Available online at www.sciencedirect.com
SCIENCE DIRECT®

Robotics and Autonomous Systems 47 (2004) 31–46

Robotics and Autonomous Systems
www.elsevier.com/locate/robot

A fuzzy-based reactive controller for a non-holonomic mobile robot

Foudil Abdessemed [a,*], Khier Benmahammed [b], Eric Monacelli [c]

[a] Dpt. d'Electronique, Université de Batna, Rue Chahid Boukhlouf, Batna 05000, Algeria
[b] Dpt. d'Electronique, Université de Setif, Setif 19000, Algeria
[c] Laboratoire LIRIS, 10 Avenue de l'Europe 78140 Vélizy, Paris, France

Received 17 December 2002; received in revised form 22 October 2003; accepted 10 February 2004

Abstract

This paper presents the theoretical development of a complete navigation problem of an autonomous mobile robot. The situation for which the vehicle tries to reach the endpoint is treated using a fuzzy logic controller. The problem of extracting the optimized IF–THEN rule base is solved using an evolutionary algorithm. A new approach based on fuzzy concepts is presented in this paper to avoid any collision with the surrounding environment when this latter becomes relatively complex. Simulation results show that the designed fuzzy controller achieves effectively any movement control of the vehicle from its current position to its end motion and without any collision.

Keywords: Mobile robot; Evolutionary programming; Obstacle avoidance; Fuzzy controller

1. Introduction

Navigation and obstacle avoidance are very important issues for the successful use of an autonomous mobile robot. When computing the configuration sequence, we allow the robot to move from one position to another. When the environment of the robot is obstacle free, the problem becomes less complex to handle. But as the environment becomes complex, motion planning needs much more treatments to allow the robot to move between its current and final configurations without any collision within the surrounding environment. The objective of this paper is to show how to guide an autonomous mobile robot in an unknown environment using a fuzzy logic controller. This present interest is largely due to the successful applications of fuzzy logic controller to a variety of industrial systems. Its main components are an inference engine and a set of linguistic IF–THEN rules that encode the behavior of the mobile robot. However, the main difficulty in designing a fuzzy logic controller is the efficient formulation of the fuzzy IF–THEN rules. If it is easy to produce the antecedent parts of a fuzzy rule base, it is however very difficult to produce the consequent parts without expert knowledge. Usually, they are generated using heuristic thinking [1,2] or modeling an expert's actions [3]. Other researchers have investigated the problem by suggesting an adaptive fuzzy control to estimate and adjust the parameters of the fuzzy controller [4,5]. Clustering algorithms could be another approach to be applied for automatically generating the fuzzy rules [6]. In recent years and due to the availability of powerful computer platform,

* Corresponding author.
E-mail address: fodil_a@hotmail.com (F. Abdessemed).

0921-8890/$ – see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2004.02.006

the theory of evolutionary algorithms starts to become popular to the problem of parameter optimization, although its origins can be traced back to the 1960s. It is regarded as a parallel global search technique evaluating many points in the parameter space and it is more likely to converge towards the global solution. Genetic algorithm as one approach to the implementation of evolutionary algorithms was used to generating the rules of the cart-pole balancing fuzzy logic controller [7]. We have proposed a genetic algorithm to extracting the rules of a fuzzy controller intended to control the end effector motion of a planar manipulator in purpose to follow a prescribed trajectory [8]. Recently, Shi et al. [9] have suggested a genetic algorithm to evolve a fuzzy expert system. In this paper, an evolutionary programming technique is proposed to design fuzzy logic control rules and seeks the optimal values of the consequent parts in order to drive an autonomous mobile robot from any position in the workspace to its final end point without any deviation. On the other hand, when the environment is cluttered with dynamic or static objects, it becomes a complex problem to plan the actions of the mobile robot. When using global methods, the general configuration model assumes that the robot world space is known a priori, see for instance [10]. Local methods, on the other hand, use transducer information to generate the best motion of the system from the sensory data obtained on line. The obstacle avoidance relies on a relatively poor, but easy modeling of the local environment. The robot reacts instantaneously to every intrusion in its perception area, like a muscle to a stimulus. This reactive avoidance defines a relation between the perception and the action. The obstacle avoidance is often associated to an anti collision behavior. The main objective is to keep away from collision. However, in this mode, the path followed is not optimal. The potential field method developed by Khatib [11], and which was an on-line collision avoidance approach, is applicable when the robot does not have a priori model of the obstacles, but senses them during motion execution. The potential function is defined over free space as the sum of an attractive potential pulling the robot towards the goal configuration and a repulsive potential pushing the robot away from the obstacles. However, even when used with graph searching techniques, the potential field method does not guarantee the attainment of the goal and may get stuck at a local minimum. To overcome this drawback, other methods have been investigated, for instance in [12], the authors proposed a powerful planner that combines the technique of the potential field with a randomized search techniques. On the other hand, fuzzy controllers are a convenient choice for systems that involve varying degrees of uncertainty. It provides tools that are of potential interest to mobile robot control. Wijesoma et al. [13] applied a fuzzy behavioral approach to navigate a mobile robot using fuzzy propositions. In this area, one can find, for instance, the work of Safiotti [14], where a behavioral based approach has been described to control mobile robot based on the use of fuzzy logic to generate collision free motion. In this paper a real time efficient method inspired from the potential field method and which uses the fuzzy principles to navigate an autonomous robot in an unknown environment is proposed. We have called it a *fuzzy image method* [15]. The obstacle avoidance behavior was designed so as to get a model of the local environment. This region has to be avoided whenever the vehicle approaches it. We note however that it would be rather difficult to compare methods or prefer a method upon others. But, the approach we propose is simpler in its design and in its formulation. In our scheme, only one rule base is used for both navigation behavior and collision avoidance behavior. The rules are optimized using an evolutionary algorithm. This paper is organized as follows. In Section 2, the kinematic model of the mobile robot is described. In Section 3, the basic principles of fuzzy controller design are introduced. Section 4 deals with the design procedure of the evolutionary programming scheme. In Section 5, the basic principles of the fuzzy image method are discussed. Section 6 treats the case of multipoint obstacle avoidance. Simulation and results are discussed in Sections 7 and 8 concludes the paper.

2. Vehicle kinematics

Many kinematic models describing the motion of various vehicle configurations have been proposed; see for instance [16,17]. In this section, a kinematic description of a mobile robot is given. The vehicle has two driving wheels at the rear corners and two passive supporting wheels at the front corners. Two DC motors independently drive the two rear wheels. The vehicle

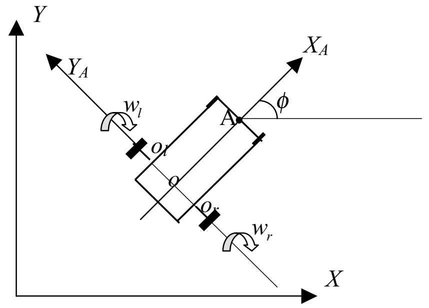

Fig. 1. Schematic representation of mobile robot.

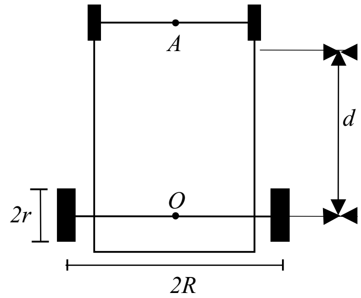

Fig. 2. Geometric characteristic of the mobile robot.

presents however two constraints. It is non-holonomic, which means that it must move in the direction of the axis of symmetry, i.e.,

$$\dot{y}_A - \dot{x}_A \tan\phi = 0, \tag{1}$$

ϕ is the heading angle of the vehicle from the X-axis of the world coordinates (see Fig. 1).

