International Journal of Power Electronics and Drive System (IJPEDS)
Vol. 17, No. 2, June 2026, pp. 1008∼1024
ISSN: 2088-8694, DOI: 10.11591/ijpeds.v17.i2.pp1008-1024

Hybrid control strategy for trajectory tracking and obstacle avoidance in differential wheeled robots: integrating PSO-NMPC, GA, and fuzzy logic

Abdennour Zeghida[1], Lotfi Farah[2], Halim Merabti[3], Abdelfateh Kerrouche[4]
[1]Laboratory of Automation and Signals Annaba (LASA), Department of Electronics, Faculty of Technology, University of Badji Mokhtar, Annaba, Algeria
[2]Electromechanical Engineering Laboratory, Department of Electromechanics, Faculty of Technology, University of Badji Mokhtar, Annaba, Algeria
[3]Research Center in Industrial Technologies CRTI, Alger, Algeria
[4]School of Computing Engineering and the Built Environment Edinburgh Napier University, Edinburgh, United Kingdom

Article Info

Article history:

Received Aug 8, 2025
Revised Feb 12, 2026
Accepted Feb 21, 2026

Keywords:

Fuzzy logic
Genetic algorithm
Obstacle avoidance
PSO-NMPC
Trajectory tracking
Wheeled robot

ABSTRACT

Mobile robots frequently encounter challenges in maintaining accurate trajectory tracking and effective obstacle avoidance in dynamic and uncertain environments. Traditional control methods, such as proportional integral derivative (PID) and standard MPC, often fail to provide the necessary adaptability and robustness for complex navigation tasks. To overcome these limitations, this study proposes a hybrid control framework for differential-drive wheeled robots that integrates particle swarm optimization–based nonlinear model predictive control (PSO-NMPC), adaptive neuro-fuzzy inference system (ANFIS) optimized by PSO, and genetic algorithm (GA) tuning. The PSO-NMPC computes optimal control inputs in real time while satisfying system constraints to ensure precise trajectory tracking, achieving an average RMSE of 0.0941 m (RMSE$_x$ = 0.0884 m, RMSE$_y$ = 0.0812 m). The ANFIS-PSO controller manages nonlinearities and environmental uncertainties for reliable obstacle avoidance, with an overall RMSE of 0.1084 m (RMSE$_x$ = 0.0761 m, RMSE$_y$ = 0.0772 m). The GA further optimizes key parameters and trajectories, ensuring global path refinement and robust obstacle clearance, achieving an overall RMSE of 0.1094 m (RMSE$_x$ = 0.1059 m, RMSE$_y$ = 0.0274 m). Simulation results in Matlab2024b confirm that the proposed hybrid framework provides precise trajectory tracking, smooth control, and robust obstacle avoidance, making it a promising solution for autonomous mobile robots operating in dynamic and uncertain environments.

Corresponding Author:

Abdennour Zeghida
Laboratory of Automation and Signals Annaba (LASA), Department of Electronics, Faculty of Technology
University of Badji Mokhtar
P.O. Box 12, 23000 Annaba, Algeria
Email: abdennour.zeghida@univ-annaba.dz

1. INTRODUCTION

Over the past few years, efficient control of robot technology has made significant progress in view of the increasing need for intelligent automation in medical, agricultural, military, and industrial applications

[1]. Mobile robots, in contrast to fixed-base manipulators, can be moved and navigate in both structured and unstructured environments. Differential-drive mobile robots (DDMR) are popular as a mobile platform due to their simple mechanical structure, high mobility, and independent actuation of the wheels on both sides for linear motion and angular motion independently [2].

Nevertheless, precise and robust motion control of DDMRs is still an open challenge. Nonlinear kinematics, wheel slippage, actuator saturation, model uncertainties, and dynamic obstacles seriously degrade the tracking performance and safety of navigation. DDMR motion relies on wheel actuators actuated by electric power and regulated using power electronic drive systems, where sharp or cycling control signals introduce uncertainties that waste will increase motor stress, energy waste, strategies, and thermal load. Hence, developing reactive and fast algorithms with smoothness, constraint-awareness, and robustness is a necessary step not only for reactive collision avoidance in robotics but also to make electric drives more energy-constrained or actuators less prone to damage.

Trajectory tracking and obstacle avoidance problems have long been addressed by traditional control strategies, including proportional integral derivative (PID) and linear model predictive control (MPC) [3]–[5]. All of these algorithms work well when the systems are linear and time-invariant, but do not perform well for nonlinear and time-varying scenarios. To alleviate such disadvantages, intelligent and bio-inspired control methodologies are being introduced, such as fuzzy logic (FL), neural networks (NNs), genetic algorithms (GAs), and particle swarm optimization (PSO) techniques [6]–[9]. FL controllers are able to cope with uncertainties but do not have predictive ability. NNs are proficient in modeling nonlinearities but need plenty of data for training. Genetic algorithm (GA) and PSO are the evolutionary algorithms that can meet strong optimization, but it is difficult to directly apply them in real-time since GA and PSO have a large amount of computation overhead [10]–[13].

A number of hybrid control approaches have been developed to integrate the benefits of traditional and intelligent techniques. Neural networks have been used in the trajectory tracking of adaptive systems [14], fuzzy controllers on DDMR control of motion regulation [15], sliding mode control on how to improve stability [16], and preventive controllers for reaching targets [17]. The neural network-based predictors and the improved MPC schemes were also reported for nonholonomic mobile robot control [18]–[20].

One important gap in the literature persists, however. Most of the studies only consider trajectory tracking and obstacle avoidance separately, but not simultaneously, in dynamic or uncertain conditions. In addition, the integration of predictive control along with global optimization and adaptive fuzzy intelligence in a unified manner that is amenable to drive-constrained robotic systems has not been fully investigated.

To solve these problems, the present work develops a hybrid intelligent control technique including PSO-based NMPC, PSO-tuned ANFIS (ANFIS-PSO), and GA tuning. The PC-based PSO-NMPC-based algorithm is utilized to output optimal control laws in the online mode for accurate path tracking. The GA module optimizes the parameters of the controller and path plan in order to improve global performance and reduce computation cost. Through the ANFIS-PSO controller, which can compensate for nonlinear and environmental uncertainties, adaptive obstacle avoidance is achieved.

The novelty of this work lies in the unified multi-layer integration of predictive control, evolutionary optimization, and adaptive fuzzy intelligence, enabling simultaneous improvement in tracking precision, control smoothness, and obstacle avoidance reliability. This integrated strategy reduces abrupt velocity variations, which is beneficial for electric drive efficiency, actuator longevity, and system stability. The main contributions of this study are:

- Development of a novel hybrid control architecture combining PSO-NMPC, ANFIS-PSO, and GA for DDMR systems.
- Real-time optimization of control parameters under system constraints.
- Integration of adaptive fuzzy intelligence for dynamic obstacle avoidance.
- Performance validation demonstrating reduced tracking error, smoother velocity profiles, and enhanced navigation stability compared with conventional NMPC and fuzzy controllers.

The rest of this manuscript is structured as follows: i) Section 2 describes the robot modeling and the proposed control framework; ii) The implementation approach is detailed in section 3; iii) The simulation and experimental results are presented in section 4; and iv) Finally section 5 summarizes and concludes the paper with directions to future work.

2. THE PROPOSED HYBRID CONTROL ALGORITHM

Examine a trajectory tracking control problem in which we derive control laws to manage the angular and linear velocities (and accelerations) of a nonholonomic wheeled mobile robot to properly follow a specified path. The control policy minimizes the root mean square error (RMSE) between the desired and actual trajectories. Tracking mistakes may arise from sensor inaccuracies, external disturbances, system noise, and wheel slip. This nonholonomic limitation prevents the mobile robot from instantaneously moving in a direction perpendicular to the wheel axis, complicating motion control design compared to a holonomic system that allows for unfettered mobility.

