ROBOFEST 4.0

TWO WHEEL SELF BALANCING ROBOT

Submitted By

Indian Institute of Technology Bhubaneswar

2024

CONTENTS

<u>OVERVIEW</u>

A Two-Wheel Auto-Balancing Robot (more commonly known as **TWABR**) employs a pair of vertically aligned wheels to achieve its primary objective of preserving equilibrium and avoiding tipping over. The wheels are driven by two motors coupled to each of them. The motors can be of a DC nature and are controlled by electrical signals using a control system based on the inclination reading and the velocity of their gravity centre. Its operation is similar to the classic inverted pendulum system. The objective of this project is to create and deploy a self-stabilising robot utilising an ESP 32 microcontroller, an MPU6050 accelerometer, a gyroscope sensor, and an L298N motor driver. The report provides a detailed description of the components that are required and how they are used. The motivation for working with the TWSBR came from the fact that it’s an interesting robot to control, not only because of its unstable and non-linear nature but also because of its application outside the academic environment.

AIM

To make a two-wheeled auto-balancing robot which is a free-standing robot that maintains its balance. The robot will balance on two wheels and use two sensors, a gyroscope and feedback to determine the current angular position versus the desired angular position. No readymade kits/chassis are allowed. No external controls are allowed.

COMPONENTS

1). Structure Components:

Components	Manufacturer
Plates	3D-Printed using Institute lab facilities
Wheel	Electronics Comp.
Shaft	Xinjinghua Shaft
Rods	Ambica Steel
Acrylic Sheets for Casing	3D-Printed using Institute lab facilities
Nut Bolt	Usha Fasteners

2). Electronics Components:

Components	Manufacturer
Microcontroller	ESP 32
Motor Driver (L298N)	Robodo

11.1 Li-Po Battery	MatLogix Technologies Private Limited
Motor	Robu
MPU6050 Gyroscope sensor	Electronic Spices (India)

Description of Components:

1. **Plates & Rods:**

   The design consists of three plates and four rods. The bottom two plates serve the purpose of providing space for all the electronic and mechanical components, while the topmost plate is intended for transportation purposes, carrying the required payload. The rods provide support to the robot's structure, as the plates are connected to these rods.

2. **Wheels & Shaft:**

   There are 2 wheels which will handle all the load and will be transmitted to the main body by the bearings. The shaft is attached to the motor to drive the wheel, there are 2 shafts for each wheel.

3. **Acrylic Sheets for Casing:**

   Two acrylic sheets will encase the electronic and mechanical parts, attached to the lower two plates from the front and back, to avoid and minimize any damage. Acrylic was chosen for its lightweight nature and its ability to be easily crafted into the desired dimensions.

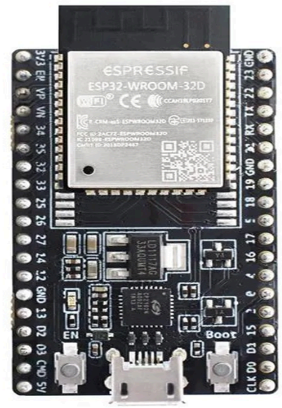
4. **Nut Bolt:**

   These are utilised to provide robust mechanical joints and secure the structure of the robot in place.

5. **11.1 Li-Po Battery:**

   The 11.1V 3S 3300mAh 40C Lipo Battery with XT60 Connector is a 3s1p battery pack with a nominal Voltage of 11.1V. This battery pack has 3 cells connected in series with a capacity of 40 C continuous discharge. The nominal capacity of the battery pack is 3300mah.

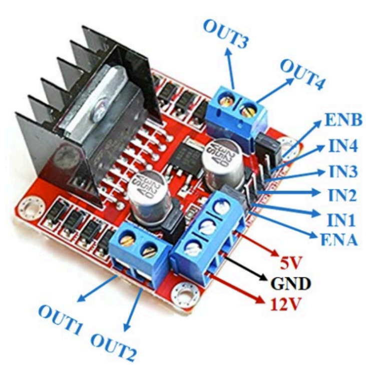
6. ESP 32 (Microcontroller):



https://robu.in/wp-content/uploads/2022/07/ESP32D-23.jpg

The ESP32, built around the Tensilica Xtensa LX6 microprocessor, features dual-core and single-core variations with clock frequencies up to 240 MHz. It encompasses 34 digital input/output pins (16 of which support PWM output), 12-bit ADC with 18 channels, 2 8-bit DAC channels, touch sensor capabilities, and an integrated 2.4 GHz Wi-Fi and Bluetooth 4.2/BLE module. The ESP32 also includes SPI, I2C, and UART interfaces, a micro-USB connection, a power jack, and a built-in reset button. Additionally, the board has a flash memory capacity ranging from 2MB to 16MB, facilitating extensive data storage and processing capabilities. It includes a CP2102 USB-to-UART bridge for easy programming and debugging. The ESP32 is designed to facilitate advanced IoT applications and can be powered via USB, AC-to-DC adapter, or battery, enabling versatile deployment across various projects.

