Design And Implementation of Two-Wheel Self-Balancing Robot Using PD Controller

CHAITRA S[1], DR. MAHESHAN C M[2]
[1]*PG Scholar (Control and Instrumentation), Department of Electrical Engineering, University of Visvesvaraya College of Engineering, Bengaluru, Karnataka, India.*
[2]*Associate Professor, Department of Electrical Engineering, University of Visvesvaraya College of Engineering, Bengaluru, Karnataka, India.*

Abstract- In order to maintain upright stability, two-wheel self-balancing robots belong to a type of underactuated, intrinsically unstable systems that need constant feedback control. The design, modelling, and practical implementation of a two-wheel self-balancing robot employing a proportional-derivative (PD) control approach are presented in this study. An inverted pendulum supported by two driven wheels serves as the model for the robot. An inertial measuring device made up of an accelerometer and gyroscope provides real-time orientation feedback. To fuse sensor data and reduce noise and drift, a complementary filter is used. LabVIEW is used to build the control algorithm on an NI myRIO platform, where the PD controller produces pulse-width-modulated motor commands. According to experimental data, the PD controller can achieve stable balancing with low oscillations and a quick transient response when the gain is tuned appropriately. The paper demonstrates that classical PD control provides a dependable and computationally effective solution for self-balancing robotic platforms, making it appropriate for real-time embedded control, prototype-level, and instructional applications.

Keywords: Self-balancing robot, inverted pendulum, PD controller, sensor fusion, NI myRIO.

I. INTRODUCTION

Self-balancing robots have received a lot of attention in robotics and control engineering because of its nonlinear, unstable dynamics and practical applications in personal transporters, service robots, and assistive devices [1, 2]. A two-wheel self-balancing robot behaves like an inverted pendulum, with the center of mass above the wheel axis. Any minor disruption leads the robot to fall unless corrective action is taken immediately via feedback control.

Accurate sensing, quick computing, and durable actuation are all required to maintain equilibrium in such systems. Accelerometers and gyroscopes are standard inertial sensors for estimating a robot's tilt angle and angular velocity. These measurements are interpreted by a controller, which creates the necessary motor orders to restore equilibrium. While advanced control methods like Linear Quadratic Regulators (LQR), sliding-mode control, and intelligent controllers offer high robustness, they frequently increase system complexity and computing strain.

Simpler classical controllers continue to be appealing for real-world and instructional applications [3], [5]. By combining derivative action that forecasts system behaviour and dampens oscillations with proportional correction based on angle error, the Proportional–Derivative (PD) controller offers efficient stabilization. The design and real-time implementation of a PD-controlled two-wheel self-balancing robot are the main topics of this study, with a focus on experimental validation, simplicity, and dependability.

II. LITERATURE REVIEW

Two-wheel self-balancing robots have been stabilized in a number of studies by using feedback control techniques and modeling them as wheeled inverted pendulum systems. A thorough dynamic model of a two-wheeled self-balancing robot was published by Kumar et al. [1], who also showed that, with the right tuning, classical control techniques may produce dependable stabilization under minor perturbations. In order to achieve stable equilibrium, their work highlighted the significance of precise system modeling and sensor feedback.

Robust control algorithms for wheeled inverted pendulum robots operating under parametric uncertainties and external disturbances were examined by Mahmoud et al. [2]. Their findings demonstrated greater robustness when compared to traditional controllers; however, these methods are less appropriate for low-cost or educational platforms due to their higher computing complexity.

Pandey et al. [3] used inertial sensors and traditional control methods to create a self-balancing robot with extra line-following capacity. Their experimental findings demonstrated that, when paired with efficient sensor fusion and real-time implementation, basic controllers can nevertheless deliver respectable performance.

State estimation and sensor reliability have also been extensively researched. Park et al. [4] showed that precise tilt estimate has a major impact on stability margins and concentrated on improving balancing performance through enhanced sensing methods. These results validate the combination of gyroscope and accelerometer data using complimentary filtering approaches.

Castillo-Zamora et al. [5] provided controller performance comparisons and assessed sliding-mode, PD, and PID controllers for stability applications. According to their research, PD control provides an advantageous trade-off between resilience, performance, and ease of implementation, especially for real-time embedded systems.

