2012 International Conference on Information Technology and e-Services

Intelligent Systems based on Reinforcement Learning and Fuzzy Logic Approaches, "Application to Mobile Robotic"

Lakhmissi Cherroun
Sciences and Technology Department,
University of Djelfa - 17000 - Algeria
E-mail : cherroun_lakh@yahoo.fr

Mohamed Boumehraz
Electrical Engineering Department,
Biskra University - 07000 - Algeria
E-mail : medboumehraz@netcourrier.com

***Abstract*-- One of the standing challenging aspects in mobile robotics is the ability to navigate autonomously. It is a difficult task, which requiring a complete modeling of the environment and intelligent controllers. This paper presents an intelligent navigation method for an autonomous mobile robot which requires only a scalar signal likes a feedback indicating the quality of the applied action. Instead of programming a robot, we will let it only learn its own strategy. The Q-learning algorithm of reinforcement learning is used for the mobile robot navigation by discretizing states and actions spaces. In order to improve the mobile robot performances, an optimization of fuzzy controllers will be discussed for the robot navigation; based on prior knowledge introduced by a fuzzy inference system so that the initial behavior is acceptable. The effectiveness of this optimization method is verified by simulation.**

***Keywords*— mobile robot, intelligent system, fuzzy controller, Q-learning, fuzzy Q-learning.**

I. INTRODUCTION

Navigation is a vital issue for the movement of autonomous mobile robot. It may be considered as a task of determining a collision-free path that enables the robot to travel through an obstacle course, starting from an initial position and ending to a goal position in a space where there are one or more obstacles, by respecting the constraints kinematics of the robot and without human intervention. The process of finding such path is also known as path planning problem [1]. Obstacle avoidance is one of the basic missions of a mobile robot. It is a significant task that must have all the robots, because it permits the robot to move in an unknown environment without collisions [1][2].

A control strategy with a learning capacity can be carried out by using the reinforcement learning; which the robot receives only a scalar signal likes a feedback. This reinforcement makes it possible the navigator to adjust its strategy in order to improve their performances. It is considered as an automatic modification of the robot behavior in its environment of navigation [3]. The reinforcement learning is a method of optimal control, when the agent starts from an ineffective solution which gradually improves according to the experience gained to solve a sequential decision problem [4].

To use reinforcement learning, several approaches are possible. The first consists in manually discretizing the problem for obtaining states and actions spaces; which could be used directly by algorithms using Q tables [4]. It is however necessary to pay attention to the choice of discretizations, so that they allow a correct learning by providing states and actions which contain a coherent rewards. The second method consists in working at continuous spaces of states and actions by using approximators of functions [5]. Indeed, to use the reinforcement learning, it is necessary to estimate correctly the quality function. This estimate can be done directly by a continuous function approximator like the neural networks or fuzzy inference systems [6][7][8]. The use of these approximators permits to work directly in continuous spaces and to limit the effects of parasites which could appear with bad discretization choices [9][10].

In this paper a reinforcement learning method is used to tune the conclusion part of fuzzy inference systems. These fuzzy controllers are used for various tasks of a mobile robot (goal seeking, wall-following and obstacle avoidance). The results obtained show significant improvements of the robot behaviors and the speed of learning.

The present paper is organized as follows: Section 2 describes the considered robot architecture. Section 3 gives the necessary background of reinforcement learning and we discuss the application of the Q-learning algorithm for a searching goal task. In section 4, we present the proposed optimization fuzzy navigators for different tasks of mobile robot. Section 5 concludes this paper.

II. MOBILE ROBOT KINEMATICS

The robot under consideration is a tricycle mobile robot with non-holonomic property that restricts its mobility in the sideways direction and with limitation of angle. The kinematic model of the mobile robot has two rear driving wheels and a passive front wheel. The inputs of this system are the steering angle α and the velocity v_r. The outputs are the robot coordinates in a novel position: (x_r, y_r, θ_r) (Fig.1). In perfect adhesion conditions (movement without sliding), this kinematic model can be described by the following equations:

$$\begin{aligned} \dot{x}_r &= v_r \cos(\theta_r) \\ \dot{y}_r &= v_r \sin(\theta_r) \\ \dot{\theta}_r &= \frac{v_r}{l} tg(\alpha) \end{aligned} \qquad (1)$$