7. L298N Motor Driver:

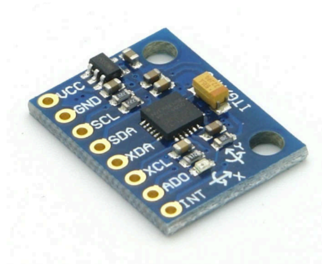


Pinout Diagram

https://images.app.goo.gl/XWcTdJpQxTTyAbzAA

L298N is a motor driver which can control the speed and direction of up to two DC motors. This motor driver is one of the cheapest and easiest ways to control DC motors. One more advantage is that it can control a stepper motor as well. The main central processing unit of the L298N motor driver is the L298N chip which comes with a big, black heat sink. The L298N IC is a dual-channel H-bridge motor driver which can run two DC motors simultaneously. This makes the L298N motor driver ideal for two and four-wheeled projects.

8. MPU6050 (accelerometer & gyroscope sensor):



MPU6050 Module

https://images.app.goo.gl/5Y3bpCs89SrEFoXF7

The MPU6050 is a Micro-Electro-Mechanical System (MEMS) that consists of a 3-axis Accelerometer and a 3-axis Gyroscope inside it. This helps us to measure acceleration, velocity, orientation, displacement and many other motion-related parameters of a system or object. This module also has a (DMP) Digital Motion Processor inside it which is powerful enough to perform complex calculations and thus free up the work for Microcontroller.

9. MOTOR:

https://bit.ly/4boZ7ZB

This 12V DC motor provides a torque of 5 Kg-cm, making it suitable for heavy-duty applications. With a rotational speed of 1000 RPM, it offers both power and efficiency. The motor features a 6mm shaft, designed for durability and stability. The geared mechanism enhances its torque capabilities, making it ideal for automotive and industrial tasks.

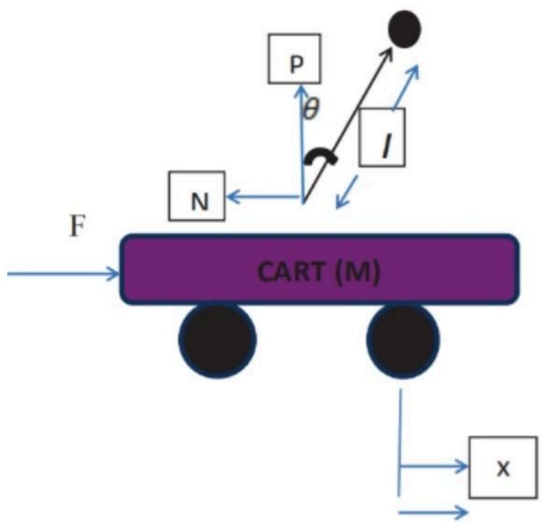
MECHANISM

This section of the report aims to give a holistic view of how the two-wheeled balancing robot will be realised and the concepts that we plan to employ to achieve our objective.

The mechanism of a self-balancing robot constitutes the integration of several data that is acquired by the utilisation of a variety of components as stated above.
It is based on the physics of an inverted pendulum.

Inverted Pendulum mechanism:

The inverted pendulum mechanism has been recognized as an extremely complicated mechanism to control as a part of control engineering. The reason behind this perpetual challenge is attributed to the non-linear of the mechanism.



The figure above shows an inverted pendulum with a cart.
A pendulum of this kind will simply fall over if not displaced slightly.

To solve this further, the mathematical model obtained is:

$$M\ddot{x} + b\dot{x} + m\ddot{x} + ml\ddot{\theta}cos\theta - ml\dot{\theta}^2 sin\theta = F$$

$$I\ddot{\theta} + ml^2\ddot{\theta} + mglsin\theta = -ml\ddot{x}cos\theta$$

Source: 2014 International Conference on Robotics and
Emerging Allied Technologies in Engineering

The above equations in linear form would be:

$$(I + ml^2)\frac{d^2\varphi}{dt^2} - mgl\varphi = ml\frac{d^2x}{dt^2}$$

$$(M + m)\frac{d^2x}{dt^2} + b\frac{dx}{dt} - ml\frac{d^2\varphi}{dt^2} = u$$

The state space models can be obtained from the equations. Then simulation software will be used to detect and confirm that without control, this system will indeed be rendered unstable.

The Robot

To be able to describe the orientation of the robot under different circumstances and to be able to identify its stable configuration, we need to define some angles and distances.

