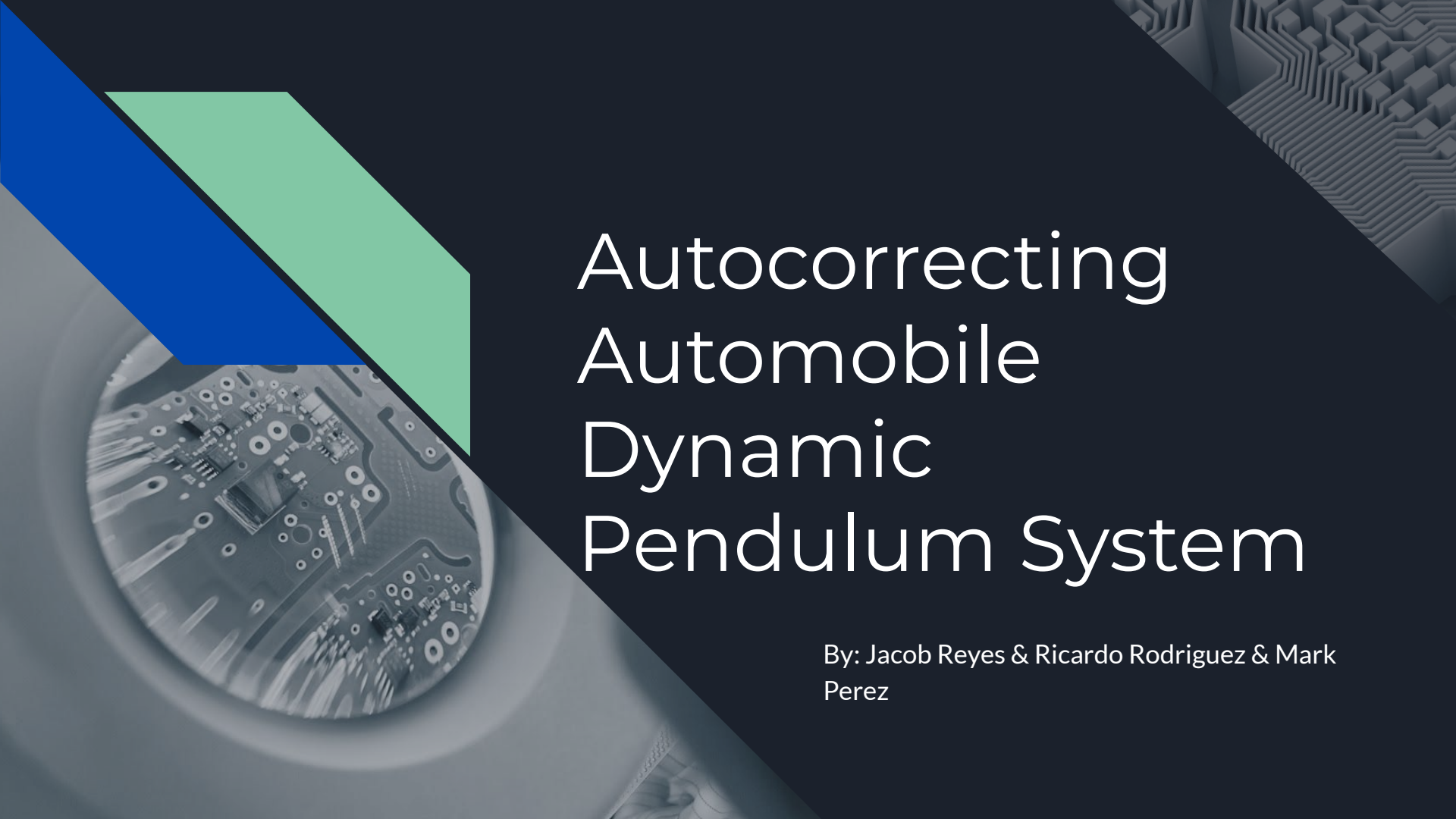


Autocorrecting Automobile Dynamic Pendulum System

By: Jacob Reyes & Ricardo Rodriguez & Mark Perez

Overview

The Autocorrecting Automobile Dynamic Pendulum System or vehicle pendulum, is a autocorrecting robot that stabilizes a fixed pendulum of mass by changing the acceleration of the vehicle in the direction the pendulum is falling.

