Research Article

International Journal of Advanced Technology and Engineering Exploration, Vol 12(122)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2024.111101462

Gain scheduling for point-to-point PID control of car-like robots under load variations

Mohd Faid Yahya[1*] and Mad Helmi Ab. Majid[2]
Faculty of Electrical Technology and Engineering, Universiti Teknikal Malaysia Melaka, 76100, Durian Tunggal, Melaka, Malaysia[1]
Faculty of Computing and Meta-Technology, Universiti Pendidikan Sultan Idris, 35900, Tanjong Malim, Perak, Malaysia[2]

Received: 12-August-2024; Revised: 27-January-2025; Accepted: 28-January-2025

Abstract
The control of car-like robots is crucial for various applications, including autonomous vehicles and industrial automation. Achieving precise and robust movement in these robots, especially under varying load conditions, necessitates advanced control strategies. Traditional proportional-integral-derivative (PID) tuning methods, such as Ziegler-Nichols and Cohen-Coon, often fall short in addressing the dynamic challenges posed by changing loads. Although adaptive and intelligent methods have been explored, they can be computationally intensive and complex to implement. There is a clear need for a more efficient and adaptive PID tuning approach that maintains simplicity while offering robustness against varying loads. This research aims to develop a gain-scheduled PID tuning method specifically designed for car-like robots, enabling them to adapt to varying load conditions during point-to-point movements. The study focuses on the kinematic model of car-like robots, operating in simulated environments with dynamically changing loads. The gain-scheduled PID controller is designed using a combination of analytical and adaptive techniques. The analytical technique aims to meet step response performance criteria, including no overshoot, a rise time of less than 1 s, and a settling time of under 1.5 s. In contrast, the adaptive technique focuses on updating the gains during point-to-point movements to accommodate load variations, ensuring optimal performance throughout the robot's operation. The results are validated across multiple load-carrying scenarios. The performance of the proposed method is benchmarked against basic PID tuning method. In three trials of random load-carrying across 10-point destinations, the basic tuning method resulted in an average completion time of 27.86 s, while the gain-scheduled tuning method achieved an average of 11.27 s. This demonstrates that the gain-scheduled approach offers superior adaptability and robustness, along with reduced computational complexity. The study successfully achieves the objective of developing a robust and efficient PID tuning method for car-like robots.

Keywords
Gain-scheduled, Car-like robot, Proportional integral derivative tuning, Pid tuning, Control system.

1.Introduction
Mobile robots have become increasingly widespread in industrial and logistics applications, with car-like robots being particularly valuable for their maneuverability and load-carrying capabilities [1, 2]. Point-to-point control systems, especially those utilizing proportional-integral-derivative (PID) controllers, remain fundamental in robotics due to their reliability and straightforward implementation [3, 4].

Recent advances in adaptive control strategies have highlighted the importance of considering load variations in mobile robot navigation, [5]. The integration of smart manufacturing principles led to increased demands for robots that can handle variable payloads while maintaining precise positioning [6, 7]. Studies have shown that traditional fixed-parameter PID controllers often struggle with varying load conditions, impacting both trajectory tracking and final positioning accuracy, [8, 9]. Modern industrial applications require robust control systems that can adapt to changing payload conditions while maintaining optimal performance metrics [10, 11].

*Author for correspondence

The primary challenge in implementing PID control for car-like robots arises from the inherent nonlinearity of their system dynamics, especially under varying load conditions [12, 13]. Traditional fixed-gain PID controllers often struggle to maintain consistent performance across different loads, resulting in degraded positioning accuracy and increased settling times [14]. Load variations amplify the coupling effects between longitudinal and lateral motions, further complicating precise point-to-point movements [15]. Moreover, ensuring real-time adjustment of control parameters while maintaining system stability poses significant computational challenges [16].

The primary objective of this paper is to develop an adaptive gain scheduling mechanism for PID controllers in car-like robots that automatically adjusts control parameters based on payload variations. The goal is to improve point-to-point positioning accuracy while maintaining system stability across different loading conditions. The research focuses on developing a computationally efficient solution suitable for real-time implementation in industrial settings.

This paper contributes to the field by presenting a novel gain scheduling framework that integrates advanced tuning techniques for PID controllers in car-like robots. An adaptive algorithm is introduced, capable of real-time adjustments based on load variations, demonstrating significant improvements in performance metrics such as settling time and overshoot. Additionally, comprehensive simulations validate the proposed method against traditional PID control strategies, highlighting its effectiveness in maintaining stability across diverse operational scenarios.

The paper is structured as follows: it begins with an introduction outlining the significance of PID tuning in robotic control systems and literature review. Following this, a detailed methodology section describes the proposed gain scheduling algorithm and its implementation. The results section presents a comparative analysis between the proposed approach and conventional methods, concluding with a discussion of implications and future research directions.

2.Literature review

A basic control system for a car-like robot typically involves point-to-point navigation, where the robot is required to move from one location to another accurately and efficiently. The PID controller is a widely used method for achieving this control due to its simplicity and effectiveness. The primary objective of PID tuning is to adjust the controller parameters (proportional, integral, and derivative gains) to achieve optimal performance in terms of stability, speed, and accuracy. This is particularly critical for mobile robots, where the dynamic behavior can change significantly depending on the load and operating conditions. As mobile robots often need to navigate complex environments and handle varying payloads, it is essential to implement robust and adaptive PID tuning methods to maintain precise control over their motion and positioning [17–19]. The challenge lies in maintaining optimal performance despite these variations, which necessitates advanced tuning strategies [20–22]. The performance of a PID controller is highly dependent on its tuning parameters, which determine how the controller responds to errors. Various methods have been proposed to address these challenges, each with distinct methodologies, results, advantages, and limitations.

Classical tuning methods, such as Ziegler-Nichols and Cohen-Coon, serve as the foundation of PID control design. The Ziegler-Nichols method relies on empirical tuning based on step response data, offering a quick and straightforward way to set PID parameters. While it provides acceptable performance for simple systems, it often leads to suboptimal results in systems with time delays or nonlinearities. For instance, although it effectively stabilizes simple systems, it may cause high overshoot and oscillations in more complex applications [23–25]. On the other hand, the Cohen-Coon method offers an improvement over Ziegler-Nichols by better accommodating time-delay systems, providing a balance between stability and response speed. Despite its advantages, it still faces limitations in nonlinear and time-varying systems and often requires further fine-tuning to meet specific performance criteria in robotic applications [26]. The advantages of classical PID tuning methods are their simplicity, speed, and effectiveness for linear systems. However, they struggle with nonlinearity, time-varying dynamics, and complex environments.

Analytical methods use the system's dynamic characteristics for precise tuning, emphasizing step response optimization to minimize overshoot, settling time, and steady-state error. For instance, optimization-based tuning for differential-drive robots has shown enhanced accuracy and response speed by tailoring the controller parameters to the system's dynamics [27, 28]. As they provide more tailored tuning for specific systems, they offer better performance in complex dynamics. However, they require high computational intensity and reliance on accurate system methods; therefore, they are less adaptable in real-time scenarios.

Adaptive PID tuning methods, such as gain-scheduling, dynamically adjust the controller parameters in real-time based on the current operating conditions. This approach is particularly useful for mobile robots that must adapt to changing loads or varying environmental conditions. Gain-scheduling has been successfully applied in various robotic systems, where it allows the controller to maintain optimal performance despite significant changes in the system's dynamics [29]. This method is crucial for applications where the robot must perform consistently across a wide range of tasks or when dealing with uncertain or varying payloads, ensuring stability and precision in motion control. By continuously updating the PID gains, gain-scheduling helps in maintaining optimal control performance, making it an essential technique in modern robotics [30]. The advantages of adaptive methods lie in their real-time adaptability and robustness to dynamic variations. However, they require accurate modeling and may struggle to perform effectively in highly nonlinear systems.

Intelligent tuning methods, such as fuzzy logic and neural networks, integrate artificial intelligence to handle uncertainties and nonlinearities. Fuzzy logic controllers, leveraging linguistic rules, have been used for path tracking in mobile robots, offering improved adaptability where traditional PID methods fall short [31–33]. This has been effectively used in mobile robot path tracking and manipulator control, where traditional PID controllers may struggle [34–36]. Neural network-based PID controllers, such as self-tuning neural networks, have demonstrated significant adaptability and performance improvements in complex systems like selective compliance articulated robot arm (SCARA) robots [37–39]. Furthermore, deep reinforcement learning has emerged as a robust dynamic gain auto-tuning approach, enhancing real-time control [40–42]. The intelligent tuning methods are highly adaptive and effective in handling nonlinearities and uncertainties. However, they are computationally demanding and require extensive training or rule design.

