An Enhanced PID-Based Motion Control Framework for Autonomous Line-Following Robot

Nguyen-Thanh-Loc Tran [1], Viet-Tien-Dung Bui [2,*], Hong-Nho Bui [3], Hoang-Nguyen Nguyen [4], Thi-Ngoc-Thao Nguyen [5], Thanh-Sang Nguyen [6], Hung-Ky Nguyen [7], Huynh-Duc-Anh Nguyen [8], Thanh-Binh Phan [9], Hoang-Sang Luong [10], Le-Minh-Tan Nguyen [11], Vo-Minh-Khoa Tran [12], Tien-Dat Nguyen [13], Huynh-Khanh-Nam Pham [14], Duc-Dat Nguyen [15], The-Nhan Nguyen [16]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16] Ho Chi Minh City University of Technology and Engineering (HCM-UTE), Ho Chi Minh City (HCMC), Vietnam

Email: [1] 23151277@student.hcmute.edu.vn, [2] 23151228@student.hcmute.edu.vn, [3] 21161345@student.hcmute.edu.vn, [4] 22161160@student.hcmute.edu.vn, [5] thaontnt@hcmute.edu.vn, [6] 22142212@student.hcmute.edu.vn, [7] 19151010@student.hcmute.edu.vn, [8] 22151005@student.hcmute.edu.vn, [9] 22151009@student.hcmute.edu.vn, [10] 22142211@student.hcmute.edu.vn, [11] 22151038@student.hcmute.edu.vn, [12] 22151025@student.hcmute.edu.vn, [13] 19151048@student.hcmute.edu.vn, [14] 22142171@student.hcmute.edu.vn, [15] 21142514@student.hcmute.edu.vn, [16] 22142181@student.hcmute.edu.vn

*Corresponding Author

***Abstract*—PID controller is widely used in automatic control systems because it is simple, reliable, and easy to apply. It is especially suitable for mobile robots, such as line-following robots. The main contribution of this work is an experimental method to tune PID parameters. Instead of using complex algorithms, the parameters are adjusted and tested directly on a real robot. This makes the method easier to apply, especially for low-cost and educational systems. Experiments were conducted to evaluate how PID parameters (Kp, Ki, and Kd) affect the robot's performance. The robot was tested on different paths, including straight lines, curves, and 90-degree turns. The results show that the optimal parameters are Kp = 65, Ki = 0.1, and Kd = 13. With these values, the robot moves smoothly, responds quickly, and follows the path accurately.**

***Keywords*—PID Control; Line-Following Control; ESP; Line-Following Robot; Low-Cost and Educational Robot**

I. INTRODUCTION

The Proportional–Integral–Derivative (PID) controller is one of the most widely used control strategies in modern automatic control systems due to its simplicity, reliability, and strong performance across a broad range of applications [1]. It has been successfully applied in industrial manufacturing processes [2], autonomous robotics [3], healthcare systems [4], vehicle guidance and stabilization [5], and other dynamic control tasks [6]. A PID controller operates through three key components: the proportional term, which responds to the current error; the integral term, which eliminates accumulated steady-state error; and the derivative term, which predicts future behavior by evaluating the rate of change of the error [7]. Owing to this structure, PID remains an essential solution for achieving stability, fast response, and accuracy in control systems.

Line-following robots represent one of the most fundamental and widely studied platforms in mobile robotics research. These robots typically rely on infrared sensors to detect the contrast between a predefined path and the surrounding surface, allowing them to autonomously follow a trajectory. Due to their relatively simple mechanical structure and low implementation cost, line-following robots are widely used in robotics education, prototype development, and experimental validation of control algorithms [8], [9].

In recent years, several advanced control approaches have been proposed to improve the performance of mobile robots. Techniques such as intelligent hybrid techniques [10], fuzzy logic combined with genetic algorithms [11], and adaptive robust control [12] have been applied to enhance trajectory tracking and motion stability. In addition, several studies have attempted to optimize PID parameters using advanced computational methods such as immune algorithms [13], MATLAB-based optimization [14], [15], fuzzy inference systems [16], [17], and particle swarm optimization [18]. These approaches can improve control accuracy and adaptability under certain conditions.

However, many of these advanced methods require complex mathematical modeling, high computational resources, or simulation-based optimization procedures. Such requirements may limit their practical applicability, particularly for low-cost robotic platforms or educational environments where simplicity, real-time implementation, and hardware constraints are important considerations. Furthermore, line-following robots operating in real environments are often affected by sensor noise, variations in ambient lighting, and nonlinear motor responses, which can degrade control performance [19], [20].

Despite the large number of studies on PID and intelligent control techniques, there remains a lack of practical research focusing on systematic experimental PID tuning strategies validated directly on real robotic platforms, particularly for low-cost mobile robots used in educational and experimental environments. Therefore, identifying a simple yet effective PID tuning approach that can be easily implemented and experimentally validated remains an important challenge in robotic motion control.

To address this issue, this study proposes a practical implementation of a discrete PID controller for a line-following mobile robot based on the ESP32 microcontroller. Instead of relying on complex optimization algorithms or

intelligent hybrid controllers, this work focuses on a systematic experimental PID tuning methodology with real-time validation on a physical robotic platform. This approach emphasizes practical implementation, simplicity, and suitability for low-cost robotic systems.

