Conservation of Energy in a Cart-Pendulum System Experiencing Elastic Collisions

Clyde Villacrusis, Nathan Joshua, Wendy Mapaye

Physics 4AL Lab 3, Fall Quarter 2023
Table 4

Abstract

The goal of this project was to confirm the existence of conservation of energy and momentum in a controlled experiment. Attempted using a cart-pendulum system made of two arduinos – one attached to a moving cart and the other hung by a string from a stand to act as a pendulum. The system underwent two successive collisions and a theoretical prediction was made for the final velocity of the cart. We hypothesized that our experimental data would agree with this prediction. To obtain the required data, the cart arduino was configured to gather distance and time while the pendulum gathered acceleration in three axes. A Python based analysis was performed to extract the required values and errors in measurement. Comparing the results, we found that 2 out of 4 runs of the experiment matched the theoretical prediction for final velocity of the cart within two standard deviations. Run 2 was in agreement with the theoretical prediction of -0.1227 m/s with -0.123 ± 0.005 m/s experimentally, while Run 4 was in agreement with the theoretical prediction of -0.0947 m/s with -0.088 ± 0.004 m/s experimentally. Moreover, Run 1 and 3 experimental data were close to their theoretical counterparts although not within two standard deviations. Therefore, we concluded our hypothesis was validated. Additionally, an impulse based prediction was made as an alternate way of calculating the final velocity of the cart but our experimental results did not match our hypothesis following this method.

Introduction

The goal of this project was to confirm the existence of conservation of energy and momentum in a controlled experiment. This was attempted using a cart-pendulum system made of two arduinos – one attached to a moving cart and the other hung by a string from a stand to act as a pendulum as depicted in **Figure 1**. The arduino on the cart is given a slight push to generate a constant initial velocity which results in an elastic collision with the arduino acting as a pendulum followed by a quick second elastic collision after the pendulum completes its swing. A theoretical prediction can then be made for the final velocity of the cart assuming conservation of energy and momentum. The complete experimental setup is shown in **Figure 2**.

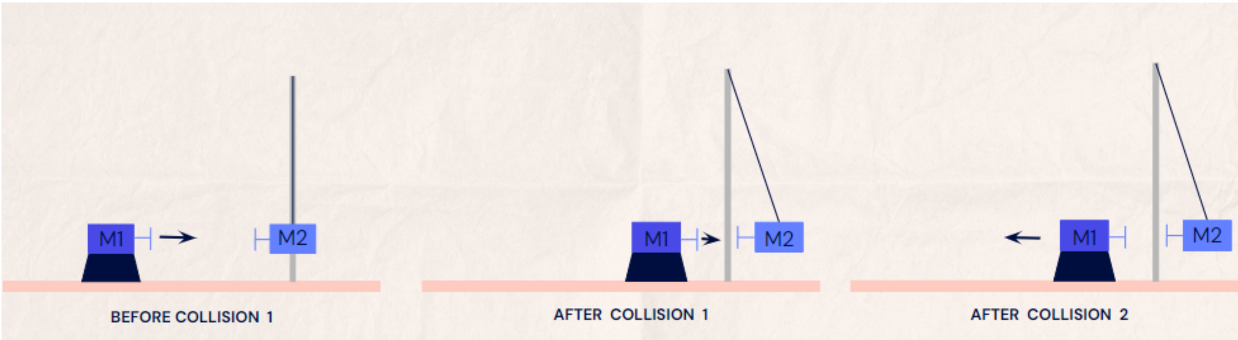


Figure 1: The Cart-Pendulum system used to perform the experiment

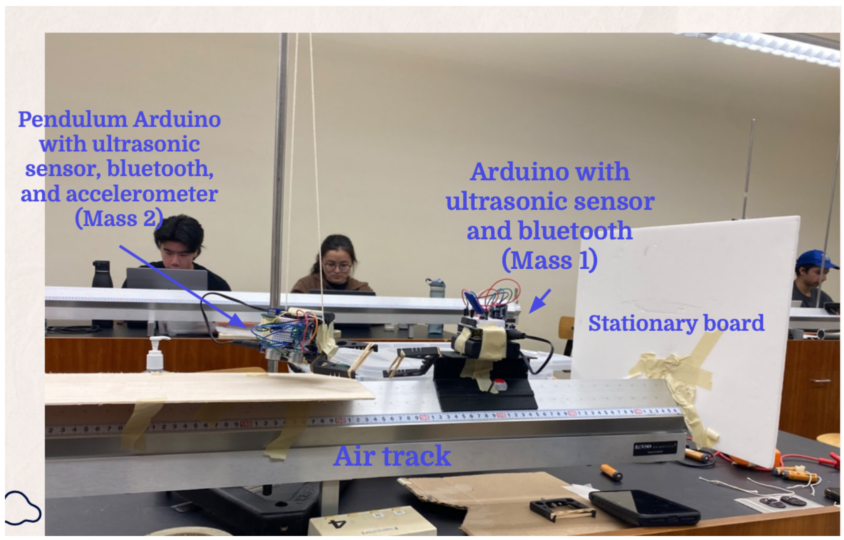


Figure 2: The Cart-Pendulum system used to perform the experiment

Elastic collisions occur when both momentum and kinetic energy are conserved in a clash of masses. This happens when no work is done by non-conservative forces such as friction. Mathematically, this is summarized by **Equations 1.1-1.2**,

