ICROS-SICE International Joint Conference 2009
August 18-21, 2009, Fukuoka International Congress Center, Japan

Hybridization of Fuzzy Q-learning and Behavior-Based Control for Autonomous Mobile Robot Navigation in Cluttered Environment

Khairul Anam[1],Prihastono[2,4],Handy Wicaksono[3,4], Rusdhianto Effendi[4], Indra Adji S[5], Son Kuswadi[5], Achmad Jazidie[4], Mitsuji Sampei[6]

[1] Department of Electrical Engineering, University of Jember, Jember, Indonesia
(Tel : +62-0331-484977 ; E-mail: kh.anam.sk@gmail.com)
[2] Department of Electrical Engineering, University of Bhayangkara, Surabaya, Indonesia
(Tel : + 62-031-8285602; E-mail: prihtn@yahoo.com)
[3] Department of Electrical Engineering, Petra Christian University, Surabaya, Indonesia
(Tel : +62-031-8439040; E-mail: handy@petra.ac.id)
[4] Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
(Tel : +62 031-599 4251; E-mail: ditto@ee.its.ac.id, jazidie@ee.its.ac.id)
[5] Electronics Eng. Polytechnics Institute of Surabaya, Surabaya Indonesia
(Tel : +62 031-5947280; E-mail: indra@eepis-its.edu , sonk@eepis-its.edu)
[6] Department of Mechanical and Control Engineering, Tokyo Institute of Technology, Tokyo, Japan
(Tel : +81-3-5734-2552; E-mail: sampei@ctrl.titech.ac.jp)

Abstract: This paper proposes hybridization of fuzzy Q-learning and behavior-based control for autonomous mobile robot navigation problem in cluttered environment with unknown target position. The fuzzy Q-learning is incorporated in behavior-based control structure and it is considered as generation of primitive behavior like obstacle avoidance and target searching. The simulation result demonstrates that the hybridization enables robot to be able to learn the right policy, to avoid obstacle and to find the target. Real implementation of this hybridization shows that the robot was able to learn the right policy i.e. to avoid obstacle.

Keywords: behavior based control, fuzzy q-learning.

1. INTRODUCTION

A cluttered environment is challenging environment for autonomous mobile robot to operate safely in that environment. The learning algorithm is needed for the robot to overcome complicated task in cluttered environment with unknown target position. For this purpose, reinforcement learning methods have been receiving increased attention for use in autonomous robot systems.

To implement the reinforcement learning, it is used Q-learning. However, since Q-learning deals with discrete actions and states, it is not possible to implement it directly in learning of autonomous robot because the robot deals with continuous action and state. 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] and to use generalization ability of feed-forward Neural Networks to store the Q-values [1-3]. The others use fuzzy logic to approximate the Q-values [4,5].

This paper uses Fuzzy Q-learning (FQL) proposed by Glorennec to approximate Q-values. FQL has been used in various field of research [6,7,8]. However, most of them were implemented in single task and simple problem.

For cluttered environment with unknown target position, it is necessary to design a control schema that involves more than one FQL to conduct the complicated tasks simultaneously. This paper focuses on hybridization between FQLs and behavior-based control for autonomous mobile robot navigation. The rest of the paper is organized as follows. Section 2 describes theory and design of control schema. Simulation and real implementation result is described in section 3 and conclusion is described in section 4.

2. THEORY

2.1 Fuzzy Q-learning (FQL)

Fuzzy Q-learning is extension of Q-learning method so that it can hold continuous states and actions. Q-learning [9] is a reinforcement learning method. In this method, 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 can be 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, fuzzy inference system can be used to approximate Q-values. This approach is based on the fact that the fuzzy inference system is universal approximators and good candidates to store Q-values [5].

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 that it can 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 vector (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 learner 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 simple ε-greedy method is used for action selection: a greedy action is chosen with probability 1-ε, and a random action is chosen with probability ε .

Good explanation about fuzzy Q-learning was presented by [5] and 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:

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

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

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

and the value of the states x :

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

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 eqution (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 :

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

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 :

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

Therefore, the updating equation (5) become :

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

2.2 Behavior Based Control

In Behavior-Based Control, the control of the robot is decomposed into several tasks using task achieving behaviors approach. Each task is called by behavior. The structure of behavior-based control is showed by fig. 1.

