iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VI **Month of publication:** June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72612

Object Carrying Two Wheeled Self Balancing Robot: A Novel Approach

Shounak Bandyopadhyay[1], Subhadip Das[2], Sukhendu Bikash Banerjee[3], Nikhil Kumar Sah[4], Pranay Majhi[5], Sourav Samanta[6]

[1]*Assistant Professor, Department of Electronics and Communication Engineering, AIEM, Mogra*
[2]*Assistant Professor, Department of Mechanical Engineering, AIEM, Mogra*
[3]*Assistant Professor, Department of Civil Engineering, AIEM, Mogra*
[4, 5, 6]*B.Tech Student, Department of Mechanical Engineering, AIEM, Mogra*

Abstract: The Two-Wheel Self-Balancing Robot is a promising application of robotics and control systems that demonstrates the ability of a machine to maintain its equilibrium on two wheels. This article utilizes an accelerometer and gyroscope sensor in conjunction with an Arduino Uno microcontroller to achieve precise balancing and control.

The sensor provides real-time data on the robot's orientation and angular velocity. The Arduino Uno processes this data to calculate the necessary adjustments to maintain the robot's balance. The control algorithm uses a proportional-integral-derivative (PID) controller to make continuous adjustments to the motor speeds, ensuring that the robot remains upright.The sensor data is collected and processed to determine the robot's tilt angle, and the PID controller computes the appropriate motor speed corrections to counteract any deviations from the desired upright position.This article serves as an excellent educational platform for learning about control systems, sensor integration, and real-time data processing in the context of robotics. The successful implementation of this robot demonstrates the practicality and potential applications of self-balancing technology in various fields, including transportation and automation. Moreover, the tuning of PID using the heuristic method is also carried out, and the robot's balancing is noticeably improved.

Keywords: Self-balancing, invertedpendulum system, Arduino UNO,PID Controller, L298N Motor driver

I. INTRODUCTION

In the fields of robotics and control engineering, self-balancing robot oriented research and technological breakthroughs have gained traction. The idea of an inverted pendulum serves as the fundamental operating principle behind self-balancing robots. Examples of inverted pendulums in real-world applications include rockets like MAXUS in its launch position, and SEGWAYS, which transport personnel as a self-balancing vehicles .Figure 1 reflects a free body diagram of an inverted pendulum with a cart.

II. LITERATURE REVIEW

By using linear and non-linear controllers as well as adaptive and self-learning algorithms, M.R.M. Romlay aimed to address the control techniques of balancing two-wheeled mobile robots. The assessment and experimentation conducted on the two-wheel mobile robot will be the main topic of the review. From self-balancing, navigating, or avoiding obstacles, the goal of mobile robots progresses to performing complex external tasks like transportation and environmental monitoring [1]. Ivoilov, Zhamud, and Trubin [2] develop a complementary filter to estimate tilt angle in order to mitigate mid- and low-frequency accelerometer distortions. To balance tilt angle and avoid erratic system movement in the horizontal plane, a proportional-integral-derivative (PID) controller is employed. Although Jung and Kim [3] used PID to regulate the angle, the cart position was uncontrollable. For mobile robot angle and position control, PID controllers are used as a comparison to neural networks (NN). Ali & Hossen construct a PID controller with a Kalman filter for accelerometer and gyroscope data [4]. Zhang, Zhang, Wang, Li, and Zhang [5] used a PID controller to keep the robot steady while it was hopping. The controller receives the angle between the mobile robot and the vertical axis and generates a torque instruction to drive the wheels and stable the robot in the vertical position. Wenxia and Wei [6] use the system's Q and R values to simulate a self-balancing robot model. Two linear controllers are combined by Gonzalez, Al-varado, and Pena [7] to create a low-cost balancing robot. The tilting angle, tilting speed, and tilting angle of the wheel are all controlled by the LQR controller. After that, the wheel's angular acceleration will be sent to the PI controller for magnitude control.