Evolutionary algorithms, such as genetic algorithms (GA) and particle swarm optimization (PSO), have been increasingly used for PID tuning due to their ability to explore a wide search space and find near-optimal solutions. These methods are particularly useful in scenarios where the control system must handle nonlinearities, time delays, or multiple objectives [43]. For instance, PSO has been used to tune fuzzy PID controllers for mobile robot trajectory control, resulting in enhanced performance and robustness [44, 45]. Similarly, GAs have been applied to optimize PID parameters in various robotic systems, demonstrating significant improvements in control accuracy and system stability [46–48]. These methods offer an alternative to traditional tuning approaches, especially for complex and high-dimensional control problems [49, 50]. Apart from mobile robots, PSO and Bayesian optimization algorithms have been employed to tune the PID controllers of robotic manipulator joints [51, 52]. More advanced evolutionary methods include the use of the bat algorithm [53], whale optimization algorithm [54], ant colony optimization [55], and artificial gorilla troop optimization [56]. They are effective for complex, nonlinear, and multi-objective optimization problems. However, similar to intelligent methods, they are computationally intensive and not always suitable for real-time implementation.

Classical methods, while foundational, lack the robustness required for complex or time-varying systems. Analytical methods provide precision but require accurate system modeling and significant computational resources. Adaptive methods like gain-scheduling offer real-time adaptability but can struggle with highly nonlinear behavior. Intelligent methods show great potential in handling uncertainties but demand extensive computational and design efforts. Evolutionary algorithms provide powerful optimization capabilities but are often limited by their computational demands and inability to adapt in real-time. The gap in the current state of PID tuning lies in developing a hybrid approach that combines the precision of analytical methods, the adaptability of gain-scheduling, and the robustness of intelligent and evolutionary techniques. Such a unified framework could enable real-time, robust PID tuning for mobile robots operating under varying load conditions, addressing the limitations of individual methods and ensuring optimal performance in diverse environments.

The objective of this research is to develop a gain-scheduled PID tuning strategy for a car-like robot that can adapt to changing loads. This approach will integrate analytical and adaptive techniques to create a robust control system. The ultimate goal is to achieve precise and stable point-to-point motion control in mobile robots, ensuring consistent performance across a wide range of tasks and environments. This research focuses on a simulation aspect of a car-like robot operating on 10 m by 10 m workspace. Specifically, the kinematics model of car-like robot is going to be derived and loading dynamics affects the speed and turning of the car-like robot.

3.Methods

The structural design of the car-like robot is illustrated in *Figure 1* showcasing its four-wheel configuration. The robot consists of a chassis with four wheels, where the front two wheels are capable of steering, enabling directional control, and the rear two wheels are responsible for propulsion. The coordinate system of the robot includes the x_b, y_b, and z_b axes representing the body-fixed frame, with x_b directed forward, y_b directed to the left, and z_b directed upward.

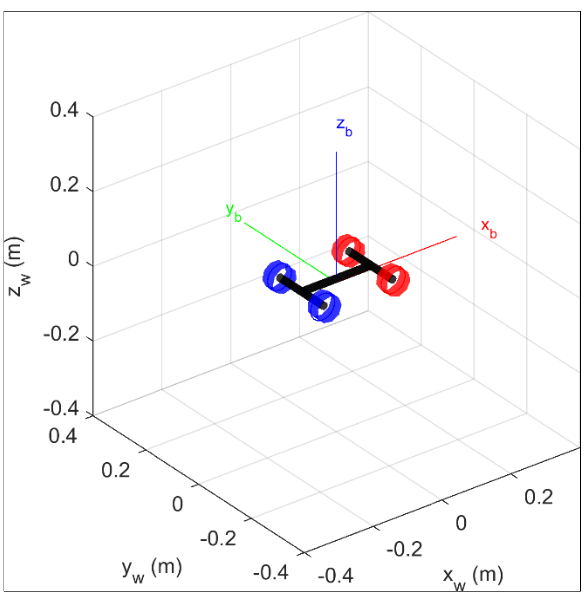

Figure 1 Car-like robot design

The kinematic model of the car-like robot can be described by the following Equations 1 to 3:

$$\dot{x} = v\,cos(\theta) \quad (1)$$

$$\dot{y} = v\,sin(\theta) \quad (2)$$

$$\dot{\theta} = \frac{v}{L}\,tan(\gamma) \quad (3)$$

where:
x and y are the global coordinate
θ is the heading angle
v is the linear velocity
L is the wheelbase
γ is the steering angle

The navigation problem involves the car-like robot moving from a starting point to an ending point through a series of intermediate waypoints as shown in *Figure 2*. The robot must traverse from a starting point to an endpoint through a series of predefined waypoints. This illustration highlights the trajectory planning challenge, which is further complicated by dynamic load variations. The figure provides context for the trajectory tracking equations and control strategies proposed in the subsequent sections.
The problem model is represented as follows:

- The robot starts at coordinates (x_{start}, y_{start})
- It needs to reach the end coordinates (x_{end}, y_{end}), passing through waypoints (x_i, y_i) where $i = 1, 2 \ldots, n$

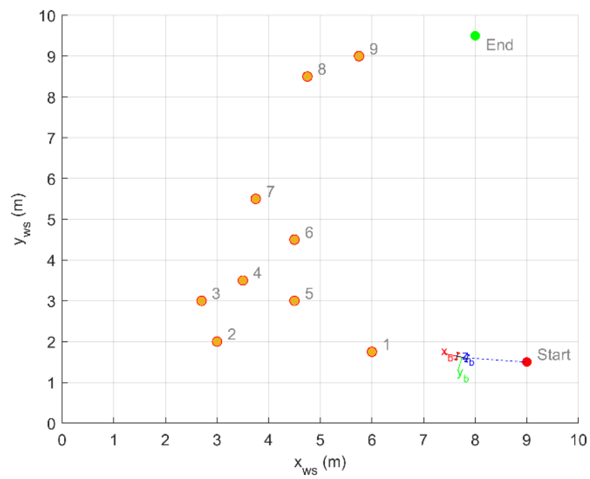

Figure 2 Navigation problem

The problem can be formulated as finding a control strategy that allows the robot to follow a desired trajectory $(x_{ws}(t), y_{ws}(t))$, while accounting for load changes. The trajectory tracking can be represented by Equations 4 and 5:

$$e_x = x_{reference}(t) - x_{actual}(t) \quad (4)$$

$$e_y = y_{reference}(t) - y_{actual}(t) \quad (5)$$

where e_x and e_y are the tracking errors in the x and y directions, respectively.

The proposed control system block diagram for the car-like robot moving point-to-point with load effect is shown in *Figure 3*. The diagram includes the multiple goals (waypoints), angle difference calculations, and the integration of throttle and steering PID controllers. The "Loading and Tuning" block is of particular importance, as it dynamically adjusts the controller gains based on load variations, ensuring smooth transitions and robust performance under different operating conditions. The main components are:

1. Multiple Goals (Start and End Points): The system starts with the defined multiple goals, including start and end points.
2. Throttle and Steering Control: The robot's throttle and steering are controlled based on the PID controller outputs.
3. Angle Difference Calculation: This block calculates the angular difference θ between the robot's current orientation and the desired direction.
4. PID Controllers: Two PID controllers are used - one for controlling the throttle and the other for the steering angle.
5. Car-Like Robot Kinematics: This block uses the kinematic model of the robot to update its position and orientation based on the control inputs.
6. Loading and Tuning: The system incorporates the effect of changing loads and uses a tuning mechanism for gain-scheduled PID control.