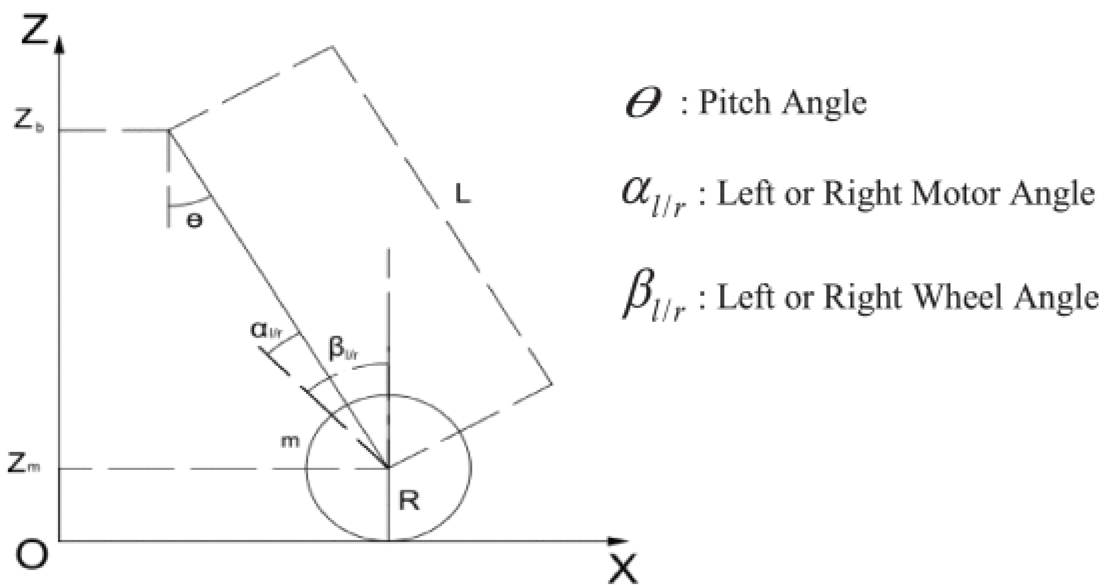


Fig. 1: XOZ modeling

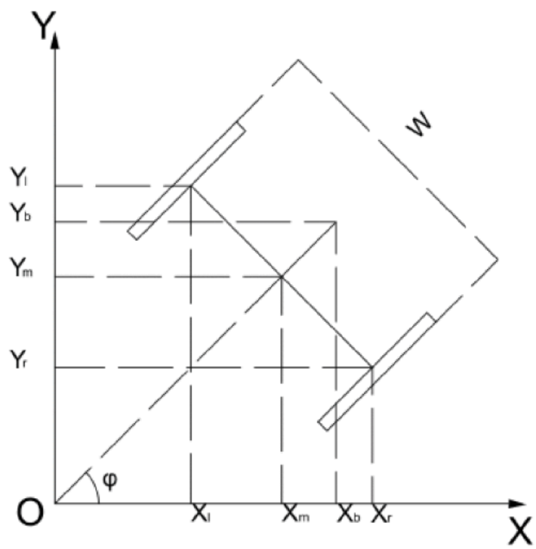


Fig2: XOY modeling

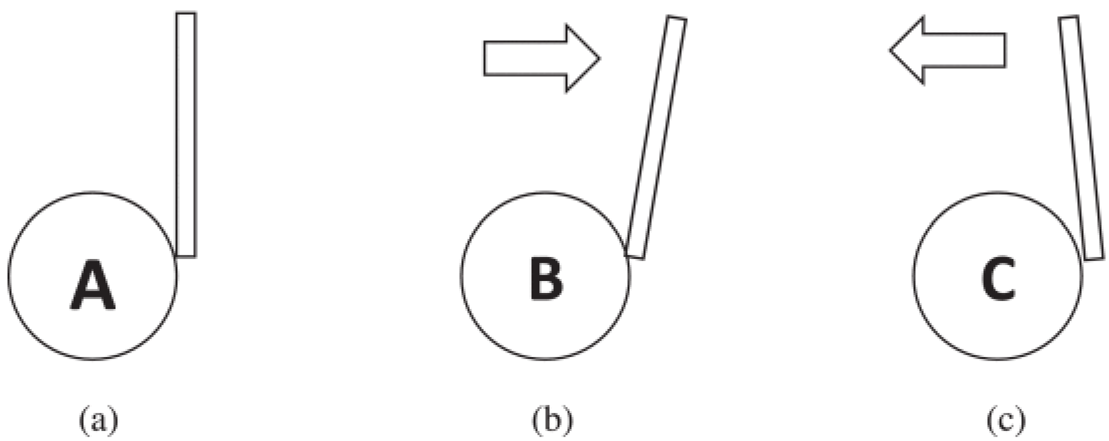


Fig. 3: (a)Describes the stable state of the two-wheeled self-balancing robot. (b) Indicates the robot leans to the right. (c) Demonstrates the robot leans to the left.

Establishing the dynamic model in this project involves the physical principles of the Lagrangian equation and Newton's Second Law in three-dimensional motion.

We would arrive at a result by a deep analysis of the relationship between the forces and motors' voltage through a series of mathematical derivatives in the X-O-Y and X-O-Z directions.

CALCULATION

a) Calculation of Center of Mass (COM)

ROBOFEST

CALCULATION of COM

Date / /
Page No.
Shivalal

mass of L298N motor Driver = 26gm

mass of Battery = 23gm

mass of Batteryholder = 15gm

mass of MPU6050 = 2.1gm

mass of ESP32 = 5gm

mass of rod $= 21 * 4 * \pi (0.5)^2 \rho_R = 494.55$ gm

(Where $\rho_R = 7.6$ g/cm^3)

mass of lower tray $= 20 * 15 * 0.3\ \rho = 90\ \rho$

mass of upper tray $= (20 * 15 * 0.3 + 20 * 2 * 2 * 0.3 + 15 * 0.3 \times 2 \times 2)\rho = (90+32)\rho$

mass of addition attach/deattach tray $= (90+32)\rho = 122\rho$

(Where $\rho = 7.6$ gm/cm^3)

mass of motor $= 130 * 2$ gm

mass of wheels $= 2 * 35$ gm

* Let us consider lower plate as our reference point to calculate COM in y-axis.

