INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 19, Issue: 3, Month: June, Year: 2024
Article Number: 6420, https://doi.org/10.15837/ijccc.2024.3.6420

IT-2 Fuzzy Control and Behavioral Approach Navigation System for Holonomic 4WD/4WS Agricultural Robot

H. Ait dahmad, H. Ayad, A. García Cerezo, H. Mousannif

Hayat Ait dahmad*
Department of Applied Physics
Cadi Ayyad University, Marrakech, Morocco
*Corresponding author: hayat.aitdahmad@ced.uca.ma

Hassan Ayad
Department of Applied Physics
Cadi Ayyad University, Marrakech, Morocco
h.ayad@uca.ma

Alfonso García Cerezo
Department of Systems Engineering and Automation
University of Malaga, Malaga, Spain
gcerezo@ctima.uma.es

Hajar mousannif
Department of Informatics
Cadi Ayyad University, Marrakech, Morocco
mousannif@uca.ac.ma

Abstract

In contemporary agriculture, researchers are focusing on precision farming and environmentally conscious techniques, leading to the advancement of autonomous agricultural mobile robots. Consequently, A navigation system using TSK Interval Type-2 fuzzy logic has been developed and, for the first time, applied to a holonomic mobile agricultural robot equipped with four-wheel drive and steer. This system employs a behavioral approach to accomplish the robot's primary objective in unfamiliar environments. It comprises four controllers: one exclusively dedicated to goal-seeking behavior, with the remaining controllers responsible for obstacle avoidance. Supervision of these controllers is conducted through the Mandani fuzzy logic approach, which combines the outputs of all four controllers, robot wheel velocities and steering angles, using a proposed equations.
Various case studies were examined and simulated utilizing MATLAB and VREP software platform to assess the navigator system's performance. The simulation outcomes showcase the efficiency of the proposed equations and the system's navigation capabilities. The stability of the entire system was thoroughly examined and confirmed, employing the Lyapunov function, in environments with and without obstacles.

Keywords: TSK-IT2 fuzzy logic, Mandani-fuzzo logic, Supervisor, behavior approach.

1 Introduction

Over recent years, a growing interest has emerged in the advancement of intelligent autonomous systems in the field of agricultural robotics. The main objective is to reduce the workload of farmers and to supplant them in the execution of repetitive, arduous, dangerous tasks or requiring a level of precision that exceeds human capabilities.

The principal challenge confronting these robots revolves around the development of intelligent navigation systems. These systems must facilitate the robots' ability to autonomously traverse unstructured environments without human aid or intervention while attaining a pre-defined objective.

Robot navigation is described as the task of computing motion sequences that allow the robot to maneuver through complicated and crowded situations. Consequently, the control system must exhibit a level of intelligence that enables it to govern the mobile robot's actions while avoiding collisions with objects in its proximity surroundings.

Various methodologies have been introduced to address the navigational complexities of the robot. These include meta-heuristic methods, artificial intelligence techniques, fuzzy systems, and hybrid systems. In terms of meta-heuristic methodologies, the author [1] introduced a fusion of two distinct meta-heuristic approaches to enhance the trajectory optimization of a mobile robot. this fusion aims to ensure the robot's navigation in a partially known environment with static obstacles. The employed strategy yielded significant outcomes in terms of trajectory optimization, as demonstrated by the obtained results. In [2] an innovative hybrid approach integrating Particle Swarm Optimization (PSO) with Self-Organizing Migration Algorithm (SOMA) has been proposed. This approach aims to achieve optimal trajectory planning and control for individual or multi-mobile robots in either static or dynamic work environments. Researchers [3] proposed a novel methodology for deriving optimal collision-free trajectories for robots navigating in cluttered environments. This approach involves a hybridization of improved particle swarm optimization (IPSO) and sine and cosine algorithms (SCA). The effectiveness of the algorithm has been established through extensive evaluations carried out both in simulations and in real working environments. A multi-objective genetic algorithm-based trajectory search method is proposed in [4] to deal with the problem of trajectory planning for mobile robots. The results demonstrate that the suggested method performs significantly in terms of selecting the optimal trajectory with reduced time complexity. In [5], an original approach to global trajectory planning is introduced. This approach is based on the use of a meta-heuristic method known as the ant colony algorithm. The heuristic parameters of this approach are adjusted via a fuzzy logic controller, with the aim of deriving an optimal global trajectory for a mobile robot in complex, static environments.

Type-1 fuzzy logic is also recognized as a powerful intelligence strategy used in the context of mobile robots to navigate unfamiliar environments. Its ability to process sensor measurements or real-world data, which often include errors, noise, and variation, makes it a valuable asset for optimal navigation. By employing linguistic rules and fuzzy logic controllers, mobile robots can plan their motion, circumvent obstacles, and navigate efficiently. This approach has outperformed other intelligent path-planning techniques and offers a reliable solution to the navigation problem in mobile robotics [6][7] [8] [9]. Additionally, its application extends to various domains beyond robotics [10] [11] [12][13].

Nevertheless, in certain specific situations, Type-1 fuzzy logic may encounter difficulties in dealing with uncertainties, mainly due to its membership functions. FL Type-1 assumes that the membership functions are completely known and that there is no uncertainty in the system. Recognizing this limitation, the evolution of type-2 fuzzy logic systems (FL Type-2) became necessary. FL Type-2 introduces the concepts of "fuzzification uncertainty" and "defuzzification uncertainty" to handle uncertainty at a higher level. This enables Type-2 fuzzy logic systems to make more accurate decisions in complex and uncertain environments. For instance, the implementation of a Type-2 fuzzy logic controller for the autonomous navigation of a mobile unicycle robot showcased its impressive resilience in unfamiliar environments, a conclusion supported by simulation results [14].In contrast to Type 1 fuzzy controllers, the Type 2 controller has proven exceptionally robust and convincingly effective. An approach was introduced in [15] [16] for embedding a type-2 fuzzy logic system in an omnidirectional mobile robot. This system was designed to follow a designated path while adeptly avoiding obstacles that may appear within its environment. The simulation results effectively demonstrated the advantages of this innovative approach. To illustrate the ability of interval type 2 controllers to handle

uncertainties, a comparative study was presented in [17] concerning a wheeled robot. The designed controllers were subjected to tests that revealed that interval type 2 fuzzy logic (IT2FL) outperformed conventional fuzzy logic (FL) controllers. Researching the performance outcomes achieved with the two types of Fuzzy Logic Systems (FLSs) forms a significant area of investigation. Several authors have addressed this challenge in diverse applications, as evidenced in [18][19][20].
Prompted by the previously outlined challenges, This paper is intended to establish a technique that combines interval type 2 fuzzy logic and a behavioral approach to enhance the autonomous navigation of a 4WD/4WS holonomic mobile robot in unfamiliar environments. The structure of the article unfolds as follows: Section 2 provides an overview of the interval-2 fuzzy logic system. Section 3 includes robot kinematics and the IT-2 fuzzy logic proposed to establish goal-seeking and obstacle-avoidance behaviors, as well as a type-1 fuzzy logic-based approach to managing these behaviors. Validation of system stability using the Lyapunov function is discussed in section 4. Section 5 shows the validation of the proposed method by testing various scenarios. Finally, a conclusion summarizing the conducted work and highlighting future prospects for this research is presented in Section 6.

