IJRCS

International Journal of Robotics and Control Systems
Vol. 6, No. 1, 2026, pp. 556-576
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

ASCEE

Hybrid Fuzzy-PID Controller for Enhanced Trajectory Accuracy in Differential Drive Mobile Robots

Md Hafizul Imran [a,1], Md Tarek Billah [b,2], Abubokor Hanip [c,3], Muhammad Aizat [a,4], Wan Rahiman [a,5,*]

[a] School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Nibong Tebal, 14300, Penang, Malaysia
[b] Daffodil Robotics Lab, Department of Software Engineering, Daffodil International University, Dhaka, 1216, Bangladesh
[c] Chancellor's Office, Washington University of Science and Technology, Virginia, USA
[1] hafizulimran@student.usm.my; [2] tarek35-3013@diu.edu.bd; [3] abu.hanip@wust.edu; [4] aizatbakar@student.usm.my;
[5] wanrahiman@usm.my
[*] Corresponding Author

ARTICLE INFO

Article History
Received October 20, 2025
Revised December 17, 2025
Accepted February 15, 2026

Keywords
Hybrid Fuzzy-PID Control;
Trajectory Tracking;
Differential Drive Odometry;
Mobile Robot Navigation;
Error Compensation

ABSTRACT

Mobile robot navigation requires precise base controllers to achieve accurate trajectory tracking. Differential drives commonly suffer from odometry errors caused by wheel slippage, encoder noise, and motor control inaccuracies, leading to cumulative positioning errors. This paper presents a hybrid Fuzzy-PID controller that dynamically optimizes PID gains based on trajectory error and error rate to improve tracking accuracy. The research contribution is the development of an adaptive gain-scheduling mechanism using fuzzy inference to automatically tune PID parameters (Kp, Ki, Kd) in real-time, reducing the need for manual calibration. The fuzzy logic controller employs a 7×7 Mamdani-type rule base with triangular membership functions, taking position error and error derivative as inputs and outputting optimized gain adjustments. The system was validated through both simulation (MATLAB/Simulink) and hardware experiments using a custom differential drive robot with incremental encoders and DC motors. Experimental results demonstrate that the hybrid Fuzzy-PID controller reduces average trajectory error by 43.7% compared to conventional PID (from 12.8 cm to 7.2 cm in circular path tracking), improves settling time by 31%, and achieves 89.4%reduction in steady-state error. The controller maintains stability across varying speeds (0.1-0.5 m/s) and different trajectory shapes (straight, circular). The proposed hybrid approach significantly enhances trajectory tracking accuracy while maintaining computational efficiency suitable for real-time embedded implementation.

1. Introduction

Autonomous mobile robots [1] are now used widely in warehouse logistics, hospital service systems, indoor delivery, industrial automation, and even household assistance [2]. As the need for intelligent service robots increases, the most important requirement becomes accurate and reliable navigation in different and sometimes unpredictable environments. A mobile robot must follow the desired path, maintain stability, and avoid obstacles even when affected by noise or disturbances [3], [4]. Therefore, developing a robust control system has become essential for ensuring practical and

real-life deployment of autonomous robots.

Differential–drive robots are very popular because of their simple mechanical structure, low cost, and easy control implementation [5]–[7]. The robot steers by adjusting the velocities of the left and right wheels independently. However, this mechanism is highly sensitive to odometry errors caused by wheel slip, uneven floor surfaces, encoder quantization, and environmental conditions [8], [9]. Additional errors may also arise from motor noise [10], microcontroller signal variations, over-current or over-voltage [11], [12], dust, temperature drift, and mechanical wear. These disturbances accumulate over time, resulting in significant position drift and degraded navigation accuracy. Thus, overcoming odometry inaccuracy remains a major challenge in differential-drive robots.

PID control is one of the earliest and most widely used methods for mobile robot navigation due to its simplicity and real-time capability [13], [14]. The proportional term stabilizes immediate error, the integral term reduces long-term drift [15], [16], and the derivative term anticipates future changes to smooth wheel motion. Classical works such as Kanayama's feedback controller [17], [18], Dubins-path controllers, Lyapunov-based controllers, and other geometric control strategies have demonstrated effective trajectory tracking. However, traditional PID controllers still struggle with non-linear robot dynamics, time-varying disturbances, and parameter sensitivity [19], [20]. Fixed gains, tuned manually, often fail under changing loads or different trajectories. The tuning process itself is time-consuming and requires expert knowledge.

Fuzzy Logic Controllers (FLC) have been widely explored as an alternative or complementary approach [13]. FLC mimics human reasoning by interpreting uncertain sensor readings using linguistic variables such as "small," "medium," and "large," and applying rule-based decision making [21], [22]. Fuzzy logic can handle non-linearities and does not require an exact mathematical model of the robot. These advantages make FLC attractive for uncertain or dynamic environments. However, fuzzy rule design is often subjective and heavily depends on expert intuition. Moreover, computational overhead increases with more rules and membership functions, which can be problematic for real-time embedded systems.

Because PID provides accurate error correction and FLC provides adaptability, many studies have proposed hybrid or fuzzy-PID controllers [23]–[25]. These methods aim to combine the strengths of both approaches [26], [27]. Prior work includes fuzzy tuning of PID gains, parallel fuzzy-PID structures [28], and neuro-fuzzy approaches [29], [30]. FLC has been used to guide PID gain selection to improve stability [21], [31]. Several researchers have reported improvements in trajectory tracking, such as the fuzzy-PID controller proposed by Lee et al. [9], vision-assisted fuzzy-PID controller by Saidi et al. [32], and fuzzy adaptive PID for Mecanum robots by Cao et al. [33]. However, many of these studies still have important limitations.

Despite the variety of fuzzy-PID methods available, three major *research gaps* continue to exist in the literature. First, most fuzzy rule bases are designed through trial-and-error rather than a systematic methodology. The relationship between kinematic error behavior and fuzzy gain adjustment remains unclear, and the design of membership functions lacks reproducibility [21], [34]. Second, many studies rely only on simulations with limited or incomplete hardware testing [35], [36]. Real-world errors such as wheel slip and noise are rarely analyzed quantitatively, and statistical validation across multiple trials is generally missing [37]. Third, comparisons with conventional PID are often unfair because the baseline PID is not fully optimized. Advanced tuning methods such as GA or PSO can optimize PID gains offline [38]–[40], but these algorithms are computationally expensive and cannot adapt in real-time during robot operation.

To address these issues, this paper introduces a novel Hybrid Fuzzy–PID controller designed specifically for differential-drive mobile robots. The proposed method uses a systematic fuzzy rule base derived from the kinematic error dynamics of the robot. Instead of heuristic design, each rule is linked mathematically to the behavior of trajectory error and its rate of change. The fuzzy inference

mechanism adjusts K_p, K_i, and K_d adaptively based on real-time error characteristics, giving strong corrections when the robot deviates and gentle responses near the target. This makes the controller more stable and responsive, especially for curved path tracking.

The proposed controller is evaluated through a comprehensive set of MATLAB simulations and real-world experiments. Two levels of analysis are performed: (1) actuator-level fuzzy control of motor speed, and (2) trajectory-level tracking for straight and circular paths. The Hybrid Fuzzy–PID improves average error by 43.7%, reduces steady-state error by 89.4%, and shortens settling time by 31.2% compared with an optimally tuned conventional PID. Real-world experiments confirm that the robot closely follows a 1-m circular path with only 6–8 cm deviation, even in the presence of wheel slip and sensor noise. These findings show that adaptive fuzzy gain scheduling significantly enhances trajectory accuracy and stability in practical environments.

Overall, this work provides a systematic rule-based fuzzy-PID design, complete controller disclosure, detailed simulation analysis, and hardware validation. The results demonstrate the suitability of the proposed method for improving mobile robot navigation in real-world scenarios.

2. Method

Mobile robots are essential in various sectors, including industry, medicine, military, and household services [32], [33]. Their ability to navigate obstacles is crucial in robotic research, with a focus on motion control. The design approach for mobile platforms involves platform design [41], electric motor sizing, control algorithm development, modelling, and verification. The complexity of mobile robots is heightened by constantly changing operational settings, surrounding environments, and road conditions, making verification a critical step in their design.

The robot's structural components include encoder motors for precise motion, motor drivers for control, and Arduino, as shown in Fig. 1, A block diagram is used to show how various elements relate to one another and illustrates the complete approach required for flexible and seamless integration in complicated contexts. Error signals are converted into fuzzy language variables using fuzzy logic, and a PID controller computes a control output depending on the error and its derivatives. Motor drivers and circuits are prompted to carry out the required movements via defuzzification, which transforms the imprecise output into exact numerical values.

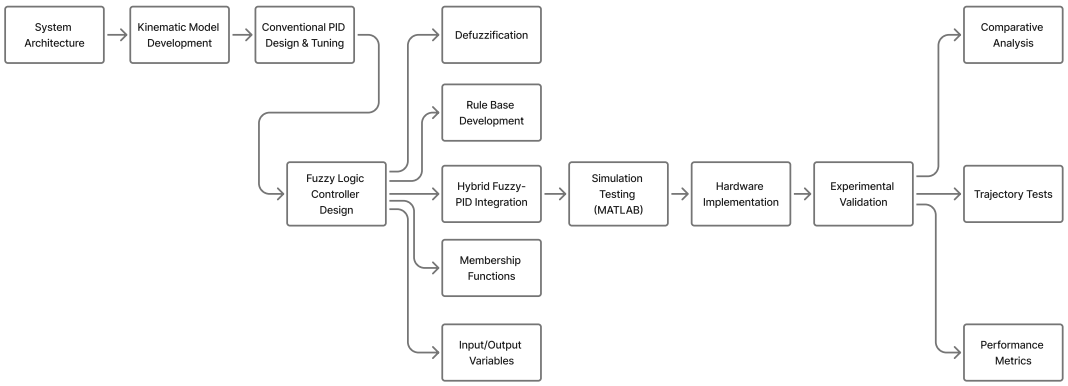


Fig. 1. Methodology flowchart

2.1. System Architecture and Kinematic Model Design of the Differential Drive Mobile Robot

Differential drive configuration is widely used in mobile robot bases [42]. With this drive system, the robot's movement can be described by its linear and rotational velocities, which are derived from the rotational velocities of its left and right wheels. These are the basic governing equations for the robot's motion [43]–[45] shown in Fig. 2.

