International Journal of Robotics and Control Systems
Vol. 5, No. 6, 2025, pp. 2918-2937
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

Real-time Trajectory Tracking in a Partially Known Environment Using Dual Fuzzy Controller

Messaouda Benzaoui [a,1,*], Fadhila Lachekhab [b,2], Mohamed Tadjine [c,3]

[a] Institute of Electrical & Electronic Engineering, University of Mhamed Bougara, Boumerdes BP3500, Algeria
[b] Faculty of Hydrocarbons & Chemistry, University of Mhamed Bougara, Boumerdes BP3500, Algeria
[c] Polytechnic National School, Algiers, Algeria
[1] m.benzaoui@univ-boumerdes.dz; [2] f.lachekhab@univ-boumerdes.dz; [3] tadjine@yahoo.fr
[*] Corresponding Author

ARTICLE INFO

Article history
Received May 16, 2025
Revised August 27, 2025
Accepted December 06, 2025

Keywords
Trajectory Tracking;
Fuzzy Logic;
Mobile Robot;
Unforeseen Obstacle

ABSTRACT

Mobile robots are favorites to industrial applications nowadays. They become popular in a several fields. This increases the need of control to get more precise and autonomous ones. From the most challenging control methods to mobile robot, we can find methods to introduce real-time control motion as well as methods that solve the problems of the environment. This paper introduces both kind of control. In fact, a system of reactive navigation containing two fuzzy controllers is developed. The first is to allow the real-time trajectory tracking of a mobile robot in a partially known environment. The second is a self-correction of the trajectory according to the unexpected obstacle that can be found differently in the environment. The dual fuzzy controller is executed simultaneously in real time. To carry out these algorithms, a planner module delivers the former. The dual controller is designed using Saphira software and MATLAB then tested. First, we sample the trajectory and supply it in different shapes. Then, we use sorting algorithm to determine the point of alignment with the reference trajectory. In the second controller, we use avoidance for static and unforeseen obstacles. Finally, we present a series of simulations for these controllers in the case of the Pioneer 2P robot in multiple environments. Results of real experiments show the satisfactory and efficiency of these algorithms. Is achieved a 95 % obstacle avoidance rate and value of trajectory tracking error of 5% so is well done and the unforeseen obstacles was avoided with the proposed method. Without divergence from the trajectory. In other words, by rejoining the desired trajectory quickly after just moving away from the obstacle.

1. Introduction

The mobile robot trajectory following problem is largely subject to searches. Specifically, to obtaining a controller that follows trajectory and avoids obstacles at the same time. The most used control algorithm implemented in this field are PID Controllers, Model Predictive Controllers, Reinforcement Learning Controllers and Fuzzy Logic Controllers [1]–[4].

The use of PID algorithms provide a powerful dynamics and excellent mobility precision [4]. This algorithm can be applied in motor control level or kinematic control. In the later control needs improvement to get the best results. Towards performance variation in system parameters. Such as robot weights [2]–[8]. This improvement can be done by affecting PID parameters or by adding other control techniques.

Many authors use time-varying PID controllers [9]–[12] and test them even for several disturbances. However, they often need another algorithm when an increase in precision or speed is needed. The meta-heuristics algorithms are the most used here [13]–[15].

Another solution is the use of fractional-order FOPID for robot mobile tracking with obstacle avoidance problems. This gives a better result than simple PID, but the tuning of parameters is needed more attention from the designer. In [16] the FOPID combined with backstepping is developed for the trajectory tracking of mobile robots. It is used in more Beetle swarm optimization algorithms to tune the parameters of the kinematic and dynamic controllers. In [17], the Gray Wolf Optimization is used. In [18], a nonlinear neural fractional-order proportional integral derivative (NNFOPID) controller is proposed for the motion control of a non-holonomic differential drive mobile robot (DDMR). The tuning of these parameters uses a modified adaptive particle swarm optimization. In [19], a Gray Wolf Optimizer is used for FO-PID controllers. Time-varying PID parameters is considered in [20], the authors design a PID controller with time-varying parameters for trajectory tracking. The [21] presents a variable parameter PID controller for a differential-drive mobile robot with a desired time-varying velocity. Moreover, in [22], a robust tuning scheme is proposed where a fuzzy rule-based set point weighting technique is proposed for an initially designed fuzzy PID controller. The [23] focuses on the design of a fuzzy-PID controller for a laser sensor-based mobile robot to detect and avoid obstacles. The PID or FOPID controllers are founded in recent literature as a hybrid algorithm with meta heuristics, adaptive, fuzzy, or neuro-fuzzy. In this case, the PID is initially designed and is built through the result [24]–[29].

Model Predictive Controllers (MPC) develop the controllers implicitly by resolving an optimization problem that can be constrained. It uses a horizon optimization. The best control series across a limited potential horizon for N steps is discussed in [30]–[33]. The need for improvement also appears to be a simpler and precise controller in [34] receding horizon control. The use of fuzzy, meta-heuristics, quadratic programming, etc. [35]–[39]. Real-time predictive control is proposed in [40]–[41], nonlinear MPC is seen in [42]–[45].

Reinforcement Learning Controller (RL) is one of the most famous heuristic approaches that is used for the navigation of mobile robots in a complicated environment. Meanwhile, RL is an algorithm that finds the best decision via experience [4], [46]. To reduce these difficulties, [47] a fuzzy based system is designed to collect data sets for training of RL. The RL is used in [10] with neural networks.

The fuzzy controller is used for mobile robotic control since it can define linguistic words and make decisions consistently with predictability. As a result, FLC is suited for operating a mobile robot due to its ability to make decisions. Even in the face of unpredictability, it subsequently employs a collection of linguistically fuzzy rules to apply expert skills in a variety of circumstances [4], [48]–[51]. Meanwhile, for insufficient data, the improvement is necessary. It can include the Adaptive fuzzy sliding-mode control design [52]. The adaptive fuzzy controller is proposed in [3].

These controls are proposed for motion planning and a collision-free control algorithm. Fuzzy logic is used where inaccurate information can be processed. Fuzzy logic allows decision-making based on a priori knowledge of the controlled system. The mathematical modeling and the implementation of tracking position control of autonomous mobile robots can be processed using fuzzy logic. In a known environment, the robot's evolution can be summarized in a series of simple convergence tasks towards the goal. However, the knowledge of its environment does not give the necessary information about the followed path. Therefore, moving the robot from an initial position A to a final position B requires a hierarchy of actions as in [53]. If the environment perceived by the

robot is unknown—it may contain unforeseen obstacles—the unexpected obstacles can lead to failures in robot navigation, which requires an adaptation to new situations.

The uncertain environments or the presence of unexpected obstacles is seen in some literatures using deep learning control [54], a probabilistic online computation constraint [55], [56] fuzzy neural network with self-learning, Reinforcement learning as in [39], Adaptive Nonlinear Sliding mode control technique in [57], [58] etc.

For all of these studies, the major steps of trajectory tracking, with or without unexpected obstacles, is first the determination of error position, then alignment with the reference trajectory, and finally the determination of the point situated on the trajectory that has a minimum distance from the current position of the mobile robots. Thus, in this paper, we propose a kind sorting algorithm as a solution to answer this subject. In fact, a number of simplifying assumptions of the environment evolution are generated, and a path following module is requested. We consider that the planner delivers a reference trajectory and the tracking trajectory controller follow it reactively by amending it if is necessary according to the current situation.

The main objective in this work is to give a powerful method for determining the alignment point with the reference path. Therefore, a fuzzy controller of position error minimization and reference trajectory alignment is developed. On the other hand, in order to take into account unexpected obstacles on the planned trajectory, we develop a fuzzy controller for the obstacle avoidance. Some situations are guessed to test the collision-free algorithm. Consequently, developing a dual fuzzy controller for trajectory tracking and unforeseen obstacle avoidance in partially known environments with unexpected obstacles.

Using the environment information, the proposed algorithm must: (1) Generate a trajectory between A and B, which is achievable by the robot. The planning data are expressed as a defined trajectory given by a sequence of points corresponding to the transition situations G_i in the environment. (2) Follow this trajectory using the perception system (sensors) and actuators. The first module (path planning) is a deliberative module that relies on a representation of the environment in the form of a real or fuzzy map.

