

Hierarchical fuzzy controller for a nonholonomic mobile robot

Maïssa BOUJELBEN, Chokri REKIK and Nabil DERBEL

Abstract—The objective of this paper is to find optimal trajectories of a mobile robot during its motion from an initial position to a desired one in an autonomous way. For that, a new approach which uses hierarchical fuzzy systems has been proposed. This approach combines fuzzy logic systems and the potential field method. Firstly, the developed controller is based on fuzzy logic systems: this controller sends the speed values of each driving wheel to the robot, while seeking the goal. Secondly, the obstacle avoidance behavior is inspired from the potential field method: the robot is attracted to the target and pushed from obstacles. Genetic algorithms are used to optimize the consequences of fuzzy rules. Simulation results illustrate the validity and the efficiency of the suggested approach.

I. INTRODUCTION

Mobile robotic is a field of research currently considered for the automation of several applications, especially for designing systems able to move in an autonomous way. The industry is the sector that benefits the most from these systems. Indeed, their application areas are numerous. For example we can find the automobile field (for transport, agriculture...) and the exploration field (of planetary, mines, hostile environments...). Obstacle avoidance is a very important research problem ensuring the robot and the human security. For that, a mobile robot must be equipped with an adequate perception system in order to give it a reactive behavior allowing it to avoid obstacles in an autonomous way and in real time. In this context, several methods of reactive obstacle avoidance were developed in order to ensure the navigation of a mobile robot in an unknown environment [1]. Among these methods, we can find: the potential field method [2], the steering angle field [3], the dynamic window [4], etc. In addition, several research projects were carried out using advanced techniques such as neural networks and fuzzy logic systems [5], [6], [7]. In the last decades, the use of fuzzy logic became very widespread to design a robust controller satisfying autonomous navigation problems.

In this paper, a Fuzzy Logic Controller “FLC” is proposed in order to control the robot in an environment without obstacles [8], [9]. The objective of this paper is to command the mobile robot in an unknown environment. For that, our contribution is to determine an hierarchical fuzzy controller allowing to avoid obstacles [10], [11]. The advantage of this

M. BOUJELBEN is with Control and Energy Management Lab (CEM Lab) University of Sfax, Sfax Engineering School, BP W,1173-3038 Sfax, Tunisia boujelben.maïssa@gmail.com

C. REKIK is with Control and Energy Management Lab (CEM Lab) University of Sfax, Sfax Engineering School, BP W,1173-3038 Sfax, Tunisia chokri.rekik@enis.rnu.tn

N. DERBEL is with Control and Energy Management Lab (CEM Lab) University of Sfax, Sfax Engineering School, BP W,1173-3038 Sfax, Tunisia nabil.derbel@ieee.org

approach is to reduce the fuzzy rule number compared with a standard fuzzy system.

This paper is divided into six sections. Second section presents the problem formulation. In section three, a description of the developed optimal fuzzy logic controller is given. This is to control the robot in an environment without obstacles. In section four, we propose an hierarchical fuzzy controller to treat obstacle avoidance problem. Section five presents numerical simulations in order to illustrate the proposed approach. Finally, section six concludes the paper.

II. PROBLEM FORMULATION

The concept of mobile robots indicates the set of robots with mobile base. These robots are intended to move autonomously in their workspace. For that, several types of mobile robots were designed in order to ensure the autonomous navigation among which we find the mobile robots with wheels which are often used. In this context, we are interested by a mobile robot with two independent driving wheels which can be oriented and commanded by acting on the speed of each wheel (see figure 1).

The kinematic model is given by:

$$\begin{cases} \frac{dX_R}{dt} = \frac{V_R + V_L}{2} \cos \alpha_R \\ \frac{dY_R}{dt} = \frac{V_R + V_L}{2} \sin \alpha_R \\ \frac{d\alpha_R}{dt} = \frac{V_R - V_L}{L} \end{cases} \quad (1)$$

where V_R and V_L are the robot right and left wheel velocities, respectively; α_R is the angle between the robot direction and the X -axis; $\dot{\alpha}_R$ is the robot angular velocity and L is the distance between two wheels.

