A Fuzzy Q-Learning Approach to Navigation of an Autonomous Robot

Sepideh Valiollahi
Department of Electrical and Computer Engineering
Babol University of Technology
Babol, Iran
s.valiollahi@stu.nit.ac.ir

Reza Ghaderi
Department of Electrical and Computer Engineering
Babol University of Technology
Babol, Iran
r_ghaderi@nit.ac.ir

Ataollah Ebrahimzadeh
Department of Electrical and Computer Engineering
Babol University of Technology
Babol, Iran
e_zadeh@nit.ac.ir

***Abstract*—The proposed algorithm takes advantage of coupling fuzzy logic and Q-learning to fulfill requirements of autonomous navigations. Fuzzy if-then rules provide a reliable decision making framework to handle uncertainties, and also allow incorporation of heuristic knowledge. Dynamic structure of Q-learning makes it a promising tool to adjust fuzzy inference parameters when little or no prior knowledge is available about the world. To robot, the world is modeled into a set of state-action pairs. For each fuzzified state, there are some suggested actions. States are related to their corresponding actions via fuzzy if-then rules based on human reasoning. The robot selects the most encouraged action for each state through online experiences. Efficiency of the proposed method is validated through experiments on a simulated Khepera robot.**

Keywords-fuzzy Q-learning; autonomous navigation; Khepera robot

I. INTRODUCTION

Autonomy is a necessity of the robots that are increasingly replacing/cooperating human in their homes and workspaces, or hazardous and unreachable environments. An autonomous robot should move purposefully and carry out specific tasks in unknown environments usually with unpredictable dynamics. An instance could be reaching a desired goal without colliding with obstacles. A robot system comprises some main interacting units: sensors, preprocessing unit, decision making unit, controllers, motors, and actuators [1, 2]. Although, all these units affect the robot performance, decision making unit as the robot brain plays the most significant role. As the focus of this study is on decision making unit, a small wheeled robot with simple infrared sensors is selected so that the other units do not demand heavy processing [3]. A robust decision making algorithm is designed to navigate the robot toward a predefined goal while avoiding obstacles.

A set of heuristic fuzzy if-then rules is designed by human reasoning and without any restrictive presumptions about the world. Incorporation of heuristic knowledge to design of fuzzy rules results in few number of rules, which in turn leads to a fast and easy implementation of planning process. The proposed method is independent of a global model or prior knowledge about the world. The only source of information comes from the locally sensed data by limited range infrared sensors. Fuzzy inputs are outputs of infrared sensors while fuzzy outputs are the robot speed and steering angle. Fuzzification of measured sensory data helps an efficient handling of uncertain, imprecise, or noisy information [2]. The robot models the world into few fuzzified sensory states. For each fuzzified state, there are some suggested actions. States are related to their corresponding actions by fuzzy if-then rules based on human reasoning. Online tuning of the fuzzy inference provides a flexible decision making system, which could adapt itself to unknown environments. Q-learning is a promising tool to tune fuzzy inference parameters due to its simple, model free, and dynamic structure [4]. Applying Q-learning, fuzzy output memberships are tuned online and through interactions with the world. Simulated experiments on Khepera robot validated efficiency of the proposed method. The rest of the paper is reviewed as follow: Section II deals with related literature, Section III provides a brief description about the Khepera robot. The proposed fuzzy Q-learning method is explained in section IV. Simulation results are depicted in section V, and finally the paper is concluded in section VI.

II. RELATED LITERATURE

The earliest methods for autonomous robot navigation were based on hierarchical architecture usually composed of four main layers: Sensing, Modeling, Planning, and Acting [1, 5, 6]. The robot builds a global model of the world using prior knowledge or sensory information, tries to plan an optimal path, and passes it to the execution layer. This routine continues until reaching the ultimate goal. After sensing stage and during the rest of the process, there is no feedback from the environment. Obviously, such methods do not work when prior knowledge about the world is unavailable, incomplete, or unreliable. Furthermore, unpredictable dynamic of real world environments plus their inherent uncertainties put hierarchal methods in pitfall of out of date models which lead to inadequate actions [2].

