PID-CONTROLLED HUMAN DETECTION ROBOT WITH VISUAL PROCESSING ON ALPHABOT-2

HOANG-THONG NGUYEN, QUOC-THANG VO, MINH-THIET LE, CONG-TUAN TRUONG, DUY-DAT TRAN, CONG-HOANG-ANH PHAM, HUU-TAI CAO, GIA-BAO LAM, HOANG-ANH TRUONG, LE-BAO-LUAN TRAN, HOANG-QUANG-MINH NGUYEN, LE-HOANG-VIET NGUYEN, THI-HONG-LAM LE*

Ho Chi Minh City (HCMC) University of Technology and Education (HCMUTE), HCMC, Vietnam

**Corresponding author: lamlth@hcmute.edu.vn*

(Received: 13 April 2025; Accepted: 13 May 2025; Published on-line: 8 June 2025)

***ABSTRACT*:** This paper presents a human detection and alert robot system based on the AlphaBot2 platform and Raspberry Pi. The system employs a camera with a HOG-based human detection algorithm to locate the target within the frame and uses an ultrasonic sensor to measure the distance to the person. Based on the horizontal offset between the person and the frame centre, the PID controller adjusts the speeds of two DC motors to guide the robot smoothly and steadily toward the person. When the robot reaches a predefined distance from the detected human target, a buzzer is triggered. Through experiments, the effectiveness of the image processing and PID algorithm is evaluated, and optimal parameter values are identified for the system.

KEY WORDS: PID Control; Human Dection Robot; Raspberry Pi; AlphaBot 2; Image Processing.

1. INTRODUCTION

In recent years, the integration of visual processing and automatic control in robotics and automation has opened new opportunities for developing intelligent systems across various applications, including surveillance [1], inspection [2], and emergency response [3]. Motivated by the goal of supporting rescue operations, this study presents the design and implementation of an autonomous robot capable of detecting humans, approaching them within a safe distance, and issuing a warning signal. To facilitate development, the AlphaBot2 platform is utilized, an integrated mobile robot chassis designed for research and educational robotics projects. The system includes a compact frame, dual motor modules, a motor driver, infrared sensors, and an interface board compatible with the Raspberry Pi. Thanks to its modular structure and ease of hardware integration, AlphaBot2 is well-suited for developing autonomous robotic systems incorporating image processing and control algorithms.

Modern object detection techniques such as YOLO (You Only Look Once) [4], SSD (Single Shot Detector) [5], and deep convolutional neural networks (CNNs) [6] have demonstrated high accuracy in real-time scenarios. In this study, the Histogram of Oriented Gradients (HOG) [7] combined with Support Vector Machine (SVM) [8] is chosen for human detection due to its simplicity, high reliability, and relatively low computational requirements. This algorithm can be easily implemented using predefined function calls in the OpenCV library in Python without requiring retraining of the detection model. In addition to human detection, HOG+SVM can also be used for posture recognition (e.g., standing, sitting, lying), handwritten character recognition, and general object detection.

With the advancement of robotic systems, mobile robots have employed advanced control strategies such as adaptive control [9], fuzzy logic [10], and model predictive control [11] to improve accuracy and robustness in dynamic environments. While these methods yield highly precise results, they often require powerful hardware, complex programming, and high development costs, factors that are unsuitable for low-cost applications or educational research. The PID (proportional–integral–derivative) control [12] strategy was chosen for this project due to its simplicity, experimental nature, and effectiveness, even with minimal investment. The PID control algorithm is implemented on a Raspberry Pi to regulate the robot's motion. By continuously adjusting motor speed based on the distance error between the robot and the detected object, the PID controller ensures smooth and stable object-approaching capability.

Previous studies have demonstrated that the process of tuning PID parameters can be automated through various methods, such as immune algorithms [13], MATLAB-based simulations [14], and fuzzy inference methods [15]. In contrast, this study adopts an empirical trial-and-error strategy to determine the most effective parameter configuration and to analyze the influence of each parameter on the robot's overall control performance. PID controllers continue to be a fundamental component in numerous application domains, including industrial automation [16], vehicular systems [17], and healthcare technologies [18].

This work addresses a critical need in rescue operations, where traditional search methods are often time-consuming, costly, and may result in victims being found too late. By proposing a low-cost, vision-based robotic system capable of detecting and approaching humans autonomously, this study offers a practical and scalable solution for rapid deployment in emergencies. Unlike prior works focusing on high-end technologies, the originality of this study lies in its integration of classical PID control with efficient human detection on a lightweight, modular platform, providing both technical insights and real-world applicability for time-sensitive, resource-limited scenarios.

Controlling a mobile robot to approach a human target in real-world environments introduces a range of technical challenges. These include external disturbances such as obstacles and surface friction, unpredictable behaviours of the robot and target due to system nonlinearity, and latency in sensor readings or actuator responses. These difficulties are further compounded in emergency rescue scenarios, where operational reliability is critical. To address these issues, the control mechanism must deliver precise responses, maintain low processing delays, and ensure overall system stability to prevent oscillatory or erratic movements.

The core aim of this work is to design and implement a vision-based human detection system integrated with a PID control algorithm for object tracking. Additionally, the research seeks to determine optimal controller parameters and assess their impact on system behaviour. Since the HOG and SVM algorithms are readily available and well-supported within the Python ecosystem, this project does not focus on developing new human detection methods. Instead, it directs its efforts toward investigating and optimizing the PID control strategy. By concentrating on the tuning of the PID parameters, the study aims to enhance the robot's ability to approach and align accurately with the detected human targets, thereby improving overall system performance in practical scenarios.

2. ALGORITHM

2.1. Image Processing

2.1.1. Histogram of Oriented Gradients (HOG)

The Histogram of Oriented Gradients (HOG) is an image feature extraction technique based on the distribution of gradient orientations of intensity. HOG is widely used in object recognition tasks, particularly in human detection. The algorithm operates by dividing the input image into small spatial regions called cells, computing gradient vectors at each pixel within a cell, and then constructing histograms of gradient directions for each cell. These histograms are subsequently normalized over larger spatial regions known as blocks, which reduces the influence of lighting conditions and contrast variations. The final output is a feature vector that effectively represents the shape and texture of objects while maintaining robustness against minor geometric and illumination changes.

