Physics 4AL

Lab 7

TA: Manoj Aravind Sankar

Unit 2 Lab Report - Motion

Disha Zambani

Hannah Wang

Ethan Wong

I. Objective

In this experiment we worked as a team in order to test the effects of friction on an air glider's velocity and acceleration. We accomplished this by using a glider, masses, an air track, and an Arduino with ultrasound sensor. We hypothesized that the addition of friction to the system would significantly decrease the distance the glider traveled on the air track, decrease its velocity, and increase its deceleration.

II. Methods

In this experiment, we used an ultrasound sensor mounted to an Arduino to measure the displacement of a glider with a reflecting surface on an air track. The Arduino setup was attached to one end of an air track and the reflecting surface was attached to an air track glider. The ultrasound sensor emitted ultrasound chirps from a transceiver and these chirps reflected off of the reflecting surface before returning to the detector. The duration of time between when the chirp was emitted and when the chirp was detected was used to find the distance between the sensor and the reflecting surface.

Data was collected for four different scenarios. In the first scenario, the glider (191g) was moving at a constant velocity. We used a level airtrack for this. We positioned the glider near the ultrasound sensor, reset the Arduino, and gave the glider an initial push away from the sensor. In the second scenario, we used the same set up as in the first except we weighed down the glider with additional mass (100g) in order to introduce friction. In the third scenario, the glider was moving with constant acceleration. For this, we propped up the end of the track that the ultrasound sensor was attached to to create a slope with a 3 degree angle so that the glider would accelerate away from the sensor when released. In the fourth scenario, we used the same setup as the third except we weighed down the glider with additional mass (100g) to introduce friction. Each trial was run three times and the best of the three trials was used for data analysis.

