DESIGN OF AN INTELLIGENT LINE-FOLLOWING AND MAZE-SOLVING ROBOT BASED ON FUZZY LOGIC AND ARDUINO

Truong-Phuong-Nam Pham[1], Minh-Khoa Nguyen[1], Vinh-Hung Lieu[1,*], Thi-Ngoc-Thao Nguyen[1], thanh-Binh Nguyen[1], Van-Hiep Nguyen[1], Thi-Hong-Lam Le[1], Trong-Bang Tran[2], Ngoc-Huy Do[1], Binh-Hau Nguyen[3]

[1]Faculty of Electrical and Electronics Engineering (FEEE), Ho Chi Minh City University of Technology and Engineering (HCM-UTE), Ho Chi Minh City (HCMC), Vietnam
[2]Department of Mechanical Engineering, Konkuk University, Seoul, Republic of Korea
[3]Posts and Telecommunications Institute of Technology, Ho Chi Minh, Vietnam

**Corresponding author: 23151256@student.hcmute.edu.vn*

(Received: 05 December 2025; Accepted: 02 February 2026; Published on-line: 01 March 2026)

ABSTRACT: This paper presents the design and implementation of an intelligent line-following and maze-solving robot based on fuzzy logic and an Arduino platform. The proposed system integrates infrared sensors for line detection, a fuzzy-PID control strategy for motion regulation, and a decision-making algorithm for maze navigation. The control approach was first validated through MATLAB/Simulink simulation and subsequently implemented on a physical robotic prototype. Experimental results conducted on a maze-structured track demonstrate stable line-tracking performance, smooth curve negotiation, accurate intersection handling, and precise stopping at the finish point. The results confirm that the proposed fuzzy-based control strategy enhances tracking accuracy, reduces oscillations, and improves overall robustness, proving its effectiveness and practicality for intelligent mobile robotic applications.

KEY WORDS: Fuzzy control, PID control, line-following robot, intelligent control.

1. INTRODUCTION

Line-following and maze-solving robots have been extensively utilized as benchmark platforms in autonomous mobile robotics due to their relevance in industrial automation, warehouse logistics, and intelligent navigation systems. These platforms serve as effective testbeds for evaluating control strategies, sensor integration, and decision-making algorithms in practical environments.

In the context of motion control, Fuzzy Logic Control (FLC) has been widely adopted to enhance line-following performance, particularly in handling nonlinear system behavior and sensor uncertainty. Prior studies, such as Diva [1] and Bach and Yi [2], demonstrated that fuzzy-based controllers improve tracking stability and reduce deviation. Furthermore, Supriadi et al. [3] reported enhanced tracking accuracy through optimization of membership functions. Despite these advantages, FLC typically exhibits limitations in steady-state accuracy and requires careful design of rules and parameters.

To overcome these limitations, hybrid control strategies integrating fuzzy logic with conventional Proportional-Integral-Derivative (PID) control have been proposed. Studies by Kumar and Singh [4], Zhang et al. [5], and Chen and Zhao [6] showed that fuzzy-PID controllers can achieve superior tracking performance, faster response, and improved robustness compared to standalone controllers. However, these approaches often introduce increased computational complexity and demand systematic parameter tuning.

On the other hand, maze-solving algorithms, such as Flood Fill and Dijkstra’s algorithm, are commonly employed for path planning in structured environments. These methods are effective for exploration and shortest-path determination, yet they primarily address high-level navigation and do not explicitly consider low-level motion control and trajectory stability.

Despite these developments, several gaps remain in the literature. Existing studies tend to focus on either line-following control or maze-solving independently, with limited integration between the two. Moreover, the interaction between low-level control stability and high-level navigation decisions has not been sufficiently explored. In addition, many works lack comprehensive validation through both simulation and real-world experimentation, particularly under complex operating conditions. These issues highlight the need for an integrated and experimentally validated control framework.

Accordingly, this study addresses the problem of developing a unified control system capable of maintaining stable line-following performance while enabling reliable and intelligent maze navigation in real-world environments. This research aims to design and implement an integrated control framework for an intelligent mobile robot by combining fuzzy logic and PID control with a maze-solving decision algorithm. The proposed system is implemented on an Arduino platform and evaluated through MATLAB/Simulink simulations as well as real-world experiments to assess both tracking performance and navigation capability. This work is positioned as a design- and application-oriented case study, emphasizing system implementation and experimental validation rather than theoretical novelty.

2. MODEL

2.1. System Architecture

The proposed system is formulated as a closed-loop control structure integrating sensing, control, and actuation subsystems, as illustrated in Fig. 1 . The sensing subsystem acquires environmental information in the form of line patterns on the surface, which are then processed by the control subsystem to generate appropriate control actions. These actions are executed by the actuation subsystem through a differential drive mechanism. This closed-loop configuration enables real-time feedback, ensuring stable trajectory tracking and robust system performance.