The suggested hybrid control system combines various intelligence and optimization strategies to improve tracking accuracy, stability, and robustness in response to these obstacles. The hybrid controller integrates the advantages of nonlinear model predictive control (NMPC), PSO, GA, and fuzzy logic control (FLC) to enhance control performance in diverse dynamic and uncertain environments. The suggested hybrid control technique amalgamates NMPC, PSO, GA, and an FLC to ensure resilient trajectory tracking and obstacle evasion in a differential wheeled robot. Figure 1 illustrates the comprehensive structure of the system [21].

The simplest study of how mechanical systems act is kinematics. To design effective mobile robots for tasks and understand how to develop control software for a specific instance of mobile robot hardware, we need to understand the mechanical behavior of the robot in mobile robotics. The DDMR kinematic modeling notations and essential parameters are summarized in Table 1. Throughout the mathematical formulation and simulation of the robot's motion, these abbreviations specify the fundamental geometric and motion-related variables, such as wheel lengths, radius, and linear or angular velocities.

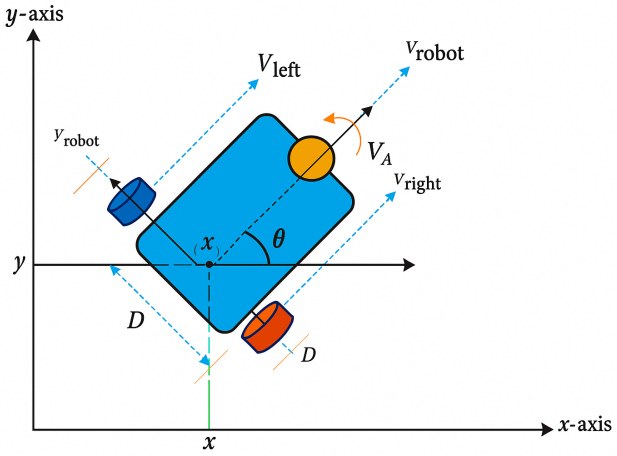


Figure 1. Demonstration of a differential drive mobile robot in the world coordinate frame

Table 1. List of abbreviations for DDMR parameters

Abbreviation	Meaning
m	Center point of the DDMR
D	Distance between the two rear wheels of DDMR (m)
r	Radius of each rear wheel of DDMR (m)
V	Linear velocity of the DDMR (m/s)
ω	Angular velocity of the DDMR (rad/s)
V_{left}	Linear velocity of the left wheel of DDMR (m/s)
V_{right}	Linear velocity of the right wheel of DDMR (m/s)

2.1. Kinematic model of the mobile robot

Kinematics is the mathematical study of motion without taking into account the forces that control motion. Robot movements are described by robot kinematics. It discusses how the system's geometric

relationships work. It establishes a connection between the control parameters, system parameters, and space behavior.

The kinematic model of a two-wheel mobile robot is shown in Figure 1. The DDMR is modeled with two driving wheels on the hub behind the truck, and a castor wheel over the top of the truck. DDMR motion and its direction are realized with the right and left wheel actuators being 2 DC motors. The DDMR motion is characterized by the linear velocities of its two wheels V_{left} and V_{right}. The DDMR linear and angular velocities V and ω of the DDMR are related to V_{left} and V_{right} as given in (1) and (2).

$$V = \frac{(V_L + V_R)}{2} \tag{1}$$

$$\omega = \frac{(V_L - V_R)}{D} \tag{2}$$

A drive with difference any change in the relative velocities of the two wheels will cause DDMR to become sensitive. Different trajectories result from even a slight difference in these velocities. In the global coordinate axis, the DDMR kinematic equations are expressed as (3)–(5) [22].

$$\dot{x}(t) = \frac{(V_R + V_L)}{2} cos\theta(t) \tag{3}$$

$$\dot{y}(t) = \frac{(V_R + V_L)}{2} sin\theta(t) \tag{4}$$

$$\dot{\theta}(t) = \frac{V_R(t) - V_L(t)}{D} \tag{5}$$

2.2. Nonlinear model predictive control

The nonlinear model predictive control (NMPC) strategy is employed to ensure precise trajectory tracking of the DDMR under nonlinear and nonholonomic constraints. The objective is to determine the control inputs V (linear velocity) and ω (angular velocity) that enable the robot to follow a reference trajectory defined by (x_{ref}, y_{ref}), as illustrated in Figure 2 [23].

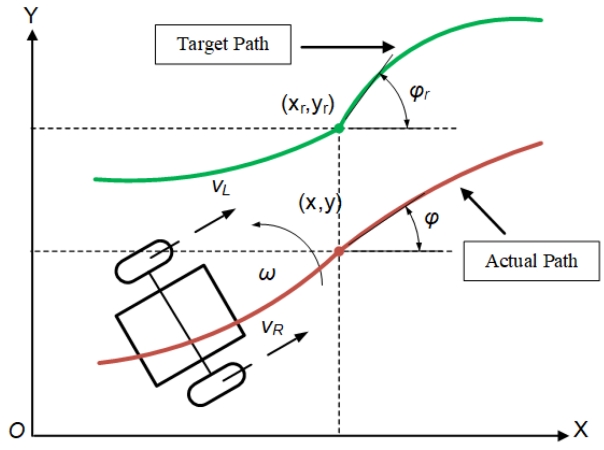


Figure 2. Nonlinear model predictive control structure for the mobile robot

Consider the discrete nonlinear system model, as in (6).

$$x(k+1) = f(x(k), u(k)) \tag{6}$$

Where $x(k)$ is the state vector, $u(k)$ is the control input, and $f(\cdot)$ represents a continuous nonlinear mapping. The control input is constrained by (7).

$$u(k) \in \mathbb{U} \subset \mathbb{R}^m \tag{7}$$

Where $\mathbb{U}$ is a compact convex set satisfying $0 \in \mathbb{U}$ and $f(0,0) = 0$. The state is also constrained to remain within the admissible region, as (8).

$$x(k) \in \mathbb{X} \tag{8}$$

The NMPC optimization problem aims to regulate the robot state toward the reference trajectory by solving a finite-horizon cost minimization problem, as (9).

$$\min_u J_N(x, k, u) \tag{9}$$

Subject to the system dynamics and constraints given in (7).

The cost function is defined as (10).

$$J_N(x, k, u) = F(x(k+N)) + \sum_{i=k}^{k+N-1} L(x(i), u(i)) \tag{10}$$

Where N is the prediction horizon. The terminal state is required to lie within the terminal region, as (11).

$$x(k+N) \in \mathbb{X}_f \subset \mathbb{X} \tag{11}$$

The terminal cost F and terminal region $\mathbb{X}_f$ ensure the closed-loop stability of the NMPC. The optimal control sequence is given by:

$$U = [u(k), u(k+1), \ldots, u(k+N-1)] \in \mathbb{U}^N$$

and only the first control input $u(k)$ is applied at each sampling instant, with the optimization repeated at the next step.

– Discrete-time robot model: Based on the kinematic model of the DDMR, the system dynamics in discrete time are expressed as (12).

$$\begin{cases} x_{k+1} = x_k + T_s V_k \cos(\theta_k) \\ y_{k+1} = y_k + T_s V_k \sin(\theta_k) \\ \theta_{k+1} = \theta_k + T_s \omega_k \end{cases} \tag{12}$$

Where x_k and y_k denote the robot position, θ_k its orientation, V_k the linear velocity, ω_k the angular velocity, and T_s the sampling period.

