

Research Article

Hybrid Type-2 Fuzzy-Sliding Mode Controller for Navigation of Mobile Robot in an Environment Containing a Dynamic Target

Dorra Ayedi ¹, Maïssa Boujelben,² and Chokri Reki²

¹Control and Energy Management Lab (CEM Lab), University of Sousse, Sousse Engineering School, BP 264 Sousse Erriadh 4023, Tunisia

²Control and Energy Management Lab (CEM Lab), University of Sfax, Sfax Engineering School, BP W, 3038 Sfax, Tunisia

Correspondence should be addressed to Dorra Ayedi; dorra.ayedii@yahoo.fr

Received 18 January 2018; Accepted 14 June 2018; Published 1 August 2018

Academic Editor: Yangmin Li

Copyright © 2018 Dorra Ayedi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We are interested in developing a new control architecture to command a mobile robot in a partially known environment. For that purpose, a hybrid control architecture is adopted, combining the reactive and the deliberative method. The reactive method consists of the hierarchical fuzzy controllers based on Type-2 Fuzzy Logic System (T2-FLS), dedicated to commanding the robot towards a mobile target while avoiding unexpected obstacles. A comparative study is made to show the efficiency of T2-FLS against Type-1 Fuzzy Logic System (T1-FLS). Additionally, the used deliberative method is the sliding mode, allowing the robot to track the mobile goal trajectory. Simulation results are given finally to test the proposed architecture.

1. Introduction

Controlling a mobile robot in an unknown environment is an attractive aspect in the field of robotic, as it is used in many applications. Therefore, researchers have developed several methods of controlling non-holonomic robot in a real environment. Those methods can be classified into three groups: reactive approaches, deliberative approaches, and hybrid approach, which is a combination between both of them [1]. The choice between those classes depends on the robot's knowledge of the environment [2].

The reactive approach does not require a prior knowledge about the environment. The robot receives at each instant information about its local environment and then moves autonomously to reach a desired point [2, 3]. This approach is adaptable to the dynamic or unexpected environment, but may have some problems such as nonconvergence or local minima [4]. Neural networks [5], potential field [6], and fuzzy logic [7] are among the methods which illustrate the reactive approach.

Unlike the reactive category, the deliberative method is based on trajectory planning, which needs to consider a complete knowledge about the environment [2]. Although it

is considered as a fast and stable navigation method, it can not be adapted in a completely unknown environment [4]. Many control methods were proposed in the domain of deliberative approach. Among those methods, we cite the PID controllers [8] and the sliding mode control [9].

As it was explained, many limitations can be faced while applying only reactive or deliberative methods [10]. They can not solve individually the problems faced in the navigation domain [11]. Therefore, those two methods are combined to form the hybrid approach. The favor of this mix is to benefit from the advantages of the two approaches [2] and provides the possibility of using the reactive behavior as well as global vision of the environment.

In the literature, Arkin and Murphy were the first who have made a combination between the reactive and the deliberative methods, which leads to the hybrid approach [1]. Zhu has used this combination in his work [12], letting the reactive approach playing a dominant role in case of conflict. Tian has designed in [13] a hybrid navigation control strategy for rescue robot. In this paper, the idea of developing a hybrid control architecture is adopted, as it suits with the environment in which the robot moves.

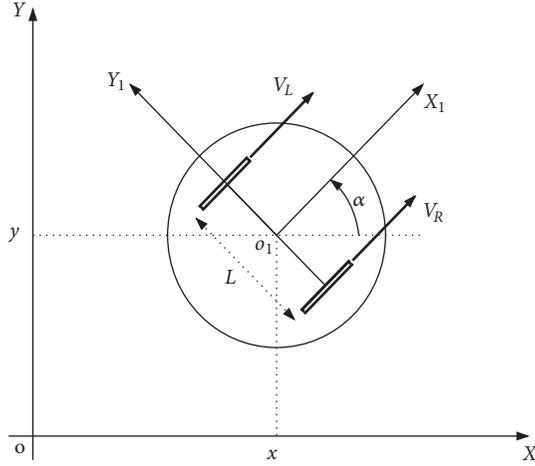


FIGURE 1: The schematic model of the Khepera II robot.

The used robot must have a considerable number of sensors distributed on its circumference. At first, the robot moves from an initial point to reach a dynamic target, which the trajectory is predefined. The use of the reactive method becomes necessary for avoiding unexpected obstacles. Fuzzy logic system was adopted for this task, as it is able to conceive robust controllers [14, 15]. The main problem which faces us is the huge number of obtained rules, due to the big number of information received by each sensor. In [16], the authors have proposed the hierarchical fuzzy controller in order to solve this problem.

On the other hand, the robot can face perturbations due to several sources. Among those sources, we mention the linguistic uncertainties, imprecise sensor information, and uncertainties in inputs or outputs [17, 18]. T1-FLS can not really handle all those uncertainties [15]. An extension of the T1-FLS have been introduced by Zadeh [19]. This new type is characterized by a third dimension which makes it able to handle all the perturbations mentioned [17].

Once the robot becomes close enough from the dynamic goal, it follows the trajectory of the target, which is a predefined safe path. In this case, the deliberative method copes better with this situation. Therefore, the sliding mode was adopted for trajectory tracking, for his fast response, and his robustness against the variation of parameters [3, 20].

The main contribution of this paper is to construct a suitable control architecture based on the association of deliberative and reactive method. Hierarchical T2-FLS and T1-FLS were implemented in the reactive controllers method and tested on the robot. A comparative study is made to show the outperformance of type 2 against type 1. For the deliberative controller, the sliding mode is proposed as a local method, to track the dynamic target until reaching a desired point.

This paper is organized as follows: Section 2 presents the robot control approach. On Section 3, reactive navigation controllers are detailed. Path planning is described on Section 4. At least, conclusion is given.

2. Robot Control Approach

The mobile robot is requested to navigate from an initial point to reach a dynamic target while avoiding obstacles. In this context, the Khepera II robot is used. Our robot is a non-holonomic robot, equipped with two independent driving wheels located on the same axis. Eight sensors are distributed on the circumference of the robot in a way that permits the detection of obstacles, regardless of their positions in the environment. A schematic model of this robot is presented in Figure 1.