2. Control Strategy

In general, path-planning strategies are global or local in relation to the information around the environment. In global path planning the robot can reach the target by following a predefined path, global path planning is off-line path planning. In local path planning, the robot moves in real-time monitoring online path planning [59]–[61].

However, Global path planning need to build a global map model, to obtain the optimal path to guide the robot to go safely toward the target point in the actual environment. In this work, it is a combine global and local path to goals to find the global optimal path with real-time obstacle avoidance planning. In this work to get a sampled point's vector as flow:

The reference trajectory is given in the form of sampled points defined by a vector G of dimension N+1:

$$G = \{G_0, G_1, \dots G_N\} \quad (1)$$

Each point G_i of this set G is characterized by:

Cartesian coordinates (x_i, y_i)

An orientation α_i of the trajectory at this point.

$$G_i = \begin{bmatrix} x_i \\ y_i \\ \alpha_i \end{bmatrix} \quad (2)$$

The trajectory is valid when the continuity conditions of the Cartesian coordinates and the continuity of the orientation are ensured. Fig. 1 illustrates the principle of trajectory tracking: Where it is in the current time "i".

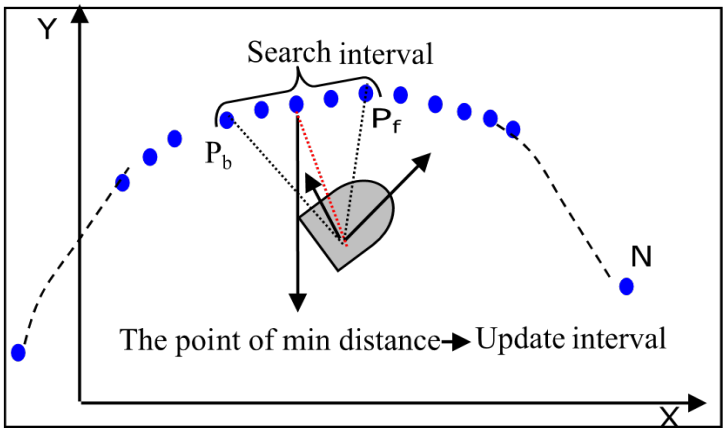

Fig. 1. Trajectory tracking principle distance point search algorithm

The η_{min} represents the minimum distance between the robot's current position and the reference trajectory, G_f represents the future point, G_{min} represents the point defining the distance η_{min}, θ_a is the angle between the direction of the robot and the orientation of the trajectory at point G_f. We will designate it later as misalignment, ϕ_{min} represents the angle between the robot's main axis and the minimum distance segment; xoy is robot frame and XOY is absolute frame.

From a known configuration of the robot, the trajectory-tracking controller must ensure:

- Convergence in position by minimizing the distance η_{min}.
- An alignment with reference trajectory by the minimization of θ_a.

2.1. Search Algorithm for the Minimum Distance Point

To carry out a search of the robot trajectory minimum distance point, we proceed as in classical sorting algorithm which consists of finding an interval $[P_b, P_r]$ of points of G_i included in the vector of G, afterward we calculate the distances of η_i from robot to different points of G_i.

We choose the width of the search interval so that the trajectory is almost linear. We would also take into account the robot dimensions and the trajectory points (sampling).

Knowing that robot state vector given by:

$$q_{cur} = \begin{bmatrix} x_{cur} \\ y_{cur} \\ \theta_{cur} \end{bmatrix} \quad (3)$$

We define the distance η_i between robot and the point G_i as follows:

$$\eta_i = \sqrt{(x_i - x_{cur})^2 + (y_i - y_{cur})^2} \quad (4)$$

Therefore, it catches the minimum distance calculated over this interval by:

$$(\eta_{min}, i_{min}) = min(\eta_i) \quad (5)$$

with $i_{min}\epsilon[P_b, P_f]$ is the index of the point that satisfies the condition (5). It corresponds to the desired point G_{min} given by (6):

$$G_{min} = \begin{bmatrix} x_{min} \\ y_{min} \\ \alpha_{min} \end{bmatrix} \tag{6}$$

After determining this point, the update of the search interval is as follows:

$$\begin{cases} If\ (i_{min} - C_1) \geq 0\ then\ \ P_b = (i_{min} - C_1) \\ \qquad\qquad \text{Else}\ \ P_b = 0 \end{cases}$$
$$\begin{cases} If\ (i_{min} + C_2) < N\ then\ \ P_f = (i_{min} + C_2) \\ \qquad\qquad \text{Else}\ \ P_f = N \end{cases} \tag{7}$$

The robot coordinate system can be transformed by using G_{min} point coordinates. They are expressed in the global former:

$${}_RG_{min} = \begin{bmatrix} {}_RX_{min} \\ {}_RY_{min} \\ {}_R\alpha_{min} \end{bmatrix} \tag{8}$$

So, the coordinates of point of G_{min} point in robot global coordinate system becomes:

$$\begin{bmatrix} {}_RX_{min} \\ {}_RY_{min} \\ {}_R\alpha_{min} \end{bmatrix} = \begin{bmatrix} \Delta_x \cos(\theta_{cur}) + \Delta_y \sin(\theta_{cur}) \\ -\Delta_x \sin(\theta_{cur}) + \Delta_y \cos(\theta_{cur}) \\ \Delta_\alpha \end{bmatrix} \tag{9}$$

With

$$\begin{bmatrix} \Delta_x \\ \Delta_y \\ \Delta_\alpha \end{bmatrix} = \begin{bmatrix} (x_{min} - x_{cur}) \\ (y_{min} - y_{cur}) \\ \alpha_{min} - \theta_{cur} \end{bmatrix}$$

We deduce that:

$$\phi_{min} = arctan(\frac{{}_RY_{min}}{{}_RX_{min}}) \tag{10}$$

The coordinates of the future point noted G_f, which will be used for the alignment with the trajectory, are determined as follows:

$$G_f = \begin{bmatrix} X_f \\ Y_f \\ \alpha_f \end{bmatrix} \quad with\ \ f = i_{min} + C_3 \tag{11}$$

Where C_3 is the number of steps to choose in the future.

Finally, the misalignment is determined by (12):

$$\theta_a = (\alpha_f - \theta_{cur}) \tag{12}$$

Once the transformation and calculation of the variables is carried out, the main objective of the trajectory-tracking controller is specified by:

- Minimizing η_{min} through ϕ_{min}, to reduce error in position.
- Minimizing θ_a to reduce error in orientation.

2.2. Problem Formulation

The planning module shown in the in Fig. 2, determines a trajectory within a modelled environment with a grid map using the methods. We notice that the proposed planner operates with a defined time horizon T, which required a previous prediction of the future states of the environments on this horizon, according to perceptive information characterizing the current situation.

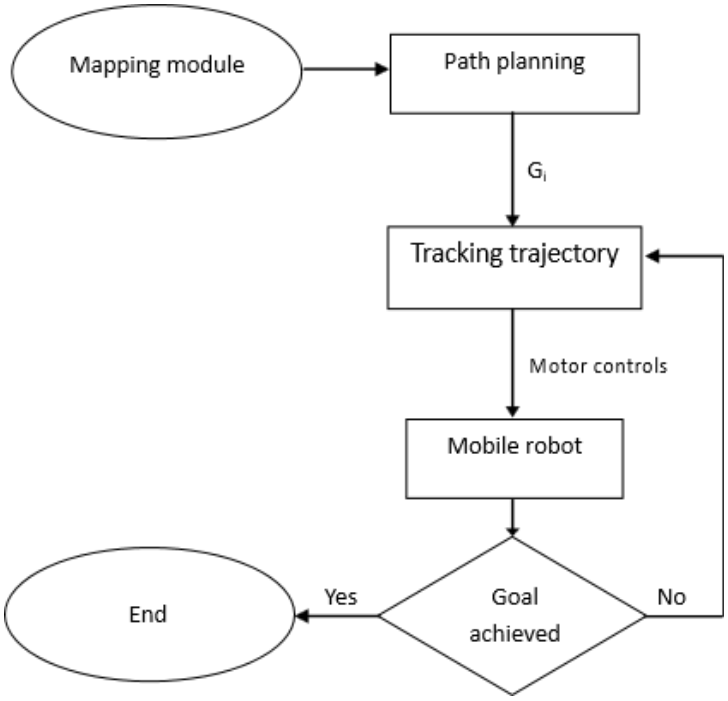

Fig. 2. Control architecture for trajectory tracking

Such prediction induces uncertainties and inaccuracies, which increase progressively with the time horizon. Therefore, two strategies are observed and defined: Uncertainties and inaccuracies are commonly modelled to generate the safest possible trajectories. In these cases, the limited space of solutions constitutes the main disadvantage. Some simplifying assumptions of the environment evolution are generated before the path following module is requested.