$$m_1u_1 + m_2u_2 = m_1v_1 + m_2v_2$$

Equation 1.1: Conservation of Momentum

$$\frac{1}{2}m_1u_1^2 + \frac{1}{2}m_2u_2^2 = \frac{1}{2}m_1v_1^2 + \frac{1}{2}m_2v_2^2$$

Equation 1.2: Conservation of Kinetic Energy

If we consider m_1 to be the mass of the moving cart and m_2 to be the mass of the pendulum, a simple rearrangement of **Equation 1.1** gives us the velocity of the pendulum post collision 1 of the experiment,

$$v_2 = \frac{m_1}{m_2}(u_1 - v_1)$$

Equation 1.3: Initial velocity of the Pendulum after collision 1

The pendulum which is now in motion with initial velocity v_2 given by **Equation 1.3** continues to conserve the energy of the system in the absence of damping forces like air-resistance. This is because the gravitational force is a conservative force and no work is done by the tension in the string since it acts in the direction perpendicular to the motion of the pendulum throughout its swing. This justification allows us to make use of the following result**,**

$$\frac{1}{2}m_2v_2^2 = \frac{1}{2}m_2v_{p_1}^2 + m_2g\Delta h$$

Equation 1.4: Conservation of Mechanical Energy in the Pendulum where v_{p1} is the velocity of the pendulum at some point along its swing after the 1st collision and Δh is the change in height

Equation 1.4 is of particular importance to us at the moment just prior to the second collision because this allows us to obtain the initial velocity of the pendulum before the second collision.

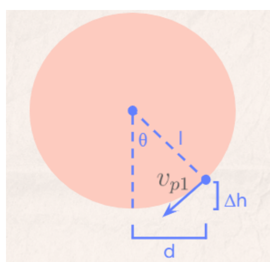


Figure 3: Pendulum before collision 2

After using some trigonometry based on the relationships in **Figure 3**, we can simplify **Equation 1.4** to solve for the initial velocity of the pendulum,

$$v_{p1} = \sqrt{v_2^2 - 2g(l - lcos\theta)}$$

Equation 1.5: Initial velocity of the pendulum before collision 2

$$cos\theta = \frac{\sqrt{l^2 - d^2}}{l}$$

Equation 1.6: Cosine of the angle between the cart and pendulum where d is the distance between the cart and the pendulum and l is the length of the pendulum

Before we are able to calculate the final velocity of the cart, we must find the final velocity of the pendulum to apply conservation of momentum. We can use the product of the length of the pendulum with the angular velocity of the pendulum post collision 2 to obtain this,

$$v_{p2} = l\theta'$$

Equation 1.7: Initial velocity of the pendulum before collision 2

Now that we have everything we need, we can use **Equation 1.1** for the second collision with the updated velocities,

$$m_1 v_1 - m_2 v_{p1_x} = m_1 v_{fc} + m_2 v_{p2_x}$$

Equation 1.8: Conservation of momentum during collision 2 where v_{fc} is the final velocity of the cart and v_{p1_x} and v_{p2_x} are x-components of the initial and final velocities of the pendulum

Notice that only the x-components of the pendulum's initial and final momentums are used since we are applying conservation of momentum in the x axis. A rearrangement of terms in **Equation 1.8** gives us the final velocity of the cart,

$$v_{fc} = v_1 - \frac{m_1}{m_2}(v_{p1} - v_{p2})cos\theta$$

Equation 1.9 : Final velocity of the cart after collision 2

Experimentally, the final velocity of the cart can be obtained using an Ultrasound Sensor that measures distance and time from a screen. In particular, the HC-SR04 Ultrasonic Sensor used in our experiment emits a chirp using its transceiver which is reflected off of an object and returns to its detector. The distance is then calculated using the following equation,

$$d = \frac{1}{2} v_s \Delta t$$

Equation 1.10: Distance between object and HC-SR04 Ultrasonic Sensor where v_s is the speed of sound and Δt is the time elapsed since the soundwave left the transceiver until it returns to the detector

The sensor can be attached to the arduino on the cart and the distance vs. time data subject to Python analysis can give us the velocity of the cart at different points along with the associated error.

The Accelerometer is a module that measures its own acceleration in the x, y and z axes. Attaching an accelerometer to the pendulum will give us valuable acceleration data that can be used to obtain angular velocity. The values for each axis will not have the appropriate units at first, which is why the accelerometer must be calibrated prior to use.

We hypothesize that the final velocity of the cart post collision 2 obtained experimentally using the Ultrasonic Sensor data will match the final velocity of the cart predicted by **Equation 1.9**.