2.1. Robot Kinematic Model. Our purpose is to control the wheels speed to command the robot. For that, the kinematic model of the used robot is given by the following system:

$$\begin{aligned}\dot{x} &= \frac{V_R + V_L}{2} \cos \alpha \\ \dot{y} &= \frac{V_R + V_L}{2} \sin \alpha \\ \dot{\alpha} &= \frac{V_R - V_L}{L}\end{aligned}\quad (1)$$

where

- (i) V_L and V_R are, respectively, the left and right wheels velocities.
- (ii) α is the angle which separates the robot direction from the X -axis.
- (iii) L is the distance between the left and the right wheels.

2.2. The Proposed Control Architecture. In this paper, a hybrid control is proposed. This approach combines reactive and deliberative methods. It permits benefiting from the advantages of the two methods [21]. Firstly, the robot moves in an unknown environment towards the mobile target, which needs the application of the reactive method. When the robot is close enough from the mobile target, path planning is sufficient, since the trajectory of the goal is known. Figure 2 presents the proposed strategy of control.

Figure 2 shows the presence of three controllers: the attraction to dynamic target controller allowing the robot to reach the mobile goal, the obstacle avoidance controller which enables it to get away from obstacles, and the path planning controller devoted for trajectory tracking.

According to the information given in each step by the environment (position of the robot, the goal, and the obstacles) and sent to the robot through the sensors, the action selection block chooses the suitable controller to command the robot. If the robot is far away from the target, the reactive method [15] is activated. In this case, the distance which separates the robot from the obstacle defines the choice between the attraction to dynamic target controller and the obstacle avoidance controller. Both of them work in a similar way and give as outputs the right and left wheels velocities V_R and V_L , except that the velocities obtained by the obstacle avoidance controller are inverted.

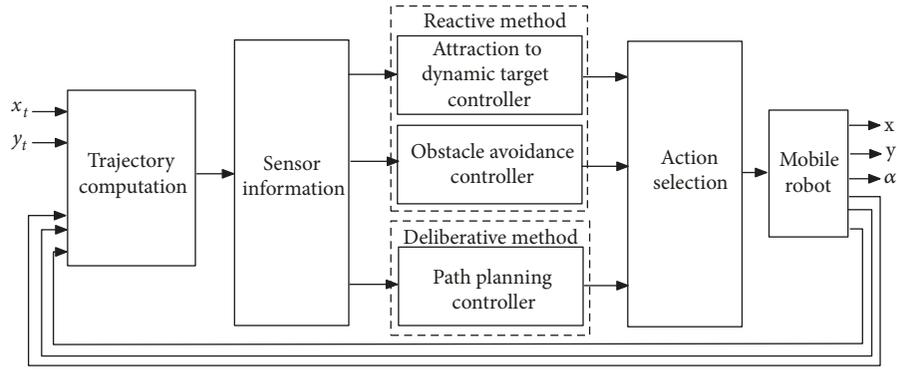


FIGURE 2: The proposed architecture of the mobile robot navigation.

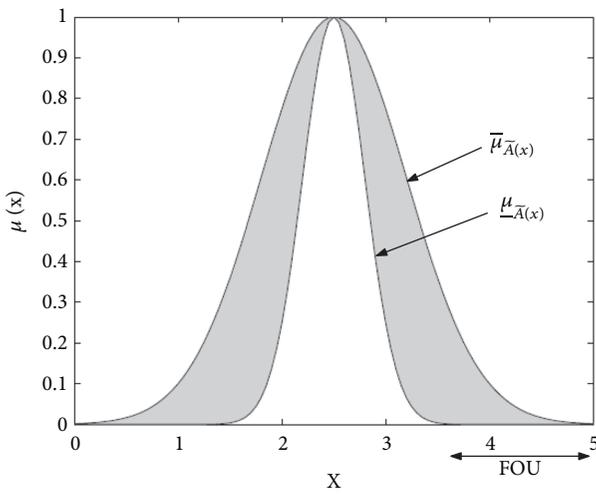


FIGURE 3: Gaussian type-2 fuzzy system.

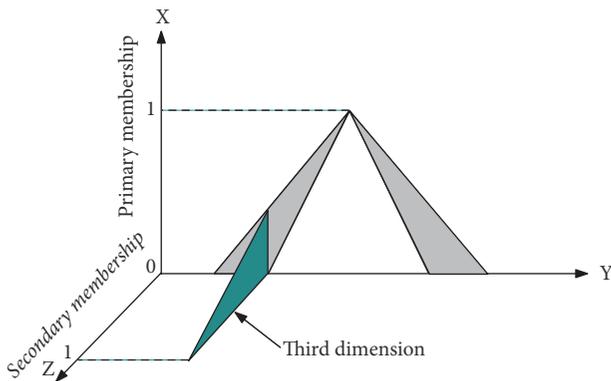


FIGURE 4: The third dimension of a triangular T2-FLS.

At each instant, the goal and robot positions are compared. When the robot becomes close to the goal position, it should follow it until reaching the desired point. In consequence, the deliberative method (path planning controller) is activated.

As a result, we obtain the right and left wheels velocities V_R and V_L , which control the robot.

3. Reactive Navigation

The environment in which the robot moves is an unknown area of navigation. Based on its sensor information, the robot must go towards the target, following a secure path by avoiding the obstacles.

The adopted approach to conceive the controllers which are dedicated for the reactive method is Type-2 Fuzzy Logic System (Type-2 FLS). This approach has become very well known in the field of control and automation, as it permits realizing robust controllers [15].

3.1. Type-2 Fuzzy Logic System. Based on Zadeh’s theory, the T2-FLS is an extension of the T1-FLS. The membership function of a T2-FLS is a fuzzy set, unlike the T1-FLS which the membership function is a crisp number [22].

The T2-FLS is characterized by the following:

- (i) A footprint of uncertainty as is shown in Figure 3, bounded by a lower and an upper membership function $\mu_{\tilde{A}(x)}$ and $\bar{\mu}_{\tilde{A}(x)}$ [23].
- (ii) A third dimension presented in Figure 4. Indeed, each variable x is associated with a primary membership function and a secondary membership function.

Those particularities guarantee additional degree of freedom [17], which push us to opt for the T2-FLS, to handle the maximum of uncertainties that the robot can face while moving towards the goal.

Mathematically speaking, the definition of a generalized T2-FLS is

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) \mid \forall x \in X, \forall u \in J_x \subseteq [0, 1]\} \quad (2)$$

where X and J_x are, respectively, the primary and the secondary domain [24, 25].

Due to the complexity of the generalized T2-FLS, the interval T2-FLS is adopted, where all $\mu_{\tilde{A}}(x, u)$ are equal to 1 [25]. A simple illustration of the third dimension of the T1-FLS, the generalized T2-FLS, and the interval T2-FLS are presented in Figure 5.