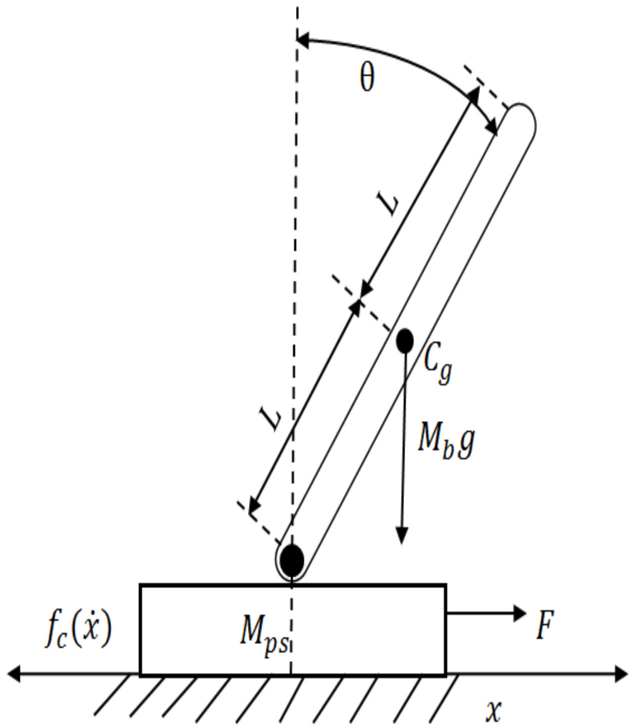


Fig 1 Free body diagram of an inverted pendulum system with cart

One significant category of mobile robots is the two-wheeled self-balancing robot. A two wheel self-balancing robot only needs two points of contact with the floor surface, unlike a regular robot. The robot is distinct from other robots since it demands a special stability control to stay upright. A self-balancing robot's basic concept is as easy as driving the wheels in the direction that the robot tilts. Small size,flexibility,low cost provides an extra mileage to this device to be used in various application in the field of control engineering. The design and development of control systems for automobiles, spacecraft, and other transportation facilities, including military transport, make use of the platform's inherent complexity. The developed hardware is used to create an automated vehicle that can carry objects and reduce human effort in workplaces, offices, and household tasks. The robot depends on PID based Kalman filter control algorithm to maintain its balancing action. The robot consists of :

1) Arduino UNO(At Mega 328) : One sort of microcontroller board developed on the ATmega328 is called the Arduino Uno where uno is an Italian word that signifies one. This board has 14 digital I/O pins, an ICSP header, a power jack, 6 analog I/Ps, a ceramic resonator operating at A16 MHz, a USB port, and 14 digital I/O pins. By connecting this board to the computer, all of these can help to increasethe microcontroller's intelligence activities. This board's power source can be accomplished via a battery, a USB cable, or an AC to DC adaptor.
2) MPU 6050 sensor module:An integrated 3-axis accelerometer and 3-axis gyroscope is the base unit of MPU6050. The robot location is sensed by this module, which, measures the tilt angle and acceleration value that are sent to the processor (ArduinoUNO) so it can be processed further.
3) L298N motor driver:The L298N is a popular integrated circuit (IC) used as a motor driver and controller in various electronic projects, particularly in robotics and automation. It is capable of driving two DC motors or a single bipolar stepper motor. The L298N IC provides bidirectional control of these motors, meaning it can control their direction of rotation as well as their speed.To use the L298N motor driver, we have to typically connect it to a microcontroller or another control circuit to provide the necessary signals to control the motors. The specific wiring and control sequences depend on the application and the type of motors we are using.
4) DC/DC Power Converter LM2596:The LM2596 is a popular voltage regulator IC that is commonly used as a DC/DC power converter or voltage step-down (buck) regulator. It is designed to take an input voltage higher than the desired output voltage and efficiently provide a stable and lower output voltage.

