EPJ Web of Conferences **344**, 01047 (2025) https://doi.org/10.1051/epjconf/202534401047
AIPTEC 2025

A motion control system for a mobile floor cleaning robot using the proportional-integral-derivative method

Ahmad Sahru Romadhon[1*], *Ibnu* Irawan[2], *Mustika* Ratnawati[3], and *Achmad* Zain Nur[4]

[1]Mechatronics Engineering Department, University of Trunodjoyo Madura, Bangkalan, 69162, Indonesia
[2]Mechanical Engineering Department, University of Trunodjoyo Madura, Bangkalan, 69162, Indonesia
[3]Industial Engineering Department, University of Trunodjoyo Madura, Bangkalan, 69162, Indonesia
[4]Electrical Engineering Department, University of Trunodjoyo Madura, Bangkalan, 69162, Indonesia

Abstract. Floor cleaning robots are designed to perform tasks in unstructured environments based on human-provided logic. Many commercially available robots use inefficient random methods, creating a need for a more precise navigation and motion control system. *The research contribution is* the practical implementation and evaluation of a hybrid motion control system using the Proportional-Integral-Derivative (PID) method. This system integrates feedback from rotary encoders and a camera to minimize movement errors and overcome odometry drift. The robot maps a 200cm x 200cm arena. It combines a rotary encoder to measure distance traveled and a camera to detect visual markers for positioning and direction. The PID algorithm processes this fused sensor data to correct the robot's path. The research successfully determined the optimal PID parameters to be Kp = 1, Ki = 0.07, and Kd = 0.08. With these parameters, the system effectively reduced the average error rate to 6.126% across three tests, demonstrating stable and accurate movement from the start to the final position.

1 Introduction

Technological advancements in the Industry 4.0 era have produced innovations aimed at simplifying human tasks in daily life[1]. One of the most beneficial applications is in the field of smart robotics, which has grown rapidly to meet various human needs, including domestic cleaning[2]. With increasing activity and dynamic lifestyles, many people do not have enough time to clean their floors regularly, which can lead to cleanliness and health problems[3]. As a solution, automatic floor cleaning robots have been developed to address this challenge.

Previous research has explored various models of floor cleaning robots. For example, Asep Kosasih and Faisal Akbar developed the simple reflex autonomous smart mopping the floor (SRASMF) model, an intelligent robot equipped with sensors and actuators to move forward, turn, and stop independently. However, many commercially available floor cleaning robots still rely on random or wall-following methods, which are often inefficient and require manual control. To improve the performance and autonomy of the robot, a more precise navigation system is needed[4][5].

This research proposes the development of a motion control system for a floor cleaning robot using the Proportional-Integral-Derivative (PID) method. This system is designed to allow the robot to operate according to a predetermined area mapping. The robot's navigation is supported by a combination of a rotary encoder sensor to accurately measure the distance traveled based on wheel rotation and a camera as a visual detector to ensure the robot stays on the correct path[6][7].

The main goal of PID control is to achieve and maintain a stable value that corresponds to a predetermined setpoint, thereby minimizing movement error. By implementing the PID algorithm, which consists of three parameters—Proportional (Kp), Integral (Ki), and Derivative (Kd)—this floor cleaning robot is expected to move stably along the designated path, from the starting position to the end, with a high degree of precision[8][9]. Thus, this research focuses on the design and implementation of a PID-based motion control system to create a reliable and efficient floor cleaning robot[10].

The integration of multiple sensor types, often referred to as sensor fusion, is a critical strategy for overcoming the inherent limitations of individual sensors in mobile robotics[11][12][13]. Rotary encoders are highly effective for odometry, providing precise data on wheel rotation to estimate the distance traveled[6]. However, they are susceptible to cumulative errors arising from wheel slippage, uneven surfaces, or mechanical imperfections, which can lead to significant positional drift over time. This drift makes relying solely on odometry unreliable for tasks requiring long-term positional accuracy[14][15].

This integration is crucial as it addresses fundamental principles of kinematics and dynamics. The rotary encoder provides odometric data based on wheel kinematics, but this is susceptible to cumulative errors (drift). To counteract this, the camera provides an external reference frame, a concept derived from optical physics,

* Corresponding author: asr0354@gmail.com

allowing the system to visually detect markers and correct this kinematic drift.