Alternatively, we may use an impulse-based approach to calculate the theoretical final velocity of the cart. Since impulses during each collision are the only contributors to change in momentum, we can use conservation of momentum to obtain the following equation,

$$v_{fc} = \frac{m_1 u_1 - \int_{t_1}^{t_2} F_1 dt - \int_{t_3}^{t_4} F_2 dt}{m1}$$

Equation 1.11 : Final velocity of the cart after collision 2 derived using impulses from collisions 1 and 2 where F_1 and F_2 are the forces of impact during each collision experienced by the cart and their integrals are impulse

In **Equation 1.11**, the impulses are subtracted since the pendulum experiences an opposite impulse with respect to the cart. Our second hypothesis is that the velocity predicted by **Equation 1.11** should also be in agreement with the experimentally obtained final velocity for the cart.

Methods

Equipment

- 2 Arduinos
- 2 Laptops
- String
- Metal stand
- HC-SR04 Ultrasonic Sensor
- 2 HC-06 Bluetooth modules
- MPU6050 Accelerometer
- Tape
- 8 1.5V batteries
- Voltmeter
- Ruler
- Weighing Scale
- Air Track
- Cart/Air Glider
- Rubber bands
- Attachment that collides Rubber bands
- Cardboard Screen

Begin by attaching the ultrasonic sensor and bluetooth module to the arduino to be placed on the cart and follow the wiring diagram given in **Figure 4** to complete the arduino setup. Similarly, attach the accelerometer and second bluetooth module to the other arduino and wire it according to **Figure 4**.

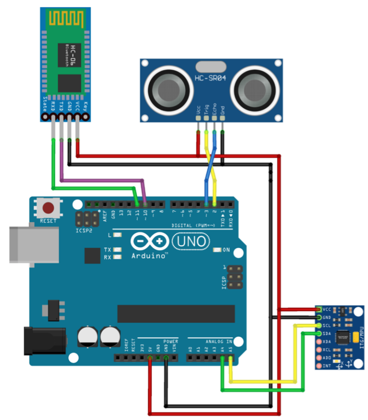


Figure 4: Pendulum Arduino wiring (ignore accelerometer wiring for Cart)

Following completion of the wiring process, each Arduino must be connected to separate computers via USB so that the required code can be uploaded from the Arduino IDEs respectively. The code is written in C++ and uses the SoftwareSerial library for serial communication between the computer and the Arduino's digital pins. Once the code is uploaded, remove the USB cable and connect the setup to a battery holder. Each battery holder contains four batteries with at least 1.4 V each. This is checked beforehand using a voltmeter.

Now set up the air track on some level surface. Fix the cardboard screen to one end of the airtrack while making sure its surface is perpendicular to the airtrack. Take the arduino to be attached to the cart, and tape the battery holder to its bottom. Tape this combined arduino setup to the cart/air glider. Attach a rubber band holder to the front of the cart facing away from the ultrasonic sensor. Weigh this entire cart setup. Now place this cart setup on the air track and make sure that the ultrasonic sensor on the arduino faces the cardboard screen.

Take the other arduino and loop a string through its holes. Then tape the battery holder to its bottom and attach a rubber band holder to its side, keeping in mind that it will be facing the cart's rubber band holder. Weigh this pendulum setup using a weighing scale.

Before using the accelerometer on the Pendulum Arduino, it must be calibrated to reflect the appropriate units in each axis. This can be done by orienting the Arduino such that the accelerometer experiences acceleration due to gravity along each axis and measuring the respective accelerometer values using the Arduino code. Using the collected accelerometer data, we can extract the fit-coefficients with numpy's polyfit() function which can then be used to convert the accelerometer values to the appropriate units. **Figures 5-7** show the x-axis, y-axis, and z-axis calibrations respectively.

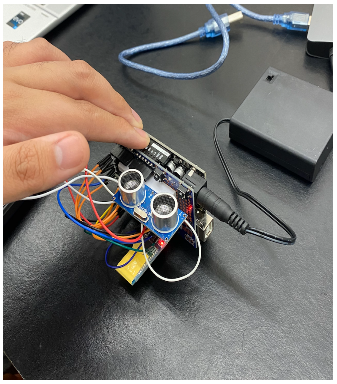


Figure 5: X-axis calibration

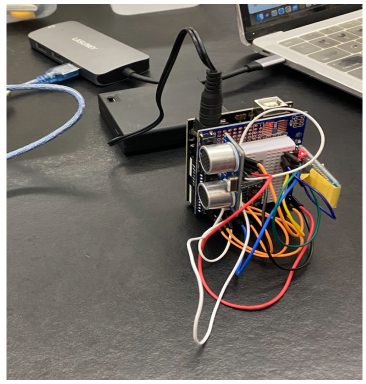


Figure 6: Y-axis calibration

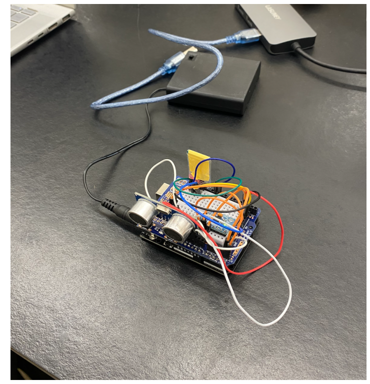


Figure 7: Z-axis calibration

After calibrating the accelerometer, tie the pendulum arduino to the metal stand using its string. Measure the length of the pendulum using a ruler. Once again, ensure that the rubber band holders face each other. This is an essential part of the experiment because the rubber bands must be the points of contact during the collisions to ensure that the collisions are nearly elastic.