978-1-4673-1479-4/12/$31.00 ©2012 IEEE

To overcome this shortcoming, reactive methods have been proposed where the world model is based on currently sensed information and updated with respect to the robot interactions with the world [7-10]. Planning is adaptive to real time events while navigating the robot to a desired goal. Fuzzy logic has proven to be an efficient tool to decision making in autonomous navigation problems owing to its efficient properties such as independency of a precise model of the world, tolerance against uncertain or noisy information, fast response, and easy implementation [11-13]. In [11], fuzzy logic is applied to individual behavior design and action coordination of the behaviors. A layered approach is employed in which a supervision layer based on the context decides which behaviors to activate and then coordinate. The applied fuzzy membership parameters in [11] are set offline, usually done by human reasoning or a trial and error process, that is not adequate in case of unexpected situations. Therefore, learning strategies are applied to adjust fuzzy inference parameters. Supervised learning algorithms usually require large amounts of training input/output data, which may be hard to obtain specially for autonomous navigations [14-16]. Unsupervised and dynamic structure of reinforcement learning makes it a promising tool to online applications [2, 17]. Q-learning is an efficient simple-structured model free reinforcement learning that is often used to adjust fuzzy parameters; this coupling is also known as fuzzy Q-learning (FQL) [18-20]. In [20] an FQL approach is applied where a fuzzy inference of eight rules is designed offline and then Q-learning is applied to tune fuzzy outputs memberships. In this paper, accuracy of the FQL approach used in [20] is improved by making changes to fuzzy rule base and an efficient definition of reinforcement signal, which has a critical role in Q-learning performance. Experimental results demonstrate an acceptable tradeoff between simplicity and accuracy of the proposed method

III. KHEPERA ROBOT

Khepera is a small mobile robot with a circular shape of 55 mm in diameter, 30 mm in height and 70 g in weight [3]. Using such a small robot, the navigation algorithm can be developed independent of a precise model of the robot. This in turn, allows an easy transfer of the designed navigation algorithm to other robots [21]. Khepera has two wheels and two small Teflon balls. Each wheel is moved by a DC motor. The robot's maximum linear and angular speeds are about 40 mm/s and 1.58 rd/s respectively [12]. Khepera has eight infrared sensors, which are composed of an emitter and an independent receiver. These sensors (S0, S1,…, S7) are arranged in a somewhat circular fashion around its body (Fig. 1) and measure distances in a short range from about 1 to 5 cm.

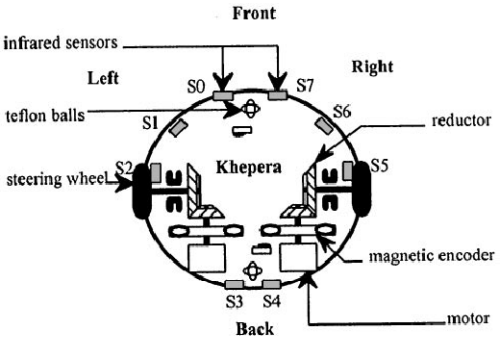

Figure 1: The Khepera Robot [12]

The sensor readings are integer values in the range of [0,1023]. A sensor value of 1023 indicates that the robot is very close to the object, and a sensor value of 0 indicates that the robot does not receive any reflection of the infrared signal [22]. Four groups of sensors are considered i.e. left, front, right, and back sensors. The considered value for each group is the maximum output of each of the merged sensors.
Left sensor: SL=max(s1,s2)
Front sensor: SF=max(s0,s7)
Right sensor: SR=max(s6,s5)

IV. THE PROPOSED FUZZY Q-LEARNING METHOD

To get a better insight to the proposed method, brief descriptions of fuzzy logic and Q-learning is provided below:

A. Fuzzy

Fuzzy logic maps an input space to an output space by a list of if-then statements called rules. The rules are structured in a human decision making format. A typical rule could be as follow:

IF Obstacle is Far THEN Speed is Low

where "*Obstacle*" and "*Speed*" are input variable and output variable respectively. Rule base is the key element in robot intelligence. A rule base should consider all possible cases of all linguistic adjectives for input variables. Linguistic adjectives, like "*Far*" and "*Low*", are described by membership functions which map the value of the variables to membership degrees ([0,1]) in each fuzzy set (Fig. 2).