Writing the classical relationships between the velocity point O, and those of points O_l and O_r, we can easily determine the linear velocity $\vec{v}_o$, and the instantaneous angular velocity $\vec{\omega}$ of the mobile robot:

$$\vec{v}_o = \vec{v}_{o\mathrm{r}} + o\vec{o}_\mathrm{r} \wedge \vec{\omega}, \tag{2}$$

$$\vec{v}_o = \vec{v}_{o\mathrm{l}} + o\vec{o}_\mathrm{l} \wedge \vec{\omega} \tag{3}$$

$$\vec{\omega} = \dot{\phi} \cdot \hat{k}, \tag{4}$$

$\hat{k}$ is the unit vector along the Z_A axis, and $\vec{v}_{o\mathrm{l}}$ and $\vec{v}_{o\mathrm{r}}$ are the linear velocities of the mobile robot points O_l and O_r, respectively. When projecting expressions (2) and (3) on the X-axis and the Z-axis, we get the expressions of v_o and $\dot{\phi}$ as follows:

$$v_o = \tfrac{1}{2} r(\omega_\mathrm{r} + \omega_\mathrm{l}), \tag{5}$$

$$\dot{\phi} = \frac{r}{2R}(\omega_\mathrm{r} - \omega_\mathrm{l}), \tag{6}$$

r and R are, respectively, the radius of the wheels and the width of the vehicle as it is shown in Fig. 2. It has been proven by Samsung and Abderrahim [18] that the vehicle converges better to its reference when controlling a point located in front of the rear wheel axis. In this paper point A, as it is obvious from Fig. 2, which is located at a distance d from point O, has been chosen to be the position control of the vehicle such that:

$$x_A = x_o + d\cos\phi, \tag{7}$$

$$y_A = y_o + d\sin\phi, \tag{8}$$

where

$$x_o(t+1) = x_o(t) + \Delta D\cos(\phi + \tfrac{1}{2}\Delta\phi), \tag{9}$$

$$y_o(t+1) = y_o(t) + \Delta D\sin(\phi + \tfrac{1}{2}\Delta\phi), \tag{10}$$

such that

$$\Delta D = \tfrac{1}{2} r(\Delta q_\mathrm{r} + \Delta q_\mathrm{l}), \tag{11}$$

$$\Delta\phi = \frac{r}{2R}(\Delta q_\mathrm{r} - \Delta q_\mathrm{l}), \tag{12}$$

where (x_o, y_o) and (x_A, y_A) denote the coordinates of points O and A, respectively, whereas Δq_r and Δq_l are the angular steps of the right and left wheels, respectively.

3. Fuzzy controller design

Originally advocated by Zadeh [19] and Mamdani and Assilian [2], fuzzy logic has become a mean of collecting human knowledge and experience and dealing with uncertainties in the control process. Now, fuzzy logic is becoming a very popular topic in control engineering. Considerable research and applications of this new area for control systems have taken place. Fuzzy control is by far the most useful application, but its successful solutions to a variety of consumer products and industrial systems, helped to attract growing attention and interests. Thereafter, a situation for which the vehicle tries to reach an end point is examined. From its design simplicity, its implementation,

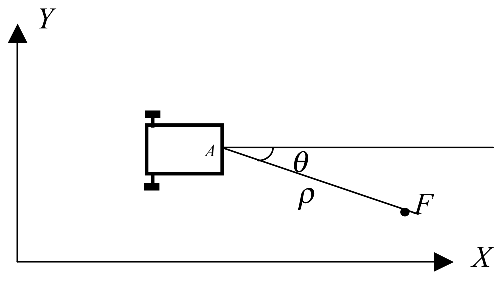

Fig. 3. Example of situation 'reaching a point'.

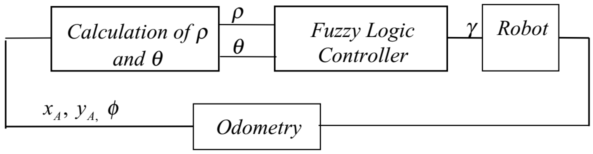

Fig. 4. Block diagram of the controlled system.

and its robustness properties, a fuzzy logic controller is used in order to control the navigation behavior of an autonomous mobile robot. If we consider the vehicle moving in a free obstacle environment, then the optimal trajectory from its current position to its end configuration is naturally a line joining these two extreme points as it is depicted for instance by Fig. 3, where θ is the angle between the symmetric axis of robot and the line that joins the control point of the robot to its final point.

If we link the points with segments, then the goal is to control the driving point A of the autonomous robot with respect to these segments and to come the closest possible to the end point. The distance ρ becomes zero when the vehicle stabilizes at its final configuration. Fig. 4 gives a schematic block diagram of this architecture. From this figure one can notice that the inputs to the fuzzy controller are ρ and θ, and its output is the steering angle γ.

3.1. Fuzzification

The best fuzzy system is implemented with five and eight triangular membership functions for the controller input variables ρ and θ, respectively, on each normalized universe of discourse with 50% overlap. A higher number is not necessary since it leads to intermediate configurations that are implicitly described. Moreover, the non-symmetry of the membership functions of the distance variable helps in a fine control. Although, there is no restriction on the form of membership functions, we choose the piecewise linear description (see Fig. 5). Associated with the map, is a collection of linguistic values:

$$L_\rho = \{\mathrm{VS}, \mathrm{S}, \mathrm{M}, \mathrm{L}, \mathrm{VL}\}, \quad (13)$$

$$L_\theta = \{\mathrm{PVL}, \mathrm{PL}, \mathrm{PM}, \mathrm{PS}, \mathrm{NS}, \mathrm{NM}, \mathrm{NL}, \mathrm{NVL}\}. \quad (14)$$

The meaning of each linguistic value in the term sets L_ρ and L_θ should be clear from its mnemonics; for example, VS stands for *very-short*, S for *short*, M for *medium*, L for *large*, and VL for *very-large*. For the term set L_θ, PVL stands for *positive-very-large*, PL for *positive-large*, PM for *positive-medium*, PS for *positive-small*, and likewise for the negative (*N*) linguistic values. The number of possible combinations is $L = NxK$, which is equal to the number of rules needed to specify a single output, where N and K are the cardinal numbers of the sets L_θ and L_ρ, respectively.

3.2. Fuzzy inference mechanism

The second step in designing an FLC is the fuzzy inference mechanism. For instance, the knowledge base

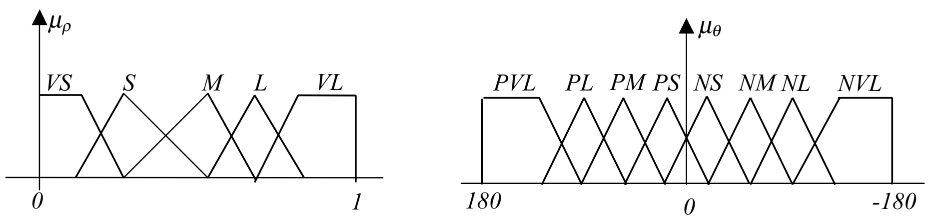

Fig. 5. Membership functions for the variables ρ and θ.

of the system consists of rules in the form

$$\text{Rule } R_{ij}:\quad \text{IF } \rho \text{ is S AND } \theta \text{ is NM THEN } \gamma \text{ is } n_p,$$
$$i = 1, 2, \dots, N,\quad j = 1, 2, \dots, K,$$
$$p = 1, 2, \dots, L,$$

such that

$$\gamma(k) = F[\rho(k), \theta(k)], \tag{15}$$

$\rho(k)$ and $\theta(k)$ are inputs to the map F, and the output $\gamma(k)$ denotes a numerical value within the interval $[-90^\circ, +90^\circ]$, characterizing the relative variation in the direction that should be taken by the vehicle to reach the final point. In this application, the operator min is used as a t-norm operator. The rules could be defined by using the human-like description of the vehicle's movement behavior [20]. But, this approximate human reasoning may lead to certain unsatisfactory rules. Furthermore, the global behavior of the vehicle may result from the combination of several basic behaviors for instance, the trajectory tracking, the vehicle speed, the obstacle avoidance. As a matter of fact, it is not obvious to define the respective rules. In the next section, and as an alternative to the aforementioned method, we suggest an evolutionary algorithm as an efficient solution for the extraction of the rules.