To counteract the drift from odometry, this study incorporates a camera for external referencing and path correction. Computer vision allows the robot to recognize features or markers in its environment[5], providing absolute position feedback that is not subject to cumulative error[16]. When the camera detects a deviation from the intended path, this information is fed back to the control system. The PID controller then uses this visual feedback, in conjunction with the data from the rotary encoders, to compute a more accurate correction, ensuring the robot maintains its trajectory with greater reliability[17][9].

Therefore, the primary contribution of this research lies in the practical implementation and evaluation of a hybrid control system that synergizes the strengths of both internal (odometry from rotary encoders) and external (visual feedback from a camera) sensing methods. By demonstrating the effectiveness of the PID controller in processing this fused sensor data, this study aims to provide a robust and cost-effective solution for precise navigation in autonomous cleaning robots, addressing the common challenge of positional accuracy in real-world domestic environments[4][18].

The research contribution is the design and implementation of a PID-based motion control system that integrates rotary encoder odometry with camera-based visual feedback. This hybrid sensor fusion approach is evaluated to minimize positional errors and ensure the robot stably follows a predetermined path, addressing the common challenge of odometry drift in autonomous robots.

2 Methodology

The research methodology covers hardware design, control system architecture, and the operational workflow of floor cleaning robots.

2.1 Research methodology

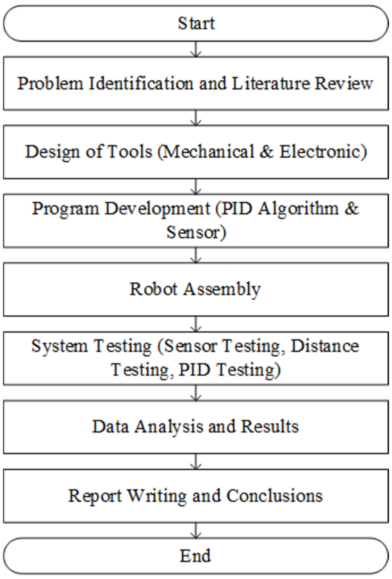

Fig. 1. Research methodology flowchart.

2.2 System design and components

The floor cleaning robot is designed with four wheels, consisting of two drive wheels and two free wheels. The mechanical design was created using Fusion 360 software, with overall dimensions of 40 cm in length, 40 cm in width, and 12 cm in height.

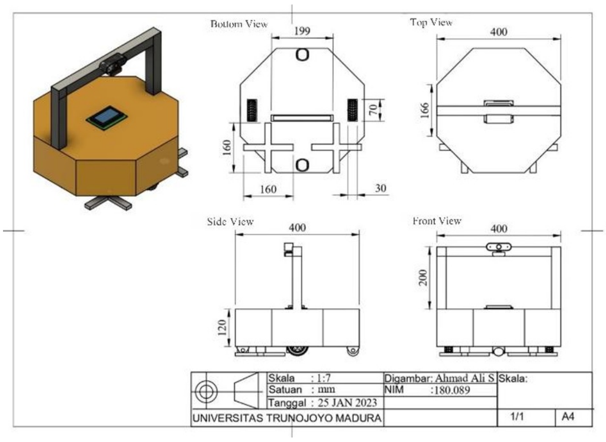

Fig. 2. 2D Design of floor cleaning robot.

This system is built using several integrated main components, including:

- Microcontroller: An Arduino ATmega 2560 serves as the central processing unit that manages input signals from sensors and generates output for actuators.
- Motion Sensor: Rotary Encoder sensors are used on both drive wheels to generate serial pulses that are translated into movement, position, and distance data[6].
- Visual Sensor: A camera functions to detect the robot's position, direction, and destination by identifying visual objects along the path[5]. In addition, a TCS34725 Color Sensor is used to detect color markers (red, green, blue) that indicate commands to stop or turn.
- Actuators: Two DC motors are used as the main wheel drives. Motor movement is controlled by a BTS7960 H-Bridge Motor Driver capable of regulating speed and direction of rotation (clockwise and counter-clockwise).
- Cleaning System: The robot is equipped with a vacuum cleaner and a rotating broom driven by a separate DC motor.
- Power Source: The entire system is powered by a battery as the main power supply.