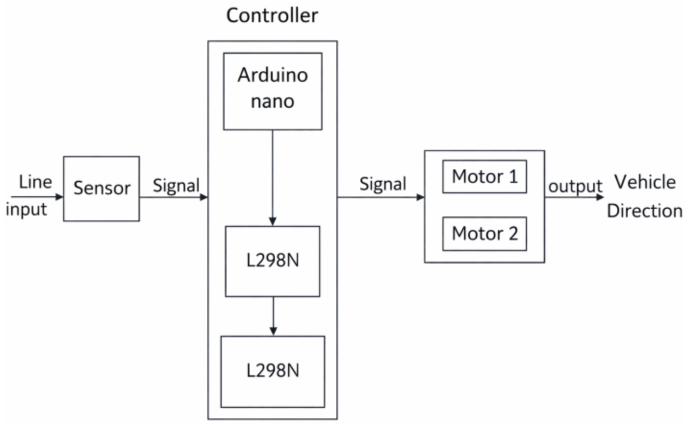


Fig. 1. Overall system architecture of the proposed robot

2.2. Sensor Configuration and Calibration

The sensing subsystem employs an array of four active infrared (IR) sensors positioned linearly at the front of the robot with an inter-sensor spacing of approximately 2 cm. Each sensor operates within a voltage range of 3.3–5 V and is equipped with an LM393 comparator, producing binary outputs where logic “1” represents line detection and logic “0” indicates the absence of the line.

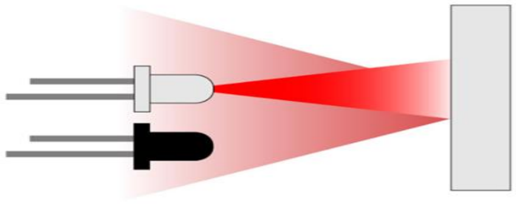


Fig. 2. Infrared sensor response on non-line surface

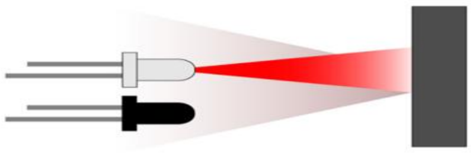


Fig. 3. Infrared sensor response on line surface

The working principle of the sensors is based on infrared light reflection characteristics. As depicted in Fig. 2 and Fig. 3 , bright surfaces reflect a higher portion of emitted infrared light, resulting in a stronger received signal, whereas dark line surfaces absorb more الضوء, leading to weaker reflections. This contrast in reflected intensity allows the system to reliably distinguish between line and background surfaces.

To ensure consistent performance, a calibration procedure is conducted under actual operating conditions. The robot is alternately placed on line and non-line surfaces while adjusting the onboard potentiometer until a clear threshold separation between logic “1” and logic “0” is achieved. This calibration step minimizes sensitivity to ambient lighting variations and enhances detection reliability.

2.3. Control Strategy

The control subsystem adopts a hybrid approach that combines fuzzy logic control with classical PID concepts. A Mamdani-type fuzzy logic controller processes two inputs, namely the tracking error $e(t)$and the change of error $de(t)$, to generate a control output that regulates the speed difference between the left and right motors. This approach enables smooth and adaptive steering correction under nonlinear conditions.

To further enhance system responsiveness and stability, PID parameters are tuned using a trial-and-error method. The integration of fuzzy reasoning and PID tuning provides a balance between robustness and control accuracy while maintaining manageable computational complexity.

2.4. Mathematical Representation of Line Tracking

The line-following behavior is modeled using an error-based control framework, where the deviation between the robot position and the reference line is defined as the tracking error $e(t)$[7]. The control action is implemented by adjusting the speed difference between the two drive motors, which can be expressed as

$$\Delta V = v_{right} - v_{left} \tag{1}$$

As indicated in Eq. (1), the value of $\Delta V$is determined by the fuzzy logic controller based on both the magnitude and direction of the tracking error. A positive error increases the speed of one motor while reducing the other, resulting in a corrective turning motion toward the line. Conversely, when $e(t)$approaches zero, $V_R$and $V_L$converge, enabling stable straight-line motion along the desired trajectory [8].

2.5. Maze Navigation Model

Beyond line tracking, the system incorporates a maze navigation capability based on pattern recognition and decision-making algorithms. The IR sensor array detects specific patterns corresponding to straight paths, left turns, right turns, and intersections [9], [10]. These patterns are processed by a rule-based navigation strategy, such as the left-hand or right-hand rule.

The navigation system is modeled as a finite-state machine consisting of several states, including line tracking, intersection detection, decision making, and motion execution [11], [12]. This structured modeling approach ensures seamless integration between sensing, control, and navigation subsystems, enabling stable movement and intelligent maze-solving performance in complex environments.

3. METHODOLOGY

3.1. Control System Overview

Based on the system model presented in Section 2, a hybrid control framework combining fuzzy logic and a Proportional-Integral-Derivative (PID) controller is developed to achieve stable and adaptive robot navigation. The system operates in a closed-loop configuration, where sensor measurements are continuously processed to generate real-time control actions for a differential-drive platform.

A conventional PID controller is initially employed to reduce tracking error and enhance stability; however, fixed gains are often inadequate under nonlinear dynamics and environmental uncertainties. To address this limitation, a fuzzy logic-based tuning mechanism is incorporated to adapt PID parameters online, thereby improving robustness and tracking accuracy, as reported in [13], [14]. The structure of the PID controller is illustrated in Fig. 4.

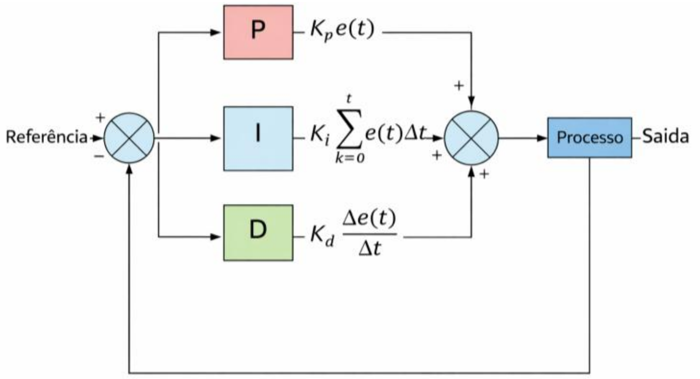


Fig. 4. Block diagram of the PID control system

3.2. Error Computation and Sensor Mapping

The tracking error is derived from the binary outputs of the infrared (IR) sensor array using a discrete mapping approach. Each sensor state is translated into a scalar error value within the range of −3 to 3, representing the lateral deviation from the line center. The mapping scheme is summarized in Table 1 , enabling efficient implementation in embedded systems.

