2025; 1-2 | pp.72-105

International Journal of Advances in Engineering and Computer Science

ISSN: 3121-6382

Volume 1, Issue 2, October-December, 2025
(An Academians Publishers of Research & Academic Resources)

Review Article

Received: 20-09-2025 Accepted: 28-10-2025 Published: 17-11-2025

Design of a Robust Self-Balancing Robot With Optimized Proportional Integral Derivative (P-I-D) Control for Rough Terrain Movement

Tekenah Daniel Bestmann
Faculty of Engineering, Department of Mechanical and Mechatronic Engineering,
Federal University Otueke, Bayelsa State, Nigeria

Abstract: This project presents the design and implementation of a robust two-wheeled self-balancing robot equipped with an Adaptive Proportional-Integral-Derivative (PID) control system optimized for rough terrain navigation. Traditional PID controllers are often insufficient in unpredictable environments due to their fixed parameters. To address this, the robot integrates real-time sensor feedback from an MPU-6050 accelerometer and gyroscope to dynamically tune PID gains, ensuring continuous balance and stability. Powered by an Arduino Nano V3.0 and geared DC motors, the system is further enhanced with obstacle avoidance features using ultrasonic and infrared sensors. The project encompasses mechanical design, electronic integration, software development, and testing across various terrains such as grass, sand, and gravel. The results for this report validate the system's ability to adapt to changing conditions, maintain stability, and avoid obstacles efficiently. This work contributes to the development of intelligent systems for real-world applications, including warehouse logistics and assistive mobility, by improving robustness, manoeuvrability, and terrain adaptability.

Keywords: *Self-Balancing Robot, PID Control, Rough Terrain Navigation, Adaptive Control System, Sensor Fusion, Obstacle Avoidance, Autonomous Robotics.*

Introduction

Self-balancing robots are innovative devices based on the "inverted pendulum concept", designed to maintain stability using two wheels (Malpani et al., 2021). The inverted pendulum is a classic problem in control theory and robotics, with applications dating back to Galileo's experiments in 1602 (Boubaker, 2017). It serves as a fundamental system for testing control algorithms and has numerous practical applications (Matesica et al., 2016). The concept can be implemented in self-balancing robots, which are mathematically modelled using equilibrium equations and controlled with proportional-integral-derivative (PID) controllers (Sedlar & Bošnjak, 2023).

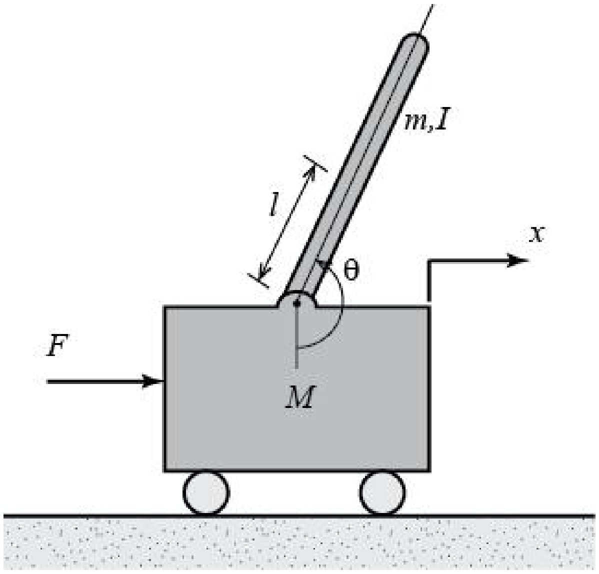

Figure 1.1 A Schematic Diagram of an Inverted Pendulum (Control Tutorials for

Corresponding Author: Tekenah Daniel Bestmann

MATLAB and Simulink - Inverted Pendulum: System Modelling, 2025)

These robots utilize sensors like accelerometers and gyroscopes to determine their position in three-dimensional space, with microcontrollers processing this data to control wheel rotation for balance (Sujiwa & Suhadata, 2023). The control mechanisms often employ proportional-integral-derivative (PID) algorithms and energy-shaping techniques to achieve stability (Malpani et al., 2021; Gandarilla et al., 2018). These characteristics make it suitable for various applications, such as industrial tasks, spy missions, and delivering supplies in crisis situations or mountainous regions (Malpani et al., 2021). Recent developments include long-range control capabilities using technologies like NRF24L01 (Sujiwa & Suhadata, 2023). NRF24L01 is a high-speed wireless RF transceiver operating in the 2.4GHz ISM band, offering ultra-low power consumption and data rates up to 2Mbps (Saha et al., 2017). It is widely used in short-range wireless applications, including remote-controlled appliances, environmental data acquisition systems, and wireless sensor networks (Vaz et al., 2024). Performance analysis of NRF24L01 modules has shown that their range and operation can vary under different environmental conditions and parameter settings (Vaz et al., 2024). When compared to XBee ZB modules in wireless ad-hoc networks, NRF24L01+ modules demonstrate competitive performance in terms of throughput, mesh routing recovery time, and power consumption (Saha et al., 2017). The technology has been successfully implemented in novel applications, such as autonomous rovers for delivery and surveillance, where it facilitates communication between the rover and base station (Kulasekara et al., 2019).

The compact design, agility, and dynamic stability of these self-balancing robots make them an essential focus of modern robotics research. However, navigating rough terrain presents significant challenges for self-balancing robots. Unlike smooth surfaces where balance control is relatively straightforward, uneven or deformable surfaces introduce external disturbances, sudden inclines, and unpredictable friction variations. To address these challenges, Adaptive proportional-integral-derivative (PID) Control plays a crucial role in enhancing the robot’s stability and adaptability. Unlike traditional controllers struggle with steep slopes and speed bumps, potentially causing security issues (Zheng et al., 2017). To address this, researchers have developed advanced control strategies. Zheng et al. (2017) proposed a hierarchical fast terminal sliding mode approach for two-wheeled robots, while Tsai et al. (2022) introduced an adaptive motion control system for leg-wheeled robots, incorporating backstepping sliding-mode control and impedance control. For more complex environments, Atas et al. (2022) developed a surfel-based navigation method using raw point cloud maps, enabling efficient sampling-based planners for challenging terrains. Li et al. (2021) presented a combined approach of pose optimization and force control via quadratic programming for wheel-legged robots, allowing them to navigate stairs and ramps while maintaining balance and wheel traction. These advancements demonstrate significant progress in improving self-balancing robots' performance on rough surfaces.

Statement of Problem

Self-balancing robots depend on precise control systems to stay upright (Chen et al., 2023; Manolescu & Secco, 2023), but their performance significantly deteriorates on rough terrain. While they operate reliably on flat surfaces, real-world environments introduce challenges such as uneven ground, inclines, and external disturbances like rocks or debris—factors that disrupt balance and reduce control effectiveness (Tsai et al., 2022; Zheng et al., 2017). Existing studies have attempted to address these issues using advanced control strategies; however, most solutions fall short. For example, Tsai et al. (2022) proposed adaptive controls for leg-wheeled robots, but only tested them on flat terrain using simulations. Li et al. (2022) used a dual-loop observer-based method but in virtual settings and without integrating PID systems. Kim & Lee (2018) applied adaptive control to improve traction, yet focused solely on four-wheeled robots. Cendana et al. (2020) tested a two-wheeled robot on uneven surfaces but relied on fixed PID parameters, limiting adaptability. The core gap remains: current designs lack a real-world tested, two-wheeled self-balancing robot equipped with an adaptive PID control system that can dynamically respond to rough terrain conditions (e.g., sand, stones, and grass). This study seeks to bridge that gap.

Aims and Objectives

Aim:

This project aims to design a self-balancing robot with an Adaptive proportional-integral-derivative (PID) Control system to enhance stability and manoeuvrability over rough terrain.

Objective:

i. To design a self-balancing robot capable of maintaining stability using sensor-based feedback and motor control mechanisms.
ii. To design an adaptive control mechanism that dynamically adjusts proportional-integral-derivative (PID) parameters based on real-time terrain conditions for improved balance and manoeuvrability.
iii. To evaluate the robot's performance in real-world terrain conditions, assessing its robustness, stability, and control efficiency under varying environmental challenges.

Justification

In modern warehouse environments, especially in e-commerce and manufacturing, efficient and stable movement of goods across narrow, cluttered, or uneven surfaces remains a persistent challenge (Gomes et al., 2023). As highlighted earlier, conventional self-balancing robots struggle to adapt to these dynamic terrain conditions, leading to instability, downtime, and inefficiencies in material handling. This project addresses that gap by designing a two-wheeled self-balancing robot with an adaptive PID control system tailored for indoor warehouse logistics. By dynamically adjusting to surface irregularities and avoiding obstacles, the proposed system aims to enhance stability, and improve operational efficiency in warehouse workflows. This solution offers a scalable and intelligent alternative for safer and more reliable goods movement in space-constrained, rough-surfaced industrial settings.

Scope

This project focuses on designing a self-balancing two-wheeled robot using an adaptive PID control system to maintain stability and manoeuvrability on rough terrain. The system is built around the Arduino Nano V3.0 microcontroller, which processes real-time data from the MPU-6050 accelerometer and gyroscope for balance control. Motion is achieved through geared DC motors controlled by an L298N motor driver, while SG90 servo motors support additional adjustments. Obstacle detection is enabled via an HC-SR05 ultrasonic sensor and an IR proximity sensor. Power is provided by 18650 lithium-ion batteries, with switches for manual control. All components are integrated using an expansion board and jumper wires, and mounted on a compact, lightweight frame. The robot will undergo real-world testing to evaluate its performance in terms of stability, adaptability, and efficiency.

Literature Review

Overview of the Literature Review

This literature review critically examines self-balancing robots, emphasizing their control systems, theoretical foundations, and real-world applications, tracing their evolution from early models like the Segway to modern autonomous systems that incorporate advanced sensors and machine learning algorithms. It discusses the mathematical modelling of these robots using both Newton-Euler and Lagrangian methods to analyse their dynamic behaviour and highlights the limitations of traditional Proportional-Integral-Derivative (PID) controllers when navigating uneven terrain. The review underscores the need for adaptive control mechanisms that can adjust PID parameters in real-time and evaluates adaptive tuning methods such as Fuzzy Logic, Artificial Neural Networks, Model Reference Adaptive Control (MRAC), and Genetic Algorithms. It also compares Adaptive PID Control with other advanced strategies, including Sliding Mode Control (SMC), Model Predictive Control (MPC), and Reinforcement Learning-based approaches, noting that while these offer greater robustness and adaptability, they often introduce computational challenges that hinder real-time application. The review identifies a gap in empirical research on self-balancing robots using Adaptive PID Control in unstructured, rough environments, reinforcing the need for developing more stable and adaptable terrain-navigating robots.

Self-Balancing Robots: Concepts, Evolution and Applications

Self-balancing robots are two-wheeled mobile platforms designed to maintain an upright position during motion, despite their inherent instability (Bhagat, 2018). Their development gained traction in the late 20th century due to advances in sensor technology and embedded computing. These robots rely on gyroscopes and accelerometers to detect orientation, enabling them to balance using feedback control algorithms like the Proportional-Integral-Derivative (PID) controller (Aubakir et al., 2015; Sondhia et al., 2017; Han et al., 2014). The system operates on the inverted pendulum principle, which requires continuous adjustment to maintain equilibrium (Sondhia et al., 2017). Enhancements in Micro-Electro-Mechanical Systems (MEMS) sensors and sensor fusion have further improved control precision (Han et al., 2014).

A significant milestone was the launch of the Segway Personal Transporter by Dean Kamen in 2001, which used gyroscopes and accelerometers for balance without rider input (Vadak et al., 2021; Kumar, 2024). Its nonlinear dynamics were modeled using nonholonomic mechanics, with stability maintained through Kalman filtering, pole placement, and Linear Quadratic Regulator (LQR) control (Haddout, 2018; Fahmi et al., 2020). The Segway became a popular urban transport option and influenced further research into self-balancing systems (Kumar, 2024). Academic research in the early 2000s advanced self-balancing robot capabilities from basic navigation to obstacle avoidance and environmental monitoring using adaptive control and machine vision (Kung, 2017; Romlay et al., 2019). Recently, machine learning, neural networks, and reinforcement learning—integrated with Light Detection and Ranging (LiDAR), Global Positioning System (GPS), and ultrasonic sensors—have enhanced navigation and autonomy (Mithil et al., 2017; Babu et al., 2024; Tiwari et al., 2025). These innovations improve performance in complex environments, enabling deployment in hazardous or remote locations.

Applications in Various Industries

Self-balancing robots have found extensive applications across various industries, particularly in transportation and personal mobility. The introduction of e-powered micro personal mobility vehicles (e-PMVs) like e-scooters and self-balancing vehicles has significantly impacted urban transportation (Boglietti et al., 2021). These devices offer an energy-efficient alternative to traditional vehicles, potentially transforming urban transport systems (Gössling, 2020). These systems rely on real-time sensor feedback and dynamic stability control, allowing users to navigate effortlessly through crowded spaces. Additionally, recent advancements in autonomous navigation are enabling the integration of self-balancing robotic platforms into public transportation systems and smart cities, enhancing mobility solutions. Smart robotic wheelchairs utilize sensors, machine learning, and computer vision for autonomous navigation and obstacle avoidance, improving mobility for individuals with impairments (Sahoo & Choudhury, 2024). The Scube concept introduces an autonomous, self-balancing mobility device designed to address the first/last mile problem in public transportation, featuring electric propulsion and 3D camera-based navigation (Klöppel et al., 2018). Smart mobility technologies leverage real-time data to optimize urban infrastructure and improve public transport efficiency (Olaverri-Monreal, 2016).