5) BO motors: Battery-operated geared DC motors are a common type of motor used in many applications, including robotics and various electronic devices.These motors are based on a DC (Direct Current) motor at their core. DC motors are known for their simplicity and reliability. They convert electrical energy from a battery or power source into mechanical motion.The addition of a gearbox to a DC motor results in a geared DC motor. The gearbox is used to reduce the motor's speed while increasing its torque output.The gear ratio of a geared DC motor defines the relationship between the motor's input speed (RPM - Rotations Per Minute) and the output speed. A higher gear ratio means the motor's output shaft will rotate more slowly but with greater torque. For our application we have used the gear ratio of 1 : 34.
6) Two wheels: The two wheels that are positioned vertically on the same axis is used in a two-wheel self-balancing robot. The wheel's main objective is to keep its equilibrium and avoid falling.
7) Power source: A switch connects a 12 V Lithium-Polymer battery, which is the power source, to the Arduino.

In earlier design approaches Arduino Mega 2560 is used as processor unit with sharp sensors like GP2D120,ultrasonic sensors HC SR-04,GY -521 as IMU unit & TB6612FNG as motor driver. To obtain adequate acceleration,proper positioning of center of gravity& fast response time we have used this design model. To make the system stable we are also using PID based Kalman filter algorithim and performance analysis is also done in this paper.

a) Controllers: The Linear Quadratic Regulator and Proportional-Integral-Derivative (PID) controls are most frequently employed to keep the robot upright. Some researchers have also investigated the use of fuzzy logic, pole-placement, and linear-gaussian control (LQG). However, in other instances, they were simply tested in simulations and never put into practice with actual robots. Either LQG or a combination of controllers are used where the robot displacement is also controlled. For instance, LQR is utilized to keep the robot upright and PID, or a cascaded PID controller, is employed to control displacement. The examined controllers, PID and LQR, are the ones for which more information will be provided in this article. PID may be the controller that is most commonly utilized. According to VanDoren, Proportional-integral-derivative controllers remain the workhorse of industrial process control, more than 60 years after their introduction. The following equation encapsulates the algorithm:

$$u(t) = Kp(t) + Ki\int e(t)\,dt + Kd\,d/dte(t)$$

Where K_p, K_i, and K_d are the tuning parameters, and u(t) is the controller's output, e(t) is the error. It is not dependent on a model of the system and is rather simple to implement. Trial and error can be implemented for fine-tuning of the parameters. This simple approach has been used to keep robots upright.