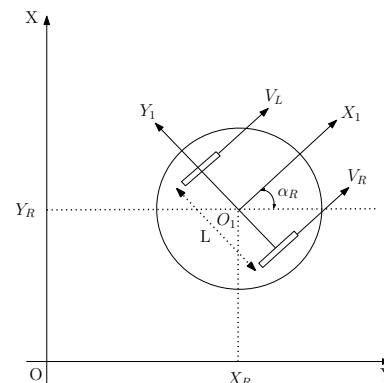


Fig. 1. The schematic model of mobile robot.

III. GENETIC SYNTHESIS OF FUZZY LOGIC CONTROLLER

Fuzzy subsets theory is used to solve problems which require human expertise (contrary to neural networks which require primarily data), i.e. fuzzy logic translates linguistic information coming from experts, to analyze situations and to make decisions in order to obtain good results. This technique is widely used to conceive the robust controller satisfying a great number of autonomous navigation problems. However the optimization of these controllers is rarely studied.

In this part, we have supposed that the environment is without obstacle. Figure 2 presents the block diagram to command the robot. The robot is equivalent to a system having two inputs (V_R and V_L which are the robot right and left wheel velocities) and three outputs (X_R , Y_R and α_R , defining the position and the orientation of the robot).

The fuzzy controller allows to orientate the robot towards its target starting from a given position. The robot position and the orientation are calculated in real time by an odometer. This controller has two inputs the distance d and the angle φ ; d is the distance separating the robot center and the desired position and φ is the angle from the robot to the target (see figure 3). Expressions of the distance d and the angle φ are given as follows:

$$d = \sqrt{(X_T - X_R)^2 + (Y_T - Y_R)^2} \quad (2)$$

$$\varphi = \theta_T - \alpha_R \quad (3)$$

with:

$$\theta_T = \arctan \frac{(Y_T - Y_R)}{(X_T - X_R)} \quad (4)$$

A. Fuzzy partition of input variables

Contrary to binary variables (true or false state), fuzzy variables present the variation field of the input variables. The distance d is chosen between 0 and 700 mm, whereas φ is defined in the interval $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Membership functions of the parameters d and φ (figures 4 and 5) are Gaussian and given by the following expression:

$$\text{Gauss}(x, \sigma, a) = \exp \left[- \left(\frac{x - a}{\sigma} \right)^2 \right] \quad (5)$$

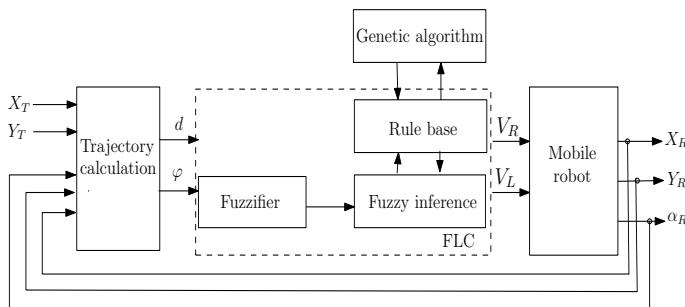


Fig. 2. Block diagram to command the robot.

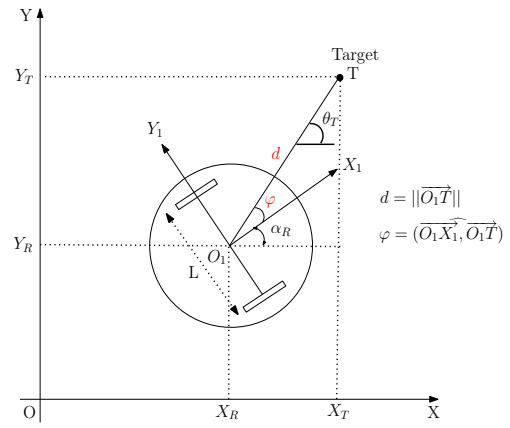


Fig. 3. Representation of the parameters d and φ .

From our experiments and the desired precision, we have associated five linguistic values for the distance d (VS: Very Small; S: Small; M: Medium; L: Large; VL: Very Large) and seven linguistic values for the angle φ (NL: Negative Large; NM: Negative Medium; NS: Negative Small; Z: Zero; PS: Positive Small; PM: Positive Medium; PL: Positive Large).

B. Rules basis

This part consists in determining relations between fuzzy input variables (the distance d and the angle φ) and fuzzy output variables (right speed V_R and left speed V_L). These rules are given by:

If (d is A_i) and (φ is B_i) then ($V_R=y_i$ and $V_L=z_i$) with $i = 1, 2, \dots, r$, where r is the rule number.