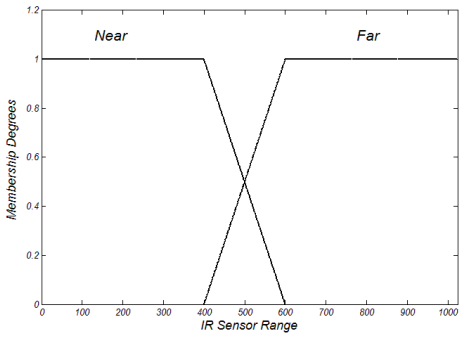

Figure 2. Fuzzy membership function

Fuzzy sets are often defined as piecewise linear shapes (triangular or trapezoidal) to reduce the computational complexity of acquiring membership degrees. Fuzzy logic is able to handle imprecise data efficiently. If input data is sensed imprecisely, the membership degree in each fuzzy set may change but the relative membership degrees over fuzzy sets remain the same qualitatively. Fuzzy operators are applied to combine input variables and manipulate the output fuzzy set for each rule. The outputs from all rules are then aggregated to form a single fuzzy set. To extract a crisp output from the aggregated fuzzy set, a defuzzification method is applied [23].

B. Q-Learning

Q-learning usually provides best fit to the requirements of real world applications due to its unsupervised, dynamic, and model free structure. The robot observes the world as a set of state-action pairs. To each action a Q value is assigned. Transition from one state to another (via proper actions) may bring it punishment or reward, and accordingly Q values are updated as (1):

$$Q(x_t, a_t) \leftarrow Q(x_t, a_t) + \beta\{r_t + \gamma V_t(x_{t+1}) - Q(x_t, a_t)\} \quad (1)$$

where x_t is the current state, a_t is the action taken at state x_t, $Q(x_t,a_t)$ is the Q value of state x_t , $V(x_{t+1})$ is an estimated value of the new state (x_{t+1}), β and γ are the learning rate and forgetting factor respectively both in range of [0 1], and r_t is the immediate reinforcement scalar signal [4].

C. Fuzzy Q-Learning

A heuristically designed fuzzy rule base with N=8 rules is applied. Two minor tasks of goal reaching and obstacle avoidance are embedded in the designed rule base. The approach is to reach the goal while avoiding encountered obstacles. Q-learning is responsible to keep a balance between these two tasks by tuning fuzzy output membership functions. Input variables are the three groups of Khepera infrared sensors mentioned in Section III i.e. SL, SF, and SR which approximately cover a 180 degrees view of the surroundings. To reduce the number of rules, just two linguistic adjectives are considered for inputs: Far (F) and Near (N) whose membership functions are depicted in Figure 2. If two or more obstacles are sensed simultaneously, the approach is to avoid the nearest obstacle. The nearest obstacle is determined by checking the value of parameter (P) as below:

If SL>SR; P=1; Else P=0; end

The outputs are the robot speed (S) and steering angle (Φ). S_i ($1 \le i \le N$) is a fixed value for each rule which can be zero (Z), C1*Vmax, C2*Vmax, or Vmax. C1 and C2 are constants smaller than one and C1<C2. Vmax is the maximum considered speed. φ_i is described by linguistic adjectives of PB(Positive Big), PS(Positive Small), Z(Zero), NS(Negative Small), and NB(Negative Big) which are symmetric i.e. PB=-NB, PS=-NS. The rule base is as follow:

R_1: IF (SL,SF,SR)=NNN THEN S_1=Z and φ_1= (2*p-1)PB

R_2: IF (SL,SF,SR)=NNF THEN S_2=C1*Vmax and φ_2=PB

R_3: IF (SL,SF,SR)=NFN THEN S3=C2*Vmax and φ_3=Z

R_4: IF SL(SL,SF,SR)=NFF THEN S_4=C2*Vmax and φ_4= PS

R_5: IF (SL,SF,SR)=FNN THEN S_5=C1*Vmax and φ_5=NB

R_6: IF (SL,SF,SR)=FNF THEN S_6=C1*Vmax and φ_6=(2*p-1)PB

R_7: IF (SL,SF,SR)=FFN THEN S_1=C2*Vmax and φ_7=NS

R_8: IF (SL,SF,SR)=FFF THEN S_1=Vmax and φ_8=Tg

Tg is the target angle relative to robot axis. Equation (2) computes the robot speed:

$$S = \frac{\sum_i \alpha_i s_i}{\sum_i \alpha_i} \quad (2)$$

where α_i is the truth value of the ith rule calculated by product method. For each rule, there are J=6 suggested steering angles ($\varphi[i,j]$, $1 \le j \le J$) to which q values are assigned (q[i,j]). J suggested singletons are distributed equally in predefined intervals. To obtain the robot steering angle a fuzzy Q-learning approach is applied as depicted in Fig. 3. q is a N*J matrix whose elements (q[i,j]) are initialized to zero. Among the J suggested actions for each rule, only the action associated with maximum q value (5) is selected and takes part in computation of the overall action see (3) and (4).

