Journal of Engineering and Technology for Industrial Applications

ITEGAM-JETIA

Manaus, v.11 n.52, p. 68-75. March / April, 2025.
DOI: https://doi.org/10.5935/jetia.v11i52.1534

ISSN ONLINE: 2447-0228

RESEARCH ARTICLE OPEN ACCESS

NAVIGATION SYSTEM AND OBSTACLE AVOIDANCE FOR MOBILE ROBOT USING TYPE-2 FUZZY LOGIC IN INCERTAIN ENVIRONMENTS

Soufiane. Hachani[1], Emira. Nechadi[2]

[1] Laboratory of Intelligent Systems (LSI), Department of Electronical University of Setif 1, Algeria.
[2] Department of Electrotechnical, University of Setif 1, Algeria

[1]http://orcid.org/ 0009-0000-4817-397X, [2]http://orcid.org/ 0000-0002-0642-7967

Email: hachani.soufiane@univ-setif.dz , emira.nechadi@univ-setif.dz

ARTICLE INFO

Article History
Received: January 06, 2025
Revised: February 20, 2025
Accepted: March 15, 2025
Published: March 31, 2025

Keywords:
Navigation System,
Mobile Robot,
Type-2 Fuzzy Logic,
Uncertain Environments,
Obstacle avoidance,

ABSTRACT

This paper introduces a reactive navigation strategy for wheeled mobile robots, utilizing type-2 fuzzy logic to manage uncertainty in dynamic environments. The approach incorporates two distinct type-2 fuzzy logic controllers, each tailored to address specific challenges in navigation. The first controller focuses on steering the robot toward its target by continuously adjusting its path in response to changing conditions. The second controller specializes in obstacle avoidance, enabling the robot to detect and maneuver around obstacles it encounters during its journey.

To evaluate the performance of the system, numerical simulations are carried out across various scenarios, including dynamic and cluttered environments, to demonstrate its robustness. Additionally, the results of the type-2 fuzzy logic approach are compared with conventional navigation techniques, such as rule-based or model-based methods. The comparison underscores the system's greater adaptability and resilience. The study concludes that type-2 fuzzy logic provides an effective and flexible solution, significantly improving both path planning and real-time decision-making in unpredictable and complex environments.

I. INTRODUCTION

The field of mobile robot navigation has gained increasing importance in recent years due to the rising use of robots in various fields, including industrial and service applications [1]. The control of motion for mobile robots involves several research areas, such as path planning and tracking algorithms [2], precise control design for trajectory following (3], and obstacle avoidance [4], [5]. The demand for robots to navigate in unknown and dynamic environments, filled with both static and dynamic obstacles, has driven the development of advanced systems for planning and navigation. Current path planning methods can be broadly categorized into two main types: global planning methods and local or reactive planning methods [6], [7].

Global path planning, also known as off-line or static path planning, refers to the process where the robot has prior knowledge of the environment and can reach its destination via a pre-defined path. This category includes various algorithms, such as graph-based methods like Dijkstra's algorithm, roadmap-based methods such as RRT and PRM, and topological methods including cell decomposition and Voronoi diagrams [8].

In contrast to global path planning algorithms, which rely on prior knowledge and environmental mapping, reactive or local navigation strategies employ sensors to monitor the robot's surroundings in real-time, allowing the robot to make quick decisions without prior knowledge of the environment. These reactive strategies often involve various algorithms such as Artificial Potential Fields (APF) [9], Control Barrier Functions (CBF) ([10]), and Fuzzy Logic Controllers (FLC) [11].

Fuzzy Logic Control (FLC) has become a key element in reactive navigation strategies due to its ability to make decisions in real-time based on sensory data. FLC has proven particularly useful for handling nonlinearities and managing dynamic, uncertain environments. Several researchers have proposed using FLC systems for mobile robot navigation. For example, [12], [13] proposed fuzzy control systems for guiding robots to their target destinations while avoiding obstacles. Furthermore, [14] developed an optimal fuzzy tracking control system based on the Takagi-Sugeno model and Linear Quadratic Regulator (LQR) for improving trajectory tracking and obstacle avoidance.

However, traditional fuzzy logic systems face difficulties in precisely defining membership values within the [0, 1] range for fuzzy systems. To address this issue, Type-2 Fuzzy Logic Systems (T2FLS) have been explored as a more effective solution [15].

These systems help in dealing with higher levels of uncertainty, making them ideal for dynamic environments where information may be incomplete or fuzzy. Similarly, researchers such as ([11]) have employed Particle Swarm Optimization (PSO) to enhance fuzzy membership functions, while others have applied neuro-fuzzy systems in sensor-based navigation [16], [17]. Additionally, Zadeh's concept of Type-2 fuzzy logic, introduced in 1975 [18], has played a significant role in enhancing robot adaptability to unforeseen environments.

In summary, this paper presents an approach to design and implement an interactive navigation system for mobile robots using type 2 fuzzy logic controllers (T2FLC). This technique aims to enhance the robot's ability to navigate dynamic and uncertain environments effectively, providing a more robust solution to the challenges of obstacle avoidance and real-time path planning.

II. MODEL OF THE MOBILE ROBOT USED IN OUR WORK

The robot used in the simulation is a single-wheeled mobile robot, driven by two independently controlled wheels powered by separate motors. It may also be equipped with passive wheels to maintain stability. The real-world robot is assumed to have a range of sensors to measure the distance to nearby obstacles and monitor its speed. The navigation approach employed in this study follows an interactive navigation strategy.