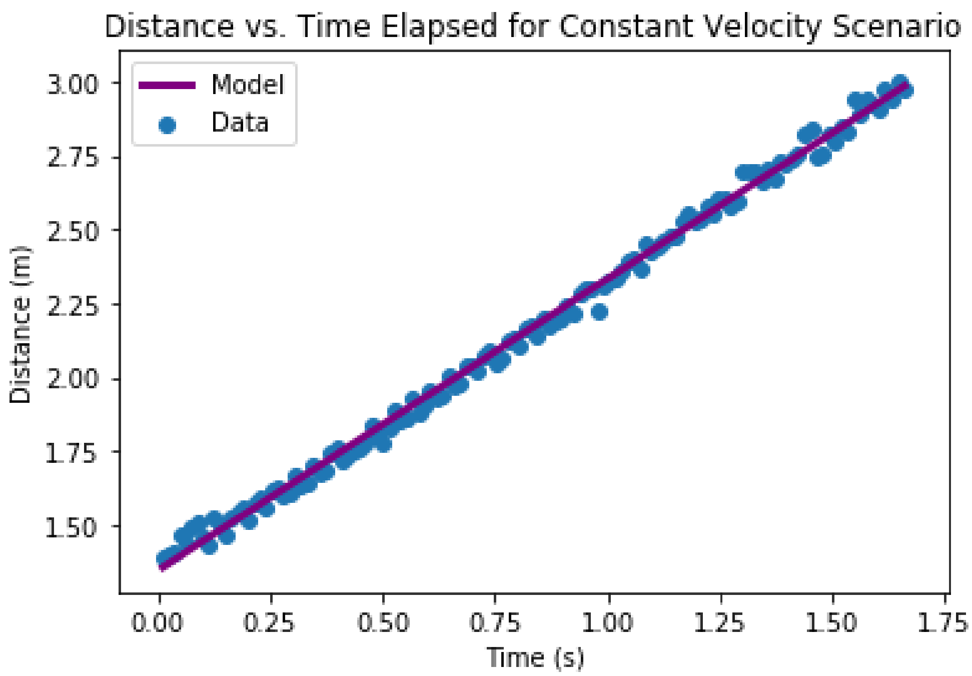
III. Analysis

In order to analyse the data that we collected, we used Python and its matplotlib library to plot the data collected and the numpy library to further analyze the plots. We were able to find values for the velocity and acceleration of the different conditions imposed upon the air track and glider system with these tools. We collected data from the Arduino serial monitor and saved it into a .csv file for each trial so that we would be able to import them into Python and create accurate graphs and conduct polyfit analysis. From the Arduino we only were able to obtain the total elapsed time of the trial and the ultrasound delay as the glider moved farther and farther away from the ultrasound sensor for each trial. These quantities are both in time (milliseconds for the first and microseconds for the former), so in order to depict more sensible relationships, we needed to conduct some preliminary calculations and unit conversions (cm to m, ms to s, µs to s, etc.) to generate the plots and, in turn, relationships between the quantities analyzed.

The relationship between ultrasound delay and distance between the ultrasound sensor and the reflecting surface can be modeled by the equation $\Delta t = \frac{2x}{v_s}$, where v_s is the speed of sound (344 m/s) and Δt is the ultrasound delay in seconds. Using this relationship, we were able to calculate and plot the position of the glider as a function of time elapsed for each scenario tested.

IV. Results

Scenario 1: Constant Velocity



R^2=0.9959365591372498

FIGURE 1 Distance vs Time Elapsed for Constant Velocity Scenario - The chart above shows the distance between the ultrasound sensor and the reflecting surface (y-axis) as a function of time elapsed (x-axis) after the reflecting surface was pushed away from the ultrasound sensor with a constant velocity (no friction). The distance was found by using the proportional relationship between ultrasound delay and distance. The data produces a linear curve with equation $y = 0.0987585x + 0.1344519$ and a slope equal to the velocity of the glider. The curve has a coefficient of determination of about 0.996.

As is shown in Figure 1, the plot for constant velocity and no friction results in a graph with measured distance on the y-axis and elapsed time on the x-axis. The graph was linear, as expected, because a linear change in the displacement equals a constant velocity.

When the glider is pushed away from the ultrasound sensor on a level frictionless surface, it has a constant velocity. The equation of the best fit curve in Figure 1 can be determined using the results array from the polyfit. It is

$$y = 0.0987585x + 0.1344519$$

where y is the distance of the glider in meters and x is the time elapsed in seconds. The best fit line is a good fit for the data as evidenced by its coefficient of determination. It tells us that that about 99.6% of the variability in distance is caused by a change in time.

This equation corresponds to the theoretical position of an object moving with constant velocity, which can be modeled with the linear function