1. t=0, observe the state x_t.
2. For each rule i, choose $\varphi[i,j]$ with maximum q value.
3. Compute $\Phi(x_t)$ and its corresponding $Q(x_t, \Phi(x_t))$.
4. Apply the action $\Phi(x_t)$. Observe the new state x_{t+1}.
5. Receive the reinforcement r_t.
6. Compute an estimate value of the new state x_{t+1} i.e. V_{xt+1} .
7. Update q[i,j].
8. t←t+1. Go to step 1.

Figure 3. Fuzzy Q-learning algorithm

$$\phi = \frac{\sum_i \alpha_i \varphi_{im}}{\sum_i \alpha_i}; 1 \le i \le N, 1 \le j \le J \quad (3)$$

$$\phi_{im} = \varphi[i,j] \text{ with } \max_j q[i,j] \quad (4)$$

In each iteration of the algorithm, maximum q values are updated applying (5),(6),(1),(7) respectively.

$$q_{im} = \max_j q[i,j] \quad (5)$$

$$Q(x_t, \Phi(x_t)) = \sum_i \alpha_i q_{im} \quad (6)$$

$$q_{im} = q_{im} + \alpha_i dQ \quad (7)$$

Choosing an appropriate reinforcment signal (r_t) is a deciding factor in the overall performance of Q-learning. r_t is usually defined with respect to the existing obejctives and constriants which depend on the problem under study. In this work, the objective is to get closer to the goal and the constraint is to keep a safe margine from obstacles. r_t is as follow:

$$r_t = \begin{cases} 30; & SL(x_{t+1}) < SL(x_t), SF(x_{t+1}) < SF(x_t), SR(x_{t+1}) < SR(x_t), \\ & \text{and } Dg(x_{t+1}) < Dg(x_t) \\ 20; & \max(SL(x_{t+1}), SF(x_{t+1}), SR(x_{t+1})) < Th, \\ & and\ Dg(x_{t+1}) < Dg(x_t) \\ 10; & SL(x_{t+1}) < SL(x_t), SF(x_{t+1}) < SF(x_t), SR(x_{t+1}) < SR(x_t) \\ 5; & \max(SL(x_{t+1}), SF(x_{t+1}), SR(x_{t+1})) < Th \\ -2; & \text{Otherwise} \end{cases}$$

where Th is a fixed threshold indicating a predefined distance margin from walls or obstacles (30 mm), and Dg is the distance between the robot and the goal.

V. SIMULATIONS

The navigation algorithms are usually evaluated and compared to each other according to specific criteria such as speed and safety. For a fast navigation the robot should choose the shortest path to reach the goal. For a safe navigation, the robot should keep away from obstacles. An optimal navigation entails a tradeoff between speed and safety. To evaluate the performance of the proposed navigation method, experiments were carried out on 30 simulated environments. The number, shape, size, and place of obstacles were changed randomly in different environments. Furthermore, in each environment, various cases were considered for the robot initial position and the goal location. Three instances of fuzzy Q-learning performance in simulated environments are demonstrated through Fig.4, Fig.6, and Fig.8; as observed the robot travels safe, smooth, and short paths to the goal. The complexity of other simulated environments is the same as the depicted instances.

