PID CONTROL FOR AUTONOMOUS OBJECT-FOLLOWING ROBOT

DUC-BAO-LONG NGUYEN*, MINH-KHOI HO, ANH-QUAN DAO, NHU-BAO TRUONG, THANH-BINH NGUYEN, QUANG-TRUNG NGUYEN, HOANG-NGHIA NGUYEN, DANG-KHOI NGUYEN, QUANG-HUNG LE, KY-TINH TRAN

Department of Automatic Control, Ho Chi Minh City University of Technology and Education (HCMUTE), Ho Chi Minh City, Vietnam

**Corresponding author: 22151111@student.hcmute.edu.vn*

(Received: 22 December 2024; Accepted: 31 January 2025; Published on-line: 1 March 2025)

ABSTRACT: This paper presents an experimental study on the PID algorithm applied to a vehicle system that maintains a fixed distance from an object. The research focuses on how changes in PID parameters affect the vehicle's behavior. The system comprises an SRF05 ultrasonic sensor connected to an Arduino, controlling the vehicle is forward and backward movements to follow the object. Through a series of experiments, analyses, and evaluations, the study explores the effectiveness of the PID controller and addresses challenges. The results help identify the optimal PID parameter values for the system systematically.

KEY WORDS: PID Control, Distance Control, Arduino, Object following robot.

1. INTRODUCTION

In the field of automatic control, the PID (Proportional-Integral-Derivative) controller is one of the most common and effective solutions for stabilizing and optimizing the performance of systems. PID is widely applied in various fields, such as autonomous robots [1], industrial production [2], healthcare [3], and vehicle control [4]. PID controller [5] operates based on three main components: the proportional component (P), which provides feedback proportional to the instantaneous error; the integral component (I), which calculates and adjusts based on the accumulated error; and the derivative component (D), which quickly responds to the rate of change of the error. Thanks to its flexibility, ease of implementation, and high efficiency in maintaining system stability, PID has become a preferred solution for handling complex dynamic control problems, including controlling a vehicle to follow a moving object.

Currently, mobile robots employ advanced methods such as the Intelligent Hybrid Technique [6], Fuzzy Logic and Genetic Algorithm [7], and Adaptive Robust Control [8]. While these methods provide high precision and adaptability, they often require complex hardware, sophisticated programming, and higher costs, which may not be practical for educational applications. The PID control method can address this issue due to its simplicity in design, experimental basis, and minimal hardware requirements. By employing this method, costs can be reduced, making it suitable for research and educational purposes. We use the Arduino platform, which is affordable, supported by a large and robust community, and well-suited for less complex research topics like ours.

Previous studies have shown that PID parameter tuning can be automated using immune algorithms [9], MATLAB [10], or fuzzy inference methods [11]. Our research focuses on an experimental trial-and-error approach to identify the optimal parameter set and assess each parameter's impact on the system.

In mobile robot systems, the problem of controlling a vehicle to follow an object has many practical applications, such as human-following robots in various fields, including healthcare [12], warehousing [13], and retail environments [14], or service robots (such as delivery robots) [15] following a person or a specific object. Our research aims to develop a vehicle capable of recognizing an object and moving forward and backward to follow it.

The problem of controlling a vehicle to follow an object presents significant challenges, including real-world environment factors like disturbances, friction, and obstacles that can affect control capability; the nonlinearity of vehicle and object behavior requiring flexible control strategies; and system delays in sensors and controllers that can impact instantaneous response. Therefore, the control system must ensure high accuracy, reduce processing delays, and maintain system stability to avoid oscillation or loss of control.

This paper aims to study and implement the PID control algorithm for a vehicle to follow an object, identify the optimal parameter set, and evaluate the impact of different parameters on the system. The research provides a systematic approach to tuning PID parameters through trial-and-error experimentation, offering practical insights into the impact of each parameter on system performance. It emphasizes the significance of low-cost solutions for educational and research purposes while enhancing understanding of PID in various systems.