3.1.1. Inference System. Figure 6 shows the structure of a T2-FLS. Comparing to the model of a T1-FLS, the difference

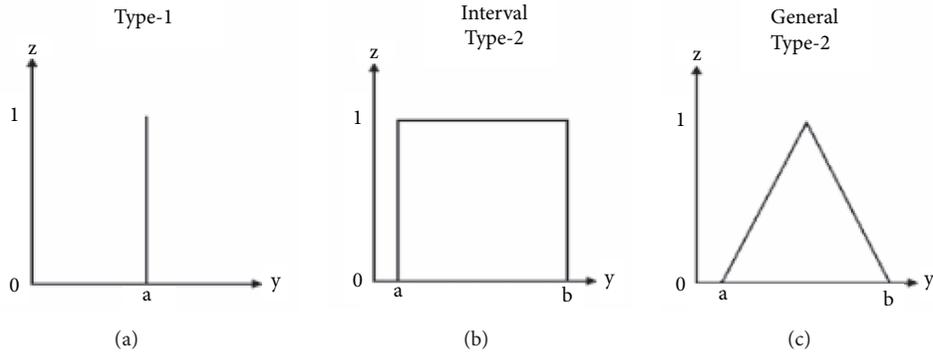


FIGURE 5: A clear view of the third dimension. (a) Type-1 fuzzy set. (b) Interval type-2 fuzzy set. (c) General type-2 fuzzy set.

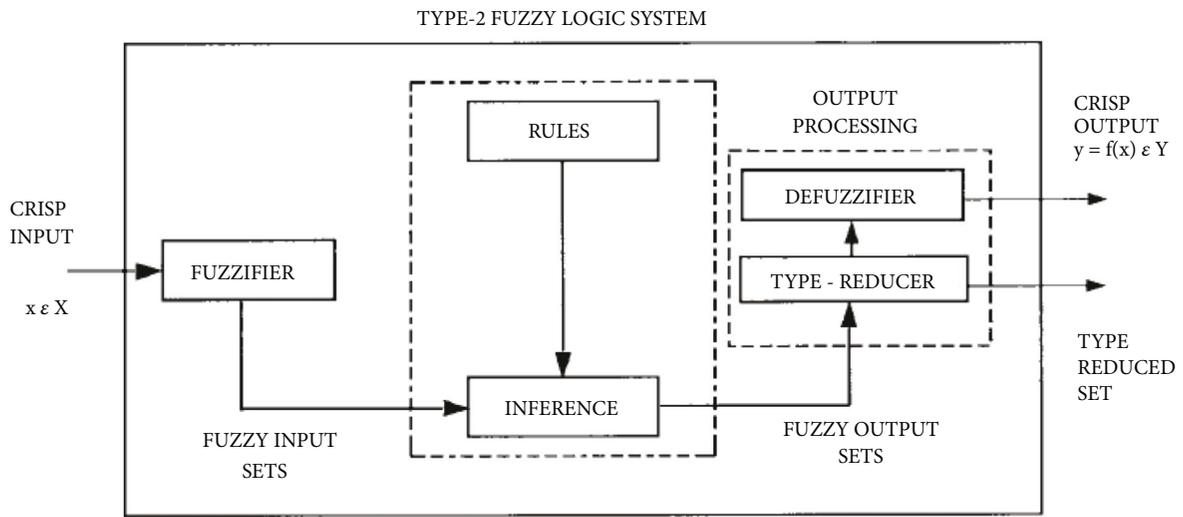


FIGURE 6: The structure of type-2 fuzzy inference system.

lies on the output processor. In addition to the defuzzifier block which exists on the T1-FLS structure, the block of type reducer is joined. Indeed, this block is dedicated to reducing the type of type-2 set to generate type-1 set [25].

Similarly to the T1-FLS, the interval T2-FLS is also based on the IF-THEN rules, except the category of the antecedents and the consequences which differs from type-1. For the case of the proposed architecture to command the Khepera II robot, the i^{th} rule is given as follows:

$$\text{If } (d \text{ is } \tilde{A}_i) \text{ and } (\varphi \text{ is } \tilde{B}_i) \text{ then } (V_R = [y_l, y_r]) \text{ and } (V_L = [z_l, z_r]),$$

where $i = 1, 2, \dots, N$, \tilde{A}_i and \tilde{B}_i are the interval type-2 fuzzy systems, $y_l, y_r, z_l,$ and z_r are, respectively, the upper and lower values of V_R and V_L .

The inference engine has as role to link rules and form firing interval set W_i

$$W_i = [\underline{w}_i, \bar{w}_i] \quad (3)$$

$$\text{with } \underline{w}_i = [\mu_{\tilde{A}_i}(d) \times \mu_{\tilde{B}_i}(\varphi)]$$

$$\text{and } \bar{w}_i = [\bar{\mu}_{\tilde{A}_i}(d) \times \bar{\mu}_{\tilde{B}_i}(\varphi)]$$

where \times represents the t-norm used, which can be the minimum or the product operation [26].

3.1.2. Output Processing. In this step, a fuzzy type-2 set goes through a type reducer to become a fuzzy type-1 set, then through a defuzzifier to obtain a crisp value.

According to [27, 28], different methods of type reduction have been mentioned and detailed. The center of sets (COS) method was adopted in this paper. The type 1 fuzzy set expression is given in the following equation.

$$Y_{cos} = [y_l, y_r] \quad (4)$$

where Y_{cos} is an interval set bounded by two limits: y_l and y_r .

To compute those two values, Karnik and Mendel have proposed an algorithm [23, 29], explained in the following steps:

KM for Computing y_l

- (1) Sort rule consequence $y_{\underline{i}}$ ($i = 1, 2, \dots, M$) in increasing order then match each w_i with his respective $y_{\underline{i}}$, so their index corresponds to the renumbered $y_{\underline{i}}$.

- (2) Initialize w_i as the following equation:

$$w_i = \frac{w_i + \bar{w}_i}{2}, \quad i = 1, 2, \dots, M \quad (5)$$

then compute

$$y = \frac{\sum_{i=1}^M y_{\underline{i}} w_i}{\sum_{i=1}^M w_i} \quad (6)$$

- (3) Find the switch point k ($1 \leq k \leq M - 1$) such that

$$y_{\underline{i}} \leq y \leq y_{\underline{i+1}} \quad (7)$$

- (4) Set

$$y_i = \begin{cases} \bar{w}_i, & i \leq k \\ w_i, & i > k \end{cases} \quad (8)$$

then compute

$$y' = \frac{\sum_{i=1}^M y_i w_i}{\sum_{i=1}^M w_i} \quad (9)$$

- (5) Check if $y' = y$. If yes, stop and set $y_l = y$ and $K = L$. If no, set $y = y'$ and go to step (3)

KM for Computing y_r

- (1) Sort rule consequence \bar{y}_i ($i = 1, 2, \dots, M$) in increasing order then match each w_i with his respective \bar{y}_i , so their index corresponds to the renumbered \bar{y}_i .