3.3. Defuzzification

There are many ways for performing defuzzification. The strategy adopted here is the height defuzzification method. This method is simple and very quick. It uses the individual scaled control outputs, and builds the weighted sum of the peak values, as it is clear from the following equation:

$$\gamma = C_{\mathrm{F}} \frac{\sum_{i=1}^{L} \mu(\gamma_i)\gamma_i}{\sum_{i=1}^{L} \mu(\gamma_i)}, \tag{16}$$

where γ_i is the support value at which the membership function reaches the maximum value. CF is the scaling factor defined for the output universe of discourse and is made equal to 90; L designates the number of rules used. For instance, if we take all the rules, then L would be equal to 40; which is the maximum number of possible combination between the number of fuzzy subsets of the distance and those of orientation.

4. Evolutionary programming

Evolutionary algorithms are a class of probabilistic optimization techniques. They are especially well suited for solving difficult optimization problems. Evolutionary algorithms are of three types: genetic algorithms, introduced by Holland during the 1960s [21] and subsequently improved by DeJong [22], Goldberg [23] and others. Evolutionary programming was developed by Fogel et al. [24], originally offered as an attempt to create artificial intelligence. Evolution strategies initially developed in Germany by Rechenberg [25] and refined by Schwefel [26], and Herdy [27], were designed initially to solve difficult parameter optimization problems. The basic idea of an optimization problem using an evolutionary algorithm is to start with a finite initial population of individuals representing a set of free parameters of the system under consideration. The population is made to evolve in a stochastic manner towards better regions of search space. Individuals with high fitness values are favored to survive for the next generation. The design of an evolutionary algorithm obeys to three main operators: selection, recombination and mutation. The main structure of a general evolutionary algorithm may be viewed in [28]. In the following, we describe the structure of the evolutionary programming developed here as a strategy for the purpose of a extracting the optimized consequences of the fuzzy rules. A $(\mu + \lambda)$-evolutionary programming is described by an octuple entity defined by the following format:

$$\mathrm{EP} = \{I(t), L, \mu, \lambda, \mathrm{sel}, p_{\mathrm{mut}}, f, g\} \tag{17}$$

for which the components are defined as follows:

$I = [a_1, a_2, \dots, a_{2L}]$	encoding chromosome
$2L \in \aleph$	length of chromosome
$\mu \in \aleph$	population size
$\lambda \in \aleph$	number of offspring $(=\mu)$
$p_{\mathrm{mut}} : I \to I$	mutation operator
$f : \mathfrak{R}^L \to \mathfrak{R}$	fitness function
$g : \mathfrak{R}^L \to \mathfrak{R}$	set of constraints

The design of the EP is based mainly on three mechanisms:

- the representation of individuals;
- implication of the variation operators;
- the generation procedure.

4.1. Representation

Each individual chromosome represents a complete rule base solution. The components: $(a_1 = m_1, a_2 = m_2, \dots, a_L = m_L)$ determine the consequent part of the fuzzy rules, and the remaining components, $(a_{L+1} = \sigma_1, a_{L+2} = \sigma_2, \dots, a_{2L} = \sigma_L)$ contain the standard deviation, which controls the mutation process. A complete string of chromosome could be written in the following way: $a_1 a_2, \dots, a_L a_{L+1} a_{L+2}, \dots, a_{2L}$, representing one individual. The set of all the individuals represent a population. If we denote by $P(k)$ a population at a time k, then we can write

$$P(k) = \bigcup_{i=1,\dots,\mu} I^i(k) \quad \text{with } I^i = \{a^i_j, i = 1, \dots, \mu, j = 1, \dots, 2L\},$$

I^i is the ith individual in which the components a_j describe the consequent parts of the rules and the standard deviations. In this application, we have rather chosen the floating point encoding instead of the binary code. The algorithm seeks many local optima and increases the likelihood of finding the global optimum, representing the problem goal.

4.2. Initialization

We begin the construction of the evolutionary programming by creating a population of μ-trial solutions. The initial population $P(0)$ originates from a start point by assigning a uniform random values to the components m_i within the interval $[-90, +90]$. The adaptable standard deviations σ_i have been assigned an ad hoc choice value equal to 0.01.

4.3. Evaluation

The fitness value of an individual is defined to be its value under the objective function. It is evaluated for each set of rules. It should be well chosen in order to give effectively the desirable quality of the controller. The objective function chosen to be the most successful to our application was found to be

$$C = \frac{100}{\text{SPE} \times \text{SDS}}. \tag{18}$$

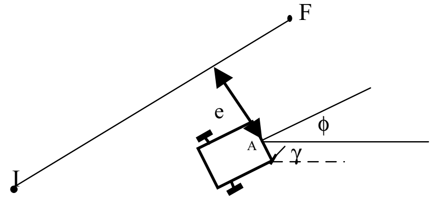

Fig. 6. Vehicle motion control.

This is a combination of two performance criteria. The first criterion becomes obvious if we refer to Fig. 6. The robot is specified in terms of its coordinates (x, y, ϕ). The objective is to make the robot come closer to the line segment in such a way the error e is reduced to zero.

SPE, designates the sum of the position errors, which is the sum of the distance errors between the actual position of the mobile robot and the reference line trajectory, and k the running time, then

$$\text{SPE} = \frac{1}{d(\text{I}, \text{F})} \sum_k |\{(x_A(k) - x_\text{I})(y_\text{F} - y_\text{I}) - (x_\text{F} - x_\text{I})(y_\text{A}(k) - y_\text{I})\}|, \tag{19}$$

such that

$$d(\text{I}, \text{F}) = \sqrt{(x_\text{F} - x_\text{I})^2 + (y_\text{F} - y_\text{I})^2} \tag{20}$$

describing the distance of the segment line, joining initial and final robot positions. The second criterion is added so that the steering angle will be smooth for a low value of the sum of its derivatives. Thus, one can write

$$\text{SDS} = \sum_k |\gamma(k) - \gamma(k-1)|, \tag{21}$$

where SDS is the sum of derivatives of the steering angle and k the sampling time.

4.4. Mutation

Mutation is an important variation operator, it is the primary search operator of evolution strategy. Based on the work of Yao and Liu [29], we have used the new mutation operator which makes use of the Cauchy distribution. When comparing the results obtained by this new operator with our previous work [30], which relied on Gaussian mutation operator, we found that the

Cauchy mutation is faster and robust in sorting out the optimal rules. This is mainly due to its higher probability of making long jumps. Cauchy density function centered at the origin is defined by

$$f_t(x) = \frac{1}{\pi} \frac{t}{t^2 + x^2}. \tag{22}$$

One child is generated from one parent by adding a Cauchy distributed random variable δ_j with the scale parameter $t = 2.5$, and is sampled a new for each value of the counter j:

$$\vec{m}^i_{\text{new}}(j) = \vec{m}^i_{\text{old}}(j) + \vec{\sigma}^i_{\text{old}}(j)\delta_j. \tag{23}$$

The step size σ is controlled by the following equation:

$$\sigma^i_{\text{new}}(j) = \sigma^i_{\text{old}}(j) \exp[\tau'\delta + \tau\delta_j]. \tag{24}$$

The parameters τ and τ' are the learning rates and which adopt the conventional values of $(\sqrt{2n})^{-1}$ and $(\sqrt{2\sqrt{n}})^{-1}$, respectively, where n is the total number of rules to be evolved (in our situation $n = 40$). To help the system operates in a stable region, we have taken into account two constraints, which are derived when the following assumptions are added:

- *Assumption 1*: The action values for which the directions are left sided are taken as positive.
- *Assumption 2*: The action values for which the directions are right sided are taken as negative.