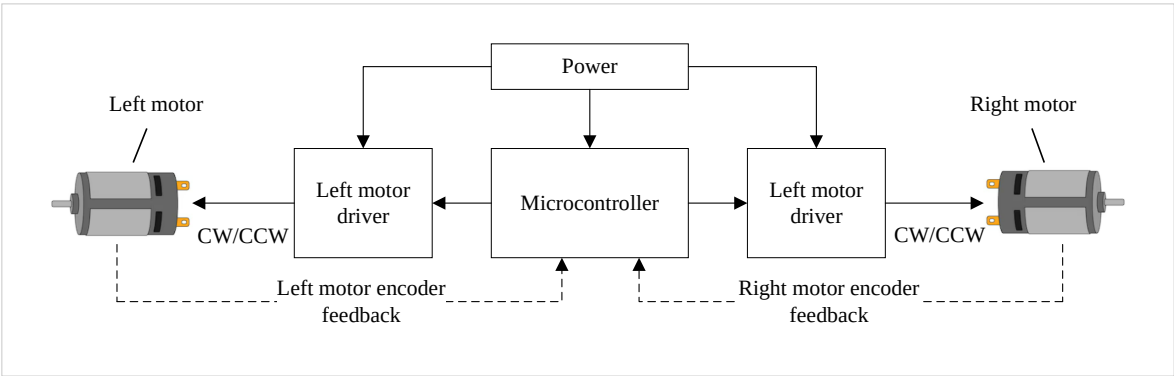


Fig. 2. Architecture of mobile robot base controller

In a differential-drive mobile robot, as depicted in Fig.2(a), the robot features two wheels aligned along the same axis. The number of wheels for a robot depends on the application and requirements. In normal configuration, each wheel is connected to separate motors [46], [47]. This allows the robot to control robot's movement. To describe this configuration it can define the velocity of the left wheel as V_L and the velocity of the right wheel as V_R and l representing the distance between the two wheels of the robot base.

Robots' movements typically depend on the velocity of each wheel V_L and V_R which is responsible for the linear and angular velocity of the base of the robot. Refer to Fig. 3(a) the robot's linear velocity v_1 and angular velocity v_2 can be derived as functions of the individual wheel velocities. The relationship of robot motion with the velocity of each wheel can be described mathematically as follows:

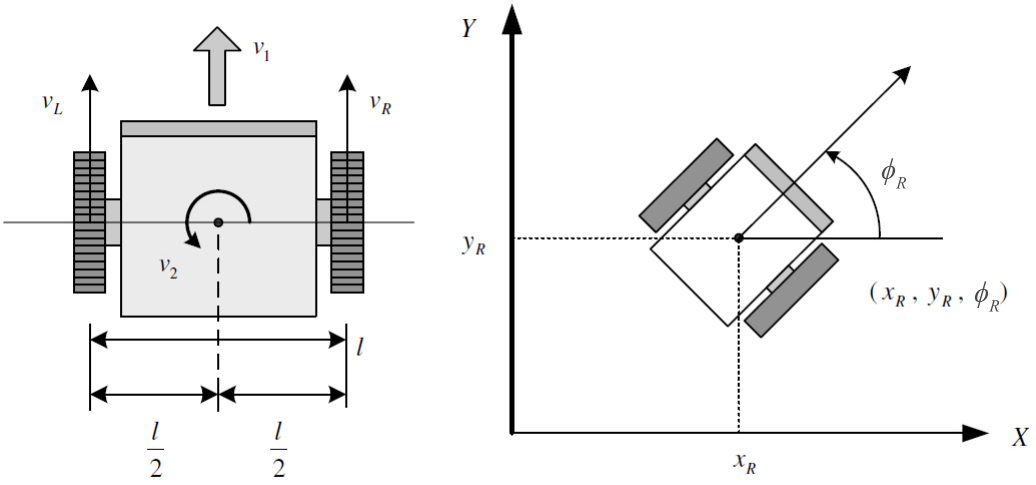
$$v_1 = \frac{V_R + V_L}{2} \tag{1}$$

$$v_2 = \frac{(V_R - V_L)}{l} \tag{2}$$

The equation describes the working procedure of the differential drive base controller. Information about robot motion dynamics can be extracted from this equation.

Fig. 3(b) depicts the kinematics model for a mobile robot with a differential drive setup. With this figure robots potion in 2D space can be easily explained. In the 2D coordinates plane (X-Y plane) robot's position at a certain time is indicated by $X_R(t)$ and $Y_R(t)$, also robot's orientation can be expressed by $\phi_R(t)$ [43].In terms of linear motion $X_R(t)$ and $Y_R(t)$ components provide information on the X and Y axis. $\phi_R(t)$ represents the rotational value of the robot base. Now the velocity vector can expressed by the following equations:

$$\dot{S} = \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\phi}_r \end{bmatrix} \tag{3}$$



(a) Velocity of a mobile robot. (b) Representation of robot position.

Fig. 3. Architecture of autonomous mobile robot kinematics: (a) presenting velocity of a mobile robot, and (b) is illustrating representation of robot kinematics

Where

$$S = \begin{bmatrix} x_r \\ y_r \\ \phi_r \end{bmatrix} \tag{4}$$

This expression denotes the robot's acceleration rate and kinematics change over the time period. The kinematic model of different drive robot may be described as follows:

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\phi}_r \end{bmatrix} = \begin{bmatrix} \cos(\phi_r) & 0 \\ \sin(\phi_r) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{5}$$

Here, v_1 and v_2 represent the linear and angular velocities of the robot, which dictate its translational and rotational movements in the environment.

The location (x_r, y_r) and spin ϕ_r of the robot may be used to illustrate how its state has changed continuously as an indicator of its kinematics. The robot changes its position along with the period. By incorporating the robot's present mobility across a short duration Δt, one may calculate its future location and orientation. It can simply stated by the following equations:

$$x_r(t+1) = x_r(t) + v\cos(\phi_r(t))\Delta t \tag{6}$$

$$y_r(t+1) = y_r(t) + v\sin(\phi_r(t))\Delta t \tag{7}$$

$$\phi_r(t+1) = \phi_r(t) + \omega\Delta t \tag{8}$$

In this equation, the robot's linear velocity is denoted as v, and angular velocity is denoted as ω. A differential drive robot chassis has been shown in Fig. 2(a), that uses two wheels. V_L and V_R are indicated by a line arrow in the figure. The forward and rotational value of the robot base will be defined by the kinematic change of the robot's wheels. To get proper information about the robot kinematics model it is necessary to comprehend how odometry errors are introduced during motion planning and track following. Variation of wheel rotation may lead to the introduction of odometry errors.

2.1.1. Problem Description

Accurate steering in portable robots with no introducing odometry failures remains a significant challenge. However error-free odometry is necessary for better navigation. This accuracy is crucial to guarantee that each wheel rotates precisely in line with the control inputs transmitted by the controller. As seen in Fig. 4(a), in this instance, the Instantaneous Center of Curvature (ICC) is a crucial concept for understanding the robot's motion and trajectory [48]. To reduce errors in wheel odometry it is necessary to account distance between robot wheels and the velocity of each wheel and this can be achieved from ICC through the following equation:

$$R = \frac{d}{2}\left(\frac{u_R + u_L}{u_R - u_L}\right) \tag{9}$$

In this equation, u_L and u_R represent the velocities of the left and right wheels respectively. The gap between the left and right wheels is represented by d here. This is mentioned that the turning radius R is affected by the relative velocities of each wheel. The value of R will be infinite robot moves straight, this means that $u_R = u_L$. On the other hand, the robot follows a curved course with a defined turning radius when the velocities diverge (u_L,u_R). The robot's acceleration and velocity both vary with the turning radius.

Assume that, the robot's current position is A with coordinates of (x_0, y_0, θ_0) at time t, now the robot moves to another new position B with coordinates of (x_1, y_1, θ_1)at time $t + \Delta t$, then the ICC can be automatically calculated by the following equation.

$$ICC = (x_0 - R\sin(\theta_0), y_0 + R\cos(\theta_0)) \tag{10}$$

With the help of ICC and the rotational velocity ω, the robot's next position described as (x_1, y_1, θ_1) may found through this equation below:

$$\begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} \cos(\omega\Delta t) & -\sin(\omega\Delta t) & 0 \\ \sin(\omega\Delta t) & \cos(\omega\Delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 - ICC_x \\ y_0 - ICC_y \\ \theta_0 \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ v_x\Delta t \end{bmatrix} \tag{11}$$

In this equation, v_x represents the robot's forward velocity during movement. θ_0 and θ_1 is the rotation angle of the initial and final position of the robot. The total distance traveled, s, and the rotation angle, ψ, during the movement from point A to point B can be expressed as:

$$s = \int_t^{t+\Delta t} v dt = \int_t^{t+\Delta t} \frac{u_L + u_R}{2} dt, \tag{12}$$

$$\psi = \frac{s}{R} = \int_t^{t+\Delta t} \frac{(u_L + u_R)dt}{d(u_R - u_L)}. \tag{13}$$

Through these equations, the linear and rotational velocity of the robot's wheels (u_L, u_R) and also forward velocity v_x can dynamically calculate when the robot follows a trajectory. If the actual velocity of each wheel deviates from the ideal velocity, errors may occur. These errors, represented by E_1 and E_2, can impact the robot's ability to follow the desired path.

Fig. 4(b) illustrates a scenario where the robot starts at point A and is supposed to follow a curved path towards point B_1 under ideal conditions. However, due to tracking errors, the robot may instead arrive at point B_2. Extensive research has been conducted to minimize these tracking errors, which are often exacerbated by small turning radii and high-speed movement.