2. ALGORITHM

2.1. Concept of PID

PID controller is a feedback loop control method widely used in automatic control systems. PID consists of three main components: proportional component (P), which responds proportionally to the current error. Integral component (I) which responds based on the accumulated error over time, and derivative component (D) which responds based on the rate of change of the error [16]

The general form of the PID control law is:

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

Where: K_p is the proportional gain, a tuning parameter, is the integral gain, a tuning parameter, K_d is the derivative gain, a tuning parameter, e(t)=SP-PV(t) is the error (SP is the setpoint, and PV(*t*) is the process variable), t is the time or instantaneous time (the present),τ is the variable of integration (takes on values from time 0 to the present).

2.2. Significance of the Components

2.2.1. Proportional Term

The proportional term produces an output value proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant K_p, which is called the proportional gain constant.

The proportional term is given by:

$$P_{out} = K_p e(t) \quad (2)$$

A high proportional gain results in a significant change in the output for a given change in the error. If the proportional gain is too high, the system can become unstable. In contrast, a slight gain produces a small output response to a significant input error and a less responsive or sensitive controller. If the proportional gain is too low, the control action may be too small when responding to system disturbances. Tuning theory and industrial practice indicate that the proportional term should contribute most of the output change.

2.2.2. Integral term

The contribution from the integral term is proportional to both the magnitude of the error and its duration. In a PID controller, the integral is the sum of the instantaneous error over time, giving the accumulated offset that should have been corrected previously. Accumulated error is multiplied by the integral gain (K_i) and added to the controller output.

The integral term is given by:

$$I_{out} = K_i \int_0^t e(\tau) \quad (3)$$

The integral term accelerates the process's movement towards the setpoint and eliminates the residual steady-state error that occurs with a pure proportional controller. However, since the integral term responds to accumulated errors from the past, it can cause the present value to overshoot the setpoint value.

2.2.3. Derivative term

The derivative of the process error is calculated by determining the slope of error over time and multiplying this rate of change by the derivative gain K_d. The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain, K_d.

The derivative term is given by:

$$D_{out} = K_d \frac{d}{dt} e(t) \quad (4)$$

Derivative action predicts system behavior and thus improves the settling time and stability of the system. An ideal derivative is not causal, so implementations of PID controllers include additional low-pass filtering for the derivative term to limit the high-frequency gain and noise. Derivative action is seldom used in practice, though – by one estimate in only 25% of deployed controllers – because of its variable impact on system stability in real-world applications.

2.3. Flowchart

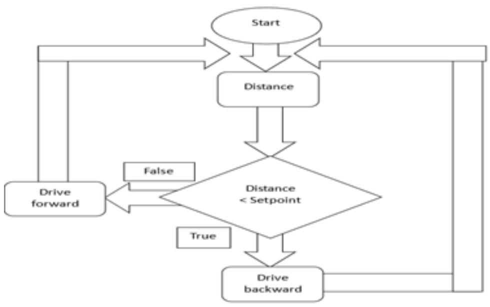

Fig.1. Flowchart

3. EXPERIMENTAL MODEL

This project uses an ultrasonic sensor to measure the distance from the model to an object. Then, the model tracks the object. The sensor readings are processed by a PID controller, which adjusts the vehicle's movements forward or backward to maintain a predetermined distance from the object.

Fig. 2. Experimental model

The components of the robot are presented below:

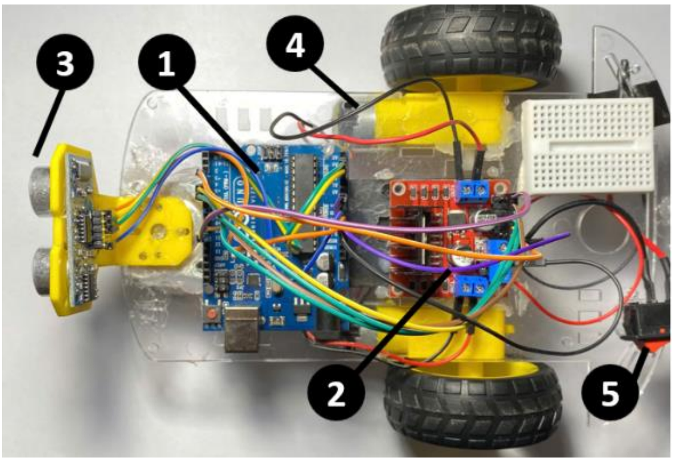

Fig. 3. Robot keeping a fixed distance from an object

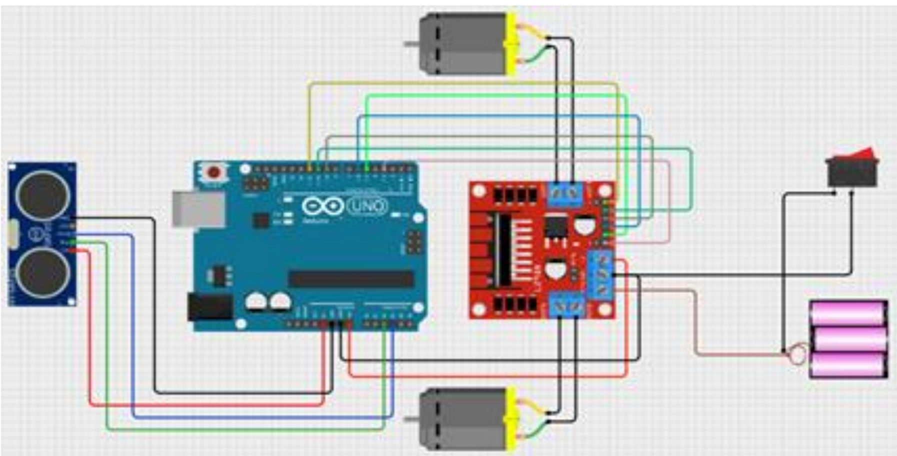

Fig. 4. Connection diagram of the circuit

Table 1. Elements of Robot

Station	Unit
1	Arduino UNO R3
2	L298N
3	SRF05-Ultrasonic Sensor
4	DC motor
5	Switch

The diagram shows a circuit built with the SRF05 ultrasonic distance sensor, the dual DC motor driver module L298, and the Arduino UNO R3 development board. We use three pins of the SRF05 ultrasonic sensor, including VCC, Trig, Echo, and GND. The VCC pin is connected to the 5V output from the Arduino. The GND pin is connected to the Arduino's GND. The Trig and Echo pins are connected to pins A2 and A3 of the Arduino, respectively. Arduino's pins 2, 5, 6, 9, and 10 control two DC motors through the L298 motor driver. These pins are connected to the inputs of the motor driver module, and the DC motors are connected to the outputs of the motor driver. ENA and ENB pins on the L298N are used to control motor speed using PWM signals from Arduino. A 12V power source is directly supplied to the L298N module to drive the two motors. Arduino gets its 5V power supply from the L298N through the Vin pin.

The robot detects and follows nearby objects, maintaining a safe distance of 25 cm. It will stop if an object moves quickly and gets more than 90 cm ahead. If the object remains stationary, the robot will move closer until it is 25 cm away and stop.

Based on the PID algorithm, the SRF05 ultrasonic sensor helps the robot measure the distance ahead and control its movement. The Arduino Uno serves as the central controller, receiving signals from the ultrasonic sensor and adjusting the speed and direction of the motors through the L298N H-bridge module. The ultrasonic sensor emits ultrasonic waves through the Trig pin and receives reflected signals through the Echo pin to calculate the distance. The distance data is sent to the Arduino for processing and adjustment to ensure the robot maintains a consistent distance from the object.