The fuzzy inference table is manually constructed from several simulations and experiment tests. Tables I and II represent the suggested fuzzy rules.

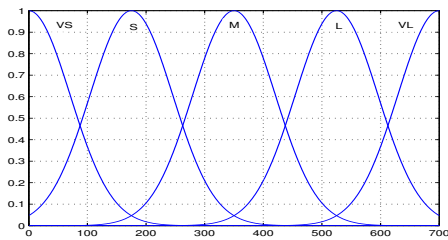


Fig. 4. Membership functions for the distance d (in mm).

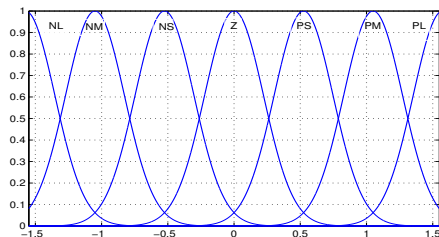


Fig. 5. Membership functions for the angle φ (in rad).

TABLE I
INFERENCE TABLE FOR THE RIGHT VELOCITY.

		φ						
		V_R	NL	NM	NS	Z	PS	PM
d	VS	S	S	Z	Z	S	M	L
	S	M	S	S	S	M	L	VL
	M	L	M	S	M	L	VL	VL
	L	L	L	M	L	VL	VL	VL
	VL	VL	VL	VL	VL	VL	VL	VL

TABLE II
INFERENCE TABLE FOR THE LEFT VELOCITY.

		φ						
		V_L	NL	NM	NS	Z	PS	PM
d	VS	L	M	S	Z	Z	S	S
	S	VL	L	M	S	S	S	M
	M	VL	VL	L	M	S	M	L
	L	VL	VL	VL	L	M	L	L
	VL	VL	VL	VL	VL	VL	VL	VL

C. Fuzzy controller outputs

Fuzzy controller outputs are the robot right and left wheel velocities. They are defined by:

$$V_R = \frac{\sum_{i=1}^r \alpha_i y_i}{\sum_{j=1}^r \alpha_j} \quad (6)$$

$$V_L = \frac{\sum_{i=1}^r \beta_i z_i}{\sum_{j=1}^r \beta_j} \quad (7)$$

with α_i and β_i are the level activation of the rule i .

The elaborated fuzzy controller is a Sugeno fuzzy logic system of order zero. So, fuzzy rule consequences are constants. Then, we have chosen the five following values as linguistic variables of the fuzzy rule consequences: Z: Zero; S: Small; M: Medium; L: Large; VL: Very Large. Based on simulations and experimentation tests, we have associated to each linguistic variable a numerical value (see figure 6). With these chosen values, the robot is able to reach its target for different configurations (see figure 7). To ameliorate the robot's trajectories, we have to change the values of fuzzy rule consequences. For that, fuzzy controller parameters are optimised using genetic algorithms.

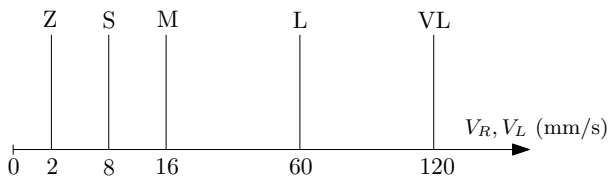


Fig. 6. Numerical values to the linguistic variables.

D. Optimisation by genetic algorithms

Optimization problem is frequently encountered in several fields. For example, when:

- the environment is very complex.
- the problem has a great number of parameters which must be optimized simultaneously.
- there are several constraints which must be satisfied simultaneously.
- the system is not-differentiable or discontinuous.

Genetic algorithms constitute a solution to these problems.

In fact, and contrary to gradient algorithms [12] which are analytical methods, genetic algorithms [13], [14] are iterative and random methods which require a few information for their use (such as the possible search space and the performance criterion). In our work, genetic algorithms are used to optimise the consequence of fuzzy rules. For that, we have adopted the following performance criterion J (see equation (8)) which expresses the distance covered by the robot during its motion from an initial position towards another desired position.

$$J = \int_0^\tau V dt \quad (8)$$

with τ is the workspace time and V is the magnitude of the robot speed.

