

COMPACT FUZZY Q LEARNING FOR AUTONOMOUS MOBILE ROBOT NAVIGATION

Handy Wicaksono¹, Khairul Anam², Prihastono³, Indra Adjie Sulistijono⁴, Son Kuswadi⁵

¹Department of Electrical Engineering, Petra Christian University

²Department of Electrical Engineering, University of Jember

³Department of Electrical Engineering, University of Bhayangkara

^{4,5}Department of Electrical Engineering, Electronics Engineering Polytechnic Institute of Surabaya

¹Jl. Siwalankerto 121 – 131 Surabaya, Indonesia

¹handywicaksono@yahoo.com

ABSTRACT

Robot which does complex task needs learning capability. Q learning is popular reinforcement learning method because it has off-line policy characteristic and simple algorithm. But it only suitable in discrete state and action. By using Fuzzy Q Learning (FQL), continuous state and action can be handled too. Unfortunately, it's not easy to implement FQL algorithm to real robot because its complexity and robot's limited memory capacity. In this research, Compact FQL (CFQL) algorithm is proposed to solve those weaknesses. By using CFQL, robot still can accomplish its task in autonomous navigation although its performance is not as good as robot using FQL.

KEY WORDS

Autonomous robot, fuzzy Q learning, navigation

1. Introduction

In order to anticipate many uncertain things, robot should have learning mechanism. In supervised learning, robot will need a master to teach it. On the other hand, unsupervised learning mechanism will make robot learn by itself. Reinforcement learning is an example of this method, so robot can learn online by accepting reward from its environment [1].

There are many methods to solve reinforcement learning problem. One of most popular methods is Temporal Difference Algorithm, especially Q Learning algorithm [2]. Q Learning advantages are it has off policy characteristic and simple algorithm. Also it is convergent in optimal policy. But it can only be used in discrete state/action. If Q table is large enough, algorithm will spend too much time in learning process [3].

In order to apply Q learning in continuous/state action, generalization can be done by using many function approximation methods. One of them is Fuzzy Inference System (FIS). It can be used in generalization in state space and it can produce whole continuous action [5]. There are some Fuzzy Q Learning structures that had been made [6] and modified [4][7]

However, FQL is difficult to be applied in real robot. Most of the researches has been done are in form of computer simulation [4], [8], [9]. Mahadevan et. al. [10] has applied Q learning on box pushing robot, but the robot uses computer as controller. This situation enlarges the size and processing time of robot. In addition, Smart et.al. [11] have applied Q learning in real robot, but it still needs supervising phase from human operator.

Difficulties in FQL implementation appears because of robot's limited memory size, low processing performance, and low power autonomy. On the other hand FQL algorithm is complex. In order to overcome those difficulties, Asadpour et.al. [12] have done some simplifications on Q learning algorithm (Compact Q Learning) by using only addition and subtraction operation, and limited number type (integer only). Although development of processor technology is getting faster nowadays, simplification of FQL algorithm will give benefit in speed of processing and also money saving.

On the other hand FQL has been applied in real robot [13], but the author does not give clear steps that have been done. So, by this research, compact FQL design method will be proposed step by step. Robot's ability to accomplish autonomous navigation and amount of receive rewards will be evaluated too. Although experiments still done in computer simulation now, in the future it will be done in real robot application.

2. Behavior Coordination

Robot should have these behaviors to accomplish autonomous navigation.

1. Wandering
2. Obstacle avoidance
3. Search target
4. Stop

Those behaviors must be coordinated so they can work synchronously in robot. Coordination method which is used in this research is Subsumption Architecture [15]. Figure 1. shows robot's behaviors coordination structure.

From the figure, it can be seen that Wandering is the lowest level behavior, so if there are another active behaviors, then Wandering won't be active. Behavior with highest priority level is obstacle avoidance (OA).