4. EXPERIMENT RESULTS

To examine how the parameter sets affect the vehicle, we conduct experiments sequentially with each set of parameters and evaluate their impact on the system. Our primary focus is to adjust the parameters of the PID controller to observe how the system responds to the setpoint. Initially, the vehicle is placed 30 cm away from the object. The survey results are presented below, with "DIS" being the distance measured by the vehicle's sensor and "Setpoint" being the initial set signal: 25 cm.

Table 2. Adjust elements of PID control

Parameter	K_p	K_i	K_d	Result
PD-1	0.01	0	0	Fig. 5.
PD-2	0.66	0	0	Fig. 6.
PD-3	2	0	0	Fig. 7.
PD-4	0.66	0.5	0	Fig. 8.
PD-5	0.66	1.4	0	Fig. 9.
PD-6	0.66	2	0	Fig. 10.
PD-7	0.66	1.4	0.1	Fig. 11.
PD-8	0.66	1.4	0.32	Fig. 12.
PD-9	0.66	1.4	1	Fig. 13.

4.1. Adjusting K_p

Initially, the controller was operated as a pure P controller, meaning it did not include I and D components. We started with a small K_p value and gradually increased it, observing the system's response improving progressively until it became unstable when K_p was further increased. We retained this K_p value to adjust the subsequent steps.