Sensor signals are sampled at a fixed interval of 10 ms and converted into digital values after threshold calibration. In addition to line tracking, specific sensor patterns are used to identify intersections and navigation states. The corresponding intersection classifications are presented in Table 2.

Table 1: Sensor patterns and corresponding tracking error values

S1	**S2**	**S3**	**S4**	**Error**
1	0	0	0	3
1	1	0	0	2
0	1	0	0	1
0	1	1	0	0
0	0	1	0	-1
0	0	1	1	-2
0	0	0	1	-3

Table 2: Intersection detection based on sensor patterns

Pattern	**Code**	**Description**
1110	100	Left turn
0001	010	Right turn
0000	102	Dead end
1111	103	Intersection or goal

To support maze navigation, a rule-based decision strategy is implemented using the left-hand rule. At intersections, the robot prioritizes turning left, followed by moving straight, turning right, and finally executing a U-turn when no other options are available. This approach ensures systematic exploration with low computational overhead.

3.3. Fuzzy Logic Controller Design

A Mamdani-type fuzzy logic controller is designed to enhance adaptability by processing the tracking error $e(t)$and its derivative $de(t)$. Both inputs are defined over the normalized range [−3, 3] and represented by five linguistic variables: Negative Big (NB), Negative Medium (NM), Zero (Z), Positive Medium (PM), and Positive Big (PB).

Triangular membership functions are adopted due to their computational simplicity and suitability for real-time embedded systems [15]. The output variable is similarly defined, and centroid defuzzification is applied to obtain a crisp control signal. The complete inference mechanism is governed by 25 IF–THEN rules, as summarized in Table 3.

The rule base is structured such that large deviations produce strong corrective actions, whereas small deviations yield smoother responses, thereby minimizing oscillations and improving steady-state performance [16], [17].

Table 3: Fuzzy rule base

de \ e	**NB**	**NM**	**Z**	**PM**	**PB**
NB	NB	NB	NM	NM	Z
NM	NB	NM	NM	Z	PM
Z	NM	NM	Z	PM	PM
PM	NM	Z	PM	PM	PB
PB	Z	PM	PM	PB	PB

3.4. PID Tuning and Adaptive Fuzzy–PID Strategy

The control signal generated by the PID controller is expressed as

$$u(t) = K_p e(t) + K_i \int e(t)\, dt + K_d \frac{de(t)}{dt} \tag{1}$$

as shown in Eq. (2). The initial PID gains are obtained using MATLAB/Simulink tuning tools, resulting in $K_p = 1.0834$, $K_i = 1.009$, and $K_d = 0.0098722$.

To enable adaptive tuning, each parameter is allowed to vary within ±30% of its nominal value. The fuzzy inference system generates incremental adjustments ΔK_p, ΔK_i, and ΔK_d, which are applied in real time to update the controller gains. This adaptive mechanism improves transient response, reduces overshoot, and enhances robustness under varying operating conditions [18], [19].

3.5. Fuzzy–PID Control Architecture

The overall architecture of the fuzzy-tuned PID controller is presented in Fig. 5. In this configuration, the tracking error and its derivative are fed into the fuzzy system to dynamically adjust PID gains. The updated controller then produces the control signal for the robot motion system, ensuring smooth and stable trajectory tracking in the presence of disturbances and nonlinearities.

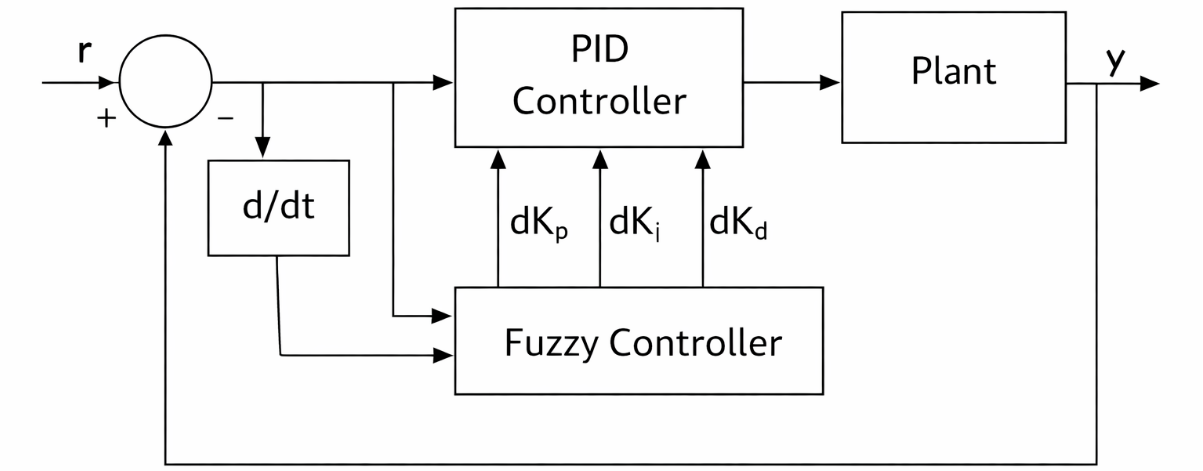


Fig. 5. Structure of the Fuzzy-tuned PID control system

3.6. Motor Control Strategy

The control output regulates the speed difference between the left and right motors based on a differential-drive model:

$$V_R = V_{base} + \Delta V, V_L = V_{base} - \Delta V \tag{2}$$

as defined in Eq. (3). Here, ΔVrepresents the control signal, while V_{base}denotes the nominal voltage.