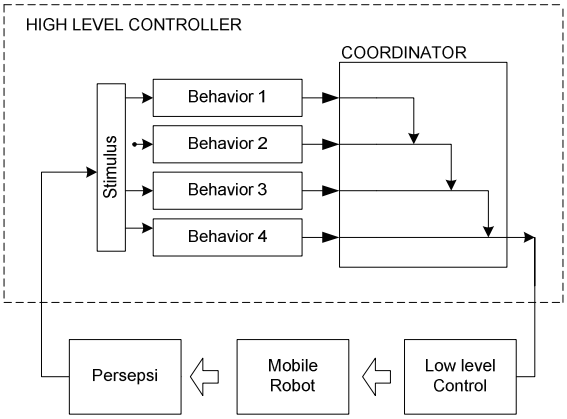

Fig. 1. Behavior-based Control Schema

Based on sensory information, each behavior yields direct responses to control robot according to certain purposes like obstacle avoidance or wall following. Behaviors with different goal can yield conflict uncompleted. Therefore, it is required effective coordination mechanism from the behaviors so that form logic and rational behaviors. This paper uses a hybrid coordinator as proposed by Carreras [11].

In Carreras's hybrid coordinator, 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 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. The formula of hybrid coordinator is showed by fig. 2.

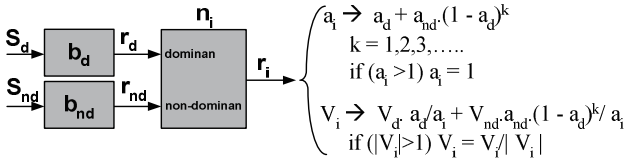

Fig 2. Mathematic formulation of node output [11]

The low-level controller is constructed from conventional control i.e. PID controller. The input is

derived from output of high-level controller. 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.

3. CONTROLLER DESIGN

3.1 Robot Design and Environment

Fig. 3 describes the robot used in the simulation. The robot has three range finder sensors, two light sensors and two touch sensors (bumpers). It is designed using *Webots*.

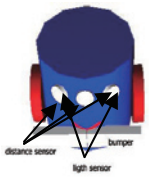

Fig. 3 Robot for simulation

Fig. 4 shows the robot for real implementation from *Bioloid*. The robot has four wheels. Although has four wheels, it use differential mechanism for controlling the wheels. It is also equipped with three infra red distance sensors for detecting obstacles and three light sensors for searching light sources as the target.

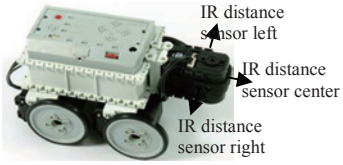

Fig. 4 Mobile Robot for real implementation

Environment model which is used in simulation is showed by fig. 5. To test the control schema, cluttered environment is created as described in it. There are many objects with various shape and position. The position of the target is hided. The area width of the environment is 2 m x 2 m. For real implementation, the environment is not determined exactly.

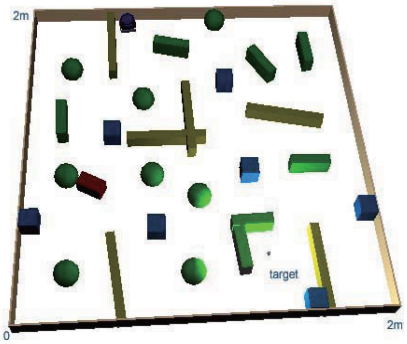

Fig.5 Model Environment for robot simulation

3.2 FQL and BBC for robot control

This paper presents collaboration between Fuzzy Q-Learning and behavior-based control. For complex environment, it is necessary to incorporate FQL in behavior-based schema. Therefore, this paper presents behavior based schema that uses hybrid coordination node [11] to coordinate some behaviors iether from FQl generation or from behavior designed in design step. Presented schema is adapted from [11] and described in figure 6.

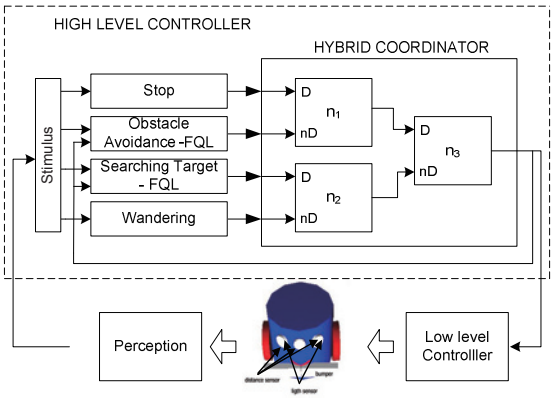

Fig. 6 Fuzzy Q-learning in Behavior based Control

In fig. 6, high-level controller consists of four behaviors and one HHCN. The four behaviors are stop, obstacle avoidance-FQL, searching target-FQL, and wandering. 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. The stop Behavior will be fully active when the any of light sensor value more than 1000.