Parameters to be optimized are the consequences y_i and z_i relating to the right and the left velocities, respectively [8]. The initial population is randomly chosen. It is formed by 10 individuals. Each individual represents the consequence of the fuzzy rules. The structure of an individual is represented as follows:

$$\text{individu}_i = \underbrace{y_1^i \ y_2^i \ y_3^i \ \dots \ y_n^i}_{V_R} \ \underbrace{z_1^i \ z_2^i \ z_3^i \ \dots \ z_n^i}_{V_L} \quad (9)$$

with $i = 1, 2, \dots, m$ and $m = 10$.

E. Simulation results

Figure 7 shows the evolution of the mobile robot. Based on this figure, we notice that the robot is able to reach the target, for different configurations. It is clear that the use of genetic algorithms, can optimize the trajectory of the mobile robot.

IV. HIERARCHICAL FUZZY CONTROLLER

In this part, we will treat a mobile robot's command problem in an unknown environment containing fixed obstacles. For that, we have proposed an approach allowing the robot to join the target and to avoid collisions with detected obstacles. The principle idea of this approach is inspired from the potential field method. The robot is then attracted to the target and pushed from obstacles. In this approach, we have applied hierarchical fuzzy systems to ensure the robot navigation in an occupied environment. These hierarchical fuzzy systems are composed from several fuzzy subsystems, of a lower size.

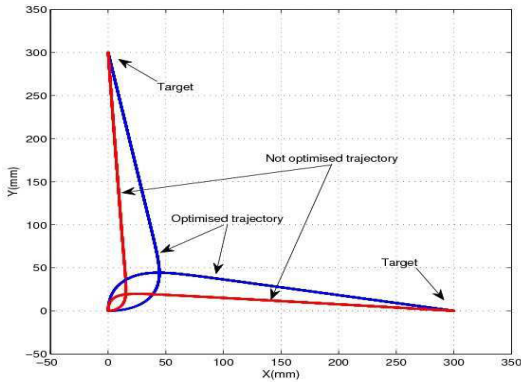


Fig. 7. Simulation results of the robot Khepera II before and after the FLC optimization.

In the case of standard fuzzy systems, the fuzzy rule number increases exponentially with the number of input variables. The advantage of hierarchical fuzzy systems is to reduce the fuzzy rule number; In this case, this number increases linearly with the number of input variables [15], [16].

This approach is inspired from Abdessemed's work [10]. It consists in proposing a new hierarchical structure to avoid obstacles. This structure allows to decrease the number of elaborated fuzzy rules. In our case, the robot contains eight sensors. Each sensor measures in real time the distance between the robot and the obstacle. If for each distance we associates five membership functions, the number of fuzzy rules will be then $5^8 = 390625$. With this high significant number of fuzzy rules, the simulation time increases. In this case, the hierarchical structure makes it possible to simplify the problem and to reduce the number of fuzzy rules to 370.

The fuzzy logic controller must receive instantaneously the following four inputs characterizing the local environment of the robot:

1. d_T : the distance separating the robot and the target.
2. φ_T : the angle between the robot axis and the target .
3. d_o : the distance separating the robot and the obstacle.
4. φ_o : the angle between the robot axis and the obstacle.

From these inputs, the fuzzy controller must be able to react following a given situation. To avoid collision, we have added a second controller which uses the same fuzzy rules elaborated previously when the environment is without obstacles. This controller is responsible to move the robot away from the obstacles. These two controllers operate as follows: in real time, the first controller computes the right and the left velocities V_{rt} and V_{lt} to find the target direction. The second controller computes the right and the left velocities V_{ro} and V_{lo} to find the obstacle direction. In fact, we assimilate that the obstacle is a target. The inversion of these two velocities (i.e. right velocity V_{ro} takes left velocity value V_{lo} and conversely) allows the robot to move away from the obstacle. Finally, the two velocities ensuring the robot movement in a cluttered environment are the sum of the velocities provided by each controller. They are expressed

as follow:

$$V_R(k) = V_{rt}(k) + V_{lo}(k) \quad (10)$$

$$V_L(k) = V_{lt}(k) + V_{ro}(k) \quad (11)$$

with

$$V_{rt} = \frac{\sum_{i=1}^r \mu_i(d_t, \varphi_t) y_i}{\sum_{j=1}^r \mu_j(d_t, \varphi_t)} \quad (12)$$

$$V_{lt} = \frac{\sum_{i=1}^r \mu_i(d_t, \varphi_t) z_i}{\sum_{j=1}^r \mu_j(d_t, \varphi_t)} \quad (13)$$

$$V_{ro} = \frac{\sum_{i=1}^r \mu_i(d_o, \varphi_o) y_i}{\sum_{j=1}^r \mu_j(d_o, \varphi_o)} \quad (14)$$

$$V_{lo} = \frac{\sum_{i=1}^r \mu_i(d_o, \varphi_o) z_i}{\sum_{j=1}^r \mu_j(d_o, \varphi_o)} \quad (15)$$

where d_o is the distance between the robot and the obstacle and φ_o is the angle from the robot to the obstacle.

