ROBOFEST 4.0

TWO WHEEL SELF BALANCING ROBOT

Submitted By

INDIAN INSTITUTE OF TECHNOLOGY,

BHUBANESWAR

Contents:

1. Description of Robot

A two-wheeled self-balancing robot operates on the principle of an inverted pendulum, using precise control to maintain an upright position. Its center of mass can shift away from the vertical due to uneven weight distribution, making it dynamically stable yet statically unstable. Built with a lightweight frame with Aluminum material, our robot features two NEMA 23 stepper motors that deliver smooth and stable movement. The control system, powered by an Arduino Nano, employs a PID control mechanism to maintain balance.
The robot is engineered to maintain its balance using a combination of elements involving sensors and a control algorithm. The detailed logical steps involved in the working of the robot have been mentioned below:

Initialization and Setup

To initialize the robot, start it from a standstill position, either lying down or standing upright with minimal vibrations. Upon powering up, the onboard MPU 6050 sensor begins calibration, indicated by a blinking LED on the Arduino Nano. If the robot is lying down, gently lift it to a vertical position once the calibration is complete. The robot will then automatically engage its balancing mechanism, using the calibrated sensor data to maintain an upright stance. After initialization, users can place additional loads or weights on top of the robot as needed, and it will adjust to maintain balance accordingly.

Data Acquisition

The microcontroller reads real-time data from the MPU 6050, which includes measurements from its built-in gyroscope and accelerometer. Using the Wire library, the microcontroller communicates with the MPU 6050 over the I2C protocol, ensuring quick data transmission while minimizing the number of required connections. The raw data obtained from the MPU 6050 is processed by the microcontroller, which performs calculations to derive the angular velocity and tilt angle. To achieve a more accurate and stable measurement of the tilt angle, a complementary filter is employed. This filter combines data from both the gyroscope and the accelerometer, leveraging the strengths of each sensor. The accelerometer helps correct long-term drift in the gyroscope data, while 3 the gyroscope smooths out the noisy, short-term variations in the accelerometer readings. Together, they provide a more precise and stable tilt angle estimation, essential for effective balancing.

Error Calculation & Control Algorithm

In the control system, the error is determined by comparing the desired upright position, calibrated as 0 degrees, with the actual tilt angle measured by the sensors. This difference, known as the positional error, indicates how much the robot deviates from the vertical alignment. Additionally, the system calculates the rate of change of this error, which helps predict the robot's future movement trajectory and assess how quickly the tilt is changing.

The PID (Proportional-Integral-Derivative) controller uses this error to dynamically adjust the motor outputs and maintain balance. The Proportional component of the PID responds to the magnitude of the error, applying a corrective force that is directly proportional to the deviation from the upright position. The Integral component addresses accumulated errors over time, correcting small, persistent biases that could cause the robot to drift. More formally, it takes care of the steady state error. Lastly, the Derivative component considers the rate of change of the error, allowing the system to anticipate future deviations and make smoother, more stable adjustments.

Motor Control

The PID controller's output signals are sent to the motor driver, which adjusts the speed and direction of the stepper motors to maintain balance. Pulse Width Modulation (PWM) signals control the effective power supplied to the motors, allowing precise acceleration or deceleration. The system utilizes an Interrupt Service Routine (ISR) that executes every 20 microseconds, generating pulses to control motor speed and direction. This high-frequency, real-time pulse generation ensures smooth and responsive motor control, enabling the robot to quickly react to changes in tilt and maintain an upright position. The ISR handles pulse timing and motor direction switching, ensuring that corrective actions are applied accurately and consistently.

Feedback Loop

Errors are recalculated continuously and the updated values are sent to the control algorithm in real time to keep adjusting the motor movements accordingly. By continuously calculating and feeding the error into the PID controller, the robot can dynamically balance itself, ensuring real-time corrections to maintain its upright stance.

2. Salient Features of Robot

Payload Capacity:

Quite conveniently and without spilling anything, the robot is expected to be able to carry a payload effectively owing to the good design and sturdy construction of the frame. The mechanical structure of the robot weighs 1200gm with a well-balanced design that enables it to carry different types of objects comfortably. Such a feature comes in handy in the delivery and material handling applications where moving about with a load and still managing to keep in an upright position is a requirement.

Line Following with PiCam 2:

PiCam 2 cameras are particularly important in line-following robots, as they provide smoother and more accurate detection of marked paths or lines. Leveraging advanced image processing capabilities, the camera allows the robot to detect lines with greater precision and reliability, even in challenging environments with variable lighting or complex patterns. This improved detection enables the robot to adjust its movement more effectively, resulting in smoother operation. This capability is especially useful in applications where the robot needs to operate in confined areas, such as storage facilities or assembly lines, where precision and adaptability are critical for performing tasks as part of a larger automated process.

Autonomous Self-Balancing:

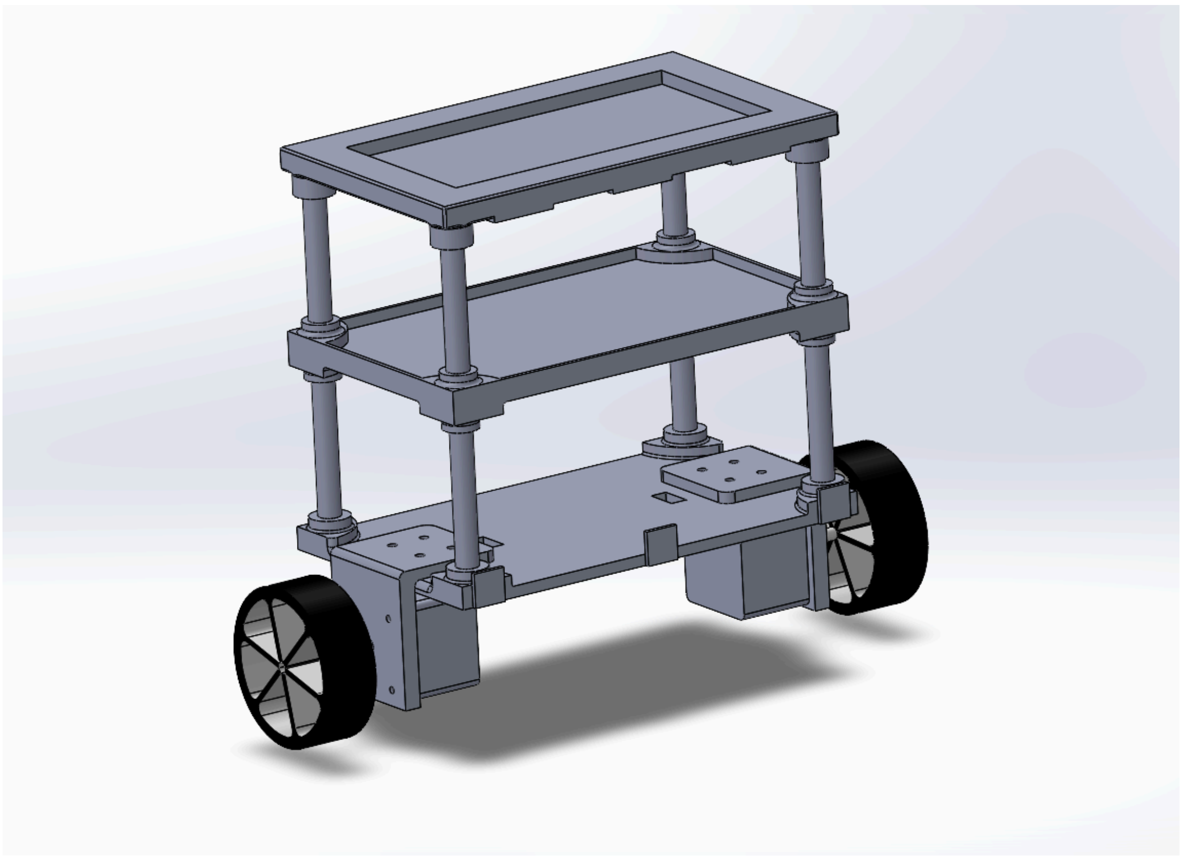
Based on the inverted pendulum structure of the robot, self-balancing can be achieved automatically with the assistance of the user being limited. The MPU 6050, which is one of the sensors installed in the robot system, detects the activities such as angle tilt and the speed at which the tilt changes or angular velocity and feeds those to the PID controller.

Variable Height Mechanism with Threaded Rods

Our robot incorporates threaded rods to enable a variable height mechanism. This design allows the plates of the robot to be positioned at any desired height by securely fastening them with nuts. This feature provides flexibility for adjusting the robot's structure to accommodate different tasks or environments, making it a versatile tool for various applications.

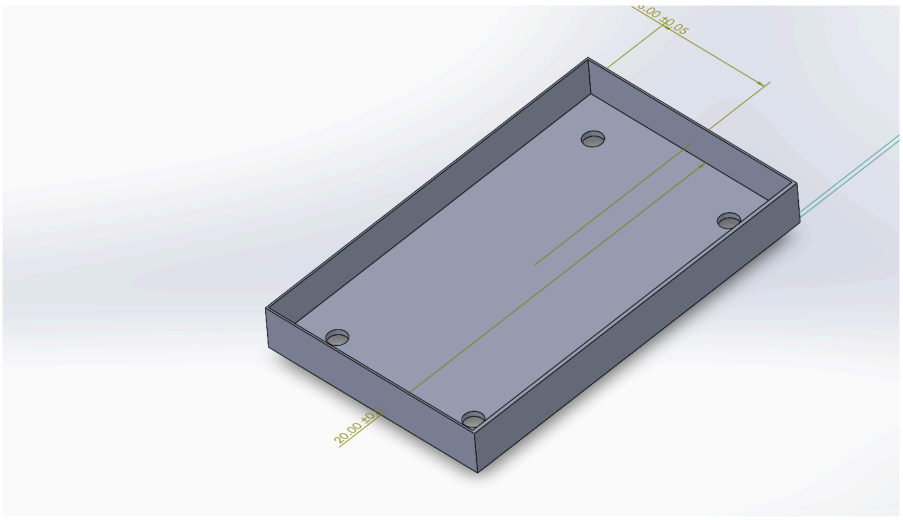
3. Description of Mechanical Design

All our components except the wheels were made from Aluminum Material to strike the balance between the payload carrying capacity and the weight of the robot itself.

