BEHAVIOUR NAVIGATION LEARNINIG USING FACL ALGORITHM

Abdelkarim Souissi and Hacene Rezine
EMP , Bordj Elbahri, Alger, Algerie
aks752005@yahoo.fr, rezine_hacene_emp@yahoo.fr

Keywords: Mobile Robot Navigation, Reactive Navigation, Fuzzy Control, Reinforcement learning, Fuzzy Actor Critic Learning.

Abstract: In this article, we are interested in the reactive behaviours navigation training of a mobile robot in an unknown environment. The method we will suggest ensures navigation in unknown environments with presence off different obstacles shape and consists in bringing the robot in a goal position, avoiding obstacles and releasing it from the tight corners and deadlock obstacles shape. In this framework, we use the reinforcement learning algorithm called Fuzzy Actor-Critic learning, based on temporal difference prediction method. The application was tested in our experimental PIONEER II platform.

1 INTRODUCTION

In this article, we propose a reinforcement training method where the apprentice explores actively its environment. It applies various actions in order to discover the states causing the emission of rewards and punishments. The agent must find the action which it must carry out when it is in a given situation It must learn how to choose the optimal actions to achieve the fixed goal. The environment can punish or reward the system according to the applied actions. Each time that an agent applies an action, a critic gives him a reward or a penalty to indicate if the resulting state is desirable or not (Sutton, 1998), (Glorennec, 2000). The task of the agent is to learn using these rewards the continuation of actions which gets the greatest cumulative reward.

Mobile robotics is a privileged application field of the training by reinforcement (Fujii, 1998), (Smart, 2002), (Babvey, 2003). This established fact is related to the growing place which takes, since a few years, an autonomous robotics without knowledge of the environment. The goal is then to regard behaviour as a function of mapping sensor-effector. The training in robotics consists of the automatic modification of the behaviour of the robot to improve its behaviour in its environment. The behaviours are synthesized starting from the simple definition of objectives through a reinforcement function.

The considered approach of the robot navigation using fuzzy inference as apprentice is ready to integrate certain errors in the information about the system. For example, with fuzzy logic we can process vague data. The perception of the environment by ultrasounds sensors and the reinforcement training thus prove to be particularly well adapted one to the other (Beom, 1995), (Fukuda 1995), (Jouffe, 1997), (Faria, 2000).

It is very difficult to determinate correct conclusions manually in a large base rule FIS to ensure the releasing from tight corner and deadlock obstacles, even when we use a gradient descent method or a potential-field technique due to the local-minimum problem. In such situations the robot will be blocked.

Behaviours made up of a fusion of a «goal seeking» and of an "obstacle avoidance» issues are presented. The method we will suggest ensures navigation in unknown environments with presence off different obstacles shape, the behaviour will be realised with SIF whose conclusions are determined by reinforcement training methods. The algorithms are written using the Matlab software after having integrated, in a Simulink block, the functions of perception, localization and motricity of the robot.

The application was tested in our experimental platform PIONEER II.

2 FACL ALGORITHM

We have selected a zero-order Takagi-Sugeno FIS apprentice due to its simplicity, universal approximator characteristics, generalization capacity and its real time applications. The input variables have a triangular and trapezoidal membership functions.

The SIF thus consists of *N* rules of the following form (Glorennec, 2000(:

If situation then Y1 = v[i]	and Y2= u[*i*, 1] with q[*i*,1]
	Y2= u[*i*, 2] with q[*i*,2]
	Y2= u[*i*, *j*] with q[*i*,*j*]
Critic	actor

In FACL algorithm (Jouffe, 1997(, each rule R_i of the apprentice has:

- a conclusion v_i used for the approximation of the evaluation function V^{π} of the current policy. "initialized to zeros".
- a set of discrete actions U_i identical for all rules.
- a vector of parameters q^i indicating the quality of the various discrete actions available and intervening in the current policy definition.

The characteristics of the input variables membership functions (number, position) are fixed. The number of rules is also fixed. Thus, the only modifiable characteristics of the apprentice are the conclusions v_i (critic) and the election of an action u[*i*,*j*] among *J* actions available (actor) in the rule R_i.