2.1.2. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised machine learning algorithm commonly used for binary classification tasks. In human detection systems, the SVM is trained to distinguish between image regions containing humans and those that do not, using HOG features as input. The SVM algorithm identifies an optimal hyperplane that separates the data into two classes with the maximum margin. When combined with HOG features, the SVM enables accurate classification of image regions, thereby identifying the presence and location of humans in real-world environments.

The HOG + SVM approach has been proven effective and highly accurate in numerous studies and applications involving human shape recognition, see Fig. 1.

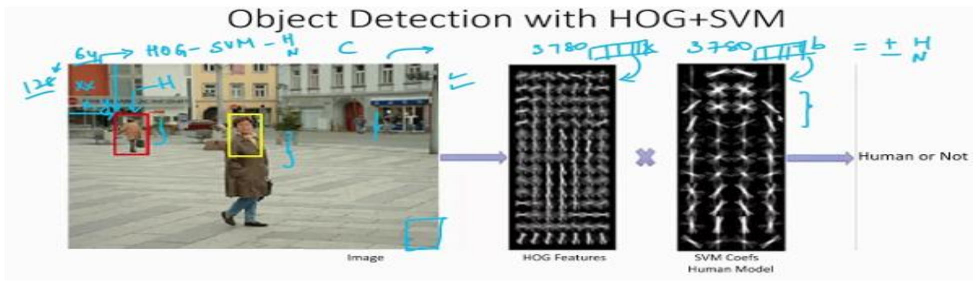

Fig. 1. Object Detection with HOG+ SVM

2.1.3. HOG-SVM for Human Detection

In this study, human detection was performed using the Histogram of Oriented Gradients (HOG) feature descriptor combined with a Support Vector Machine (SVM) classifier. The HOG parameters were set as follows: cell size of 8×8 pixels, block size of 2×2 cells (16×16 pixels), block stride of 8×8 pixels, and nine orientation bins covering 0° to 180°. A linear SVM kernel was used for classification due to its efficiency in binary detection tasks. The window size for detection was 64×128 pixels, which is standard for pedestrian detection.

During real-time processing, frames captured by the Raspberry Pi camera were analyzed using OpenCV's hog.detectMultiScale() function with a scale factor of 1.05 and a detection threshold of

-0.5 to reduce false positives. The SVM was trained using positive and negative samples from the INRIA Person Dataset with a regularization parameter C = 0.01, optimized using the Sequential Minimal Optimization (SMO) algorithm. The resulting bounding box coordinates were passed to the motion control system for PID-based regulation.

2.2. PID

2.2.1. Concept of PID

A proportional–integral–derivative controller (PID controller or three-term controller) is a feedback-based control loop mechanism commonly used to manage machines and processes that require continuous control and automatic adjustment. It is typically used in industrial control systems and various other applications where constant control through modulation is necessary without human intervention.

The general form of the PID control law is:

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

Where: is the proportional gain, a tuning parameter, is the integral gain, a tuning parameter, 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), is the time or instantaneous time (the present), is the variable of integration (takes on values from time 0 to the presen).

2.2.2. Significance of the Components

The Proportional (P) term responds to the present error, generating an output proportional to its magnitude. By applying immediate corrective action, the P term minimizes errors quickly.

The proportional term is given by:

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

The Integral (I) term addresses any persistent errors or long-term deviations from the setpoint by accumulating the error over time. By integrating the error signal, the I term ensures that the system approaches and maintains the setpoint accurately, eliminating steady-state errors.

The integral term is given by:

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

The Derivative (D) term anticipates future changes in the error by evaluating its rate of change. This approach dampens oscillations and stabilizes the system, especially during transient responses.

The derivative term is given by:

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

Figure 2 is a diagram that visually represents the three main components of a PID (Proportional-Integral-Derivative) controller in control systems. The three blocks labelled "Integral (I)," "Proportional (P)," and "Derivative (D)" correspond to the fundamental parts of a PID controller, which are used to regulate processes such as motor speed, temperature, or position. The illustration to the right depicts the typical PID control symbol, showing how these three components work together to adjust and stabilize the system output. The combined action of these components allows the system to respond accurately and quickly to changes or deviations, maintaining desired performance and reducing error.

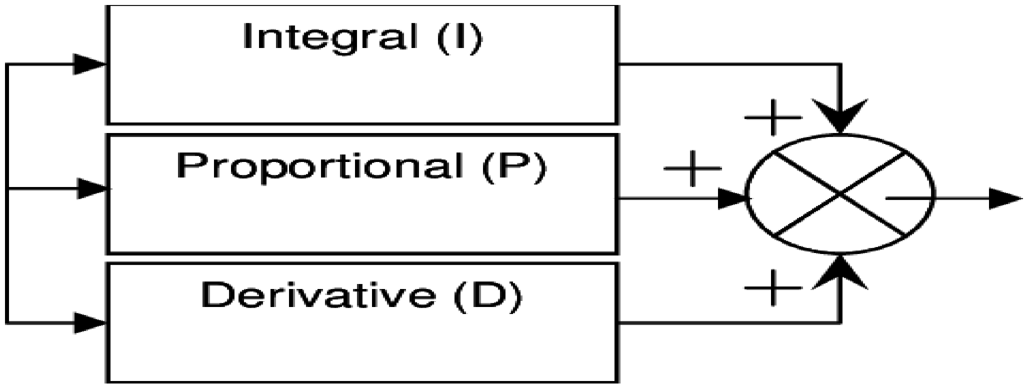

Fig. h2. Structure of a PID controller

2.3. Flowchart

Figure 3 describes the flowchart of a process for a robotic or sensor-based system. The process begins with the system starting up and initiating a rotating motion, likely to scan its surroundings. Once the system is in operation, it checks for the presence of a person. If a person is detected, the system proceeds to move forward toward them. As it approaches, the system monitors the distance to the person. When the distance reaches 50 cm, a buzzer is activated, serving as a signal or alert. After the buzzer sounds, the process concludes, and the system stops. This setup is typically used in automated systems, where a robot approaches a person and alerts them upon reaching a predetermined proximity.

