Robotica (2024), **42**, pp. 2801–2824
doi:10.1017/S0263574724001140

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Trajectory tracking control of a mobile robot using fuzzy logic controller with optimal parameters

Tesfaye Deme Tolossa[1], Manavaalan Gunasekaran[2], Kaushik Halder[1], Hitendra Kumar Verma[1], Shyam Sundar Parswal[1], Nishant Jorwal[1], Felix Orlando Maria Joseph[1] and Yogesh Vijay Hote[1]

[1]Department of Electrical Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand, India
[2]Department of Electrical and Electronics Engineering, CIT, Combature, Tamil Nadu, India
Corresponding author: Felix Orlando Maria Joseph; Email: m.orlando@ee.iitr.ac.in

Received: 22 February 2024; **Revised:** 11 June 2024; **Accepted:** 17 June 2024; **First published online:** 19 September 2024

Keywords: wheeled mobile robot; membership function; particle swarm optimization; fuzzy logic controller; path tracking; experimental evaluation

Abstract

This work investigates the use of a fuzzy logic controller (FLC) for two-wheeled differential drive mobile robot trajectory tracking control. Due to the inherent complexity associated with tuning the membership functions of an FLC, this work employs a particle swarm optimization algorithm to optimize the parameters of these functions. In order to automate and reduce the number of rule bases, the genetic algorithm is also employed for this study. The effectiveness of the proposed approach is validated through MATLAB simulations involving diverse path tracking scenarios. The performance of the FLC is compared against established controllers, including minimum norm solution, closed-loop inverse kinematics, and Jacobian transpose-based controllers. The results demonstrate that the FLC offers accurate trajectory tracking with reduced root mean square error and controller effort. An experimental, hardware-based investigation is also performed for further verification of the proposed system. In addition, the simulation is conducted for various paths in the presence of noise in order to assess the proposed controller's robustness. The proposed method is resilient against noise and disturbances, according to the simulation outcomes.

1. Introduction

The autonomous two-wheeled differential drive (DD) mobile robot's popularity has significantly increased over time. This is due to the wide range of real-world applications of the mobile robotics. Autonomous mobile robotics applications have become increasingly important in recent years in fields like agriculture, industry, space exploration, and road inspection [1–5].

As a result of recent advancements in artificial intelligence (AI), robots are currently used to carry out from basic human needs to complicated and autonomous actions to execute various tasks on a large scale [3, 6, 7]. Autonomous mobile robots can be employed in many household tasks like food delivery, floor cleaning, item placement, etc., because of technological improvements [8–10]. Thus, a different methods are employed in order to carry out these various activities without the need for human intervention. These techniques include fuzzy logic controllers (FLCs) [1, 2], neural network (NN) technique [11], hybrid fuzzy controller [12], genetic algorithm (GA) [13], algorithm of simulated annealing (SA) [14], particle swarm optimization (PSO) [15], adaptive neuro-fuzzy (ANFIS) [16], model-predictive control (MPC) techniques [17], etc.

Path tracking control is primarily used to steer the robot along predefined trajectories based on their position, acceleration, and speed as functions of time. This path tracking can be represented by a variety of shapes, such as curves, straight lines, and segments of circles. One of the biggest challenges of the autonomous mobile robot navigation is trajectory tracking [18, 19].

According to the reviewed literature, many researchers have tracked the motion of mobile robots using different types of controllers. The most commonly used methods for path tracking include model-based control, fuzzy logic control, NNs control, and GA approach [19]. In ref. [20], the tracking-error learning control (TELC) technique is introduced for accurate mobile robot path tracking. However, this tracking error is limited to steady-state behavior control. The model-based predictive system is employed for path tracking and steering control of the mobile robot in ref. [21]. However, predictive control is computationally challenging and sensitive to model errors. In ref. [22], adaptive sliding mode control (SMC) was applied for path tracking of two-wheeled mobile robots to overcome external forces and inertia uncertainty. The SMC reduces disturbances, but there are certain problems with the controller, such as complexity and tuning issues. The problem of accurately tracking the control of a two-wheel mobile robot following a path using a backstepping control strategy is addressed in ref. [23]. Backstepping control's shortcomings include the model's complexity, sensitivity to errors, and difficulties with real-time implementation.

The model-based control strategy requires the mathematical models of the system being controlled to track the desired trajectories. One of the most popular model-based techniques is the Jacobian pseudo-inverse technique, which is especially well known for its efficiency in certain trajectory tracking tasks. However, this method may suffer from certain limitations related to singularity, convergence, accumulation of errors, and computational complexity [18, 24, 25].

Fuzzy and NN controllers are static, nonlinear systems that may be precisely designed to meet system needs and function according to the principles of human reasoning. The selection between fuzzy logic and NN controllers depends on factors such as system complexity, availability of data, interpretability requirements, and computational complexity [19, 26]. A variety of research studies have been presented regarding NN-based trajectory tracking of mobile robots. These methods commonly use input variables, such as wheel angles to determine the position and orientation of mobile robot. Forward kinematics is used to produce training data. The NN method is highly interested in robotic control systems due to their ability to effectively approximate nonlinear functions [19, 26]. The network architecture and types of neurons used in a NN determine the performance of the controlled system. NN-based control is limited due to computational difficulties arising from complex networks and require large amounts of training data to guarantee accurate path tracking [18, 19, 27].

On the other hand, fuzzy logic uses human knowledge and experience to provide a non-mathematical approach to solving control problems [28]. This method makes it easier to process data at a higher level and integrate rule-based computing effectively. Fuzzy logic has been increasingly used in autonomous vehicle control in recent decades due to its extensive popularity in a variety of industries [29]. It is particularly effective in handling nonlinear systems with uncertainties and often avoids the need for detailed system models. Due to their effective learning capabilities, fuzzy systems are well suited for developing inference algorithms for real-time motion planning. Fuzzy controllers are more adaptable in real-time applications as compared to NNs. Significantly, FLC exhibits faster response times than a NN when both are applied to the same system with identical load conditions [19, 26]. A mobile robot's local and global navigation, path planning, steering, and control are all future tasks that could be handled by fuzzy logic. The advantage of using fuzzy controllers is ability to deal with imprecise input and uncertaint [1, 29].

FLC algorithm is a promising solution due to its inherent resilience, resilience to input disturbances, and capacity for handling noise [19, 26, 30]. The fact that FLC has been successfully implemented in challenging applications demonstrates its outstanding capabilities for handling nonlinearities. In ref. [30], robustness is defined as the system's capacity to continue operating in the presence of noise or disturbances from both the internal and the external. Fault tolerance is also defined as system ability to continue providing services in the presence of faults that arise during development or operation, including component damage [30, 31]. In order to achieve fault tolerance and guarantee system stability and reliability, the reliable controllers are essential [29, 30].

The trajectory tracking of the DD mobile robot is designed using a fuzzy-based proportional integral derivative (PID) controller in ref. [32]. Selecting the appropriate and optimized number of rules and

membership functions for a fuzzy-based PID controller is complicated. In ref [33], electric wheelchair trajectory tracking using FLC control strategy is designed. Fuzzy sliding model-based control is used for controlling an autonomous DD mobile robot by considering the friction torques in refs. [34, 35]. In ref. [36], fuzzy sliding mode and backstepping controllers are used to track the motion of a two-wheeled robot to overcome model errors and external disturbances. The complexity of modeling and potential overfitting for particular operating conditions are some issues with fuzzy sliding mode controllers, which can limit the system's performance. A fuzzy logic approach for efficiently controlling mobile robots under nonholonomic constraints was described in ref. [37]. In ref. [38], a comparison of the fuzzy-based PID and traditional PID under various mobile robot dynamics is presented. However, fuzzy logic performs better than other alternative controllers for nonlinear, complex, and undefined problems.

The FLC is characterized by adaptability, resilience, simplicity, and reliability that make it useful in a variety of applications have been emphasized in previous studies [2, 30]. However, with the large number of fuzzy controllers, the membership function parameter design is essentially based on trial and error [19]. Finding an accurate and effective controller is difficult when using the heuristic-based method of membership function selection. The large number rules are also another challenge for fuzzy control because the length of computations increases and slows the response of the system [28, 35].

Tuning the membership function is a major difficulty of the FLC design, which has been addressed by researchers employing a variety of optimization strategies over the past 10 years [19, 28]. Selecting parameters for the membership function has a big impact on creating reliable and efficient fuzzy controller. But the majority of the membership function selection is based on a trial-and-error methodology that lacks autonomy and learning [19, 28, 35]. We have developed optimized membership function parameters for mobile robot control in response to these relevant problems.

The kinematic model of a two-wheeled mobile robot with FLC-based nonholonomic DD trajectory tracking is the main objective of this study. This work concentrates on the kinematic model by considering the position, orientation, and velocities of the mobile robot without including dynamic forces and torques. For trajectory tracking control of nonholonomic mobile robots, FLC is commonly used due to its robustness, resilience, and effectiveness in handling uncertainties and nonlinearities. However, design of the huge number rule and membership functions are common problems for fuzzy controllers, which increases the length of computation times and decreases efficiency the controller [19, 28].

To address these challenges, we have proposed a novel approach that uses PSO to optimize membership function parameters and GA to minimize the size of the rule base. PSO approach improve rule aggregation and feature selection to reduce computing costs and improve system performance. Even though many strategies have been used to modify FLC membership functions, automating this process still remains challenging [19]. Here, we have introduced an innovative approach to maximize the tuning membership function parameters of the proposed method for mobile robot control. To the best of our knowledge, PSO-based FLC membership function optimization has not been studied in any literature previously for mobile robot control. Our goal is to improve the performance of controllers in real-world scenarios.

The study deals with the use of FLC to control and track the trajectory of a two-wheeled DD mobile robot. Here, we have ensured that the velocity is uniform and the path planning is tracked precisely using the proposed method. Two inputs and two outputs are presented in the proposed controller. The mobile robot's distance error and the orientation angle error are the inputs of controller. Likewise, the linear and rotational angular velocities of the two-wheeled DD mobile robot are the outputs of FLC. The initial position and orientation angle, in this case, are considered to be nonzero. For the position control of mobile robots, self-localization is a notably important feature. It is impossible to navigate the robot without knowing its location. The robot's current location can be ascertained using a variety of different techniques, such as visual-based localization, the dead-reckoning approach, and other methods [1, 2]. The vision camera is employed in this research work to locate the position and orientation angle of the mobile robot.