The primary contributions of this work are as follows:

- A complete implementation of a low-cost discrete PID controller for a line-following robot using readily available components (ESP32, L298N, and TCRT5000 sensors).
- A systematic evaluation of the individual effects of Kp, Ki, and Kd on robot performance across straight paths, curved trajectories, and 90° turns.
- Experimental identification of optimal PID parameters for stable and smooth robotic motion.
- An analysis of performance limitations related to environmental factors such as lighting conditions and sensor noise, providing insights for future improvements.

II. ALGORITHM

A. Concept of PID Control

Fig. 1 a PID controller (Proportional–Integral–Derivative) is a feedback control technique widely used in automatic control systems. It continuously calculates an error value e(t) as the difference between the desired setpoint (SP) and the measured process variable (PV). The controller output u(t) is determined by combining three terms: proportional (P), integral (I), and derivative (D), as expressed in (1) [21]:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \quad (1)$$

Where Kp – proportional gain, Ki – integral gain, Kd – derivative gain, e(t)=SP−PV(t) – control error.

Each term of the PID controller contributes differently to the system response:

- Proportional term (P): produces a control output proportional to the instantaneous error. A larger Kp yields a faster response but may introduce oscillations, while a smaller Kp results in a slower and less sensitive system.
- Integral term (I): eliminates steady-state error by accumulating past errors over time. However, an excessively high KiK_iKi can cause overshoot and instability due to error accumulation.
- Derivative term (D): responds to the rate of change of error, improving transient performance and reducing overshoot. In practical applications, the derivative action is often filtered to reduce sensitivity to high-frequency noise [22].

The combined effect of the P, I, and D components enables the PID controller to achieve fast response, reduced steady-state error, and stable control performance.

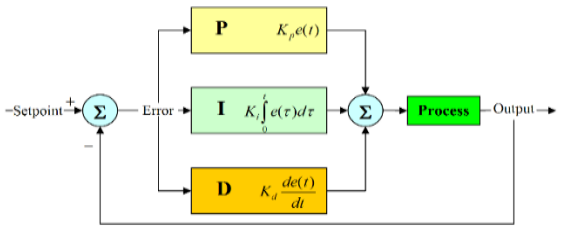

Fig. 1. PID Controller

B. Line-Following Algorithm

The line-following robot uses five infrared sensors (TCRT5000) labeled s1 s5 to detect the black line on a white surface. Each sensor outputs logic ‘0’ when detecting the line and ‘1’ otherwise. The combination of sensor readings indicates the robot’s lateral deviation from the path, as shown in Table 1.

Before conducting the experiments, the infrared sensors were calibrated to ensure reliable detection under the experimental lighting conditions. The reflected signals from both the black line and the white background were measured, and the threshold levels of the TCRT5000 modules were adjusted accordingly. Additionally, the sensors were mounted at a fixed distance from the ground to maintain stable readings and reduce measurement noise.

Table 1. Sensor value and position of the robot

Sensor Value	Position of Robot
00100	Center
10000	Far right
00001	Far left

Based on the sensor pattern, an error value (E) is assigned from –5 to +5, representing the deviation level. The microcontroller calculates the correction signal using the discrete PID equation:

$$u(k) = u(k-1) + Kp + KiTe(k) + \frac{Kd}{T}[e(k) - 2e(k-1) + e(k-2)] \quad (2)$$

Where T is the sampling period in this implementation, the PID controller is executed with a sampling period of T = 10 ms, which provides a balance between control responsiveness and the computational capability of the ESP32 microcontroller.

The output u(k) determines the PWM duty cycles applied to the left and right motors via the H-bridge driver (L298N). When the robot drifts left, the right motor speed increases and the left motor speed decreases, and vice versa. This closed-loop correction ensures the robot continuously follows the line smoothly and accurately.

C. PWM Control

Pulse Width Modulation (PWM) is used to control the average voltage supplied to the DC motors. By varying the duty cycle of a constant-frequency square wave, the motor speed can be adjusted [23], [24]:

Increasing the duty cycle → higher average voltage → higher speed, Decreasing the duty cycle → lower voltage → lower speed.

PWM offers high efficiency, low power loss, and easy implementation using microcontrollers, making it a suitable technique for motor speed control in mobile robots.

D. Motor Speed Control Using PID Output

The motor speed control based on the PID output is illustrated in the flowchart shown in Fig. 2. First, the microcontroller reads the line deviation through the function vitrixe(), which returns the error value E. The discrete PID controller then computes the control signal U(k) using the current and past error values.

To ensure stable operation, the control signal is constrained within a predefined range [−max,+max]. When the PID output reaches its saturation limits, specific correction rules are applied to maximize the robot's turning capability. Under normal operation, the left and right motor commands are obtained by adding and subtracting the PID output from the nominal reference value (Mid).

Finally, the calculated values are written to the PWM registers (OCR1A and OCR1B), which determine the duty cycles for the left and right motors. This closed-loop feedback mechanism allows the robot to continuously adjust motor speeds, enabling smooth and accurate line-following performance under varying conditions.