Motor speed is controlled using pulse-width modulation (PWM) generated by an Arduino Nano. The applied voltage is limited to a maximum of 7 V to comply with driver specifications. The control loop operates at a sampling time of 10 ms, ensuring adequate responsiveness for real-time navigation [20].

3.7. Maze-Solving Algorithm

To enable autonomous navigation, a rule-based maze-solving algorithm based on the left-hand rule is implemented. The robot selects its motion at each intersection using a prioritized decision hierarchy: turning left, moving straight, turning right, and performing a U-turn when necessary.

The navigation logic is modeled as a finite-state machine comprising line tracking, intersection detection, decision making, and motion execution states. The decision process utilizes sensor patterns defined in Table 2, ensuring seamless integration between perception and control. The overall algorithm flow is illustrated in Fig. 6.

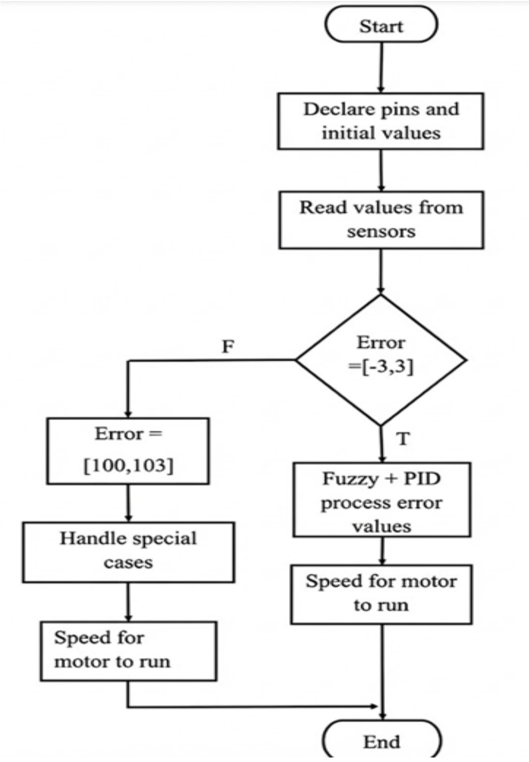


Fig. 6. Flowchart of the proposed fuzzy-PID control algorithm for line-following and maze-solving robot

3.8. Simulation and Experimental Validation

To validate the proposed control strategy, both simulation and experimental studies are conducted. A DC motor model is implemented in MATLAB/Simulink based on its electrical and mechanical characteristics, including resistance, inductance, back electromotive force, inertia, and torque constants, as shown in Fig. 7.

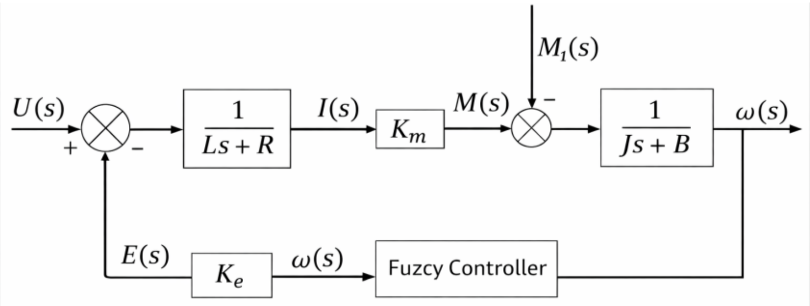


Fig. 7. Equivalent electrical and mechanical model of the DC motor

The parameters used in the simulation are summarized in Table 4. A closed-loop configuration is employed to evaluate system performance in terms of transient response, stability, and error convergence.

The system is subsequently implemented on a hardware platform consisting of an Arduino Nano, an L298N motor driver, DC motors, and an IR sensor array, as depicted in Fig. 8. The controller operates with 8-bit PWM resolution, enabling consistent motor speed control and facilitating reproducibility on similar embedded platforms.

Table 4: DC motor parameters

Parameter	Symbol	Value	Unit
Back EMF constant	Ke	0.01	Vs/rad
Torque constant	Km	0.01	Nm/A
Rotor inertia	J	0.01	kg·m²
Resistance	R	1	Ω
Inductance	L	0.5	H
Rated speed	n	150	rpm