- (2) Initialize w_i as the following equation:

$$w_i = \frac{w_i + \bar{w}_i}{2}, \quad i = 1, 2, \dots, M \quad (10)$$

then compute

$$y = \frac{\sum_{i=1}^M \bar{y}_i w_i}{\sum_{i=1}^M w_i} \quad (11)$$

- (3) Find the switch point k ($1 \leq k \leq M - 1$) such that

$$\bar{y}_i \leq y \leq \bar{y}_{i+1} \quad (12)$$

- (4) Set

$$y_i = \begin{cases} w_i, & i \leq k \\ \bar{w}_i, & i > k \end{cases} \quad (13)$$

then compute

$$y' = \frac{\sum_{i=1}^M \bar{y}_i w_i}{\sum_{i=1}^M w_i} \quad (14)$$

- (5) Check if $y' = y$. If yes, stop and set $y_r = y$ and $K = R$. If no, set $y = y'$ and go to step (3)

At last, the values calculated in this algorithm pass through the defuzzification block to obtain the final defuzzified output:

$$Y = \frac{y_l + y_r}{2} \quad (15)$$

Type-2 FLS is used in the development of the attraction to dynamic target controller and the obstacle avoidance controller.

3.2. Design of the Proposed Type-2 Fuzzy Logic Controllers. The robot should find a safe path to reach the mobile goal. For this task, the attraction to dynamic target controller and the obstacle avoidance controller are activated.

The first controller has as inputs

- (i) d : the distance between the robot and the mobile target,
- (ii) φ : the angle separating the robot orientation and the mobile target.

For the second controller, the inputs are

- (i) d_o : the distance between the robot and the obstacle,
- (ii) φ_o : the angle separating the robot orientation and the obstacle.

Those parameters are presented in Figure 7.

For the attraction to dynamic target controller, five membership functions are chosen for the distance d , while seven membership functions define the angle, as is detailed in [14].

If the action selection block chooses the second controller dedicated for obstacle avoidance, the use of the eight sensors of the Khepera II mobile robot is requested. It permits detecting the obstacles in different directions. However, a disadvantage occurs if we use one standard fuzzy controller: the number of obtained rules is huge, which increases the simulation time. To solve this problem, the hierarchical fuzzy system is used.

3.3. Hierarchical Type-2 Fuzzy Logic System. The hierarchical fuzzy system consists of attributing each sensor information to a controller, then regrouping them as is shown in Figure 8.

The choice of this structure is based on gathering the robot sensors according to angle measurement (Figure 9).

The first layer of the hierarchical fuzzy system contains eight controllers associated with the sensors measurements. Each controller gives the left and right wheels velocities, and another index denoted I_o^i . This index is dedicated to providing a collision degree between the robot and the obstacle.

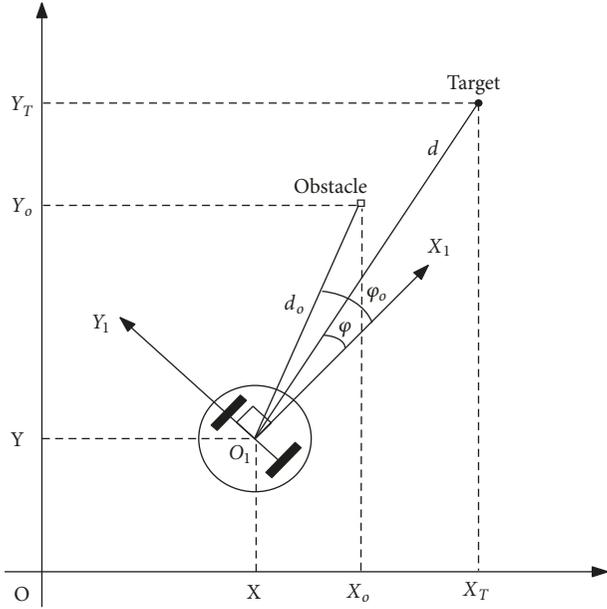


FIGURE 7: Representation of the parameters.

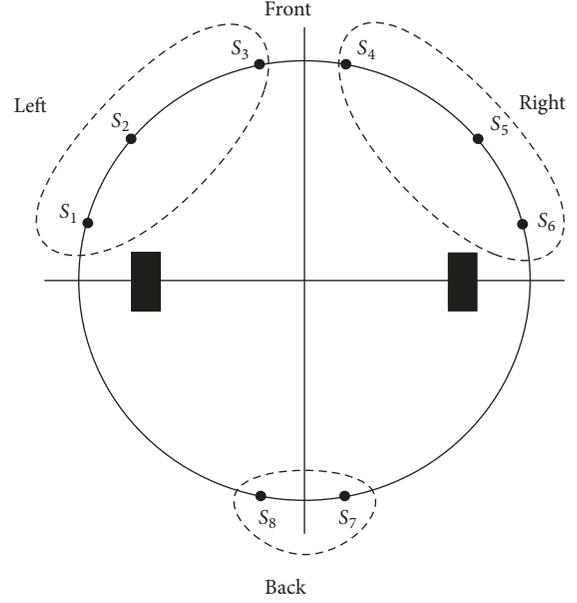


FIGURE 9: Decomposition of the eight sensors of the robot.

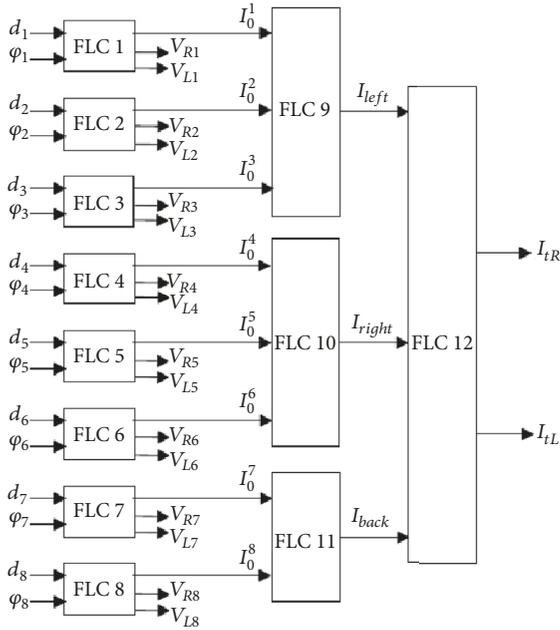


FIGURE 8: Block diagram of the hierarchical system.