According to the literature now in publication, PD control in conjunction with dependable sensor fusion appears to be a workable and effective option for two-wheel self-balancing robots, particularly in situations where simplicity, real-time performance, and ease of implementation are the main design requirements.

III. METHODOLOGY

Fig. 1 shows the proposed two-wheel self-balancing robot's control technique. System initialization is the first step in the process, during which the motor drivers, proportional-derivative (PD) controller parameters, and inertial measurement unit (IMU) are configured. This stage guarantees consistent system performance and accurate sensor calibration.

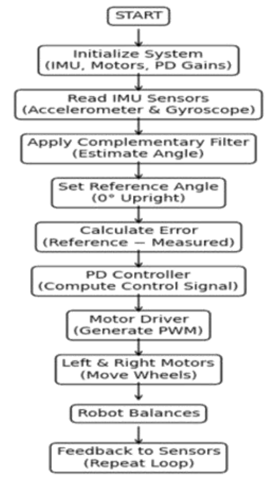

Figure 1: Flow Diagram

Accelerometer and gyroscope readings are among the real-time data that the controller continuously gathers from the IMU. A complementary filter is used to fuse the sensor outputs since raw sensor data contains noise and drift. To get a precise estimate of the robot's tilt angle, this filtering method combines the accelerometer's long-term stability with the gyroscope's quick dynamic responsiveness.

Zero degrees is the reference angle for the upright posture. The tilt error is computed as the discrepancy between the observed and reference angles. To produce a remedial control signal, the PD controller processes this error. The derivative term enhances system damping by taking the error's rate of change into account, while the proportional term reacts to the tilt error's amount.

Pulse width modulation (PWM) signals are produced from the control signal via the motor driver circuit. The robot can counteract disruptions and regain balance by using these signals to control the speed and direction of its left and right motors. Stable balancing performance and constant feedback are made possible by the closed-loop architecture of the complete system.

© FEB 2026 | IRE Journals | Volume 9 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I8-1714038

IV. SYSTEM MODELING AND CONTROL STRATEGY

A. Inverted Pendulum Model.

An inverted pendulum fixed on a moving platform can be used to represent the two-wheel self-balancing robot. Let θ represent the robot body's tilt angle in relation to the vertical. Regulating θ to zero is the controller's goal.

When higher-order nonlinearities are ignored, the linearized dynamics surrounding the upright position can be written as

$J\ddot{\theta}+b\theta-mgl=U$ (1)

where J is the moment of inertia, b is damping, m is the robot's mass, g is the gravitational acceleration, l is the distance to the center of mass, and u is the control input produced by the motors.

B. PD Control Law

The PD controller computes the control input based on the tilt angle error and its derivative [5], [6]:

$u(t)=Kp\times e(t)+Kd\times(de(t))/dt$ (2)

Where e(t)=θref−θ(t)e(t)=\θ_{ref} - \θ(t)e(t)=θref−θ(t), and Kp and Kd are the proportional and derivative gains, respectively. The proportional term provides restoring torque, while the derivative term improves damping and transient response.

V. SENSOR FUSION AND FEEDBACK IMPLEMENTATION

A. Accelerometer and Gyroscope Measurement

The gyroscope's measurement of angular velocity yields precise short-term motion data, but when integrated over time, it experiences drift. Although it is susceptible to noise and vibration, the accelerometer measures tilt angle with respect to gravity.

B. Complementary Filter

The Complementary filter is used to provide a trustworthy approximation of the tilt angle [4], [7]:

$\theta=\alpha(\theta+\dot{\theta}_{gyro}\Delta t)+(1-\alpha)\theta_{acc}$ (3)

where the filter coefficient is represented by α. This method combines the accelerometer's long-term stability with the gyroscope's short-term accuracy.

VI. HARDWARE AND SOFTWARE IMPLEMENTATION

A. Hardware Architecture

Two DC gear motors, an inertial measurement device, motor driver circuits, and an NI myRIO-1900 controller make up the robot. Real-time data collection, signal processing, and motor control are made possible via the myRIO platform.