This work uses the second approach. We consider that the planner delivers a nominal trajectory, and the tracking trajectory controller follows it reactively by amending it if necessary, according to the current situation. Moreover, in the considered partially known environments, the fixed obstacles are modelled. The planning module generates a free trajectory, which the robot should follow. However, during the evolution, the robot may encounter unexpected (unforeseen) obstacles that interrupt its trajectory.

We note that, to perform the real experimentation, we use a mobile robot Pioneer 2P (Fig. 3) of the Activimdia firm [62]. The robot has eight ultrasonic sensors Si through eight transducers, which are placed as in Fig. 4. The front sonars are positioned as follows: one on each side and six on the front, separated by twenty degrees [62].

The main objective of this work is to create a fuzzy controller to track the nominal trajectory delivered by the path planning module. In our tracking experiments, we assume that the environment is uncluttered. That means the robot follows a path without using the obstacle avoidance module.

Fig. 3. Experimental platform of Pioneer 2P

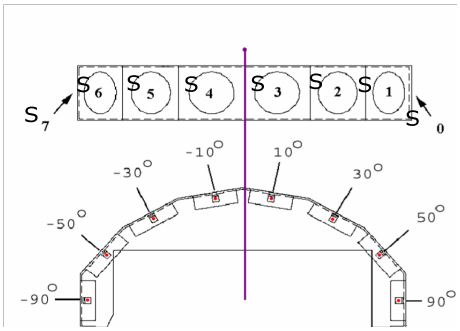

Fig. 4. Ultrasonic sonar position of Pioneer 2P

For each robot position in the absolute coordinate system, inputs values can be calculated for each moment of T. Therefore, it is necessary to elaborate a search algorithm of the point located on the trajectory and closest to the current position of the robot. The Fig. 5 carry on the proposed minimum distance point search.

3. Design of the Trajectory Tracking Controller

3.1. Fuzzy Controller Trajectory Tracking

To acquire the fuzzy controller, the first step in design is to identify the inputs/outputs to the controller. According to trajectory tracking for mobile robot, this needs different types of information: We need to know where the robot is located at each time, for this, we use sensor; we need to get the destination or the direction where the robot will move. Aimed at this, rotation of the robot in the appropriate direction needs the two angles as defined in (5) and (10). Therefore, from this information, we took the input variables as η_{min}, θ_a, and ϕ_{min}.

Each input of the fuzzy controller (η_{min},θ_a, ϕ_{min}) is decomposed into fuzzy sub-sets to get membership functions (see Fig. 6). The used labels are near and far, for the distance η_{min} and N (negative), Z (zero), P (positive) for the angles: θ_a, ϕ_{min}.

To get output of the fuzzy controller, the necessary information is to identify what we want from fuzzy controller: is to make the robot move to the destination. To accomplish this task. The robot has two controlled wheels. In general [48], [49], [63], the outputs to the controller are the speed of DC motors directly fixed to the right and left wheels of the robot. In this paper, the output variables are identified as the rotation velocity V_{rot} which gives order to DC motor to turn left or right with a fixed values and as translation one V_{ts} which gives order to robot to go left or right. The values of V_{rotS} . Fig. 7 shows an overview of the used fuzzy controller.

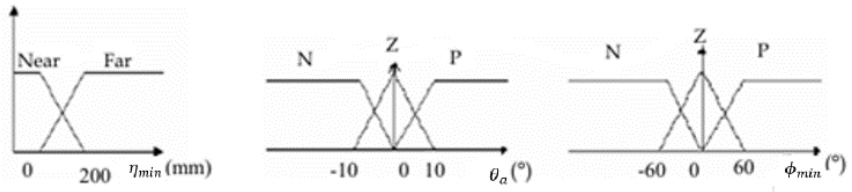

Fig. 5. Membership functions for input/output variables (fuzzification)

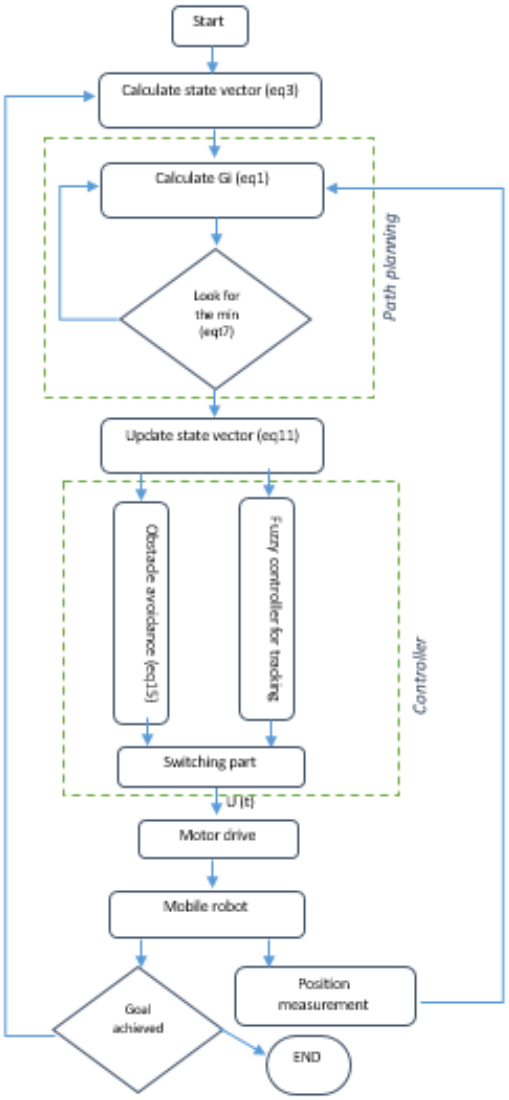

Fig. 6. Flow chart of the proposed method

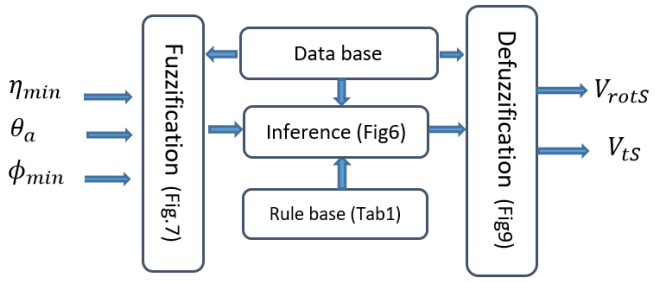

Fig. 7. The overview of the used fuzzy tracking controller

The translation speed output V_{tS} is partitioned into three fuzzy sub-sets: Zero (Z), Medium (Mm) and Maximum (Max). The rotation speed V_{rotS} is divided into three fuzzy subsets: Turn Left (TL), Zero (ZE) and Turn Right (TR) (see Fig. 8). We get the fuzzy tracking controller as in Fig. 9. This partition provides a concise rule base as shown in Table 1 and Table 2.

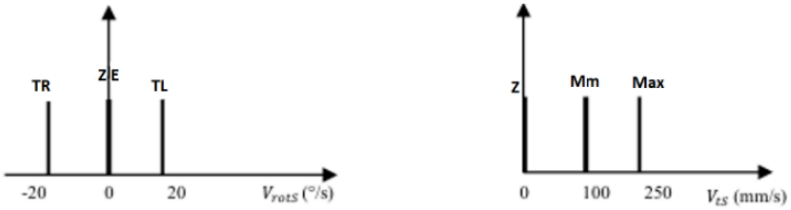

Fig. 8. The Membership functions for the output variables (defuzzification)

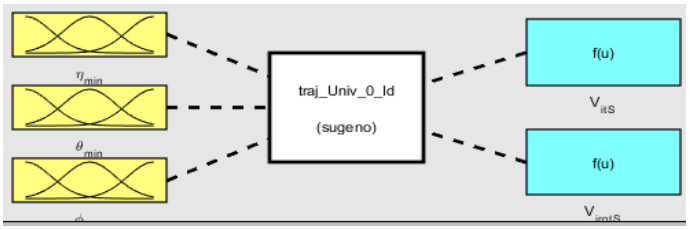

Fig. 9. The fuzzy tracking controller

Table 1. Tracking Rule base of V_{tS} & V_{rotS} respectively

B_a / B_{min}	N	Z	P
N	Z	Mm	Mm
Z	Mm	Max	Mm
P	Mm	Mm	Z

Table 2. Rule base for tracking

B_a / B_{min}	N	Z	P
N	TR	TR	ZE
Z	TR	ZE	TL
P	ZE	TL	TL

Using the consequent membership functions of the rules, the defuzzifier transforms the linguistic variables and get a crisp output value. This produces a non-fuzzy value. In the simulation, to get this result, the 'evalfis' function of MATLAB is used.