Figure 4 is a flowchart that outlines a process for a system utilizing a PID controller, likely for line following or similar navigation tasks. The process begins with the system starting up and reading the pixel error, which indicates any deviation from a desired path or line. The PID controller processes this error to determine corrective actions.

If the error is greater than zero, the system adjusts by making the right motor slower and the left motor faster to correct the path. If the error is equal to zero, meaning the system is on the correct path, both motors are kept at the same speed. If the error is less than zero, indicating deviation in the opposite direction, the system compensates by making the right motor faster and the left motor slower. These adjustments help the system maintain or return to the desired trajectory.

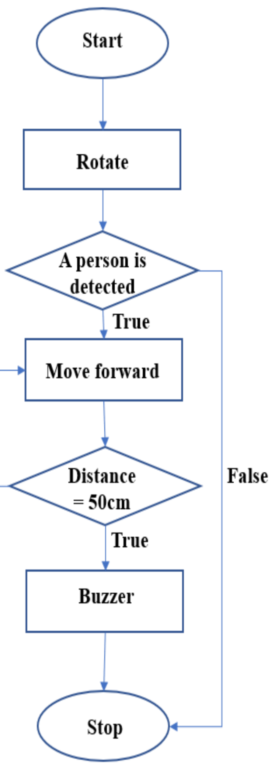

Fig. 3. System overview

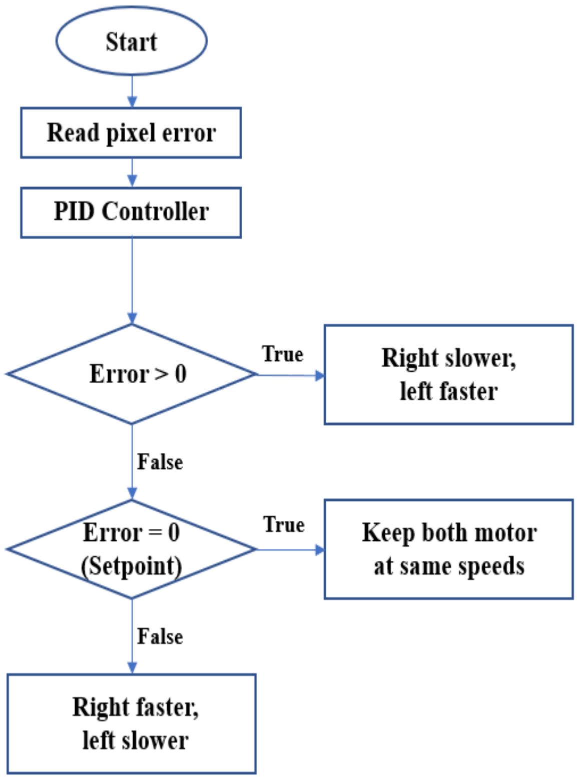

Fig. 4. PID controller

3. EXPERIMENTAL MODEL

This project is built using the AlphaBot2-Pi robotic platform, which integrates a Raspberry Pi with a camera module and several onboard sensors to detect the presence of a human, see Fig. 5. The Raspberry Pi serves as the central controller, processing real-time video from the PiCamera using OpenCV and a HOG-based person detection algorithm.

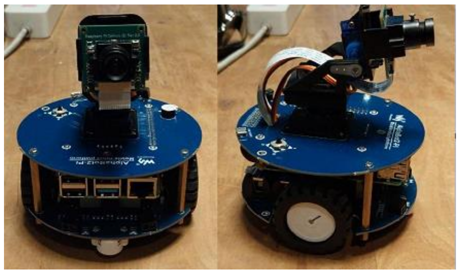

Fig. 5. Experimental model

The main components of the robot and the structure of the AlphaBot2 are illustrated below; see Fig. 6 to 9:

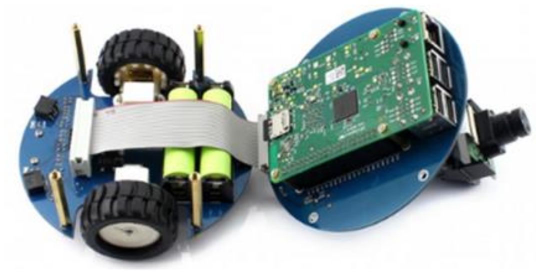

Fig. 6. Raspberry Pi is mounted on the AlphaBot2 chassis using standoffs and connected via GPIO.

Fig. 7. Raspberry Pi Camera

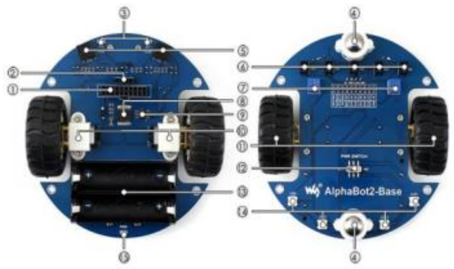

Fig. 8. AlphaBot2- Base

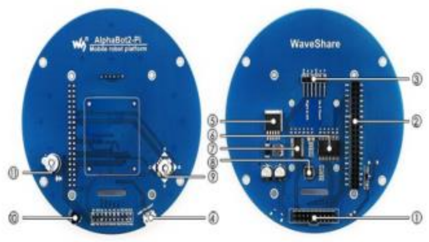

Fig. 9. AlphaBot2- Pi

Table 1: Elements of AlphaBot 2- Base

Station	Unit
1	AlphaBot2 control interface
2	Ultrasonic module interface
3	Obstacle avoiding indicators
4	Omni-direction wheel
5	ST188
6	ITR20001/T
7	Potentiometer
8	TB6612FNG
9	LM393
10	N20 micro gear motor
11	Rubber wheels
12	Power switch
13	Battery holder
14	WS2812B
15	Power indicator

Table 2: Elements of AlphaBot 2- Pi

Station	Unit
1	AlphaBot2 control interface
2	Raspberry Pi interface
3	Servo interface
4	USB TO UART
5	LM2596
6	TLC1543
7	PCA9685
8	CP2102
9	Joystick
10	IR receiver
11	Buzzer

Regarding the AlphaBot2-Pi platform, a mobile robot chassis is specifically designed for integration with the Raspberry Pi. The platform features a two-layer stackable structure, where the lower layer consists of the AlphaBot2 mainboard, housing dual DC motors with encoders, a motor driver (TB6612FNG), line-tracking sensors, an ultrasonic interface, and infrared obstacle sensors. The upper layer is reserved for mounting the Raspberry Pi board, which serves as the central processing unit for all vision and control algorithms.