The contributions of this research paper are as follows: (1) design of an optimized FLC for a two-wheel DD mobile robot that can easily navigate in the plane; (2) design of the PSO approach for

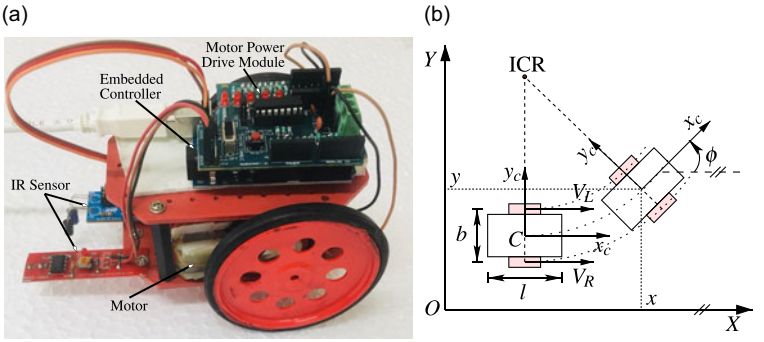

Figure 1. *(a) Two-wheel differential drive mobile robot. (b) Mobile robot coordinate system.*

optimization of membership function parameters; (3) conducting MATLAB-based simulations to evaluate the proposed controller performance and comparing it with existing standard kinematic controllers such as minimum norm solution (MNS), Jacobian transpose-based controllers, and closed-loop inverse kinematics (CLIK); (4) extensive experimental is performed to test the efficiency of the proposed method in real-world scenarios; and (5) evaluating the robustness of the proposed controller in the presence of sensor noise.

The remainder of the paper is organized as follows: in Section II, the kinematic modeling of the wheeled mobile robot is presented. FLC design is discussed in Section III. Section IV presents the simulation findings and discussions, and Section V presents the conclusion.

2. Kinematics model

In this work, a two-wheel DD robot is employed to evaluate the proposed controller. The prototype of the mobile robot with two-wheel DD is shown in Figure 1(a). The coordinate system of proposed mobile robot is shown Figure 1(b). Two DC motors are used for driving each individual wheel. This robot can move either in forward or backward but not in the sides. The mobile robot center is assigned by the coordinate system $(C : x_C, y_C)$. It is assumed that the mobile robot will travel in a horizontal plane. We assign the arbitrary coordinate system $O : XY$ to the working plane. The position of the two wheeled mobile robot C is defined as (x, y, ϕ), x is the coordinate of C projected on x-axis, y is the coordinate C projected on y-axis, and ϕ is the orientation angle of coordinate system C with respect to the coordinate system O. The two-wheeled mobile robot's kinematic model is designed at velocity level so that the wheels side slippage are negligible. The movement of the two-wheeled mobile robot on a curved trajectory is restricted to having the same instantaneous center of rotation in order to guarantee that the side slips are zero. The simplified constraints for the mobile robot with zero side slips are given as [39]:

$$\dot{y}\cos\phi - \dot{x}\sin\phi = 0 \tag{2a}$$

$$\dot{x}\cos\phi + \dot{y}\sin\phi + \frac{b}{2}\dot{\phi} = r\dot{\theta}_R \tag{2b}$$

$$\dot{x}\cos\phi + \dot{y}\sin\phi - \frac{b}{2}\dot{\phi} = r\dot{\theta}_L \tag{2c}$$

where the angular velocity components of the right and left wheels are defined by $\dot{\theta}_R$ and $\dot{\theta}_L$, respectively, while r is the wheel radius. The derived and simplified mobile robot kinematic model with all the aforementioned constraints is as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cos\phi & \frac{r}{2}\cos\phi \\ \frac{r}{2}\sin\phi & \frac{r}{2}\sin\phi \\ \frac{r}{b} & -\frac{r}{b} \end{bmatrix} \begin{bmatrix} \dot{\theta}_R \\ \dot{\theta}_L \end{bmatrix} \tag{2d}$$

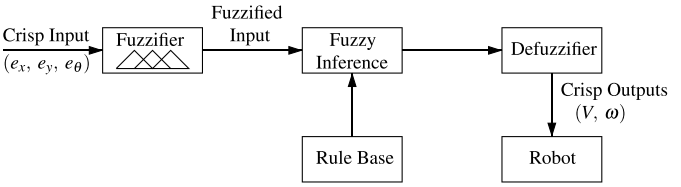

Figure 2. *Block diagram of mobile robot control using an FLC.*

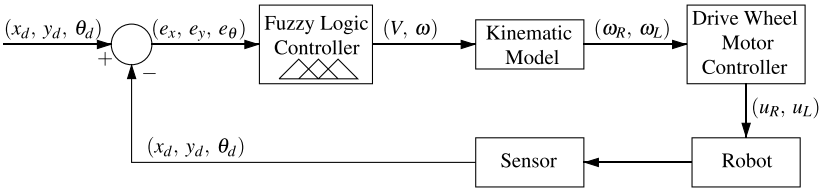

Figure 3. *Structure of fuzzy logic controller.*

The wheel servo motors have an optical encoder. Motor encoders are used to measure the wheel velocities of the mobile robot. The velocities of the mobile robot $[\dot{x}\,\dot{y}, \dot{\phi}]^{\mathrm{T}}$ are computed using Equation (2d), and the integral of these velocities provides the position $[x, y, \phi]^{\mathrm{T}}$ of the mobile robot with respect to a fixed coordinate system.

Let $V = \sqrt{\dot{x}^2 + \dot{y}^2}$ represent the linear velocity of two-wheel mobile robot, while the linear velocity components of the right and left wheels, respectively, given by $V_R = r\dot{\theta}_R$ and $V_L = r\dot{\theta}_L$. Therefore, the equation can be written as:

$$\begin{bmatrix} V \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix} \begin{bmatrix} V_R \\ V_L \end{bmatrix} \tag{2e}$$

From Equation (2e), the velocities of the wheels are derived in terms of mobile robot velocity and yaw rate $\dot{\phi}$ as:

$$\begin{bmatrix} V_R \\ V_L \end{bmatrix} = \begin{bmatrix} 1 & \frac{b}{2} \\ 1 & -\frac{b}{2} \end{bmatrix} \begin{bmatrix} V \\ \dot{\phi} \end{bmatrix} \tag{2f}$$

3. Fuzzy logic controller design

In this section, we are going to design FLC to fuzzify and deduce rules from the inputs and then defuzzify the outputs. Here, the FLC is designed with two inputs (distance and orientation angle errors) and two outputs (linear and rotational angular velocities). Figure 2 displays the FLC block diagram proposed to control two-wheeled DD mobile robots. Figure 3 depicts the FLC's structure.

There are four steps for designing a fuzzy controller: fuzzification, inference mechanism system, rule base, and defuzzification.

3.1. Fuzzification

The fuzzy membership function is an essential tool for identifying the fuzziness inherent in the data and usually transforms the crisp input values supplied to the fuzzy inference system. The first step of the fuzzy inference system is called fuzzification, which involves transforming exact input values into fuzzy language variables with membership degrees between 0 and 1. The fuzzy membership function provides a visual representation of each value's degree of membership in a specified fuzzy set.

The two input controllers are the distance error e_d (which represents the difference between the desired position and the robot's current position) and the orientation angle error e_θ (which represents the difference between the desired orientation angle and the robot's current orientation angle). The linear and angular velocities are the outputs of the proposed controller. The values of the two inputs are given by Equations (3.1a) and (3.1b):

$$e_d = \sqrt{(x_T - x)^2 + (y_T - y)^2} \tag{3.1a}$$

$$e_\theta = tan^{-1}\left(\frac{y_T - y}{x_T - x}\right) \tag{3.1b}$$

where x_T and y_T are the target positions of mobile robot. The e_d and e_θ are distance and orientation errors, respectively. In this study, fuzzification is accomplished by the application of the triangle structure membership function. The orientation error input employs five linguistic variables. These linguistic variables are listed as follows:

- ZA → Zero Angle
- PA → Positive angle
- VPA→ Very Positive Angle
- NA → Negative Angle
- VNA → Very Negative Angle

Similarly, the distance error linguistic variables are also given as follows:

- ZA → Zero Distance
- SD → Small Distance
- MD → Medium Distance
- LD →Long Distance
- VLD→ Very Long Distance

The following linguistic variables are used to determine output linear velocity membership functions:

- ZS → Zero Speed
- SS → Slow Speed
- MS → Medium Speed
- FS → Fast Speed
- VFS →Very Fast Speed

The following linguistic variables are also used to determine output angular velocity membership functions:

- ZAV → Zero Angular Velocity
- MPAV → Medium Positive Angular Velocity
- MNAV → Medium Negative Angular Velocity
- LPAV → Large Positive Angular Velocity
- LNAV → Large Negative Angular Velocity

Different membership functions are used to describe orientation error (e_θ) and distance error (e_d). Figures 4 and 5 show the input membership functions of orientation and distance errors, respectively. The membership functions for linear and angular velocity outputs are shown in Figure 6a and b, respectively.

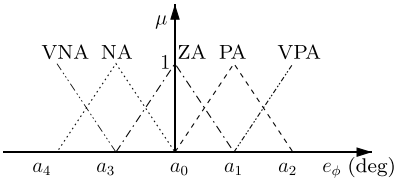

Figure 4. *Input membership functions for orientation angle error.*

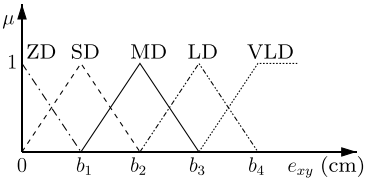

Figure 5. *Input membership functions for distance error.*

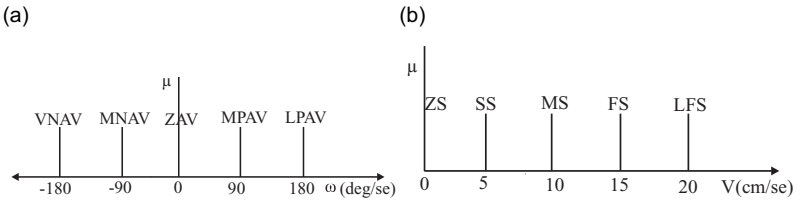

Figure 6. *Membership functions for output angular velocity (ω) and linear velocity (V), respectively.*

Tuning the membership function is a major difficulty of the FLC design, which has been addressed by researchers employing a variety of optimization strategies over the past 10 years [28, 35]. Selecting parameters for the membership function has a big impact on creating trustworthy and efficient FLC designs. The majority of the membership function selection are based on a trial-and-error methodology that lacks autonomy and learning [28].