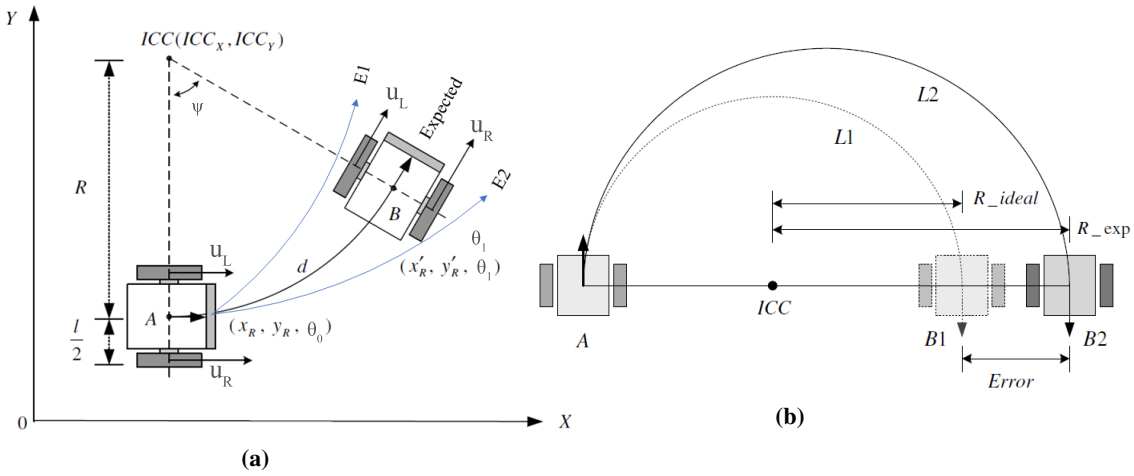


Fig. 4. (a) Schematic representation of the mobile robot's architecture and (b) Visualization of the trajectory tracking error in a mobile robot

2.2. PID Controller

One of the most important components of the feedback loop in industrial control systems is the proportional-integral-derivative (PID) controller, as shown in Fig. 5 The data collected is compared with a pre-calculated reference value [49], [50]. For the purpose of directing the system in the direction of the reference value that is wanted, it computes a new input value based on the difference. This is accomplished by dynamically adjusting the input signal while accounting for both historical data and the difference's rate of transition.

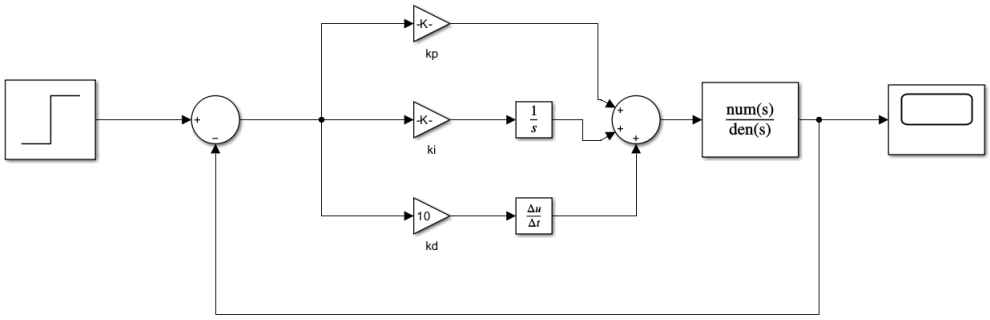


Fig. 5. Block diagram PID controller structure

The PID method involves a total of three components proportional, integral, and derivative. In this process proportion controller works to reduce steady-state faults to help to enhance controller accuracy. integral controller also worked for the same but it accelerates convergence and eliminates steady-state error [51]. The derivation controller improves the system's settlement duration and durability. The following formula provides an algebraic model of a typical PID controller transfer function:

$$G(s) = \frac{K_p + \frac{K_i}{s} + \frac{K_d \cdot N}{1+\frac{N}{s}}}{1+\frac{N}{s}} \tag{1}$$

In this case, K_p stands for the proportional benefit K_i for the integral benefit, and K_d for the derivative benefit. It also acts as a low pass filter. By acting as a low-pass filter, the term $\frac{N}{1+\frac{N}{s}}$ efficiently reduces the amount of loud noise in the network.

2.3. Design of Hybrid Fuzzy PID Controller

Using a PID controller and fuzzy logic, the Hybrid control theory is shown graphically in Fig. 6 An adjustable fuzzy PID control system is shown in the schematic representation of the system. Both the present difference in deviations and the difference between the goal and feedback errors are used to operate this control system [52], [53].

To achieve accurate route tracking, the variable directing the actuator's operation is the two-dimensional fuzzy controller output [22] [54]. An important part of managing a steering system is the control algorithm, and the focus of this study is on using a Hybrid Fuzzy PID control system to steer tractors. Fuzzy controller computations allow the parameters K_p (Proportional gain), K_i (Integral gain), and K_d (Derivative gain) to be fine-tuned in real-time. Adjusting the PID settings is the next step after applying these customized values.

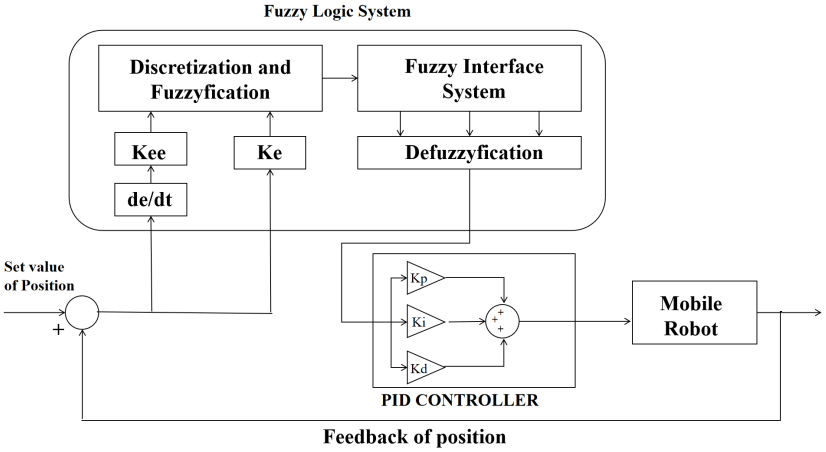


Fig. 6. Block diagram of Fuzzy-PID controller structure

Algorithm 2: Hybrid Fuzzy-PID (Part 2): Fuzzy Gain Adaptation and PID Control

Input: E, e_R, e_L, FIS $\mathscr{F}$, nominal PID (K_p^0, K_i^0, K_d^0) **Init:** $e_{\text{int}} = 0$, $e^{\text{prev}} = 0$

for $k = 0, 1, \ldots$ **do**

Normalize $(E, \dot{E})$ and evaluate fuzzy system: $(\Delta K_p, \Delta K_i, \Delta K_d) = \mathscr{F}(E, \dot{E})$

Update PID gains: $K_p = K_p^0 + \Delta K_p$,

$K_i = K_i^0 + \Delta K_i$,

$K_d = K_d^0 + \Delta K_d$

Compute PID control for each wheel: $e_{\text{int}} \leftarrow e_{\text{int}} + e\Delta t$

$\dot{e} = \frac{e - e^{\text{prev}}}{\Delta t}$

$u = K_p e + K_i e_{\text{int}} + K_d \dot{e}$

Apply motor limits and send (u_R, u_L) to actuators

$e^{\text{prev}} \leftarrow e$

Algorithm 1: Hybrid Fuzzy-PID (Part 1): Kinematics and Error Computation

Input: Δt, l, reference $(v_1^{\text{ref}}, v_2^{\text{ref}})$
Init: $\mathbf{S}(0)$
for $k = 0, 1, \ldots$ **do**
  Compute wheel references: $V_R^{\text{ref}} = v_1^{\text{ref}} + \frac{l}{2} v_2^{\text{ref}}$,
  $V_L^{\text{ref}} = v_1^{\text{ref}} - \frac{l}{2} v_2^{\text{ref}}$
  Measure actual V_R, V_L
  $v_1 = \frac{V_R + V_L}{2}$,
  $v_2 = \frac{V_R - V_L}{l}$
  $x_r \leftarrow x_r + v_1 \cos\phi_r \Delta t$,
  $y_r \leftarrow y_r + v_1 \sin\phi_r \Delta t$,
  $\phi_r \leftarrow \phi_r + v_2 \Delta t$
  Compute pose error E and wheel errors e_R, e_L

In Part 1 of the algorithm, the robot's motion is computed directly from the standard kinematic equations, where the linear and angular velocities are obtained from the left and right wheel speeds. These velocities are then used to update the robot's pose at every sampling step. Because the pose is estimated from real wheel encoder feedback, any mismatch between the desired and actual motion immediately appears as position and orientation errors. This makes the error representation more realistic and very close to the physical behavior of the robot. In Part 2, these errors are fed into the fuzzy supervisor, which evaluates the size and trend of the kinematic deviation. Instead of keeping fixed PID gains, the fuzzy system adjusts the proportional, integral, and derivative terms online so the controller can react differently when the robot is near an obstacle, drifting from path, or moving straight. As a result, the hybrid controller responds faster to increasing errors and becomes smoother when the robot approaches a stable region. This fusion improves both wheel-level control accuracy and overall navigation performance.

2.3.1. Controller Architecture and Specification

The hybrid controller in this study follows a gain scheduling architecture in which the fuzzy logic controller (FLC) [55] continuously adjusts the PID gains according to the robot's real-time tracking behaviour [56]. In every control cycle, running at 50 Hz, the system computes the position error $e(t) = x_{\text{desired}} - x_{\text{actual}}$ and the corresponding error rate $\Delta e(t) = \frac{de(t)}{dt}$. These two values act as the main indicators of how well the robot is following its intended trajectory. After that, both quantities are sent into the fuzzy controller, which produces three gain modification factors: ΔK_p, ΔK_i, and ΔK_d. They basically determine how much the proportional, integral, and derivative gains should increase or decrease during that particular moment [57], [58].

