The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Hardware design of autonomous snake-like robot for reinforcement learning based on environment
-Discussion of versatility on different tasks-

Kazuyuki Ito, Akihiro Takayama, Toshiharu Kobayashi

Abstract—**In this paper, we propose the design of a robot with a snake-like body based on a test environment. We explore the abstraction of state-action spaces for reinforcement learning. Additionally, we discuss the versatility of the proposed mechanism by showing that different tasks can be completed by simply changing the reward of the reinforcement learning. Finally, we mention the importance of a body design based on an environment by considering the concept of ecological niches.**

I. Introduction

Recently, robots that are able to adapt and operate in unknown environments have attracted a great deal of attention. The snake-like robot is one such robot, and there are expectations that it will eventually be deployed in real-world rescue operations and other similar tasks [1-8].

To control snake-like robots autonomously, various approaches have been proposed. Reinforcement learning is one of the more effective candidates because it allows a high degree of autonomy and versatility [9-18]. Using this method, the robot can learn through trial and error without a supervisor. By applying reinforcement-learning methods for robot control, the robot can adapt to an unknown environment autonomously.

Conventional reinforcement learning does have two serious drawbacks for practical use. The first drawback is the curse of dimensionality [7], and the second drawback is the inability to generalize between similar situations and actions. Because of the curse of dimensionality, the time required for learning is increased exponentially. It often becomes impossible to complete the learning process within a reasonable time limit. Moreover, if the learning process is completed on time, even trivial changes in the environment restart the learning process. This results from the inability to generalize acquired learning. In order to realize the necessary level of automation that the snake-like robot would require to operate in the real world, we have to solve these two problems without losing autonomy or versatility.

In our previous work, to solve these problems, we proposed the utilization of real-world properties for abstracting state-action spaces. For example, we designed the body of the snake-like robot by taking into account the dynamics of the real world. We showed the effectiveness of the proposed snake-like robot by demonstrating its ability to complete a task of going towards a light while avoiding obstacles [15]. Unfortunately, the previous experiment was limited to only one task, and the versatility of the proposed mechanism was not explored.

In this paper, we discuss the versatility of the proposed mechanism by showing that different tasks can be completed by simply changing the reward value of the reinforcement learning. We also discuss the importance of the robot's environment-based body design using the framework of an ecological niche.

II. Conventional Works

In conventional works, various approaches employing the previous knowledge of tasks have been proposed in order to reduce the learning time. One of the most effective and well-known approaches has been to divide a given task into smaller tasks and then learn each of these tasks independently. This approach is very effective and applicable for practical use, only if the task is fixed beforehand. This approach has problems with regard to autonomy and versatility; prior knowledge of the task is required for dividing the task. Both autonomous learning and versatility are lost due to the necessity of prior knowledge.

Our current study is focused on methods that require no prior knowledge and that allow a high degree of autonomy and versatility. Unfortunately, universal learning algorithms suffer from the curse of dimensionality; therefore, the robot's learning process is very time consuming.

Next, we consider approaches for generalization between similar situations and actions. In conventional works, various approaches for generalizing state-action spaces are proposed [12, 16-18]. Usually, the generalization process is carried out

Manuscript received February 28, 2009.

K. Ito is with Hosei University, 3-7-2, Kajino-cho, Koganei, Tokyo, 184-8584, Japan. (phone: +81-42-387-6093; fax: +81-42-387-6381; e-mail: ito@ hosei.ac.jp).

A. Takayama is with DAIHATSU MOTOR CO., LTD, 1-1, Daihatu-cho, Ikeda, Osaka, 563-8651, Japan.

T. Kobayashi is with Hosei University, 3-7-2, Kajino-cho, Koganei, Tokyo, 184-8584, Japan.

by combining reinforcement learning with other techniques. Some of these techniques include the use of neural networks, Fuzzy logic, or a stochastic approach. By combining techniques, the size of state-action spaces is decreased, and the load of reinforcement learning is reduced; however, these approaches still have problems.

On one hand, if we employ a learning algorithm, such as the addition of neural networks, greater time is required for generalization. As a result, the total learning time, including the time taken for reinforcement learning and generalization is still too long to be practical.

On the other hand, if we employ techniques in which a designer creates a state-action space by using the prior knowledge of a given task, it implies that versatility and autonomy are lost for previously stated reasons. Therefore, it is difficult to achieve generalization in a reasonable amount of time without losing versatility and autonomy.

