BEHAVIOR BASED CONTROL AND FUZZY Q-LEARNING FOR AUTONOMOUS FIVE LEGS ROBOT NAVIGATION

Prihastono[1,3], Handy Wicaksono[1,5], Khairul Anam[4], Rusdhianto Effendi[1], Indra Adji S.[2],
Son Kuswadi [2], Achmad Jazidie[1]
[1] Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[2] Electronics Eng. Polytechnics Institute of Surabaya, Jln. Raya ITS, Sukolilo, Surabaya Indonesia
[3] Engineering Faculty, Bhayangkara University,Jln A. Yani 114 Surabaya, Indonesia
[4] Department of Electrical Engineering, Faculty of Engineering, University of Jember, Indonesia
[5] Department of Electrical Engineering, Petra Christian University, Surabaya, Indonesia
Email : prihtn@yahoo.com

ABSTRACT

This paper presents collaboration of behavior based control and fuzzy Q-learning for five legs robot navigation systems. There are many fuzzy Q-learning algorithms that have been proposed to yield individual behavior like obstacle avoidance, find target and so on. However, for complicated tasks, it is needed to combine all behaviors in one control schema using behavior based control. Based this fact, this paper proposes a control schema that incorporate fuzzy q-learning in behavior based schema to overcome complicated tasks in navigation systems of autonomous five legs robot.

In the proposed schema, there are two behaviors which is learned by fuzzy q-learning. Other behaviors is constructed in design step. All behaviors are coordinated by hierarchical hybrid coordination node. Simulation results demonstrate that the robot with proposed schema is able to learn the right policy, to avoid obstacle and to find the target. However, Fuzzy q-learning failed to give right policy for the robot to avoid collision in the corner location.

Keywords : behavior based control, fuzzy q-learning

1 INTRODUCTION

Autonomous five legs robot navigation system is a one of active area of legged robot research. To implement such a robot system, it is important for the system to properly react in an unknown environment by learning its actions through experience. For this purpose, reinforcement learning methods have been receiving increased attention for use in autonomous robot systems.

One method that has been widely used is Q-learning. However, since Q-learning deals with discrete actions and states, an enormous amount of states may be necessary for an autonomous robot to learn an appropriate action in a continuous environment. Therefore, Q-learning can not be directly used to such a case due to the problems of the curse of dimensionality.

To overcome this problem, variations of the Q-learning algorithm have been developed. Different authors have proposed to use the generalization of statistical method (hamming distance ,statistical clustering)[1], of generalization ability of feed-forward Neural Networks to store the Q-values[1-3]. Another approach consist in extending Learning into fuzzy environments [4,5] and was called by fuzzy q-learning. In this approach, prior knowledge can be embedded into the fuzzy rules which can reduce training significantly. Therefore, this approach is used in this paper.

Fuzzy Q-learning (FQL) has been used in various field of research, such as robot navigation[2,3], control system[6], robot soccer[7], game[8], and so on[9]. In five legs robot navigation, FQL has been used to generate tasks for navigation purposes like obstacle avoidance[10], wall following[11]. However, most of them was implemented in single task and simple problem. For more complicated problems, it is necessary to design a schema control that involves more than one FQL to conduct the complicated tasks simultaneously. This paper is focused on collaboration between FQLs and behavior-based control in autonomous five legs robot navigation. The rest of the paper is organized as follows. Section 2 describes theory and design of control schema. Simulation result is described in section 3 and conclusion is described in section 4.

2 THEORY AND DESIGN

2.1 Fuzzy Q-learning

Fuzzy Q-learning methods may be considered as an extension of its original version of Q-learning. Q-learning [12] is a reinforcement learning method where the learner builds incrementally a Q-value function which attempts to estimate the discounted future rewards for taking action from given states. Q-value function described by following equation :

$$\hat{Q}(s_t,a_t) = Q(s_t,a_t) + \alpha\left[r_{t+1} + \gamma.V(s_{t+1}) - Q(s_t,a_t)\right] \quad (1)$$

where r is the scalar reinforcement signal, α is the learning rate, γ is a discount factor.