The FACL algorithm uses two types of training: temporal differences for training the critic reinforcement's predictions, and a competition process between available actions for the actor (Jouffe, 1997).

2.1 Critic

The role of the critic is to approximate the evaluation function which constitutes a better criterion for the choice of the actions than that represented by the primary reinforcements. The critic value in the state S_t is inferred from the conclusions vector v_t

$$V_t(S_t) = \sum_{R_i \in A_t} v_t^i . \alpha_{R_i}(S_t) = v_t . \phi_t^T \quad (1)$$

Where ϕ_t^T represents the apprentice perception at time step *t* (i.e. contains the truth value of activated rules A_t for the state S_t).

The approximation error of the critic is given by the temporal difference error TD as follows:

$$\tilde{\varepsilon}_{t+1} = r_{t+1} + \gamma . V_t(S_{t+1}) - V_t(S_t), \quad (2)$$

The critic uses this error to update the conclusions vector v by a traditional stochastic gradient descent:

$$\begin{aligned} v_{t+1} &= v_t + \beta . \tilde{\varepsilon}_{t+1} . \nabla_v V_t(S_t), \\ &= v_t + \beta . \tilde{\varepsilon}_{t+1} . \phi_t \end{aligned} \quad (3)$$

2.2 Actor

Concerning the actor, the local actions are elected for each rule activated on the basis of quality of these actions, and also according to a policy exploration which we will see further.

The global action for the state S_t is then determined by the inference of these locally elected actions:

$$\begin{aligned} U_t(S_t) &= \sum_{R_i \in A_t} Election_{U_i}(q_t^i) . \alpha_{R_i}(S_t) \\ &= Election(q_t) . \phi_t^T \end{aligned} \quad (4)$$

Where ***Election*** is a function returning the action elected for each activated rule.

The training of the optimal policy thus consists in adjusting the vector of parameters *q* so that the induced policy is improved.

Again the TD error provides a measurement of this quality improvement. Then, we obtains the following rule for training the actor

$$q_{t+1}^i(U_t^i) = q_t^i(U_t^i) + \tilde{\varepsilon}_{t+1} . \alpha_{R_i}(S_t), \forall R_i \in A_t \quad (5)$$

The above expression shows that a positive error TD implies that the action has been just applied is preferable than the t-optimal action. It is necessary to increase the quality of the action being applied.

Reciprocally, if the TD error is negative, it is necessary to decrease the quality of the action being applied because it led the system in a state whose evaluation is lower than that expected.

2.3 Eligibility Traces

The traces implementation of the critic and the actor rise directly from the incremental version of the TD.

Let $\bar{\phi}_t$ be the trace of the critic at step *t*, $\bar{\phi}_t$ is a short term memory of the visited states by the apprentice. This memory is based on the apprentice perception, i.e.: truth values of the rules:

$$\begin{aligned}\bar{\phi}_t &= \sum_{k=0}^{t} (\gamma\lambda)^{t-k}.\phi_k, \\ &= \phi_t + \gamma\lambda \sum_{k=0}^{t-1} (\gamma\lambda)^{t-1-k}.\phi_k, \\ &= \phi_t + \gamma\lambda.\bar{\phi}_{k-1,} \end{aligned} \quad (6)$$

where λ is the proximity factor of the critic.

An equivalent trace for the actor consists in memorizing the actions applied in the states. We use a short term memory of the truth values of each rule according to each action available in these rules. Let $e_t^i(U^i)$ be the trace value of the action U^i in the rule R_i at the step *t* (Jouffe, 1997):

$$e_t^i(U^i) = \begin{cases} \gamma\lambda'.e_{t-1}^i(U^i) + \phi_t^i, (U^i = U_t^i), \\ \gamma\lambda'.e_{t-1}^i(U^i), \quad else \end{cases} \quad (7)$$

where λ' is the proximity factor of the actor.