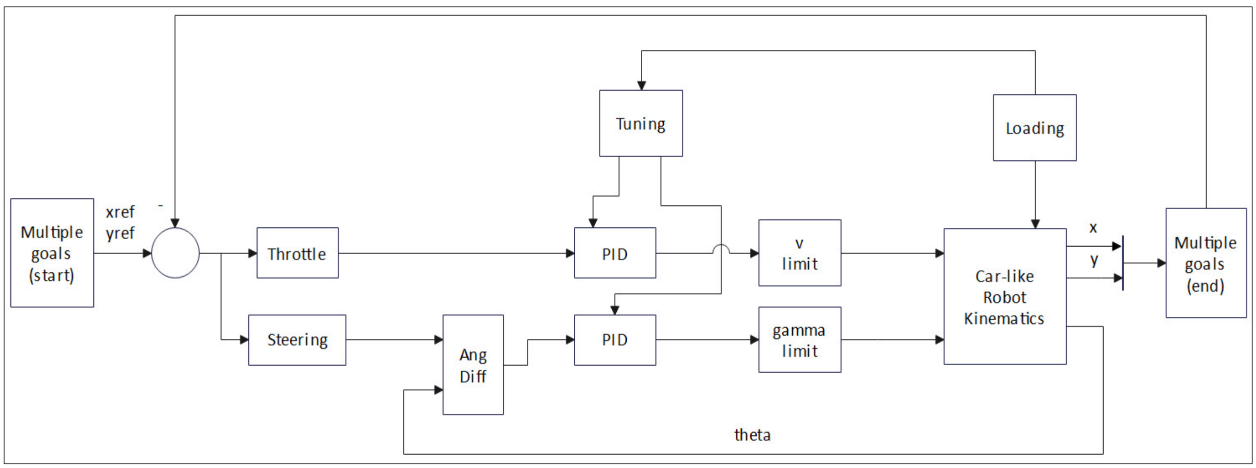

Figure 3 Proposed control system block diagram

The PID model is defined by the following Equation 6:

$$u(t) = K_p e(t) + K_i \int e(t)\, dt + K_d \frac{de(t)}{dt} \qquad (6)$$

where:

- $u(t)$ is the control input (throttle or steering angle),
- $e(t)$ is the error between the desired and actual values,
- K_p, K_i, K_d are the proportional, integral, and derivative gains, respectively.

The presence of varying loads affects the robot's dynamics. The block diagram accounts for this by incorporating a loading block that adjusts the robot's parameters based on the current load. For every 1 kg added to the car-like robot, the speed and turning angle would reduce by 10%. The maximum load was set to 9 kg. The proposed method employs a gain-scheduled PID controller, where the gains K_p, K_i, K_d are adjusted based on the load and the robot's state. This adaptive tuning ensures optimal performance across different operating conditions. The tuning block dynamically updates the PID gains, thereby improving the robot's ability to follow the desired path accurately.

The development of the gain-scheduled PID control system for the car-like robot is based on several key assumptions. First, it is assumed that the sensors provide accurate readings of the robot's position and orientation. Accurate sensor feedback is essential for maintaining the integrity of the control system, as any inaccuracies could introduce errors in the feedback loop, potentially leading to instability or degraded performance. This assumption is particularly valid for simulation environments or scenarios using high-precision sensors like encoders and inertial measurement units (IMUs). However, in real-world applications, sensor inaccuracies, such as noise or bias, would need to be mitigated using techniques like filtering or sensor fusion.

Another assumption is that no slippage occurs between the wheels and the ground. The kinematic model assumes constant traction between the wheels and the ground, ensuring the validity of the motion equations. Slippage, caused by factors like sudden changes in load or surface irregularities, could invalidate the model and lead to inaccuracies in motion control. While this assumption simplifies the study, real-world applications may require enhancements such as the inclusion of slip dynamics or traction control mechanisms to account for variable surface conditions.

The third assumption is that environmental conditions, such as friction between the wheels and the ground, remain constant throughout the operation of the robot. Consistent frictional forces eliminate external uncertainties in the control design and tuning process, making the system more predictable. However, in practical scenarios, environmental factors like wet surfaces or uneven terrain could introduce variations, necessitating adaptive control strategies to maintain performance.

In addition to these assumptions, specific constraints are defined for the system. The robot's maximum allowable load is set at 9 kg. This constraint ensures safe operation and prevents overloading, which could result in mechanical failure or performance degradation. Increasing the load capacity would require a redesign of the system, including stronger components and more powerful actuators, which falls outside the scope of this study.

The performance targets for the PID controller represents another crucial constraint. These include a rise time of less than 1s, no overshoot, and a settling time of less than 1.5s. These targets are standard in precision control applications and ensure the robot's navigation is both efficient and stable. Meeting these criteria minimizes oscillatory behavior and ensures a responsive system capable of adapting to varying load conditions. Failure to meet these performance metrics could compromise the robot's practical usability, highlighting the importance of robust tuning and control design.

The experiments were conducted entirely in a simulation environment using MATLAB, where all programming and controller implementation were developed from scratch. As this study is simulation-based, no physical hardware was used, and the robot's behavior was modeled to reflect its kinematic characteristics without the need for sensor feedback. MATLAB served as the simulation platform, enabling precise implementation of the gain-scheduled PID control strategy. A fixed time step of 0.01 s was utilized to ensure high-resolution simulation and accurate point-to-point tracking.

For each load condition, the simulations were repeated three times to evaluate the robot's performance under random load variations during fixed point-to-point movements. This repetition helped ensure the robustness of the proposed control method. As the study was purely simulation-based, sensors were not incorporated into the model; instead, all required states, such as position, orientation, and velocity, were directly derived from the simulation framework. This setup provided a controlled, reproducible environment for analyzing the effectiveness of the gain-scheduled PID tuning method, isolating it from hardware imperfections and external disturbances.

The gain-scheduled PID tuning involves adjusting the controller gains K_p, K_i, K_d in real-time based on the current load conditions. This adaptive tuning mechanism ensures that the control system can handle varying loads without compromising performance as per Equations 7 to 9.

$$K_p(t) = f_p(load) \tag{7}$$

$$K_i(t) = f_i(load) \tag{8}$$

$$K_d(t) = f_d(load) \tag{9}$$

where f_p, f_i, and f_d are functions that determine the gains based on the load. This research focuses solely on the kinematics of the robot, acknowledging the significant impact of load variations on its movement, while excluding its other dynamic properties. Therefore, the function f serves as a representation of the gain value adjustments determined through trial and error to achieve the desired performance when the robot carries a load.

The algorithm for gain-scheduled PID tuning of car-like robot is given as follow:

Start
Set desired_positions = $[(x1, y1),\ (x2, y2),\ \dots]$
Set load = 0 # Initial load in kg
Initialize K_{p_speed}, K_{i_speed}, K_{d_speed}, $K_{p\_steering}$, $K_{i\_steering}$, $K_{d\_steering}$
for each load from 0 kg to 9 kg:
    Run simulation with current PID values
    while not near desired_position:
        Observe robot motion
        if motion not smooth:

Adjust transition distance
Adjust K_{p_speed}, K_{i_speed}, K_{d_speed}, $K_{p\_steering}$, $K_{i\_steering}$, $K_{d\_steering}$ through trial and error
if step_response not desired:
Adjust K_{p_speed}, K_{i_speed}, K_{d_speed}, $K_{p\_steering}$, $K_{i\_steering}$, $K_{d\_steering}$ to achieve:
Overshoot = 0
Rise time < 1 s
Settling time < 1.5 s
Record K_{p_speed}, K_{i_speed}, K_{d_speed}, $K_{p\_steering}$, $K_{i\_steering}$, $K_{d\_steering}$ for current load
Increase load by 1 kg
Set gain schedule based on recorded PID values
End

The description of the algorithm is shown below:

1. Start: Begin the process.
2. Set desired_positions: Define the desired points the robot needs to reach. These points are predetermined based on the task requirements or a specified path.
3. Set load: Initialize the load to 0 kg.
4. Initialize PID values: Set the initial values for the PID parameters for both speed and steering.
5. For each load from 0 kg to 9 kg: Iterate through the load values, incrementing by 1 kg each time.
   - Run simulation with current PID values: Execute the simulation with the current set of PID values.
   - While not near desired_position: Continue adjusting as long as the robot has not reached the desired position.
     - Observe robot motion: Monitor the robot's movement.
     - If motion not smooth: If the movement is not smooth, adjust the transition distance.
     - Adjust PID values through trial and error: Manually fine-tune the PID parameters to achieve the desired motion characteristics.
     - If step_response not desired: Ensure the step response meets the specified criteria (no overshoot, rise time < 1s, settling time < 1.5s).
       - Adjust PID values to achieve desired step response: Further adjust the PID parameters to meet these criteria.
     - Record PID values: Document the PID values for the current load.
   - Increase load by 1 kg: Move to the next load increment.
6. Set gain schedule: Establish the gain schedule based on the recorded PID values for different loads.
7. End: Conclude the process.

