Application of reinforcement learning to hyper-redundant system
Acquisition of locomotion pattern of snake like robot

K. Ito and F. Matsuno

Graduate school of Interdisciplinary Science and Engineering, Tokyo Institute of Technology
4259 Nagatsuta, Midori, Yokohama, 226-8502 Japan
TEL 045-924-5546, FAX 045-924-5546
Email kazuyuki@cs.dis.titech.ac.jp, Matsuno@dis.titech.ac.jp

Abstract

We consider a hyper-redundant system that consists of many uniform units. The hyper-redundant system has many degrees of freedom and it can accomplish various tasks. Applying the reinforcement learning to the hyper-redundant system is very attractive because it is possible to acquire various behaviors for various tasks automatically.

In this paper we present a new reinforcement learning algorithm "Q-learning with propagation of motion". The algorithm is designed for the multi-agent systems that have strong connections. The proposed algorithm needs only one small Q-table even for a large scale system. So using the proposed algorithm, it is possible for the hyper-redundant system to learn the effective behavior. In this algorithm, only one leader agent learns the own behavior using its local information and the motion of the leader is propagated to another agents with time delay. The reward of the leader agent is given by using the whole system information. And the effective behavior of the leader is learned and the effective behavior of the system is acquired.

We apply the proposed algorithm to a snake-like hyper-redundant robot. The necessary condition of the system to be Markov decision process is discussed. And the computer simulation of learning the locomotion is demonstrated. From the simulation results we find that the task of the locomotion of the robot to the desired point is learned and the winding motion is acquired. We can conclude that our proposed algorithm is effective to the snake like hyper-redundant system and our analysis of the condition, that the system is Markov decision process, is valid.

Keywords:

Q-learning; hyper-redundant system; reinforcement learning; snake-like robot; propagation of motion

1. Introduction

We consider a hyper-redundant system that consists of many uniform units. The hyper-redundant system has many degrees of freedom and it can accomplish various tasks. Applying the reinforcement learning to the hyper-redundant system is very attractive because it is possible to acquire various behaviors automatically.

The reinforcement learning [3][4][5] has been much attention for the control method of real robots [6][7]. It does not need priori knowledge and has higher capability of reactive and adaptive behaviors. In the reinforcement learning, the designer has to prepare only one controller, and the different control laws are acquired automatically for each different task.

Q-learning [5] is regarded as one of the most typical methods of the reinforcement learning. In the Markov decision process, applying the Q-learning to the system, the optimal behaviors are acquired. Actually the reinforcement learning is applied to some simple tasks and their effectiveness is demonstrated [3]-[9]. However increasing of the action-state space makes it difficult to accomplish the learning process. So applying the reinforcement learning to the hyper-redundant system is very difficult and in the most of all previous works, the application of the learning is restricted to simple tasks with relatively small action-state space.

Considering these points we present a new reinforcement learning algorithm "Q-learning with propagation of motion". The algorithm is designed for the multi-agent systems that have strong connections. The proposed algorithm needs only one small Q-table even for the large scale system. So using the proposed algorithm, it is possible for the hyper-redundant system to learn an effective behavior. In the algorithm, only one leader agent learns the own behavior using its local information and the movement of the leader is propagated to another agents with time delay. The reward of the leader agent is given using the whole system information. And the effective behavior of the leader is learned and the effective behavior of the system is acquired.

In this paper we apply the proposed method to a snake-like hyper-redundant robot. It is composed of units that connect each other in series as the line form. The unit is regard as an agent and each agent has physical interaction each other. So we consider the snake-like robot as the multi-agent systems that have strong connections.

In the relative works of the snake-like robot, Hirose demonstrated the locomotion of snake-like robots based on the analysis of the locomotion of real snakes [1]. Iwasaki proposed the control law of locomotion of a snake-like robot based on precise physical model [2]. However these ways of locomotion are implemented by designer, and the snake robot can not adapt to a given environment automatically. In the methodology of controller design, the designer has to construct the different control laws for each different task so the load of work of the designer is very large. So the adaptive methodology that the control law is acquired automatically should be necessary.

To demonstrate the effectiveness of our proposed approach, the computer simulations of learning the locomotion of the robot to the desired position are carried out. And the necessary condition for the system to be the Markov decision process is discussed.