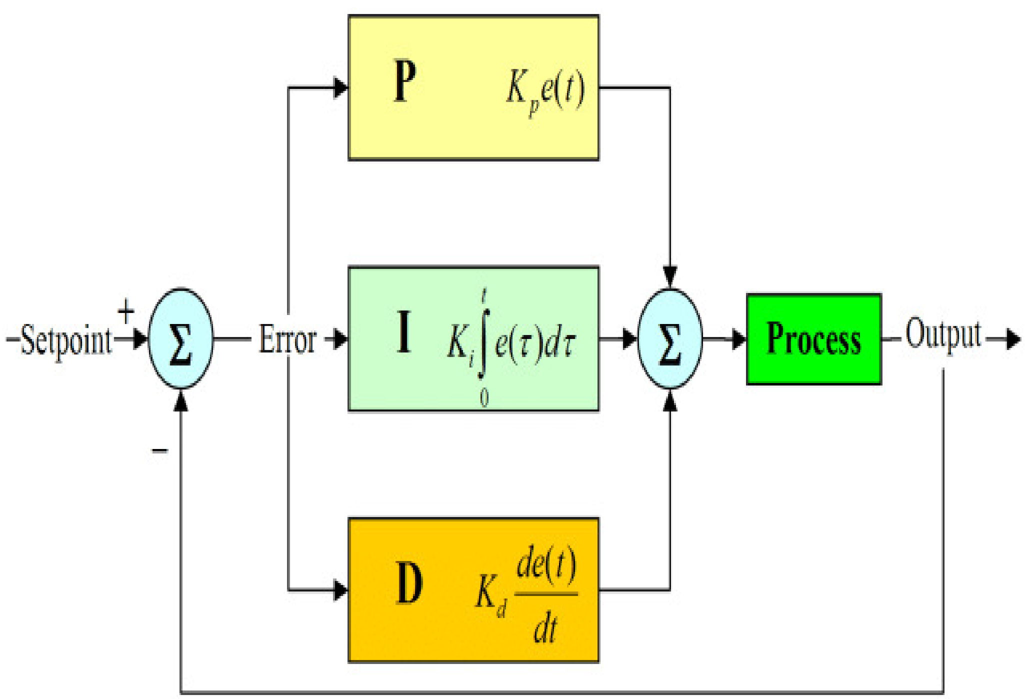


Fig 2 PID controller

b) Sensors :It is well acknowledged in the literature that utilizing a gyroscope or an accelerometer alone to determine the tilt angle is not particularly trustworthy. This mainly results from the fact that each of these instruments have a bias in their measurements, are influenced by noise, and have a temperature-dependent bias. Therefore, to estimate the tilt angle, we combine both terms.The gyroscope calculates angular velocity using the 'thieta symbol', which is expressed in radians per second or degrees per second. It makes intuitive sense to calculate the tilt angle by integrating the angular velocity. In relation to free fall, the accelerometer measures acceleration. As a result, when an object is tilted, the force is split into components in the object's x, y, and z axes.

c) Kalman filter:A Discrete Kalman Filter (DKF) is a recursive algorithm used for state estimation in dynamic systems. It operates in cycles or iterations as it continuously updates its estimate of the state based on new measurements. The signals from the accelerometer and gyroscope are put together using the Kalman filter to determine the tilt angle. The Kalman Filter implements a sort of feedback by first estimating the process state using the time update equations and then estimating the measurement using the measurement update equation.

