Toronto Metropolitan University
Department of Mechanical, Industrial, and Mechatronics Engineering
Faculty of Engineering & Architectural Science

Department of Mechanical, Industrial, and Mechatronics Engineering

Please select your current program below:

- [x] ~~Mechanical Engineering~~
- [ ] **Industrial Engineering**
- [ ] **Mechatronics Engineering**

Course Number	MEC830
Course Title	Mechatronics Systems Design
Semester/Year	F2024
Section Number	04
Group Number	11

Project No. 2

Assignment Title	Project 2
Submission Date	Nov 22, 2024
Due Date	Nov 22, 2024

Student Name	Student ID (xxxx1234)	Signature*
Harsh Singh	xxxx0937	H.S.
Sarvin Jeevakaran	xxxx86052	J. Sarvin
William Martin	xxxx42092	W.M
Amish Salwan	xxxxx2453	A.S

(Note: Remove the first 4 digits from your student ID)

**By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: http://www.ryerson.ca/senate/policies/pol60.pdf.*

Table of Contents

Introduction

An inverted pendulum is a combination of a pivot and a mass that oscillates about the said pivot. Unlike a regular pendulum, the location of the mass is positioned above the pivot, meaning that unless an appropriate controller is applied to the system, the inverted pendulum will be unstable to the smallest disturbances [1].

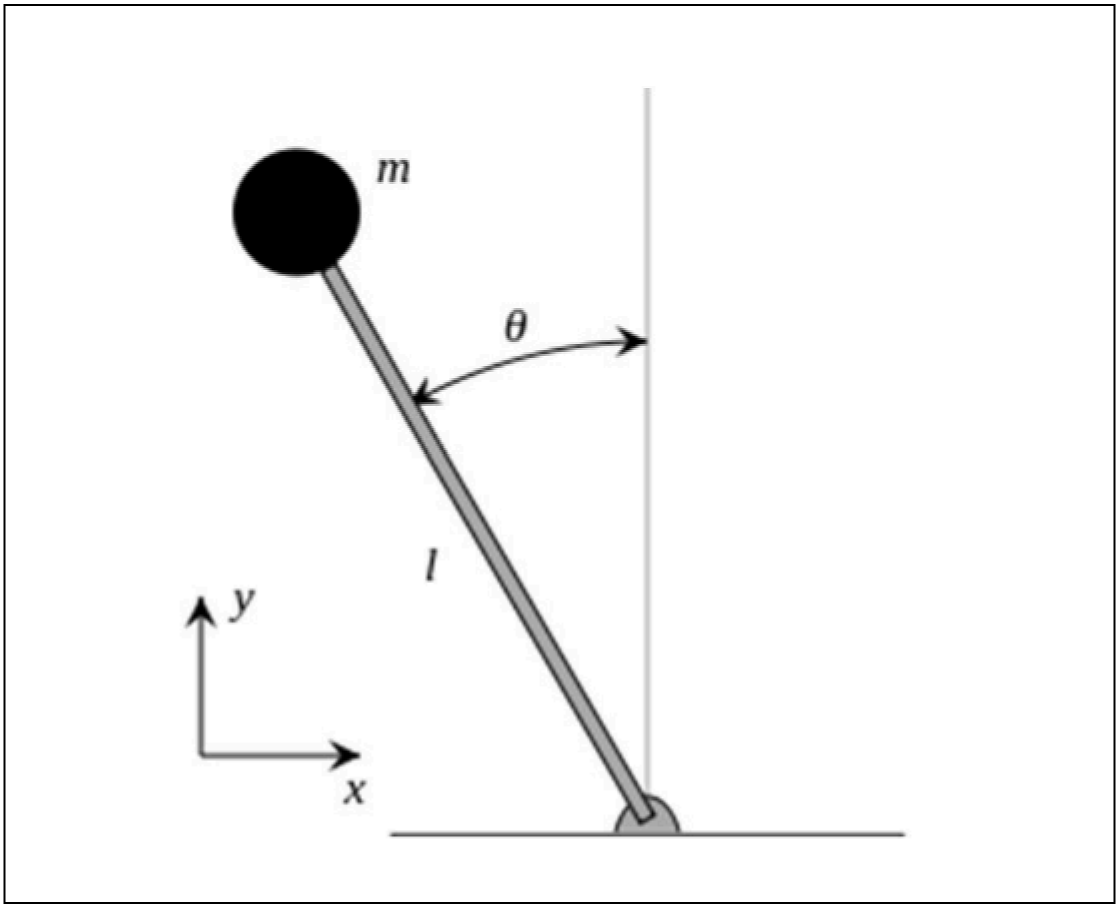


Figure 1 - Schematic of Inverted Pendulum

Within this project, a big emphasis is on the ability to stabilize the extremely unstable system. Using several mathematical modeling through many governing laws such as Newton's laws of motion, a system of equations that demonstrate the linear motion of the pivot and the angular motion of the pendulum can be attained. The aforementioned modeling is extremely useful in determining the exact physical properties that need to be controlled in order to implement the control systems to stabilize the inverted pendulum system. There are two main control systems that will be used to assess the ability of the inverted pendulum system to stabilize itself; a PID controller, and a Pole Placement controller.

A Proportional-Integral-Derivative controller, otherwise known as a PID controller, is a closed-loop feedback control system that functions by determining the error that is experienced by a specific system and applying feedback to the inputs of the system in an attempt to stabilize the system. Depending on the nature and the behavior of the error experienced, the P, I and D gain values can be modified to mitigate the encountered error as much as possible. Alternatively, a Pole Placement controller is also a closed-loop feedback control system that provides the input to the system to mitigate the error so that the target value can be reached as swiftly as possible. Instead of the PID gain values, the feedback gains are chosen depending on the location of the poles of the linearized equations.

Concurrency

For this project, a systematic approach by the name of concurrent engineering was adopted. It is a method of designing and developing products, in which the different stages run simultaneously, as opposed to running consecutively [2]. It was decided to use this approach due to the variety of advantages that this methodical approach offers. Because of multiple components of engineering working together, it is more efficient to gather feedback throughout the design process. This approach, when executed appropriately, reduces the delay in product design, as well as reduces the number of changes throughout the design process. Combining all of these allows the final product to be of utmost quality, by making sure all departments of engineering are satisfied with the product that is to be delivered to the end customerFour major components of concurrent engineering need to be considered so that this approach's full potential can be realized. These components are Organization, Communication, Design Problem, and Product Development. All four of these components are discussed in detail, including the evolution of the project throughout all stages of the design process. Organization and Communication are discussed below, whereas the Design Problem and Product Development are thoroughly discussed later in the Design Process and the Solution Formation sections of the report.

Organization

In order for the project to be successful, a proper organizational structure was pre-established so that all of the team members are aware of the roles and responsibilities that are expected of them. Breaking down the project into different components, and determining the skills-set needed to effectively accomplish tasks related to that component was extremely helpful. Subsequently, relating those skills-sets to the team members allowed the entire team to discuss the comfort and ability they have in order to accomplish the tasks related to a specific component. It is also important to note that the majority of the tasks needed some assistance from 3 or more team members, and very rarely was one task completed by one person by themselves. The table below shows the team members and their roles and responsibilities within the project.

Name of Team Member	Roles and Responsibilities
Amish Salwan	CAD Modelling Build Product Testing CAD Reporting and Engineering Drawings
Harsh Singh	Arduino Code Product Testing Top-Down/Bottoms-up Approach/Fishbone Diagram (FMEA) Concurrency Human Machine Interface
Sarvin Jeevakaran	Product Testing Arduino Coding Mind Map & Black Box Model Design Process Novelty
William Martin	FMEA chart MBSE diagrams Product Testing Wiring Diagram 3D Printing

Figure 2: Responsibility Table

Communication

The communication between team members is done through the use of WhatsApp messaging service. Any updates throughout the project are done by updating the entire team through this messaging service, allowing everyone to understand the status of the project as soon as possible. There were multiple meetings scheduled throughout the project so that the project can be completed right on the deadline. The completion of the project report is done through the use of Google Docs, which allows all the team members to contribute to the contents of the report at the same time. The use of Google Docs also allows revision history, in case a crucial piece of information needs to be recovered at any moment in time. As for the Arduino code that was utilized, it was uploaded to Github, allowing precise version control for the code that was run for the inverted pendulum system throughout the project.

Failure Mode and Effects Analyses (FMEA)

Top-Down Approach

A top-down approach for designing the inverted pendulum will be utilized in the early stages of the design process to identify failures and logic failures for a product as a whole, and then delve into the causes of its subsequent subsystems and components. This functional approach is a major part of the Failure Mode and Effects Analyses, as it allows the designers to design to cater to the potential failures that can occur in each every system of their designs. Possible subsystems of the inverted pendulum system and their functionalities are listed in the table below.

Subsystem	Function
Pendulum Arm	Connects to the cart and is allowed to rotate freely
Pendulum Cart	Stabilizes the arm by moving linearly
Sensor	Measures the angular and linear parameters of the system
Electronics	Connects multiple components and provides power to the system

Figure 3: Top-down approach table

Some possible ways the inverted pendulum system might experience failures are listed below. In addition, it is important to understand which subsystems the failures affect, which will also be explored.

- The inverted pendulum system cannot stabilize itself

This failure can occur due to a variety of factors such as sensor issues, improper design of the pendulum arm, pendulum cart not supplying enough restoring force, or the electronics not being wired properly. In this case, the inverted pendulum would result in not being able to stabilize itself to the upright position in case a disturbance is applied to it.

- Too much noise detected from the sensor for angular and linear measurements

In case of noise due to sensor measurements in the angular and linear measurements, the sensor would send the noise-filled data to the microcontroller, which would use the closed-loop feedback control system in an attempt to stabilize the system. However, it would detect the noisy data, resulting in the instability of the inverted pendulum.

- Insufficient power supplied to the motors due to the battery

Insufficient power supply to the motors from the battery would result in the pendulum cart not moving fast enough to stabilize the pendulum. Therefore, it is important to choose batteries with enough voltage so that the motors can be run at the appropriate levels. This would impact the pendulum cart, the electronics, and the pendulum arm of the inverted pendulum system.

- Wear and tear of the mechanical components

In the case of long-term usage of the inverted pendulum, the multiple mechanical components that the system is composed of will eventually wear out over time. In this case, the pendulum cart and the pendulum arm will be affected by this wear and tear of the mechanical components. Replacement of these components would need to be done efficiently so that the stability of the inverted pendulum system can be maintained.

Bottoms-Up Approach

Similar to the top-down approach, a bottoms-up approach is a design strategy that is used for identifying specific failures from the subsystems, and then predicting its corresponding effects on the entire subsystem. This structural approach is another major part of the Failure

Mode and Effects Analyses, allowing the engineers to specifically analyze the subsystems of a bigger system. Possible subsystems of the inverted pendulum system, their potential failures, and the corresponding effects on the entire system are listed in the table below.

Figure 4: Bottoms-up approach

Subsystem	Potential Failures	Effects on the system
Pendulum Arm	Wear over time	Over time, the stability of the pendulum arm will decrease, as the wear of the pendulum arm worsens.
	Pendulum arm snaps	The pendulum arm snapping would result in a broken system. Unable to stabilize itself. Also contributes to potential safety hazards
	Wear and tear of pivot joints	Would result in instability of the system over time as the wear and tear of pivot joints worsens
	Fracture or bending of the arm	Fracture or bending of the arm would result in decreasing stability depending on the bending/fracture of the arm
	Loose connections at the pivot joint	Would result in instability of the system. Potential safety hazards in case the connections come off while the system is in motion with people nearby
Pendulum Cart	Worn-out wheels	Would result in decreased friction between the cart and driving surface over time Resulting in instability of the system
	Weak base of the cart	Weak base would result in the cart not being to support the load of the pendulum and other mechanical components on top of it
	Insufficient friction between wheels and surface	Would result in instability of the system if the cart slides too much, not allowing the pendulum arm to stay upright

	Inadequate acceleration of cart	Would result in the pendulum angle to continuously increase without having enough restoring force from the motors to stabilize the system
Sensors	Calibration issues	Inadequate calibration would result in the systematic errors in measuring angle, resulting in instability of the system no matter the gain values of the controller
	Loose wiring	Interrupts the data flow to the sensor, allowing only part of the data to be read by the microcontroller
	Vibration damage from the jittery motion	Reduces the reliability of the sensors, which would likely introduce noise to the measurements
	Large sampling time	Increases the time taken for the sensor to read the angle values. Would result in a larger response time, which would decrease the stability of the system
	Sensor drift over time	Pendulum would naturally deviate from the upright position without the gain values correcting it
Electronics	Insufficient power to motors	The power supplied to the motor would not be enough to move the cart effectively, which would result in the pendulum arm staying upright
	Loose wiring to the microcontroller	Interrupts the data flow to the sensor and the motor, allowing only part of the data to be read and transmitted by the microcontroller
	Inadequate battery size	The power supplied to the motor would not be enough to move the cart effectively, which would result in the pendulum arm staying upright
	Battery depletion	Decreases the amount of power sent to the motor over a large amount of time without being charged. Would result in instability of the system

Ishikawa (Fishbone) Diagram

An Ishikawa diagram, also known as the Fishbone diagram, is a tool of the Failure Mode and Effects Analysis which is used to identify a system's subsystems and the potential failures that can occur within those subsystems. Aligning with the project at hand, the spine of the diagram represents the inverted pendulum system itself. The ribs of the diagram demonstrate the several subsystems of the inverted pendulum system, which are: the pendulum cart, the pendulum arm, the sensor, and the electronics. Additionally, the secondary ribs outline the potential failures that can occur within each subsystem.