The Raspberry Pi is mounted on top of the AlphaBot2 board using four brass standoffs and screws, which provide a stable and secure connection while ensuring proper alignment of the 40-pin GPIO header. This direct stacking configuration eliminates the need for jumper wires and allows seamless hardware interfacing through the GPIO pins. The PiCamera module (Rev 2.0) is connected to the Raspberry Pi via the CSI (Camera Serial Interface) port. It is affixed using a custom bracket or adhesive mount positioned at the front of the robot, with a forward-facing orientation to capture a live video feed of the surrounding environment.

Once a person is detected, their horizontal position relative to the frame centre is used to calculate the positional error, which is input into a PID controller. The PID controller then adjusts the speed of the two DC motors, creating differential speed to steer the robot and maintain alignment with the detected person. If no human is detected within the camera frame, the robot initiates a search behaviour by rotating in place. This scanning motion is performed by applying opposite speeds to the motors, enabling the robot to rotate until a person is detected or a timeout occurs.

The system's mobility is achieved via two independently driven wheels powered by the DC motors. Speed and direction control are executed through GPIO12 (PWM0) and GPIO13 (PWM1) for pulse-width modulation, and GPIO5, GPIO6, GPIO20, and GPIO21 for motor direction signals. An ultrasonic distance sensor (HC-SR04) is mounted at the front of the chassis to measure the proximity of detected targets. It connects to GPIO17 (Trig) and GPIO27 (Echo), with a voltage divider applied to the Echo signal to step it down from 5V to 3.3V, ensuring compatibility with the Raspberry Pi's input thresholds. Power for the entire system is provided by a 12V battery pack regulated through an LM2596 buck converter, which steps down the voltage to 5V. This 5V line powers the Raspberry Pi as well as the AlphaBot2 peripherals. A push-button connected to GPIO7 allows manual activation or deactivation of the detection system.

When the distance between the robot and the detected person is less than 50 centimetres, the buzzer is activated as an alert signal. The integration of camera vision, ultrasonic sensing, PID control, and motor feedback enables the AlphaBot2-Pi to operate autonomously and respond intelligently to its environment.

All components are securely mounted and wired using the standard AlphaBot2 assembly kit; see Tables 1 and 2. The compact and modular design of the chassis facilitates efficient cable management and ensures robustness during motion. This setup enables the AlphaBot2-Pi to function as a fully autonomous mobile robot capable of detecting humans and navigating accordingly using real-time image processing and PID-based motion control.

4. EXPERIMENTAL RESULTS

This section presents the experimental results obtained from the human detection and distance-based alerting system implemented on the AlphaBot2 platform. First, the performance of the visual detection algorithm is evaluated by assessing its accuracy under varying distances and angles between the robot and the target person. Following this, the effectiveness of different PID parameter sets is analyzed to identify the optimal configuration that minimizes alignment error and reduces the time required for the robot to reach the predefined alert distance. The results aim to validate the robustness and responsiveness of the proposed system in a simulated environment.

Due to the limitations of the camera's detection range, pre-captured images of a human figure were used to simulate human presence for the robot's visual recognition process, see Fig. 10.

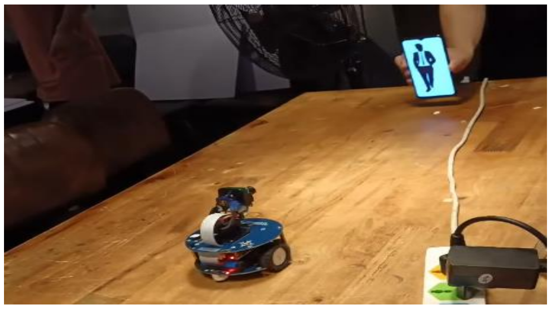
Fig. 10. Robot Approaching Simulated Human Image

4.1. Human Detection

4.1.1. When Robot is Aligned with the Human

To evaluate the effectiveness of the system in this case, the robot was positioned directly in front of the person at distances ranging from 50 cm to 3 meters under stable indoor lighting conditions. The 3-meter mark was determined as the maximum range at which the system could reliably detect a person, given the hardware limitations of the prototype. Detection results at each distance were recorded over 20 test iterations. The detection results at each distance were recorded and summarized in Table 3.

Table 3: Human detection accuracy at various distances

Distance to human (cm)	Detection Attempts	Successful Detections	Detection Accuracy (%)
50	20	20	100
100	20	18	90
150	20	16	80
200	20	15	75
250	20	13	65
300	20	10	50

Based on practical observations, aside from the issue of distance, the incorrect detections primarily occur due to false positives, where the algorithm mistakenly identifies objects with shapes similar to that of a human, such as chair legs, table legs, or lamp posts, as a person. These background objects often create gradient patterns similar to a standing figure, leading the HOG-based detection system to register them as human detections incorrectly. Additionally, the limited camera resolution and suboptimal lighting conditions exacerbate the issue by reducing the contrast and clarity of actual human contours. These factors highlight the need for improved feature differentiation or the integration of additional sensor data to minimize false alarms.

4.1.2. When Robot is Misaligned with Human

In the case when the robot is misaligned with the person (i.e., the person is not immediately visible), the robot must rotate to search for the target. Experiments were conducted with four different angular deviations and two distance levels (100 cm and 200 cm). Each condition was tested 20 times, and the results are presented in Tables 4 and 5.