Figure 4 and *Figure 5* show the reference and actual x, y-position signals of the robot over simulation time. The blue line represents the reference positions the robot should reach, while the orange line represents the actual positions achieved by the robot. The goal is to minimize the difference between the reference and actual x, y positions. The minimal deviation between these lines demonstrates the controller's precision in following the desired path. Together, *Figures 4* and *Figure 5* provide a comprehensive view of the robot's accuracy in 2D space, highlighting the gains achieved through the proposed PID tuning method. *Figure 6* depicts the error signals for both x and y positions over simulation time. The blue line represents the error in the x position, and the orange line represents the error in the y position. The decreasing error magnitudes indicate the effectiveness of the gain-scheduled PID controller in reducing positional deviations. This figure provides insight into the controller's stability and its ability to achieve convergence to the desired trajectory under dynamic load variations. As per the system design, the robot is programmed to prioritize reaching within 0.5 meters of the target point before proceeding to the next.

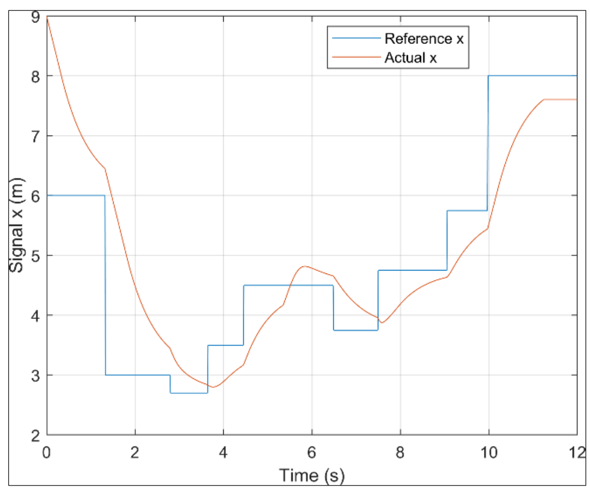

Figure 4 Signal for x coordinate

The tuning process for the PID controller was carried out systematically to ensure robust performance under varying load conditions. The K_p, K_i, and K_d gains were initialized based on trial-and-error simulations, leveraging prior knowledge of the system's dynamics. The primary objective during initialization was to minimize the rise time, overshoot, and settling time while ensuring stability.

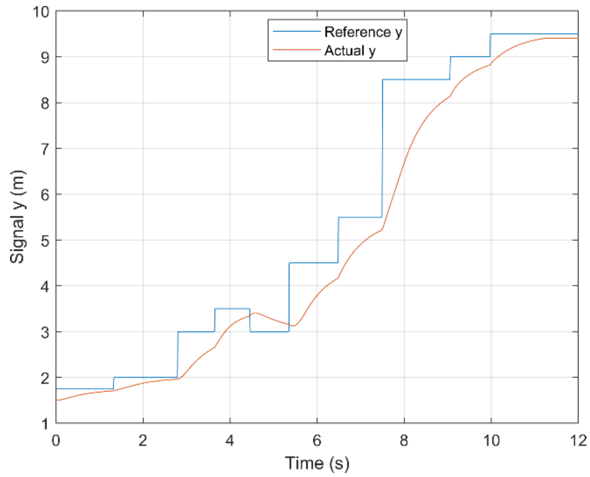

Figure 5 Signal for y coordinate

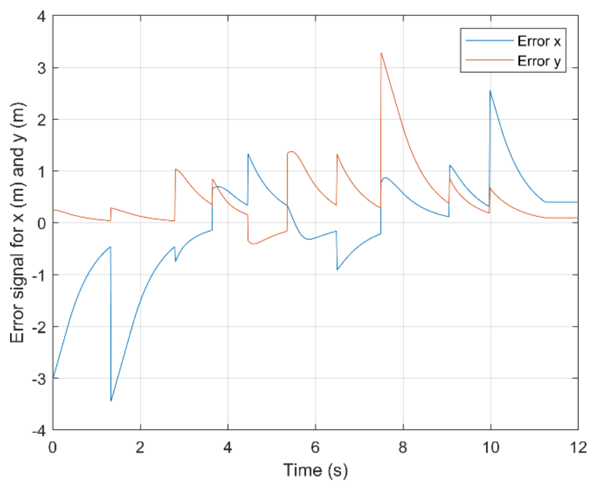

Figure 6 Error signal

For the specific case of point-to-point navigation, the integral and derivative terms K_i, and K_d were omitted during the tuning process. For point-to-point movements, steady-state errors are less critical since the task primarily focuses on transient response. Including K_i could introduce unnecessary overshoot or oscillations, particularly for tasks that require quick transitions. The derivative term, while useful in improving damping and reducing overshoot, was found to contribute marginally to performance improvement in point-to-point navigation tasks. The simplicity of excluding K_d also reduced computational complexity, which is advantageous for real-time applications.

However, for other navigation tasks, such as trajectory tracking or line following, the inclusion of K_i and K_d becomes essential to address cumulative errors and enhance stability. These tasks demand higher precision and robustness against disturbances, where the contributions of integral and derivative actions are critical. This aspect is acknowledged as a potential area for future research, where the gain-scheduled approach can be extended to dynamically tune K_i and K_d based on the task requirements.
The proportional gains K_p were adjusted adaptively using the gain-scheduling mechanism. This allowed the controller to respond effectively to load variations by increasing or decreasing K_p values to maintain the desired transient and steady-state performance. The gain scheduling was performed by correlating load changes with the required control effort, as documented in *Table 1*. This approach ensured that the controller remained efficient and robust for the intended point-to-point navigation tasks while providing a foundation for extending the method to more complex scenarios.

Table 1 lists the gain schedule for the PID parameters at different load levels. For each load increment, the corresponding PID values for both steering and throttle are provided. The table serves as a reference for implementing gain-scheduled PID control based on the load. It facilitates the adjustment of PID parameters to maintain optimal performance as the load changes. Notice that the increment of the gains, specifically for Kp1 and Kp2 are nonlinear whereas other gains remain 0. The integral and derivative properties do not need to be tuned because they depend on the type of navigation problem. For the point-to-point problem, adjustments are made only to two gains (Kp1 and Kp2). However, for other navigational tasks such as line following, path following, or moving to a specific pose, tuning the integral and derivative gains becomes essential, which is beyond the scope of this paper.

Table 1 Gain schedule for handling load variation

Loading (kg)	**Gain schedule**					
	Steering			**Throttle**		
	Kp1	**Ki1**	**Kd1**	**Kp2**	**Ki2**	**Kd2**
0	1.5	0	0	4	0	0
1	1.67	0	0	4.1	0	0
2	1.88	0	0	4.5	0	0
3	2.15	0	0	5.7	0	0
4	2.5	0	0	6.5	0	0
5	3	0	0	7.5	0	0
6	3.7	0	0	8.5	0	0
7	5	0	0	13	0	0
8	7.5	0	0	21	0	0
9	15	0	0	40	0	0

The gain adjustments are nonlinear due to the inherent complexities of the dynamic system, particularly in how the car-like robot responds to varying loads. Load variations introduce changes in inertia and friction, which affect the system's kinematics and control effort in a nonlinear manner. Linear gain adjustments were insufficient during initial tests to meet performance criteria such as eliminating overshoot, achieving a rise time of less than 1 s, and maintaining a settling time under 1.5 s. Nonlinear adjustments, on the other hand, allowed the gains to scale appropriately, providing precise control for both small and large load variations. This empirical approach, as reflected in *Table 1*, demonstrates how f_p, f_i, and f_d increase disproportionately with load, ensuring that the control system responds adequately across the entire operating range.

The nonlinear gain adjustments have a significant impact on stability and performance. By tailoring the gains to specific load conditions, nonlinear gain adjustments enhance stability margins, preventing oscillations and instability caused by under-tuned or over-tuned gains. This approach also improves robustness against parameter uncertainties and external disturbances, ensuring that the robot maintains steady trajectory tracking regardless of abrupt load changes. From a performance perspective, the nonlinear scheduling ensures optimal control across a wide range of loads, balancing responsiveness and smoothness. The results demonstrated consistent performance, including reduced completion times and stable trajectory tracking, even under dynamically changing loads. Additionally, nonlinear gains improve energy efficiency by minimizing control effort for lighter loads while maintaining adequate force for heavier loads. Nonlinear gain adjustments are essential for ensuring stability and achieving optimal performance in systems with dynamic and complex behaviors.