The updates of the parameters preset by (3) and (5) for all the rules and actions become:

$$v_{t+1} = v_t + \beta\tilde{\varepsilon}_{t+1}\bar{\phi}_t, \quad (8)$$

$$q_{t+1} = q_t + \tilde{\varepsilon}_{t+1}e_t, \quad (9)$$

2.4 Description of the FACL Execution Procedure

The execution in a time step can be divided into six principal stages. Let *t+1* the step of the current time; the apprentice then has applied the action U_t elected in the previous time step and on the other hand has received the primary reinforcement r_{t+1} for the transition from the state S_t to S_{t+1}. After the calculation of the truth values of the rules, the six stages are as follows (Jouffe, 1997):

1- Calculation of the t-optimal evaluation functions of the current state by the critic:

$$V_t(S_{t+1}) = v_t.\phi_{t+1}^T \quad (10)$$

As proposed by Baird (Baird, 1995), to eliminate the instability source due to the approximation by SIF, we do not perform a gradient descent on the residual of Bellman defined by ε but on the *average residual quadratic of Bellman.* The modification to be made on the trace of eligibility is then given by the following relation:

$$\bar{\phi}_t \leftarrow \bar{\phi}_t - \gamma\rho\phi_{t+1}, \ \rho \in [0,1] \quad (11)$$

2- TD Error calculation:

$$\tilde{\varepsilon}_{t+1} = r_{t+1} + \gamma.V_t(S_{t+1}) - V_t(S_t), \quad (12)$$

3- Update of training rates corresponding to the critic parameters for all rules in three stages:

In order to accelerate its training speed and to avoid instability, we adopt a heuristic adaptive training rate (Jouffe, 1997). The implementation of heuristic is carried out by the means of Delta Bar Delta rule:

$$\begin{aligned} &- \quad \delta_t^i = \tilde{\varepsilon}_{t+1}.\bar{\phi}_t^i \\ &- \quad \beta_{t+1}^i = \begin{cases} \beta_t^i + k \quad if \quad \bar{\delta}_{t-1}^i.\delta_t^i > 0, \\ \beta_t^i(1-\psi) \quad if \quad \bar{\delta}_{t-1}^i.\delta_t^i < 0, \\ \beta_t^i \quad else \end{cases} \\ &- \quad \bar{\delta}_t^i = (1-\psi)\delta_t^i + \psi\bar{\delta}_{t-1}^i \end{aligned} \quad (13)$$

Where in our case:

- β_t^i is the training rate of the critic for the rule R_i in the time step t;

- $\delta_t^i = \tilde{\varepsilon}_{t+1}.\bar{\phi}_t^i$ represent the variation brought to the parameter of the critic in which we propose to integrate the trace of eligibility directly and not simply the partial derivative;

- $\bar{\delta}_t^i = (1-\psi)\delta_t^i + \psi\bar{\delta}_{t-1}^i$ represent the exponential average

This rule increases linearly the training rates in order to prevent that it do not become too large, and decrementing it in an exponential way to ensure that it decrease quickly.

4- Training of the critic and the actor by updating the vector v and the matrix q :

$$v_{t+1} = v_t + \tilde{\varepsilon}_{t+1}.\beta_{t+1}.\bar{\phi}_t^T \quad (14)$$

$$q_{t+1} = q_t + \tilde{\varepsilon}_{t+1} e_t \quad (15)$$

5- It is again necessary to calculate the t-optimal evaluation function of the current state by the critic but this time with the lately updated parameters:

$$V_{t+1}(S_{t+1}) = v_{t+1}.\phi_{t+1}^T \quad (16)$$

This value will be used for the TD error calculation in the next step of time.

6- Now it remains the choice of the action to be applied in the state S_{t+1} .

We consider the case of continuous type actions, which is inferred from the various actions elected in each rule,

$$U_{t+1}(S_{t+1}) = \sum_{R_i \in A_{t+1}} Election_U(q_{t+1}^i).\alpha_{R_i}(S_{t+1}), \forall U \in U,$$

Where Election is defined by:

$$Election_U(q_{t+1}^i) = ArgMax_{U \in U}(q_{t+1}^i(U) + \eta^i(U) + \rho^i(U)) \quad (17)$$

