Hexagon-Based Q-Learning for Object Search with Multiple Robots

Hyun-Chang Yang, Ho-Duck Kim, and Kwee-Bo Sim

School of Electrical and Electronics Engineering, Chung-Ang University
221, Heukseok-Dong, Dongjak-Gu, Seoul 156-756, Korea
e-mail : hoduck@wm.cau.ac.kr, kbsim@cau.ac.kr

Abstract

This paper presents the hexagon-based Q-leaning for object search with multiple robots. We organized an experimental environment with five small mobile robots, obstacles, and an object. Then we sent the robots to a hallway, where some obstacles were lying about, to search for a hidden object. In experiment, we used three control algorithms: a random search, an area-based action making (ABAM) process to determine the next action of the robots, and hexagon-based Q-learning to enhance the area-based action making process.

Keywords **:** Hexagon-based Q-Learning, Multiple robots, Area-Based Action Making, Markovian

1. Introduction

Nowadays, robots are performing human's work in dangerous field, such as rescue jobs at fire-destroyed building or at gas contaminated sites; information retrieval from deep seas or from space; and weather analysis at extremely cold areas like Antarctica. Sometimes, multiple robots are especially needed to penetrate into hard-to-access areas, such as underground insect nests, to collect more reliable and solid data.

Multiple robot control has received much attention to offer a new way of controlling multiple agents more flexibly and robustly. Ogasawara used distributed autonomous robotic systems to control multiple robots that transport a large object[1]. In this paper, we propose an area-based action making (ABAM) process, which is the basis of hexagon-based Q-learning, to control multiple robots against collision and lead individual robot to search through its own trajectory.

Reinforcement learning allows an agent to actively decide an action policy based on explorations of its environment. During exploration of an uncertain state space with reward, an agent can learn what to do by continuous tracking of its state history and appropriately propagating rewards through the state space[2]. In our research, we focused on Q-learning as a reinforcement learning technique. Because Q-learning is a simple way to solve Markovian action problems with incomplete information and on the basis of the action-value function Q that maps state-action pairs to expected returns [3]. In addition to this simplicity, Q-learning can adopt to the real world situation. For example, the state space can be matched with the physical space of the real world. An action also can be regarded as physical robot movement. In this paper, we propose the hexagon-based Q-learning to enhance the area-based action making process so that the learning process can better adapt to real world situations.

The organization of this paper is as follows. In chapter 2, the area-based action making process is introduced. In chapter 3, hexagon-based Q-learning adaptation is presented. In chapter 4, experimental results from the application of three different searching methods to find the object are presented. In chapter 5, conclusions are presented.

2. Area-Based Action Making Process

Area-based action making (ABAM) process is a process that determines the next action of a robot. The reason why this process is referred to ABAM is that a robot recognizes surrounding not by distances, from itself to obstacle, but by areas around itself. The key idea of the ABAM process is to reduce the uncertainty of its surrounding. It is similar with the behavior-based direction change, to control the robots [4]. The robots recognize the shape of its surrounding, then take an action (turn and move forward) to where the widest space will be guaranteed. Consequently, each robot can avoid an obstacle and collision with other robots. Figure 1 depicts the different actions taken by distance-based action making (DBAM) and by ABAM in the same situation [5]. Our small mobile robot has the six emitter-detector infrared sensor pairs, which are placed at an angle of 60 degrees with one another to cover 360 degrees. The advantage of ABAM is illustrated by the

following example. Figure 2 presents the result of each action making process by DBAM and ABAM. In both case, the robot is surrounded by 4-obstacles. By DBAM, the robot will be confused because it perceives that there is no obstacle in the southeast direction, and then it will try to keep tracking to the southeast. Finally, it will get stuck between two obstacles. This scenario is shown in the left picture in Fig. 2. By ABAM, however, the robot will calculate the areas of its surrounding, and then it will recognize that an action to the northeast will guarantee the widest space. Therefore, the robot will change its direction to the northeast. This scenario is presented in the right picture in Fig. 2.

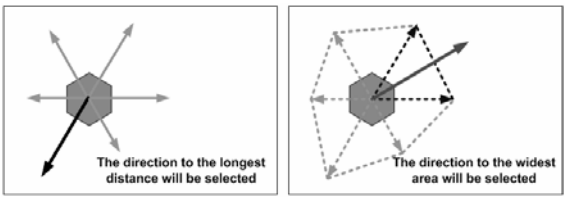

Fig. 1. The different actions will be taken by DBAM and by ABAM in the same situation

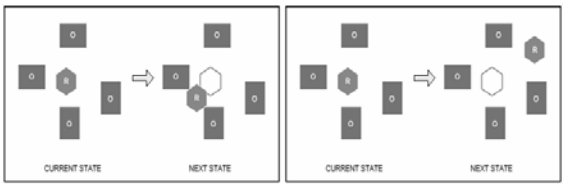

Fig. 2. An illustrative example of robot maneuvers by DBAM(left) and by ABAM(right)

In addition to the obstacle avoidance, ABAM also make the robots to search their own space [6]. This feature is advantageous when 2 or 3 robots meet at the same place. When they face each other, each robot will try to find more wide space. Consequently, the robot will change its direction to avoid the other robots and start to search in its own space again.

3. Hexagon-Based Q-Learning

Q-learning is a well-known algorithm for reinforcement learning. It leads the agent to acquire optimal control strategies from delayed rewards, even when the agent has no prior knowledge of the effects of its actions on the environment [7][8]. Figure 3 is an illustrative example to explain Q-learning algorithm more clearly. The 'R' stands for a robot or agent. The values upon the arrows are relevant $\hat{Q}$ values with the state transition. For example, the value $\hat{Q}(s_1, a_{right}) = 72$, where a right refers to the action that moves R to its right [8].

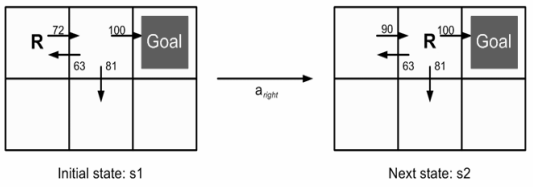

Fig. 3. An illustrative example of Q-learning

If the robot takes the action to the right, the value will be updated for this entry where r = 0, γ = 0.9 are predetermined values. The formula is presented below.

$$\begin{aligned} \hat{Q}(s,a) &\leftarrow r + \gamma \max_{a'} \hat{Q}(s',a') \\ &\leftarrow 0 + 0.9 \max\{63,81,100\} \\ &\leftarrow 90 \end{aligned} \quad (1)$$

The Q-learning for our robot system was adapted to enhance the ABAM process. The adaptation can be performed with a simple and easy modification, named hexagon-based Q-learning. Figure 4 is an illustrative example of hexagon-based Q-learning. In Fig. 4, intuitively, we know that the only thing that was changed is the shape of state space. We changed the shape of the space, from a square to a hexagon, so that the robot can recognize its surrounding by 6-areas. According to this adaptation, the robot takes an action to 6-direction and has 6-table entry $\hat{Q}$ value. In the left of Fig. 4, the robot is in the initial state. Now, if the robot decides that +60 degree guarantee the widest space after calculation of its 6-areas of surrounding, the action of the robot would be $a_{+60°}$. After the action is taken, if Area3 is the widest area, the value of $\hat{Q}(s_1, a_{+60°})$ will be updated by the formula (2) in the Q-learning algorithm as

$$\begin{aligned} \hat{Q}(s_1, a_{init}) &\leftarrow r + \gamma \max_{a'} \hat{Q}(s_2, a_{+60}) \\ &\leftarrow 0 + \gamma \max\{Area1, Area2, \cdots, Area6\} \\ &\leftarrow \gamma Area3 \end{aligned} \quad (2)$$

where 0 is the predetermined immediate reward. After the movement from the initial state to the 1st next state, immediate reward becomes the difference between the sum of total area before action is taken and the sum of total area after action is taken.

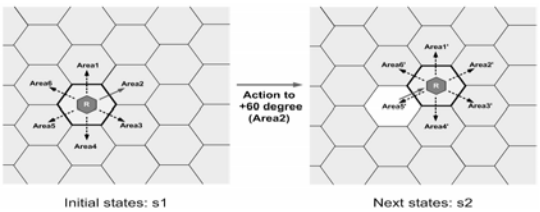

Fig. 4. Hexagon-Based Q-learning

Ultimately, the robot can determine its trajectory by learning this $\hat{Q}$ value. In the real world experiment, however, battery consumption is a problem. If the robot has to perform infinite iterations to complete task, total system will fail. Therefore, a system must be set up to cancel the former action and move back to the earlier state, when the former action causes any bad reward or result. The hexagon-based Q-learning algorithm is presented in Table 1.

Table 1. Hexagon-based Q-learning algorithm

For each s, a initialize the table entry $\widehat{Q}(s,a)$ to zero
Calculate each 6-areas at the current state s
Do until task is completed.
- Take an action a to the widest area
- Receive immediate reward r
- Observe the new state s'

If $\widehat{Q}(s',a)$ is greater or equal than $\widehat{Q}(s,a)$
- Update the table entry for $\widehat{Q}(s,a)$
- $s \leftarrow s'$

If $\widehat{Q}(s',a)$ is too less than $\widehat{Q}(s,a)$
- Move back to the previous state
- $s \leftarrow s$

4. Experiment Results

We performed experiments by using three different control methods: random search, ABAM, and enhanced ABAM by hexagon-based Q-learning. In the first part of this chapter, we introduce our self-made small mobile robot system. Then, we present experimental result with three different control methods.

4.1 Architecture of Small Mobile Robot

Our small mobile robot system consisted of four sub-parts and a main micro-controller part. The sub-parts were camera vision, sensor, motor, and Bluetooth communication module. Each sub-part had its own controller to perform its unique function more efficiently.

Figure 5 shows the appearance, anatomy, and functional block diagram of the robot. The main components of the robot are as follows. For the eye of the robot, Movicam II made by Kyosera is used. It is the CCD camera and its size is 30×47×29 mm. The robot has six infrared sensors, emitter and detector pairs, to measure the distance around itself. The detector is ST-1kla, high sensitivity NPN silicon phototransistors. NMB PG25L-024 stepping motor is used as the driving part. Its characteristics are the following: drive voltage-12V, drive method 2-2 phase and 0.495° step angle.

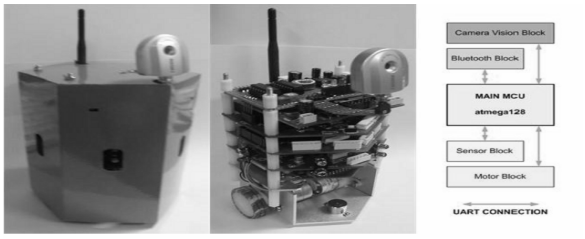

Fig. 5. Appearance (left), anatomy (center), and functional block diagram (right) of the robot

4.2 Experiments

The task of the robots is a follows: "Find the hidden object while tracking through an unknown hallway." We set up the color of the object as green and that of 5-robots as orange. The object was a stationary robot having the same shape. It was a located at a hidden place near the obstacle. The 5-robots, which try to search the object, recognize the object by the object's color and shape. The 5-robots will decide whether they have finished the task by detecting the object after each action is taken. First, we used the random search control method to find the hidden object. The main controller generated a random number and decided the next action corresponding to this number. Random search is not so strong method to control the robot efficiently. Therefore, random did not perform well. Moreover, it is very time and power consuming in the real world situation. In Fig. 6, the white arrow points out the object (same in Fig. 7 and Fig. 8). During random search, even though the robots are within a close distance to the search object, some robots failed to find the object.

Fig. 6. 5-robots are searching the object using random search

Second, we applied ABAM to the robots. With the feature of ABAM, the robots sense their environment by 6-infrared sensors and calculate 6-area with these values. When the calculation is done, each robot tries to move to where the widest area will be guaranteed. In our 2nd experiment, after the robots started to move, each robot spread out into the environment. Consequently, the ABAM performed better than random search. Figure 7 shows that the two robots, which is located the right side of the object, succeeded to complete the task. These two robots are designated by black arrow in Fig. 7.

Fig. 7. 5-robots are searching the object using ABAM

Finally, we adopted the hexagon-based Q-learning to ABAM as a modified control method. This method allowed the robots to reduce the probability of wrong judgment and compensated wrong judgment by reinforcement learning. Each robot tried to search its own area as in the 2nd experiment, however, it canceled the decided action if the action caused negative (or bad) immediate reward value. The search with hexagon-based Q-learning is presented in Fig. 8. The results of our experiment are presented in Fig. 9. With random search, one robot found the object at the 2nd trial and 6th trial, although these detections can be considered as just coincidence. Therefore, we can say the random search has no remarkable meaning. With ABAM, the robots performed better than with random search, with the average performance above 1 during the all trial. Finally, with the adaptation of hexagon-based Q-learning to ABAM, the results were remarkable.

Fig. 8. 5-robots are searching the object using hexagon-based Q-learning

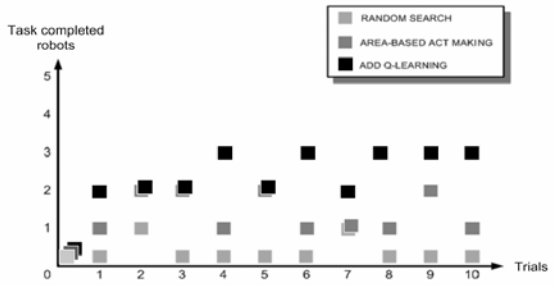

Fig. 9. Experimental result with 3 different control methods

5. Conclusion

In this paper, we presented the area-based action making process and hexagon-based Q-learning to search the object, hidden in unknown space, for 5 of our self-made small mobile robots. The experimental results from the application of the three different control methods in the same environmental situations were presented. The area-based action making process and hexagon-based Q-learning can be a new way for robot to search an object in unknown space. This algorithm also makes the agents avoid obstacles during their search. In our research, first, we need to clarify the problem of accessing to the object.

This means that if multiple robots are to carry out a task such as object transporting or block stacking, the robots need to recognize the object then approach to it. Therefore, we need to develop the robust accessing algorithm. Naturally, some grippers need to be attached to both sides of the robot. Second, our robot systems should be improved so that the main part and the sub-parts adhere more strongly. In addition, stronger complex algorithms such as Bayesian learning or TD(λ) method should be adapted. Third, a self-organizing Bluetooth communication network should be built so that robots can communicate with each other robustly even if one or more robots are lost. Finally, the total system should be refined.

Acknowledgment:

This research was supported by the Development of Social Secure Robot using Group Technologies of Growth Dynamics Technology Development Project by Ministry of Commerce, Industry and Energy, Korea.

References

[1] G. Ogasawara, T. Omata, T. Sato, "Multiple movers using distributed, decision-theoretic control," *Proc. of Japan-USA Symposium on Flexible Automation 1*, pp. 623-630, 1992.

[2] D. Ballard, "*An Introduction to Natural Computation*," The MIT Press Cambridge, 1997.

[3] J. Jang, C. Sun, E. Mizutani, "Neuro-Fuzzy and Soft Computing," *Prentice-Hall New Jersey*, 1997.

[4] W. Ashley, T. Balch, "Value-based observation with robot teams (VBORT) for dynamic targets," *Proc. of Int. Conf. on Intelligent Robots and Systems*, 2003.

[5] J. B. Park, B. H. Lee, and M. S. Kim, "Remote Control of a Mobile Robot Using Distance-Based Reflective Force," *Proc. of IEEE Int. Conf. on Robotics and Automation 3*, pp. 3415-3420, 2003.

[6] P. Ögren, N. E. Leonard, "Obstacle Avoidance in Formation," *Proc. of IEEE Int. Conf. on Robotics and Automation 2*, pp. 2492-2497, 2003.

[7] T. Mitchell, "Machine Learning," *McGraw-Hill Singapore*, 1997.

[8] C. Clausen, H. Wechsler, "Quad-Q-Learning," *IEEE Trans. on Neural Network 11*, pp. 279-294, 2000.