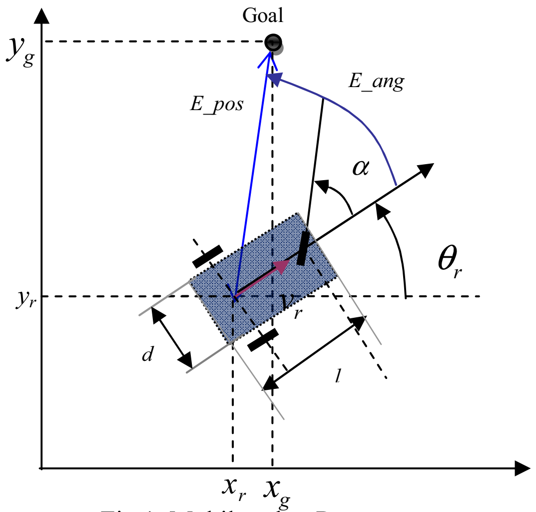


Fig.1. Mobile robot Parameters

III. REINFORCEMENT LEARNING

In reinforcement learning, an agent learns to optimize an interaction with a dynamic environment through trial and error. The agent receives a scalar value or reward with every action it executes. The goal of the agent is to learn a strategy for selecting actions such that the expected sum of discounted rewards is maximized [4].

In the standard reinforcement learning model, an agent is connected to its environment via perception and action. At any given time step t, the agent perceives the state s_t of the environment and selects an action a_t. The environment responds by giving the agent scalar reinforcement signal r_t and changing into state s_{t+1} (see Fig.2). The agent should choose actions that tend to increase the long run sum of values of the reinforcement signal. It can learn to do this overtime by systematic trial and error, guided by a wide variety of algorithms [4][11].

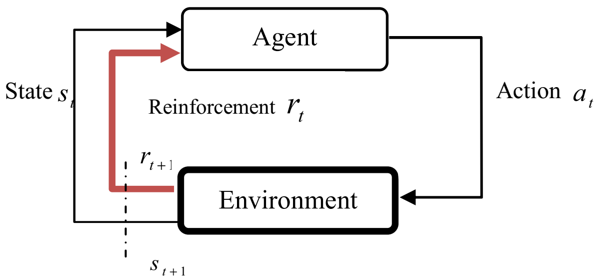


Fig.2. Reinforcement Learning Structure

The agent goal is to find an optimal policy, $\pi : \{S, A\} \rightarrow [0,1]$, which maps states to actions that maximize some long run measure of reinforcement. In the general case of the reinforcement learning problem, the agent's actions determine not only its immediate rewards, but also the next state of the environment. As a result, when taking actions, the agent has to take the future into account. Generally the value function is defined in a problem of the form of a Markovian decision-process *PDM* by:

$$V_\pi(s) = E_\pi(R_t | s_t = s) = E_\pi\left(\sum_{k=1}^{\infty} \gamma^k r_{t+k} | s_t = s\right) \quad (2)$$

Where $\gamma \in]0,1[$ is a factor to regulate the importance of future returns.

The most algorithms of reinforcement learning use a quality function noted Q-function, representing the value of each pair state-action to obtain an optimal behavior [4][12]. It gives for each state, the future return if the agent pursues this policy π :

$$Q^\pi(s,a) = E_\pi(R_t | s_t = s, a_t = a) \quad (3)$$

The optimal quality is:

$$Q^*(s,a) = \max_\pi Q^\pi(s,a) \quad (4)$$

We obtain then:

$$Q^*(s,a) = E(r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a) \quad (5)$$

The learning by temporal differences *(TD)* is a combination of Monte Carlo methods and that of dynamic programming. These methods allow to learn directly without having a model of the environment by evaluating the action without needing to arrive at the final goal [4].

A. Q-learning

It is the more popular of temporal difference algorithms [12]. The idea of Q-learning is to learn a Q-function that maps the current state s_t and action a_t to a utility value $Q^\pi(s_t, a_t)$, that predicts the total future discounted reward that will be received from current action a_t. In that it learns the optimal policy function incrementally as it interacts with the environment after each transition (s_t, a_t, r_t, s_{t+1}). This update is done by observation of the instantaneous transitions and their rewards associated by the following equation [4][12]:

$$Q(s_t,a_t) \leftarrow Q^\pi(s_t,a_t) + \alpha\left[r_t + \gamma \max_{a\in A(s)} Q(s_{t+1},a_t) - Q(s_t,a_t)\right] \quad (6)$$

Where $\alpha \in [0,1]$ is a learning rate that is either a small constant that goes to zero.