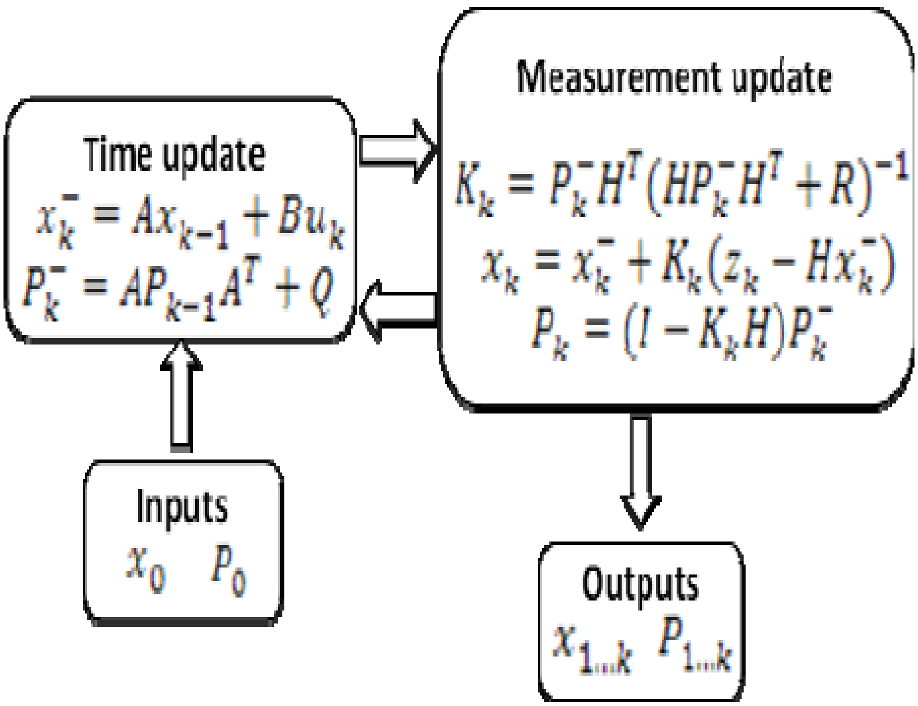


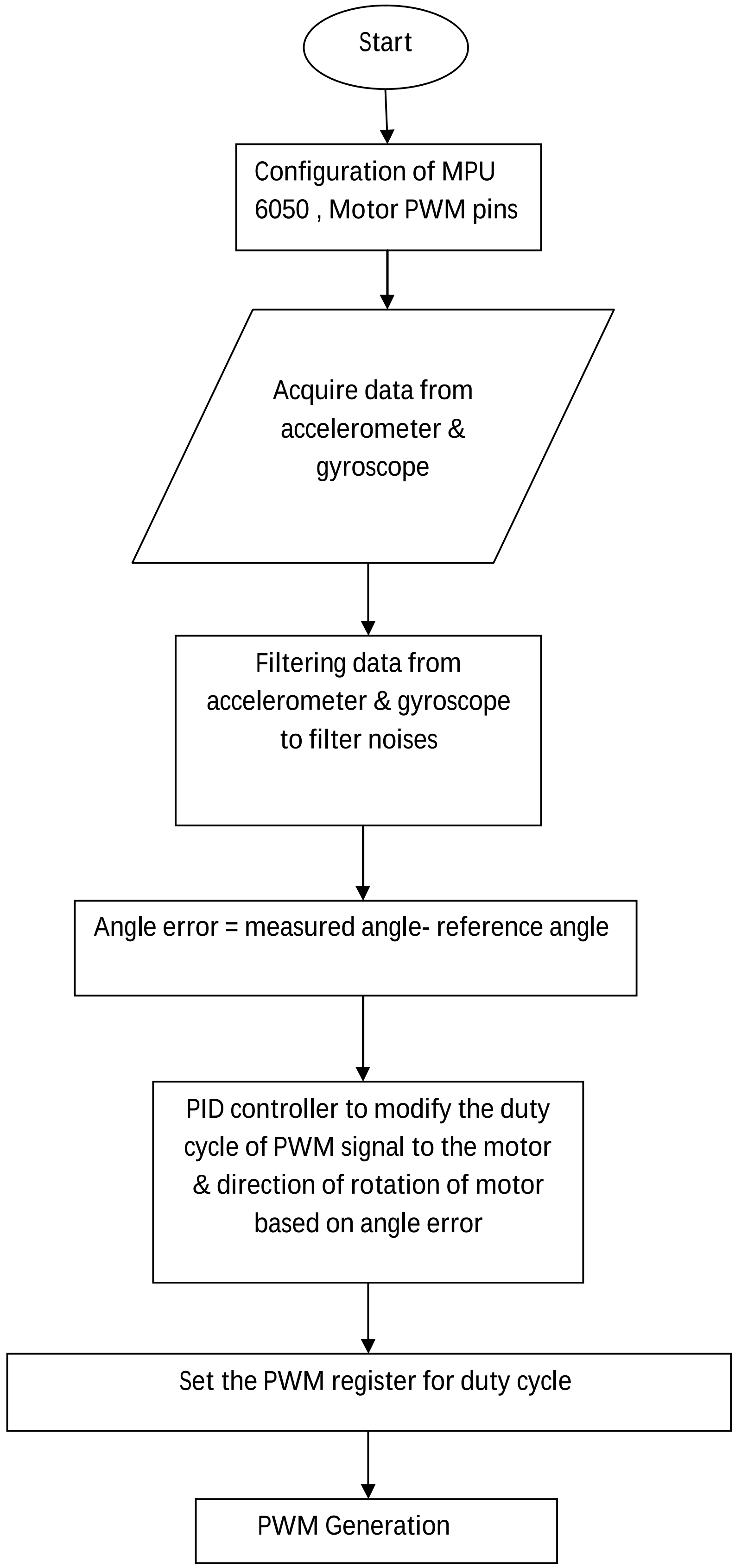
Fig 3 ongoing discrete kalman filter cycle

d) Tuning: Tuning a self-balancing robot, such as the one using an MPU-6050 sensor and Arduino Uno, is a crucial step to ensure that it can maintain its balance effectively. Tuning involves adjusting the parameters of the control algorithm, typically a PID (Proportional-Integral-Derivative) controller, to achieve stable and responsive control.A good control system exhibits several key characteristics (stability, accuracy,responsiveness, robustness,minimization of steady stae error, predictability etc.) that contribute to its effectiveness and reliability in regulating a process or system. These characteristics are essential for ensuring that the control system meets its intended objectives.