The obstacle avoidance-FQL behavior is one of behavior 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.

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}. The robot is rewarded when it is faced toward the light source, and receives punishment in the other cases.

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

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

For real implementation, this paper uses control schema described by fig.7. The input of obstacle avoidance-FQL behavior 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 255. The zero value means that the object is far from the robot. On the contrary, the 255 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}.

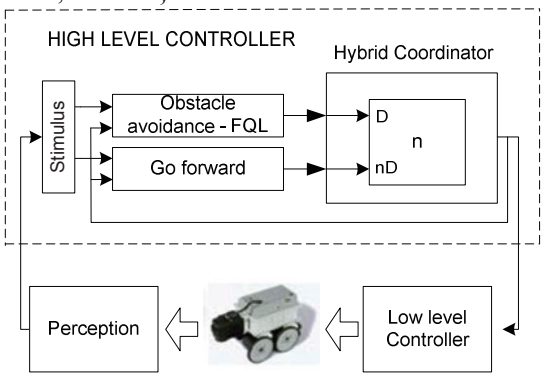

Fig. 7 Control Structure for real implementation

The reinforcement function for real implementation is different from simulation one. If the distance of the robot from objects more than 252, 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 = 200, 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:

$$r = \begin{cases} -1 & d_s > 252 \\ 0 & if\ d_s > d_k \\ 1 & otherwise \end{cases} \quad (10)$$

where d_s is the shortest distance provided by any of IR sensor while performing the action.

Go forward behavior has task to move the robot in forward direction. Activation parameter is 1 over time. The output is speed setting for the low level controller.

4. RESULT

4.1 Simulation Result

To test performance of the control schema, five trials have been conducted. The main goal is to teach the robot so that it can find and get the target without any collision with the object that was encountered in a cluttered environment with unknown target position. The learning parameters value used in this paper are: α = 0.0001, γ= 0.9, and λ =0.3.

Fig. 8 shows the simulation result of reward accumulation of FQL-obstacle avoidance for five trials. For all of trials, robot has succeeded to reach the target. However, the time spent to reach the target is different. One trial spent more time than the others did. In that trial, the robot have collided more obstacles than the others have.

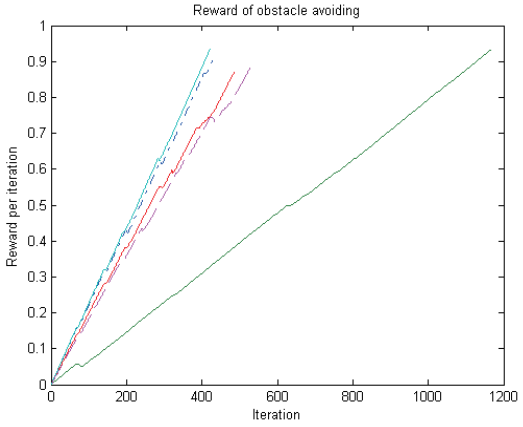

Fig 8. Reward accumulation of FQL-obstacle avoidance

The local reward in fig. 9 gives more information about the performance of FQL-obstacle avoidance. Robot got many positive rewards and few negative rewards.

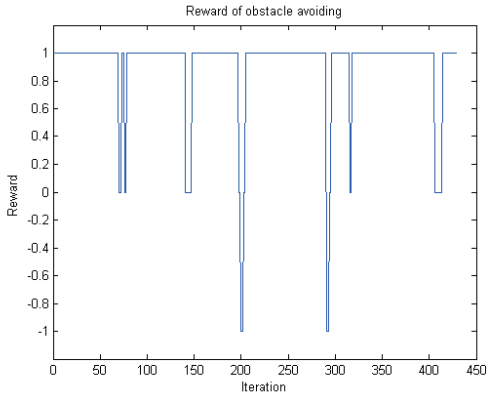

Fig 9. Local reward of FQL-obstacle avoidance

The performance of FQL-target searching can be analyzed from figure 10 and 11. 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 succeeds to detect the existence of the target.