The quality functions are stored at table form: a line associates the qualities of the various actions for a given state. Firstly, when the table does not contain sufficient data, a random component is added in order to restrict the eligible actions with the small number of the actions already tested. As the table fills, this random component is reduced in order to allow the exploitation of received information and to obtain a good performance [4].

B. Goal seeking behavior using Q-learning algorithm

At each step the robot must define the state in which it is, and starting from this state, it must make a decision on the action to be carried out. According to the result obtained during the execution of this action, it either is punished, to decrease the probability of execution of the same action in the future, or rewarded, to support this behavior in the similar situations.

For a goal seeking task by a mobile robot, the around space is divided into sectors according to the angle between the orientation of the robot and that of the target noted *E_ang,* and the distances between the robot and the target noted *E_pos* or the position error. The delivered actions are: advanced, turn right and turn left. These actions are chosen by the exploration-exploitation policy *(PEE)* in order to explore the state spaces. During the learning phase, the robot receives the following values as reinforcement signals:

$$r=\begin{cases}4, & \text{If the robot reach the target.}\\ 3, & \text{If } E\_pos \text{ decrease and } E\_ang \;=\; 0.\\ 2, & \text{If } E\_pos \text{ and } E\_ang \text{ decrease.}\\ -1, & \text{If } E\_pos \text{ decrease and } E\_ang \text{ increase.}\\ \text{-}2, & \text{If } E\_pos \text{ increase.}\\ \text{-}3, & \text{If a collision is occured with the environnement.}\end{cases} \tag{7}$$

In order to generalize the robot navigation for all possible situations, the training is made with a random initial position of the robot and target in each episode. Figure 3 shows the robot paths obtained after a learning task. As depicted, the robot moves toward the target from its initial position (the robot can reach the target in all cases) by executing discreet actions. Like a learning indicator, figure 4 shows the average return per trial performance of the controller during the learning process. It is observed that the behavior improves during the time. And it is satisfactory to realize this task.