III. Proposed Framework

A. Importance of design based on environment

As mentioned in section II, it is difficult to solve the curse of dimensionality and the inability to generalize without losing versatility and autonomy. However, higher organisms in the real world can learn in real time in spite of the fact that they learn by trial and error, and acquired policies can be applied generally. Using a conventional framework, we cannot explain why these organisms can learn general policies within such a short time.

We are forced to consider the fact that the conventional framework has two causes that cannot explain the behavior of organisms. One is the notion that the learning process can be expressed as an algorithm using mathematics. This would imply that almost all studies on reinforcement learning would belong to the information or software sciences. The other notion is that all prior knowledge results in a loss of versatility. Certainly, we can observe that employing the prior knowledge of tasks causes the loss of versatility; however, we also have to consider that employing the prior knowledge of an environment does not always cause a loss of versatility.

Generally, organisms live in various places, and we tend to think that they have a universal mechanism for adapting to their environment. We tend to try to describe this universal learning mechanism independently from the real environment; however, if we focus on one species, the environment in which the species lives is restricted. The adaptive mechanism of each species is dependent on its environment—it is not universal. Each species has a body suited to its environment, and it adapts itself by utilizing the properties of that environment. In biology, this is called an ecological niche.

Therefore, there is a possibility that higher organisms utilize the properties of their environment to improve their learning efficiency. Even if a body is dependent on its environment, its versatility and autonomy are not lost. This is because an organism lives in its environment and can always utilize properties of its environment.

We must consider that the problems caused by the curse of dimensionality and the lack of generality could be solved not through the improvement of the universal learning algorithm but by the improvement of the body so that it can utilize the properties of the environment.

Keeping this in mind, we designed the body of the snake-like robot by utilizing the prior knowledge of the environment it would be operating in. This will allow the robot to utilize the properties of the environment for abstracting state-action spaces. The objective of this experiment is to show that versatility for different tasks is maintained even if we design the robot by utilizing the prior knowledge of the environment.

B. Environment

Fig. 1 shows an example of the environment used. There is one light source that the robot must move towards, and many obstacles. The obstacles are wooden sticks, plastic pipes, and sponges. The obstacles are placed randomly, and the spaces between them are 1.5 times greater than the width of the robot. The robot may touch the obstacles, and it can use reactive force to move. Some obstacles are not sufficiently fixed in place; therefore, they can be moved when sufficient force is applied. The amount of force necessary to move these obstacles is unknown, and the robot does not have any information regarding the environment. The robot does not know the positions of the obstacles or which obstacles are fixed in place. In addition, the positions of the obstacles are changed after the robot has learned them.

There are many obstacles in the experiment environment, and the snake-like robot has many degrees of freedom; therefore, the curse of dimensionality is a serious problem. All acquired learning should be generally applicable for adapting the robot to changes in the environment.

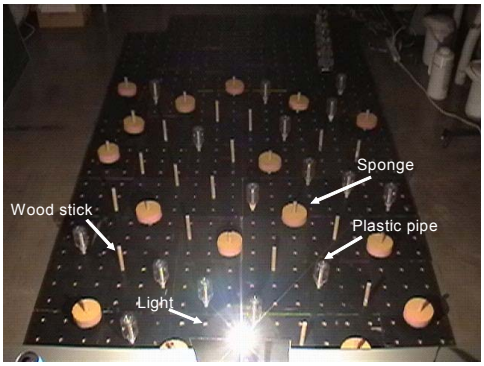

Fig. 1 Environment

C. *Snake-like robot*

Fig. 2 Proposed snake-like robot

Fig. 2 shows the snake-like robot we employed in this experiment. We have improved our previous robot [15]. This robot has the ability to turn in a small radius, and the range of the light sensor module is improved. With regard to the versatility of the proposed framework, the mechanism for abstracting state-action spaces is the same as that in our previous paper [15]. Details of the design are provided in the subsection below.

D. *Hardware design of body*

We employed the physical properties of the environment like a law of motion. Fig. 3 shows the passive mechanism of the proposed snake-like robot [15]. We did not use any actuators for joints; therefore, all the joints were passive. Additionally, we did not employ force sensors for the body or angle sensors for the joints. Two wires were embedded in the robot, and the length of these wires was controlled by a motor embedded into the rear end (Fig. 4). Rubbers that generate conservative force were embedded between all the links. The head of the robot is an acute-angled triangle, and it has a small free wheel attached to the tip to avoid reactive force from obstacle.

