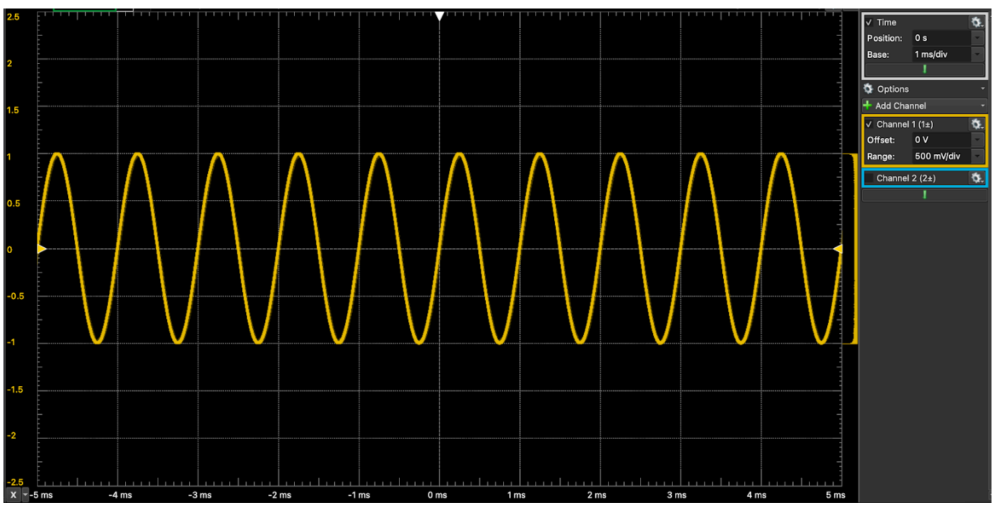


Figure 1.1.0: Screen capture of the sine wave with an amplitude of 1V and period of 1ms.

The sine wave was triggered by moving the yellow triangular slider on the right side of the screen (the trigger level) to a point on the wave in which the oscillations can no longer be seen, and it looks like a standing wave. On this sine wave, the trigger level was 0V, because it was a 1V amplitude sine wave with no offset. Triggering the wave gives a very useful display, where the pulses are all of equal shape and size, making it much easier to read the figures of merit of the wave. Regarding rule #9, hypothesizing data in this lab was extremely important. If a piece of data is not as expected, it is much more smart and efficient to immediately try to find and fix the issue, rather than find out later you did the lab wrong. In this case, I inputted a 1V amplitude, 1ms period, 1 KHz frequency, and 0 V offset. So, it was easy to expect that I would then see a wave with these properties as well. Using rule #9, I hypothesized that the waveform would create a wave with these same properties, and sure enough it did.