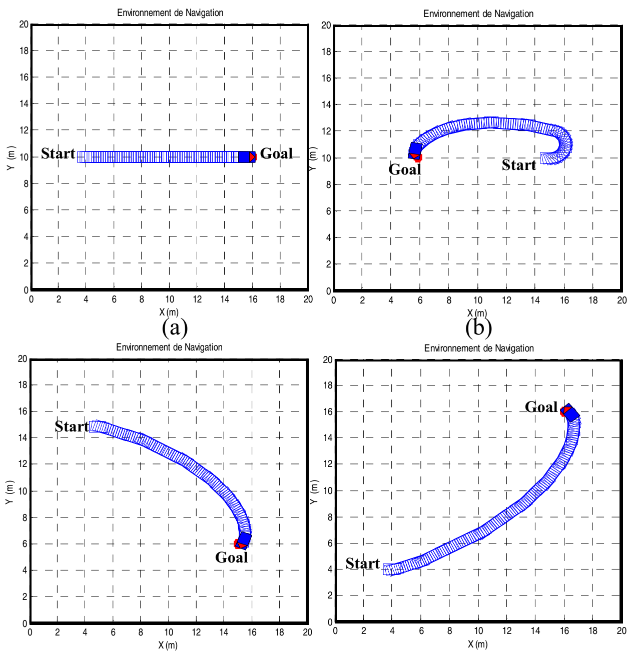


Fig.3. Goal seeking using Q-learning algorithm

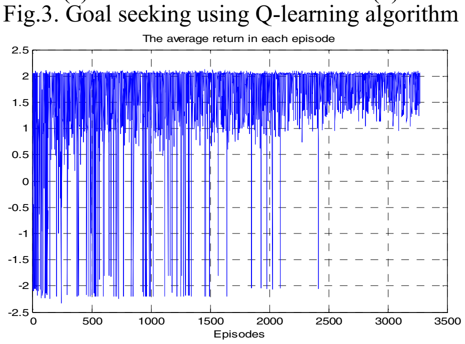


Fig.4. Average values of the reinforcement

Several implementations of the Q-learning algorithm were applied by varying the number of states and actions suggested to obtain an acceptable behavior. The increasing of the state-action pairs makes it possible to improve the behavior of the robot, but requires a more significant memory capacity and time learning (see table.1). The use of the Q-learning algorithm requires the storage of the Q-functions for all pairs (state- action). In the discrete problems with low dimension; we can use tables. But in the case of continuous spaces of states and actions like the mobile robot navigation task; the number of situations is infinite and the representation of the Q-function by tables is difficult. The universal approximators like the neural networks and the fuzzy inference systems offer promising solutions for approximating the Q-values [6][8].

TABLE I.
NUMBER OF EPISODES

Number of States	Proposed actions	Number of episodes
09	03	3260
11	03	4100
13	03	6200
19	05	8950

In order to improve the mobile robot performances, we use in this work, fuzzy controllers optimized by reinforcement learning. These controllers are characterized by the introduction of prior knowledge so that the initial behavior is acceptable.

IV. FUZZY SYSTEMS OPTIMIZATION BY Q-LEARNING

Fuzzy inference systems (FIS) are promising solutions for representing the quality functions with continuous spaces of states and actions [7][11][13]. The task consists in approaching the Q-function by a FIS:

$$s \rightarrow y = \hat{Q} = FIS(s) \tag{8}$$

The idea of this optimization is to propose several conclusions for each rule and to associate each conclusion by a quality function which will be evaluated during the time. The training process permit to obtaining the best rules that maximizing the future reinforcements [7][11][12]. This fuzzy version of the Q-learning algorithm is named fuzzy Q-learning algorithm presented in table 2. The initial rule base using a zero order Takagi-Sugeno model is composed therefore of m rules and N conclusions such as the equation.9 [11][14]:

$$\begin{aligned} \text{if } s \text{ is } S_i \text{ Then } \quad & y = a[i,1] \text{ with } q[i,1] = 0 \\ \text{or } \quad & y = a[i,2] \text{ with } q[i,2] = 0 \\ & \dots \\ \text{or } \quad & y = a[i,N] \text{ with } q[i,N] = 0 \end{aligned} \tag{9}$$

where $q(i,j)$ with $i = 1...m$ and $j = 1...N$, are potential solutions whose values are initialized to 0. During the learning, the conclusion of each rule is selected by means of an exploration-exploitation policy noted *EEP* where $EEP(i) \in \{1...N\}$ such as:

$$a_{\varepsilon-gloutonne} = \begin{cases} \arg\max_{a \in A(s_t)} Q(s_t, a) \text{ with } \varepsilon \text{ probabilty} \\ \text{Random action } A(s_t) \text{ with } 1-\varepsilon \text{ probability} \end{cases} \tag{10}$$

In this case, the inferred action is given by:

$$A(s) = \sum_{i=1}^{m} w_i(s).a[i, EEP(i)] \tag{11}$$

And the quality of this action will be:

$$\hat{Q}(s, A(s)) = \sum_{i=1}^{N} w_i(s).q[i, EEP(i)] \tag{12}$$

TABLE II.
FUZZY Q-LEARNING ALGORITHM

1. Choose the FIS structure.
2. Initialize randomly $q[i,j]$, $i=1,..m$ (m: rule number). $j=1,..N$ (N: Number of proposed conclusions).
3. $t=0$, observe the state s_t
4. For each rule i, compute $w_i(s_t)$
5. For each rule i, choose a conclusion with the EEP.
6. Compute the action $A(s_t)$ and correspondence quality $Q(s_t, A(s_t))$
7. Apply the action $A(s_t)$. Observe the new state s'_t.
8. Receive the reinforcement r_t.
9. For each rule i, compute $w_i(s'_t)$.
10. Compute a new evaluation of the state value.
11. Update parameters $q[i,j]$ using this evaluation.
12. $t \leftarrow t+1$, Go to 5.

A. *Goal seeking behavior using fuzzy Q-learning*

The application is summarized in implementation of a fuzzy controller for mobile robot navigation. Its rule base is improved on-line by using a reinforcement value. In this algorithm, the robot has an initial rule base which defines the possible situations for the designed task.