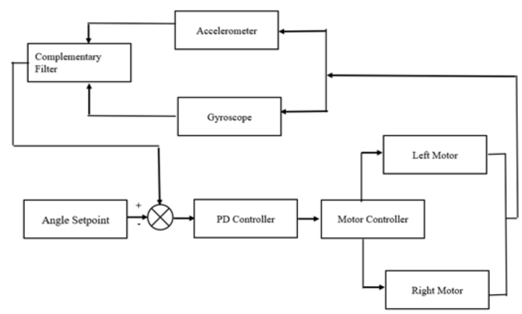

Figure 2: Block diagram representation of PD Control for Self-Balancing Robot

Equilibrium maintenance is made possible by the relationship between the gyro-accelerometer sensor fusion and the PD control algorithm. In order to properly balance the robot in the face of external disturbances or changes in its environment, this combination of sensors and control method demonstrates a robust self-balancing robot that can dynamically and autonomously modify its motor outputs. The adaptability

B. Software Implementation

The control algorithm is created using LabVIEW. MyRIO interfaces are used to collect sensor data, which is then filtered by the complementing filter and processed by the PD controller. The motors are powered by pulse-width modulation (PWM) signals that are generated from the controller output. Fast corrective motion is made possible by the duty cycle, which controls motor speed and direction.

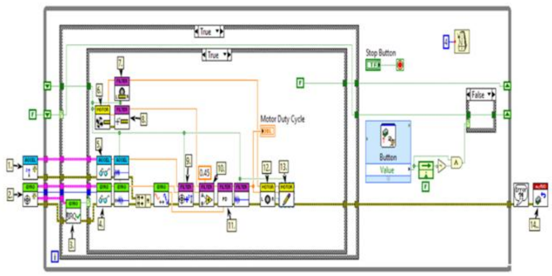

Figure 3: Self balancing robot code.
Source: https://link.springer.com/chapter/10.1007/978-981-97-6806-6_32.

The computed modifications are converted into Pulse Width Modulation (PWM) signals, which regulate the motors' rotating power supply. The motor's speed is determined by the duty cycle of the PWM signals, which enables real-time dynamic modifications. Furthermore, using devices like H-bridge motor drivers, the direction of the PWM signal can be changed to regulate the direction of motor rotation. The self-balancing robot can traverse and maintain stability in a variety of surroundings thanks to its complex control system, which combines sensor fusion, PID control, erientation offset correction, and PWM signal modulation to continuously analyze and adjust its status. This all-encompassing strategy guarantees the robot's ability to react to both static and dynamic changes, resulting in a reliable and well-rounded performance.

VII. EXPERIMENTAL RESULTS AND DISCUSSION

In accordance with validation procedures frequently documented in experimental balancing platforms, experimental testing were carried out by manually perturbing the robot and monitoring its response [1], [3]. The robot swiftly and with little overshoot returned to the upright posture with appropriately adjusted PD gains. Oscillations were induced by excessive proportional gain, and delayed response was caused by insufficient derivative gain. The findings show that the PD controller offers dependable real-time performance and steady balancing at a reasonable computational cost.

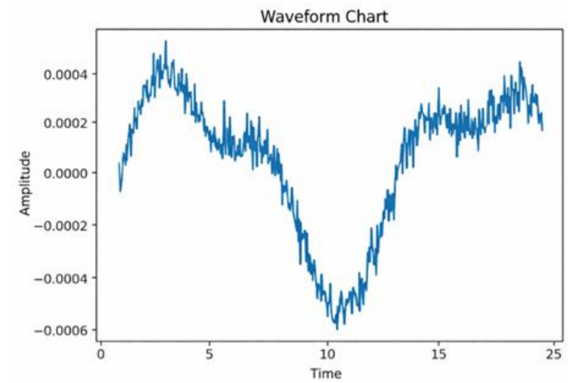

Figure 4: Gyro sensor output

The figure 4 shows self-balancing robot equipped with a proportional-derivative (PD) controller, the gyroscope sensor plays a crucial role in maintaining balance. The gyroscope measures the rate of angular displacement, providing real-time data on the robot's tilt or inclination. The graph output of the gyroscope sensor reflects the angular velocity or rate of change in the robot's orientation over time.