4.Results

The initial experiment was conducted to test the robot's performance without any load using poorly tuned PID parameters. As illustrated in *Figure 7*, the robot's trajectory deviates significantly from the desired path. The path is marked by multiple oscillations and an overall inefficient route, indicating the inadequacy of the tuning. The robot struggles to follow the set trajectory, resulting in an elongated and erratic movement. This emphasizes the need for proper tuning methods to achieve stable and accurate navigation.

In contrast, *Figure 8* demonstrates the robot's performance with well-tuned PID parameters. The trajectory closely follows the desired path with minimal deviation. The improved tuning results in a smoother and more efficient route, showcasing the effectiveness of the gain-scheduled PID tuning. The robot reaches each waypoint accurately and follows a consistent path, indicating enhanced stability and control.

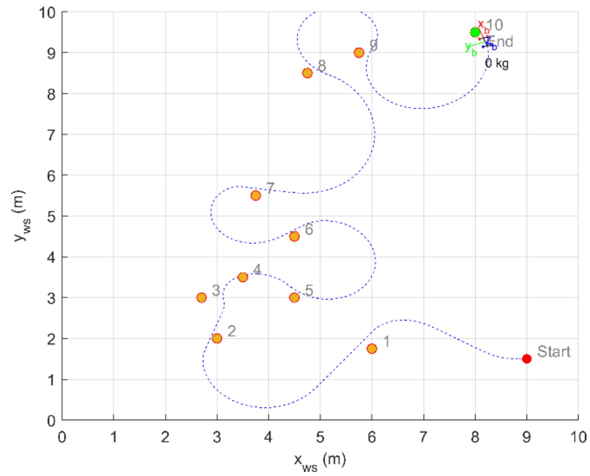

Figure 7 Bad tuning (no load condition with $K_{p\_speed} = 1.5$, $K_{i\_speed} = 3$, $K_{d\_speed} = 2$, $K_{p\_steeering} = 3$, $K_{i\_steering} = 3$, $K_{d\_steering} = 1$)

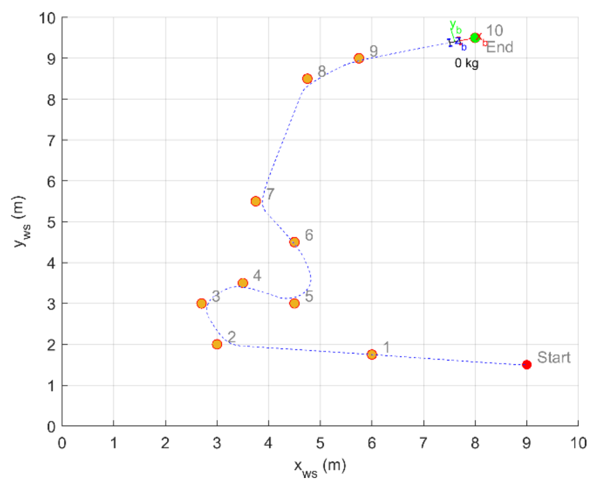

Figure 8 Good tuning (no load condition with $K_{p\_speed} = 1.5$, $K_{i\_speed} = 0$, $K_{d\_speed} = 0$, $K_{p\_steeering} = 4$, $K_{i\_steering} = 0$, $K_{d\_steering} = 0$)

In evaluating the performance of the gain-scheduled PID controller, key response characteristics were measured, including rise time, settling time, and overshoot. These parameters were analyzed for varying load conditions, with the controller consistently maintaining a rise time of less than 1 s, settling time of less than 1.5 s, and negligible overshoot, as detailed in the description of the algorithm.

Table 2 presents the overall completion time for the robot to travel from the start to the last position under fixed loading conditions. The analysis of the provided data in *Table 2* highlights significant differences between the performance of the no tuning and gain-scheduled approaches in terms of completion time, consistency, and predictability. The average completion time for the no tuning approach is 24.93 s, which is considerably higher than the 11.26 s achieved with gain-scheduling. This difference indicates a substantial improvement in performance when gain-scheduling is applied, showcasing its effectiveness in reducing completion time.

In terms of consistency, gain-scheduling demonstrates better stability with a very low standard deviation of 0.05 s, compared to the 17.38 s observed for the no tuning approach. This low variation indicates that gain-scheduling consistently delivers nearly the same performance regardless of the load applied, while the no tuning method exhibits significant variability.

The predictability of the system is further evident from the confidence intervals. For gain-scheduling, the 95% confidence interval is extremely narrow, ranging from 11.22 to 11.30 s. This highlights a high degree of reliability and precision in its performance. On the other hand, the no tuning approach has a much wider confidence interval, spanning from 11.57 s to 38.28 s. This wide range reflects the variability and unpredictability associated with this method, particularly as the load increases.

Table 2 Fixed loading from start position to last position test

Loading (kg)	Overall completion time (s)	
	No tuning	**Gain-schedule**
0	11.24	11.24
1	12.36	11.24
2	13.81	11.24
3	15.72	11.22
4	18.35	11.25
5	22.11	11.26
6	27.79	11.41
7	37.46	11.25
8	65.49	11.24
9	n/a	11.24

Based on the data from *Table 2*, the plot of completion time comparing no tuning and gain-scheduled tuning is presented in *Figure 9. It* reveals that gain-scheduling effectively maintains a consistent completion time across all loading conditions, even under significant load variations. In contrast, the no tuning approach shows a clear trend of increasing completion times with heavier loads, emphasizing its limitations in handling varying conditions. Overall, this analysis highlights the advantage of gain-scheduled tuning in achieving efficient, consistent, and reliable performance.

Table 3 provides data on the robot's performance under random loading conditions, detailing the coordinates and load for each position. Based on *Table 3*, the average completion time for the no-tuning method is 27.86 s, whereas the gain-scheduled method significantly reduces it to 11.27 s. This demonstrates that gain-scheduled tuning provides a notable improvement in efficiency, making it much faster in handling tasks involving random loads.

When analyzing the standard deviation, the no tuning method shows a value of 7.382 s, indicating considerable variability in its performance. In contrast, the gain-scheduled method achieves a standard deviation of just 0.006 s, which is exceptionally low. This suggests that gain-scheduled tuning not only improves speed but also ensures consistency in performance across different trials.

The 95% confidence intervals further support these observations. For the no tuning method, the confidence interval is wide, with a margin of ±5.674 s, reflecting high uncertainty and variability in its completion times. On the other hand, the gain-scheduled method has a confidence interval of just ±0.004 s, reflecting extremely reliable and stable outcomes.

Finally, the lower and upper bounds for the no tuning method range from 22.186 s to 33.534 s, whereas for the gain-scheduled method, the bounds are extremely tight, ranging from 11.262 s to 11.271 s. This indicates that the gain-scheduled approach provides predictable results within a very narrow range, as opposed to the wide variability seen in the no-tuning method. The gain-scheduled PID tuning method significantly outperforms the no-tuning approach in terms of efficiency, reliability, and consistency. It is evident that gain-scheduled tuning is highly effective for managing random loads, providing faster and more predictable control in such scenarios.

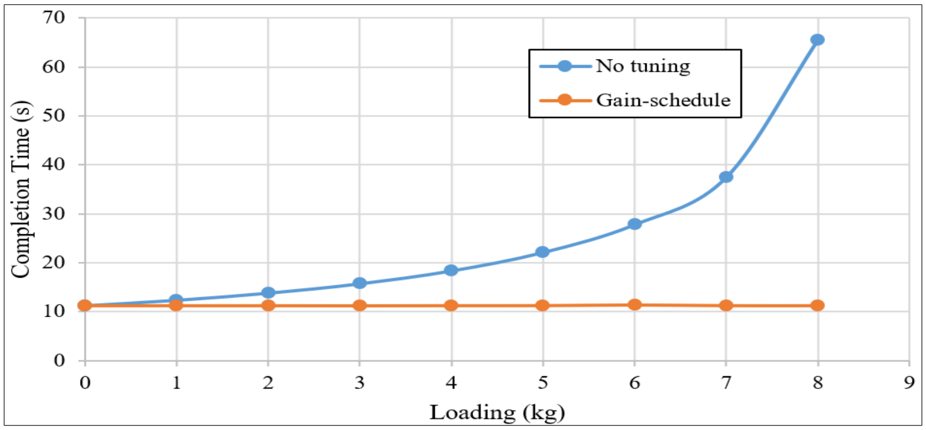

Figure 9 Plot of completion time between no tuning and gain-schedule

Table 3 Random loading from start position to last position test

Position	Coordinate (m)		Random load trial (kg)		
	x	**y**	**First**	**Second**	**Third**
1	6	1.75	3	2	5
2	3	2	7	2	5
3	2.7	3	1	5	1
4	3.5	3.5	8	2	7
5	4.5	3	6	8	1
6	4.5	4.5	8	5	7
7	3.75	5.5	6	2	3
8	4.75	8.5	4	7	1
9	5.75	9	1	4	3
10	8	9.5	2	8	3
Overall completion time (s)		**No tuning**	35.33	27.68	20.57
		Gain-schedule	11.27	11.27	11.26