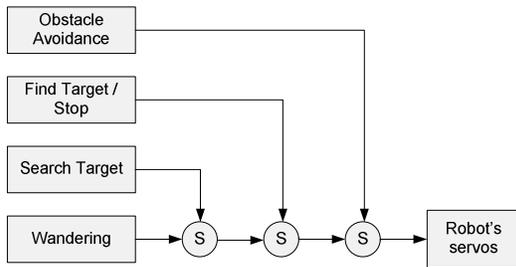


Figure 1. Subsumption Architecture for autonomous navigation robot

3. Robot Learning

3.1 Q Learning

Reinforcement learning is one of unsupervised learning method which learns from agent's environment. Agent (such as: robot) will receive reward from its environment. This method is simple and effective for online and fast process in such an agent like robot. Figure 2. shows reinforcement learning basic scheme.

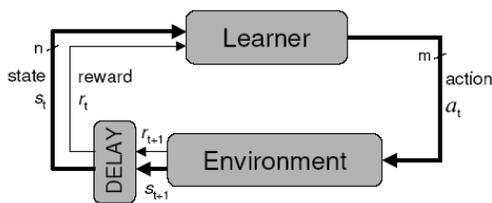


Figure 2. Reinforcement learning basic scheme (Perez, 2003)

Q learning is most popular reinforcement learning method because it is simple, convergent, and off policy. So it is suitable for real time application such as robot. Q learning algorithm is described in Figure 3.

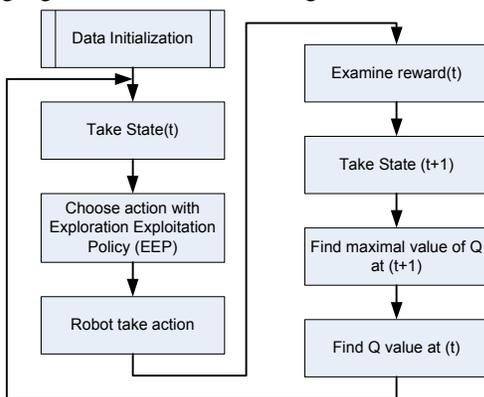


Figure 3. General flow chart of Q learning

Simple Q value equation that used in this algorithm is shown below.

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)] \quad (1)$$

where :

$Q(s,a)$: component of Q table (state, action)

s : state s' : next state

a : action a' : next action

r : reward α : learning rate

γ : discount factor

3.2 Fuzzy Q Learning

Generalization of Q learning is needed when continuous state and action are used. In this case, Q function table will increase to save the new state-action pair. So, learning process needs very long time and big size memory capacity. As the effect, this method is difficult to be applied. By using fuzzy logic as generalization tool, agent can work in continuous state and action. Fuzzy Inference System (FIS) is universal approximator and a good candidate to save Q value.

In Fuzzy Q Learning (FQL), learning isn't done in each state on the state space. So optimization in some representative states are needed. In this case, fuzzy interpolation can be used to predict state and action [7]. Figure 4. shows flow chart of FQL algorithm.

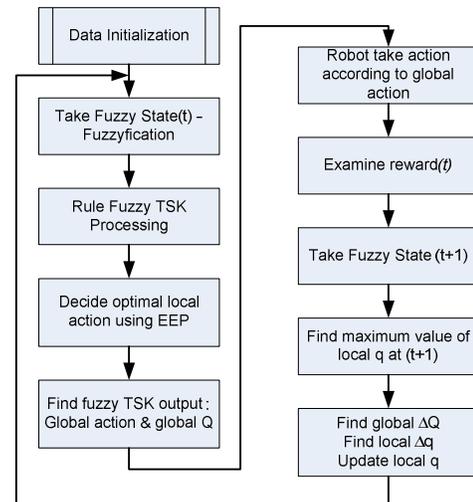


Figure 4. General flow chart of fuzzy Q learning

3.3 Compact Fuzzy Q Learning

CFQL algorithm is made based on some suggestion of Asadpour et.al. [14]. It is said that memory consumption saving on processor can be done by considering these things below.