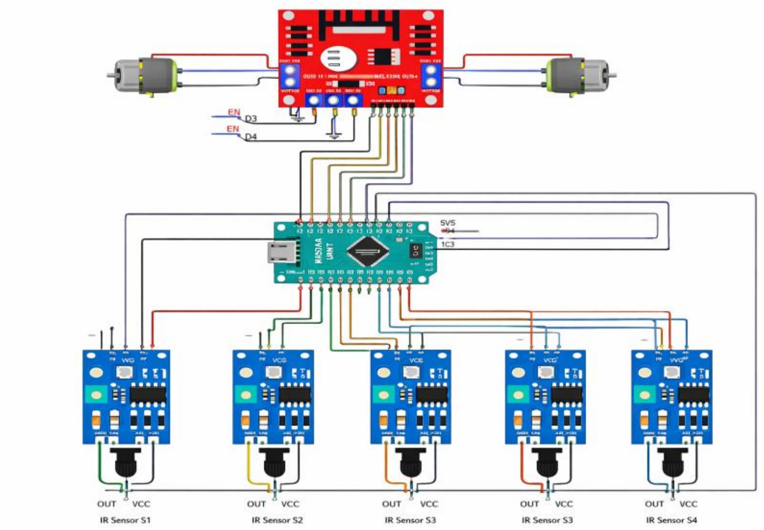


Fig. 8. Experimental hardware setup of the proposed robot system

4. RESULTS

4.1. Experimental Platform

The experimental validation is conducted using a line-following and maze-solving robot developed based on the proposed control architecture. The system integrates an Arduino Nano microcontroller, a four-channel infrared sensor array, an L298N motor driver, and two DC motors configured in a differential-drive arrangement.

The control algorithm, including fuzzy-based PID tuning and real-time speed regulation, is initially validated through MATLAB/Simulink simulations before being implemented on the physical platform. The experimental hardware configuration is presented in Fig. 9.

To evaluate system performance under realistic conditions, the test track is designed with straight segments, curved paths, and multiple intersections, enabling comprehensive assessment of tracking accuracy, stability, and navigation capability.

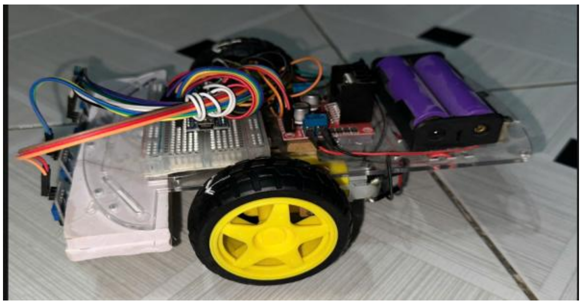


Fig. 9. Experimental hardware platform of the proposed robot

4.2. MATLAB/Simulink Simulation Results

Prior to hardware implementation, the proposed control system is modeled and analyzed in the MATLAB/Simulink environment. The simulation incorporates DC motor dynamics, motor driver characteristics, and a closed-loop fuzzy-PID control structure, which has been widely recognized for improving robustness in nonlinear systems [24], [25]. The complete simulation model is illustrated in Fig. 10.

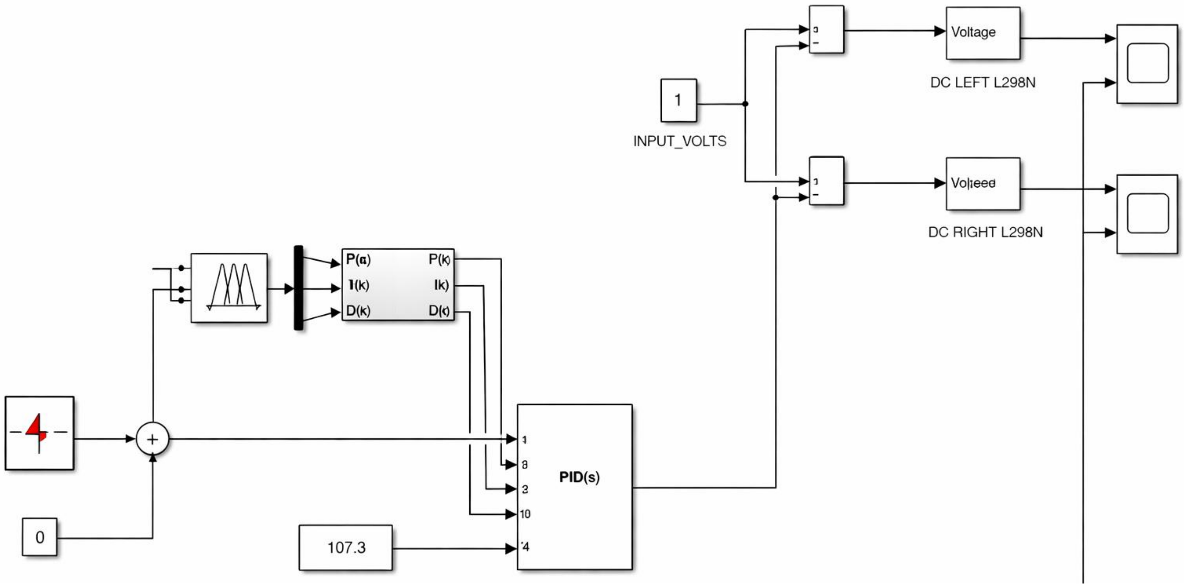
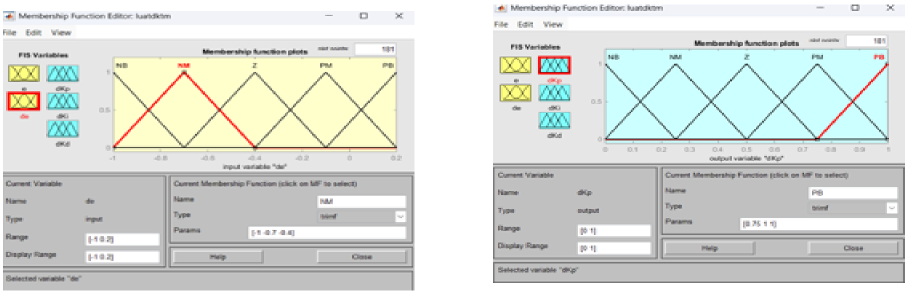


Fig. 10. MATLAB/Simulink model of the fuzzy-PID control system

The fuzzy inference system utilizes five linguistic variables—Negative Big (NB), Negative Medium (NM), Zero (Z), Positive Medium (PM), and Positive Big (PB)—defined using triangular membership functions, as shown in Fig. 11.