Table 4: Human detection accuracy at different angle deviations (distance is 100cm)

Angle Deviation (Degrees)	Detection Attempts	Successful Detections	Detection Accuracy (%)
45	20	18	90
90	20	17	85

135	20	16	80
180	20	14	70

Table 5: Human detection accuracy at different angle deviations (distance is 200cm)

Angle Deviation (Degrees)	Detection Attempts	Successful Detections	Detection Accuracy (%)
45	20	14	70
90	20	14	70
135	20	13	65
180	20	12	60

From the results and observations, it can be seen that when the robot is placed at an angular offset from the person's position, its ability to detect the target significantly decreases due to the person being outside the direct field of view of the camera. The robot must initiate a search behaviour involving sequential rotation, which introduces latency and is susceptible to environmental factors such as lighting conditions, obstructions, or objects with human-like shapes. At larger offset angles (e.g., 135° and 180°), detection becomes more difficult, as the camera needs to scan a wider area to locate the person, leading to increased response time and a higher rate of detection failures. Furthermore, if the robot does not rotate quickly enough or does not follow the correct scanning direction, it may fail to bring the person into the frame, resulting in missed detections.

4.2. PID Controller

In order to evaluate the control performance with different PID gain settings and identify the optimal set, the robot was initially placed at a position 2 meters away from the human target in a straight line. From this starting point, experimental data were collected, including the time taken for the robot to reach the target position at a distance of 50 cm from the human, the time when the robot emitted a buzzer alert and stopped, the distance between the robot's stopping point and the 50 cm mark. If the recorded distance is negative, it indicates the robot stopped before reaching the target. Conversely, a positive value means the robot overshot the target. All experimental results corresponding to the different PID settings are detailed in Tables 6, 7, and 8.

Another important metric collected was the pixel error of the robot. This error represents the deviation between the desired target position (centre of the camera's frame) and the actual detected position of the human within the camera frame at the moment the robot stops and emits the buzzer signal. Additionally, the variation of the pixel error throughout the robot's motion was recorded and plotted in graphs (Figures 11–19) to illustrate better the influence of the PID controller on system stability. If the detected human position deviates to the left of the centre of the camera frame, the pixel error will have a negative value; if it deviates to the right, the pixel error will have a positive value. When the detected position and the centre of the frame are perfectly aligned, the pixel error will be zero. At a distance of 50 cm, based on calculations, each pixel corresponds to approximately 0.183 mm. This conversion allows readers to interpret the final pixel error values better; however, for cases where the robot did not precisely stop at 50 cm, slight deviations in the corresponding distance per pixel may occur.

To determine the optimal set of PID parameters for the system, a trial-and-error method was employed. This approach was chosen because it is the most straightforward to implement and allows direct observation of how each parameter affects the system's behaviour, which is particularly valuable for educational and research purposes. The tuning was conducted sequentially to isolate the influence of each term. First, the integral and derivative gains were set to zero, and only the proportional gain (Kp) was adjusted. This allowed us to assess the system's basic responsiveness. Once a suitable Kp was identified, producing a stable response with moderate oscillation, the derivative gain (Kp) was introduced while keeping Ki=0. The Kd value was increased gradually to reduce overshoot and dampen the oscillations. Finally, with Kp and Kd fixed, the integral gain (Ki) was tuned to eliminate any steady-state error, completing the PID parameter search.

4.2.1. Tuning Kp

First, we adjusted the proportional gain (Kp) while setting Ki and Kd to zero. After finding the optimal Kp value, we proceeded to tune Ki and finally Kd.

Table 6: Effect of Kp values on experimental datas

Kp	Time to reach 50 cm (s)	Time to Stop and Buzz (s)	Distance at Stop vs Target (cm)	Final Pixel Error (px)	Plot
0.2	8.5	9.3	+10	110	Fig.11.
0.55	7.5	8.0	+5	51	Fig.12.
1.2	9.0	9.7	+12	90	Fig 13.

With the initial Kp set to 0.2, the robot was able to move towards the target; however, the time to reach the 50 cm mark was relatively long, the robot overshot the target, and the buzzer alert was delayed by 0.8 seconds relative to arrival. Moreover, the pixel error at stopping was large, and as shown in Figure 11, the system exhibited significant overshoot and instability.

Increasing Kp to 0.55 resulted in much more favourable outcomes, as evidenced in Table 6. The corresponding pixel error graph clearly showed reduced overshoot and improved system stability. The robot's stopping position became more acceptable, and the final pixel error was less than half of the previous case.

Further increasing Kp to 1.2, however, worsened the results compared to Kp = 0.2, with increased overshoot and severe instability (Figure 13). Thus, the optimal proportional gain was determined to be Kp = 0.55.

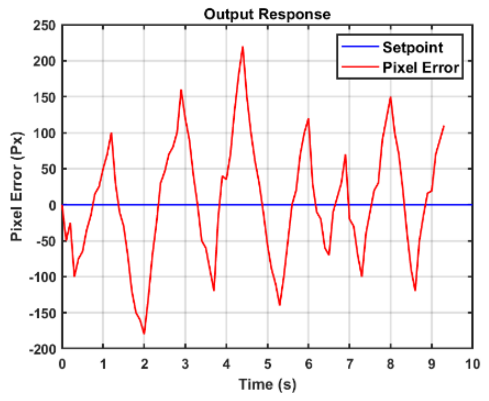

Fig. 11. PID's result (Kp= 0.2, Ki= 0, Kd= 0)

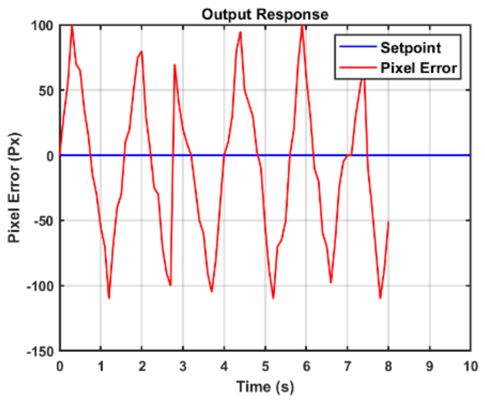

Fig. 12. PID's result (Kp= 0.55, Ki= 0, Kd= 0)

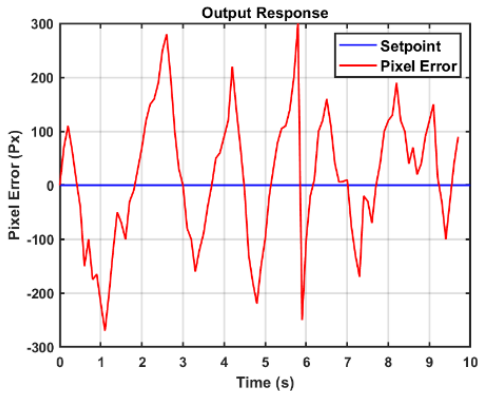

Fig. 13. PID's result (Kp= 1.2, Ki= 0, Kd= 0)

4.2.2. Tuning Ki

With Kp fixed at 0.55 and Kd still at zero, we gradually increased Ki to determine the optimal integral gain.