3.2. Fuzzy Controller Unforeseen Obstacle

We consider a safety distance such that the robot must not cross it "D_{free}" As the robot is far from this value, the avoidance procedure is ineffective and the robot is safe as long. As soon as it approaches, avoidance must begin. At that moment, tracking of desired trajectory is useless!

Starting from this principle, we built controller. Taking this distance D_{free} as the goal to create the fuzzy controller. It seems clear that a Mamdani model is sufficient. We divide the space seen by the obstacle detector sensors into three parts, then, it creates three inputs for the fuzzy controller (Fig. 10 and Table 3). The output is always V_{rot} and V_{tS} but with a repulsive effect.

This is justified in the case of mobile robot navigation. Indeed, depending on the distance between the sensors and their positioning to detect obstacles far from "D_{free}, and due to the constant values of the inductive sensor, blind spots appear. This indicates that obstacle detection is limited. The transition

from tracking to avoidance can be done automatically, whether the sensors are activated or disactivated.

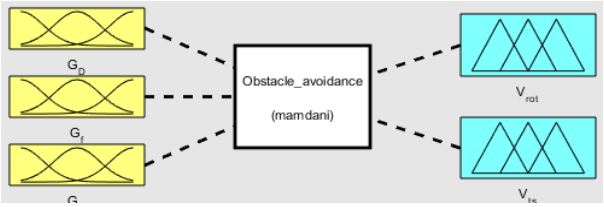

Fig. 10. The proposed Fuzzy avoidance

Table 3. Obstacle avoidance rule base of V_{ts} & V_{rots} respectively

G_D \ G_G	P	M	L
P	TR	TR	TR
M	TR	TR	TR
L	TR	TR	TR

G_D \ G_G	P	M	L
P	TR	TL	TR
M	TR	TR	TL
L	TR	TR	TR

4. Simulation of the Proposed Controller

Unlike in [48], [49] in this work the number of fuzzy sub-sets for each input is reduced to three. Also, recovers accuracy with Sugeno model. In [48] uses for inputs: inductive sensor (divided to five fuzzy subsets), the "Proximity Sensors": Left Front and right divided each to three fuzzy subsets (C: Near, M: Mid, FR: Far), for the output, the linear velocity divided to five fuzzy subsets and the angular velocity to seven fuzzy subsets. It uses a Mamdani type. External disturbances are used in experimentation to test the influence of using Fuzzy type2.

In [49] uses as six inputs: Distance to the destination (divided to three fuzzy subsets), Direction of destination from current location (five fuzzy subsets), Front obstacle (three fuzzy subsets), Left obstacle (three fuzzy subsets), Right obstacle (three fuzzy subsets), Back obstacle (three fuzzy subsets). As outputs uses 2: left motor speed (seven fuzzy subsets) and right one (seven fuzzy subsets). It is used a Mamdani and simulation tests are based more on obstacle detection.

To choose the number of fuzzy subsets or linguistic values for each input must done carefully. Certainly, the more fuzzy subsets number, the more the controller will be accurate but the more expensive to implement the electronic device required. Consequently, designer considers optimization of cost and accuracy.

The obstacle avoidance controller used in this work is switched with the tracking controller as in [62], who uses two fuzzy controller one to avoid a dynamic obstacle as a primary task and the second to do the trajectory task, but in this paper the tracking is considered as primary task.

4.1. Alignment Point with the Reference Path

The defined fuzzy controller is simulated using different forms of reference trajectories (See Fig. 11 and Fig. 12). When it shows a rectilinear form (Fig. 11), the robot initially is on the trajectory but with a different orientation as in Fig. 11–(A). The controller minimizes the error distance and then the alignment error with the trajectory in Fig. 11–(B).

In the second test, a circular reference trajectory is given by (Fig. 12–(A)). and the robot orientation is different of the reference trajectory. Initially, the robot performs a rotation on sit to rejoin the nearest point on the trajectory (Fig. 12–(B)).

The Fig. 13 shows the performed trajectories by the robot to track the reference trajectory (using eq5). Consequently, the developed algorithm provides G_i point in the reference path, which has the minimum distance to the robot's current position. Further, fuzzy controller generates the necessary path.