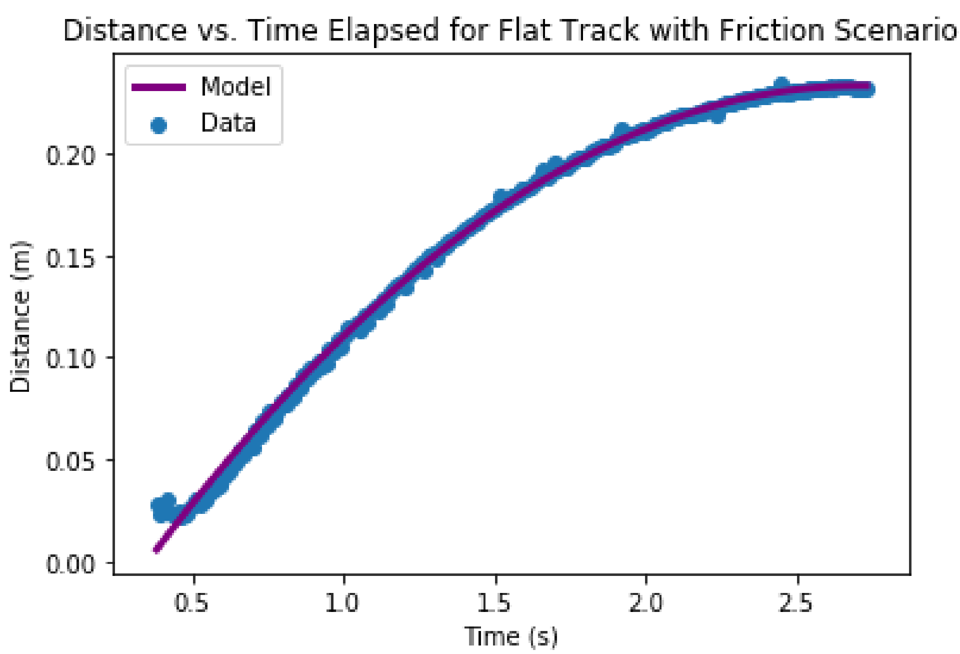
$$x = vt + x_0$$

where x is the distance of the glider from the ultrasound sensor, v is the velocity of the glider, t is the time elapsed, and x_0 is the initial position of the glider.

By combining the information above, the velocity of the glider is found to be equal to the slope of the graph. Therefore, the velocity of the glider is approximately 0.0988 m/s.

Scenario 2: Flat Track with Friction

In scenarios involving a force of friction acting upon the glider, the distance over time graph was seen as quadratic. This is because the friction force introduced a negative acceleration to the system. Since the plot was quadratic, we used a polyfit of the second order. We added a mass of 100 grams to our already 191 gram glider to introduce said friction.



R^2=0.9974016582816866

FIGURE 2 Distance vs Time Elapsed for Flat Track with Friction Scenario - The chart above shows the distance between the ultrasound sensor and the reflecting surface (y-axis) as a function of time elapsed (x-axis) in the flat track with friction scenario. The data produces a second degree curve with equation $y = -0.041716x^2 + 0.226565x - 0.074626$. The slope of the curve is decreasing, indicating that there was a negative acceleration acting on the glider. The curve has a coefficient of determination of about 0.997.

As is shown in Figure 2, the plot for velocity with friction introduced results in a graph that is decreasing and quadratic in form. The velocity eventually becomes zero evidenced by the leveling off of the distance plot in the graph. This is due to the fact that the glider has the extra force of friction acting on it which means the glider will eventually come to a stop in its motion if we kept it going after we took the sample data. We were not able to demonstrate this because the ultrasound readings taken would be highly inaccurate some distance from the arduino to the glider.

The equation of the best fit curve in Figure 2 is

$$y = -0.041716x^2 + 0.226565x - 0.074626$$

The best fit curve seems to be a very good fit for the data as the plot and curve are almost identical. However, since the curve is nonlinear, we cannot look at the coefficient of determination as a metric for goodness of fit. A more complex analysis would have to be done to confirm the goodness of fit of the best fit curve using nonlinear regression.

This equation corresponds to the theoretical position of an object moving with constant acceleration, which can be modeled with the kinematic equation:

$$x = \frac{1}{2}at^2 + v_0 t + x_0$$

Therefore, the acceleration is approximately -0.0834 m/s^2 and the initial velocity imposed was approximately 0.227 m/s.

The coefficient of friction can be found using the acceleration. Friction is the only force acting on the glider and can be defined as

$$F_f = \mu F_N = \mu mg$$

where F_f is the force of friction in Newtons, μ is the coefficient of friction, and F_N is the normal force in Newtons. The normal force is equal in magnitude to the force of gravity, which is equal to mass times acceleration due to gravity (mg).