For a goal seeking task, the controller uses the angle between the orientation of the robot and the target noted *E_ang* and the distance between the position of the robot and that of the target noted *E_pos*. The objective is to generate the steering angle α. The membership functions of *E_pos* and *E_ang* are depicted in Fig.5 and Fig.6 :

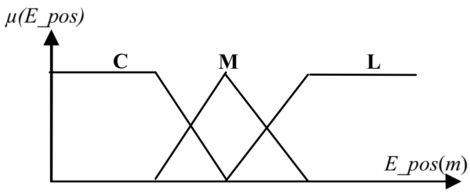


Fig.5. The membership functions of E_pos

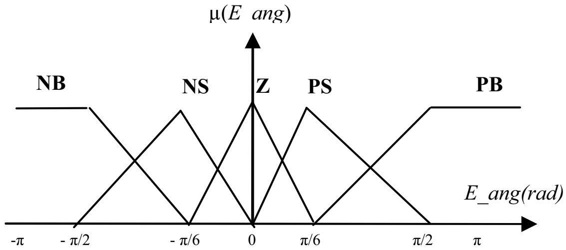


Fig.6. The membership functions of E_ang

With the following linguistic variables: **C:** Close, **M:** Medium, **L:** Large, **Z:** Zero **PS:** Positive Small **PB:** Positive Big **NB:** Negative Big **NS:** Negative Small.

During the learning, the robot receives the same reinforcement values that used in the previous section. In the learning phase, in order to optimize the used navigation controller, the initial positions are selected randomly, where each episode starts with a random position and finishes when the robot reached the target or strikes the limits of its environment. For each rule, 3 conclusions are proposed. After a training time, the robot chooses for each rule the conclusion corresponding to the best Q-function $q[i,j]_{j=1}^{N}$.

For a random position; the paths of the robot using fuzzy Q-learning algorithm are depicted in figure 7. We observe the improvement of the robot behavior. The figure 8 shows the maximization of the average values of the received reinforcements. In all cases, the robot moves toward the target for any initial position by executing continuous actions. The learning is faster than the previous using Q-learning algorithm (see Fig.8).

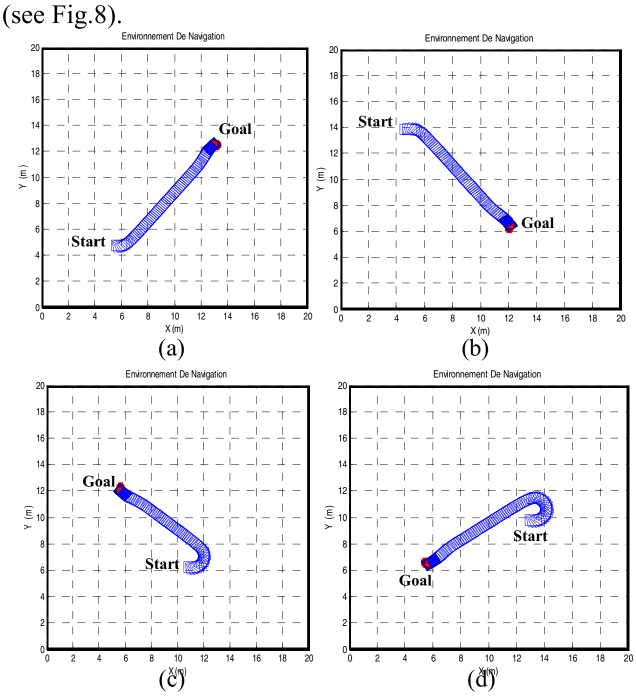


Fig.7. Goal seeking using fuzzy Q-learning algorithm

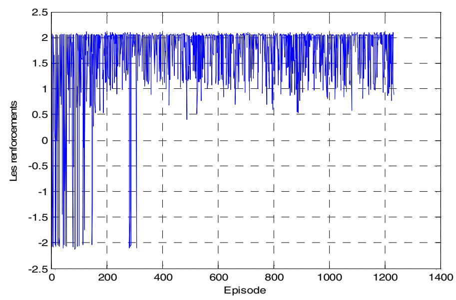


Fig.8. Maximization of the robot rewards

B. *Wall-Following behavior*

For a wall-following task, the fuzzy Q-learning algorithm is used to generate actions that maintain the robot parallel to the wall to be followed. This approach is used with an imprecise knowledge by proposing numerical interpretations for the output linguistics variables (steering angle).