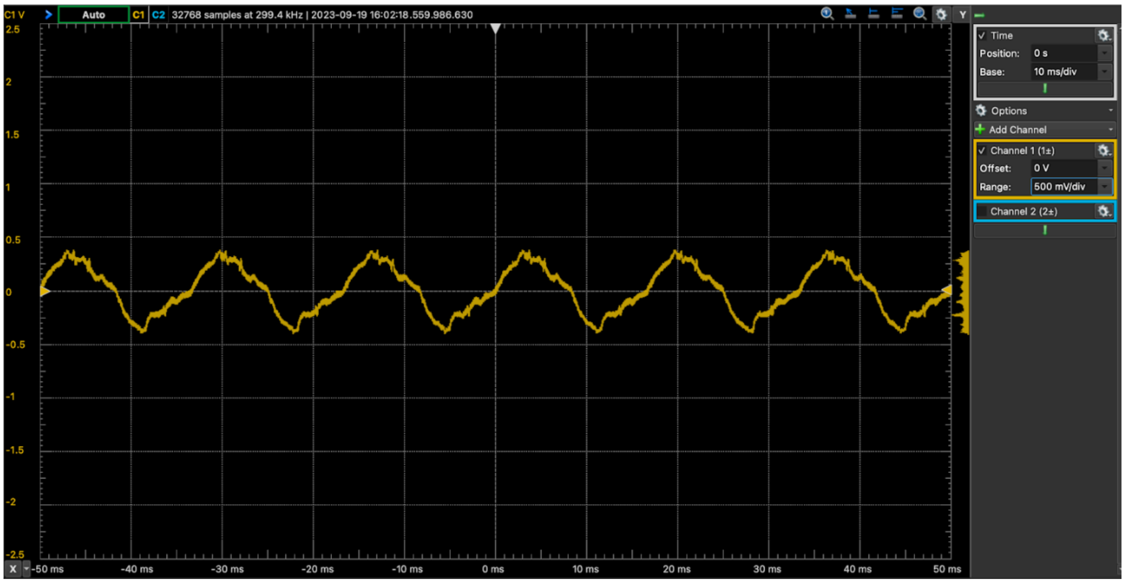


Figure 1.1.1: Screen capture of 60 Hz pickup on Waveform.

I measured this 60Hz wave by leaving the +1 connector pin floating in space, so that it was measuring the frequency of the surrounding air, which is a standard 60Hz in our AC power. This wave had a lot of noise, and depended a lot on the environment. The closer I would move to the wave, the more it would suppress, and other things like my mouse and keyboard near the connector pin made changes to the wave. Because of this, the wave was hard to trigger, because I could not get the wave to stay relatively constant. However, the waveform simulator still shows a rough sine wave, with amplitude of ~4V (this doesn't matter too much because it's dependent on the signal) and a frequency of ~16ms using the eyeball test. Using rule #9, I expected a frequency of 16.6ms, because 1/16.6 = 60Hz. So, the waveform passed rule #9 and therefore looks as expected.

Overall, this experiment is extremely useful because it introduces us to the AD3 scope. It explains what each connection and wire does, which is all brand new knowledge, and explains how to use the waveform software. It is also a necessity to understand each figure of merit and what it represents, not only in an ideal circuit on paper, but also in a real life situation. Being able to physically adjust the voltage, frequency, period, etc, and being able to instantaneously visualize the new waveform really help me understand how each one interacts with another.

Experiment 2

Introduction:

- The purpose of this lab is to learn to convert raw measurements into a few figures of merit using standard repetitive waveforms such as the sine, square, and triangle waves.

1) Experimenting & Discussion:

Using just the eyeball, the period looks to be 1ms. The frequency (1/1ms) = 1Khz. The peak to peak is 2V, and the amplitude and average are 1V. From previous experiments, I know that the RMS of a sine wave is roughly 0.7 times that of the peak amplitude value. Using this, I expect the RMS value to be ~ (0.7)(1) = 0.7 V.

The DMM measured 0.2 mV (DC RMS) before the offset was set to 1V.
After setting the offset to 1V, the DC RMS measured by the DMM was 0.997 V, which is close to 1V as expected. The AC RMS measured by the DMM was 0.689 V. These values make sense and are what we expected. They are only off by roughly 1%, which can be attributed to the small errors within each piece of equipment present in the lab.

The voltage values make sense, 1V average, 2V peak to peak, and 1V amplitude, because this is a 1V sine wave with a +1V offset. The peak to peak value (2V) is always twice the amplitude(1V), and if the wave had no offset, the average voltage would be 0V, so it was easy to confirm these values were correct.