– System states and control inputs as shown in Table 2:

Table 2. System variables and descriptions

Symbol	Description
$x = [x, y, \theta]^T$	State vector (position and orientation)
$u = [V, \omega]^T$	Control input vector (linear and angular velocities)

– Optimization problem formulation: at each sampling instant k, NMPC solves the finite-horizon optimization problem, as (13).

$$\min_U \; J_N(x_k, U) = \sum_{i=0}^{N-1} \Big[(x_{k+i|k} - x_{ref})^T Q (x_{k+i|k} - x_{ref}) + u_{k+i|k}^T R \, u_{k+i|k} \Big] + (x_{k+N|k} - x_{ref})^T P (x_{k+N|k} - x_{ref}) \tag{13}$$

Subject to (14)–(16).

$$x_{k+i+1|k} = f(x_{k+i|k}, u_{k+i|k}) \tag{14}$$

$$u_{min} \leq u_{k+i|k} \leq u_{max} \tag{15}$$

$$x_{min} \leq x_{k+i|k} \leq x_{max} \tag{16}$$

Where Q, R, and P are positive definite weighting matrices, and N is the prediction horizon.

– Performance metric: the tracking performance is evaluated using the root mean square error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[(x_{ref,i} - x_i)^2 + (y_{ref,i} - y_i)^2]} \tag{17}$$

– Continuous-time model representation: in continuous time, the nonlinear kinematic model of the DDMR is given as (18).

$$\dot{x}(t) = V(t)\cos(\theta(t)), \qquad \dot{y}(t) = V(t)\sin(\theta(t)), \qquad \dot{\theta}(t) = \omega(t) \tag{18}$$

The linear and angular velocities are related to the left and right wheel velocities V_L and V_R by (19).

$$V(t) = \frac{V_R(t) + V_L(t)}{2}, \qquad \omega(t) = \frac{V_R(t) - V_L(t)}{L} \tag{19}$$

Discretizing the above model using a sampling time Δt yields:

$$x_{k+1} = x_k + \Delta t\, V_k \cos(\theta_k), \quad y_{k+1} = y_k + \Delta t\, V_k \sin(\theta_k), \quad \theta_{k+1} = \theta_k + \Delta t\, \omega_k \tag{20}$$

At each sampling instant, the NMPC optimization problem can be formulated as (21).

$$\begin{aligned} \min_U\ J = \sum_{i=0}^{N-1} \Big[& (x_{k+i} - x_{k+i}^{ref})^T Q (x_{k+i} - x_{k+i}^{ref}) \\ & + (u_{k+i} - u_{k+i}^{ref})^T R (u_{k+i} - u_{k+i}^{ref}) \Big] + (x_{k+N} - x_{k+N}^{ref})^T Q_f (x_{k+N} - x_{k+N}^{ref}) \end{aligned} \tag{21}$$

Subject to (22)–(24).

$$x_{k+i+1} = f(x_{k+i}, u_{k+i}) \tag{22}$$

$$u_{\min} \leq u_{k+i} \leq u_{\max} \tag{23}$$

$$x_{\min} \leq x_{k+i} \leq x_{\max} \tag{24}$$

After solving the optimization, only the first control input $u_k^* = [V_k^*, \omega_k^*]^T$ is applied to the robot actuators. The process is then repeated at the next sampling instant using updated state measurements. This receding-horizon strategy enables NMPC to effectively handle nonlinearities, input constraints, and disturbances, achieving smooth and robust trajectory tracking performance.

2.3. Particle swarm optimization

The particle swarm optimization (PSO) was first presented by Kennedy and Eberhart in 1995 [24]. It draws from the collective social behavior in flocks of birds or schools of fish [25]. Each member of the search population (particle) is a possible solution to the optimization problem, and it evolves based on its own experience and that of its neighbours. Every particle has an individual best position pbest i (pbest), which is the fitness value of it found so far. In the global PSO version, gbest is used as the best solution found by all particles. For each particle. Each particle maintains its personal best position, denoted as p_k^i (pbest), which corresponds to the best position it has achieved so far according to a defined fitness function.

In the global version of PSO, the global best position p_k^g (gbest) represents the best solution discovered by any particle in the swarm. The velocity and position of each particle are updated iteratively using (25) and (26).

$$V_{k+1}^i = V_k^i + c_1 r_1 (p_k^i - x_k^i) + c_2 r_2 (p_k^g - x_k^i) \tag{25}$$

$$x_{k+1}^i = x_k^i + V_{k+1}^i \tag{26}$$

Where the parameters are defined in Table 3.

Table 3. Notation used in the PSO algorithm

Symbol	Description
x_k^i	Position of the i^{th} particle at iteration k
V_k^i	Velocity of the i^{th} particle at iteration k
p_k^i	Local best position of the i^{th} particle (*pbest*)
p_k^g	Global best position among all particles (*gbest*)
c_1, c_2	Acceleration coefficients (cognitive and social constants)
r_1, r_2	Random numbers uniformly distributed in [0, 1]

At each iteration, the velocity of each particle is influenced by two main components: the cognitive component, representing the particle's own experience, and the social component, representing the collective knowledge of the swarm. The balance between these two tendencies enables effective exploration and convergence toward the global optimum.

The main steps of the PSO algorithm are as follows: i) Initialize a swarm of N particles with random positions x_i and velocities V_i; ii) Evaluate the fitness of each particle according to the objective function; iii) Update each particle's personal best ($pbest$) and identify the global best ($gbest$); iv) Update particle velocities and positions using (25) and (26); and v) Repeat until the stopping condition (e.g., maximum iterations or minimal error) is reached. Through this mechanism, PSO efficiently balances exploration and exploitation in the search space, making it a robust optimization method for tuning control parameters in nonlinear systems such as the DDMR.

Integration of PSO-NMPC: The combination of PSO with NMPC aims to improve the trajectory tracking performance of the DDMR under dynamic constraints and model uncertainties. The PSO algorithm is utilized to optimize the control inputs and prediction horizon parameters of the NMPC in order to achieve a more accurate and stable motion response.

In the conventional NMPC framework, the control problem is formulated as a nonlinear optimization problem that minimizes a cost function representing the tracking error and control effort. However, due to the high nonlinearity of the robot dynamics and the presence of multiple local minima, the optimization process may not converge to a global optimum. PSO provides a global search capability to enhance the NMPC optimization process by refining the control sequence initialization or tuning the weighting parameters of the cost function.

The cost function to be minimized at each sampling instant is defined as (27).

$$J_N(x,u) = \sum_{i=k}^{k+N-1} \left[(x(i) - x_{ref}(i))^T Q (x(i) - x_{ref}(i)) + u(i)^T R u(i) \right] \tag{27}$$

Where: $x(i)$ represents the predicted robot states (position and orientation); $x_{ref}(i)$ denotes the reference trajectory states; $u(i)$ is the control input vector $[V, \omega]^T$; Q and R are positive definite weighting matrices that balance tracking accuracy and control effort; and N is the prediction horizon length.

In the proposed PSO-NMPC approach, each particle in the PSO swarm represents a candidate control input sequence over the prediction horizon:

$$U = [u(k), u(k+1), \ldots, u(k+N-1)]$$

The fitness of each particle is evaluated using the cost function (27), which reflects the predicted tracking performance over the horizon. The PSO algorithm searches for the optimal U^* that minimizes this cost while satisfying the system constraints defined in the NMPC model.

The integration process proceeds as follows: i) Initialize the swarm with random control input sequences U_i; ii) Evaluate the cost function $J_N(x, u)$ for each particle; iii) Update $pbest$ and $gbest$ based on individual and global fitness; iv) Use $gbest$ as the optimal control sequence to initialize or replace NMPC optimizer at the current time step; v) Apply the first control input $u(k)$ to the DDMR system; and vi) Repeat the process at the next sampling instant with updated states.