- Using integer type number only in program (without floating type number), although it can increase number range used in the program.

- Using *unsigned number* only (without negative sign numbers).
- Choosing to use addition – subtraction operation than multiplication – division operation.
- Don't use Exploitation Exploration Policy which contains complex equation (i.e: Boltzman distribution). *Greedy* or ϵ -*greedy* method can be used here.

In order to implement this algorithm in robot, Subsumption Architecture will be used here in Figure 5.

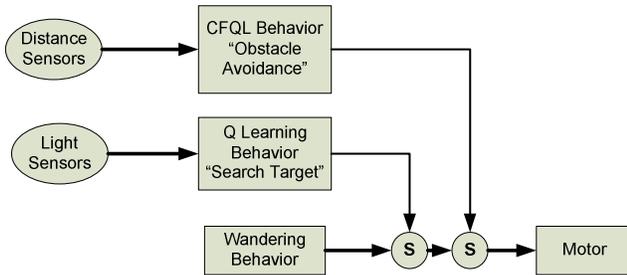


Figure 5. Robot architecture using CFQL behavior

Compact Fuzzy Q learning in this research only used in robot's obstacle avoidance behavior because search target behavior have some random characteristic. Figure 5. shows scheme of CFQL behavior implementation.

Next step is adjustment of distance sensor's membership functions. This is ideal distance sensor in robotic simulator software (Webbots 5.5.2). Triangle membership function (MF) will be used here as shown in Figure 6. This MF needs little modification to avoid floating type number like shown in Figure 7.

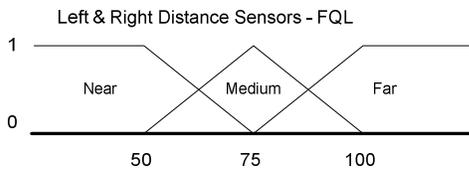


Figure 6. Membership function of left & right distance sensor – FQL

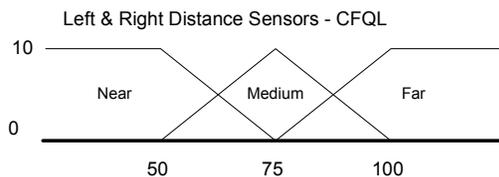


Figure 7. Membership function of left & right distance sensor - CFQL

In FQL algorithm, there are 3 kind of actions that produced here : turn left, forward, and turn right, which is described in Figure 8. In order to avoid negative number, those actions will be modified like shown in Figure 9.

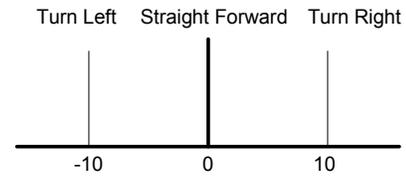


Figure 8. Three possible actions in FQL

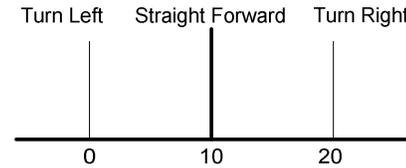


Figure 9. Three possible actions in CFQL

Fuzzy Takagi Sugeno Kang (TSK) will be used here. Rule base that has been used appears as 9 rules description.