If we refer to Table 1, the following constraints are deduced from the above two assumptions:

$$m_{ij} \in [0, 90], \quad 1 \le i \le 5,\ 1 \le j \le 4,$$
$$m_{ij} \in [-90, 0], \quad 1 \le i \le 5,\ 5 \le j \le 8. \tag{25}$$

After having the new generation, a fitness score is evaluated for each new member $\vec{m}^i$. The best individuals are those for which the objective function yields the greatest value.

Table 1
Fuzzy logic rules of the controller

ρ	θ							
	PVL	PL	PM	PS	NS	NM	NL	NVL
VS	m_{11}	m_{12}	m_{13}	m_{14}	m_{15}	m_{16}	m_{17}	m_{18}
S	m_{21}	m_{22}	m_{23}	m_{24}	m_{25}	m_{26}	m_{27}	m_{28}
M	m_{31}	m_{32}	m_{33}	m_{34}	m_{35}	m_{36}	m_{37}	m_{38}
L	m_{41}	m_{42}	m_{43}	m_{44}	m_{45}	m_{46}	m_{47}	m_{48}
VL	m_{51}	m_{52}	m_{53}	m_{54}	m_{55}	m_{56}	m_{57}	m_{58}

4.5. Selection

The next generation is selected based solely on the fitness values of the individuals. The selection is conducted by taking a random uniform sample of individuals of size equal to half the population size among that of all parents and offspring. Each solution out of the offspring and parent individuals is evaluated against a ‘*Q*’ randomly chosen individuals. For each comparison, a ‘*win*’ is assigned if an individual score is better or equal to that of its opponent.

5. Obstacle avoidance using fuzzy image method

To explain the process, we present the details of how each of the components of the obstacle fuzzy controller operates. The main idea of the fuzzy image method (FIM), comes from the reactive control behavior. The proper action that should be taken to avoid the collision, is to shift the vehicle either to the left or to the right. Therefore, the reactive control system requires a fast and on-line adaptive process that allows the system to deal with the novel situation without having to perform a deep and extensive analysis of the situation.

5.1. Description principle

If we assume that the robot has to move from point *A* to point *F*, as it is shown in Fig. 3, without colliding the objects in its path, then the fuzzy controller developed in the preceding sections has to give the appropriate actions to the wheels to sort out the path of the mobile robot in the free configuration space. The fuzzy controller must therefore receive instantaneously the following four input data of the local environment:

1. The distance from the robot to the target, ρ_t.
2. The angle between the robot and the target, θ_t.
3. The distance from the robot to the obstacle, ρ_o.
4. The angle between the robot and the obstacle, θ_o.

The subscripts t and o stand for target and obstacle. The main basis of the rule based system, consists of taking into consideration these input data and decide the appropriate steering angle that should be resulted. For instance, the following *i*th rule could be used to represent this complex multi-input single-output system:

Rule R^i: IF ρ_t is A_1 AND

θ_t is B_1 AND ρ_o is A_2 AND θ_o is B_2 THEN γ is n^i,

where A_1, B_1, A_2, and B_2 are the linguistic terms associated with the variables ρ_t, θ_t, ρ_o and θ_o, respectively; n^i is the value of the desired steering angle of the *i*th rule. One way to obtain the appropriate output may be encoding the fuzzy logic knowledge base rule as an individual in a genetic algorithm population. The genetic algorithm is then used to retrieve the best knowledge base rule to achieve the desired objective through a maximum fitness criterion. In this case, the fitness criterion should take in account the obstacle constraint. This is possible but this design could lead to an explosive set of rules that could slow down the process. In our case, we have observed and verified that the rule base, founded previously in the last section when the mobile robot evolves in a free obstacle environment, could be used as well to generate the final steering angle when the environment is cluttered with obstacles. For that purpose, we have included an obstacle fuzzy controller that uses the same inference engine developed further, and it is fired only when the vehicle enters the detection area. It must generate then an action which is responsible for moving the robot away from the obstacle such that

$$\gamma_o(k) = F[\rho_o(k), \theta_o(k)] = -n_o. \tag{26}$$

This action is simply the negative value of the steering angle needed to conduct the vehicle toward the obstacle. Now, to keep the robot stuck to a certain resultant path towards the target while it is being pushed away by any static or dynamic object, and to prevent any divergence, the resulting steering angle is made to be composed of two actions as far as the robot remains in the detection area. First, the fuzzy controller computes the appropriate steering angle in order to put the mobile robot in the direction of the target. Next, the fuzzy controller computes the steering angle necessary to put the mobile robot in the direction of the obstacle. If the negative signed is assigned to this action, then the robot will be pushed away from the obstacle. The resulting steering angle that prevents the mobile robot to collide with the obstacle while maintains its motion toward the target is written as the sum of these two actions; one can write

$$\gamma(k) = \gamma_t(k) + \gamma_o(k) \tag{27}$$

or

$$\gamma(k) = m_t - n_o, \tag{28}$$

where k is the running time. As noted previously, the anti collision module is fired only when the two following requirements are satisfied:

$$(1)\ d \le d_s, \qquad (2)\ |\theta| \le \theta_s, \tag{29}$$

where d_s is the influence distance and θ_s the influence zone direction. This will delimit the detection area that will have a canonical shape. Any object in this canonical region emanating from the robot can be detected by the sensors mounted in front of the robot. The straight lines issued from point A, of the mobile robot, in direction of the obstacle intersect its edge shown by the bolded segment in Fig. 7. This part of the obstacle belongs to the visibility region and its presence indicates that the robot will collide if remedial actions are not taken.

A ultrasonic sensor system mounted in front of the robot and made to rotate back and forth by a stepping motor in a suitable chosen angle, could be one way of the realization of the obstacle detection module. Since the obstacle fuzzy controller module is fired only when the vehicle enters the region of detection, then the fired rules are reduced only to that region; this leads consequently to the relation $M < L$; where M and L stands for the maximum number of rules used by the obstacle and the target fuzzy control modules

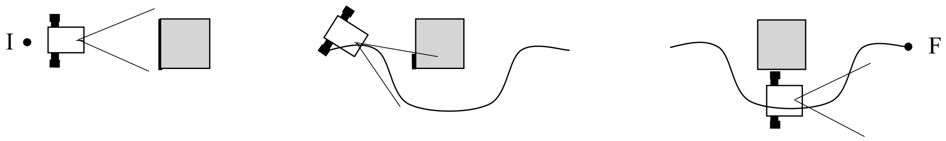

Fig. 7. Reactive navigation control in case of obstacle presence.

```
/* Determine the current state of the vehicle */
  Orientation();
/* Select a motion case*/
   if (θ < −θLIMIT )
     right_back_rot();
     elseif (θ < θLIMIT )
   left_back_rot();
```

Fig. 8. Rotation procedure algorithm.

respectively. When the danger of collision is discarded, i.e. the mobile robot leaves the area of detection, the obstacle controller is disabled and the mobile robot under the influence only of the target controller draws its path towards the target. Therefore, if we make use of the relationship given by (16), we get the expression of the resulting steering angle:

$$\gamma(k) = C_{\mathrm{F}} \left[\frac{\sum_{i=1}^{L} \mu_i m_i}{\sum_{i=1}^{L} \mu_i} - \frac{\sum_{j=1}^{M} \mu_j m_j}{\sum_{j=1}^{M} \mu_j} \right]. \quad (30)$$

Two basic primitive behaviors have been introduced to further increase the flexibility of the autonomous robot. Only one primitive action is activated at a time. The first primitive action is to rotate the mobile robot back to the left, and the second one rotates it back to the right. The algorithm used for this behavior is shown in Figs. 8 and 9 illustrates a sample run of the mobile robot making a rotation back to the left.