1. *Fuzzy Navigator*

The fuzzy controller uses as variables: the distances between the robot and the obstacle in three directions (opposite d_f, on the right d_r and on the left d_l). The actions are the steering angle and the velocity of the robot. To simplify the studied navigation strategy, the distances from the obstacle in the three directions of the robot are fuzzified with two membership functions (Fig.9), where: **N**: near **F**: far, and the output labels are: **NB:** Negative Big **PS:** Positive small **M:** Medium **S:** Small **Z:** Zero **PB:** Positive Big **NS:** Negative Small.

$$\mu_N(d) = \min\left(\max\left(0, \frac{d-d_2}{d_1-d_2}\right), 1\right) \tag{13}$$

$$\mu_F(d) = \min\left(\max\left(0, \frac{d-d_1}{d_2-d_1}\right), 1\right)$$

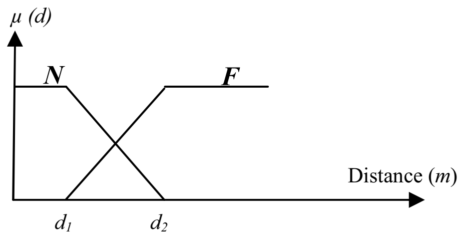


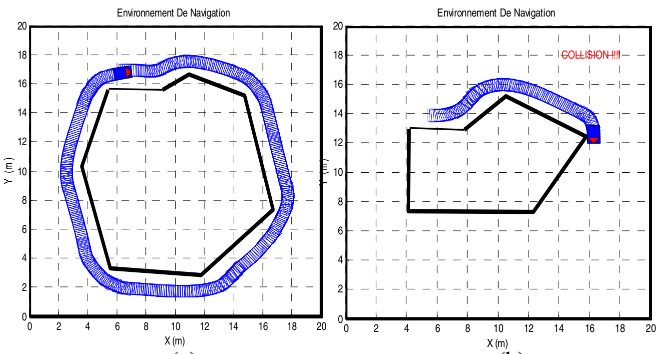
Fig.9. Input membership functions

Firstly, the strategy used for this task is expressed symbolically by the fuzzy rules presented at table 3.

TABLE III.
RULE BASE FOR THE WALL-FOLLOWING

Steering Angle / *Velocity*			*distance d_l* N		*distance d_l* F	
			distance d_f **N**	*distance d_f* **F**	*distance d_f* **N**	*distance d_f* F
distance d_r	**F**	α	*NB*	*NS*	*PB*	*NM*
		V_r	*Z*	*M*	*Z*	*M*
	N	α	*PB*	*Z*	*PB*	*PS*
		V_r	*Z*	*M*	*S*	*S*

The results obtained using this fuzzy controller are given in figures 10 (a-b):



(a) (b)
Fig.10 (a) Wall following using fuzzy controller
(b) Collision with the obstacles

The used fuzzy navigator gives acceptable results to achieve this task as shown in figure 10-a. But in cases when the obstacle contains corners, the behavior is bad and the robot cannot avoid the collisions (Fig.10-b). As a solution for this problem, we propose an on-line optimization of this fuzzy controller rule-base using reinforcement signal; this conducts to use an optimization approach (FQL algorithm).

2. *Fuzzy Q-Learning with an imprecise knowledge*

The used reinforcement values are:

$$r = \begin{cases} -2, & \text{If a collision is occured,} \\ -1, & \text{If } d_i < d_1,\ i = 1 \ldots 3, \\ 0, & \text{Others.} \end{cases} \tag{14}$$

This signal will be employed to select the best numerical interpretation of the used linguistic terms by proposing three interpretations for each output label (steering angle). After a learning process, the optimization results are shown in figures 11 (a-b). It is observed that the robot is able to move (navigate) in its environment without collision with the obstacles.