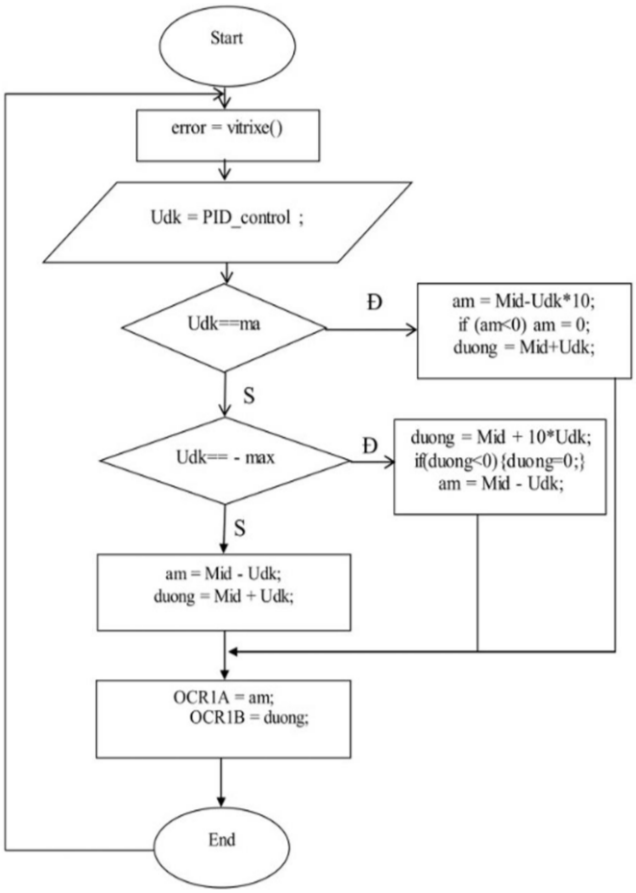

Fig. 2. Flowchart of the PID control algorithm for line-following robot

III. Experimental Model

A. Overview and Block Diagram

This project implements a line-following robot that uses infrared sensors (TCRT5000) to detect and track a line on the floor. The sensor data are processed by an ESP32 microcontroller, which controls two DC motors through the L298N motor driver module [25]. A discrete PID controller is applied to balance the motor speeds, allowing the robot to follow the path smoothly and maintain stability during turns or sudden direction changes.

Fig. 3 to Fig. 5 present different views of the experimental robot platform, including the sensor arrangement and motor driver configuration.

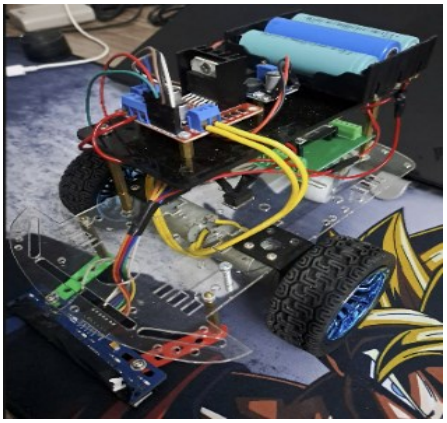

Fig. 3. Front view of the experimental model

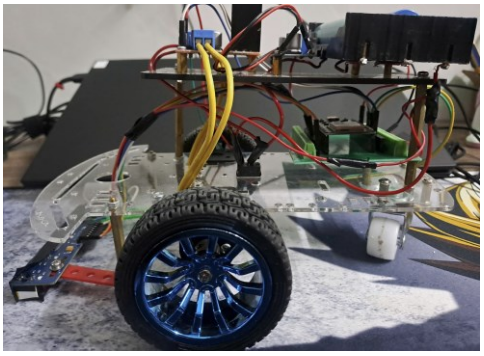

Fig. 4. Side view of the experimental model

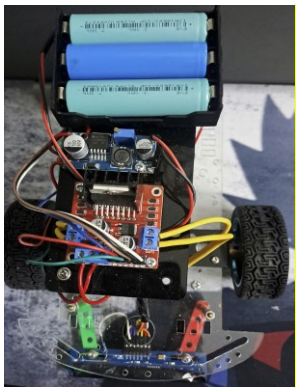

Fig. 5. Top view of the experimental model

B. Functional Description

From Table 2, components are as follows:

- Power Supply Unit: Provides 5 V for the ESP32 and sensors, and 12 V for the L298N motor driver to operate two DC motors.
- Sensor Unit: Consists of five TCRT5000 [26] infra-red sensors arranged in a row. These sensors detect the contrast between the black line and the bright background and output digital signals (0 or 1) to the microcontroller [27].
- Control Unit: The ESP32 [28] reads the sensor signals, calculates the positional error of the robot relative to the line, and applies the PID control algorithm to generate PWM signals for the motors.
- Motor Driver Unit: The L298N module receives PWM and direction control signals from the ESP32 to drive the motors forward or backward [29].

When the robot deviates from the line, the PID controller increases the speed of the motor on the opposite side, steering the robot back to the correct path. This enables smooth and accurate line tracking even at higher speeds [30].

Table 2. Elements of Robot from Fig. 6 and Fig. 7

Station	Unit
1	L298N
2	Line sensor
3	DC motor
4	ESP32
5	Battery
6	Switch
7	LM2596