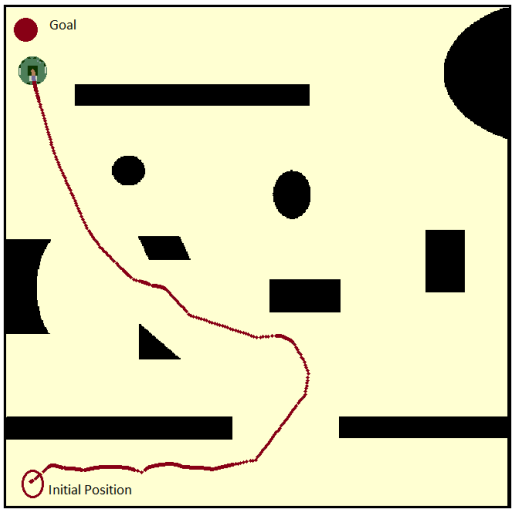

Figure 4. The most repeated path (over 20 runs) generated by fuzzy Q-learning method

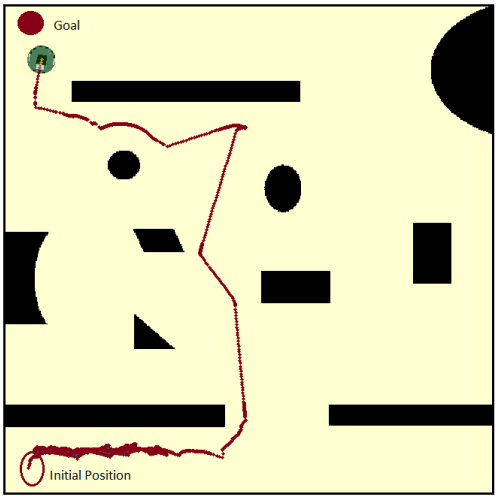

Figure 5. The most repeated path (over 20 runs) generated by fuzzy method

Each experiment is repeated for 20 times to ensure the reliability of obtained results. The average behavior of the proposed method is summarized quantitatively in TABLE I according to its speed, safety, and probability of reaching the goal. An obstacle collision is recorded when the distance of the robot from an obstacle or a wall gets less than 10mm. To obtain the percentages produced in TABLE I, results of 20 runs were averaged. These averages were again averaged over different cases (at least four cases) of the robot initial position and the goal location in one simulated environment. Finally, the percentages gathered from 30 different environments are averaged to constitute the ultimate results. As can be seen, applying the proposed fuzzy Q-learning method, the robot is 95% probable to reach the goal, and may go beyond the safe margin from obstacles or walls by 15%.

TABLE I. PERFORMANCE OF FUZZY Q-LEARNING METHOD

Goal Hits[a]		Goal Misses
95%		
Obstacle Collisions	Steps	**5%**
15%	**550**	

a. All the percentages are the average results over 30 simulated environments.

TABLE II. PERFORMANCE OF FUZZY METHOD

Goal Hits[a]		Goal Misses
70%		
Obstacle Collisions	Steps	**30%**
20%	**850**	

a. All the percentages are the average results over 30 simulated environments.

To spot the effective role of Q-learning in tuning fuzzy inference parameters, the same experiments were repeated for a fuzzy rule base whose parameters were set offline and without Q-learning (TABLE II). The outputs membership functions in fuzzy method are singletons fixed at the mean value of intervals considered for fuzzy Q-learning method. Comparison between two methods reveals the superiority of fuzzy Q-learning over fuzzy method in terms of speed, safety, and probability of reaching the goal. The maximum number of steps is considered 3000 for both methods. Three instances of fuzzy method performance are visualized in Fig.5, Fig.7, and Fig.9.

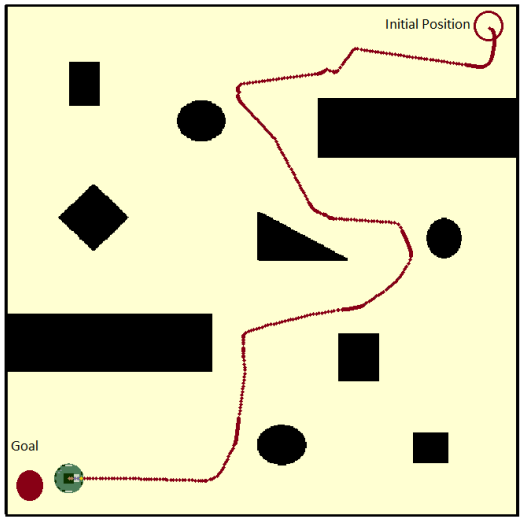

Figure 6. The most repeated path (over 20 runs) generated by fuzzy Q-learning method