This hybrid PSO-NMPC control framework combines the predictive and constraint-handling capability of NMPC with the global optimization strength of PSO. The result is an adaptive and robust controller capable of maintaining accurate trajectory tracking even under disturbances, slippage, and parameter uncertainties in nonlinear DDMR dynamics.

3. DESIGN OF THE FUZZY LOGIC CONTROLLER

The fuzzy logic controller (FLC) is designed to manage both navigation and obstacle avoidance tasks simultaneously for the DDMR. Fuzzy logic control provides a robust and flexible approach for dealing with nonlinear and uncertain robotic environments where precise mathematical modeling is difficult. The fundamental structure of the fuzzy controller consists of three key stages: fuzzification, inference, and defuzzification.

3.1. Determination of input and output variables

In several previous studies, fuzzy logic has been successfully implemented in mobile robot obstacle avoidance and path-tracking systems. For the proposed DDMR system, the input variables are defined as the positional error and orientation error relative to the reference trajectory. These represent the deviation between the robot's current position and the desired reference (x_{ref}, y_{ref}).

The two output variables correspond to the left and right wheel velocities, which determine the robot's motion and turning behavior. The fuzzy logic control system thus generates independent wheel velocity commands that enable the robot to follow the reference path smoothly while avoiding obstacles.

i) Right wheel velocity (V_{right}): The velocity command applied to the right wheel of the robot. This value adjusts dynamically based on the error inputs to guide the robot toward the desired trajectory.
ii) Left wheel velocity (V_{left}): The corresponding command for the left wheel. By adjusting V_{left} and V_{right} independently, the DDMR can perform precise turning and path correction maneuvers.

3.2. Design of the initial fuzzy logic controller

The initial FLC is designed with simplicity and adaptability in mind, employing error-driven inputs and a Sugeno-type inference structure to enable efficient learning and fine-tuning. The hybrid ANFIS–PSO framework later refines these fuzzy parameters to enhance performance in dynamic environments. The proposed fuzzy controller uses two input variables — the reference X_{ref} and Y_{ref} coordinates — representing the robot's horizontal and vertical positional deviations. These serve as the main feedback signals for trajectory tracking.

The membership functions for the input variables (position and orientation errors) are defined using five linguistic terms: very small (VS), small (S), medium (M), large (L), and very large (VL). These fuzzy sets are modeled using overlapping triangular and trapezoidal membership functions, ensuring smooth transitions and robust control performance. The very small term centers around zero, promoting precise fine-tuning near the reference, while the wider sets (L and VL) allow effective correction of large deviations.

$$\text{trimf}(x; a, b, c) = \begin{cases} 0, & x \leq a \text{ or } x \geq c \\ \dfrac{x-a}{b-a}, & a < x \leq b \\ \dfrac{c-x}{c-b}, & b < x < c \end{cases} \tag{28}$$

This fuzzy structure ensures smooth control transitions and stable motion even in the presence of disturbances or modeling uncertainties. Figures 3 and 4 illustrate the input membership functions used for the fuzzy logic controllers of the right and left wheel velocities, respectively.

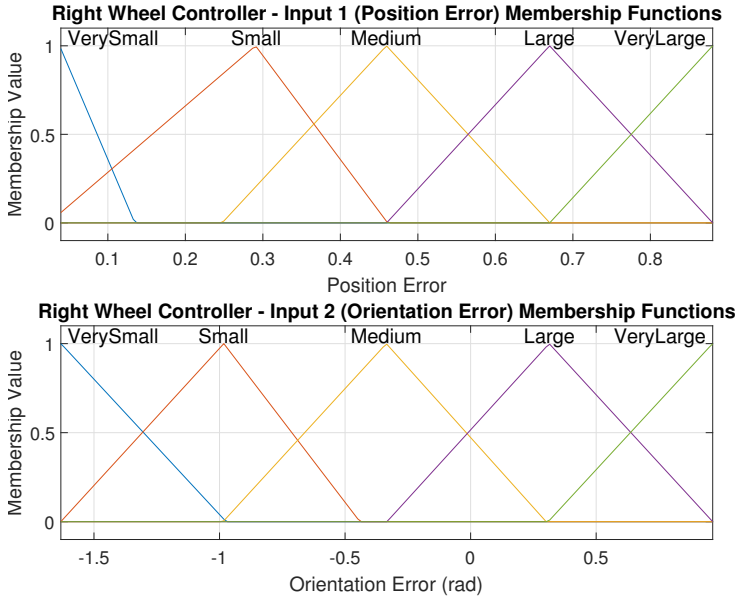


Figure 3. Input membership functions for the right wheel velocity controller

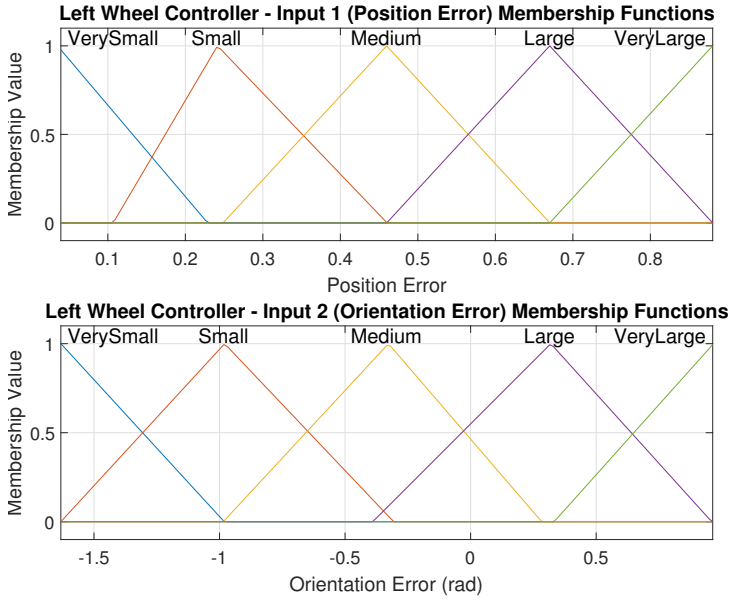


Figure 4. Input membership functions for the left wheel velocity controller

3.3. Fuzzy inference and control surface generation

The fuzzy inference process applies a set of IF–THEN rules that determine the output wheel velocities based on the current positional and orientation errors. The defuzzification step then converts these fuzzy outputs into crisp velocity values for V_L and V_R. Figures 5 and 6 illustrate the ANFIS-generated control surfaces for the left and right wheel velocities, respectively. These surfaces capture the nonlinear relationships between input errors (position and derivative of position) and the output velocity commands.

The surfaces exhibit smooth curvature transitions, reflecting consistent control actions and ensuring stable robot movement. Steeper gradients correspond to stronger corrective actions for large trajectory deviations, while flatter regions indicate finer adjustments near the target path. This fuzzy structure, enhanced through ANFIS and PSO tuning, provides reliable and adaptive control performance for the DDMR under dynamic and uncertain environments.