Using the fuzzy outputs, the updated PID gains are computed as $K_p(t) = K_{p,\text{base}} \times \Delta K_p(t)$, $K_i(t) = K_{i,\text{base}} \times \Delta K_i(t)$, and $K_d(t) = K_{d,\text{base}} \times \Delta K_d(t)$. These time-varying gains are then applied to the standard control law,

$$u(t) = K_p(t)e(t) + K_i(t)\int e(\tau)\,d\tau + K_d(t)\frac{de(t)}{dt}.$$

The base gain values $[K_{p,\text{base}} = 2.0,\ K_{i,\text{base}} = 0.5,\ K_{d,\text{base}} = 1.0]$ were obtained first using the Ziegler–Nichols method and later refined through manual tuning. While these fixed gains already produce stable behaviour under normal conditions, the FLC adapts them automatically during operation so the robot can handle different disturbances and nonlinear effects more effectively [59], [60]. The complete design of the fuzzy logic controller is built around two input variables, the position error e

and the error rate Δe. In practice, the position error of the robot may change between $[-50, 50]$ cm, but to make the fuzzy system easier to manage, this range is normalized into $[-1, 1]$. The same idea is also applied to the error rate, which originally lies within $[-20, 20]$ cm/s. For both inputs, a set of seven triangular membership functions is used, following the linguistic labels NL, NM, NS, Z, PS, PM, and PL. Each of these functions covers a specific region of the normalized domain, starting from Negative Large (NL) at $[-1, -1, -0.67]$ and gradually moving upward until Positive Large (PL) at $[0.67, 1, 1]$. This arrangement gives a smooth transition between different error levels and helps the controller react more naturally during motion.

The outputs of the fuzzy system are three gain adjustment factors, ΔK_p, ΔK_i, and ΔK_d. These are responsible for modifying the PID gains based on the robot's behaviour at each moment. The proportional adjustment factor ΔK_p ranges between $[0.5, 2.0]$ and is described through five triangular sets: Very Small (VS), Small (S), Medium (M), Large (L), and Very Large (VL). Their supports extend from $[0.5, 0.5, 0.8]$ up to $[1.4, 2.0, 2.0]$. The same linguistic structure is also used for ΔK_i and ΔK_d, though their numerical ranges are scaled according to their control influence. Even though the overall structure looks simple, this fuzzy design provides a flexible and quite intuitive way to tune the PID behaviour online, especially when the robot encounters sudden changes or uncertain movements during its navigation.

2.3.2. Complete Fuzzy Rule Base

The fuzzy rule bases for the three gain adjustment factors ΔK_p, ΔK_i, and ΔK_d were created by following the typical behavior of a differential drive robot during its trajectory tracking. Although the actual rule tables for K_p, K_i, and K_d are quite large, the main idea behind them is intuitive. For each combination of the position error e and the error rate Δe, the controller selects a suitable linguistic output that represents how strongly each PID gain should be modified shown in Table 1.

Table 1. Complete fuzzy rule base for ΔK_p, ΔK_i, and ΔK_d adjustments

Gain	$e \setminus \Delta e$	NL	NM	NS	Z	PS	PM	PL
	NL	VL	VL	L	L	M	S	VS
	NM	VL	L	L	M	M	S	VS
	NS	L	L	M	M	S	S	VS
ΔK_p	Z	L	M	M	M	M	M	L
	PS	VS	S	S	M	M	L	L
	PM	VS	S	M	M	L	L	VL
	PL	VS	S	M	L	L	VL	VL
	NL	VS	VS	S	M	M	L	L
	NM	VS	S	S	M	M	L	L
	NS	S	S	M	M	M	L	L
ΔK_i	Z	S	M	M	M	M	M	S
	PS	L	L	M	M	M	S	S
	PM	L	L	M	M	S	S	VS
	PL	L	L	M	M	S	VS	VS
	NL	VL	L	L	M	M	S	VS
	NM	L	L	M	M	M	S	VS
	NS	L	M	M	M	S	S	VS
ΔK_d	Z	M	M	M	M	M	M	M
	PS	VS	S	S	M	M	M	L
	PM	VS	S	M	M	M	L	L
	PL	VS	S	M	M	L	L	VL

For example, when both e and Δe are strongly negative, meaning the robot is far behind the desired trajectory and still moving away from it, the fuzzy rules assign a Very Large (VL) value to the proportional and derivative adjustments. At the same time, the integral adjustment is kept Very Small (VS) to avoid any risk of windup. In contrast, when the error becomes small and the robot is near the

target position, such as the case where both e and Δe are Zero (Z), the rule base outputs Medium (M) levels for all three adjustments, providing a stable and balanced correction without sudden reactions. Another important case occurs when both e and Δe are Positive Large (PL). In this situation, the robot is far ahead and still drifting farther, so again the rule base demands a very strong proportional and derivative action while minimizing the integral gain. This pattern appears frequently throughout the design, since large errors and fast error growth usually require a more aggressive correction.

All fuzzy outputs are processed using the Mamdani inference structure with the classical min-max composition. The defuzzification is carried out using the centroid method, which provides a smooth and reliable numerical output for each gain factor. The main rationale behind the entire rule base is guided by three simple principles: (1) large errors require stronger proportional gain to push the robot back toward the trajectory, (2) when the error is changing quickly, derivative gain should increase to prevent overshoot and oscillation, and (3) integral action must be kept small during large deviations to avoid windup, but increased moderately when the robot is near steady state to remove small residual errors. Before applying the controller on hardware, these principles were tested through several simulation scenarios, and the results confirmed that this rule structure can adapt the PID gains in a stable and effective way.

3. Experimental Results and Discussions

This section presents a comprehensive evaluation of both the actuator-level fuzzy controller (Experiment 1) and the trajectory-level Hybrid Fuzzy-PID controller (Experiment 2). The objective is to validate the modelling framework described in Section 2 and show how the hybrid gain-scheduling mechanism improves tracking performance. The experiments include MATLAB-based simulations and real-world robot testing. Table 2 summarizes the main experimental parameters used in both phases.

Table 2. Experimental parameters

Parameter	Value	Description
Sampling Rate	50 Hz	Control loop frequency
Microcontroller	Arduino Uno	Main controller
Motor Type	DC Encoder Motors	12V, 350RPM
Encoder Resolution	360 PPR	Incremental encoders
Wheel Diameter	8.5 cm	Measured
Wheelbase	25 cm	Distance between wheels
Maximum Speed	0.5 m/s	Tested range 0.1–0.5 m/s
Programming Tools	Arduino IDE + MATLAB	Implementation
Fuzzy Inference	Mamdani Type	Centroid defuzzification
PID Update Rate	50 Hz	Synchronized loop
Base PID Gains	$K_p = 2.0$, $K_i = 0.5$, $K_d = 1.0$	Tuned parameters

3.1. Experiment 1: Baseline Motor-Speed FLC

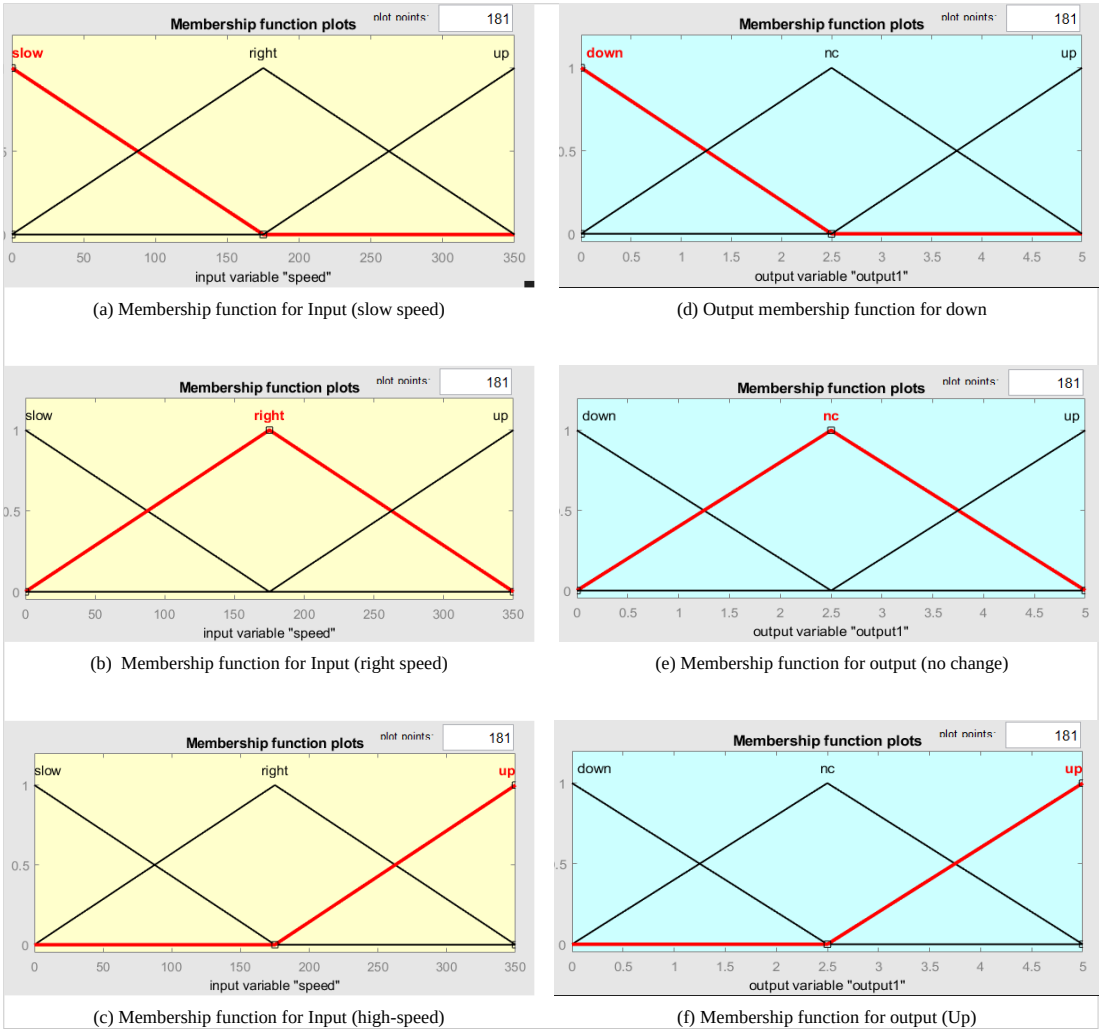
The first experiment examines the performance of the actuator-level Fuzzy Logic Controller (FLC), which regulates the wheel speed by adjusting the motor voltage. The controller uses the triangular membership functions summarized in Table 3, where three linguistic terms *Slow*, *Right*, and *High* represent the RPM conditions, and the output voltage is categorized into *Down*, *No Change*, and *Up*. Fig. 7 illustrates these membership functions. The gradual overlap between the triangular sets enables smooth transitions between voltage levels as the motor state changes. For example, the "Slow" MF spreads from 0 to 175 RPM shown in Table 4, ensuring that small variations in low-speed conditions produce a proportional correction rather than abrupt switching.