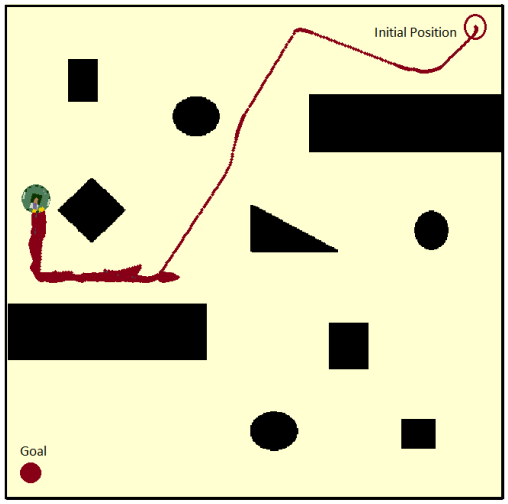

Figure 7. The most repeated path (over 20 runs) generated by fuzzy method

As can be seen in Fig.5, applying the fuzzy method, the robot moves forward and backward to pass the corridor (see the thick line near the initial position) while such problems do not occur with fuzzy Q-learning method. Fig.7 and Fig.9 illustrate situations where the fuzzy method fails to navigate the robot to the goal locating behind an obstacle. Such failures with fuzzy method are due to its inflexible and offline parameter adjustment which is not responsive for all situations. On the other hand, the success of fuzzy Q-learning in similar situations is due to online tuning of fuzzy parameters by selecting the most reinforced actions from several candidates.

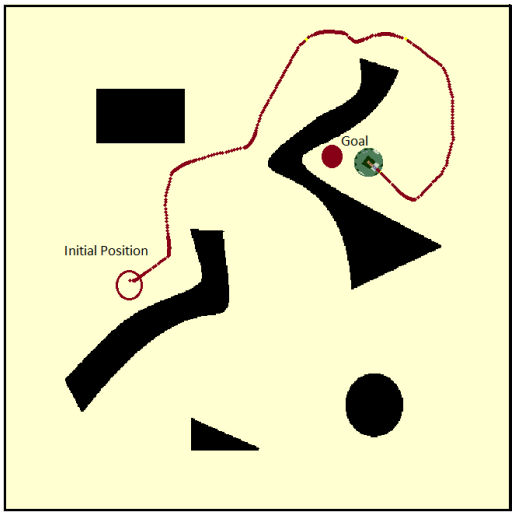

Figure 8. The most repeated path (over 20 runs) generated by fuzzy Q-learning method

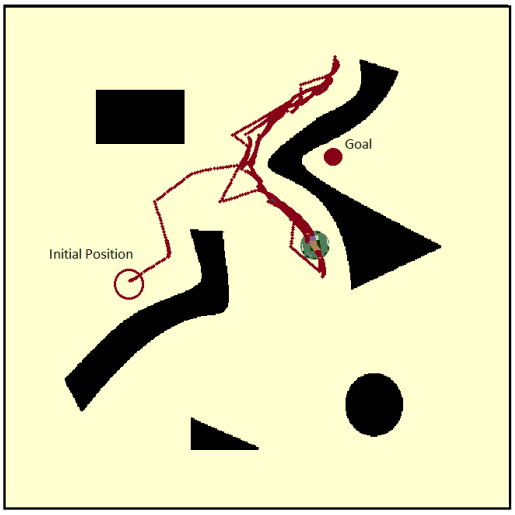

Figure 9. The most repeated path (over 20 runs) generated by fuzzy method

VI. CONCLUSION

A fuzzy Q-learning method is proposed to address autonomous robot navigation. A set of fuzzy rules are designed heuristically, and Q-learning is applied to tune fuzzy inference parameters. Performance of the proposed method is experimented on a simulated Khepera robot. The obtained results validated efficiency of the proposed navigation algorithm quantitatively and qualitatively in terms of speed, safety, and probability of reaching the goal. To highlight the role of Q-learning, the same experiments were repeated with a fuzzy approach without Q-learning. Comparisons revealed that online tuning of fuzzy inference helped a faster, safer, and smoother navigation, and provided effective solutions to get rid of obstacles blocking the goal which lead to an enhanced probability of goal reaching.

REFERENCES

[1] C. Galindo, J.-A. Fernández-Madrigal, and J. González, "Improving Efficiency in Mobile Robot Task Planning Through World Abstraction," IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION vol. 20, no. 4, 2004.