By applying Newton’s second law, we find that

$$F_f = \mu mg = ma\,.$$

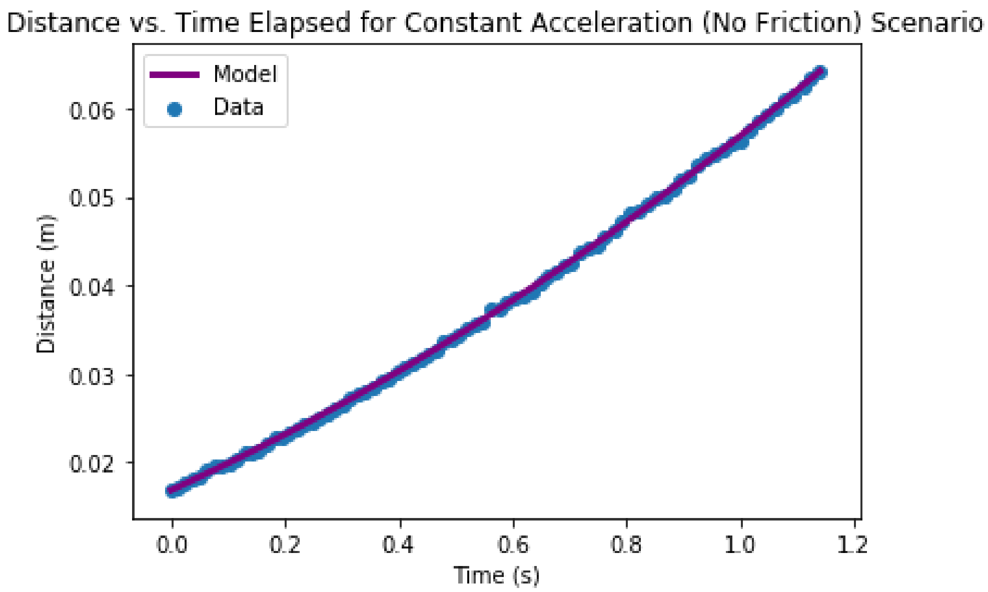
Therefore, the coefficient of friction can be found with the equation

$$\mu = \frac{a}{g}$$

For this trial of this scenario, we find that the coefficient of friction is .00851.

<u>Scenario 3: Constant Acceleration (inclined air-track)</u>

The analysis method used for Scenario 2 can be extended to Scenarios 3 and 4, as they also involve constant acceleration.



R^2=0.9997842597192487

FIGURE 3 Distance vs. Time Elapsed for Constant Acceleration Scenario - The chart above shows the distance between the ultrasound sensor and the reflecting surface (y-axis) as a function of time elapsed (x-axis) in the constant acceleration with no friction scenario. The data produces a second degree curve with equation $y = 0.0106608x^2 + 0.0293041x + 0.0168964$. The slope of the curve is increasing, indicating that there was a positive acceleration acting on the glider. The curve has a coefficient of determination of about 0.999.

As is shown in Figure 3, the plot for acceleration without friction results in a graph that is increasing and quadratic in form. This makes sense because as the glider moves farther down the incline created, it will pick up speed, as it is moving in the direction of gravity and the gravitational force is the only force acting on the glider. Our plot looks quite linear but if we were able to take more data in a more extended period of time, we would be able to see that the plot is indeed quadratic as said. We were not able to demonstrate this because the ultrasound readings taken would be highly inaccurate some distance from the arduino to the glider.

The equation of the best fit curve in Figure 3 is

$$y = 0.0106608x^2 + 0.0293041x + 0.0168964$$

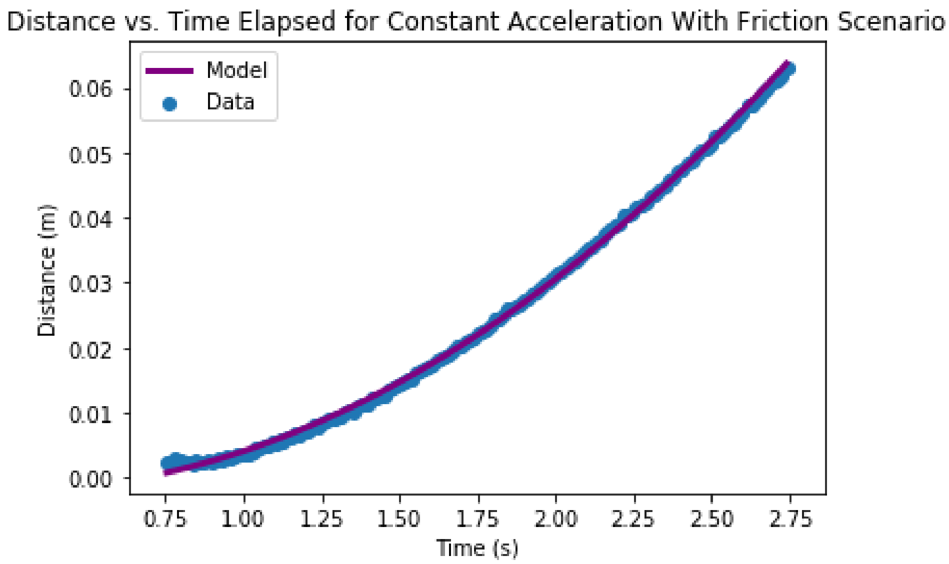
Using a similar method of analysis in Scenario 2, the acceleration is approximately 0.0213 m/s^2 and the initial velocity imposed was approximately 0.0293 m/s.