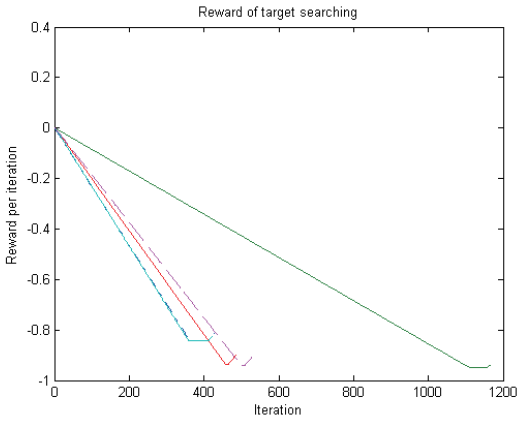

Fig 10. Reward accumulation of FQL-target searching

The performance of FQL-target searching can be analyzed from figure 10 and 11. 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 succeeds to detect the existence of the target.

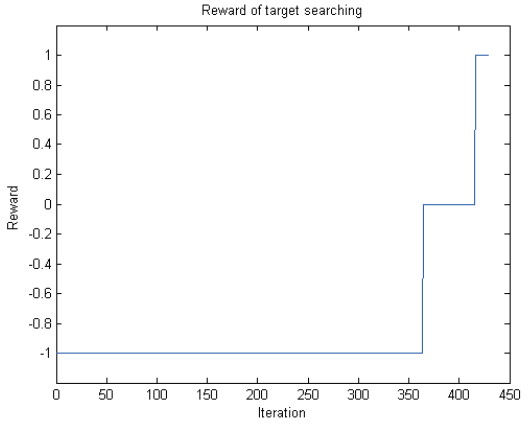

Figure 11. Local reward of FQL-target searching

Another simulation conducted to measure the performance of the FQL is simulation of learning ability of the robot to get the target from different starting point. There are three different starting points. Fig.12 shows the simulation result.

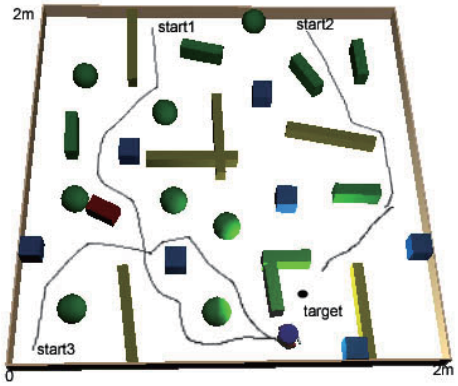

Fig 12. Robot trajectory for different starting point simulation

The trajectory result of fig. 12 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 encountered.

Fig. 13 is test of FQL-target searching. There is only one target but the target position was moved to another place after the robot got the target. In the first effort, the robot must get the first target position. After getting the target, the robot must find and go to the target again because it was moved to second position. Finally, robot could find the third target position after it reaches the second target position target. The trajectory result gives information that the robot was able to track the target wherever target is.

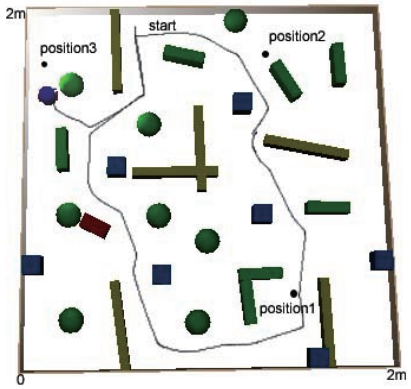

Fig. 13 Robot trajectory for different target position simulation

4.2 Real Implementation

This paper presents real implementation of hybridization fuzzy Q-learning and behavior-based control and the result is embedded robot with learning ability. The main goal of the learning for embedded robot is difference from the simulation one. The goal is to teach the robot so that it can avoid any object encountered in a cluttered environment. The learning parameters value used in this paper are $\alpha = 0.0001$, $\gamma = 0.9$, and $\lambda = 0.3$.

Fig. 14 shows the simulation result of reward accumulation of FQL-obstacle avoidance for embedded robot. For the figure, it is known that the robot needs more learning to improve its performance. However, it also shows that the embedded robot has succeeded to learn the policy.