2. Hyper-redundant systems and snake-like robot

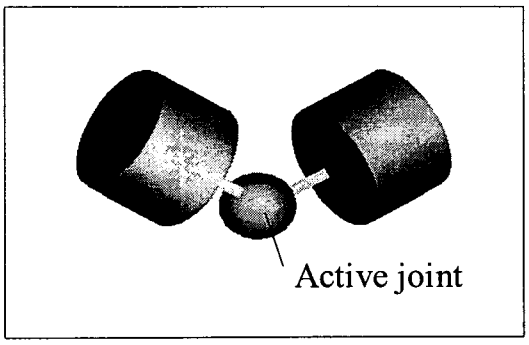

Fig. 1 Basic unit

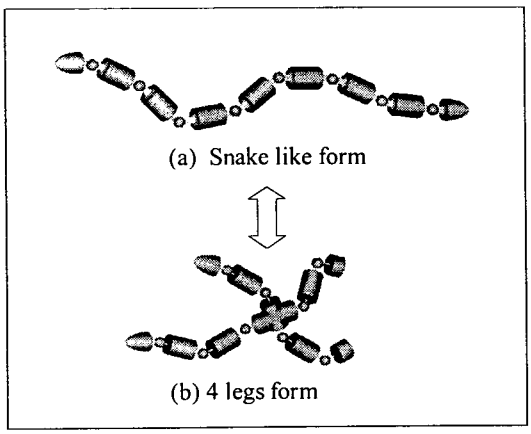

Fig. 2 Hyper-redundant mechanical systems

Fig. 1 shows a unit of a hyper-redundant mechanical system. The unit has one active joint and it can combine another units. A hyper-redundant system is composed of many uniform units and separation and recombination are possible (as shown in Fig. 2). By changing the form of combination, the hyper-redundant system can adapt itself to various environment and various tasks that are imposed. The hyper-redundant mechanical system is regarded as the adaptive hardware system, and the reinforcement learning is regarded as an adaptive software system. By combining these adaptive hardware system and software system, real adaptive system can be constructed. So the applying the reinforcement learning to the hyper-redundant system is very attractive and it might be effective.

The snake-like robot is one of the typical and simple forms of the hyper-redundant systems. It is composed of units that connect each other in series as the line form. The snake-like robot has many degrees of freedom and various movements are possible.

In this paper we consider one unit of the snake-like robot as the one agent. And each agent can control own joint locally.

3. Q-learning

Reinforcement learning is the method of acquisition of policy. In the unknown world, by repeating try and error, the agent learns the effective policy using information of reward only.

Q-learning is a reinforcement learning algorithm proposed by Watkins [1]. 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-vale 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 α is a learning rate ($0<\alpha<1$), and γ is a discount rate ($0 \le \gamma \le 1$).

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.

4. Problem of Q-learning for applying to hyper-redundant system

The hyper-redundant system has many degrees of freedom. Generally, the size of a state-action space is expressed as an exponential function of degrees of freedom. So the size of the state-action space of the hyper-redundant system is very large. In the Q-learning (not only Q-learning but also another various reinforcement learning algorithm has same problem) Q-vales is composed of all state-action space, so if the state-action space is large then implementation of Q-learning is impossible. And suppose the implementation is possible, the learning time is also exponential function of the number of degrees of freedom and the acquisition of the task is impossible.

To reduce the size of the state-space, suppose Q-value is composed of a sub-set of the state-space, then generally another problem occurs. This problem is caused by partial observation. Because of the partial observation, the

different states are observed as the same state. In this case even if the world constitutes a Markov decision process, the world can be observed non-Markov decision process. And it causes the obstruction for achieving the learning.

Some reinforcement learning algorithms for solving the partially observable Markov decision problem are proposed [8][9], but their application has been limited to a simple problem because they need a larger memory space or they use a probabilistic method. And it is difficult to apply them to the hyper-redundant system.

5. Proposed method

5. 1 Propagation of motion

A leader agent learns own behavior using its local information only. And the motion of the leader agent is propagated to other agents with time delay. The reward of the leader agent is given using information from the whole system. In this sub-section we describe the formulation of propagation of motion and the details of the method of the composition of the state-action space are explained in the next sub-section.

The motion of the leader agent is learned using Q-learning and the motion of it propagates the next agent, and similarly the motion propagates toward the end agent.