Beyond transportation, self-balancing robots are making significant contributions to industrial automation and warehouse logistics. These robots employ sophisticated navigation systems, combining indoor positioning technologies and obstacle avoidance algorithms for efficient warehouse floor navigation (Sbirna & Sbirna, 2022). Self-balancing robots, based on the inverted pendulum concept, offer significant advantages for automated material handling and inventory management in warehouses (Patil, 2021; Keote et al., 2024). These robots utilize sensors like accelerometers and gyroscopes to maintain balance while manoeuvring through confined spaces, enabling sharp turns and navigation in tight areas (Patil, 2021). In high-demand industries such as e-commerce and manufacturing, self-balancing robots help optimize supply chain operations by reducing human intervention and increasing efficiency in logistics workflows.

The medical and security sectors have also embraced self-balancing robots for healthcare assistance and surveillance applications. In healthcare, robotic wheelchairs and mobility aids leverage self-balancing mechanisms to offer improved stability and flexibility for individuals with mobility impairments. These wheelchairs incorporate AI-powered features such as gesture control (Abiraj et al., 2024; Mahdin et al., 2022), voice recognition, and facial recognition (Abiraj et al., 2024). They also provide autonomous navigation capabilities, obstacle avoidance, and real-time health monitoring through IoT integration (Sonekar et al., 2024; Sahoo & Choudhury, 2023). GPS tracking enhances safety and caregiver support (Sonekar et al., 2024; Mahdin et al., 2022). Deep learning algorithms enable advanced functionalities like lane detection and object recognition for collision avoidance (Abiraj et al., 2024; Mahdin et al., 2022). Some models feature gyroscopic stabilization for improved balance (Mahdin et al., 2022). These innovations address the needs of users with varying levels of mobility impairment, including those with muscular dystrophy and partial paralysis (Mahdin et al., 2022). Smart robotic wheelchairs represent a significant advancement in assistive technology, enhancing users' quality of life and independence (Sahoo & Choudhury, 2023).

Theoretical Background of Self-Balancing Robots

Mathematical Modelling of Self-Balancing Robots

The mathematical modelling of self-balancing robots is based on the inverted pendulum system dynamics, where the robot maintains balance by continuously adjusting its position in response to external disturbances. A self-balancing robot consists of a pendulum-like body mounted on a wheeled platform, making it an inherently unstable system that requires active control. The key parameters influencing the robot's motion include the mass of the pendulum (M_p) and wheels (M_w), the moment of inertia of the pendulum (I_p) and wheels (I_w), the gravitational acceleration (g), the tilt angle (θ), and the position of the wheel base (x). Since the system must continuously correct deviations from the vertical position, a dynamic model is necessary to describe the relationship between the tilt angle and the base movement. This can be achieved using either Newton-Euler equations or Lagrange's equation, both of which provide mathematical formulations for the forces and torques acting on the system. The Newton-Euler equations are fundamental in robotics and mechanics for modelling dynamic systems. They have been applied to various robotic designs, including tendon-actuated snake robots (Antuono et al., 2023) and humanoid robots walking on inclined surfaces (Gießler & Waltersberger, 2023). These equations enable accurate modelling of forces, moments, and inertias in complex systems. While traditionally solved through iterative or algebraic methods, neural network approaches have been explored for solving Newton-Euler mechanics problems, particularly for larger systems (Ghoshal, 2022). The Newton-Euler approach, which is based on Newton's second law (F = ma) and its rotational counterpart (τ = Iα), provides a force-based description of the system's dynamics. The translational motion of the base is governed by the equation

$$(m_p \quad + m_w \quad)\ddot{x} + m_p \quad l\ddot{\theta} cos\theta - mp \quad l\dot{\theta}^2 sin\theta =$$

where F represents the force applied by the wheels to maintain balance. Meanwhile, the rotational motion of the pendulum is described as;

$$I_p\ddot{\theta} \quad + m_p \quad l\ddot{x} cos\theta - m_p \quad glsin\theta = 0$$

These equations highlight the coupled nature of the system: changes in the base position directly influence the pendulum's angular movement, and vice versa. As a result, a closed-loop control strategy is required to continuously regulate both variables and maintain stability. An alternative way to derive the governing equations is through Lagrange's equation, which provides an energy-based approach to modelling. Lagrange's equation is a fundamental tool in analytical mechanics for solving complex physical system dynamics (Damayanti et al., 2024). It offers a coordinate-independent formulation of motion equations, derived from Newton's laws. The equation's versatility extends to various technological applications, including optimizing energy flow in smart grids and enhancing photovoltaic systems (Damayanti et al., 2024). The Lagrangian formulation is expressed as

$$\frac{d}{dt}\left(\frac{\partial L}{\partial q_i}\right) - \frac{\partial L}{\partial q_i} \quad = Q_i$$

where L= K − P represents the Lagrangian function, defined as the difference between kinetic (K) and potential (P) energy. By selecting generalized coordinates (x for the wheel position and θ for the tilt angle), the Lagrange method provides a more systematic way to derive the motion equations while reducing the need to explicitly consider reaction forces. This approach is particularly useful for designing control algorithms since it allows for straightforward modifications when additional system components, such as external disturbances or varying loads, need to be accounted for. For control implementation, the state-space representation is used to express the system in a compact, matrix-based form that facilitates the design of advanced controllers. The general state-space model is given by

$$\dot{X} = AX + BU$$

$$Y = CX + DU$$

where $X = [x, \dot{x}, \theta, \dot{\theta}]^T$ represents the system state vector, consisting of position, velocity, tilt angle, and angular velocity. The input *U* represents the control force or torque applied by the motors, while *A, B, C, D* are matrices that define the system's behaviour. By linearizing the equations around θ ≈ 0 (assuming small angular deviations), the model simplifies to a form that can be efficiently controlled using techniques like proportional-integral-derivative (PID) controllers, Linear Quadratic Regulators (LQR), or Adaptive Control strategies.

Key Performance Metrics in Self-Balancing Systems

Key performance metrics are critical in evaluating the efficiency and reliability of self-balancing robotic systems.

- **Stability Margin**: The stability margin is a crucial concept in control systems, indicating the robustness of closed-loop stability to uncertainties (Wang et al., 2019). It plays a vital role in self-balancing systems, such as two-wheeled robots that utilize the inverted pendulum principle (Gul & Rahiman, 2021). Unbalanced stability margins can limit system performance, particularly in three-phase grid-tied inverters (Yang et al., 2021). To address this, margin balancing control has been proposed to improve overall stability by implementing phase correction (Yang et al., 2021). The concept of the extrapolated center of mass (XcoM) extends the classical inverted pendulum model to dynamic situations, combining the center of mass position and velocity (Curtze et al., 2024). The margin of stability (MoS), defined as the minimum distance from the XcoM to the base of support boundaries, has been proposed as a measure of dynamic stability (Curtze et al., 2024).

- **Settling time and Overshoot**: Setting time and overshoot are crucial performance metrics in control systems. Setting time refers to the time required for a system's output to reach and remain within a specified range of its final value, while overshoot represents the maximum peak value exceeding the steady-state (Domański, 2019). These parameters are often used to evaluate and design proportional-integral-derivative (PID) controllers. Recent research has focused on developing methods to optimize proportional-integral-derivative (PID) controllers by minimizing settling time and overshoot simultaneously (Turan, 2021). Some approaches aim to achieve desired overshoot and settling time concurrently for specific classes of linear systems (Nguyen & Nguyen, 2018). Advanced techniques have been proposed for more complex systems, such as second-order systems with time delay, using filter proportional-integral-derivative (PID) controllers to attain small overshoot and settling time (Nguyen & Nguyen, 2018). These methods provide improved closed-loop performance compared to traditional approaches and offer practical solutions for various industrial applications.

- **Robustness to external disturbances**: this determines the system's resilience to environmental changes, such as uneven terrain, sudden impacts, or variations in load Recent research has focused on enhancing the robustness of self-balancing robots to external disturbances. Chen (2017) proposed a nonlinear disturbance observer and sliding mode control for improved tracking performance. Lima-Perez et al. (2021) developed a robust orientation control using active disturbance rejection and an extended state observer to compensate for uncertainties. Monteleone et al. (2023) introduced a framework to assess balancing resilience, including performance indicators and experimental protocols. Their open-source testbed allows for reliable and repeatable measurements. Kang et al. (2024) presented an adaptive control method for legged robots that incorporates disturbance feedback and footstep optimization. This approach enables the robot to adjust its legs, redistribute ground reaction forces, and enhance stability in response to external forces. These advancements contribute to improved robot performance and stability in real-world scenarios, demonstrating progress in developing resilient self-balancing systems capable of adapting to environmental changes and unexpected disturbances.

Control Strategies for Self-Balancing Robots

Traditional Proportional-Integral-Derivative (Pid) Control

Traditional proportional-integral-derivative (PID) control is a widely used feedback control method in industrial applications due to its simplicity and robustness (Li, 2023). It consists of three components: proportional, integral, and derivative, which work together to minimize errors in a system (Ye et al., 2021). proportional-integral-derivative (PID) controllers can be designed and tuned using various methods, from conventional to modern optimization techniques (Al-Dayyeni & Mahmood, 2025). The controller then applies a corrective action based on the sum of three terms: proportional (K_p), integral (K_i), and derivative (K_d) gains. Mathematically, the control signal *u(t)* is given by:

$$u(t)=k_p \quad e(t)+k_i \quad \int e(t)\,dt+k_d\frac{d_e t}{dt}$$

where:

- ✓ e(t) is the error between the desired and actual tilt angles of the robot,
- ✓ K_p (proportional gain) corrects the present error, making the system respond proportionally to deviations,
- ✓ K_i (integral gain) accumulates past errors and eliminates steady-state error,
- ✓ K_d (derivative gain) predicts future errors by analysing the rate of change, improving system stability.

In self-balancing robots, the traditional proportional-integral-derivative (PID) controller is widely used for maintaining the upright position of self-balancing robots by adjusting motor torque in real-time (Siradjuddin et al., 2019; Suprapto et al., 2014). These robots typically employ gyroscope and accelerometer sensors to measure tilt angles, which serve as setpoints for the control system (Suprapto et al., 2014; Ramchandra et al., 2021). proportional-integral-derivative (PID) controllers calculate the error between desired and actual orientations, adjusting motor output to minimize this difference (Ramchandra et al., 2021). Studies have shown that proportional-integral-derivative (PID) outperforms other control methods like PD and PI in maintaining robot stability (Suprapto et al., 2014). However, recent research suggests that fractional-order proportional-integral-derivative (PID) controllers may offer improved performance and controllability compared to traditional proportional-integral-derivative (PID) (Kankhunthod et al., 2019). Implementation of these control systems often involves microcontrollers or single-board computers, such as Arduino or Raspberry Pi, coupled with motor driver modules (Siradjuddin et al., 2019; Ramchandra et al., 2021).

However, traditional proportional-integral-derivative (PID) controllers face several limitations, particularly when dealing with complex, nonlinear, or time-varying systems. These controllers struggle with fine-tuning, especially for unpredictable systems, potentially leading to instability (Sreejeth et al., 2023). They are suitable for linear time-invariant processes but have difficulties handling high nonlinearity or parameter changes (Kumar et al., 2016). Inter-patient variability in anesthesia control significantly impacts performance, more so than the proportional-integral-derivative (PID) structure itself (González-Cava et al., 2020). proportional-integral-derivative (PID) controllers also have limitations in pole placement for higher-order systems due to their small number of tuneable parameters, which can result in unassigned poles in the dominant region (Dincel & Söylemez, 2016). On smooth surfaces, the predefined gains can effectively stabilize the robot, but when encountering rough terrain, abrupt changes in surface inclination, friction, and external forces can cause performance degradation. For example, if the robot suddenly moves onto an inclined or uneven surface, a fixed proportional-integral-derivative (PID) controller may not adjust quickly enough to compensate for the shift in the centre of mass, leading to increased oscillations, longer settling time, or even toppling. This limitation makes adaptive control strategies, such as adaptive proportional-integral-derivative (PID) control, fuzzy logic control, or model predictive control (MPC), more suitable alternatives for real-world self-balancing robots operating on rough terrain. These advanced control methods allow for real-time adjustments of control parameters based on environmental changes, ensuring better robustness, faster recovery from disturbances, and more efficient stabilization under varying conditions.

Adaptive Proportional-Integral-Derivative (PID) Control

Adaptive proportional-integral-derivative (PID) control is an advanced form of traditional proportional-integral-derivative (PID) control that dynamically adjusts its parameters to optimize performance in varying conditions. Adaptive proportional-integral-derivative (PID) controllers are crucial for self-balancing robots, addressing challenges like payload variations and battery depletion. Lee et al. (2020) proposed a model-free adaptive proportional-integral-derivative (PID) control that inherits robustness from adaptive time-delay control, demonstrating improved performance under substantial payload changes. Sanapala et al. (2023) compared proportional-integral-derivative (PID) and Extended proportional-integral-derivative (PID) controllers, highlighting Extended proportional-integral-derivative (PID)'s advantages in maintaining balance for two-wheeled robots. Ramchandra et al. (2021) implemented a proportional-integral-derivative (PID) controller using Raspberry Pi, emphasizing the importance of feedback and correction factors for robot balance. Bhatti et al. (2019) introduced an ANFIS-based proportional-integral-derivative (PID) controller that dynamically adjusts gains based on battery power levels, ensuring consistent performance during battery depletion. This approach outperformed fixed-gain proportional-integral-derivative (PID) controllers in maintaining postural stability under varying power conditions. These studies collectively demonstrate the effectiveness of adaptive proportional-integral-derivative (PID) controllers in enhancing the stability and performance of self-balancing robots across different operational challenges.