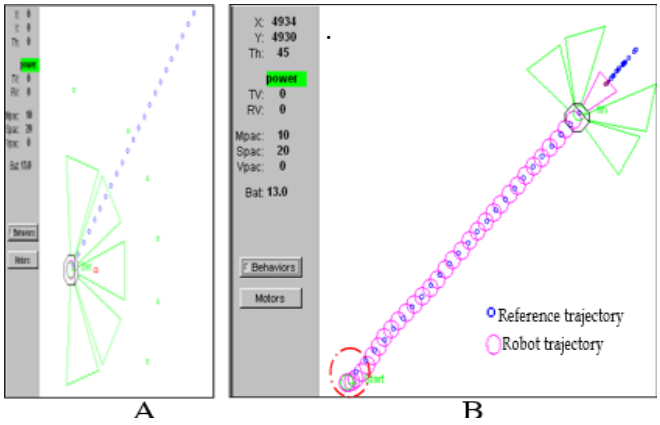

Fig. 11. Evolution of the robot on a rectilinear trajectory

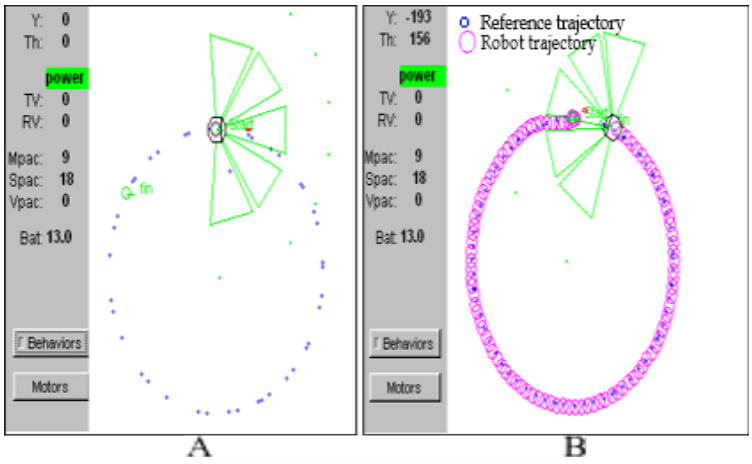

Fig. 12. Evolution of the robot on a circular trajectory tracking

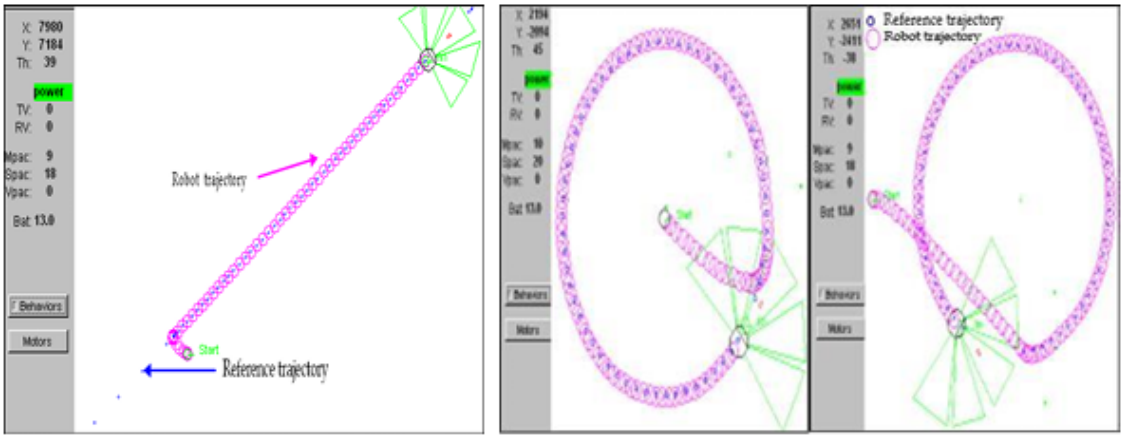

Fig. 13. Straight and Curvature trajectory tracking

4.2. Case Unforeseen Obstacles on the Trajectory

The previous case treats trajectory tracking in an unencumbered environment. However, during displacement, the robot may encounter unexpected obstacles. Therefore, a fuzzy obstacle avoidance controller is developed. The robot avoids collision with unforeseen (unexpected) obstacles in the environment. Consequently, the developed control achieves both behaviors tracking trajectory and avoiding obstacles. A switching between the two behaviors is determined using the sensors reading S_i. In each situation encountered by the robot A_i, a priority to one behavior is given. Knowing that each of them works well when acting alone. This switching is used to determine the appropriate transition moments between the commands of the two-controller's-trajectory tracking and obstacle avoidance- Using (7), the switching between the two behaviors, based on the sensor readings, is given with (13), (14) and (15).

$$\text{Trajectory tracking} \begin{cases} If\ G_F \geq 350\ mm \\ and\ \ G_D \geq 350\ mm \\ \text{and}\ \ G_G \geq 350\ mm \end{cases} \tag{13}$$

where

$$\begin{aligned} G_D &= \min(S_0, S_1, S_2) \\ G_F &= \min(S_2, S_3, S_4, S_5) \\ G_G &= \min(S_5, S_6, S_7) \end{aligned} \tag{14}$$

$$\text{Obstacle avoidance} \begin{cases} If\ G_F \leq 350\ mm \\ or\ \ G_D \leq 350\ mm \\ \text{or}\ \ G_G \leq 350\ mm \end{cases} \tag{15}$$

And S_i stands for the sensors reading in the situations A_i.

5. Simulation Results

5.1. Random form path with unforeseen obstacles

Fig. 14 shows a robot trajectory in an environment congested by obstacles close to the reference trajectory, presented in small blue circles; the trajectory simulated by using the algorithm already developed presented by pink circles. This figure shows that the robot follows correctly the trajectory and bypasses unforeseen obstacles, which clash with the path; the robot avoids the obstacles without moving away from the reference trajectory.

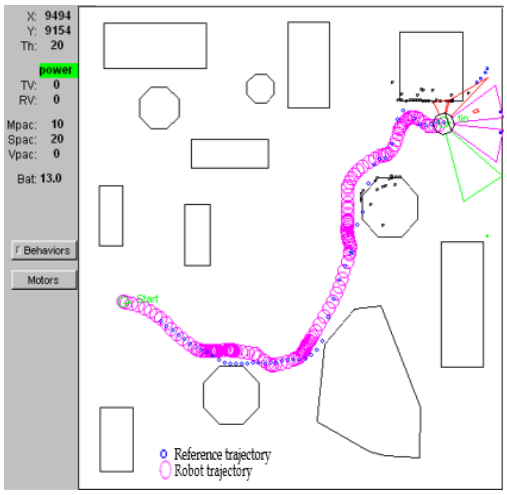

Fig. 14. Avoidance of unexpected obstacles on the reference trajectory

5.2. Circular Path with Unforeseen Obstacles

In the second simulation, shown in Fig. 15, the robot is brought to follow a circular trajectory between A1 and A2. By using the control algorithm developed in this work, trajectory tracking is, perfectly done, with satisfactory precision. In fact, the reference and actual trajectories are superimposed. However, between A2 and A3 it appears a small deviation because of the black object that inter almost in the desired path, therefore the robot deviates from the reference trajectory to avoid the unexpected obstacle as clearly shown in Fig. 16. Likewise, on the section between the points A3 and A4, the robot joins the well tracked. In sections A4 − 5 and A5 A6 − A7, the robot encounters again unforeseen obstacles whether inside or outside the circle. Controller sends the order to the robot motors to turn right or left with the satisfactory angle. Then it joins again the reference trajectory.

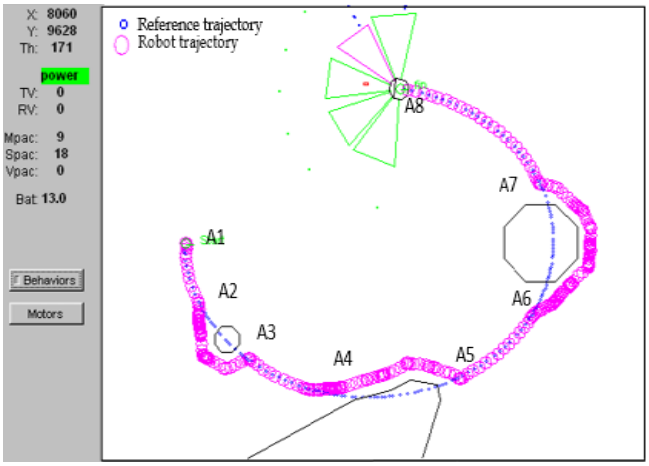

Fig. 15. Trajectory tracking and avoidance of unexpected obstacles

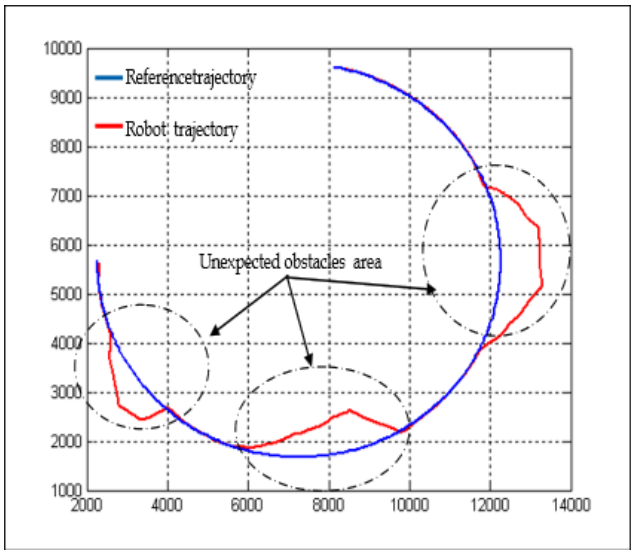

Fig. 16. Reference trajectory and robot trajectory paths

5.3. Random path with Unforeseen Obstacles

This simulation, shown in Fig. 17 and Fig. 18, the desired path is circular, the unforeseen obstacles are interrupting the robot to follow the reference trajectory. The robot avoids fixed obstacles and follows correctly the trajectory defined between the starting points and the goal. Fig. 17 shows the reference trajectory delivered by the path-planning module. The robot manages correctly to follow this trajectory. Fig. 18 shows the reference trajectory (blue) and the robot performed trajectories (red) simultaneously. In Fig. 17–(A), an unexpected obstacle intersects the reference trajectory. The control

system then uses the obstacle avoidance module to allow the robot to bypass it, and then the robot manages to join the reference trajectory in Fig. 17–(B).