III. PROPOSED SYSTEM

1) Working Principle: Inverted pendulum configuration is the cornerstone for two-wheeled self balancing robots, which rely on dynamic balancing systems for balancing and movement. If a self-balancing robot tilts by an angle, the Arduino processor will get this value. The L298N motor drive will receive instructions from the Arduino in the form of new rpm values, positive or negative depending on the direction of tilt, after the Arduino processes these values using the PID-based Kalman Filter method. These instructions will tell the motors to compensate for the tilt angle. As a result, the bot's center of mass will encounter a pseudoforce that will apply a torque in the opposite direction of tilt.

2) Algorithm for self-balancing :



3) Block diagram of the proposed model:

The below mentioned block diagram illustrates the detailed principle of working of this article. The working of each block is already discussed.

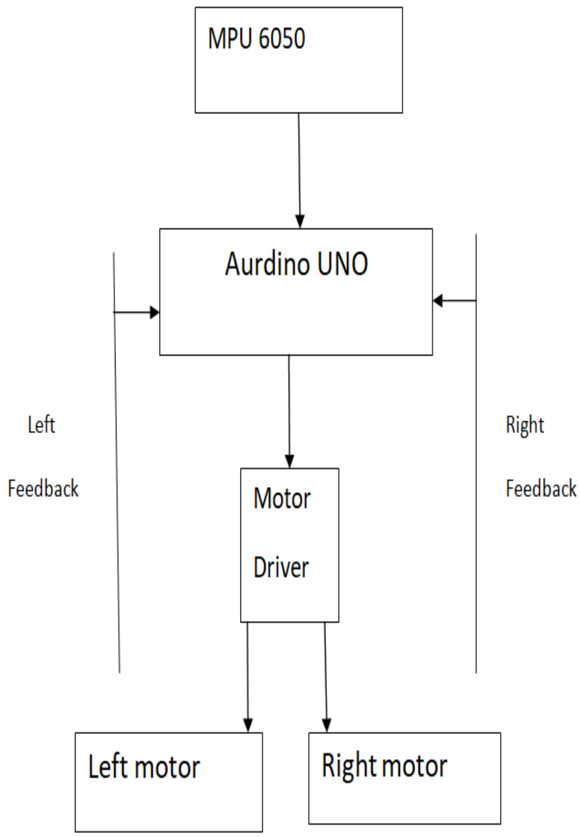


Fig 4 Block diagram for proposed model

The entire robot is balanced on two wheels that have the necessary grip and provide enough friction. Two steps must be taken in order to achieve the robot's verticality: first, the robot's angle of inclination must be measured; second, motors must be regulated to move forward or backward in order to create an angle of 0°.We discover that the robot becomes active and starts out in a little tilted position when the supply is begun. Here, the battery supplies power to both the Arduino UNO and the MPU sensor, which primarily consists of two sensors: the gyroscope and the accelerometer. The accelerometer measures the tilt angle, the gyroscope sends the calculation of angular velocity to the robot's wheels through the Arduino, after that sends it towards the motor driver to operate the robot's wheels. The robot's wheels move in the direction that it is falling, stopping it from falling. As a result, we can observe that every component is interfaced properly and operating synchronously. Figure 5 shows a front view image of our proposed model.