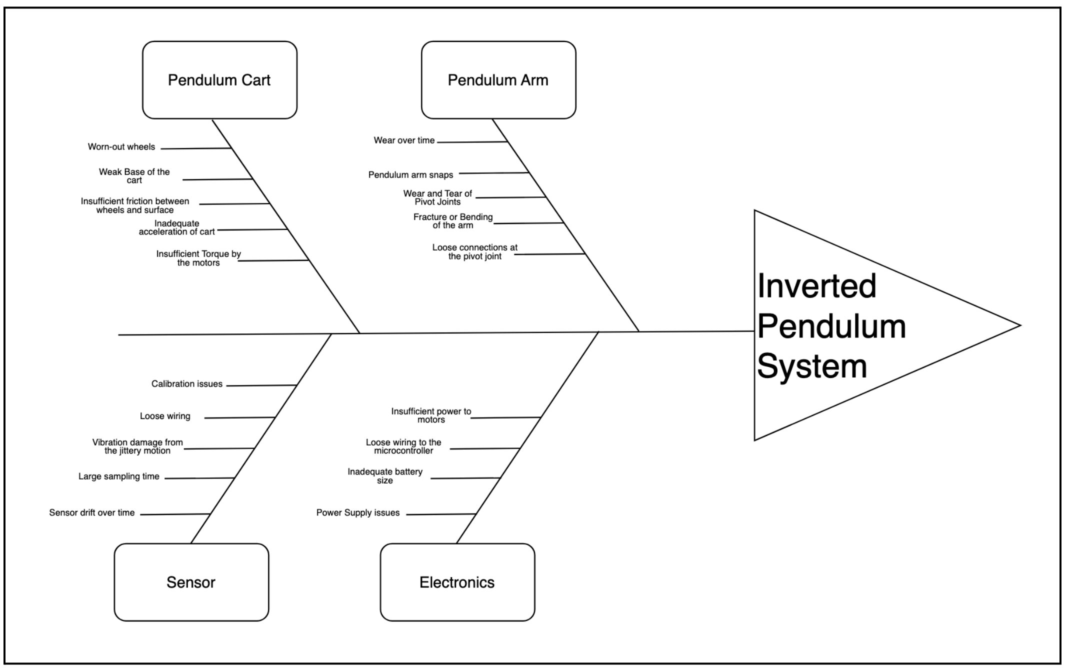


Figure 5: Ishikawa diagram of the mechanism

FMEA Form

FAILURE MODE AND EFFECTS ANALYSIS

Item:	Inverse Pendulum Cart Mechanism	Responsibility:	W. Martin	1
Model:	Current	Prepared by:	W. Martin	1 of 1
Core Team:	William Martin, Sarvin Jeevakaran, Harsh Singh, Amish Salawan			11/22/2024 Rev: 1

Process Function	Potential Failure Mode	Potential Effect(s) of Failure	Sev	Potential Cause(s)/ Mechanism(s) of Failure	Occur	Current Process Controls	Detec	RPN	Recommended Action(s)	Action Results: Actions Taken	Sev	Occ	Det	RPN
Movement Control of Wheels	Misallignemnt/slippage	Pendulum cannot stablize effectively	7	Improper machine set up and inproper installation of wheels	5	Regular inspection of wheels and alignment	4	140	Use wheels with better grip, ensure even installation	Replaced wheels with rubber-coated ones and adjusted mounting for alignment	6	3	3	54
Pendulum movement when under failure	Pendulum falls and strikes Arduino board	Arduino board is damaged, leading to system failure	9	Improper machine set up	4	Making sure that when the pendulum seems to be falling with full momentum, we kill and stop the mechanism	8	288	Relocate Arduino to a safer position and add a physical barrier to protect it	Relocated Arduino to a more protected position on the chassis; 3D-printed a base for the pendulum to shield the Arduino.	5	2	3	30
Powering the Arduino	Power supply cord too short to reach Arduino	Arduino cannot be powered, causing the system to not start	8	Positioning of Arduino too far from power source; fixed-length power cable	3	Ensuring the positions of all the compontents on the chassis fit and pose no obstructions with connections	9	216	Relocate Arduino closer to the power source or modify the design to bring the power supply closer.	3D-printed the pendulum base, which provided a new mounting point closer to the power source, resolving the issue.	2	1	3	6
Creating the pendulum base	The original 3-D printed pendulum base too thin and brittle	Pendulum base breaks under load, causing instability or damage to components	7	Incorrect print settings (low infill or thin walls), unsuitable material choice	5	Inspection of the printed base	5	175	Increase base thickness, use higher infill density, and consider a stronger material like PETG or ABS	Reprinted a stronger base with increased thickness and infill density using PETG for durability.	2	1	2	4
Pendulum attachment	Dowel loosens from chassis mount	Pendulum falls, destabilizing the system	8	Insufficient clamping or loose connection	4	Manual check for tightness	7	224	Use stronger adhesive on dowel holder or switch to a sturdier attachment method	Applied suppler glue adhesive and reinforced the dowel mount with an additional bracket on each side	7	3	5	105

Figure 6: FMEA Form

Possible risks in the inverted pendulum balancing system are evaluated using the Failure Modes and Effects Analysis (FMEA) chart. In order to guide corrective operations, it identifies the failure modes, their occurrence, severity, and detection. Wheel misalignment/slippage obtained a severity rating of 7, a detection rating of 4, and an occurrence rating of 5. The RPN was lowered from 140 to 54 by using improved grip wheels and making sure the installation was done evenly.

The severity of the pendulum hitting the Arduino board was 9, the occurrence was 4, and the detection was 8. The RPN was lowered from 288 to 30 by moving the Arduino and adding a 3D-printed platform for protection. The severity was 8, the occurrence was 3, and the detection was 9, when the power supply cord was too short. In order to move the Arduino closer to the power source and lower the RPN from 216 to 6, a new pendulum base was 3D printed. Lastly, with a severity of 7, occurrence of 5, and detection of 5, the original pendulum foundation was very thin and brittle. The RPN decreased from 175 to 4 when we recreated the base using thicker, tougher material.

Design Process

Problem Definition

The general objective of this project is to design a mechatronics system that is capable of stabilizing an inverted pendulum using different types of controllers. The problem can be defined based on four categories, which are the product characteristics, functional requirements, constraints, and performance metrics.

Product Characteristics

Firstly, an important aspect is that the system must be designed to be safe as to avoid posing a risk to users. Due to the nature of the inverted pendulum mechanism that is to be implemented, the system will include moving masses which could possibly injure users during real world operations. This risk to physical safety must be minimized for this design.

The system must also be designed to be reliable and consistent. The design must be able to achieve the same end results for any set of conditions to endure the system can effectively perform its intended task of balancing the pendulum.

Other than the performance and safety-based product characteristics, it is also important that the length of just the pendulum should be between 40 to 50 centimeters long, while the mass is between 50-100 grams.

Functional Requirements

One function of the designed system is that it must stabilize the angle of the pendulum. This requirement is important as it is the main aspect of this design problem as a whole. This function also includes being able to respond to disturbances to maintain the balance of the pendulum.

Another function of the design is it must prevent damage to its surroundings. With the movement of the pendulum being a possible safety hazard, the design must take into account the ability to prevent significant impacts to both protect users and the system itself.

Constraints

A constraint for the design is that it must maintain the angle of the pendulum within 2° of the vertical. This is required to ensure the results of the system are within an acceptable range for stabilization of the pendulum.

One more constraint that will affect the design is the system must be capable of responding to disturbances within 0.5 seconds of them being applied. This is to ensure the design is made to be both effective and efficient.

There are constraints that are specified within the design problem with respect to the generic design of the inverted pendulum system. In addition to the functional constraints described above, it is also important to create a design that would cater to the constraints defined within the design problem. These constraints are listed below:

- The pivoting joint must not have an actuator attached to it, and must rotate freely.
- Only the electric components within the ELEGOO kit should be used.
- A financial budget of $100 is available for purchasing any item that is not available.
    - It is important to mention that this budget does not include the cost of 3D printing any relevant parts.

Performance Metrics

The main performance metric of this design would be for the angle of the stabilized pendulum. The error between the desired and actual pendulum angle will be used as a reference to evaluate how well the design performed its primary task.

To gauge the safety of the system, the design will be rated on a 5 point scale based on the perceived danger or risk to users and the environment. This will give a reference to the safety of the design and how it should be modified.

To evaluate the efficiency of the design, the actual and desired response times will be used. The error between these values will give a reference to how well the system performs under applied disturbances.

Problem Definition Chart

Figure 7: Problem Definition Chart

Product Characteristics	Functional Requirements	Constraints	Performance Metrics
Reliable	Consistently Stabilize Pendulum	Stable Angle = 0 ± 2° Response Time < 0.5 sec	Error = (Desired - Actual) / Actual
Safe	Prevent damage to surroundings	Limit range of pendulum rotation	Safety = (5 - Rating) / 5

Solution Formation

After thoroughly analyzing the problem and its subsequent categories such as the Product Characteristics, Functional Requirements, etc., a series of iterative steps need to be performed before a final conclusion on the design of the inverted pendulum can be decided. This section focuses on these steps, and discusses in detail the qualities and features of the design that is required for the inverted pendulum and its corresponding system.

Mind Map

To gain a better understanding of the problem and how to formulate a solution, the problem was broken down into various sub-problems. Each of these sub-problems can be analyzed individually to determine what systems or components can be implemented to achieve certain functions.

The design problem was broken down to the following sub-problems; angle measurement, response system, physical interface, and control methods. For each of these problems, multiple different solutions were determined and the most appropriate options were selected to form the ideal solution.

For the problem of measuring the pendulum angle, two types of sensors were considered. The first was a rotary encoder, which is a sensor that measures rotational motion by converting angular position data to an analog signal. This solution would require the sensor being mounted at the pivot point of the pendulum. The second option was to use an accelerometer and gyroscope module. This is a device that is capable of measuring linear accelerations and angular velocities of an object. This solution would allow for the sensor to be mounted anywhere on the pendulum.

For the response system problem, the main goal for the solution was to provide a method to move the pendulum back to a stable position. The first solution for this was to use a cart base for the pendulum. This would involve affixing the pendulum at a pivot placed on top of a linearly moving cart which would move left and right in response to the pendulum angle. The second option would be to attach the pendulum to a linear actuator. Similar to the cart solution, the pendulum would pivot and be moved left or right based on the pendulum angle. One more solution was to make use of propellers affixed to the top of the pendulum. This solution would require using the propulsion force from the propellers to rotate the pendulum based on its angle.

For the problem of the control system, two solutions were considered. The first was PID control, which is a control method that uses calculated error between actual and desired inputs as well as gain values to determine the required output to reduce the error. In the case of this design the error from the pendulum angle would be used to control the response system. The second option was using pole placement. This is a control method that uses the properties of a system to determine the required gain output to obtain specified poles. In the case of this design, this method would determine the control of the response system to obtain stable poles based on the desired angle and motion of the pendulum.

For the problem of providing a physical interface to the system two solutions were considered. The first solution was the implementation of a set of wired push buttons to the system that could be assigned to different functions such as shutting down the system. The second option was to implement wireless communications. This would especially serve the same purpose as the wired buttons but with the advantage of less wires.

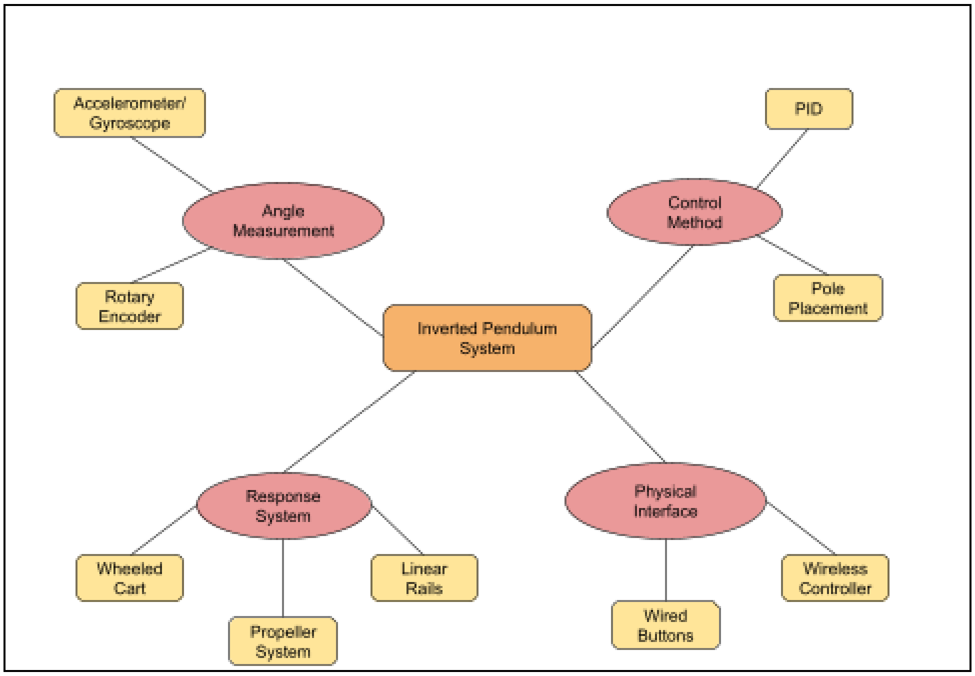


Figure 8: Solution Formulation Mind Map

From analyzing the solutions of each sub-problem, an initial design was formulated based on what was thought to be the most effective and appropriate to achieve the design goals. For the angle measurement solution, the accelerometer and gyroscope module was chosen due to its versatility in measuring multiple variables in different directions, allowing for more flexibility in the construction of the design. For the response system, the cart solution was chosen. The cart was chosen because the use of a wheeled base would allow for more freedom for movement without being too complex. For the control method both PID and Pole Placement was chosen to demonstrate how the system can function with different types of controllers. Finally for the physical interface the use of wired buttons was chosen due to its simplicity compared to wireless interfacing.

Black Box Model

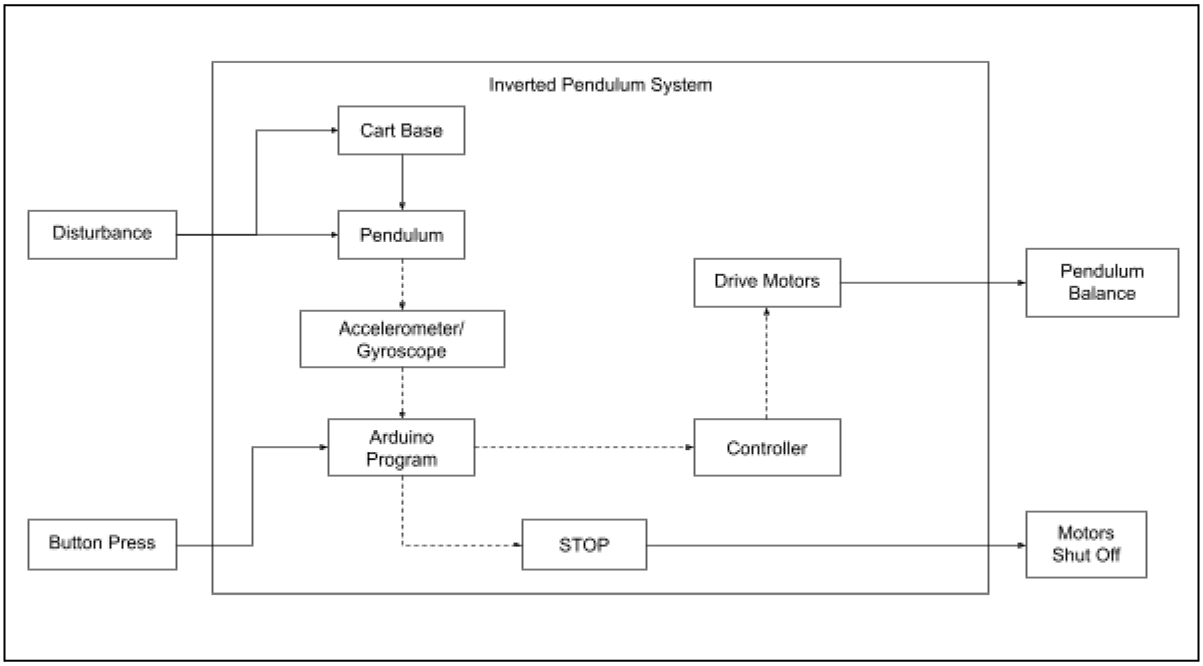


Figure 9: System Black Box Model

When formulating a design using the selected solution the system was analyzed based on its main inputs and outputs. For the inverted pendulum system the main input of interest was the disturbance applied to the pendulum or cart. As for the output, the main one was the balancing and stabilization of the pendulum. Based on this information the sub-system solutions were laid out showing the relations between them. The interactions of the sub-systems were primarily linear with the following order of functions. The pendulum would receive a disturbance force either directly or indirectly from the cart which would then send information to the accelerometer and gyroscope module. This information would then be sent to the arduino program to be interpreted and sent to the controller code, which would in turn drive the motors to balance the pendulum. This model describes the primary function of the design solution that will be implemented.