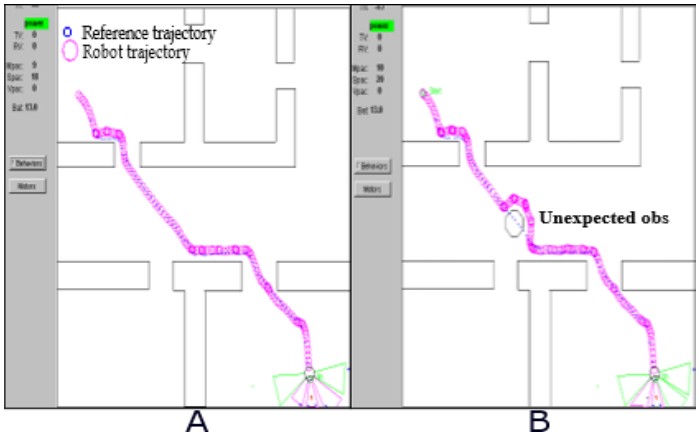

Fig. 17. Reference and robot performed trajectories

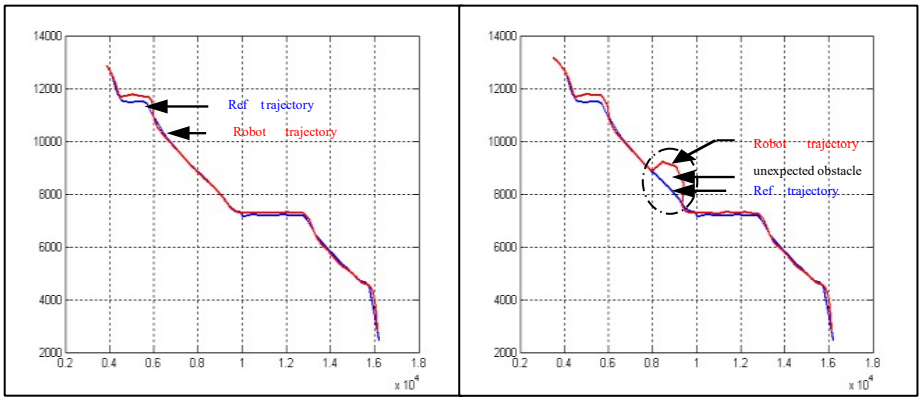

Fig. 18. Reference and robot performed trajectories

6. Real Experimentations on Pioneer 2P

This study allows the tracking of a planned trajectory in a partially known environment of a holonomic wheeled mobile robot Pioneer 2P. The Pioneer II robot is controlled by Saphira software developed by Kurt Konolige of the International Artificial Intelligence Laboratory (IRS) [62]. From version 8.x, several core functions of Saphira have been moved completely to the ARIA software "Activmedia Robot Interface for Application". The updating frequency includes other functionalities motion control and data logging of robot with laser and embedded camera in the robotic, Floor plan of the test environment is approximately 4 Hz.

Saphira and Aria are written in the C++ language, and their APIs use the properties of C++ objects to provide an efficient interface to robot programming. For programming ease, a block diagram of Simulink is used with APIs (S-Function) of MATLAB, which groups together the main functions of Saphira of perception, localization and commands.

6.1. Tracking Trajectory without Obstacles

In this test, the robot follows the circular trajectory shown in Fig. 15 right. The robot initial position is different from the reference trajectory. In first step, it catches the trajectory by eliminating the position and alignment error in Fig. 19 represents the different real movements of the robot to track this trajectory.

6.2. Tracking Trajectory with Unforeseen Obstacle

The planned trajectory has a sinusoidal curvature. In this real test, an unexpected obstacle is on the reference path of the robot. Fig. 20 shows that the robot manages to avoid the unforeseen obstacle and then follow the planned trajectory.

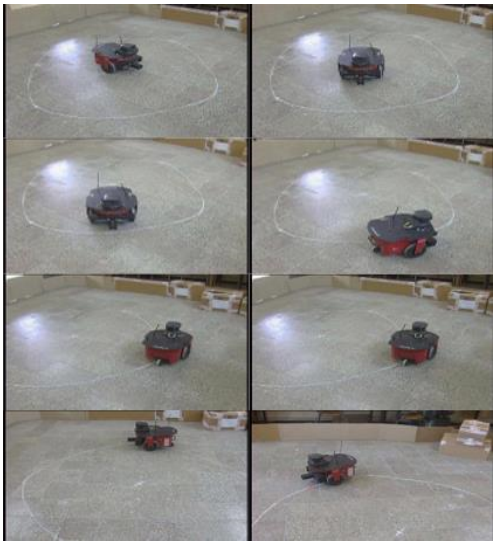

Fig. 19. Real experimentation of tracking trajectory without unforeseen obstacle

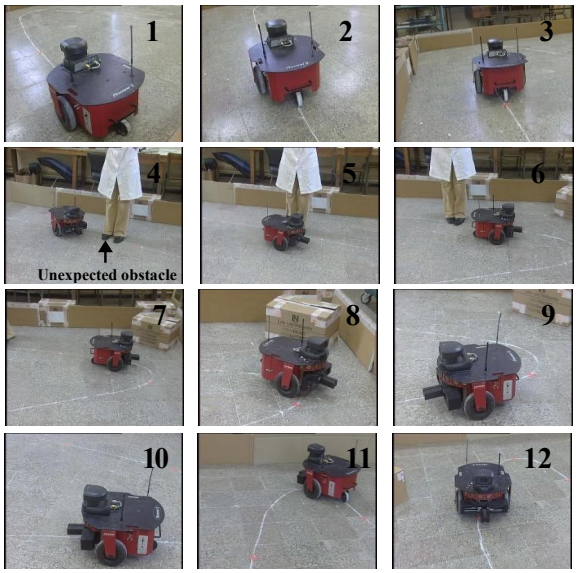

Fig. 20. Real experimentation of tracking trajectory with unforeseen obstacle

7. Conclusion

In this paper, we have developed a navigation system for the robot based on fuzzy controllers, which allows tracking a reference trajectory. The fuzzy controller tracks the sampled reference path delivered by the planner. When the robot operates in an environment consisting of fixed and unexpected obstacles, the navigation module uses the obstacle avoidance fuzzy controller. The switching between the two behaviors is done correctly according to a simple logic. An efficient search algorithm of the nearest point has been developed. All programs elaborated are simulated with Simulink of MATLAB and Sapphira software. Based on our results and observations, for both simulations and real experiments on the robot, the obtained results are considered satisfactory since tracking error is 95% without obstacle. Nevertheless, the presence of the obstacle does not allow to

judge tracking well: the more the obstacle enters the more the trajectory is modified but t the algorithm quickly rejoins the desired path as soon as the robot moves away.

As a perspective of further work, we will try to optimize the membership functions of the fuzzy systems by using artificial intelligent techniques, we are thinking of testing this method for other types of mobile robots in a dynamic environment. This may require the addition of a hybrid control with DL for example.

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

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] Z. Sun, H. Xie, J. Zheng, and Z. Man, "Path-following control of mecanum-wheels omnidirectional mobile robots using nonsingular terminal sliding mode," *Mechanical Systems and Signal Processing*, vol. 147, p. 107128, 2021, https://doi.org/10.1016/j.ymssp.2020.107128.

[2] F. Cañadas-Aránega, J. C. Moreno, and J. L. Blanco-Claraco, "A PID-based control architecture for mobile robot path planning in greenhouses," *IFAC-PapersOnLine*, vol. 58, no. 7, pp. 503–508, 2024, https://doi.org/10.1016/j.ifacol.2024.08.112.

[3] J. Nie, Y. Wang, Z. Miao, Y. Jiang, H. Zhong, and J. Lin, "Adaptive fuzzy control of mobile robots with full-state constraints and unknown longitudinal slipping," *Nonlinear Dynamics*, vol. 106, no. 4, pp. 3315–3330, Dec. 2021, https://doi.org/10.1007/s11071-021-06933-y.

[4] T. Zhao, P. Qin, and Y. Zhong, "Trajectory tracking control method for omnidirectional mobile robot based on self-organizing fuzzy neural network and preview strategy," *Entropy*, vol. 25, p. 248, 2023, https://doi.org/10.3390/e25020248.

[5] S. MohandSaidi and R. Mellah, "Real-time speed control of a mobile robot using PID controller," in *Artificial Intelligence and Its Applications*, B. Lejdel, E. Clementini, and L. Alarabi, Eds., Lecture Notes in Networks and Systems, vol. 413. Cham: Springer, 2022, pp. 675–688, https://doi.org/10.1007/978-3-030-96311-8_51.

[6] J. H. Park, "Optimal controller design for a mobile robot using genetic algorithm and adaptive PID," *IEEE Access*, vol. 13, pp. 86167–86184, 2025, https://doi.org/10.1109/ACCESS.2025.3570472.

[7] R. Haq, H. Prayitno, I. Sucahyo, and E. Rahmawati, "A low-cost mobile robot based on proportional integral derivative (PID) control system and odometer for education," *Journal of Physics: Conference Series*, vol. 997, no. 1, p. 012046, 2018, https://doi.org/10.1088/1742-6596/997/1/012046.

[8] B. M. Yousuf, A. Saboor Khan, and S. Munir Khan, "Dynamic modeling and tracking for nonholonomic mobile robot using PID and back-stepping," *Advanced Control for Applications: Engineering and Industrial Systems*, vol. 3, no. 3, p. e71, 2021, https://doi.org/10.1002/adc2.71.