We consider the snake-like hyper-redundant robot that consists of N units. It means that the number of joints is N and the number of links is $N+1$. In this case the head unit of the snake-like robot can be regarded as the leader agent. We assume that all joint angles are controlled locally and they are regulated to desired angles within each time step.

$$\theta_n(t_i) = \theta_{nd}(t_i) \qquad (2)$$

In (2) $\theta_n(t_i)$ is the relative joint angle of the n-th ($2 \le n \le N$) joint at time t_i and $\theta_{nd}(t_i)$ is the desired value of $\theta_n(t_i)$.

The desired relative joint angle of Joint 1 (head joint) at time t_{i+1} is given as

$$\theta_{1d}(t_{i+1}) = \theta_1(t_i) + \Delta\theta_1(t_i) \qquad (3)$$

where $\Delta\theta_1(t_i)$ is the deviation of the joint angle of Joint 1 and it is the action of Q-learning. The deviation of the joint angle of the n-th joint is given as

$$\Delta\theta_n(t_i) = \Delta\theta_{n-1}(t_{i-1}) \qquad (4)$$

and the desired relative joint angle of the n-th joint is given as

$$\theta_{nd}(t_{i+1}) = \theta_n(t_i) + \Delta\theta_n(t_i). \qquad (5)$$

From (3)-(5), the motion of Joint 1 propagates to the latter joints.

5. 2 Composition of state-action space

We construct the state-action space using head unit information only. So the size of the state-action space is reduced compared to the case when the whole information is used. The action is the deviation of the joint angle of Joint 1 $\Delta\theta_1(t_i)$. The state is composed of the top position of the head, the absolute angle of the head link, the relative joint angle of Joint 1 $\theta_1(t_i)$ and the actions that are used from p steps past to 1 step past $\{\Delta\theta_1(t_{i-1}), \cdots, \Delta\theta_1(t_{i-p})\}$.

Using this composition method, the state-action space is composed of the head unit information only and the states of other units and actions are determined automatically.

In the next sub-section we consider the condition so that the world can be observed as the Markov decision process.

5.3 Condition to be Markov decision process

In the previous sub-section we construct the state-action space using only head unit (which is the representative part of robot) information. In general, under the composition, different shapes of robot can be observed as the same state and it causes the partial observable problem. And the propagation of motion causes the destruction of Markov property, because the robot motion depends on the actions that were used from N-1 steps past to 1 step past. In this section we consider the condition for the system to be complete observable Markov decision process.

At first we consider the partial observable problem. From (2)-(5), the n-th joint angle at time t_i can be written as (6).

$$\theta_n(t_i) = \theta_n(t_0) - \theta_1(t_0) + \theta_1(t_i) - \sum_{j=i-(n-1)}^{i-1} \Delta\theta_1(t_j) \qquad (6)$$

From equation (6), the n-th joint angle at time t_i can be expressed by using the initial joint angle of the n-th joint, initial joint angle of the head joint, the joint angle of the head joint at time t_i, and the actions which are used from n-1 steps past to 1 step past. And we do not need any other joint angles to calculate $\theta_n(t_i)$. It is very important. Now, suppose the initial shape is fixed and $p \ge n-1$, the joint angle of the n-th joint can be determined uniquely by using the head unit information only. The maximum value of n is N, so the condition not to occur the partial observable problem is "Initial shape is fixed and $p \ge N-1$". Under the condition, all joint angles can be observed completely in the composed state space using our proposed method. So the partial observable problem does not occur.

Next we consider Markov property. Because of propagation of motion, the actions that were used at from N-1 steps past to 1 step past is remained in the dynamics (3)-(5). By adding the past actions to the state, the past actions that remain in the dynamics are recognized as the different states. The maximum number of steps of actions that remain in the system is N-1. So, to satisfy Markov

property, we should set p to satisfy $p \geq N-1$.

Summarizing this section, we obtain that the condition for the system to be the complete observable Markov decision process is "Initial shape is fixed and p is set so as to satisfy the inequality $p \geq N-1$".

5.4 Size of state-action space

In this sub-section, we compare the original size of the state-action space with that of our proposed method.