2 Fuzzy logic systems with type 2 interval modeling architecture

The development from type-1 to type-2 fuzzy logic sets, initiated by Lotfi Zadeh [21], marks an evolution in defining an element's degree of membership through a fuzzy membership function that operates within the [0,1] interval. This advancement provides a superior framework for modeling and addressing uncertainties, presenting a significant improvement over the capabilities of type-1 fuzzy sets.
The interval type-2 fuzzy set, represented as $\tilde{A}$, is recognized as a specialized subset of the general type-2 fuzzy set. Its design is particularly focused on reducing the computational demands inherent in handling uncertainties related to membership degrees. It is characterized by a three-dimensional membership function, and can be expressed as follows:

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} 1/(x, u) \tag{1}$$

where $x \in X$ and $u \in J_x \subseteq [0, 1]$ are the primary and secondary variables, J_x is the primary membership and the integral sign indicates the union of all possible values of x and u.
The IT-2 fuzzy system closely resembles a Type-1 fuzzy system in its basic structure, as depicted in Figure 1, comprising fuzzification, a rule base, an inference mechanism, and output processing. However, a key distinction is the inclusion of a "type reducer" in the IT-2 system, which converts its output fuzzy sets into Type-1 fuzzy sets before defuzzification, streamlining the data for further processing and decision-making.

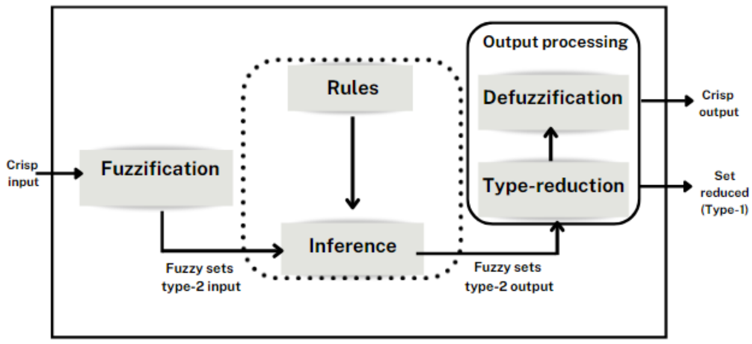


Figure 1: IT2-FLC architecture.

2.1 Fuzzification

The fuzzification interface is the initial step in fuzzy logic, converting crisp or physical values into Type-2 fuzzy sets. It does this by associating each input with an interval determined by two values: the lower membership value (LMF) on the left, derived from the lower membership function, and the upper membership value (UMF) on the right, obtained from the upper membership function. These intervals capture membership uncertainty.

2.2 Rule base and inference mechanism

As previously stated, the differentiation between a Type 1 and a Type 2 system is restricted to the characteristics inherent in the membership functions relating to premises and consequences. However, the structure of the rule remains unchanged. Consider N IT2 fuzzy rules. The i^{th} rule is expressed as follows:

$$\tilde{R}^i = IF\, x_1\, is\, \tilde{F}_1^i\, and\, x_2\, is\, \tilde{F}_2^i\, and\, ...\, and\, x_p\, is\, \tilde{F}_p^i\, , THEN\, y\, is\, \tilde{G}^i \tag{2}$$

Where the x_j are inputs, $\tilde{F}_j^i$ are antecedent sets such that (j = 1 . . ., p), y is the output, and $\tilde{G}^i$ is the consequent set.
The inference mechanism of an IT-2 fuzzy system calculates an IT-2 output fuzzy set according to each rule. The fuzzy rule base 2 is then used to perform a relation between an input vector and a scalar output.

2.3 Type reduction and defuzzification procedures

The "output processing" block in fuzzy logic comprises two key steps. First, The "type reducer" reduces the output of the T2 fuzzy set generated by the inference system to a T1 fuzzy set, often referred to as the "centroid," by estimating a range defined by lower c_L and upper c_R limits. This involves iterative type-reduction techniques. Second, the "defuzzification interface" then refines the type-1 fuzzy set resulting from the interval to produce a precise outcome, using designated defuzzification algorithms. The defuzzified output can be calculated as follows:

$$y = \frac{c_L + c_R}{2} \tag{3}$$

3 Developed IT-2 fuzzy logic navigator system

An intelligent control system using Interval-2 Takagi-Sugeno-Kang (IT-2 TSK) fuzzy logic has been developed to facilitate robot navigation. The IT-2 TSK fuzzy logic model, a variant of the TSK fuzzy system, uses IT-2 fuzzy sets. First presented in [22], this model offers a new approach to generating the output of a fuzzy logic system. Unlike Mamdani's method, which relies on fuzzy sets to represent the output, the TSK model adopts mathematical functions for this purpose. In this model, each rule delimits the output through a linear equation, thus moving away from the conventional use of fuzzy sets to specify the output.
The primary objective of this developed system is to ensure the robot safely reaches its intended destination within an unknown environment without encountering collisions. To achieve this, we divided its overall mission into two sub-missions: the goal-seeking controller and the obstacle avoidance controller. The latter has been subdivided into three obstacle avoidance controllers: "front", "right" and "left".
Once the simulation is launched, all these controllers operate simultaneously. Consequently, we have implemented a supervisor using Type-1 Mandani fuzzy logic to evaluate their respective weights and establish priorities. This supervisor makes decisions based on the data collected by the robot's sensors to determine the most appropriate control strategy for the given circumstances. Figure 2 provides a visual representation of our navigation architecture.

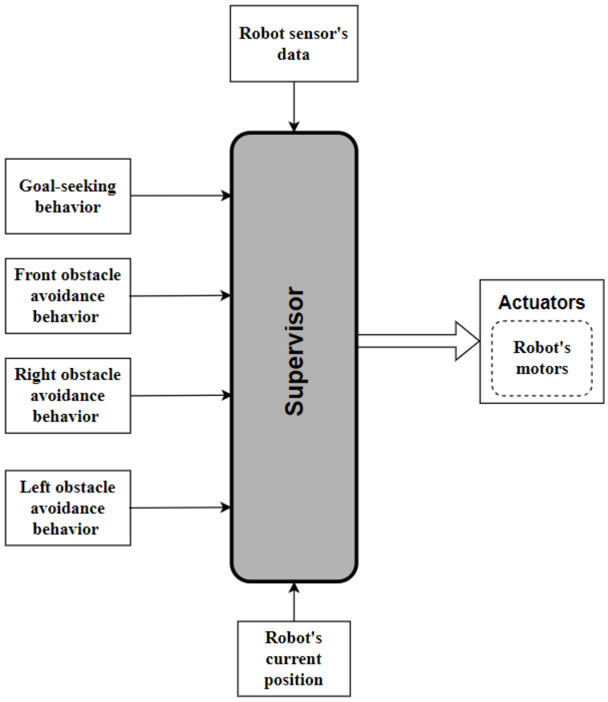


Figure 2: Navigator system architecture.