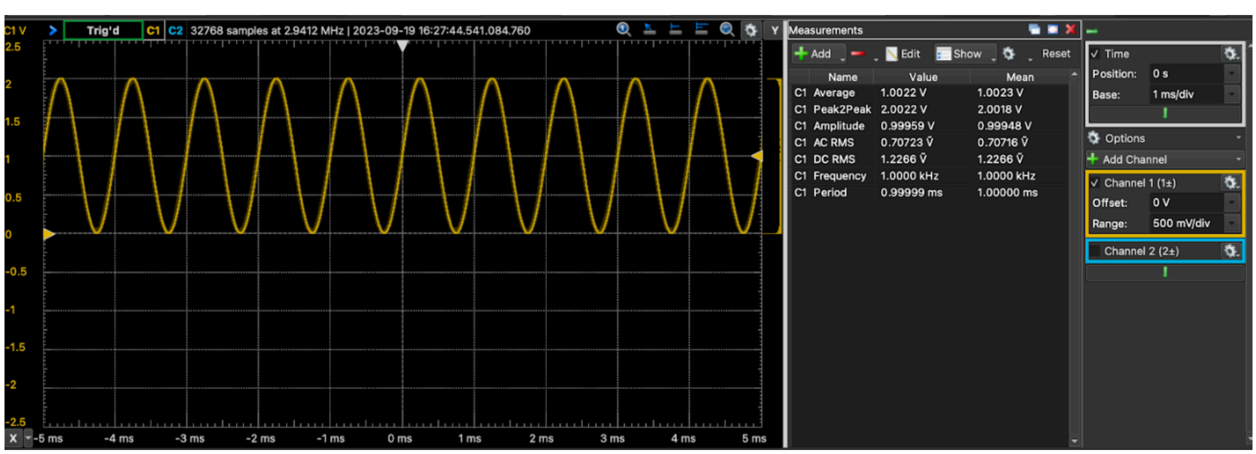


Figure 1.2.0: Screen capture of the sine wave with measurements turned on

Experiment 3

Introduction:

- This experiment's purpose is to apply the skills previously learned about the waveform and DMM, and apply them to measure signals from a real source, the pins on an Arduino board.

1) Procedure:
   Set up the signals on the arduino, make the connections between the arduino, breadboard and AD3 scope, perform the measurements, and then compare with the DMM measurements and interpret the results.

2) Experimental Data & Discussion:

Measuring the DC voltages on the arduino board, each value (vin, 5v, 3.3v) with both the scope and DMM, each voltage value came out as expected with only a slight (~1% error), as normal. These are single ended measurements because we are measuring one potential with a common respect to a common reference voltage (ground).

After setting up the digitalWrite on pin 13, the tone on pin 12, and the PWM signal on pins 10 and 6, I used rule #9 to guess what I might see, and then experiment to find the true value and make sure my understanding is correct.

For pin 13, I expect to see a square wave that is on for 1ms, then off for 2ms. This would mean that the period would be 3ms, and frequency 1/3ms = 330 Hz. After experimenting, the simulation read 3.0147ms for the period, and 331.71Hz frequency.

For pin 12, I expect to see a square wave that is on for 1ms, then off for 1ms. This would mean that the period would be 2ms, and frequency 1/2ms = 500 Hz. After experimenting, the simulation read 2.0352ms for the period, and 491.36Hz frequency.

All measurements matched up with rule #9 as expected.

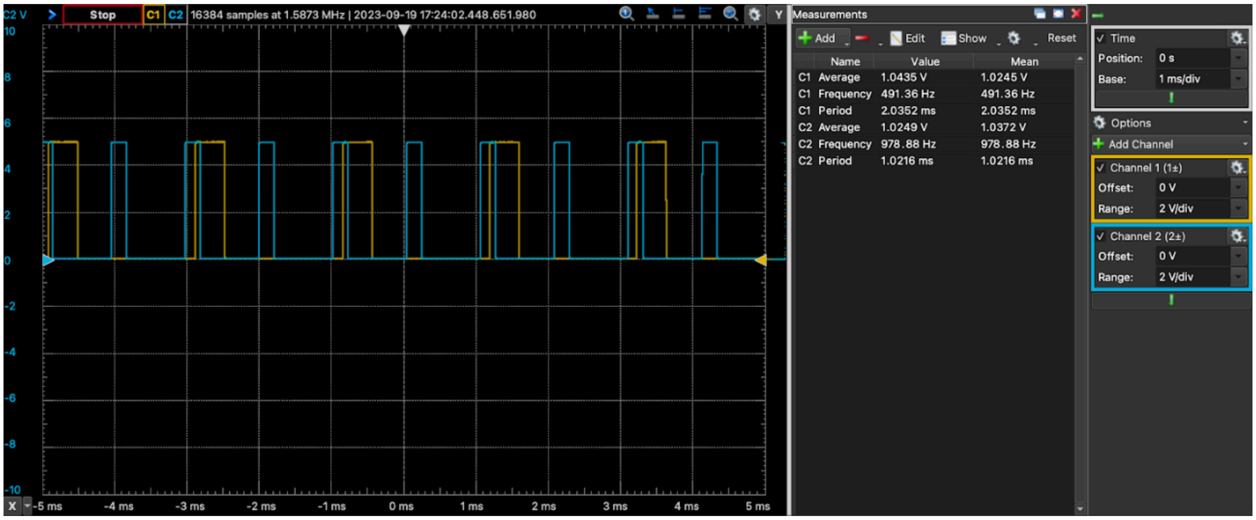


Figure 1.3.0: Waveform showing the PWM signal on channel 1 (pin 10) and channel 2 (pin 6)