In this work, the PSO technique has been employed to find the optimal value of all input membership function's parameters. Here, root mean square error (RMSE) between actual (x, y) and desired position (x_d, y_d) of mobile robot is considered as cost function, which is shown in Equation (3.1c):

$$Cost\ function = \sqrt{\frac{1}{n}\sum_{i=1}^{n}((x_d(i) - x(i))^2 + (y_d(i) - y(i))^2)} \tag{3.1c}$$

The objective of this algorithm is to find the optimal value of nine parameters of input membership function while minimizing the cost value. In this PSO algorithm, nine swarms in total have been taken into account, and each swarm contains 50 particles. The swarm minimizes the cost function to advance toward the optimal position depending on the personal best attribute of each particle and the global best attribute. Velocity and position of the swarm are updated, according to the rules given by Equations (3.1d) and (3.1e):

$$\begin{aligned} V_j(k+1) = {} & wV_j(k) + c_1 r_1(P_{Best_j}(k) - P_j(k)) \\ & + c_2 r_2(G_{Best}(k) - P_j(k)) \end{aligned} \tag{3.1d}$$

$$P_j(k+1) = P_j(k) + V_j(k+1) \tag{3.1e}$$

Here, $P_j(k+1)$ and $V_j(k+1)$ are the j^{th} particle's position and velocity at $(k+1)^{th}$ iteration, respectively. $P_{Best_j}(k)$ is the personal best position of the j^{th} particle at k^{th} iteration. $G_{Best}(k)$ is the global best of the swarm at the k^{th} iteration. The variables c_1 and c_2 are the coefficients of acceleration; w is the inertia

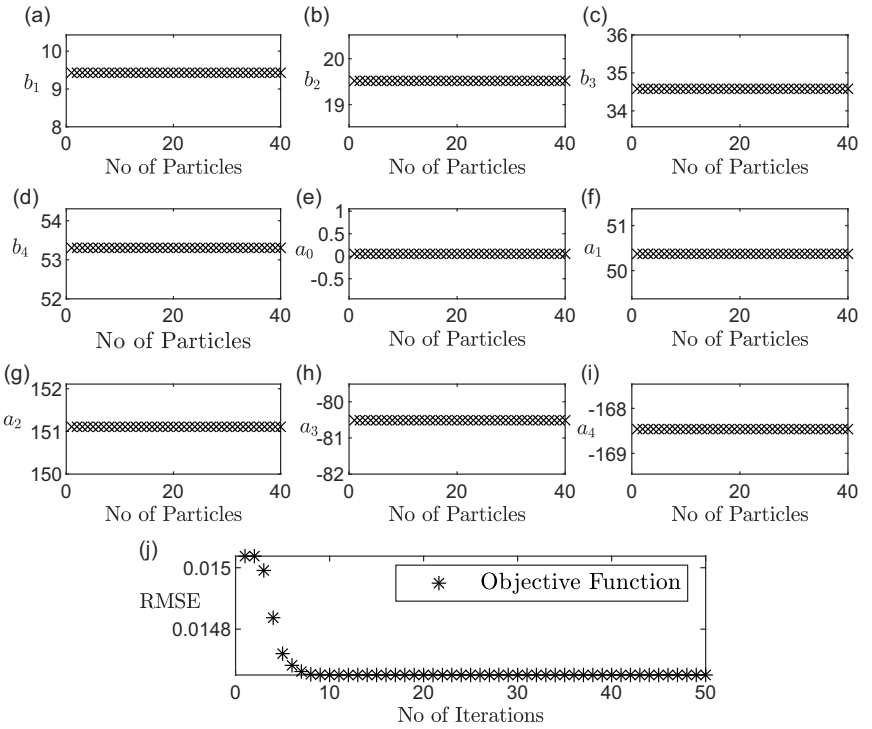

Figure 7. *Results of PSO (optimal values of nine parameters and fitness function profile).*

weight; and r_1 and r_2 are the random integers in the range [0, 1]. Figure 7(a)–(i) illustrate particles' positions of all swarms at the last iteration and also cost function (Figure 7(j)) profile during the off-line optimization. From the view point of Fig. 7(a)–(i), optimal values for nine parameters are: ($a_0 = 0.0532, a_1 = 50, a_2 = 151, a_3 = -80, a_4 = -168, b_1 = 9.428, b_2 = 19.51, b_3 = 34.58, b_4 = 54.58$).

3.2. Rule base and inference system

The fuzzy controller operates by taking precise values as input for the errors in orientation angle and distance. These values are then fuzzified to provide membership degrees that correspond to appropriate fuzzy sets. This process is handled by the inference system, which turns the unclear input data into an output by adjusting the linear velocities of the robot's left and right wheels based on a rule basis. The rule base is a set of statements that, utilizing if-then logic to link the inputs and outputs. We have employed five lingusitic variables for each inputs (orientation angle and distance errors) to simplify the reasoning.

The formation of the rule base for fuzzy controller is also another an essential process that enhances the system's control. The complex interactions between input variables and desired output velocities are represented by the rules. These rules allow the system to follow the desired trajectory effectively by considering various error scenarios. The performance of the controller is adjusted through incremental changes to the fuzzy universe of discourse that improve tracking accuracy and robustness in a variety of operational conditions [28, 40]. Once membership functions have been defined, it is necessary to formulate fuzzy rule bases. We have developed an expert system with 25 if-then rules. A total of 25 rules base are given as follows for both linear and angular velocities:

1. If $e_d = ZD$ and $e_\theta = ZA$, then $V = SS$ and $\omega = ZAV$.
2. If $e_d = ZD$ and $e_\theta = PA$, then $V = ZS$ and $\omega = MPAV$.
3. If $e_d = SD$ and $e_\theta = ZA$, then $V = SS$ and $\omega = ZAV$.

Table I. *Rule base of the output linear velocity.*

e_d (cm) / e_θ(deg)	ZA	PA	NA	VPA	VNA
ZD	SS	ZS	ZS	ZS	ZS
SD	SS	SS	SS	SS	SS
MD	MS	MS	MS	MS	MS
LD	FS	FS	FS	FS	FS
VLD	VFS	VFS	VFS	VFS	VFS

Table II. *Rule base of the output angular velocity.*

e_d(cm) / e_θ(deg)	ZA	PA	NA	VPA	VNA
ZD	ZAV	MPAV	MNAV	LPAV	LNAV
SD	ZAV	MPAV	MNAV	LPAV	LNAV
MD	ZAV	MPAV	MNAV	LPAV	LNAV
LD	ZAV	MPAV	MNAV	LPAV	LNAV
VLD	ZAV	MPAV	MNAV	LPAV	LNAV

4. If $e_d = SD$ and $e_\theta = PA$, then $V = SS$ and $\omega = MPAV$.

.

.

.

25. If $e_d = VLD$ and $e_\theta = VNA$, then $V = VFS$ and $\omega = LNAV$.

The rule base for the output linear velocity of two-wheeled DD mobile robots using FLC control is given in Table I. The FLC rule base for the mobile robot's output angular velocity is given in Table II.

The two-wheel DD mobile robot under investigation has a maximum linear velocity of 20 cm/s. The actual velocities of the robot must be taken into account while designing the FLC to satisfy control objectives. Fuzzy control mechanisms are incorporated at both the input and output stages as part of this design. The language variable for linear velocity (V) is divided into equal interval in the universe of discourse [0, 20 cm/s], and this range is defined by five constant membership functions. In a similar manner, the angular velocity (ω) has five corresponding membership functions and is divided into equal intervals within the universe discourse [−180, 180 deg/s].

Here, we have presented the GA to automate and simplify the development of fuzzy control rules. GA uses genetic reduction approaches to find the minimum number of rules required to build fuzzy models in addition to automatically producing fuzzy rules. Computational complexity is reduced during the learning phase by removing redundant rules and merging conflicting rules [28, 40]. Fuzzy controller rule is represented by the following structure:

R^i=If (e_d is $A^i_{1h_1}$ and e_θ is $B^i_{2h_2}$ . . .) Then (V is C_m) and (ω is D_m)

In this case, distance error, orientation error, linear velocity, and angular velocity are represented by the membership functions A, B, C, and D, respectively. The linguistic variables in the input and output increase exponentially with the number of fuzzy rules. For each crisp input, $h = 1, 2, 3, \cdots, n$, the number of linguistic variables for the crisp input is given by $h_1, h_2, h_3, \ldots, h_n$ and the crisp output linguistic variable is represented by $h_1, h_2, h_3, h_4, \ldots, h_m$. The candidate rules r is given as r=$h_1 \times h_2 \times h_3 \times h_4, \cdots, h_n$.

The output of the fuzzy rule in the fuzzy model is either a constant or a linear combination of the input variables. For FLC, the total candidate rules increase by $p = r \times h_m$, if the consequent value is constant and related to the linguistic variables of output. As a result, in order to determine the proper control action, developing such a controller requires a large amount of computing time per step. In order

Table III. *Specification of the mobile robot.*

Physical parameters	l	b	r	m
	0.15 m	0.12 m	0.035 m	0.35 kg
Operating conditions	$V_{\min}$	$V_{\max}$	$\omega_{\min}$	$\omega_{\max}$
	1.5 V	6.0 V	$\pi/2$ (rad/s)	4π (rad/s)

to generalize fuzzy control rules, the weight factor w_m^i must be included in order to assess whether a rule should be included in the rule base. The generalized rule base structure is rewritten as:

R^i=If (e_d is $A^i_{1h_1}$ and e_θ is $B^i_{2h_2}$ . . .) Then (V is $w^i_m C_m$) and (ω is $w^i_m D_m$)

→ If $\sum_{m=1}^{h_m} w_m^i = 0$, if the i^{th} rule has no consequences and then it should be removed from the candidate rule base.

→ If $\sum_{m=1}^{h_m} w_m^i = 1$, if the i^{th} rule has a single consequence and hence it should be included straight into the rule base.

→ If $\sum_{m=1}^{h_m} w_m^i = 2$ or more, if the i^{th} rule has two or more consequences, we have applied the following equation to merge the two or more consequences into one consequence: $V_N = \frac{1}{2}\sum_{i=1}^{n} Z_{C_i}$, where the number of consequences is denoted by n, the center numeric value of each fuzzy rule consequence is represented by Z_{C_i}, and the final numerical value is represented by V_N. Several results are produced by selecting the linguistic variable for V_N that has the highest membership value.