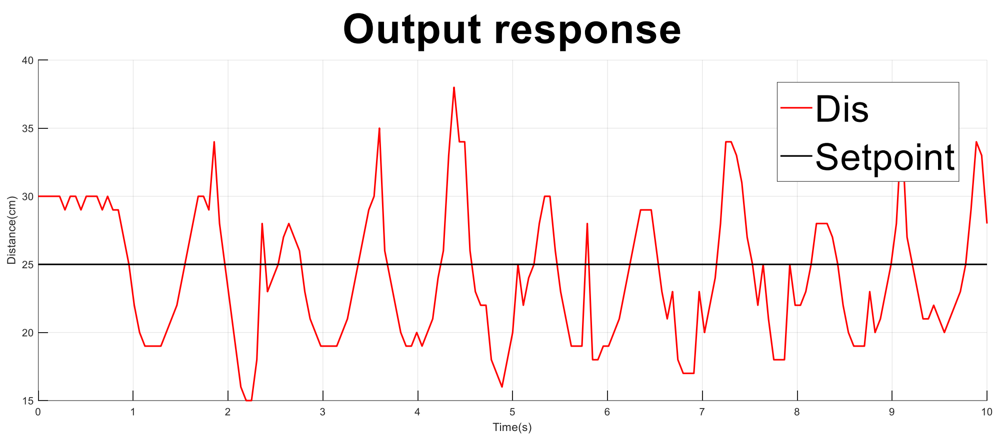

Fig. 5. PID-1's result

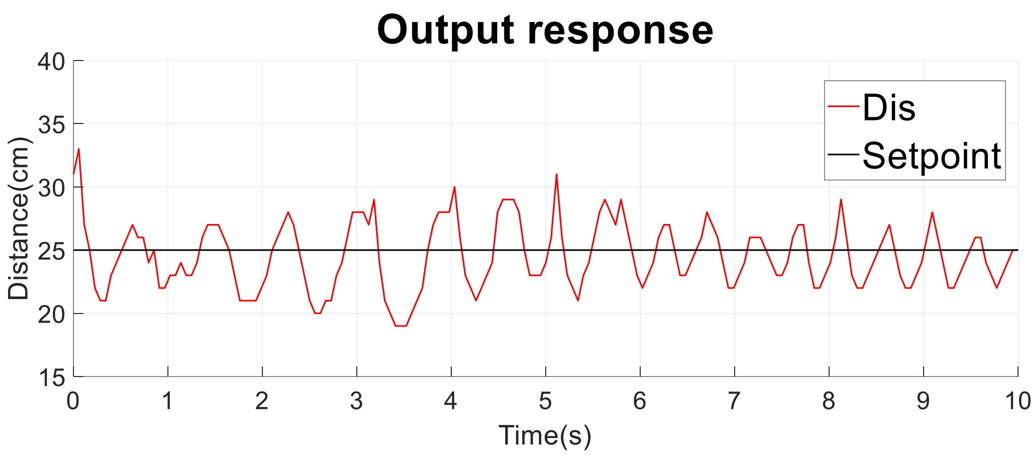

Fig. 6. PID-2's result

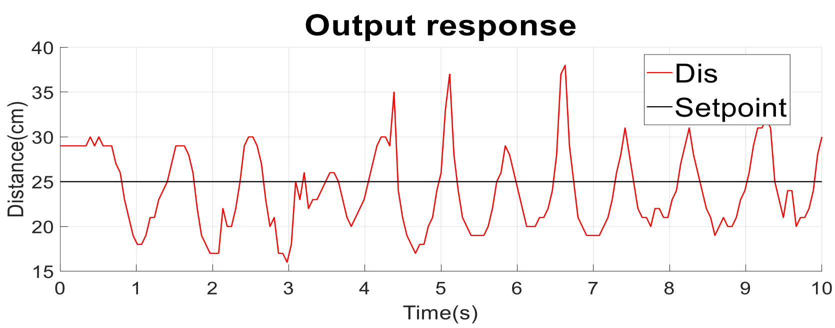

Fig. 7. PID-3's result

After adjusting K_p while keeping K_i and K_d values at zero, we observed an improvement in the vehicle's performance; its response capability increased slightly but was still not entirely satisfactory. If the K_p value is very low, the vehicle oscillates significantly, becomes unstable, and experiences high overshoot, as seen in Fig. 5. By adjusting the K_p value slightly, we noticed that after a period, the vehicle reduced its error and experienced less overshoot but could not eliminate it (Fig. 6.). Further increasing K_p caused the vehicle to return to an unstable state, as shown in Fig. 7. Since system performed best at $K_p = 0.66$, we retained this value and proceeded to adjust K_i.

4.2. Adjusting K_i

In the second step, the controller was operated as a PI controller. The integral component ensured the system reduced error over time with the K_p value retained from the previous step. We adjusted K_i starting from a small value and gradually increasing it until the system began to oscillate if this value was increased further. The K_p and K_i values were then retained for the next step.