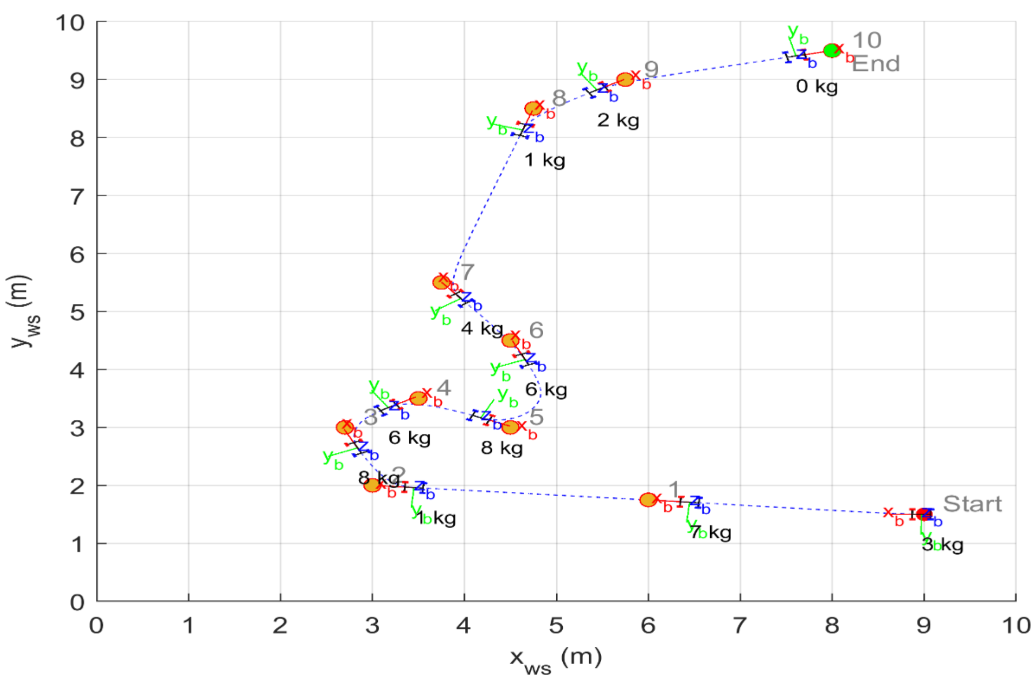

Figure 10 Random load effect on the car-like robot moving point-to-point for first random load trial

The first random load trial is shown in *Figure 10*. Despite the dynamic changes in load, the robot maintains a stable and consistent path, reaching the desired waypoints efficiently. This figure exemplifies the adaptability of the gain-scheduled PID controller in handling unpredictable load variations, a critical feature for real-world applications.

5.Discussion

The experimental results demonstrate the critical importance of proper PID tuning in robotic navigation, especially under varying load conditions. The system's response characteristics, particularly rise time, settling time, and overshoot, are indicative of its stability and efficiency. The gain-scheduled PID tuning effectively minimized overshoot while maintaining rapid response times under all tested loading conditions. These results demonstrate the controller's robustness and suitability for point-to-point navigation tasks in environments with varying loads. The poorly tuned system failed to follow the desired path accurately, resulting in inefficient and unstable movements. In contrast, the gain-scheduled PID tuning showcased significant improvements in trajectory accuracy and stability, both for no-load and load conditions.

In the first experiment, the gain-scheduled tuning method proved its robustness by maintaining consistent performance across different fixed loads, effectively mitigating the impact of increasing load on the robot's navigation efficiency. This indicates its suitability for applications where the load is known and fixed. In the second experiment, the gain-scheduled method excelled under random loading conditions, providing consistent and reliable performance across all trials. This adaptability highlights its potential for real-world applications where load conditions may vary dynamically.

The experimental results underscore the effectiveness of gain-scheduled PID tuning in addressing the challenges of dynamic load variations in car-like robots. The significant improvements in trajectory accuracy, stability, and completion time, as compared to traditional PID tuning methods, highlight the potential of this approach for real-world applications.

Our findings complement and extend prior studies on PID tuning in robotics. For instance, Serrano-Pérez et al. [17] demonstrated the utility of offline robust tuning for omnidirectional robots but acknowledged the limitations in handling real-time dynamic changes. Our gain-scheduled approach fills this gap by providing a real-time adaptive solution that maintains performance across a range of operating conditions.

Lee et al. [14] explored adaptive PID tuning for robotic manipulators under varying payloads. Their approach focused on linear dynamics, while our study addresses nonlinearities inherent in car-like robots. Additionally, Campos et al. [44] utilized PSO for mobile robot trajectory control, achieving high precision, though at the cost of computational efficiency. Our method offers a simpler yet effective alternative by dynamically adjusting PID gains based on load variations, ensuring real-time adaptability without the need for complex optimization processes.

The proposed gain-scheduled PID tuning method has the capability for far-reaching implications in industries such as logistics, manufacturing, and autonomous transportation. Robots equipped with this control system can adapt seamlessly to varying payloads, minimizing downtime and maximizing operational efficiency. For example, in warehouse automation, robots must navigate efficiently while carrying loads of different weights. The ability to dynamically adjust control parameters ensures consistent performance, enhancing overall productivity.

Moreover, this study provides a framework for improving the adaptability of autonomous vehicles in scenarios involving dynamic weight distribution, such as passenger cars or delivery drones. The emphasis on maintaining stability and reducing computational complexity makes this approach particularly valuable for embedded systems with limited processing power.

While this research focuses on gain-scheduled PID tuning, it opens avenues for integrating advanced control strategies. Intelligent methods such as fuzzy logic or neural networks, as discussed by Mourad and Youcef [31], could complement gain-scheduling by introducing additional adaptability to handle highly nonlinear dynamics. For instance, neural networks could predict the required gain adjustments based on sensor data, further enhancing the robustness of the control system.

Additionally, our approach stands to gain from hybrid methodologies combining gain-scheduling with evolutionary optimization techniques. Such integrations, as highlighted by Campos et al. [44], may allow for finer adjustments in environments with extreme variability.

In addition to performance metrics, computational overhead is a key consideration for the real-time feasibility of gain-scheduled PID controllers. Gain scheduling introduces computational requirements due to the real-time calculation and adjustment of PID gains based on load variations. These computations may impact the controller's response time, especially in embedded systems with limited processing power.

A preliminary analysis of computational demands indicates that the proposed gain-scheduled PID controller involves repetitive matrix operations and look-up table interpolations. These operations are lightweight and manageable on modern embedded platforms, such as those using ARM Cortex-M processors. However, scalability challenges may arise when adapting the algorithm for more complex environments or higher-dimensional problems. Future work should include detailed profiling of computational latency and memory usage to ensure compatibility with resource-constrained hardware.

This consideration is critical for embedded implementation in real-world applications where system responsiveness and reliability are paramount. The ability to maintain computational efficiency while achieving robust control remains an area for further optimization. The gain-scheduled PID tuning method significantly enhances the performance and stability of the car-like robot, making it a viable solution for navigating point-to-point with changing loads. Future work could explore further optimization techniques and real-world implementations to validate and extend these findings.

5.1Limitation

Gain-scheduled PID control for car-like robots, while effective for handling varying loads, presents several limitations. It heavily depends on accurate modeling of the robot and is time-consuming to implement and tune for different operating conditions. Ensuring smooth transitions between gain sets is challenging, with risks of instability or oscillations. The approach may struggle with scalability as the number of operating conditions increases. Furthermore, gain-scheduled PID tuning might face difficulties in handling sudden changes in environmental disturbances, as these scenarios can introduce abrupt variations in system dynamics, leading to performance degradation or instability. This limitation highlights the need for robust mechanisms to adapt to unanticipated changes and nonlinearities, which could lead to overfitting and reduce generalizability and overall performance. A complete list of abbreviations is listed in *Appendix I.*

6.Conclusion and future work

This paper presents a comprehensive study on gain-scheduled PID tuning for a car-like robot tasked with point-to-point movement under varying load conditions. The primary objective was to develop a robust PID tuning method that adapts to changing loads, thereby ensuring optimal performance across different operating scenarios. The research successfully achieved its primary objective by integrating gain-scheduling into the PID control framework. This approach was validated through extensive simulation tests, demonstrating significant improvements in the robot's tracking accuracy and stability compared to traditional PID tuning methods.