3.2.1. Defuzzification

The control system's inference engine is responsible for making decisions via approximative reasoning. Fuzzy inference results are obtained by defuzzification, which is the process of obtaining a single number from the combined fuzzy set output. Defuzzification is essentially accomplished by an algorithm that chooses the best crisp value from the fuzzy collection. We have used the center of gravity (CoG) technique for defuzzification in our study. The CoG is located at center of the body and represents the point at which the majority of an object's weight is concentrated. Mathematically, CoG is given as follows:

$$Z_{crisp} = \frac{\int \mu(z)\, z \times dz}{\int \mu(z)\, dZ}$$

where Z_{crisp} is represents the crisp (non-fuzzy) output value obtained through defuzzification, while $\int \mu(z)$ is represents the membership function of the fuzzy set at a particular value z. Finally, these specifications can be executed for fuzzy controller through the running of basic MATLAB code on suitable computer systems.

4. Results and discussions

To demonstrate the effectiveness of the suggested FLC with optimal membership function parameters, both the simulation and experimental results are provided and analyzed in this section. The two-wheeled mobile robot parameters used for simulation and experimental purposes are presented in Table III. The minimum and maximum active operating conditions of the driving wheels are also presented in Table III. If the output results are outside these specific operating conditions, the drive wheels will be in the dead zone or saturation region.

4.1. Simulation results

The proposed method and other control techniques are simulated using MATLAB software. The performance and reliability of the proposed controller for a two-wheel DD mobile robot are assessed using

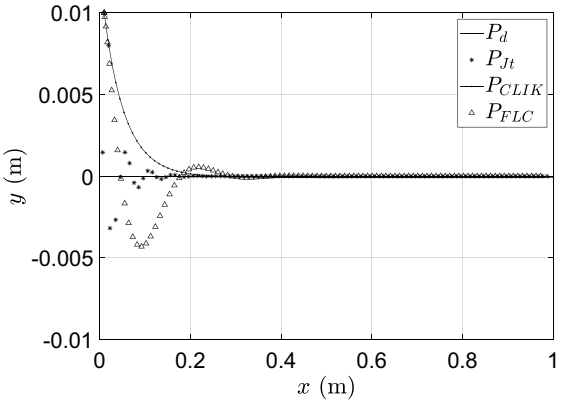

Figure 8. *Straight-line trajectory tracking simulation results with nonzero initial conditions.*

a variety of trajectory tracking approaches, including straight-line, circular, triangular, and Lissajious paths. Furthermore, different types of controllers are employed for comparative analysis to highlight the advantages of the proposed approach. This study used three controllers for comparison purposes: MNS, CLIK, and Jacobian transpose-based controllers. Here, the controller's performance is evaluated using the following parameters: RMSE, maximum tracking error, standard deviation error, and controller effort. The position and orientation of the two-wheeled mobile robot are assumed to be nonzero initially, which are given as $x_0 = 0.01$ m, $y_0 = 0.01$ m, and $\phi_0 = -0.2$ rad.

4.1.1. Straight-line path tracking

Figure 8 presents the simulation outcomes for tracking a straight line using different controllers. In the figure, the desired trajectory represented by P_d, while P_{Jt} is the actual path with Jacobian transpose-based controller, P_{CLIK} is the actual path with CLIK, and P_{FLC} is the actual path with the proposed FLC. The path tracking result for the MNS controller is not displayed in Figure 8 due to a substantial discrepancy between the desired path and the actual path. On the other hand, Table V displays the tracking errors for each controller. The settling time t_s of the Jacobian transpose-based controller is equal to 0.1 s, which is smaller than the other controllers. However, the controller efforts presented in Table IV ($\omega_R = 42.4$ rad/s, $\omega_L = 48.2$ rad/s) are much larger than the other controllers and not within the defined practical limits, which may lead to saturation in a practical scenario. Even though the FLC has the largest settling time of 0.2 s, the controller effort is smaller ($\omega_R = 3.2$ rad/s, $\omega_L = 3.2$ rad/s) and within the practical limits. The tracking performance of the controllers is recorded in Table V. FLC has the lowest RMSE as compared to other controllers. The CLIK controller has the least maximum position error $\max\{e_{xy}\} = 0.0141$ m and the least standard deviation $\sigma\{e_{xy}\} = 0.0024$ m with respect to the desired path.

4.1.2. Circular path tracking

Figure 9 displays the simulation outcomes for a circle path tracking using the proposed and different other controllers. In this figure, the desired trajectory is represented by P_d, while P_{Jt} is the actual path with Jacobian transpose-based controller, P_{CLIK} is the actual path of the CLIK, and P_{FLC} is the actual path with the proposed FLC. The performance of the MNS-based controller is given in Table V. In this case, the MNS RMSE$\{e_{xy}\} = 0.3716$ m, and it is very high as compared to other controller's performance. The circle tracking using a Jacobian transpose-based controller has a shorter settling time (t_s) when compared to the other controllers. However, the controller requires significantly greater effort or higher angular velocities ($\omega_R = 42.4$ rad/s and $\omega_L = 48.2$ rad/s), which could lead to saturation in real-world applications. In circular tracking, even though the FLC's settling time is longer than listed controllers, the controller efforts are still within reasonable bounds at $\omega_R = 3.2$ rad/s and $\omega_L = 2.8$ rad/s. Among the

Table IV. *Controller efforts of various controllers on different paths with nonzero initial conditions.*

Methods	**Path**	**max$\{\omega_R\}$ (rad/s)**	**max$\{\omega_L\}$ (rad/s)**	**$\int\{\lvert\omega\rvert\}$ (rad)**
Jacobian transpose	Straight-line	42.4	48.2	74.5
	Circular	42.4	48.2	196.9
	Triangular	43.2	48.2	129.4
	Lissajous	42.4	48.2	429.4
Closed-loop inverse kinematics	Straight-line	**2.9**	**2.8**	57.17
	Circular	**3.5**	**2.8**	**177.3**
	Triangular	53.2	59.0	111.3
	Lissajous	3.1	3.3	412.0
Proposed fuzzy controller	**Straight-line**	**3.2**	**3.2**	**55.66**
	Circular	**3.5**	**2.8**	**177.3**
	Triangular	**8.9**	**4.6**	**101.9**
	Lissajous	**3.1**	**3.2**	**411.0**

Table V. *Tracking the performance of the different controllers on various paths with nonzero initial conditions.*

Methods	**Path**	**RMS$\{e_{xy}\}$ (m)**	**max$\{e_{xy}\}$ (m)**	**$\sigma\{e_{xy}\}$ (m)**
Minimum norm solution	Straight -line	0.2988	0.1870	0.0560
	Circular	0.3716	0.02168	0.0664
	Triangular	0.2691	0.1384	0.0391
	Lissajous	0.5061	0.4831	0.1387
Jacobian transpose-based controller	Straight-line	0.1188	0.0233	0.0010
	Circular	0.1185	0.0232	0.0005
	Triangular	0.1190	0.0233	0.0013
	Lissajous	0.1184	0.0233	0.0004
Closed-loop inverse kinematics	Straight-line	0.0289	**0.0141**	**0.0024**
	Circular	0.0785	**0.0141**	**0.0006**
	Triangular	0.0475	**0.0141**	**0.0035**
	Lissajous	0.0813	**0.0141**	**0.0005**
FLC without optimization	Straight-line	0.0182	0.0479	0.0076
	Circular	0.0172	0.0604	0.0085
	Triangular	0.0199	0.0748	0.0128
	Lissajous	0.0168	0.0597	0.0058
Proposed optimized FLC	**Straight-line**	**0.0168**	**0.0401**	**0.0065**
	Circular	**0.0149**	**0.0508**	**0.0073**
	Triangular	**0.0163**	**0.0518**	**0.0093**
	Lissajous	**0.0146**	**0.0393**	**0.0045**

listed controllers, FLC has the lowest RMSE value. Regarding the planned path at each instant in time, the CLIK controller has the lowest standard deviation error $\sigma\{e_{xy}\} = 0.0006$ m and the lowest maximum position error $\max\{e_{xy}\} = 0.0141$ m, when compared to the FLC, Jacobian transpose-based controller, and MNS.

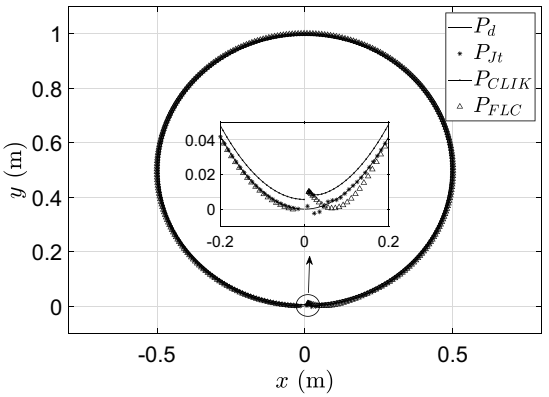

Figure 9. *Circular path trajectory tracking simulation results with nonzero initial conditions.*

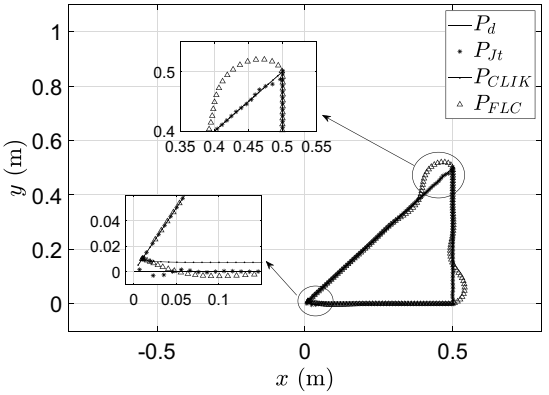

Figure 10. *Triangular path trajectory tracking simulation results with nonzero initial conditions.*

4.1.3. Triangular path tracking

Here, the simulation outcomes for tracking a triangular path using different controllers, namely Jacobian transpose-based controller, CLIK, FLC, and MNS, are explained. The desired path is represented by P_d in the figure, while the actual paths of the proposed FLC, CLIK, and Jacobian transpose-based controller are represented by P_{FLC}, P_{CLIK}, and P_{Jt}, respectively. The simulation results for the Jacobian transpose-based controller, CLIK, and FLC are displayed in Figure 10, while the MNS simulation result is not included because there is a significant discrepancy between the desired and actual paths. However, the performance data of all controllers, including MNS, are presented in Table V. The settling time (t_s) of the Jacobian transpose-based controller is 0.05 s, which is the shortest among all the controllers tested. However, the controller efforts shown in Table IV ($\omega_R = 43.2$ rad/s and $\omega_L = 48.2$ rad/s) are significantly greater than those of the other controllers, which is practically infeasible. FLC has a longer settling time compared to other controllers, but the controller effort is within acceptable limits ($\omega_R = 8.9$ rad/s and $\omega_L = 4.6$ rad/s). The tracking error simulation results are shown in Table V for all controllers, including MNS. FLC has the lowest RMSE of all the controllers, indicating that it performs better in terms of overall tracking accuracy. However, the CLIK controller has the least maximum position error $\max\{e_{xy}\} = 0.0141$ m and the least standard deviation $\sigma\{e_{xy}\} = 0.0035$ m compared to the other controllers for circular path tracking.