The best fit curve seems to be a very good fit for the data as the plot and curve are almost identical. However, since the curve is nonlinear, we cannot look at the coefficient of determination as a metric for goodness of fit. A more complex analysis would have to be done to confirm the goodness of fit of the best fit curve using nonlinear regression.

The expected acceleration for this scenario is 0.513 m/s^2. This value was calculated using the procedure described in the Data section below under the Constant acceleration with no friction table. We can see that the expected acceleration is much larger than the obtained value for acceleration in this scenario. As stated in the Data section under the Constant acceleration with no friction table this must have been due to some collection error. We also see that the relationship (increasing, quadratic) obtained for acceleration and the setup of the glider is as expected, so we can dismiss the error due to some insignificant human error. If the trials were to be redone, we would make sure to identify and remove the error to produce more accurate results.

Scenario 4: Constant Acceleration (Inclined Track) with Friction

Like in Scenario 2, to introduce friction into the glider system, we added 100 grams of mass to the 191 gram glider. However, unlike in Scenario 2, the surface is now an incline, so the effect that the frictional force has on the glider system will differ.



R^2=0.9991088501729468

FIGURE 4 Distance vs Time Elapsed for Constant Acceleration with Friction Scenario - The chart above shows the distance between the ultrasound sensor and the reflecting surface (y-axis) as a function of time elapsed (x-axis) in the constant acceleration with friction scenario. The data produces a second degree curve with equation $y = 0.01051036x^2 - 0.00503341x - 0.0014087$. The slope of the curve is increasing, indicating that there was a positive acceleration acting on the glider. The curve has a coefficient of determination of about 0.999.

As is shown in Figure 4, the plot for acceleration with friction results in a graph that is increasing and quadratic in form. This might not make sense at first since adding friction to the system should cause the glider to slow down. However, after more analysis, the negative velocity and positive acceleration obtained from the Python analysis shows that the glider is indeed slowing down.

The equation of the best fit curve in Figure 4 is

$$y = 0.01051036x^2 - 0.00503341x - 0.0014087$$

Using a similar method of analysis in Scenario 2, the acceleration is approximately 0.0210 m/s^2 and the initial velocity imposed was approximately 0.00503 m/s.

The best fit curve seems to be a very good fit for the data as the plot and curve are almost identical. However, since the curve is nonlinear, we cannot look at the coefficient of determination as a metric for goodness of fit. A more complex analysis would have to be done to confirm the goodness of fit of the best fit curve using nonlinear regression.

The expected acceleration for this scenario is 0.513 m/s^2. This value was calculated using the procedure described in the Data section below under the Constant acceleration with no friction table. We can see that the expected acceleration is much larger than the obtained value for acceleration in this scenario. As stated in the Data section under the Constant acceleration with no friction table this must have been due to some collection error. We also see that the relationship between the positive acceleration and negative velocity indicates the glider slowing down, which is expected when friction is opposing motion in the system, so we can dismiss the error due to some insignificant human error. If the trials were to be redone, we would make sure to identify and remove the error to produce more accurate results.