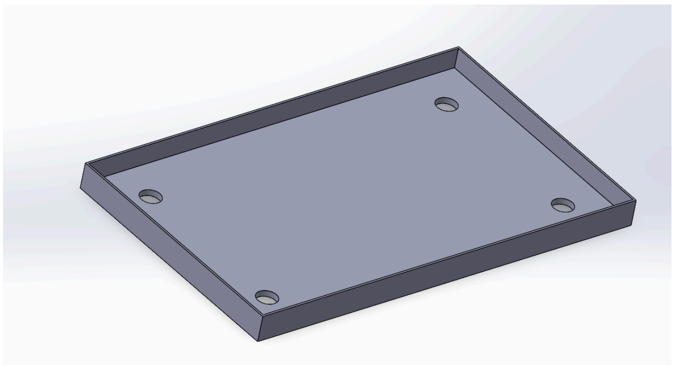


1. Plates:

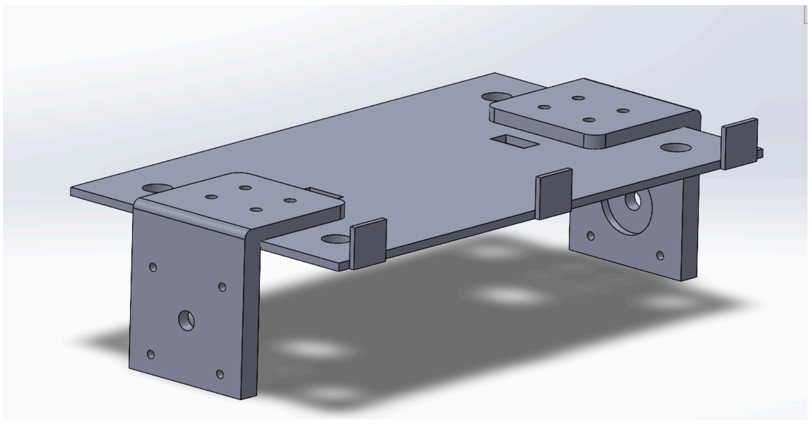
- **Purpose:**
  The plates form the main structural body of the robot, providing mounting surfaces for electronic components, batteries, and other payloads.
- **Details:**
  - **Top Plate:**
    - Thickness: **0.3 cm**
    - Dimension: **20 × 12 cm**
    - Features side borders to prevent materials (like payloads) from falling off.
    - Likely used for mounting lighter components like sensors, controllers, and displays.



  - **Middle Plate:**
    - Thickness: **0.1 cm** (thinner for weight reduction while maintaining structural integrity).
    - Dimension: **20 × 12 cm**
    - Also includes **side borders** for securing components and materials.
    - Main mounting surface for the robot's heavier components, such as the battery and control board.



- **Bottom Plate:**
  - Thickness: **0.3 cm**
  - Dimension: **20 × 12 cm**
  - Acts as the foundation of the robot.
  - Houses the motors, wheels, and threaded rods.



- **Material:**
  All plates are made of **aluminum** to ensure a lightweight design while offering high durability. Aluminum reduces the overall weight of the robot, improving its agility and balance.

2. Hexagonal Bolts:

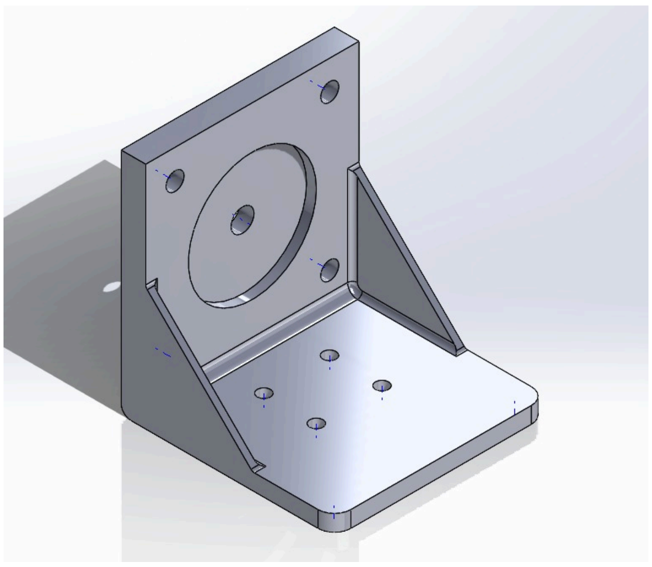
- **Purpose:**
  These bolts are used to securely fasten the plates and other components to ensure structural rigidity and prevent vibration during operation.
- **Details:**

    - Quantity: **24 bolts**
    - These bolts are strategically placed at key points, such as the corners of the plates and where the motors and rods are attached, to distribute loads evenly.

3. Fully Threaded Rods:

- **Purpose:**
  The threaded rods hold the three plates together in a **symmetric arrangement**, ensuring stability and alignment of the robot's structure.
- **Details:**
    - Length: **22 cm**
    - Diameter: **1 cm**
    - The rods pass through all three plates, with hexagonal nuts securing them at both ends.
    - The design ensures that the plates remain equidistant and aligned vertically.
    - These rods also contribute to the load-bearing capacity of the structure, allowing it to hold a significant weight (e.g., **1 liter of water**).

4. Side Flanges:



- **Purpose:**
  The side flanges are designed to **hold the motors securely** in place, preventing unwanted movement or misalignment.

- **Details:**
    - The flanges are mounted on the bottom plate, where they align with the motors' mounting points.
    - These components ensure that the torque generated by the motors is effectively transmitted to the wheels without compromising stability.

5. Aluminum Material:

- **Purpose:**
  Aluminum is used for all mechanical components to achieve a balance between lightweight construction and strength.
- **Benefits:**
    - Lightweight: Reduces the weight the motors must balance, improving response time and stability.
    - High Durability: Ensures the structure can withstand mechanical stress and impacts during operation.
    - Corrosion Resistance: Prolongs the lifespan of the robot, even in humid conditions.

6. Symmetry in Design:

- **Purpose:**
  A symmetric design is critical for a two-wheeled self-balancing robot to ensure even weight distribution. This simplifies the control logic for balancing and reduces the chances of tipping over.

Load-Bearing Capacity:

- The robot is designed to **hold a payload of 1 liter of water (~1 kg)**.
- This is achieved through:
    - The structural integrity of the **threaded rods and plates**, which distribute the weight evenly.

- The rigidity provided by **24 hexagonal bolts**, ensuring no component shifts during operation.