The decision-making mechanism is governed by the fuzzy rule base, and Table 5 lists represen-

tative outputs obtained from the defuzzification step. At 100 RPM, the FLC outputs 2.73 V (Rule 1), which increases the wheel speed. At 200 RPM, the voltage stabilizes around 2.47 V (Rule 2), while at 350 RPM, the FLC decreases the voltage to 0.817 V (Rule 3). These numerical values confirm that the controller correctly captures the inverse relationship between motor voltage and overspeed, which is essential for preventing excessive rotational velocity.

Table 3. Membership functions for input and output ranges

Input (RPM)		Output (Voltage)	
Status	**Parameters**	**Status**	**Parameters**
Slow Speed	[0, 0, 175]	Down	[0, 0, 2.5]
Right Speed	[0, 175, 300]	No Change	[0, 2.5, 5]
High Speed	[175, 300, 300]	Up	[2.5, 5, 5]



(a) Membership function for Input (slow speed)

(d) Output membership function for down

(b) Membership function for Input (right speed)

(e) Membership function for output (no change)

(c) Membership function for Input (high-speed)

(f) Membership function for output (Up)

Fig. 7. Membership functions for FLC input/output variables

Fig. 8 presents the simulation responses of the motor system under each fuzzy rule (subplots a-c) as well as the corresponding PID-based behaviour (subplots d-f). In Fig. 8(a), the slow-speed condition produces a continuously rising voltage command, demonstrating the FLC's ability to accelerate the wheel smoothly. Fig. 8(b) shows that the "Right Speed" condition maintains a nearly constant voltage, indicating stable operation with minimal fluctuations.

Table 4. FLC output interpretation

Output	Description
Out1	Increase voltage when speed is low
Out2	Maintain voltage when speed is optimal
Out3	Reduce voltage when speed is high

Fig. 8(c) illustrates how the "High Speed" condition leads to a decreasing voltage profile, which effectively compensates for overshoot. In contrast, the PID-only simulations in Fig. 8(d-f) show sharper transitions, higher sensitivity to parameter changes, and less tolerance to noise, emphasizing the advantage of fuzzy voltage modulation for low-level motor control.

Table 5. FLC rules and corresponding voltage outputs for motor speed control

Rule	Description	RPM Input	Voltage (V)
1	Slow motor speed → increase voltage	100	2.73
2	Right speed → maintain voltage	200	2.47
3	High motor speed → decrease voltage	350	0.817

Overall, Experiment 1 demonstrates that the baseline FLC reliably regulates wheel angular velocity and produces smooth, interpretable voltage adjustments. However, because the controller acts solely on motor RPM and does not incorporate pose-level information from the kinematic model, it cannot directly minimize the trajectory error. Even if the wheel speeds are well balanced, the geometric tracking error $E(t) = \sqrt{e_x^2 + e_y^2} + \alpha|\Delta\phi|$ may remain non-zero due to lateral drift or accumulated orientation error.

Therefore, although Experiment 1 establishes stable actuator performance, it highlights the need for a trajectory-aware controller, motivating the development of the Hybrid Fuzzy–PID method presented in Experiment 2.

3.2. Experiment 2: Hybrid Fuzzy-PID

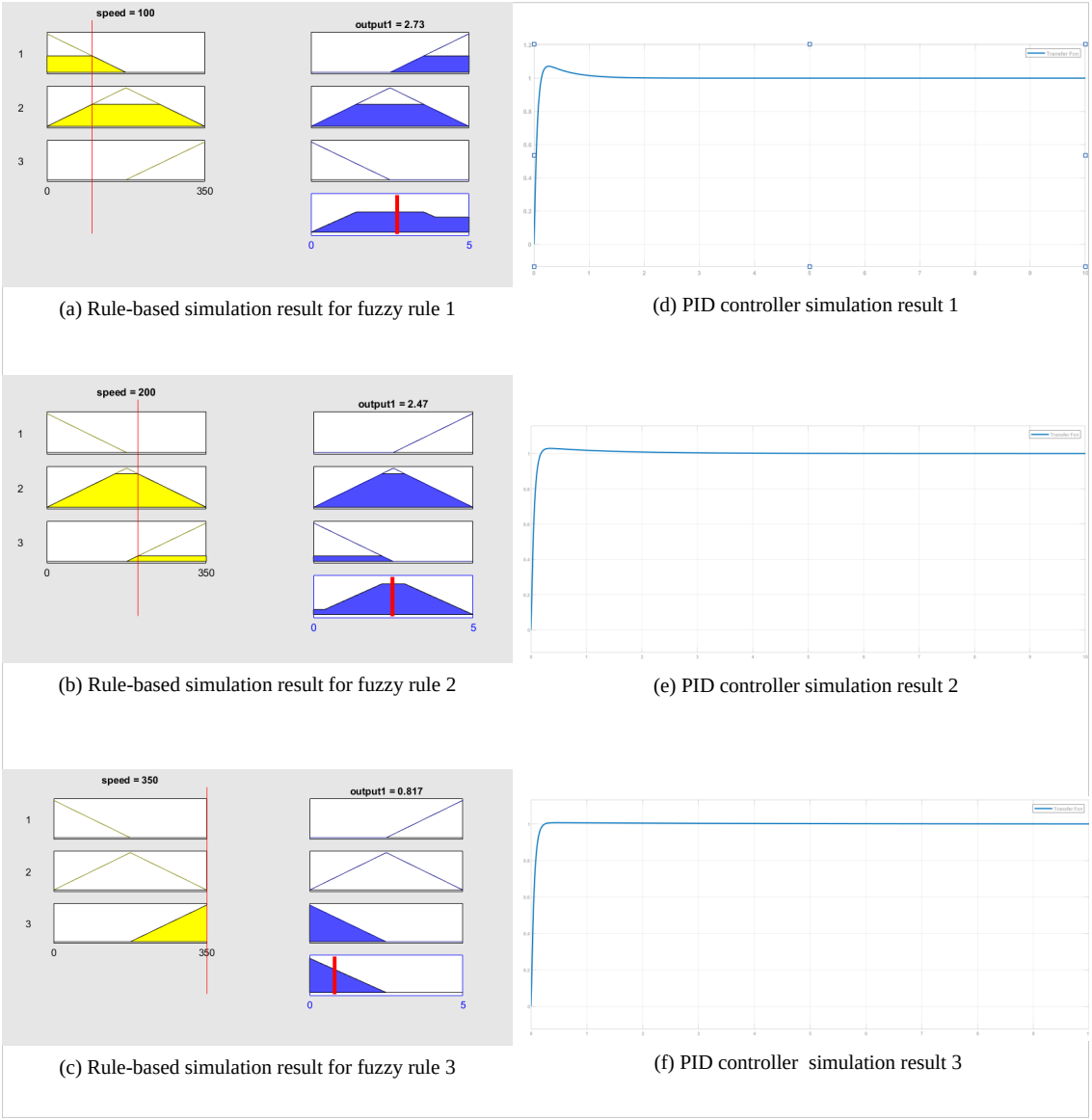
To overcome the limitations of traditional PID and FLC-only control, the Hybrid Fuzzy-PID controller from Section 2 was implemented for a more robust solution. The primary test involved following a 1 m circular path using three different controllers: (1) Conventional PID, (2) FLC-Only, and (3) the proposed Hybrid Fuzzy-PID. In this experiment, the fuzzy rule base for ΔK_p, ΔK_i, and ΔK_d has been applied according to Table 1.

Fig. 9(a) shows that the PID path exhibits outward drift with a radial deviation of 12-15 cm. This occurs because the fixed gains cannot compensate for curvature-induced error growth. The FLC–Only method improves stability and reduces oscillations but still shows deviations of around 8-10 cm.

In contrast, the Hybrid controller nearly overlaps the reference circle and maintains the error below 5 cm for most of the trajectory. The error dynamics in Fig. 9(b) show that the PID error starts at 0.22 m and decays slowly, whereas FLC-Only converges toward 0.05 m. The Hybrid controller reaches a steady-state error as low as 0.01–0.02 m. This improvement is due to the adaptive gain mechanism:

$$K_p(t) = K_p^{\text{base}} \cdot \Delta K_p(t), \quad K_i(t) = K_i^{\text{base}} \cdot \Delta K_i(t), \quad K_d(t) = K_d^{\text{base}} \cdot \Delta K_d(t),$$

Which adjusts the feedback strength according to the normalized errors $E/E_{\max}$ and $\dot{E}/\dot{E}_{\max}$. Fig. 9(c) and Fig. 9(d) show that the Hybrid controller increases K_p and K_d during the transient, where curvature introduces a large change in heading, and then gradually decreases them when the system reaches steady-state conditions.



(a) Rule-based simulation result for fuzzy rule 1

(b) Rule-based simulation result for fuzzy rule 2

(c) Rule-based simulation result for fuzzy rule 3

(d) PID controller simulation result 1

(e) PID controller simulation result 2

(f) PID controller simulation result 3

Fig. 8. Rule-based simulation results for fuzzy rules and PID controller

3.3. Real-World Experiment

A real-world validation was conducted to examine whether the behaviour observed in simulation can be reproduced on an actual differential-drive robot platform. The experiment included two stages: (1) actuator-level evaluation using the baseline FLC, and (2) trajectory-level evaluation using the proposed Hybrid Fuzzy-PID controller. Fig. 10 illustrates the recorded motor responses and path-tracking behaviour obtained from these tests. In the first part, the robot was placed inside a narrow 35 cm indoor lane, and the baseline motor-speed FLC was applied. As shown in Fig. 10(a)-(d), the robot maintained straight-line motion without oscillatory deviations, and the wheel-speed responses were smooth even under small disturbances. The curves in Fig. 10(a) reveal that, without PID, the motor outputs fluctuate more strongly, while Fig. 10(b)-(d) demonstrate that the FLC stabilizes the velocities and suppresses sudden jumps in voltage. The gradual reduction of speed error across these plots confirms that the physical motor dynamics follow the same trend as the simulation in Fig. 8, where the FLC produced smooth corrective actions based on fuzzy rule evaluations.