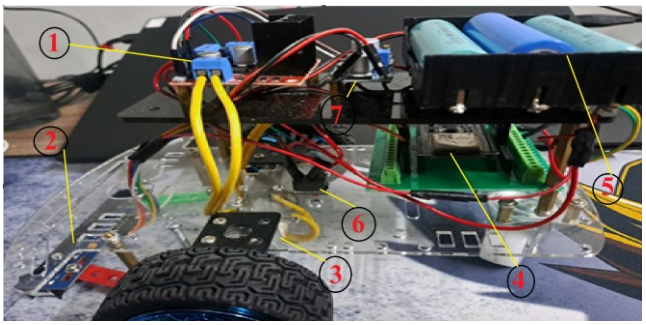

Fig. 6. Positions and components of the robot

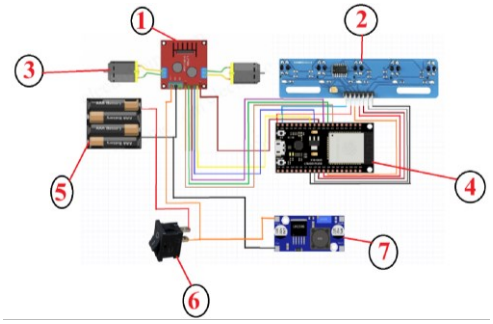

Fig. 7. The circuit connection diagram of the system

C. Working Principle

The infrared sensors continuously read reflected light values from the surface. When the robot deviates from the line, the ESP32 calculates the positional error and applies the discrete PID algorithm to modify the PWM signals of both motors [31]. This feedback mechanism allows the robot to quickly return to the correct path, achieving smoother and more stable motion compared to traditional control methods [32].

IV. Experiment Results

Initially, we set the speed to 120/255, but it was too fast, making it difficult to tune the PID parameters. Therefore, we reduced the speed to 50/255, which made it easier to find suitable parameter values. The results of our experiments are presented in Table 3 and Table 4.

Table 3. Adjust Elements of PID Control

Parameter	Kp	Ki	Kd	Result
PID-1	65	0.1	13	Fig. 8
PID-2	66	0.1	13	Fig. 9
PID-3	67	0.1	13	Fig. 10
PID-4	65	0.2	13	Fig. 11
PID-5	65	0.1	14	Fig. 12
PID-6	65	0.1	12	Fig. 13

The experimental tests were conducted on a track consisting of multiple segments, including straight paths, 90° turns, and curved sections. The corresponding output responses of the robot were observed for each trajectory condition.

To provide a clearer evaluation of the controller's performance, several basic quantitative indicators were considered during the experiments. These include the tracking error, which represents the deviation of the robot from the reference line, the maximum lateral deviation, and the settling behavior, which describes how quickly the robot returns to the correct path after a disturbance.

A. Adjusting Kp

At first, a very small value of Kp was selected and gradually increased while observing the behavior of the vehicle until it was able to follow the line steadily. The main evaluation criteria during this stage were the tracking sensitivity and the oscillation level of the robot.

Fig. 8 shows the response of the robot when the PID parameters were initially set to Kp = 65, Ki = 0.1, and Kd = 13. The motor output indicates that the vehicle maintained relatively stable motion across different track sections.

Fig. 9 and Fig. 10 illustrate the system response when the proportional gain was further increased while keeping Ki and Kd constant. After adjusting Kp while keeping Ki and Kd constant at 0.1 and 13, respectively, we observed noticeable changes in the line-following performance of the vehicle. When Kp = 65 as shown in Fig. 9, the motor output indicates that the vehicle moved smoothly and maintained a stable trajectory along the line, demonstrating satisfactory performance.

However, when Kp was slightly increased to 66, the vehicle began to exhibit mild oscillations and a slightly slower response, as indicated by the motor output waveforms in Fig. 10. Although the motion was less smooth, the vehicle was still able to traverse the entire path without losing the line.

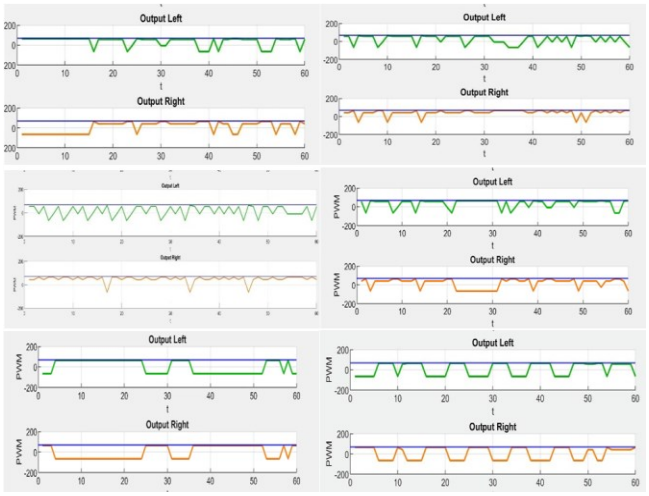

Fig. 8. PID-1's result

As Kp was further increased to 67, the vehicle's performance deteriorated significantly. It deviated from the intended trajectory, particularly at 90° right turns where it lost the line and rotated repeatedly while attempting to reacquire it. After some time, the vehicle managed to find the line

again, but with considerable delay. In curved sections, slight deviations were also observed.

These results indicate that an excessive increase in Kp causes the system to become unstable, reducing both tracking accuracy and smoothness. Therefore, Kp = 65 was selected as the optimal value for subsequent tuning of Ki.