Table 7: Effect of Ki values on experimental datas

Ki	Time to reach 50 cm (s)	Time to Stop and Buzz (s)	Distance at Stop vs Target (cm)	Final Pixel Error (px)	Plot
0.1	6.5	6.7	+3	25	Fig.14
0.25	5.6	5.9	+1.5	14	Fig.15
0.4	7.1	7.8	+2	37	Fig.16

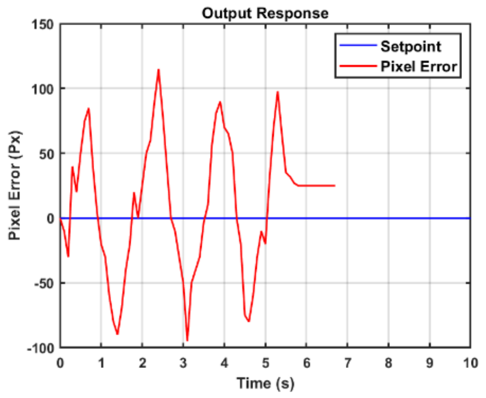

Fig. 14. PID's result (Kp= 0.55, Ki= 0.1, Kd= 0)

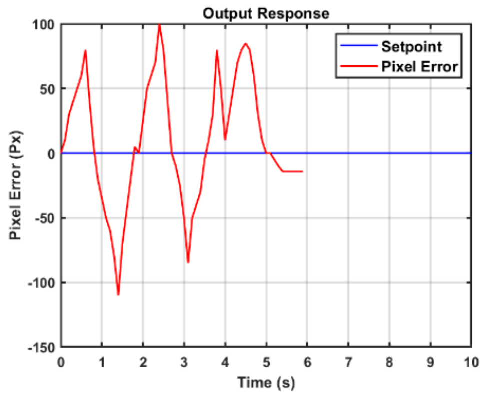

Fig.15. PID's result (Kp= 0.55, Ki= 0.25, Kd= 0)

Starting with Ki = 0.1, significant improvements were observed. The robot reached the target faster, the buzzer delay was reduced, and the stopping accuracy improved markedly. The final pixel error decreased to 25, and the system became more stable with a clear convergence to the setpoint near the end of the motion (Figure 14).

Increasing Ki to 0.25 further enhanced performance. The system exhibited greater stability, and the final pixel error decreased even more (Figure 15).

However, at Ki = 0.4, the system performance deteriorated. Instability re-emerged (Figure 16), and in practical observation, the robot initially stopped at the 50 cm point but failed to maintain the stop, moving forward after about 1.5 seconds. The final pixel error was significantly higher compared to the previous two cases.

Therefore, the optimal integral gain was determined to be Ki = 0.25.

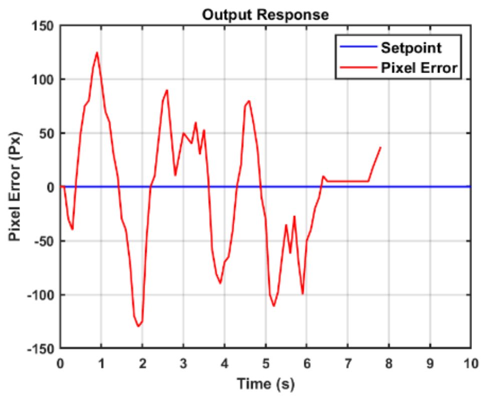

Fig. 16. PID's result (Kp= 0.55, Ki= 0.4, Kd= 0)

4.2.3. Tuning Kd

With Kp=0.55 and Ki=0.25, we adjusted derivative gain (Kd) to optimize system performance.

Table 8: Effect of Kd values on experimental datas

Kd	Time to reach 50 cm (s)	Time to Stop and Buzz (s)	Distance at Stop vs Target (cm)	Final Pixel Error (px)	Plot
0.05	4.5	4.6	+0.5	7	Fig.17.
0.15	4.0	4.0	0	0	Fig.18.
0.30	-	4.1	-5	12	Fig.19.

With Kd = 0.05, clear improvements were noted. The system showed better stability, and the response signal closely tracked the setpoint (Figure 17). However, some residual errors remained.

Increasing Kd to 0.15 resulted in near-perfect control performance. The robot reached the 50 cm mark in just 4 seconds, immediately stopped and emitted the buzzer precisely at the target position. The robot aligned exactly with the detected human with a final pixel error of zero. The graph (Figure 18) showed excellent setpoint tracking and rapid convergence.

When Kd was further increased to 0.3, the system once again became unstable. Although the overshoot was relatively low, the robot could not closely approach the target value (Figure 19). In practice, the robot stopped approximately 5 cm before the 50 cm mark and immediately activated the buzzer.