The traces eligibility updates for the critic and the actor are given by the two following formulas:

$$\bar{\phi}_{t+1} = \phi_t + \gamma\lambda.\bar{\phi}_t, \quad (18)$$

$$e_{t+1}^i(U^i) = \begin{cases} \gamma\lambda'.e_t^i(U^i) + \phi_{t+1}^i, (U^i = U_{t+1}^i), \\ \gamma\lambda'.e_t^i(U^i), \quad else \end{cases} \quad (19)$$

2.5 Proposed Exploration / Exploitation

The actions election strategy which we used is a combination of directed and random exploration (Jouffe, 1997). The global election function, applied to each rules, is then defined by:

$$Election_U(q) = ArgMax_{U \in U}(q(U) + \eta(U) + \rho(U)) \quad (20)$$

Where U represents the set of available discrete actions in each rules, and q the associated vector quality. $\eta(U)$ the random exploration term, and $\rho(U)$ the directed exploration term.

The term η is in fact a random values vector. It corresponds to a vector ψ of values sampled according to an exponential law, standardized in order to take into account the q qualities size scale:

$$s_f(U) = \begin{cases} 1 \quad si \quad Max(q(U)) = Min(q(U)) \\ \dfrac{s_p(Max(q(U)) - Min(q(U)))}{Max(\psi)}, \quad else \end{cases} \quad (21)$$

$$\eta(U) = s_f(U).\psi, \quad (22)$$

Where s_p represents the maximum size of the noise relative to the qualities amplitude, s_f is the corresponding normalisation factor. Used alone, this term of exploration allows a random choice of actions when all qualities are identical and allows in the other case, to test only the actions of which quality satisfied:

$$q(U) \geq q(U^*) - s_p(Max(q(U)) - Min(q(U))) \quad (23)$$

This term then allow us to avoid the choice of bad actions (Jouffe, 1997).

The term of directed exploration ρ permit the test of actions not having been applied often. It is thus necessary to memorize the times number where the action was elected. This term is defined in the following way: $\rho(U) = \dfrac{\theta}{e^{n_t(U)}}$ (24)

Where θ represents a positive factor which permit to equilibrate the directed exploration, $n_t(U)$ the applications number of action U at the time step *t*. In the case of actions of the continuous type, $n_t(U)$ corresponds to a number of applications of the discrete action U in the considered rule. Let U_t^i the discrete action elected at the step of time t in the rule R_i ; the update of this variable is determined by the following equation:

$$n_t(U^i) = n_{t-1}(U^i) + 1, \quad \forall R_i \in A_t, U^i = U_t^i \quad (25)$$

3 EXPERIMENTAL PLATFORM

Pioneer P2-dx with its modest size lends itself well to navigation in the tight corners and encumbered

spaces such as the laboratories and small offices. It has two driving wheels and a castor wheel.

Figure 1: Pioneer II Robot Photo.

To detect obstacles, the robot is equipped with a set of ultrasounds sensors. It supports eight ultrasonic sensors placed in its front (fig. 2). The range of measurements of these sensors lies between 10cm as minimal range and 5m as maximum range.

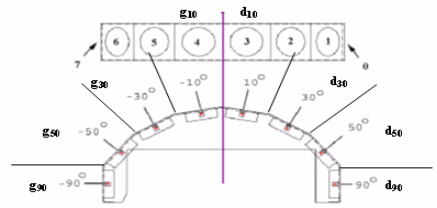

Figure 2: Position of the ultrasonic sensors used in the robot Pioneer II.

The software Saphira/Aria (Konolige, 2002a), (Konolige, 2002b) allows the control of the robot (C/C++ programming). We have integrated functions of Saphira (perception, localization and motricity) in Simulink using API (S-Function), Thus we can benefit from MatLab computing power and simplicity of Simulink to test our algorithms and to control the robot Pioneer II.

4 EXPERIMENTATION

The task of the robot consists in starting from a starting point achieving a fixed goal while avoiding the static obstacles of convex or concave type. It is realised by the fusion of two elementary behaviours «*goal seeking*» and «*obstacle avoidance* ".

4.1 The «goal seeking» Behaviour

For the "goal seeking" behaviour, we consider three membership functions for the input θ_{Rb} (robot-goal angle), and two membership functions for the input ρ_b (robot-goal distance) (Figure 3). The base of knowledge consists of six fuzzy rules. The SIF controller has two output for the actor (rotation speed Vrot and the translation speed Vit) and one output for the critique who is related to the evaluation function of the pair actions (Vrot,Vit).