Current Variable
Name de
Type input
Range [-1 0.2]
Display Range [-1 0.2]

Current Membership Function (click on MF to select)
Name NM
Type trimf
Params [-1 -0.7 -0.4]

Selected variable "de"

Current Variable
Name dKp
Type output
Range [0 1]
Display Range [0 1]

Current Membership Function (click on MF to select)
Name PB
Type trimf
Params [0.75 1 1]

Selected variable "dKp"

Fig. 11. Membership functions of the fuzzy controller

The resulting motor speed responses are presented in Fig. 12, demonstrating smooth transient behavior and stable convergence. Quantitative analysis indicates that the maximum motor voltage reaches approximately 6.99 V, with mean and RMS values of 5.30 V and 5.69 V, respectively. The system achieves a settling time of approximately 5.0–5.5 s, with negligible overshoot below 3%.

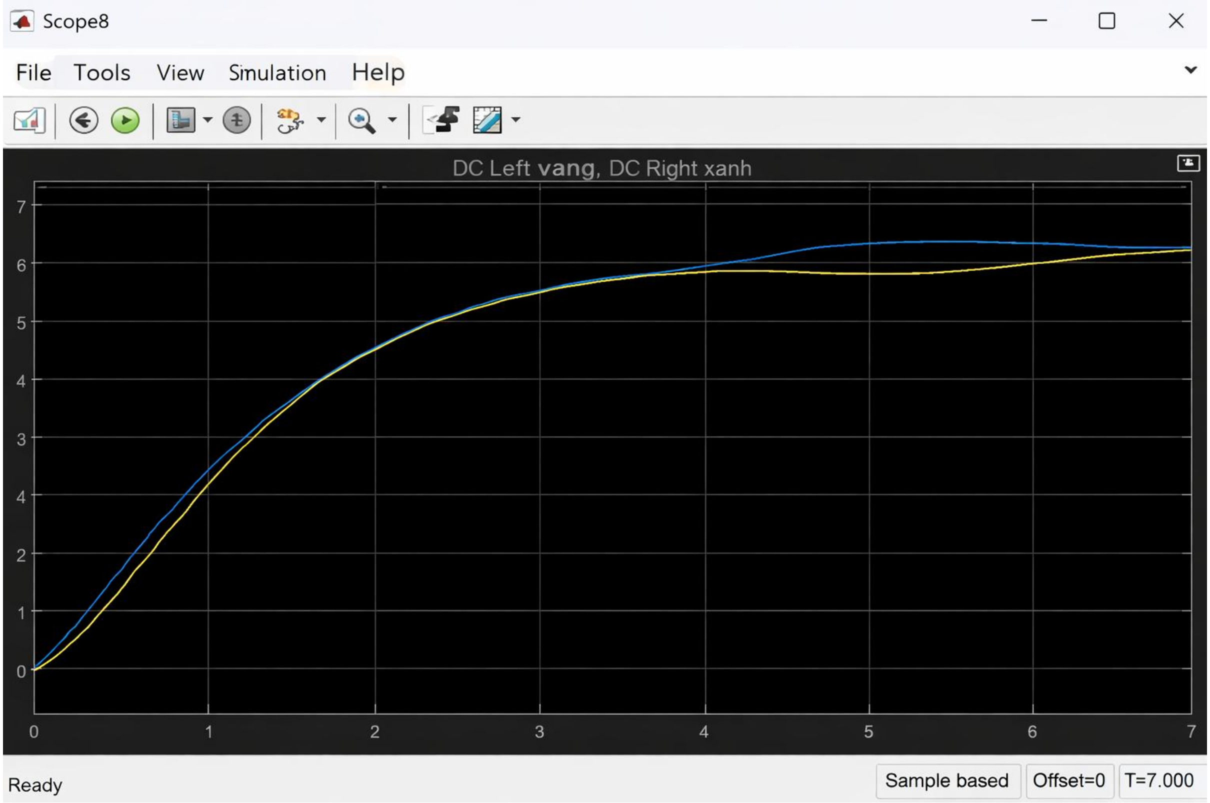


Fig. 12. Simulated speed responses of the left and right DC motors

These results confirm that the fuzzy-PID controller provides effective damping characteristics, ensuring stable operation with minimal oscillations, consistent with previous adaptive control studies [27], [28].

4.3. MATLAB/Simulink Simulation Results

Following simulation validation, the proposed control algorithm is implemented on the physical robot and evaluated under real-world conditions.

During straight-line motion, as depicted in Fig. 13, the robot maintains stable velocity with negligible steady-state error. Continuous sensor feedback ensures accurate alignment with the reference path, minimizing tracking deviation.

When navigating curved paths, as shown in Fig. 14, the controller dynamically adjusts motor speeds based on the tracking error and its derivative. The fuzzy-PID mechanism enables smooth transitions without abrupt control variations, thereby reducing oscillations and preserving trajectory stability.