The primary objective of any robotic navigation system is to direct the robot towards a predefined target area. The secondary objective is to prevent collisions with obstacles. Both objectives are achieved by providing the robot with the necessary commands to minimize the discrepancy between its current position and the target location. For obstacle detection, appropriate sensors, such as ultrasonic sensors, are employed.

It is assumed that the robot-target configuration is represented by TP = [xT, yT, θT]. Additionally, the error vector between the robot's actual position and the nearest obstacle is defined using two variables: DRO and θRO. Similarly, the error between the robot's actual position and the target location can be calculated by considering two parameters: DRT and θRT, as shown in (Figure.1).

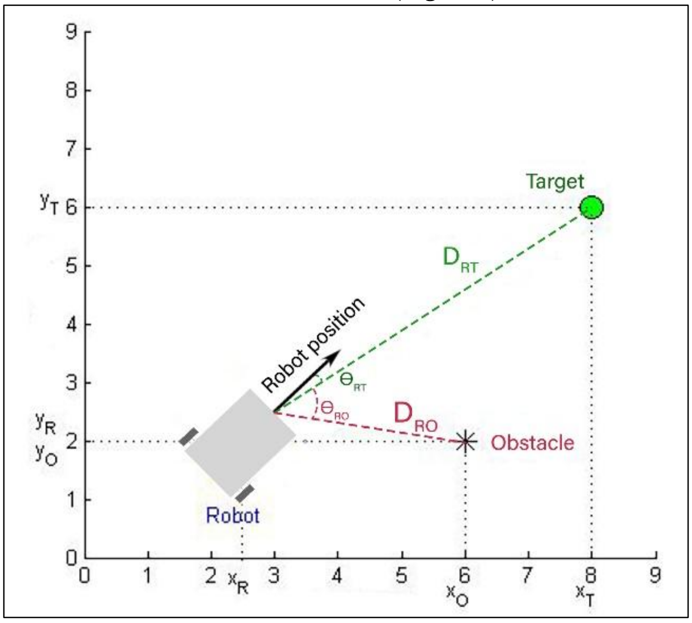


Figure 1: mobile robot used in our work.
Source: Authors, (2025).

Explanation of the abbreviations in (Fig.1):

(x_r, y_r) : The robot position.

(x_T, y_T) : The coordinates of the target point.

V: Linear velocity.

ω: Angular velocity.

V_r : The speed of the right wheel.

V_l : The speed of the left wheel.

θ_R : The orientation of the robot.

θ_T: The orientation of the target.

θ_{RT} : The angle between the current orientation of the robot and that of the target.

D_{RT} : The distance between the robot and the target.

III. STATE REPRESENTATION

State representation in the context of mobile robots refers to how the various variables that describe a robot's state are modeled and used to control its behavior. This typically includes variables such as position, orientation, velocity, and other relevant parameters that completely characterize the robot's state at any given time.

For a two-wheeled robot, the state representation is often expressed as a state vector that includes the robot's position on the plane (x, y), its orientation (θ), and sometimes linear and angular velocities (v, ω). The aim is to relate control inputs, such as wheel speeds, to changes in these state variables.

In summary, state representation is used to model the evolution of the robotic system over time using differential equations that describe the robot's dynamics. These models are essential for control, trajectory planning, and autonomous navigation.

The kinematic model of the mobile robot is given by:

$$\begin{cases} \dot{x} = V.\cos(\theta_R) \\ \dot{y} = V.\sin(\theta_R) \\ \dot{\theta} = \omega \end{cases} \quad (1)$$

The total speed of the robot can be approximated as the average of the speeds of the individual wheels. Specifically, the total linear speed of the robot is given by the sum of the speeds of the left and right wheels divided by two. This approach assumes that both wheels are acting identically and that the robot is moving in a straight line, providing a simplified model of its forward speed. Substituting this gives us:

$$\begin{cases} \dot{x} = \frac{V_r+V_l}{2}.\cos(\theta_R) \\ \dot{y} = \frac{V_r+V_l}{2}.\sin(\theta_R) \\ \dot{\theta} = \frac{V_r-V_l}{2L} \end{cases} \quad (2)$$

IV. DISCRETE MODEL FOR ROBOT MOTION

We use the discrete version of this model, which is expressed as follows: the robot's motion is represented in discrete time steps, where the state of the robot at each step is updated based on the control inputs (such as wheel speeds) and the system's dynamics. This approach allows us to approximate the continuous model in a computationally feasible way, with the state being updated at each time interval according to the robot's kinematic equations.

$$\begin{cases} x_{k+1} = x_k + \frac{V_{r_k}+V_{l_k}}{2}.T.\cos(\theta_{R_k}) \\ y_{k+1} = y_k + \frac{V_{r_k}+V_{l_k}}{2}.T.\sin(\theta_{R_k}) \\ \theta_{R_{k+1}} = \theta_{R_k} + T.\frac{V_{r_k}-V_{l_k}}{2L} \end{cases} \quad (3)$$

V. DISTANCE AND ANGLE COMPUTATION BETWEEN ROBOT AND TARGET

To calculate the distance between the robot and its target, we use the Euclidean distance formula. This formula computes the straight-line distance between two points in a 2D plane, where the robot's current position and the target's position are represented by their respective coordinates (x_R, y_R) for the robot and (x_T, y_T) for the target.

The Euclidean distance D; which is a crucial parameter for the fuzzy logic controller to adjust the robot's movement towards the target effectively.