With channel 1 plugged into pin 10, and channel 2 plugged into pin 6, I expect to see two waves oscillating, one with double the period of the other. I expect to see this because of the source code, where after high voltage, the delay is 1 ms, and after low voltage, the delay is 2ms. This is the reason for the two different waves coming from the two pins.
After conducting the experiment, my prediction was correct. Channel 1 (pin 10) has a period of ~2ms, while channel 2's period is ~1ms. The two PWM signals also have a different frequency, which comes with having different periods (frequency = 1/period). Essentially, the waveform from pin 6 is oscillating twice as fast.

I then made a slight change to the source code, changing the duty cycle. The duty cycle is the ratio of time a circuit is on compared to the time the circuit is off. I changed the duty cycle so that there was a 10ms "off" period. Using rule #9, I expect the wave to oscillate slower and have a larger period, lower frequency. After experimentation, the increase in duty cycle did in fact raise the period of the wave, and lower the frequency.

With a LED and a 1k resistor in series, I expected the voltage across the LED to be around 1.6V and the light to be turning off and on at around 500Hz. After experimenting with the waveform, the frequency was in fact 500Hz, but the true voltage across the LED was ~1.92V.
I then predicted the voltage across the resistor and LED in series to be 5V, because that is the supplied voltage. After measuring, the voltage across both components came out to be ~4.986V. Both of the previous measurements are single ended, meaning the voltage drop was measured with respect to a common ground.
Using a double-ended measurement, (the +1 and -1 connectors), I then measured the voltage across only the resistor. Using rule #9, I expected this value to be ~3.1V, because it must add up to 5V when summed with the voltage across the LED, which was previously measured to be ~1.9V. After doing the simulation and adding the measurements, the results made sense. The real voltage across the resistor was ~3.205V. It is slightly off when adding up to 5V, but that can be explained by small inaccuracies and imperfections in the equipment.

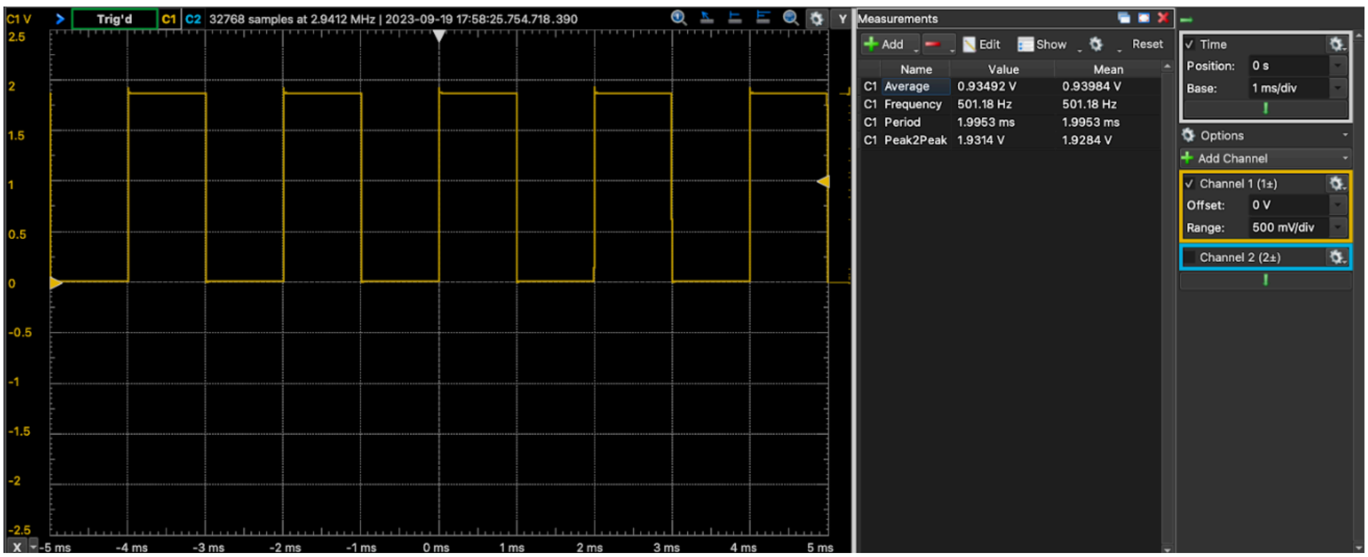
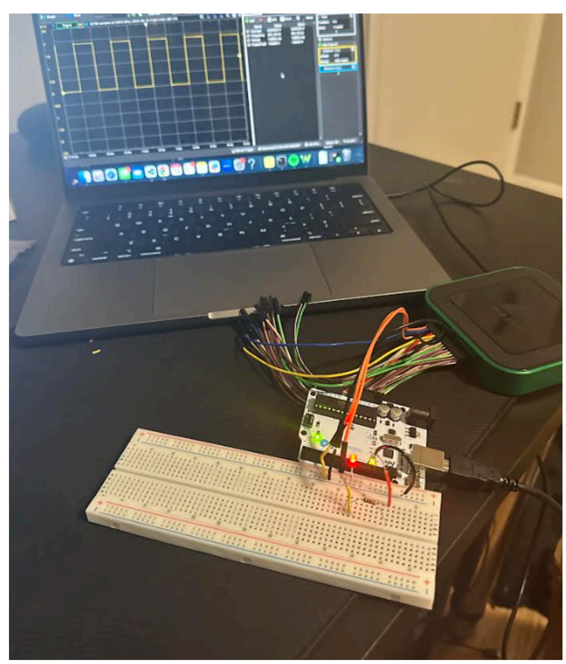