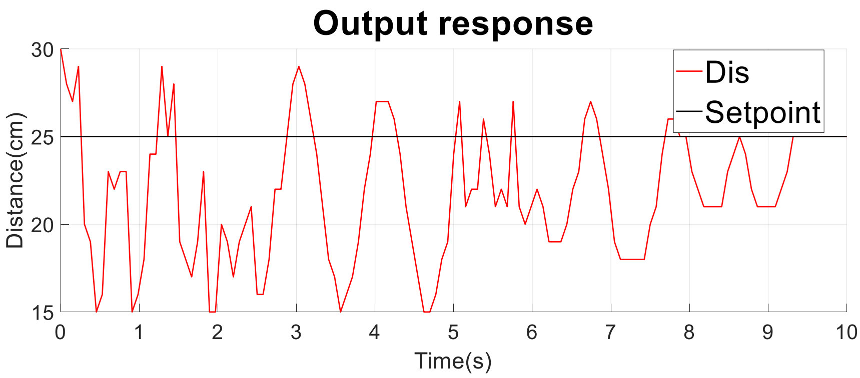

Fig. 8. PID-4's result

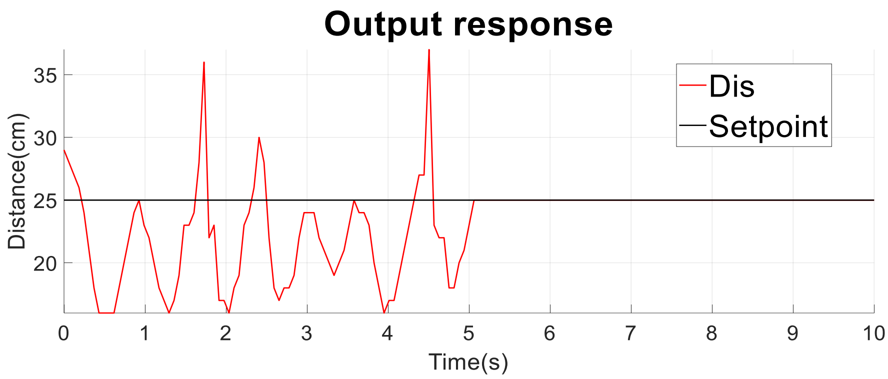

Fig. 9. PID-5's result

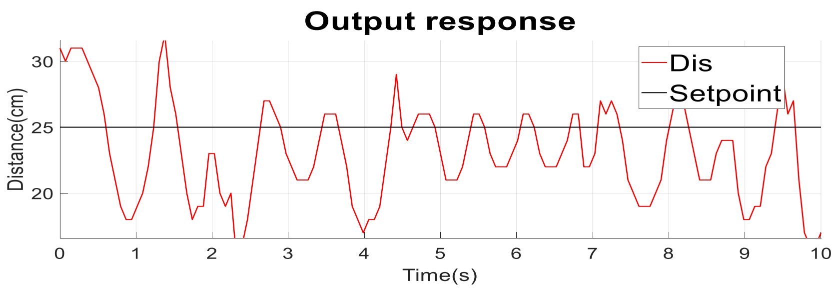

Fig. 10. PID-6's result

We kept the K_p value stable at 0.66 and sequentially changed the K_i value. As shown in Fig. 8, with the increase of K_i, the vehicle significantly reduces the error and, after a period, can stop precisely at the setpoint and stabilize. As K_i continues to increase, as seen in Fig. 9, the vehicle reduces the error, but the settling time remains long. If the K_i coefficient becomes too large, the vehicle becomes unstable (Fig. 10). We retained K_p = 0.66 and K_i = 1.4 to adjust K_d.

4.3. Adjusting K_d

After adjusting Ki, the system became stable with a small error but required a relatively long settling time. We refined the controller by adding the D component, aiming to reduce the settling time without increasing the error. The adjustment followed the same process described above, ultimately resulting in the optimal parameter set for the system.