Now, to have a picture of the proposed method, suppose we represent the control steering angle by seven linguistic variables having the shape given by Fig. 10; uniformly distributed on the universe of discourse in which the limits are -90° and $+90^{\circ}$.

Let us suppose, for instance, the following eight consecutive rules fired instantaneously at the time instant k:

- R1: IF ρ_{t} is L AND θ_{t} is PS THEN γ_{t} is PS.
- R2: IF ρ_{t} is L AND θ_{t} is PM THEN γ_{t} is PM.

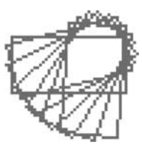

Fig. 9. Simulation result of a left rotation.

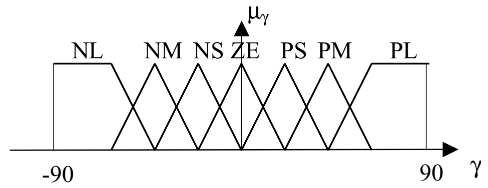

Fig. 10. Membership functions of the control steering angle.

- R3: IF ρ_{t} is M AND θ_{t} is PS THEN γ_{t} is PS.
- R4: IF ρ_{o} is M AND θ_{t} is PM THEN γ_{t} is PM.
- R5: IF ρ_{o} is M AND θ_{o} is PS THEN γ_{o} is PS.
- R6: IF ρ_{o} is M AND θ_{o} is PM THEN γ_{o} is PM.
- R7: IF ρ_{o} is S AND θ_{o} is PS THEN γ_{o} is PS.
- R8: IF ρ_{o} is S AND θ_{o} is PM THEN γ_{o} is PM.

In the next sub-section we will use a graphical interpretation to see how the resulting steering angle is obtained when these rules are applied.

5.2. Graphic interpretation

A graphic interpretation is illustrated in Figs. 11 and 12. The membership functions representing the control steering angle are weighted according to the corresponding input changes and the different control contributions as shown in Fig. 12. For a pair of distance and orientation from the current configuration of the mobile robot to the target, four sets of control steering angles exist. At the same time, for a pair of distance and orientation and for the same configuration of the mobile robot with respect to the obstacle, four sets of control steering angles exist. Therefore, for a unique configuration of the mobile robot in the workspace, eight sets of control steering angles result as shown in Fig. 13. In this figure, we see that we have concatenated the eight sets on the same universe of discourse of the steering angle where we have taken the image, with respect to the vertical axis, of the four sets resulting from obstacle avoidance instead of the real ones. This is true, since it is desirable to have a negative value for the control steering angle as a result of the obstacle presence. Now, to determine the consequent action to be taken from these eight possible contributions, the height defuzzification method is chosen. In this example, the output steering angle is obviously seen to be located somewhere in the negative

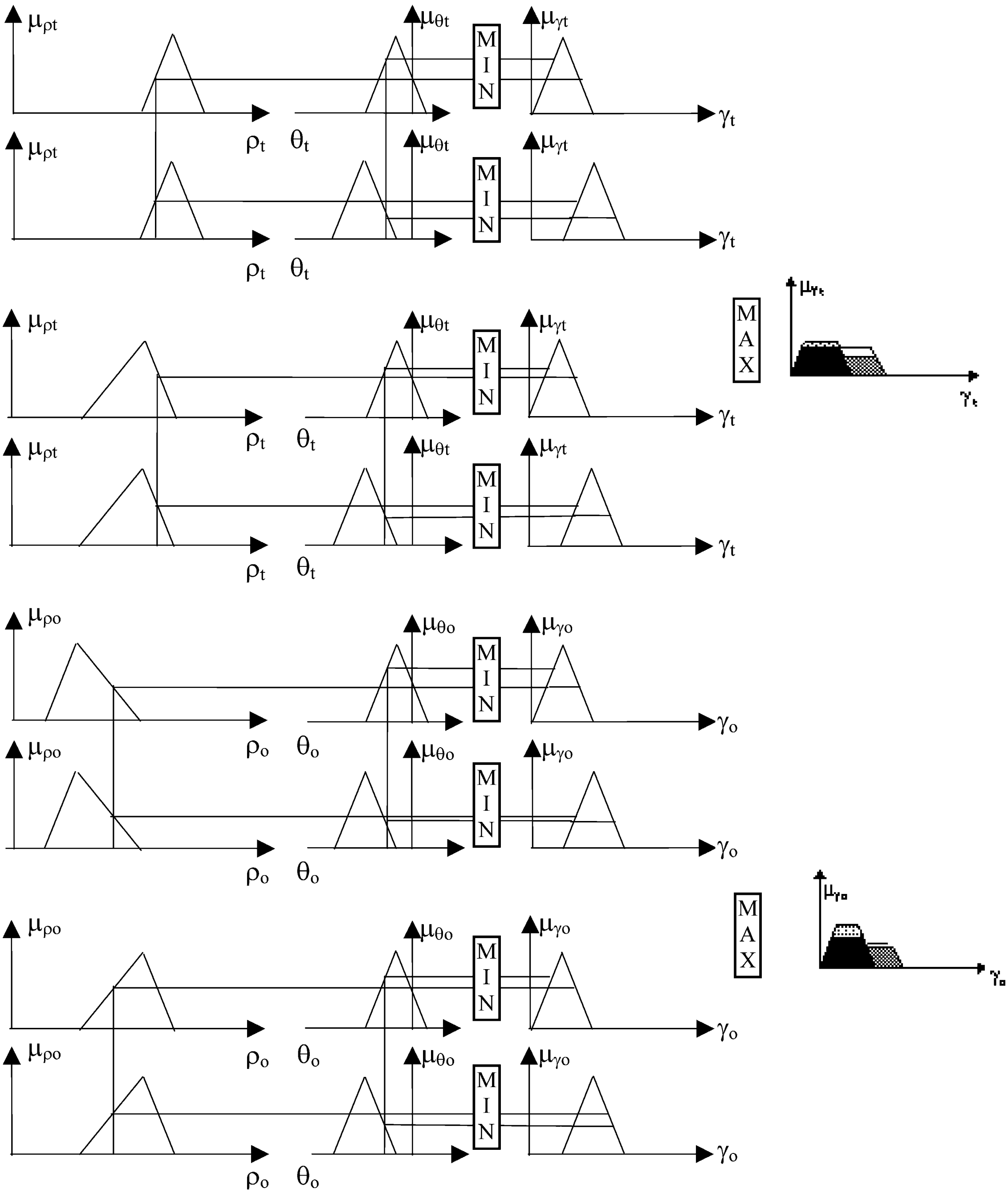

Fig. 11. Graphical representation of the control rules.

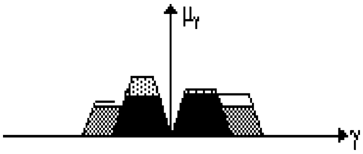

Fig. 12. Determination of the control steering angle from the obtained membership functions.

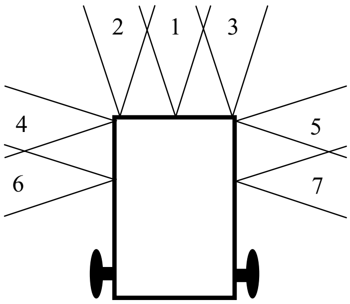

Fig. 13. The robot with its ultrasonic perception system.

axis of the universe of discourse. This means that a deviation to the right is required. This action will be of the same value if the height defuzzification process is split into a part treating the path tracking and a part treating the collision avoidance.