According to Figure 9, those indexes are regrouped into three groups (according to sensor's positions in the robot) and are considered as inputs to three other controllers which form the second layer. As a consequence, we obtain three indexes: I_{left} , I_{right} , and I_{back} , which have the same meaning of the other indexes. The third and last layer provides as outputs I_{tR} and I_{tL} . Those indicators help for the robot guidance as they illustrate the absence degree of obstacles. More details and inference tables are given in [16, 30].

4. Path Planning

Once the robot is close enough from the mobile goal, the action selection block passes from the reactive approach to the deliberative approach. The trajectory of the goal is determined, and the robot should just follow the path of the target until reaching the desired point. For that, we have proposed the sliding mode method to conceive the path planning controller [9, 20].

The purpose of the sliding mode controller is to compute the appropriate linear and angular velocities so that the robot can follow exactly the target [3] by converging the tracking error to zero [31].

For that we consider the reference position $p_r = (x_r, y_r, \alpha_r)^T$ and the current position $p = (x, y, \alpha)^T$. The position error between p_r and p is denoted as $p_e = (x_e, y_e, \alpha_e)^T$, and it is expressed as

$$\begin{bmatrix} x_e \\ y_e \\ \alpha_e \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_r - x \\ y_r - y \\ \alpha_r - \alpha \end{bmatrix} \quad (16)$$

This tracking error is illustrated in Figure 10.

The velocity error is then obtained by deriving the error tracking p_e [20], which gives

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\alpha}_e \end{bmatrix} = \begin{bmatrix} y_e w - v + v_r \cos \alpha_e \\ -x_e w + v_r \sin \alpha_e \\ w_r - w \end{bmatrix} \quad (17)$$

where v_r and w_r are, respectively, the linear and the angular velocities of the mobile target.

Equation (1), which represents the adopted Khepera II mobile robot model, is a multiple-input nonlinear system.

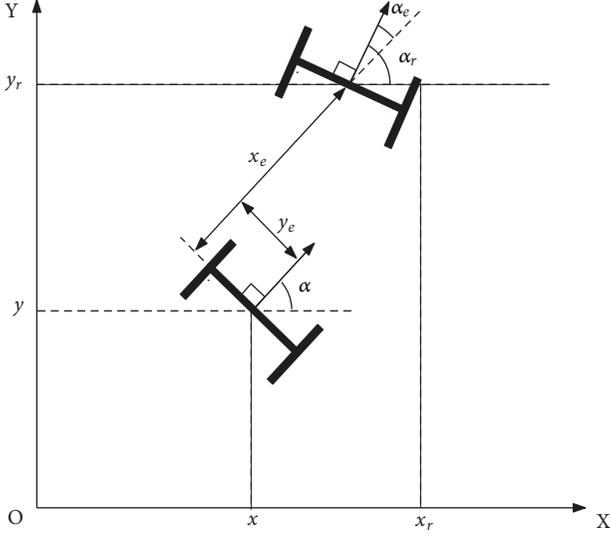
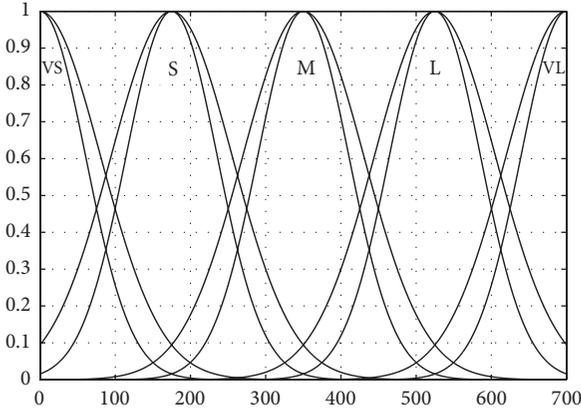


FIGURE 10: Tracking error.

FIGURE 11: Type-2 membership function of the input d (in mm).

According to [3, 20], x_e is chosen equal to zero at the first switching surface to simplify the problem. Therefore, the Lyapunov candidate function proposed in [3, 20] and adopted in this paper is

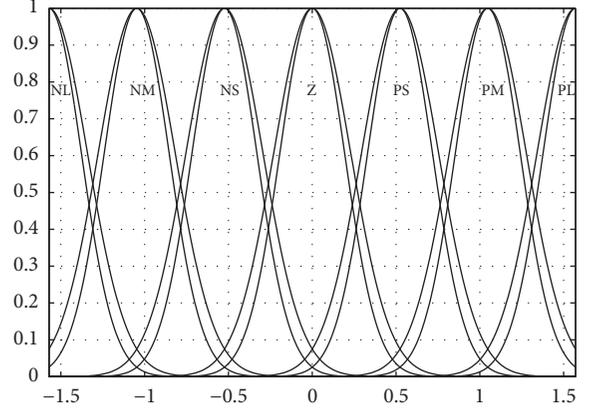
$$V = \frac{1}{2}y_e^2 \quad (18)$$

where the time derivative is expressed by

$$\begin{aligned} \dot{V} &= y_e \dot{y}_e = y_e (-x_e w + v_r \sin(\alpha_e)) \\ &= -x_e y_e w - v_r y_e \sin(\arctan(v_r y_e)) \end{aligned} \quad (19)$$

The switching function condition is defined as $\alpha_e = -\arctan(v_r y_e)$. We ensure that $v_r y_e \sin(\arctan(v_r y_e)) \geq 0$ which always satisfies the global condition $\dot{V} \leq 0$. The proof is detailed in [20]. Consequently, the vector of the switching surface is expressed in the following equation.

$$s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} x_e \\ \alpha_e + \arctan(v_r y_e) \end{bmatrix} \quad (20)$$

FIGURE 12: Type-2 membership function of the input φ (in rad).

s_1 converges to 0 when x_e converges to 0. In a similar way, s_2 converges to 0 when $\alpha_e + \arctan(v_r y_e)$ converges to 0, which means that α_e converges to $-\arctan(v_r y_e)$. Consequently, we obtain the convergence of α_e and y_e to 0.