1. If $ir_0 = \text{far}$ and $ir_1 = \text{far}$ then actions are (a11, a12, a13) which are suitable with (q11, q12, q13)
2. If $ir_0 = \text{far}$ and $ir_1 = \text{medium}$ then actions are (a21, a22, a23) which are suitable with (q21, q22, q23)
3. If $ir_0 = \text{far}$ and $ir_1 = \text{near}$ then actions are (a31, a32, a33) which are suitable with (q31, q32, q33)
4. If $ir_0 = \text{medium}$ and $ir_1 = \text{far}$ then actions are (a41, a42, a43) which are suitable with (q41, q42, q43)
5. If $ir_0 = \text{medium}$ and $ir_1 = \text{medium}$ then actions are (a51, a52, a53) which are suitable with (q51, q52, q53)
6. If $ir_0 = \text{medium}$ and $ir_1 = \text{near}$ then actions are (a61, a62, a63) which are suitable with (q61, q62, q63)
7. If $ir_0 = \text{near}$ and $ir_1 = \text{far}$ then actions are (a71, a72, a73) which suitable with (q71, q72, q73)
8. If $ir_0 = \text{near}$ and $ir_1 = \text{medium}$ then actions are (a81, a82, a83) which suitable with (q81, q82, q83)
9. If $ir_0 = \text{near}$ and $ir_1 = \text{near}$ then actions are (a91, a92, a93) which suitable with (q91, q92, q93)

In simple table form, those rules can be written in Table 1.

Table 1
Simple rule bases of fuzzy TSK

	NF1	NF2	NF3
MF1	1	2	3
MF2	4	5	6
MF3	7	8	9

4. Simulation Result

4.1 Robot

Robot used here is wheeled robot that has two distance sensors and two light sensors. It only uses two motors. The complete parts of robot can be shown in Figure 10.

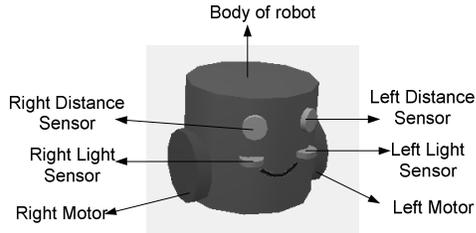


Figure 10. Wheeled robot used in simulation

Webots 5.5.2 software from Cyberbotics has been fully used to simulate and test the performance of robot.

4.2 Q Learning Simulation

In this section, wheeled robot with Q learning behaviors (obstacle avoidance and search target) will be tested. Reward design for this robot shown below:

- $r = 1$, if left distance sensor ≤ 100 and right distance sensor ≤ 100
- 0 , if left distance sensor ≤ 100 and right distance sensor > 100 or right distance sensor ≤ 100 and left distance sensor > 100
- -1 , if left distance sensor > 100 and right distance sensor > 100

It can be concluded from reward design that less distance sensor value means robot is getting farther from the obstacle. So robot will get positive reward and vice versa.

Figure 11. shows rewards which are accepted by robot in obstacle avoidance behavior in 500 iterations.

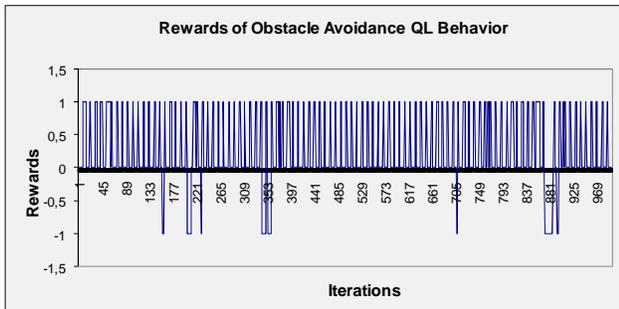


Figure 11. Rewards which are accepted by robot for QL obstacle avoidance behavior

From the Figure 11., it can be seen that robot accept positive rewards consistently. Negative rewards still accepted by robot shows that obstacle around the robot is complex. After some seconds, robot can accomplish its mission well.

Here is the figure of robot's accumulated rewards.

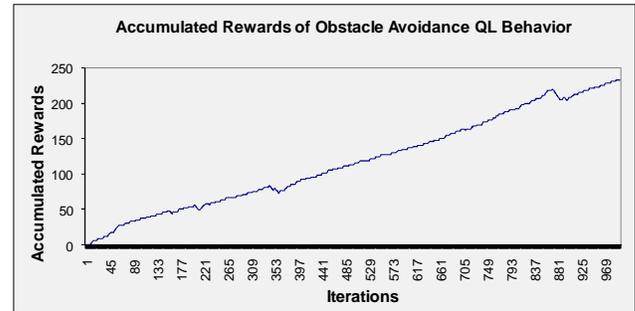


Figure 12. Accumulated rewards which are accepted by robot for QL obstacle avoidance behavior

By seeing at Figure 12., it is clear that accumulated rewards that accepted by robot is getting bigger through time.