Now turn on the air supply for the air track and set it to level 6 to make friction negligible. In addition to assuming elastic collision and ignoring friction, assume negligible effect of air resistance. We are now ready to begin the experiment.

To run the experiment, push the cart for an instant towards the pendulum and track the ultrasonic sensor and accelerometer data until about 10 seconds after both collisions are completed. Repeat this for multiple trials.

Once we obtain the data, we can move on to data analysis using Python to extract the required velocities and angular velocity. To find the velocities of the cart before collisions 1 and 2 and after collision 2 along with the associated errors, we can use numpy's polyfit() function. Section the graph into three parts: pre-collision 1, post-collision 1 and post-collision 2. Extract the best-fit coefficients and covariance matrices for the distance vs. time graphs for each section. The slope of the best fit lines will give us the velocity and the square root of the covariance of the slope will give us the error in the velocity. The velocity obtained from the post-collision 2 graph is our experimentally obtained final velocity of the cart.

To find the angular velocity of the pendulum post-collision 2, we need to use the y-axis accelerometer data that we obtained. On graphing the y-axis accelerometer data vs. time, sine-curve-like oscillations will be visible after two large upside down peaks. This is observed in all y-axis accelerometer data vs. time graphs. The two large upside down peaks correspond with the moments of collision. We are interested in the oscillations after that. We can use scipy's curve_fit() function to fit these oscillations to the following guess function,

$$a_y = Asin(\omega t + \phi)$$

Equation 2.1 : Guess function to fit y-axis accelerometer data oscillations

The best fit-curve that we obtain can be used to find the angular displacement of the pendulum post-collision 2 using the following equation,

$$\theta = \sin^{-1}(\frac{a_y}{g})$$

Equation 2.2 : Angular displacement of pendulum post-collision 2 where a_y is the accelerometer data in the y-axis

This is because the acceleration that the arduino experiences in y-axis is $gsin\theta$ considering gravity is the only force at work. We can perform theoretical differentiation on **Equation 2.2** to get the angular velocity of the pendulum post-collision 2,

$$\theta' = \frac{1}{g\sqrt{1-(\frac{a_y}{g})^2}} A\omega cos(\omega t + \phi)$$

Equation 2.3: Angular velocity of the pendulum post-collision 2

We can use the angular velocity obtained using **Equation 2.3** along with the initial velocity of the cart obtained using the ultrasonic sensor to calculate the theoretical final velocity using **Equation 1.9**.

Alternatively we can use impulses to calculate the final velocity using **Equation 1.11**. We can find the impulses for each collision by performing numerical integration over the product of the large upside down peaks corresponding to each collision with the cart's mass using scipy's simpson() function. We can plug the results into **Equation 1.1** to get yet another theoretical prediction for the final velocity of the cart.

Results

To start with, we found that the mass of the cart was 430 ± 0.5 g while the mass of the pendulum was 233 ± 0.5 g. We also found that the length of the pendulum was 44.2 ± 0.05 cm. Additionally, we found that the maximum distance before we started to observe noisy ultrasonic sensor data was 45 cm. Then, by following the process in the **Methods** section, we performed 4 trials of the experiment and obtained the following graphs for Distance vs. Time for the moving cart:

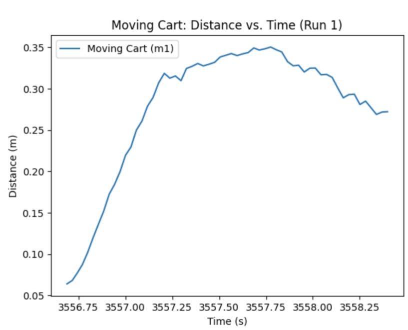


Figure 8: Run 1(Raw data)

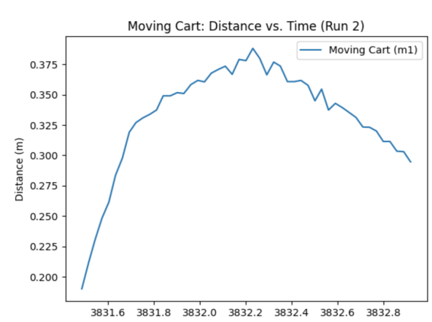


Figure 9: Run 2 (Raw data)

