EMBEDDED LEARNING ROBOT WITH FUZZY Q-LEARNING FOR OBSTACLE AVOIDANCE BEHAVIOR

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: *Fuzzy Q-learning is extending of Q-learning algorithm that uses fuzzy inference system to enable Q-learning holding continuous action and state. This learning has been implemented in various robot learning application like obstacle avoidance and target searching. However, most of them have not been realized in embedded robot. This paper presents implementation of fuzzy Q-learning for obstacle avoidance navigation in embedded mobile robot. The experimental result demonstrates that fuzzy Q-learning enables robot to be able to learn the right policy i.e. to avoid obstacle*.

Keywords: fuzzy q-learning, obstacle avoidance

EMBEDDED LEARNING ROBOT USING FUZZY Q-LEARNING FOR OBSTACLE AVOIDANCE BEHAVIOR

ABSTRACT

Fuzzy Q-learning is extending of Q-learning algorithm that uses fuzzy inference system to enable Q-learning holding continuous action and state. This learning has been implemented in various robot learning application like obstacle avoidance and target searching. However, most of them have not been realized in embedded robot. This paper presents implementation of fuzzy Q-learning for obstacle avoidance navigation in embedded mobile robot. The experimental result demonstrates that fuzzy Q-learning enables robot to be able to learn the right policy i.e. to avoid obstacle.

Keywords : behavior based control, fuzzy q-learning

1. Introduction

In unstructured environment, a big change may be happen suddenly. To overcome it, robot control must be able to change its control action to adapt with new condition. Therefore, it is needed control system for robot that can learn its environment.

Because the environment is unstructured and unknown, unsupervised learning is suitable used for enabling the robot to learn its environment. For this purpose, reinforcement learning methods have been receiving increased attention for use in autonomous robot systems. Reinforcement learning can be realized using 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)[3], of generalization ability of feed-forward Neural Networks to store the Q-values[3,5,9]. Another approach consist in extending Learning into fuzzy environments [4] 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 widely used as a method that can enable robot to learn its environment by on line. Various behavior could be generated as long as the robot operates in its environment such as obstacle avoidance and target searching. Anam et all used FQL to generate some behaviors in behavior-based control to control autonomous robot in cluttered environment[7]. Meng Jo et all have developed the FQL and implemented it in mobile robot navigation[8].

However, the implementations of FQL algorithm are restricted in simulation area or in robot that is controlled by personal computer. They are rare implemented in embedded system. Anam et all used it in simulation using webots[7]. Meng Jo et all used it ini khepra robot that is connected to personal computer[8]. This paper focused on implementation of FQL to embedded system. The rest of the paper is organized as follows. Section 2 and 3 describes theory and of fuzzy q-learning respectively. Experimental result is described in section 4 and conclusion is described in section 5.

2. Fuzzy Q-learning

Fuzzy Q-learning methods may be considered as an extension of its original version of Q-learning. Q-learning [8] 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 :

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

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 :

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

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
6. Update q-values.

3. Method

3.1 Robot and Environment

Figure 1 describes the robot used in the practice. It was built from Bioloid Robot module[1]. It has three range finder sensors, three light sensors and one sound sensor. All those sensors are embedded in AX-S1 module. It also uses 4 servo motor of AX-12 that is used in continuous mode. Main Controller of the robot is CM-5. CM-5 is the CPU of Bioloid from atmel ATMEGA128 which acts as the brain of the robot. A button is installed inside to be used as an input device and can function as a remote control. Rechargable batteries are also installed in it. CM-5 communicate with others by serial communication.