3.1 IT-2 fuzzy logic controller for goal-seeking

Before delving into the tasks our robot will undertake, let's return to the kinematic equations governing the holonomic mobile robot equipped with four-wheel drive and steer (4WD/4WS), as conveyed by the widely recognized equations:

$$\dot{X}_r = v\cos(\theta_r) \tag{4}$$

$$\dot{Y}_r = v\sin(\theta_r) \tag{5}$$

$$\dot{\theta}_r = \omega_r \tag{6}$$

Linear and angular velocities are obtained using the expressions below:

$$v = \frac{v_f \cos\delta_f + v_r \cos\delta_r}{2} \qquad \omega_r = \frac{v(\tan\delta_f - \tan\delta_r)}{L_r + L_f} \tag{7}$$

To navigate to the intended destination, two inputs are considered, as shown in Figure 3: the distance between the robot and the goal, referred to as e_{Pos}, and the angular error, denoted as e_{Ang}, which indicates the difference in orientation between the target and the robot.
Regarding the controller's output,we have identified two output variables: the velocity, represented by v and the steering angle, indicated by δ for both the front and rear wheels. The input variables are obtained by the following equations:

$$e_{Pos} = \sqrt{(X_T - X_r)^2 + (Y_T - Y_r)^2} \tag{8}$$

$$e_{Ang} = \arctan\left((Y_T - Y_r)/(X_T - X_r)\right) - \theta_r \tag{9}$$

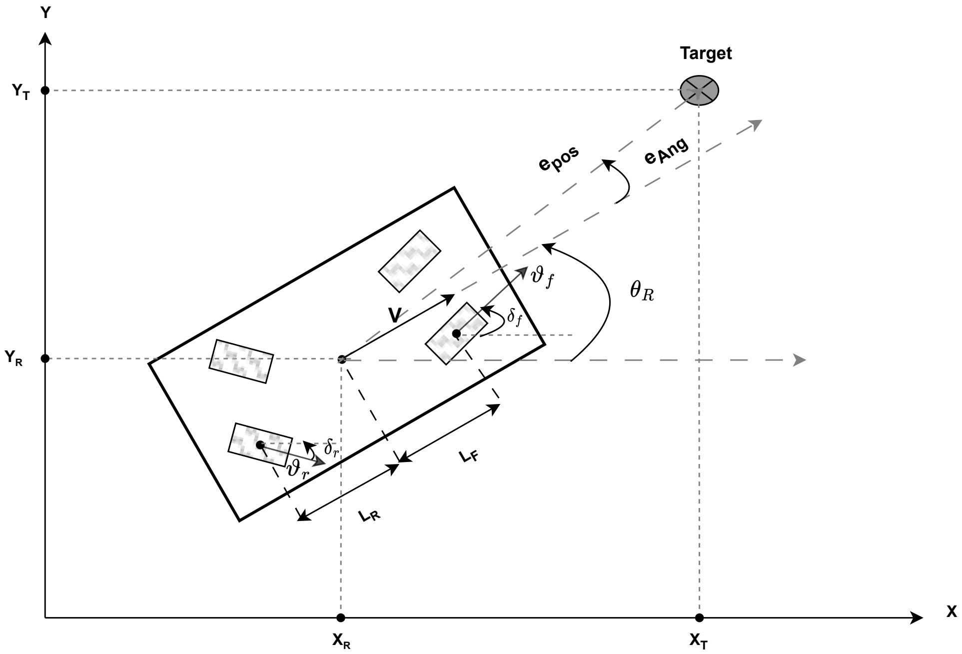
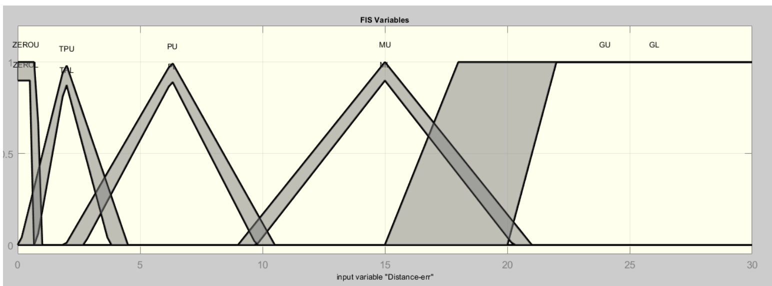


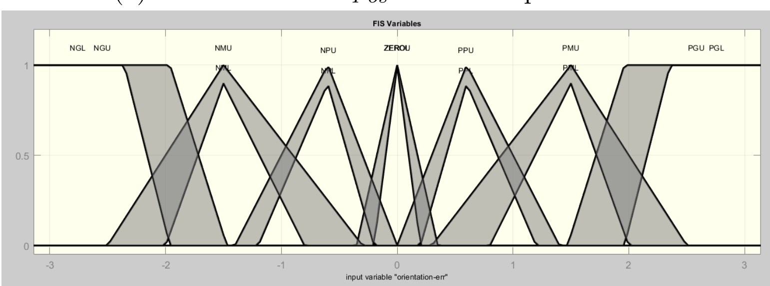
Figure 3: 4WD/4WS mobile robot model.

where (X_r, Y_r, θ_r) represents the present robot state , while (X_T, Y_T) denote the target position.

Considering the characteristics of the robot and the specifics of its operating environment, we determined the appropriate universe of discourse for all inputs and outputs. For each input, we used a mixture of trapezoidal and triangular membership functions. For the control values, we simplified them into singletons. Parameters for these membership functions were selected through a process of trial and error. This approach enabled us to adjust our parameters iteratively, finally identifying those that ensured that the robot would perform in the expected operational scenarios.
The membership functions of the inputs are depicted in Figures 4a and 4b, whereas the outputs are illustrated in Figures 5a and 5b.

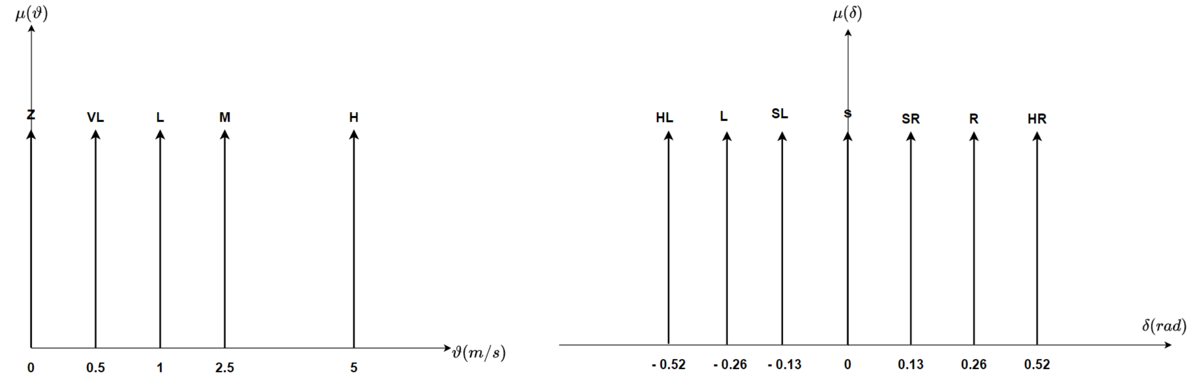


(a) Distance error e_{Pos} membership functions.



(b) Orientation error e_{Ang} membership functions.

Figure 4: IT-2 Fuzzy input membership functions.



(a) Wheel Velocity membership functions.

(b) Steering angleδ membership functions.

Figure 5: IT-2 Fuzzy output membership functions.