distance of motor's COM = −2.25 cm

Distance of wheels's COM = −2.25 cm

Distance of rods' COM = 10.5 cm

Distance of top plate COM = 21 cm

Distance of battery = −1 cm

Distance all other components avg. weight (12gm) and avg. COM is = 0.5 cm

⟹ As this is symmetric ~~vehicle~~ robot in x & z axis hence we need <u>NOT</u> to calculate COM in those axis.

$$y_{com} = \frac{(0 * 90\rho) + (132\rho * 21) - 2.25(260) - 2.25(70) - 1(92) + 0.5(12.1) + 10.5(494.55)}{222\rho + 440.1 + 494.55}$$

$$y_{com} = \frac{2772\rho + 4367.32}{222\rho + 934.65} = 9.7 \text{ cm from lower tray.}$$

⟹ <u>NOTE</u> : We choose <u>$\rho = 7.6$ g/cm^3</u> because it is density of stainless steel which is hard and tough material and along with that it is maintaining the mass of overall body within the range we needed.

- Net mass (without additional tray) = <u>2.621 kg</u>
- Net mass (with tray) = <u>3.625 kg</u>.

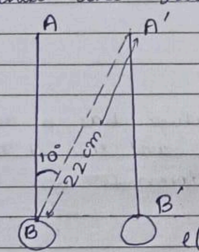
⟹ We ~~will~~ will place our additional tray on COM on the body so that it will <u>NOT</u> affect the posn of COM of whole body and hence will be easy to balance.

b) Calculation of Torque and RPM

Under free fall condition, an object would take $t = 0.247\,s$ to reach the ground from a height of 30cm is [using: $S = ut + \frac{1}{2}at^2$]

Thus, as a limiting condition we aim to balance our robot within 0.2s.

Also, the maximum deviation / inclination that the robot can balance = 10°



AB → Initial stable position

A'B → Intermediate unstabilised position.

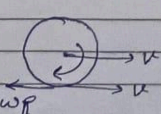
A'B' → final stabilised position

$l(AA') = 22 \sin 10°$

$= 3.8202\,cm$

required speed $v = \frac{s}{t} = \frac{3.8202\,cm}{0.2\,s} = 19.101\,cm/s$

From the concept of rotation applied at the wheels, we know:



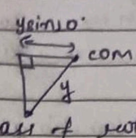
$v = \omega R$

Radius of wheel, $R = 3.465\,cm$

$\therefore \omega = \frac{v}{R} = \frac{19.101}{3.465} = 5.51\,rad/s$

$= 52$ RPM

• Let the COM; y coordinate be 'y'



$\tau = My\sin 10°$

where M = Mass of robot

~~M = 1937g = 1.937Kg~~

~~y =~~ M = 2.6Kg

y = 9.7cm

$\tau = 2.6 \times 9.7 \times \sin 10$

$\tau = 4.37$ Kgcm

thus we use a motor which has a rating of 1000RPM and 5Kgcm torque and use a motor driver to bring the RPM down to 60.

PID Control

To maintain the equilibrium and stability of the robot, it is necessary to generate a suitable control signal that drives the motors in the proper direction and at the proper speed. We will accomplish this by employing a well-known control loop algorithm called a PID controller.
PID stands for Proportional Integral Derivative. The controller has three control actions:

Proportional Control (Kp): Multiplies current error by Kp to provide immediate response proportional to error magnitude.
Integral Control (Ki): Accumulates errors over time and adjusts control output to eliminate steady-state errors using Ki.
Derivative Control (Kd): Calculates error rate of change and adjusts control to anticipate system behavior, contributing to stability with Kd.

Its control architecture is feedback, where the input to the controller is the system error signal e(t). It yields an error value by calculating the difference between a point of reference and an actual point.
The output of a PID controller, which is equal to the control input, is calculated in the time domain from the feedback error as follows:

$$u(t) = Kp.e(t) + Ki\int e(t)dt + Kd\frac{de(t)}{dt}$$

Particularly in the derivative term, PID controllers are susceptible to measurement noise that can amplify high-frequency noise and result in erroneous control actions. In addition, the output of MPU6050 signals is frequently tainted with noise. To mitigate this, a ***Kalman filter*** is implemented, which focuses primarily on the issue of noise sensitivity in PID controllers. The Kalman filter improves the precision of state estimations and assists in mitigating the detrimental impacts of measurement noise on the control system through the reduction of sensor noise.
Here is how this filter works and its role in the control system.

- **State Estimation**: Kalman filter processes accelerometer and gyroscope data to estimate tilt angle, angular velocity, and possibly angular acceleration.
- **Sensor Fusion**: Combines accelerometer and gyroscope data for more accurate state estimation than either sensor alone.
- **Noise Reduction**: Mitigates sensor noise effects by incorporating statistical models into the estimation process.

- **Dynamic System Model**: Integrates the dynamic model of the system to describe state variable evolution based on control inputs and system dynamics.
- **Feedback for Control**: Provides estimated state feedback to the PID controller, guiding control actions to drive the robot towards the desired state, such as an upright position.