Project objective / Goals

The main objective of this project is to design and develop a vehicle that will house a pendulum, as well as a controller that will stabilize the pendulum using motors to accelerate the vehicle.

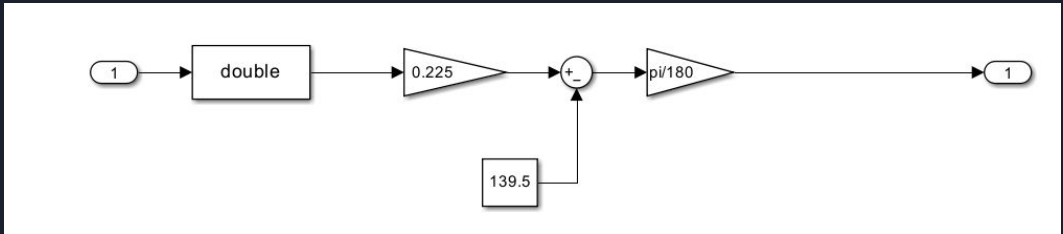
This project will allow us to gain a better understanding of control systems used for dynamic applications in robotics.

Program Development

The first component we shall focus on would be the setup for converting the analog voltage of the potentiometer to a coherent value of radiants. Where we will get 0 radians when the potentiometer is perfectly placed in the middle to simulate the pendulum being at a 90 degree angle from the horizontal line.

Conditional:

$$Radians = ((AI \times 0.225) - 139.5) \times \frac{\pi}{180}$$



Program Development

The MatLab Function, where “u” will represent our output for the potentiometer setup. A conditional if statement to program this function. We must firstly zero out the output “Y#” as it will represent an H-Bridge Mode we will not be using if both motors are on at the same time.

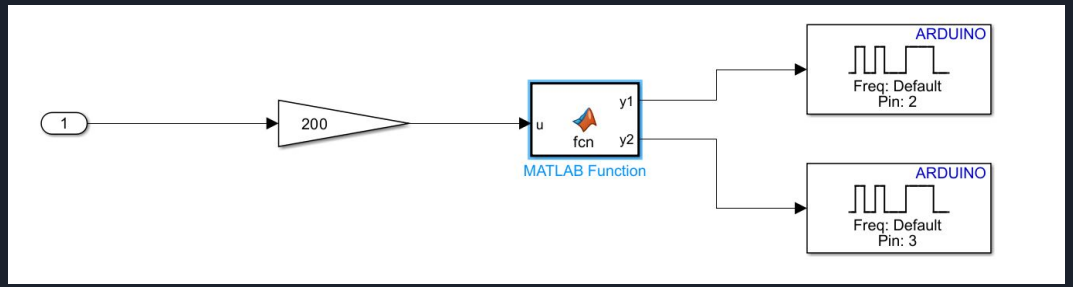
```
function [y1,y2]= fcn(u)

if u >= 0
    y2 = 0;
    y1 = u;
else
    y1 = 0;
    y2 = -u;
end
```

Resembles a Bang-Bang Controller.