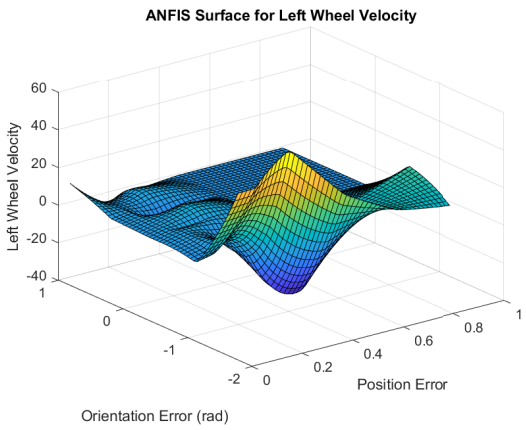


Figure 5. ANFIS control surface for left wheel velocity

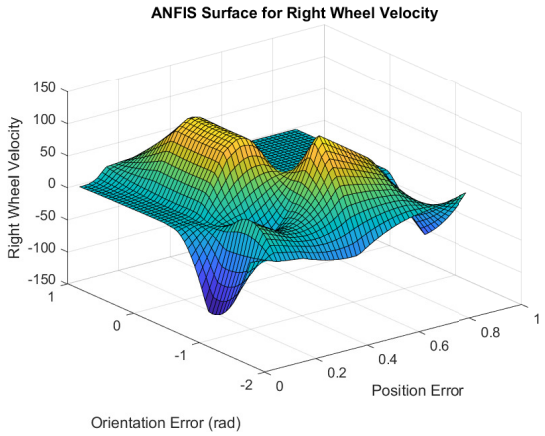


Figure 6. ANFIS control surface for right wheel velocity

3.4. Genetic algorithm for DDMR path planning and optimization

GA is an evolutionary optimization method inspired by natural selection and genetics. GA is particularly useful for global search in complex, multimodal spaces and has been widely used for path planning, parameter tuning, and trajectory optimization for mobile robots. In this work, GA is applied to i) generate collision-free paths for the DDMR and ii) tune discrete NMPC/PSO parameters when required.

– Chromosome representation: a chromosome encodes a path from start to goal using two common methods:
  i) Waypoint sequence: A chromosome is a vector of waypoints $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. Each waypoint can be represented by a fixed-length binary or real-valued gene.
  ii) Control sequence: A chromosome encodes a sequence of control inputs over a horizon $U = [u_0, u_1, \ldots, u_{H-1}]$ where $u_i = [V_i, \omega_i]$. This is convenient when integrating directly with NMPC.
– Fitness function: the fitness function must reflect path quality and feasibility. A typical fitness F to *maximize* (or cost to minimize) is:

$$F = -\Big(w_1 \cdot \text{Length} + w_2 \cdot \text{ClearancePenalty} + w_3 \cdot \text{Smoothness} + w_4 \cdot \text{Time}\Big)$$

  Where: Length $= \sum_{i=2}^{n} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$; ClearancePenalty is a large penalty if any segment intersects obstacles (or inversely proportional to the minimal obstacle distance); Smoothness $= \sum |\Delta\theta_i|$ penalizes sharp turns (helps respect nonholonomic constraints); Time estimates traversal time given wheel limits. Weights w_i are chosen according to priority (e.g., safety w_2 large).
– Constraints and repair: ensure chromosomes satisfy:
  i) Workspace bounds: $x_{\min} \le x \le x_{\max}, y_{\min} \le y \le y_{\max}$.
  ii) Collision-free: Waypoints and connecting segments must avoid obstacles.
  iii) Kinematic feasibility: Curvature and turning radius constraints consistent with DDMR dynamics. Optionally repair infeasible paths by smoothing or local replanning.
– GA parameters (recommended starting values), as shown in Table 4.

Table 4. Suggested GA parameters for DDMR path planning

Parameter	Suggested value
Population size	30–150
Crossover probability	0.7–0.9

– Integration with NMPC and PSO
  i) Path → NMPC: GA produces a sequence of waypoints. NMPC then tracks the smoothed path by generating wheel commands while respecting kinematic constraints and obstacles using predictive constraints.

ii) Parameter tuning: GA can be used offline to tune NMPC cost weights Q, R or PSO hyperparameters (population and inertia) by optimizing long-term metrics (RMSE).
iii) Hybrid online usage: Use GA for occasional re-planning when environment changes drastically; NMPC/PSO handle short-term tracking and local optimization.

– Algorithm 1 presents the GA used for trajectory optimization. A population of candidate solutions (waypoints or control inputs) is iteratively improved while enforcing feasibility through a repair mechanism. Each solution is evaluated using a fitness function, and genetic operators (selection, crossover, and mutation) are applied to explore the search space, with elitism preserving the best individuals. The final solution provides a collision-free and smooth trajectory used as a reference for the robot controller.

Algorithm 1 Pseudocode

```
Initialize population P with random chromosomes (waypoints or controls)
for gen = 1 : max_gen do
  for each individual i in P do
    repair chromosome i to satisfy bounds/kinematics
    simulate path or predict trajectory
    compute fitness F(i)
  end for
  select parents from P (tournament/roulette)
  perform crossover to create offspring
  apply mutation to offspring
  evaluate offspring fitness (with repair if needed)
  form new population using elitism
  if stopping criterion met break
end for
return best chromosome → smooth and convert to reference trajectory
```

4. RESULTS AND DISCUSSION

In this section, simulation results and performance of integrated hybrid control design approach where PSO–NMPC–GA–FLC will be demonstrated for DDMR. Furthermore, a model of ANFIS is utilized to improve the prediction and adaptive adjustment of velocity. All simulations were realized using a MATLAB/Simulink platform to demonstrate the performance of the system in trajectory tracking and obstacle avoidance.

4.1. Simulation setup

The DDMR model dimensions and kinematics parameters, such as wheel radius, axle width, and kinematic constraints, were set in the mindset of typical mobile robots. The simulated workspace contained a number of dynamic obstacles, and the desired paths were predefined reference trajectories $x_{ref}(t)$, $y_{ref}(t)$. The controller parameters of the NMPC–PSO were tuned for best real time tracking difference and response dynamics during obstacle avoidance operations. The main configuration parameters are as follows:

– Prediction horizon: $N_p = 5$ time steps
– Weighting factors: $r_1 = r_2 = 1$, $c_1 = c_2 = 1$
– Dynamic obstacle modeling enabled

4.2. Trajectory tracking and velocity analysis

The simulation results show that the NMPC–PSO controller can achieve accurate trajectory tracking, as well as stable movement even in a complex dynamic environment. The predictive control methods allow for the prediction of future states of the robot and real-time determination of corrective actions. As depicted in Figure 7, the actual trajectory is successfully tracked with little lateral deviation from the required path, confirming that non-linear kinematics and external disturbances are well treated by the controller.

The PSO optimized NMPC parameters improve the convergence and tracking error compared to traditional NMPC configurations. The wheel velocity responses (Figure 8) are remarkable smooth and continuous with no sharp transitions or oscillations. This behavior validates that control inputs are dynamically feasible and realizable on actual hardware, guaranteeing mechanical stability and actuator protection.

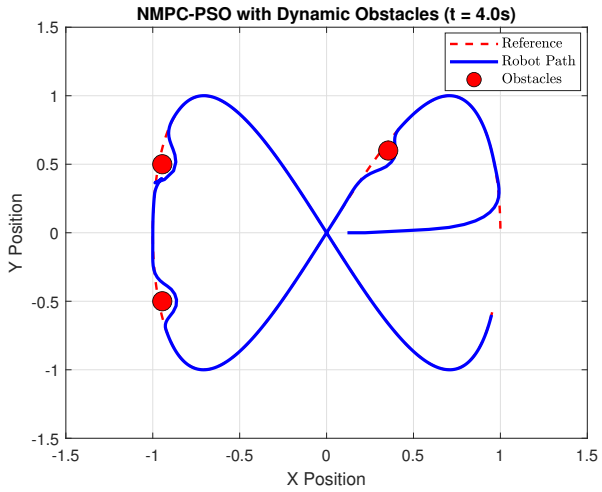


Figure 7. Trajectory tracking performance: reference vs. actual path with dynamic obstacle avoidance