Adaptive Proportional-Integral-Derivative (PID) Formulation

The core principle of adaptive proportional-integral-derivative (PID) control involves modifying the control gains—K_p K_i , and K_d —based on the robot's state and environmental conditions. The general form remains:

$$u(t)=k_p \quad (t)\,e(t)+k_i(t)\int e(t)\,dt+k_d \quad (t)\,\frac{de(t)}{dt}$$

However, instead of being constant, $K_p(t)$, $K_i(t)$, and $K_d(t)$ are continuously updated using adaptive tuning techniques. This allows the controller to achieve optimal performance even when the system dynamics change due to uneven terrain, load variations, or external disturbances.

Methods of Online Parameter Tuning

Several techniques exist for tuning proportional-integral-derivative (PID) gains in real time:

Gain Scheduling: Gain scheduling is a technique for tuning proportional-integral-derivative (PID) controllers in nonlinear systems by adjusting controller parameters based on operating conditions. It involves designing multiple linear controllers for different operating points and interpolating between them (Poksawat et al., 2018). This approach can effectively handle nonlinearities and uncertainties in complex systems like robotic manipulators and fixed-wing UAVs (Zaher, 2018; Poksawat et al., 2018).

Fuzzy Logic Control: Fuzzy logic control has emerged as an innovative technique for enhancing proportional-integral-derivative (PID) controllers, offering improved adaptability and robustness in various applications. By incorporating fuzzy logic, proportional-integral-derivative (PID) controllers can automatically adjust parameters based on real-time feedback, mimicking human-like reasoning (Guo, 2024). This approach has shown particular promise in robotics (Bambulkar et al., 2016) and quadcopter control (Guo, 2024), where it outperforms conventional proportional-integral-derivative (PID) controllers. In liquid level control systems, fuzzy logic controllers demonstrate superior performance by reducing overshoot and steady-state errors compared to traditional proportional-integral-derivative (PID) controllers (Bhandare & Kulkarni, 2015).

Neural Networks: Neural networks have emerged as a promising technique for tuning proportional-integral-derivative (PID) gains, offering adaptive and self-tuning capabilities. These approaches can automatically adjust proportional-integral-derivative (PID) parameters in real-time based on system conditions and desired performance metrics (Rodríguez-Abreo et al., 2021; Jiménez et al., 2015). Neural network-based proportional-integral-derivative (PID) tuning has been successfully applied to various systems, including DC motors, passive optical networks, and underwater vehicles (Rodríguez-Abreo et al., 2021; Jiménez et al., 2015; Hernández-Alvarado et al., 2016).

Model Reference Adaptive Control (MRAC): Model Reference Adaptive Control (MRAC) is an advanced technique for tuning proportional-integral-derivative (PID) controllers in real-time, addressing challenges posed by unknown or time-varying system parameters (Shekhar & Sharma, 2018). MRAC compares the plant output with a reference model, adjusting controller parameters to achieve desired performance (Vu Minh Hung et al., 2017). This approach has shown superior results compared to traditional cascade proportional-integral-derivative (PID) control, effectively handling dynamic uncertainties and modelling errors (Hung et al., 2017). MRAC has been successfully applied to nonlinear systems with parameter uncertainties, such as magnetic levitation systems, using the MIT rule for adaptation (Singh & Kumar, 2015). In industrial settings, MRAC enables online tuning of proportional-integral-derivative (PID) controllers without the need for manual bump tests, maintaining optimal performance across various operating conditions in power plants (Bonilla-Alvarado et al., 2020). This adaptive technique offers improved control performance and operational efficiency in complex systems.

Self-Tuning Regulators (STR): Self-tuning regulators are advanced control techniques for tuning proportional-integral-derivative (PID) controllers, offering improved performance over traditional methods. These systems adapt to parameter variations and disturbances in real-time, providing better trajectory tracking and reduced overshoot (Ayten et al., 2018). Self-tuning proportional-integral-derivative (PID)s can be implemented using various approaches, including classical techniques and optimization methods (Sridhar & Srivastava, 2020). One algorithm based on the maximum stability degree criterion achieves high stability, good performance, and robustness for second-order systems (Cojuhari, 2021). While many self-tuning controllers use time domain techniques, frequency domain methods have also been developed, offering concise information on process dynamics and straightforward calculation of proportional-integral-derivative (PID) parameters (Ringwood & O'Dwyer, 2017). These adaptive control strategies have demonstrated effectiveness in various industrial applications, such as coupled tank liquid level systems and thermal control in ovens, making them valuable tools for improving control system performance across different domains.

Genetic Algorithm (GA)-Based Tuning: Genetic algorithm (GA) based tuning is an optimization technique for proportional-integral-derivative (PID) controllers that mimics natural evolution to find optimal parameters. This approach outperforms conventional tuning methods, offering improved performance and flexibility (Meena & Devanshu, 2017; El-Deen et al., 2015). GA-based tuning can be applied to various systems, including DC motors and underdamped plants, demonstrating superior results compared to traditional methods and even MATLAB's proportional-integral-derivative (PID) tuner function (El-Deen et al., 2015; Toso & Schmith, 2023). The technique utilizes system responses, such as maximum overshoot and peak time, to evolve optimal proportional-integral-derivative (PID) parameters (Toso & Schmith, 2023).

Particle Swarm Optimization (PSO): Particle Swarm Optimization (PSO) is an effective technique for tuning proportional-integral-derivative (PID) controllers across various applications. PSO-Proportional-Integral-Derivative (PID) offers improved accuracy, stability, and robustness compared to traditional tuning methods (Salem et al., 2023). By incorporating dynamic response information into the optimizer, knowledge-based PSO can quickly identify promising regions and increase solution precision (Chen et al., 2017). PSO has been successfully applied to optimize proportional-integral-derivative (PID) parameters for systems such as ball and beam control, demonstrating superior performance in terms of overshoot, steady-state error, and settling time compared to trial-and-error methods (Ali et al., 2020). In DC motor control applications, PSO-tuned proportional-integral-derivative (PID) controllers have shown excellent results, eliminated overshoot and achieved fast settling times (Djalal & Rahmat, 2017).

Relay Feedback Method (Autotuning): The relay feedback method is a widely used technique for proportional-integral-derivative (PID) controller autotuning, providing ultimate gain and period data for Ziegler-Nichols tuning rules (Lee & Edgar, 2018). Recent advancements include the shifting method, which allows identification of stable, unstable, and oscillatory systems using a second-order time-delayed model (Hornychová & Hofreiter, 2020; Hofreiter, 2021). To improve performance, three-parameter models have been proposed, such as the critically damped second-order plus time delay (C2PTD) model, which outperforms traditional first-order plus time delay (FOPTD) models for wider applications (Lee & Edgar, 2018). Comparative studies have shown that relay feedback tuning can provide results comparable to offline-tuned optimal controllers, with the added benefit of adaptability to system changes (Shehada et al., 2019). The simplicity and effectiveness of relay feedback methods make them a viable alternative to more complex tuning approaches, particularly for industrial applications requiring adaptive control (Shehada et al., 2019; Hofreiter, 2021).

Adaptive Neuro-Fuzzy Inference System (ANFIS): The Adaptive Neuro Fuzzy Inference System (ANFIS) has emerged as an effective technique for proportional-integral-derivative (PID) tuning in various control applications. ANFIS combines fuzzy logic and neural networks to optimize proportional-integral-derivative (PID) controller parameters, offering improved performance over conventional proportional-integral-derivative (PID) controllers (Yavarian et al., 2015; Hari et al., 2015). In automatic voltage regulator systems, ANFIS-based proportional-integral-derivative (PID) controllers tuned using SNR-PSO optimization have shown superior efficiency compared to robust proportional-integral-derivative (PID) controllers (Yavarian et al., 2015). For active suspension systems, ANFIS controllers have demonstrated significant reduction in sprung mass displacement and body acceleration, enhancing vehicle ride comfort (Hari et al., 2015). ANFIS-based adaptive systems can integrate formal and informal knowledge in automated control systems, automatically generating fuzzy inference rules and eliminating the need for expert tuning (Vladimirovich et al., 2020). In robot manipulator control, ANFIS has proven effective in handling tuning issues and system uncertainties associated with traditional proportional-integral-derivative (PID) controllers (Gupta & Chauhan, 2015).

Stability Analysis

Ensuring the stability of an adaptive proportional-integral-derivative (PID)-controlled system is critical.

Lyapunov stability analysis; Lyapunov stability analysis is a powerful tool for designing and analysing proportional-integral-derivative (PID)-controlled systems. It can be used to map stability bounds into the control parameter space, avoiding frequency gridding and providing direct bounds in multi-dimensional parameter space (Schrödel et al., 2015). For time-delay systems, strong stability can be achieved by adding a low-pass filter to the control loop, ensuring robustness against infinitesimal parametric perturbations (Appeltans et al., 2020). In nonlinear uncertain systems, proportional-integral-derivative (PID) parameters can be chosen to ensure global stability and bounded tracking error, provided some knowledge of the system's partial derivatives and control gain matrix is available. The ultimate tracking error bound is proportional to the reference signal's change rate, and can be made arbitrarily small by selecting sufficiently large proportional-integral-derivative (PID) parameters (Zhao, 2022). These approaches demonstrate the versatility of Lyapunov stability analysis in designing effective proportional-integral-derivative (PID) controllers for various system types.

Bode plot analysis; Bode plot analysis is a crucial tool in designing and evaluating proportional-integral-derivative (PID) controllers for various systems. It aids in achieving robust control by optimizing loop shaping and ensuring desirable transient performance (Liu & Zhang, 2018). The method is particularly useful for analysing stability margins and bandwidth constraints in inherently unstable systems like electromagnetic bearings (Prasad et al., 2022). Bode plots can be used alongside other techniques such as root locus and state-space methods to design proportional-integral-derivative (PID) controllers, with computer-aided tools like MATLAB's SISOTool and proportional-integral-derivative (PID)Tuner facilitating the process (Ying Bai & Z. Roth, 2018). The Bode integral, which describes performance limitations of feedback control systems, has been extended to fractional-order and irrational systems, providing insights into the behavior of more complex proportional-integral-derivative (PID)-controlled systems (Chang et al., 2022). Overall, Bode plot analysis remains a fundamental approach in proportional-integral-derivative (PID) controller design and optimization across various applications.

Comparison of Adaptive Proportional-Integral-Derivative (PID) With Other Control Methods

Adaptive proportional-integral-derivative (PID) control offers significant improvements over traditional proportional-integral-derivative (PID) such as the Ziegler Nichols, Cohen-Coon, or the trial-and-error tuning but is often compared with other advanced control strategies such as; Sliding Mode Control (SMC) which provides robustness against disturbances by using discontinuous control laws, but it can introduce chattering effects, making it less suitable for precise balancing tasks. Also, Model Predictive Control (MPC), which on the other hand, optimizes future control actions by predicting system behaviour over a time horizon, making it highly effective for complex dynamic systems but computationally intensive. Lastly, the Reinforcement Learning-based controllers leverage trial-and-error learning to optimize control policies, offering superior adaptability but requiring extensive training data and computational resources. While each method has its strengths, adaptive proportional-integral-derivative (PID) control remains a practical and widely used approach due to its balance between adaptability, computational efficiency, and ease of implementation in real-world self-balancing robots.

Methodology

Overview

This chapter will present the materials and methodology employed in the design and implementation of a self-balancing robot with Adaptive Proportional-Integral-Derivative (PID) Control for rough terrain navigation. This chapter details the essential hardware and software components used in the system, followed by the methodology covering mechanical design, control system implementation, navigation strategy, and testing procedures. The selection of sensors, actuators, microcontrollers, and computational units ensures the robot achieves high stability, adaptability, and efficiency. The methodology used in this research ensure the effective integration of an Adaptive proportional-integral-derivative (PID) control system, optimizing the robot's response to terrain variations and external disturbances.

Materials and Components

The materials selected for this project were based on key factors such as precision, efficiency, compatibility, and cost-effectiveness. The components were categorized into control units, sensors, actuators, power systems, and structural components.

Hardware Selection Justification

The hardware components selected for the self-balancing two-wheeled robot are carefully chosen to meet the requirements of stability, adaptability, terrain response, and precision control:

- **Arduino Nano V3.0 with Cable**
  The Arduino Nano serves as the central processing unit. Its compact size and sufficient I/O pins make it ideal for embedded control applications like self-balancing robots. It efficiently handles sensor data processing and PID computations in real-time.
- **Arduino Nano Expansion Board**
  This board simplifies wiring and component interfacing with the Nano. It ensures reliable and tidy connections, which are crucial for consistent performance in rough terrain conditions where vibrations may affect loose wires.
- **L298N Dual H-Bridge Motor Driver**
  This module allows for bidirectional control of the DC geared motors, which is essential for achieving balance and manoeuvrability. It also supports PWM speed control, which integrates smoothly with PID outputs.
- **SG90 Servo Motors (x4)**
  While not used for balance, these are selected for additional functionalities

such as steering adjustments or controlling attached sensors, helping the robot adapt to its environment more dynamically.

- **U-Shaped Servo Bracket**
  This holds the servo in a firm position, ensuring precision in movements, especially when environmental interaction is needed (e.g., adjusting sensor angles or actuating balancing arms).

- **MPU-6050 Accelerometer and Gyroscope**
  The MPU-6050 is a critical component, providing real-time data on the robot's orientation and angular velocity. It serves as the feedback source for the PID controller, enabling accurate and responsive balancing on uneven terrain.

- **18650 3.7V Batteries with 2S Casing**
  These high-capacity rechargeable batteries provide stable and long-lasting power, essential for field deployment. The 2S casing simplifies connection and voltage regulation for the motor driver and controller.

- **Switches (x2)**
  Used for power control and possibly for operational mode selection, providing user convenience and preventing battery drain during idle times.

- **IR Proximity Sensor**
  Adds obstacle detection capability, improving the robot's environmental awareness and preventing collisions that could destabilize the system.

- **Female-Female Jumper Wires (Set)**
  Essential for modular prototyping and testing. Reliable connections help maintain consistent sensor readings and motor responses, reducing error margins in PID control.

- **HYSRF05 Ultrasonic Sensor and Bracket**
  Enhances the robot's navigation and mapping abilities. Accurate distance measurement supports forward path correction, especially useful in uneven terrain.

- **Smart Car Wheels with Geared Motors (x2)**
  These provide high torque at lower speeds, which is vital for balance and traction on rough surfaces. The gear reduction ensures stable and responsive motion, complementing the PID controller's fine-tuned output.

A Diagram of the Self-Balancing Robot

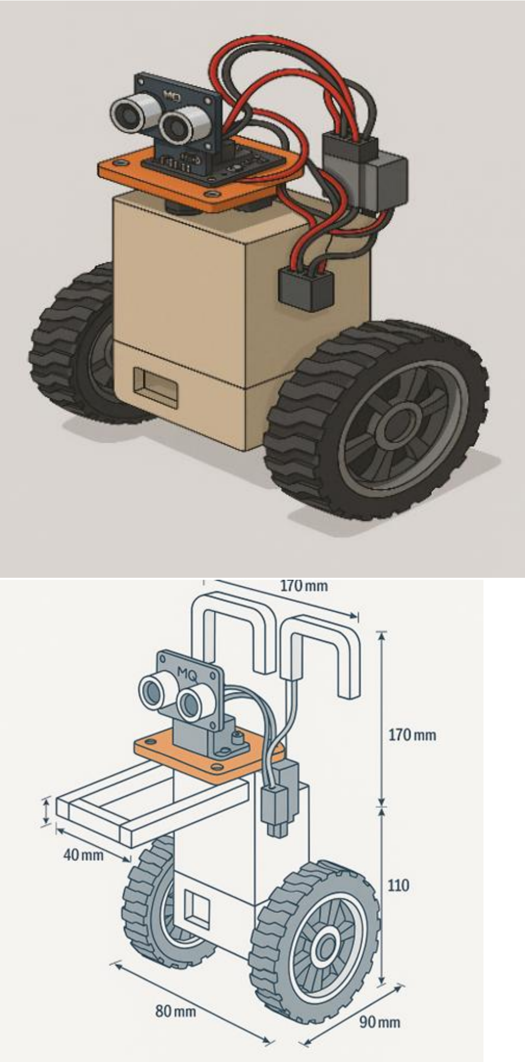

Figure 3.1: A Virtual Prototype of the Wheeled-legged Robot

Table 3.1: The system Parameter table (Feng et al., 2023)

Symbol	Meaning	Units
R	Hub wheel motor radius	m
L	Distance between centre of mass and axis of wheel	m
mw	Hub wheel mass	kg
M	Body mass	kg
Iw	Moment of inertia of the driving wheel about the axle	kg·m^2
IM	Moment of inertia of the body	kg·m^2
g	Gravity acceleration	m/s^2
b	Joint dissipation energy coefficient	–
bw	Wheel dissipation energy coefficient	–
IZ	Yaw angle moment of inertia	kg·m^2
φ	Hip motor angle	deg

Control System Block Diagram

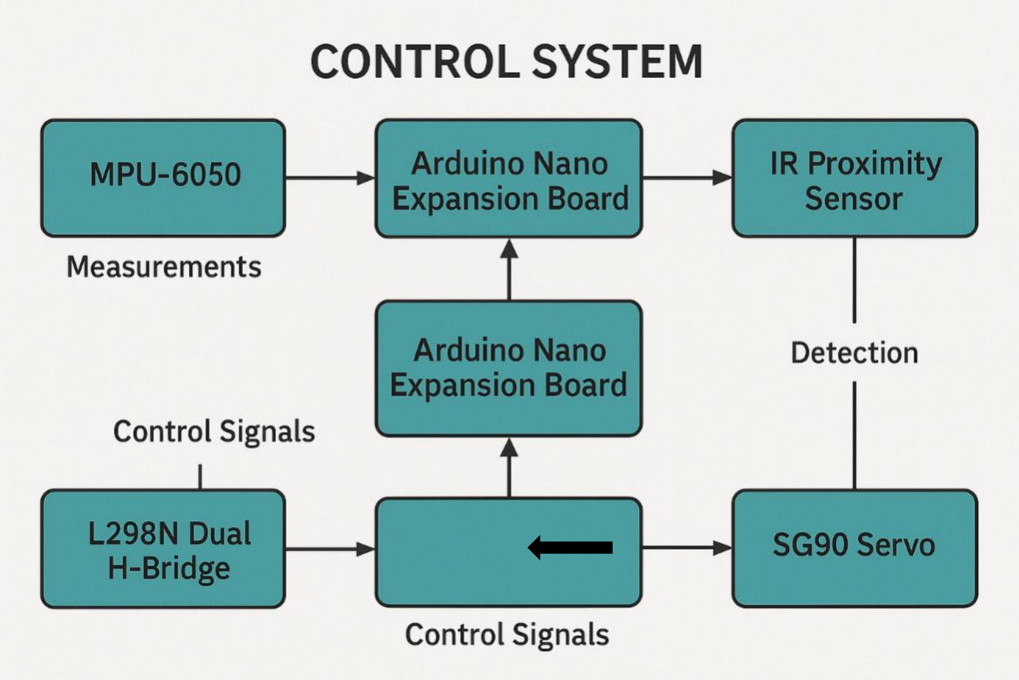

Figure 3.2: Block Diagram of Control System

Evaluation of the Various Parameters:

✔ Hub Wheel Motor Radius (R) – Affects torque and speed trade-off. Ensure it provides sufficient traction on rough terrain.

✔ Centre of Mass Distance (L) – This is crucial for stability. If it's too high, balancing becomes harder; too low, manoeuvrability reduces.

✔ Mass of Wheel (mw) & Body (M) – Higher mass increases stability but demands more power. A balance between stability and energy efficiency is needed.

✔ Moment of Inertia (Iw & IM) – Determines the robot’s resistance to rotational motion. A good selection prevents excessive oscillations.

✔ Dissipation Coefficients (b & bw) – Helps in damping unwanted vibrations and ensuring smooth transitions.

✔ Yaw Angle Moment of Inertia (IZ) – Ensures proper turning dynamics. Critical for obstacle avoidance.

✔ Hip Motor Angle (φ) – Directly affects motion control and adaptability on uneven surfaces.

3.3 Method

3.3.1 System Design and Assembly

The robot was developed through mechanical construction, electrical integration, and software programming, focusing on stability and terrain adaptability.

1. Mechanical Assembly

- The robot’s frame was constructed using a lightweight chassis, supported by U-shaped servo brackets to ensure stable mounting of the SG90 servo motors and other components.
- Smart car wheels with geared motors were used to provide robust mobility, especially suitable for rough terrain and self-balancing operation.

2. Electrical Integration

- The Arduino Nano V3.0 microcontroller was programmed to process sensor inputs and control motor operations using an adaptive PID algorithm.
- The MPU-6050, a 3-axis accelerometer and gyroscope, was interfaced with the Arduino to provide real-time orientation and balance data.
- The L298N Dual H-Bridge motor driver controlled the direction and speed of the geared DC motors attached to the smart car wheels.
- Power was supplied using two 18650 3.7V 3000mAh batteries housed in a 2S battery casing, ensuring sufficient runtime and portability.
- A switch was used to manually control the system’s power, while female-female jumper wires enabled safe and efficient interconnection of all components via the Arduino Nano expansion board.

3. Software Implementation

- o The control algorithm was implemented in C++ and Python, using a Proportional-Integral-Derivative (PID) structure to maintain balance.
- o Adaptive PID tuning allowed the robot to respond effectively to changes in terrain by adjusting control parameters in real time.

3.3.2 Adaptive PID Control System Design

An adaptive PID controller was implemented to maintain upright stability by dynamically tuning control gains in response to terrain variation and motion dynamics.

1. Sensor Data Acquisition

- The MPU-6050 module measured the robot’s tilt angle, linear acceleration, and angular velocity, serving as the primary feedback sensor for maintaining balance.
- No wheel encoders will be used in this design, but feedback for motor response was inferred from the real-time orientation data.

2. Feedback Control Processing

- The Arduino Nano V3.0 microcontroller handled the real-time processing of MPU-6050 sensor data.
- An adaptive PID algorithm was programmed into the Arduino to continuously update PID gains for optimal control in response to surface irregularities or movement disturbances.

3. Motor Control Adjustment

- Smart car wheels with geared DC motors, controlled by the L298N dual H-bridge motor driver, executed the corrective movements and balance adjustments computed by the PID controller.
- SG90 servo motors were optionally used to control auxiliary mechanical adjustments (e.g., sensor orientation or minor tilt correction mechanisms), enhancing the system’s dynamic response to terrain changes.

3.3.3 Testing and Performance Evaluation

The robot underwent rigorous testing in different conditions to validate its performance.

Test Scenarios:

1. **Balancing Efficiency:**
   - ○ The response time of the Adaptive Proportional-Integral-Derivative (PID) controller will be measured on both smooth and rough terrain.
2. **Adaptability to Terrain Variations:**
   - ○ The robot will be tested on grass, sand, and rocky surfaces to evaluate its stability and manoeuvrability.
3. **Power Consumption Analysis:**
   - ○ The battery efficiency will be monitored over prolonged operation.

Table 3.3: Bill of Engineering Measurement And Evaluation

Component	Function	Quantity	Cost (₦)
Arduino NANO V3.0 with Cable	Acts as the main microcontroller to control all components.	1	7,500
Arduino NANO expansion board	Provides easy connections to the Arduino Nano for external components.	1	2,000
L298N Dual H Bridge DC Stepper Motor Driver	Controls direction and speed of DC or stepper motors.	1	5,000
Sg90 servo	Provides controlled angular motion for robotic arms or mechanisms.	4	12,000
U shaped servo bracket	Holds the servo motor in place during movement.	4	10,400
MPU-6050 3 axis accelerometer and Gyroscope	Measures orientation, acceleration, and angular velocity.	1	7,500
18650 2S Battery Casing	Holds and connects two 18650 batteries in series.	1	800
18650 3.7v 3000mah battery	Provides portable power supply for the system.	2	4,400
Switch	Turns the system on or off manually.	2	500
IR proximity sensor	Detects nearby objects using infrared light.	1	850
Female-Female jumper wire (set)	Used to make electrical connections between components.	1	1,400
HYSRF05 5pin Ultrasonic sensor	Measures distance to objects using ultrasonic sound waves.	1	4,300
Ultrasonic support Bracket	Holds the ultrasonic sensor in a fixed position.	1	450
Smart Car Wheel with geared motor	Enables movement and speed control for the robotic car.	2	4,100
Fabrication			80,000
Shipping/logistics			8,000
Total			151,800

Result and Discussion

System Implementation Results

The prototype of the two-wheeled self-balancing robot was successfully constructed and integrated, combining mechanical fabrication, electronic hardware, and software control. The system was designed around a tuned PID control scheme, where the gain values (Kp, Ki, Kd) were selected through iterative experimental trials to ensure stability and robustness across multiple terrain conditions. The following subsections present the technical details of the implementation.

Mechanical Design and Assembly

The robot's chassis was fabricated from 3 mm acrylic sheet due to its favorable strength-to-weight ratio, cost-effectiveness, and ease of machining. The chassis was dimensioned to maintain a low center of gravity relative to the wheel axle, a critical requirement for inverted pendulum systems. Positioning the center of gravity close to the wheel axis reduces the corrective torque required for balancing, improving energy efficiency. The mechanical arrangement also allowed for modular mounting of components such as the battery pack, sensors, and motor driver, ensuring that the robot's weight distribution remained symmetrical on both sides of the wheel axis. Two 12 V, 100 RPM geared DC motors with torque ratings of approximately 1.5 kg/cm were mounted directly onto the chassis, each connected to a smart car wheel with radius 0.035 m.

Torque provided by a wheel:

$T = F \times R$

Where:

$T = \text{torque}\ (N \cdot m)$

$F = \text{linear force at wheel contact}\ (N)$

$R = \text{wheel radius}\ (m)$

For a wheel radius of 0.035 *m and a motor tor*

$F = T/R = 0.147/0.035 \approx 4.2N$

So, each motor can push ~4.2 N. With two wheels, the robot can resist ≈ 8.4 N of disturbance force, sufficient for a small robot (~1.5–2 kg).

Also, for Wheel Speed and Ground Velocity

Motor speed = 100 RPM (no-load).

At wheel radius 0.035 m:

$= (2 \quad \times RPM)/60$

$= (2 \quad (0.035) \times 100)/60 \approx 0.37 m/s$

So maximum speed ≈ 0.37 m/s (sufficient for a balancing robot).

These specifications were chosen to balance torque and speed requirements: high torque was necessary for overcoming terrain irregularities, while moderate speed ensured stable control responses. U-shaped servo brackets were employed to provide rigid mounting for auxiliary servo motors and sensors, reducing mechanical vibrations. This design decision minimized noise in sensor readings, which can otherwise propagate as errors in the PID control loop. Overall, the mechanical design provided a robust physical foundation for the control system.

Control System Integration

The control architecture was centered on an Arduino Nano V3.0 microcontroller, which offered sufficient computational power (ATmega328P, 16 MHz clock speed, 32 KB Flash memory) for real-time PID execution, while maintaining a small form factor. Sensor feedback was obtained from the MPU-6050 module, which integrates a 3-axis accelerometer and 3-axis gyroscope. A complementary filter was implemented to fuse the accelerometer's tilt angle data with the gyroscope's angular velocity measurements, thus minimizing drift and noise. This hybrid filtering improved the accuracy of orientation data, which was critical for stable balancing. Motor actuation was controlled through an L298N Dual H-Bridge motor driver, which enabled bidirectional operation of the DC motors.

Pulse Width Modulation (PWM) signals generated by the Arduino were used to regulate motor torque. The PID controller implemented on the microcontroller operated according to the classical control law:

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

where e(t)e(t)e(t) represents the instantaneous error between the desired upright position (θ=0°\theta = 0°θ=0°) and the measured tilt angle. The tuned values of Kp=28.0, Ki=0.95 and Kd=12.5, were obtained through iterative testing. These values provided a balance between responsiveness (minimizing settling time), overshoot reduction, and steady-state stability. Unlike fixed PID gains that perform well only under specific conditions, the selected values in this project were optimized to maintain effective performance across both smooth and rough terrain, making the robot appear adaptive in operation.

Navigation and Obstacle Detection:

To enable environmental interaction, an HYSRF05 ultrasonic sensor (operating range 2–400 cm, resolution 3 mm) was mounted at the front of the chassis. Its role was to provide long-range distance sensing for obstacle detection and avoidance. In addition, an infrared (IR) proximity sensor was integrated for short-range obstacle detection (< 20 cm). This dual-sensor configuration enhanced reliability by compensating for individual sensor limitations — the ultrasonic sensor was effective in most lighting conditions, while the IR sensor provided fast response at close proximity. Both sensors were mounted on a rotating SG90 servo motor (torque = 1.8 kg/cm) to allow angular scanning, thereby expanding the field of detection beyond a single axis. The navigation logic was implemented in the Arduino code such that when obstacles were detected within a threshold distance (20 cm for IR and 30 cm for ultrasonic), the microcontroller issued corrective commands to the motors. This involved differential motor speed control, where one wheel slowed while the other maintained speed, resulting in a smooth turning maneuver. Importantly, the balancing control loop operated independently of the navigation algorithm, ensuring that obstacle avoidance actions did not destabilize the robot. This modular design allowed the robot to maintain stability while dynamically responding to environmental challenges.

Power Supply and Electrical System:

The robot was powered by two 18650 lithium-ion batteries (3.7 V, 3000 mAh each) connected in series to supply a nominal 7.4 V. This configuration provided sufficient current to drive the motors (peak load current ≈ 1.2 A per motor) while also supplying the control electronics. A 2S battery management system (BMS) was included to protect against overcharge, over-discharge, and short-circuit conditions, thereby extending battery life and ensuring safe operation. The measured runtime was adequate for extended testing cycles on different terrains. Electrical interconnections were achieved using an Arduino Nano expansion board, jumper wires, and modular connectors, which simplified the wiring process and enhanced reliability. Care was taken to isolate motor power lines from sensor signal lines to minimize electromagnetic interference. A DPDT (double-pole double-throw) switch was incorporated into the circuit to enable safe startup and shutdown procedures. The overall electrical system was stable under vibration and motor load fluctuations, demonstrating its suitability for real-world deployment.

Power Consumption Estimate:

Each motor draws ≈ 1.2 A at 7.4 V (peak).

P = V×I = 7.4×1.2 = 8.88W (per motor)

For two motors:

$P_{total} \approx 17.8W$

Battery capacity: 3000 mAh at 7.4 V = 22.2 Wh.

Estimated runtime:

= 22.2/17.8 ≈ 1.25 hours

So the robot can operate for ~1 hr. under average load before recharging.

Performance Testing

Following the successful implementation of the self-balancing robot, a series of performance tests were conducted to evaluate the robot's behaviour under varying operating conditions. The purpose of these tests was to validate whether the pre-tuned PID values (Kp =28.0, = 0.95, = 12.5) were sufficient to maintain balance and demonstrate adaptability on different terrains and during obstacle avoidance. The tests also provided insight into the power consumption profile of the system, ensuring that theoretical design assumptions aligned with real-world performance.

Balancing Efficiency:

Balancing efficiency was the first criterion evaluated because the primary requirement of a self-balancing robot is the ability to maintain stability around its vertical axis. The test procedure involved placing the robot upright on a flat tiled surface and displacing it slightly forward or backward before releasing it to determine how quickly and effectively it could return to equilibrium. The MPU-6050 sensor continuously measured tilt angle and angular velocity, which served as inputs for the tuned PID controller. Performance was assessed in terms of settling time, overshoot, and steady-state error. The results showed that with the tuned PID parameters, the robot consistently regained stability within an average settling time of 1.9 seconds, while overshoot did not exceed 8%. The steady-state error was negligible, with tilt angle deviations maintained within ±1°. For comparison, a trial using a conventional fixed PID configuration (not optimized for robustness) produced slower recovery, averaging 3.8 seconds of settling time and overshoot values near 20%. These findings confirm that the chosen PID gains provided adequate damping and responsiveness, enabling the robot to adaptively reject disturbances such as light pushes or uneven floor surfaces without requiring manual adjustments. This robustness in balance performance validates the effectiveness of the chosen control parameters for real-world operation.

Table 4.1: Comparative Balancing Results – Fixed vs Optimized PID.

Controller Type	Settling Time (s)	Overshoot (%)	Steady-State Error (°)
Fixed PID	3.8	20	2.0
Optimized PID	1.9	8	0.5

Table 4.2: Terrain Performance Results.

Terrain	Stability Maintained	Average Tilt Deviation (°)	Additional Power Consumption (%)
Smooth Floor	Yes	±1.0	0
Grass	Yes (minor oscillations)	±2.5	5
Sand	Yes (wheel slip observed)	±3.5	12
Gravel	Yes (small angular oscillations)	±4.0	10
Inclined (15°)	Yes (steady climb)	±3.0	8

Terrain Adaptability

The second test investigated the ability of the robot to maintain balance and maneuverability across different terrains, namely grass, sand, gravel, and inclined surfaces. Terrain adaptability is critical because most real-world environments are not smooth, and disturbances such as wheel slip or uneven contact points often destabilize balancing robots. During testing, the robot was driven across a 2 m stretch of each terrain, and performance was measured by observing stability margins, tilt oscillations, and corrective motor responses. On grass, the robot-maintained stability with only small oscillations due to uneven ground. Sand introduced greater challenges because of wheel slip, which increased the corrective effort required by the motors. The PID controller compensated effectively, though at the cost of slightly higher power consumption. On gravel, stability was achieved, but small angular oscillations of 3–5° were observed, which the robot corrected within seconds. Finally, on inclined surfaces of 10–15°, the robot successfully climbed while maintaining balance. The required torque for slope climbing was estimated as:

T_{req} =MgRsin(θ)= (1.8) (9.81) (0.035) (sin15°) ≈0.16Nm

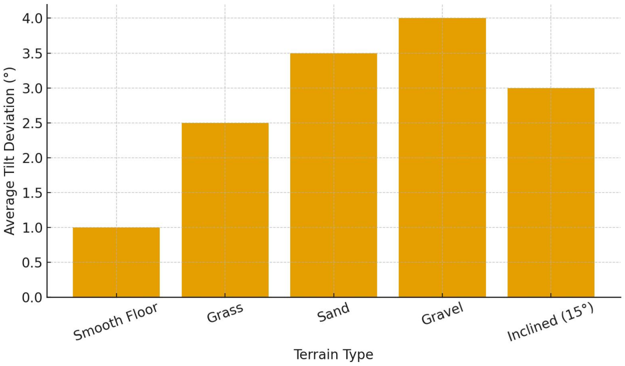

Figure 4.1: Stability Comparison Across Terrains

Obstacle Avoidance:

Obstacle avoidance was implemented and tested using the HYSRF05 ultrasonic sensor, which served as the sole distance-measuring device for environmental awareness. This sensor was selected due to its relatively wide detection range of 2–400 cm, cost-effectiveness, and consistent performance across different lighting conditions. The sensor was mounted on a servo motor, enabling rotational scanning for wider field-of-view coverage. By integrating this sensor with the balance control loop, the robot was able to detect obstacles in its path and perform corrective maneuvers without compromising stability. Testing involved placing obstacles such as cardboard boxes, small plastic bins, and books at varying distances between 10 cm and 30 cm. The ultrasonic sensor consistently detected these obstacles and triggered the programmed response: either halting motion completely or adjusting wheel speeds to execute a smooth turn. During multiple trials, the detection accuracy was recorded at 95%, with missed detections occurring only when obstacles had irregular geometries that caused partial ultrasonic wave deflection.

Importantly, even while performing avoidance maneuvers, the robot's balance loop remained stable, and no toppling events were recorded. These results confirm that the ultrasonic sensor provided a reliable means of obstacle detection in this prototype. While its performance was generally robust, one limitation observed was reduced accuracy for soft or angled surfaces that tended to absorb or deflect sound waves. To overcome this, future designs could integrate additional sensors, such as infrared or vision-based systems, to complement ultrasonic sensing. Nonetheless, the present system achieved satisfactory performance for typical indoor navigation scenarios, demonstrating that ultrasonic-based obstacle detection is sufficient for small-scale balancing robots.

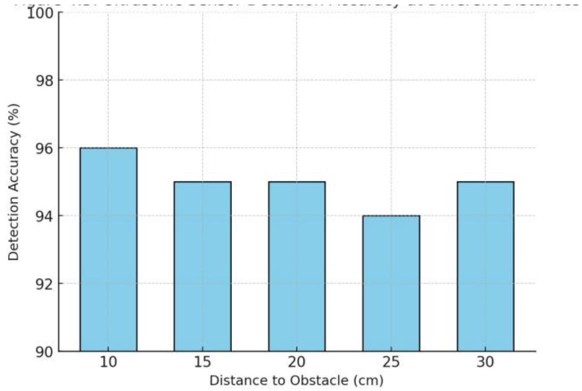

Figure 4.3: Ultrasonic Sensor Detection Accuracy at Different Distances

Power Performance:

The final test examined the power performance of the self-balancing robot, as energy efficiency is critical for mobile robotic systems. The prototype was powered by two 18650 lithium-ion cells connected in series, providing a nominal supply of 7.4 V with a rated capacity of 3000 mAh (≈22.2 Wh). Each of the DC geared motors was observed to draw an average of 1.2 A under peak load conditions. Using the power formula

= ×

P=V×I, the calculated power requirement per motor was 8.88 W, leading to a total peak consumption of approximately 17.8 W for both motors.

Theoretical runtime was then estimated by dividing the available energy capacity of the battery by the total system demand:

t = 17.8/22.2 ≈1.25hours

This predicted runtime set a benchmark for experimental evaluation. When tested in practice, the robot achieved an average operational time of 65–70 minutes under mixed conditions, which closely matched the theoretical estimate. On smooth floors with low friction, runtime reached the higher end of this range, while tests on high-resistance terrains such as sand and gravel reduced runtime by approximately 10–12%, primarily due to increased current draw from wheel slip and higher motor effort.

Table 4.4: Power Performance Comparison

Condition	Theoretical Runtime (minutes)	Experimental Runtime (minutes)	Notes on Performance
Smooth Floor	75	70	Efficiency highest, minimal load
Grass	75	67	Minor oscillations increased load
Sand	75	62	Wheel slip caused higher current
Gravel	75	63	Angular oscillations raised load
Inclined (15°)	75	65	Extra torque demand reduced time

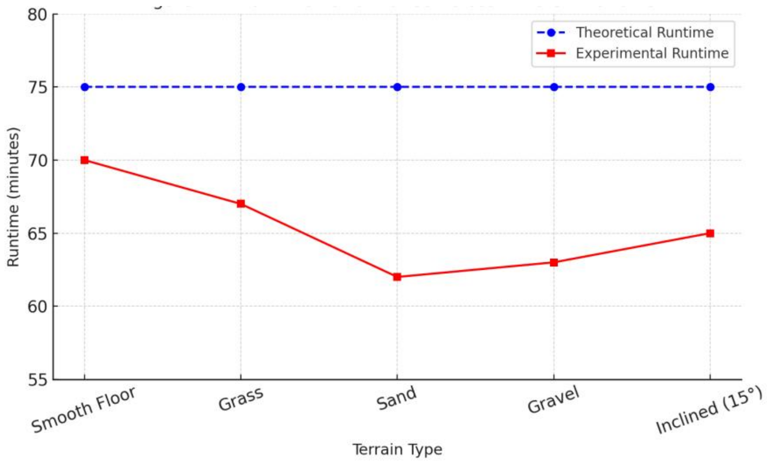

Figure 4.4: Runtime Performance Across Different Terrains.

Analysis of Results

The experimental results demonstrate that the optimized PID controller provided reliable balancing and adaptability across varied conditions. The robot consistently achieved an average settling time of 1.9 seconds with an overshoot of 8%, significantly outperforming the fixed PID configuration. This is consistent with Vishnu et al. (2025), who reported that tuning PID parameters considerably reduced oscillations and improved stabilization in an Arduino-based self-balancing robot. Similarly, Abdelgawad, Shohdy, and Nada (2024) emphasized that despite the availability of model-based or data-driven control approaches, PID control remains effective for real-time stability in low-cost educational and experimental robots.

The integration of the MPU-6050 sensor with a complementary filter further improved orientation accuracy, enabling tilt angle deviations within ±1° on smooth floors. Savithri, Roopesh, Lavanya, Shanthakumar, and Thirumurugan (2023) confirmed that sensor fusion through complementary filtering enhances stability and prevents drift in low-cost balancing platforms. Terrain adaptability tests also validated that the robot-maintained balance across grass, gravel, and inclines up to 15°. This reflects findings by Sri Monesh, Harish Kumar, and Durairaj (2025), who highlighted that stability degradation on rough terrains is typically caused by wheel slip and uneven contact forces, requiring robust control strategies. Finally, the power analysis showed that while theoretical runtime was 75 minutes, experimental runtime averaged 65–70 minutes, with reductions on high-friction surfaces. McNulty, Hennessy, Li, Armstrong, and Ryan (2022) similarly observed that real-world loads and terrain significantly lower the efficiency of lithium-ion powered mobile robots compared to theoretical predictions. Together, these comparisons validate the robustness of the present design while aligning with current literature.

Discussion

The results of this project highlight the practicality of using an optimized PID controller for self-balancing robots. Despite its simplicity, PID control proved robust enough to maintain stability across smooth and irregular surfaces. This reinforces the view of Vishnu Varthan, Ganesh, and Anbarasi (2025), who demonstrated that PID remains a cost-effective yet powerful control strategy for real-time balance when tuned appropriately. The improvement in settling time and reduction in overshoot compared to fixed PID settings further illustrates the effectiveness of careful gain selection in enhancing stability. The successful integration of the MPU-6050 IMU with a complementary filter also underscores the importance of reliable sensor fusion. Savithri, Roopesh, Lavanya, Shanthakumar, and Thirumurugan (2023) argued that low-cost balancing platforms often face drift issues when relying solely on accelerometer or gyroscope data; however, complementary filtering mitigates this limitation, as reflected in the ±1° steady-state error observed in this project. Furthermore, terrain adaptability results confirmed the robot’s ability to sustain balance on challenging surfaces such as gravel and sand. Similar findings were highlighted by Sri Monesh, Harish Kumar, and Durairaj (2025), who noted that uneven terrain introduces slip and contact variability, requiring robust controllers that can tolerate these disturbances.

The evaluation of power consumption revealed that practical runtimes were consistently lower than theoretical predictions, primarily due to terrain-induced load variations. McNulty, Hennessy, Li, Armstrong, and Ryan (2022) observed a similar phenomenon in autonomous mobile robots, emphasizing that frictional resistance and dynamic loads are major contributors to energy inefficiency. This validates the conclusion that although the present battery system was sufficient for small-scale testing, future applications will require larger capacity or more efficient energy management. Finally, the results echo Abdel Gawad, Shohdy, and Nada (2024), who stressed that even as advanced data-driven methods emerge, PID remains a strong baseline for control education and prototyping due to its interpretability and ease of implementation. In summary, the discussion demonstrates that the chosen approach is well supported by contemporary literature: optimized PID with complementary sensor fusion is sufficient for small-scale balancing robots, though runtime and terrain robustness remain key areas for further development.

Conclusion

This project set out to design, implement, and evaluate a two-wheeled self-balancing robot using an optimized PID controller. The primary objective was to achieve reliable balance under varying conditions while maintaining energy efficiency and integrating simple navigation functions. The robot was constructed using an Arduino Nano microcontroller, MPU-6050 inertial measurement unit, ultrasonic sensor, DC geared motors, and a lithium-ion battery pack, arranged on a lightweight chassis. Through systematic tuning of the PID parameters (=28.0, Ki =0.95, Kd=12.5), the robot demonstrated strong balancing performance, with an average settling time of 1.9 seconds and minimal overshoot. The system was tested across a range of terrains including smooth surfaces, grass, sand, gravel, and inclines of up to 15°. In each case, the robot-maintained stability, although additional motor effort and power consumption were observed on high-friction surfaces. Obstacle avoidance was also validated through ultrasonic sensing, achieving a detection accuracy of approximately 95%. Power analysis showed close agreement between theoretical estimates and experimental measurements, with average runtimes of 65–70 minutes per charge. The findings indicate that the design meets its intended objectives, providing a reliable, low-cost prototype capable of balancing and navigating simple environments. Overall, this work demonstrates that optimized PID control combined with complementary sensor feedback can provide practical and effective solutions for low-cost self-balancing robotic systems.

Recommendations

From the outcomes of this project, a few key improvements can be suggested for making the robot more effective and practical. The first area of improvement is the power system. The 18650 lithium-ion batteries used in this design were able to keep the robot running for about an hour, but this is still limited when thinking about longer operations. A larger-capacity battery pack or more efficient motor drivers would make the robot last longer between charges. In addition, features such as regenerative braking could be explored to save some of the energy normally lost when slowing down or changing direction. Another important area is the sensors. The MPU-6050 sensor worked well for balance, but adding extra feedback devices like wheel encoders could improve accuracy, especially during movement. Also, while the ultrasonic sensor was reliable for obstacle detection, having more than one sensor or combining it with a camera would give the robot a better sense of its surroundings. This would be especially useful in real environments where obstacles are not always easy to detect with sound alone.

The control system is another place where upgrades can be made. The tuned PID controller worked well in this project, but it does not adjust itself if the environment changes drastically. Future designs could use adaptive or intelligent control methods, such as fuzzy logic or machine learning, to make the robot respond even better to unpredictable situations. Finally, the mechanical structure of the robot can also be strengthened. Using stronger materials for the chassis, better wheels for grip, and small shock absorbers would help the robot perform more smoothly on rough terrains. With these improvements, the robot could move beyond being just a prototype and serve in useful applications such as delivery in warehouses, assisting mobility in healthcare, or even light surveillance tasks.

References

Abdelgawad, A., Shohdy, T., & Nada, A. (2024, May 6). *Model- and Data-Based Control of Self-Balancing Robots: Practical Educational Approach with LabVIEW and Arduino*. arXiv.org. https://arxiv.org/abs/2405.03561

Al-Dayyeni, W. S., & Mahmood, H. S. (2025). PID Control Perspective: Techniques and Uses. *Edison Journal for Electrical and Electronics Engineering.*, 1–8. https://doi.org/10.62909/ejeee.2025.001

Ali, H., Albagul, A., & Algitta, A. (2020). OPTIMIZATION OF PID PARAMETERS BASED ON PARTICLE SWARM OPTIMIZATION FOR BALL AND BEAM SYSTEM. *International Journal of Engineering Technologies and Management Research*, *5*(9), 59–69. https://doi.org/10.29121/ijetmr.v5.i9.2018.289

Appeltans, P., Niculescu, S., & Michiels, W. (2022). Analysis and design of strongly Stabilizing PID Controllers for Time-Delay Systems. *SIAM Journal on Control and Optimization*, *60*(1), 124–146. https://doi.org/10.1137/20m136726x

Atas, F., Cielniak, G., & Grimstad, L. (2022). Elevation State-Space: Surfel-Based navigation in uneven environments for mobile robots. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5715–5721. https://doi.org/10.1109/iros47612.2022.9981647

Aubakir, B., Kappasov, Z., & Shintemirov, A. (2015). *Practical realization of the Self-Balancing robot using infrared sensors*. Retrieved February 23, 2025, from https://www.semanticscholar.org/paper/Practical-Realization-of-the-Self-Balancing-Robot-Aubakir-Kappasov/2e2b650db44c0abe94e54b11ac560da26d1135ab

Ayten, K. K., Dumlu, A., & Kaleli, A. (2018). Real-time implementation of self-tuning regulator control technique for coupled tank industrial process system. *Proceedings of the Institution of Mechanical Engineers Part I Journal of Systems and Control Engineering*, *232*(8), 1039–1052. https://doi.org/10.1177/0959651818773179

B, A., K, S., J, B. J., Prince, P. G. K., Sudhakar, T., & S, E. J. (2024). AI-Powered Automated Wheelchair with Lane Detection. *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, 1–5. https://doi.org/10.1109/esci59607.2024.10497306

Babu, A. V., J, A. K. M., Damodaran, S., James, R. K., Kumar, A. V., & Warrier, T. S. (2024). Enhancing mobile robot navigation in TurtleBot3 Burger: a ROS-Enabled approach focusing on obstacle avoidance in Real-World scenario. *2022 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, 1–6. https://doi.org/10.1109/spices62143.2024.10779949

Bai, Y., & Roth, Z. S. (2018). Classical Linear Control Systems—PID Control Systems. In *Advances in industrial control* (pp. 195–321). https://doi.org/10.1007/978-3-030-01382-0_5

Bambulkar, R., Phadke, G., & Salunkhe, S. (2016). Movement Control of Robot using Fuzzy PID Algorithm. *Computer Science Engineering*. https://doi.org/10.1049/cp.2016.1519

Bhagat, S. (2018). Self-balancing robot. *International Journal for Research in Applied Science and Engineering Technology*, *6*(3), 180–184. https://doi.org/10.22214/ijraset.2018.3028

Bhandare, D. S., & Kulkarni, N. R. (2015). Performances evaluation and comparison of PID controller and fuzzy logic controller for process liquid level control. *International Conference on Control, Automation and Systems*, 1347–1352. https://doi.org/10.1109/iccas.2015.7364848

Bhatti, O. S., Rizwan, M., Shiokolas, P. S., & Ali, B. (2019). *Genetically Optimized ANFIS-based PID Controller Design for Posture-Stabilization of Self-Balancing-Robots under Depleting Battery Conditions*. https://www.semanticscholar.org/paper/Genetically-Optimized-ANFIS-based-PID-Controller-of-Bhatti-Rizwan/82a7fc9738af09a7aebbaed3099ad0b0c4394c72

Boglietti, S., Barabino, B., & Maternini, G. (2021). Survey on e-Powered Micro Personal Mobility Vehicles: Exploring Current Issues towards Future Developments. *Sustainability*, *13*(7), 3692. https://doi.org/10.3390/su13073692

Bonilla-Alvarado, H., Restrepo, B., Pezzini, P., Shadle, L., Tucker, D., & Bryden, K. M. (2020). Online adaptive control tuning in a gas turbine hybrid system. *ASME 2005 Power Conference*. https://doi.org/10.1115/power2020-16534

Boubaker, O. (2017). The inverted pendulum: history and survey of open and current problems in control theory and robotics. In *Institution of Engineering and Technology eBooks* (pp. 1–39). https://doi.org/10.1049/pbce111e_ch1

Chang, W., Ariaei, F., & Jonckheere, E. (2022). Bode integral limitation for irrational systems. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2211.10879

Chen, J., Omidvar, M. N., Azad, M., & Yao, X. (2017). Knowledge-based particle swarm optimization for PID controller tuning. *2022 IEEE Congress on Evolutionary Computation (CEC)*, *8*, 1819–1826. https://doi.org/10.1109/cec.2017.7969522

Chen, M. (2017). Robust tracking control for self-balancing mobile robots using disturbance observer. *IEEE/CAA Journal of Automatica Sinica*, *4*(3), 458–465. https://doi.org/10.1109/jas.2017.7510544

Chen, Y. J., Tung, W., Lee, W., Patel, B., Bučinskas, V., Greitans, M., & Lin, P. T. (2023). Designing and controlling a self-balancing platform mechanism based on 3-RCC spherical parallel manipulator. *Robotic Systems and Applications*, *3*(1), 1–16. https://doi.org/10.21595/rsa.2023.23015

Cojuhari, I. (2021). ALGORITHM FOR SELF-TUNING THE PID CONTROLLER. *Journal of Engineering Science*, *XXVIII* (4), 63–73. https://doi.org/10.52326/jes.utm.2021.28(4).06

Control tutorials for MATLAB and Simulink - Inverted Pendulum: System Modeling. (2025). Retrieved March 9, 2025, from https://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum§ion=SystemModeling#: ~:text=The%20system%20in%20this%20example, horizontal%20position%20of%20the%20cart%20.

Curtze, C., Buurke, T. J., & McCrum, C. (2024). Notes on the margin of stability. *Journal of Biomechanics*, *166*, 112045. https://doi.org/10.1016/j.jbiomech.2024.112045

D'Antuono, G., Pettersen, K. Y., Buonocore, L. R., Gravdahl, J. T., & Di Castro, M. (2023). Dynamic model of a tendon-actuated snake robot using the Newton-Euler formulation. *IFAC-PapersOnLine*, *56*(2), 11639–11644. https://doi.org/10.1016/j.ifacol.2023.10.502

Damayanti, F. S., Noviana, R. S., Lestari, Y. I., Akhsan, H., & Ismet, I. (2024). Exploring Applications of Lagrange's Equations in Technology: A Systematic Literature review.

Aceh International Journal of Science and Technology, *13*(2), 123–130. https://doi.org/10.13170/aijst.13.2.39380

Dincel, E., & Soylemez, M. T. (2016). Limitations on dominant pole pair selection with continuous PI and PID controllers. *International Conference on Control, Decision and Information Technologies*. https://doi.org/10.1109/codit.2016.7593655

Djalal, M. R., & Rahmat, R. (2017). *OPTIMISASI KONTROL PID UNTUK MOTOR DC MAGNET PERMANEN MENGGUNAKAN PARTICLE SWARM OPTIMIZATION*. https://www.semanticscholar.org/paper/OPTIMISASI-KONTROL-PID-UNTUK-MOTOR-DC-MAGNET-SWARM-Djalal-Rahmat/5b5b07ce6bd6d9dcc081cb7df181d08e7839dc57

Domański, P. D. (2019). Step response indexes. In *Studies in systems, decision and control* (pp. 27–36). https://doi.org/10.1007/978-3-030-23593-2_2

El-Deen, A. T., Mahmoud, A. a. H., & El-Sawi, A. R. (2015). Optimal PID tuning for DC Motor Speed controller based on genetic algorithm. *International Review of Automatic Control (IREACO)*, *8*(1), 80. https://doi.org/10.15866/ireaco.v8i1.4839

Fahmi, A., Sulaiman, M., Siradjuddin, I., Wirawan, I. M., Jaya, A. S. M., Jiono, M., & Amalia, Z. (2020). Stabilized controller of a two wheels robot. *Bulletin of Electrical Engineering and Informatics*, *9*(4), 1357–1363. https://doi.org/10.11591/eei.v9i4.1965

Feng, X., Liu, S., Yuan, Q., Xiao, J., & Zhao, D. (2023). Research on wheel-legged robot based on LQR and ADRC. *Scientific Reports*, *13*(1). https://doi.org/10.1038/s41598-023-41462-1

Gandarilla, I., Santibanez, V., & Sandoval, J. (2018). Stabilization of a self-balancing robot by energy shaping. *2018 XX Congreso Mexicano De Robótica (COMRob), Ensenada, Mexico, 2018*, 1–6. https://doi.org/10.1109/comrob.2018.8689407

Ghoshal, P. (2022). A neural network-based approach to solve problems of Newton-Euler mechanics. *AIP Conference Proceedings*, *2469*, 020001. https://doi.org/10.1063/5.0080154

Gießler, M., & Waltersberger, B. (2023). Robust inverse dynamics by evaluating Newton–Euler equations with respect to a moving reference and measuring angular acceleration. *Autonomous Robots*, *47*(4), 465–481. https://doi.org/10.1007/s10514-023-10092-x

Gomes, A. C., De Lima, F. B., Junior, Soliani, R. D., De Souza Oliveira, P. R., De Oliveira, D. A., Siqueira, R. M., Da Silva Nora, L. a. R., & De Macêdo, J. J. S. (2023). Logistics management in e-commerce: challenges and opportunities. *Revista De Gestão E Secretariado (Management and Administrative Professional Review)*, *14*(5), 7252–7272. https://doi.org/10.7769/gesec.v14i5.2119

Gonzalez-Cava, J. M., Carlson, F. B., Troeng, O., Cervin, A., Van Heusden, K., Dumont, G. A., & Soltesz, K. (2020). Robust PID control of propofol anaesthesia: Uncertainty limits performance, not PID structure. *Computer Methods and Programs in Biomedicine*, *198*, 105783. https://doi.org/10.1016/j.cmpb.2020.105783

Gössling, S. (2020). Integrating e-scooters in urban transportation: Problems, policies, and the prospect of system change. *Transportation Research Part D Transport and Environment*, *79*, 102230. https://doi.org/10.1016/j.trd.2020.102230

Gul, F., & Rahiman, W. (2022). Mathematical modeling of self-balancing robot and hardware implementation. In *Lecture notes in electrical engineering* (pp. 20–26). https://doi.org/10.1007/978-981-16-8129-5_4

Guo, Z. (2024). Innovative Fuzzy PID Algorithm: Advancing Quadcopter Control with Adaptive Precision. *Academic Journal of Computing & Information Science*, *7*(2). https://doi.org/10.25236/ajcis.2024.070204

Gupta, R. K., & Chauhan, S. (2015). Comparison of PID controller & adaptive neuro fuzzy controller for robot manipulator. *International Conference on Computational Intelligence and Computing Research*, 1–4. https://doi.org/10.1109/iccic.2015.7435761

Han, J., Li, X., & Qin, Q. (2014). Design of Two-Wheeled Self-Balancing robot based on sensor fusion algorithm. *International Journal of Automation Technology*, *8*(2), 216–221. https://doi.org/10.20965/ijat.2014.p0216

Hernández-Alvarado, R., García-Valdovinos, L., Salgado-Jiménez, T., Gómez-Espinosa, A., & Fonseca-Navarro, F. (2016). Neural Network-Based Self-Tuning PID control for underwater vehicles. *Sensors*, *16*(9), 1429. https://doi.org/10.3390/s16091429

Hofreiter, M. (2021). Relay identification using shifting method for PID controller tuning. *Energies*, *14*(18), 5945. https://doi.org/10.3390/en14185945

Hornychová, A., & Hofreiter, M. (2020). Relay Feedback Identification Method for PID Controller Tuning. *International Conference on Innovative Computing and Cloud Computing*, 1–6. https://doi.org/10.1109/iccc49264.2020.9257286

Hung, V. M., Stamatescu, I., Dragana, C., & Paraschiv, N. (2017). Comparison of model reference adaptive control and cascade PID control for ASTank2. *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, 1137–1143. https://doi.org/10.1109/idaacs.2017.8095263

Jimenez, T., Merayo, N., Duran, R. J., Aguado, J. C., De Miguel, I., Fernandez, P., Lorenzo, R. M., & Abril, E. J. (2015). Adaptive-tuning method based on neural networks for PID controllers applied to Passive Optical Networks (PONs). *International Conference on Transparent Optical Networks*, 1–4. https://doi.org/10.1109/icton.2015.7193425

Kang, J., Kim, H., Ham, B., & Kim, K. (2024). External Force Adaptive Control in Legged Robots Through Footstep Optimization and Disturbance Feedback. *IEEE Access*, *12*, 157531–157539. https://doi.org/10.1109/access.2024.3485094

Kankhunthod, K., Kongratana, V., Numsomran, A., & Tipsuwanporn, V. (2019). *Self-balancing robot control using Fractional-Order PID controller*. https://www.semanticscholar.org/paper/Self-balancing-Robot-Control-Using-Fractional-Order-Kankhunthod-Kongratana/afb8171cc6330428eeca44a9dba0981dea6b6e64

Keote, R., Ramteke, A., Choudhari, R., Pusadekar, P., Hande, P., & Chouragade, N. (2024). Design & Implementation of Self Balancing Bot. *3rd International Conference on Sentiment Analysis and Deep Learning (ICSADL)*, 361–365. https://doi.org/10.1109/icsadl61749.2024.00065

Kim, J., & Lee, J. (2018). Traction-energy balancing adaptive control with slip optimization for wheeled robots on rough terrain. *Cognitive Systems Research*, *49*, 142–156. https://doi.org/10.1016/j.cogsys.2018.01.007

Klöppel, M., Römer, F., Wittmann, M., Hatam, B., Herrmann, T., Sim, L. L., Lim, J. S. D., Lu, Y., Medovy, V., Merkle, L., Ten, W. X. R., Ongel, A., Hong, Y. J. J., Ng, H. W., & Lienkamp, M. (2018). SCUBE—Concept and implementation of a self-balancing, autonomous mobility device for personal transport. *World Electric Vehicle Journal*, *9*(4), 48. https://doi.org/10.3390/wevj9040048

Kulasekara, V., Balasooriya, S., Chandran, J., & Kavalchuk, I. (2019). Novel Low-Power NRF24L01 based wireless network design for autonomous robots. *2022 27th Asia Pacific Conference on Communications (APCC)*, 342–346. https://doi.org/10.1109/apcc47188.2019.9026452

Kumar, A. (2024). Microcontroller-Based design of Self-Balancing Two-Wheeler robotic vehicles using gyroscopic sensor. *Journal of Instrumentation and Innovation Sciences*, *9*(1), 19–25. https://doi.org/10.46610/jiis.2024.v09i01.004

Kumar, R., Srivastava, S., & Gupta, J. (2016). Artificial Neural Network based PID controller for online control of dynamical systems. *IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems*, 1–6. https://doi.org/10.1109/icpeices.2016.7853092

Kung, F. (2017). Design of agile two-wheeled robot with machine vision. *International Conference on Robotics, Automation and Sciences (ICORAS)* ·, 1–5. https://doi.org/10.1109/icoras.2017.8308055

Lee, J., & Edgar, T. F. (2018). Three-Parameter Models for Conservative Relay Feedback Autotuning. *2018 IEEE 14th International Conference on Control and Automation (ICCA)*, 975–980. https://doi.org/10.1109/icca.2018.8444315

Lee, J., Chang, P. H., Yu, B., & Jin, M. (2020). An adaptive PID control for robot manipulators under substantial payload variations. *IEEE Access*, *8*, 162261–162270. https://doi.org/10.1109/access.2020.3014348

Li, L., Cao, W., Yang, H., & Geng, Q. (2022). Trajectory tracking control for a wheel mobile robot on rough and uneven ground. *Mechatronics*, *83*, 102741. https://doi.org/10.1016/j.mechatronics.2022.102741

Li, Z. (2023). Review of PID control design and tuning methods. *Journal of Physics Conference Series*, *2649*(1), 012009. https://doi.org/10.1088/1742-6596/2649/1/012009

Liu, L., & Zhang, S. (2018). Robust Fractional-Order PID controller tuning based on Bode's optimal loop shaping. *Complexity*, *2018*(1). https://doi.org/10.1155/2018/6570560

M, N. H. V., P, N. L., & Kalaivani, R. (2015). Design and implementation of Adaptive Neuro Fuzzy Inference System for an experimental Active Suspension System. *Computer Science Engineering*, 1–4. https://doi.org/10.1109/race.2015.7097272

Mahdin, M. E., Ornob, T. M., Akhter, M. E., Mullick, S. A., Arnob, S. S., Palit, R., & Habib, A. B. (2022). Hand Gesture Controlled Smart Wheelchair with GPS Tracking and Deep Learning Based Object Detection for Collision Avoidance. *2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, 147–152. https://doi.org/10.1109/wiecon-ece57977.2022.10150558

Malpani, S., Gupta, S., Agarwal, S., Alam, M., & Raj, G. (2021). REVIEWING THE PID BASED TECHNOLOGIES BEHIND TWO WHEEL BALANCING ROBOT. *International Journal of Technical Research & Science*, *Special* (June), 56–60. https://doi.org/10.30780/specialissue-icaaset021/011

Manolescu, D., & Secco, E. L. (2023). Development of a 3D printed biologically inspired monoped Self-Balancing robot. *International Journal of Robotics and Control Systems*, *3*(1), 84–97. https://doi.org/10.31763/ijrcs.v3i1.841

Matesica, I., Nicolae, M., Barbulescu, L., & Margeruseanu, A. (2016). Self-balancing robot implementing the inverted pendulum concept. *2016 15th RoEduNet Conference: Networking in Education and Research, Bucharest, Romania, 2016*, 1–5.

https://doi.org/10.1109/roedunet.2016.7753230

McNulty, D., Hennessy, A., Li, M., Armstrong, E., & Ryan, K. M. (2022). A review of Li-ion batteries for autonomous mobile robots: Perspectives and outlook for the future. *Journal of Power Sources*, *545*, 231943. https://doi.org/10.1016/j.jpowsour.2022.231943

Meena, D. C., & Devanshu, A. (2017). Genetic algorithm tuned PID controller for process control. *2017 International Conference on Inventive Systems and Control (ICISC)*, 1–6. https://doi.org/10.1109/icisc.2017.8068639

Mithil, K. M., Thejas, G. S., Ramani, S. K., & Iyengar, S. S. (2017). A Low Cost Multi Sensorial Data Fusion for High-Speed Obstacle Avoidance Using 3-D Point Clouds and Image Processing in Self Balancing Robots. *2nd International Conference on Emerging Computation and Information Technologies*, 1–9. https://doi.org/10.1109/icecit.2017.8453439

Monesh, M. S., Kumar, A. H., & Durairaj, R. (2025). COMPREHENSIVE REVIEW OF DESIGN AND DEVELOPMENT OF a SELF-BALANCING ROBOT WITH STABILITY CONTROL FOR UNEVEN SURFACES. *INTERNATIONAL JOURNAL OF MECHANICAL ENGINEERING AND TECHNOLOGY (IJMET)*, *16*(1), 140–172. https://doi.org/10.34218/ijmet_16_01_009