2.3 Block diagrams and control systems

The robot motion control system is designed using the Proportional-Integral-Derivative (PID) method to minimize movement errors[9]. The architecture of this control system is illustrated in the block diagram in Figure 3.

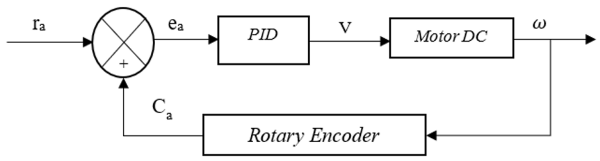

Fig. 3. PID control system block diagram.

The control process works in a feedback loop:

1. The setpoint (r_a) is set as the reference value or desired position target.
2. The Rotary Encoder sensor provides feedback in the form of actual wheel position data (ω), which becomes the control variable (C_a).
3. The difference between the setpoint and the actual data is calculated as the error value (e_a).
4. The PID controller processes this error value to generate an adjusted control signal (V).
5. This control signal is then sent to the DC motor to correct the speed and direction of rotation, so that the robot's movement follows the specified path.

2.4 Electronic circuits

The electronic design of this floor cleaning robot aims to integrate all components so that the system can function as a whole. The main circuit connects the microcontroller to various sensors and actuators.

2.4.1 Rotary encoder sensor circuit

The rotary encoder sensor is used to convert the rotational movement of the wheel into data that can be read by the Arduino ATmega 2560 microcontroller. The following is a microcontroller circuit with a rotary encoder sensor in Figure 4.

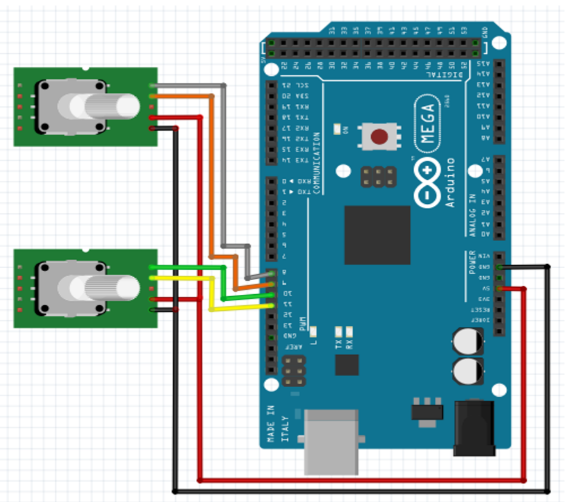

Fig. 4. Microcontrollers with rotary encoder sensors.

Here are the steps to connect a microcontroller with a rotary encoder sensor:

1. Connect VCC to the microcontroller's 5v.
2. Connect Ground to the microcontroller's ground.
3. Connect CLK to pin 8 of the microcontroller.
4. Connect DT to pin 9 of the microcontroller.
5. Connect CLK to pin 10 of the microcontroller.
6. Connect DT to pin 11 of the microcontroller.

2.4.2 BTS7960 motor driver circuit

The BTS7960 motor driver functions to control the voltage and rotation direction of the wheel drive DC motor. This driver allows the 12V motor to be controlled with a 5V signal from the microcontroller. The following is a diagram of the BTS7960 motor driver, DC motor, and microcontroller circuit in Figure 5.

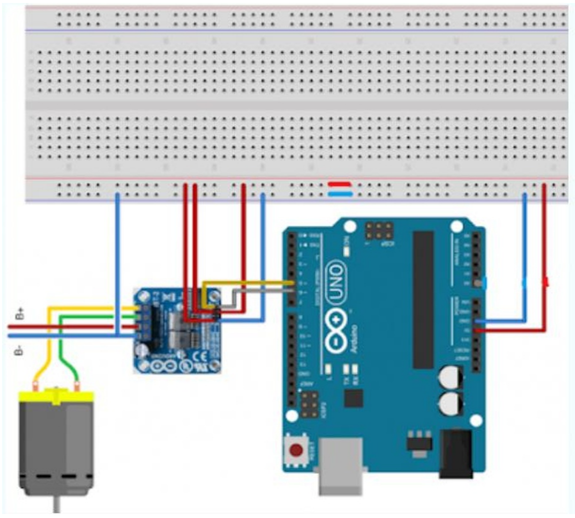

Fig. 5. BTS7960 motor driver, DC motor, and microcontroller.