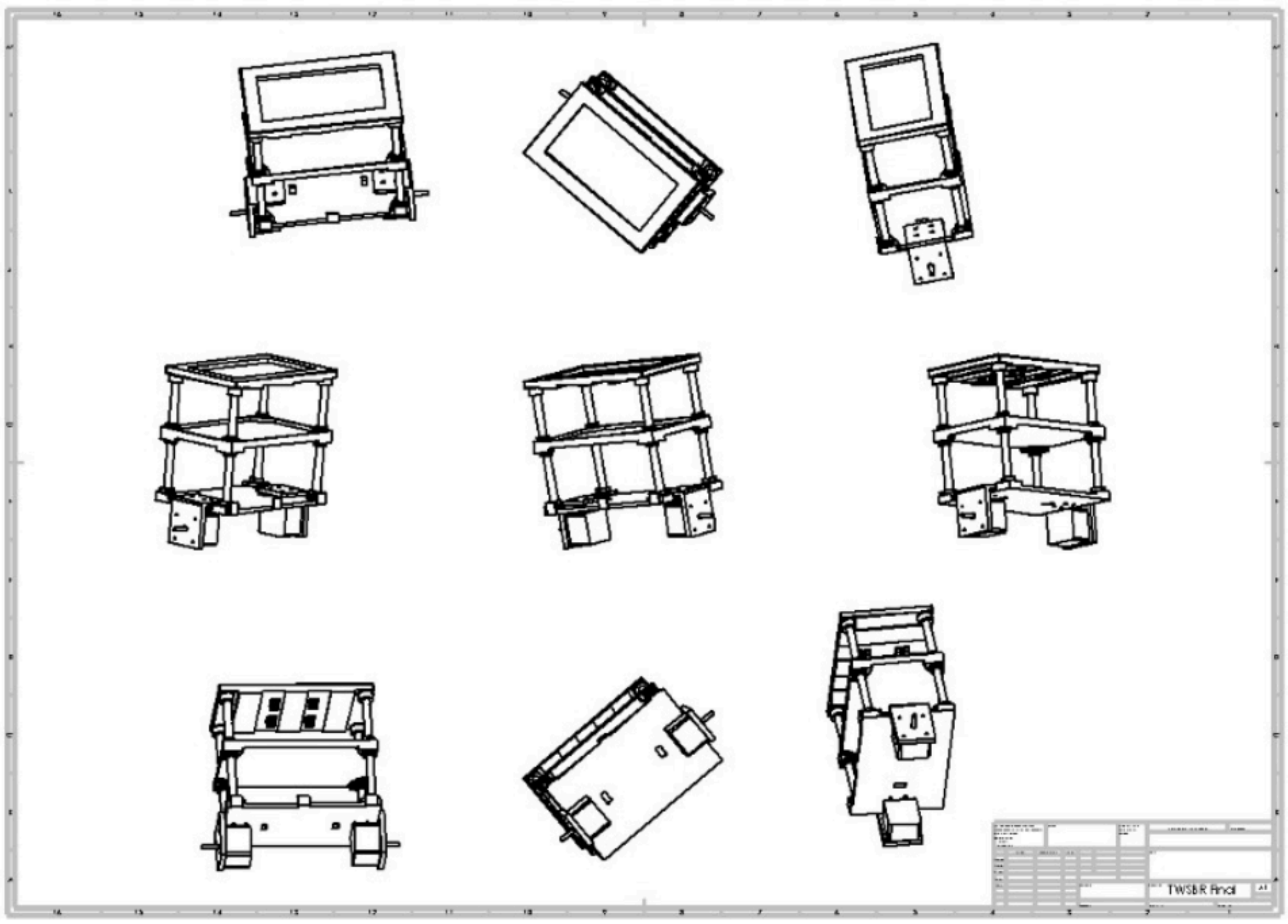
Mechanical Design in the Context of Two-Wheeled Balance:

1. **Center of Gravity (CoG):**
    - The **middle plate** likely holds the heavier components (e.g., batteries) to keep the CoG low, improving balance.
    - Symmetry ensures that the CoG remains centered between the wheels.
2. **Motor and Wheel Mounting:**
    - The **side flanges** ensure that motors are mounted in line with the robot's axis, minimizing tilt or misalignment issues that could destabilize the robot.
3. **Durability and Stability:**
    - The use of aluminum and the rigid connections between components ensure that vibrations or external forces (e.g., bumps) do not compromise the robot's balance.

4. Description of all the logical steps in the working of mechanical design

The mechanical design of the two-wheeled self-balancing robot was developed with innovation and adaptability in mind. The use of fully threaded rods provides the flexibility to adjust the length of the structure, allowing the robot to accommodate materials of varying sizes beyond the initial setup. This adaptability ensures that the design can be easily modified for diverse applications without compromising structural integrity. During development, multiple configurations were tested to determine the structure that provided the best balance and response. The chosen design, featuring three lightweight aluminum plates and symmetric alignment, proved to be optimal for stability and weight distribution. The middle plate, being thinner, keeps the center of gravity low, while the top and bottom plates ensure robustness.

The robot’s lightweight structure, combined with high durability, minimizes the load on the motors, enabling efficient performance. The symmetric design simplifies control by evenly distributing weight across the two wheels, crucial for balancing. Additionally, the structure was tailored to handle a range of PID control values, ensuring stability across various tuning parameters. The side borders and flanges enhance functionality by securely holding materials and motors. This thoughtful, adaptable, and efficient design not only balances the robot effectively but also provides versatility for future modifications.