CAD MODEL

*ALL THE CAD MODELS ARE MADE ON THE SOLIDWORKS SOFTWARE

Two Wheeled Self Balancing Robot CAD Model

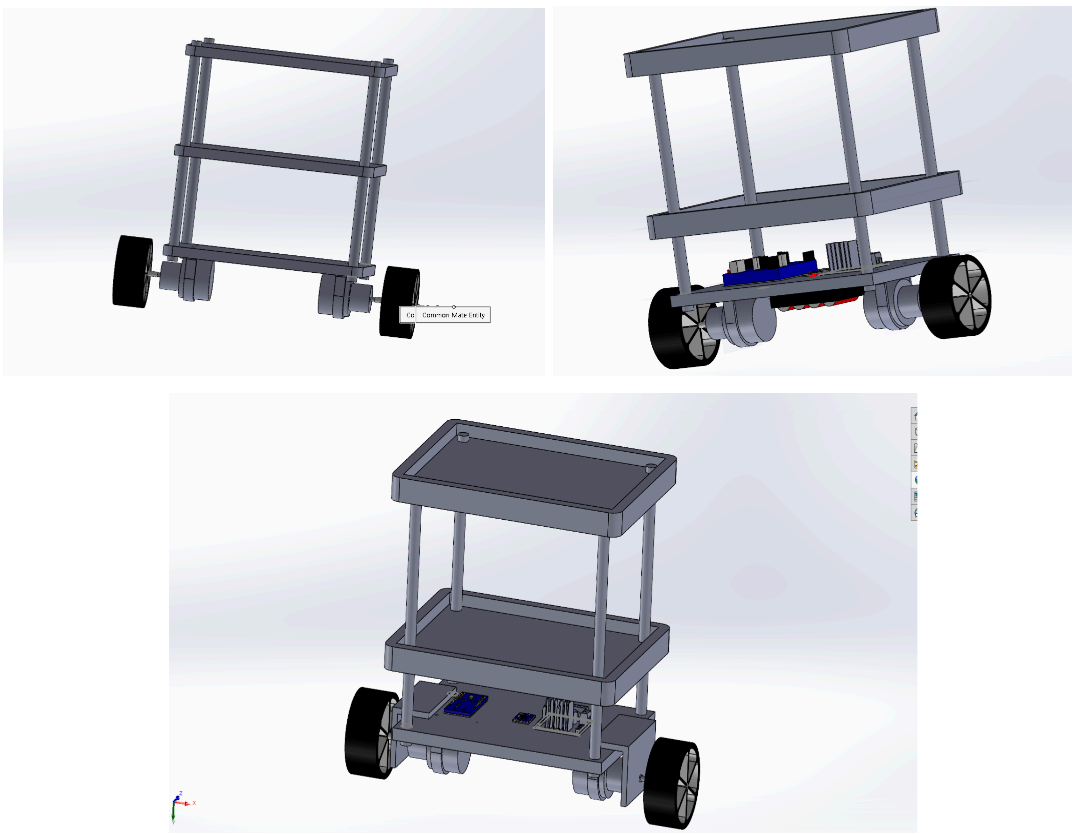


Fig 1: Final Design of CAD Model along with its stages.
https://github.com/Divyakumar6163/TWSBR/blob/main/CAD%20MODEL/TWSBR.SLDASM

This is the overall CAD model that includes every part we will be using for the two-wheeled self-balancing robot. The model is 20 cm tall, 10 cm wide(wheels included), and 15 cm long. After analyzing a number of designs, we ultimately developed the one seen above. In the final model, we placed the battery and its casing underneath the last plate instead of the original placement. This change was made to better utilize the available space and more effectively arrange the remaining components. Additionally, at first, all of the plates were spaced equally apart, but later on, we adjusted this because we needed more room between the upper locations to accommodate everything, and we also reduced the space between the final two because that is where we would place all of the electronic components in order to maximize efficiency.

Wheel

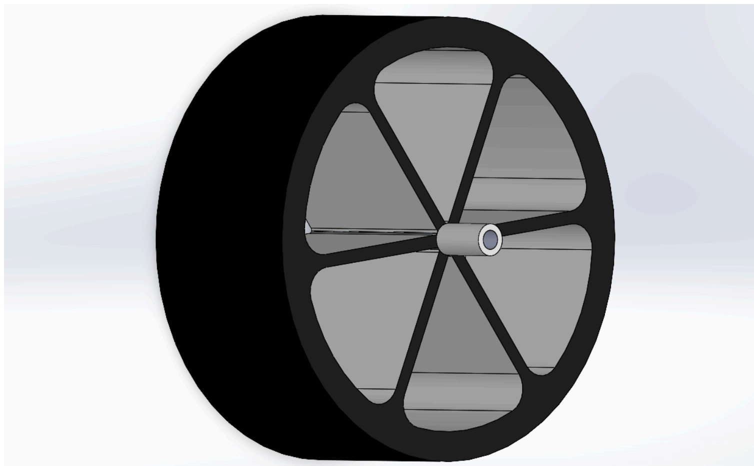


Fig 2: CAD Model for Wheel of TWSBR
https://github.com/Divyakumar6163/TWSBR/blob/main/CAD%20MODEL/Wheel.SLDPRT

The wheel is 6.5 cm in diameter and 3 cm thick. There will be an extension of about 2.2 cm for placement into the motor shaft. The aforementioned dimensions are displayed in the cad model This wheel fits completely within the tolerance limit and is the most accurate and precise fitment. The diameter of the extension part in the wheel is 3mm.