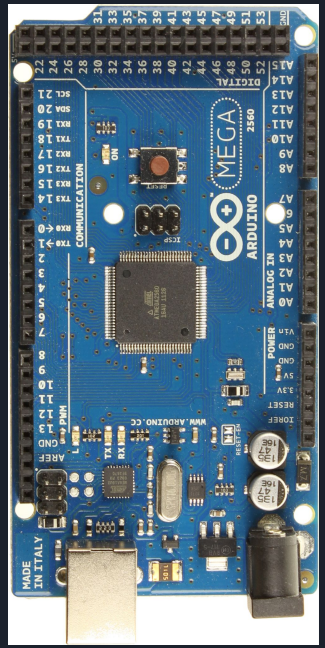
Program Development

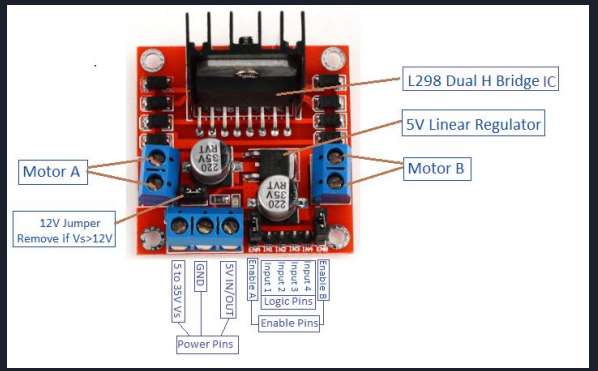
The following figure below is the configuration in which our MatLab Function will find itself. We find that “1” is acting as the output for the potentiometer setup after it being connected to a PID Controller.This PID will hold uncommon characteristics due to the need for real time fast configurations. This means the PID controller would have a proportional value of 1000, an integral value of 1, an derivative value of 1, and a filter coefficient of 100. This contributed highly to our software success for when to choose the correct bang bang configuration. We must multiply the potentiometer value by 200 to gain an appropriate torque PWM Value of the Motor. (200/255 = 78%)



Components of Automated Car Avoidance System

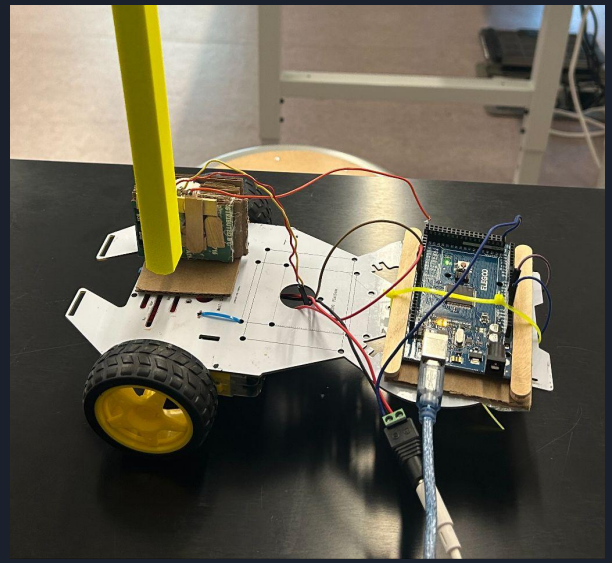
- Arduino Mega 2560
- L298N H-Bridge Motor Driver
- Small 1.5-3V Electric Motor (x2)
- Gearbox Assembly / Wheels
- 9 Volt power supply cable
- 10k Ohm linear potentiometer



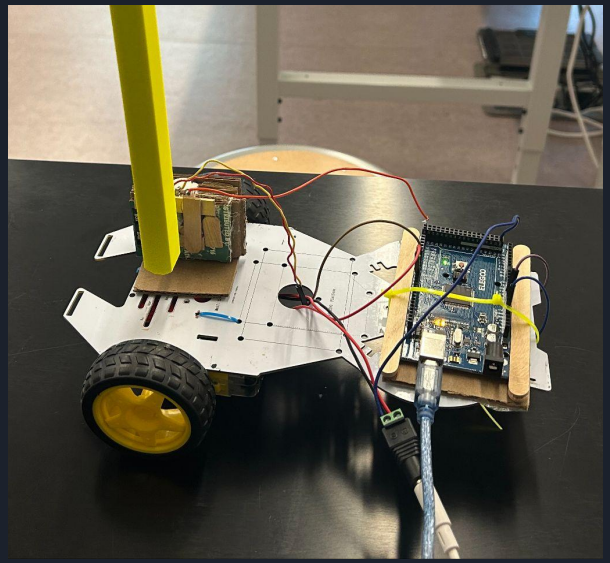


Hardware Development

- Wiring the Potentiometer and Motors
- Pre-Assembly Testing
- Preparing the Robot Base
  - Attaching Gearbox Assembly
- Securing the Arduino and Motor Driver
- Creating an Arduino Platform
- Building the Potentiometer Mount
  - Designing and 3D Printing the Pendulum
  - Attaching the Pendulum