V. Data

1. Parameter fits for constant velocity:
   Each of the individual velocities and initial positions were found by taking the experimental data and plotting and analyzing them using the Python tools. Each time we conducted a trial, the force with which we pushed the cart varied and this the velocity on the glider also varied from trial to trial.

Initial position (m)	Velocity (m/s)
0.134	0.0988
0.145	0.0865
0.124	0.126

2. Flat with friction:
   Each of the individual velocities and accelerations were found by taking the experimental data and plotting and analyzing them using the Python tools. The coefficient of friction was calculated using Newton's third law. The net force was found using the fact that it is equal to the mass times the acceleration of the system (the glider). From this, we know that the friction force is the only force acting on the glider during the trial. From this we can say that the net force is equal to the friction force. Since the friction force is equal to the coefficient of friction times the normal force, we can solve for the coefficient of friction by setting it equal to the mass times the acceleration of the system over the normal force where the normal force is just the mass of the system times the acceleration

due to gravity (since there is no incline, the normal force is the same as the gravitational force). A simplified explanation of this process can be found in the results section with the proper equations. The coefficients obtained were all quite similar as shown in the table below:

Initial position (m)	Velocity (m/s)	Acceleration (m/s^2)	Coefficient of friction
- 0.0821	0.567	- 0.0839	.00856
- 0.0746	0.227	- 0.0834	.00851
- 0.0534	0.332	- 0.0820	.00837

3. Constant acceleration with no friction:
   Each of the individual velocities and accelerations were found by taking the experimental data and plotting and analyzing them using the Python tools. The error column in the table below was found using the provided equation
   (theoretical acceleration - measured acceleration) / theoretical acceleration
   where the theoretical acceleration was calculated using Newton's third law. When attempting this calculation, we must first take into consideration the fact that the system is now an incline. We know that the equation for the net force on a mass on an incline is
   $$\sum F = (sin - \mu cos)mg$$
   however, the coefficient of friction here is 0, so we can say that the net force is equal to $mgsin$ which is also equal to mass times acceleration. We can manipulate the equation obtained by this relationship to say that the theoretical acceleration is equal to $gsin$. For all three trials we chose to use an angle of 3 degrees, so the calculated theoretical acceleration comes out to be 0.513 m/s^2. We can see that the error is rather high for all data collected which must have been due to some collection error. We also see that the relationship (increasing, quadratic) obtained for acceleration and the setup of the glider is as expected, so we can dismiss the error due to some insignificant human error. If the trials were to be redone, we would make sure to identify and remove the error to produce more accurate results.

Initial position (m)	Velocity (m/s)	Acceleration (m/s^2)	Error
0.0169	0.0293	0.0213	95.8%
0.0158	0.0354	0.0235	95.4%
0.0152	0.0512	0.0267	94.8%

4. Constant acceleration with friction
   Each of the individual velocities and accelerations were found by taking the experimental data and plotting and analyzing them using the Python tools. The expected acceleration with no friction was calculated using the same method as the previous section. Once

again, we see that the actual acceleration we obtained versus the expected acceleration are magnitudes off which must have been due to some collection error. We also see that the relationship (increasing, quadratic) obtained for acceleration and the setup of the glider is as expected, so we can dismiss the error due to some insignificant human error. If the trials were to be redone, we would identify and remove the error.

Initial position (m)	Velocity (m/s)	Acceleration (m/s^2)	Expected acceleration with no friciton (m/s^2)
- 0.00141	- 0.00503	0.0210	0.5132
- 0.00190	- 0.00456	0.0202	0.5132
- 0.00132	- 0.00521	0.0223	0.5132