5. Software used for Robot-Making

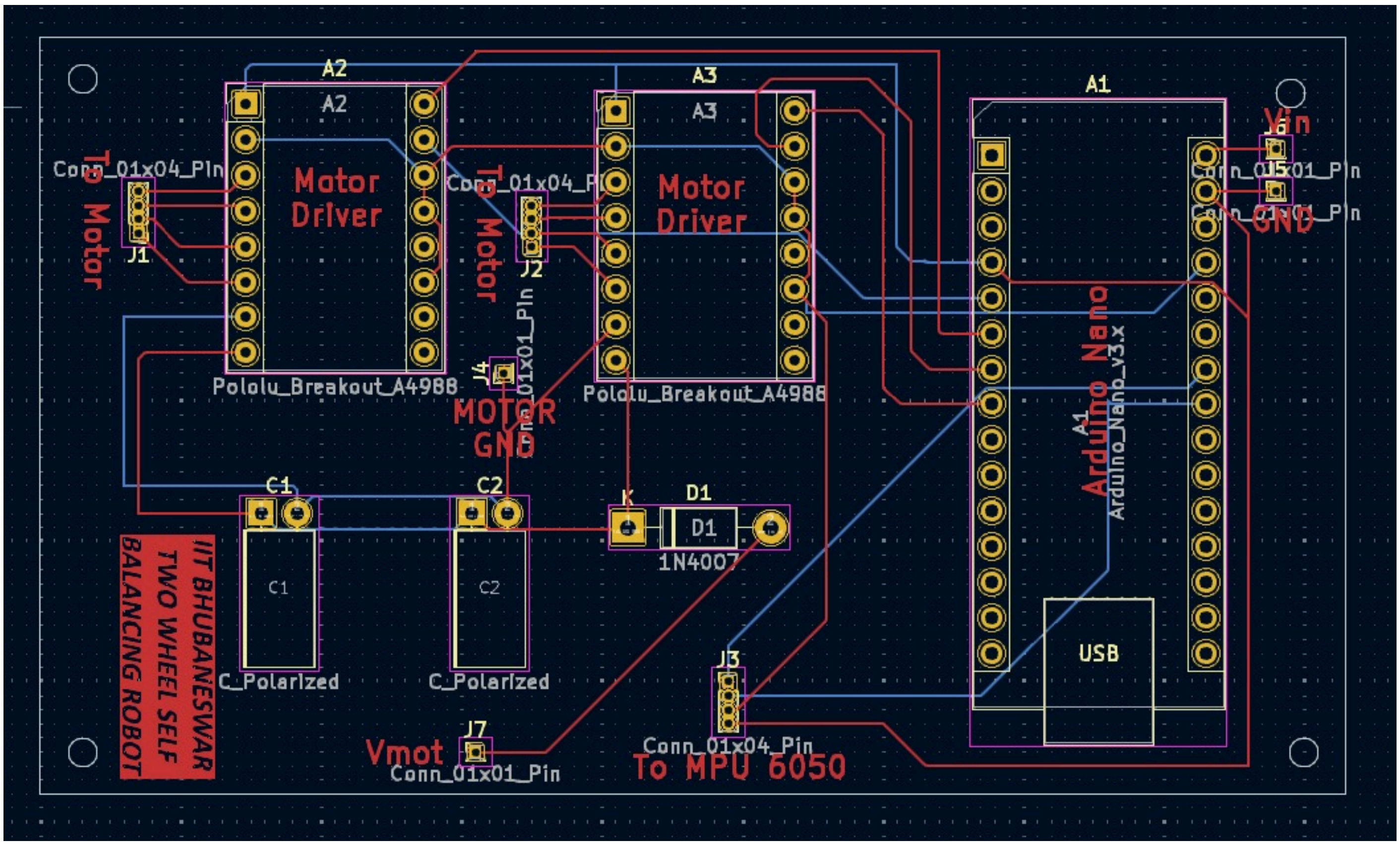
The development of our two-wheeled self-balancing robot involved leveraging a wide range of software tools to streamline the design, coding, testing, and analysis processes. Below is an overview of the software utilized and their specific roles:

1. Arduino IDE
The Arduino IDE served as the primary platform for programming the microcontroller. It was used to implement the robot's control logic, including the PID control mechanism, motor control, and real-time processing of sensor data from the MPU-6050 accelerometer and gyroscope. The straightforward and reliable environment provided by Arduino IDE made coding and debugging efficient.

2. SolidWorks
The SolidWorks software was employed for creating the CAD model of the robot's mechanical structure. It helped in visualizing the robot's design, testing the stability of the frame, and ensuring that the placement of components like motors, batteries, and the control board was optimal for weight distribution. The accurate modeling facilitated seamless fabrication of the mechanical parts.

3. MATLAB
MATLAB was used for simulating and testing various aspects of the system before implementation. This included analyzing the dynamics of the self-balancing mechanism, tuning the PID controller parameters, and visualizing sensor data to ensure the accuracy and responsiveness of the control system.

4. KiCad
For the design of the Printed Circuit Board (PCB), KiCad was used. It allowed us to create schematics, design the layout of the control circuitry, and produce the files needed for PCB manufacturing. This ensured precise and reliable connections between electronic components.



5. Fritzing

Fritzing was utilized for creating circuit diagrams to document the robot's electronic connections. This software simplified the process of visualizing and sharing the wiring between components like the MPU-6050, motor driver, and the microcontroller.

6. Visual Studio Code (VS Code)

VS Code was used for Python programming, particularly in the initial stages of testing sensor data and creating auxiliary scripts for system analysis. Its extensive library support and debugging tools helped in rapid prototyping and evaluation of sensor fusion algorithms.

7. Ansys

The Ansys software was employed for mechanical analysis. Using Ansys, we conducted stress analysis on the robot's frame and evaluated the durability of critical components under dynamic loads. This ensured the design's robustness, especially when subjected to vibrations and impacts during operation.

Each of these software tools played a crucial role in ensuring that the robot's design, fabrication, and programming were executed with precision, resulting in a well-functioning prototype.

6. Details of the Manufacturing / Fabrication of the Mechanical Assembly

Step 1: Preparing the Threaded Rods

- **Cut to Length**: Using a hacksaw or angle grinder, cut the threaded rods to equal lengths of 22 cm.
- **Deburr Edges**: Smoothen the sharp edges of the rods using a file or grinder to prevent injuries during assembly and ensure easy threading.

Step 2: Preparing the Plates

- **Cut to Dimensions**: Take high-quality aluminum sheets and cut them to the desired dimensions of 20 × 12 cm. Ensure precise cutting to achieve uniformity. The top and bottom plates are 0.3 cm thick for added strength, while the middle plate is 0.1 cm thick to reduce weight.
- **Welding the Borders**: Use Tungsten Inert Gas (TIG) welding to attach aluminum side borders to the middle and top plates. These borders prevent materials from falling off and add structural integrity.
- **Finish Edges**: Smooth the edges of the plates using sandpaper or a grinder for safety and a clean appearance.