ESP 32

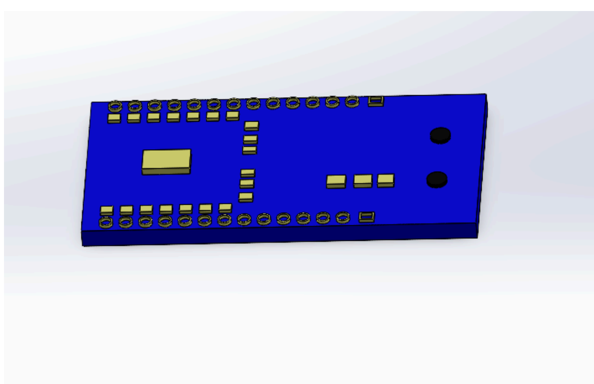


Fig 3: ESP 32 (MicroController)
https://github.com/Divyakumar6163/TWSBR/blob/main/CAD%20MODEL/ESP32.SLDPRT

The ESP32 microcontroller model shown above was created by taking into account just the crucial components of the microcontroller. Located above the plate is the microcontroller.

Battery Casing

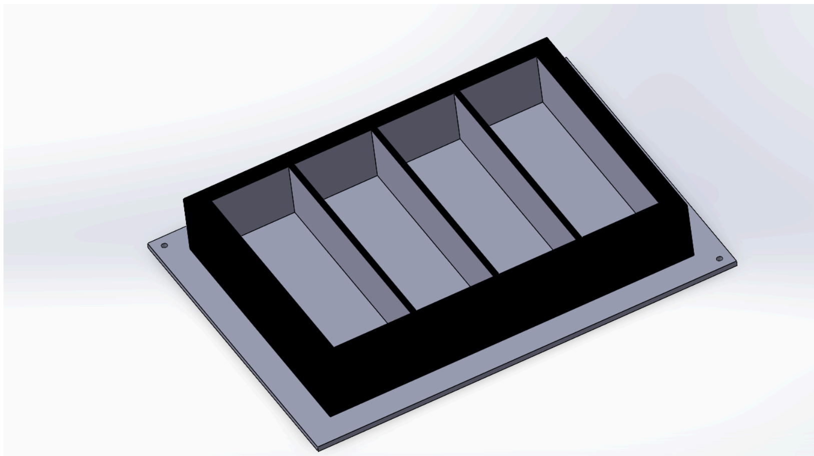


Fig 4: CAD of Battery Casing
https://github.com/Divyakumar6163/TWSBR/blob/main/CAD%20MODEL/Batter%20Casing.SLDPRT

A Cad model Battery casing, is made to hold four AA batteries (11.1 V each) at once and to keep the batteries separate from one another. Battery separation is essential to prevent the batteries from losing touch with one another. The battery shell is 7.6 cm by 5.4 cm in total. The precise dimensions are 4.4 x 6.6 cm.

Battery

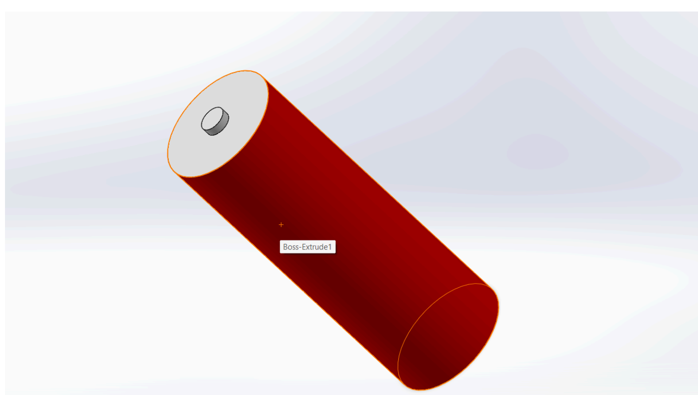


Fig 5: A cell to power the system
https://github.com/Divyakumar6163/TWSBR/blob/main/CAD%20MODEL/Battery.SLDPRT

The battery that we are going to use to supply the project with power is represented by the CAD model that can be seen above. Each of the batteries has the capacity to supply 11.1V, and we are utilizing four of them.