D_{RT} is given by:

$$D_{RT} = \sqrt{e_{RTx}^2 + e_{RTy}^2} \quad (4)$$

Where:

e_x: The error between robot x_R and target x_T.

e_y: The error between robot y_R and target y_T.

$$\begin{cases} e_{RTx} = x_T - x_R \\ e_{RTy} = y_T - y_R \end{cases} \quad (5)$$

To calculate the direction or angle between the robot and the target, we need to determine the relative angle at which the robot should turn to face the target. This angle, denoted θRT, can be calculated using the inverse tangent function. The robot's direction is represented by the angle θR (the angle between the robot's direction and the reference axis), while the target's position is given by the coordinates (x_T, y_T) and the robot's current position by (x_R,y_R).

$$\theta_T = \tan^{-1}\left(\frac{e_{RTy}}{e_{RTx}}\right) \quad (6)$$

The direction angle θRT can be calculated as follows:

$$\theta_{RT} = \theta_T - \theta_R \quad (7)$$

VI. STRUCTURES OF ROBOT-TRGET FLC-RT

The fuzzification, inference, and defuzzification processes are applied in the navigation behavior (Fig.2); using two inputs: the distance between the mobile robot and the target (D_{RT}), and the angle between the robot's current orientation and the target's orientation (θ_{RT}). The controller's outputs are the velocities of the left (Vl) and right (Vr) wheels.

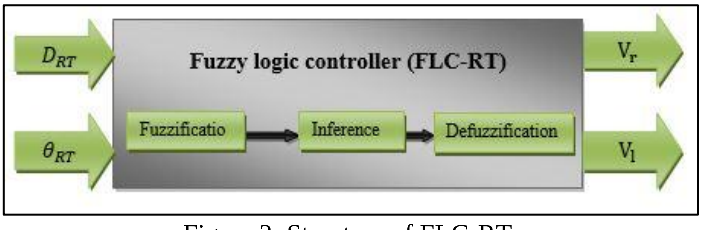


Figure 2: Structure of FLC-RT.
Source: Authors (2025).

VII. DISTANCE AND ANGLE COMPUTATION BETWEEN ROBOT AND OBSTACLE

To calculate the distance between the robot and an obstacle, we use the Euclidean distance formula. This formula computes the straight-line distance between two points in a 2D plane, where the robot's current position is represented by coordinates (x_R,y_R) and the obstacle's position is given by (x_O,y_O). The Euclidean distance D_{RO} is a key parameter for the fuzzy logic controller to adjust the robot's movement and avoid collisions effectively.

The distance D_{RO} is given by:

$$D_{RO} = \sqrt{e_{ROx}^2 + e_{ROy}^2} \quad (8)$$

Where:

e_{ROx}: The error between robot x_R and obstacle x_O.

e_{ROy}: The error between robot y_R and obstacle y_O.

$$\begin{cases} e_{ROx} = x_O - x_R \\ e_{ROy} = y_O - y_R \end{cases} \quad (9)$$

To calculate the direction or angle between the robot and an obstacle, we need to determine the relative angle at which the robot should turn to face the obstacle. This angle, denoted θ_{RO}, can be calculated using the inverse tangent function. The robot's direction is represented by the angle θ_R (the angle between the robot's direction and the reference axis), while the obstacle's position is given by the coordinates (x_O,y_O) and the robot's current position by (x_R,y_R).

$$\theta_{RO} = \tan^{-1}\left(\frac{e_{ROy}}{e_{ROx}}\right) \quad (10)$$

VIII. STRUCTURES OF ROBOT-OBSTACLE FLC-RO

The fuzzification, inference, and defuzzification processes are applied in the obstacle avoidance behavior (Fig. 3), using two inputs: the distance between the mobile robot and the obstacle (DRO), and the angle between the robot's current orientation and the obstacle's orientation (θRO). The controller's outputs are the velocities of the left (Vl) and right (Vr) wheels.

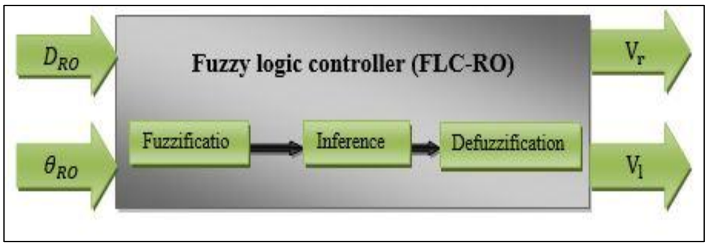


Figure 3: Structure of FLC-RO.
Source: Authors (2025)

IX. BLURRED NAVIGATION CONTROLLER STRUCTURE WITH OBSTACLE AVOIDANCE

The fuzzing, reasoning, and de-fuzzification operations are implemented using two separate fuzzy logic controllers (FLCs) for the navigation and obstacle avoidance behaviors in the system (Figure 4). The FLC for robot and target navigation (FLC-RT) uses two inputs: the distance between the moving robot and the target (DRT) and the angle between the robot's current direction and the target direction (θRT). The controller's outputs are the left (Vl) and right (Vr) wheel speeds, which guide the robot toward its target position. When the robot is navigating toward the target, the FLC-RT is active, ensuring efficient movement. If an obstacle is detected, the system switches to the FLC for robot obstacle avoidance (FLC-RO), which uses the distance between the robot and the obstacle (DRO) and the angle between the robot's current direction and the obstacle direction (θRO) as inputs. The FLC-RO outputs adjust the robot's movement to avoid collisions while continuing to move toward its target. This dynamic switching ensures smooth and safe navigation, while balancing the two goals of reaching the target and avoiding obstacles.