Human Machine Interface

This section of the Solution Formation covers the instances where the inverted pendulum system comes in contact with a human, whether that being a user, bystander, or a potential customer. Catering primarily towards the safety first, and then the ergonomics, this section will delve in detail about the features that would elevate this iteration of the design from other

Due to the nature of the project and its corresponding design, it is important to include an ‘E-STOP’ button. The purpose of the ‘E-STOP’ button is to terminate all operations of the system when the button is pressed. In the off-chance that the inverted pendulum does seem to hit an unlikely object, the E-STOP button can be pressed to void all operations at that moment in time to prevent any unwanted collisions or accidents. It is also important that this E-STOP button is located in a position on the system that is swiftly and easily accessible. In addition, it is also important that the appearance of the E-STOP button is easily distinguishable from other features of the inverted pendulum system, so that it can be easily identified during a moment of panic. Implementation of these two features through the E-STOP button will result in termination of any/all operations that the system is executing when an unwanted situation occurs.

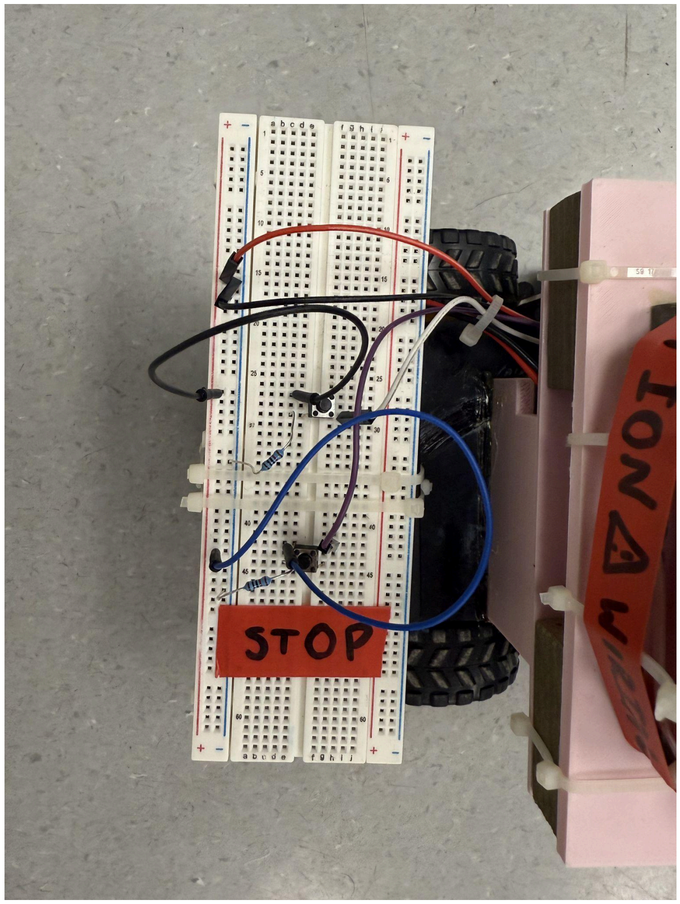


Figure 10 - ‘Emergency Stop Button’ to stop the motion of the cart

Additionally, there are labels throughout the cart that alert the humans nearby of the potential dangers of the inverted pendulum system. The Figure 11 cautions the users about the open wiring of the inverted pendulum system. Open wiring is a massive danger to the users and the system as, firstly, it would pose a great safety hazard to the user operating the system. Secondly, in the case of the open wiring, it is easily able to become loose, allowing only partial data flow to the sensor and the motors. The color scheme of red was chosen with black font color so that the contrast is good. Doing this allows the users to quickly identify and understand that there is something to be cautioned about regarding the design.

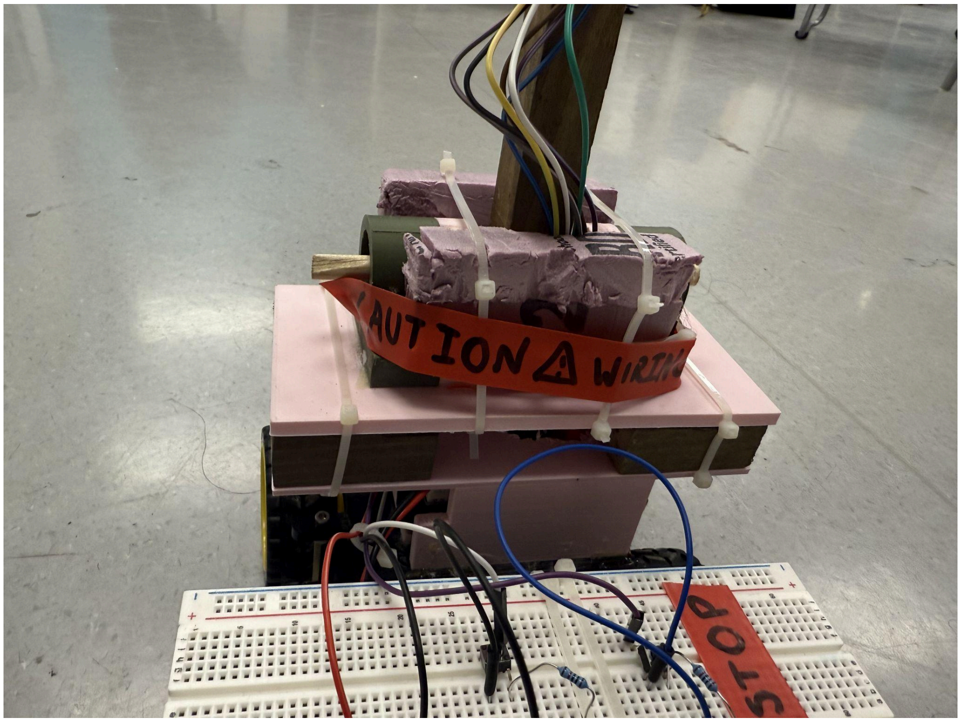


Figure 11 - Open wiring to caution the users

Another label was added to the system that cautions the users about the random and powerful movement of the inverted pendulum system. Once again, it allows the users and nearby people to understand that the mechanism moves with erratic movement and directions. The color scheme for this label was done using yellow background and black font color, allowing the label to catch the user's attention and give them a moment to read the label.

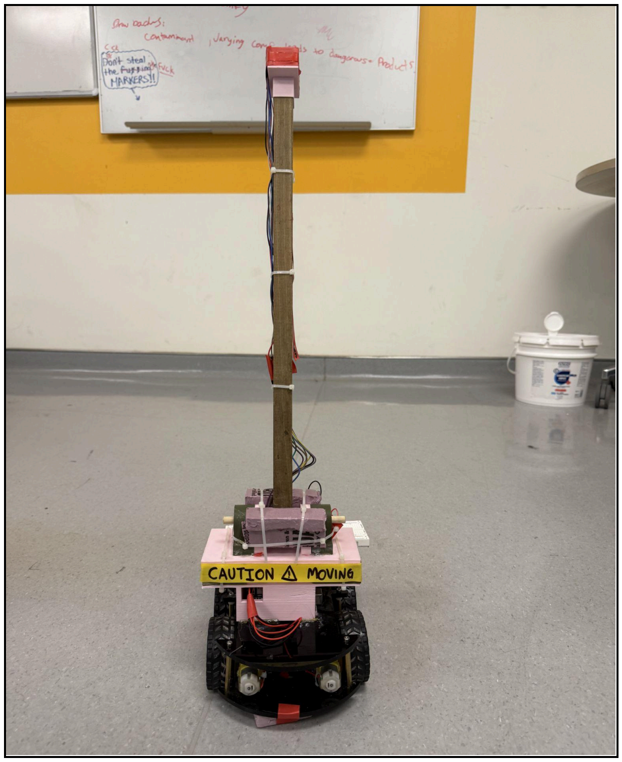


Figure 12 - Label allowing the user to understand that the inverted pendulum causes erratic movements