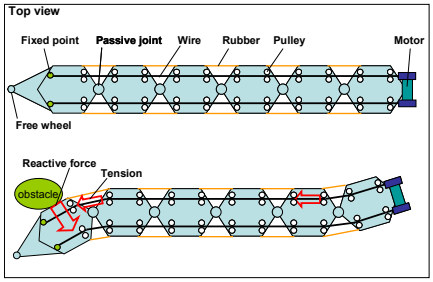

Fig. 3 Passive mechanism

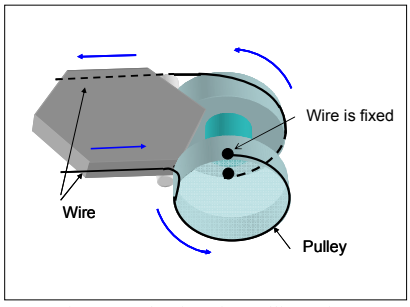

Fig. 4 Mechanism for pulling wires

Fig. 5 shows that when the robot contacts obstacles, some free joints are moved by the reactive force. The affected links pull the wire on one side, and the wire pulls the other links to compensate for the change in direction of the robot. When the robot has avoided the obstacle, the rubber pulls the affected links, returning them to their initial straight shape. If more than one obstacle hits the body, the resultant force of the reactive forces determines which links are moved.

As a result of the wire constraints and the dynamics of the rubbers, the robot moves as shown in Fig. 5. If the length of the wires on both sides is equal and the obstacles are set uniformly, the expected result is straight movement while avoiding the obstacles (Fig. 5 a)). If the length of the wire of the left side is shorter than the wire on the right side, the expected movement is a turn to the left (Fig. 5 b)).

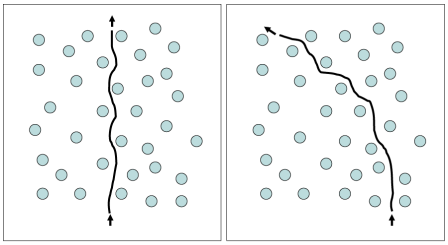

Fig. 5 Expected behavior

The movement of each joint is realized passively in this mechanism. Which joint should be moved, and to what extent is determined by the dynamics of the real world. The reactive force from the obstacles is used directly to move the joints. The state of each joint has an effect on the balance of wire constraints; therefore, these states are used to control the robot. These states do not have to be measured; the necessary calculation is processed by the dynamics of the real world.

The moving direction of the robot is determined by one equilibrium point. This point is the difference between the lengths of the wires. We can reduce the dimension of state of the body to only one by using the difference between the lengths of the wires as the state of the body. In this experiment, we used the angle of the tail motor, instead of the difference between the lengths of the wires as the state of the body; the result was the same.

We can also reduce the dimensions of the action space to only one. We define the action as to move the tail motor for changing the lengths of the wires.

E. *Hardware design for sensing*

We employed many CdS cells, and we used an equilibrium point of the output of the CdS cells as the direction. CdS cells are electrical cells that convert light intensity to electrical resistance. They have directional characteristics, and the front

side of a CdS provides the best response. Fig. 6 shows a CdS cell, a module of CdS cells, and their layout in the robot. We developed one module using 6 CdS cells, as shown in Fig. 6 b), and we embedded the module into every link of the robot, as shown Fig. 6 c).

We calculated Equation (1) in parallel using an analog electrical circuit, and we used the result as the direction. N is the number of links, and x is the direction ($0 < x < 5$). When light is put at the front side of the robot, x is 2.5, and if x is smaller than 2.5, it implies that the light is on the right side of the robot. If x is greater than 1.5, it implies that the light is on the left side of the robot.

$$x = \frac{\sum_{i=1}^{6}(1/R_{i2}) + 2\sum_{i=1}^{6}(1/R_{i3}) + 3\sum_{i=1}^{6}(1/R_{i4}) + 4\sum_{i=1}^{6}(1/R_{i5}) + 5\sum_{i=1}^{6}(1/R_{i6})}{\sum_{j=1}^{6}\sum_{i=1}^{6}(1/R_{ij})} \quad (1)$$

We used x to trigger the reward used in reinforcement learning.

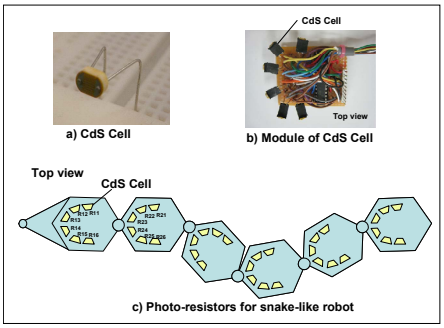

Fig. 6 Sensing system using CdS cells

IV. Experiment

A. Task

We created a task that uses a light source as a goal. We placed the light source at the center of the environment.