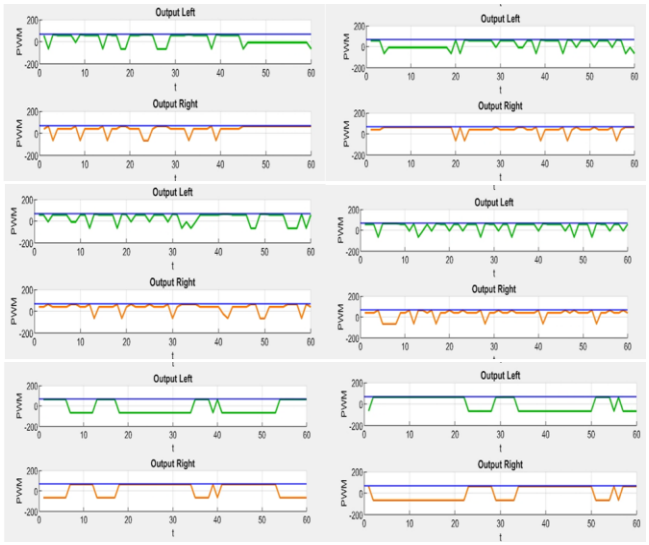

Fig. 9. PID-2's result

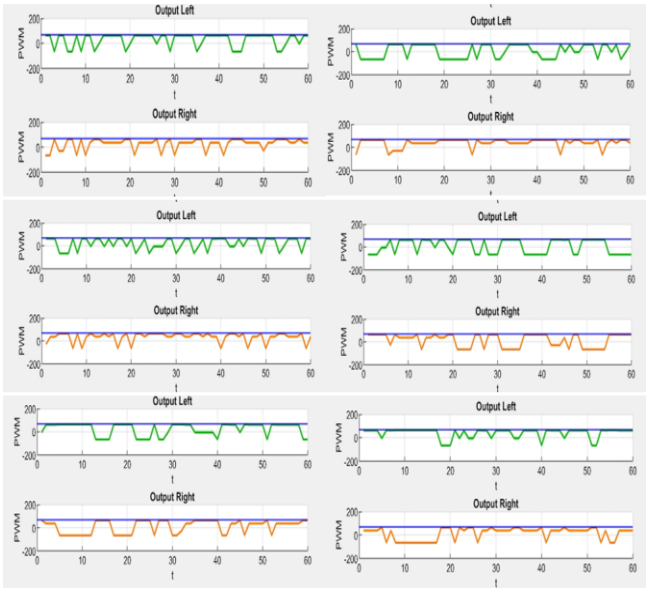

Fig. 10. PID-3's result

B. Adjusting Ki

In the second step, the controller was operated as a PID controller. The integral gain Ki is responsible for eliminating the steady-state error, which is the remaining error when only Kp is used, and the vehicle cannot perfectly follow the line. It accumulates the error over time, helping the vehicle maintain a more stable trajectory on straight sections. However, if Ki is too large, the vehicle may experience overshoot or oscillation due to excessive error accumulation. The Ki value was adjusted starting from a small value and gradually increased until the system began to oscillate if increased further. The Kp and Ki values were then retained for the next step.

We maintained constant values of Kp = 65 and Kd = 13 while varying the integral gain, Ki. As illustrated in Fig. 8, increasing Ki resulted in system instability, characterized by visible vibrations and a sluggish response. It appears that the stability threshold was exceeded at Ki = 0.1, as even a minor increase to 0.2 caused significant performance deterioration—the vehicle lost its smooth motion and even deviated from the trajectory, as clearly evidenced by the erratic motor output waveforms shown in Fig. 11. We attribute this phenomenon to excessive oscillations generating substantial inertia, which caused the vehicle to drift off the track. Consequently, we decided to retain Kp = 65 and Ki = 0.1 for the subsequent adjustment of Kd.

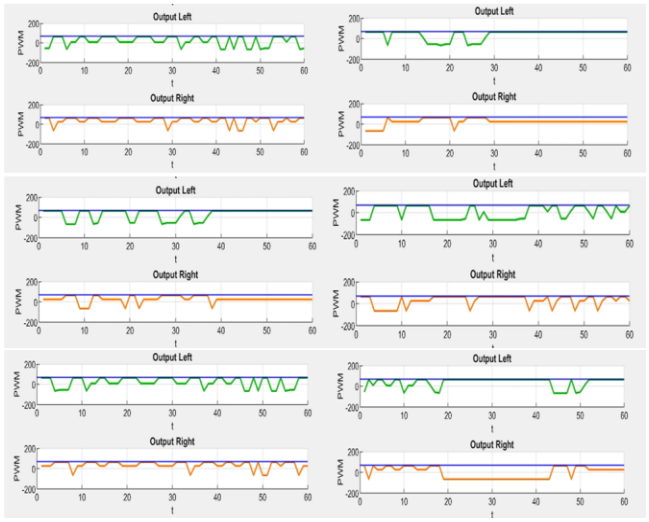

Fig. 11. PID-4's result

C. Adjusting Kd

After adjusting Ki, we identified suitable parameters that allowed the system to stabilize with a small error. We then proceeded to investigate Kd to complete the controller. The adjustment process was carried out in the same manner as the previous steps, ultimately resulting in the optimal parameter set for the system.

We maintain constant values of Kp = 65 and Ki = 0.1 while varying the derivative gain, Kd. Increasing Kd to 14 results in moderate oscillations, as illustrated in Fig. 12. Although the vehicle occasionally deviated from the trajectory, it successfully reacquired the line; however, the overall stability was inferior compared to the Kd = 13 configuration. Conversely, decreasing Kd to 12 results in a slower system response. While the vehicle managed to complete the course despite deviations, its performance varied across different track sections. Specifically, it remained stable on 3-line sections but showed deviations followed by recovery on 2-line sections. Notably, performance at 90° turns was inconsistent: the robot occasionally tracked well but at other times failed to detect the turn, leading to overshoot, as clearly depicted in Fig. 13.