MBSE Diagrams

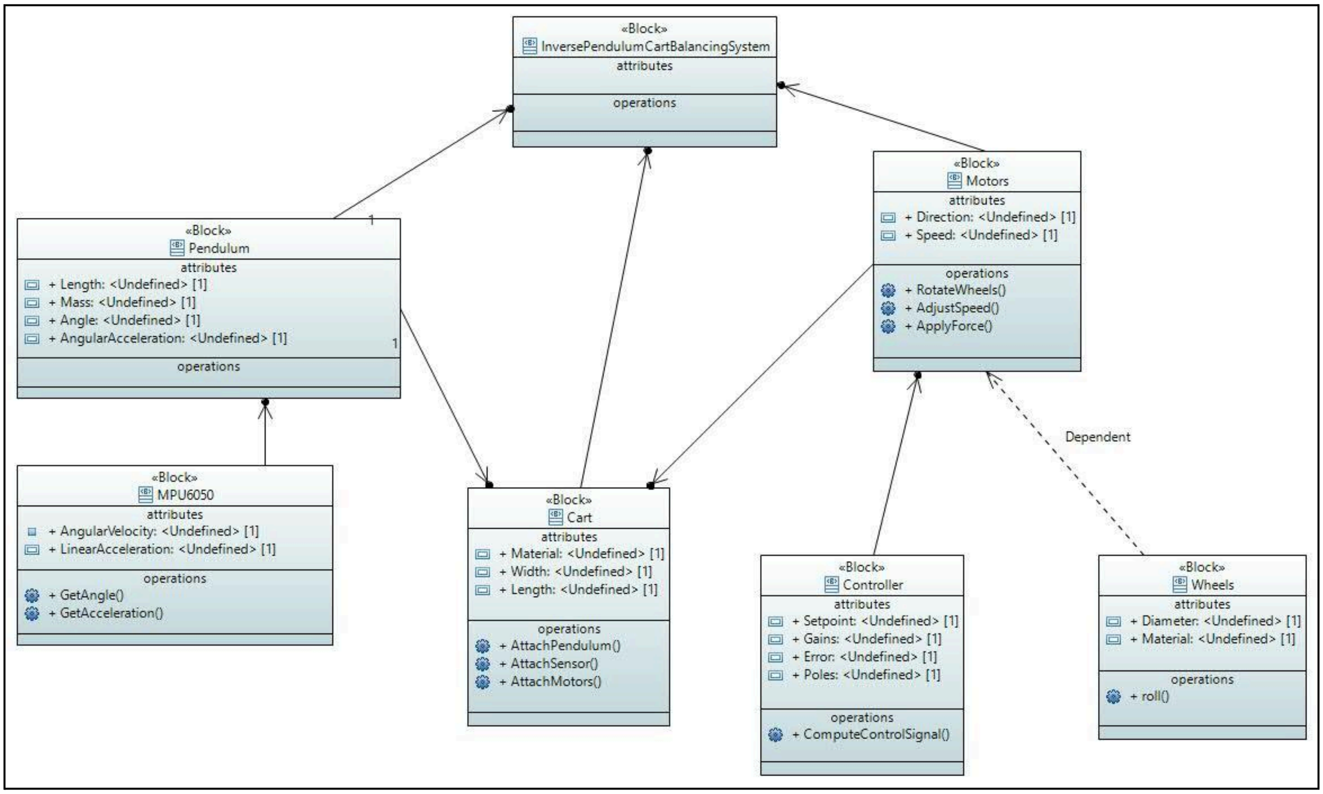


Figure 13: MSBE Block Definition Diagram

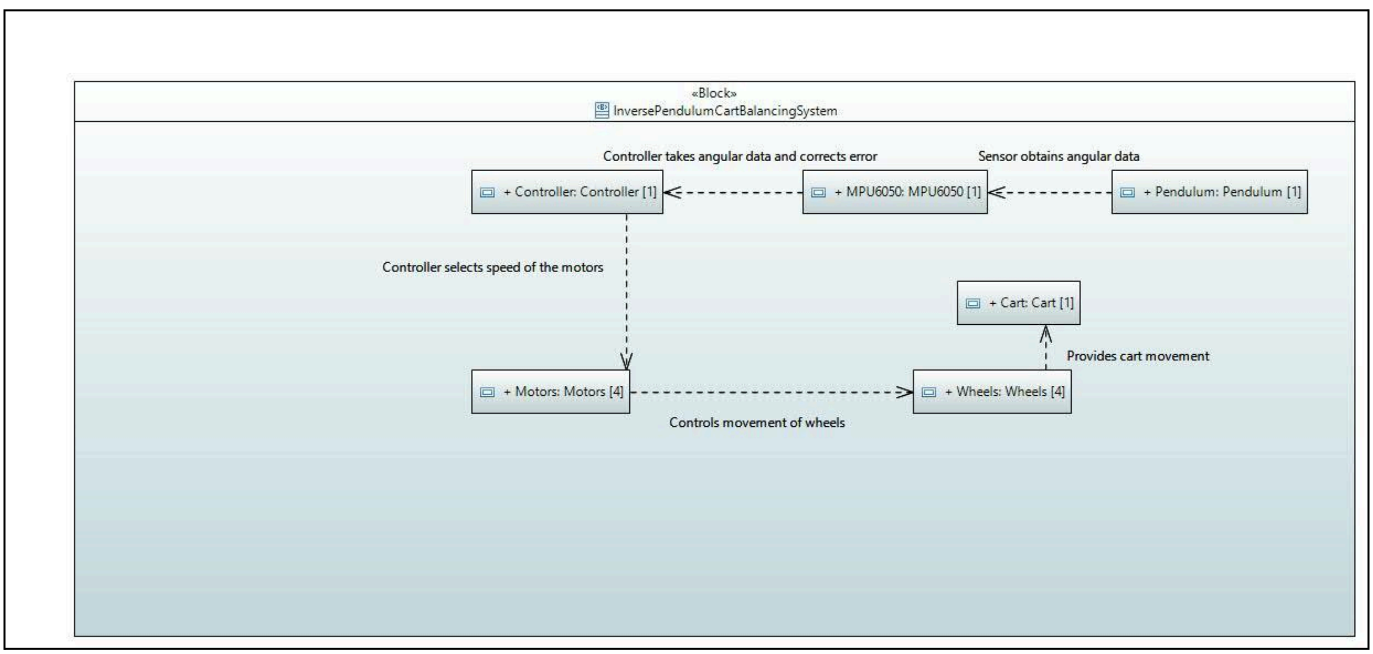


Figure 14: MSBE Internal Block Diagram

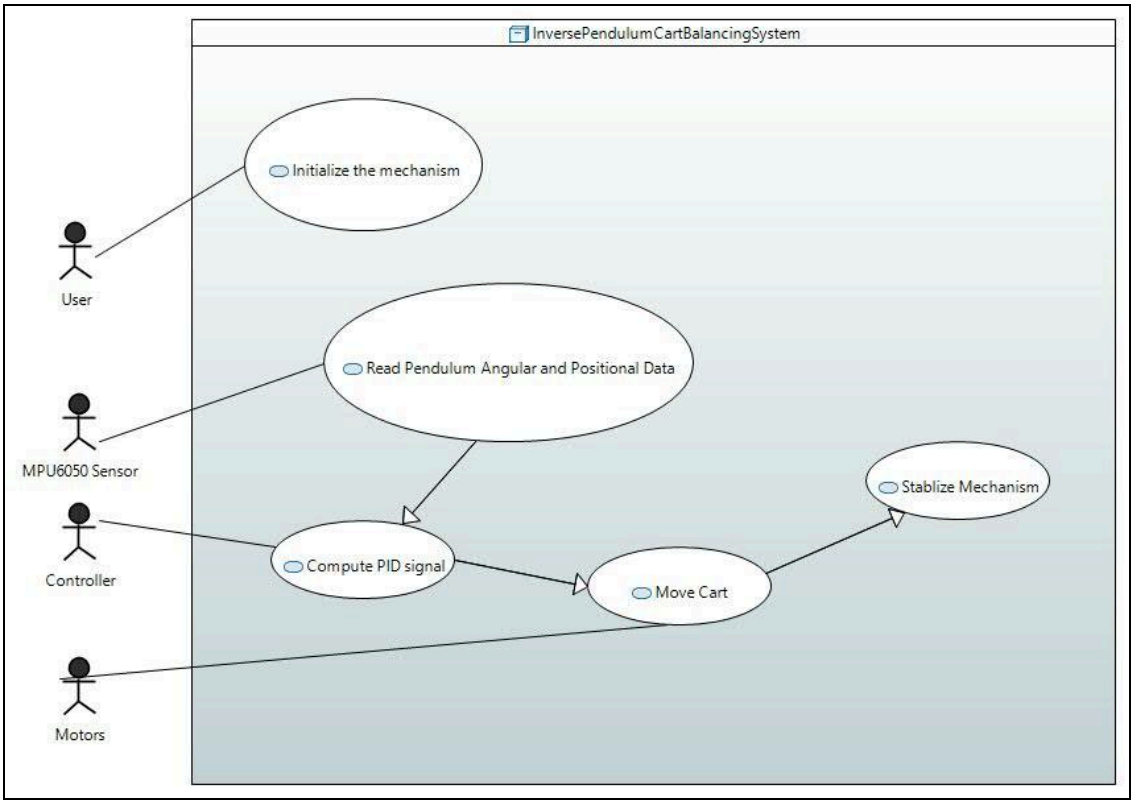


Figure 15: MSBE Use Case Diagram

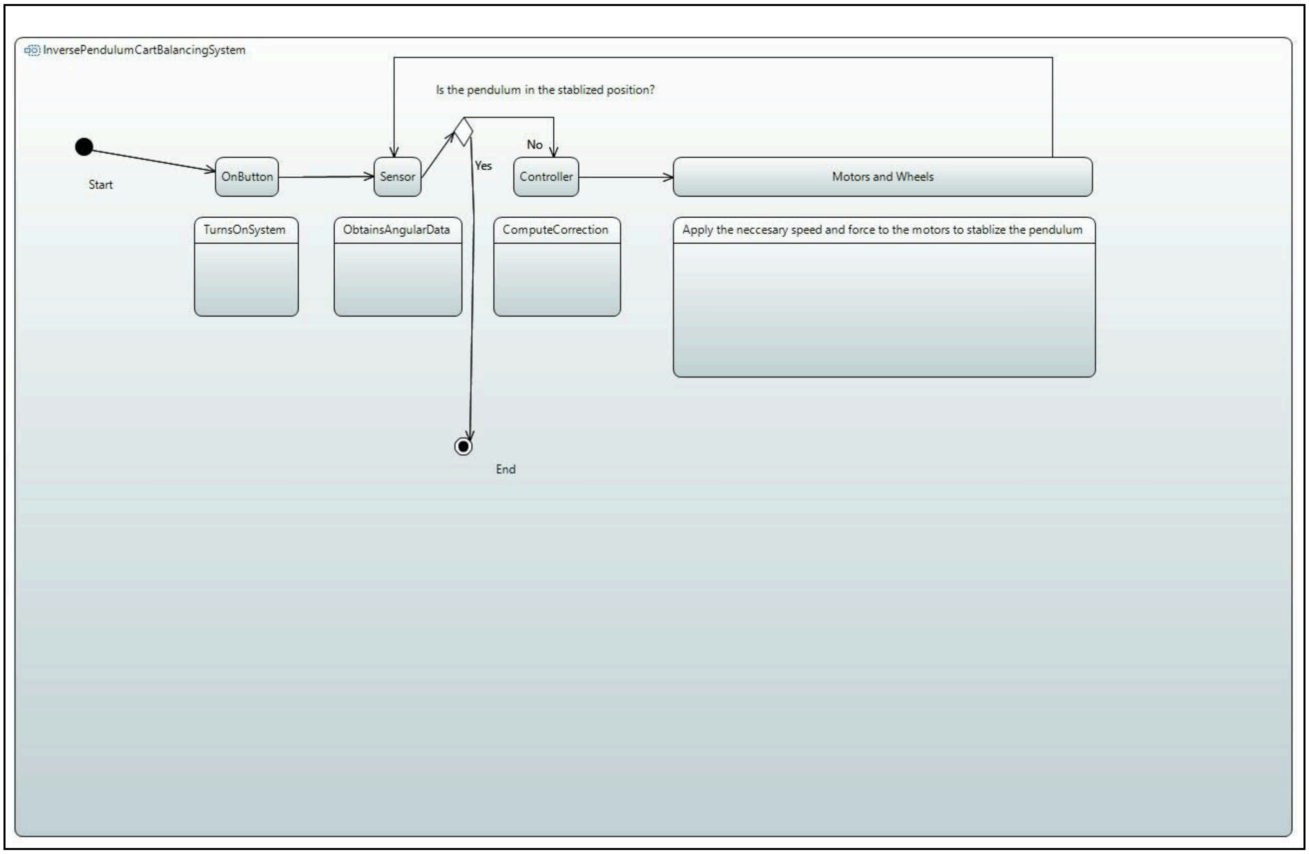


Figure 16: MSBE Activity Diagram

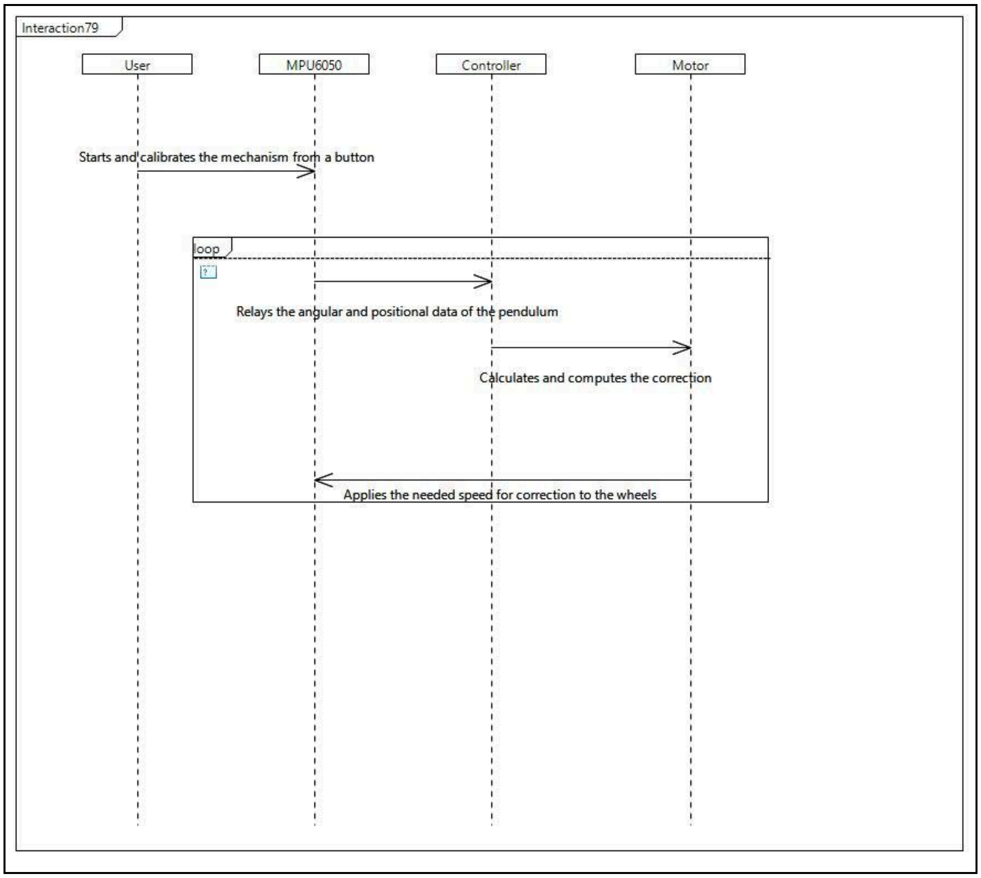


Figure 17: MSBE Sequence Diagram

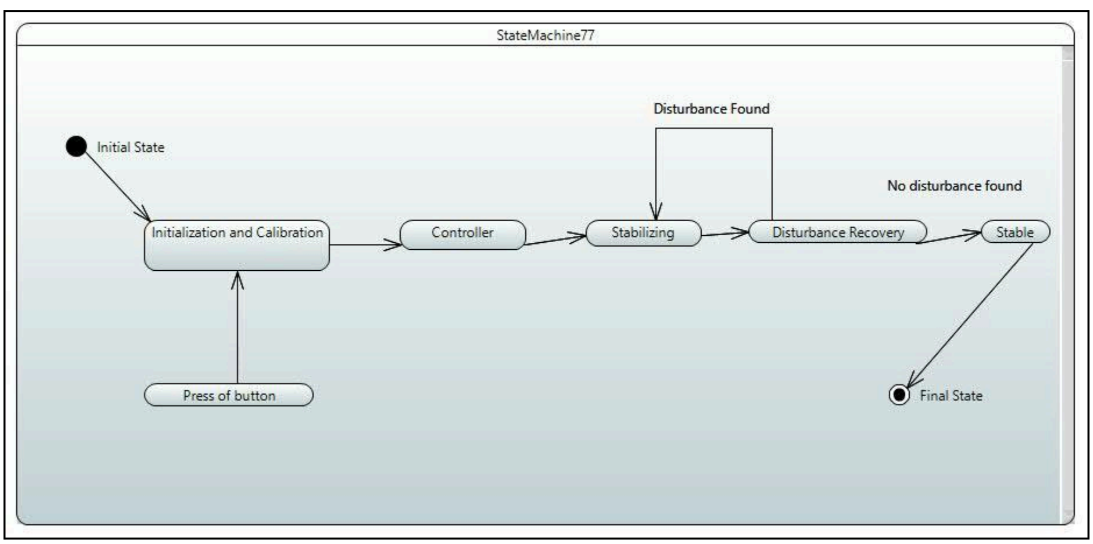


Figure 18: MSBE State Machine Diagram

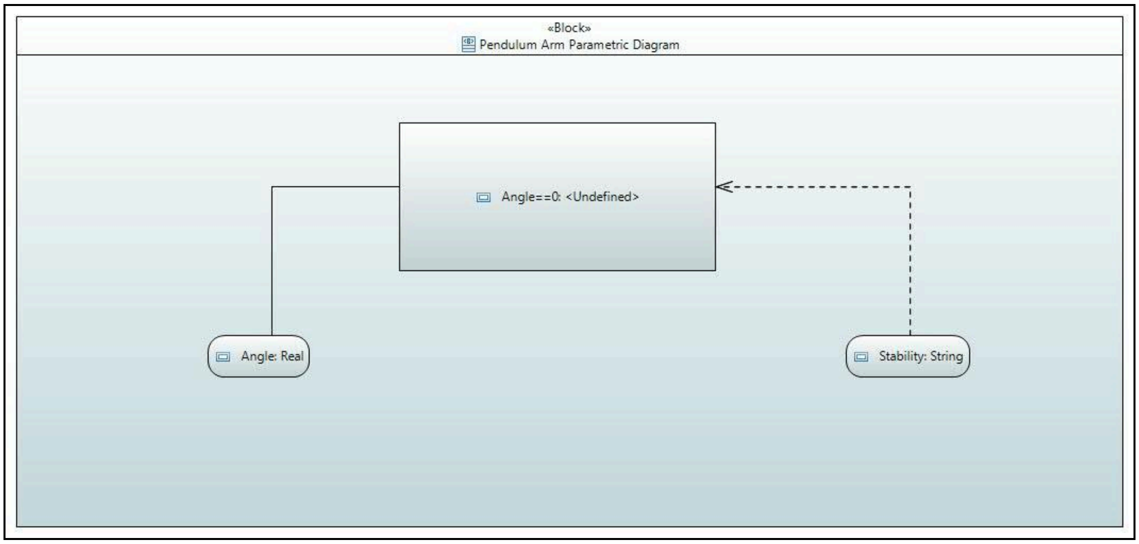


Figure 19: MSBE Parametric Diagram

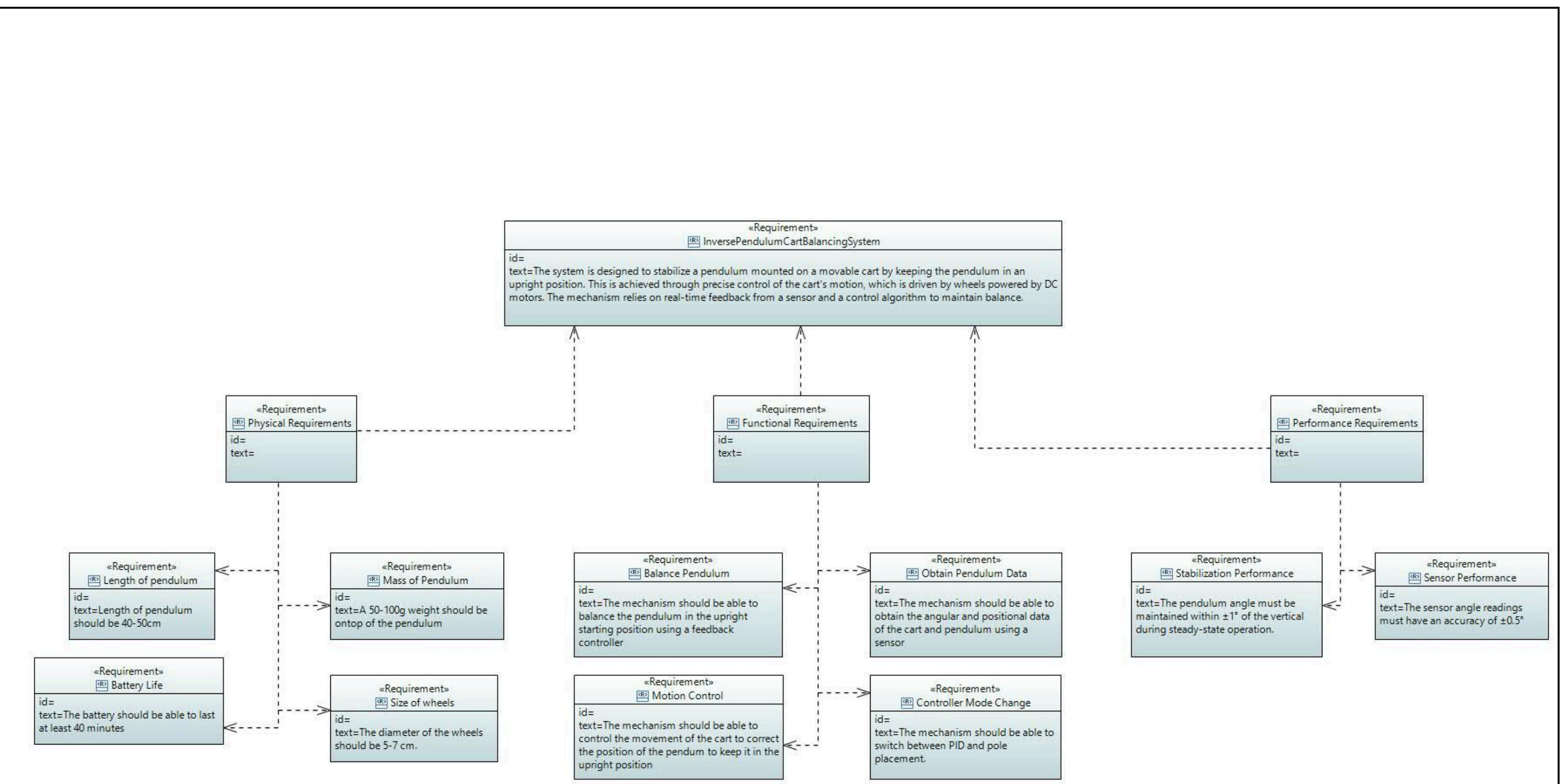


Figure 20: MSBE Requirements Diagram

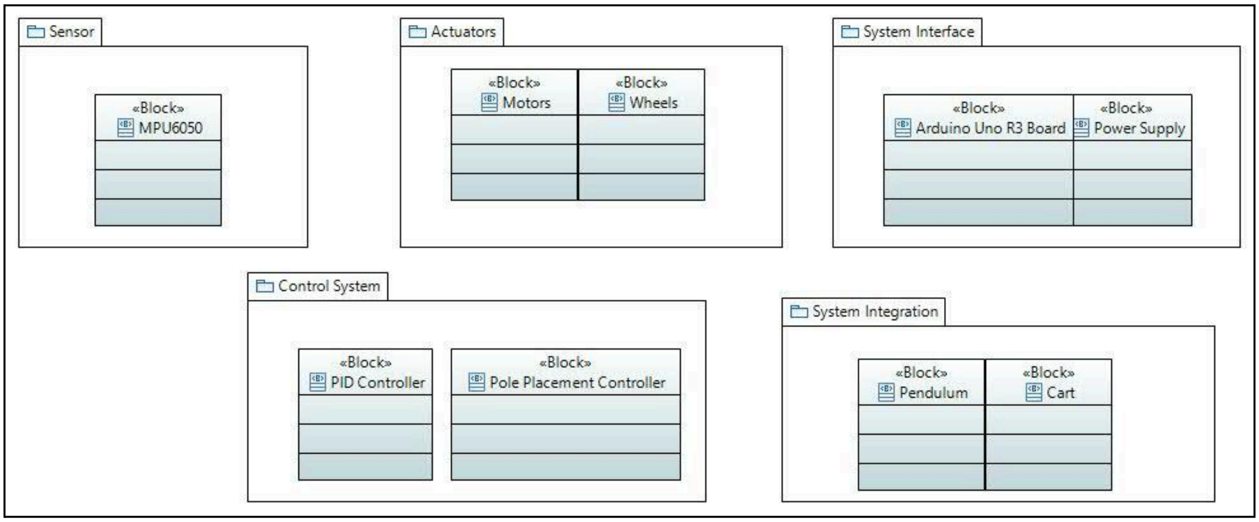


Figure 21: MSBE Package Diagram

Modeling

Mathematical Modeling

In order to implement the inverted pendulum and the control systems that will stabilize it, it is necessary to develop mathematical equations using governing laws and principles that demonstrate the motion of the inverted pendulum. There are two equations of interest; the cart's linear motion, and the pendulum's angular motion. While there are a few ways of determining these equations of motion, this project will focus on the Lagrangian method, which focuses on the difference in kinetic and potential energy, rather than Newton's complex forces [2]. The mathematical modeling using the Lagrangian method and the parameters in Figure 1 was done in the following way to determine the system's equations of linear and angular motion

$$Lagrangian = Kinetic\ Energy\ [T] - Potential\ Energy\ [V]$$

$$\Rightarrow T = \frac{1}{2}M\frac{dx}{dt}^2 + \frac{1}{2}m[\frac{dx}{dt}^2 + (l\frac{d\theta}{dt})^2 + 2l\frac{dx}{dt}\frac{d\theta}{dt}cos\theta]$$

$$\Rightarrow V = -mglcos\theta$$

The Pole Placement controller requires that the system's equations of motions are in a linearized form. Using the principle of small angle approximation, this can be done in the following way