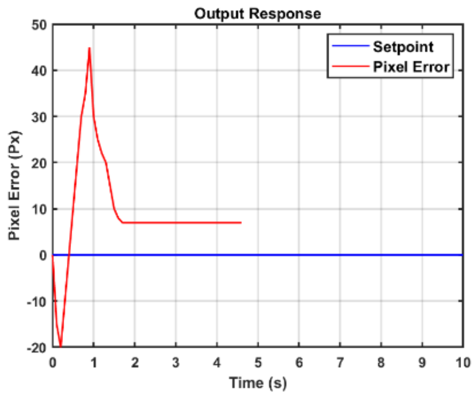

Fig. 17. PID's result (Kp= 0.55, Ki= 0.25, Kd= 0.05)

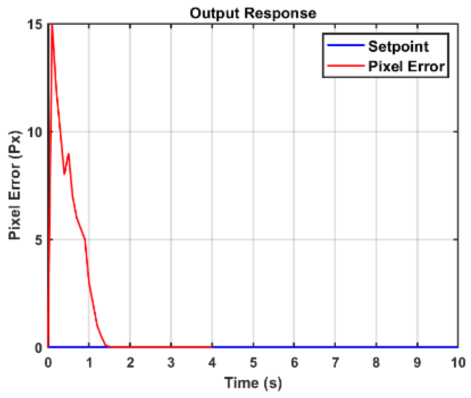

Fig. 18. PID's result (Kp= 0.55, Ki= 0.25, Kd= 0.15)

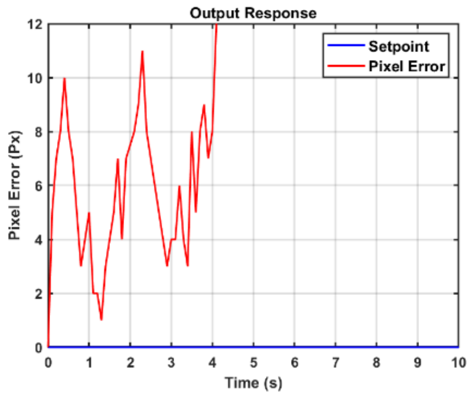

Fig. 19. PID's result (Kp= 0.55, Ki= 0.25, Kd= 0.15)

Thus, the optimal derivative gain was determined to be Kd = 0.15.

4.2.3. Optimal Results

Thus, the optimal PID parameters are determined to be Kp = 0.55, Ki = 0.25, and Kd = 0.15. With this set of parameters, the robot achieved the shortest time to reach the target, a final pixel error of zero, and stopped precisely at the desired position while immediately sounding the alarm.

Next, this optimized PID set was tested under conditions where the robot was initially positioned at angular deviations of 45°, 90°, 135°, and 180° relative to the person. The results and corresponding graphs are presented in Table 9 and Figures 20, 21, 22 and 23. As anticipated, since the robot required additional time to rotate and locate the person, the time taken to reach the target increased significantly with larger initial angular deviations. Other performance metrics also showed slight variations, indicating an increase in error; however, overall system performance remained stable and within acceptable limits. The system continued to demonstrate a degree of stability and maintained good tracking performance toward the setpoint over a reasonable period. These outcomes further confirm the effectiveness and robustness of the optimized PID parameters. (It should be noted that the results were collected from instances where the robot successfully detected the human).

Table 9: Experimental datas for cases when the robot is misaligned with the person

Angle Deviation (Degrees)	Time to reach 50 cm (s)	Time to Stop and Buzz (s)	Distance at Stop vs Target (cm)	Final Pixel Error (px)	Plot
45	4.5	4.6	+1	3	Fig.20.
90	5.5	5.7	+1.5	4	Fig.21.
135	6.2	6.4	+2	8	Fig.22.
190	7.1	7.4	+3	10	Fig.23.

Compared to the paper titled "Stable PID Control for Mobile Robots" by Carmona R. et al. (2018) [19], both systems share the goal of achieving stable and accurate robot movement using PID control. However, while Carmona’s work derived PID parameters analytically through a kinematic model and Lyapunov stability analysis, our approach relied on experimental tuning based on real-time robot feedback. Despite the methodological differences, both studies highlight the effectiveness of PID control in ensuring reliable system behaviour in dynamic conditions.

A notable similarity is the pivotal role PID control plays in generating smooth and stable robot responses in both structured and unstructured environments. Carmona's results demonstrated trajectory tracking errors on the order of a few millimetres, well-suited for high-precision path following in controlled settings. In contrast, our results illustrate the practicality of PID control in vision-based tracking tasks, where visual inputs are subject to constant variation. The optimized PID parameters not only improved the responsiveness of the system but also mitigated oscillation and instability when tracking a moving subject.

These findings reinforce the importance of selecting and optimizing PID parameters tailored to the specific application context. Our results contribute to the broader understanding of PID control in autonomous robotic systems and demonstrate its practical value in real-time person-following and rescue support scenarios using vision-based perception.