VI. Conclusion

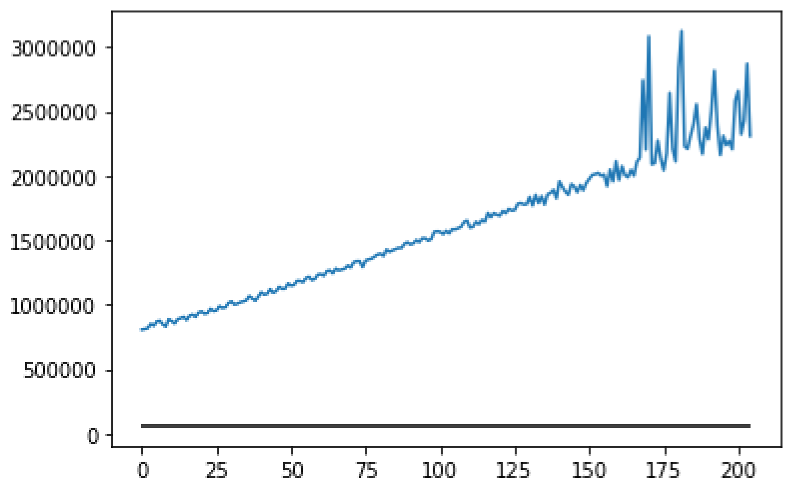
During the course of this lab experiment we were able to confirm the properties of motion under different circumstances: under constant velocity (with and without friction) as well as under constant acceleration (with and without friction). As expected, when there was no friction introduced into the system, and no incline, the glider would move at the same initial velocity imposed onto it in each trial. Since there were no other forces acting on the glider during the trial, the glider had no acceleration. When friction was added to the air glider, we could see that it introduced a force (frictional) into the system. The force, which we know to be acting in the opposite direction of motion, caused the glider to decelerate, lose velocity and eventually (not shown in the experiment) stop. When friction was removed and the surface was raised to an incline, the glider accelerated downwards because the only force acting on the glider was that of gravity. When friction was added back in, but this time on the incline, the glider is definitely slowing down as evidenced by the velocity and acceleration being in opposite directions. This is due to the frictional force opposing the motion of the glider. In Scenario 4, we may be able to say that the force of gravity exceeds the frictional force and that is why the curve is increasing. Further experimentation can be done to compare frictional forces on gliders of different mass or material.

VII. Appendix

Code used to analyze constant velocity dataset:

```
from google.colab import drive
drive.mount('drive')
import numpy as np
read_in_array = np.loadtxt('/content/drive/My Drive/Physics 4AL/Lab Report 1/v1.csv',delimiter=',')
```

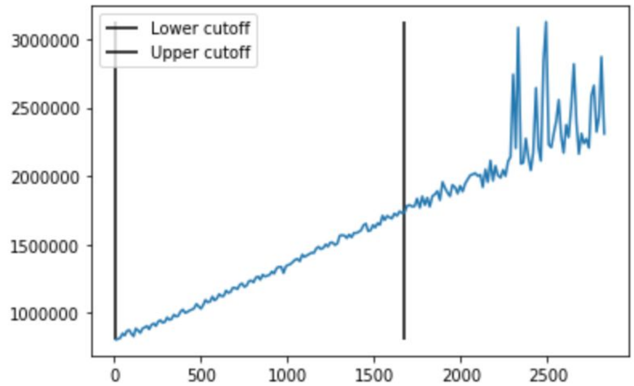
```
time_millis = read_in_array[:,0]

delay = read_in_array[:,1]
import matplotlib.pyplot as plt

array_indices = np.arange(len(delay))

# Plot array indices vs delay
maximum_delay = 60000

# Plotting a horizontal line for indices
plt.hlines(maximum_delay, np.min(array_indices), np.max(array_indices))
plt.plot(array_indices, delay)
```