$$\Rightarrow (M + m)\frac{d^2x}{dt} - ml\frac{d^2\theta}{dt} = F(t)$$

$$\Rightarrow ml\frac{d^2\theta}{dt} + ml^2\frac{d^2\theta}{dt} = 0$$

Schematics

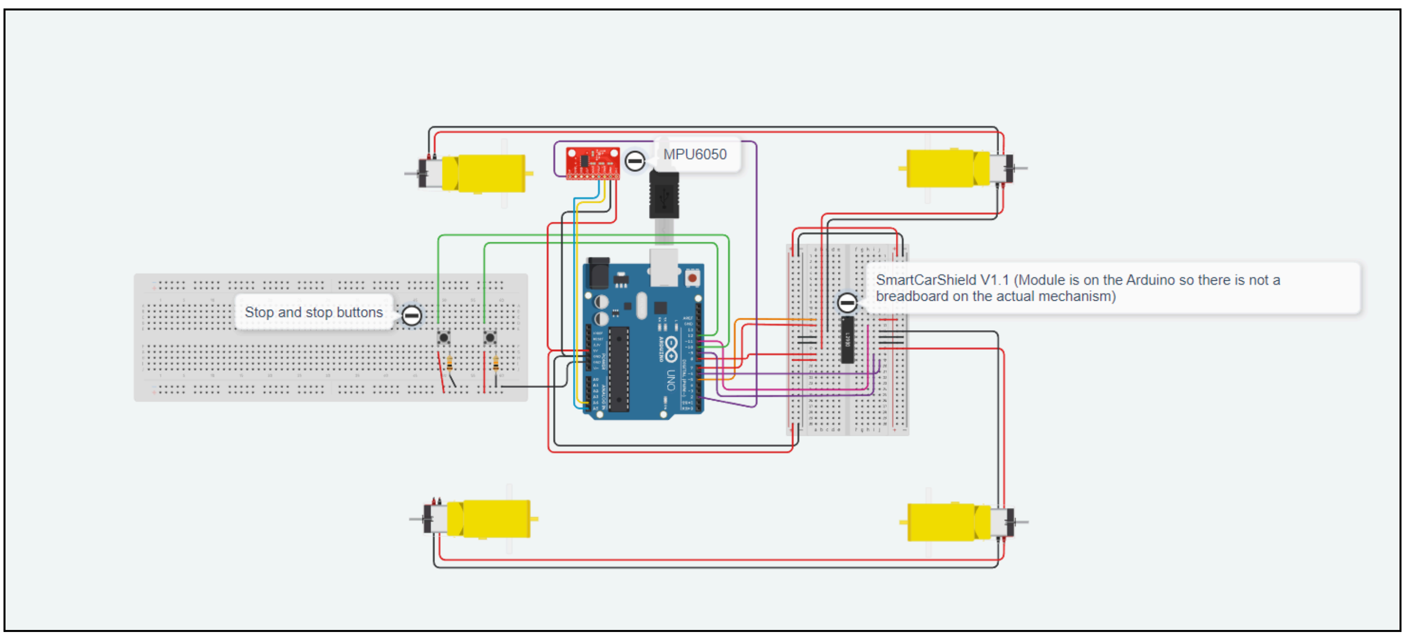


Figure 22 - Wiring Diagram (1)

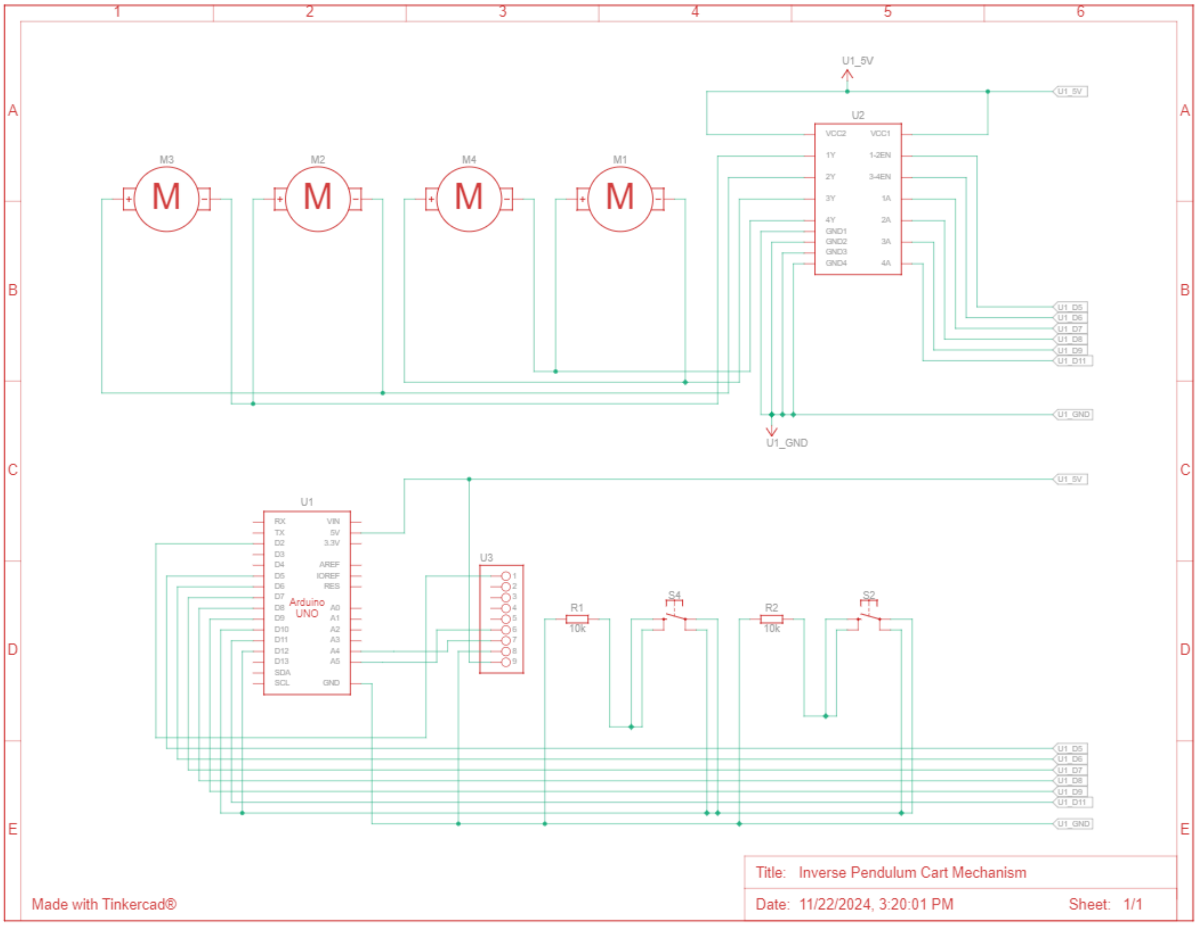


Figure 23 - Wiring Diagram (2)

<u>CAD</u>

In this project, CAD modeling focused on designing components integral to the inverted pendulum system, including a Weight Holder Pendulum. This project required an iterative design process to achieve the required dimensional accuracy for stability and compatibility with other system components. Modeling in this work was done using SolidWorks, based on the dimensions of parts such as the Arduino or bearings, and tolerances were made for fitment.

For images of the engineering drawings please refer to the appendix or consult the files attached to the submission.

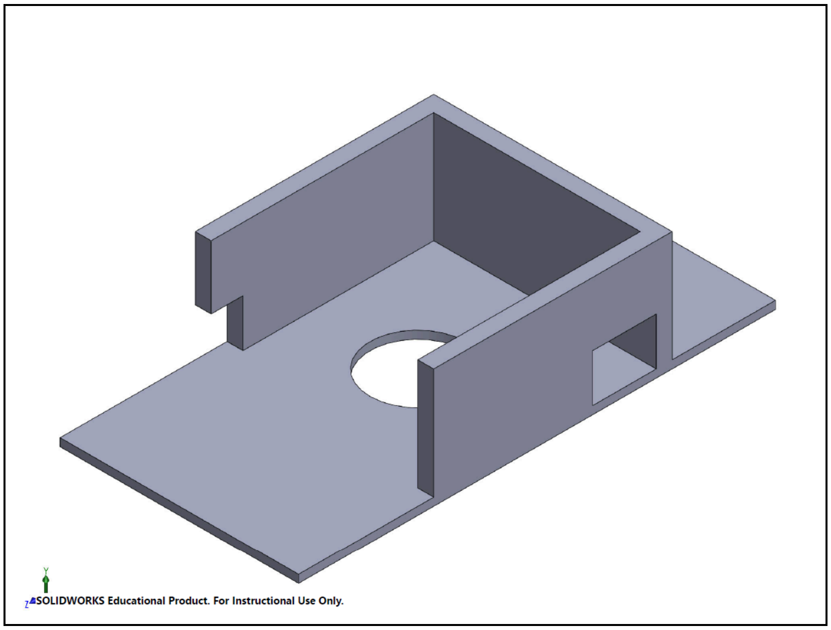


Figure 23: CAD image of Arduino Cover

This design was created to house the base of the pendulum holder, allowing the pendulum to remain in the center of the cart. Many iterations and designs were made but this final design was chosen due to its structural integrity.

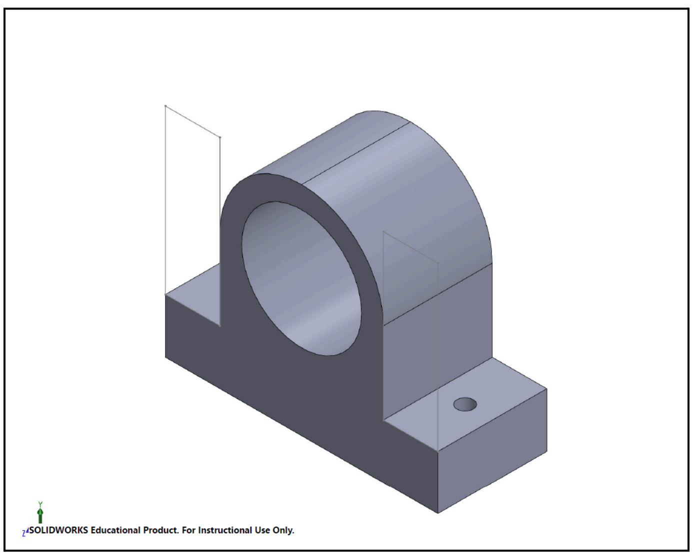


Figure 24: CAD image of Pendulum Holder (old)

The initial design for the pendulum holder design was made around a bearing. It was thought that the bearing would help the pendulum have smoother rotation. While this was true, it was allowing for too much motion of the pendulum. This made it difficult to stabilize the pendulum as it would move in a direction too freely. It was then determined that a new holder had to be built.

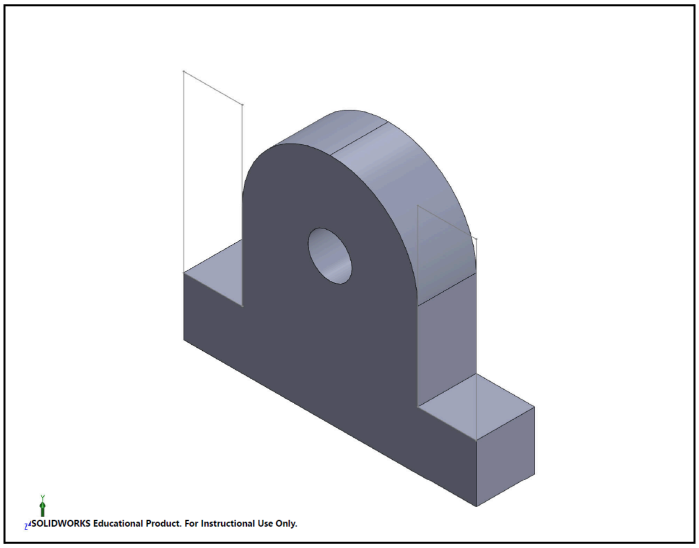


Figure 25: CAD image of Pendulum Holder (New)

The new pendulum holder design included a smaller hole for the axle of the pendulum. This created a slightly greater resistance for the pendulum to move, allowing for finer movement/adjustments when stabilizing the pendulum.