[9] P. Chotikunnan and R. Chotikunnan, "Dual design PID controller for robotic manipulator application," *Journal of Robotics and Control*, vol. 4, no. 1, pp. 23–34, 2023, https://doi.org/10.18196/jrc.v4i1.16990.

[10] T. T. K. Ly, N. H. Thai, and L. T. Phong, "Design of neural network-PID controller for trajectory tracking of differential drive mobile robot," *Vietnam Journal of Science and Technology*, vol. 62, no. 2, pp. 374–386, 2024, https://doi.org/10.15625/2525-2518/18066.

[11] M. Gheisarnejad and M. H. Khooban, "An intelligent non-integer PID controller-based deep reinforcement learning: Implementation and experimental results," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 4, pp. 3609–3618, 2021, https://doi.org/10.1109/TIE.2020.2979561.

[12] H. Ammar, M. Ibrahim, A. Azar, and R. Shalaby, "Gray wolf optimization of fractional order control of 3-omni wheels mobile robot: Experimental study," in *Proceedings of the 2020 16th International Computer Engineering Conference (ICENCO)*, Cairo, Egypt, 2020, pp. 147–152, https://doi.org/10.1109/ICENCO49778.2020.9357384.

[13] A. Al-Jodah, S. J. Abbas, A. F. Hasan, A. J. Humaidi, A. S. M. Al-Obaidi, A. A. Al-Qassar, and R. F. Hassan, "PSO-based optimized neural network PID control approach for a four wheeled omnidirectional mobile robot," *International Review of Applied Sciences and Engineering*, vol. 14, no. 1, pp. 58–67, 2023, https://doi.org/10.1556/1848.2022.00420.

[14] H. Khan, S. Khatoon, P. Gaur, M. Abbas, C. A. Saleel, and S. A. Khan, "Speed control of wheeled mobile robot by nature-inspired social spider algorithm-based PID controller," *Processes*, vol. 11, no. 4, p. 1202, 2023, https://doi.org/10.3390/pr11041202.

[15] H. Goud, P. C. Sharma, K. Nisar, M. R. Haque, A. A. A. Ibrahim, N. S. Yadav, P. Swarnkar, M. Gupta, and L. Chand, "Metaheuristics algorithm for tuning of PID controller of mobile robot system," *Computers, Materials & Continua*, vol. 72, no. 2, pp. 3481–3492, 2022, https://doi.org/10.32604/cmc.2022.019764.

[16] J. D. L. Xu, S. Baoye, and C. Maoyong, "A combined backstepping and fractional-order PID controller to trajectory tracking of mobile robots," *Systems Science & Control Engineering*, vol. 10, no. 1, pp. 134–141, 2022, https://doi.org/10.1080/21642583.2022.2047125.

[17] L. Dong, Z. He, C. Song, and C. Sun, "A review of mobile robot motion planning methods: From classical motion planning workflows to reinforcement learning-based architectures," *Journal of Systems Engineering and Electronics*, vol. 34, no. 2, pp. 439–459, 2023, https://doi.org/10.23919/JSEE.2023.000051.

[18] G. A. R. Ibraheem, A. T. Azar, I. K. Ibraheem, and A. J. Humaidi, "A novel design of a neural network based fractional PID controller for mobile robots using hybridized fruit fly and particle swarm optimization," *Complexity*, vol. 2020, p. 3067024, 2020, https://doi.org/10.1155/2020/3067024.

[19] J. S. Ling Leong, K. T. Kin Teo, and H. P. Yoong, "Four wheeled mobile robots: A review," in *IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, Kota Kinabalu, Malaysia, 2022, pp. 1–6, https://doi.org/10.1109/IICAIET55139.2022.9936855.

[20] F. Lachekhab, D. Acheli, M. Tadjine, and Y. Meraihi, "Heuristic and learning method for obstacle avoidance with mobile robot," in *2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH)*, 2021, pp. 34–39, https://doi.org/10.1109/IHSH51661.2021.9378725.

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

[22] P. Mitra, C. Dey, and R. K. Mudi, "Fuzzy PI controller with dynamic set point weighting," in *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA)*, S. C. Satapathy, S. K. Udgata, and B. N. Biswal, Eds., Berlin, Heidelberg: Springer, 2013, pp. 51–58, https://doi.org/10.1007/978-3-642-35314-7_7.

[23] A. Rehman and C. Cai, "Autonomous mobile robot obstacle avoidance using fuzzy-PID controller in robot's varying dynamics," in *2020 39th Chinese Control Conference (CCC)*, Shenyang, China, 2020, pp. 2182–2186, https://doi.org/10.23919/CCC50068.2020.9188467.

[24] L. Panta, "Comparative analysis of NMPC and fuzzy PID controllers for trajectory tracking in omni-drive robots: Design, simulation, and performance evaluation," *International Journal of Fuzzy Systems*, vol. 27, no. 6, pp. 1691–1701, Sep. 2025, https://doi.org/10.1007/s40815-024-01866-1.

[25] X. Gao, R. Gao, P. Liang, Q. Zhang, R. Deng, and W. Zhu, "A hybrid tracking control strategy for nonholonomic wheeled mobile robot incorporating deep reinforcement learning approach," *IEEE Access*, vol. 9, pp. 15592–15602, 2021, https://doi.org/10.1109/ACCESS.2021.3053396.

[26] N. Y. Allagui, F. A. Salem, and A. M. Aljuaid, "Artificial fuzzy-PID gain scheduling algorithm design for motion control in differential drive mobile robotic platforms," *Computational Intelligence and Neuroscience*, vol. 2021, no. 1, p. 5542888, 2021, https://doi.org/10.1155/2021/5542888.

[27] M. Gheisarnejad and M. H. Khooban, "An intelligent non-integer PID controller-based deep reinforcement learning: implementation and experimental results," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 4, pp. 3609–3618, Apr. 2021, https://doi.org/10.1109/TIE.2020.2979561.

[28] D. Babunski, J. Berisha, E. Zaev, and X. Bajrami, "Application of fuzzy logic and PID controller for mobile robot navigation," in *2020 9th Mediterranean Conference on Embedded Computing (MECO)*, Budva, Montenegro, 2020, pp. 1–4, https://doi.org/10.1109/MECO49872.2020.9134317.

[29] I. Carlucho, M. D. Paula, and G. G. Acosta, "An adaptive deep reinforcement learning approach for MIMO PID control of mobile robots," *ISA Transactions*, vol. 102, pp. 280–294, 2020, https://doi.org/10.1016/j.isatra.2020.02.017.

[30] P. Li, S. Wang, H. Yang, and H. Zhao, "Trajectory tracking and obstacle avoidance for wheeled mobile robots based on EMPC with an adaptive prediction horizon," *IEEE Transactions on Cybernetics*, vol. 52, no. 12, pp. 13536–13545, 2022, https://doi.org/10.1109/TCYB.2021.3125333.

[31] 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*, vol. 54, no. 9–10, pp. 1319–1325, 2021, https://doi.org/10.1177/00202940211043070.

[32] J. Pankert and M. Hutter, "Perceptive model predictive control for continuous mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6177–6184, Oct. 2020, https://doi.org/10.1109/LRA.2020.3010721.

[33] S. Stavrinidis and P. Zacharia, "An ANFIS-based strategy for autonomous robot collision-free navigation in dynamic environments," *Robotics*, vol. 13, no. 8, p. 124, 2024, https://doi.org/10.3390/robotics13080124.

[34] P. N. Dao, H. Q. Nguyen, T. L. Nguyen, and X. S. Mai, "Finite horizon robust nonlinear model predictive control for wheeled mobile robots," *Mathematical Problems in Engineering*, vol. 2021, no. 1, pp. 6611992, 2021, https://doi.org/10.1155/2021/6611992.

[35] Y. Kiwon, "Design of deep neural network-based model predictive controller for a car-like mobile robot," *International Journal of Mechanical Engineering and Robotics Research*, vol. 11, no. 8, pp. 606–613, 2022, https://doi.org/10.18178/ijmerr.11.8.606-613.

[36] J. Tang, S. Wu, B. Lan, Y. Dong, Y. Jin, G. Tian, W.-A. Zhang, and L. Shi, "GMPC: Geometric model predictive control for wheeled mobile robot trajectory tracking," *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4822–4829, 2024, https://doi.org/10.1109/LRA.2024.3381088.