The steps for assembling the BTS7960 motor driver, DC motor, and microcontroller are as follows:

1. Connect RPWM to pin 5 of the microcontroller.
2. Connect LPWM to pin 6 of the microcontroller.
3. Connect R_EN to 5V of the microcontroller.
4. L_EN is connected to the microcontroller's 5V.
5. VCC is connected to the microcontroller's 5V.
6. GND is connected to the microcontroller's GND.
7. B (+) is connected to the V+ input voltage from the battery.
8. B (-) is connected to the V- input voltage from the battery.

2.4.3 TCS34725 color sensor circuit

The TCS34725 sensor is used to detect color markers (red, green, blue) in the area that the robot will pass through. The following is the TCS34725 color sensor circuit and microcontroller in Figure 6.

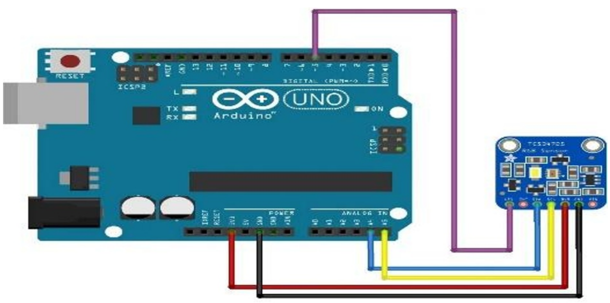

Fig. 6. TCS34725 sensor and arduino.

The steps for connecting the TCS34725 sensor and microcontroller are as follows:

1. Connect VCC to the 5v power supply.
2. Connect GND to the GND pin.

3. Connect SCL to the SCL pin.
4. Connect SDA to the SDA pin.
5. Connect the LED to pin 5.

2.5 System workflow

The operational workflow of the floor cleaning robot is illustrated in a flowchart. The process begins with system initialization, where the camera, vacuum cleaner, and broom are activated. The robot then begins to move along a pre-mapped path. The flowchart can be seen in Figure 7.

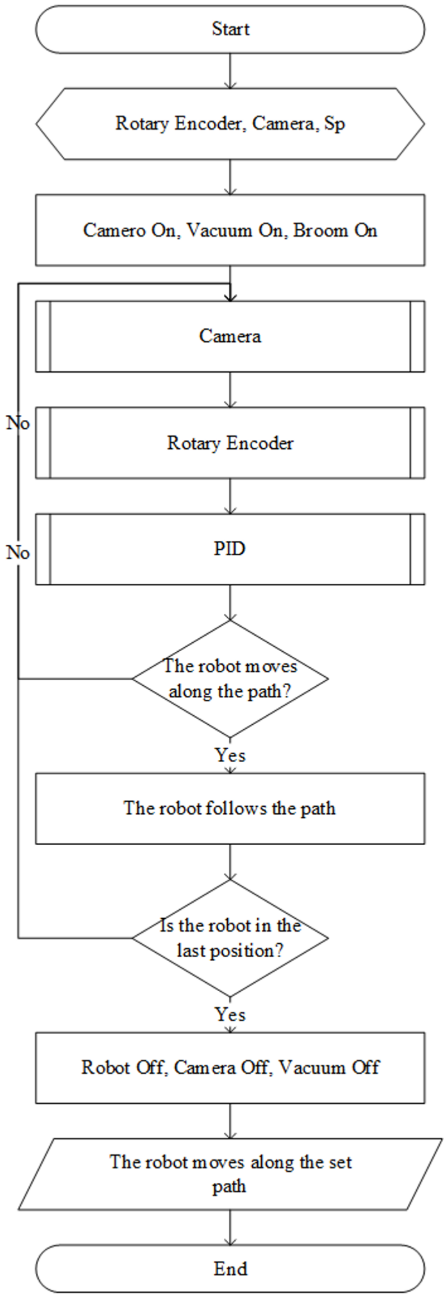

Fig. 7. Flowchart of system workflow.

During operation, the system continuously performs the following steps:

- The camera and rotary encoder provide real-time position and direction data.
- The PID controller analyzes this data to detect any deviations from the path. If an error is detected, the PID calculates the necessary correction to return the robot to its path.
- This process continues until the robot detects that it has reached its predetermined end position.
- Once the task is complete, all systems (robot, camera, vacuum cleaner) are automatically deactivated.