D. Optimal Results

Further experiments were conducted with various combinations of parameters, and based on the experimental results in Fig. 14 is the trajectory, and Fig. 15 is the optimal result graph with the optimal parameters determined as Kp = 65, Ki = 0.1 and Kd = 13. With these values, the vehicle

moves stably in a straight line, moves smoothly without being sluggish, turns effectively on curves, and handles 90° corners well. However, on 2-line straights, the vehicle sometimes oscillates slightly to the left or right but quickly corrects itself and continues straight. On 3-line straights, the vehicle moves completely smoothly and straight, showing excellent performance

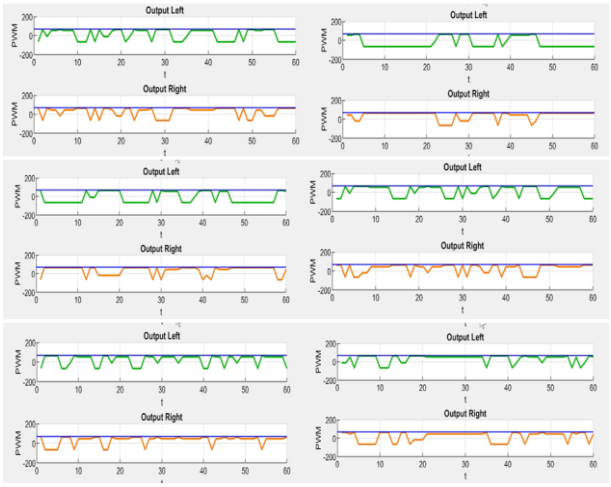

Fig. 12. PID-5's result

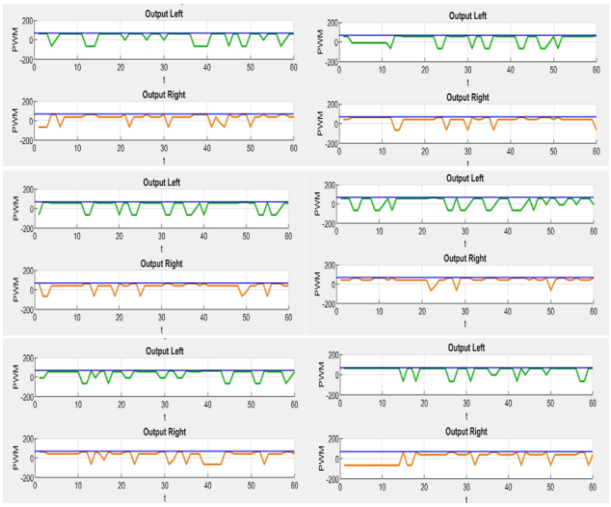

Fig. 13. PID-6's result

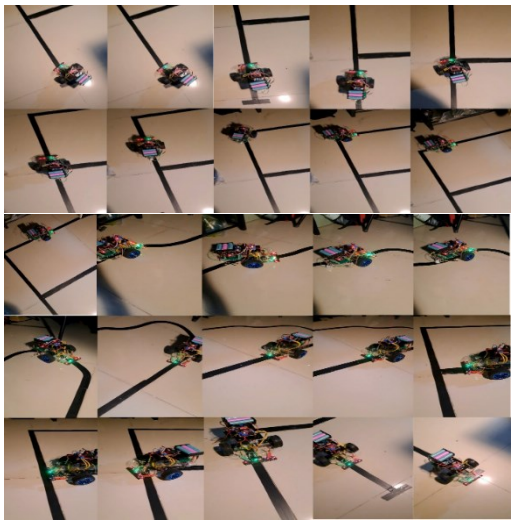

Fig. 14. The trajectory of the robot with optimal parameters

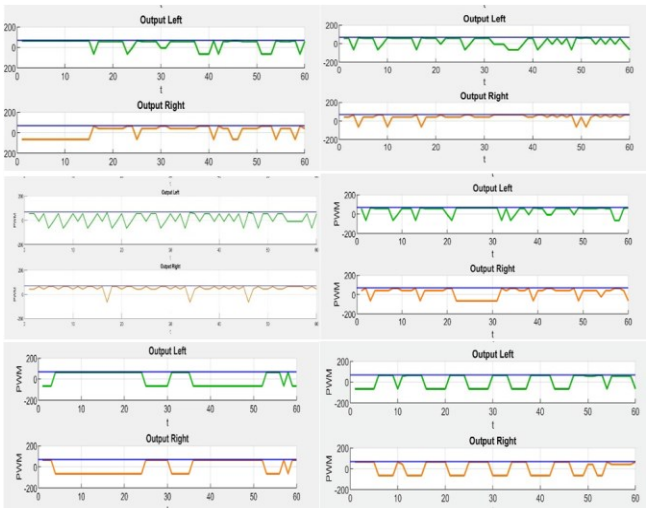

Fig. 15. Output result of the optimal parameters

Table 4. Comparison of Results with Different PID Parameters

Case	Kp	Ki	Kd	Tracking Error	Max Deviation	System Behavior
PID-1	65	0.1	13	Low	Small	Stable motion
PID-2	66	0.1	13	Medium	Moderate	Slight oscillation
PID-3	67	0.1	13	High	Large	Unstable
PID-4	65	0.2	13	High	Large	Strong oscillation
PID-5	65	0.1	14	Medium	Moderate	Moderate oscillation
PID-6	65	0.1	12	Medium	Moderate	Slow response

V. CONCLUSION

This study presented the design and implementation of a PID-based line-following robot using an ESP32 microcontroller and infrared sensors. The objective was to investigate the influence of PID parameters on the tracking performance and stability of the robot.