Positioning error is defined using five membership functions: B (Big), M (Medium), S (Small), VS (Very Small) and Z (Zero). Concurrently, the angular error is defined by a set of seven membership functions: PS (Positive Small), PM (Positive Medium), PB (Positive Big), NB (Negative Big), NM (Negative Medium), NS (Negative Small) and Z (Zero).
For the output linguistic fuzzy sets, they are represented as Z (Zero), VL (Very Low), L (Low), M (Moderate), H (High), SL (Slight Left), L (Left), HL (Hard Left), S (Straight), SR (Slight Right), R (Right), and HR (Hard Right).

In pursuit of the global goal, and after defining membership functions, we formulated fuzzy rule bases. Using the Takagi-Sugeno approach, we established connections between input and output variables. Table 1 presents an overview of these fuzzy rule bases. For instance, the entry at position (1, 1) in Table 1 can be expressed as follows:

$$IF\, e_{Pos}\, is\, Z\, and, e_{Ang}\, is\, NB\, , THEN\, \delta_f = S,\, \delta_r = S\, and\, v_f = v_r = L. \tag{10}$$

Table 1: Fuzzy rules designed for achieving the goal-seeking task.

				Orientation Error e_{Ang}						
				NB	NM	NS	Z	PS	PM	PB
Distance Error e_{Pos}	Z	δ	δ_f	S	S	S	S	S	S	S
			δ_r	S	S	S	S	S	S	S
		$v_f = v_r$		L	M	M	Z	M	M	L
	VS	δ	δ_f	HL	HL	L	S	R	HR	HR
			δ_r	HR	HR	R	S	L	HL	HL
		$v_f = v_r$		VL	VL	VL	VL	VL	VL	VL
	S	δ	δ_f	HL	HL	L	S	R	HR	HR
			δ_r	HR	HR	R	S	L	HL	HL
		$v_f = v_r$		L	L	L	L	L	L	L
	M	δ	δ_f	L	L	SL	S	SR	R	R
			δ_r	R	R	SR	S	SL	L	L
		$v_f = v_r$		L	M	M	M	M	M	L
	B	δ	δ_f	L	SL	SL	S	SR	SR	R
			δ_r	R	SR	SR	S	SL	SL	L
		$v_f = v_r$		M	H	H	H	H	H	M

3.2 IT-2 fuzzy logic controller for obstacle avoindance

Obstacle avoidance constitutes a fundamental behavior inherent in nearly all robotic applications. Its primary objective is to prevent any obstacle from impeding the robot's progress toward accomplishing its primary goal.
To enable our robot to recognize obstacles in various directions, we have partitioned its overall perception area into three distinct detection zones (as depicted in Fig.6). Within each zone, we have integrated three ultrasonic sensors to improve the accuracy of obstacle position assessment concerning our robot. These zones are defined as specified:

- Front obstacle avoidance (FOA) utilizes the distances d1, d2, and d3.
- Left obstacle avoidance (LOA) utilizes the distances d3, d7, and d5.
- Right obstacle avoidance (ROA) utilizes the distances d2, d6, and d4.

To navigate around obstacles, three controllers based on IT-2 fuzzy logic of the Takagi-Sugeno type have been developed. In all three controllers, the input is derived from the robot's proximity to obstacles, as measured by ultrasonic sensors. The same partitioning of the universe of discourse for these sensors has been employed.
Concerning the controller's output, we have defined the outputs to be the steering angle and wheel speed. These outputs are characterized by the same set of fuzzy sets utilized in the goal-seeking behavior.
The linguistic variable distance is defined by two fuzzy sets: C(Close) and F(Far) as shown in Figure 7.
The rules for all three zones are displayed in Table 2 .

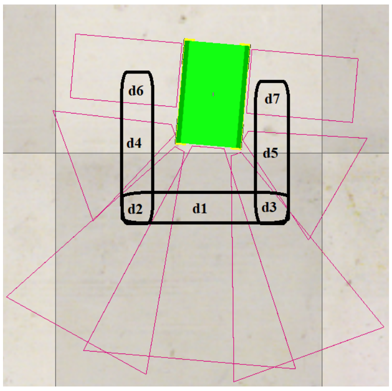


Figure 6: Position of the ultrasonic sensors.

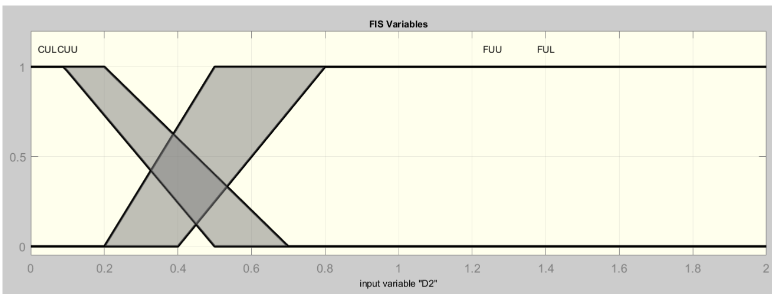


Figure 7: Membership Functions Describing Obstacle Distance.

Table 2: Fuzzy rules for the obstacle avoidance task in the front (a), right (b), and left (c) directions.

			d3			
			C		F	
			d1			
			C	F	C	F
d2	C	δ_f	HL	HL	HR	HR
		v_f	M	M	M	M
	F	δ_f	HL	L	HL	S
		v_f	M	M	M	Z

(a)

			d6			
			C		F	
			d4			
			C	F	C	F
d2	C	δ_f	HR	R	HR	R
		v_f	M	M	M	M
	F	δ_f	R	S	R	S
		v_f	M	Z	M	Z

(b)

			d7			
			C		F	
			d5			
			C	F	C	F
d3	C	δ_f	NG	NM	NG	NM
		v_f	M	M	M	M
	F	δ_f	NM	NM	NM	Z
		v_f	M	M	M	Z

(c)

3.3 The proposed Mandani-Method Fuzzy Logic Supervisor

A significant challenge arises when multiple behaviors compete for control of the robot. These conflicting behaviors often demand simultaneous control actions, especially in complex navigation scenarios. As a solution, we proposed developing a supervisor utilizing the Mandani-Fuzzy Logic technique. The purpose of this supervisor is to arbitrate between these behaviors. The core concept behind this approach is to assign priorities to each behavior based on the robot's state, achieved through weights determined by the supervisor.

We have established a prioritized order for the four behaviors. Goal-seeking behavior is of utmost importance and we have assigned it a weight represented by W = 1, which guarantees that it will always be executed. Frontal obstacle avoidance takes priority over the other two obstacle avoidance behaviors, as it represents a more immediate danger to the robot. Left and right behaviors are equally weighted. The control variables governing robot motion are calculated using the proposed equations 11 . These equations combine the outputs of each behavior controller, weighted according to the output of the supervisor.

$$\begin{aligned} \delta_f &= \delta_{f_{GS}} + W * \delta_{f_{FOA}} + (1-W) * \delta_{f_{ROA}} + (1-W) * \delta_{f_{LOA}} \\ v_f &= v_{f_{GS}} + W * v_{f_{FOA}} + (1-W) * v_{f_{ROA}} + (1-W) * v_{f_{LOA}} \\ \delta_r &= -\delta_f \\ v_r &= -v_f \end{aligned} \tag{11}$$

The implementation of the supervisor is summarized in Figure 8.