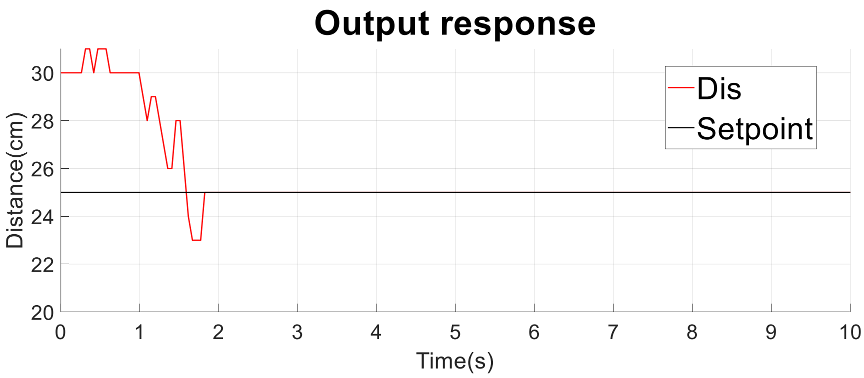

Fig. 11. PID-7's result

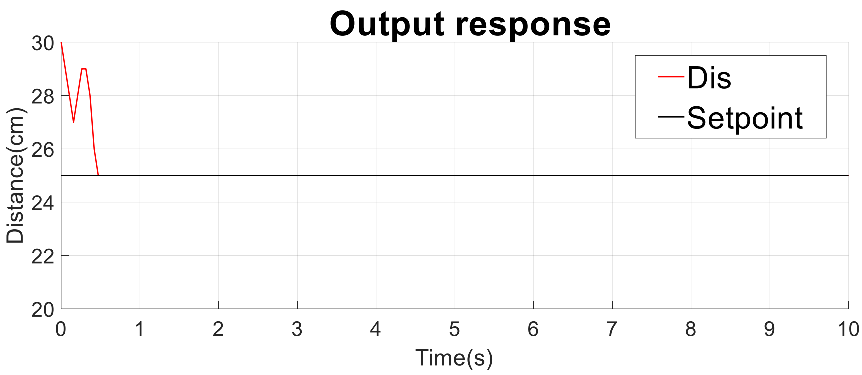

Fig. 12. PID-8's result

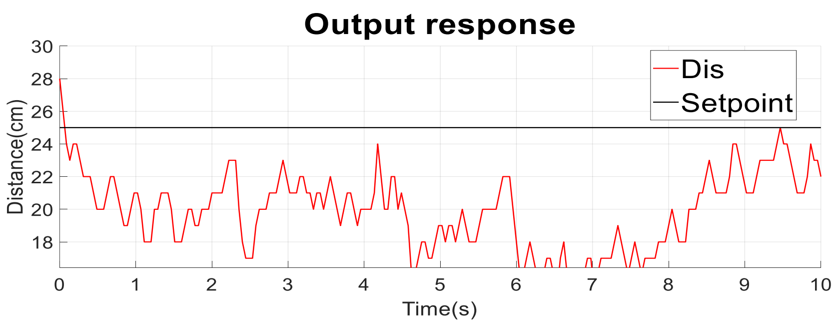

Fig. 13. PID-9's result

We kept K_p = 0.66 and K_i = 1.4 while varying the K_d values. With a slight increase in K_d, the vehicle's settling time decreased (Fig. 11). As K_d was further increased, as shown in Fig. 12, the vehicle's settling time continued to decrease without increasing the overshoot, allowing the vehicle to reach a stable state more quickly. However, if K_d was increased excessively, the vehicle became overly cautious of the setpoint and failed to reach the desired position, leading to instability (Fig. 13). We selected the value K_d = 0.32.

4.4. Optimal Results

After conducting experiments with various parameters, the team found the optimal parameters: K_p=0.66, K_i=11.4, K_d =0.32. With these parameters, the vehicle can quickly reach the setpoint and stop swiftly with low overshoot. Additionally, if the object moves, the vehicle can smoothly follow it.

5. CONCLUSION

In this study, the research team successfully designed and implemented a PID control system for a vehicle to follow an object. Through experimentation and evaluation, the PID control system demonstrated precise and rapid adjustments, enabling the vehicle to maintain its distance from the object and follow it successfully. This confirms the effectiveness and broad applicability of the PID control method in automation and mobile robotics systems.

Additionally, the experimental results showed that the system has good stability and responds well to various conditions. It further emphasizes the system's potential for real-world applications, particularly in fields such as industry, transportation, and services. However, this study has certain limitations that need to be considered. First, the experiments were conducted in a controlled environment, and the system's performance in more complex and dynamic real-world conditions has not been thoroughly tested. The vehicle encounters difficulties in real-world environments with many obstacles, as it can only follow objects within the front range, not to the sides or top and bottom. The system's operation also heavily depends on the sensors. Since the vehicle uses the SRF05 ultrasonic sensor, there is considerable noise, and the measurable distance is limited to 2-450 cm. The ultrasonic sensor emits a cone-shaped wave, so the farther the distance, the higher the error. Another drawback is that, due to the reliance on ultrasonic waves, the ultrasonic sensor system can be affected by interference from high-frequency signals, leading to significant impact in various sound environments. Additionally, PID parameters were tuned through a trial-and-error method, which is suitable for research and educational purposes but may not be ideal in real-world situations that require quickly finding the optimal parameters for the vehicle. Finally, while effective in simple scenarios, the PID control system may need further optimization to handle more complex behaviors or dynamic environments.