Through systematic experimental tuning, the optimal controller parameters were identified as Kp = 65, Ki = 0.1, and Kd = 13. With this configuration, the robot demonstrated stable motion, smooth trajectory tracking, and reliable performance when navigating straight paths, curved segments, and 90° turns.

The main contribution of this work lies in the systematic experimental tuning approach applied to a low-cost robotic platform. Unlike many existing studies that rely on complex optimization or intelligent control methods, the proposed method demonstrates that carefully tuned classical PID control can still achieve reliable line-following performance using simple hardware components. This approach provides a practical and cost-effective solution for mobile robotics and educational applications.

In addition to its practical relevance, the developed experimental platform can serve as a useful educational tool for studying feedback control, embedded systems, and robotic motion control.

Future work may focus on improving the system performance through adaptive PID tuning, fuzzy control strategies, or machine learning–based optimization methods. Further experiments on more complex track configurations and higher operating speeds could also be conducted to evaluate the robustness of the proposed approach.

ACKNOWLEDGMENT

This research was funded by Ho Chi Minh City University of Technology and Engineering, Vietnam, under grant No. SV2026-216. We want to give thanks to the PhD. Van-Dong-Hai Nguyen (HCM-UTE) due to his supervision for this study. Link of operation of the model is: https://youtube.com/shorts/dEiUP1Y-SKw?feature=share.

REFERENCES

[1] N. Razmjooy, Z. Vahedi, V. V. Estrela, R. Padilha, and A. C. B. Monteiro, "Speed Control of a DC Motor Using PID controller based on improved whale optimization algorithm," in *Lecture Notes in Electrical Engineering*, vol. 696, 2021, pp. 153–167, https://www.doi.org/10.1007/978-3-030-56689-0_8.

[2] M. Y. Alwardat and P. V. Balabanov, "Speed control of DC motor using PID controller based on MATLAB," *Vestnik Tambovskogo Gosudarstvennogo Tehnicheskogo Universiteta*, vol. 27, no. 2, pp. 195–202, 2021, https://www.doi.org/10.17277/vestnik.2021.02.pp.195-202.

[3] A. Eskandari and R. Vatankhah, "Opposition-based particle swarm optimization-aided neural fractional order PID pitch control for variable pitch wind turbines," *International Journal of Dynamics and Control*, vol. 13, no. 6, p. 216, 2025, https://www.doi.org/10.1007/s40435-025-01703-9.

[4] A. Duque-Torres, C. Klammer, S. Fischer, R. Ramler, and D. Pfahl, "Metamorphic testing for optimisation: A case study on PID controller tuning," *Information and Software Technology*, vol. 188, p. 107872, 2025, https://www.doi.org/10.1016/j.infsof.2025.107872.

[5] P. T. M. S. Parvathy Thampi and G. Raghavendra, "Intelligent model for automating PID controller tuning for industrial water level control system," in *Proceedings - 2021 International Conference on Design Innovations for 3Cs Compute Communicate Control, ICDI3C 2021*, pp. 149–155, 2021, https://www.doi.org/10.1109/ICDI3C53598.2021.00039.

[6] T. Rakib and M. A. R. Sarkar, "Design and fabrication of an autonomous fire fighting robot with multisensory fire detection using PID controller," in *2016 5th International Conference on Informatics, Electronics and Vision, ICIEV 2016*, pp. 909–914, 2016, https://www.doi.org/10.1109/ICIEV.2016.7760132.

[7] J. Te Shi, H. H. Gu, C. D. Shi, and T. Y. Ma, "PID control of medical centrifuge based on improved PSO-BP neural network," in *2023 8th International Conference on Intelligent Computing and Signal Processing, ICSP 2023*, pp. 831–834, 2023, https://www.doi.org/10.1109/ICSP58490.2023.10248482.

[8] A. L. Salih, M. Moghavvemi, H. A. F. Mohamed, and K. S. Gaeid, "Modelling and PID controller design for a quadrotor unmanned air vehicle," in *2010 IEEE International Conference on Automation, Quality and Testing, Robotics, AQTR 2010 - Proceedings*, pp. 74–78, 2010, https://www.doi.org/10.1109/AQTR.2010.5520914.

[9] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559–576, 2005, https://www.doi.org/10.1109/TCST.2005.847331.

[10] F. B. Rajshekhar, B. S. Kailash, K. N. Bibishan, and S. Heena, "Brainy Line Following Robot with Obstacle Detection," *International Journal of Advanced Research in Science, Communication and Technology*, pp. 292–301, 2025, https://www.doi.org/10.48175/IJARSCT-29244.

[11] A. Mohan, "Line-following robot for automatic delivery in classrooms or libraries," *Universal Research Reports*, vol. 12, no. 3, pp. 93–103, 2025, https://www.doi.org/10.36676/urr.v12.i3.1588.

[12] P. K. Mohanty and D. R. Parhi, "Navigation of an autonomous mobile robot using intelligent hybrid technique," in *Proceedings of 2012 IEEE International Conference on Advanced Communication Control and Computing Technologies, ICACCCT 2012*, pp. 136–140, 2012, https://www.doi.org/10.1109/ICACCCT.2012.6320757.