To design a control law, the switching surface is expressed as a saturation function [32], supposing that $\gamma = \arctan(v_r y_e)$:

$$\begin{aligned} \dot{s} &= \begin{bmatrix} \dot{s}_1 \\ \dot{s}_2 \end{bmatrix} = \begin{bmatrix} -k_1 \text{sat}(s_1) \\ -k_2 \text{sat}(s_2) \end{bmatrix} = \begin{bmatrix} \dot{x}_e \\ \dot{\alpha}_e + \frac{\partial \gamma}{\partial v_r} \dot{v}_r + \frac{\partial \gamma}{\partial y_e} \dot{y}_e \end{bmatrix} \\ &= \begin{bmatrix} y_e w - v + v_r \cos \alpha_e \\ w_r - w + \frac{\partial \gamma}{\partial v_r} \dot{v}_r + \frac{\partial \gamma}{\partial y_e} (-x_e w + v_r \sin \alpha_e) \end{bmatrix} \end{aligned} \quad (21)$$

The obtained control law is

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} y_e w + v_r \cos \alpha_e + k_1 \text{sat}(s_1) \\ w_r + \frac{\partial \gamma}{\partial v_r} \dot{v}_r + \frac{\partial \gamma}{\partial y_e} (v_r \sin \alpha_e) + k_2 \text{sat}(s_2) \end{bmatrix} \quad (22)$$

where $\partial \gamma / \partial v_r = y_e / (1 + (v_r y_e)^2)$ and $\partial \gamma / \partial y_e = v_r / (1 + (v_r y_e)^2)$.

5. Simulation Results

In this part, we present the inputs simulation and some simulations results to test the validity of the control architecture proposed and to compare the performance of T1-FLS and T2-FLS.

5.1. Inputs Simulations. As was explained previously, we have two inputs: the distance d and the angle of direction φ . d varies between 0 and 700 mm and is divided into five fuzzy subsets: VS (very small), S (small), M (medium), L (large), and VL (very large). φ is defined in the interval $[-\pi/2, \pi/2]$, and seven fuzzy subsets are associated with it which are NL (negative large), NM (negative medium), NS (negative small), Z (zero), PS (positive small), PM (positive medium), and PL (positive large) [3]. Figures 11 and 12 illustrate the simulations of those two inputs.

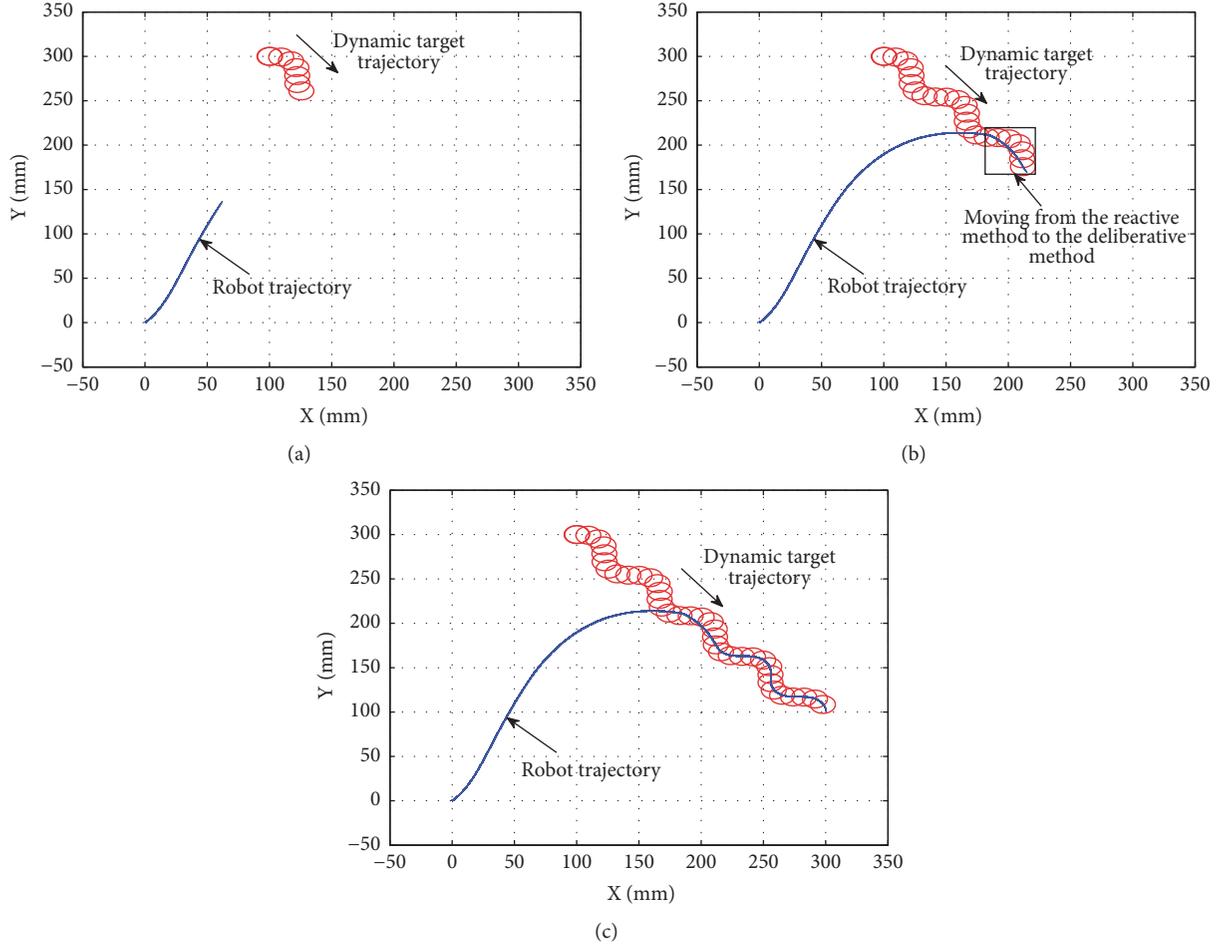


FIGURE 13: Path planning for the mobile robot using T1-FLS. (a)The robot tries to reach the target using the reactive approach. (b) The transition between the reactive and the deliberative approach. (c) Trajectory tracking applying the deliberative approach.

5.2. *Trajectory Simulation.* The objective in this part is to test the efficiency of the proposed architecture. For that purpose, we consider a robot which starts from an initial point $(X, Y) = (0, 0)$ and tries to get closer to a dynamic target until reaching the desired final point. Firstly, we have considered a safe environment, using T1-FLS as a reactive approach. Figure 13 illustrates clearly the steps of progress of the robot.