3 Results and discussion

This section presents the results of the system implementation and analysis, including the determination of PID parameters (tuning), a comparative analysis of system performance with and without the PID controller, the integration of the control system with sensor fusion, and a discussion of its limitations.

3.1 Analysis of System response: Before and after PID implementation

To evaluate the effectiveness of the PID controller, the robot's motion dynamics were tested under two conditions: without PID control and with PID control implemented. The difference in performance was significant, highlighting the controller's crucial role in achieving stable and accurate navigation.

Without a PID controller, the robot's movement was highly unstable. As shown in the test results on Figure 8 the system exhibited a progressive increase in error over time. This phenomenon, known as cumulative drift, is caused by uncorrected odometry errors from factors like minor wheel slippage and mechanical imperfections. The error value continuously deviated from the setpoint, indicating a lack of a corrective mechanism and an inability to maintain response stability. The system could not compensate for deviations, resulting in an ever-increasing steady-state error and ultimately causing the robot to fail to follow the designated path.

After implementing the PID controller, the system's dynamics changed dramatically. The PID controller provided a robust feedback mechanism that actively corrected deviations from the setpoint. As illustrated in Figure 10, the system demonstrated stable convergence toward the desired trajectory, significantly reducing the steady-state error and eliminating uncontrolled drift. This corrective action ensures that the robot can reliably maintain its course, which is essential for autonomous navigation tasks.

The performance of the PID controller was further evaluated by observing key response characteristics such as steady-state error, overshoot, and settling behavior. With the optimal parameters (Kp=1, Ki=0.07, Kd=0.08), the system demonstrated a rapid convergence to the setpoint with minimal oscillation, indicating a well-damped response. Compared to systems without PID, which exhibited cumulative drift and increasing error, the PID-controlled motion maintained trajectory stability throughout the testing path.

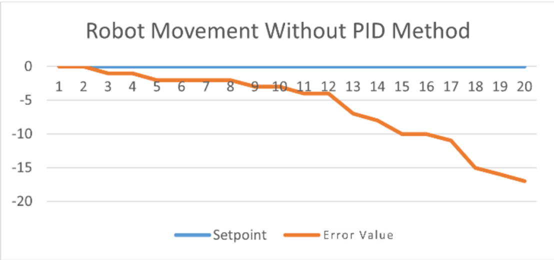

Fig. 8. Robot movement without PID control.

3.2 Analysis of PID parameter tuning (Kp, Ki, Kd)

The process of determining the optimal PID parameters—Proportional (Kp), Integral (Ki), and Derivative (Kd)—was carried out through a series of systematic tuning tests, as detailed in the research report. Each parameter plays a unique role in shaping the system's response:

- Proportional Gain (Kp): This parameter is responsible for speeding up the system's response to an error. A higher Kp value results in a faster reaction, but if set too high, it can lead to significant overshoot and oscillation around the setpoint.
- Integral Gain (Ki): The integral term works to eliminate residual or steady-state error by accumulating past errors over time. This ensures that the robot eventually reaches the exact target position. However, an excessively high Ki can cause instability.
- Derivative Gain (Kd): The derivative term helps to stabilize the system by anticipating future errors based on their current rate of change. It acts as a dampener, reducing overshoot and oscillations caused by the Kp and Ki terms.

Five tuning attempts were made to find the best balance. The table below summarizes how different parameter values affected the robot's performance:

Table 1. PID tuning test results.

Tuning Attempt	Kp	Ki	Kd	Observed Response
1	0,857	1,725	1,824	Highly unstable, severe oscillation, and large overshoot.
2	0,132	1,0	0,575	Irregular and unpredictable response.
3	0,112	1,265	0,112	Slow response with some oscillation remaining.
4	1,0	1,0	1,0	Faster response but still exhibited noticeable overshoot.
5 (Optimal)	1,0	0,07	0,08	Stable, fast response with minimal overshoot and no oscillation.

The response from the fifth and optimal tuning test is visualized in Figure 9.