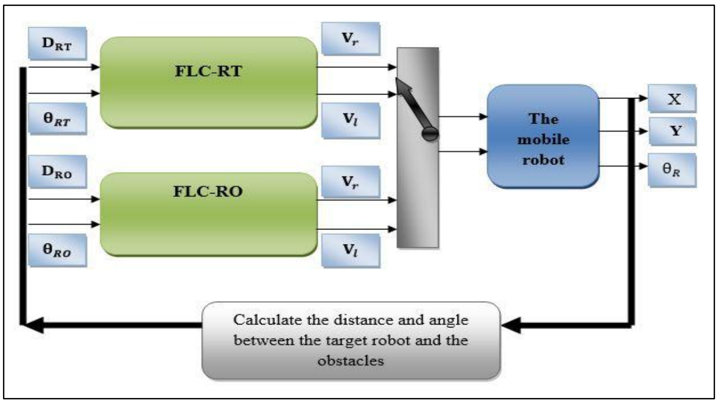


Figure 4: Structure of the navigation controller with obstacle avoidance.
Source : Authors (2025)

Then, the selected fuzzy logic controllers, as shown in (Fig.5) and (Fig. 6), should be designed to ensure that the distances D_{RT} and θ_{RT} between the robot and the target are minimized, i.e., ensure that $D_{RT} \rightarrow 0$, $\theta_{RT} \rightarrow 0$ when $t \rightarrow \infty$. It takes D_{RT} and θ_{RT} as inputs and produces Vl and Vr as outputs. The same is true for obstacle avoidance in Figure 4 because the inputs are : D_{RO} and θ_{RO}

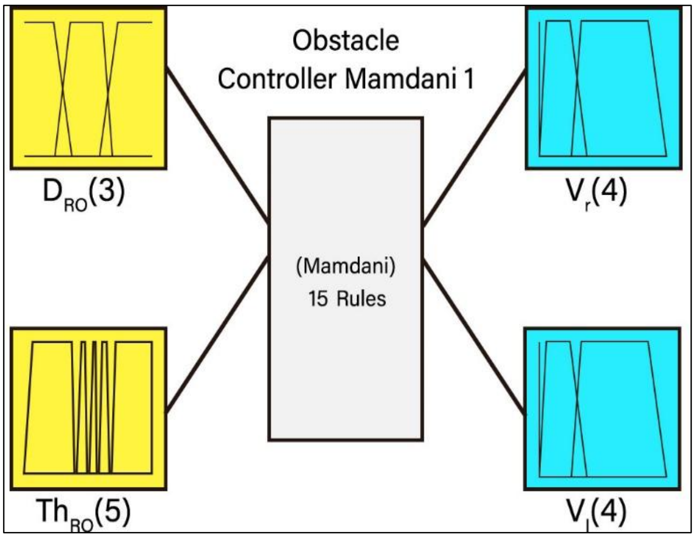


Figure 5 : Fuzzy rules set of the Target FLC-RT
Source : Authors (2025)

X. FUZZY LOGIC RULE TABLES.

We introduced fuzzy rules according to several experiments from which we extracted the following tables:
The rule base for the target fuzzy controller FLC-RT is specified in Table 1.

Table 1: Fuzzy rule sets of the target FLC-RT.

D_{RT}/θ_{RT}	NB	N	Z	P	PB
S	Z/Z	Z/Z	Z/Z	Z/Z	Z/Z
M	F/UM	F/M	M/M	M/F	F/UM
B	F/UM	F/M	F/F	M/F	F/UM

Source: Authors (2025).

The linguistic variables for the inputs in this controller are:
D_{RT} = (**S**:Small, **B** :Big, **Z** :Zero)
θ_{RT} = (**NB :** Negativebig, **N :** Negative, **Z :** Zero,
P : Positive, **PB :** Positivebig)
The linguistic variables for the outputs in this controller are:
Vr = **F:** Fast, **M :** Medium, **UM :** Under-Medium, **Z :** Zero.
V_l = **F:** Fast, **M :** Medium, **UM :** Under-Medium, **Z :** Zero.
Examples for this fuzzy control rule is:
if θ_{RT} is N and D_{RT} is M, then v_r is F and v_l is M
if θ_{RT} is Z and D_{RT} is S, then v_r is Z and v_l is Z

The rule base for the obstacle avoidance FLC-RO fuzzy controller is specified in Table 1.

Table 2: FLC-O fuzzy rule set for obstacle avoidance.

D_{RO}/θ_{RO}	NB	N	Z	P	PB
S	M/F	Z/M	Z/F	M/Z	F/M
M	F/F	M/M	M/M	M/M	F/F
B	F/F	F/F	F/F	F/F	F/F

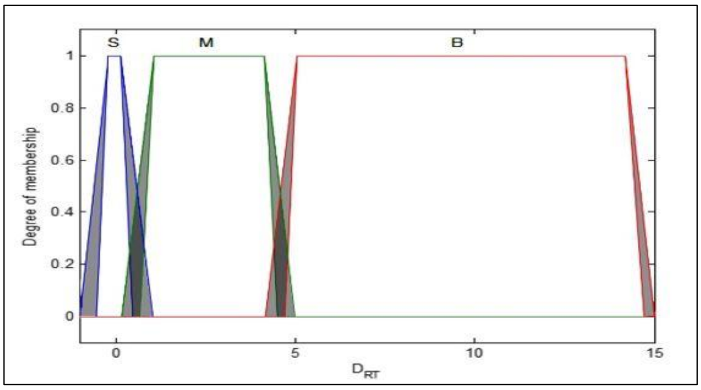
Source: Authors (2025).