Step 3: Drilling the Plates

- **Mark Holes**: Carefully measure and mark the positions for the four rods and other mounting components on the plates using a ruler and marker. Ensure uniformity for proper alignment.
- **Drill Holes**: Use a drill press for precision to create holes matching the rod diameter (1 cm). Drill additional holes as needed for motor mounts, flanges, and electronic components.

Step 4: Assembly of the Frame

- **Thread Nuts**: Attach nuts and washers to the threaded rods and secure them tightly to each plate. This ensures that the plates are rigidly held in position.
- **Arrange the Plates**:
    - **Bottom Plate**: Acts as the foundation, holding the motors, wheels, and other base components.
    - **Middle Plate**: Designed to support the electronics (e.g., control boards, sensors, and batteries) while maintaining a low center of gravity for stability.
    - **Top Plate**: Used for sensors and an aesthetic cover, providing a clean and organized finish.
- **Tighten Assembly**: Using two wrenches, firmly tighten the nuts and washers on either side of each plate, ensuring the frame is sturdy and well-aligned.

Step 5: Wheel and Motor Mounting

- **Attach Motors**: Mount the motors to the bottom plate using custom aluminum flanges or brackets. Ensure the motors are aligned and secure to prevent vibrations or misalignment during operation.
- **Connect Wheels**: Fix the wheels to the motor shafts tightly using screws or clips. Verify that the wheels rotate smoothly and are balanced.
- **Align Assembly**: Double-check that the wheels are perfectly aligned with the robot’s center to ensure stable and straight motion during operation.

Step 6: Integration of Components

- **Mount Electronics**: Securely attach the control board, battery, and sensors to the middle plate using screws and brackets. Arrange the wiring neatly to avoid tangling.
- **Payload Adaptability**: Utilize the adjustable threaded rods to modify the distance between plates, allowing the robot to accommodate payloads of different sizes.

Step 7: Final Adjustments

- **Symmetry Check**: Ensure the entire structure is symmetric, as an uneven frame could affect balance and stability.

- **Weight Distribution**: Verify that the weight is distributed evenly across the two wheels to optimize the robot's balancing capabilities.

This innovative mechanical design leverages adjustable threaded rods for flexibility, enabling the robot to adapt to different payloads and configurations. By using aluminum for lightweight durability, welding for precise side borders, and a symmetric structure, the design ensures balance and stability while supporting a wide range of PID control settings. These steps collectively create a reliable and adaptable two-wheeled self-balancing robot.

7. Electronic Components and Their Working

Arduino Nano (Microcontroller)

The microcontroller used in our robot is the **Arduino Nano**. It features:

- **22 digital I/O pins** and **8 analog inputs**, making it highly versatile for managing sensors and motor drivers.
- Operates at **5V** with a clock speed of **16MHz**, providing sufficient processing power for real-time control.
- Programmed via the **Arduino IDE** using the USB interface, which simplifies code development and deployment.
  The Arduino Nano is responsible for executing the **PID control algorithm** to maintain balance, as well as interfacing with the MPU-6050 and motor drivers. Its small size makes it ideal for space-constrained applications like our two-wheeled robot.

Raspberry Pi 4 (Single-Board Computer)

The **Raspberry Pi 4** is integrated into the robot to enable path-following capabilities using image processing techniques. Its role includes:

- Capturing images from the ground using a connected **Raspberry Pi Camera Module** to detect the path or tape for line-following.

- Processing the captured images with Python libraries like **OpenCV** for path detection and generating appropriate motor control signals.
- Communicating with the Arduino Nano to adjust the robot's movement and direction based on the detected path.
  The Raspberry Pi 4’s powerful quad-core processor and ample memory enable efficient handling of image processing tasks in real time, making it suitable for advanced navigation tasks.

Motors (Stepper Motors)

The robot uses **stepper motors** for driving the wheels. These motors are chosen for their:

- High precision in controlling the movement, essential for maintaining balance.
- Ability to provide consistent torque at low speeds, ensuring smooth operation during line-following or self-balancing tasks.

Resistors and Capacitors

- **Resistors**:
  - Limit the current flow into sensitive components, protecting them from damage due to excessive current.
- **Capacitors**:
  - Smooth out voltage fluctuations in the power supply, especially near the stepper motor drivers.
  - Filter noise and suppress electromagnetic interference (EMI) caused by switching currents.
  - Ensure stable operation of the motors and improve their accuracy, which is critical for balancing and path-following.

Diodes

Diodes are used to protect the electronic circuit by:

- Blocking reverse current flow that could potentially damage sensitive components like the Arduino Nano or motor drivers.

- Providing unidirectional current flow, which is essential for the safe operation of the motors and power supply components.

Integration of Components

- The Arduino Nano handles the self-balancing aspect, processing data from the MPU-6050 and controlling the motors via the A4988 drivers.
- The Raspberry Pi 4 enables advanced navigation by performing image processing for line-following.
- Resistors, capacitors, and diodes ensure the stability and protection of the overall electronic circuit, improving reliability and performance.

Together, these components form a robust and efficient system capable of both self-balancing and autonomous path-following.

8. Description of all the logical steps in the working of electronic components

MPU6050 and Arduino Nano: Balancing the Robot

The **MPU6050**, an integrated accelerometer and gyroscope module, plays a pivotal role in the self-balancing mechanism of the robot. It continuously monitors the tilt angle and angular velocity of the robot by detecting changes in its orientation. This real-time sensor data is sent to the **Arduino Nano**, the central microcontroller, via an I2C interface. The Arduino Nano processes this data using a **PID control algorithm**, which calculates the necessary adjustments required to maintain the robot's balance. The algorithm uses proportional, integral, and derivative terms to compute precise corrective actions based on the tilt angle and the rate of change in tilt.