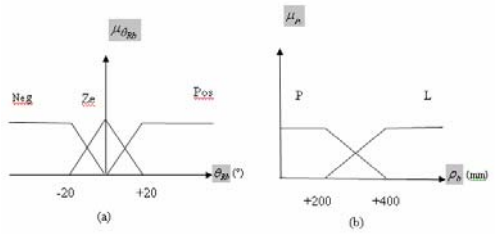

Figure 3: membership functions for θ_{Rb} (a) and ρ_b (b).

From the heuristic nature of FACL algorithm, we carried out several tests to determine the values of the parameters which accelerate the training speed and to obtain good performances for the apprentice. After a series of experiments, we found the following values:

$\theta = 50, \lambda = \lambda' = 0.9, \rho = 0.5$, $sp = 0.1$ and $\gamma = 0.9$

Available actions for all rules, are as follow {-20°/s, -10°/s, -5°/s, 0°/s, +5°/s, +10°/s, +20°/s} for the rotation velocity and {0 mm/s, 150 mm/s, 350mm/s} for translation, which gives 21 possible actions in each fuzzy rule.

The reinforcement function is defined as follows:

- If the robot is far from the goal, the reinforcement is equal to

- 1 if ($\theta_{Rb} . \dot{\theta}_{Rb} < 0$) & Vit=0
- 1 if (-1°< θ_{Rb} <+1°) & Vit $\neq 0$
- 0 if ($\theta_{Rb} . \dot{\theta}_{Rb} = 0$) & Vit=0
- -1 else

- If the robot is close to the goal, the reinforcement is equal to

- 1 if ($\theta_{Rb} . \dot{\theta}_{Rb} < 0$) & Vit=0
- -1 else

Figure (4) shows the trajectory of the robot during the phase of training and validation.

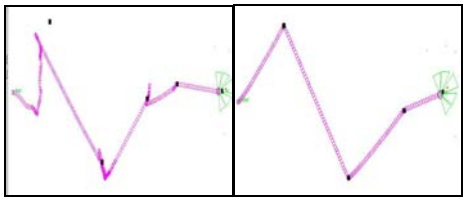

Figure 4: Trajectory of the robot during and after training.

4.2 The «obstacle avoidance» Behaviour

For this behaviour, we have determined the translation speed of the robot proportional to the distance from the frontal obstacles, with a maximum value of 350 mm/s (fig.5).

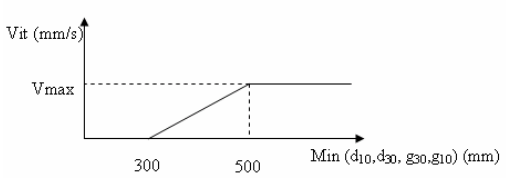

Figure 5: Translation speed Vit.

Inputs of the fuzzy controller for this behaviour are the minimal distances provided by the four sets of sonar {min(d_{90},d_{50}), min(d_{30},d_{10}), min(g_{10},g_{30}), min(g_{90},g_{50})}, with respectively three membership functions for each one (fig. 6).

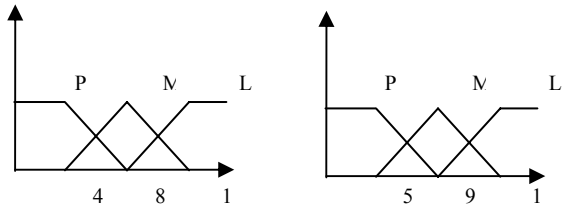

Figure 6: input membership functions for the fuzzy controller Vrot (a) :{min(d_{90},d_{50}), min(g_{90},g_{50})} (b) :{min(d_{10},d_{30}), min(g_{10},g_{30})}.

The set of the action $\boldsymbol{U}$ common to all rules consists of five actions {-20°/s, -10°/s, 0°/s, +10°/s, +20°/s}.