Battery & Battery Casing

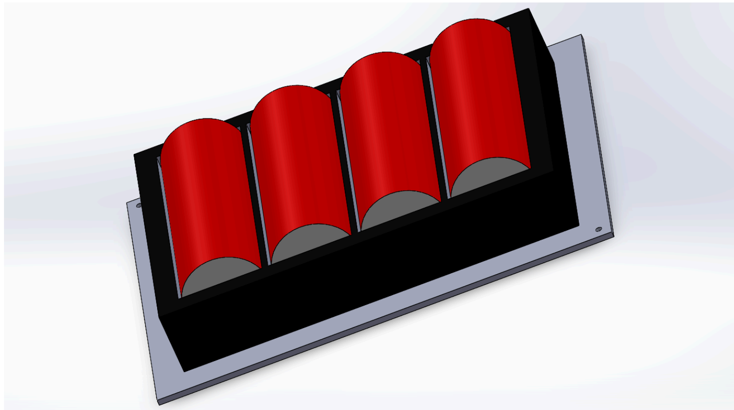


Fig 6: Assembly of battery and battery casing
https://github.com/Divyakumar6163/TWSBR/blob/main/CAD%20MODEL/Battery%20Assembly.SLDASM

This is the assembly of the battery and battery casing which is directly going to connect with the main body i.e. main casing skeleton of the model.

Main Casing

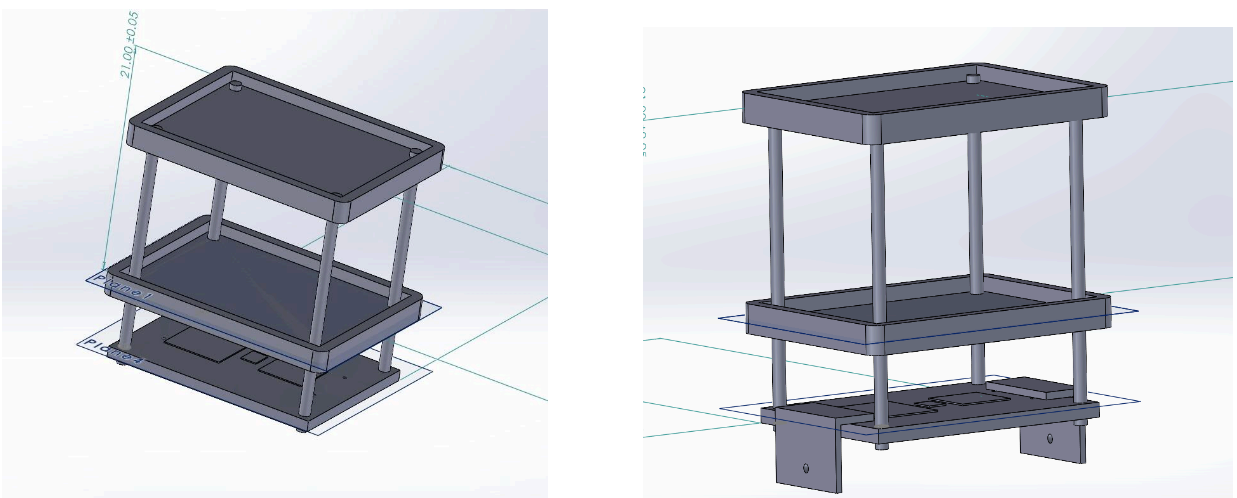


Fig 7: Outer Structure of Robot
https://github.com/Divyakumar6163/TWSBR/blob/main/CAD%20MODEL/main%20casing.SLDPRT

This shows the two stages of designing the casing of the robot which serves as the skeleton of the project. The fetal structure on the left throws light on the initial idea of the frame. Eventually, we decided to add an extension to the model so that a bearing can be attached. This bearing would prevent the bending of the shaft and consequently save the motor from damage.

MPU 6050

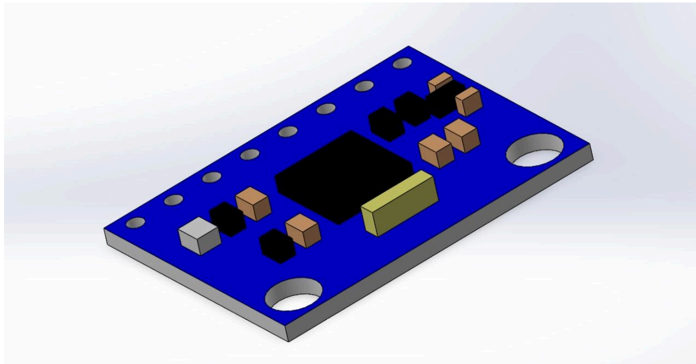


Fig 8: CAD of gyroscope
https://github.com/Divyakumar6163/TWSBR/blob/main/CAD%20MODEL/MPU%206050.SLDPRT

The above figure shows the CAD model of the gyroscope. The model that we are going to be using for this project is the MPU6050. It is going to give us the motion-related parameters of the robot which is going to be utilized in the equations to stabilize the robot and balance it well.

Motor

Fig 9: Motor and clip to hold motor on the body of robot.
https://github.com/Divyakumar6163/TWSBR/blob/main/CAD%20MODEL/motor.SLDPRT

This is the motor the robot is going to control the movement. Initially, there were no bearings then we added them as it is the most important part of the robot to avoid the motor from being destroyed.