The input data for the fuzzy logic supervisor includes obstacle distances measured in three directions. These distances are determined using the following equations:

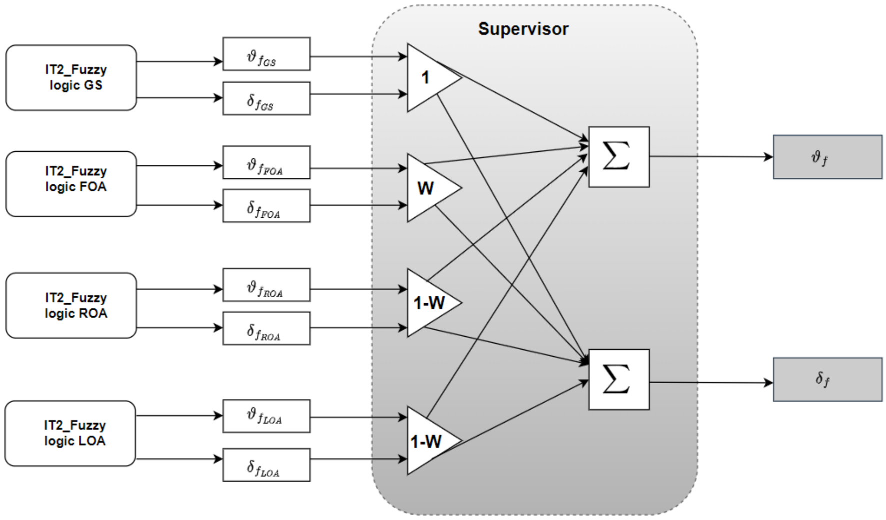


Figure 8: The suggested supervisor architecture.

$$
\begin{aligned}
d_{Front} &= min(d_1, d_2, d_3) \\
d_{Right} &= min(d_2, d_4, d_6) \\
d_{Left} &= min(d_3, d_5, d_7)
\end{aligned}
\tag{12}
$$

The distance between obstacles is described using three fuzzy sets: (VC) Very Close, (C) Close, and (F) Far. The membership functions for these sets are illustrated in Figure 9, and the weight membership functions can be seen in Figure 10. The weight is calculated based on the rules outlined in Table 3.

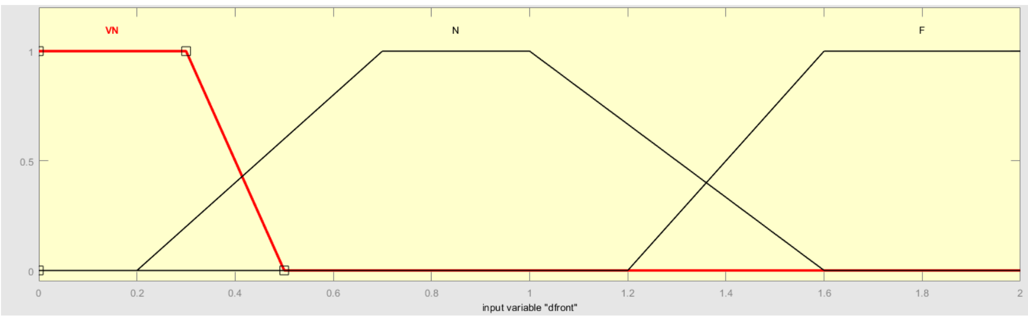


Figure 9: Obstacle distance fuzzy set.

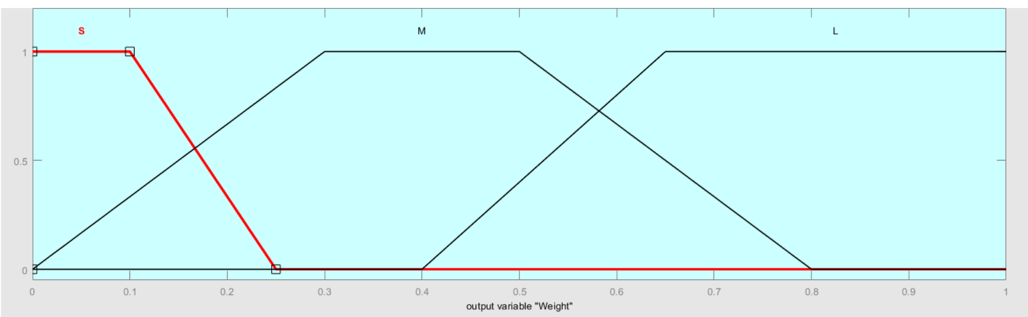


Figure 10: Weighting fuzzy set.

Table 3: Fuzzy rules for the supervisor.

			d_{Left}								
			VC			M			F		
			d_{Right}								
			VC	M	F	VC	M	F	VC	M	F
d_{Front}	VC	W	L	L	L	L	L	L	L	L	L
	M		L	L	L	L	L	L	L	L	L
	F		L	L	L	L	M	S	L	S	S

4 Stability verification using lyapunov function

In ensuring the robot's stability during its core mission of navigating toward a target in an environment with clustered obstacles, we conducted a stability evaluation of the entire control system using the Lyapunov method. Considering that we worked with four controllers and switched between them throughout the mission, there was a potential for introducing instability into the overall control system. Therefore, we selected the positive candidate equation based on quadratic form [23]:

$$V = \frac{1}{2}e_{Pos}^2 + \frac{1}{2}e_{Ang}^2 \tag{13}$$

Using Figure 3, an alternative representation for the position error can be derived:

$$e_{Pos} = v\cos\left(e_{Ang}\right) \tag{14}$$

After computing the derivative of the Lyapunov function by using the kinematic model and the equation14 , the result is as follows:

$$\dot{V} = [-v^2\cos\left(e_{Ang}\right)\sin\left(e_{Ang}\right) + e_{Ang}][\tan\left(e_{Ang}\right) + \frac{2v\tan\left(\delta_r\right)}{L_f + L_r}] \tag{15}$$

We can establish the negativity of the derivative of V by examining the results presented in Figures 15b and 20b. This confirms that $\dot{V} \leq 0$ both in environments without obstacles and in environments with obstacles, thus guaranteeing stability.

5 Simulation results

To evaluate the accuracy of the proposed approach to guiding a 4WD/4WS mobile robot over unstructured terrain, we first designed a model of this robot in SOLIDWORKS software with the necessary specifications. Subsequently, the model built, as shown in Figure 11, was exported from SOLIDWORKS in "URDF" format. This file was then imported into the Vrep simulator. The Virtual Robot Exploration Platform (Vrep) is recognized as an adaptable and comprehensive tool for the rapid development of 3D simulations. It features capabilities for modeling, modifying, programming, and simulating a robot in an environment that closely simulates real-world scenarios. Simulation results were produced on a computer configured with an Intel Core i7 processor, running at a speed of 2.4 GHz and equipped with 16 GB RAM, with Matlab 2017b installed as the software environment.

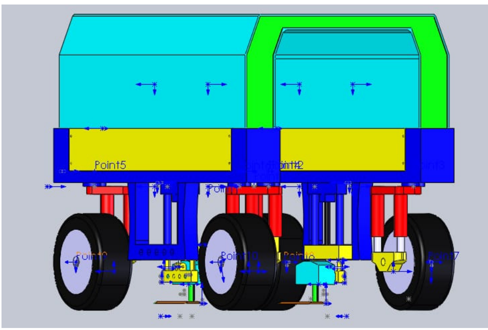


Figure 11: Robot modeling on the SOLIDWORKS platform.