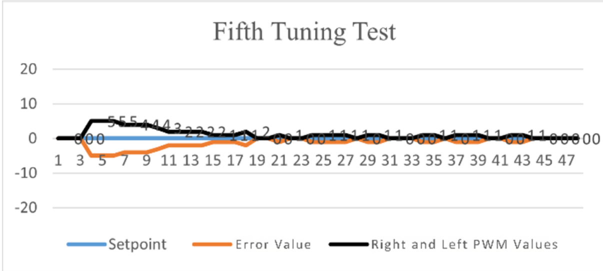

Fig. 9. Optimal PID tuning test results.

3.3 Justification for optimal parameter selection

The combination of Kp=1, Ki=0.07, and Kd=0.08 was chosen as the optimal set of parameters. This decision is justified by the system's performance during the fifth tuning attempt, which demonstrated the best trade-off between responsiveness and stability.

With these parameters, the robot achieved a stable response with minimal overshoot and was able to settle quickly at the setpoint without excessive oscillation, as shown in Figure 10. In comparison, earlier trials with higher Kp or Ki values resulted in instability, while lower values led to a sluggish response and residual errors. The fifth tuning attempt successfully minimized the positional error to an average of 6.126%, proving its effectiveness for the specific mechanical and environmental conditions of the robot.

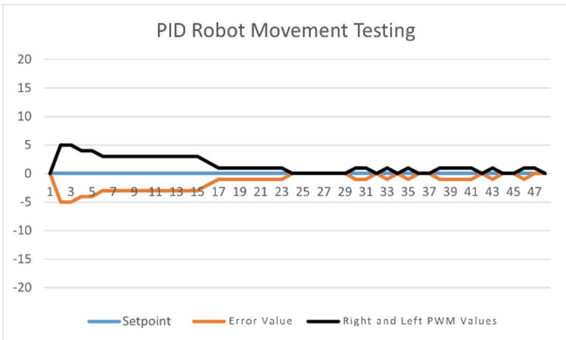

Fig. 10. Robot movement with PID control.

PID was chosen over other advanced control techniques (e.g., Fuzzy, Adaptive Learning MPC [19], or Deep Neural Networks [20]) due to its computational simplicity and compatibility with real-time embedded systems[21]. Although PID is effective, its performance remains sensitive to mechanical variations such as surface friction and weight distribution, which explains the residual error of 6.126%. These insights highlight the necessity of precise tuning to accommodate hardware characteristics.

3.4 Discussion

The implementation of the PID controller was analyzed based on the structure requested by the reviewer, focusing on key findings, comparison with existing work, implications, and limitations.

The primary finding of this research is the determination of optimal PID parameters, which were found to be Kp = 1, Ki = 0.07, and Kd = 0.08. As demonstrated in the results Figure 10, these parameters produced a stable and well-damped response. With these values, the control system successfully minimized the robot's movement error within the 200cm x 200cm arena, achieving a stable average error rate of 6.126% across three comprehensive tests.

This 6.126% error rate signifies a substantial improvement over the uncontrolled system (without PID), where the error was shown to accumulate rapidly, reaching a value of 17 and causing the robot to fail its designated path. When compared to similar studies, our findings are consistent and competitive. For instance, research by Dian Tresnawan et al., which also utilized a 200cm x 200cm mapped area, reported a stable error rate of 6.66%. Our result of 6.126% represents a slightly more precise performance. Furthermore, work by Bimantyoso Hamdikatama et al., which also employed a PID controller with rotary encoders, confirmed that this method significantly improves positional accuracy, supporting the methodology chosen for this study. Unlike the simple reflex autonomous model proposed by Asep Kosasih et al., this research implements a fixed-path mapping corrected by PID-based sensor fusion.

The implication of these findings is that a PID controller, when implemented as the core of a hybrid control loop leveraging sensor fusion (camera and rotary encoders)[22], is a highly effective and computationally simple strategy for this application[9][11]. The system successfully compensates for the cumulative odometry drift inherent in encoder-only systems. The camera provides external correction 16that recalibrates the robot's position, ensuring the long-term navigational accuracy that neither sensor could achieve alone[11].

The main strength of this system is its demonstrated ability to drastically reduce positional error and maintain path stability. However, this study has limitations, as indicated by the residual average error of 6.126%. This error is attributed to several uncompensated, real-world factors. These include non-linear mechanical factors such as minor wheel slippage and, significantly, an unbalanced load created by the placement of a laptop on the robot's chassis[23]. Furthermore, the use of two free-wheel casters was observed to contribute to minor instability. Future mechanical designs could mitigate this by using three or four omni-directional wheels.