In order to deal with large continuous state, generalization must be incorporated in the state representation. Generalization ability of fuzzy inference system (FIS) can be used to facilitate generalization in the state space and to generate continuous action [10].

Each fuzzy rule *R~* is a local representation over a region defined in the input space and it memorizes the parameter vector *q* associated with each of these possible discrete actions. These Q-values are then used to select actions *so* as to maximize the discounted sum of reward obtained while achieving the task. The rules have the form [4]:

If x is S_i then action = a[i,1] with *q*[i,1]
or a[i,2] with *q*[i,2]
or a[i,3] with *q*[i,3]
...
or a[i,J] with *q*[i,J]

where the state S_i are fuzzy labels and *x* is input vektor (x_1,…., x_n), a[i,J] is possible action and *q*[i,J] is q-values that is corresponding to action a[i,J], and J is number of possible action. The learning robot has to find the best conclution for each rule i.e. the action with the best value.

In order to explore the set of possible actions and acquire expereince through reinforcement signals, the local action are selected using using an exploration-exploitation strategy based on the state-action quality, i.e., *q* values. Here, the simple ε-greedy method is used for action selection: a greedy action is chosen with probability 1-ε, and a random action is used with probability ε . The exploration probability is set by $\varepsilon = \frac{2}{10+T}$ where *T* is the number of trial. The exploration probability is intended to control the necessary trade-off between exploration and control, which is gradually eliminated after each trial.[10]

Let i° be selected action in rule i using action selection mechanims that was mentioned before and i^* such as $q[i,i^*] = \max_{j \le J} q[i,j]$. The infered action a is :

$$a(x) = \frac{\sum_{i=1}^{N} \alpha_i(x) \text{ x } a(i,i^\circ)}{\sum_{i=1}^{N} \alpha_i(x)} \qquad (2)$$

The actual Q-value of the infered action, a, is :

$$Q(x,a) = \frac{\sum_{i=1}^{N} \alpha_i(x) \text{ x } q(i,i^\circ)}{\sum_{i=1}^{N} \alpha_i(x)} \qquad (3)$$

and the value of the states *x* :

$$V(x,a) = \frac{\sum_{i=1}^{N} \alpha_i(x) \text{ x } q(i,i^*)}{\sum_{i=1}^{N} \alpha_i(x)} \qquad (4)$$

If x is a state, a is the action applied to the system, y the new state and r is the reinforcement signal, then *Q(x,a)* can be updated using equtions (1) and (3). The difference between the old and the new *Q(x,a)* can be thought of as an error signal, $\Delta Q = r + \gamma V(y) - Q(x,a)$, than can be used to update the action q-values. By ordinary gradient descent , we obtain :

$$\Delta q[i,i^{\text{o}}] = \varepsilon \text{ x } \Delta Q \frac{\alpha_i(x)}{\sum_{i=1}^{N} \alpha_i(x)} \qquad (5)$$

Where ε is a learning rate.

To speed up learning, it is needed to combine Q-learning and Temporal Difference (TD(λ)) method[4] and is yielded the eligibility e[i,j] of an action y :

$$e[i,j] = \begin{cases} \lambda\gamma e[i,j] + \frac{\alpha_i(x)}{\sum_{i=1}^{N} \alpha_i(x)} & \text{if } j = i^\circ \\ \lambda\gamma e[i,j] & \text{elsewhere} \end{cases} \qquad (6)$$

Therefore, the updating equation (5) become :

$$\Delta q[i,i] = \varepsilon \text{ x } \Delta Q \text{ x } e[i,j]. \qquad (7)$$

The algorithm of fuzzy q-learning as has been expalined before is described below .

1. Observe the state x.
2. for each rule, choose the actual consequence using e-greedy seceltion
3. compute global consequence a(x) and its corresponding Q-value Q(x,a)
4. Apply the actiion a(x). Let y be the new state
5. receive the reinforcement r

Update q-values.

