



Self-Balancing Robot Control with Mobile App Using PID Algorithm and Bluetooth Communication

Purvi Laad¹, Sakshi Yadav², Rishu Rai³, Vaibhav Ahirwar⁴, Neenansha Jain⁵, Neelima Singh Baghel⁶
^{1,2,3,4,5,6} Department of Electronics & Communication Engineering (ECE)
^{1,2,3,4,5,6} Oriental College of Technology, Bhopal MP, India

Abstract— This research paper presents the design and implementation of a two-wheeled self-balancing robot controlled via a smartphone mobile application using Bluetooth wireless communication. The robot employs an inverted pendulum model and maintains its upright position through a PID (Proportional-Integral-Derivative) control algorithm. The tilt angle is measured by a GY-521 module (MPU6050) which integrates a 3-axis accelerometer and 3-axis gyroscope. Sensor fusion is applied through a complementary filter to obtain a reliable and accurate tilt estimate. Two NEMA 17 stepper motors, driven by A4988 motor driver ICs, actuate the wheels based on PID output. The HC-05 Bluetooth module enables real-time wireless command transmission from an Android mobile application to the microcontroller. The system successfully demonstrates stable balancing along with directional control (forward, backward, left, right) without losing equilibrium. This work integrates core concepts of embedded systems, control theory, sensor interfacing, and wireless communication, making it a comprehensive platform for robotics education and research.

Keywords— Self- Balancing Robot, PID Controller, Stepper Motor, Mobile App Control.

I. INTRODUCTION

This Robotics has emerged as a key technological domain with applications in industrial automation, healthcare, transportation, and smart systems. Among various robotic systems, self-balancing robots represent a challenging and interesting application due to their reliance on dynamic stability control. These robots operate based on the inverted pendulum principle, where stability is maintained by continuously adjusting the wheel motion.

A self-balancing robot integrates sensors, microcontrollers, and control algorithms to detect tilt and apply corrective actions. The PID controller is widely used due to its simplicity, reliability, and ease of implementation in real-time systems. With the advancement of wireless communication technologies, mobile-based control has become increasingly popular, allowing users to remotely control robotic systems through smartphones.

This work focuses on the development of a mobile-controlled self-balancing robot, combining control systems, embedded design, and wireless communication to achieve stable and user-friendly operation.

II. LITERATURE REVIEW

Extensive research has been carried out in the domain of self-balancing robots and control systems, focusing on improving stability, accuracy, and real-time responsiveness. Among various control strategies, the Proportional-Integral-Derivative (PID) controller has been extensively adopted due to its simple structure, low computational complexity, and effective performance in nonlinear dynamic systems. Several studies have demonstrated that properly tuned PID controllers can achieve stable balancing even under external disturbances and parameter variations, particularly when implemented with high-frequency feedback loops and optimized using classical tuning methods such as Ziegler-Nichols and trial-and-error approaches [1-3].

The self-balancing robot is commonly modeled as an inverted pendulum system, which represents an inherently unstable nonlinear system. The mathematical modeling of such systems is typically derived using Newton-Euler equations or Lagrangian dynamics, leading to linearized state-space representations around the equilibrium point. These models are essential for controller design and stability analysis. Researchers frequently utilize simulation platforms such as MATLAB/Simulink to validate system dynamics, optimize control parameters, and predict system

behavior prior to hardware implementation, thereby reducing development cost and time [4-6].

Accurate estimation of the tilt angle is critical for maintaining balance. In this context, sensor fusion techniques have been widely investigated to combine data from accelerometers and gyroscopes. Individual sensors suffer from limitations such as noise, bias, and drift; hence, fusion algorithms are required to improve reliability. Complementary filters are commonly used due to their computational efficiency and ease of implementation in embedded systems, where they effectively combine high-frequency gyroscope data with low-frequency accelerometer signals. On the other hand, Kalman filter-based approaches provide optimal state estimation and superior accuracy, although they require higher computational resources, making them more suitable for advanced systems [7-9]

Wireless communication has become an integral part of modern robotic systems. Bluetooth-based control, particularly using modules such as HC-05, has been widely implemented due to its simplicity, low power consumption, and ease of interfacing with microcontrollers. Studies indicate that Bluetooth Serial Port Profile (SPP) enables reliable real-time communication between mobile devices and embedded systems within a limited range, typically up to 10 meters. Mobile applications developed using platforms such as Android Studio or MIT App Inventor allow intuitive user interfaces for controlling robot motion, including directional and speed commands [10-12].