4 Conclusion

Based on the research, implementation, and analysis conducted, the following conclusions are drawn:

1. Contribution & Results: The application of the Proportional-Integral-Derivative (PID) control method, with optimal parameters found to be Kp = 1, Ki = 0.07, and Kd = 0.08, was proven successful. The system's contribution lies in its effective use of sensor fusion (camera and rotary encoder) to minimize movement error to an average of 6.126% in a 200cm x 200cm arena.
2. Comparative Performance: In contrast, tests conducted without PID control showed highly unstable movement. The robot's path deviated from the setpoint to an error value of 17, confirming that the robot fails to reach its destination without the PID control system.
3. Limitations: Despite its success, the system retains a residual error of 6.126%. This limitation is attributed to mechanical factors, including an unbalanced load from an onboard laptop and the use of two free-wheel casters which reduced stability.
4. Future Work: For future development, several improvements are recommended. This includes exploring alternative control methods (such as Fuzzy Logic or Adaptive-PID), Model Predictive Control (MPC) [19][8][24], Deep Learning [20], or more advanced sensor fusion-based navigation methods such as Visual-Inertial Odometry [12] [25][26]. Mechanical enhancements should focus on achieving better stability by using four drive wheels (eliminating free-wheel casters) and refining components, such as soundproofing the vacuum cleaner for quieter operation.

References

1. L. Reid, G. Sisel, Digital care at home: Exploring the role of smart consumer devices. Health Place. **80**, 102961 (2023). https://doi.org/10.1016/j.healthplace.2022.102961
2. A. Joon, W. Kowalczyk, Design of Autonomous Mobile Robot for Cleaning in the Environment with Obstacles. Applied Sciences. **11**, 8076 (2021). https://doi.org/10.3390/app11178076
3. H. Herdiansya, Y. Pernando, Design Prototype Robot Vacuum Cleaner Based on Arduino Microcontroller with Autonomic Mode. TIN: Terapan Informatika Nusantara. **4**, 190–197 (2023). https://doi.org/10.47065/tin.v4i3.4215
4. P. K. Panigrahi, S. K. Bisoy, Localization strategies for autonomous mobile robots: A review. Journal of King Saud University - Computer and Information Sciences. **34**, 6019–6039 (2022). https://doi.org/10.1016/j.jksuci.2021.02.015
5. S. Cebollada, L. Payá, M. Flores, A. Peidró, O. Reinoso, A state-of-the-art review on mobile robotics tasks using artificial intelligence and visual data. Expert Systems With Applications. **167**, 114195 (2021). https://doi.org/10.1016/j.eswa.2020.114195
6. I. Akbar, N. Ismail, T. D. Rachmilda, Design of Angle Position and Temporary Velocity Detector Using Rotary Encoder KY-040. SENTER 2019: Seminar Nasional Teknik Elektro 2019. 287–293 (2020)
7. B. Artono, W. Nugroho, R. Wahyudi, Color-Based Image Processing for Autonomous Human Following Trolley Robot Navigation with Camera Vision. Journal of Computer Science and Engineering (JCSE). **5**, 20–38 (2024). https://doi.org/10.36596/jcse.v5i1.727