2.2 Behavior Based Control

This paper considers hierarchical control structure (fig. 1) that showing two layers : high level controller and low level controller.

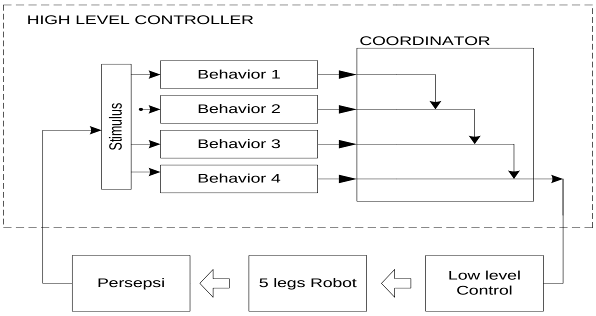


Figure 1. Behavior based Control Schema

High level controller is behavior-based layer that consists of a set of behaviors and a coordinator. This paper uses hybrid coordinator that was proposed by carreras[13]. The hybrid coordinator takes advantage of competitive and cooperative approaches. The hybrid coordinator allows the coordination of a large number of behaviors without the need of a complex designing phase or tuning phase. The addition of a new behavior only implies the assignment of its priority with reference to other behaviors. The hybrid coordinator uses the priority and behavior activation level to calculate the output of the layer, which is the desired control action input to the low-level control system Therefore, the response of each behavior is composed of the activation level and the control action , as illustrated in Fig. 2[13].

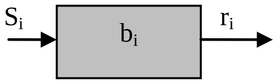


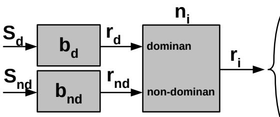
Figure 2. Behavior Normalization [5]

Before entering the coordinator, each behavior is normalized as described in figure 7. In figure 7, S_i is i^{th} behavior and r_i is i^{th} result of behavior normalization that consist of expected control action v_i and activation level (degree of behavior), $a_i \rightarrow 0 – 1$. Behavior coordinator uses r_i behavior responses to compose control action of entire system. This process is executed each sampling time of high level controller.

The coordination system is composed of set of n_i nodes. Each node has two inputs and one output. The inputs are dominant input and non-dominant input. The response that is connected to dominant input has higher priority than the response that is connected to non-dominant input. The node output consists of expected control action v_i and activation level a_i.

When dominant behavior is fully activated, i.e. a_d=1, node output is same as dominant behavior. In this case, the node behaves like competitive coordination.

However, when dominant behavior is partly activated, i.e. $0 < a_d < 1$, the node output is combination of two behaviors, dominant behavior and non-dominant behavior. When a_d=0, the node output will behave like non-dominant behavior. Set of nodes construct a hierarchy called ***Hierarchical Hybrid Coordination Nodes*** (HHCN).



$$\begin{cases} a_i \rightarrow a_d + a_{nd}.(1 - a_d)^k \\ \quad k = 1,2,3,\ldots.. \\ \quad \text{if } (a_i > 1)\ a_i = 1 \\ V_i \rightarrow V_d.\ a_d/a_i + V_{nd}.a_{nd}.(1 - a_d)^k/\ a_i \\ \quad \text{if } (|V_i|>1)\ V_i = V_i/|\ V_i\ | \end{cases}$$

Figure 3. Mathematic formulation of node output [13]

The low-level controller is constructed from conventional control i.e. PID controller. The input is derived from output of high-level controller, that is velocity setting that must be accomplished by motor. This controller has responsibility to control speed motor so that the actual speed motor is same or almost same as the velocity setting from high-level controller.

2.3 Robot Design and Environment model

To test our proposed schema, cluttered environment is created as described in figure 5. The figure 5 is. considerd as cluttered environment because some reasons. The first, there are many objects with various shape and position. Second, the position of the target is hided. This condition give a difficulty to robot to find the target directly.