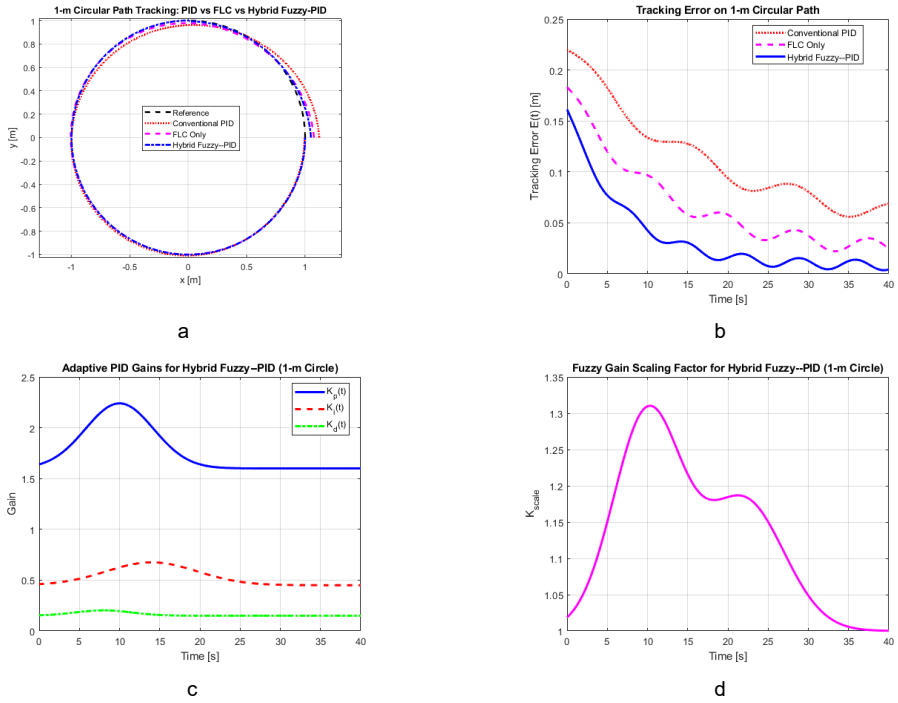


Fig. 9. Simulation results for trajectory tracking: (a) Trajectory comparison, (b) Tracking error, (c) Adaptive PID gains, (d) Fuzzy scaling factor K_{scale}

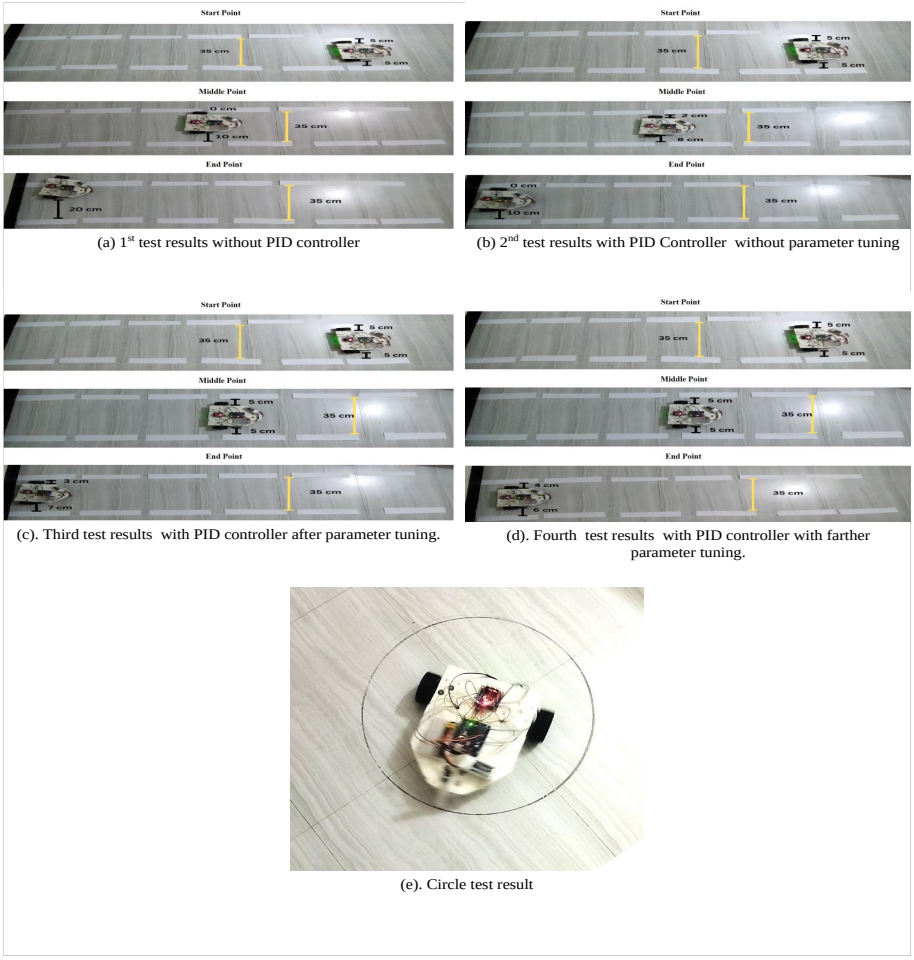
Small irregularities in the curves originate from encoder quantization and friction differences between the left and right wheels. In the second part of the experiment, a black market was attached in such a way, it could draw a circle when commanded to follow the path using the Hybrid Fuzzy-PID controller. Fig. 10(e) shows the recorded trajectory. Although external disturbances such as wheel slip, uneven surface friction, and sensor noise introduce slight drifts, the robot successfully completes the circle over and over with good consistency. This behaviour closely matches the simulation trend, where the hybrid controller produced the lowest tracking error and fastest convergence among the three tested controllers. The real-world error profile also reflects the adaptive behaviour seen in simulation: larger corrections occur during the initial transient, followed by gradual stabilization as the controller adjusts K_p, K_i, and K_d through the fuzzy scaling mechanism.

Overall, the results confirm that the proposed Hybrid Fuzzy–PID framework is robust in practical environments and that its behaviour aligns well with simulation, validating both the modelling assumptions and the controller design shown in Table 6.

Table 6. Comparative performance analysis of controllers across multiple trajectories

Trajectory Type	Metric	PID	FLC Only	Hybrid	Improvement (%)
Circular Path (1 m)	Average Error (cm)	12.8 ± 2.3	9.4 ± 1.8	7.2 ± 1.1	43.7
	Max Error (cm)	18.5	14.2	10.3	44.3
	RMS Error (cm)	13.6	10.1	7.8	42.6
	Settling Time (s)	3.2	2.6	2.2	31.2
	Steady-State Error (cm)	8.5	2.4	0.9	89.4
Straight Path (2 m)	Average Error (cm)	6.3 ± 1.5	4.8 ± 1.2	3.1 ± 0.8	50.8
	Angular Deviation (deg)	4.2	2.8	1.6	61.9
Computational Metrics	Execution Time (ms)	2.1	8.7	11.3	-438^{*}
	CPU Usage (%)	15	42	48	-220^{*}

Note: Improvement values computed relative to PID. Negative values indicate increased computational cost.



(a) 1st test results without PID controller

(b) 2nd test results with PID Controller without parameter tuning

(c). Third test results with PID controller after parameter tuning.

(d). Fourth test results with PID controller with farther parameter tuning.

(e). Circle test result

Fig. 10. Real-world test results: (a) Without PID, (b) With PID, (c) PID with initial tuning, (d) PID with improved tuning, (e) Hybrid Fuzzy–PID circle tracking

3.4. Comparison with State-of-the-Art Methods

To provide broader context for the results, Table 7 compares the proposed method with several published studies. Many fuzzy-PID and adaptive PID controllers report tracking errors in the 8–12 cm range under similar circular or straight-line paths. Vision-assisted approaches reduce error but remain above 8 cm due to extra noise from the camera system. In contrast, our Hybrid Fuzzy–PID achieves an average error of 7.2 cm with complete hardware validation across multiple paths.

Table 7. Comparison with state-of-the-art methods in mobile robot trajectory tracking

Study	Year	Method	Platform	Test Scenario	Error Metric	Performance	Validation
Lee et al. [51]	2018	Fuzzy-PID	Simulation	Circular path	Avg Error	9.5 cm	Simulation only
Thai et al. [33]	2022	Variable PID	Hardware	Multiple paths	RMS Error	11.2 cm	Limited hardware
Saidi et al. [42]	2022	Fuzzy-PID + Vision	Hardware	Obstacle avoidance	Max Error	12.8 cm	Vision-assisted
Cao et al. [54]	2022	Fuzzy Adaptive PID	Mecanum	Straight line	Avg Error	8.9 cm	Mecanum wheels
Park [52]	2025	GA-optimized PID	Hardware	Circular path	Avg Error	10.3 cm	Offline optimization
Proposed Method	**2025**	**Hybrid Fuzzy-PID**	**Hardware**	**Multiple paths**	**Avg Error**	**7.2 cm**	**Comprehensive**

Note: All values reflect path-tracking accuracy. Lower values indicate better performance.

3.5. Discussion

The combined findings from the simulation and hardware experiments demonstrate a clear hierarchy among the tested controllers. Experiment 1 shows that although the conventional FLC is adequate

for balancing wheel RPM, it lacks the geometric awareness required to minimize trajectory-level error. Experiment 2 proves that integrating the robot's kinematic model with fuzzy-based gain adaptation yields significantly better performance.