Simulation result of search target behavior for 1000 times iteration can be seen here. The reward design shown below:

- $r = -2$, if left light sensor ≤ 300 and right light sensor ≤ 300
- -1 , if left light sensor ≤ 300 and right light sensor > 300 or right light sensor ≤ 300 and left light sensor > 300
- 2 , if left light sensor > 300 and right light sensor > 300

The same conclusion with preceding design can be applied here.

Here is figure that shows rewards which is accepted by robot in search target behavior.

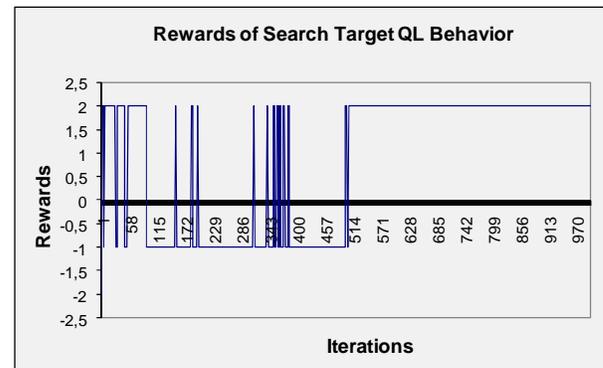


Figure 13. Rewards which are accepted by robot for QL search target behavior

From the figure above, it can be seen that in the beginning robot often accept negative rewards. It is happened because robot still in target searching process. But after it find the target, the robot getting closer to the target so it accept positive rewards.

Accumulated rewards which is accepted by robot shown in figure below. It described the same fact with preceding figure.



Figure 14. Accumulated rewards which are accepted by robot for QL search target behavior

Overall robot behaviors can be seen by its capability in doing autonomous navigation by avoiding the obstacles and find the target. Here are robot performances in autonomous navigation from 3 different start positions.

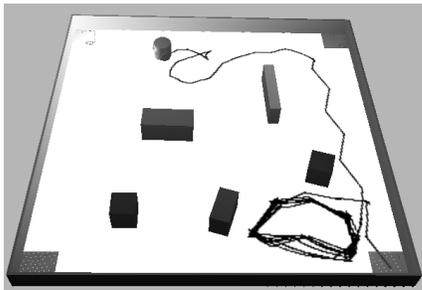


Figure 15. Robot trajectory from 1st start position

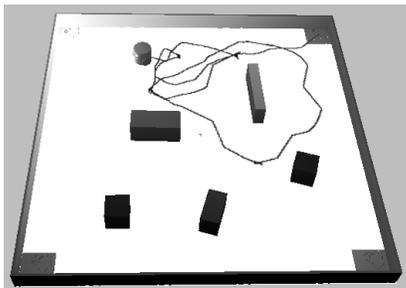


Figure 16. Robot trajectory from 2nd start position

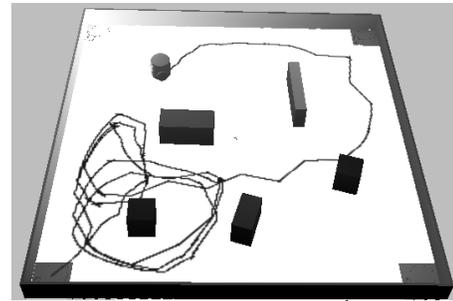


Figure 17. Robot trajectory from 3rd start position

From simulation result, it appears that robot succeed to accomplish its mission well. Although in some condition the robot has been wandering around in the same area, but at last robot can get out of the “stuck” condition.