Once the corrective action is determined, the Arduino Nano generates **direction** and **step signals** that are sent to the **A4988 motor drivers**. These signals dictate the rotational direction and precise steps the motors must take to stabilize the robot. The **A4988 drivers** translate these signals into precise current flows to the stepper motors, ensuring accurate and smooth motor operation. The stepper motors, in turn, adjust their position and rotation to compensate for the imbalance detected by the MPU6050. This iterative process, occurring within milliseconds, allows the robot to maintain its balance even when subjected to external disturbances or uneven surfaces.

IR Sensors: Line Following Capability

The robot is equipped with **infrared (IR) sensors** that add an additional layer of functionality by enabling it to follow a predefined path or line. These sensors detect the contrast between the tape (line) on the ground and the surrounding surface. The IR sensors continuously send this data to the Arduino Nano, which interprets it to determine the robot's current position relative to the line.

Using this information, the Arduino Nano adjusts the motors' movements to align the robot back onto the path. This line-following capability is integrated seamlessly with the self-balancing mechanism, allowing the robot to navigate while maintaining its upright position.

Power Supply and Stability

The robot’s electronic components are powered by a **DC-DC converter**, which steps down the input voltage to a regulated 5V output. This ensures that all components, including the Arduino Nano, MPU6050, IR sensors, and motor drivers, receive stable and appropriate power.

To further enhance power stability, **capacitors** are strategically placed in the circuit near the motor drivers. These capacitors smooth voltage fluctuations and suppress noise generated by the motors during operation. By reducing electromagnetic interference (EMI), the capacitors also improve motor performance and ensure precise adjustments, which are critical for both balancing and navigation.

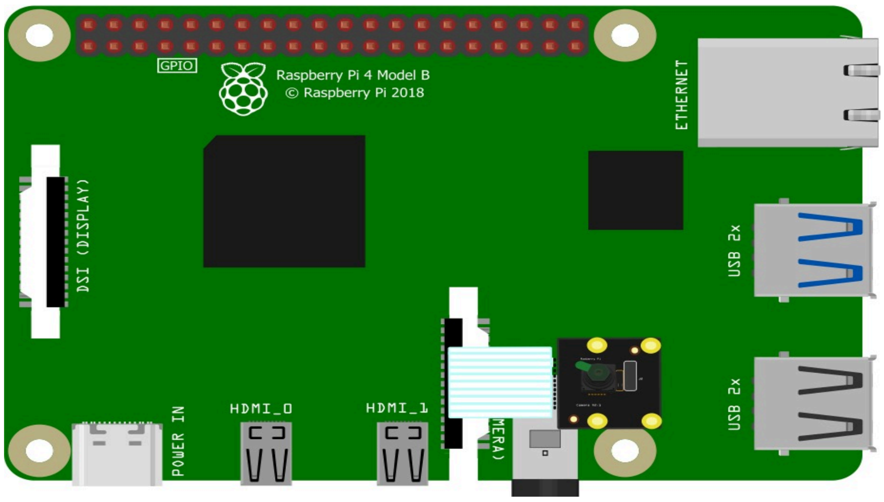
System Integration and Efficiency

Together, the MPU6050, Arduino Nano, A4988 drivers, stepper motors, and IR sensors create a cohesive system. The MPU6050 detects imbalances, while the Arduino Nano calculates and sends corrective signals to the motors, ensuring stable operation. Meanwhile, the IR sensors enable the robot to navigate predefined paths effectively.

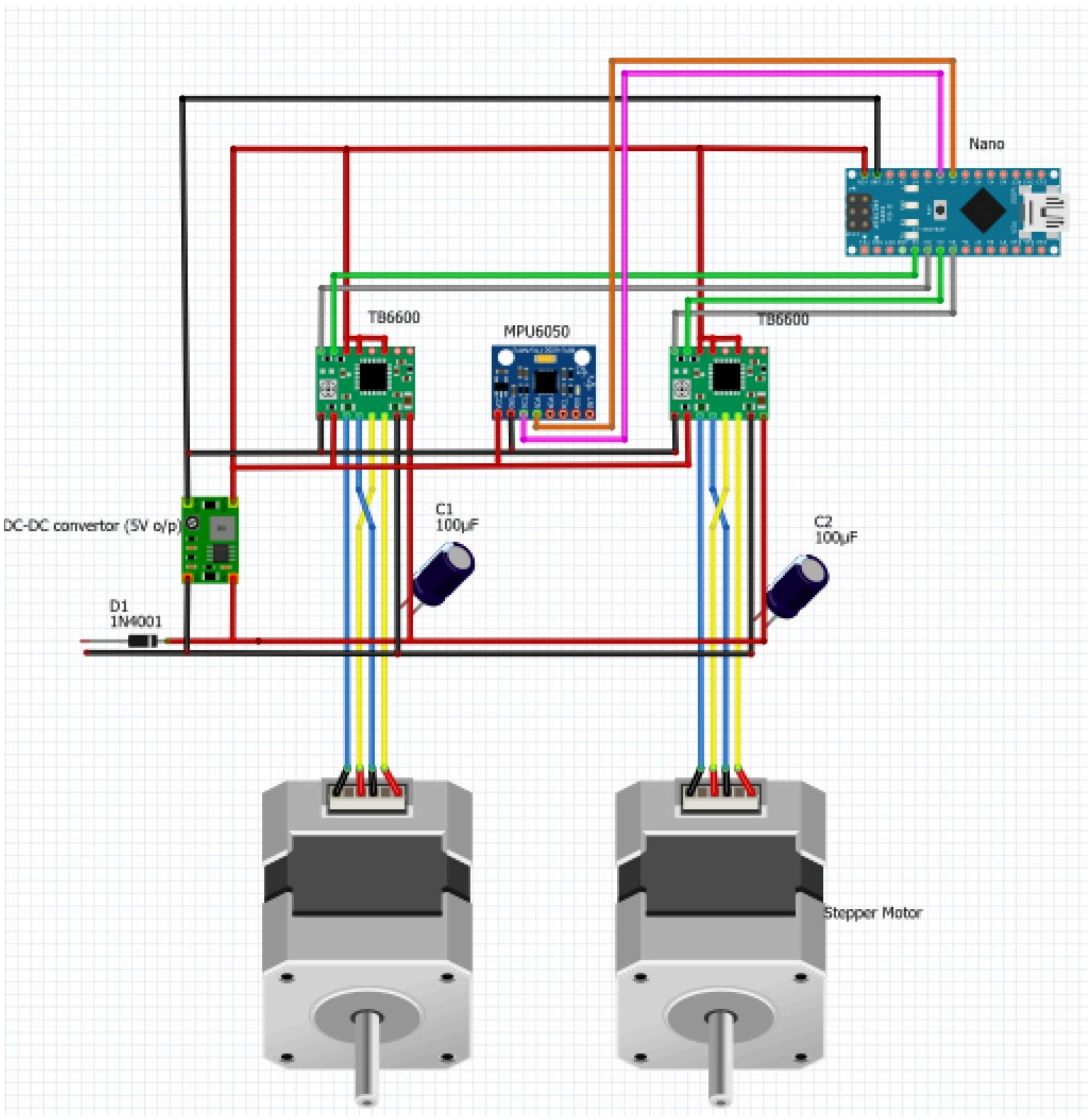
The integration of a robust power supply, capacitors, and motor drivers ensures the reliability of the system even under varying conditions. This synergy between components makes the robot a versatile and efficient platform capable of self-balancing and autonomous navigation tasks.