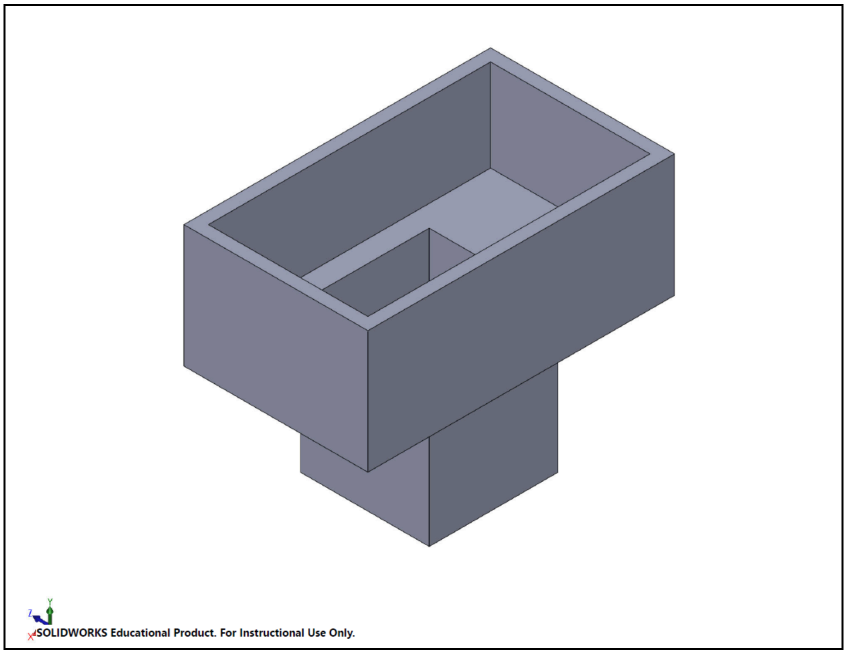


Figure 26: CAD image of Weight Holder

This design was made to attach to the pendulum, house 50g of weight, and the MPU6050. It was attached using glue but was designed to have a tight fit to ensure the system had constant weight. The MPU6050 was attached on top using tape to ensure data measurements were accurate and no slipping occurred.

Simulation

The simulation was created to model the system virtually to test its behavior. Using MATLAB Simulink a model of the system was constructed which consisted of two main geometrical components, being the cart and the pendulum. Simple geometry was used to model the components with the cart and pendulum being represented by rectangular boxes. The system model was set up to measure the position of the cart and the angle of the pendulum as well as receive a force and an input. To simulate the PID control structure a PID controller block was implemented which calculated error between a defined set point of zero degrees and the angle of the pendulum. The PID output was used to calculate response force. The geometric system itself was connected to output the cart position and pendulum angle with it taking force input from the PID controller. Impulse forces were also added to simulate the application of disturbances to the system. This configuration allowed for the behavior of the system to be observed in terms of how well the PID controller stabilized the pendulum. The system was tested by setting an arbitrary disturbance and modifying the PID gains. Trial and error was used to determine the gain that provided adequate stabilization to the pendulum. The results of this simulation were used as a reference for the constructed prototype control system.

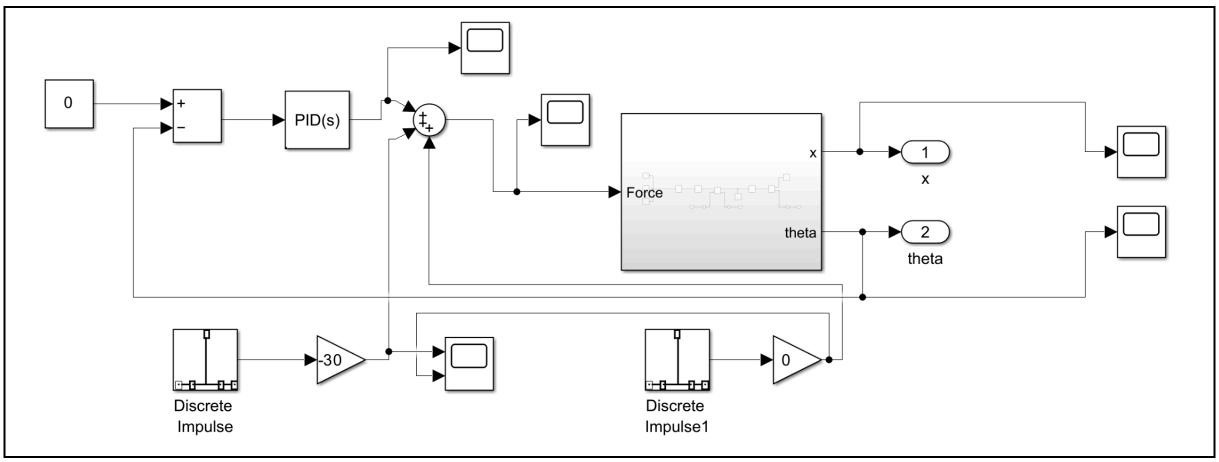


Figure 27: Simulink Simulation Block Diagram

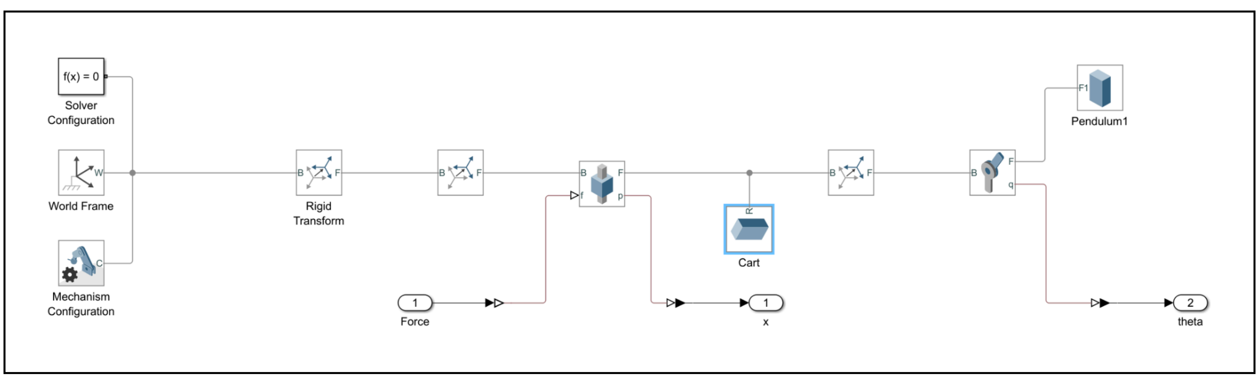


Figure 28: Simulink Inverted Pendulum Geometric Model

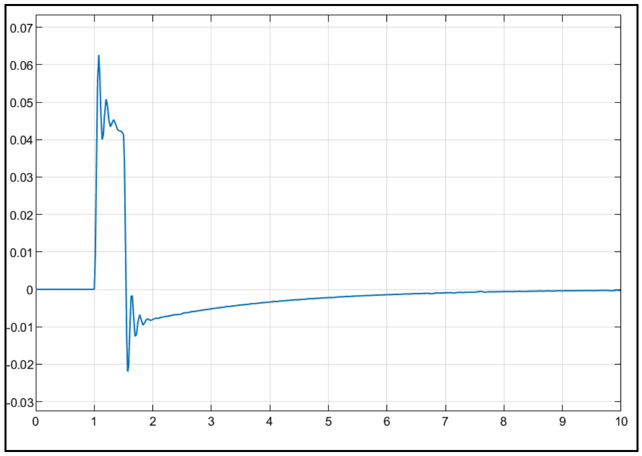


Figure 29: Pendulum Angle Plot

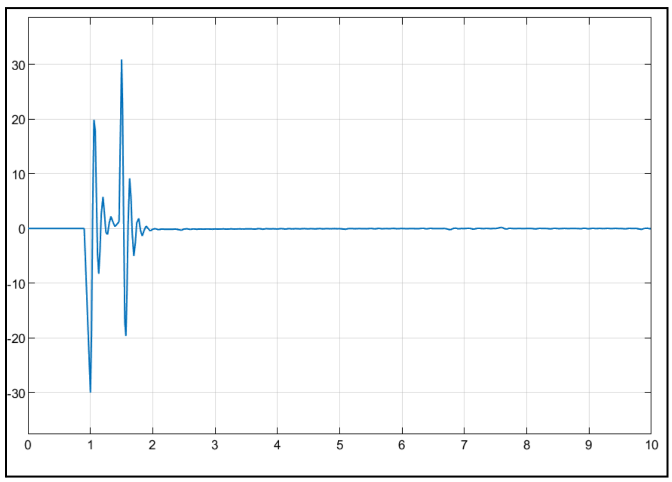


Figure 30: PID Output Plot

Implementation

The design for the implementation and prototyping of the solution had three main categories: mechanical construction, electrical system, and programming.

Mechanical Design

The mechanical portion of this design includes the pendulum itself, the cart base,and the pivot point of the pendulum. The pendulum was constructed out of wood and made to be 45cm long. The pendulum also had an additional component affixed to the top of it to carry a tip mass ranging from 50g to 100g. For the constructed design a tip mass of about 50g was chosen. The cart base serves primarily as a platform to mount the other components such as the DC motors, Arduino microcontroller, and pendulum. To allow for free rotation of the pendulum, the connection between the cart base and the bottom end of the pendulum was constructed as a pinned joint. This mechanical design shows the basic structure of the inverted pendulum system.

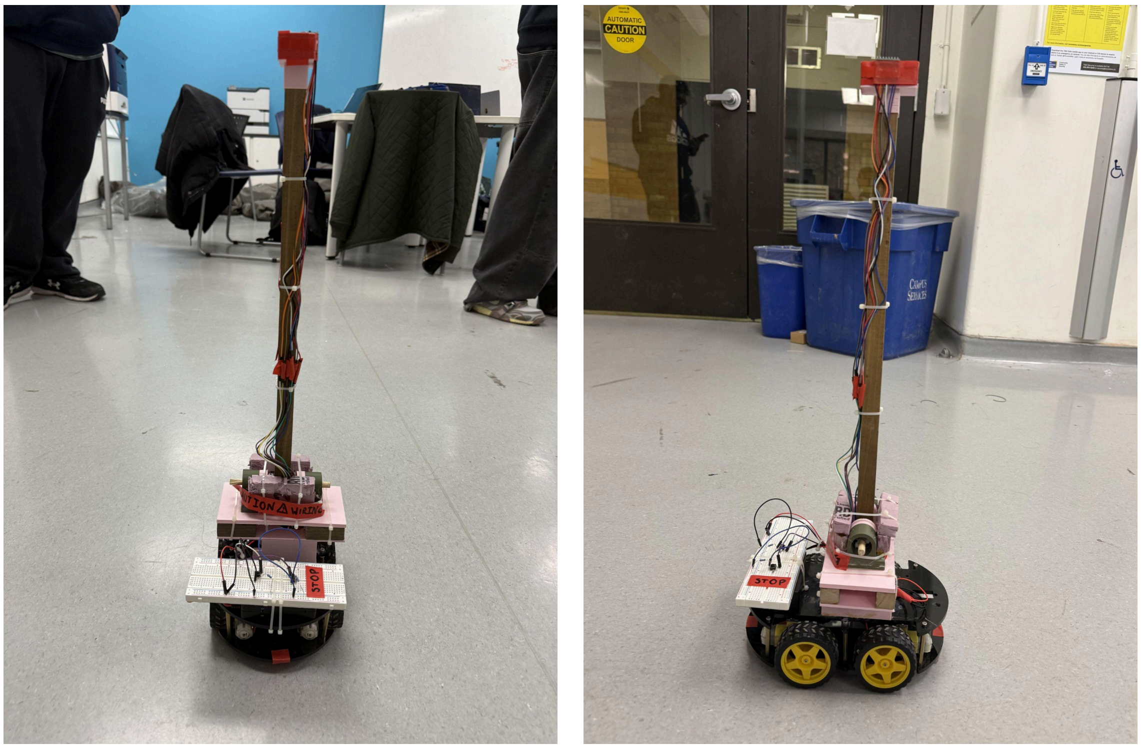


Figure 31- Isometric views of the inverted pendulum system

Electrical Design

The electrical portion of the design includes the various sensors and actuators used to control the system motion and measure data. Three main electrical components were utilized in this design solution: the MPU 6050 module, DC motors, and an ultrasonic sensor. The MPU 6050 module is a sensor that combines an accelerometer and gyroscope which is capable of measuring linear acceleration and angular velocity. This component is mounted to the pendulum to allow for measurements of its tilt angle. The DC motors are the primary component of the disturbance response system. A set of four DC motors with attached wheels are mounted to the cart base and allow it to move. These are used to move the cart in response to changes in pendulum angle in order to stabilize the pendulum upright. Finally, the ultrasonic sensor is a sound based sensor that is used to measure distances. In this design, this sensor is mounted to the side of the cart base and is used to measure cart displacement for use in the pole placement controller. Each of these electrical components are wired to connect with an Arduino Uno microcontroller board and motor driver shield.

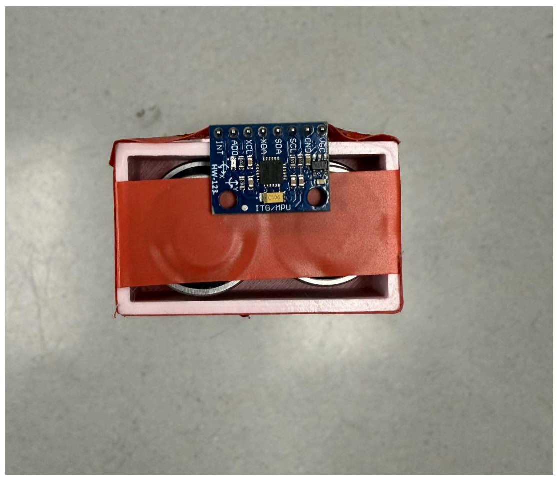


Figure 32 - MPU6050 mounted on top of the Pendulum Arm

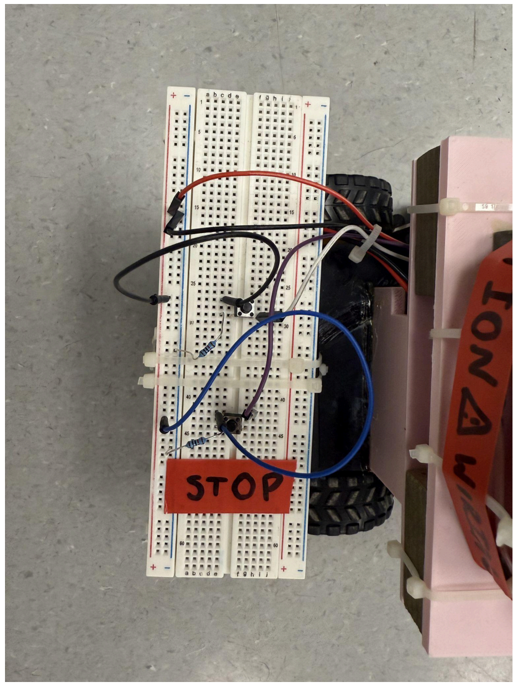


Figure 33 - Wiring of button on to the Breadboard connected to the Microcontroller

Control/Programing

The programming portion of this design includes the Arduino code used to interface with and control the electrical components and sensors of the system. This also includes the implementation of the PID and pole placement controllers. The code is written to take in signals from the two sensors, the MPU 6050 and ultrasonic sensor, and use the received data to perform calculations. The data is then processed by the code using one of the two control algorithms to output control signals to the DC motors.