[13] G. Narvydas, R. Simutis, and V. Raudonis, "Autonomous mobile robot control using fuzzy logic and genetic algorithm," in *2007 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS*, pp. 460–464, 2007, https://www.doi.org/10.1109/IDAACS.2007.4488460.

[14] W. Shi, L. Xu, S. Chen, and Y. Liang, "Path tracking for wheeled mobile robot based on adaptive robust control," in *Proceedings of the 31st Chinese Control and Decision Conference, CCDC 2019*, pp. 3086–3091, 2019, https://www.doi.org/10.1109/CCDC.2019.8833456.

[15] D. H. Kim, D. H. Hong, and J. I. Park, "Auto-tuning of reference model based PID controller using immune algorithm," in *Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002*, pp. 483–488, 2002, https://www.doi.org/10.1109/CEC.2002.1006282.

[16] H. J. Live, H. B. Wang, X. M. Zhu, Z. H. Shen, and J. Y. Chen, "Simulation research of fuzzy auto-tuning PID controller based on matlab," in *Proceedings - 2017 International Conference on Computer Technology, Electronics and Communication, ICCTEC 2017*, pp. 180–183, 2017, https://www.doi.org/10.1109/ICCTEC.2017.00047.

[17] L. Ye, "MATLAB Simulation of PID Control Algorithm," *Journal of Theory and Practice of Engineering Science*, vol. 4, no. 03, pp. 28–45, 2024, https://www.doi.org/10.53469/jtpes.2024.04(03).06.

[18] H. Jiang, "Overview and development of PID control," *Applied and Computational Engineering*, vol. 66, no. 1, pp. 187–191, 2024, https://www.doi.org/10.54254/2755-2721/66/20240946.

[19] I. P. Diva, "Simulation of fuzzy logic controller as direction adjustment for line following robot," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3S1, 2024, https://www.doi.org/10.23960/jitet.v12i3s1.4590.

[20] A. A. Al Barazi and E. Aykut, "Automated tuning of PID controllers using a generalized discrete PID structure with particle swarm optimization," *IEEE Access*, vol. 13, pp. 185994–186013, 2025, https://www.doi.org/10.1109/ACCESS.2025.3625055.

[21] D. Çelik, N. Khosravi, M. A. Khan, M. Waseem, and H. Ahmed, "Advancements in nonlinear PID controllers: A comprehensive review," *Computers and Electrical Engineering*, vol. 129, pp. 1–43, 2026, https://www.doi.org/10.1016/j.compeleceng.2025.110775.

[22] P. Teppa-Garran, G. Bohórquez, and G. Garcia, "Optimal tuning of PID-type controllers," *Journal of Applied Research and Technology*, vol. 23, no. 2, pp. 145–154, 2025, https://www.doi.org/10.22201/icat.24486736e.2025.23.2.2470.

[23] O. Jasim, "Design and implementation of a practical board for generating a PWM signal to control the speed and direction of a DC motor," *Journal of Research in Engineering and Computer Sciences*, vol. 3, no. 03, pp. 01–08, 2025, https://www.doi.org/10.63002/jrecs.303.960.

[24] A. Faroqi, M. A. Ramdhani, F. Frasetyio, and A. Fadhil, "dc motor speed controller design using pulse width modulation," *IOP Conference Series: Materials Science and Engineering*, vol. 434, p. 012205, 2018, https://www.doi.org/10.1088/1757-899X/434/1/012205.

[25] J. Sang, "Application of PID control," *Science and Technology of Engineering, Chemistry and Environmental Protection*, vol. 1, no. 11, 2025, https://www.doi.org/10.61173/8c9hkg58.

[26] S. Yamanoor, *Introduction to the ESP32 Platform*. Springer Books, 2025, https://www.doi.org/10.1007/979-8-8688-1570-6_1.

[27] L. Chen, J. Zhang, and Y. Wang, "Wireless car control system based on Arduino Uno R3," in *2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC)*, IEEE, pp. 1783–1787, 2018, https://www.doi.org/10.1109/IMCEC.2018.8469286.

[28] D. N. David, D. Hamdani, and I. Setiawan, "Developing Physics Teaching Aids on Moment of Inertia Material using Arduino Nano and TCRT5000 Sensor," *Jurnal Ilmiah Pendidikan Fisika*, vol. 7, no. 1, p. 10, 2023, https://www.doi.org/10.20527/jipf.v7i1.6889.

[29] M. İ. Daşkın, E. Özkan, and K. B. Cihan, "Line follower robot with PID control," 2024. https://www.doi.org/10.13140/RG.2.2.24983.89760.

[30] Q. Ariyansyah and A. Ma'arif, "DC Motor Speed Control with Proportional Integral Derivative (PID) Control on the Prototype of a Mini-Submarine," *Journal of Fuzzy Systems and Control*, vol. 1, no. 1, pp. 18–24, 2023, https://www.doi.org/10.59247/jfsc.v1i1.26.

[31] F. Ben Aicha, "Optimized embedded AI: efficient implementation of CNNs on ESP32-CAM for real-time image classification," *Computing*, vol. 107, no. 11, p. 206, 2025, https://www.doi.org/10.1007/s00607-025-01559-z.

[32] S. K. Suman and V. K. Giri, "Speed control of DC motor using optimization techniques based PID Controller," in *Proceedings of 2nd IEEE International Conference on Engineering and Technology, ICETECH 2016*, pp. 581–587, 2016, https://www.doi.org/10.1109/ICETECH.2016.7569318.