One of the key achievements is adaptive performance. The gain-scheduled PID controller effectively adjusted its parameters in real-time, maintaining high performance despite variations in load. This adaptability was crucial in minimizing the overshoot and settling time, ensuring smooth and precise point-to-point navigation. The second key achievement is robustness. By incorporating load variations into the control strategy, the proposed method exhibited robust performance across a wide range of operating conditions. This robustness is essential for real-world applications where external disturbances and load changes are inevitable. The last key achievement is in term of comparative analysis. The proposed methodology was compared to the basic tuning method for benchmarking. The results highlighted the superiority of the gain-scheduled PID approach in overall completion time, validating its effectiveness and applicability.

The gain-scheduled PID tuning method proposed in this paper represents a significant advancement in the control of car-like robots. By addressing the challenges posed by varying loads, this method ensures reliable and efficient point-to-point navigation. Future work will focus on extending this approach to more complex robotic platforms and exploring its integration with other advanced control techniques.

Future research should aim to extend the applicability of this approach by testing it in real-world environments to evaluate its robustness and reliability under diverse conditions. Incorporating advanced techniques, like machine learning models, could improve the precision of gain adjustments by predicting system dynamics and adapting to unforeseen changes. Additionally, the scalability of the method should be examined for larger robotic systems or multi-robot environments, where interdependencies and cooperative dynamics add complexity. Hybrid systems combining gain-scheduling with intelligent or evolutionary methods, such as neural networks or optimization algorithms, show promise for improving adaptability and robustness in dynamic scenarios. These future directions could significantly enhance the versatility and impact of the proposed gain-scheduled PID tuning strategy.

Acknowledgment
The authors express their gratitude to the Fakulti Teknologi dan Kejuruteraan Elektrik (FTKE) and Universiti Teknikal Malaysia Melaka (UTeM) for their support in this research and publication.

Conflicts of interest
The authors have no conflicts of interest to declare.

Data availability
None.

Author's contribution statement
Mohd Faid Yahya: Data compiling, analysis, investigation, conception, preparing first version, reviewing and revising written content. analysis and interpretation of findings. **Mad Helmi Ab. Majid**: Inspection of data, reviewing and revising written content, and examination of issues.

References
[1] Wang Z, Li P, Li Q, Wang Z, Li Z. Motion planning method for car-like autonomous mobile robots in dynamic obstacle environments. IEEE Access. 2023; 11:137387-400.

[2] Velasco-sánchez EP, Bañón MÁ, Candelas FA, Puente ST, Torres F. ViKi-HyCo: a hybrid-control approach for complex car-like maneuvers. IEEE Access. 2024; 12:65428-43.

[3] Carron A, Bodmer S, Vogel L, Zurbrügg R, Helm D, Rickenbach R, et al. Chronos and CRS: Design of a miniature car-like robot and a software framework for single and multi-agent robotics and control. In international conference on robotics and automation 2023 (pp. 1371-8). IEEE.

[4] Pastrana MA, Oliveira LH, Mendes DA, Silva DL, Mendoza-peñaloza J, Muñoz DM. Implementation of a PID controller using online tuning applied to a mobile robot obstacle following/avoidance. In argentine conference on electronics 2024 (pp. 10-5). IEEE.

[5] Leong PY, Ahmad NS. Exploring autonomous load-carrying mobile robots in indoor settings: a comprehensive review. IEEE Access. 2024; 12:131395-417.

[6] Jochman T, Voltr V, Kubáček V, Švec O, Burget P, Hlaváč V. Integrating augmented reality within digital twins for smart robotic manufacturing systems. In 29th international conference on emerging technologies and factory automation 2024 (pp. 1-8). IEEE.

[7] Kang BG, Park HS, Lee JM, Yun JM. Methodology on the cyber-physical system construction for a user-friendly smart clothing manufacturing robot system. In 33rd IEEE international conference on robot and human interactive communication 2024 (pp. 412-6). IEEE.

[8] Nagaya K, Ohhira T, Hashimoto H. Developing power-assisted two-wheeled luggage-carrying robot for stair-lifting using admittance control. In international conference on mechatronics 2023 (pp. 1-6). IEEE.

[9] Gu S, Meng F, Liu B, Chen X, Yu Z, Huang Q. Adaptive interactive control of human and quadruped robot load motion. IEEE/ASME Transactions on Mechatronics. 2024:1-12.

[10] Andronas D, Kampourakis E, Papadopoulos G, Bakopoulou K, Kotsaris PS, Michalos G, et al. Towards seamless collaboration of humans and high-payload robots: an automotive case study. Robotics and Computer-Integrated Manufacturing. 2023; 83:1-17.

[11] Ma J, Lian B, Wang M, Dong G, Li Q, Wu J, et al. Optimal design of a parallel assembling robot with large payload-to-mass ratio. Robotics and Computer-Integrated Manufacturing. 2023; 80:102474.

[12] Xiao L. Precise movement control of agricultural robot car based on PID control algorithm. In 2nd international conference on image processing and computer applications 2024 (pp. 1-6). IEEE.

[13] Georgieva T, Ivanov A, Anchev A. Payload stabilization in multidirectional robot motion by applying trajectory analysis algorithm and feedback encoders. In international conference automatics and informatics 2023 (pp. 179-85). IEEE.

[14] Lee J, Chang PH, Yu B, Jin M. An adaptive PID control for robot manipulators under substantial payload variations. IEEE Access. 2020; 8:162261-70.

[15] Wang Y, Li Z, Wang Y, Yang Y. Design of a position tracking controller for quadrotor uavs with variable mass. In 7th international conference on intelligent robotics and control engineering 2024 (pp. 52-8). IEEE.

[16] Borkar KK, Aljrees T, Pandey SK, Kumar A, Singh MK, Sinha A, et al. Stability analysis and navigational techniques of wheeled mobile robot: a review. Processes. 2023; 11(12):1-36.

[17] Serrano-pérez O, Villarreal-cervantes MG, Rodríguez-molina A, Serrano-pérez J. Offline robust tuning of the motion control for omnidirectional mobile robots. Applied Soft Computing. 2021; 110:107648.