Khepera II robot has eight sensors distributed on its contour as it is shown on figure 8. To simplify the problem, we joined together:

- Sensors C_1 , C_2 and C_3 in order to test the obstacle's existence on the robot's left side.
- Sensors C_4 , C_5 and C_6 in order to test the obstacle's existence on the robot's right side.
- Sensors C_7 and C_8 in order to test the obstacle's existence on the robot's back side.

From this decomposition, we have obtained three new positions (on the right, on the left and on the back) to locate the obstacles. The fuzzy hierarchical structure suggested is presented in figure 9.

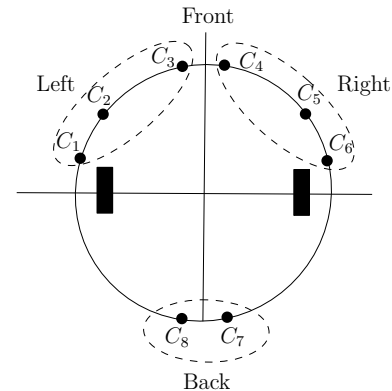


Fig. 8. Position of the eight sensors .

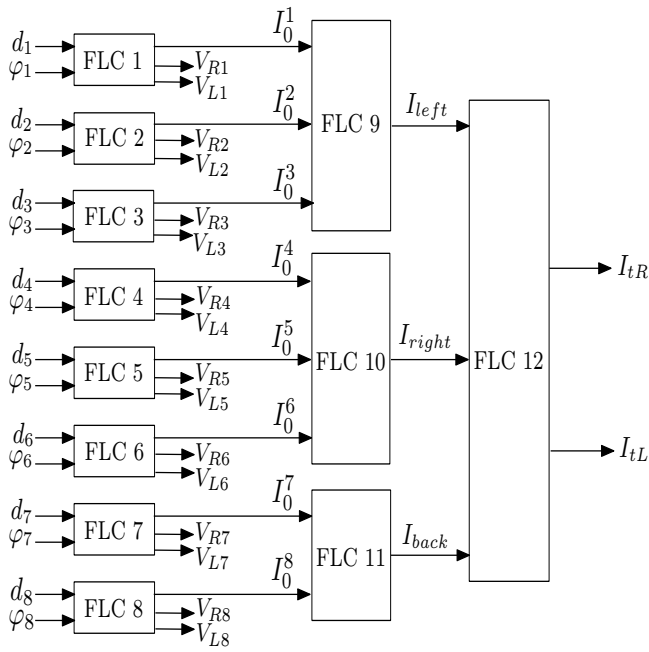


Fig. 9. Block diagram of the hierarchical fuzzy system.

In this structure, we notice the existence of twelve fuzzy controllers. The eight first ones are related to each sensor of the robot. Each controller has three outputs: the index I_0^i which indicates the collision degree between the robot and the obstacle and the two speeds: right (V_{Ri}) and left (V_{Li}), $i = 1, \dots, 8$, and

$$V_{Ri} = \frac{\sum_{j=1}^r \mu_j(d_i, \varphi_i) y_j}{\sum_{k=1}^r \mu_k(d_i, \varphi_i)} \quad (16)$$

$$V_{Li} = \frac{\sum_{j=1}^r \mu_j(d_i, \varphi_i) z_j}{\sum_{k=1}^r \mu_k(d_i, \varphi_i)} \quad (17)$$

The set of the ninth fuzzy controllers has as inputs the indices I_0^1 , I_0^2 and I_0^3 and gives as output a new index I_{left} which indicates the collision degree between the robot and the obstacle on the left side. In the same way, the indices I_0^4 , I_0^5 and I_0^6 give through the controller FLC 10, a new index I_{right} which indicates the collision degree between the robot and the obstacle on the right side. The indices I_0^7 and I_0^8 give through the controller FLC 11 a new index I_{back} which indicates the collision degree between the robot and the obstacle on the back of the robot. Finally, the controller FLC 12 has as inputs the indices I_{left} , I_{right} and I_{back} and gives as outputs two indices I_{tR} and I_{tL} which indicates the absence degree of obstacles, so that the robot can find its target.