9. Electronic connection block diagram

Raspberry Pi and Camera:



Microcontroller Circuit:



10. Futuristic application of Robot as well as in commercialization

Two-wheeled self-balancing robots hold significant potential for futuristic applications and commercialization. Their unique design, combining stability, maneuverability, and compactness, makes them ideal for a variety of use cases. Here are some visionary applications and commercial opportunities:

Futuristic Applications

1. Personal Mobility Devices

- **Advanced Wheelchairs**: Self-balancing technology can create sleek, highly maneuverable wheelchairs for users with mobility challenges, enhancing accessibility in tight spaces.
- **Urban Commuters**: Compact, two-wheeled personal transporters for urban commuting could replace scooters or bicycles, offering a stable ride on uneven terrain.

2. Service Robots

- **Retail and Hospitality**: Robots equipped with shelves or trays can deliver items in stores, restaurants, or hotels, maintaining stability while moving between customers or rooms.
- **Healthcare Assistance**: Robots could carry medical supplies, medications, or even monitor patients autonomously in hospitals or care homes.

3. Last-Mile Delivery

- **Parcel and Food Delivery**: Compact robots can navigate urban environments efficiently, delivering parcels or food to customers' doorsteps, reducing human involvement and delivery times.

4. Surveillance and Security

- **Patrol Robots**: Equipped with cameras and sensors, these robots can patrol premises, monitor activity, and even respond to potential security breaches in warehouses, factories, or campuses.
- **Disaster Response**: Robots with self-balancing capabilities can traverse challenging terrains, providing real-time data or delivering critical supplies during emergencies.

5. Education and STEM Learning

- Self-balancing robots are excellent platforms for teaching robotics, control systems, and programming to students, making them a staple in educational institutions worldwide.

6. Entertainment and Interaction

- **Social Companions**: Robots designed as companions for children, the elderly, or even as interactive toys.
- **Events and Promotions**: Futuristic robots can act as attention-grabbing entities for brand promotions, offering an engaging experience.

Commercialization Potential

1. Consumer Electronics

- **Mass-Produced Mobility Devices**: Companies can create sleek, affordable self-balancing scooters or personal transporters for urban use.
- **Smart Home Integration**: Robots for domestic use, integrated with home automation systems for tasks like fetching items, cleaning, or monitoring security.

2. Logistics and Warehousing

- **Automated Trolleys**: Robots can transport goods between shelves or assist workers in warehouses.
- **Inventory Management**: Equipped with cameras and scanners, they can update inventory in real-time.

3. Healthcare Market

- **Patient Monitoring**: Robots designed to autonomously monitor patients' health metrics or remind them to take medications.
- **Assistive Devices**: Commercially available self-balancing wheelchairs or crutches.

4. Retail and E-Commerce

- **Smart Shopping Carts**: Robots that follow customers, carrying their purchases and scanning items for seamless checkout.
- **Delivery Bots**: Integration into logistics chains for rapid and cost-effective last-mile deliveries.

5. Agriculture

- **Precision Farming**: Self-balancing robots equipped with sensors and cameras can monitor crops, apply fertilizers, or detect pests with high precision.

6. Defense and Law Enforcement

- **Reconnaissance Bots**: Compact, stealthy robots capable of navigating challenging environments for surveillance or bomb disposal.
- **Crowd Control**: Deployment in public spaces to manage crowds or deliver announcements.

Challenges and Considerations for Commercialization

- **Cost**: Reducing production costs for affordability while maintaining high-quality performance.
- **Safety**: Ensuring stability and fail-safe mechanisms in diverse environments.
- **Battery Life**: Enhancing battery technology for extended operation.
- **Regulation**: Compliance with local laws for autonomous systems and mobility devices.
- **Public Acceptance**: Ensuring users trust and adopt the technology.

Conclusion

The self-balancing two-wheeled robot can revolutionize industries by combining intelligent automation and mobility. With advancements in AI, sensors, and battery technologies, these robots can evolve into indispensable tools in transportation, logistics, healthcare, and beyond, making them a valuable investment for companies targeting innovative markets.

Signed By:

Team Members:

1. Ayush Gupta
2. Divya Kumar
3. Jiya Chakraborty
4. Rachit Jain
5. Vivek Singh

Mentor:

Dr. Satyanarayan Panigrahi
Associate Professor
School of Mechanical Sciences

Head of Institute:

Prof. Shreepad Karmalkar
Director, IIT Bhubaneswar