It is obvious that the robot can move towards the target and tracks the goal's trajectory until reaching the final point, which proves the validity of the proposed control architecture.

In the next simulations (Figures 14 and 15), the robot moves in environments containing obstacles and tries to reach a moving target using the T1-FLS and the T2-FLS.

In Figure 14, the robot starts from the initial point $(X, Y) = (0, 0)$. It succeeds to avoid the obstacles and to reach the circular trajectory of the target using the FLS (type 1 and type 2). Then, the sliding mode is applied for path tracking, until attaining the final point. Figure 15 treated the case of sinusoidal trajectory of the dynamic target, with changing the obstacles position and number. The robot adapts with those

TABLE 1: Comparison between distances covered with T1-FLS and T2-FLS.

	Figure 14	Figure 15
T1-FLS	459.1522	527.1141
T2-FLS	447.8955	515.9508
Relative errors (%)	2.5	2.2

changes and achieves the navigation and the tracking task properly.

The comparison between the distances covered by the robot after implanting T1-FLS and T2-FLS is presented as numerical values in Table 1. This table shows that T2-FLS always provides better results comparing to type 1, giving optimized distances.

6. Conclusion

In this paper, we have proposed a hybrid control law to command a mobile robot in a real environment containing

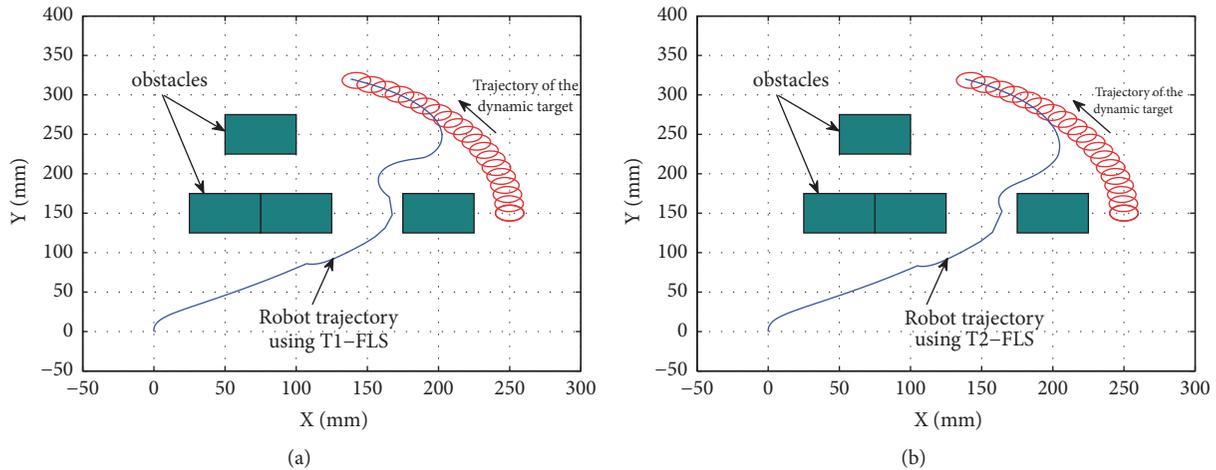


FIGURE 14: Simulation result with circular trajectory of the dynamic target using T1-FLS (a) and T2-FLS (b).

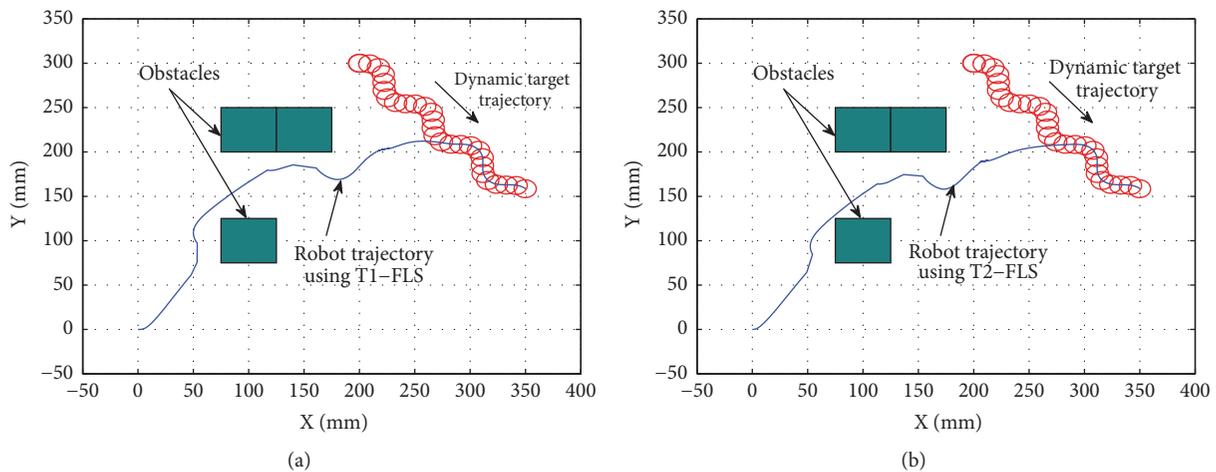


FIGURE 15: Simulation result with sinusoidal trajectory of the dynamic target using T1-FLS (a) and T2-FLS (b).

static obstacles and dynamic target. The control architecture adopted combines reactive approach and deliberative approach. Indeed, the hierarchical fuzzy logic system is applied to bring the robot from an initial portion to become close to the trajectory of the goal. The robot goes after the dynamic target using the sliding mode. Simulations results are presented to test the validity of this algorithm and to prove the outperformance of the T2-FLS comparing to the T1-FLS in the reactive controllers.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] D. Nakhaeinia, S. Hong Tang, S. B. Mohd Noor, and O. Motlagh, "A review of control architectures for autonomous navigation of mobile robots," *International Journal of Physical Sciences*, vol. 6, no. 2, pp. 169–174, 2011.
- [2] N. Hacene and B. Mendil, "Toward safety navigation in cluttered dynamic environment: A robot neural-based hybrid autonomous navigation and obstacle avoidance with moving target tracking," in *Proceedings of the 3rd International Conference on Control, Engineering and Information Technology, CEIT 2015*, pp. 1–6, May 2015.
- [3] M. Boujelben, C. Reikik, and N. Derbel, "A hybrid fuzzy-sliding mode controller for a mobile robot," *International Journal of Modelling, Identification and Control*, vol. 25, no. 3, pp. 155–164, 2016.
- [4] M. Hank and M. Haddad, "Hybrid control architecture for mobile robots navigation in partially known environments," in *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2014*, pp. 513–521, Austria, September 2014.