Expressions of the robot's right and left wheel's velocity ensuring the robot movement and the obstacle avoidance

become:

$$V_R = I_{tR} V_{tR} + \sum_{i=1}^n I_0^i V_{Li} \quad (18)$$

$$V_L = I_{tL} V_{tL} + \sum_{i=1}^n I_0^i V_{Ri} \quad (19)$$

where n is the sensor number.

The inference rules of the index I_0^i are given by table III. This table indicates the collision degree between the robot and the obstacle. For that, we have chosen three values which are 0, 1 and 2 to indicate the obstacle position compared to the robot. The value 0 indicates that there are no obstacles, the value 1 indicates that the obstacle is relatively near to the robot, whereas the value 2 indicates that the obstacle is very near to the robot.

According to table III, we can notice that the fuzzy inference system is used to analyze the obstacle position with respect to the robot. This table contains 35 rules base. For example, to explain these rules the following rule is taken :

If (d_i is VS) and (φ_i is Z) then (I_0^i is 2)

i.e.: if the obstacle is at a very small distance and in the same direction of the robot then there is a danger of collision. The expression of the index I_0^i is the following:

$$I_0^i = \frac{\sum_{j=1}^{35} \mu_j(d_i, \varphi_i) I_j}{\sum_{k=1}^{35} \mu_k(d_i, \varphi_i)} \quad (20)$$

To each index I_0^i we associate three Gaussian membership functions as shown in figure 10, with Z: Zero, M: Medium and L: Large.

Tables IV, V and VI represent respectively the inference rules of indices I_{left} , I_{right} and I_{back} .

Now, we consider the indices I_{left} , I_{right} and I_{back} as

TABLE III

COLLISION DEGREE BETWEEN THE ROBOT AND THE OBSTACLE

I_0^i		φ_i						
		NL	NM	NS	Z	PS	PM	PL
d_i	VS	2	2	2	2	2	2	2
	S	1	2	2	2	2	2	1
	M	0	1	1	1	1	1	0
	L	0	0	0	1	0	0	0
	VL	0	0	0	0	0	0	0

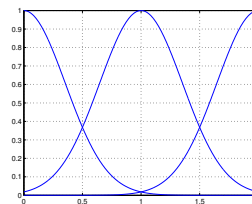


Fig. 10. Membership functions for the index I_0^i .

TABLE IV
DISCRETE VALUES FOR THE INDEX I_{left}

I_{left}	I_0^1									
	Z			M			L			
	Z	M	L	Z	M	L	Z	M	L	
I_0^3	Z	0	1	2	0	1	0	1	1	2
	M	1	1	2	1	1	2	1	1	2
	L	2	2	2	2	2	2	2	2	2

TABLE V
DISCRETE VALUES FOR THE INDEX I_{right}

I_{right}	I_0^4									
	Z			M			L			
	Z	M	L	Z	M	L	Z	M	L	
I_0^6	Z	0	1	2	1	1	2	2	2	2
	M	0	1	2	1	1	2	2	2	2
	L	1	1	2	1	1	2	2	2	2

TABLE VI
DISCRETE VALUES FOR THE INDEX I_{back}

I_{back}	I_0^7			
	Z	M	L	
I_0^8	Z	0	1	2
	M	1	1	2
	L	2	2	2

inputs of the fuzzy controller FLC 12 in order to calculate the indices I_{tL} and I_{tR} .

In tables VII and VIII, we have also used three numerical values as indices:

- The index 0: shows that the obstacle is very near to the robot; The robot must then modify its direction.
- The index 1: shows that the obstacle is relatively near to the robot; The robot can advance with a low speed and modify its direction.
- The index 2: shows that there is no obstacle; So, the robot can advance safely.