One of the two controllers that are implemented is PID control. This is a control algorithm that uses the error between a measured and desired value to calculate an appropriate output signal that can be used to reduce the error. The behavior of the controller is based on the proportional, derivative, and integral gains, which are values set by the user. For use in this design, the error is calculated between the desired pendulum angle of 0 degrees and the actual angle of the pivoting pendulum and the output signal is used to set the speed and direction of the DC motors. The gain values were determined using trial and error to obtain optimal results, which resulted in a k_p of 120, k_i of 1.3, and k_d of 4. The PID algorithm was programmed into the arduino code to calculate the PID output based on the pendulum angle, where the output is used to control the speed of the motors.

The second type of implemented controller is pole placement. This is a controller that uses the location of poles to determine a stable configuration for a system. This is done through the derivation of a K matrix which represents the matrix of values that allows the system to have stable poles based on the equation x_dot = Ax + Bu. For the purposes of this design the linear quadratic regulator (LQR) method was used to determine the K matrix using MATLAB. The matrices for A and B were determined based on the equations of motion of the system and the system force variables respectively. This method also uses Q and R matrices, which represent the weights that the state variables and inputs respectively have on the output of the system. Using the K matrix values obtained from MATLAB the equation for the pole placement output was implemented into the arduino code, where it was also used to control the motor speed based on the equation u = -Ku, where u is the output of the controller.

Bill of Materials

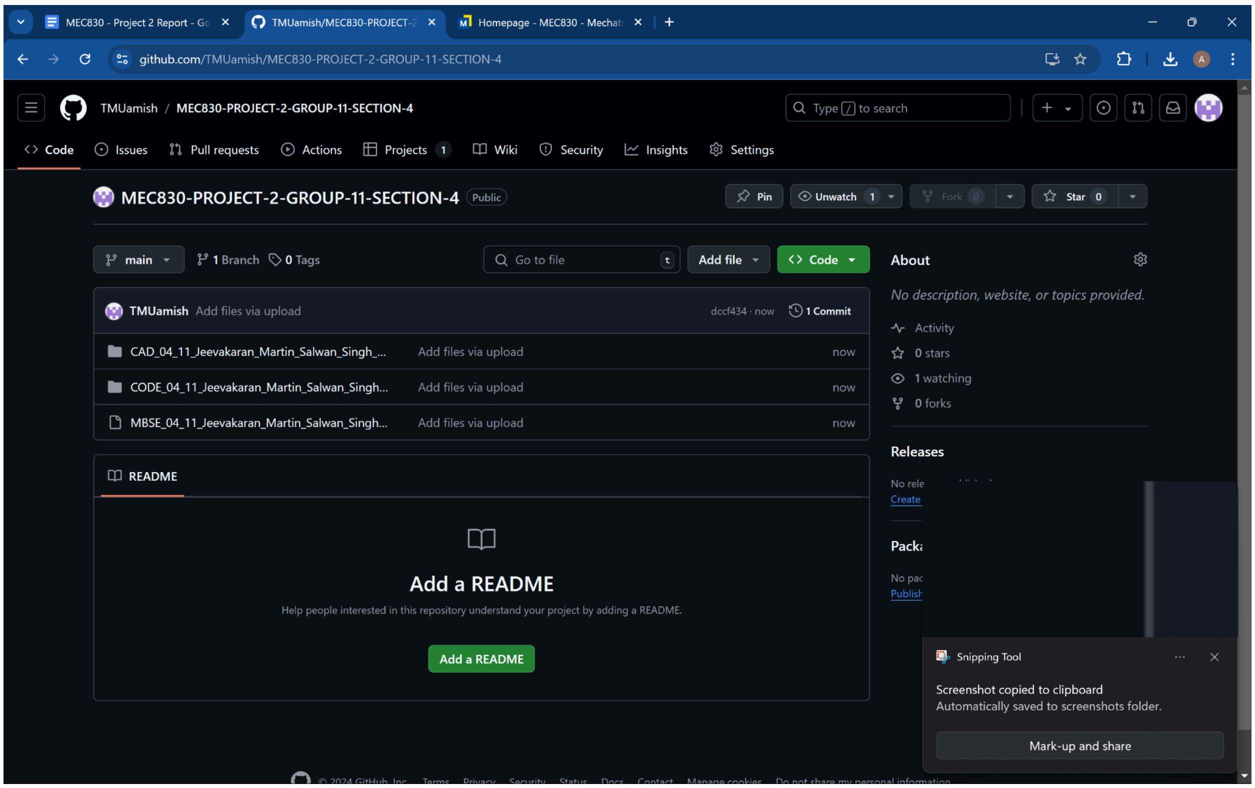
Item	Cost (CAD)	Quantity
1"x1"x48"Wood stick	$1.66	1
Foam block	$10.00	1
Plastic for chassis (2 pack)	$15.00	1
Support metal pole	$0.25	6
T10 screw and bolt (combo)	$0.10	20
3D printing filament (total ft)	$5.00	1
Wheel (2 pack)	$1.75	2
Power supply	$5	1
DC motor	$1.90	4
Tape	$1	1
Zip ties (300 pack)	$12	1
MPU6050 (4-pack)	$13	1
Total Cost	$77.26	

Figure 34 : Table of Bill of Material

Novelty

The design that was chosen was based on the implementation of various design aspects of an inverted pendulum system. The pendulum part of the design followed a standard structure consisting of a long beam with a point mass at its end. The inverted pendulum system also required that the pendulum is capable of free rotation at its base, which was also implemented in the design. The remainder of the design was selected based on research and planning of what components and systems would provide effective means to accomplish the design goals. The design that was chosen utilizes a car as the base to hold the pendulum and the method to which the system responds to disturbance. This type of design allows for flexibility in the use and modification of the system. The use of a wheeled car allows for more freedom of movement making it easier to stabilize the pendulum. This design is also useful as it would be relatively easy to implement it for use in real world situations due to the simplicity of construction and components.

Version Control



GitHub Link: https://github.com/TMUamish/MEC830-PROJECT-2-GROUP-11-SECTION-4.git

Conclusions & Recommendations

This project served as a good introduction to understanding multiple complex engineering concepts. These concepts, such as Failure Modes and Effects Analyses, Design Processes, MBSE Diagrams, Simulation, CAD, Closed-loop feedback controllers, etc., were all related to the inverted pendulum system. The overarching task was to build an inverted pendulum system that would stabilize the pendulum arm using two closed-loop feedback controllers; PID controller, and Pole Placement controller.

The design process to construct an inverted pendulum system was done using a Concurrent engineering approach, where multiple tools for analysis of the design product was done. These tools included the Failure Mode and Effects Analysis, the Design process analysis, the solution formation, the modeling, and implementation to fully construct the entire design product. The implementation of the real-life inverted pendulum was done after thoroughly analyzing the design process iteratively, with the final design only costing $77.26. In terms of the result of the design product, the inverted pendulum system was able to stabilize any disturbance that was applied to it. Both controllers, the PID and the Pole Placement controllers, both were able to stabilize the pendulum arm for approximately 12 seconds after a slight disturbance was applied to it.

One recommendation that can be proposed is that there is more time given for the preparation of the project. Given only three weeks to completely build the entire inverted pendulum was a huge constraint, and the majority of the items took some time to deliver. Therefore, not enough time was devoted to testing the gain values of the PID and Pole Placement controllers. Given more time, we would be able to test and fine tune the parameters so that the inverted pendulum can stabilize itself for a longer amount of time instead of the aforementioned 12 seconds.

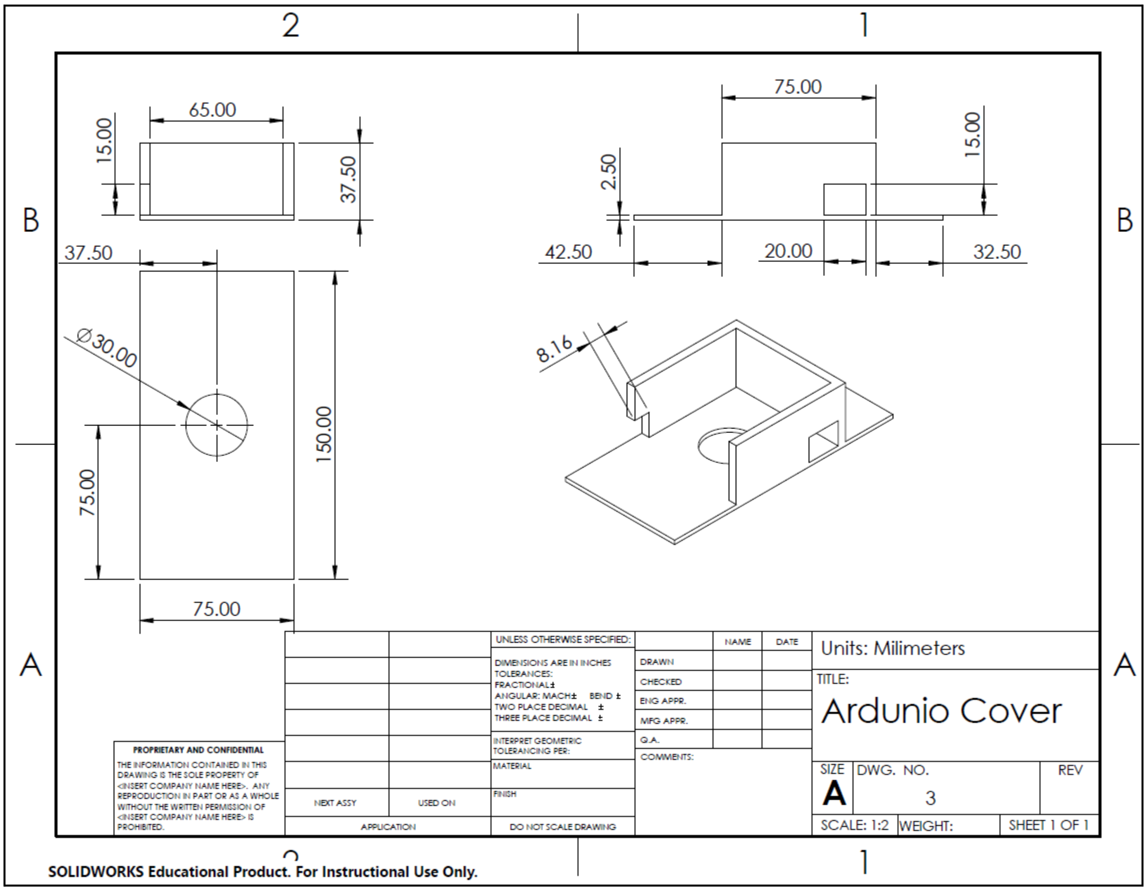
References

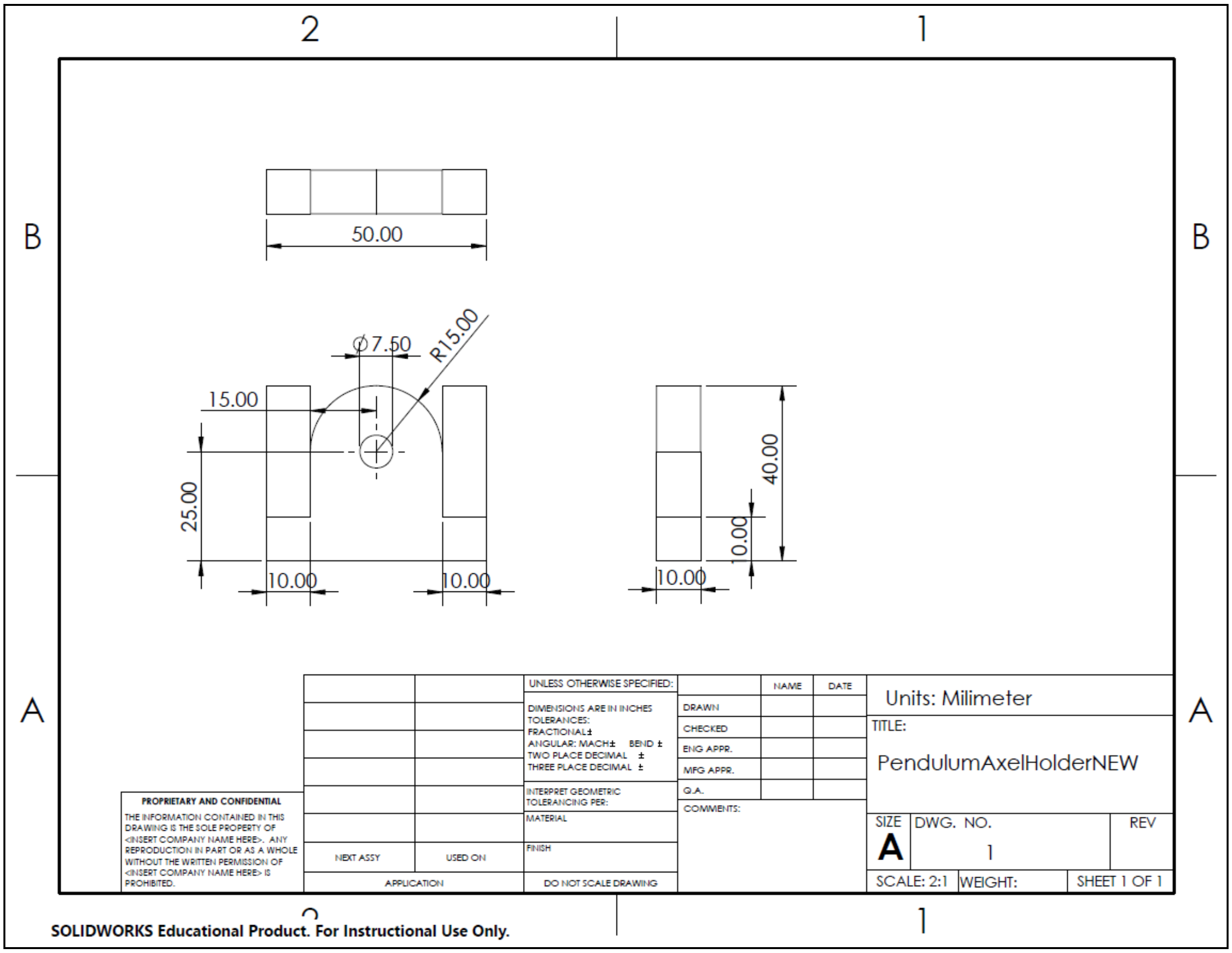
[1] https://wwrichard.net/2013/01/15/what-is-an-inverted-pendulum/