4.1.4. Lissajous path tracking

Figure 11 displays the outcomes simulation of the trajectory tracking for the Lissajous path with various controllers. The desired trajectory is represented by P_d in the figure, whereas the actual paths are

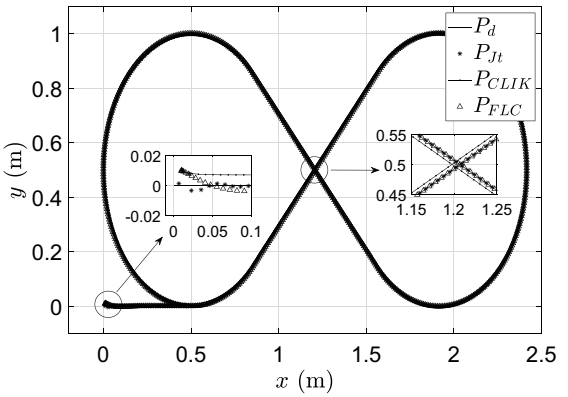

Figure 11. *Lissajous path trajectory tracking simulation results with nonzero initial conditions.*

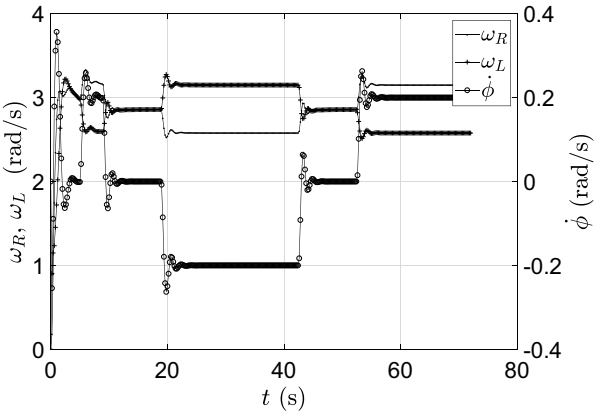

Figure 12. *Controller efforts velocities of the mobile robot while tracking Lissajous path using FLC with nonzero initial conditions.*

represented by P_{Jt}, P_{CLIK}, and P_{FLC} for the Jacobian transpose-based controller, CLIK, and proposed FLC, respectively. Since the performance of the MNS-based controller is poor, the tracking trajectory of the MNS is not included in Figure 11. Table V shows the trajectory tracking errors for all controllers, including MNS. The maximum controller efforts demanded by the Jacobian transpose-based controller are $\omega_R = 43.2$ rad/s and $\omega_L = 48.2$ rad/s, which is not in defined practical limits. On the other hand, FLC has the longest settling time, but its controller effort is within acceptable limits ($\omega_R = 3.1$ rad/s and $\omega_L = 3.2$ rad/s). The CLIK controller has the lowest maximum position error $\max\{e_{xy}\} = 0.0141$ m and lowest standard deviation $\sigma\{e_{xy}\} = 0.0005$ m in relation to the planned path at every time, whereas FLC also has the lowest RMSE of all the controllers. The zoomed view of the trajectory in Figure 11 shows that there is a steady-state error in the case of the CLIK controller.

The controller efforts of the FLC for Lissajous path tracking is presented in Figure 12. It is clear that the angular velocities of the wheels are less than 3.5 rad/s, and the orientation rate (yaw rate) of the mobile robot is also in the range of ± 0.4 rad/sec. The trajectory tracking error of the FLC with respect to time is presented in Figure 13. The position error of the mobile robot in the *x*-*y* plane is within the range of ± 0.025 m. Due to the nonzero initial position, the maximum distance error e_{xy} is equal to 0.025m. The orientation error e_ϕ of Lissajous path tracking for FLC with respect to time is also shown in Figure 13. This figure demonstrates that the error is large at the initial stage because of the initial orientation being nonzero, and after a few seconds, it is reduced to a small value, which is varied within a range of ± 0.05 rad (Figure 14).

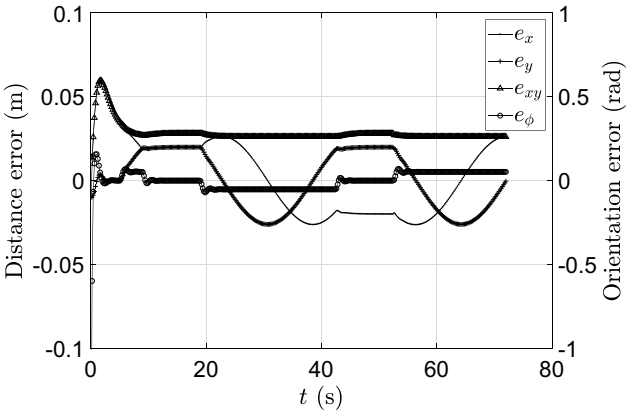

Figure 13. *Position errors of the mobile robot while tracking Lissajous path using FLC with nonzero initial conditions.*

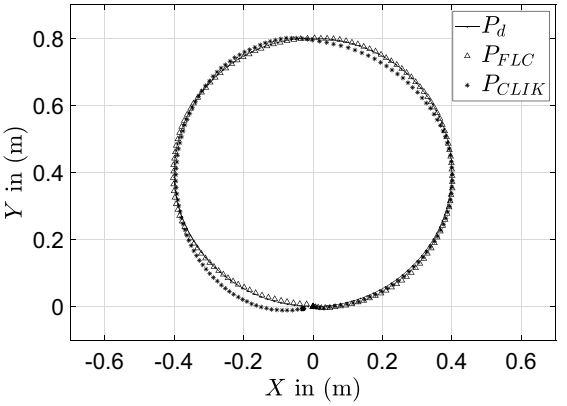

Figure 14. *Experimental results of path tracking on a circular path.*

4.2. Proposed method simulation results considering the noise

As discussed in the previous section, FLC is proposed for the trajectory tracking of the two-wheeled DD mobile robot for this study. FLC is preferable because it can handle the uncertainty and inaccurate data associated with controlling systems compared to other controllers [1, 2, 30]. In particular, this controller shows the capacity to reduce tracking errors while maintaining task efficiency when sensor noise or disturbance is added to the system. Several previous studies have demonstrated that a fuzzy controller performs effectively in nonlinear environments [3–5, 30].

Fuzzy control can also improve the overall robot system's robustness [30]. In order to verify the robustness of the proposed system, an investigation is carried out to examine the controller's actions when noise affects the robot's localization system. The purpose of this study is to evaluate the robustness of controller, particularly with regard to trajectory tracking tasks. Basically, analyzing the controller's response to noise in the localization system-induced errors or uncertainty in the robot's trajectory tracking provides important information on the effectiveness and robustness of the control system. This type of analysis helps to increase the robustness and reliability of the controller in real-world scenarios. Four different types of trajectory tracking paths are used for simulation purposes to verify the proposed controller in this study. To verify the robustness of the proposed controller, randomly generated sensor noise is considered and simulated using various trajectory tracking paths.

Table VI. *The proposed conventional FLC, type 3 FLC, FLC-based SMC, optimized FLC, and FLC in the presence of noise are compared depending on varous criteria.*

Methods	**Path**	**RMS$\{e_{xy}\}$ (m)**	***Max*$\{e_{xy}\}$ (m)**	**$\sigma\{e_{xy}\}$ (m)**
Normal FLC	Straight-line	0.0182	0.0679	0.0078
	Circular	0.0172	0.0604	00.0065
	Triangular	0.0178	0.0546	00.0103
	Lissajous	0.0168	0.0499	0.0058
FLC-based SMC [36]	Straight-line	0.0173	0.0515	0.0067
	Circular	0.0162	0.0521	0.0061
	Triangular	0.0169	0.0597	0.0105
	Lissajous	0.0159	0.0710	0.0109
Type-3 FLC [31]	Straight-line	0.0169	0.0507	0.0064
	Circular	0.01537	0.0513	0.0057
	Triangular	0.0178	0.538	0.0204
	Lissajous	0.0153	0.0399	0.0049
FLC with noise	Straight-line	0.0188	0.0864	0.0124
	Circular	0.0178	0.0748	0.0125
	Triangular	0.0213	0.1196	0.0212
	Lissajous	0.0174	0.0710	0.0109
Optimized FLC	**Straight-line**	**0.0168**	**0.0401**	**0.0065**
	Circular	**0.0149**	**0.0508**	**0.0073**
	Triangular	**0.0163**	**0.0518**	**0.0093**
	Lissajous	**0.0146**	**0.0393**	**0.0045**

4.2.1. Straight-line path trajectory tracking simulation results in presence of noise

The simulation results for straight-line trajectory tracking of the proposed FLC method incorporating presence noise and various controllers are shown in Figure 15. The desired trajectory is denoted by P_d, while Jacobian transpose-based controllers, CLIK, and FLC represented by P_{Jt}, P_{CLIK}, and P_{FLC} are the actual trajectories for each respective controller. This figure shows when 0.01 randomly generated noise is simulated with the proposed system. The controller efforts remain within the expected operating range even the noise is added to system. The comparison of FLCs with and without noise is shown in Table VI. This outcome highlights the FLC controller's stability and resilience, as well as demonstrating its capacity to continue operating even in the case of errors and uncertainties in the system.

4.2.2. Circular path trajectory tracking simulation results in presence of noise

Figure 16 shows the circular trajectory tracking simulation results. This simulation results include the effect of presence noise along with other controllers, including the proposed FLC technique. In this approach, P_d represents the desired trajectory. In the meantime, the Jacobian transpose-based controller, CLIK, and FLC represent the actual trajectories, which are represented by P_{Jt}, P_{CLIK}, and P_{FLC}, respectively.

The figure shows when proposed method is simulated with randomly generated sensor in order to assess the stability and robustness of the proposed approach. The controller's efforts remain within the expected operating range even when noise is introduced. The comparison of robust fuzzy and proposed FLC without noise is presented in Table VI. This demonstrates the FLC controller's robustness and stability as well as its capacity to continue operating even when the system is disturbed by uncertainty and system failures.