The linguistic variables for the inputs in this controller are:
D_{RO} = (**S:** Small, **M:** Medium, **B :** Big)

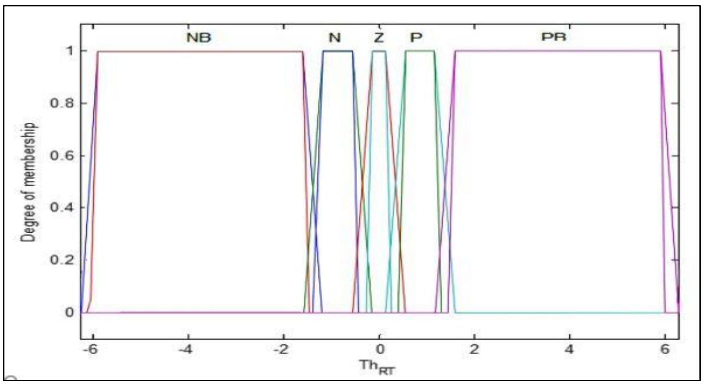
θ_{RO} = (**NB:** Negativebig, **N:** Negative, **Z:** Zero,
P : Positive, **PB :** Positivebig)
The linguistic variables for the outputs in this controller are:
Vr = **F:** Fast, **M :** Medium, **Z :** Zero.
Vl = **F:** Fast, **M :** Medium, **Z :** Zero.
Examples for this fuzzy control rule is:
• if θ_{RO} is P and D_{RO} is B, then v_r is F and v_l is F
• if θ_{RO} is PB and D_{RO} is S, then v_r is F and v_l is M

XI. MEMBERSHIP FUNCTION

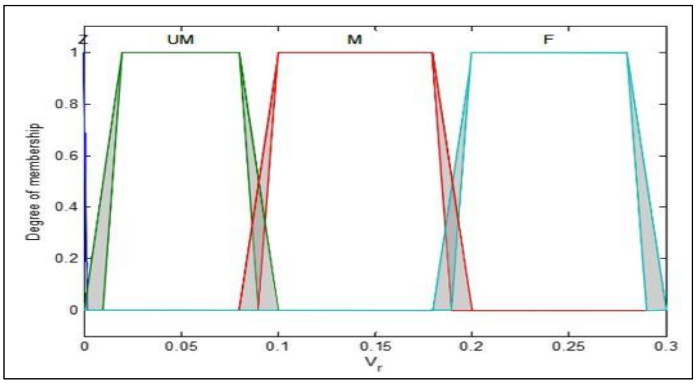
We present a series of schematic diagrams that show the membership functions (Figure 6-a, 6-b, 6-c, 6-d) of FLC-RT and (Figure 7-a, 7-b, 7-c, 7-d) of FLC-RO ; and for both the inputs and outputs of a fuzzy logic controller used in a mobile robot navigation system. These diagrams are essential for visualizing how a fuzzy logic system interprets different sensor inputs and translates them into control actions for the robot. The membership functions determine the degree of truth for different input values, allowing the system to make decisions based on fuzzy rules rather than binary logic.



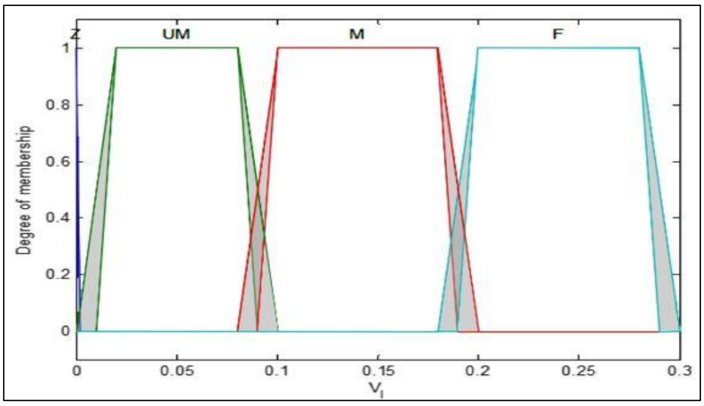
(a): MF-INPUT1-D_{RT}.