6. Multipoint obstacle avoidance

To generalize the proposed approach to a complete navigation in a cluttered environment, we have developed a perception system that permits a deep analysis of the obstacle situations by adding extra sensors all around the autonomous robot to sense the obstacles located in different and unknown places. This approach is inspired from the works of Brooks [31] and Tongshai et al. [32]. But instead of using behavior-based approach implemented using different levels of behaviors, we have rather used a cooperative and a concurrent behavior-based on fuzzy control techniques. The main interesting thing in this approach is the use of only one rule base for both goal seeking and obstacle avoidance behaviors. Dividing the space into sub-spaces solves the situation for which the robot tries to navigate in a cluttered environment. We have used seven ultrasonic sensors arranged in the way shown by Fig. 13. Therefore the number of obstacle steering angles associated to each sensor increases with the number of sensors. Hence, the resulted steering angle responsible of the orientation of the robot given by Eq. (30) takes now the following general expression:

$$\gamma(k) = \kappa\left(\beta_{\mathrm{t}}\gamma_{\mathrm{t}} - \sum_{i=1}^{\eta}\beta_{\mathrm{o}}^{i}\gamma_{\mathrm{o}}^{i}\right), \qquad (31)$$

κ is an adjusting factor chosen here to be equal to 0.1, whereas η characterizes the number of ultra sonic sensors. β_{t} and β_{o} are indices that weigh the importance of the target and the obstacle steering angles respectively. In this work, the relative importance of an obstacle to the robot is determined by its distance and its orientation. To weigh this importance, one can use a fuzzy index, which is coupled to the perception environment. The rule base defines the relationship between the distance, the orientation and the index of importance. The block diagram of Fig. 14 shows more

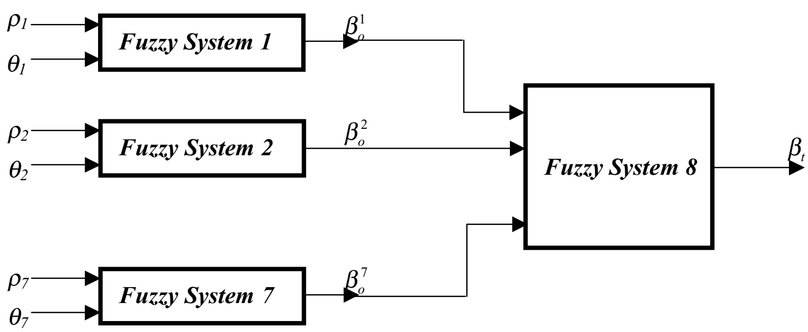

Fig. 14. Block diagram of the fuzzy weighting factor generator system.

Table 2
Fuzzy logic rules of the obstacle index of importance

ρ	θ							
	PVL	PL	PM	PS	NS	NM	NL	NVL
β_o^j								
VS	1	2	3	4	4	3	2	1
S	1	1	2	3	3	2	1	1
M	0	0	1	2	2	1	0	0
L	0	0	0	1	1	0	0	0
VL	0	0	0	0	0	0	0	0

clearly the generation of the fuzzy weighting factors β_t and β_o.

The analysis of the obstacle situation with respect to the robot leads us to choose five discrete values for the index of importance. Table 2 shows the construction of the rule base between adjacent sensors chosen with respect to the danger of collision. For instance, one rule is constructed as

IF (distance is VS) AND (orientation is PL) THEN

(big attention must be taken $I_i = 4$).

The index of importance is a function of 13 linguistic variables $(n_\rho + n_\theta)$. The defuzzification process results in the appropriate weighting factor β_o:

$$\beta_o^j = \frac{\sum_{i=1}^{40} \alpha_{ij} I_{ij}}{\sum_{i=1}^{40} \alpha_i}. \tag{32}$$

Now to obtain the target weighting factor β_t associated to the target steering angle, we fuzzify the obtained obstacle weighting factor into four fuzzy sets having the triangular membership shape as is it is shown in Fig. 15.

Then, we construct the rule base tables between adjacent sensors. The final output, which represent in our case the target index factor γ_t is given by the weighted average over all rules. To measure the danger of collision, we have assigned indices in a descending order according to the importance of the obstruction with respect to the robot configuration. These values are plugged in Tables 3–8. Thus, the target weighting

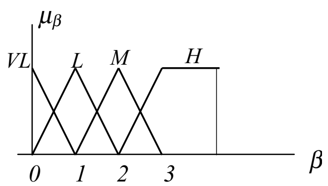

Fig. 15. Membership functions of the index factor β_o.

Table 3
Fuzzy logic rules of the target index of importance

β_2	β_1			
	VL	L	M	H
I_1				
VL	3	3	3	2
L	3	2	2	0
M	3	2	1	0
H	2	0	0	0

Table 4
Fuzzy logic rules of the target index of importance

β_3	β_1			
	VL	L	M	H
I_2				
VL	3	3	3	2
L	3	2	2	0
M	3	2	1	0
H	3	0	0	0

Table 5
Fuzzy logic rules of the target index of importance

β_4	β_2			
	VL	L	M	H
I_3				
VL	3	3	2	2
L	3	2	2	1
M	3	2	0	0
H	3	1	0	0

Table 6
Fuzzy logic rules of the target index of importance

β_5	β_3			
	VL	L	M	H
I_4				
VL	3	3	2	2
L	3	2	2	1
M	3	2	0	0
H	3	1	0	0

Table 7
Fuzzy logic rules of the target index of importance

β_6	β_4			
	VL	L	M	H
I_5				
VL	3	3	2	2
L	3	2	2	1
M	3	2	0	0
H	3	1	0	0

Table 8
Fuzzy logic rules of the target index of importance

β_7	β_5			
	VL	L	M	H
I_6				
VL	3	3	2	2
L	3	2	2	1
M	3	2	0	0
H	3	1	0	0

factor γ_t is a function of 48 fuzzy variables. The resulting singular value is retrieved by a weighted average of the height defuzzification method, which includes all the six data rule tables:

$$\beta_t = \frac{\sum_{j=1}^{6}\sum_{i=1}^{16}\mu_{ij}I_{ij}}{\sum_{j=1}^{6}\sum_{i=1}^{16}\mu_{ij}}. \qquad (33)$$

Table 9
The resulting extracted rules upon applying "EP"

ρ	θ							
	PVL	PL	PM	PS	NS	NM	NL	NVL
VS	55.5	50	49.7	49.8	−43.4	−50	−0.42	−0.17
S	0.0	50	0.0	0.0	−46.2	0.0	0.0	−50.1
M	2.8	0.0	49.7	49.8	0.0	0.0	−0.12	−0.04
L	0.0	25.8	49.7	49.6	−47.4	−50	−0.12	−0.19
VL	0.0	0.0	50	49.5	0.0	−50	−0.4	0.0

7. Simulation results

The proposed scheme was evaluated on the described autonomous robot fuzzy controller. The fuzzy rule base was evolved with the aim to improve the performance criterion (18). The parameter values used in the simulation are, vehicle length 105 cm, vehicle width 100 cm, and the distance from the origin of the wheel axis to the control point located in front of the car is 70 cm. Table 9 shows the optimal values of the consequences of the fuzzy control rules obtained by the evolutionary programming algorithm. This algorithm, was run with a population size equal to 20 for 100 generations. To investigate the ability of the mobile robot to reach the end point located arbitrarily at the origin, simulations are carried out at different starting positions in the workspace. Fig. 16 shows the experimental workspace and the successful control