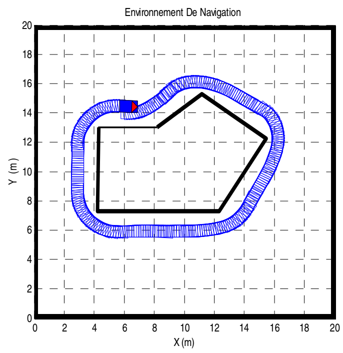
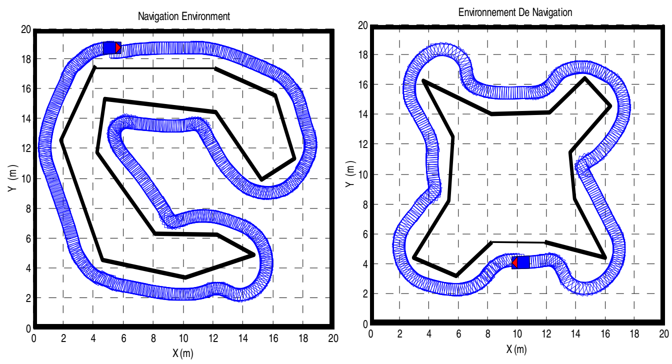


Fig.11. Improvement of the previous behavior
(That using the optimized fuzzy navigator)



(a) (b)
Fig. 12 (a-b) Wall-following with different forms

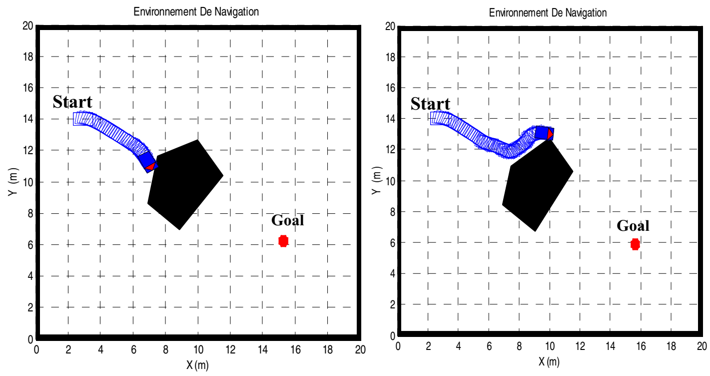
C. *Obstacle avoidance behavior without prior knowledge*

The task consists to equip the robot with the ability of avoiding the near detected obstacles and reaching the goal without being stuck in local minima and without collision with obstacles. For this purpose we use the fuzzy Q-learning algorithm presented at table 1. The same rule-base was used with the following reinforcements:

$$r = \begin{cases} -4, & \text{if a collision occured,} \\ -1, & \text{if } d_i \text{ decrease and } d_i < \frac{l_c}{2}, \\ 0, & \text{otherwise,} \end{cases} \tag{16}$$

In this case, the rewards values are used to define the best conclusion part from the three proposed conclusions for each rule: α_1, α_2 and α_3.

Figures 13 (a-b) show the mobile robot paths in the first episodes (in learning task). As depicted, collisions with the obstacles are produced at the first time. After a training phase, the robot obtains the best behavior to reach the target (by selecting the conclusion that maxisimise the total rewards). The robot can avoid the obstacles and moves in the direction of the target. If there is a near obstacle, it chooses the turn right action (figure 14). Other situations are presented in figures (15 and 16) for a corridor navigation and navigation to the target with a wall-following bahavior.



(a) (b)