The reinforcement function is defined as follows:

- +1 if min {min(d_{90},d_{50}),min(d_{30},d_{10})} < min {min(g_{10},g_{30}),min(g_{90},g_{50})} & Vrot > 0
- +1 if min {min(d_{90},d_{50}),min(d_{30},d_{10})}> min {min(g_{10},g_{30}),min(g_{90},g_{50})} & Vrot < 0
- -1 otherwise

The figure (7) shows the evolution of the robot during the training phase.

On the figure (8), we represent the time evolution of the robot. It shows that the robot is able to be released from the tight corners and deadlock obstacles shape.

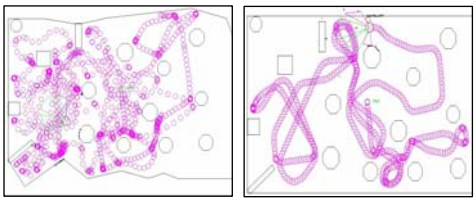

Figure 7: Trajectories of the robot during and after training.

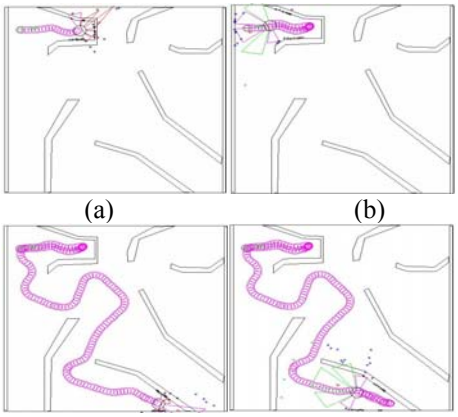

Figure 8: Time evolution of the robot after training.

4.3 Fusion of the Two Behaviours

The goal of the two elementary behaviours fusion is to allow the robot navigation in environments composed by fixed obstacles of convex or concave type and to achieve a fixed goal while ensuring its safety, witch is a fundamental point in reactive navigation.

The suggested solution consists in considering the whole input variables of the two behaviours « *goal seeking* » & « *obstacles avoidance* » associated with distributed reinforcement function by the means of a weighting coefficient between the two behaviours (0.7 for obstacles avoidance and 0.3 for the goal seeking).

The fuzzy controller input are six who are the minimal distances provided by the four sets of sonar {min(d_{90},d_{50}), min(d_{30},d_{10}), min(g_{10},g_{30}), min(g_{90},g_{50})}, with respectively three membership functions for the side sets and two membership functions for the frontal sets, to which we add the input θ_{Rb} with three membership functions, and ρ_b with two membership functions. The rules base

thus consists of 216 fuzzy rules. The translation speed of the robot is proportional to the distance from the frontal obstacles. The FACL algorithm parameters are slightly modified as follows: θ=20 & sp =0.9

Figure (9) represents the type of trajectory obtained during the training phase. The robot manages to avoid the obstacles and to achieve the goal assigned in environments encumbered by maintaining a reasonable distance between the obstacle and its with side dimensions. Figure (10) illustrates the satisfying behaviour of the robot after training. The various trajectories obtained for the same environment and the same arrival and starting points are due primarily to the problems of the perception of the environment by the robot and are related to the phenomena of sonar readings, these differences confirm at the same time the effectiveness of the training algorithm.

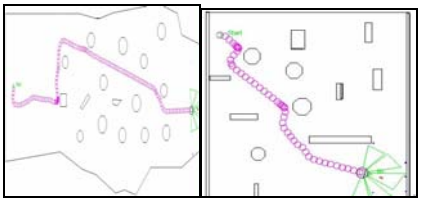

Figure 9: Trajectories of the robot in training phase.

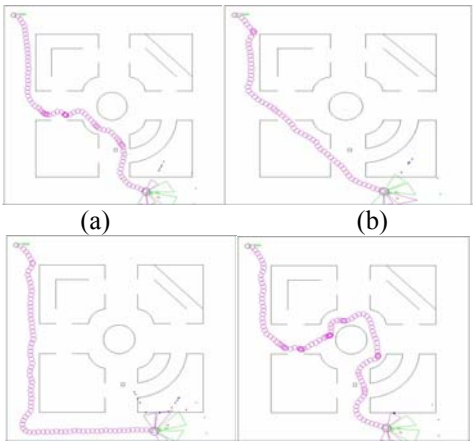

Figure 10: Various trajectories of the robot after training.

