A study of reinforcement learning with knowledge sharing

Kazuyuki Ito
Okayama University

Yoshiaki Imoto
Okayama University

Hideaki Taguchi
Okayama University

Akio Gofuku
Okayama University

Proceedings of the 2004 IEEE
International Conference on Robotics and Biomimetics
August 22 - 26, 2004, Shenyang, China

A study of reinforcement learning with knowledge sharing

-Applications to real mobile robots-

Kazuyuki Ito, Yoshiaki Imoto, Hideaki Taguchi and Akio Gofuku
Department or Laboratory
Okayama University
3-1-1, Tushima-naka
Okayama, Japan
email: kazuyukiito@ieee.org

Abstract— **In this paper, we consider multi-agent system in which every agents have own tasks that differs each other. We propose a method that decreases learning time of reinforcement learning by using the model of environment. In the proposed algorithm, the model is created by sharing the experiences of agents each other. To demonstrate the effectiveness of the proposed method, simulations of a puddle world and experiments of a maze world have been carried out. As a result effective behaviors have been obtained quickly.**

I. Introduction

Recently, applications of robots to home use are getting much attention. For example cleaning-robot, pet-robot and so on. And reinforcement learning is also getting much attention because reinforcement learning can be one of effective controller for autonomous robots [1]-[18]. It does not need prior knowledge, and behaviors to complete given tasks are obtained automatically by repeating trial and error.

However, in the reinforcement learning, 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 use prior knowledge and the other is the method that utilized experiences that are given during the learning.

In the former, the cost of learning is reduced by adding some sub-rewards or dividing given task into some sub-tasks. 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 [1], 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 tasks, so acquired knowledge of environment can not be reused to other tasks.

And in the previous works of multi-agent systems, acquired policy is also dependent on task so acquired knowledge can not be reused to other tasks. When every agents have own tasks and the tasks are different each other, the agents could not share experiences each other. So, applications of previous works are restricted to some cooperative tasks [6], [17]. They can cooperate only when they treat the identical task.

In other words, in the previous works, agents can not reuse own experiences and can not share own experiences to other agents.

In the field in which activities of personal robots are wanted, many personal robots share the same environment, for example, room, house, garden and so on. And they have different tasks each other. So, if they can share and utilize own experiences each other independently of tasks, the learning becomes efficient.

In our previous work, we addressed this issue and proposed a reinforcement learning algorithm for multi-agent system in which agents can share their experiences each other even if they have different tasks[18]. And effectiveness of the proposed algorithm was demonstrated by simulations.

In this paper, we improve the proposed algorithm to apply it to real robots. To demonstrate the effectiveness of the proposed algorithm, simulations and experiments are carried out.

II. Problem domain

We consider multi agent system. Every agent has a 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.

0-7803-8641-8/04/$20.00 ©2004 IEEE

III. Proposed Algorithm

A. 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.

In this paper, 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.

B. 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.

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.

$$N(i,t+1,s,a,s') = N(i,t,s,a,s') + 1 \tag{1}$$

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 (2).

$$\hat{P}(i,t+1,s,a,s') = \frac{N(i,t,s,a,s')}{\sum_{s''\in s'} N(i,t,s,a,s'')} \tag{2}$$

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 (3).

$$\hat{R}_k(i,t+1,s,a,s') = \eta\hat{R}_k(i,t,s,a,s') + (1-\eta)r_k(i,t,s,a,s') \tag{3}$$

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.

C. Method to share 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 (4) to (8) and (9) to (13) respectively. For agent i, at first, we calculate N of agent i by using own experience.

when agent i went to s' from s by using a

$$N(i,t+1,s,a,s') = N(i,t,s,a,s') + 1 \tag{4}$$

Then calculate N of agent i by using experience of agent j.

when agent j went to s' from s by using a

$$N(i,t+1,s,a,s') = N(i,t,s,a,s') + 1 \tag{5}$$

We can obtain estimated transition probability by using equation below.

$$\hat{P}(i,t+1,s,a,s') = \frac{N(i,t,s,a,s')}{\sum_{s''\in s'} N(i,t,s,a,s'')} \tag{6}$$

Next we calculate $\hat{R}$ of agent i. At first, we employ experience of agent i.

when agent i went to s' from s by using a

for all $k' \in k_u$

(where k_u is the universal set of reward)

$$\hat{R}_{k'}(i,t+1,s,a,s') = \eta\hat{R}_{k'}(i,t,s,a,s') + (1-\eta)r_{k'}(i,t,s,a) \tag{7}$$

Next we employ experience of agent j.

when agent j went to s' from s by using a

for all $k' \in k_c$

(where k_c is the set of common reward)

$$\hat{R}_{k'}(i,t+1,s,a,s') = \eta\hat{R}_{k'}(i,t,s,a,s') + (1-\eta)r_{k'}(j,t,s,a) \tag{8}$$

$\hat{P}$ and $\hat{R}$ of agent j is calculated by the same way.

when agent j went to s' from s by using a

$$N(j,t+1,s,a,s') = N(j,t,s,a,s') + 1 \tag{9}$$

when agent i went to s' from s by using a

$$N(j,t+1,s,a,s') = N(j,t,s,a,s') + 1 \tag{10}$$

$$\hat{P}(j,t+1,s,a,s') = \frac{N(j,t,s,a,s')}{\sum_{s''\in s'} N(j,t,s,a,s'')} \tag{11}$$

when agent j went to s' from s by using a

for all $k' \in k_u$

$$\hat{R}_{k'}(j,t+1,s,a,s') = \eta\hat{R}_{k'}(j,t,s,a,s') + (1-\eta)r_{k'}(j,t,s,a) \tag{12}$$

when agent i went to s' from s by using a

for all $k' \in k_c$

$$\hat{R}_{k'}(j,t+1,s,a,s') = \eta\hat{R}_{k'}(j,t,s,a,s') + (1-\eta)r_{k'}(i,t,s,a) \tag{13}$$

When more than three agents share their experience each other, equations above are calculated against every agent.

IV. Application to Q-learning

A. Q-learning

Q-learning is a reinforcement learning algorithm proposed by Watkins [19]. In the Q-learning, we assume that the world constitutes a Markov decision process. The agent has Q-value that is composed of the pair of states s and actions a. By repeating trials, the Q-value is renewed using following rule.

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\{r(s,a) + \gamma \max_{a'} Q(s',a')\} \tag{14}$$

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

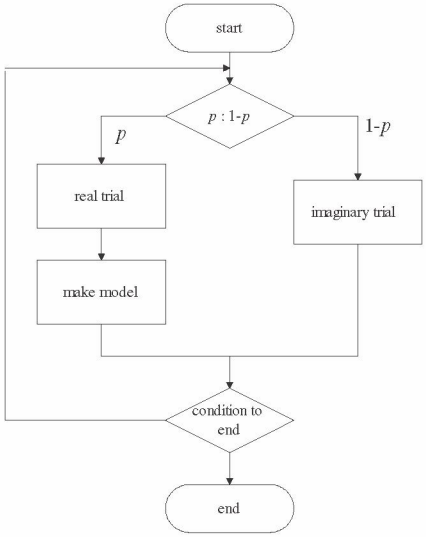

Fig. 1. Flowchart

TABLE I

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 II

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)

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.

B. 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 flowchart 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 $1-p$ $(0 < p < 1)$.

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 of agent i is calculated by usual equation of Q-learning.

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

Then model of environment is created as written in subsection III-B and III-C.

In the learning in the imaginary world, the transition of the state is determined based on the probability of $\hat{P}$ and Q-value of agent i is calculated using the model of environment as follows.

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

V. SIMULATION

In this section we pick up our previous results of simulation[18] to discuss effectiveness of sharing experiences by multi mobile robots, and in the next section we demonstrate that the effectiveness of the proposed algorithm is also valid in the real 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 I. The goal positions are different each other. So every agent has to learn own independent policy for each task. Therefore the Q-table of every agent is different each other and it is impossible to share their Q-table and their policy, and it is impossible to share the knowledge each other by the conventional method. So the tasks in the puddle world are valid to demonstrate the effectiveness and originality of the proposed method. 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 agents, and the negative reward is independent of the agent. So the negative reward can be considered as common rewards.

A. Effect of the proposed method

In this subsection, we consider the effect of the proposed method written in section III and IV. 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, ratio of number of real trial and number of imaginary trial is 1:10.

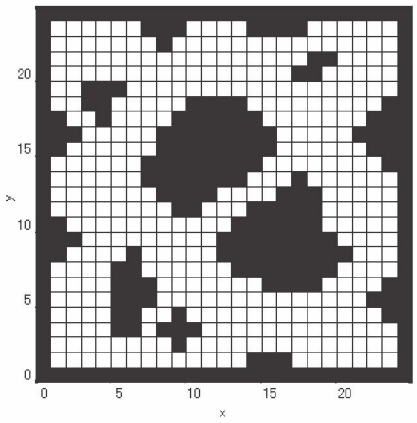

Fig. 2. Puddle world

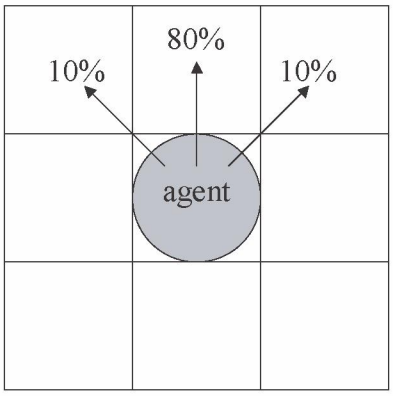

Fig. 3. Probability of transit

Third to sixth are the proposed algorithm. In Case3, every agent tries to learn independently. In Case4, we divide the 8 agents into 4 groups. Every group has 2 agents and they share their experiences with the other agent in the group. In Case5, we divide the 8 agents into 2 groups that have 4 agents. And the agents share their experiences with every other agent in their group. And in Case6, all agents share their experiences. Details are written in Table II.

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

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

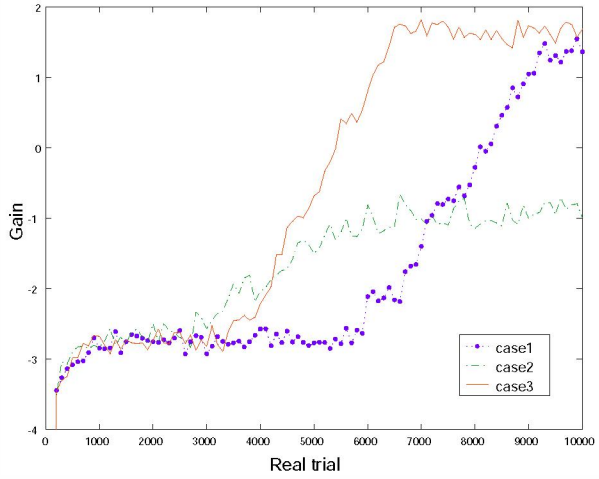

Fig. 4. Simulation result (Case1 to Case3)

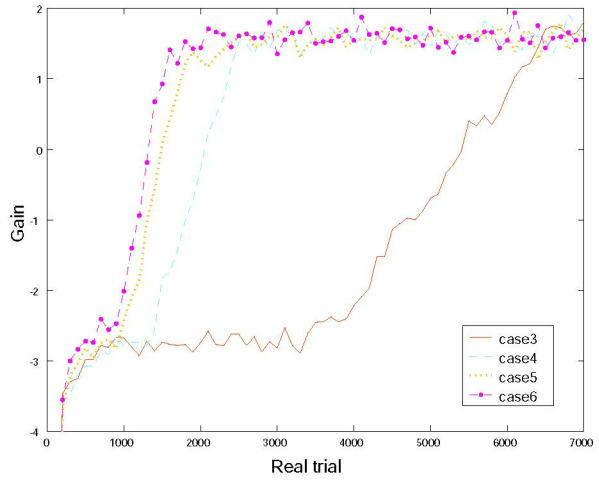

Fig. 5. Simulation result (Case3 to Case6)

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. From Fig.5, 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.

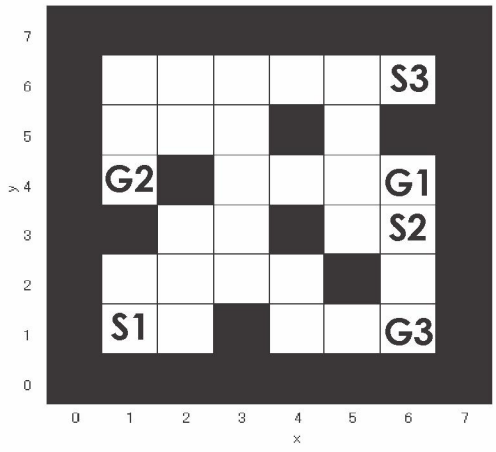

Fig. 6. Maze

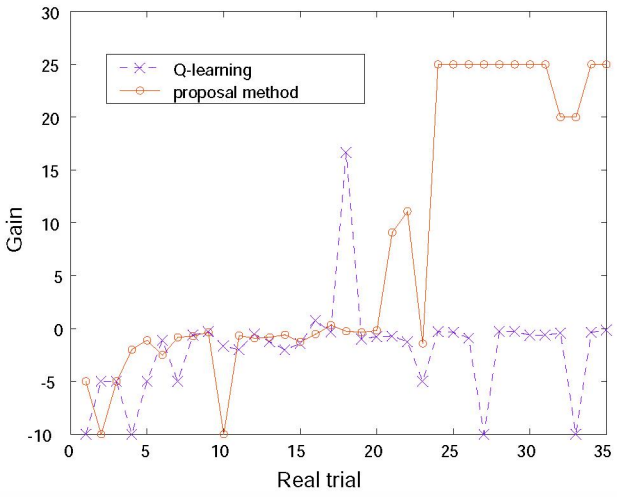

Fig. 7. Experiment result

VI. Application to Real Robots

In this section, we apply the proposed method to real robots in order to demonstrate validity in applying it to real world.

We consider a maze shown in Fig. 6, and 3-mobile robots. Where S1, S2, S3 are start positions of Robot1, Robot2, Robot3, respectively and G1, G2, G3 are goal positions of Robot1, Robot2, Robot3, respectively. An aim of task is to learn a policy to move to the goal positions with avoiding walls.

We employ Khepera as the mobile robots. It has 8 infrared proximity sensors and can detect walls. It also has 2 DC brushed servo motors with incremental encoders and can estimate own position by dead-reconing (In the learning, when errors of position become big enough, the position of the Khepera is adjusted to the center of the cell by human operator).

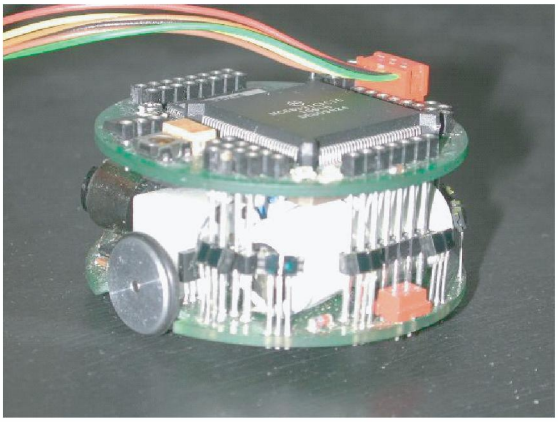

Fig. 8. Khepera

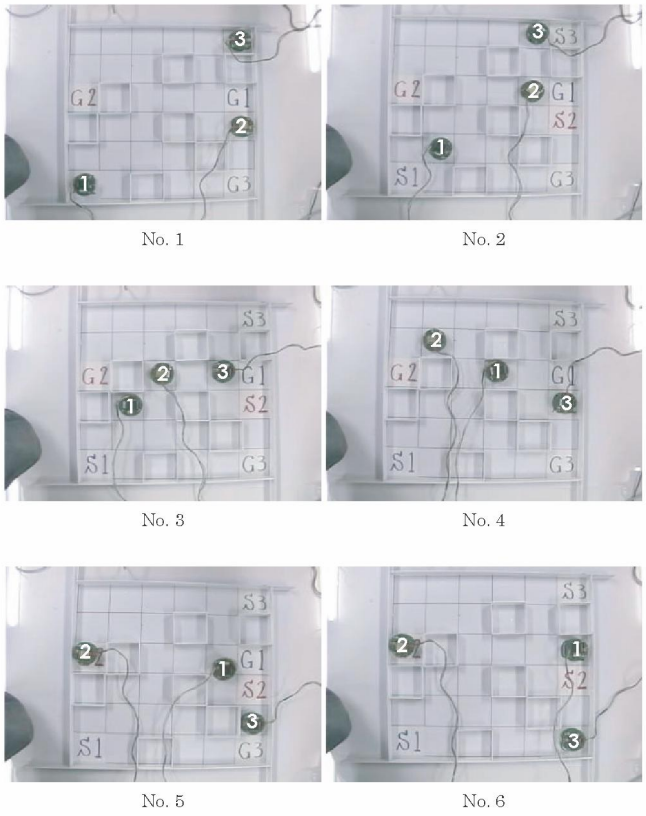

Fig. 9. Acquired behaviors

Configuration of the proposed algorithm is set as a same as the simulation written in Section V.

In addition to the proposed algorithm, conventional Q-learning has been carried out in order to be compared with the proposed algorithm.

Fig. 7 and Fig. 9 shows an example of experiment result. Fig. 7 is transition of gains of proposed method and conventional Q-learning. Fig. 9 is acquired behaviors of the proposed method. From Fig. 7, we can find that by using

the proposed method, effective behaviors are acquired quickly than conventional Q-learning. We have carried out another 3 experiments by using the proposed method, and in all experiments, effective behaviors were acquired quickly than conventional Q-learning.

We can conclude that the proposed method is effective not only for simulated imaginary world but also for real maze world.

VII. Conclusion

In this paper, we have improved our proposed reinforcement learning method to apply it to real mobile robots. In the proposed 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. And learning by real robots in real time became possible.

To demonstrate the effectiveness of the proposed method, experiments of maze world have been carried out. As a result, the learning time has been reduced and effective behaviors have been acquired.

We can conclude that the proposed algorithm is effective for multi agent system in the real world.

References

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

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

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

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] K. Ito, T. Kamegawa, and F. Matsuno. Extended QDSEGA for controlling real robots -acquisition of locomotion patterns for snake-like robot-. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 791–796, 2003.

[12] K. Ito and A. Gofuku. Hybrid autonomous control for heterogeneous multi-agent system, -combining of centralized reinforcement learning and distributed rule-based control-. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2500–2505, 2003.

[13] K. Ito and A. Gofuku. Emergence of adaptive behaviors by redundant robots, -robustness to changes environment and failures-. In *Proc. of Congress on Evolutionary Computation*, pages 2572–2579, 2003.

[14] K. Ito, A. Gofuku, and M. Takeshita. Necessity of body image in applying reinforcement learning to redundant robots. In *Proc. of the 2nd International Symposium on Adaptive Motion of Animals and Machines*, pages WeP–III–3, 2003.

[15] H. Kimura, T. Yamashita, and S.Kobaysahi. Reinforcement learning of walking behavior for a four-legged robot. In *Proc. of 40th IEEE Conference on Decision and Control*, pages 411–416, 2001.

[16] M. Asada, Y. Katoh, M. Ogino, and K. Hosoda. A humanoid approaches to the goal - reinforcement learning based on rhythmic walking parameters -. In *Proc. of the International Symposium of RoboCup2003*, pages CD–ROM, 2003.

[17] S. Kobayashi K. Miyazaki, S. Arai. A theory of profit sharing in multi-agent reinforcement learning. *Transactions of the Japanese Society for Artificial Intelligence*, 14(6):1156–1164(in Japanese), 1999.

[18] K. Ito, Y. Imoto, A. Gofuku, and M. Takeshita. A study of reinforcement learning with knowledge sharing for distributed autonomous system. In *Proc. of IEEE international Symposium on Computational Intelligence in Robotics and Automation*, pages 1120–1125, 2003.

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