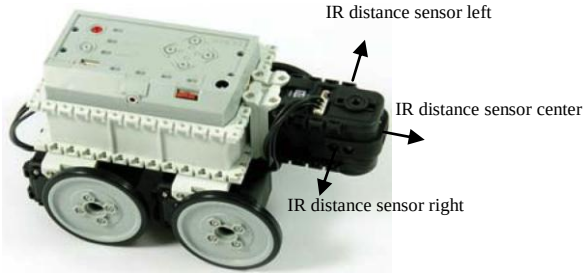


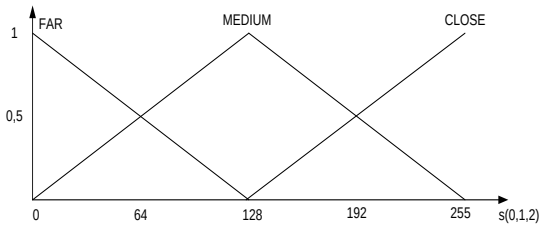
Figure 1. Robot design

To test the FQL algorithm, simple unstructured environment is created and described in figure 2. There are two objects with various shape and position. This condition gives a difficulty to robot to operate savely in it. The area width of the environment is about 1 m x 0.5 m.

Figure 2. Environment model

3.2 Fuzzy Q-learning for Obstacle Avoidance

The obstacle avoidance-FQL is fuzzy Q-learning algorithm that can construct obstacle avoidance behavior by learning the environment. This behavior has task to avoid every object encountered and detected by the range finding sensors. The input of the sensors is distance data between robot and the object from three IR(infra red) range finder sensors. The Output is integer value from 0 to 255. The zero value means that the object is far from the robot and the 255 value means that the robot has collided the object. Figure 3 show the membership function of input for obstacle avoidance-FQL.



Fi

Figure 3. Input Membership Function FQL

Different from conventional fuzzy logic, FQL output is estimation value of q-function that has uncertain values. However, each estimation value is according to control action that will be given to the robot. Therefore, output membership function is represented by action values, not q-function values. The action set consist of five actions: {turn-right, rather turn-right, move-forward, rather turn-left, turn-left}. This paper uses singleton membership function for output membership function as described in figure 4.

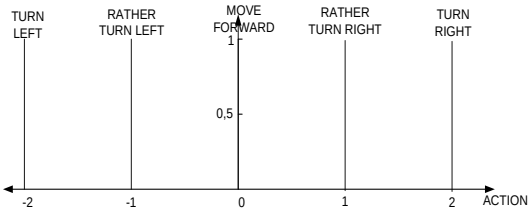


Figure 4. Output Membership Function

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 250, 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 = 180, the penalty value is 0. Otherwise, the robot is rewarded by 1. The components of the reinforcement that teaches the robot keep away from obstacles are:

$$r = \begin{cases} -1 & if \;\; collision,\; d_s > 250 \\ 0 & if \;\; d_s > d_k \\ 1 & otherwise \end{cases} \qquad (8)$$

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.

3.3 Embedded Fuzzy Q-learning

To make fuzzy q-learning algorithm compatible with AVR ATMEGA 128, it is necessary to modify the algorithm that have been implemented in [6]. Some simplifications conducted in original algorithm are :

a. Minimization the use of floating value variable. This can reduce memory used.

b. Choosing of proper time sampling. This paper used 0,1 ms for time sampling.

c. The use of AVR ATMEGA 128 for learning only. It receives sensors data from sensor modules and gives control command to actuators. Sensor modules are separated from the main processor and the communicate with main processor by serial communication.

d. The use of internal EEPROM ATMEGA 128 to store learning results, i.e. q-learning table as result of fuzzy approximation. Therefore, the learning still can continue although the robot was lost of power by restoring the memory in internal EEPROM.

4. Result

To test performance of the learning algorithm in real implementation, four experiments has been conducted. The main goal is the robot has to avoid any collision with the object that was encountered. The parameters values that are used in this paper are α = 0.0001, γ= 0.9, and λ =0.3.