4.3 Fuzzy Q Learning Simulation

In this simulation, the steps that have been used in preceding simulation will be followed. Here is simulation result of obstacle avoidance behavior for 1000 iterations. Reward design used here is same with preceding behavior.

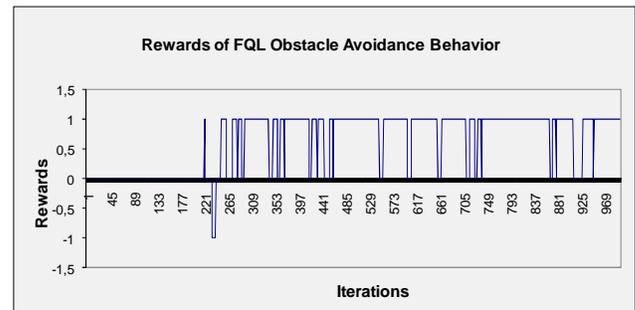


Figure 18. Rewards which are accepted by robot for FQL obstacle avoidance behavior

From the figure above, it can be seen that in the beginning robot receive zero and negative rewards. But after that, robot keep on getting positive rewards. The rewards which are accepted by FQL behavior is more and more consistent than QL behavior (see Figure 11.).

Accumulated rewards are appeared on Figure 18. In 1000 iterations, robot with FQL behavior accepts more than 600 rewards, while robot with QL behavior only accepts 200 rewards (see Figure 12.).

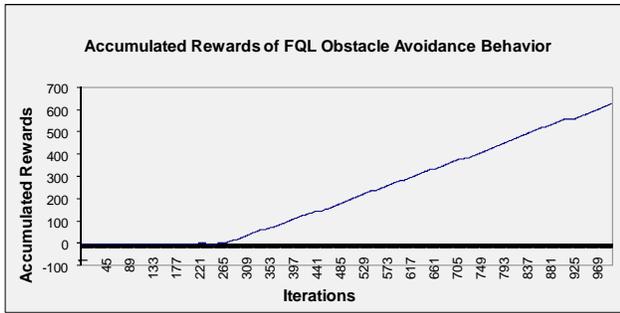


Figure 19. Accumulated rewards which are accepted by robot for FQL obstacle avoidance behavior

By using the same rule with preceding simulation, here are the simulation results

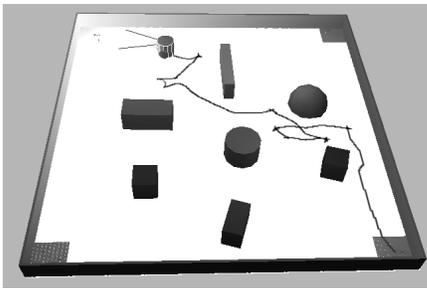


Figure 20. Robot's trajectory by using FQL from start position 1

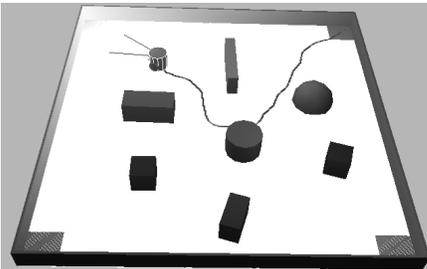


Figure 21. Robot's trajectory by using FQL from start position 2

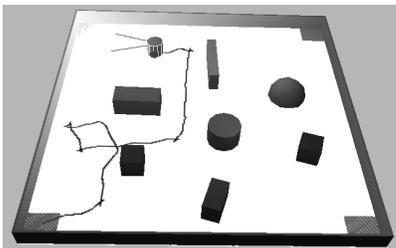


Figure 22. Robot's trajectory by using FQL from start position 3

From Figure 20 - 22, it can be seen that robot succeed to complete its mission. If the results are compared with Q Learning implementation (Figure 14 – 16), it is clear that

this robot is faster in finding the target and its movement is smoother too than the preceding robot.