In addition to classical PID control, several advanced control techniques have been explored to enhance system robustness and performance. These include fuzzy logic controllers, Linear Quadratic Regulators (LQR), adaptive control methods, and neural network-based approaches. Such techniques are capable of handling system nonlinearities and uncertainties more effectively, offering improved disturbance rejection and adaptive behavior. However, their implementation complexity, higher computational requirements, and need for precise system modeling limit their practical use in low-cost and resource-constrained systems. Consequently, PID controllers remain the most widely preferred solution for educational, prototyping, and embedded robotic applications [13-15].

III. METHODOLOGY

The proposed system for the mobile-controlled self-balancing robot is developed through a systematic sequence of design, integration, and control steps. The overall methodology is described as follows:

A. System Design and Architecture

The system consists of four major subsystems: sensing unit, control unit, actuation unit, and communication unit. The sensing unit acquires real-time tilt data, which is processed by the control unit to generate corrective signals. These signals are applied to the actuation unit (motors) to maintain balance, while the communication unit enables user interaction through a mobile application.

B. Sensor Data Acquisition

The tilt angle and angular velocity of the robot are measured using the MPU6050 sensor module. The accelerometer provides linear acceleration data, whereas the gyroscope measures angular motion. These signals are continuously transmitted to the microcontroller using the I²C communication protocol.

Table-2 Details of Component

Subsystem	Components	Function
Sensing	GY-521 (MPU6050)	Measures tilt angle and angular velocity via I2C
Power Supply	12V Li-Po Battery	Powers motors, drivers, and logic circuits
Processing	Microcontroller (Arduino)	Runs PID algorithm, sensor fusion, Bluetooth parsing
Actuation	NEMA 17 + A4988	Drives wheels with precise step control
Communication	HC-05 Bluetooth	Receives remote commands from mobile app
User Interface	Android App	Sends directional commands to robot wirelessly
Mechanics	Frame, Wheels, Brackets	Provides structural support and mobility

C. Sensor Fusion and Angle Estimation

Raw sensor outputs are affected by noise and drift; therefore, a sensor fusion technique is implemented. A complementary filter is used to combine accelerometer and gyroscope data, producing an accurate and stable estimate of the tilt angle. This estimated angle serves as the primary feedback parameter for control.

D. Error Calculation

The system aims to maintain the robot in a vertical (upright) position. The desired angle is set to zero degrees. The error signal is calculated as the difference between the desired angle and the measured tilt angle:

$$\text{Error} = \theta_{\text{desired}} - \theta_{\text{measured}}$$

This error determines the corrective action required.

E. PID Control Implementation

A Proportional–Integral–Derivative (PID) controller is implemented to minimize the error and stabilize the system.

- The **Proportional (P)** term responds to the present error
- The **Integral (I)** term compensates accumulated past errors
- The **Derivative (D)** term predicts future error trends

The combined PID output generates control signals for motor actuation.

F. Motor Driver and Actuation

The PID output is transmitted to the A4988 stepper motor driver in the form of STEP and DIR signals. The driver regulates current and controls the rotation of NEMA 17 stepper motors. The motors adjust wheel movement to counteract the tilt, thereby maintaining balance.

G. Wireless Communication and Control

An HC-05 Bluetooth module is used for wireless communication between the robot and a mobile application. The user sends commands such as forward, backward, left, and right through the app. These commands are received by the microcontroller via UART communication and are integrated with the balancing control algorithm.

H. Real-Time System Operation

The system operates in a continuous feedback loop:

1. Sensor data acquisition
2. Angle estimation
3. Error computation
4. PID control action
5. Motor adjustment

Simultaneously, user commands are processed without disturbing the balancing mechanism, ensuring stable and responsive robot operation.

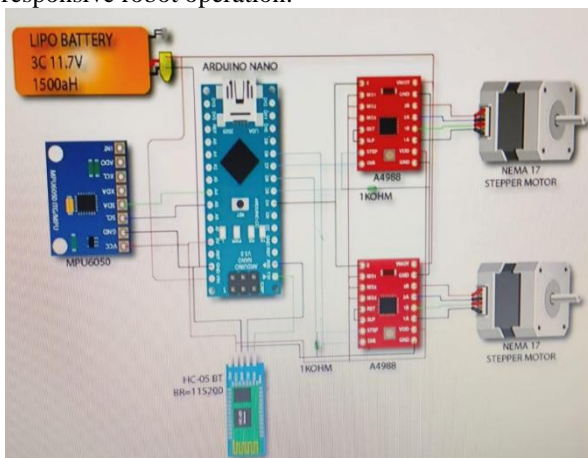


Fig:1 Block Diagram

IV. RESULTS AND DISCUSSION

The developed self-balancing robot system was implemented using an Arduino-compatible microcontroller operating at a control loop frequency of approximately 100 Hz. The modular firmware architecture ensured efficient execution of sensing, filtering, control, and actuation processes in real time.