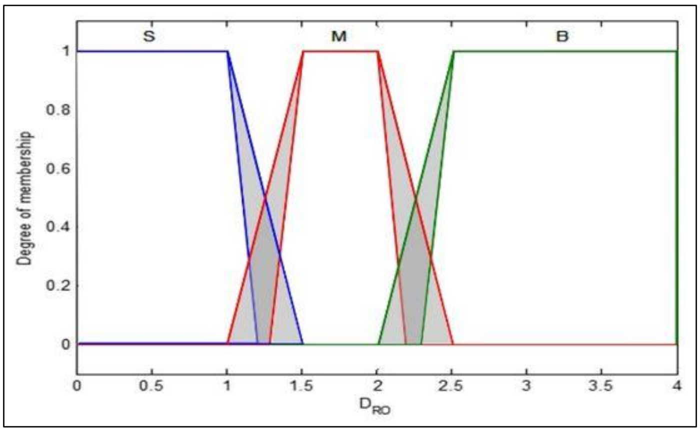
(b): MF-INPUT2- θ_{RT}



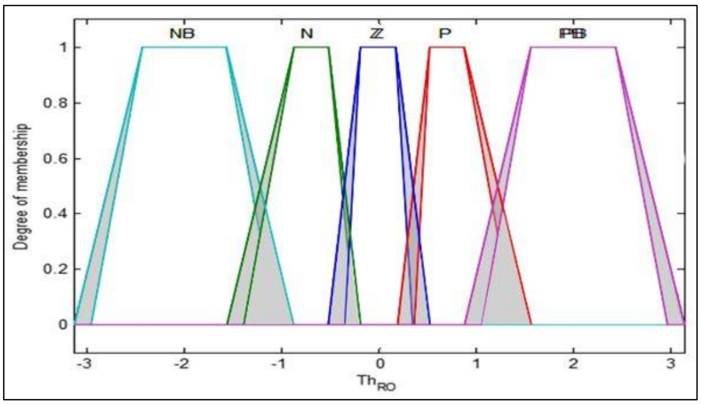
(c): MF-OUTPUT1-V_r



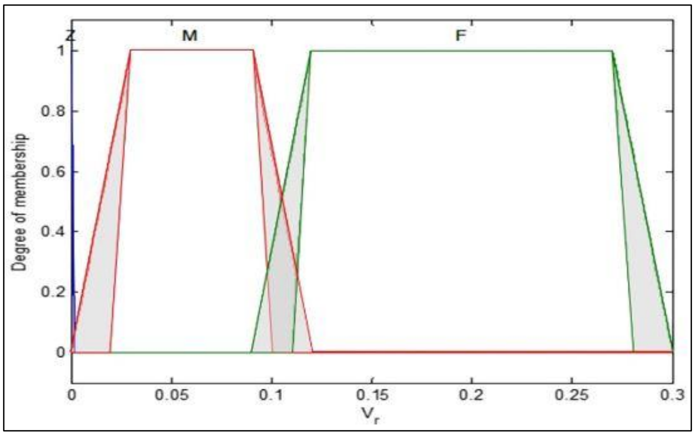
(d): MF-OUTPUT2-V_l
Figure 6: Membership function of the FLC-RT input/output variables.
Source: Authors (2025)



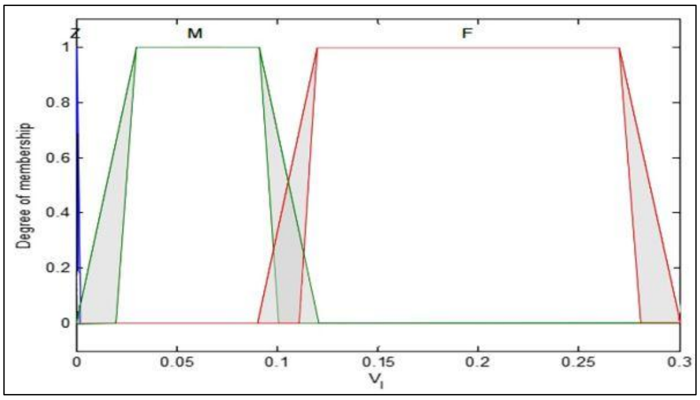
(a): MF-INPUT1-D_{RO}



(b): MF-INPUT2- θ_{RO}



(c): MF-OUTPUT1-V_r



(d): MF-OUTPUT2-V_l

Figure 7: Membership function of the FLC-RO input/output variables.

Source: Authors (2025)

XII. OUTDOOR LIGHTING

The mobile robot employed in our experiments is the Pioneer 3-DX, depicted in (Figure 8-a). This robot is a compact, lightweight, two-wheel, two-motor differential-drive system, making it ideal for indoor laboratory or classroom environments. The wheel axle length for this model is L = 46. It is equipped with a front SONAR sensor, a battery, wheel encoders, and a microcontroller running ARCOS firmware. The Advanced Robotics Interface for Applications (ARIA) serves as an effective platform for integrating user control software, as it efficiently handles low-level client-server interactions, including serial communication, command and status packet processing, cycle timing, multi-threading, and accessory control management. The experiments were conducted with a sampling time of T = 0.3 s.

Figure 8.a: Mobile robot for test *P-3DX*.

Source : Authors (2025)

The robot's position is determined using optical quadrature encoders. The test prototype is equipped with eight sonar sensors, numbered as shown in (Figure 8-b). The measured distance to the nearest obstacle is determined as the minimum value from all sensors, expressed as:

$$dr=\min(d1, d2, \ldots, d8) \qquad (11)$$

where :

di represents the distance to the obstacle measured by the i-th ultrasonic sensor. The angle between each consecutive pair of sensor directions is 20 degrees, except for the four side sensors (so0, so7, so8, and so15), where the angle between them is 40 degrees.

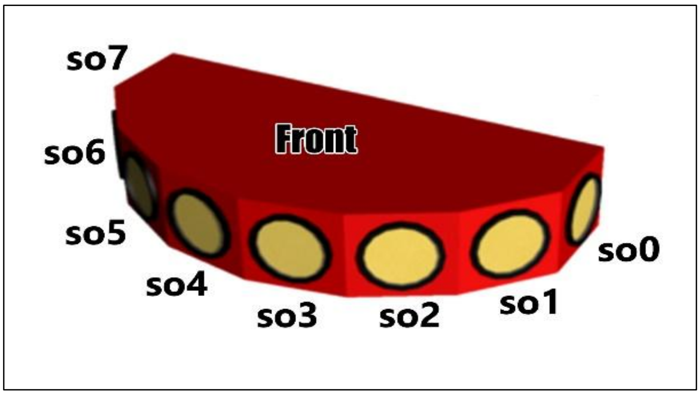
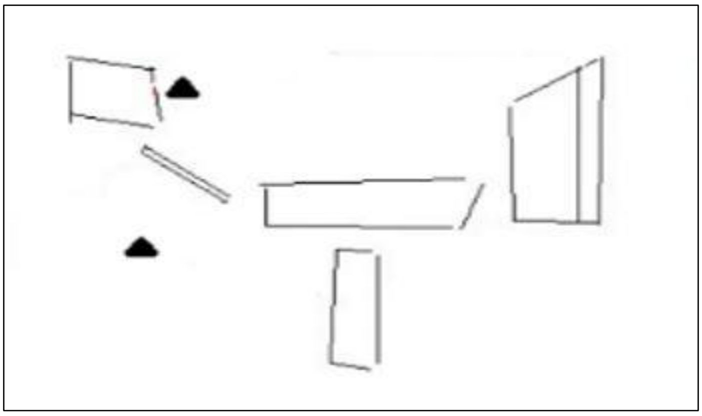