- [5] I. Engedy and G. Horváth, "Artificial neural network based mobile robot navigation," in *Proceedings of the WISP 2009 - 6th IEEE International Symposium on Intelligent Signal Processing*, pp. 241–246, Hungary, August 2009.
- [6] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
- [7] H. Maaref and C. Barret, "Sensor-based navigation of a mobile robot in an indoor environment," *Robotics and Autonomous Systems*, vol. 38, no. 1, pp. 1–18, 2002.
- [8] R. Rojas and A. G. Förster, "Holonomic control of a robot with an omnidirectional drive," *KI-Künstliche Intelligenz*, vol. 20, no. 2, pp. 12–17, 2006.
- [9] R. Solea, A. Filipescu, and U. Nunes, "Sliding-mode control for trajectory-tracking of a wheeled mobile robot in presence of uncertainties," in *Proceedings of the 7th Asian Control Conference (ASCC '09)*, pp. 1701–1706, Hong Kong, August 2009.
- [10] M. Hank and M. Haddad, "A hybrid approach for autonomous navigation of mobile robots in partially-known environments," *Robotics and Autonomous Systems*, vol. 86, pp. 113–127, 2016.
- [11] D. Nakhaeinia, P. Payeur, T. S. Hong, and B. Karasfi, "A hybrid control architecture for autonomous mobile robot navigation in unknown dynamic environment," in *Proceedings of the 11th IEEE International Conference on Automation Science and Engineering, CASE 2015*, pp. 1274–1281, Sweden, August 2015.
- [12] Y. Zhu, T. Zhang, J. Song, and X. Li, "A new hybrid navigation algorithm for mobile robots in environments with incomplete knowledge," *Knowledge-Based Systems*, vol. 27, pp. 302–313, 2012.
- [13] F. Tian, S. Ge, and H. Zhu, "A navigation control strategy with hybrid architecture for rescue robot," in *Proceedings of the 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS 2010)*, pp. 675–680, Xiamen, China, October 2010.
- [14] M. Boujelben, C. Reikik, and N. Derbel, "Hierarchical fuzzy controller for a nonholonomic mobile robot," in *Proceedings of the 2012 20th Mediterranean Conference on Control and Automation, MED 2012*, pp. 341–347, Spain, July 2012.
- [15] D. Ayedi, M. Boujelben, and C. Reikik, "Interval type-2 TSK fuzzy approach for autonomous mobile robot control in presence of uncertainties," in *Proceedings of the 2017 14th International Multi-Conference on Systems, Signals & Devices (SSD)*, pp. 280–286, March 2017.
- [16] M. Boujelben, C. Reikik, and N. Derbel, "Hierarchical fuzzy controller to avoid mobile obstacle for a mobile robot," in *Proceedings of the 2013 10th International Multi-Conference on Systems, Signals and Devices, SSD 2013*, pp. 1–8, March 2013.
- [17] H. A. Hagra, "A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 4, pp. 524–539, 2004.
- [18] I. Maalej, C. Reikik, D. B. H. Abid, and N. Derbel, "Interval type-2 Takagi-Sugeno-Kang fuzzy logic approach for three-tank system modeling," in *Proceedings of the 2014 IEEE 23rd International Symposium on Industrial Electronics, ISIE 2014*, pp. 144–149, IEEE, June 2014.
- [19] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning I," *Information Sciences*, vol. 8, pp. 199–249, 1975.
- [20] J. H. Lee, C. Lin, H. Lim, and J. M. Lee, "Sliding mode control for trajectory tracking of mobile robot in the RFID sensor space," *International Journal of Control, Automation, and Systems*, vol. 7, no. 3, pp. 429–435, 2009.
- [21] A. Benzerrouk, L. Adouane, and P. Martinet, "Stable navigation in formation for a multi-robot system based on a constrained virtual structure," *Robotics and Autonomous Systems*, vol. 62, no. 12, pp. 1806–1815, 2014.
- [22] K. Tai, A.-R. El-Sayed, M. Biglarbegian, C. I. Gonzalez, O. Castillo, and S. Mahmud, "Review of recent type-2 fuzzy controller applications," *Algorithms*, vol. 9, no. 2, p. 19, 2016.
- [23] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: theory and design," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 5, pp. 535–550, 2000.
- [24] J. M. Mendel and R. I. B. John, "Type-2 fuzzy sets made simple," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 117–127, 2002.
- [25] L. Amador-Angulo, O. Mendoza, J. R. Castro, A. Rodríguez-Díaz, P. Melin, and O. Castillo, "Fuzzy sets in dynamic adaptation of parameters of a bee colony optimization for controlling the trajectory of an autonomous mobile robot," *Sensors*, vol. 16, no. 9, p. 1458, 2016.
- [26] N. Karnik and J. Mendel, "Introduction to type-2 fuzzy logic systems," in *Proceedings of the 1998 IEEE International Conference on Fuzzy Systems IEEE World Congress on Computational Intelligence*, vol. 2, pp. 915–920, IEEE, Anchorage, AK, USA.
- [27] N. N. Karnik, J. M. Mendel, and Q. Liang, "Type-2 fuzzy logic systems," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 6, pp. 643–658, 1999.
- [28] N. N. Karnik and J. M. Mendel, "Type-2 fuzzy logic systems: type-reduction," in *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, pp. 2046–2051, IEEE, 1998.
- [29] M. Manceur, *Commande robuste des systèmes non linéaires complexes [Ph.D. thesis]*, Reims, 2012.
- [30] C. Reikik, M. Jallouli, and N. Derbel, "Optimal trajectory of a mobile robot using hierarchical fuzzy logic controller," *International Journal of Computer Applications in Technology*, vol. 53, no. 4, pp. 348–357, 2016.
- [31] M. N. Dharmaweera, S. Y. Khoo, and Z. Man, "Navigation of four-wheel-steering mobile robots using robust fault-tolerant sliding mode control," in *Proceedings of the 11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010*, pp. 1375–1380, IEEE, December 2010.
- [32] I. González, S. Salazar, and R. Lozano, "Chattering-free sliding mode altitude control for a quad-rotor aircraft: Real-time application," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1–4, pp. 137–155, 2014.



Hindawi

Submit your manuscripts at
www.hindawi.com