Figure 4 describe the robot that was used in the testing of proposed schema. The robot has three range finder sensors, and two light sources

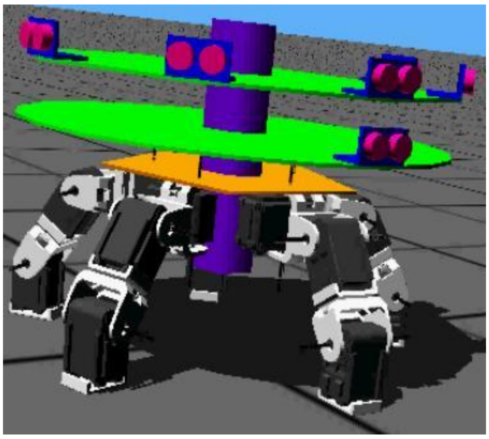


Figure 4. Robot design

Environment model which is used in this paper is showed by figure 5.

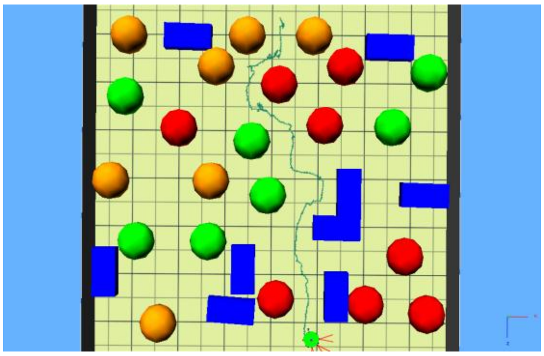


Figure 5. Environment model for simulation purpose

2.4 FQL and BBC for robot control

This paper presents collaboration between Fuzzy Q-Learning and Behavior based control. Most of authors have developed fuzzy q-learning to generate a behavior that is constructed by learning continuously to maximize discounted future reward. However, most of them only focus on generating a behavior for simple environment as showed by Deng[10], Mr Jo [11]. For complex environment, it is necessary to incorporate FQL in behavior-based schema. Therefore, this paper proposes behavior based schema that uses hybrid coordination node [13] to coordinate some behaviors iether from FQl generation or from behavior that is designed in design step. Proposed schema is adapted from [13] and described in figure 6.

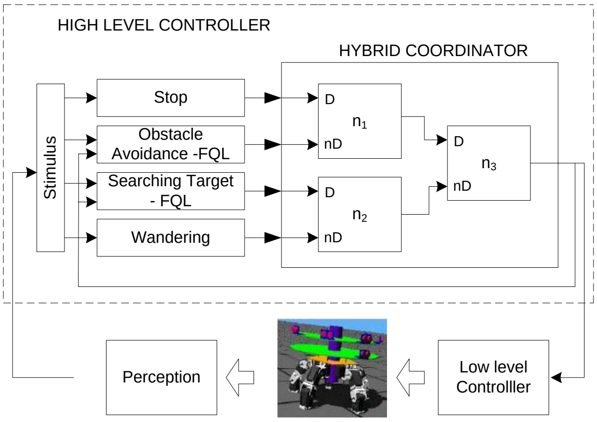


Figure 6. Fuzzy Q-learning in Behavior based Control

In figure 6, High-level controller consists of four behaviors and one HHCN. The four behaviors are stop, obstacle avoidance-FQL, searching target-FQL, and wandreing. Stop behavior has highest priority and wandering behavior has lowest priority. Each behavior is developed separately and there is no relation between behaviors. The output of high-level controller is speed setting to low level controller and robot heading.

The wandering behavior has task to explore the robot environment to detect the existence of target. Activation parameter, a_{tm}, is 1 over time. The output is speed setting that is vary every few seconds.

The obstacle avoidance-FQL behavior is one of behavior that is generated by Fuzzy Q-learning. This behavior has task to avoid every object which is encountered and detected by the ranging finding sensors. The input is distance data between robot and the object from three IR range finder sensors. Output of the range finder sensors is integer value from 0 to 1024. The zero value means that the object is far from the robot. On the contrary, the 1024 value means that the robot has collided the object. The action set consists of five actions: {turn-right, little turn-right, move-forward, little turn-left, turn-left}.