Figure 8.b: Sonars

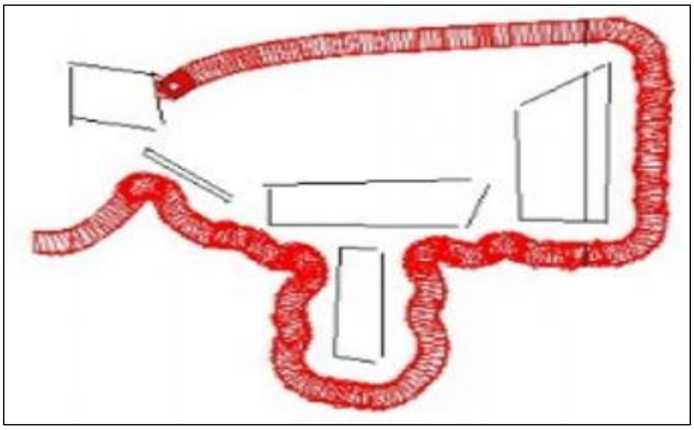
Source : Authors (2025)

XIII. ENVIRONMENT WITH STATIONARY OBSTACLES

In the first test, we tackled the challenge of moving the robot from an initial position, defined by SP = [0, 0], to a target position SP = [3, 2] (represented by black-filled triangles) within a crowded environment with five fixed polygonal obstacles, as shown in (Figure 9-a). It is important to note that the target configuration is located within a dangerous area due to its proximity to the obstacles. The results of applying the proposed method are presented in (Figure 9-b). These results highlight the effectiveness of the proposed approach in guiding the robot towards its target configuration while adhering to the kinematic constraints and successfully avoiding collisions with the obstacles.



(a): Test-1 environment



(b): Navigation result.

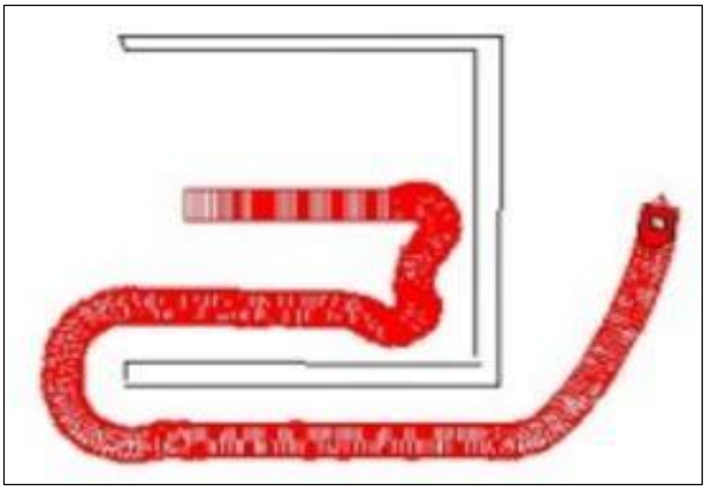
Figure 9: Test-1 result.

Source: Authors (2025).

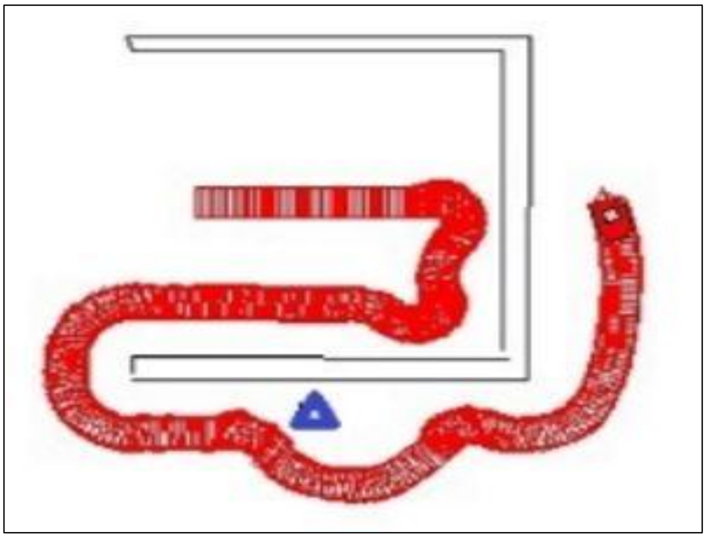
XIV. ENVIRONMENT WITH DYNAMIC OBSTACLES

The second test, the results of which are shown, was conducted in the workspace environment shown in (Figure 10-a). This environment has a U-shaped (concave) structure, which poses a significant challenge in the field of robotics, as many interactive navigation methods are prone to getting stuck at local minima. The results highlight the effectiveness of the proposed method in overcoming the local minima problem. In addition.

The same scenario was tested with a dynamic obstacle, represented by a blue triangle in (Figure 10-b). The results from this scenario also confirm the robustness of the proposed approach in dealing with dynamic obstacles.



(a): Navigation in U-shaped.



(b): Navigation in dynamic environment.
Figure 10: Test-2 result
Source : Authors (2025)

XV. CONCLUSIONS

This paper presents a local navigation strategy aimed at ensuring the safe operation of mobile robots in dynamic and uncertain environments. The approach is inspired by human reasoning and involves the creation of two behavioral planners utilizing a type-2 fuzzy logic controller. The first controller guides the robot towards its target, while the second controller directs the robot away from obstacles. A detailed, step-by-step explanation of the developed controllers is provided. To demonstrate the effectiveness of the proposed method, numerical tests across various scenarios and environments are presented. Future work will focus on adapting the proposed approach to facilitate robot navigation in 3D environments.