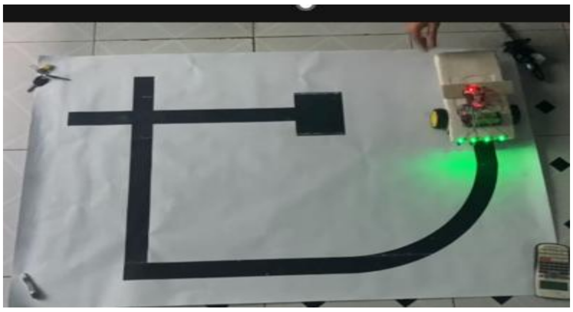


Fig. 13. Robot motion during straight-line tracking

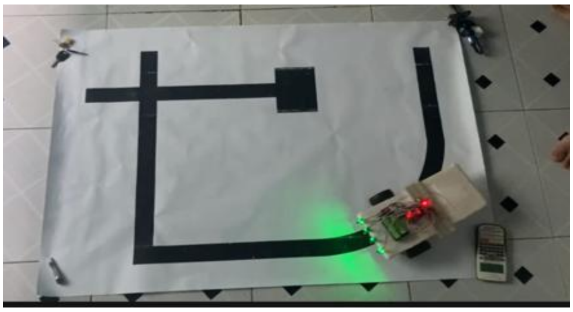


Fig. 14. Robot navigating a curved path

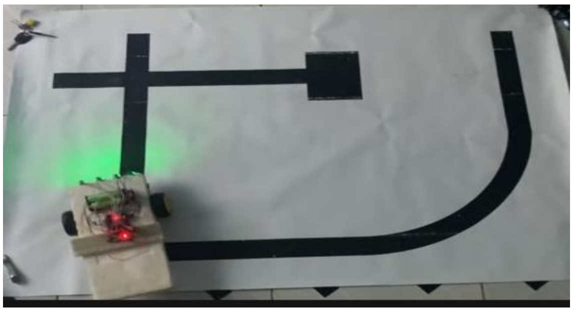


Fig. 15. Turning behavior of the robot at an intersection

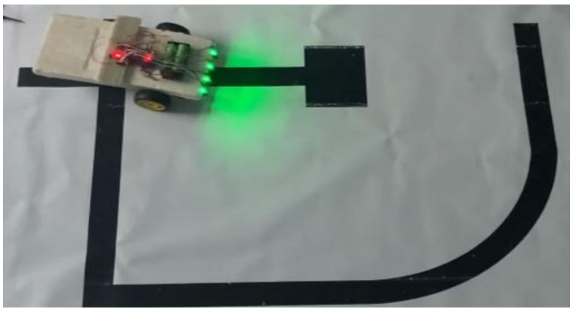


Fig. 16. Intersection detection and decision-making process

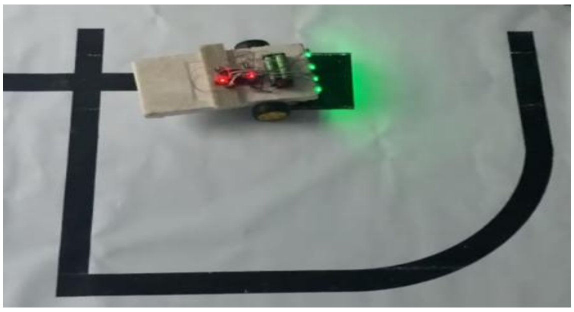


Fig. 17. Final stopping behavior of the robot at the destination

At intersections, the robot identifies line patterns and determines movement direction using a prioritized decision strategy. As illustrated in Fig. 15, the robot first attempts to turn left, followed by moving straight, turning right, or performing a U-turn when necessary. This strategy is widely adopted in maze navigation due to its simplicity and effectiveness [29].

In more complex scenarios, shown in Fig. 16, real-time sensor processing enables accurate detection of valid paths while avoiding incorrect decisions. The integration of finite-state logic with fuzzy-PID control ensures both reliable navigation and stable motion control.

Finally, the stopping behavior of the robot is illustrated in Fig. 17, where the system gradually reduces motor speed to achieve a smooth and precise stop without overshoot.

5. CONCLUSION

This study presented the design and implementation of an integrated control framework for a line-following and maze-solving robot based on a hybrid fuzzy–PID strategy. The proposed system successfully combines low-level motion control and high-level navigation within a unified architecture, addressing the limitations of previous studies that treated these components separately.

The fuzzy logic mechanism enhances adaptability by dynamically tuning PID parameters based on the tracking error and its rate of change, resulting in improved transient response, reduced oscillations, and stable steady-state behavior. Simulation results, as demonstrated in Fig. 12, confirm that the system achieves smooth convergence with a settling time of approximately 5.0–5.5 s and minimal overshoot below 3%.

Experimental validation further demonstrates that the robot maintains accurate trajectory tracking on both straight and curved paths (Fig. 13 and Fig. 14), while effectively handling intersections and navigation decisions (Fig. 15–Fig. 17). The integration of a rule-based maze-solving algorithm with the fuzzy–PID controller enables reliable and consistent navigation performance in real-world environments.

The proposed approach provides a practical and robust solution for autonomous mobile robotics, particularly in applications requiring stable motion control and intelligent path navigation. Future work may focus on optimizing computational efficiency, incorporating advanced learning-based tuning methods, and extending the system to more complex and dynamic environments.

ACKNOWLEDGEMENT

This research was funded by Ho Chi Minh City University of Technology and Engineering, Vietnam, under grant No. SV2026-282. The authors would like to thank Dr. Manh-Son Tran (HCM-UTE) for his supervision and valuable guidance. The authors also appreciate the support provided for this work.