ACKNOWLEDGEMENT

This paper belongs to the HCMUTE project for students in the year 2025. We thank Dr. Hai Van Dong Nguyen for his invaluable supervision and support throughout our research. The link to the operation of the model is shown: https://www.youtube.com/watch?v=xQvY5oIFcQ0&feature=youtu.be.

REFERENCES

[1] Tawfiqur Rakib, M. A. Rashid Sarkar (2016)," design and fabrication of an autonomous fire fighting robot with multisensor fire detection using PID controller", 2016 International

Conference on Informatics, Electronics and Vision (ICIEV), pp. 909-914, doi: 10.1109/ICIEV.2016.7760132

[2] Parvathy Thampi M.S, G. Raghavendra (2021), "Intelligent Model for Automating PID Controller Tuning for Industrial Water Level Control System", 2021 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), 149-155, doi: 10.1109/ICDI3C53598.2021.00039

[3] JiTe Shi, et al (2023)," PID control of medical centrifuge based on improved PSO-BP neural network", 2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP), pp. 831-834, doi: 10.1109/ICSP58490.2023.10248482

[4] Atheer L. Salih et al (2010), "Modelling and PID controller design for a quadrotor unmanned air vehicle", International Conference on Automation, Quality and Testing, Robotics, vol.1, pp. 1-5, doi: 10.1109/AQTR.2010.5520914

[5] Ang K.H et al (2005), "PID control system analysis, design, and technology", in IEEE Transactions on Control Systems Technology, vol.13, no.4.doi: 10.1109/TCST.2005.847331

[6] Prases.K. Mohanty et al(2012)," Navigation of an autonomous mobile robot using intelligent hybrid technique", 2012 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), doi: 10.1109/ICACCCT.2012.6320757

[7] Gintautas Narvydas et al (2007)," Autonomous Mobile Robot Control Using Fuzzy Logic and Genetic Algorithm", 2007 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, doi: 10.1109/IDAACS.2007.4488460

[8] Wenrui Shi et al (2019),"Path tracking for wheeled mobile robot based on adaptive robust control". 2019 Chinese Control And Decision Conference (CCDC), doi: 10.1109/CCDC.2019.8833456

[9] Shi Zhongzhi (2002)," Auto-tuning of reference model based PID controller using immune algorithm", Computational Intelligence, Proceedings of the World on Congress on, pp. 483-388,doi: 10.1109/CEC.2002.1006282

[10] Tengku Luqman Tengku Mohamed et al(2010)," Development of Auto Tuning PID Controller Using Graphical User Interface (GUI) ", 2010 Second International Conference on Computer Engineering and Applications (ICCEA 2010), vol.1, doi: 10.1109/ICCEA.2010.101

[11] H.J. Live, et al (2017),"Simulation Research of Fuzzy Auto-Tuning PID Controller Based on Matlab", 2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC), pp. 180-183, doi: 10.1109/ICCTEC.2017.00047

[12] Yuxuan Tian (2022),"Design of Autonomous Following Vehicle for Rehabilitation", 2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), doi: 10.1109/ICAICA54878.2022.9844510

[13] Rey, R., et al,"Human-robot co-working system for warehouse automation", (2019), 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2019: 578–585, doi: 10.1109/ETFA.2019.8869178

[14] Nishimura, S. et al, (2006)," A study of robotizing daily items for an autonomous carrying system-development of person following shopping cart robot", 2006 9th International Conference on Control, Automation, Robotics and Vision, pp. 1–6 , doi: 10.1109/ICARCV.2006.345272

[15] Murad Mehrab Abrar et al, (2020)" An Autonomous Delivery Robot to Prevent the Spread of Coronavirus in Product Delivery System", 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), doi: 10.1109/UEMCON51285.2020.9298108

[16] Haowen Jiang (2024), "Overview and development of PID control", Applied and Computational Engineering , 66(1):187-191, doi: 10.54254/2755-2721/66/20240946