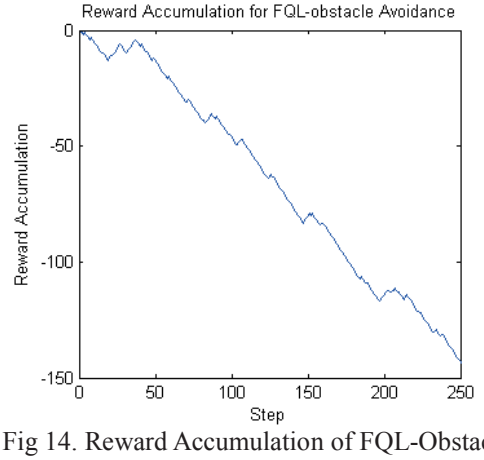

Fig 14. Reward Accumulation of FQL-Obstacle Avoidance for Embedded Robot

The local reward in fig. 15 gives more information about the performance of FQL-obstacle avoidance. Robot got positive rewards and negative rewards. But negative rewards are more often than positive rewards. Therefore the robot needs more experiment.

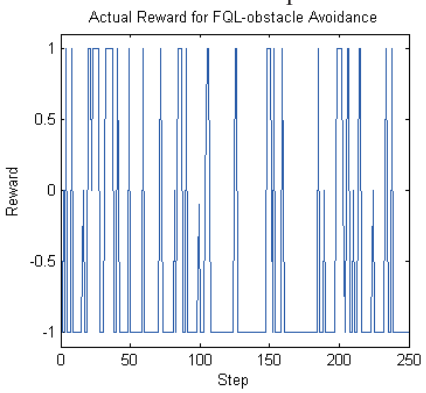

Fig 15. Local reward of FQL-obstacle avoidance for embedded robot

Figure 16 and 17 shows the experimental result of embedded robot in avoiding object, spin left and right respectively. This results show that the embedded robot can learn the given policy, i.e. to avoid obstacle encountered.

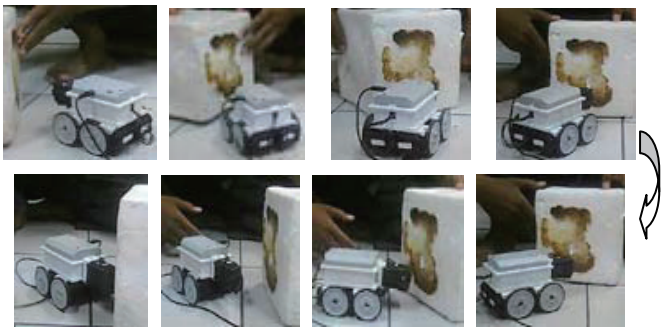

Fig . 16 Learning process of Fuzzy Q-learning for obstacle avoidance (spin left)

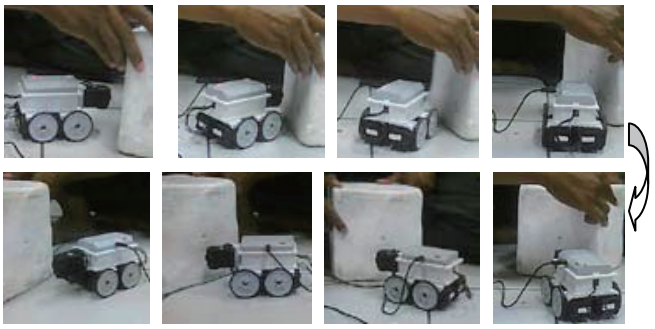

Fig . 17 Learning process of Fuzzy Q-learning for obstacle avoidance (spin right)

5. ACKNOWLEDGMENT

This work is being supported by Japan International Cooperation Agency (JICA) through Technical Cooperation Project for Research and Education on Information & Communication Technology (PREDICT-ITS) Batch III.

6. CONCLUSION

This paper presents hybridization of fuzzy Q-learning and behavior-based control for autonomous mobile robot navigation problem in cluttered environment. Simulation results demonstrate that the robot with this schema was able to learn the right policy, to avoid obstacle and to find the target. Experimental result with bioloid robot showed that the robot can learn the policy. However, its performance is not so good and needs more experiments.

REFERENCES

[1]. C. Touzet, "Neural Reinforcement Learning for Behavior 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., ”Mobile Robot Navigation Using Neural Q Learning”, Proceeding *of the Third International Conference on Machine learning and Cybernetics*, Shanghai, China, Vol. 1,p. 48 – 52, 2004

[3]. Huang, BQ, Cao, GY, Guo, M. ,"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, 2005

[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 International Conference on Fuzzy System*, Vol. 2, No. 1,hal. 659 – 662, 1997

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

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

[8]. 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*, Page(s):29 – 33, 2000

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

[10]. 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