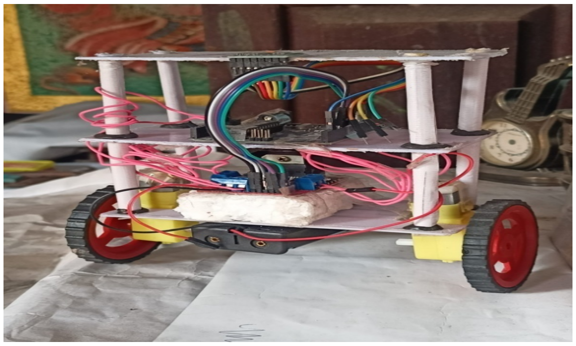
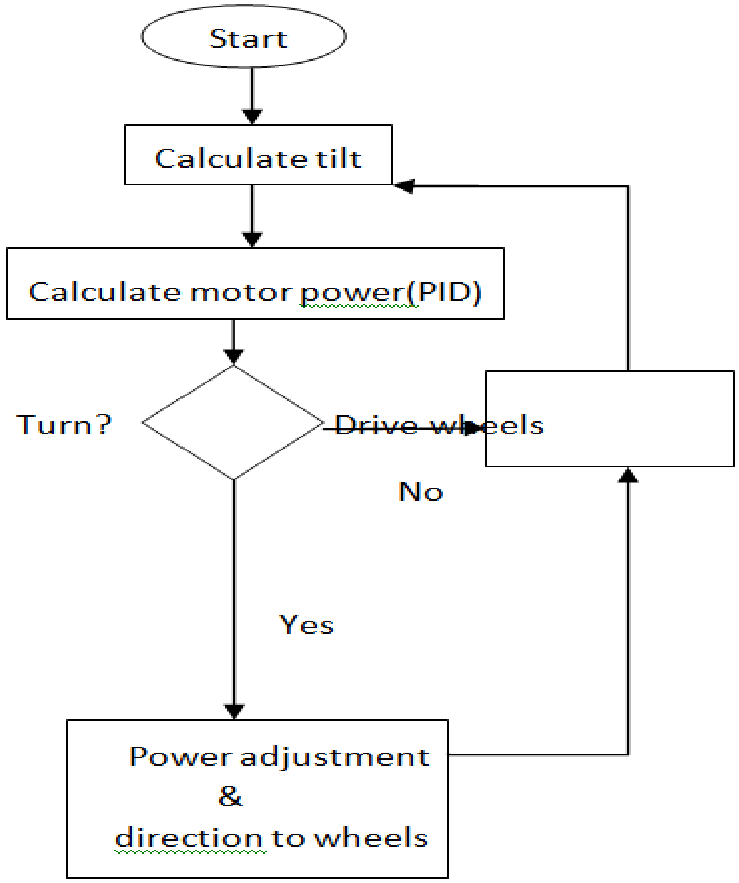


Fig 5 Complete picture of the model(Front view)

4) System Architecture:



IV. CONCLUSION

With the expectation of making the robot balance on its own using PID and Kalman Filter, the project's initial goals were all accomplished. The speed of the open loop offers line with fairly smooth values with only a few unexpected odd ones. The algorithm is also excellent, offering a very low settling time of the motor's. Testing of the PID algorithm for the motor has been done while maintaining the reference speed constant at varied voltages. The voltage variation has no impact on the speed of the motor implies that the PID Algorithm is functioning properly.Last but not least, it may be concluded that a learning platform was created through the creation and design of an self-balancing robot operating at its best.

REFERENCES

[1] M.R.M. Romlay, M.I. Azhar, S.F. Toha, M.M. Rashid"Two-wheel Balancing Robot; Review on Control Methods and Experiments" ; International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7, Issue-6S, March 2019.

[2] A. Y. Ivoilov, V. A. Zhmud, and V. G. Trubin, "The tilt angle estimation in the inverted pendulum stabilization task," 2018 Moscow Workshop on Electronic and Net-working Technologies MWENT 2018, vol. 2018-March, 2018.