A. System Performance Evaluation

The integration of the MPU6050 sensor with I²C communication enabled continuous acquisition of tilt and angular velocity data. The complementary filter effectively fused accelerometer and gyroscope signals, producing a stable and noise-reduced tilt angle. The filter coefficient ($\alpha \approx 0.98$) provided an optimal balance between short-term responsiveness and long-term stability.

The PID controller demonstrated satisfactory performance in maintaining the upright position of the robot. The proportional term ensured rapid correction of tilt deviations, while the integral term minimized steady-state error. The derivative component effectively reduced oscillations, resulting in smooth stabilization. The controller output was directly mapped to the motor stepping frequency, allowing precise adjustment of motor speed and direction.

B. Motor Control Analysis

The use of A4988 stepper motor drivers enabled accurate control of NEMA 17 motors through STEP and DIR signals. Motor speed was regulated by varying the pulse delay, which was dynamically adjusted based on PID output. Microstepping (1/8 step mode) significantly reduced mechanical vibrations and improved motion smoothness, contributing to enhanced balancing performance.

C. Wireless Control Performance

The Bluetooth-based communication system provided reliable real-time interaction between the robot and the mobile application. The use of simple ASCII command transmission ensured low latency and efficient command parsing. The system successfully executed directional commands (forward, backward, left, right, and stop) while maintaining balance, demonstrating effective integration of user input with the control algorithm.

D. System Stability and Limitations

Experimental observations indicate that the robot maintains stable operation under small disturbances and moderate external inputs. However, system performance is sensitive to PID tuning parameters and sensor calibration. Excessive gain values may lead to oscillations, while improper filtering may introduce noise in angle estimation. Despite these limitations, the system achieves reliable balancing on flat surfaces with consistent response characteristics.

V. Conclusion

This study presented the design and implementation of a mobile-controlled self-balancing robot using a PID-based control strategy. The system successfully integrates sensor fusion, real-time control, stepper motor actuation, and wireless communication within a compact embedded framework.

The complementary filter provided accurate tilt estimation, while the PID controller ensured effective stabilization of the inherently unstable inverted pendulum system. The use of A4988 drivers and microstepping improved motor performance, resulting in smooth and precise motion control. Furthermore, the Bluetooth-enabled mobile interface allowed intuitive and flexible user interaction without compromising system stability.

Overall, the proposed system demonstrates that a cost-effective and computationally efficient control approach can achieve reliable self-balancing performance. The developed platform serves as a practical foundation for further research in advanced control techniques, autonomous robotics, and intelligent mobility systems.

References

- [1]. K. J. Åström and T. Hägglund, *PID Controllers: Theory, Design, and Tuning*, 2nd ed., ISA, 1995.
- [2]. N. S. Nise, *Control Systems Engineering*, 7th ed., Wiley, 2015.
- [3]. J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, vol. 64, pp. 759–768, 1942.
- [4]. H. K. Khalil, *Nonlinear Systems*, 3rd ed., Prentice Hall, 2002.

- [5]. B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, Springer, 2016.
- [6]. MathWorks, "MATLAB/Simulink User Guide," 2020.
- [7]. S. Colton, "The Balance Filter: A Simple Solution for Integrating Accelerometer and Gyroscope Measurements," MIT, 2007.
- [8]. R. G. Valenti, I. Dryanovski, and J. Xiao, "A Linear Kalman Filter for MARG Orientation Estimation," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 9, pp. 2033–2044, 2016.
- [9]. G. Welch and G. Bishop, "An Introduction to the Kalman Filter," UNC Chapel Hill, 2006.
- [10]. Bluetooth SIG, "Bluetooth Core Specification v5.0," 2019.
- [11]. M. Banzi and M. Shiloh, *Getting Started with Arduino*, Maker Media, 2014.
- [12]. MIT, "App Inventor Documentation," 2021.
- [13]. L. A. Zadeh, "Fuzzy Logic," *IEEE Trans. Information Theory*, vol. 11, no. 3, pp. 338–353, 1965.
- [14]. D. E. Kirk, *Optimal Control Theory: An Introduction*, Prentice Hall, 1970.
- [15]. S. Haykin, *Neural Networks and Learning Machines*, 3rd ed., Pearson, 2009.