5.1 Simulation in an obstacle-free environment

In this simulation, we have chosen the simplest scenario: navigating in an environment without obstacles. The purpose of this selection is to emphasize the robot's goal-seeking behavior and how it adjusts its steering angles and speed based on position and angle errors.
In Figure 12 , The robot must navigate from its initial position at coordinates ($x_r = -8.9m$, $y_r = -10.5m$) to the target point ($x_T = 10m$, $y_T = 10m$). It successfully achieved this goal, as evidenced by the illustration in Figure 13. The absence of obstacles meant that the predominant behavior observed throughout the simulation was goal-seeking.
At the start of the simulation, the robot decelerates to align itself with the target, as shown in Figure 14a. Once successfully aligned, the robot accelerates, slowly reducing its steering angle, as shown in Figure 14b. The robot then gradually slows down as it approaches the target. This gradual deceleration prepares the robot for a complete stop, ensuring a smooth and precise arrival at the target location.
With regard to stability, Figures 15a and 15b show that stability is maintained when only the goal-seeking behaviour is activated.

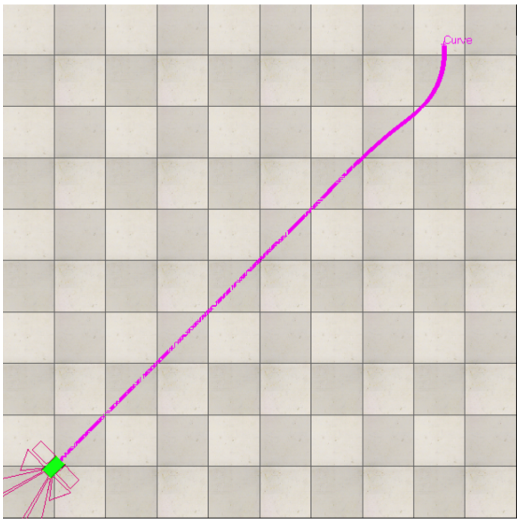


Figure 12: Robot's trajectory in an obstacle-free environment.

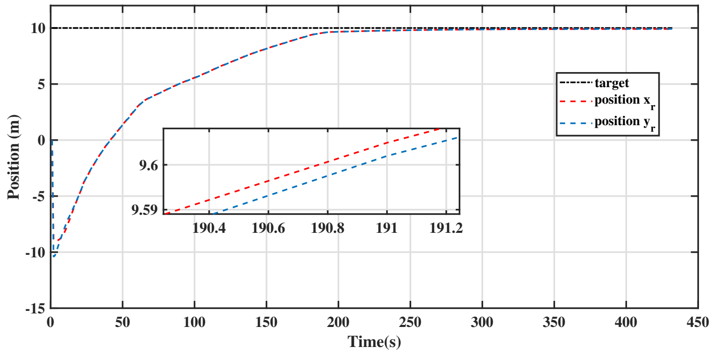
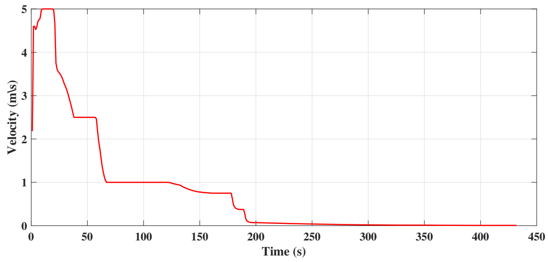
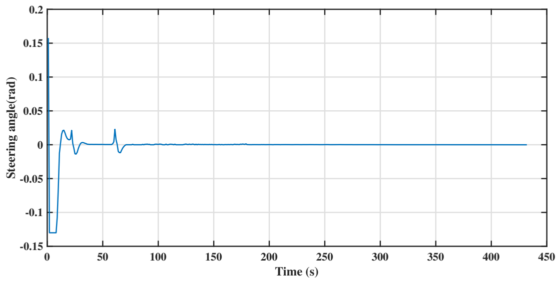


Figure 13: Separate robot trajectories in an obstacle-free environment.

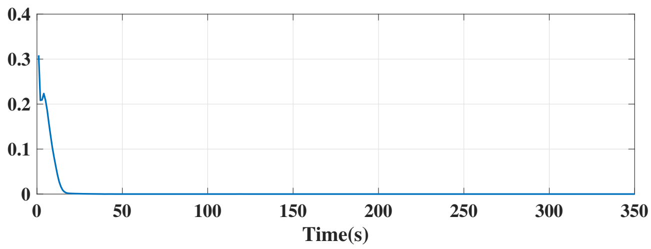


(a)

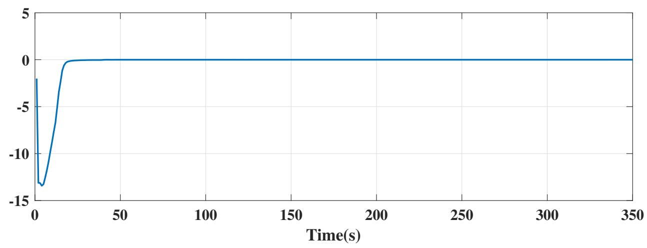


(b)

Figure 14: (a) The robot's velocity, (b) The steering angle during navigation.



(a)



(b)

Figure 15: (a) Lyapunov function , (b) derivative of Lyapunov function in an obstacle-free environment.

5.2 Simulation in an environment occupied by obstacles

The environment is filled with cylindrical and cubic obstacles, requiring the robot to navigate from its initial position ($x_r = 8.9m$, $y_r = 9m$) to its destination ($x_T = -10m$, $y_T = -10m$). The robot's trajectory comprises four distinct phases: AB, BC, CD and DE (see Figure 16).

During the AB phase, priority is given to target convergence behavior, which is activated at the start of the simulation. In section BC, the robot detects an obstacle on its left, and the left obstacle avoidance behavior is triggered. The same applies to the CD phase. When the robot approaches point D and detects an obstacle on its right, the corresponding behavior is activated. Finally, in the DE phase, the target convergence behavior again takes priority, as the path to the target becomes clear. This behavior continues until the robot stops at the target point.
In Figure 17 , environment n° 2 replicates environment n° 1, differing only in the adjusted coordinates of the starting and destination points. This adjustment was made to assess the proposed controller under various conditions. Remarkably, our robot adeptly maneuvered around the obstacles in this environment, ultimately achieving the designated objective.
In environment n° 3 (Figure 18), The robot begins its itinerary between two walls, using its sensors to detect obstacles on its right and left sides. Initially, the robot alternates between right and left obstacle avoidance behaviors until it identifies a wall directly in front of it. In this case, the front obstacle avoidance behavior is activated. With the target behind this wall, the robot follows the wall using the left obstacle avoidance behavior until its left sensors no longer detect the wall. At this point, the robot smoothly adjusts its direction towards the target. Throughout this process, the goal convergence behavior takes priority and guides the robot until it reaches the target position.
In environment n° 4 the robot's objective is to navigate along a corridor, avoiding obstacles to reach its destination. Figure 19 illustrates the robot's navigation trajectory and shows a sequence of tasks. Initially, the robot follows the corridor walls, alternating left and right obstacle avoidance behaviors. Upon exiting the corridor, the robot switches to forward obstacle avoidance behavior, continuing its trajectory toward the destination. It carefully adjusts its trajectory when confronted with a wall on the right, a cylindrical obstacle on the left, and successfully avoids these obstacles to reach its goal.