Figure 1.3.1: Scope screen capture of voltage across the LED with measurements

Figure 1.3.2: picture of circuit with Arduino, LED, resistor and scope with connections



Experiment 4

Introduction:

- The purpose of this lab is to measure a real signal with the scope, bring it into the simulation environment, and compare how close the real measurement is from the ideal, simulated result. This is important to gain experience building a model of an ideal signal and hacking it to match the simulation to the target result.

1) Procedure:
   Firstly, set up the Wavegen for a sine wave. This requires the waveform software, and the analog discovery scope. Then, measure the waveform with the scope, and export the measured voltage and time into an excel sheet. Plot the waveform for the measured values into a scatter plot. Then, create a model to calculate voltage of an ideal sine wave that is dependent on amplitude, phase, period, and offset. Once the model is complete, plot another scatterplot using the model's values of voltage vs time. Finally, plot the residual error difference between the measured values and the modeled values, and then analyze and interpret the difference between the two.

2) Experimental Data

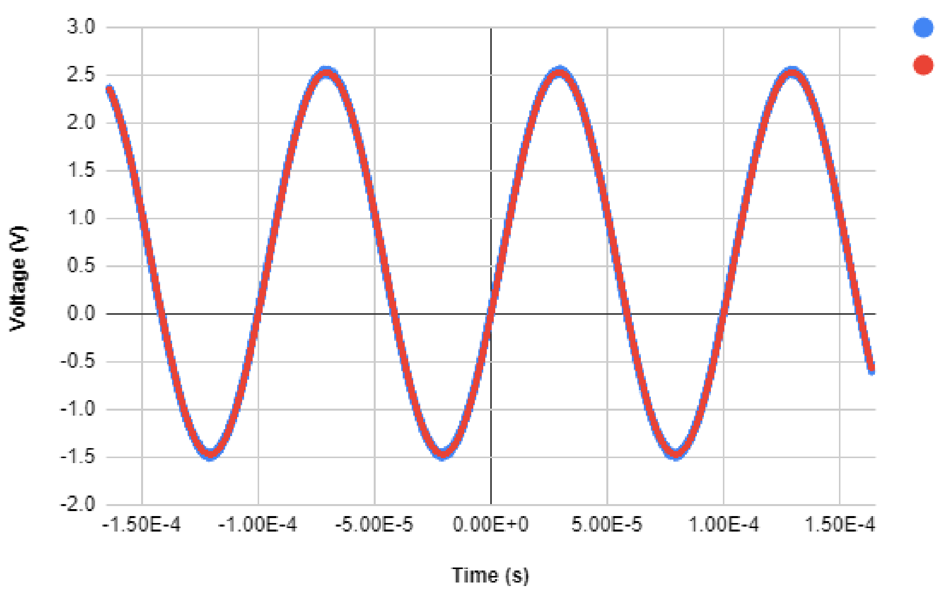


Figure 1.4.0: Scatter Plot of Voltage (V) vs Time (sec) showcasing both the modeled and actual values.