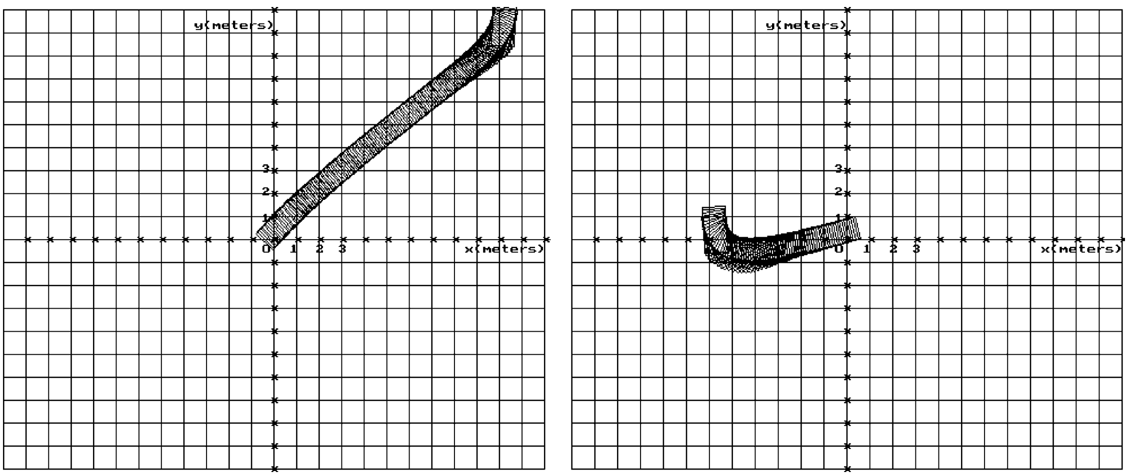

Fig. 16. Navigation control free path from two different configurations.

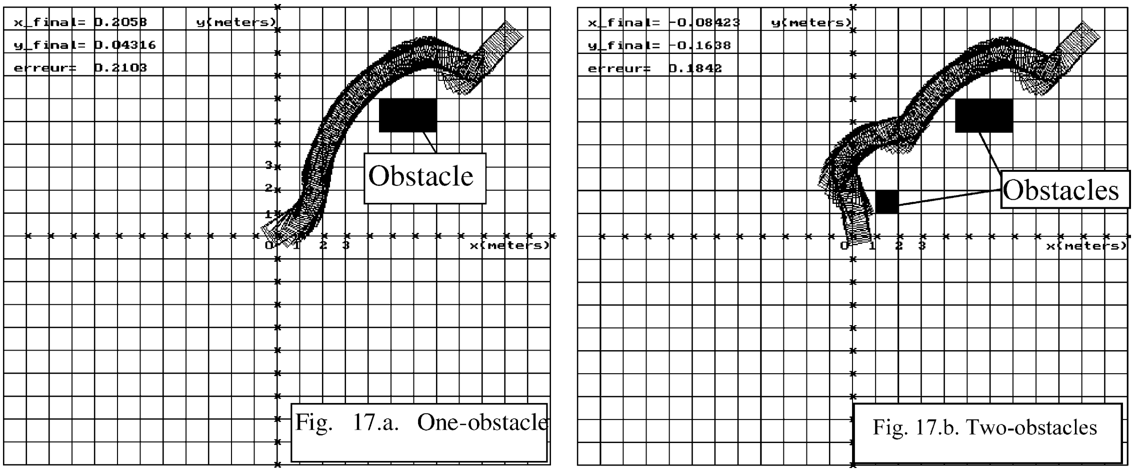

Fig. 17. Example of vehicle controlled motion with obstacle avoidance: (a) one obstacle; (b) two obstacles.

operation. Fig. 17 illustrates the navigation–obstacle avoidance strategy. The simulation results exhibit that the mobile robot can start moving from an initial position, avoids the square-like obstacles, and reaches a goal position. Fig. 17(a) shows the case study of one obstacle, while b shows the case of two obstacles. Based on the evaluation qualitative method of the obstacles configuration from the information of the sensors, the robot succeeds to reach the goal position in an environment cluttered with obstacles. Fig. 18 shows this case study for two different initial robot configurations. Situation for which the robot succeeds in boarding a passageway with obstacles is depicted in Fig. 19, and treatment of local minima is considered in Fig. 20. One can see the robot succeeding in coming out of this situation by choosing a side loop line. To describe what happens we can say that, once the obstacle is detected both behaviors are activated. But, since the fuzzy propositions of the obstacle behavior interest are stronger than those of target behavior interest, the keep away behavior dominates, making the robot abandon the line to the goal. Later when the path is clear, the obstacle behavior is inhibited and navigation to the goal regains importance. Fuzzy

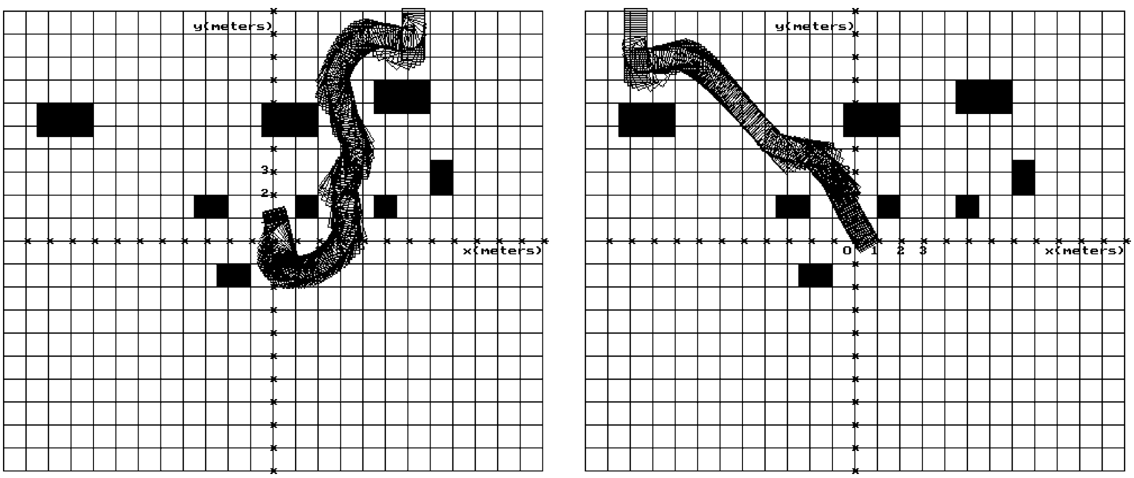

Fig. 18. Example of vehicle controlled motion with a cluttered obstacle environment from two different starting points.

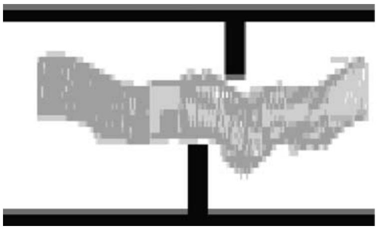

Fig. 19. Evolution of the robot in a passageway with obstacles.

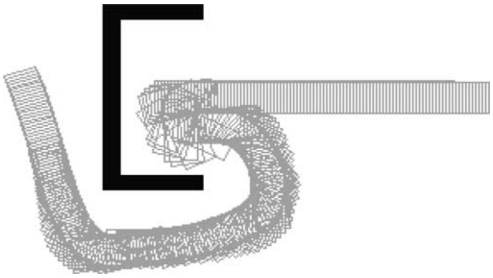

Fig. 20. Treatment of a local minimum.

logic proved to be a satisfactory control strategy and has shown a good degree of robustness face to a large variability and uncertainty in the parameters. Overall, this project this project has served as a platform of a whole work dealing with mobile manipulation, where separation of tasks is considered.

8. Conclusion

In this paper, a theoretical development of a complete navigation procedure of a mobile robot in a cluttered environment has been described. To formalize the imprecise reasoning processes, a method using a fuzzy logic controller has been presented. The problem of extracting the IF–THEN rule base is carried out via an evolutionary programming method. They were tuned to minimize the tracking trajectory error. Simulations were carried out on a non-holonomic mobile robot to test the performances of the proposed fuzzy controller. In an efficient manner the mobile robot was seen to reach any end position, and whatever its initial configuration in the *x–y* plane. Moreover, a new method of obstacle avoidance has been presented in this article as part of the navigational procedure. This allows a reactive control, which results in an efficient behavior towards unforeseeable situations.