[37] Z. Jian, Z. Yan, X. Lei, Z. Lu, B. Lan, X. Wang, and B. Liang, "Dynamic control barrier function-based model predictive control to safety-critical obstacle-avoidance of mobile robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023, pp. 3679–3685, https://doi.org/10.1109/ICRA48891.2023.10160857.

[38] J. Wei and B. Zhu, "Model predictive control for trajectory-tracking and formation of wheeled mobile robots," *Neural Computing and Applications*, vol. 34, pp. 16351–16365, 2022, https://doi.org/10.1007/s00521-022-07195-4.

[39] È. Pairet, J. D. Hernández, M. Carreras, Y. Petillot, and M. Lahijanian, "Online mapping and motion planning under uncertainty for safe navigation in unknown environments," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3356–3378, Oct. 2022, https://doi.org/10.1109/TASE.2021.3118737.

[40] G. Bai, L. Liu, Y. Meng, S. Liu, L. Liu, and W. Luo, "Real-time path tracking of mobile robot based on nonlinear model predictive control," *Nongye Jixie Xuebao / Transactions of the Chinese Society of Agricultural Machinery*, vol. 51, no. 9, pp. 47–52, 2020, https://doi.org/10.6041/j.issn.1000-1298.2020.09.006.

[41] J. G. Romero, E. Nuño, E. Restrepo, R. Cisneros, and M. Morales, "A smooth time-varying PID controller for nonholonomic mobile robots subject to matched disturbances," *Journal of Intelligent and Robotic Systems*, vol. 105, p. 13, 2022, https://doi.org/10.1007/s10846-022-01622-3.

[42] O. Y. Ismael, M. Almaged, and A. I. Abdulla, "Nonlinear model predictive control-based collision avoidance for mobile robot," *Journal of Robotics and Control (JRC)*, vol. 5, no. 1, pp. 142–151, Jan. 2024, https://doi.org/10.18196/jrc.v5i1.20615.

[43] K. Alireza and I. Sharifi, "Resilient nonlinear model predictive control for formation-containment of multi-mobile robot systems," *Robotics and Autonomous Systems*, vol. 189, p. 104983, 2025, https://doi.org/10.1016/j.robot.2025.104983.

[44] Y. Hu, H. Su, J. Fu, H. R. Karimi, G. Ferrigno, E. De Momi, and A. Knoll, "Nonlinear model predictive control for mobile medical robot using neural optimization," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 12, pp. 12636–12645, 2021, https://doi.org/10.1109/TIE.2020.3044776.

[45] L. Guan, Y. Lu, Z. He, and X. Chen, "Intelligent obstacle avoidance algorithm for mobile robots in uncertain environment," *Journal of Robotics*, vol. 2022, no. 1, p. 8954060, 2022, https://doi.org/10.1155/2022/8954060.

[46] J.-Y. Zhai and Z.-B. Song, "Adaptive sliding mode trajectory tracking control for wheeled mobile robots," *International Journal of Control*, vol. 92, no. 10, pp. 2255–2262, 2019, https://doi.org/10.1080/00207179.2018.1436194.

[47] O. Gamal, X. Cai and H. Roth, "Learning from fuzzy system demonstration: Autonomous navigation of mobile robot in static indoor environment using multimodal deep learning," in *Proc. 2020 24th Int. Conf. System Theory, Control and Computing (ICSTCC)*, Sinaia, Romania, 2020, pp. 218–225, https://doi.org/10.1109/ICSTCC50638.2020.9259786.

[48] F. Cuevas, O. Castillo, and P. Antonio, "Design of a control strategy based on type-2 fuzzy logic for omnidirectional mobile robots," *Journal of Multiple-Valued Logic & Soft Computing*, vol. 37, pp. 107–136, 2021, https://www.researchgate.net/publication/360244831_Design_of_a_Control_Strategy_Based_on_Type-2_Fuzzy_Logic_for_Omnidirectional_Mobile_Robots.

[49] A. Wondosen and S. Dereje, "Fuzzy logic controller design for mobile robot outdoor navigation," *arXiv preprint*, arXiv:2401.01756, 2024, https://doi.org/10.48550/arXiv.2401.01756.

[50] M. Zangeneh, E. Aghajari, and M. Forouzanfar, "A review on optimization of fuzzy controller parameters in robotic applications," *IETE Journal of Research*, vol. 68, no. 6, pp. 4150–4159, 2022, https://doi.org/10.1080/03772063.2020.1787878.

[51] R. Chai, H. Niu, J. Carrasco, F. Arvin, H. Yin, and B. Lennox, "Design and experimental validation of deep reinforcement learning-based fast trajectory planning and control for mobile robot in unknown environment," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 5778–5792, 2024, https://doi.org/10.1109/TNNLS.2022.3209154.

[52] J.-T. Huang Chiu, "Adaptive fuzzy sliding mode control of omnidirectional mobile robots with prescribed performance," *Processes*, vol. 9, no. 12, p. 2211, 2021, https://doi.org/10.3390/pr9122211.

[53] S. Stavrinidis and P. Zacharia, "An ANFIS-based strategy for autonomous robot collision-free navigation in dynamic environments," *Robotics*, vol. 13, no. 8, p. 124, 2024, https://doi.org/10.3390/robotics13080124.

[54] J. Wang, Z. Liu, H. Chen, Y. Zhang, D. Zhang and C. Peng, "Trajectory tracking control of a skid-steer mobile robot based on nonlinear model predictive control with a hydraulic motor velocity mapping," *Applied Sciences*, vol. 14, no. 1, p. 122, 2024, https://doi.org/10.3390/app14010122.

[55] R. Chai, H. Niu, J. Carrasco, F. Arvin, H. Yin and B. Lennox, "Design and experimental validation of deep reinforcement learning-based fast trajectory planning and control for mobile robot in unknown environment," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 5778–5792, Apr. 2024, https://doi.org/10.1109/TNNLS.2022.3209154.

[56] J. Wang, M. T. Fader and M. Joshua, "Learning-based model predictive control for improved mobile robot path following using Gaussian processes and feedback linearization," *Journal of Field Robotics*, vol. 40, no. 5, pp. 1014–1033, 2023, https://doi.org/10.1002/rob.22165.

[57] J. A. Abdulsaheb and D. J. Kadhim, "Classical and heuristic approaches for mobile robot path planning: A survey," *Robotics*, vol. 12, no. 4, p. 93, 2023, https://doi.org/10.3390/robotics12040093.

[58] D. Wang, W. Wei, Y. Yeboah, Y. Li, and Y. Gao, "A robust model predictive control strategy for trajectory tracking of omni-directional mobile robots," *Journal of Intelligent & Robotic Systems*, vol. 98, no. 2, pp. 439–453, 2020, https://doi.org/10.1007/s10846-019-01083-1.

[59] L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang, "Path planning techniques for mobile robots: Review and prospect," *Expert Systems with Applications*, vol. 227, pp. 120254, 2023, https://doi.org/10.1016/j.eswa.2023.120254.

[60] A. Muhammad, M. A. H. Ali, and I. H. Shanono, "A review on intelligent mobile robot path planning techniques," in *2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2021, pp. 53–58, https://doi.org/10.1109/ISCAIE51753.2021.9431819.

[61] M. S. Alam, M. U. Rafique, and M. U. Khan, "Mobile robot path planning in static environments using particle swarm optimization," *arXiv preprint*, arXiv:2008.10000, 2020, https://doi.org/10.48550/arXiv.2008.10000.

[62] F. Lachehab, M. Tadjine and M. Kesraoui, "Experimental evaluation of new navigator of mobile robot using fuzzy Q-learning," *International Journal of Engineering Systems Modelling and Simulation*, vol. 11, no. 2, pp. 50–59, 2019, https://doi.org/10.1504/IJESMS.2019.101670.

[63] P. M. Tuan, N. D. Tai, T. Q. Huy and N. T. Thinh, "Flexible path planning of mobile robot for avoiding the dynamic obstacles using fuzzy controllers," *International Journal of Mechanical Engineering and Robotics Research*, vol. 13, no. 1, pp. 126–132, 2024, https://doi.org/10.18178/ijmerr.13.1.126-132.