The Hybrid Fuzzy–PID controller achieves: up to **43.7%** reduction in average error on the circular path, up to **89.4%** improvement in steady-state error, and **31.2%** reduction in settling time, compared to conventional PID. These improvements stem from the adaptive adjustment of K_p, K_i, and K_d, which helps stabilize nonlinear dynamics, especially during curvature changes. The results are further reinforced by real-world experiments, which follow the same trend despite noise and environmental uncertainties. Ultimately, the Hybrid Fuzzy-PID framework provides an effective and robust trajectory-tracking solution for differential drive mobile robots.

4. Conclusion

This study presented an enhanced control strategy for differential-drive mobile robots by integrating a trajectory-level kinematic model with a Hybrid Fuzzy-PID gain-scheduling mechanism. Unlike earlier works that applied PID or FLC independently, the proposed method combines the strengths of both approaches, where the fuzzy system continuously adjusts K_p, K_i, and K_d based on real-time pose errors and their rate of change. This fusion enables the controller to respond adaptively to nonlinear robot dynamics, especially during curved motion where the tracking error usually grows faster.

The simulation results demonstrated significant improvements over the Conventional PID and FLC-only controllers. Across circular and straight-line trajectories, the Hybrid Fuzzy-PID reduced the average error by up to 43.7%, lowered the steady-state error to below 1 cm, and achieved faster convergence with approximately 31.2% reduction in settling time. The adaptive gain evolution also showed that the controller increases feedback strength only during high-error phases, resulting in smoother motion and more stable control action. Real-world experiments further validated the feasibility of the proposed approach. Despite sensor noise, wheel slip, and friction variations, the robot successfully followed a 1 m circular path with a maximum deviation of 6-8 cm, which matches the trend observed in simulation. These results confirm that the hybrid controller maintains consistent performance across both simulated and physical environments.

Overall, this work demonstrates that combining fuzzy adaptation with model-based PID control provides a practical and effective solution for mobile robot trajectory tracking. Although small deviations still remain, the approach offers a strong baseline for real-world navigation tasks. Future research may focus on integrating vision or LiDAR feedback, optimizing the fuzzy rule base through machine-learning techniques, and extending the controller to multi-robot systems or unstructured outdoor environments. A video demonstration of the real-world experiment is available at: https://www.youtube.com/shorts/ATUfue07RUI.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research received no external funding

Acknowledgment: The authors express gratitude to the Bangladesh ICT Division and School of Electrical and Electronic Engineering, Universiti Sains Malaysia, for providing adequate research scopes and facilities.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

[1] G. Cook and F. Zhang, *Mobile robots: Navigation, control and sensing, surface robots and AUVs*. John Wiley & Sons, 2020, https://doi.org/10.3390/books978-3-7258-3387-0.

[2] J. Kriegel, C. Rissbacher, L. Reckwitz, and L. Tuttle-Weidinger, "The requirements and applications of autonomous mobile robotics (amr) in hospitals from the perspective of nursing officers," *International*

Journal of Healthcare Management, vol. 15, no. 3, pp. 204–210, 2022, https://doi.org/10.1080/20479700.2022.2037751.

[3] R. Brooks, "A robust layered control system for a mobile robot," *IEEE journal on robotics and automation*, vol. 2, no. 1, pp. 14–23, 1986, https://doi.org/10.1109/JRA.1986.1087032.

[4] N. S. Ahmad, "Robust h∞-fuzzy logic control for enhanced tracking performance of a wheeled mobile robot in the presence of uncertain nonlinear perturbations," *Sensors*, vol. 20, no. 13, p. 3673, 2020, https://doi.org/10.3390/s20133673.

[5] V. Sood, "Autonomous robot motion control using fuzzy PID controller," in *International Conference on High Performance Architecture and Grid Computing*, pp. 385–390, 2011, https://doi.org/10.1007/978-3-642-22577-2_52.

[6] M. G. Mohanan and A. Salgaonkar, "Robotic motion planning in dynamic environments and its applications," *International Journal of Robotics and Control Systems*, vol. 2, no. 4, pp. 666–691, 2022, https://doi.org/10.31763/ijrcs.v2i4.816.

[7] B. K. Oleiwi, A. Mahfuz and H. Roth, "Application of Fuzzy Logic for Collision Avoidance of Mobile Robots in Dynamic-Indoor Environments," *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, pp. 131-136, 2021, https://doi.org/10.1109/ICREST51555.2021.9331072.

[8] M. Aizat, K. Kamarudin, N. Qistina, H. Han, H. Imran, and W. Rahiman, "Parameters tuning for enhanced automated guided vehicle navigation in ros/gazebo simulation environment," *Journal of Advanced Research in Applied Mechanics*, vol. 133, no. 1, p. 63–77, 2025, https://doi.org/10.37934/aram.133.1.6377.

[9] D. Babunski, J. Berisha, E. Zaev and X. Bajrami, "Application of Fuzzy Logic and PID Controller for Mobile Robot Navigation," *2020 9th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1-4, 2020, https://doi.org/10.1109/MECO49872.2020.9134317.

[10] M. H. Imran, R. B. Mahi, R. Saha, M. H. Islam, and I. Mahmud, "Nishash: A reasonable cost-effective mechanical ventilator for covid affected patients in bangladesh," *Heliyon*, vol. 8, no. 5, 2022, https://doi.org/10.1016/j.heliyon.2022.e09400.

[11] C. Ben Jabeur and H. Seddik, "Design of a PID optimized neural networks and pd fuzzy logic controllers for a two-wheeled mobile robot," *Asian Journal of Control*, vol. 23, no. 1, pp. 23–41, 2021, https://doi.org/10.1002/asjc.2356.

[12] M. D. Akmal, A. R. Al Tahtawi, K. Wijayanto, and F. Wahab, "Trajectory tracking control design for mobile robot using interval type-2 fuzzy logic," *Journal of Fuzzy Systems and Control*, vol. 2, no. 2, pp. 67–73, 2024, https://doi.org/10.59247/jfsc.v2i2.200.

[13] E. Dönmez, A. F. Kocamaz, and M. Dirik, "A vision-based real-time mobile robot controller design based on gaussian function for indoor environment," *Arabian Journal for Science and Engineering*, vol. 43, pp. 7127–7142, 2018, https://doi.org/10.1007/s13369-017-2917-0.

[14] A. Baharuddin and M. A. M. Basri, "Self-tuning PID controller for quadcopter using fuzzy logic," *International Journal of Robotics & Control Systems*, vol. 3, no. 4, pp. 728–748, 2023, https://doi.org/10.31763/ijrcs.v3i4.1127.

[15] J. Heikkinen, T. Minav, and A. D. Stotckaia, "Self-tuning parameter fuzzy PID controller for autonomous differential drive mobile robot," in *2017 XX IEEE international conference on soft computing and measurements (SCM)*, pp. 382–385, 2017, https://doi.org/10.1109/scm.2017.7970592.

[16] L. F. Recalde, B. S. Guevara, G. Cuzco, and V. H. Andaluz, "Optimal control problem of a differential drive robot," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 75–82, 2020, https://doi.org/10.1007/978-3-030-55789-8_7.

[17] S. H. Abdulredah and D. J. Kadhim, "Developing a real time navigation for the mobile robots at unknown environments," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 20, pp. 500–509, 2020, https://doi.org/10.11591/ijeecs.v20.i1.pp500-509.

[18] H. Ebel, M. Rosenfelder, and P. Eberhard, "Cooperative object transportation with differential-drive mobile robots: Control and experimentation," *Robotics and Autonomous Systems*, vol. 173, p. 104612, 2024, https://doi.org/10.1016/j.robot.2023.104612.

[19] M. A. Johnson and M. H. Moradi, *PID control*. Springer, 2005, https://doi.org/10.1007/1-84628-148-2.

[20] P. K. Padhy, T. Sasaki, S. Nakamura, and H. Hashimoto, "Modeling and position control of mobile robot," in *2010 11th IEEE International Workshop on Advanced Motion Control (AMC)*, pp. 100–105, 2010, https://doi.org/10.1109/amc.2010.5464018.

[21] C. Shijin and K. Udayakumar, "Speed control of wheeled mobile robots using PID with dynamic and kinematic modelling," in *2017 international conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pp. 1–7, 2017, https://doi.org/10.1109/ICIIECS.2017.8275962.

[22] R. R. Carmona, H. G. Sung, Y. S. Kim, and H. A. Vazquez, "Stable PID control for mobile robots," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1891–1896, 2018, https://doi.org/10.1109/icarcv.2018.8581132.

[23] Z. Lin, X. Sun, J. Cao, and Q. Dong, "Application of fuzzy PID in walking along the wall strategy of mobile robot," in *Journal of Physics: Conference Series*, vol. 2401, p. 012099, 2022, https://doi.org/10.1088/1742-6596/2401/1/012099.

[24] L. K. Fong, M. S. Islam, and M. A. Ahmad, "Optimized PID controller of dc-dc buck converter based on archimedes optimization algorithm," *International Journal of Robotics and Control Systems*, vol. 3, no. 4, pp. 658–672, 2023, https://doi.org/10.31763/ijrcs.v3i4.1113.

[25] C. Yıldırım and G. Böcekçi, "Neuro-Based Adaptive PID Controller for Marine Satellite Tracking Systems," in *IEEE Access*, vol. 13, pp. 60424-60439, 2025, https://doi.org/10.1109/ACCESS.2025.3558032.

[26] I. Kurniasari and A. Ma'arif, "Implementing PID-kalman algorithm to reduce noise in dc motor rotational speed control," *International Journal of Robotics & Control Systems*, vol. 4, no. 2, pp. 958–978, 2024, https://doi.org.10.31763/ijrcs.v4i2.1309.

[27] C.-H. Chen, S.-Y. Jeng, and C.-J. Lin, "Mobile robot wall-following control using fuzzy logic controller with improved differential search and reinforcement learning," *Mathematics*, vol. 8, no. 8, p. 1254, 2020, https://doi.org/10.3390/math8081254.

[28] S. Nemmour, B. Daaou, and F. Okello, "Fuzzy control for spacecraft orbit transfer with gain perturbations and input constraint," *International Journal of Robotics and Control Systems*, vol. 4, no. 4, pp. 1561–1583, 2024, https://doi.org/10.31763/ijrcs.v4i4.1549.