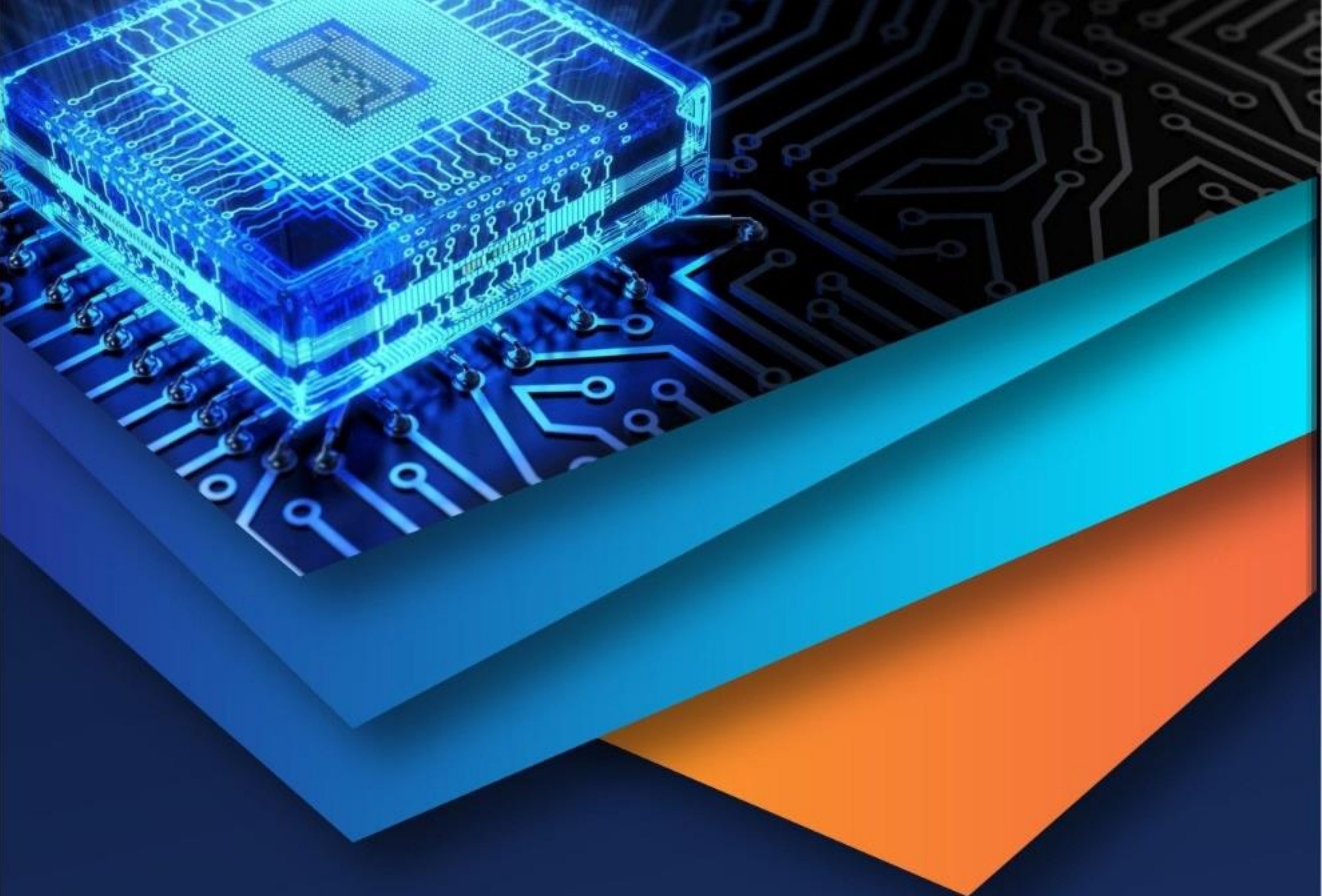
[3] S. Jung and S. S. Kim, "Control experiment of a wheel-driven mobile inverted pendulum using neural network," IEEE Trans. Control Syst. Technol., vol. 16, no. 2, 2008, pp. 297–303.

[4] M. I. Ali and M. M. Hossen, "A two-wheeled self-balancing robot with dynamics model," 4th Int. Conf. Adv. Electr. Eng. ICAEE 2017, vol. 2018–January, 2018, pp. 271–275.

[5] Y. Zhang, L. Zhang, W. Wang, Y. Li, and Q. Zhang, "Design and implementation of a two-wheel and hopping robot with a linkage mechanism," IEEE Access, vol. 6, 2018, pp. 42422-42430.

[6] S. Wenxia and C. Wei, "Simulation and Debugging of LQR control for two-wheeled self-balanced robot", 2017 pp. 2391–2395.

[7] C. Gonzalez, I. Alvarado, and D. M. La Peña, "Low cost two-wheels self-balancing robot for control education," IFAC-Papers Online, vol. 50, no. 1, 2017, pp. 9174–9179.



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY