Boomboom Balancino Bot (BBB): Gyro-Stabilized Straight-Line Robot Using Mpu6050 Feedback Control

Luisito Bernardo., Dwight Baines Camposano., Lynard Espiritu., Judemar Silmar., Steve Villa., Engr. Bernard C. Fabro

Computer Engineering Department, Eulogio "Amang" Rodriguez Institute of Science and Technology, Nagtahan, Sampaloc, Manila, 1016 Philippines

DOI: https://doi.org/10.51244/IJRSI.2026.13010025

Received: 07 January 2026; Accepted: 09 January 2026; Published: 23 January 2026

ABSTRACT

This project presents the design and implementation of a BBB (BoomboomBalancinoBot) gyro-stabilized straight-line mobile robot that demonstrates the core principles of feedback and control systems using low-cost embedded hardware. The system uses an ESP32-C3 microcontroller, an MPU6050 inertial sensor, and a TB6612FNG motor driver to control a two-wheel differential-drive robot with a rear caster wheel. The robot estimates its yaw (heading) by integrating the MPU6050's gyroscope Z-axis readings, then applies a closed-loop PID controller to minimize heading error by adjusting left and right motor speeds. A web-based interface allows users to command motion and switch between open-loop (feedback OFF) and closed-loop (feedback ON) operation, enabling clear demonstration of drift versus automatic correction under disturbances such as wheel slip or external pushing. The project highlights how real-time sensor feedback improves stability and accuracy in mobile robot navigation, closely relating to heading-hold control used in autopilot and course-keeping systems.

Keywords: Feedback Control, PID Controller, MPU6050 Gyroscope, Heading Hold, Differential Drive Robot

INTRODUCTION

Almost identical and inexpensive differential-drive robots deviate from a straight line as real-world imperfections such as motor mismatches, uneven wheel friction, surface irregularities, and wheel slip cause yaw drift even if a "forward" command is given [1] [2]. The project thus adds a closed-loop feedback controller that gets yaw-rate from an MPU-6050 gyroscope and continuously changes the left- and right-motor PWM signals to hold a commanded heading. This means that the approach is more about yaw-stabilization/heading-hold than the full inverted-pendulum balancing, thus a set-point, error, and corrective output can be measured [3].The hardware stack (ESP32-C3, TB6612FNG driver, two N20 DC motors, rear caster wheel, SSD1306 OLED, and Wi-Fi web interface) is following recent low-cost implementations which have been successful in combining ESP32, MPU-6050, and PID control for stable motion [4] [5].As the system switches between open-loop (no correction) and closed-loop operation, it demonstrates the main control conceptsset-point tracking, disturbance rejection, and PID tuningthat are commonly included in the robotics curricula and are directly applicable to practical platforms such as warehouse AGVs/AMRs, drone yaw stabilization, and autonomous vehicle steering [6] [7]. Hence this cheap, reproducible platform can be used for both educational and research purposes, thus facilitating the further investigation of sensor-based motion control in real-world conditions.

Background of the Project

Manylow-cost mobile robots can move forward, backward, and turn, but they often failto move straight consistently. This happens due to common real-world factors such as motor mismatch, unequalwheelfriction, uneven surfaces, and wheelslip. As a result, the robot drifts and curves even when the user commands "forward." To address this, our project focuses on feedback and controlsystems using the MPU6050IMU (gyroscope) to measure yaw rotation and a controlalgorithm that adjusts motor speeds to maintain a desired heading.Instead of balancing uprightlike an inverted pendulum, this project demonstrates a more stable and measurable

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue I January 2026

controlobjective: yaw stabilization / heading hold. The system uses an ESP32-C3 Super Mini as the controller, aTB6612FNG motor driver to drive two N20DC motors, a rear caster wheelfor stable support, a web-based controlinterface for command input, and a smallOLED display for real-time status

Importance and Relevance of the Study

This project is relevant because:

- It provides a practical demonstration of a closed-loop feedback control system using a real sensor (MPU6050) and real actuators (DC motors).
- It highlights the difference between open-loop control (no correction) and closed-loop control (automatic correction) using a simple UI toggle.
- It demonstrates key control concepts such as setpoint, error, controller output, and disturbance rejection.
- It is closely related to real-world systems such as warehouse robots (AGVs/AMRs), drones (yaw stabilization), and autonomous vehicle direction control.

Problem Statement

Most basic differential-drive robots cannot maintain a straight path reliably due to real world disturbances and motor differences. There is a need for a simple educational platform that demonstrates how a robot can automatically correct its direction using feedback and control and show clear evidence of that improvement through open-loop vs closed-loop comparison.

Thus, the main problem addressed is:

How can we design and implement a low-cost mobile robot using an ESP32-C3, MPU6050 sensor, and TB6612FNG motor driver that maintains a desired heading through a feedback control system and demonstrates open-loop vs closed-loopbehavior?

General Objective

To design and implement a gyro-stabilized straight-line robot using an ESP32-C3 Super Mini, MPU6050, TB6612FNG, and N20 motors that can hold a desired heading through feedback control and provide a simple web interface and OLED display for demonstration.

Specific Objectives

The project aims to:

- Design and assemble a mobile robot hardware system using an ESP32-C3, MPU6050, TB6612FNG motor driver, two N20 DC motors with wheels, and a rear caster wheel for stability.
- Interface the MPU6050 with the ESP32-C3 through the I²C protocol and acquire yaw-rate data from the gyroscope for heading estimation.
- Implement a feedback control algorithm using PD or PID-based yaw stabilization to minimize heading drift and reject external disturbances.
- Control motor speed and direction through the TB6612FNG by adjusting left and right PWM signals based on the controller output.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue I January 2026

- Develop a Wi-Fi–based web interface and OLED display to control throttle and steering, toggle feedback on and off, visualize system variables in real time, and evaluate open-loop versus closed-loop performance.

RELATED LITERATURE

Feedback Control in Differential-Drive Robots

Differential-drive mobile robots are prone to heading drift because of motor mismatch, uneven friction, and surface disturbances. Recent control literature for wheeled mobile robots emphasizes closed-loop regulation to improve stability and robustness (Borkar et al., 2023; Villalobos-Aranda et al., 2024). In this study, the feedback objective is simplified to heading-hold control so that the effects of feedback can be clearly observed in a low-cost educational platform.

PID Control for Embedded Mobile Robots

PID control remains a practical baseline for embedded robotic systems because it is straightforward to implement and tune, while still providing meaningful disturbance rejection. Recent embedded robot projects continue to adopt PID-based control loops for real-time demonstration and STEM education (Rajendran et al., 2025). Similarly, this work applies a PID/PD-style controller to correct heading error by differentially adjusting left and right motor PWM outputs.

IMU-Based Heading Estimation and Sensor Error Sources

Heading estimation using low-cost IMUs typically relies on integrating the gyroscope yaw-rate to obtain a relative yaw angle. However, measurement noise and gyro bias can accumulate and produce drift, making calibration and bias compensation important (Zheng et al., 2023; Pimpalkar, 2024). To address these issues, the BBB system performs an initial stationary bias calibration and then applies the calibrated yaw-rate stream as feedback for the controller.

Heading-Hold Control for Course-Keeping Systems

Heading-hold (course-keeping) control is commonly used to maintain a desired direction despite disturbances. Malu and Majumdar (2024) discussed heading control concepts in mobile robot navigation, while recent control design studies continue to highlight robustness requirements for wheeled robots (Xu et al., 2022; Villalobos-Aranda et al., 2024). In contrast to full localization, the present work focuses on yaw stabilization during straight-line motion to provide a direct, measurable comparison between open-loop and closed-loop behavior.

Table 1. Comparison Matrix of Related Studies and Current Research

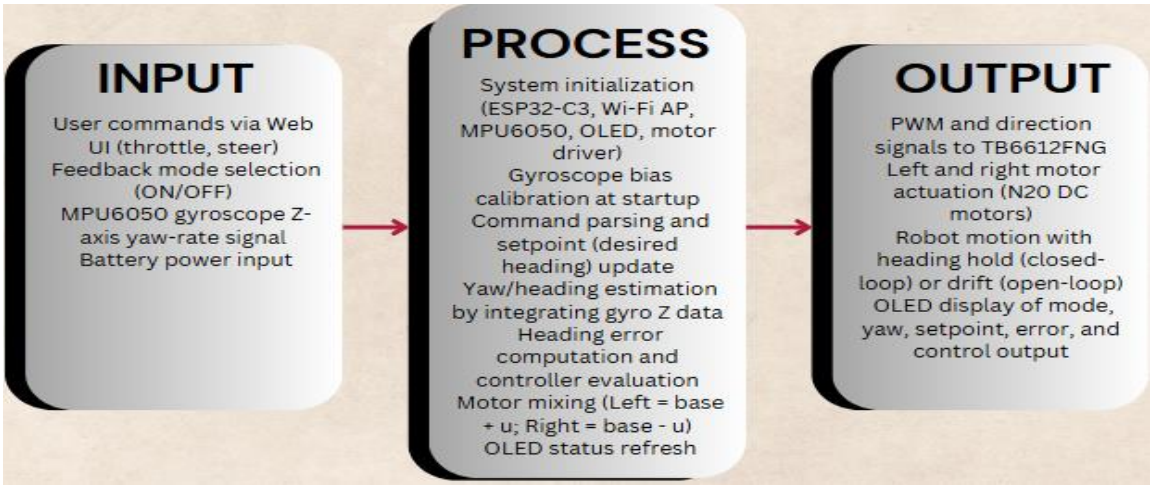
Study	Sensor Used	Platform/Technology	Key Feature(s)	Gap Addressed by the Study
Xu et al. (2022)	Not specified (model-based study)	Trajectory-tracking control for differential-drive robots	Backstepping + fractional-order PID for trajectory tracking under skidding/sliding conditions	Focuses on trajectory tracking design/optimization rather than a simple low-cost heading-hold demonstration platform
Borkar et al. (2023)	Multiple (review)	Review of wheeled mobile robot navigation/control	Survey of stability analysis and navigational techniques for WMRs	Does not provide a compact embedded implementation for classroom demonstration
Villalobos-Aranda et al. (2024)	Not specified	Differential-drive robot control	Almost-global trajectory tracking controller with	Addresses full trajectory tracking; does not target low-cost IMU-based heading-

		(theoretical + experimental)	stability analysis and experiments	hold for open-loop vs closed-loop comparison
Zheng et al. (2023)	Inertial sensors (IMU)	IMU measurement systems	Modeling and compensation of inertial sensor errors (noise, bias, drift)	Does not integrate the compensation approach into a complete low-cost mobile robot heading-hold platform
Rajendran et al. (2025)	IMU (MPU6050-class) + motors	ESP32-based embedded robot platform	PID-based embedded control for educational robotic demonstration	Primarily targets balance/stability problems; heading-hold for differential-drive motion is not the main focus
Malu & Majumdar (2024)	IMU / motion sensors	Mobile robot kinematics and control (survey)	Discusses kinematics, localization, and control concepts including heading-related motion	Lacks a detailed low-cost hardware implementation focused on yaw stabilization and disturbance rejection
Current Study	MPU6050 gyroscope (yaw-rate)	ESP32-C3 differential-drive robot with TB6612FNG	Web-controlled open-loop vs closed-loop yaw stabilization (heading hold) using PID/PD correction	Provides a complete, low-cost, and reproducible platform for demonstrating real-time feedback control effects

METHODOLOGY

The system is made up of a low-budget differential-drive mobile robot that demonstrates heading stabilization through feedback control. An ESP32-C3 microcontroller is used as the brain of the system tasked with sensor data acquisition, control algorithm execution, motor actuation, and user interaction. Information on yaw-rate is taken from an MPU6050 inertial measurement unit via the I²C interface and is used to estimate heading deviations during motion. The controller then uses this feedback to adjust the left and right motor speeds through a TB6612FNG dual H-bridge motor driver to follow the desired heading.The system is capable of running in two modes: open-loop control, which is the motor operation without any correction, and closed-loop control, which is using gyro feedback for real-time heading correction. A web-based interface offers users the capability to change motion parameters and switch feedback on or off, whereas a 128×64 OLED display shows the system status, yaw data, heading error, and controller output in real-time. Combined, these components provide a platform where feedback and control concepts in mobile robotics can be developed, observed, and evaluated.

Figure 1. IPO Framework



The figure shows the Input-Process-Output phases of theBBB Gyro-Stabilized Straight-Line Robot System.

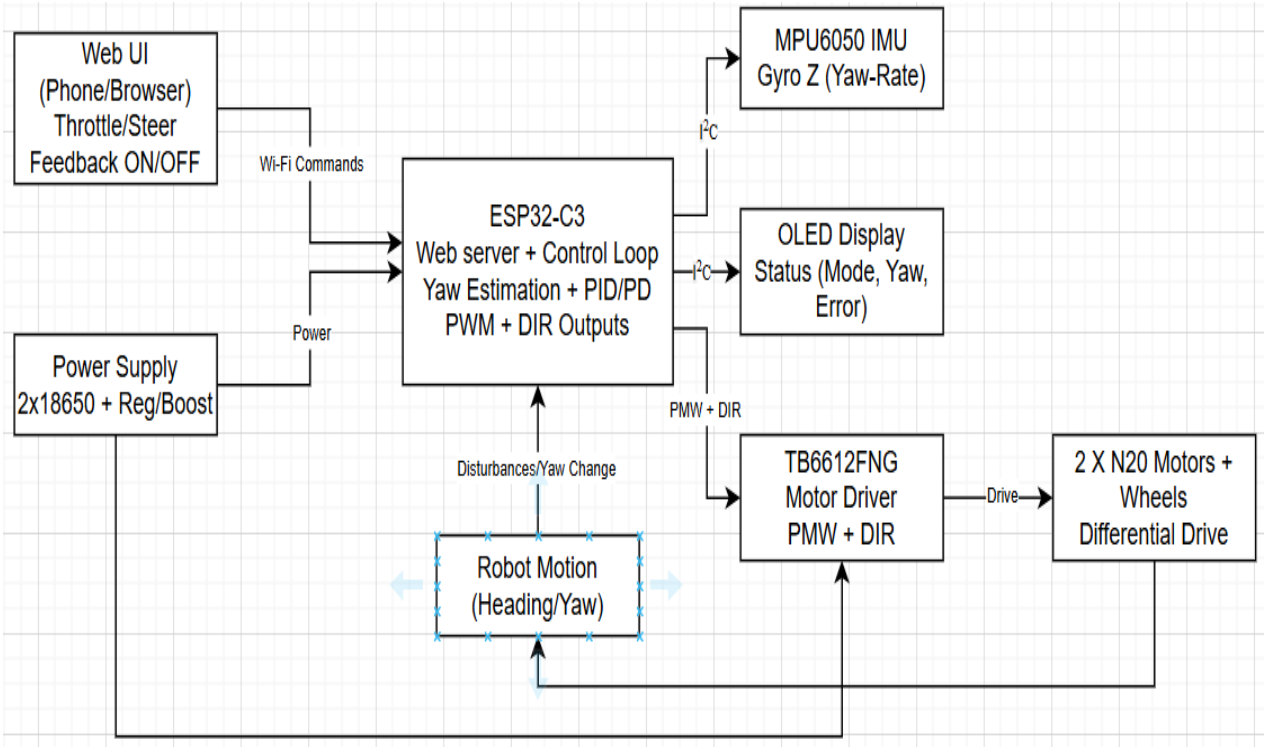
ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue I January 2026

Input: The input side is composed of commands from the user and data from sensors. Throttle,steering,rand feedback mode selection are made available via a web-based interface. The MPU6050 gyroscope gives Z-axis yaw-rate data, and battery supplies power to all the components of the system.

Process: The process step is done by the ESP32-C3 microcontroller. At system startup, the Wi-Fi module, MPU6050, OLED display, and motor driver are initialized, then gyroscope bias calibration is done. User commands are deciphered to specify the desired heading setpoint. The robot's yaw is determined by double integrating gyroscope data, and the heading error is calculated. The controller measures this error and produces a correction signal if the feedback mode is on. The motor mixing technique uses this correction to the base speed bysignalingone motor to go faster and the other slower. The OLED constantly updates the system status information.

Output: The control signals and system feedback make up the output. PWM and direction signals are TB6612FNG motor driver outputs left and right DC motors. The robot in closed-loop mode maintains its heading, whereas in open-loop mode it drifts. The OLED is used monitor the operating mode, yaw value, setpoint, error, and controller output live.

Figure 2. Block Diagram



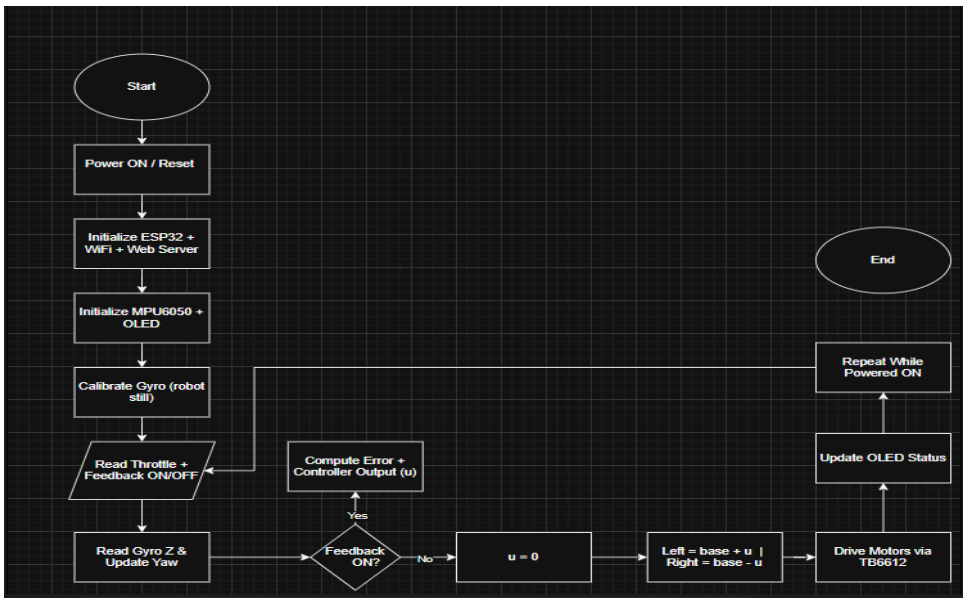
The block diagram shows the layout of the differential-drive robot, which has a heading-hold feature. Through a web-based interface and Wi-Fi, user throttle, steering, and feedback mode selection commands are sent to the ESP32-C3 microcontroller. The ESP32-C3 is a web server and the central control unit running the yaw estimation and PID or PD control loop.

Yaw-rate data from the MPU6050 IMU are sent to the controller via the I²C interface and used to calculate the robot's heading. Operating mode, yaw, and control error are among the system status information that is displayed on an OLED for real-time monitoring. The controller, based on the computed control output, sends PWM and direction signals to the motor driver TB6612FNG, which separately drives the left and right N20 DC motors running in a differential conjugation.

The motors allow the robot to move, while the changes in heading caused by external disturbances and mechanical asymmetries are detected by the gyroscope and thus complete the closed feedback loop. The dual-18650 battery pack with regulation circuitry is the source of the power, which keeps the controller, sensors, and actuators functioning stably.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue I January 2026

Figure 3. Flowchart



The flowchart visually represents the operational progression of the feedback-controlled robot on wheels one proposed. At the time of power-up or reset, the ESP32-C3 sets up a network Wi-Fi module, web server, MPU6050 sensor, and OLED display in that order, and then a gyroscope calibration is carried out if the robot is not moving. When performing a task, the system checks the user throttle commands, feedback enable status, and yaw-rate data from the MPU6050 to update the heading direction at every cycle operation.The controller when feedback control is, it calculates the heading error and issues a corrective control signal; otherwise, the output of the control is zero for an open-loop operation. The left and right motor velocities are changed by the TB6612FNG driver to impart the resultant motor commands. The system's state and the variables of the controller are shown on the OLED display, and the process continues indefinitely as long as the robot has power.

Real-Time Control Conditions And Loop Timing

The robot applies yaw stabilization only under defined operating conditions. First, the user enables Feedback ON from the web interface. Second, a forward or reverse command must exceed a small throttle deadband so that the robot is in motion; when the robot is stationary, motor outputs remain OFF and the controller output is held at zero to prevent unintended actuation. Upon enabling feedback (or at the start of a motion command), the current yaw estimate is captured as the heading setpoint and is maintained until the user changes the command (e.g., a steering input or a new setpoint request).

During each control cycle, the ESP32-C3 reads the gyroscope Z-axis yaw-rate from the MPU6050, applies the calibrated bias offset, and integrates the result to update the yaw estimate. The heading error is computed as the difference between the setpoint and the estimated yaw, then a PID/PD controller produces a correction term u. Motor mixing is performed as:

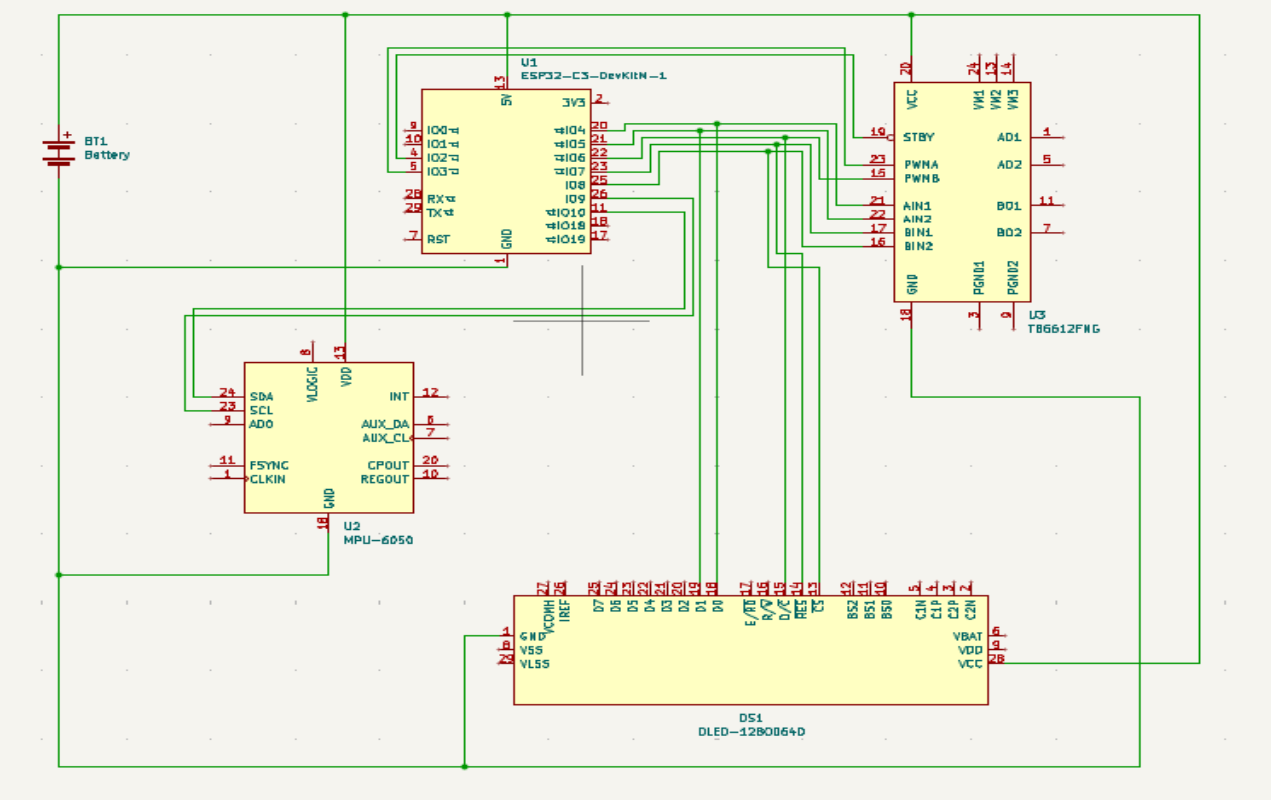
$$\textit{Left PWM} = \textit{basePWM} + u \textit{ and Right PWM} = \textit{basePWM} - u.$$

The correction term is bounded to avoid motor saturation and to ensure safe operation.

Because closed-loop performance depends on timing, time-critical sensing and control are executed at a fixed sampling interval, while web-server processing and display refresh are handled so they do not dominate the loop time. Excessive web requests or frequent display updates can increase effective delay; therefore, the implementation prioritizes the control loop over user-interface updates to maintain stable correction behavior.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue I January 2026

Figure 4. Schematic Diagram



The schematic shows the hardware setup of the proposed differential-drive robot equipped with gyro-based feedback control. An ESP32-C3 development board is utilized as the main controller, taking care of sensor acquisition, control computation, motor actuation, and system monitoring. Yaw-rate feedback is acquired from an MPU6050 inertial measurement unit linked through the I²C interface, thus enabling real-time heading estimation.Motor control is realized by a TB6612FNG dual H-bridge driver which obtains PWM and direction signals from the ESP32-C3 to motorize the left and right motors separately. A 128×64 OLED display, operating on the same I²C bus, shows the system status and control variables in real-time. All the parts are supplied with power from a single battery source with a shared ground, thus making a neat and cost-effective embedded platform suitable for feedback and control experiments.

Components and Their Functions

Inputs / Sensors

- **MPU6050 (Gyroscope Z)**

Measures yaw rotation rate; used as feedback to detect heading drift and disturbances.

Controller

- **ESP32-C3 Super Mini**

  Hosts the web server, receives commands, reads sensor, runs feedback controller, and outputs PWM to motors.

Outputs / Actuators

- **TB6612FNG**

motor driver Drives two N20 motors with direction + PWM.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue I January 2026

- **2× N20 motors + wheels**

Provide forward motion and differential steering corrections.

- **Rear caster wheel**

Provides third-point support for stability (not self-balancing).

Indicators / User Interface

- **OLED display**

Shows mode, feedback status, yaw estimate/setpoint, error, and correction output.

- **Web UI (phone browser)**

Tilt/manual control and a Feedback ON/OFF toggle to demonstrate open-loop vs closed-loop behavior

Limitations And Real-Time Considerations

Although the BBB platform demonstrates the advantages of closed-loop heading correction, several practical limitations must be considered for real-time deployment. First, the yaw angle is obtained by integrating gyroscope yaw-rate; therefore, residual bias and noise may accumulate over time and gradually shift the estimated heading. This effect is reduced by stationary bias calibration at startup, but long runs and temperature variation can still introduce drift (Zheng et al., 2023; Pimpalkar, 2024).

Second, the control response is affected by loop timing and actuator constraints. The controller can correct small to moderate disturbances (e.g., minor wheel slip or light pushing), but the correction may be insufficient under higher-speed motion or large disturbances when motor saturation, wheel traction limits, or mechanical misalignment dominate. In addition, user commands are delivered through Wi-Fi; network latency or disconnections may delay command updates and are not suitable for safety-critical applications.

Finally, the system does not use wheel encoders or absolute position sensing. As a result, the robot can hold a relative heading but cannot guarantee straight-line travel over long distances on uneven surfaces. This limitation is intentional for demonstration purposes: it highlights the difference between open-loop operation (high drift sensitivity) and closed-loop operation (automatic heading correction), while keeping the hardware low-cost and reproducible.

RESULTS & DISCUSSION

The experiment was about the robot's motion behavior when it is controlled by open-loop and closed-loop. In the former case, the robot could not hold the straight trajectory, as the repeated trials after the first one showed a constant (drifting) change in direction though the motor commands were the same. This refers to the fact that the limitations of cheap differential-drive robots are very obvious as a result of which motor mismatch, uneven wheel friction, and surface irregularities bring about unavoidable disturbances.

When feedback was used, travel in a straight line was visibly better. The use of the MPU6050 to monitor the yaw rate had the effect of enabling the controller to detect deviations in the direction of the fist and then adjust the motor speeds in the next moment, thus resulting in a more stable and even lie of the land. The difference became even more apparent when the robot was disturbed. The lack of feedback meant the robot would not fix its direction after being pushed or turned manually. Turning the closed-loop control on meant that the robot was able to adjust its yaw and go back to the desired direction. Thus, the robot was capable of disturbance rejection through the continuous adjustment of left and right motor outputs.

Information about present time variables such as mode of operation, status of feedback, yaw estimate, error in heading, and controller output was shown on the OLED. At the same time, a web-based interface made it possible

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue I January 2026

to switch the feedback on and off during running, which enabled the direct hearing of system behavior. In general, the system was able to perform like this, with the ESP32-C3 being dependable in handling sensing, control computation, motor actuation, and user interaction. The results show that just inertial sensor feedback is enough to help low-cost mobile robots follow a straight line thus confirming gyro-based heading control as a good method for educational and prototyping use even in the absence of wheel encoders or absolute position sensing.

Requirements

The functional requirements of the heading-hold differential-drive robot set forth the characteristics and control abilities that allow the device to behave in a certain way. The unit constantly gathers yaw-rate information via the MPU6050 gyroscope to sense any changes in the direction of the heading while the movement is going on. The ESP32-C3 evaluates the data from the sensor device instantly and, if the feedback control is turned on, determines the suitable control commands for the next step. The motors' velocity and rotation sense are tweaked by the TB6612FNG motor driver using the modulation of the left and right PWM signals to keep the heading accurate. The system is capable of running two modes. When in open-loop mode, the robot is under the control of user commands only and no correction is made. If in closed-loop mode, the PD or PID feedback controller will be working to compensate any disturbance and mechanical asymmetry. A Wi-Fi web interface is available to users for throttle and steering control and also for switching the feedback ON or OFF. An OLED display is used to show on real-time the system information such as mode of operation, yaw value, setpoint, error in control, and the output of the controller.

The non-functional requirements indicate the attributes of performance, reliability, and usability. The control loop functions non-stop with a very short delay in order to provide a timely adjustment of the heading error. With the system implementation, firstly the stability and safety are ensured by the fact that the motors will not be running at uncontrolled levels during the sensor initialization and startup. The sensor data together with the screen refresh are without wiggle or delay. The robot behaves steadily even when the surface has very few irregularities or when there are small external disturbances. The overall system is designed to be cheap, simple, and functional for presenting feedback control principles at the level of demonstrations inclassrooms.

Table 2. Variables and Conditions of the BBB Gyro-Stabilized Straight-Line Robot System

Variable/Component	Type (Input/Output)	Parameter Measured/Controlled	Condition or Range	SystemResponse/Action
Web UI (Wi-Fi control page)	Input	Motion command (throttle/steer) and mode command	Throttle: 0-100% (or PWM equivalent); steer: left/center/right	Transmits command values to ESP32-C3; updates motion request
Feedback toggle (Web UI)	Input	Control mode selection	ON / OFF	ON: heading-hold correction enabled; OFF: open-loop operation
MPU6050 gyroscope (Z-axis)	Input	Yaw-rate (angular velocity)	Configured gyro range (e.g., ±250 deg/s)	Provides real-time feedback for heading estimation and control
Gyro bias calibration	Input / Initialization	Gyro Z zero-offset (bias)	Performed at startup while robot is stationary	Bias is computed and subtracted from subsequent gyro readings
ESP32-C3 Super Mini	Controller	Real-time processing and PWM generation	3.3 V logic; continuous loop execution	Acquires inputs; estimates heading; computes error;

				applies control; drives outputs
Heading setpoint (desired heading)	Internal variable	Target heading reference	Captured at motion start / when feedback is enabled	Used as reference for heading-hold control
Heading error (e)	Internal variable	Setpoint minus measured heading	Wrapped to bounded range (e.g., -180 to +180 deg)	Feeds controller to compute correction output
PID/PD controller	Control process	Correction output (u)	Based on tuned gains (Kp, Ki,Kd) and loop period	Generates differential correction to minimize heading error
TB6612FNG motor driver	Output (Driver)	Motor direction and speed control	PWM duty cycle with DIR logic	Drives both motors according to mixed control outputs
Left N20 DC motor	Output (Actuator)	Left wheel speed	PWM varies with base + u; clamped to limits	Speed is adjusted to counter heading drift
Right N20 DC motor	Output (Actuator)	Right wheel speed	PWM varies with base - u; clamped to limits	Speed is adjusted complementary to left motor for correction
OLED display (SSD1306, I2C)	Output (Indicator)	System status display	I2C bus shared with MPU6050	Displays feedback mode, yaw, setpoint, error, and control output
Power supply (2x18650 + regulator/boost)	Power source	System and motor supply	Nominal pack voltage with regulated rails as required	Provides power to ESP32, sensors, OLED, driver, and motors

Table 3. Functional Test Cases of the BBB Gyro-Stabilized Straight-Line Robot System

Test #	Input Condition	Observed Output	Expected Output	Pass/Fail	Remarks / Behavior Explanation
1	System powered ON; no throttle command (idle)	Motors OFF; robot stationary; OLED indicates standby	System remains idle; no motion without user command	Pass	Safe default state; prevents unintended activation
2	Forward throttle command; Feedback OFF	Robot advances and exhibits gradual drift/curving	Open-loop operation exhibits drift under mismatch/disturbance	Pass	Demonstrates drift when feedback is disabled

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue I January 2026

3	Forward throttle command; Feedback ON	Robot advances with reduced drift; path is straighter	Closed-loop heading hold maintains desired heading	Pass	Sensor feedback improves straight-line motion
4	Feedback ON; external push/rotation applied during motion	Robot corrects and returns toward reference heading	Disturbance is rejected; heading error is reduced	Pass	Controller produces correction output u to compensate
5	Startup calibration performed while robot is stationary	Yaw estimate stabilizes; drift is reduced after calibration	Gyro bias is measured and compensated	Pass	Bias compensation improves integration accuracy
6	Toggle Feedback OFF to ON while moving forward	Correction begins; heading error decreases over time	Controller engages and maintains heading setpoint	Pass	Supports live switching for demonstration
7	Toggle Feedback ON to OFF while moving forward	Correction stops; drift becomes observable	Open-loop operation resumes with no correction applied	Pass	Clear closed-loop vs open-loop comparison
8	Throttle command set to zero while moving	Motors stop promptly; robot halts; OLED updates	Robot stops immediately upon stop command	Pass	Confirms stop response and PWM disable behavior
9	Rapid Web UI command changes (quick throttle/steer taps)	Robot responds without latching or unintended continuous motion	System accepts fast inputs without false activation	Pass	Command handling remains stable under rapid updates
10	High throttle with large heading error (high correction demand)	PWM outputs saturate at limits; motion remains controllable	PWM is clamped to safe limits; system remains stable	Pass	Output limiting prevents overdrive and instability

CONCLUSIONS AND RECOMMENDATIONS

This work presents a feedback and control system realized on a low-cost differential-drive mobile robot that uses an IMU for heading stabilization.Performance comparison between open-loop and closed-loop operations demonstrated a significant improvement in the directional stability and a reduction in the heading drift caused by motor mismatch and external disturbances. Using real-time yaw feedback, the system is able to constantly sense the heading error and correct it immediately by the controller, which in turn, allows the robot to move in a straighter line more consistently during closed-loop operation. The system is stable and repeatable, and the operator is given real-time monitoring through a web interface and an OLED display. These results show that the platform is a practical and measurable educational tool for learning about real-time feedback control through cheap components.

Further studies should be carried out to refine accuracy, ease of use, and robustness. The incorporation of wheel encoders would enable sensor fusion with the IMU for more accurate heading and distance estimation. A web interface for PID gain adjustment would make the testing and tuning processes quicker. Battery voltage measurement and display on the OLED would make the system more aware of the operation. By implementing turn-to-angle control, one could use heading feedback to perform accurate rotations. Mechanical enhancements such as wheel alignment, reduced caster friction, and optimized weight distribution will definitely lessen the effects of disturbances. Along with Wi-Fi, Bluetooth control support would give the user more options and increase system flexibility.

ACKNOWLEDGEMENT

The researchers highly appreciate their project adviser, Engr. Bernard C. Fabro, who with his valuable guidance, great patience, and constant support encouraged the conduct of this study. His advanced knowledge, fresh critical thinking, and helpful suggestions were instrumental in redirecting the research and raising the level of output. Besides the work's achievement, the researchers' academic progression and engineering practice understanding were enriched by his mentorship.

The researchers likewise thank the faculty members, instructors, and laboratory personnel for sharing their knowledge, giving technical assistance, and providing access to facilities and resources that were used for the development and testing of the system. Their collaboration and professional encouragement were main factors that helped the project to be refined and validated.

Also, great appreciation is extended to the researchers' classmates and peers whose suggestions, feedback, and encouragement have assisted in enhancing the clarity and impact of the study. The researchers' families are specially thanked for their unfailing understanding, love, and encouragement which were the researchers' source of strength during this work.

Finally, the researchers would like to dedicate their deepest gratitude to God who gave them the strength, wisdom, and perseverance to carry out the research. It was by His direction and grace that this study was brought to completion.

REFERENCES

1. Borkar, K. K., Aljrees, T., Pandey, S. K., Kumar, A., Singh, M. K., Sinha, A., Singh, K. U., & Sharma, V. (2023). Stability Analysis and Navigational Techniques of Wheeled Mobile Robot: A Review. Processes, 11(12), 3302. https://doi.org/10.3390/pr11123302 https://rjpn.org/ijcspub/papers/IJCSP25C1052.pdf
2. Villalobos-Aranda, C., Pliego-Jiménez, J., Montañez-Molina, C., & Arellano-Delgado, A. (2024). An Almost Global Trajectory Tracking Controller for Differential-drive Wheeled Mobile Robots. International Journal of Control, Automation and Systems, 22, 3684–3693. https://doi.org/10.1007/s12555-024-0218-4https://doi.org/10.13140/RG.2.2.22102.98886
3. Xu, L., Du, J., Song, B., & Cao, M. (2022). A combined backstepping and fractional-order PID controller to trajectory tracking of mobile robots. Systems Science & Control Engineering, 10(1), 134–141. https://doi.org/10.1080/21642583.2022.2047125https://doi.org/10.1088/1742-6596/1402/4/044021
4. Zheng, T., Xu, A., Xu, X., & Liu, M. (2023). Modeling and Compensation of Inertial Sensor Errors in Measurement Systems. Electronics, 12(11), 2458. https://doi.org/10.3390/electronics12112458https://doi.org/10.59247/csol.v2i2.150
5. Pimpalkar, S. (2024). Enhancing Robotic Localization with IMUs: A Fundamental Technology for Precise Navigation. Analog Dialogue, 58 (August 2024). https://www.analog.com/en/resources/analog-dialogue/articles/robotic-localization-with-imus.htmlhttps://labs.dese.iisc.ac.in/embeddedlab/two-wheel-self-balancing-line-tracker-bot/
6. Rajendran, J., Dhanasekharan, K., Subburaj, N., Thi Mai, N., Kamal, M. A. S., & Prapan, S. (2025). Development of Proportional-Integral-Derivative Based Self-Balancing Robot Using ESP32 for STEM Education. Engineering Proceedings, 92(1), 24. https://doi.org/10.3390/engproc2025092024https://thesai.org/Downloads/Volume16No10/Paper_88-Experimental_Validation_of_an_Adaptive_Controller_for_a_Mecanum_Wheel_Robot.pdf

7. Loganathan, A., & Ahmad, N. S. (2023). A systematic review on recent advances in autonomous mobile robot navigation. Engineering Science and Technology, an International Journal, 40, 101343. https://doi.org/10.1016/j.jestch.2023.101343https://globaljournals.org/GJRE_Volume14/1-Kinematics-Localization-and-Control.pdf
8. Seo, D., & Kang, J. (2025). Controller Design for Active Four-wheel Steering and Four-wheel Independent Drive-based Mobile Robot: Enhancing Cornering Performance in Negative Phase. International Journal of Control, Automation and Systems, 23, 235–248. https://doi.org/10.1007/s12555-024-0631-8https://www.ijert.org/design-and-control-for-differential-drive-mobile-robot
9. Malu, S. K., & Majumdar, J. (2024). Kinematics, Localization and Control of Mobile Robots. Global Journal of Researches in Engineering. https://www.gjrest.org/GJRE_Volume14/1-Kinematics-Localization-and-Control.pdfhttps://icmas.eu/Journal_archive_files/Vol_15-Issue1-2020-PDF/27-34_MAROSAN.pdf
10. Hirpo, B. D., & Zhongmin, W. (2017). Design and Control for Differential Drive Mobile Robot. International Journal of Engineering Research & Technology. https://www.ijert.org/design-and-control-for-differential-drive-mobile-robot
11. Espressif Systems. (2023). ESP32-C3 Series Datasheet. https://www.espressif.com/sites/default/files/documentation/esp32-c3_datasheet_en.pdf

ABOUT THE AUTHORS

Luisito Bernardo is a Bachelor of Science in Computer Engineering student with interests in embedded systems, control systems, and hardware–software integration. He contributed to the system design, implementation of the control logic, and overall coordination of the project development and testing activities.

Dwight Baines Camposano is a Bachelor of Science in Computer Engineering student with a focus on microcontrollers, sensor interfacing, and system integration. His contributions include hardware assembly, sensor calibration, and support in system testing and validation.

Lynard Espiritu is a Bachelor of Science in Computer Engineering student with interests in robotics, feedback control, and system analysis. He assisted in the development of the control methodology, documentation of experimental procedures, and analysis of systembehavior.

Judemar Silmar is a Bachelor of Science in Computer Engineering student whose interests include embedded programming and electronics. He contributed to circuit implementation, component integration, and troubleshooting during system development.

Steve Villa is a Bachelor of Science in Computer Engineering student who served as the Project Manager of the study. He was primarily responsible for project planning, task coordination, and overall system integration. He made the most significant contribution to the hardware development, including component selection, circuit assembly, wiring, and troubleshooting. His leadership ensured timely completion of milestones and effective collaboration among team members during system development and testing.

Engr. Bernard C. Fabro is a Professional Computer Engineer and A Professor at Eulogio “Amang” Rodriguez Institute of Science and Technology. He has over 15 years of teaching experience in computer engineering, specializing in robotics, programming, and control systems. His research interests include automation, deep learning applications, and smart systems, with several published works in international conferences and journals.