8. D. Chu, H. Li, C. Zhao, T. Zhou, Trajectory Tracking of Autonomous Vehicle Based on Model Predictive Control With PID Feedback. IEEE Transactions on Intelligent Transportation Systems. **24**, 2239–2250 (2023). https://doi.org/10.1109/TITS.2022.3150365
9. K. A. Rihem Farkh, Vision Navigation Based PID Control for Line Tracking Robot. Intelligent Automation & Soft Computing. **35**, 901–911 (2023). https://doi.org/10.32604/iasc.2023.027614
10. M. Jasim Mohamed, B. K. Oleiwi, A. T. Azar, A. R. Mahlous, Hybrid controller with neural network PID/FOPID operations for two-link rigid robot manipulator based on the zebra optimization algorithm. Frontiers in Robotics and AI. **11**, (2024). https://doi.org/10.3389/frobt.2024.1386968
11. V. Ušinskis, M. Nowicki, A. Dzedzickis, V. Bučinskas, Sensor-Fusion Based Navigation for Autonomous Mobile Robot. Sensors. **25**, (2025). https://doi.org/10.3390/s25041248
12. D. Solodar, I. Klein, VIO-DualProNet: Visual-inertial odometry with learning based process noise covariance. Engineering Applications of Artificial Intelligence. **133**, 108466 (2024). https://doi.org/10.1016/j.engappai.2024.108466
13. M. Filipozzi, L. Iafolla, C. Seppi, G. Rauter, P. Cattin, Miniaturized optical absolute rotary encoder: A proof of concept. Sensors and Actuators A: Physical. **394**, 116879 (2025). https://doi.org/10.1016/j.sna.2025.116879
14. H. Küçükerdem, C. Yilmaz, H. T. Kahraman, Y. Sönmez, Autonomous control of unmanned aerial vehicles: applications, requirements, challenges. Cluster Computing. **28** (2025). https://doi.org/10.1007/s10586-025-05418-6
15. P. Panigrahi, S. Bisoy, Localization Strategies for Autonomous Mobile Robots: A review. Journal of King Saud University - Computer and Information Sciences. **34**, (2021). https://doi.org/10.1016/j.jksuci.2021.02.015
16. V. Ušinskis, M. Nowicki, A. Dzedzickis, V. Bučinskas, Sensor-Fusion Based Navigation for Autonomous Mobile Robot. Sensors. **25**, (2025). https://doi.org/10.3390/s25041248
17. R. Farkh, K. Aljaloud, Vision Navigation Based PID Control for Line Tracking Robot. Intelligent Automation and Soft Computing. **35**, 901–911 (2023). https://doi.org/10.32604/iasc.2023.027614
18. I. Suwarno, W. A. Oktaviani, Y. Apriani, D. U. Rijalusalam, A. Pandey, Potential Force Algorithm with Kinematic Control as Path Planning for Disinfection Robot. Journal of Robotics and Control (JRC). **3**, 107–114 (2022). https://doi.org/10.18196/jrc.v3i1.11528
19. K. Zhang, Q. Sun, Y. Shi, Trajectory Tracking Control of Autonomous Ground Vehicles Using Adaptive Learning MPC. IEEE Transactions on Neural Networks and Learning Systems. 1–11 (2021). https://doi.org/10.110.1109/TNNLS.2020.3048305
20. Y. Xiao, X. Zhang, X. Xu, X. Liu, J. Liu, Deep Neural Networks With Koopman Operators for Modeling and Control of Autonomous Vehicles. IEEE Transactions on Intelligent Vehicles. 1–12 (2022). https://doi.org/10.1109/TIV.2022.3180337
21. M.-L. Zhang, Y.-J. Zhang, X.-L. He, Z.-J. Gao, Adaptive PID Control and Its Application Based on a Double-Layer BP Neural Network. Processes. **9**, (2021). https://doi.org/10.3390/pr9081475
22. L. Saad Saoud et al., HuBot: A biomimicking mobile robot for non-disruptive bird behavior study. Ecological Informatics. **85**, 102939 (2025). https://doi.org/10.1016/j.ecoinf.2024.102939
23. A. D. Sabiha, M. A. Kamel, E. Said, W. M. Hussein, ROS-based trajectory tracking control for autonomous tracked vehicle using optimized backstepping and sliding mode control. Robotics and Autonomous Systems. **152**, 104058 (2022). https://doi.org/10.1016/j.robot.2022.104058
24. T. Tolossa et al., Trajectory tracking control of a mobile robot using fuzzy logic controller with optimal parameters. Robotica. **42**, 1–24 (2024). https://doi.org/10.1017/S0263574724001140
25. N. Cohen, I. Klein, Inertial Navigation Meets Deep Learning: A Survey of Current Trends and Future Directions. Results in Engineering. **24**, 103565 (2024). https://doi.org/10.1016/j.rineng.2024.103565
26. C. Chen, X. Pan, Deep Learning for Inertial Positioning: A Survey. IEEE Transactions on Intelligent Transportation Systems. **25**, 10506–10523 (2024). https://doi.org/10.1109/TITS.2024.3381161