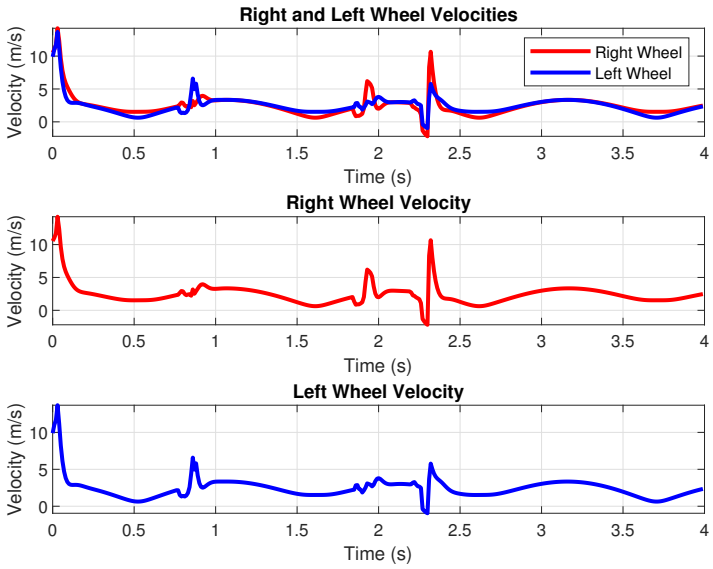


Figure 8. Wheel velocity profiles generated by the NMPC–PSO controller

4.3. Tracking error and RMSE evaluation

The tracking error, defined as:

$$e(t) = \sqrt{(x - x_{ref})^2 + (y - y_{ref})^2}$$

was evaluated throughout the simulation, the average positional error stayed under a low threshold, however. This approach ensured high tracking accuracy and stable motion performance. The DDMR was able to avoid all moving obstructions and the distance and path. The RMSE metric was used to assess trajectory accuracy for both x and y coordinates. Table 5 summarizes the results, showing that the hybrid NMPC–PSO approach significantly outperforms the conventional NMPC and PSO-only methods.

Table 5. RMSE comparison for trajectory tracking performance

Controller type	RMSE (x-axis) [m]	RMSE (y-axis) [m]	Average RMSE [m]
Proposed NMPC–PSO	0.088440	0.081192	0.094061

4.4. ANFIS-PSO performance

Figure 9 shows the path-planning of the two-wheeled differential-drive robot with dynamic obstacles. The robot can move in a collisionless manner, which shows that the proposed ANFIS-PSO control structure is efficient. The right- and left-wheel velocity profiles in Figure 10, we compare actual measured velocities with ANFIS-predicted ones in Figure 9. The estimated velocities (red curve) smoothly fit the actual measurements (blue line), ensuring that the ANFIS model can capture the nonlinear characteristics of wheel motion.

The instantaneous prediction errors are shown in Figure 11. The most serious deviations are observed when the robot suddenly accelerates to high speeds, or if it changes its running direction to avoid obstacles. Overall, the position-tracking accuracy is good: The final RMSE in x-direction is 0.076061, and that in y-direction is 0.077182; the total RMSE of position is 0.108362. The average position error is 0.088718 m) and the standard deviation is 0.062299. The highest single wave error eigenvalue is 0.880399, the lowest single wave error eigenvalue is 0.039364 and at least 90% of the positional errors are less than 0.120803. These results demonstrate that the ANFIS-PSO framework achieves robust and accurate wheel-velocity prediction as well as reliable trajectory tracking, even under dynamic and complex maneuvering conditions.

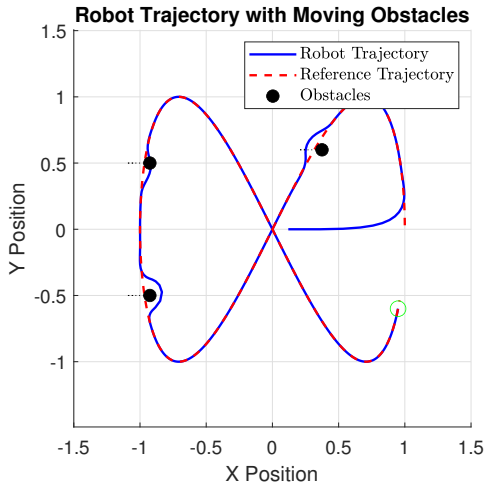


Figure 9. Robot trajectory in the presence of dynamic obstacles

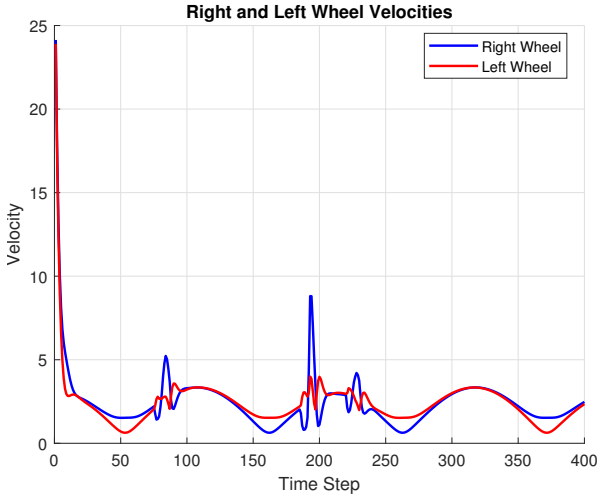


Figure 10. Comparison of actual and ANFIS-predicted wheel velocities

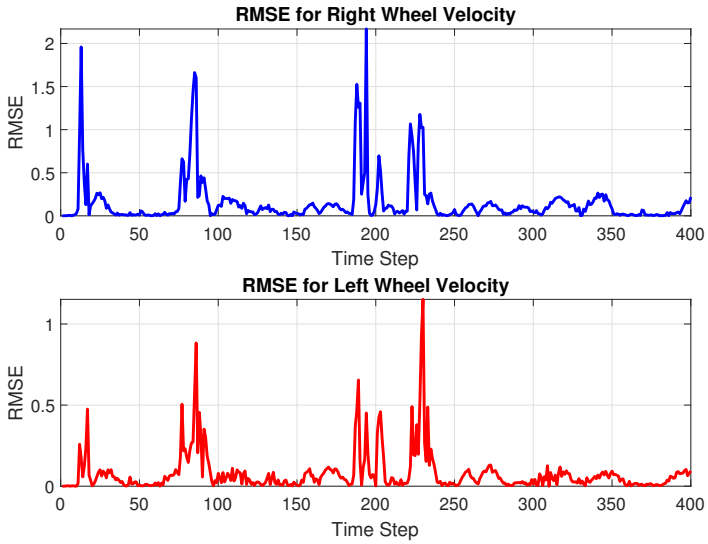


Figure 11. RMSE of wheel-velocity prediction by the ANFIS model

4.5. Genetic algorithm path planning

In this part, we implement a robot path planning strategy using a GA to enable a differential-drive robot to navigate around dynamic obstacles while tracking a predefined reference trajectory. The resulting robot navigation behavior is illustrated in Figure 12. GA can produce feasible collision-free trajectories even with dynamic obstacles, this result strengthens the appropriateness of evolutionary methodology for reactive path planning. The fitness values gradually decrease during the optimization, which suggests that the convergence is stable. This demonstrates that the GA can continuously evolve candidate paths toward lower tracking error, smoother control inputs, and greater obstacle clearance, as shown in Figure 13.

Figure 13 show the corresponding wheel velocity profiles during the GA-driven navigation experiment. This figure also demonstrating how actions can be obtained from the algorithm that generate valid left- and right-wheel commands within actuator constraints. The tracking performance of the GA based planner is also evaluated by RMSE metrics listed in Table 6.

The total position RMSE is 0.1094 m, and it concludes RMPM is able to keep the robot following a reference trajectory quite close with respect to dynamic obstacles. The RMSE values in each axis dimension demonstrate the material variation between axis dimensions, particularly that there are significant trails on DARTEM trajectories (x-axis 0.1059 m), but remain small for the y-axis direction of travel (0.0274 m) indicating most trajectory variations avoid motion. The last tracking error 0.0244 m ensures that the robot' reaches the reference path again after evasions. Even if a huge instantaneous error of 0.8804 m is observed — usually at high speed navigation to avoid obstacles , the medium mean error (0.0411 m) and standard deviation (0.1015 m) show that the GA will provide stable acceptable tracking overall performance.