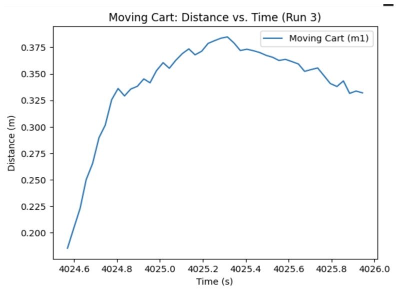


Figure 10: Run 3(Raw data)

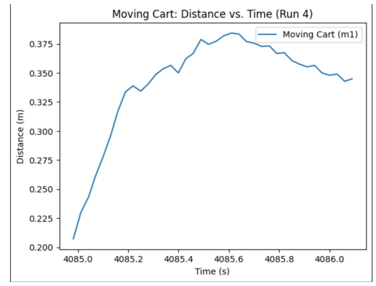


Figure 11: Run 4 (Raw data)

Before using numpy's polyfit(), we sliced the data into 3 sections, pre-collision 1, post-collision 1 and post-collision 2 using array indices. Using polyfit() we obtained the following best-fit lines for the pre-collision 1 Distance vs. Time graphs:

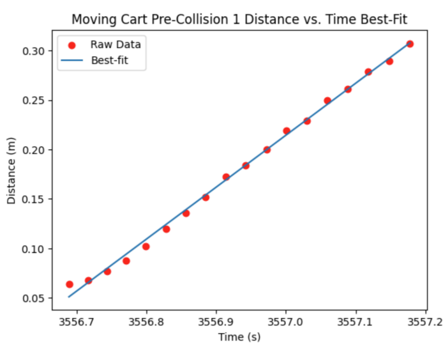


Figure 12: Run 1 (Pre-collision 1)

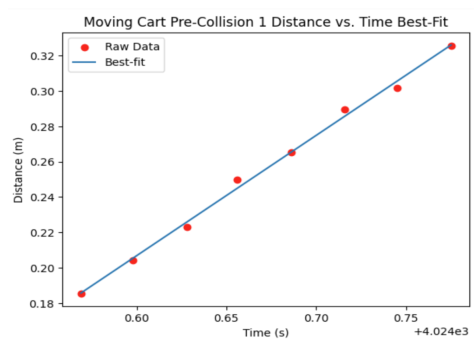


Figure 13: Run 2 (Pre-collision 1)

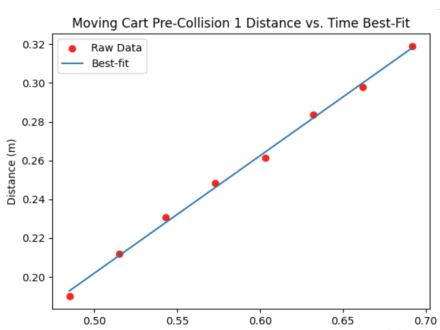


Figure 14: Run 3 (Pre-collision 1)

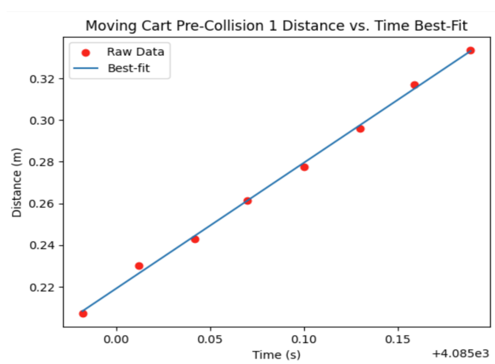


Figure 15: Run 4 (Pre-collision 1)

Table 1 displays the results for the velocity of the cart pre-collision 1:

Table 1: Pre-Collision 1 Velocities for Cart (Mass 1)

Run #	**Velocity (m/s)**
1	0.525 ± 0.008
2	0.606 ± 0.0129
3	0.681 ± 0.0175
4	0.603 ± 0.0111

Next, we obtained the following best-fit lines for the post-collision 1 Distance vs. Time graphs:

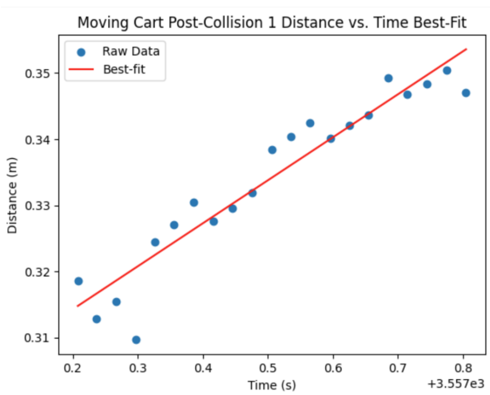


Figure 16: Run 1 (Post-Collision 1)