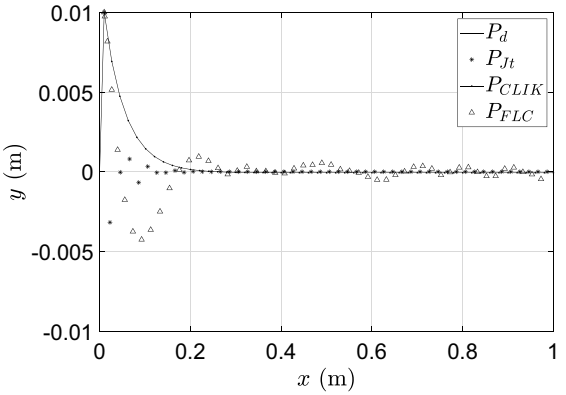

Figure 15. *Straight-line path trajectory tracking simulation results with nonzero initial conditions with presence of noise.*

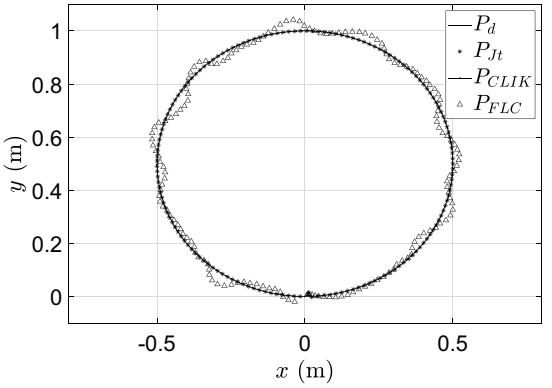

Figure 16. *Circular path trajectory tracking simulation results with nonzero initial conditions with presence of noise.*

4.2.3. Triangular path trajectory tracking simulation results in presence of noise

Figure 17 shows results of the triangular trajectory tracking simulation results in presence of noise. This comprehensive simulation results explores the effects of presence noise and other types controllers. In this figure, the desired trajectory tracking is represented by P_d, while the actual trajectories FLC, CLIK, and Jacobian transpose-based controllers are represented by P_{FLC}, P_{CLIK}, and P_{Jt}, respectively.

The proposed method is simulated with random noise for traingular path tracking as shown in the figure. The controller exhibits remarkable resilience to external disturbances as controller efforts are effectively remained within the specified working range even in the presence of noise. The performance comparison between the robust fuzzy method and the suggested fuzzy methodology is shown in Table VI. This result highlights the FLC controller's effectiveness in practical situations by demonstrating its ability handling nonlinearities and uncertainties while maintaining stability and robustness.

4.2.4. Lissajious path trajectory tracking simulation results in presence of noise

The simulation result of the proposed method with noise for Lissajious path trajectory tracking is shown in Figure 18. This comprehensive simulation study examines the effects of noise presence on the proposed novel optimized FLC method. The desired trajectory tracking is represented by P_d. Concurrently,

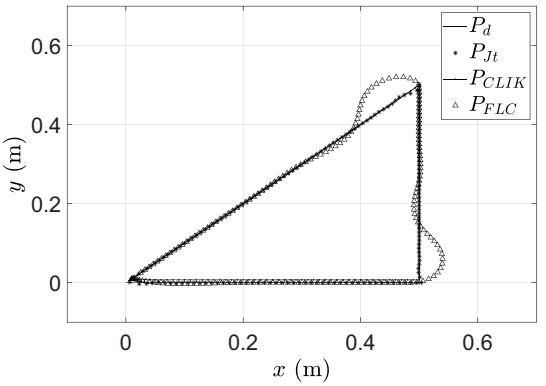

Figure 17. *Triangular path trajectory tracking simulation results with nonzero initial conditions with presence of noise.*

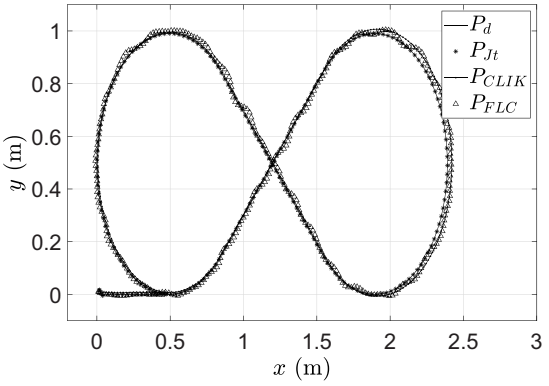

Figure 18. *Lissajious path trajectory tracking simulation results with nonzero initial conditions with presence of noise.*

the actual paths FLC, CLIK, and Jacobian transpose-based controllers are represented by P_{FLC}, P_{CLIK}, and P_{Jt}, respectively.

The controller has exceptional resistance to external disruptions, as it successfully maintains controller efforts within the designated operating range even when noise is present. The robust fuzzy approach and the proposed fuzzy approach are compared in terms of performance in Table VI. This result emphasizes the FLC controller's effectiveness in practical situations by demonstrating its ability to handle uncertainty and nonlinearities while maintaining stability.

Figure 19 shows the RMSE for orientation and position during Lissajious trajectory tracking using the FLC technique with system noise. This result shows the stability and robustness of the proposed FLC in case uncertainties and disturbance by noise. The FLC technique demonstrates its reliability in scenarios with changing variables and disturbances by maintaining a constant performance when compared to the classical controller even when noise is applied to the system.

The tracking of a Lissajious path trajectory while simulating noise are shown in Figure 20. The proposed fuzzy controller maintains the angular velocities of each wheel within the given operating ranges, even when noise is present. This result is evidence of the FLC controller's stability and resilience, demonstrating its effectiveness in unpredictable real-world situations.

Additionally, the constant stability of angular velocities within allowed bounds highlights the reliability of the FLC technique in coordinating accurate and regulated movements of the mobile robot. This ensures improved trajectory tracking by giving the controller's performance trust, especially in dealing with changing external conditions and disturbances.

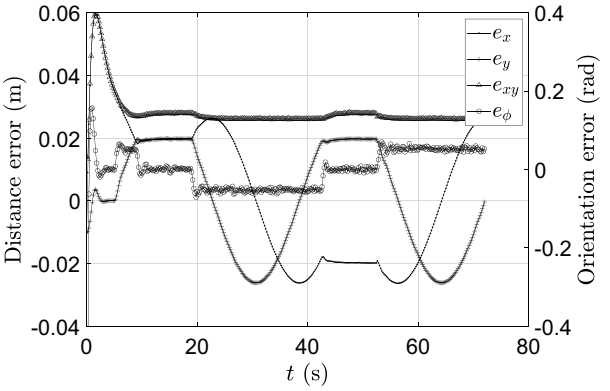

Figure 19. *Position and orientation errors of mobile robot simulation results while following the Lissajious path trajectory tracking with nonzero initial conditions with presence of noise.*

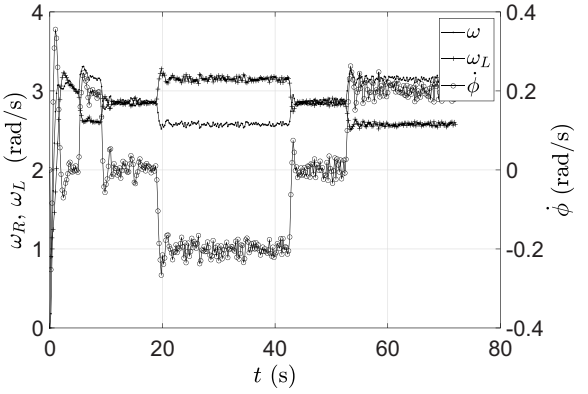

Figure 20. *Angular velocities of mobile robot simulation results while following the Lissajious path trajectory tracking with nonzero initial conditions with presence of noise.*

4.3. Experimental results

According to the simulation findings from Section 4.1, the FLC-based path tracking controller is effective performance than other controllers like MNS, CLIK, and Jacobian transpose-based controllers. Experiments were carried out using circular and Lissajious path tracking trajectories to evaluate the efficiency of the controllers. Figure 14 shows experimental results of the 0.4 m radius circular path using FLC and CLIK. The robot's actual trajectories are represented by P_{FLC} and P_{CLIK}, respectively, for the proposed FLC and CLIK, whereas the desired trajectory is represented by P_d as shown in the figure. The maximum distance errors for the proposed P_{FLC} and P_{CLIK}, $\max\{e_d\}$ is 0.001 m and 0.003 m, respectively. Since FLC tracking error is smaller than CLIK, it can be concluded that FLC controller is capable of tracking the circular path more accurately than CLIK.

Experimental results for the Lissajous path using FLC and CLIK are presented in Figure 21. The desired trajectory is represented by P_d, while the actual trajectory using a FLC is represented by P_{FLC}, and the actual trajectory using CLIK is denoted by P_{CLIK}. The maximum distance error $\max\{e_d\}$ for paths P_{FLC} and P_{CLIK} is 0.005 m and 0.007 m, respectively. The tracking error of FLC is less than that of the CLIK method, demonstrating that the FLC controller can track the complex Lissajous path with greater accuracy than CLIK controller. Figure 22 shows the two-wheeled mobile robot's experimental snapshots while tracking the circular path. The experimental snapshots of the two-wheeled drive robot while tracking the Lissajious path are also presented in Figure 23.