4.4 Compact Fuzzy Q Learning Simulation

In this section, simulation of robot using compact fuzzy Q learning (CFQL) will be presented. Simulation results from CFQL obstacle avoidance behavior for 500 iterations are shown in Figure 23. There are no negative rewards given here in order to follow CFQL rule. Here is the reward design:

- $$r = \begin{cases} 0 & , \text{ if left distance sensor } \leq 100 \text{ and right distance sensor } \leq 100 \\ 1 & , \text{ if left distance sensor } \leq 100 \text{ and right distance sensor } > 100 \text{ or right distance sensor } \leq 100 \text{ and left distance sensor } > 100 \\ 2 & , \text{ if left distance sensor } > 100 \text{ and right distance sensor } > 100 \end{cases}$$

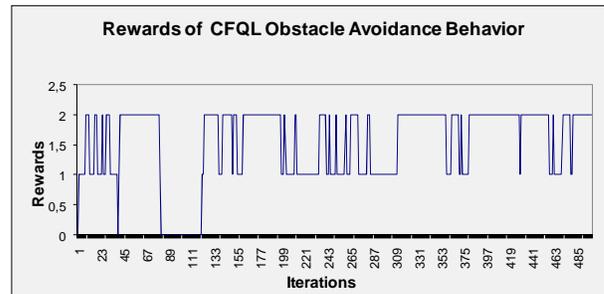


Figure 23. Rewards which are accepted by robot for CFQL obstacle avoidance behavior

Accumulated rewards which are accepted by robot is appeared on Figure 24. It can be seen that in the early stage the robot has been accepted zero and negative rewards. But after several time it continually receives positive rewards. Rewards that has been received by robot with CFQL behavior are not as much as ones that received by FQL robot. But the decreasing is not significant.

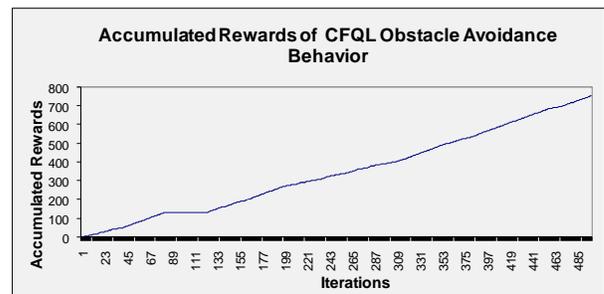


Figure 24. Accumulated rewards which are accepted by robot for CFQL obstacle avoidance behavior

By using the same rule with preceding simulation, here are the simulation results.

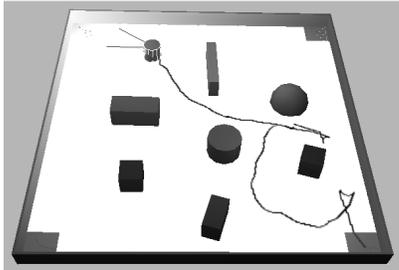


Figure 25. Robot's trajectory by using CFQL from start position 1

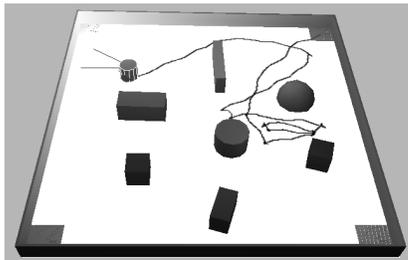


Figure 26. Robot's trajectory by using CFQL from start position 2

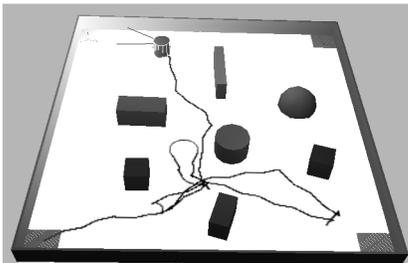


Figure 27. Robot's trajectory by using CFQL from start position 3

From 3 pictures above, it is shown that results given by CFQL are not as well as ones that given by FQL (Figure 20 – 22), but robot with CFQL behavior still can accomplish its mission in avoiding obstacle and finding the target.

5. Conclusion

This paper has been described about design of Compact Fuzzy Q Learning (CFQL) algorithm in robot's autonomous navigation problem. Its performance compared than Q Learning and Fuzzy Q Learning also examined here. From the simulation result, it can be seen that all robots can accomplish its mission to avoid the obstacles and find the target. But robot using FQL algorithm gives the best performance compared than the others because it has the shortest and smoothest path. Although performance of robot using CFQL is below one

using FQL, however it still has shorter and smoother path than one using Q Learning. So it can be concluded that usage of CFQL algorithm in robot's autonomous navigation application is satisfied.

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).

References

- [1] P. Y. Glorennec, Reinforcement Learning : An Overview, *Proceedings of European Symposium on Intelligent Techniques*, Aachen, Germany, 2000.
- [2] C. Watkins and P. Dayan, Q-learning, Technical Note, *Machine Learning*, Vol 8, 1992, pp.279-292.
- [3] M.C. Perez, *A Proposal of Behavior Based Control Architecture with Reinforcement Learning for an Autonomous Underwater Robot*, Tesis Ph.D., University of Girona, Girona, 2003.
- [4] C. Deng, and M. J. Er, Real Time Dynamic Fuzzy Q-learning and Control of Mobile Robots, *Proceedings of 5th Asian Control Conference*, vol. 3, 2004, pp. 1568-1576.
- [5] L. Jouffle, Fuzzy Inference System Learning by Reinforcement Methods, *IEEE Transactions on System, Man, and Cybernetics – Part C : Applications and Reviews*, Vol. 28, No. 3, 1998, pp. 338 – 355.
- [6] P.Y. Glorennec, and L. Jouffle, Fuzzy Q-learning, *Proceeding of the sixth IEEE International Conference on Fuzzy Sistem*, Vol. 2, No. 1, 1997, pp. 659 – 662
- [7] C. Deng, M.J. Er, and J. Xu, Dynamic Fuzzy Q-learning and Control of Mobile Robots, *Proc. of 8th International Conference on Control, Automation, Robotics and Vision*, Kunming, China, 2004.
- [8] I.H. Suh, J.H. Kim, J.H. dan F.C.H. Rhee, Fuzzy-Q Learning for Autonomous Robot Systems, *Proceedings of the sixth IEEE international Conference on Neural Networks*, Vol. 3, 1997, pp. 1738 – 1743..
- [9] R. Hafner, and M. Riedmiller, Reinforcement Learning on a Omnidirectional Mobile Robot, *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, Las Vegas, 2003, pp. 418 – 423.
- [10] S. Mahadevan, S. and J. Connell, “Automatic Programming of Behavior Based using Reinforcement Learning”, *Proceeding of the Eighth International Workshop on Machine Learning*, 1991, pp. 328-332.
- [11] W.D. Smart, and L.P. Kaelbling, Effective Reinforcement Learning for Mobile Robots, *Proceeding*

of International Conference on Robotics and Automation, 2002.

[12] M. Asadpour, and R. Siegwart, Compact Q-Learning for Micro-robots with Processing Constraints, *Journal of Robotics and Autonomous Systems*, vol. 48, no. 1, 2004, pp. 49-61.

[13] P. Ritthipravat, T. Maneewarn, D. Laowattana, and J. Wyatt, A Modified Approach to Fuzzy Q-Learning for Mobile Robots, *Proceedings of 2004 IEEE International Conference on Systems, Man and Cybernetics*, Vol. 3, 2004, pp. 2350 – 2356.

[14] R. Brooks, A Robust Layered Control System For a Mobile Robot, *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, 1986, pp. 14 – 23.