COST ANALYSIS

Component	Cost
Micro-controller	₹ 500 - ₹ 2000
Motors	₹ 200 - ₹ 800
Motor Drivers	₹ 100 - ₹ 150
Battery	₹ 200 - ₹ 300
Accelerometer & Gyroscope sensor	₹ 200 - ₹ 600
Plates/Casing & Rods	₹ 500 - ₹ 700
Wheels	₹ 50 - ₹ 100
3D Printing Filament	₹ 700 - ₹ 900

*The overall Estimated cost to make such a project is **₹ 2450 - ₹ 5550** at the "**Proof of Concept" stage.***

APPLICATIONS

Our project constitutes a two-wheeled self-balancing robot which is accommodated in a design that consists of three plates and largely focuses on domestic utility. The bottom two plates serve the purpose of providing space for all the electronic components and the topmost plate is going to be utilized for transportation purposes. This versatile model also holds the potential to cater to industries with customizable modifications to suit the nature of the application. The design and compact nature of this robot make it an easy alternative for personal transportation of goods and surveillance. This project is thus a step towards redefining the industrial landscape and presents our take on how a robot with such a vast potential can be designed.

Being actively stabilised means that the robot can correct for any disturbances which may otherwise cause a statically stable robot to tip over, even with a higher centre of mass. Two-wheeled robots can be taller, with a small footprint, making them quite suitable for indoor environments, because they can fit through narrow corridors and tight corners. Because two-wheeled robots can lean fore and aft to re-stabilize themselves, they are also able to maintain stability on inclines, by leaning into the slope.

FUTURE SCOPE OF DEVELOPMENT

From the point of view of future development, the following areas hold tremendous scope:

1. Sensors: Using sensory devices such as an ultrasonic sensor that intimates the robot regarding the surroundings in which it is operating can be procured and put to use to identify obstacles.

2.Navigation: Surveying its surroundings is a feature that would boost the utility of a self balancing robot beyond the current scope. This can be realized by a combination of machine learning into the system. This could be a massive step towards complete autonomous navigation.

3. Payload Capacity: The maximum weight that such robots can carry is often a limiting condition. More the limit is pushed, the more the range of operation of that particular robot expands.

4. Energy Efficiency: Efficient consumption of energy in today's world is of prime importance considering the scarce nature of resources. Thus using powering devices (batteries) and powered devices (motors), of better ratings and efficiency is essential in order to improve the performance of the robot

5. Mechanical Design: Calculations often have a large scope for precision along with some mechanical considerations. The incorporation of these along with experience into the model can act as a catalyst to magnify the efficiency of a TWSBR.

TIMELINE

***We postponed every step in the timeline by 40 days due to the ideation deadline extension.**

Here is the brief structure of our Timeline and we will stick to it.

1. **July 2024: Gearing Up**
    - July 1 - July 15: Procurement Phase
      Allocate 1.5 weeks for researching, sourcing, and purchasing necessary components and materials.
    - July 16 - July 22: Assembly Phase
      Allocate 1 week for the assembly of the bot, ensuring all components are correctly integrated.

2. **Coding and Programming Phase**
    - July 16 - Augest 6: Concurrent Coding
      - Begin coding simultaneously with assembly to maximize efficiency.
      - Develop the software architecture for the self-balancing mechanism.

3. **PID Control Implementation**
    - August 7 - Augest 13: PID Tuning and Integration
      -Dedicate a week to implement and fine-tune the Proportional Integral Derivative (PID) control algorithm.
      -Test and adjust parameters to achieve optimal balance and stability.

4. **Testing and Iteration**
    - Augest 14 - Augest 27: Intensive Testing
      - Conduct rigorous testing to evaluate the bot's performance under various conditions.
      - Identify limitations and areas for improvement through iterative testing.

5. **Documentation and Finalization**
    - Augest 28 - Augest 31: Documentation and Submission
      -Compile documentation detailing the design, development process, and testing results.
      - Ready with the "Proof of Concept" for submission to ROBOFEST4.0.

**All the dates mentioned above are tentative and subject to change per the guidelines provided.

DIMENSIONS

Size of Robot Proposed for Proof of Concept

- Length in cm: 15cm
- Width in cm: 10cm
- Height in cm: 20cm

The size of the Robot proposed as a prototype

- Length in cm: 45cm
- Width in cm: 30cm
- Height in cm: 60cm

ABOUT THE TEAM

Mentor:

Dr. Satyanarayan Panigrahi
Associate Professor
School of Mechanical Sciences

Ph.D. Mechanical (Technical Acoustics) - 2007
Indian Institute of Science, Bangalore

Research Interests

Technical Acoustics, Industrial Noise Control, Automotive noise control, Acoustic based Condition Monitoring of Machines, and Musical Acoustics.

Team Members:

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

REFERENCES

- [1] C. Iwendi, M. A. Alqarni, J. H. Anajemba, A. S. Alfakeeh, Z. Zhang and A. K. Bashir, "Robust Navigational Control of a Two-Wheeled Self-Balancing Robot in a Sensed Environment," in IEEE Access, vol. 7, pp. 82337-82348, 2019, doi: 10.1109/ACCESS.2019.2923916.
  keywords: {Mobile robots;Robot sensing systems;Mathematical model;Wheels;Robot kinematics;Kalman filters;TWSBR;32-bit Microcontroller;robust control;PD-PI;sensed environment;IoT},

- Ronald Ping Man Chan, Karl A. Stol, C. Roger Halkyard,
  Review of modelling and control of two-wheeled robots,
  Annual Reviews in Control,
  Volume 37, Issue 1,

GITHUB LINK: https://github.com/Divyakumar6163/TWSBR