Regarding stability in environments with obstacles, Figures 20a and 20b show the Lyapunov function and its derivative accurately calculated in environment n° 4. It is clear that the Lyapunov function converges, despite the fact that fluctuations occur due to the transition from one behavior to another.

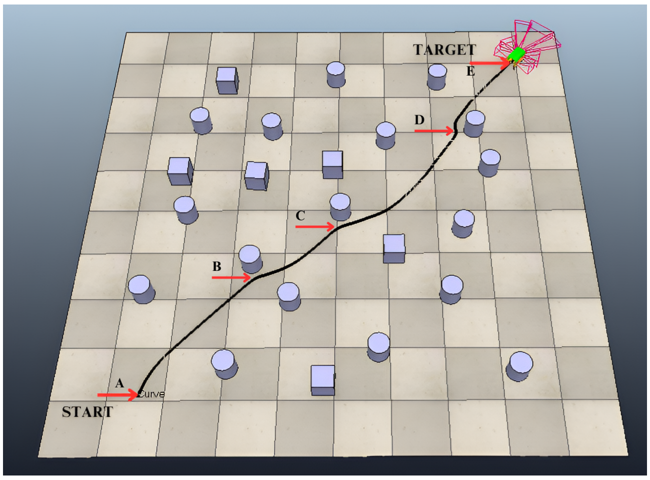


Figure 16: Environment n° 1.

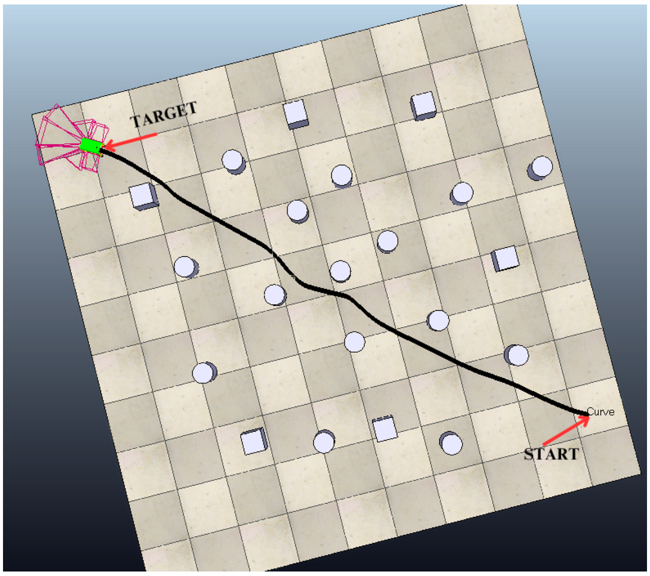


Figure 17: Environment n^o 2.

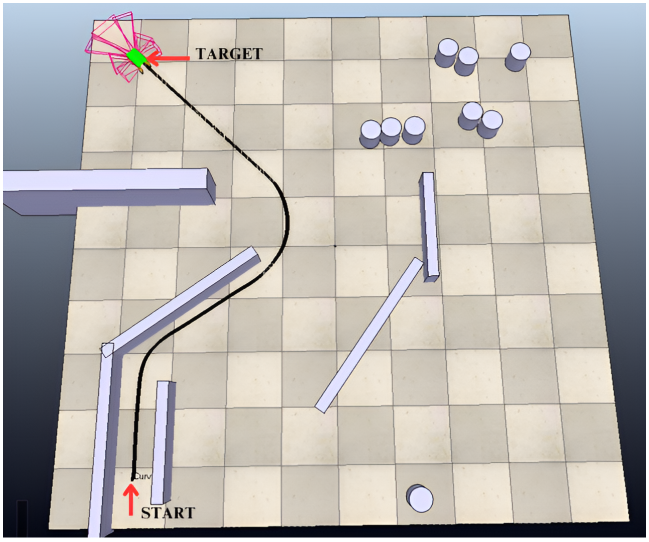


Figure 18: Environment n^o 3.

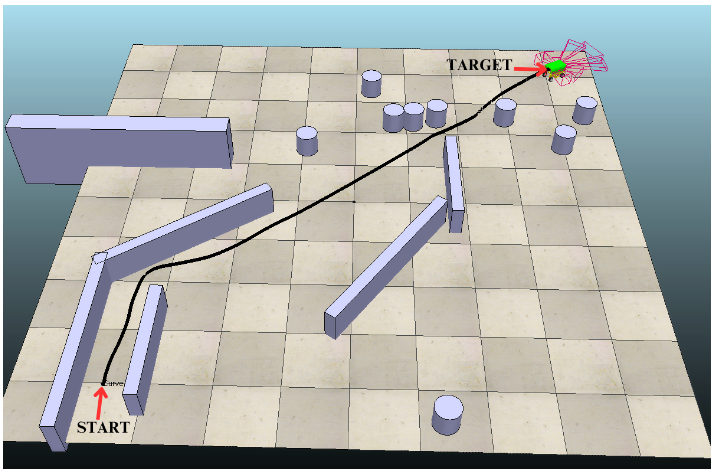
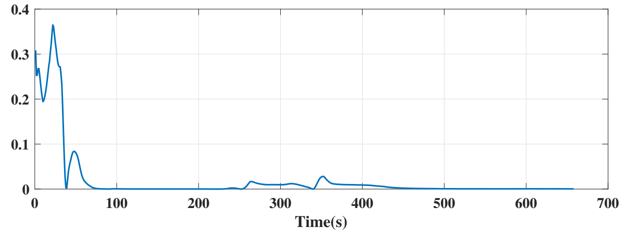
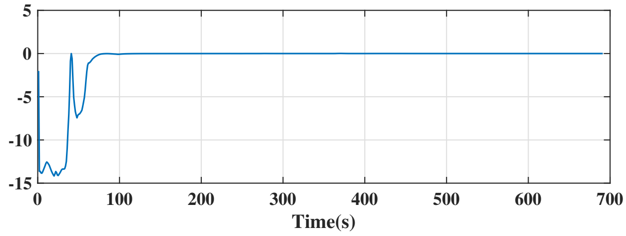


Figure 19: Environment n^o 4.



(a)



(b)

Figure 20: (a) Lyapunov function , (b) derivative of Lyapunov function in an environment with obstacles.

6 Conclusion

In this research paper, a solution to the challenge of navigating a mobile robot equipped with four-wheel drive and steer (4WD/4WS) in obstacle-cluttered environments is proposed. The introduced control systems are based on IT2 fuzzy behavior, breaking down the global navigation task into simpler sub-tasks.

Four specialized behaviors have been developed, each intended for a specific objective. However, these behaviors are insufficient when operating individually to ensure autonomous robot navigation. To address this limitation, an arbitration strategy is introduced. This strategy selects which behaviors to activate, assigning them weights that vary based on their priorities, the target's position, sensor inputs, and the robot's current state.

An equation has been proposed to aggregate all the outputs of the different controllers, i.e. the speed and steering angle generated by the four behaviors. These outputs are weighted by the values provided by the type 1 fuzzy supervisor, ultimately providing the appropriate outputs for a specific situation.