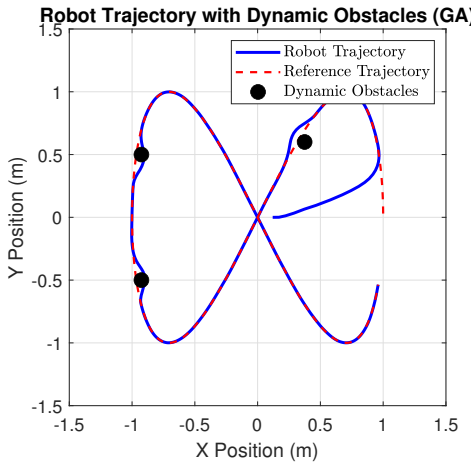


Figure 12. Robot navigation using the GA-based planner while avoiding dynamic obstacles

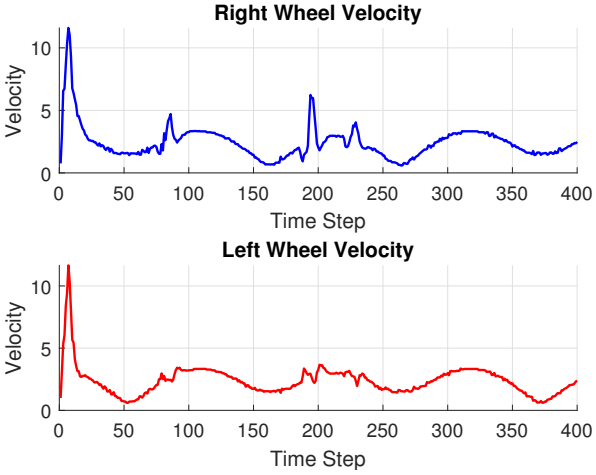


Figure 13. Left and right wheel velocity profiles generated during GA-based path planning

Table 6. Tracking performance metrics for GA-based path planning

Metric	Value	Units
RMSE (x-axis)	0.1059	m
RMSE (y-axis)	0.0274	m
RMSE total (average)	0.1094	m
Final tracking error	0.0244	m

4.6. Comparative performance evaluation

In order to compare the three approaches (NMPC–PSO, ANFIS–PSO, and GA) into an unified metric evaluation the trajectory-tracking accuracy and error metrics are summarized in this section. The analysis is based on both total RMSE, individual RMSE in the four directions, mean error tracking (MET), standard deviation and maximum instantaneous deviation. A summary of the four performances is shown in Table 7.

As indicated in Table 7, the NMPC–PSO obtains the best overall RMSE indicating better trajectory-tracking performance. The ANFIS–PSO model exhibits excellent side-slip performance with

intelligent velocity predictor and is robust to fast maneuvers. The expected total RMSE is a little higher for the GA-based planner, its mean error is acceptable and zero at final states returns to the reference path with a low final tracking error which presents confident global path-planning ability under dynamic obstacles.

Results show that the NMPC–PSO hybrid controller can provide optimal trade-off between tracking accuracy, control smoothness and computational burden. The lowest RMSE for the ANFIS–PSO model implies its excellent learning capability but with a high computational cost in training and testing. Meanwhile, the GA-based planner also achieves strong ability in obstacle avoidance and global path finding performance, it has a lower convergence rate and lager steady-state error than these of NMPC–PSO approach.

In general, the NMPC–PSO controller exhibits better versatility and real-time capability and is capable of coping with dynamic and uncertain environments. The combination of ANFIS for velocity prediction and GA to be used for global path refinement also improves system robustness and accuracy, supporting that hybrid DDMR navigation using intelligent methods is an effective scheme for the value-added operation.

Table 7. Comparison of tracking performance across NMPC–PSO, ANFIS–PSO, and GA

Method	RMSE (x)	RMSE (y)	Total RMSE	Mean error	Max error
NMPC–PSO	0.0884	0.0812	0.0941	–	–
ANFIS–PSO	0.0761	0.0772	0.1084	0.0887	0.8804
GA	0.1059	0.0274	0.1094	0.0411	0.8804

5. CONCLUSION

We can conclude that the proposed hybrid control system comprising GA and two ANFIS-PSOs and NMPC-PSO has generated an efficient and reliable performance for DD mobile robot navigation. The NMPC-PSO controller from simulation generated the smallest total trajectory-tracking error with an average RMSE of 0.0941 m (RMSE$_x$ = 0.0884 m, RMSE$_y$ = 0.0812 m), guaranteeing a smooth control and obstacle avoidance while running in real time. Both the RMSE of prediction and dealing with nonlinear dynamics were (0.1084 m (RMSE$_x$ = 0.0761 m, RMSE$_y$ = 0.0772 m)) in the velocity control and learning (ANFIS-PSO) part across all experiments, giving a precise estimate on wheel velocities. The GA-based various path planner showed high-quality global path optimization and dynamic obstacle avoidance, but a higher tracking error (overall RMSE = 0.1094 m (RMSE$_x$ = 0.1059 m, RMSE$_y$ = 0.0274 m)). In summary, the hybrid approach can effectively utilize the merits of these two tools and provides an efficient and accurate method that can be successfully employed in various unknown dynamic environments for intelligent mobile robot navigation.

FUNDING INFORMATION

This research has not received any specific grant from funding agencies in the public, commercial, or not for profit sectors.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Abdennour Zeghida	✓	✓	✓	✓		✓			✓	✓	✓		✓	
Lotfi Farah	✓	✓	✓	✓		✓		✓	✓		✓	✓	✓	
Halim Merabti	✓	✓	✓	✓		✓	✓	✓	✓		✓	✓	✓	
Abdelfateh Kerrouche	✓	✓		✓		✓			✓		✓		✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest with respect to the research, authorship, and publication of the article.

DATA AVAILABILITY

The authors confirm that the data supporting the findings of this study are available upon request from the corresponding author.

REFERENCES

[1] D. Jovcic, "Series LC DC circuit breaker," *High Voltage*, vol. 4, no. 2, pp. 130–137, Jun. 2019, doi: 10.1049/hve.2019.0003.

[2] M. Crenganis, C. Biris, and C. Girjob, "Mechatronic design of a four-wheel drive mobile robot and differential steering," *MATEC Web of Conferences*, vol. 343, p. 08003, 2021, doi: 10.1051/matecconf/202134308003.

[3] S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous Robots*, vol. 13, no. 3, pp. 207–222, 2002, doi: 10.1023/A:1020564024509.

[4] J. Li, J. Sun, L. Liu, and J. Xu, "Model predictive control for the tracking of autonomous mobile robot combined with a local path planning," *Measurement and Control (United Kingdom)*, vol. 54, no. 9–10, pp. 1319–1325, 2021, doi: 10.1177/00202940211043070.

[5] B. D. Hirpo and W. Zhongmin, "Design and control for differential drive mobile robot," *International Journal of Engineering Research and Technology*, vol. 6, no. 10, pp. 327–334, 2017, [Online]. Available: www.ijert.org.

[6] A. D. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, "A decentralized privacy-preserving healthcare blockchain for IoT," *Sensors*, vol. 19, no. 2, p. 326, Jan. 2019, doi: 10.3390/s19020326.

[7] Z. Sun, S. Hu, H. Xie, H. Li, J. Zheng, and B. Chen, "Fuzzy adaptive recursive terminal sliding mode control for an agricultural omnidirectional mobile robot," *Computers and Electrical Engineering*, vol. 105, p. 108529, Jan. 2023, doi: 10.1016/j.compeleceng.2022.108529.