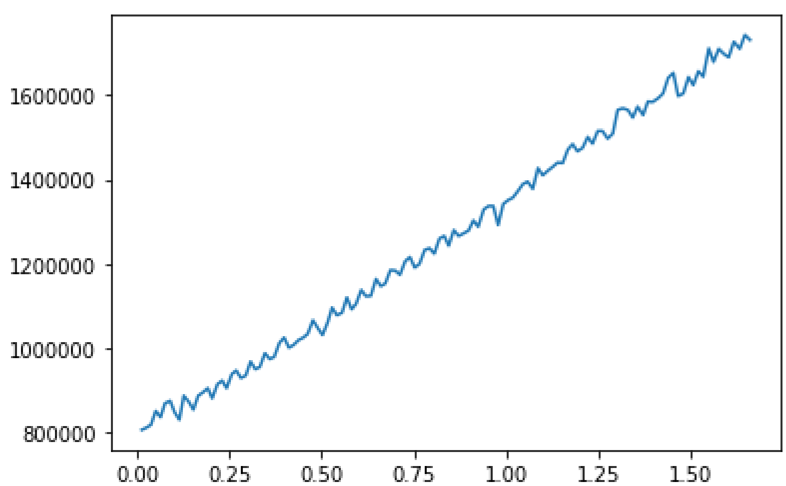

```
lower_index = 0
lower_time_limit = time_millis[lower_index]
print('The lower time cutoff is ' + str(lower_time_limit))

upper_index = 125
upper_time_limit = time_millis[upper_index]
print('The upper time cutoff is ' + str(upper_time_limit))

# These are the minimum and maximum delay plotted
lower_plot_limit = np.min(delay)
upper_plot_limit = np.max(delay)
```

```
# Plot a vertical line at the lower time limit from the bottom to the top
of the plot
plt.vlines(lower_time_limit, lower_plot_limit, upper_plot_limit, label =
'Lower cutoff')
plt.vlines(upper_time_limit, lower_plot_limit, upper_plot_limit, label =
'Upper cutoff')
plt.plot(time_millis, delay)
plt.legend()
```

```
The lower time cutoff is 12.0
The upper time cutoff is 1676.0
<matplotlib.legend.Legend at 0x7fa3b2295780>
```

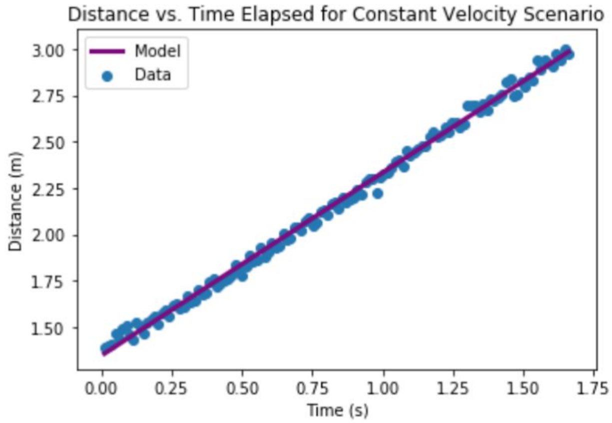



```
clipped_time = 0.001*time_millis[lower_index:upper_index]
clipped_delay = delay[lower_index:upper_index]

distance = 0.5*344*0.000001*clipped_delay[lower_index:upper_index]

plt.plot(clipped_time, clipped_delay)
```



```
coeffs = np.polyfit(clipped_time, distance, 1)
x_array = np.linspace(np.min(clipped_time), np.max(clipped_time), 100)
y_array = coeffs[0] * x_array + coeffs[1]
results = {}
print(coeffs)
results['polynomial'] = coeffs.tolist()
p = np.poly1d(coeffs)
yhat = p(clipped_time)
ybar = np.sum(distance)/len(distance)
ssreg = np.sum((yhat-ybar)**2)
sstot = np.sum((distance-ybar)**2)
results['determination'] = ssreg / sstot
print(results)

plt.plot(x_array, y_array, label = 'Model', color = 'purple', linewidth =
3)
plt.scatter(clipped_time, distance, label = 'Data')



# Your x and y labels here
plt.xlabel('Time (s)')
plt.ylabel('Distance (m)')

#Title
plt.title('Distance vs. Time Elapsed for Constant Velocity Scenario')
```

```
[0.98758504 1.34451904]
{'polynomial': [0.9875850374449981, 1.344519043702925], 'determination': 0.9959365591372498}
Text(0.5, 1.0, 'Distance vs. Time Elapsed for Constant Velocity Scenario')
```



Similar code was used to analyze all other datasets.