References

[1] L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision process, IEEE Transactions on Systems, Man and Cybernetics 3 (1973) 28–44.

[2] E.H. Mamdani, S. Assilian, Application of fuzzy algorithms for control of simple dynamic plant, Proceedings of the Institute of Electrical Engineers 121 (1974) 1585–1588.

[3] M. Sugeno, M. Nishida, Fuzzy control of a model car, in: Fuzzy Sets and Systems, Elsevier, North Holland, 1985, pp. 103–113.

[4] L.X. Wang, Stable adaptive fuzzy control of nonlinear systems, IEEE Transactions on Fuzzy Systems 1 (2) (1993) 146–155.

[5] T.T. Tay, S.W. Tan, Fuzzy systems as parameter estimator of nonlinear dynamic function, IEEE Transactions on Systems, Man and Cybernetics, Part B 27 (2) (1997) 313–326.

[6] S. Abe, M.S. Lan, A method for fuzzy rules extraction directly from numerical data and its application to pattern classification, IEEE Transactions on Fuzzy Systems 3 (1995) 18–28.

[7] C.L. Karr, Design of the cart-pole balancing fuzzy logic controller using a genetic algorithm, in: Proceedings of the SPIE—The International Society for Optical Engineering, 1991, pp. 26–36.

[8] F. Abdessemed, K. Benmahammed, Fuzzy *If–Then* rules selection using genetic algorithms for control, in: Proceedings of the Fourth European Workshop on Fuzzy Decision Analysis and Recognition Technology, EFDAN'99, Dortmund, Germany, 1999, pp. 111–117.

[9] Y. Shi, R. Elberhart, Y. Chen, Implementation of evolutionary fuzzy systems, IEEE Transactions on Fuzzy Systems 7 (2) (1999) 109–119.

[10] T.L. Pérez, RA simple motion planning algorithm for general robot manipulators, in: Proceedings of the Fifth AAAI, Philadelphia, 1986.

[11] O. Khatib, Real time obstacle avoidance for manipulators and mobile robots, International Journal of Robotics and Research 5 (1) (1986).

[12] J. Barraquand, J.C. Latombe, RA Monte-Carlo algorithm for path planning with many degrees of freedom, in: Proceedings of the IEEE Conference on Robotics and Automation, 1990.

[13] W.S. Wijesoma, P.P. Khaw, E.K. Teoh, Sensor modeling and fusion for fuzzy navigation of an AGV, International Journal of Robotics and Automation 16 (1) (2001) 14–25.

[14] A. Safiotti, Fuzzy logic in autonomous robotics: behavior coordination, in: Proceedings of the Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Spain, 1997, pp. 573–578.

[15] F. Abdessemed, K. Benmahammed, E. Monacelli, Fuzzy image method to obstacle avoidance control, in: Proceedings

of the IASTED International Conference on Artificial Intelligence and Soft Computing, Banff, Canada, July 17–19, 2002, pp. 603–608.
[16] Y. Kanayama, A. Nilpour, C.A. Lelm, A locomotion control method for autonomous vehicle, in: Proceedings of the IEEE International Conference on Robotics and Automation, 1988, pp. 1315–1317.
[17] I.J. Cox, Blanche—An experiment guidance and navigation of an autonomous robot vehicle, IEEE Transactions on Robotics and Automation 7 (1991) 193–204.
[18] C. Samsung, K.A. Abderrahim, Mobile robot control. Part I. Feedback control of a nonholonomic wheeled cart in Cartesian space, Technical Report, INRIA, France, 1990.
[19] L.A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353.
[20] B. Beaufrere, Application de la logique floue à la planification de trajectoire des robots mobiles dans un environnement inconnu, Thèse de Doctorat, Poitier University, France, 1994.
[21] J.H. Holland, Outline for a logical theory of adaptive systems, Journal of the Association for Computing Machinery 3 (1962) 277–314.
[22] K.A. DeJong, An analysis of the behavior of a class of genetic adaptive systems, Ph.D. Dissertation, University of Michigan, Ann Arbor, 1975.
[23] D.E. Goldberg, Genetic algorithm in search, in: Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[24] L.J. Fogel, A.J. Owens, M.J. Walsh, Artificial Intelligence through Simulated Evolution, Wiley, New York, 1966.
[25] I. Rechenberg, Evolutionsstrategie: Optimierung technisher systeme nach prinzipien der biologischen evolution, Frommann-Holzboog, Stuttgart, Germany, 1973.
[26] H.P. Schwefel, Evolutionsstrategie und numerische optimierung disertation, Technishe Univeritat, Berlin, Germany, 1975.
[27] M. Herdy, Reproductive isolation as strategy parameter in hierarchically organized evolution strategies, in: Parallel Problem Solving from Nature, vol. 2, Elsevier, Amsterdam, The Netherlands, 1992, pp. 207–217.
[28] T. Bäck, U. Hammel, H.P. Schwefel, Evolutionary computation: Comments on the history and current state, IEEE Transactions on the Evolutionary Computation 1 (1) (1997) 3–12.
[29] X. Yao, Y. Liu, Fast evolution strategies, evolution programming. Part VI, in: Proceedings of the Sixth International Conference, Ep97, Indianapolis, IN, USA, April 13–16, 1997.
[30] F. Abdessemed, E. Monacelli, K. Benmahammed, On using evolutionary programming for a mobile robot fuzzy motion controller, in: Proceedings of the 2000 IEEE International Symposium on Intelligent Control, 2000, pp. 37–42.
[31] R.A. Brooks, A robust layered control system for a mobile robot, IEEE Journal of Robotics and Automation RA-2 (1) (1986) 14–23.
[32] S. Tongshai, S. Suksakulshai, D.M. Wilkes, N. Sarkar, Sonar behavior-based fuzzy control for mobile robot, in: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Nashville, TN, October 8–11, 2000.

Foudil Abdessemed is currently Associate Professor of Electrical Engineering at the university of Batna, and member of the IASTED. He received his Engineering degree in electrical engineering from the National Electrical Engineering Institute of Boumerdes at Algeria in 1985, the Magister degree in electrical engineering from the University of Batna in 1994, and the Ph.D. degree in robotics and control from the University of Batna in 2003. He worked as a researcher for 2 years at LIRIS laboratory at Versailles University, France in 2001, Assistant Professor at the University of Batna in Algeria, and worked as an Engineer for the EN-SIDER factory at Algeria in 1985. His research interests include Robotics, Intelligent Systems and Control, and Multi-agent Systems.

Khier Benmahammed is an IEEE senior member, he has received his M.S. degree in electrical and computer engineering from the University of Colorado at Boulder, his M.A. in mathematics and his Ph.D. in electrical and computer engineering from the Louisiana State University at Baton Rouge, in 1980, 1984 and 1986, respectively. He has held teaching and research positions at Louisiana State University, Baton Rouge, the University of Minnesota, Duluth and Alfred University, New York, and a visiting position at the University of Reims, Champagne-Ardenne, France. He is currently a professor of electrical engineering and the Director of the Intelligent Systems Laboratory at the University Ferhat Abbas of Setif, Algeria. His current research interests are in robust adaptive neural fuzzy systems, signal/image processing, multidimensional systems, and robotics.

Eric Monacelli received his Ph.D. in 1993 in robotics at Pierre et Marie Curie University (Paris). Since 1995, he is Associate Professor at LIRIS laboratory at Versailles University. His field of interest concerned cooperating mobiles robots: control architecture for conflict management like hardware failures or tactical blocages using fuzzy logic and reinforcement laws. He works also on assistive robotics for handicapped persons like smart powered wheelchairs or rollators. He is member of different societies like IFRATH and CIT on the handicapped field.