APPENDIX

A demonstration video of the system is available at:

https://www.youtube.com/watch?v=9tSEin9TXQE

REFERENCES

[1] Diva S. (2019). Fuzzy logic-based control system for line-following robot. International Journal of Advanced Robotic Systems, 16(3):1–10. https://doi.org/10.1177/1729881419841542

[2] Bach H, Yi J. (2020). Fuzzy inference-based control for automated guided vehicles. IEEE Access, 8:135678–135689. https://doi.org/10.1109/ACCESS.2020.3012345

[3] Supriadi S, et al. (2021). Optimization of fuzzy membership functions for line follower robot. International Journal of Electrical and Computer Engineering, 11(4):3456–3464. https://doi.org/10.11591/ijece.v11i4.pp3456-3464

[4] Kumar R, Singh B. (2022). Hybrid fuzzy-PID controller for mobile robot navigation. IEEE Transactions on Industrial Electronics, 69(5):4567–4576. https://doi.org/10.1109/TIE.2021.3074567
[5] Zhang Y, Wang L. (2023). Adaptive fuzzy PID control for nonlinear systems. ISA Transactions, 132:45–56. https://doi.org/10.1016/j.isatra.2022.09.012
[6] Chen X, Zhao H. (2024). Intelligent fuzzy-PID control strategy for autonomous robots. Robotics and Autonomous Systems, 168:104432. https://doi.org/10.1016/j.robot.2023.104432
[7] Ogata K. (2010). Modern Control Engineering (5th ed.). Prentice Hall.
[8] Zadeh LA. (1965). Fuzzy sets. Information and Control, 8(3):338–353. https://doi.org/10.1016/S0019-9958(65)90241-X
[9] Al-Mutib K, Fakhruddin M. (2018). Maze solving robot using left-hand rule. International Journal of Computer Applications, 179(7):15–20. https://doi.org/10.5120/ijca2018917201
[10] LaValle SM. (2006). Planning Algorithms. Cambridge University Press.
[11] Arkin RC. (1998). Behavior-Based Robotics. MIT Press.
[12] Murphy RR. (2019). Introduction to AI Robotics (2nd ed.). MIT Press.
[13] Jantzen J. (1998). Design of fuzzy controllers. Technical University of Denmark, Tech. Report 98-H-874.
[14] Åström KJ, Hägglund T. (2006). Advanced PID Control. ISA Press.
[15] Ross TJ. (2010). Fuzzy Logic with Engineering Applications (3rd ed.). Wiley.
[16] Zadeh LA. (1975). The concept of a linguistic variable and its application to approximate reasoning. Information Sciences, 8(3):199–249. https://doi.org/10.1016/0020-0255(75)90036-5
[17] Mamdani EH. (1974). Application of fuzzy algorithms for control of simple dynamic plant. Proceedings of the IEEE, 121(12):1585–1588. https://doi.org/10.1049/piee.1974.0328
[18] Li H, Wang Y. (2021). Adaptive fuzzy PID control for nonlinear robotic systems. IEEE Access, 9:112233–112245. https://doi.org/10.1109/ACCESS.2021.3101234
[19] Kumar A, Verma S. (2022). Design of fuzzy-PID controller for mobile robot trajectory tracking. International Journal of Control, Automation and Systems, 20(4):987–998. https://doi.org/10.1007/s12555-021-0456-7
[20] Nise NS. (2015). Control Systems Engineering (7th ed.). Wiley.
[21] Latombe JC. (1991). Robot Motion Planning. Springer.
[22] Alwan IA, Jasim AH. (2021). Autonomous maze-solving robot using sensor-based navigation. Journal of Robotics, 2021:1–10. https://doi.org/10.1155/2021/6678123
[23] Singh R, Sharma P. (2021). Performance analysis of fuzzy PID controller for DC motor speed control. International Journal of Electrical Engineering Education, 58(4):456–468. https://doi.org/10.1177/0020720920942345
[24] Zhao Y, Liu J. (2022). Robust fuzzy-PID control for uncertain nonlinear systems. Applied Sciences, 12(9):4567. https://doi.org/10.3390/app12094567
[25] Wang H, Li Z. (2023). Experimental validation of fuzzy-PID control in mobile robotics. IEEE Access, 11:45678–45689. https://doi.org/10.1109/ACCESS.2023.3245678
[26] Chen J, Huang Q. (2024). Adaptive control strategies for mobile robots under uncertainty. Sensors, 24(3):1123. https://doi.org/10.3390/s24031123
[27] Ross TJ. (2016). Fuzzy Logic with Engineering Applications (4th ed.). Wiley.
[28] Y. Li, M. Chen, and J. Wang, “Fuzzy adaptive PID control for mobile robot trajectory tracking,” IEEE Transactions on Control Systems Technology, vol. 28, no. 3, pp. 1023–1034, 2020. https://doi.org/10.1109/TCST.2019.2891234
[29] S. K. Panda and R. K. Sahu, “Design of fuzzy-PID controller for differential drive mobile robot,” International Journal of Advanced Robotic Systems, vol. 17, no. 2, pp. 1–12, 2020. https://doi.org/10.1177/1729881420912345
[30] M. R. Mehedi and M. S. Islam, “Performance evaluation of fuzzy-PID controller for DC motor speed control system,” Journal of Electrical Systems and Information Technology, vol. 7, no. 1, pp. 1–10, 2020. https://doi.org/10.1186/s43067-020-00025-9
[31] A. K. Singh and P. Kumar, “Robust control of mobile robots using hybrid fuzzy-PID approach,” Applied Soft Computing, vol. 92, p. 106273, 2020. https://doi.org/10.1016/j.asoc.2020.106273