The expression of indices I_{tR} and I_{tL} are given by the following equations, respectively:

$$I_{tR} = \frac{\sum_{j=1}^3 \sum_{i=1}^3 \mu_{ij} I_{ij}^R}{\sum_{l=1}^3 \sum_{k=1}^3 \mu_{kl}} \quad (21)$$

$$I_{tL} = \frac{\sum_{j=1}^3 \sum_{i=1}^3 \mu_{ij} I_{ij}^L}{\sum_{l=1}^3 \sum_{k=1}^3 \mu_{kl}} \quad (22)$$

TABLE VII
DISCRETE VALUES FOR THE INDEX I_{tL}

I_{tL}	I_{left}									
	Z			M			L			
	Z	M	L	Z	M	L	Z	M	L	
I_{back}	Z	2	1	0	2	2	0	2	2	0
	M	2	1	0	2	2	0	2	2	0
	L	2	1	0	2	1	0	2	2	0

TABLE VIII
DISCRETE VALUES FOR THE INDEX I_{tR}

I_{tR}	I_{left}									
	Z			M			L			
	Z	M	L	Z	M	L	Z	M	L	
I_{back}	Z	2	2	2	1	1	2	0	0	2
	M	2	2	2	1	1	2	0	0	2
	L	2	2	2	1	2	2	0	0	2

V. SIMULATION RESULTS

Based on sensor information, the robot must be able to reach its target without collision with obstacles. For that, several tests for different configurations of the initial orientation, the obstacle positions and the target position have been carried out. According to simulation results (see figures 11 and 12), it's clear that the robot can move in an environment containing one or two obstacles. This validates the approach of hierarchical fuzzy systems used to command the robot in an environment containing static obstacles.

Tables IX and X presents a comparison between several criterion values before and after optimisation, for different positions of the target and obstacles. It's obvious that the obtained trajectories using genetic algorithms are the best ones.

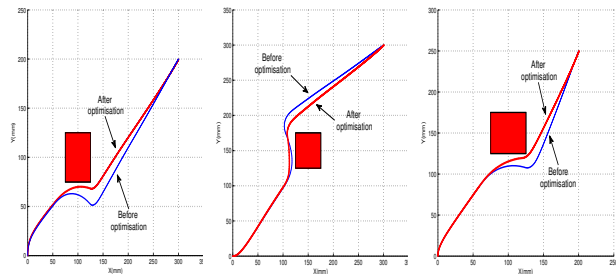


Fig. 11. Simulation results: one obstacle

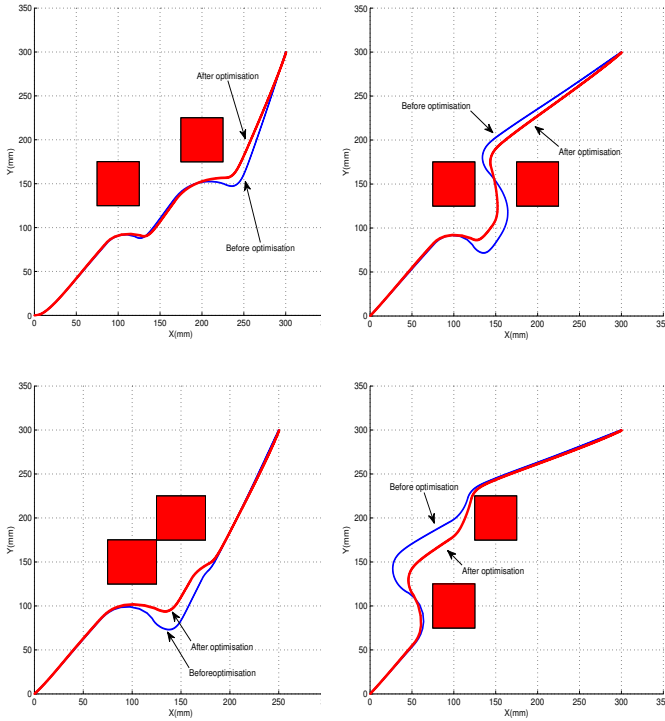


Fig. 12. Simulation results: two obstacles

TABLE IX

COMPARISON BETWEEN DISTANCES COVERED BEFORE AND AFTER OPTIMIZATION: ONE OBSTACLE.

Depart point	(0,0)	(0,0)	(0,0)
Target position	(300,200)	(300,300)	(200,250)
Obstacle position	(100,100)	(150,150)	(100,150)
Initial orientation (rad)	$\frac{\pi}{2}$	0	$\frac{\pi}{2}$
Distance before optimisation (mm)	385.27	469.68	343.78
Distance after optimisation (mm)	374.65	455.44	332.23
Relative errors (%)	2.83	3.12	3.47

TABLE X

COMPARISON BETWEEN DISTANCES COVERED BEFORE AND AFTER OPTIMIZATION: TWO OBSTACLES.