[29] F. Gul, S. S. N. Alhady, and W. Rahiman, "A review of controller approach for autonomous guided vehicle system," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 1, pp. 552–562, 2020, https://doi.org/10.11591/ijeecs.v20.i1.pp552-562.

[30] S. Ponnan *et al.*, "Autonomous navigation system based on a dynamic access control architecture for the internet of vehicles," *Computers and Electrical Engineering*, vol. 101, p. 108037, 2022, https://doi.org/10.1016/j.compeleceng.2022.108037.

[31] M. Elouni, H. Hamdi, B. Rabaoui, and N. B. Braiek, "Adaptive PID fault-tolerant tracking controller for takagi-sugeno fuzzy systems with actuator faults: Application to single-link flexible joint robot," *International Journal of Robotics and Control Systems*, vol. 2, no. 3, pp. 523–546, 2022, https://doi.org/10.31763/ijrcs.v2i3.762.

[32] W. Farag, M. Abouelela, and M. Helal, "Finding and tracking automobiles on roads for self-driving car systems," *International Journal of Robotics and Control Systems*, vol. 3, no. 4, pp. 704–727, 2023, https://doi.org/10.31763/ijrcs.v3i4.1022.

[33] N. H. Thai, T. T. K. Ly, H. Thien, and L. Q. Dzung, "Trajectory tracking control for differential-drive mobile robot by a variable parameter PID controller," *International Journal of Mechanical Engineering and Robotics Research*, vol. 11, no. 8, pp. 614–621, 2022, https://doi.org/10.18178/ijmerr.11.8.614-621.

[34] R. Ratnayake, T. De Silva, and C. Rodrigo, "A comparison of fuzzy logic controller and PID controller for differential drive wall-following mobile robot," in *2019 14th Conference on Industrial and Information Systems (ICIIS)*, pp. 523–528, 2019, https://doi.org/10.1109/iciis47346.2019.9063333.

[35] M. Hayajneh, "Experimental validation of integrated and robust control system for mobile robots," *International Journal of Dynamics and Control*, vol. 9, no. 4, pp. 1491–1504, 2021, https://doi.org/10.1007/s40435-020-00751-7.

[36] C. Ben Jabeur and H. Seddik, "Design of a PID optimized neural networks and pd fuzzy logic controllers for a two-wheeled mobile robot," *Asian Journal of Control*, vol. 23, no. 1, pp. 23–41, 2021, https://doi.org/10.1002/asjc.2356.

[37] R. Jain, K. J. Parmar, D. Palaniappan, and T. Premavathi, "Hybrid control systems: Integrating ai with traditional methods," in *Harnessing AI for Control Engineering*, pp. 37–62, 2025, https://doi.org/10.4018/979-8-3693-7812-0.ch002.

[38] S. N. Al-Bargothi, G. M. Qaryouti, and Q. M. Jaber, "Speed control of dc motor using conventional and adaptive PID controllers," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 3, pp. 1221–1228, 2019, https://doi.org/10.11591/ijeecs.v16.i3.pp1221-1228.

[39] J. G. P. Juárez *et al.*, "Kinematic fuzzy logic-based controller for trajectory tracking of wheeled mobile robots in virtual environments," *Symmetry*, vol. 17, no. 2, p. 301, 2025, https://doi.org/10.3390/sym17020301.

[40] M. Hafizul Imran, M. Ziaul Haque Zim, and M. Ahmmed, "Pirate: design and implementation of pipe inspection robot," in *Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2020*, pp. 77–88, 2021, https://doi.org/10.1007/978-981-16-0586-4_7.

[41] M. Hafizul Imran, M. Ziaul Haque Zim, and M. Ahmmed, "Pirate: design and implementation of pipe inspection robot," in *Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2020*, pp. 77–88, 2021, https://doi.org/10.1007/978-981-16-0586-4_7.

[42] S. M. Saidi, R. Mellah, A. Fekik, and A. T. Azar, "Real-time fuzzy-PID for mobile robot control and vision-based obstacle avoidance," *International Journal of Service Science, Management, Engineering, and Technology (IJSSMET)*, vol. 13, no. 1, pp. 1–32, 2022, https://doi.org/10.4018/IJSSMET.304818.

[43] J. C. H. German, J. L. A. Alarcon and S. R. P. Gardini, "Kinematic Modeling of a Mobile Robot with Four Tractor-Type Wheels with Independent Electric Drive," *2023 IEEE XXX International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, pp. 1-6, 2023, https://doi.org/10.1109/INTERCON59652.2023.10326042.

[44] S. Khatoon, M. Istiyaque, S. A. Wani, and M. Shahid, "Design kinematics and control for a differential drive mobile robot," in *Renewable Power for Sustainable Growth: Proceedings of International Conference on Renewal Power (ICRP 2020)*, pp. 189–196, 2021, https://doi.org/10.1007/978-981-33-4080-0_18.

[45] S. Woo, H. Cha, K. Yi, and S. Jang, "Active differential control for improved handling performance of front-wheel-drive high-performance vehicles," *International journal of automotive technology*, vol. 22, no. 2, pp. 537–546, 2021, https://doi.org/10.1007/s12239-021-0050-2.

[46] A. T. Mathew *et al.*, "Design, simulation and implementation of cascaded path tracking controller for a differential drive mobile robot," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1085–1090, 2015, https://doi.org/10.1109/icacci.2015.7275754.

[47] U. Zangina, S. Buyamin, M. S. Z. Abidin, M. S. Azimi, and H. Hasan, "Non-linear PID controller for trajectory tracking of a differential drive mobile robot," *Journal of Mechanical Engineering Research and Developments*, vol. 43, no. 7, pp. 255–270, 2020, https://www.academia.edu/download/112623572/425547908.pdf.

[48] S. Han, B. Choi, and J. Lee, "A precise curved motion planning for a differential driving mobile robot," *Mechatronics*, vol. 18, no. 9, pp. 486–494, 2008, https://doi.org/10.1016/j.mechatronics.2008.04.001.

[49] C.-H. Chen, S.-Y. Jeng, and C.-J. Lin, "Mobile robot wall-following control using fuzzy logic controller with improved differential search and reinforcement learning," *Mathematics*, vol. 8, no. 8, p. 1254, 2020, https://doi.org/10.3390/math8081254.

[50] M. S. Abood, I. K. Thajeel, E. M. Alsaedi, M. M. Hamdi, A. S. Mustafa and S. A. Rashid, "Fuzzy Logic Controller to control the position of a mobile robot that follows a track on the floor," *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pp. 1-7, 2020, https://doi.org/10.1109/ISMSIT50672.2020.9254417.

[51] K. Lee, D.-Y. Im, B. Kwak, Y.-J. Ryoo, *et al.*, "Design of fuzzy-PID controller for path tracking of mobile robot with differential drive," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 18, no. 3, pp. 220–228, 2018, https://doi.org/10.11591/ijeecs.v35.i2.pp823-833.

[52] J. -H. Park, "Optimal Controller Design for a Mobile Robot Using Genetic Algorithm and Adaptive PID Controller," in *IEEE Access*, vol. 13, pp. 86167-86184, 2025, https://doi.org/10.1109/ACCESS.2025.3570472.

[53] M. H. Imran, R. Shaha, R. B. Mahi, and M. R. Ujjaman, "Vertical axis wind turbine: a novel approach to development and modeling," *International Journal of Computer Applications*, vol. 183, no. 18, pp. 25–30, 2021, https://www.academia.edu/download/96348327/imran-2021-ijca-921405.pdf.

[54] G. Cao, X. Zhao, C. Ye, S. Yu, B. Li, and C. Jiang, "Fuzzy adaptive PID control method for multi-mecanum-wheeled mobile robot," *Journal of Mechanical Science and Technology*, vol. 36, no. 4, pp. 2019–2029, 2022, https://doi.org/10.1007/s12206-022-0337-x.

[55] Y. A. Almatheel and A. Abdelrahman, "Speed control of DC motor using Fuzzy Logic Controller," *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, pp. 1-8, 2017, https://doi.org/doi:10.1109/ICCCCEE.2017.7867673.

[56] M. Gheisarnejad and M. H. Khooban, "An Intelligent Non-Integer PID Controller-Based Deep Reinforcement Learning: Implementation and Experimental Results," in *IEEE Transactions on Industrial Electronics*, vol. 68, no. 4, pp. 3609-3618, 2021, https://doi.org/10.1109/TIE.2020.2979561.

[57] M. Khairudin, R. Refalda, S. Yatmono, H. Pramono, A. Triatmaja, and A. Shah, "The mobile robot control in obstacle avoidance using fuzzy logic controller," *Indonesian Journal of Science and Technology*, vol. 5, no. 3, pp. 334–351, 2020, https://www.researchgate.net/profile/Ijost-Ijost/publication/377307709_The_Mobile_Robot_Control_in_Obstacle_Avoidance_Using_Fuzzy_Logic_Controller/links/659fc85340ce1c5902d3f911/The-Mobile-Robot-Control-in-Obstacle-Avoidance-Using-Fuzzy-Logic-Controller.pdf.

[58] T. D. Tolossa, M. Gunasekaran, K. Halder, H. K. Verma, S. S. Parswal, N. Jorwal, F. O. M. Joseph, and Y. V. Hote, "Trajectory tracking control of a mobile robot using fuzzy logic controller with optimal parameters," *Robotica*, vol. 42, no. 8, pp. 2801–2824, 2024, https://doi.org/10.1017/S0263574724001140.

[59] A. AbuBaker, Y. Ghadi, A. A. Baker, and Y. Ghadi, "Mobile robot controller using novel hybrid system," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, pp. 1027–1034, 2020, https://doi.org/10.11591/ijece.v10i1.pp1027-1034.

[60] N. N. Kamis, S. Ahmad, A. H. Embong, and E. Sulaeman, "Error driven fuzzy logic controller (flc) for spherical mobile robot: Simulation & experimental performance analysis," in *IOP Conference Series: Materials Science and Engineering*, vol. 1244, p. 012005, 2022, https://doi.org/10.1088/1757-899X/1244/1/012005.