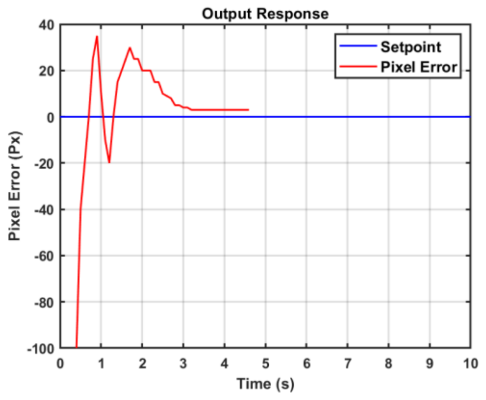

Fig. 20. 45˚ Deviation Results

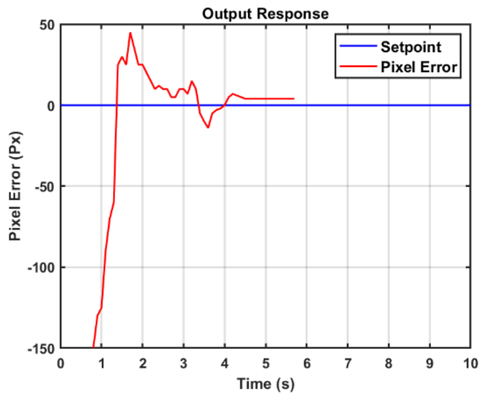

Fig. 21. 90˚ Deviation Results

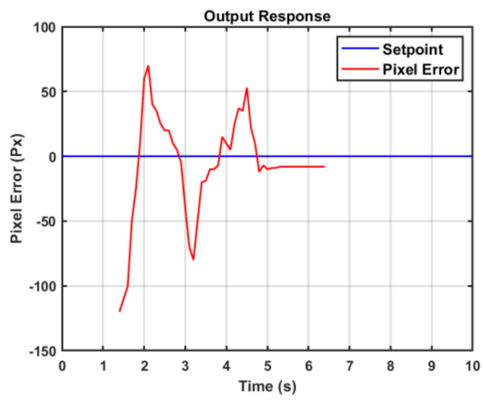

Fig. 22. 135˚ Deviation Results

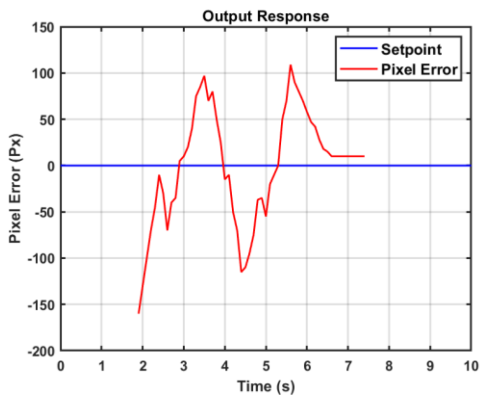

Fig. 23. 180˚ Deviation Results

5. CONCLUSION

After the study, the team successfully designed and implemented a robotic system capable of detecting humans and utilizing a PID controller to approach a fixed position relative to the human before sounding an alert. Through multiple experiments and evaluations, the human detection system based on image processing and the PID control method demonstrated highly promising results, showing precise and rapid control performance that enabled the robot to complete its tasks effectively. This further highlights the strong potential of image processing techniques and PID control in the fields of automation and mobile robotics.

With the success of this research, further development of such systems could significantly benefit rescue operations, where working conditions are extremely hazardous and minimizing human presence is crucial for ensuring safety. Robots of this kind are, therefore, highly necessary. However, as this study was conducted as an educational project by students with limited funding and technical support, the system still has certain limitations. First, experiments were conducted only in stable environments, not in actual rescue sites or environments simulating rescue scenarios; thus, challenges such as obstacles and uneven terrain for the robot's movement remain unaddressed. Another significant limitation of the system is the use of printed human images for detection instead of real human subjects. Although we also conducted tests with actual individuals and observed some successful detections, constraints related to hardware, such as low-resolution sensors, limited camera quality, and restricted mobility and autonomous navigation capabilities, made the use of static human images the most practical and optimal approach for this study. Moreover, only images of upright-standing humans were used for experimentation, while cases involving individuals in other postures have not yet been considered.

Additionally, the trial-and-error approach to tuning PID parameters is not ideal for real-world situations where unexpected variables are common. For instance, when the robot was initially misaligned with the human target, control errors immediately appeared, unlike in the ideal straight-line alignment case. Overall, for the system to operate effectively in real-world rescue missions, improvements in both control algorithms and hardware quality are essential to accomplish the intended rescue tasks reliably.

ACKNOWLEDGMENT

The authors want to thank Assoc. Prof. Van-Thuyen Ngo, Ph.D. for his supervision in completing this research. We also want to give thanks to FEEE for helping us find the place to operate the research and model. The link to the operation of the system is shown the link: https://www.youtube.com/shorts/6GXfFkXkm_0.

REFERENCE

[1] Shal T.A. et al (2013), " Intelligent surveillance robot", in 2013 International Conference on Electrical Communication, Computer, Power, and Control Engineering (ICECCPCE), doi: 10.1109/ICECCPCE.2013.6998745

[2] Mohammed M.N. et al (2018), " Design and Development of Pipeline Inspection Robot for Crack and Corrosion Detection", in 2018 IEEE Conference on Systems, Process and Control (ICSPC), doi: 10.1109/SPC.2018.8704127

[3] Shu Y. et al (2020), " The design of rescue robot based on disaster rescue", in 2020 International Conference on Innovation Design and Digital Technology (ICIDDT), doi: 10.1109/ICIDDT52279.2020.00054

[4] Redmon J. et al (2016), " You Only Look Once: Unified, Real-Time Object Detection", in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), doi: 10.1109/CVPR.2016.91

[5] Shuai Q. et al (2020), " Object detection system based on SSD algorithm", in 2020 International Conference on Culture-oriented Science & Technology (ICCST), doi: 10.1109/ICCST50977.2020.00033

[6] Chauhan R. et al (2018), " Convolutional Neural Network (CNN) for Image Detection and Recognition", in 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC) ,doi: 10.1109/ICSCCC.2018.8703316

[7] Dalal N. et al (2005), " Histograms of oriented gradients for human detection", in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), doi: 10.1109/CVPR.2005.177

[8] Hearst M.A. et al (1998), " Support vector machines", in IEEE Intelligent Systems and their Applications, vol.13, no.4, doi: 10.1109/5254.708428

[9] Shekhar A. et al (2018), " Review of Model Reference Adaptive Control", in 2018 International Conference on Information , Communication, Engineering and Technology (ICICET), doi: 10.1109/ICICET.2018.8533713

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

[11] Bemporad A. (2007), " Model Predictive Control Design: New Trends and Tools", in Proceedings of the 45th IEEE Conference on Decision and Control, doi: 10.1109/CDC.2006.377490

[12] 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

[13] Shi Z. (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

[14] Mohamed T.L.T. 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

[15] Live H.J. 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

[16] Xu L. et al (2016), " Design of the PID controller for industrial processes based on the stability margin", in 2016 Chinese Control and Decision Conference (CCDC), doi: 10.1109/CCDC.2016.7531552

[17] Chen X. et al (2019), " Modeling and Simulation of Vehicle Braking System Based on PID Control", in 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), doi: 10.1109/ICSESS47205.2019.9040743

[18] Shi J. et al (2022), " Research on control of medical permanent magnet synchronous Motor based on fuzzy adaptive fractional order PID", in 2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA), doi: 10.1109/CVIDLICCEA56201.2022.9824616

[19] Carmona R. et al (2018), " Stable PID Control for Mobile Robots", in 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), doi: 10.1109/ICARCV.2018.8581132