The reinforcement function is directly derived from the task definition, which is to have a wide clearance to the obstacles. Reinforcement signal *r* penalizes the robot whenever it collides with or approaches an obstacle. If the robot collides or the bumper is active or the distance more than 1000, it is penalized by a fixed value, i.e. -1. if the distance between the robot and obstacles is more than a certain threshold, d_k = 300, the penalty value is 0. Otherwise, the robot is rewarded by 1. The component of the reinforcement that teaches the robot keep away from obstacles is:

$$r = \begin{cases} -1 & if\ collision,\ d_s > 1000 \\ 0 & if\ d_s > d_k \\ 1 & otherwise \end{cases} \qquad (8)$$

where d_s is the shortest distance provided by any of IR sensor while performing the action. The value of activation parameter, is proportional to the distance between the sensors and the obstacle..

The searching target behavior has task to find and go to target. The goal is to follow a moving light source, which is displaced manually. The two light sensors are used to measure the ambient light on different sides of the robot. The sensors value is from 0 to 1024.. The action set consists of five actions: {turn-right, little turn-right, move-forward, little turn-left, turn-left, backward}. The robot is rewarded when it is faced toward the light source, and receives punishment in the other cases.

$$r = \begin{cases} -1 & if\ d_s < 300 \\ 0 & if\ d_s < 800 \\ 1 & otherwise \end{cases} \qquad (9)$$

where d_s is the largest value provided by any of light sensor while performing the action.

The stop Behavior will be fully active when the any of light sensor value more than 1000. The goal is to stop the robot when it reaches the light source in certain distance.

3 SIMULATION RESULT

To test performance of the proposed structure control, eight experiments has been conducted. The main goal is the robot has to find and get the target without any collision with the object that was encountered and to reach the target in as quick as possible in cluttered environment figure 5.

From the task definition, there are three performance indicators. The First is robot ability to get the target. The second is robot ability to avoid collision with the obstacle and the last is the time that was needed by the robot to reach the target.

The parameters values that are used in this paper are :

$\alpha = 0.0001$; $\lambda = 0.3$; $\gamma = 0.9$

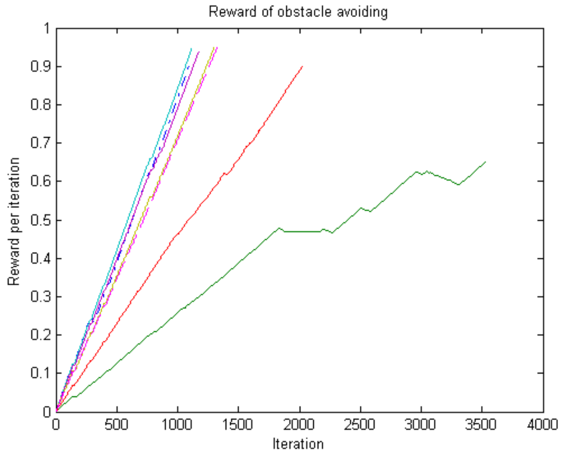


Figure 7. Reward accumulation of FQL-obstacle avoidance

Figure 7 shows the simulation result for eight trials for reward accumulation of FQL-obstacle avoidance. For all of trials, robot has succeeded to reach the target. But the time that was spent to reach the target is different. There are one trial that spent more time than the others. In the trial, the robot have collided more obstacles than the others.

