A study of reinforcement learning with knowledge sharing for distributed autonomous system

Kazuyuki Ito
Okayama University

Yoshiaki Imoto
Okayama University

Akio Gofuku
Okayama University

Mitsuo Takeshita
Okayama University

Proceedings 2003 IEEE International Symposium on
Computational Intelligence in Robotics and Automation
July 16-20, 2003, Kobe, Japan

A study of reinforcement learning with knowledge sharing for distributed autonomous system

Kazuyuki Ito
Dept. of Systems Engineering
Fucluty of Engineering
Okayama University
www.usm.sys.okayama-u.ac.jp/~kazuyuki

Yoshiaki Imoto
Dept. of Systems Engineering
Fucluty of Engineering
Okayama University

Akio Gofuku
Dept. of Systems Engineering
Fucluty of Engineering
Okayama University

Mitsuo Takeshita
Dept. of Systems Engineering
Fucluty of Engineering
Okayama University

Abstract

Reinforcement learning is one of effective controller for autonomous robots. Because it does not need priori knowledge and behaviors to complete given tasks are obtained automatically by repeating trial and error. However a large number of trials are required to realize complex tasks. So the task that can be obtained using the real robot is restricted to simple ones.

Considering these points, various methods that improve the learning cost of reinforcement learning had been proposed.

In the method that uses priori knowledge, the methods lose the autonomy that is most important feature of reinforcement learning in applying it to the robots.

In the Dyna-Q, that is one of simple and effective reinforcement learning architecture integrating online planning, a model of environment is learned from real experience and by utilizing the model to learn, the learning time is decreased. In this architecture, the autonomy is held, however the model depends on the task, so acquired knowledge of environment can not be reused to other tasks.

In the real world, human beings can learn various behaviors to complete complex tasks without priori knowledge of the tasks. We can try to realize the task in our image without moving our body. After the training in the image, by trying to the real environment, we save time to learn. It means that we have model of environment and we utilize the model to learn. We consider that the key ability that makes the learning process faster is construction of environment model and utilization of it.

In this paper, we have proposed a method to obtain an environment model that is independent of the task. And by utilizing the model, we have decreased learning time. We consider distributed autonomous agents, and we show that the environment model is constructed quickly by sharing the experience of each agent, even when each agent has own independent task. To demonstrate the effectiveness of the proposed method, we have applied the method to the Q-learning and simulations of a puddle world are carried out. As a result effective behaviors have been obtained quickly.

1 Introduction

Reinforcement learning [15, 12, 9, 8, 7, 10, 1, 11] is one of effective controller for autonomous robots [6, 13, 16, 18, 14, 3, 4, 5]. Because it does not need priori knowledge and behaviors to complete given tasks are obtained automatically by repeating trial and error. However a large number of trials are required to realize complex tasks. So the task that can be obtained using the real robot is restricted to simple ones.

Considering these points, various methods that improve the learning cost of reinforcement learning had been proposed. We can divide them into two categories. One is the methods that utilize priori knowledge and the other is the method that utilized experience that is given during the learning.

In the former, the cost to learn is reduced by adding some sub-rewards or dividing given task into some subtasks. Though the learning time is decreased extremely, the methods lose the autonomy that is most important feature of reinforcement learning in applying it to the robots.

In the latter, improvement of the efficiency of the learning is attempted. In the Dyna-Q, that is one of simple and effective reinforcement learning architecture integrating online planning, a model of environment is learned from real experience. By utilizing the model to learn, the learning time is decreased. In this architecture, the autonomy is held, however the model depends on the task, so acquired knowledge of environment can not be reused to other tasks.

In the real world, human beings can learn various behaviors to complete complex tasks without priori knowledge of the tasks. We can try to realize the task in our image without moving our body. After the training in the image, by trying to the real environment, we save time to learn. It means that we have model of environment and we utilize the model to learn. The model of environment is obtained during the learning of various tasks and some of it is given by other persons. We consider that the key ability that makes the learning process faster is construction of environment model and utilization of it.