[2] A. Saffiotti, "The uses of fuzzy logic in autonomous robot navigation," Soft Computing, vol. 1, pp. 180-197, 1997.

[3] F. Mondada, E. Franzi, and P. Lenne, "Mobile robot miniaturization: a tool for investigation in control algorithms," in the International symposium on experimental robotics, Kyoto, 1993, pp. 336-341.

[4] P. Y. Glorennec, "Reinforcement Learning: an Overview," in ESIT 2000, Aachen, Germany, September 2000.

[5] O. Khatib, "Real-time Obstacle Avoidance for Manipulators and Mobile Robots," in IEEE Conference on Robotics and Automation, 1985, pp. 500-505.

[6] J. Velagic, B. Lacevic, and B. Perunicic, "A 3-level autonomous mobile robot navigation system designed by using reasoning/search approaches," Robotics and Autonomous Systems, vol. 54, pp. 989-1004, 2006.

[7] A. H. Fagg, D. Lotspeich, and G. A. Bekey, "A Reinforcement-Learning Approach to Reactive Control Policy Design for Autonomous Robots" in IEEE Conference on Robotics and Automation, 1994.

[8] R. C. Arkin, "Motor schema based navigation for a mobile robot," in the IEEE Int. Conf. on Robotics and Automation, 1987, pp. 264-271.

[9] R. A. Brooks, "A robust layered control system for a mobile robot," IEEE Journal of Robotics and Automation,, vol. 1, no. 2, pp. 14-23, 1986.

[10] J. R. Firby, " An investigation into reactive planning in complex domains," in the AAAI Conf., 1987, pp. 202-206.

[11] A. Fatmi, A. A. Yahmadi, L. Khriji, and N. Masmoudi, "A Fuzzy Logic Based Navigation of a Mobile Robot," World Academy of Science, Engineering and Technology vol. 22, 2006.

[12] H. Maaref, and C. Barret, "Sensor-based fuzzy navigation of an autonomous mobile robot in an indoor environment," Control Engineering Practice vol. 8, pp. 757-768, 2000.

[13] E. H. Ruspini, "Fuzzy logic in the Flakey robot," in the Int. Conf. on Fuzzy Logic and Neural Networks (IIZUKA), Iizuka, JP, 1990, pp. 767-770.

[14] M. Obayashi, T. Kuremoto, and K. Kobayashi, "A Self-Organized Fuzzy-Neuro Reinforcement Learning System for Continuous State Space for Autonomous Robots," in CIMCA.2008, 2008.

[15] S. Kermiche, M. L. Saidi, and H. A. Abbassi, "Gradient Descent Adjusting Takagi-Sugeno Controller for a Navigation of Robot Manipulator," Journal of Engineering and Applied Science vol. 1, no. 1, pp. 24-29, 2006.

[16] P.G. Zavlangas, and S.G. Tzafestas, "Motion control for Mobile Robot Obstacle Avoidance and Navigation: A Fuzzy Logic-Based Approach," Systems Analysis Modelling Simulation, vol. 43, no. 12, pp. 1625-1637 2003.

[17] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement Learning: A Survey," Journal of Artificial Intelligence Research vol. 4, pp. 237-285, 1996.

[18] M. J. Er, and C. Deng, "Online Tuning of Fuzzy Inference Systems Using Dynamic Fuzzy Q-Learning," IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, vol. 34, no. 3, JUNE 2004.

[19] L. Jouffe, "Fuzzy Inference System Learning by Reinforcement Methods," IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, vol. 28, no. 3, AUGUST 1998.

[20] I. Lovrek, R. J. Howlett, and L. C. Jain, "A Simple Goal Seeking Navigation Method for a Mobile Robot Using Human Sense, Fuzzy Logic and Reinforcement Learning," KES 2008, Part I, LNAI 5177, pp. 666-673, 2008.

[21] M. Benreguieg, P. Hoppenot, H. Maaref, E. Colle, and C. Barret, "Fuzzy navigation strategy: application to two distinct autonomous mobile robots," Robotica, vol. 15, pp. 609-615, 1997.

[22] "Khepera User Manual," http://www.k-team.com.

[23] R. Kruse, J. Gebhardt, and F. Klawonn, Foundations of Fuzzy Systems: Wiley and Sons, 1994.