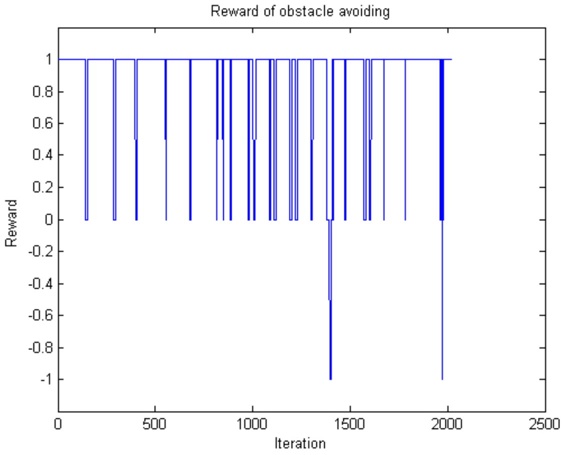


Figure 8. Local reward of FQL-obstacle avoidance

The local reward figure 8 gives more information about the performance of FQL-obstacle avoidance. Robot got many rewards and few penalties.

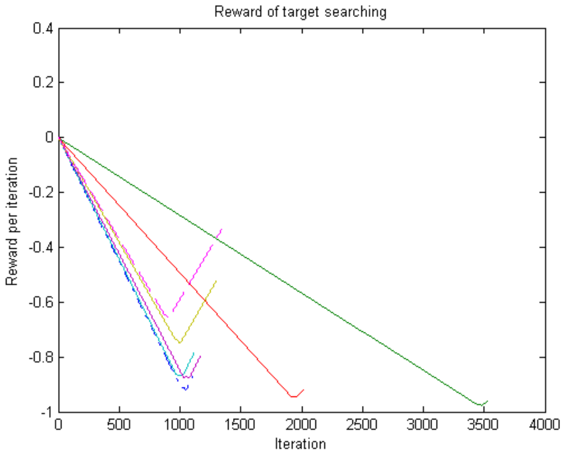


Figure 9. Reward accumulation of FQL-target searching

The performance of FQL-target searching can be analyzed from figure 9 and 10. The reward accumulation tends to go -1. In this condition, robot was trying to find target and the target was still outside scope of the robots. Therefore, in this step, robot was penalized by -1. After exploring the environment, the robot succeed to detect the existence of the target.

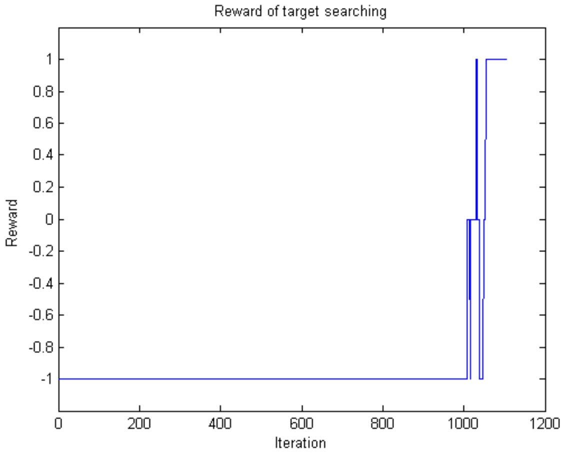


Figure 10. Local reward of FQL-target searching

Another test that was accomplished to measure the performance of the FQL is to test the learning ability of the robot to get the target from different starting point. There are three different starting points. The result of simulation is showed by figure 11.

The trajectory result of figure 11 gives information that robot was able to reach and get the target although it started from different point and it was able to avoid almost all of obstacles that was encountered. It also gives some points that the robot have collided the wall or obstacles.

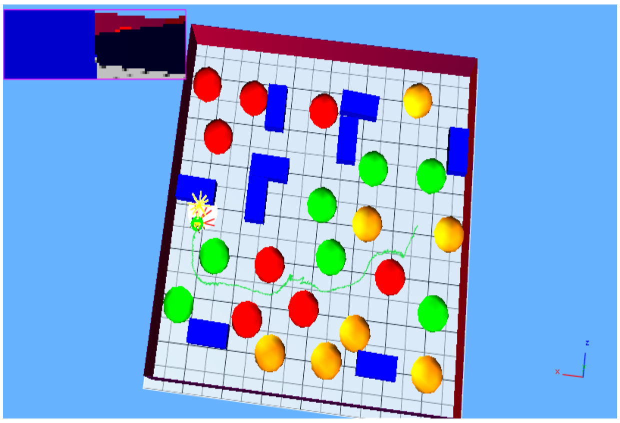


Figure 11. Robot trajectory from different starting point testing

4 CONCLUSION

This paper proposes control schema for navigation system of autonomous five legs robot in complicated environment by incorporating the fuzz q-learning to behavior based control. Two behaviors were generated by fuzzy q-learning by learning the environment continuously. Simulation results demonstrate that the robot with proposed schema is able to learn the right policy, to avoid obstacle and to find the target. However, Fuzzy q-learning failed to give right policy for the robot to avoid collision in the corner location.