[18] Somefun OA, Akingbade K, Dahunsi F. The dilemma of PID tuning. Annual Reviews in Control. 2021; 52:65-74.
[19] Borase RP, Maghade DK, Sondkar SY, Pawar SN. A review of PID control, tuning methods and applications. International Journal of Dynamics and Control. 2021; 9:818-27.
[20] Abdulnabi AR, Esmaile MF. Automatic mapping and localization in large-scale cyclic using K-nearest neighbours. International Journal of Advanced Technology and Engineering Exploration. 2022; 9(97):1802-11.
[21] Ibrahim MF, Huddin AB, Zaman MH, Hussain A, Anual SN. An enhanced frontier strategy with global search target-assignment approach for autonomous robotic area exploration. International Journal of Advanced Technology and Engineering Exploration. 2021; 8(75):283-91.
[22] Joseph SB, Dada EG, Abidemi A, Oyewola DO, Khammas BM. Metaheuristic algorithms for PID controller parameters tuning: review, approaches and open problems. Heliyon. 2022; 8(5):1-29.
[23] Africa AD, Chua JO, Solis JL. PID tuning of speed controller using ziegler-nichols and manual method DC motor. In 15th international conference on humanoid, nanotechnology, information technology, communication and control, environment, and management 2023 (pp. 1-6). IEEE.
[24] Hegedus ET, Birs IR, Ghita M, Ionescu CM, De KR, Muresan CI, et al. Optimal fractional order PID based on a modified Ziegler-Nichols method. In international conference on electrical, computer, communications and mechatronics engineering 2022 (pp. 1-6). IEEE.
[25] Mazlan NN, Thamrin NM, Razak NA. Comparison between Ziegler-Nichols and AMIGO tuning techniques in automated steering control system for autonomous vehicle. In international conference on automatic control and intelligent systems 2020 (pp. 7-12). IEEE.
[26] Utami AR, Yuniar RJ, Giyantara A, Saputra AD. Cohen-coon PID tuning method for self-balancing robot. In international symposium on electronics and smart devices 2022 (pp. 1-5). IEEE.
[27] Manoharan SK, Raghavan D, Megalingam RK, Parakat P, Sudheesh SK. Fine tuning mobility: PID driven velocity control optimization for two wheel differential drive robot. In Parul international conference on engineering and technology 2024 (pp. 1-6). IEEE.
[28] Dib F, Benaya N, Ben MK, Boumhidi I. Comparative study of optimal tuning PID controller for manipulator robot. In the proceedings of the international conference on smart city applications 2022 (pp. 252-61). Cham: Springer International Publishing.
[29] Kenmochi K, Dai Y, Hirakoso N. An optimal fusion of the position control part and the force control part in the bilateral control system. In 12th global conference on consumer electronics 2023 (pp. 545-6). IEEE.
[30] Ramos-velasco LE, Parra-vega V, García-rodríguez R, Vega-navarrete MA, Trejo-ramos C, Olguín-díaz E. Knowledge-based self-tuning of PID control gains for continuum soft robots. Engineering Applications of Artificial Intelligence. 2024; 133:108447.
[31] Mourad A, Youcef Z. Fuzzy-PI controller tuned with ICA: applied to 2 DOF robot control trajectory. In international conference on information, communication and automation technologies 2022 (pp. 1-6). IEEE.
[32] Wahyuni S, Fuad M, Hilda AD, Umam F, Latif M. Enhancing mobile robot stability and roaming capability using fuzzy-PID control method. In 9th information technology international seminar 2023 (pp. 1-6). IEEE.
[33] Kristiyono R, Wiyono W. Autotuning fuzzy PID controller for speed control of BLDC motor. Journal of Robotics and Control. 2021; 2(5):400-7.
[34] Razali MR, Faudzi AA, Shamsudin AU. Position and angular control using fuzzy tuned PID controller for mobile robot path tracking. In 5th international symposium in robotics and manufacturing automation 2022 (pp. 1-6). IEEE.
[35] Yilmaz BM, Tatlicioglu E, Savran A, Alci M. Adaptive fuzzy logic with self-tuned membership functions based repetitive learning control of robotic manipulators. Applied Soft Computing. 2021; 104:107183.
[36] Ben JC, Seddik H. Design of a PID optimized neural networks and PD fuzzy logic controllers for a two-wheeled mobile robot. Asian Journal of Control. 2021; 23(1):23-41.
[37] Freire EO, Rossomando FG, Soria CM. Self-tuning of a neuro-adaptive PID controller for a SCARA robot based on neural network. IEEE Latin America Transactions. 2018; 16(5):1364-74.
[38] Rodríguez-abreo O, Rodríguez-reséndiz J, Fuentes-silva C, Hernández-alvarado R, Falcón MD. Self-tuning neural network PID with dynamic response control. IEEE Access. 2021; 9:65206-15.
[39] Yildirim S, Savas S, Andruskiene J. Controller gain tuning of a nonholonomic mobile robot via neural network predictor. In 25th international conference electronics 2021 (pp. 1-6). IEEE.
[40] Park J, Kim H, Hwang K, Lim S. Deep reinforcement learning based dynamic proportional-integral (PI) gain auto-tuning method for a robot driver system. IEEE Access. 2022; 10:31043-57.
[41] Zhang L, Hou Z, Wang J, Liu Z, Li W. Robot navigation with reinforcement learned path generation and fine-tuned motion control. IEEE Robotics and Automation Letters. 2023; 8(8):4489-96.
[42] Mate S, Pal P, Jaiswal A, Bhartiya S. Simultaneous tuning of multiple PID controllers for multivariable systems using deep reinforcement learning. Digital Chemical Engineering. 2023; 9:1-12.
[43] Rahayu ES, Ma'arif A, Cakan A. Particle swarm optimization (PSO) tuning of PID control on DC motor. International Journal of Robotics and Control Systems. 2022; 2(2):435-47.

[44] Campos J, Jaramillo S, Morales L, Camacho O, Chávez D, Pozo D. PSO tuning for fuzzy PD+ I controller applied to a mobile robot trajectory control. In international conference on information systems and computer science 2018 (pp. 62-8). IEEE.
[45] Charkoutsis S, Kara-mohamed M. A particle swarm optimization tuned nonlinear PID controller with improved performance and robustness for first order plus time delay systems. Results in Control and Optimization. 2023; 12:1-18.
[46] Abajo MR, Sierra-garcía JE, Santos M. Evolutive tuning optimization of a PID controller for autonomous path-following robot. In16th international conference on soft computing models in industrial and environmental applications 2022 (pp. 451-60). Springer International Publishing.
[47] Goud H, Sharma PC, Nisar K, Haque MR, Ibrahim AA, Yadav NS, et al. Metaheuristics algorithm for tuning of PID controller of mobile robot system. Computers, Materials & Continua. 2022; 72(2):3481-92.
[48] Suseno EW, Ma'arif A. Tuning of PID controller parameters with genetic algorithm method on DC motor. International Journal of Robotics and Control Systems. 2021; 1(1):41-53.
[49] Valluru SK, Singh M. Optimization strategy of bio-inspired metaheuristic algorithms tuned PID controller for PMBDC actuated robotic manipulator. Procedia Computer Science. 2020; 171:2040-9.
[50] Zidan A, Tappe S, Ortmaier T. Auto-tuning of PID controllers for robotic manipulators using PSO and MOPSO. In informatics in control, automation and robotics: 14th international conference, Madrid, Spain, 2017 (pp. 339-54). Springer International Publishing.
[51] Liu Y, Jiang D, Yun J, Sun Y, Li C, Jiang G, et al. Self-tuning control of manipulator positioning based on fuzzy PID and PSO algorithm. Frontiers in Bioengineering and Biotechnology. 2022; 9:1-36.
[52] Roveda L, Forgione M, Piga D. Robot control parameters auto-tuning in trajectory tracking applications. Control Engineering Practice. 2020; 101:104488.
[53] Faraj MA, Abbood AM. Fractional order PID controller tuned by bat algorithm for robot trajectory control. Indonesian Journal of Electrical Engineering and Computer Science. 2021; 21(1):74-83.
[54] Ghith ES, Tolba FA. Real-time implementation of tuning PID controller based on whale optimization algorithm for micro-robotics system. In 14th international conference on computer and automation engineering 2022 (pp. 103-9). IEEE.
[55] Yao X, Shi X, Zhang X, Mei X. An ant colony optimization parameter tuning method based on uniform design for path planning of mobile robots. In international conference on advances in electrical engineering and computer applications 2022 (pp. 1183-91). IEEE.
[56] Ghith ES, Tolba FA. Tuning PID controllers based on hybrid arithmetic optimization algorithm and artificial gorilla troop optimization for micro-robotics systems. IEEE Access. 2023; 11:27138-54.

Mohd Faid Yahya graduated from the International Islamic University Malaysia in 2009 with a Bachelor's Degree in Mechatronics Engineering. In 2013, he obtained an M.Sc. in Mechatronics Engineering from the same university. He then earned his Ph.D. in Robotics from Universiti Sains Malaysia in 2019. Currently, he is a Senior Lecturer in the Engineering Department, Faculty of Electrical Technology and Engineering, Universiti Teknikal Malaysia Melaka. His research interests include Robot Control Systems, Visual Servoing, Machine Vision, and Artificial Intelligence.
Email: faid@utem.edu.my

Mad Helmi Ab. Majid received the B.Eng. degree in Mechatronics Engineering and the M.S. Mechatronics Engineering from International Islamic University Malaysia (IIUM), Malaysia, in 2009 and 2012 respectively. He holds a Ph.D. degree in robotics from Universiti Science Malaysia, Penang, in 2019. Currently, he is a senior lecturer at the Department of Software Engineering and Smart Technology, Faculty of Computer and Meta-Technology, Universiti Pendidikan Sultan Idris, Malaysia. His research interests include Robotics, Artificial Intelligence, Internet of Things, Smart Systems, Mechatronics System Design, Swarm Intelligence, Biomimetics, Optimization, Cooperative Robotics, Educational Robotics, and STEM Education.
Email: madhelmi@meta.upsi.edu.my

Appendix I

S. No.	Abbreviation	Full forms
1	GA	Genetic Algorithms
2	PID	Proportional-Integral-Derivative
3	PSO	Particle Swarm Optimization
4	SCARA	Selective Compliance Articulated Robot Arm