Red plot = Modeled Values
Blue plot = Measured Values

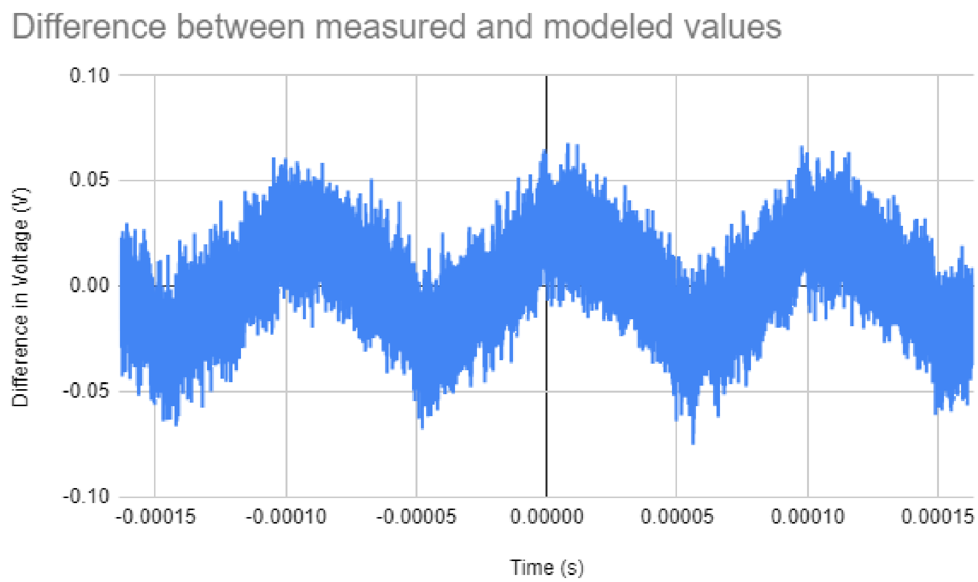


Figure 1.4.1: Line chart showcasing the error between the modeled and measured values

3) <u>Discussion:</u>

From the wavegen, I was supplied with over 32,700 different measurements of voltage vs. time. This means that the line formed in the scatterplot was extremely precise. After pasting the data into excel and forming the plot for both the modeled and measured voltage values, it came time to hack the model to get better agreement. When I first put both curves on top of eachother, they looked to be almost matching from a glance. But once I zoomed in, their differences were more than apparent. I then started adjusting the phase to get both curves to lay on top of each other as much as possible. From there, I adjusted the amplitude, aiming to match the slopes of the curves with each other. Once the waves looked very similar, I slightly tweaked the offset of the modeled curve so that both curves' peak amplitude values matched.

Plotting the graph of the difference between the curves, I noticed that my model was roughly $\mp$ 0.06 Volts off. The signal is 2V, so I am roughly 3% off. I am unhappy with this error, because I know it is possible to have a much smaller difference between the model and the actual values. However, with over 32,700 values, it took over a few minutes to see the graph change every time I changed a value. Because of this, it became really difficult to inch my error down and down until a better match was found.

My final values for the model were as follows:

Amplitude: 2.02V

Period: 0.0001 s

Offset: 0.53V

Phase: -0.285 radians

Though my model was ~3% off from the true values, I know I can improve on my model next time by using a quicker computer, which will allow me to do more trial and error changing up the dependent variables in the model.
This lab illustrates that with just some algebra, a model can be made extremely accurately, which is impressive considering my true plot has over 32,700 data values. This goes to show that while it is necessary to see simulations, analyze them, and get true data, a model can be almost as good and it takes a small percentage of time vs. doing the experiment. With this being said, it is helpful to see the simulation to truly understand what is going on versus just estimating. It is also helpful to compare the two, because without the simulation, I would not be able to know how far off my model is.