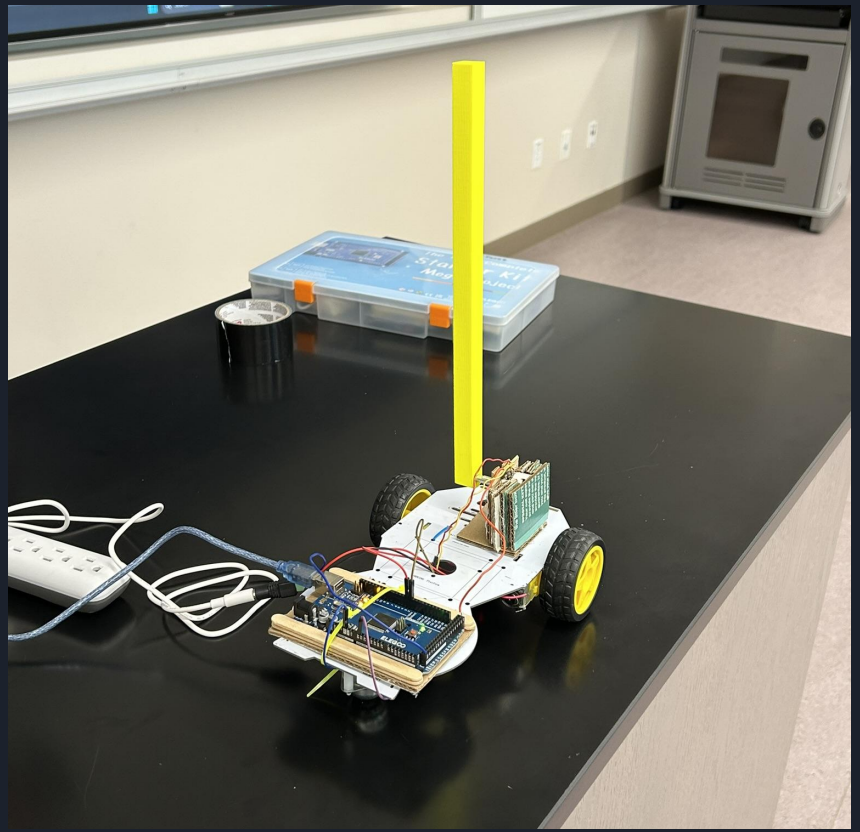


Completed Vehicle

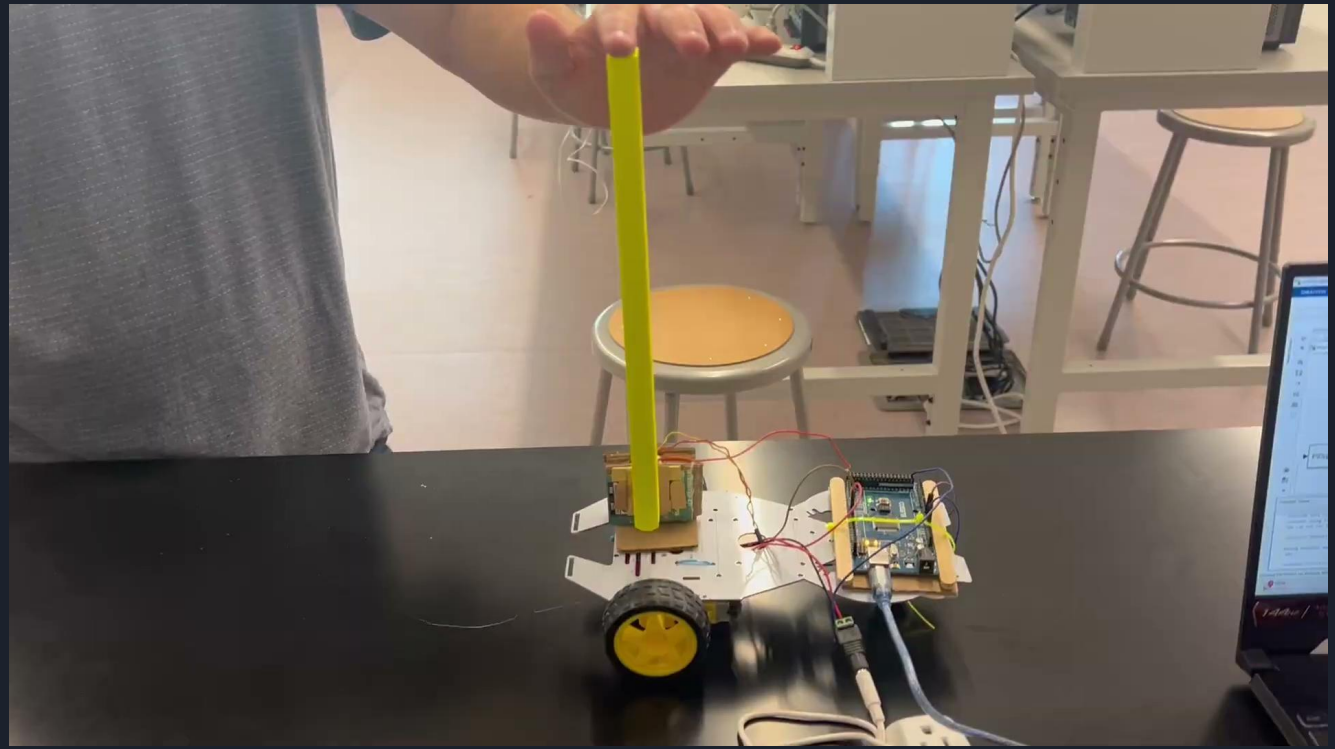


Testing and Results

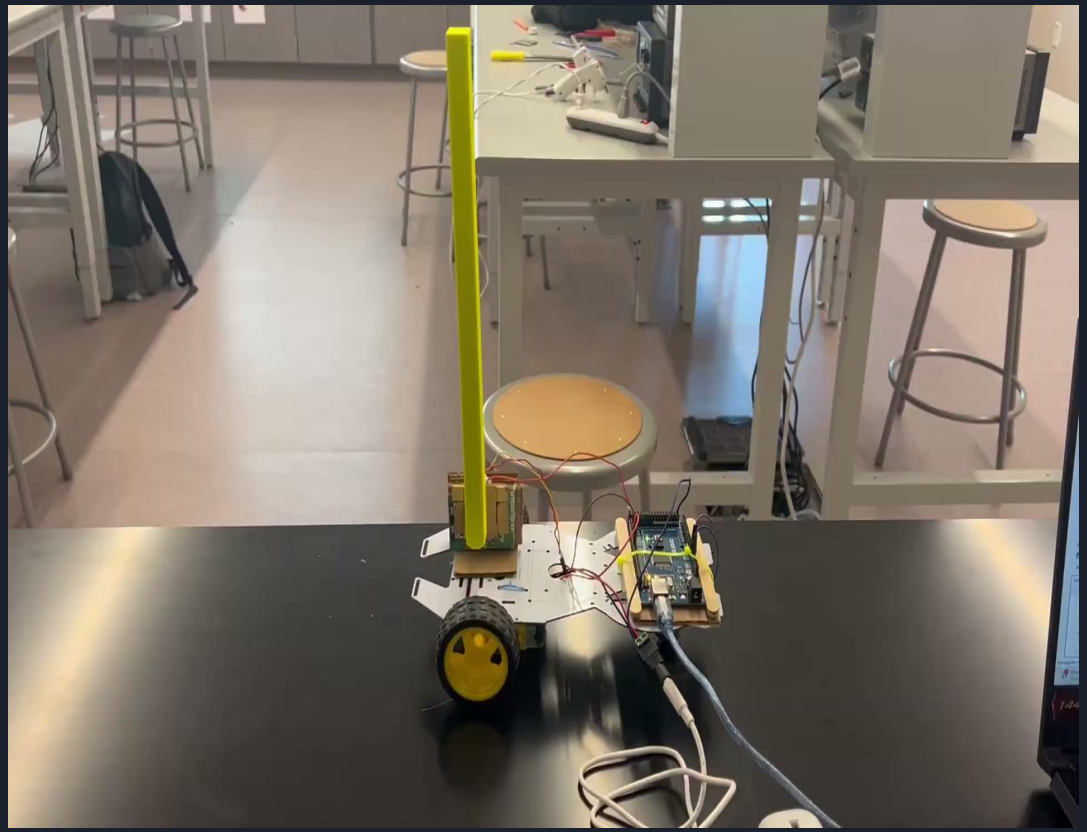
- Types of Testing Done:
  - Placement of the Hardware Components on the Chassis of the Robotic Vehicle
  - Proportional Integral Derivative (PID) Values within Simulink Control System
  - Length of the Pendulum Rod
- Results:
  - Overall Function of the Pendulum Car was Practical
  - Improvements Can be Made to Increase the Productivity of the System