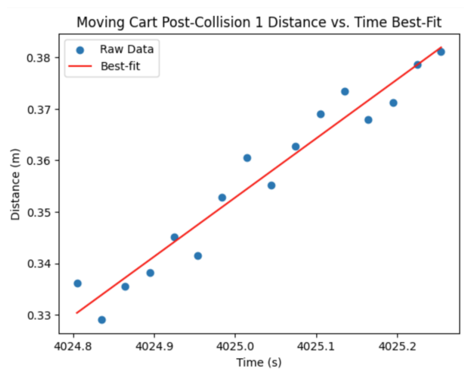


Figure 17: Run 2 (Post-Collision 1)

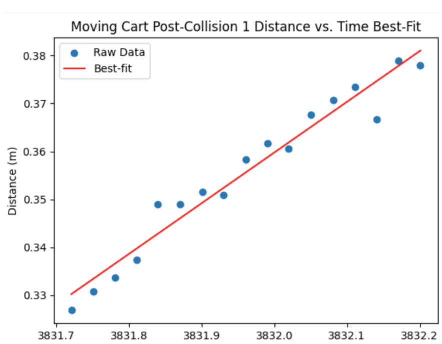


Figure 18: Run 3 (Post-Collision 1)

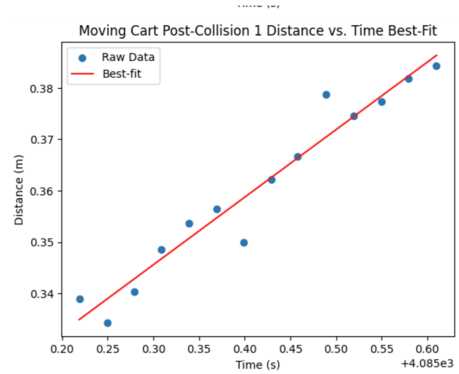


Figure 19: Run 4 (Post-Collision 1)

Table 2 displays the results for the velocity of the cart post-collision 1:

Table 2: Post-Collision 1 Velocities for Cart (Mass 1)

Run #	**Velocity (m/s)**
1	0.065 ± 0.005
2	0.106 ± 0.006
3	0.115 ± 0.007
4	0.132 ± 0.009

We then obtained the following best-fit lines for the post-collision 2 Distance vs. Time graphs:

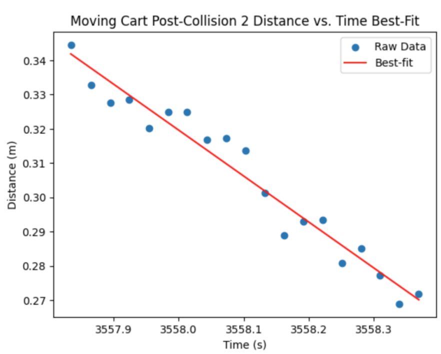


Figure 20: Run 1 (Post-Collision 2)

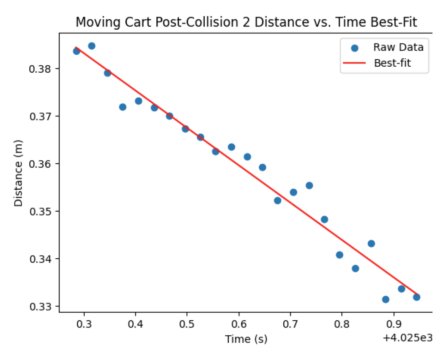


Figure 21: Run 2 (Post-Collision 2)

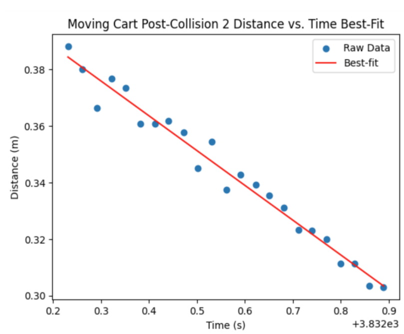


Figure 22: Run 3 (Post-Collision 2)

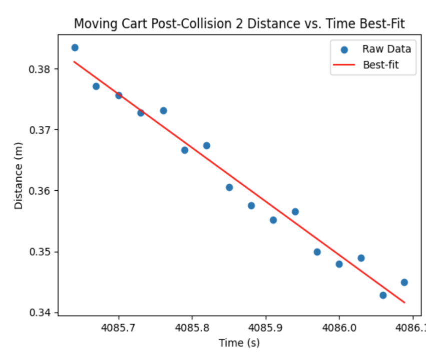


Figure 23: Run 4 (Post-Collision 2)

Table 3 displays the results for the final velocity of the cart post-collision 2:

Table 3: Post-Collision 2 – Final Velocities for Cart (Mass 1)

Run #	**Final Velocity (m/s)**
1	-0.134 ± 0.007
2	-0.123 ± 0.005
3	-0.079 ± 0.003
4	-0.088 ± 0.004

We then used scipy's curve_fit() function to obtain best fit curves for the post-collision 2 y-axis accelerometer data:

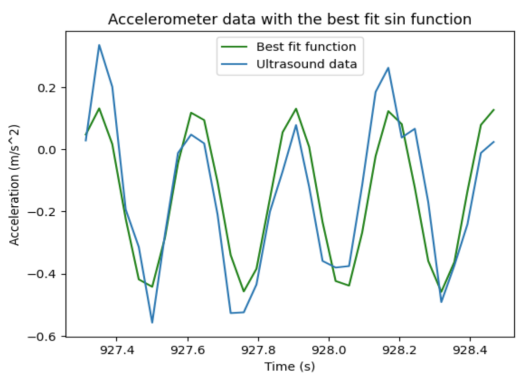


Figure 24: Run 1 (Accelerometer data)

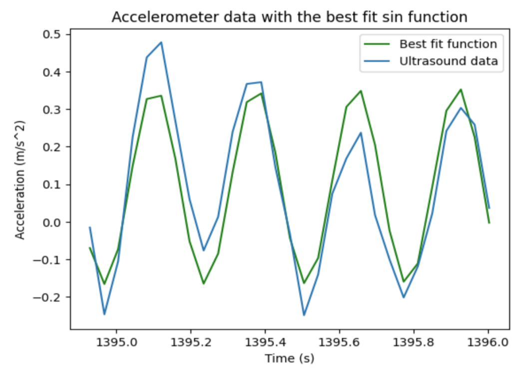


Figure 25: Run 2 (Accelerometer data)

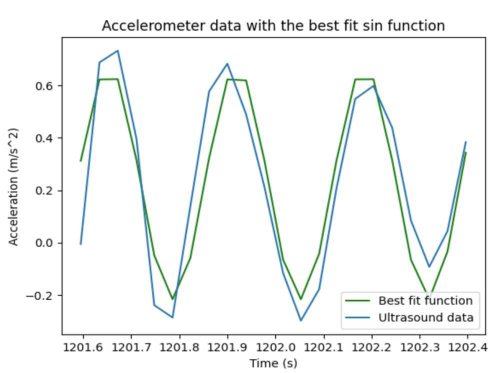


Figure 26: Run 3 (Accelerometer data)

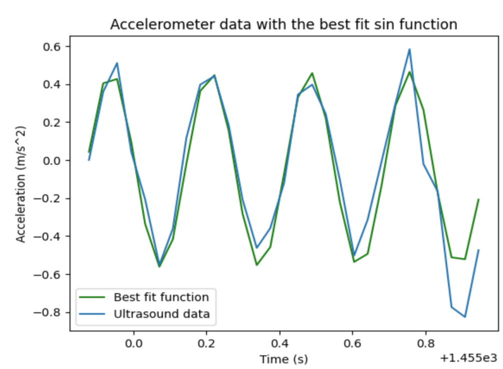


Figure 27: Run 4 (Accelerometer data)

We used these best-fit curves to find the angular displacement of the pendulum post-collision 2 using **Equation 2.2**. The derivative of this angular displacement gave us the angular velocity of the pendulum post-collision 2. This was then used to calculate the theoretical final velocity of the pendulum using **Equation 1.9**.

Table 4 displays the results for the experimental vs. theoretical final velocities of the car post-collision 2t:

Table 4: Experimental vs. Theoretical Final Velocities

Run #	Experimental Final Velocity (m/s)	Theoretical Final Velocity (m/s)
1	-0.134 ± 0.007	-0.2284
2	-0.123 ± 0.005	-0.1227
3	-0.079 ± 0.003	-0.1803
4	-0.088 ± 0.004	-0.0947

Run 2 and 4 are within the 2 standard deviations of the expected theoretical values for the final velocities. Runs 1 and 3 did not fall within 2 standard deviations, but they are close to the expected velocity values.

We also used an alternate method to find the final velocity of the cart involving impulse calculations. The following graphs show the raw y-axis accelerometer data vs. time:

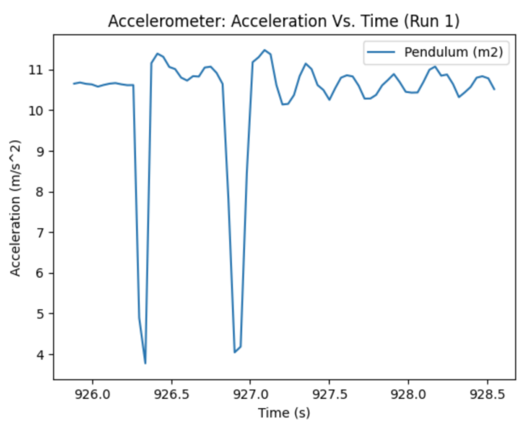


Figure 28: Run 1 (Accelerometer Data)