[8] M. Ghiasi *et al.*, "Resiliency/cost-based optimal design of distribution network to maintain power system stability against physical attacks: a practical study case," *IEEE Access*, vol. 9, pp. 43862–43875, 2021, doi: 10.1109/ACCESS.2021.3066419.

[9] Q. Liu, Y. G. Lu, and C. X. Xie, "Optimal genetic fuzzy obstacle avoidance controller of autonomous mobile robot based on ultrasonic sensors," *2006 IEEE International Conference on Robotics and Biomimetics*, ROBIO 2006, pp. 125–129, 2006, doi: 10.1109/ROBIO.2006.340327.

[10] H. Wang and X. P. Liu, "Adaptive shared control for a novel mobile assistive robot," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 6, pp. 1725–1736, 2014, doi: 10.1109/TMECH.2014.2299213.

[11] Y. D. Zhao, H. W. Ha, D. E. Kim, S. I. Han, and J. M. Lee, "Adaptive neural network based PI sliding mode control of nonholonomic mobile robot," *2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence, URAI 2015*, pp. 473–478, 2015, doi: 10.1109/URAI.2015.7358807.

[12] P. K. Padhy, T. Sasaki, S. Nakamura, and H. Hashimoto, "Modeling and position control of mobile robot," *International Workshop on Advanced Motion Control, AMC*, pp. 100–105, 2010, doi: 10.1109/AMC.2010.5464018.

[13] W. He, S. S. Ge, Y. Li, E. Chew, and Y. S. Ng, "Neural network control of a rehabilitation robot by state and output feedback," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 80, no. 1, pp. 15–31, 2015, doi: 10.1007/s10846-014-0150-6.

[14] B. S. Park, S. J. Yoo, J. B. Park, and Y. H. Choi, "Adaptive neural sliding mode control of nonholonomic wheeled mobile robots with model uncertainty," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 1, pp. 207–214, 2009, doi: 10.1109/TCST.2008.922584.

[15] F. M. Raimondi and M. Melluso, "A new fuzzy robust dynamic controller for autonomous vehicles with nonholonomic constraints," *Robotics and Autonomous Systems*, vol. 52, no. 2–3, pp. 115–131, 2005, doi: 10.1016/j.robot.2005.04.006.

[16] Y. Pan, W. Wei, and K. Furuta, "Hybrid sliding sector control for a wheeled mobile robot," *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 222, no. 8, pp. 829–837, 2008, doi: 10.1243/09596518JSCE529.

[17] D. Vidhyaprakash, S. Karthikeyan, M. Periyasamy, K. Kalaimurugan, and S. Navaneethasanthakumar, "Positioning of two-wheeled mobile robot to control wheelslip by using the wheel rotate planning technique," *Journal of Scientific and Industrial Research*, vol. 78, no. 12, pp. 879–884, 2019.

[18] D. Gu and H. Hu, "Neural predictive control for a car-like mobile robot," *Robotics and Autonomous Systems*, vol. 39, no. 2, pp. 73–86, 2002, doi: 10.1016/S0921-8890(02)00172-0.

[19] Z. Li, J. Deng, R. Lu, Y. Xu, J. Bai, and C.-Y. Su, "Trajectory-tracking control of mobile robot systems incorporating neural-dynamic optimized model predictive approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 6, pp. 740–749, Jun. 2016, doi: 10.1109/TSMC.2015.2465352.

[20] J. G. Ortega and E. F. Camacho, "Mobile robot navigation in a partially structured static environment, using neural predictive control," *Control Engineering Practice*, vol. 4, no. 12, pp. 1669–1679, 1996, doi: 10.1016/S0967-0661(96)00184-0.

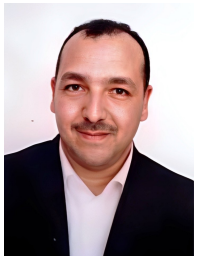
[21] H. Merabti, K. Belarbi, and B. Bouchemal, "Nonlinear predictive control of a mobile robot: a solution using metaheuristcs," *Journal of the Chinese Institute of Engineers*, vol. 39, no. 3, pp. 282–290, Apr. 2016, doi: 10.1080/02533839.2015.1091276.

[22] H. Merabti, B. Bouchemal, K. Belarbi, D. Boucherma, and A. Amouri, "Non linear predictive control of a LEGO mobile robot," *AIP Conference Proceedings*, vol. 1618, pp. 647–650, 2014, doi: 10.1063/1.4897824.

[23] J. Zhang, D. Wei, R. Gao, and Z. Xia, "A trajectory tracking and obstacle avoidance approach for nonholonomic mobile robots based on model predictive control," in *2020 IEEE 16th International Conference on Control and Automation (ICCA)*, Oct. 2020, pp. 1038–1043. doi: 10.1109/ICCA51439.2020.9264573.
[24] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, 2017, vol. 4, pp. 1942–1948. doi: 10.1109/ICNN.1995.488968.
[25] Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, 2001, vol. 1, pp. 81–86. doi: 10.1109/CEC.2001.934374.

BIOGRAPHIES OF AUTHORS

Abdennour Zeghida is a Ph.D. in Embedded Systems at the University of Badji Mokhtar–Annaba, Algeria, where he has been pursuing his doctoral studies since 2019. He received the Master's degree in Automation and Network Computing from the same university. He currently serves as Head of Automation, Mechatronics, and Robotics at East Lighting. He also works as an engineer at Smart Control and Automation and holds a Siemens TIA Portal programming certification, reflecting his expertise in industrial automation and control systems. His research interests include optimization techniques, advanced control systems, artificial intelligence, and robotic applications. He can be contacted at email: abdennour.zeghida@univ-annaba.dz.



Lotfi Farah received the B.Eng. and Ph.D. degrees in Arabic handwritten recognition from the University of Badji Mokhtar Annaba, Algeria, in 1995 and 2000, respectively. From 2000 to 2012, he was a Research Associate at the University of Chérif Messadia, Algeria. He is currently a researcher at Badji Mokhtar University, Annaba, within the Génie Électromécanique Laboratory. His research interests include artificial intelligence, fuzzy systems, neural networks, photovoltaic modeling and control, energy conversion, and power electronics. He has authored and co-authored several conference and journal papers and serves as a reviewer for international journals in artificial intelligence, including environmental progress and sustainable energy (Wiley) and computer science (USA). He can be contacted at email: lotfi.farah@univ-annaba.dz.

Halim Merabti received the Engineer degree in Electrical Engineering from Constantine University, Constantine, Algeria. He obtained the Ph.D. degree in Electronics in 2015 from the same university. He is currently a senior researcher (Grade A) at the Research Center in Industrial Technologies (CRTI), Algeria. His research interests include control systems, artificial intelligence, robotics, and computational methods. He can be contacted at email: merabtihalim@yahoo.fr.

Abdelfateh Kerrouche received the Ph.D. degree in Measurement and Instrumentation from City, University of London, UK, in 2009, funded by the Sustainable Bridges EU Project. He obtained the M.Phil. degree in Electronics from Pierre and Marie Curie University (Paris VI), France, and the M.Sc. degree in Electronics from the University of Marne-la-Vallée, France. From 2010 to 2015, he was a Research Associate at Heriot-Watt University, where he developed 3D ray-tracing models for luminescent solar concentrators and designed systems for water sampling and pathogen detection within the AquaValens EU Project. Since 2016, he has been a lecturer and Sensors/Systems Consultant at Edinburgh Napier University. He has published more than 15 papers in international journals and conferences. His research interests include pathogen detection, water sample preparation, sensors, aquaculture, and renewable energy. He can be contacted at email: a.kerrouche@napier.ac.uk.