B. Setting of Q-learning

We used the example of typical Q-learning [10] for reinforcement. We did not modify the algorithm, because the aim of this experiment was not to improve learning algorithms, but to discuss the versatility of the proposed mechanism. Equation (2) shows Q-learning.

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\{r(s,a) + \gamma \max_{a'} Q(s',a')\} \quad (2)$$

s: state, a: action, r: reward, : learning rate, : discount rate

We set α as 0.2 and γ as 0.5. Action is selected using the ε-greedy method, and the probability of random selection is 0.1. The duration of one trial was 50 s, and calculations were performed using Equation (2) every 2.5 s.

Table 1 shows the state of the light direction. The resulting values are the outputs of Equation (1). Table 2 shows the state of the body. These values represent the angle of the tail motor. As shown in Tables 1 and 2, the number of dimension of the state space is two, and the total number of states is 35.

Table 3 lists the actions executed by the robot. The actions consisted of turning the tail motor, which pulled the wires.

TABLE I
State of the light direction s

state	direction
0	[0, 1.19]
1	(1.19, 1.45]
2	(1.45, 2.05]
3	(2.05, 2.95]
4	(2.95, 3.55]
5	(3.55, 3.80]
6	(3.80, 5]

TABLE II
Sate of the body

state	0	1	2	3	4
angel [deg]	−50	−25	0	25	50

TABLE III
Action

action	motion
0	turn tail motor −25 degree
1	hold tail motor
2	turn tail motor+25 degree

When the state of the light direction was 1 and the state of the body was 1, $r = 100$ was given as a reward. When the state of the light direction was 3, 4, 5, 6, or the light went out of the observable range, $r = -100$ was given as a penalty. The trial was then be halted, and the next trial started from the initial position.

C. Experiment

We conducted 50 trials using the environment shown in Fig. 7, and we applied acquired policies for the environment shown in Fig. 8. In Figs. 7 and 8, the broken line is an auxiliary line that expresses a circle; the robot cannot see the line.

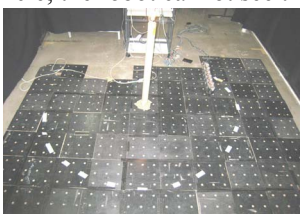

Fig. 7 Environment for learning

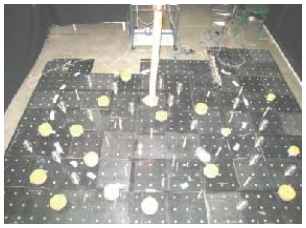

Fig. 8 Environment with obstacles

Fig. 9 shows acquired behavior and Fig. 10 shows the realized behavior using the acquired policy.

We found that the behavior of the robot revolving around the light was acquired in real time, and the learned behavior was generally applicable in a different environment. In particular, the results showed that the acquired policy from the simple environment was applicable to the complex environment, which is a very promising outcome.