Depart point	(0,0)	(0,0)	(0,0)	(0,0)
Target position	(300,300)	(300,300)	(250,300)	(300,300)
Obstacles positions	(100,150) (200,200)	(100,150) (200,150)	(100,150) (150,200)	(100,100) (150,200)
Initial orientation (rad)	0	$\frac{\pi}{4}$	$\frac{\pi}{4}$	$\frac{\pi}{4}$
Distance before optimisation (mm)	470.48	516.61	448.69	507.11
Distance after optimisation (mm)	456.12	469.93	422.69	477.49
Relative errors (%)	3.14	9.93	6.15	6.20

VI. CONCLUSION

In this paper, we have treated the autonomous navigation problem of a wheeled mobile robot in an environment con-

taining obstacles. For that, we have developed an approach which uses hierarchical fuzzy systems to bring the robot towards the desired position. An optimization of the different obtained trajectories has been carried out using genetic algorithms. Simulation results show the efficiency of the proposed approach.

REFERENCES

- [1] Th. Fraichard and H. Asama, "Inevitable collision states - a step towards safer robots", *Advanced Robotics*, 18 (10): 1001-1024, (2004).
- [2] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", *International Journal of Robotics Research*, 5 (1): 90-98, (1986).
- [3] W. Feiten, R. Bauer and G. Lawitzky, "Robust obstacle avoidance in unknown and cramped environments", *IEEE International Conference on Robotics and Automation* (1994).
- [4] D. Fox, W. Burgard and S. Thrun, "The Dynamic Window Approach to Collision Avoidance", *IEEE Robotics and Automation Magazine*, 4 (1): 23-33, March (1997).
- [5] A. Fahmi, L. Khriji, A. Al-Yahmadi and N. Masmoudi, "Contribution of fuzzy logic based autonomous robot navigation in unknown environment", *Proc. of IEEE conf. on Signals, System and Decision, SSD'07*, March 19-22, Hammamet, Tunisia, (2007).
- [6] H. Maaref and C. Barret, "Sensors-based fuzzy navigation of an autonomous mobile robot in an indoor environment", *Robotics and Autonomous System*, 38: 1-18, (2002).
- [7] M. Sugeno and M. Nishida, "Fuzzy control of a model car", *Fuzzy Sets and Syst.*, 16: 103-113, (1985).
- [8] N. Yousfi, Ch. Rekik, M. Jallouli and N. Derbel, "Optimised fuzzy controller for mobile robot navigation in a cluttered environment", *Conf. SSD'2010*, Amman, Jordan, Juin 27-30, (2010).
- [9] R.-E. Precup, H. Hellendoorn, "A survey on industrial applications of fuzzy control", *Computers in Industry* 62 (3): 213-226, (2011).
- [10] F. Abdessemed, K. Benmahammed and E. Monacelli, "A fuzzy-based reactive controller for a non-holonomic mobile robot", *Robotics and Autonomous Systems*, 47: 31-46, (2004).
- [11] H. Mezaache and F. Abdessemed, "Simulation of the navigation of a mobile robot by the Q-Learning using artificial neuron networks", University Hadj Lakhdar of Batna-Algeria-Faculty of Engineering Department of Electronics, (2009).
- [12] H. Nomura, I. Hayashi and N. Wakami, "A learning method of fuzzy inference rules by descent method", *Proc. of IEEE Int. Conf. on fuzzy systems*, USA, 203-210, (1992).
- [13] D. E. Goldberg, "Genetic Algorithms in Search, Optimisation, and Machine Learning", *New York: Addison-Wesley professional*, (1989).
- [14] J. J. Holland, "Les AGs, pour la science", ndeg. 179, p. 44, (1992).
- [15] M. Brown, K.M. Bossley, D.J. Mills and C.J. Harris, "High dimensional neurofuzzy systems: overcoming the curse of dimensionality", *Proc. IEEE Internat. Conf.*, 2139-2146, (1995).
- [16] G.V.S. Raju, J. Zhou and R.A. Kisner, "Hierarchical fuzzy control", *Int. J. Control*, 54 (5): 1201-1216, (1991).