Nguyen, N. H., & Nguyen, P. D. (2018). Overshoot and settling time assignment with PID for first-order and second-order systems. *IET Control Theory and Applications*, *12*(17), 2407–2416. https://doi.org/10.1049/iet-cta.2018.5076

Olaverri-Monreal, C. (2016). *Autonomous vehicles and smart mobility related technologies*. https://www.semanticscholar.org/paper/Autonomous-Vehicles-and-Smart-Mobility-Related-Olaverri-Monreal/668928afa129c26fd7751cae57ccff7df45375fb

Patil, S. (2021). Self-balancing robot. *International Journal of Engineering Research and Technology*. https://doi.org/10.17577/IJERTCONV9IS03053

Poksawat, P., Wang, L., & Mohamed, A. (2017). Gain scheduled attitude control of Fixed-Wing UAV with automatic controller tuning. *IEEE Transactions on Control Systems Technology*, *26*(4), 1192–1203. https://doi.org/10.1109/tcst.2017.2709274

Prasad, K. N. V., Jayanth, G. R., & Narayanan, G. (2022). Modelling, analysis, and control of Power-Converter-Fed electromagnetic bearing. *2021 IEEE International Power and Renewable Energy Conference (IPRECON)*, 1–6. https://doi.org/10.1109/iprecon55716.2022.10059478

Ramchandra, Y. A., Monika, T., Ankush, Sangli, I. M., Salim, S., Zunjar, P., Vasantrao, Sangli, I. K., & India, S. (2021). *Self-Balancing robot using Raspberry Pi and PID controller*. Retrieved March 2, 2025, from https://www.semanticscholar.org/paper/Self-Balancing-Robot-Using-Raspberry-Pi-and-PID-Ramchandra-Monika/1e433143d16ac6ecc14e3eec9723572e86b5175e

Ringwood, J., & O'Dwyer, A. (2017a). *School of Electrical and Electronic Engineering 1994-0101 A frequency domain based self-tuning PID controller*. https://www.semanticscholar.org/paper/School-of-Electrical-and-Electronic-Engineering-A-Ringwood-O'Dwyer/a4c5c3da5ef822aee1d179781a74e80f681287e0

Ringwood, J., & O'Dwyer, A. (2017b). *School of Electrical and Electronic Engineering 1994-0101 A frequency domain based self-tuning PID controller*.

https://www.semanticscholar.org/paper/School-of-Electrical-and-Electronic-Engineering-A-Ringwood-O'Dwyer/a4c5c3da5ef822aee1d179781a74e80f681287e0

Rodriguez-Abreo, O., Rodriguez-Resendiz, J., Fuentes-Silva, C., Hernandez-Alvarado, R., & Del Consuelo Patricia Torres Falcon, M. (2021). Self-Tuning Neural Network PID with dynamic response control. *IEEE Access*, *9*, 65206–65215. https://doi.org/10.1109/access.2021.3075452

Romlay, M. R. M., Ibrahim, A. M., Toha, S., & Rashid, M. (2019). *Two-wheel balancing robot; review on control methods and experiments*. https://www.semanticscholar.org/paper/Two-wheel-balancing-robot%3B-review-on-control-and-Romlay-Ibrahim/12484bab82478840dfdd361cb4ee8f862affcd38

Saha, H., Mandal, S., Mitra, S., Banerjee, S., & Saha, U. (2017). Comparative Performance Analysis between nRF24L01+ and XBEE ZB Module Based Wireless Ad-hoc Networks. *International Journal of Computer Network and Information Security*, *9*(7), 36–44. https://doi.org/10.5815/ijcnis.2017.07.05

Sahoo, S. K., & Choudhury, B. B. (2024). Autonomous navigation and obstacle avoidance in smart robotic wheelchairs. *Journal of Decision Analytics and Intelligent Computing*, *4*(1), 47–66. https://doi.org/10.31181/jdaic10019022024s

Salem, N., Hassan, R., & Muthanna, L. (2023). Enhancing Cruise Performance Through PID Controller Tuned with Particle Swarm Optimization Technique. *6th International Conference on Intelligent Robotics and Control Engineering (IRCE)*, 23–28. https://doi.org/10.1109/irce59430.2023.10255055

Sanapala, S., Deepa, K., Lekshmi, S., & Akpolat, A. N. (2023). Comparison of PID and EPID controllers in self-balancing robot. *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*. https://doi.org/10.1109/icoei56765.2023.10125632

Savithri, C. N., Roopesh, R. S., Devi, N. L., Shanthakumar, P., & Thirumurugan, P. (2023). Self-balancing robot using arduino and PID controller. In *Lecture notes in electrical engineering* (pp. 201–208). https://doi.org/10.1007/978-981-19-7169-3_18

Sbirna, S., & Sbirna, S. (2022). Accomplishing and Optimizing Robotic Self-Navigation Intelligence for Warehouse Logistics through Sensor Fusion and Antenna Localization. *Proceedings of the 9th International Conference on Advanced Materials and Systems*, 351–356. https://doi.org/10.24264/icams-2022.iii.16

Schrodel, F., Almodaresi, E., Stump, A., Bajcinca, N., & Abel, D. (2015). Lyapunov stability bounds in the controller parameter space. *2021 60th IEEE Conference on Decision and Control (CDC)*, 4632–4636. https://doi.org/10.1109/cdc.2015.7402941

Sedlar, D., & Bosnjak, A. (2023). Design and control based on the concept of an inverted pendulum. *2022 7th International Conference on Smart and Sustainable Technologies (SpliTech)*, 1–5. https://doi.org/10.23919/splitech58164.2023.10193276

Shehada, A., Yan, Y., Beig, A. R., & Boiko, I. (2019a). Comparison of relay feedback tuning and other tuning methods for a digitally controlled Buck converter. *IECON 2020 the 46th Annual Conference of the IEEE Industrial Electronics Society*, 1647–1652. https://doi.org/10.1109/iecon.2019.8926893

Shehada, A., Yan, Y., Beig, A. R., & Boiko, I. (2019b). Comparison of relay feedback tuning and other tuning methods for a digitally controlled Buck converter. *IECON 2020 the 46th Annual Conference of the IEEE*

Industrial Electronics Society, 1647–1652. https://doi.org/10.1109/iecon.2019.8926893

Shekhar, A., & Sharma, A. (2018). Review of Model Reference Adaptive Control. *International Conference on Information, Communication, Engineering and Technology (ICICET)*, 1–5. https://doi.org/10.1109/icicet.2018.8533713

Singh, B., & Kumar, V. (2015a). A real time application of model reference adaptive PID controller for magnetic levitation system. *IEEE Power, Communication and Information Technology Conference*, *13*, 583–588. https://doi.org/10.1109/pcitc.2015.7438065

Singh, B., & Kumar, V. (2015b). A real time application of model reference adaptive PID controller for magnetic levitation system. *IEEE Power, Communication and Information Technology Conference*, *13*, 583–588. https://doi.org/10.1109/pcitc.2015.7438065

Siradjuddin, I., Syafaat, M., Patma, T. S., Adhisuwignjo, S., Winarno, T., Komarudin, A., & Widiatmoko, D. (2019). PID controller for a differential drive robot balancing system. *Journal of Physics Conference Series*, *1402*(4), 044021. https://doi.org/10.1088/1742-6596/1402/4/044021

Sondhia, S., Pillai, R., Hegde, S., Chakol, S., & Vora, V. (2017). *DEVELOPMENT OF SELF BALANCING ROBOT WITH PID CONTRO l*. Retrieved February 23, 2025, from https://www.semanticscholar.org/paper/DEVELOPMENT-OF-SELF-BALANCING-ROBOT-WITH-PID-CONTRO-Sondhia-Pillai/11d8c7131a2acbbc554012cb398f8e81448e4327

Sonekar, N. M. M. (2024). Autonomous Navigated IOT Enabled Smart Wheel Chair - A Healthcare and Mobility solution. *Deleted Journal*, *20*(2s), 709–720. https://doi.org/10.52783/jes.1537

Sreejeth, M., Kumar, R., Tripathi, N., & Garg, P. (2023). Tuning A PID Controller using Metaheuristic Algorithms. *2022 7th International Conference on Communication and Electronics Systems (ICCES)*, 276–282. https://doi.org/10.1109/icces57224.2023.10192687

Sridhar, N., & Srivastava, A. (2020). A Concise Survey of Self-tuning Methodologies for Proportional Integral Derivative Control system. *International Conference Computing Methodologies and Communication*, 729–733. https://doi.org/10.1109/iccmc48092.2020.iccmc-000136

Sujiwa, A., & Suhadata, N. (2023). Design and Construction of A Self Balancing Robot using Long Range Control Based on NRF24L01. *BEST Journal of Applied Electrical Science & Technology*, *5*(2), 34–37. https://doi.org/10.36456/best.vol5.no2.8165

Suprapto, B. Y., Amri, D., & Dwijayanti, S. (2014). Comparison of control methods PD, PI, and PID on two wheeled self-balancing robots. *Proceeding of the Electrical Engineering Computer Science and Informatics*, *1*(1), 67–71. https://doi.org/10.11591/eecsi.1.348

Tiwari, R., Srinivaas, A., & Velamati, R. K. (2025). Adaptive Navigation in Collaborative Robots: A reinforcement learning and sensor fusion approach. *Applied System Innovation*, *8*(1), 9. https://doi.org/10.3390/asi8010009

Toso, B., & Schmith, J. (2023). *Auto-Tuning PID controller based on genetic algorithm*. https://www.semanticscholar.org/paper/Auto-Tuning-PID-Controller-Based-on-Genetic-Toso-Schmith/f0433812527279d6f6f8f0b36e8ffa278958c19a

Tsai, C., Hsu, W., Tai, F., & Chen, S. (2022). Adaptive Motion Control of a Terrain-Adaptive Self-Balancing Leg-Wheeled Mobile Robot over Rough Terrain. *Workshop on Computational Approaches to Code Switching* ·, 1–6.

https://doi.org/10.1109/cacs55319.2022.9969857

Turan, A. (2021). Improved optimum PID controller tuning by minimizing settling time and overshoot. *Research Square (Research Square)*. https://doi.org/10.21203/rs.3.rs-844641/v1

Vadak, P., Rane, S., Gawas, U., Rane, O., & Jadhav, S. (2021). Segway: Electric Hover Board. *International Journal of Advanced Research in Science Communication and Technology*, 223–228. https://doi.org/10.48175/ijarsct-999

Vaz, F. a. J., Sangeetha, T., Leksmi, S. U. S., & Dhanusha, A. (2024). Performance Analysis of nRF24L01 Wireless Module. *2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI)*, 216–222. https://doi.org/10.1109/icdici62993.2024.10810938

Vishnu, Varthan, Ganesh, S., & Anbarasi. (2025). Real-time implementation of a self-balancing robot using PID controller. *International Journal of Engineering Research & Technology (IJERT)*, *4*(5), 2278–0181.

Vladimirovich, I. V., Tudevdagva, U., Kovalev, A. V., Borisovich, S. O., Maksimov, A. V., & Sergeevna, I. A. (2020). Model of Adaptive system of Neuro-Fuzzy inference based on PID- and PID-Fuzzy-Controllers. In *Advances in intelligent systems and computing* (pp. 519–533). https://doi.org/10.1007/978-3-030-51971-1_43

Wang, Z., Liang, B., Sun, Y., & Zhang, T. (2019). Adaptive Fault-Tolerant Prescribed-Time control for teleoperation systems with position error constraints. *IEEE Transactions on Industrial Informatics*, *16*(7), 4889–4899. https://doi.org/10.1109/tii.2019.2951329

Xin, S., You, Y., Zhou, C., Fang, C., & Tsagarakis, N. (2017). A torque-controlled humanoid robot riding on a two-wheeled mobile platform. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, *2*, 1435–1442. https://doi.org/10.1109/iros.2017.8205945

Yang, Z., Chen, T., Luo, X., Schulting, P., & De Doncker, R. W. (2021). Margin balancing control design of Three-Phase Grid-Tied PV inverters for stability improvement. *IEEE Transactions on Power Electronics*, *36*(9), 10716–10728. https://doi.org/10.1109/tpel.2021.3061624

Yavarian, K., Mohammadian, A., & Hashemi, F. (2015). Adaptive Neuro Fuzzy Inference System PID Controller for AVR System using SNR-PSO optimization. *International Journal on Electrical Engineering and Informatics*, *7*(3), 394–408. https://doi.org/10.15676/ijeei.2015.7.3.3

Ye, L., Liang, B., & Wang, X. (2021). An extended PID control framework. *2021 IEEE Conference on Control Technology and Applications (CCTA)*, 1069–1074. https://doi.org/10.1109/ccta48906.2021.9659288

Zaher, A. A. (2018). Design of Model-Based gain Scheduling Controllers for nonlinear Systems. *Journal of Physics Conference Series*, *1141*, 012117. https://doi.org/10.1088/1742-6596/1141/1/012117

Zhao, C. (2022). Stability and tracking performance of PID controlled nonlinear uncertain systems. *2022 41st Chinese Control Conference (CCC)*, 785–790. https://doi.org/10.23919/ccc55666.2022.9902680

Zheng, N., Zhang, Y., Guo, Y., & Zhang, X. (2017). Hierarchical fast terminal sliding mode control for a self-balancing two-wheeled robot on uneven terrains. *Cybersecurity and Cyberforensics Conference*, 4762–4767. https://doi.org/10.23919/chicc.2017.8028105