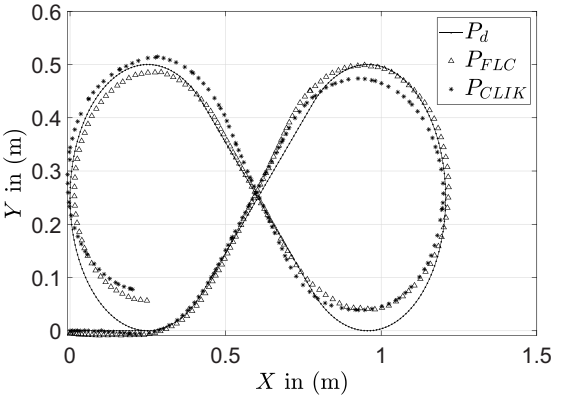

Figure 21. *Experimental results of path tracking on a Lissajous path using proposed controller and CLIK.*

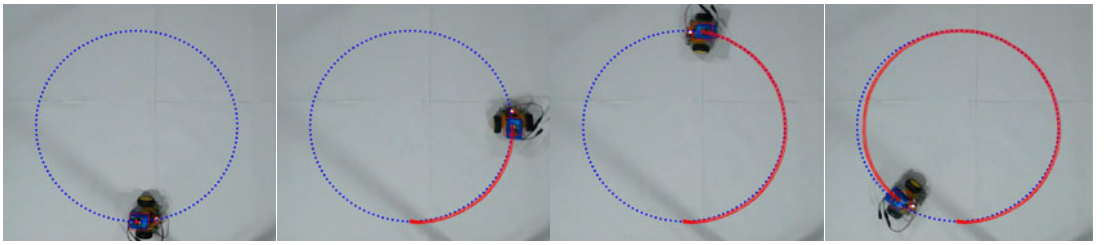

Figure 22. *Experimental snapshots during circular path tracking by the robot.*

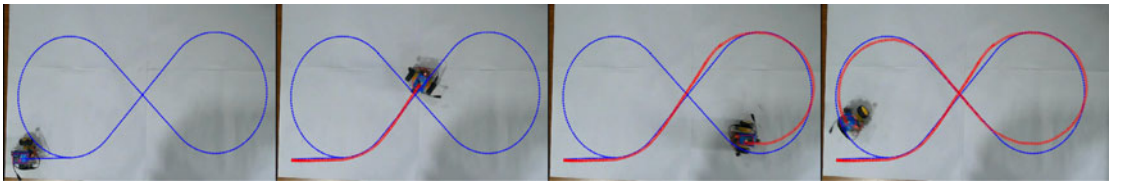

Figure 23. *Experimental snapshots during Lissajous path tracking by the robot.*

4.4. Discussions

Mobile robots consist various technologies, including sensors, actuators, electrical components, information processing, and communication systems [32]. In recent decades, the mobile robots have become more widely used in a variety of applications, including the industrial and service sectors [41]. The increasing application of mobile robots in many engineering fields recent years has led to an increasing awareness among researchers regarding the significance of investigating and improving autonomous vehicle technology. The development of efficient controllers is an essential factor in improving performance mobile robot [1–3, 31].

Among the primary challenges in controlling mobile robots are trajectory tracking and stability [18, 19, 45–47]. The goal of path tracking is to precisely lead the robot along predefined tracks within predetermined times by utilizing geometric parameters [48]. Because of its resilience and adaptability, FLC can be used in a variety of control applications [30].

Fuzzy logic has used in many different fields, including identification of faces, unmanned aerial aircraft, medical diagnostics, and transportation systems. In several industries, including chemical, automotive, and aerospace, fuzzy logic is used to support decision-making, system control, and optimization processes [31, 41, 49].

In recent years, many researchers have used mobile robot trajectory tracking using FLCs [1–3, 19, 28, 31, 35]. However, tuning the membership functions of FLC poses challenges due to its complexity and time-consuming nature [19, 28, 35]. In this study, we have proposed trajectory tracking using FLC specifically designed for two-wheeled DD mobile robots. In order to optimize the membership function parameters and enable effective tuning, we have implemented the PSO algorithm. Here, the proposed method is validated through MATLAB simulation and experimental results. To assess the effectiveness of the FLC further, we have tested different path tracking scenarios, such as circular, triangular, straight-line, and Lissajous paths. Triangular and Lissajous paths are more challenging to track than straight-line and circular paths, which shows the strength and efficiency of the proposed controller compared to other related paths.

Moreover, the proposed approach's resilience is thoroughly assessed by adding randomly generated sensor noise data using MATLAB simulation. This extensive evaluation covers multiple tracking trajectories and reveals the system's resilience to external noise. The results of the simulation demonstrate the effectiveness of the suggested methodology in preserving reliable performance even when there are external factors that affect the controller.

We have also evaluated our FLC's performance by comparing it with three alternative controllers (MNS, CLIK, and Jacobian transpose-based controller) with different trajectory tracking tasks conducted under the same circumstances. Controller effort, maximum tracking error, standard deviation error, and RMSE are some of the critical metrics used for the evaluation. Controller efficiency and trajectory tracking precision of simulation results show that the FLC performs better than the CLIK, MNS, and Jacobian transpose-based controllers. The lowest maximum position error and standard deviation error are shown by the CLIK controller. However, the FLC provides accurate trajectory tracking with reduced RMSE and controller effort. Even though the FLC requires greater computing time than MNS and CLIK, it is more flexible and easy to use in a variety of robotic applications [19, 34]. Conversely, the controller-based Jacobian transpose is limited to specific types of mobile robots and requires extensive computations for inverse kinematics. The trajectory tracking capability of the FLC is different from other controllers because of its capacity to handle uncertainty, interpolation ability, and inference systems [42, 50, 51].

A comparative analysis between conventional FLC, type 3 FLC, FLC-based SMC, FLC with noise, and optimized FLC using PSO-based optimization are presented in Table VI. The comparison is based on simulation results generated by all controllers and assessed by using performance criteria. The presented results demonstrate that the optimized FLC approach produced significant improvements in trajectory tracking performance. Performance parametric evaluations such as RMSE, maximum distance error, and standard distance error are improved using the optimized FLC. These improvements come as a result of the membership function parameter tuning using PSO algorithms, which produces optimized values that are more appropriate for the functioning of the proposed controller. The significant improvements in tracking performance and robustness are also displayed by the type 3 FLC and FLC-based SMC. SMC has the capacity to overcome external disturbances, model uncertainties, and parameter variations, making it one of the most practical techniques. However, the chattering phenomenon affects SMC [52]. Adaptive SMC and optimal SMC with a state-dependent Riccati equation are two methods that have been proposed to mitigate this effect. However, adaptive SMC increases the system response delay, which can make it challenging for wheeled mobile robot (WMR) to accurately track the trajectory [52].

PSO exhibits superior convergence rates compared to other algorithms, attributed to its direct exploration strategy, which proves advantageous in scenarios prioritizing efficiency. Furthermore, PSO's minimal parameter requirements and simplicity make it easier to implement and apply, which appeals especially to users looking for simple optimization solutions. Moreover, PSO's ability to achieve a balance between exploration and exploitation guarantees efficient search space navigation, which adds to its application in a variety of optimization circumstances.

Table VII presents comparative studies of the proposed FLC in comparison to various alternative control strategies that have been documented in the literature. The table shows that the proposed controller performs better than other controllers in terms of tracking error, stability, rate of convergence, and computational complexity.

Table VII. *Comparing the suggested approaches with other available controllers from various literature sources.*

References	Methodology	Position tracking error (m)	Computational complexity	Experimentally verified	Rate of convergence and stability
Antonelli et al. [4]	Finite-time global tracking control method.	0.0452	High	No	Converges and stable.
Zhang et al. [6]	Adaptive fuzzy feedback control method	0.00242	High	No	Converges and stable.
Abdelwahab et al. [42]	Z-number-based FLC	-	High	Yes	Converges and stable.
Abubakr et al. [43]	Adaptive dynamic approach	0.0485	High	Yes	Coverges and stable.
Jang et al. [44]	Multiobjective FLC	0.016	High	Yes	Converges and stable.
Proposed method	**FLC**	**0.0168**	**Relatively low**	**Yes**	**Converges and stable.**

5. Conclusion and future scope

This paper has proposed an optimized FLC for position control and trajectory tracking of the DD mobile robot. The effectiveness of the controller was compared with several control strategies, such as CLIK, MNS, and Jacobian transpose-based controllers, based on the control performance criterion. The performance of each control technique was evaluated based on the controller effort, RMSE, maximum error, and standard deviation error. The results showed that the FLC-based controller outperformed as compared to other techniques in terms of controller effort and RMSE. The simulation and experimental results show the potential of the FLC method for position control and trajectory tracking in mobile robots.

Future studies may examine how to improve obstacle avoidance capabilities using FLCs. In real-world applications, this development would greatly increase the flexibility and resilience of mobile robotic systems. In order to further enhance system performance, future development should take into account combining advanced FLC with SMC.

Author contributions. All authors actively contributed to the design and composition of the article. Tesfaye Deme Tolossa wrote the entire article and improved the results. The entire work has been carried out under the leadership of M. Felix Orlando.

Financial support. No any research funding is included for support in this study.

Competing interests. No any conflicts of interest.

Ethical approval. No any experiment is included relating with human and animal particpants.

References

[1] M. H. Haider, H. Ali, A. A. Khan, H. Zheng, M. U. M. Bhutta, S. Usman, P. Zhi and Z. Wang, "Autonomous Mobile Robot Navigation using Adaptive Neuro Fuzzy Inference System, **In:** *IEEE Inter. Conf. on innovations and Development of Information Technologies and robotics (IDITR)*, (2022) pp. 93–99.

[2] A. Alsharkawi, M. A. Fetyani, E. M. Ijaabo and H. Khasawneh, "Adaptive Neuro-Fuzzy Inference System for a Three-Wheeled Omnidirectional Mobile Robot. **In:** *IEEE International Conference on Applied Engineering (ICAE)*, (2020) pp. 1–6.

[3] X. Alouache and Q. Wu, "Genetic Algorithms for Trajectory Tracking of Mobile Robot Based on PID Controller," **In:** *2018 IEEE 14th Inter. Conference on Intelligent Computer Communication and Processing (ICCP)*, (2020) pp. 237–241.

[4] G. Antonelli, S. Chiaverini and G. Fusco, "Fuzzy-logic-based approach for mobile robot path tracking," *IEEE Trans Fuzzy Syst* **15**(2), 211–221 (2007).

[5] M. H. Ko, B.-S. Ryuh, K. C. Kim, A. Suprem and N. P. Mahalik, "Autonomous greenhouse mobile robot driving strategies from system integration perspective: Review and application," *IEEE/ASME Trans Mech* **20**(4), 1705–1716 (2015).

[6] J. Zhang, Q. Gong and J. Wang, "Finite-time global trajectory tracking control for uncertain wheeled mobile robots," *IEEE Access* **8**, 187808–187813 (2020).

[7] G. S. Lima, V. R. F. Moreira and W. M. Bessa, "Accurate trajectory tracking control with adaptive neural networks for omnidirectional mobile robots subject to unmodeled dynamics," *J Braz Soc Mech Sci* **45**(1), 48 (2023).

[8] A. K. Kashyap, K. P. Lagaza and A. Pandey, "Dynamic Path Planning for Autonomous Mobile Robot using Minimum Fuzzy Rule Based Controller with Avoidance of Moving Obstacles," **In:** *IEEE International Conference on Recent Innovations in Electrical, Electronics and Communication Engineering (ICRIEECE)*, (2018) pp. 3330–3335.

[9] E. Kayacan, "Closed-loop error learning control for uncertain nonlinear systems with experimental validation on a mobile robot," *IEEE/ASME Trans Mech* **24**(5), 2397–2405 (2019).

[10] C.-L. Hwang, "Comparison of path tracking control of a car-like mobile robot with and without motor dynamics," *IEEE/ASME Trans Mech* **21**(4), 1801–1811 (2016).