5. ACKNOWLEDGEMENT

This work is being supported by Japan International Cooperation Agency (JICA) through Technical Cooperation Project for Research and Education Development on Information and Communication Technology in Sepuluh Nopember Institute of Technology (PREDICT - ITS).

REFERENCE

[1]. C. Touzet,"Neural Reinforcement Learning for Behaviour Synthesis", *Robotics and Autonomous Systems*, Special issue on Learning Robot: the New Wave, N. Sharkey Guest Editor, 1997

[2]. Yang, GS, Chen, ER, Wan, C.(2004), ”Mobile Robot Navigation Using Neural Q Learning”, *Proceeding of the Third International Conference on Machine learning and Cybernatics*, Shanghai, Cina, Vol. 1,p. 48 – 52

[3]. Huang, BQ, Cao, GY, Guo, M.(2005) ,"Reinforcement Learning Neural Network to The Problem Of Autonomous Mobile Robot Obstacle Avoidance "*IEEE Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, Guangzhou, Vol. 1, p. 85-89

[4]. Jouffe,L,"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

[5]. Glorennec, P.Y., Jouffe,L, ”Fuzzy Q-learning”, *Proceeding of the sixth IEEE Internasional Conference on Fuzzy Sistem*, Vol. 2, No. 1, 1997,hal. 659 – 662

[6]. CharlesW. Anderson1, Douglas C. Hittle2, Alon D. Katz2, and R. Matt Kretchmar, "Synthesis of Reinforcement Learning, Neural Networks, and PI Control Applied to a Simulated Heating Coil", *Elsevier : Artificial Intelligence in Engineering*, Volume 11, Number 4, October 1997 , pp. 421-429(9)

[7]. Tomoharu Nakashima, Masayo Udo, and Hisao Ishibuchi, "Implementation of Fuzzy Q-Learning for a Soccer Agent", *The IEEE International Conference on Fuzzy Systems*, 2003

[8]. Ishibuchi, H, Nakashima, T., Miyamoto, H., Chi-Hyon Oh,"Fuzzy QLearning for a Multi-Player Non-Cooperative Repeated Game", *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*,Volume 3, Issue , 1997 Page:1573 - 1579 vol.3

[9]. Ho-Sub Seo, So-Joeng Youn, Kyung-Whan Oh, "A Fuzzy Reinforcement Function for the Intelligent Agent to process Vague Goals", 19th International Conference of the North American Fuzzy Information Processing Society-NAFIPS, 2000, Page(s):29 - 33

[10]. C. Deng, M. J. Er and J. Xu, "Dynamic Fuzzy Q-Learning and Control of Mobile Robots", *8th International Conference on Control, Automation, Robotics and Vision*, Kunming, China, 6-9th December 2004

[11]. Meng Joo Er, Member, IEEE, and Chang Deng, "Online Tuning of Fuzzy Inference Systems Using Dynamic Fuzzy Q-Learning", *IEEE Transactions On Systems, Man, And Cybernetics*, Vol. 34, No. 3, June 2004

[12]. Watkins C., Dayan P.(1992),"Q-learning,Thechnical Note", *Machine Learning*, Vol 8, hal.279-292

[13]. Carreras, M, Yuh, J, Batlle, J, Ridao, P “A Behavior-Based Scheme Using Reinforcement Learning for Autonomous Underwater Vehicles”, *IEEE Journal Of Oceanic Engineering, Vol. 30*, No. 2, April 2005.