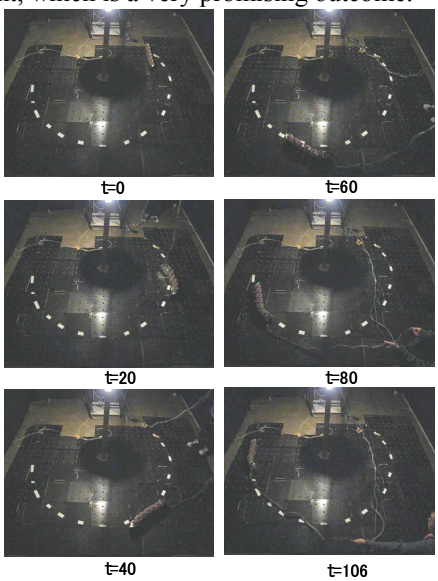

Fig. 9 Acquired behavior

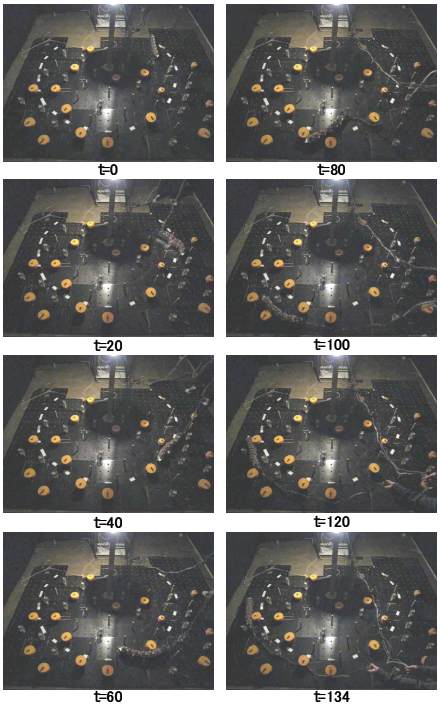

Fig. 10 Realized behavior using acquired policies

V. Discussion

First, we confirmed that the curse of dimensionality has been solved for our purposes. In the proposed mechanism, the size of dimension of the state space is only two and that of the action space is only one, though the snake-like robot has 6 links and there are many obstacles in the environment. We found that the size of the state-action space is significantly reduced, and the curse of dimensionality is overcome. The learning process of the experiment was conducted by a real robot, and it was completed in an acceptable time.

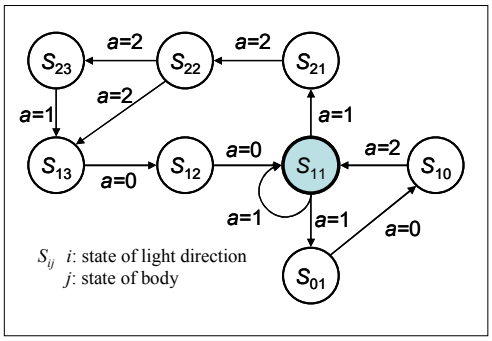

Fig. 11 State transition graph in an environment without obstacles

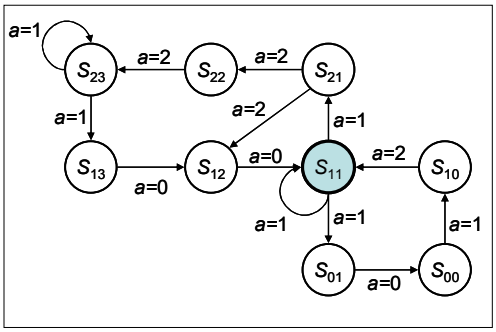

Fig. 12 State transition graph in an environment with obstacles

Fig. 11 shows the state transition graph of Fig. 9, and Fig. 12 shows state transition graph of Fig. 10. From these figures, we can observe that the state transition in both the graphs is very similar. This implies that a learning machine using Q-learning can adapt to different environments as if they were a similar environment, because of the functionality of the machine's body. A learning machine can apply acquired knowledge to different environments. In other words, by using the universal properties of the environment, the body of the robot can generalize changes of that environment. Therefore, we can confirm that the inability to generally apply learned principles is solved by the proposed mechanism.

Fig. 13 shows behavior going towards a light source that was acquired in our previous paper [15]. In that case, a reward was given when the light was on the front side of the robot. We found that by changing the reward, different behaviors were acquired. When designing the body of the proposed

mechanism, we utilized this prior knowledge. It is important to differentiate that the knowledge used was not knowledge of the task, but knowledge of the environment. This body can be applied to many different tasks, as long as the environment has similar properties to the one used in this experiment. That fact effectively demonstrates the versatility of the proposed robot.

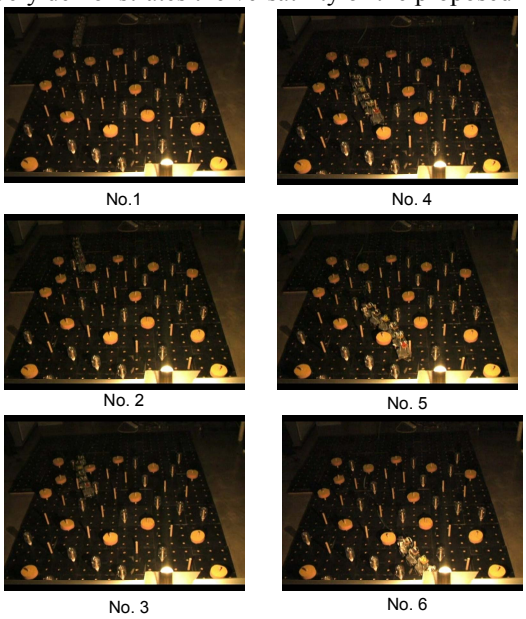

Fig. 13 Behavior going towards a light source

We can conclude that by designing the body around the properties of an environment, we can abstract state-action spaces. As a result, the problems of the curse of dimensionality and the inability to generally apply learned knowledge can be solved without losing versatility.

The important role of a designer must be considered when developing an autonomous robot controlled by reinforcement learning. The designer must assess what the universal and changeable properties of an environment are and understand how to implement these factors in the design of a mechanized body. These design considerations are what allow for the generalization between similar situations and actions.

VI. Conclusion

In this study, we explored the autonomous control of a snake-like robot that learns through reinforcement. We observed that the hardware design must take the environment into account. The universal properties of the environment allow the robot to generalize between similar actions and situations.

To demonstrate the validity of the proposed framework, an experiment was conducted, and we confirmed that by designing the body for utilizing the properties of the environment, we can abstract state-action space. As a result, the problems of the curse of dimensionality and the inability to generalize between similar situations and actions can be solved without losing versatility.

Our future efforts will focus on applying the proposed framework to more complex environments, such as a rubble-strewn rescue operation.

References

[1] Masayuki Arai, Toshio Takayama, Shigeo Hirose, Development of Souryu-III :Connected Crawler Vehicle for Inspection inside Narrow and Winding Spaces, Proc. of Int. Conf. Intelligent Robots and Systems, pp. 52-57, 2004

[2] K. L. Paap, T. Christaller, and F. Kirchner, A robot snake to inspect broken buildings, in Proc. of Int. Conf. Intelligent Robots and Systems, pp. 2079-2082, 2000.

[3] A. Wolf, H.B. Brown, R. Casciola, A. Costa, M. Schwerin, E. Shamas, H. Choset, A mobile hyper redundant mechanism for search and rescue tasks, in Proc. of Int. Conf. Intelligent Robots and Systems, pp. 2889-2895, 2003

[4] Tetsushi Kamegawa, Tatsuhiro Yamas, Hiroki Igarashit and Fumitoshi Matsuno, Development of The Snake-like Rescue Robot "KOHGA", Proc. of the 2004 IEEE International Conference on Robotlcs & Automation, pp. 5081-5086, 2004.

[5] Hiroya Yamada, Makoto Mori and Shigeo Hirose, Stabilization of the head of an undulating snake-like robot, Proc. of the 2007 International Conference on Intelligent Robots and Systems, pp.3566-3571, 2007.

[6] K. Ito and Y. Fukumori, Autonomous control of a snake-like robot utilizing passive mechanism, Proceedings of the 2006 IEEE International Conference on Robotics and Automation, pp. 381-386, 2006.

[7] K. Ito, T. Kamegawa, F. Matsuno, Extended QDSEGA for Controlling Real Robots -Acquisition of Locomotion Patterns for Snake-like Robot-, Proc. of IEEE Int. Conf. on Robotics and Automation, pp 791-796, Sep. 14-19, 2003

[8] R. Murai, K. Ito, Fumitoshi Matsuno, An intuitive human-robot interface for rescue operation of a 3D snake robot, Proc. of 12th IASTED Int. Conf. on Robotics and Applications, pp. 138-143, 2006

[9] R. S. Sutton. Reinforcement Learning: An Introduction. The MIT Press, 1988.

[10] C.J.H. Watkins and P. Dayan, Technical note Q-learning, Machine learning 8, pp.279-292, 1992.

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

[12] 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, pp 411–416, 2001.

[13] Eiji Uchibe, Minoru Asada, and Koh Hosoda, Behavior Coordination for a Mobile Robot Using Modular Reinforcement Learning, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1329-1336, 1996.

[14] K. Ito, F. Matsuno, Reinforcement Learning for Redundant Robot -Solution of state explosion problem in real world-, Proc. of ROBIO'05 Workshop on Biomimetic Robotics and Biomimetic Control, pp. 36-41, 2005.

[15] K. Ito, Y. Fukumori, A. Takayama, Autonomous control of real snake-like robot using reinforcement learning -abstraction of state-action space using properties of real world-, Proc. of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing, pp.389-394, 2007.

[16] D. Gu and H. Hu, Reinforcement learning of fuzzy logic controller for quadruped walking robots, Proceedings of 15th IFAC World Congress, Barcelona, Spain, July 21-26, 2002.

[17] C. Anderson and Z. Hong. Reinforcement Learning with Modular Neural Networks for Control. Proceedings of NNACIP'94, the IEEE International Workshop on Neural Networks Applied to Control and Image Processing, 1994.

[18] A. Likas, Reinforcement Learning Using the Stochastic Fuzzy Min-Max Neural Network, Neural Processing Letters 13, 213-220, 2001.