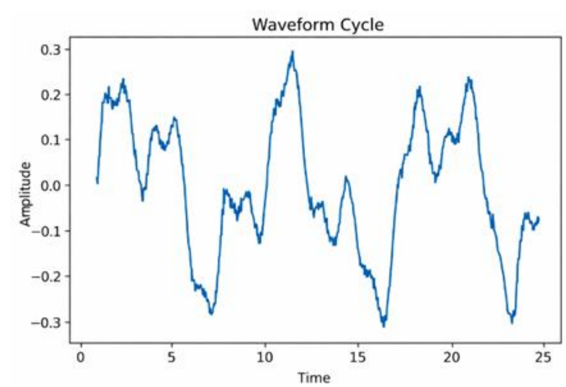

Figure 5: motor duty cycle output

The figure 5 shows duty cycle of the motor is a crucial factor that directly influences the robot's balance. The duty cycle represents the percentage of time the motor is active within a given time period. In the PD controller, the proportional component reacts to the current error between the desired and tilt angles of the robot, while the derivative component considers the rate of change of this error

Table 1: The values of Self balancing robot

Parameter	Value
Gyro offset value	91
Proportional value	6.0002
Derivative value	0.132

Measured orientation	39.12
Measured position	12.4
Orientation adjustmment	1.381
Left motor value	0.27
Right motor value	0.27
Current	0.625 mA
Voltage	4.8 V

VIII. CONCLUSION AND FUTURE WORK

In this paper, a two-wheel self-balancing robot with a PD controller was designed and implemented. Experimental findings verify that, when paired with efficient sensor fusion, PD control is adequate to stabilize the inverted pendulum system. Rapid prototyping and instructional platforms can benefit from the approach's simplicity and dependability. Trajectory tracking, autonomous navigation, and the use of sophisticated control techniques like adaptive, optimum, or disturbance-rejection-based control approaches are possible future projects.

REFERENCES

[1] V. Kumar et al., "Design, Dynamic Modelling, and Control of a Two-Wheeled Self-Balancing Robot," IEEE Access, 2022.

[2] M. S. Mahmoud et al., "Robust Control Design of Wheeled Inverted Pendulum Assistant Robot," ISA Transactions, 2017.

[3] D. Pandey et al., "Self-Balancing Robot on Two Wheels with Line Following Capability," International Journal of Engineering Research, 2021.

[4] J. H. Park et al., "Development of a Self-Balancing Robot with a Control Moment Gyroscope," Robotics and Autonomous Systems, 2018.

[5] J. J. Castillo-Zamora et al., "Comparison of PD, PID and Sliding-Mode Controllers for Stability," IEEE Access, 2018.

[6] Maheshan C M, et al., " Feed-forward neural network approach in navigation control and PI speed control of mobile robot," International journal of electrical and electronics -2016.

[7] Maheshan C .M, et al., " Line Tracking Robot Using Labview," IJSART ,2018.

[8] Maheshan C .M ,et al., " Feed-forward neural network approach in navigation control and PI speed control of mobile robot," International journal of electrical and electronics research 4 (3), 95-101,2015.

[9] Harsha Govindray Vernekar and C. M. Maheshan, " Flow rate Controlled Water Sprinkler using myRIO controller and LabVIEW ," Journal of Electrical Engineering and Automation 2022.

[10] Durgesh Pandey et al., "Self balancing robot on two wheels with line following capability," International Research Journal of Modernization in Engineering Technology and Science, January 2021.

[11] Muhammad Junaid Rabbani et al., "Trajectory Tracking and Stabilization of Nonholonomic Wheeled Mobile Robot Using Recursive Integral Backstepping Control," https: //www.mdpi.com/journal/electronics, July 2021.

[l2] Vinod Kumar et al., "Design, Dynamic Modelling, and Control of a Two-wheeled Self-Balancing Robot (TWSBR). " International Journal of Mechanical Engineering, August 2022.

[13] F. Grasser, A. D et al., "JOE: A mobile, inverted pendulum," IEEE Trans. Ind. Electron., vol. 49, no. 1, p.107-114. Feb. 2002.

[14] Nguyen Gia Minh Thao et al., "A PID backstepping controller for twowheeled self-balancing International Forum robot," 10.1109/IFOST.2010.5668001. on Strategic Technology 2010, Ulsan, 2010, pp. 76-81.

[15] W. An and Y. Li, "Simulation and control of a two wheeled self-balancing International Conference robot," 2013 IEEE on Robotics and Biomimetics (ROBIO), Shenzhen, 2013, pp. 456-461.