In the previous works of reinforcement learning, the model of environment depends on the task. So the model can not be used to learn other tasks.

In this paper, we propose a method to obtain an en-

0-7803-7866-0/03/$17.00 ©2003 IEEE

vironment model that is independent of the task, and we decrease learning time. We consider distributed autonomous agents, and we show that the environment model is constructed quickly by sharing the experience of each agent, even when each agent has own independent task. To demonstrate the effectiveness of the proposed method, we apply the method to the Q-learning and simulations of a puddle world are carried out.

2 Q-learning

Q-learning is a reinforcement learning algorithm proposed by Watkins [17]. In the Q-learning, we assume that the world constitutes a Markov decision process. The agent has the Q-value that is composed of the pair of states s and actions a. By repeating the trial, the Q-value is renewed using following rule.

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

where s is the state, a is the action, r is the reward, α is the learning rate and γ is the discount rate. By the infinite iteration of trials, the optimal policy is acquired and it can run along the optimal trajectories by selecting the action of maximum Q-value at each time.

3 Problem domain

We consider distributed autonomous agents. Every agent has the same body and they live in the identical environment. The environment consists of Markov Decision Process. The transition probability is the same without regard to differences of the agents. Each agent has own task that differ from other agents' , and it tries to accomplish the task independently. There are some kinds of reward in the environment, and the agents have ability to distinguish the difference of the reward.

The problem we solve in this paper is how to reduce the learning cost of every agent in that world.

4 Proposed Algorithm

4.1 Outline

We propose a method to construct the model of environment by utilizing experiences of agents that have own tasks.

The idea to realize the method is written as follows. We distinguish the rewards by the cause of that. Each agent has own model of transportation probability and model of the expected value of the next reward for each kind of reward. By utilizing other agent's experiences to construct own model, the construction becomes quicker. However every agent has different task, so the model of expected rewards can not be always utilized, so we pay attention to common rewards that independent of the task they depend on the environment. The agents utilize the common rewards for constructing the own model.

By trying the task in the model, the trial in the real world can be reduced, so the time to accomplish learning process can be reduced. In this paper, we define the imaginary trial as the trial in the model and define the real trial as the trail in the real world.

4.2 Estimation of environment model

In this subsection, we describe the method to estimate environment model. We define estimated value of the transition probabilities for agent i in time t as $\hat{P}(i,t,s,a,s')$ and we also define the estimated expected value of the next reward $\hat{R}_k(i,t,s,a,s')$. Where s is current state, a is current action, s' is next state and k is kinds of reward.

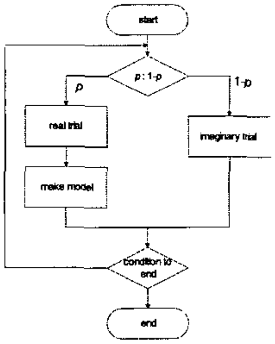

Figure 1: Flowchart

At first we explain the method to calculate $\hat{P}(i,t,s,a,s')$.

In the real trial, we calculate the equation below for each trial.

$$N(i,t+1,s,a,s') = N(i,t,s,a,s') + 1 \quad (2)$$

where $N(i,t,s,a,s')$ means the number of times that an agent i has visited to s' from s by using a. The $\hat{P}(i,t,s,a,s')$ can be given by the equation (3).

$$\hat{P}(i,t+1,s,a,s') = \frac{N(i,t,s,a,s')}{\sum_{s'' \in s'} N(i,t,s,a,s'')} \quad (3)$$

Next we explain the method to calculate $\hat{R}_k(i,t,s,a,s')$. To calculate $\hat{R}_k(i,t,s,a,s')$, we employ equation (4).

$$\hat{R}_k(i,t+1,s,a,s') = \eta\hat{R}_k(i,t,s,a,s') + (1-\eta)r_k(i,t,s,a,s') \quad (4)$$

where $r_k(i,t,s,a,s')$ is the real reward that is given by the environment and $\eta (0 < \eta < 1)$ is a coefficient that determines the renewal rate.

4.3 Shearing of experiences

In this subsection, we explain the method to share experiences with other agent. When the agent i and j share their experience, the models of the environment are calculated by equation (5) to (9) and (10) to (14) respectively.

$$N(i,t+1,s,a,s') = N(i,t,s,a,s') + 1 \quad (5)$$

$$N(i,t+1,s,a,s') = N(j,t,s,a,s') + 1 \quad (6)$$

$$\hat{P}(i,t+1,s,a,s') = \frac{N(i,t,s,a,s')}{\sum_{s'' \in s'} N(i,t,s,a,s'')} \quad (7)$$

$$\hat{R}_k(i,t+1,s,a,s') = \eta\hat{R}_k(i,t,s,a,s') + (1-\eta)r_k(i,t,s,a) \quad (8)$$

$$\widehat{R_{k'}}(i,t+1,s,a,s') = \eta\widehat{R_{k'}}(i,t,s,a,s') + (1-\eta)r_{k'}(j,t,s,a) \quad (9)$$

$$N(j,t+1,s,a,s') = N(j,t,s,a,s') + 1 \quad (10)$$

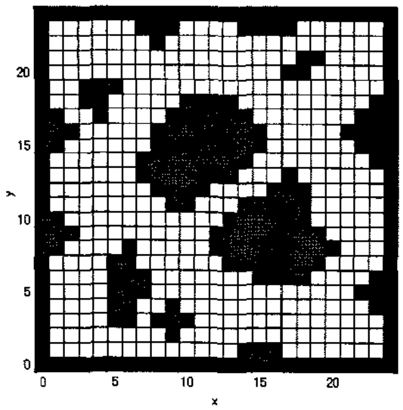

Figure 2: Puddle world

$$N(j,t+1,s,a,s') = N(i,t,s,a,s') + 1 \qquad (11)$$

$$\hat{P}(j,t+1,s,a,s') = \frac{N(j,t,s,a,s')}{\sum_{s''\in s'} N(j,t,s,a,s'')} \qquad (12)$$

$$\hat{R}_k(j,t+1,s,a,s') = \eta\hat{R}_k(j,t,s,a,s')+(1-\eta)r_k(j,t,s,a) \qquad (13)$$

$$\hat{R_{k'}}(j,t+1,s,a,s') = \eta\hat{R_{k'}}(j,t,s,a,s')+(1-\eta)r_{k'}(i,t,s,a) \qquad (14)$$

where k' means common reward. When more than three agents share their experience each other, equations above are calculated against every agent.

5 Application to Q-learning

In this subsection, we realize a reinforcement learning algorithm by applying the proposed method to the Q-learning.

Fig.1 shows the flow chart of the algorithm. In this algorithm, Q-learning is carried out in the real world by the probability p, and in the imaginary world in the probability $1-p$.

At the learning of real world, the model of environment is created, and by utilizing the model to learn in the imaginary world, the cost to learn can be decreased, because the time to learn in the imaginary world is extremely shorter than the time to learn in the real world.

At the learning in the real world, Q-value is calculated by usual equation of Q-learning.

$$Q(i,t+1,s,a) = (1-\alpha)Q(i,t,s,a)$$
$$+\alpha\{r_k(i,t,s,a) + \gamma \max_{a'} Q(i,t,s',a')\} \qquad (15)$$

Then model of environment is created as written in subsection 4.2 and 4.3.

In the learning in the imaginary world, Q-value is calculated using the model of environment as follows.

$$Q(i,t+1,s,a) = (1-\alpha)Q(i,t,s,a)$$
$$+\alpha\{\sum_{k''\in k}(\hat{R}_{k''}(i,t,s,a,s'))+\gamma \max_{a'} Q(i,t,s',a')\} \qquad (16)$$

Table 1: Start and Goal of agents

	Start(x,y)	Goal(x,y)
agent1	(2,2)	(22,22)
agent2	(22,22)	(2,2)
agent3	(2,22)	(22,2)
agent4	(22,2)	(2,22)
agent5	(22,12)	(2,12)
agent6	(2,12)	(22,12)
agent7	(12,2)	(12,22)
agent8	(12,22)	(12,2)

Table 2: Group of agent that share experience

	agent
case1	Conventional Q-learning (not shar)
case2	Dyna-Q (not shar)
case3	Proposed method (not shar)
case4	(1,2) (3,4) (5,6) (7,8)
case5	(1,2,3,4) (5,6,7,8)
case6	(1,2,3,4,5,6,7,8)

6 Simulation

In this section, we demonstrate the effectiveness of the proposed method by applying it to distributed autonomous agents in a puddle world. We consider 25×25 puddle world written in Fig.2. The black grids mean puddles. There are 8 agents and every agent has own goal position written in Table 1. Every agent has four actions (go to forward grid, go to back grid, go to right grid and go to left grid). The environment consists of probabilistic transition. With probability of 80%, the agent goes along the selected action. With probability of 10%, the agent goes to one side of selected action as written in Fig.3. The aim of the task is to reach own goal with avoiding the puddles. There are 2 kinds of rewards, one is positive rewards that are given at the goal position and the other is negative reward that is given at the puddles. In this task, the positive reward is dependent on the agent, and the negative reward is independent of the agent. So the negative reward can be considered as common rewards. Parameters of this simulation are written in Table 3

6.1 Effect of the proposed method

In this subsection, we consider the effect of the proposed method written in section 4 and 5. We compare the proposed method to conventional Q-learning and Dyna-Q. We consider 6 cases. First one is original Q-learning. Every agent has own Q-table and tries to learn independently. Second one is Dyna-Q. Every agent has own Q-table and tries to learn independently too. But in the Dyana-Q, every agent can utilize own experience by using it as the environment model. In this simulation, the number of calculation of Q-value using the model is 10 per that of real world. Third to sixth one is the proposed algorithm. In Case3, every agent tries to learn independently. In Case4, every two agents share their experiences respectively, in Case5, four agents share their experiences, and in Case6, all agents share their experiences. Details are written in Table 2.

Fig.4 and Fig.5 show the simulation result of Case1 to Case3 and Case3 to Case6 respectively. x axis means

Table 4: Start and Goal of agents in Case7 to Case9

	Number of agents	start	goal
case7	2	1(2,2) 2(2,2)	1(22,22) 2(22,22)
case8	2	1(2,2) 2(22,22)	1(22,22) 2(2,2)
case9	4	1(2,2) 2(22,22) 3(2,22) 4(22,2)	1(22,22) 2(2,2) 3(22,2) 4(2,22)

Table 3: parameter

	parameter
Learning rate	α=0.5
Discount rate	γ=0.95
Model learning rate	η=0.9
ϵ-greedy	ϵ=0.1
Reward	goal : 200 puddle : -150(25%) -100(50%) -50 (25%)

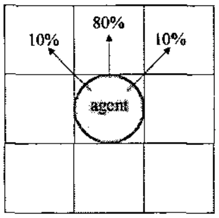

Figure 3: Probability of transit

the number of real trials and y axis means the gains. In this simulation, gain is calculated by dividing sum of rewards by the total steps of each trial. Fig.7 shows the number of times that agent1 has visited to each grid in Case3. x-axis and y-axis mean the horizontal position and vertical position of Fig.2, respectively. From Fig.7 , we can find that the route along upper left is obtained.

At first we compare the proposed method to conventional method. From Fig.4, we can find that the convergence of proposed method is quicker than conventional Q-learning and stable level of gain is equals to conventional Q-learning. It means that the acquired environment model is effective to improve learning speed and moreover the acquired policy is optimal.

In the Dyna-Q, the convergence is quick but stable level is not high, because the environment model of Dyna-Q does not take probabilistic transition into consideration.

We can conclude that the proposed algorithm is more effective than conventional methods even when the agents learn independently.

Next we consider the effect of sharing of experience. Fig.6 shows the accumulation of steps in imaginary trial. Fig.8 shows the sum of real trials of 8 agents in the stable point of Fig.4 and Fig.5. From Fig.5 to Fig.8, we can find that the convergence becomes quicker by sharing experience. However, it does not mean that agents accomplished the one task by cooperation. Each agent has own independent task and own independent Q-table. Each agent obtain own policy to complete the own task. The simulation result means that the learning speed of distributed autonomous system is improved by sharing the common experience even when each agent has different tasks. It quite differs from conventional approaches that realize one task by cooperative agents and it means our proposed approach is original and effective.

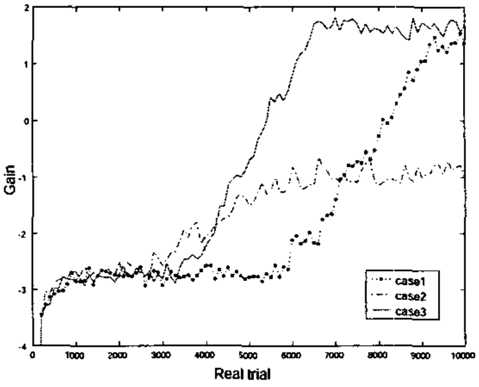

Figure 4: Simulation result (Case1 to Case3)

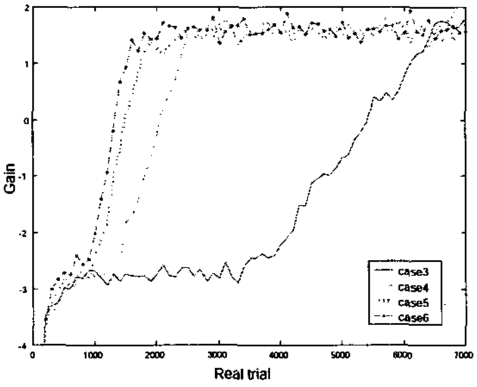

Figure 5: Simulation result (Case3 to Case6)

6.2 Created model of environment

In this subsection, we consider the created model of the environment. We employ 2 or 4 agents, and consider the effect of differences of sphere of each agent by changing the start positions and the goal position. Table 4 and Fig.9 to Fig.11 shows the positions of start and goal.

Fig.12 and Fig.13 show simulation result. Figures in Fig.13 show the number of times that any agents have visited to each grid at each points that is written in Fig.12.

From Fig.13, we can find that In Case7, the model

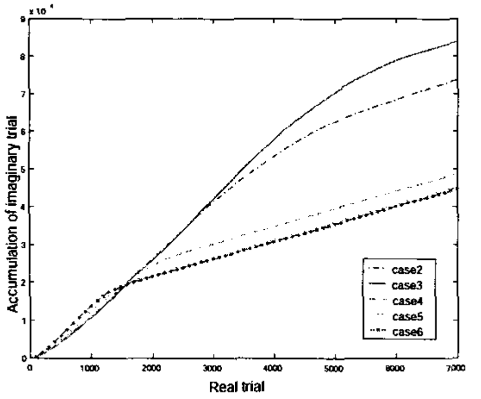

Figure 6: Accumulation of steps in imaginary trial

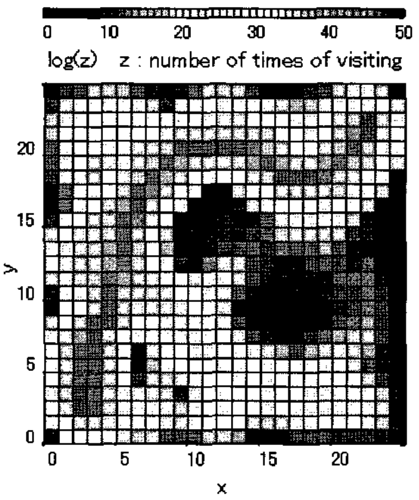

Figure 7: Number of Times of Visiting

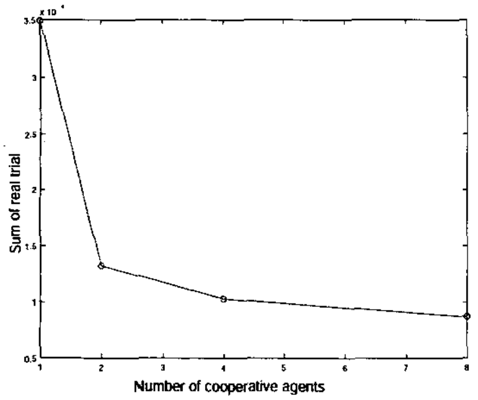

Figure 8: Sum of real trials to realize task

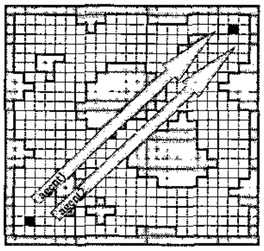

Figure 9: Case7

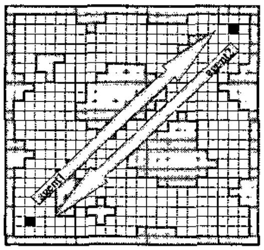

Figure 10: Case8

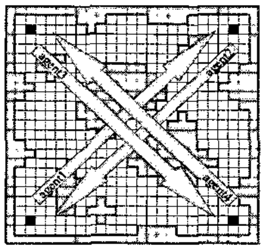

Figure 11: Case9

has been created from one side, in Case8, the model has been created from two opposite side and in Case9, the model has been created from four opposite side. We can consider that by covering lack of own experience each other, the model is created quickly. And the result shown in Fig.12 supports this consideration.

Finally, we confirm our research field again. In this simulation, we had employed the puddle world as an example, so the environment model means the map of puddle world. However, our proposed algorithm is applicable for general reinforcement learning problems. So the proposed method differs from the research of how to create the map of environment[2] and we have treated more general learning problems.

7 Conclusion

In this paper, we have proposed new reinforcement learning method for distributed autonomous system. In the method, agents share their experiences each other to create the environment model and by utilizing the model to learn, cost to learn has been reduced.

To demonstrate the effectiveness of the proposed approach, simulations of puddle world have been carried out. As a result, the learning time has been reduced. And especially, when the number of agents increased, the effect of the proposed algorithm has been increased.

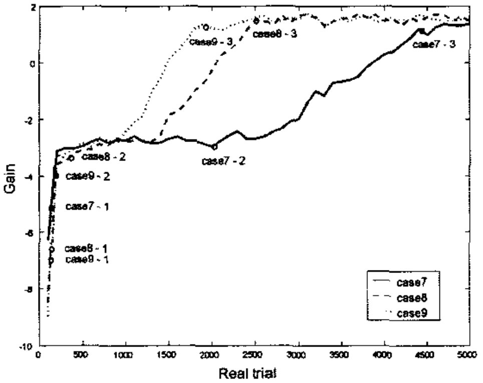

Figure 12: Simulation result (Case 7 to Case9)

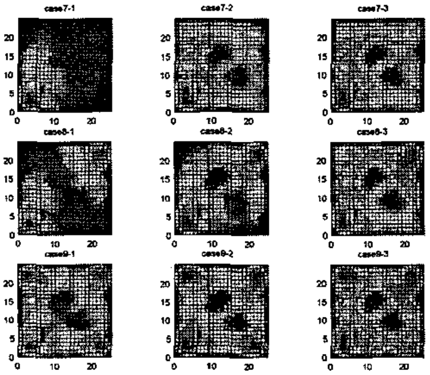

Figure 13: Number of Times of Visiting

We can conclude that the proposed algorithm is effective for distributed autonomous systems that share the identical world.

References

[1] K. Doya, H. Kimura, and M. Kawato. Neural mechanisms of learning and control. *IEEE Control Syatems Magazine*, 21(4):42–44, 2001.

[2] K. Ishioka, K. Hiraki, and Y. Anzai. Marsha: Design and implementation of map acquiring system for multiple autonoumous mobile robots. *Journal of the Robotics Society of Japan*, 12(6):846–856(in Japanese), 1994.

[3] K. Ito and F. Matsuno. Application of reinforcement learning to hyper-redundant system -acquisition of locomotion pattern of snake like robot-. In *Proc. The Pacific Asian Conference on Intelligent Systems*, pages 65–70, 2001.

[4] K. Ito and F. Matsuno. A study of Q-learning: Dynamic structuring of exploration space based on genetic algorithm. *Transactions of the Japanese Society for Artificial Intelligence*, 16(6):510–520(in Japanese), 2001.

[5] K. Ito and F. Matsuno. Applying QDSEGA to the multi legged robot. *Transactions of the Japanese Society for Artificial Intelligence*, 17(4):363–372(in Japanese), 2002.

[6] K. Ito and F. Matsuno. A study of reinforcement learning for the robot with many degrees of freedom -acquisition of locomotion patterns for multi legged robot-. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 3392–3397, 2002.

[7] T. Jakkola, S. P. Singh, and M. I. Jordan. Reinforcement learning algorithm for partially observable markov decision problems. In *Advances of Neural Information Processing Systems 7*, pages 345-352, 1994.

[8] L. P. Kaelbling. Hierachical learning in stocastic domains. In *Proc. of the 10th International Conference on Machine Lerning*, pages 167–173, 1993.

[9] L. P. Kaelbling and M. L. Littman. Reinforcement learning : A survey. In *Journal of Artificial Intelligence Research 4*, pages 237–285, 1996.

[10] H. Lee, H. Kamaya, and K. Abe. Labeling Q-learning in hidden state environments. In *Proc. of the 6th Int. Symp. on Artificial life and Robotics*, pages 208-211, 2001.

[11] L. Lin. Scaling up reinforcement lerning for robot control. In *Proc. of the 10th Int. Conf. on Machine Lerning*, pages 182–189, 1993.

[12] A. McGovern, D. Precup, B. Ravindran, S. Singh, and R. S. Sutton. Hierarchical optimal control of mdps. In *Proceedings of the 10th Yale Workshop on Adaptive and Learning Systems*, pages 186–191, 1998.

[13] N. Ono and K. Fukumoto. A modular approach to multi-agent reinforcement learning. In *Distributed Artificial Intelligence Meets Machine Learning: Learning in Multi-agent Environments*, pages 25–39, 1997.

[14] S. Sagara, T. Danjoh, M. Tamura, and R. Katoh. Resolved motion rate control of a free-floating underwater robot with horizonal planar 2-link manipulator. In *Proc. of the 6th Int. Symp. on Artificial life and Robotics*, pages 113–116, 2001.

[15] R. S. Sutton. *Reinforcement Learning: An Introduction.* The MIT Press, 1998.

[16] M. Svinin, S. Ushio, K. Yamada, and K. Ueda. Emergent systems of motion patterns for locomotion robots. In *Proc. of Int. Workshop on Emergent Synthesis*, pages 119–126, 1999.

[17] C. J. C. H. Watkins and P. Dayan. Technical note Q-learning. *Machine Learning*, 8:279–292, 1992.

[18] K. Yamada, K. Ohkura, M. Svinin, and K. Ueda. Adaptive segmentation of the state space based on bayesian discrimination in reinforcement learning. In *Proc. of the 6th Int. Symp. on Artificial life and Robotics*, pages 168–171, 2001.