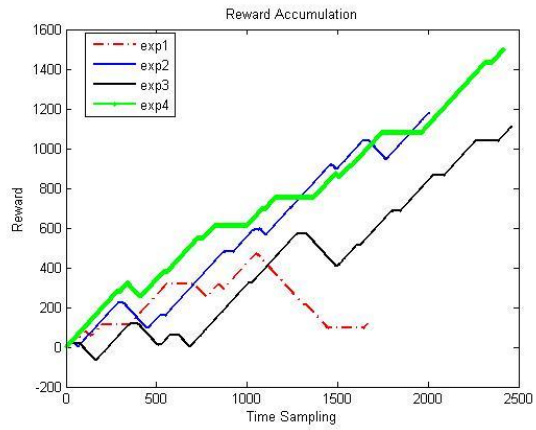


Figure 5. Reward accumulation

Figure 5 shows the result for four trials for reward accumulation. For all of trials, robot has succeeded to maximize the reward accepted. It also shows that the robot have learnt environment by improving the reward accepted by addition of experiment.

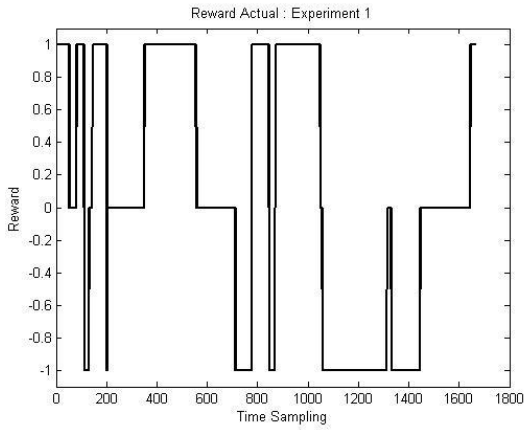


Figure 6. Actual Reward for First Experiment

The actual reward figure 6 gives more information about the performance of FQL-obstacle avoidance. Robot got many rewards and penalties for first experiment. Comparing with fourth experiment in figure 7, after several experiment robot has succeeded to learn the environment.

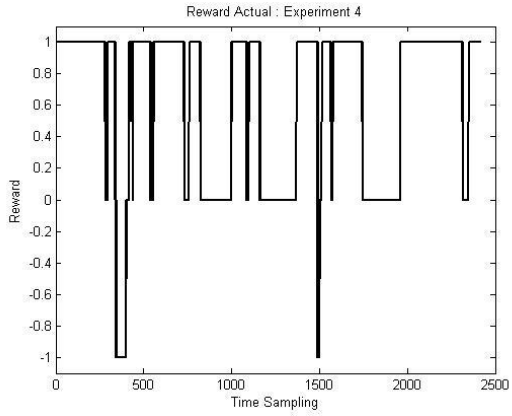


Figure 7. Actual Reward for Fourth Experiment

Figure 8. Robot Trajectory in Beginning Experiment

Figure 8 describes trajectory result of robot in the beginning experiment. Red circles explain the robot collision with object or wall. Whereas Figure 9 describes trajectory result of robot in the last of experiment. It can be concluded that the robot could learn the right policy that is to avoid any object encountered.

Figure 9. Robot Trajectory in Last Experiment

5. Conclusion

This paper presented implementation of fuzzy q-learning for robot learning in embedded system. Experimental results demonstrate that the robot with fuzzy q-learning was able to learn the right policy, to avoid any obstacle encountered.

Reference

[1]. Anonymous,*Bioloid User's Guide,*Robotis Co. Ltd, 2007

[2]. 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

[3]. 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

[4]. 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

[5]. 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, 2005, Vol. 1, p. 85-89

[6]. Khairul Anam, " Behavior-Based Control Scheme With Fuzzy Q-Learning For Autonomous Mobile Robot Navigation System", Master Thesis, Dept. of Electrical Engineering, ITS Surabaya ,2008

[7]. Khairul Anam, Son Kuswadi, Rusdhianto E,"Fuzzy Q-Learning and Hybrid Coordination Node for Autonomous Mobile Robot Navigation in Cluttered Environment", *Proceeding of International Conference on Advanced Computational Intelligence and Its Applications, Universitas Indonesia Jakarta*,2008

[8]. 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

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

[10]. 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