XVI. AUTHOR'S CONTRIBUTION

Conceptualization: Soufiane Hachani and Emira Nechadi.
Methodology: Soufiane Hachani and Emira Nechadi.
Investigation: Soufiane Hachani and Emira Nechadi..
Discussion of results: Soufiane Hachani and Emira Nechadi.
Writing – Original Draft: Soufiane Hachani.
Writing – Review and Editing: Soufiane Hachani and Emira Nechadi.
Resources: Emira Nechadi.
Supervision: Emira Nechadi.
Approval of the final text: Soufiane Hachani and Emira Nechadi.

XVII. DISCLAIMER

The authors declare that they received no financial support or grants from any public, commercial, or non-profit entities for this research. All the views expressed in this work are solely those of the authors.

XVIII. REFERENCES

[1] ABDESSEMED, F.; BENMAHAMMED, K.; MONACELLI, E. A fuzzy-based reactive controller for a non-holonomic mobile robot. *Robotics and Autonomous Systems*, v. 47, n. 1, p. 31–46, 2004. DOI: 10.1016/j.robot.2004.02.006.

[2] ADEPT TECHNOLOGY, Inc. Pioneer 3-DX Robot: Operation Manual. Livermore: Adept Technology, 2010. 120 p.

[3] AIT SAADI, A.; SOUKANE, A.; MERAIHI, Y.; BENMESSAOUD GABIS, A.; MIRJALILI, S.; RAMDANE-CHERIF, A. UAV path planning using optimization approaches: A survey. *Archives of Computational Methods in Engineering*, v. 29, n. 6, p. 4233–4284, 2022. DOI: 10.1007/s11831-022-09742-7.

[4] AKKA, K.; KHABER, F. Optimal fuzzy tracking control with obstacles avoidance for a mobile robot based on Takagi-Sugeno fuzzy model. *Transactions of the Institute of Measurement and Control*, v. 41, n. 10, p. 2772–2781, 2018. DOI: 10.1177/0142331218811462.

[5] AL-KHAZRAJI, A.; ZAYTOON, J. Observer-Based indirect adaptive sliding mode control design and implementation for a class of nonlinear systems. In: *NONLINEAR Estimation and Application*. New York: Nova Science Publishers, 2012. P. 229.

[6] AMES, A. D. et al. Control barrier functions: Theory and applications. In: 2019 18th European Control Conference (ECC). IEEE, 2019. DOI: 10.23919/ecc.2019.8796030.

[7] BOUTALBI, O.; BENMAHAMMED, K.; BOUKEZATA, B. An adaptive finite-time stable control law for manipulator robots with unknown parameters. *International Journal of Robust and Nonlinear Control*, v. 31, n. 11, p. 5218–5243, 2021. DOI: 10.1002/rnc.5538.

[8] BOUTALBI, O.; BENMAHAMMED, K.; HENNI, K.; BOUKEZATA, B. A high-performance control algorithm based on a curvature-dependent decoupled planning approach and flatness concepts for non-holonomic mobile robots. *Intelligent Service Robotics*, v. 12, n. 2, p. 181–196, 2019. DOI: 10.1007/s11370-018-00270-7.

[9]CHOSET, H.; LYNCH, K.; HUTCHINSON, S.; KANTOR, G.; BURGARD, W.;KAVRAKI, L.; THRUN, S. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.

[10] HAIDER, M. H. et al. Robust mobile robot navigation in cluttered environments based on hybrid adaptive neuro-fuzzy inference and sensor fusion. *Journal of King Saud University - Computer and Information Sciences*, v. 34, n. 10, Part B, p. 9060–9070, 2022. DOI: 10.1016/j.jksuci.2022.08.031.

[11] KARNIK, N. N.; MENDEL, J. M.; LIANG, Q. Type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems*, v. 7, n. 6, p. 643–658, 1999.

[12] KHATIB, O. Real-time obstacle avoidance for manipulators and mobile robots. In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, v. 2, p. 500–505. Institute of Electrical and Electronics Engineers, 1985. DOI: 10.1109/robot.1985.1087247.

[13] LAVALLE, S. *Planning Algorithms*. Cambridge University Press, 2006.

[14] LIU, L.; WANG, X.; YANG, X.; LIU, H.; LI, J.; WANG, P. Path planning techniques for mobile robots: Review and prospect. *Expert Systems with Applications*, v. 227, p. 120254, 2023. DOI: 10.1016/j.eswa.2023.120254.

[15] MATHWORKS. Type-2 fuzzy inference systems. Available at: https://la.mathworks.com/help/fuzzy/type-2-fuzzy-inference-systems.html. Accessed on: Nov. 28, 2024.

[16] MENDEL, J. M. *Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions*. 2nd ed. Springer, 2017. DOI: 10.1007/978-3-319-51370-6.

[17] NGUYEN, T. T. V.; PHUNG, M. D.; TRAN, Q. V. Behavior-based navigation of mobile robot in unknown environments using fuzzy logic and multi-objective optimization. *International Journal of Control and Automation*, v. 10, n. 2, p. 349–364, 2017. DOI: 10.14257/ijca.2017.10.2.29.

[18] OUADAH, N.; OURAK, L.; BOUDJEMA, F. Car-like mobile robot oriented positioning by fuzzy controllers. *International Journal of Advanced Robotic Systems*, v. 5, n. 3, p. 25, 2008. DOI: 10.5772/5603.