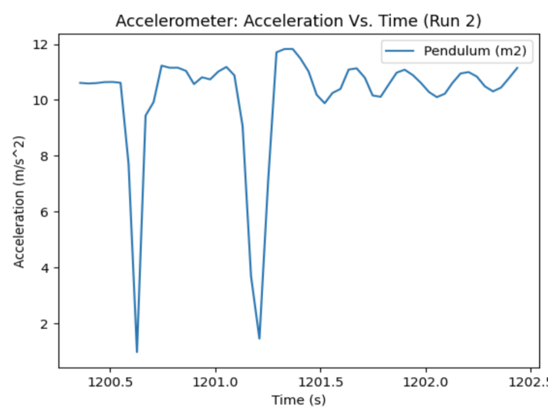
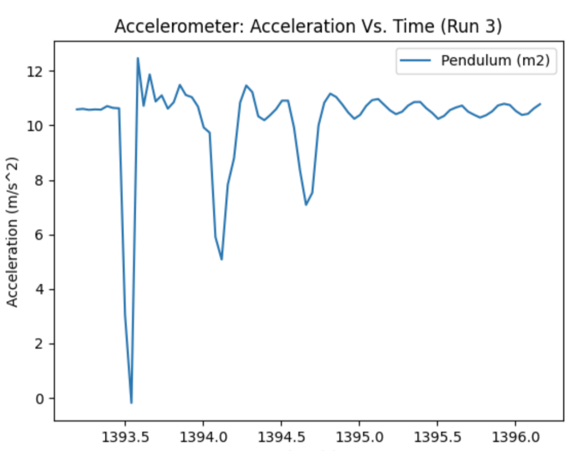


Figure 29: Run 2 (Accelerometer Data)



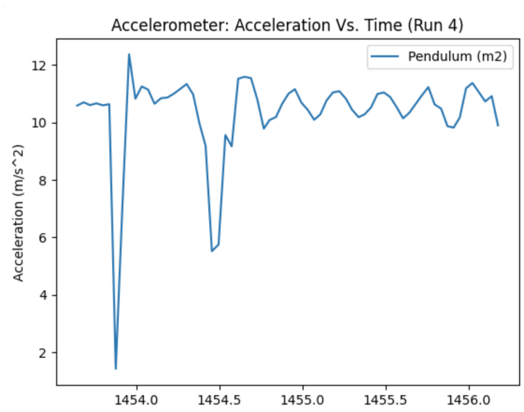


Figure 30: Run 3 (Accelerometer Data)

Figure 31: Run 4 (Accelerometer Data)

Numerically integrating the product of the peaks with the cart's mass using scipy.simpson() gave us the impulse values for each collision which we plugged into **Equation 1.11**.

Table 5 displays the final velocities of the cart when using the Impulse Method:

Table 5: Final Velocities using Impulse Method

Run #	**Final Velocity (m/s)**
1	-1.2909
2	-3.8895
3	-4.0231
4	-2.0818

Clearly, none of these values match with our experimental and theoretical results. This is most likely due to the extremely short time intervals for the accelerometer to generate data combined with noise produced by numerical integration.

Conclusion

The goal of this project was to confirm the existence of conservation of energy and momentum in a controlled experiment. We would achieve this goal if the final velocity of the cart in our cart-pendulum experiment matched our theoretical prediction for the final velocity since the theoretical prediction was made using the assumption that energy and momentum are conserved. Therefore, we hypothesized that the final velocity of the moving cart would agree with the final velocity obtained using **Equation 1.9.**

Through our experiment we found that Runs 2 and 4, i.e. 2 out of 4 total runs were within 2 standard deviations of the experimental final velocity of the cart. Moreover, Runs 1 and 3 were quite close to their theoretical predictions. Therefore we believe that our results validate our hypothesis.

We also tried another method using impulses to find the final velocity of the cart. However, the observed results did not match with our experimental and theoretical velocity values. This is most likely due to the extremely short time intervals for the accelerometer to capture data combined with the noise produced by numerical integration.

One source of error in our experiment was the slight movement of the pendulum in the z-axis after collisions. If we could isolate the pendulum's movement to the xy plane, it would give us more accurate data. Another source of error is the approximation of the arduino based pendulum as a point mass. If we could find a way to include the moment of inertia of the arduino and treat it as a physical pendulum instead of a simple pendulum, we would get more accurate results. Our decision to ignore air resistance could also have been a large source of error in our experiment. This can be fixed by using some sort of vacuum chamber to limit the effect of air as a damping force in the experiment. Lastly, our rubber bands may not have been elastic enough and may have lost more energy to heat than expected decreasing the amount of energy that was conserved in the clash between Arduinos.

Overall, this project demonstrates that the conservation of energy and momentum is experimentally found to be valid and can be used to model various systems in physics.

References

Galkin, Alexander. “Unit 2 - Motion Lab 2D: Accelerometer.” Physics 4AL. 18 Oct. 2023, UCLA. https://docs.google.com/presentation/d/1Bou9drotM-jY4O6TRgDbmZg3dIgils99/edit#slide=id.p7

“Propagation of uncertainty.” Wikipedia, The Free Encyclopedia, Wikimedia Foundation, 7 September 2023. https://en.wikipedia.org/wiki/Propagation_of_uncertainty

Thank you to our TA Alexander Galkin and the UCLA Physics 4AL Labs