Fig.13. Robot paths in learning task, (a) episode 1, (b) episode 20

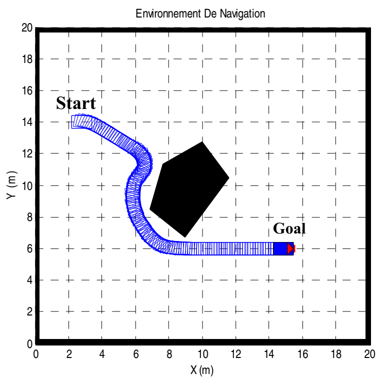


Fig.14. Robot trajectory after the learning process

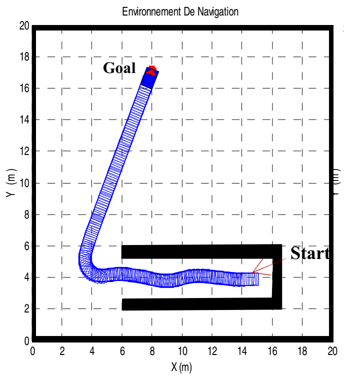


Fig.15. Navigation in a corridor

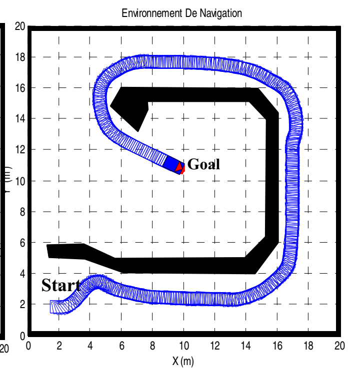


Fig.16. Navigation with a wall-following

V. CONCLUSION

Fuzzy Controller can be effectively tuned via reinforcement learning. In this paper, we presented an intelligent technique for the mobile robot navigation. This technique is based on the optimization of the fuzzy controllers (the conclusion part) in order to maximize the return function. The Q-learning algorithm is a powerful tool to obtain an optimal behavior which requires only one scalar signal likes a feedback indicating the quality of the applied action.

The idea of fuzzy Q-Learning algorithm consists at fuzzy inference systems optimization by using a reinforcement signal. This signal makes the navigator able to adjust his strategy in order to improve its performances. This algorithm combines the advantages of the two techniques and regarded on the one hand as a method of a fuzzy inference systems optimization, and on the other hand as a natural extension of the basic Q-learning algorithm to continuous state and action spaces. The optimization of membership function parameters and number of rules will improve the performance of the proposed method.

REFERENCES

[1] Shuzhi Sam Ge, Frank L. Lewis, Autonomous *Mobile Robots, Sensing, Control, Decision, Making and Applications*, CRC, Taylor and Francis Group, 2006.

[2] Maaref H., and Barret C.," Sensor Based Navigation of an Autonomous Mobile Robot in an Indoor Environment", *Control Engineering Practice*, Vol. 8, pp. 757-768, 2000.

[3] L. M. Zamstein, A. A. Arroyo, E. M. Schwartz, S. Keen, B. C. Sutton, and G. Gandhi, "Koolio: Path Planning using Reinforcement Learning on a Real Robot Platform " *FCRAR, 19th Florida Conference on Recent Advances in Robotics*, Miami, Florida, May 25-26, 2006.

[4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, 1998.

[5] S. T. Hagen and B. Krose, "Q-Learning for Systems with Continuous State and Action Spaces," *in Proc. BENELEARN, 10th Belgian-Dutch Conference in Machine Learning*, 2000.

[6] C. Touzet, *L'Apprentissage par Renforcement*, Paris:Masson, 1999.

[7] M. Boumehraz , K. Benmahammed, M. L Hadjili and V. Wertz, " Fuzzy Inference Systems Optimization by Reinforcement Learning," *Courrier du Savoir* , no. 01, pp. 09-15, 2001.

[8] P. Y. Glorennec, and L. Jouffe, "Fuzzy Q-learning," in *Proc. 1997, FUZZ-IEEE'97, 6th IEEE International Conference on Fuzzy Systems*, pp. 659-662, 1997.

[9] C. Ye, N. H.C. Yung and D. Wang, "A Fuzzy Controller with Supervised Learning Assisted Reinforcement Learning Algorithm for Obstacle Avoidance" *IEEE Trans. Syst., Man, and Cybern.* B, vol. 33, no.1, pp.17-27, 2003.

[10] L. Khriji and al," Mobile Robot Navigation Based on Q-learning Technique", *International journal of advanced Robotic System*, Vol. 8, No. 1, pp 45-51, 2011.

[11] P.Y. Glorennec, " Reinforcement Learning: an Overview," in Proc. 2000, *ESIT'2000, European Symposium on Intelligent Techniques*, pp. 17-35, 2000.

[12] C. Watkins and P. Dayan, "Q-Learning", *Machine Learning*, vol. 8, pp. 279-292, 1992.

[13] K. M. Passino and S. Yurkovich, *Fuzzy Control,* Menlo Park:Addison Wesley, 1998.

[14] M. Sc. Mykhaylo Konyev, "Using fuzzy Inference System as a Function Approximator of a State Action Table," *Advanced Aspects of Theoretical Electrical Engineering,* Sozopol, Bulgaria, 2005.