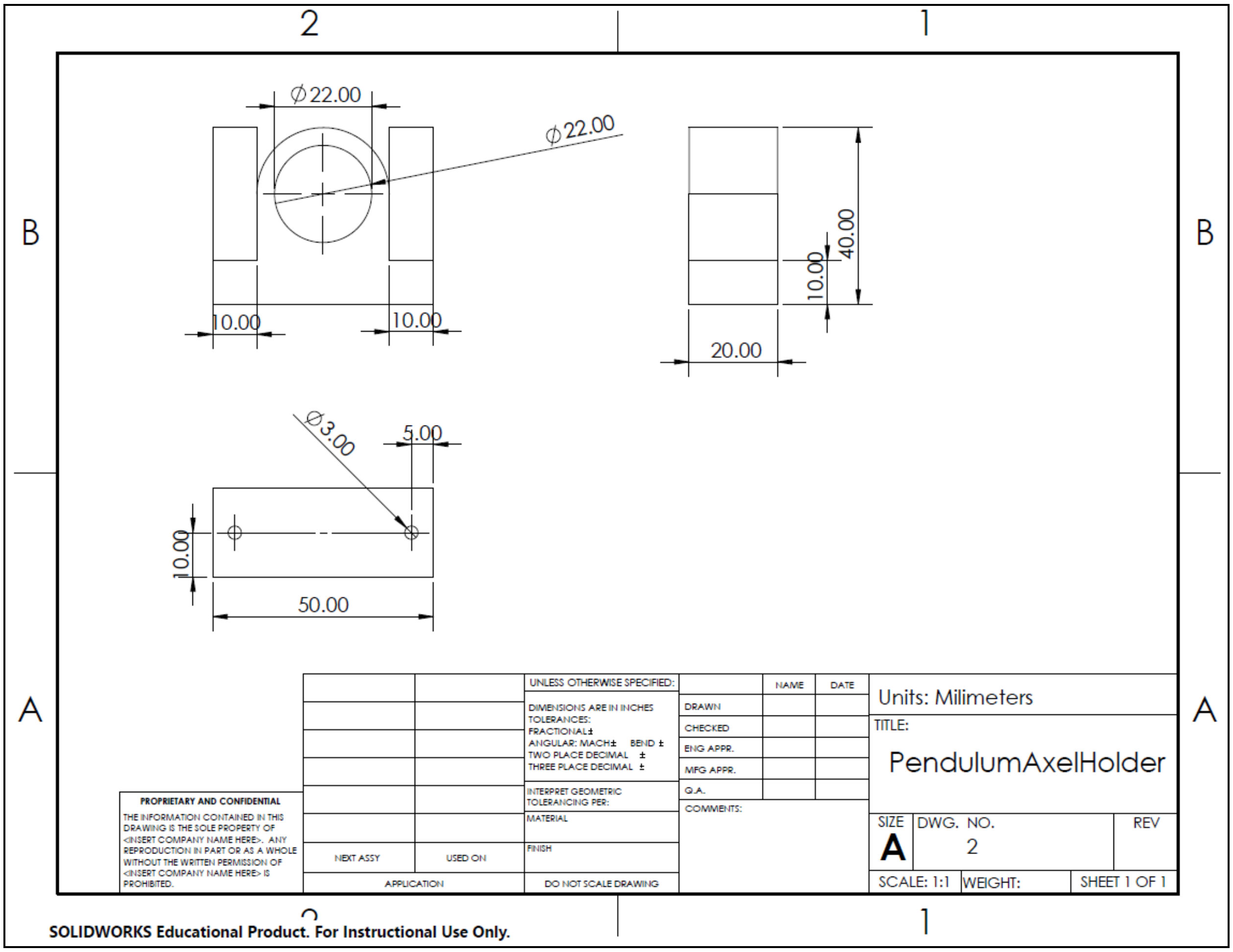
[2]https://www.concurrent-engineering.co.uk/what-is-concurrent-engineering#:~:text=Concurrent%20engineering%2C%20also%20known%20as,improved%20productivity%20and%20reduced%20costs.

Appendix

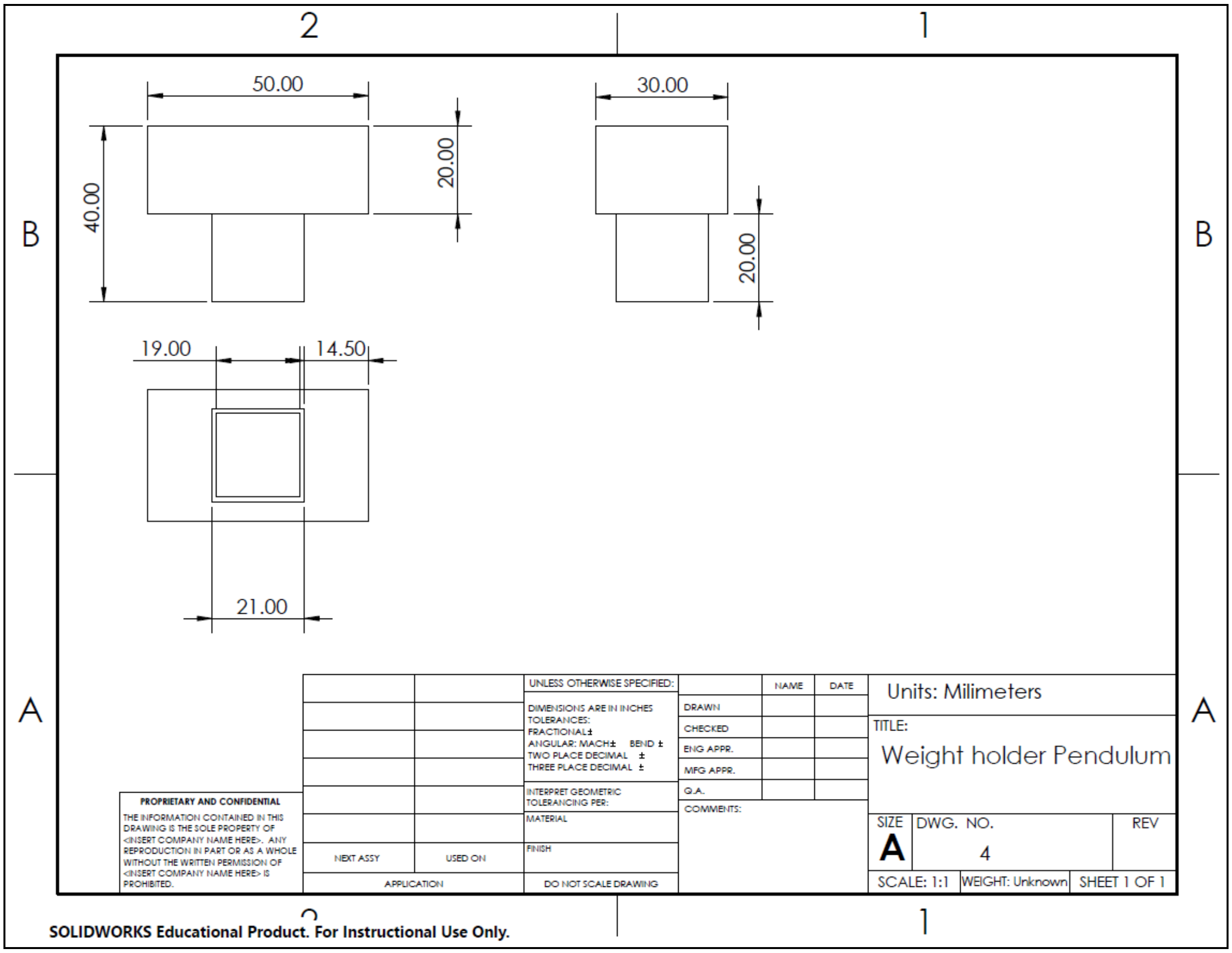
Engineering Drawings




50.00
Ø7.50
R15.00
15.00
25.00
10.00
10.00
40.00
10.00
10.00
Units: Milimeter
TITLE:
PendulumAxelHolderNEW
SIZE A
DWG. NO. 1
REV
SCALE: 2:1
WEIGHT:
SHEET 1 OF 1
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL±
ANGULAR: MACH± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
NEXT ASSY
USED ON
APPLICATION
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
SOLIDWORKS Educational Product. For Instructional Use Only.



Ø22.00
Ø22.00
10.00
10.00
40.00
10.00
20.00
Ø3.00
5.00
10.00
50.00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL±
ANGULAR: MACH± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
Units: Milimeters
TITLE:
PendulumAxelHolder
SIZE
A
DWG. NO.
2
REV
SCALE: 1:1
WEIGHT:
SHEET 1 OF 1
NEXT ASSY
USED ON
APPLICATION
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
SOLIDWORKS Educational Product. For Instructional Use Only.


2
1
50.00
30.00
40.00
20.00
20.00
19.00
14.50
21.00
B
A
UNLESS OTHERWISE SPECIFIED:
NAME
DATE
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL±
ANGULAR: MACH± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
NEXT ASSY
USED ON
APPLICATION
DO NOT SCALE DRAWING
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
Units: Milimeters
TITLE:
Weight holder Pendulum
SIZE
A
DWG. NO.
4
REV
SCALE: 1:1
WEIGHT: Unknown
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.

Arduino Code

Pole Placement

```
#include <Wire.h>
#include <MPU6050_light.h>

MPU6050 mpu(Wire);

float output = 0;

#define PWMA 5    // Controls power to right motor
#define PWMB 6    // Controls power to left motor
#define DIRA 7    // Controls direction of right motor, HIGH = FORWARD, LOW = REVERSE
#define DIRB 8    // Controls direction of left motor, HIGH = FORWARD, LOW = REVERSE
#define STBY 3    // Place H-Bridge in standby if LOW, Run if HIGH

float angleX = 0;        // Angle in the X direction
float angVelX = 0;       // Angular velovity in the X direction
float positionX = 0;     // Position in the X direction
float velocityX = 0;     // Velocity in the X direction

float setpoint = 0;

//Pole Placement Code
//==================================================================
float K[1][4] = {
    {-3.1623, -3.9827, 15.2011, 3.3551}      //Found in MATLAB
};

float state[4] = {0, 0, 0, 0};

float controlInput = 0;

float computeControlInput() {
    float u = 0;
    for (int i = 0; i < 4; i++) {
        u += -K[0][i] * state[i];
    }
    return u;
}
//==================================================================

void setup() {
  Serial.begin(115200);
  Wire.begin();

  // Initialize the MPU6050
  Serial.println("Initializing MPU6050...");
  byte status = mpu.begin();
  if (status != 0) {
    Serial.print("MPU6050 initialization failed with code: ");
    Serial.println(status);
    while (1); // Stop the program if initialization fails
  }

  Serial.println("MPU6050 initialized successfully!");

  // Calibrate the sensor
  Serial.println("Calibrating MPU6050... Keep the device still.");
  delay(1000);
  mpu.calcOffsets(); // Calculate offsets (gyro and accel)
  Serial.println("Calibration complete!");

  pinMode(PWMA, OUTPUT);      //set IO pin mode OUTPUT
  pinMode(PWMB, OUTPUT);
  pinMode(DIRA, OUTPUT);
  pinMode(DIRB, OUTPUT);
  digitalWrite(STBY, HIGH);  //Enable Motors to run
  digitalWrite(PWMA, LOW);  // Fully on
  digitalWrite(PWMA, LOW);  // Fully on

  setpoint = mpu.getAngleX();
}
```

```
void loop() {

  mpu.update(); // Update sensor readings

  // Get the angle in X direction
  angleX = mpu.getAngleX();

  // Get the angular velocity in X direction
  angVelX = mpu.getGyroX();

  //Pole Placement Controller
  //-----------------------------------------------------------------
  state[0] = 0;                 // Cart position
  state[1] = 0;                 // Cart velocity
  state[2] = angleX;                    // Pendulum angle
  state[3] = angVelX;                   // Pendulum angular velocity

  output = computeControlInput()*2;
  //-----------------------------------------------------------------

  // Motor control
  int motorSpeed = constrain(abs(output), 0, 255); // Constrain speed to 0-255 (PWM range)

  if (angleX < setpoint) {
    // Move all motors forward to correct backward tilt
    digitalWrite(DIRA, HIGH);     // Forward direction on Right
    digitalWrite(DIRB, HIGH);     // Forward direction on Left
    analogWrite(PWMA, motorSpeed);    // Power on Right
    analogWrite(PWMB, motorSpeed);    // Power on Left
  } else {
    // Move all motors backward to correct forward tilt
    digitalWrite(DIRA, LOW);     // Reverse direction on Right
    digitalWrite(DIRB, LOW);     // Reverse direction on Left
    analogWrite(PWMA, motorSpeed);    // Power on Right
    analogWrite(PWMB, motorSpeed);    // Power on Left
  }

  // Print debug information
  Serial.print("Angle X: ");
  Serial.print(angleX);
  Serial.print(" | Output: ");
  Serial.println(output);
}
```

PID

```
#include <Wire.h>
#include <MPU6050_light.h>

MPU6050 mpu(Wire);

// PID Parameters
float Kp = 90;     // Proportional gain (adjust for faster response)
float Ki = 6;      // Integral gain (adjust to reduce steady-state error)
float Kd = 4;     // Derivative gain (adjust for smoother damping)

float setpoint = 0.0;       // Desired angle (upright position)
float integral = 0.0;       // Integral term
float previousError = 0.0; // Previous error for derivative calculation
float integralMax = 50;     // Limit for integral windup

// Motor Pins
#define PWMA 5     // Right motor PWM
#define PWMB 6     // Left motor PWM
#define AIN 7      // Right motor direction
#define BIN 8      // Left motor direction
#define STBY 3     // H-Bridge standby

unsigned long previousTime = 0;
unsigned long controlInterval = 5; // Control loop interval in ms (faster sampling)
float angleX = 0;                  // Angle in the X direction
float filteredAngle = 0;           // Filtered angle
float alpha = 0.95;                // Complementary filter weight

void setup() {
  Serial.begin(115200); // Faster debugging
  Wire.begin();

  // Initialize the MPU6050
  Serial.println("Initializing MPU6050...");
  byte status = mpu.begin();
  if (status != 0) {
    Serial.print("MPU6050 initialization failed with code: ");
    Serial.println(status);
    while (1); // Stop the program if initialization fails
  }

  Serial.println("MPU6050 initialized successfully!");

  // Calibrate the sensor
  Serial.println("Calibrating MPU6050... Keep the device still.");
  delay(1000);
  mpu.calcOffsets();
  Serial.println("Calibration complete!");

  // Initialize motor pins
  pinMode(PWMA, OUTPUT);
  pinMode(PWMB, OUTPUT);
  pinMode(AIN, OUTPUT);
  pinMode(BIN, OUTPUT);
  pinMode(STBY, OUTPUT);
  digitalWrite(STBY, HIGH); // Enable motors

  setpoint = mpu.getAngleX();
}
```

```
void loop() {
  unsigned long currentTime = micros(); // Use micros() for higher precision
  if (currentTime - previousTime >= controlInterval * 1000) {
    float dt = (currentTime - previousTime) / 1000000.0; // Time step in seconds
    previousTime = currentTime;

    mpu.update(); // Update sensor readings as quickly as possible

    // Complementary filter for stabilizing angle measurement
    angleX = mpu.getAngleX();
    filteredAngle = alpha * (filteredAngle + mpu.getGyroX() * dt) + (1 - alpha) * angleX;

    // Calculate the error
    float error = setpoint - filteredAngle;

    // Calculate integral term with windup prevention
    integral += error * dt;
    if (integral > integralMax) integral = integralMax;
    else if (integral < -integralMax) integral = -integralMax;

    // Calculate derivative term
    float derivative = (error - previousError) / dt;

    // Calculate PID output
    float output = Kp * error + Ki * integral + Kd * derivative;

    // Update previous error
    previousError = error;

    // Motor control
    int motorSpeed = constrain(abs(output), 0, 255); // Constrain speed to 0-255 (PWM range)

    if (output > 0) {
      // Move forward
      digitalWrite(AIN, HIGH);     // Right motor forward
      digitalWrite(BIN, HIGH);     // Left motor forward
      analogWrite(PWMA, motorSpeed);
      analogWrite(PWMB, motorSpeed);
    } else {
      // Move backward
      digitalWrite(AIN, LOW);     // Right motor backward
      digitalWrite(BIN, LOW);     // Left motor backward
      analogWrite(PWMA, motorSpeed);
      analogWrite(PWMB, motorSpeed);
    }

    // Print debug information
    Serial.print("Angle X: ");
    Serial.print(filteredAngle);
    Serial.print(" | Error: ");
    Serial.print(error);
    Serial.print(" | Output: ");
    Serial.println(output);
  }
}
```