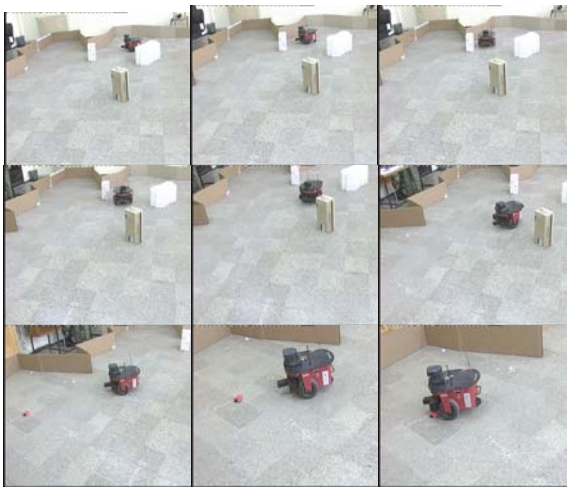

Figure 11: Trajectory of the real robot after training.

Figure (11) illustrates the satisfying behaviour of the real robot which evolves/moves in an unknown environment and which manages to achieve the fixed goal.

5 CONCLUSION

FACL Algorithm makes it possible to introduce generalization into the space of the states and the actions, and a Sugeno type order zero SIF conclusions adaptation incrementally, and this only by the means of the interactions between the apprentice and his environment. The reinforcement function constitutes the measure of the performance of the required behaviour solution. Also, fusion of the behaviours « goal seeking» & « obstacle avoidance » is presented by using a combined reinforcement function. The simulation and experimentation results in various environments are satisfactory.

REFERENCES

Babvey 03 S. Babvey, O. Momtahan, M. R. Meybodi, " Multi Mobile Robot Using Distributed Value function Reinforcement Learning", IEEE, International Conference on Robotics &Automation, September 2003.

Baird 95 L.C. Baird, "A Residual Algorithms: Reinforcement Learning with Function Approximation", Proceedings of the Twelfth International Conference on Machine Learning, 1995.

Beom 95 H.R. Beom, H. S. Cho, " A Sensor-Based Navigation for a Mobile Robot Using Fuzzy Logic

and Reinforcement Learning ", IEEE, Transactions on Systems Man and Cybernetics , vol.25, NO.3 ,pp.464-477, March 1995.

Faria 00 G. Faria, R. A.F Romero "Incorporating Fuzzy Logic to Reinforcement Learning", IEEE, pp.847-852 ,Brazil,2000.

Fujii 98 T. Fujii, Y. Arai , H. Asama, I. Endo, "Multilayered Rienforcement Learning For Complicated collision Avoidance problems", Proceedings of IEEE, International Conference on Robotics &Automation, pp.2186-2191,May 1998.

Fukuda 95 T. Fukuda, Y. Hasegawa, K. Shimojima, F. Saito,"Reinforcement Learning Method For Generating Fuzzy Controller", Department of Micro system Engineering, Nagoya University, pp.273-278, Japan, IEEE, 1995.

Glorennec 00 P. Y. Glorennec, "Reinforcement Learning: An Overview", INSA de Rennes ESIT'2000, Aachen, pp.17-35, Germany, September 2000.

Jouffe 97 L. Jouffe, "Apprentissage de Système D'inférence Floue par des Méthodes de Renforcement", Thèse de Doctorat, IRISA, Université de Rennes I, Juin 1997.

Konolige 02a K. G. Konolige 'Saphira Référence', Edition Doxygen, Novembre 2002.

Konolige 02b K. G. Konolige 'Saphira Robot Control Architecture', Edition SRI International, Avril, 2002

Smart 02 W. D. Smart, L. P. Kaelbling , " Effective Reinforcement Learning For Mobile Robots", Proceedings of the IEEE, International Conference on Robotics &Automation, pp.3404-3410, May 2002.

Sutton 98 R.S. Sutton, A. Barto, "Reinforcement Learning", Bradford Book, 1998.