[11] M. Asai, G. Chen and I. Takami, " Neural Network Trajectory Tracking of Tracked Mobile Robot," **In:** *IEEE 16th Interernational Multi-Conference on Systems, Signals and Devices (SSD)*, (2019) pp. 225–23.

[12] T.-Y. Wang and C.-D. Chang, "Hybrid Fuzzy PID Controller Design for a Mobile Robot," **In:** *IEEE International Conference on Applied System Invention (ICASI)*, (2018) pp.650–653.

[13] T. Y. Chang and C. D. Chang, "Genetic Algorithm Based Parameters Tuning for the Hybrid Intelligent Controller Design for the Manipulation of Mobile Robot," **In:** *IEEE 16th International Conference on Industrial Engineering and Applications (ICIEA)*, (2019) pp. 810–813.

[14] J. Cheng and B. Wang. Flocking Control of Mobile Robots via Simulated Annealing Algorithm," **In:** *IEEE 39th Chinese Control Conf (CCC)*, (2020) pp. 3931–3935.

[15] S. I. A. Meerza, M. Islam and D. M. M. Uzzal, "Q-learning Based Particle Swarm Optimization Algorithm for Optimal Path Planning of Swarm of Mobile Robots," **In:** *IEEE 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, (2019) pp. 1–5.

[16] Y.-H. Chen and C.-D. Chang, "An Intelligent ANFIS Controller Design for a Mobile Robot," **In:** *IEEE Inter. Conference on Applied system Invention (ICASI)*, (2018) pp. 445–448.

[17] S. J.C.L.Barreto, A. G. S. Conceicao, C. E. T. Dorea, L. Martinez and E. R. Pier, "Design and implementation of model-predictive control with friction compensation on an omnidirectional mobile robot," *IEEE/ASME Trans Mech* **19**(2), 467–476 (2014).

[18] N. Tan, Z. Zhu and P. Yu, "Neural-Network-Based Control of Wheeled Mobile Manipulators with Unknown Kinematic Models," **In:** *IEEE International Symposium on Autonomous Systems*, (2021) pp. 212–216.

[19] C. Lakhmissi, "Comparison Between Fuzzy, Neural and Neuro-Fuzzy Controllers for Mobile Robot Path Tracking," **In:** *IEEE, Conference on Control, Engineering and Information Technology*, (2015). PP. 1–6.(CEIT),

[20] E. Kayacan and G. Chowdhary, "Tracking error learning control for precise mobile robot path tracking in outdoor environment," *J Intell Robot Syst* **95**(3-4), 975–976 (2019).

[21] J. Cai, H. Jiang, L. Chen, J. Liu, Y. Cai and J. Wang, "Implementation and development of a trajectory tracking control system for intelligent vehicle," *J Intell Robot Syst* **94**(1), 251–264 (2019).

[22] J.-Y. Zhai and Z.-B. Song, "Adaptive sliding mode trajectory tracking control for wheeled mobile robots," *Int J Control* **92**(10), 2255–2262 (2019).

[23] M. Deremetz, R. Lenain and B. Thuilot, "Path Tracking of a Two-Wheel Steering Mobile Robot: An Accurate and Robust Multi-Model off-Road Steering Strategy," **In:** *IEEE International Conference on Robotics and Automation (ICRA)*, (2018) pp. 3037–3044.

[24] S. Kumar, P. PremKumar, A. Dutta and L. Behera, "Visual motor control of a 7DOF redundant manipulator using redundancy preserving learning network," *Robotica* **28**(6), 795–8102010 (2010).

[25] A. Noormohammadi-Asl, M. Saffari and M. Teshnehlab, "Neural Control of Mobile Robot Motion Based on Feedback Error Learning and Mimetic Structure," **In:** *IEEE 26th Iranian Conference on Electrical Engineering*, 2018) pp. 778–783. (ICEE2018).

[26] N. Singh and B. Panjwani, "Comparison of neural network and fuzzy logic control for nonlinear model of two link rigid manipulator," *Int J Control Autom* **7**(4), 417–428 (2014).

[27] H. Gao, X. Wang and J. Hu, "Adaptive Tracking Control of Mobile Robots Based On Neural Network and Sliding Mode Methods," **In:** *IEEE 38th Youth Academic Annual Conference of Chinese Association of Automation*, (2023) pp. 962–967.(YAC).

[28] P. C. Shill, M. A. H. Akhand, M. D. Asaduzzaman and K. Murase, "Optimization of fuzzy logic controllers with rule base size reduction using genetic algorithms," *Int J Inform Technol Decision Making* **14**(05), 1063–1092 (2015).

[29] S. E. Oltean, M. Dulau and R. Puskas, "Position Control of Robotino Mobile Robot using Fuzzy Logic," **In:** *IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, (2010) pp. 1–6.

[30] T. Zhang, W. Zhang and M. M. Gupta, "Resilient robots: Concept, review, and future directions," *Robotics* **6**(4), 22 (2017).
[31] A. Mohammadzadeh, H. Taghavifar, Y. Zhang and W. Zhang, "A fast non singleton type-3 fuzzy predictive controller for nonholonomic robots under sensor and actuator faults and measurement errors," **In:** *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (IEEE, 2024).
[32] D. K. Tiep, K. Lee, D.-Y. Im, B. Kwak and Y.-J. Ryoo, "Design of fuzzy-PID controller for path tracking of mobile robot with differential drive," *Int J Fuzzy Log Intell Syst* **18**(3), 220–228 (2018).
[33] K. Maatoug, NJAH. M. and M. Jallouli, "IEEE 19th Inter. Conf. On Sciences and Techniques of Automatic Control and Computer Engineering (STA)," **In:** *Electric Wheelchair Trajectory Tracking Control Based On Fuzzy Logic Controller*, (2019) PP. 191–195.
[34] W. U. H.M. and M. Q. Zaman, "LiDAR based trajectory-tracking of an autonomous differential drive mobile robot using fuzzy sliding mode controller," *IEEE Access* **10**, 33713–33722 (2022).
[35] M. I. Solihina, C. Y.Chuana and W. Astutib, "Optimization of fuzzy logic controller parameters using modern metheuristic algorithm for gantry crane system (GCS)," *Mater Today Proceed* **29**, 168–172 (2020).
[36] X. Wu, P. Jin, T. Zou, Z. Qi, H. Xiao and P. Lou, "Backstepping trajectory tracking based on fuzzy sliding mode control for differential mobile robots," *J Intell Robot Syst* **96**(1), 109–121 (2019).
[37] S. Khesrani, A. Hassam, O. Boutalbi and M. Boubezoula, "Motion planning and control of nonholonomic mobile robot using flatness and fuzzy logic concepts," *Int J Dyn Control* **9**(4), 1660–1671 (2021).
[38] A. U. Rehman and C. Cai, "Autonomous Mobile Robot Obstacle Avoidance using Fuzzy-PID Controller in Robot's Varying Dynamics," **In:** *IEEE 39th Chinese Control Conference*, (2020) pp. 2182–2186. (CCC).
[39] D. K. Tiep, K. Lee, D. Im, B. Kwak and Y. J. Ryoo, "Adaptive tracking control of a nonholonomic mobile robot," *IEEE Trans Robot Autom* **16**(5), 609–615 (2000).
[40] B. Lacevic, J. Velagic and N. Osmic, "Design of Fuzzy Logic Based Mobile Robot Position Controller using Genetic Algorithm," **In:** *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, (2007) pp. 1–6.
[41] H. Singh, M. M. Gupta, T. Meitzler, Z.-G. Hou, K. K. Garg, A. M. G. Solo and L. A. Zadeh, "Real-life applications of fuzzy logic," *Adv Fuzzy Syst* **2013**, 1–3 (2013).
[42] M. Abdelwahab, V. Parque, A. M. R. F. Elbab, A. A. Abouelsoud and S. Sugano, "Trajectory tracking of wheeled mobile robots using Z-number based fuzzy logic," *IEEE Access* **8**, 18426–18441 (2020).
[43] O. A. Abubakr, M. A. Jaradat and M. F. Abdelhafez, "Intelligent optimization of adaptive dynamic window approach for mobile robot motion control using fuzzy logic," *IEEE Access* **10**, 119368–119378 (2022).
[44] C.-F. Juang, C.-Y. Chou and C.-T. Lin, "Navigation of a fuzzy-controlled wheeled robot through the combination of expert knowledge and data-driven multi objective evolutionary learning," *IEEE Trans Cybern* **52**(8), 7388–7401 (2022).
[45] K. Shojaei, A. M. Shahri, A. Tarakameh and B. Tabibian, "Adaptive trajectory tracking control of a differential drive wheeled mobile robot," *Robotica* **29**(3), 391–402 (2011).
[46] T. P. Nascimento, C. E. T. Dórea and L. M. G. Gonçalves, "Nonholonomic mobile robots trajectory tracking model predictive control: A survey," *Robotica* **36**(5), 676–696 (2018).
[47] S. Kumar and D. R. Parhi, "Trajectory planning and control of multiple mobile robot using hybrid MKH-fuzzy logic controller," *Robotica* **40**(11), 3952–3975 (2022).
[48] H. Liu and J. Dai, "An approach to carton-folding trajectory planning using dual robotic fingers," *Robot Auton Syst* **42**(1), 47–63 (2003).
[49] D.-N. Bui and M. D. Phung, "Radial basis function neural networks for formation control of unmanned aerial vehicles," *Robotica* **1-19**(6), 1842–1860 (2024).
[50] S. Peng and W. Shi, "Adaptive fuzzy output feedback control of a nonholonomic wheeled mobile robot," *IEEE Access* **6**, 43414–43424 (2018).
[51] L. Cheng, Z.-G. Hou, M. Tan and W. J. Zhang, "Tracking control of a closed-chain five-bar robot with two degrees of freedom by integration of an approximation-based approach and mechanical design," *IEEE Trans Syst Man Cybern Part B (Cyberne)* **42**(5), 1470–1479 (2012).
[52] M. H. Korayem, M. Safarbali and N. Y. Lademakhi, "Adaptive robust control with slipping parameters estimation based on intelligent learning for wheeled mobile robot "," *ISA Trans* **147**, 577–589 (2024).

Cite this article: T. D. Tolossa, M. Gunasekaran, K. Halder, H. K. Verma, S. S. Parswal, N. Jorwal, F. O. Maria Joseph and Y. V. Hote (2024). "Trajectory tracking control of a mobile robot using fuzzy logic controller with optimal parameters", Robotica **42**, 2801–2824. https://doi.org/10.1017/S0263574724001140