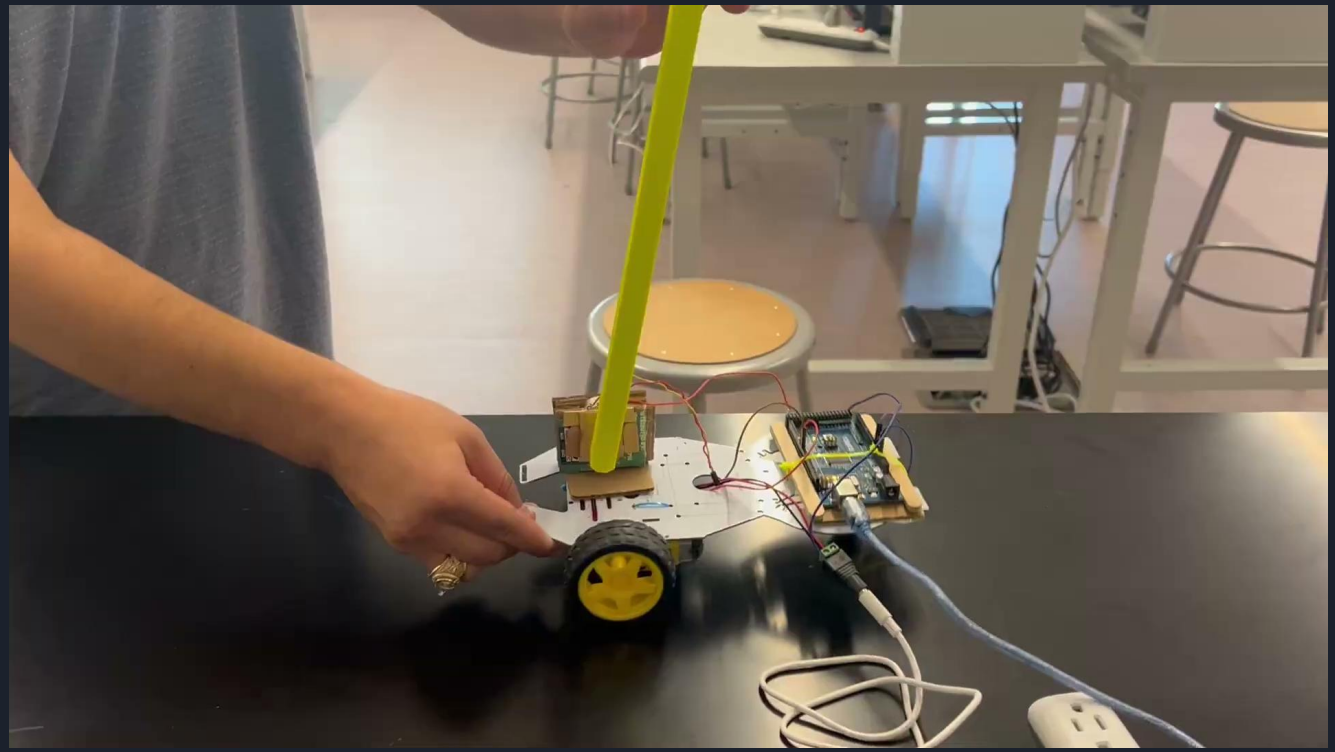
Proof of Concept:



First Trial:

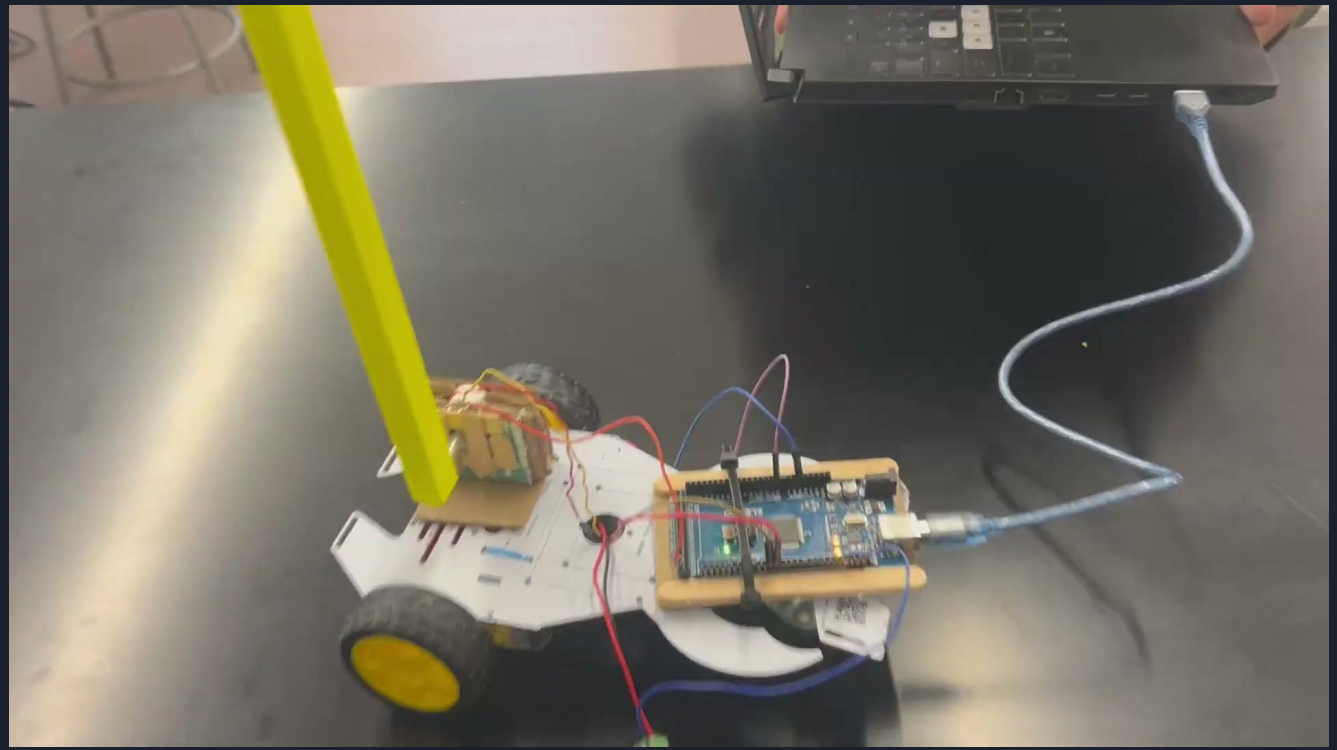


Second Trial:

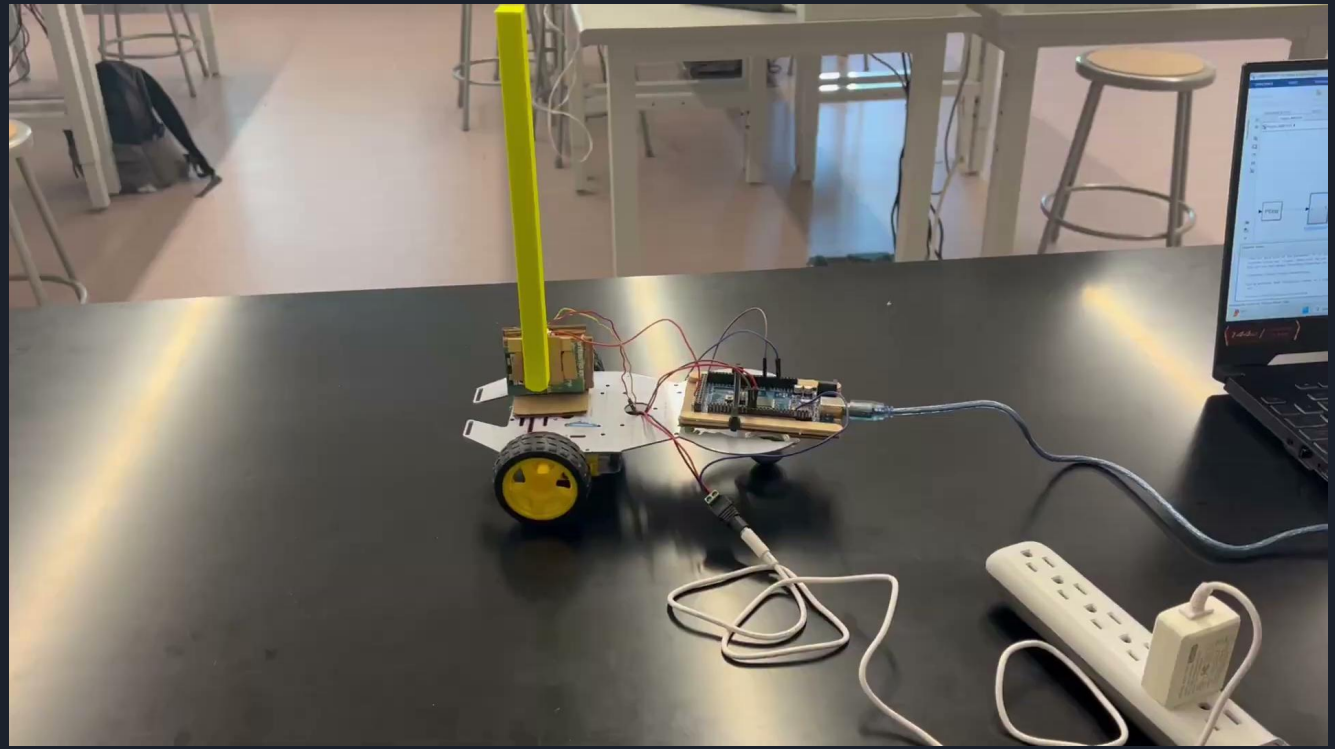


Pendulum Car Demonstration

Final Trial ¾ Pendulum Length



Issues We Faced



Constraints and Future Improvements

Constraints:

- **External Power Reliant:** The vehicle needed to stay connected to an outlet for power. This limited the operational distance of the autocorrecting vehicle pendulum system, restricting the vehicle's mobility.
- **Wired Connection to Computer:** The Arduino Mega 2560 board had to be directly connected to a computer to simulate the program, and the short wire restricted the car's movement even further.
- **Limited Testing Area:** The effectiveness of the system was hampered by a limited testing area, as the project would perform better on an infinitely long, uninterrupted stretch of road for safe operation and testing.
- **Torque of Motors:** The motors we used in the vehicle were small motors with not much torque. The motors had barely enough torque to move the robot under its own weight.

Constraints and Future Improvements

Future Improvements:

- **Battery Pack:** Adding a battery pack to replace the reliance on an outlet would provide greater mobility for the pendulum car, though it may increase the car's weight.
- **Higher Power Motors:** Motors with more torque would allow the vehicle to accelerate more quickly, thus stabilizing the pendulum more efficiently.
- **Alternative Software:** Using software other than MATLAB that allows direct downloading and running of the program without a computer connection would improve movement and reduce connection constraints. This could involve using C or C++ in the Arduino software.
- **Larger Testing Area:** Conducting the experiment in a large, low-friction area (such as an open space or hallway) would minimize space limitations and allow for better testing and performance of the pendulum car.

Thanks And Gig ‘Em