Various experiments have been conducted in diverse contexts, demonstrating the efficiency of our technique. In all situations, the robot successfully reached any position within its working environment while effectively avoiding obstacles.This performance indicates that our methodology can be considered conservative, since it guarantees the robot's ability to achieve its goals without any collisions. The use of dual inference mechanisms and a multitude of rules is likely to increase computation. To alleviate this problem, we suggest using tabular fuzzy logic in future work. This approach, especially when implemented, is advantageous for preserving the conservative nature of our strategy.

Given the difficulties encountered in fine-tuning adhesion parameters, our prospective efforts will focus on refining these parameters via meta-heuristic techniques. In addition, we intend to reduce the variability resulting from changes in the controller.

References

[1] Oultiligh, A.; Ayad, H.; Elkari, A.; Mjahed, M.; El Gmili, N. (2021); A Hybrid PSO-GWO Algorithm for Robot Path Planning in Uncertain Environments, *International Review of Automatic Control (IREACO)*, 14, 360, 2021.

[2] Muni, M.K.; Pradhan, P.K.; Dhal, P.R. et al. (2023); Improving navigational parameters and control of autonomous robot using hybrid SOMA–PSO technique, *Evolutionary Intelligence*, 1-17, 2023.

[3] Paikray, H.K.; Das, P.K.; Panda, S. (2021); Optimal Multi-robot Path Planning Using Particle Swarm Optimization Algorithm Improved by Sine and Cosine Algorithms, *Arabian Journal for Science and Engineering* , 46, 3357–3381, 2021.

[4] Suresh, K.S.; Ravichandran, K.S.; Venugopal, S.b. (2022); Multi-objective genetic algorithm for mobile robot path planning in industrial automation *Journal of Intelligent Fuzzy Systems*, 1-14, 2022.

[5] Tao, Y.; Gao, H.; Ren, F.; Chen, C.; Wang, T.; Xiong, H.; Jiang, S. (2021); A Mobile Service Robot Global Path Planning Method Based on Ant Colony Optimization and Fuzzy Control, *Applied Sciences*, 11, 3605, 2021.

[6] Lin, C.J.; Chang, M.Y; Tang, K.H; Huang, C.K (2023); Navigation Control of Ackermann Steering Robot Using Fuzzy Logic Controller, *Sensors and Materials*, 35, 781, 2023.

[7] Sultanova, A.B.; Aliev, R. A.; Kacprzyk, J.; Pedrycz, W.; Jamshidi, Mo.; Babanli, M. B.; Sadikoglu, F. (2023); Fuzzy Logic-Based Planning of the Behavior of Autonomous Vehicles, *International Conference on Applications of Fuzzy Systems, Soft Computing and Artificial Intelligence Tools (ICAFS), 15th, 2022*, 610, 2022.

[8] Batti, H.; Jabeur, C. B.; Fourati, H., (2022); Fuzzy Logic Based Control for Autonomous Mobile Robot Navigation and Obstacles Avoidance, *IEEE Information Technologies and Smart Industrial Systems (ITSIS), 2022*, 1-6, 2022.

[9] Hayat, A. D.; Hassan, A.; Hajar, M.; Ali, E. A. (2022); Fuzzy Logic Controller for 4WD/4WS Autonomous Agricultural Robotic, *IEEE International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), 3rd, 2022*, 1-6, 2022.

[10] Bouteraa, Y.; Ben Abdallah, I.; Elmogy, A.; Ibrahim, A.; Tariq, U.; Ahmad, T. (2020); A Fuzzy Logic Architecture for Rehabilitation Robotic Systems, *International Journal of Computers Communications and Control (IJCCC)*, 15, 2020.

[11] Chanchí, G. E.; Ospina, M.; Saba, M.(2023); Floods risk determination through a fuzzy logic system in developing countries. Case study Magdalena River, Colombia.s, *International Journal of Computers Communications and Control (IJCCC)*, 18, 5103, 2023.

[12] Wu, H.; Xu, Z.; (2020); Fuzzy Logic in Decision Support: Methods, Applications and Future Trends , *International Journal of Computers Communications and Control (IJCCC)*, 16, 2020.

[13] Aslam, M. S.; Li, Q.; Hou, J. (2021); Fault detection for asynchronous T–S fuzzy networked Markov jump systems with new event-triggered scheme, *IET Control Theory and Applications*, 15(11), 1461-1473, 2021.

[14] Al-Mallah, M.; Ali, M.; Al-Khawaldeh, M. (2022); Obstacles Avoidance for Mobile Robot UsingType-2 Fuzzy Logic Controller, *Robotics*, 11, 130, 2022.

[15] Cuevas, F.; Castillo, O.; Cortés-Antonio, P. (2020). Omnidirectional Four Wheel Mobile Robot Control with a Type-2 Fuzzy Logic Behavior-Based Strategy, *Studies in Computational Intelligence*, 862, 49–62, 2020.

[16] Cuevas, F.; Castillo, O.; Antonio, P. (2021). Design of a Control Strategy Based on Type-2 Fuzzy Logic for Omnidirectional Mobile Robots, *Journal of Multiple-Valued Logic and Soft Computing*, 37, 107–136, 2021.

[17] Kasmi, B.; Hassam, A. (2021); Comparative Study Between Fuzzy Logic and Interval Type-2 Fuzzy Logic Controllers for the Trajectory Planning of a Mobile Robot, *Engineering, Technology and Applied Science Research*, 11, 2, 7011–7017, 2021.

[18] Saidi, Y.; Nemra , A.; Tadjine, A.(2020); Robust mobile robot navigation using fuzzy type 2 with wheel slip dynamic modeling and parameters uncertainties, *International Journal of Modelling and Simulation*, 40, 6, 397-420, 2020.

[19] Muthugala, M. A. V. J.; Samarakoon, S. M. B. P.; Mohan Rayguru, M.; Ramalingam, B.; Elara, M. R. (2020); Wall-Following Behavior for a Disinfection Robot Using Type 1 and Type 2 Fuzzy Logic Systems, *Sensors (Basel)*, 20(16), 4445, 2020.

[20] Cuevas, F.; Castillo, O.; Cortes, P. (2020); Towards a control strategy based on type-2 fuzzy logic for an autonomous mobile robot, *Hybrid intelligent systems in control, pattern recognition and medicine*, 301-314, 2020.

[21] Zadeh, L.A.(1975); The Concept of a Linguistic Variable and its Application to Approximate Reasoning, *Information Sciences*, 8, 199–249, 1975.

[22] Liang, Q., Mendel, J.M. (1999); An introduction to type-2 TSK fuzzy logic systems, *FUZZ-IEEE'99. 1999 IEEE International Fuzzy Systems. Conference Proceedings (Cat. No.99CH36315)*, 3, 1534-1539, 1999.

[23] Galluzzo, M., Cosenza, B., Aronadio, G. (2009); Stability Analysis of Type2 Fuzzy Logic Controllers, *Chemical Engineering Transactions*, 17, 2009.

Cite this paper as:

Ait dahmad, H.; Ayad, H.; García Cerezo, A.; Mousannif H. (2024). IT-2 Fuzzy Control and Behavioral Approach Navigation System for Holonomic 4WD/4WS Agricultural Robot, *International Journal of Computers Communications & Control*, 19(3), 6420, 2024.
https://doi.org/10.15837/ijccc.2024.3.6420