Let N_{rp} be the number of region of the absolute position state (for example, position of the head and absolute attitude angle of the head link), N_{ra} be the number of region of the action for one joint, N_{rs} be the number of region of the state for one joint. In general, we design N_{ra} and N_{rs} to satisfy the following inequality.

$$N_{rs} \geq N_{ra} \qquad (7)$$

If the inequality (7) is not satisfied, the different states that are transited using different actions may be recognized as the same state.

The original size of the state space S_{os} is given as equation (8) and the original size of the action space S_{oa} is given as equation (9).

$$S_{os} = N_{rp} \times (N_{rs})^N \qquad (8)$$

$$S_{oa} = (N_{ra})^N \qquad (9)$$

The size of the state space of our proposed method S_{ps} is given as equation (10) and the size of the action space of proposed method S_{pa} is given as equation (11).

$$S_{ps} = N_{rp} \times N_{rs} \times (N_{ra})^{N-1} \qquad (10)$$

$$S_{pa} = N_{ra} \qquad (11)$$

From (8) and (10), we can find that the dimension of original state space and proposed state space are equal, but using the inequality (7) we find that $S_{ps} \leq S_{os}$ is satisfied for all N.

From (11), we can find that the S_{pa} is constant and is independent on the number of links N. And form (9) and (11) when N=1, S_{oa} and S_{pa} are equal, but when N>1, the relation $S_{pa} < S_{oa}$ is satisfied. And the number of links becomes lager our proposed method becomes superior.

We can conclude that the size of the state space and the action space can be reduced by using our proposed method. Especially the size of the action space is a constant that is independent on the number of links, so our proposed method is effective. In this paper we consider the snake-like robot as a typical example of the hyper-redundant systems. Using similar procedure we would obtain the similar conditions, for locomotion pattern generation of a multi-legged robot, the decision making strategy for multiple mobile robots, and so on.

6. Simulation

In this section, we implement the proposed method and show the acquired locomotion for a 10-links snake-like robot.

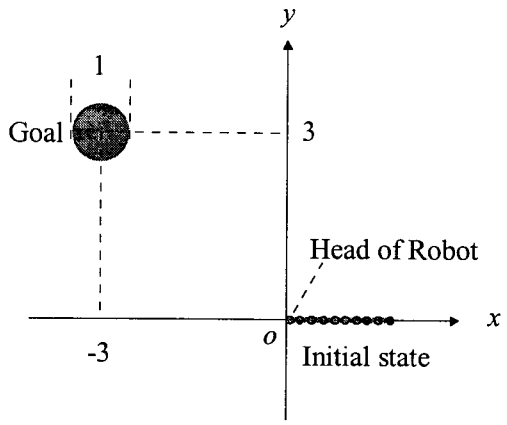

Fig. 3 Task of simulation

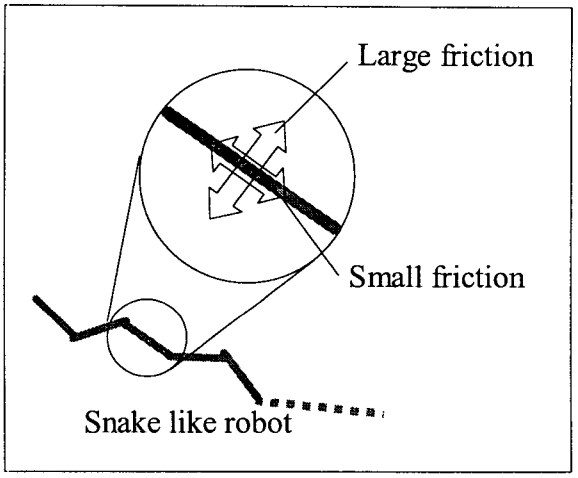

Fig. 4 Robot model for simulation

6.1 Task

In this paper we consider the task of locomotion to the desired position of a 10-links (9-units) snake-like robot. Let us define the coordinate as depicted in Fig. 3. The initial position of the head unit is (0,0) and the initial shape is a straight shape on the x-axis as depicted in Fig. 3. The desired position is set as $(-3, 3)$. And the aim of task is acquiring the locomotion pattern and moving the head of the robot to the desired position.

6.2 Robot model for simulation

In this simulation we employ the snake like robot model with friction proposed by Yamauchi et al. [2]. In this model, the friction between links and the ground is assumed as depicted in Fig. 4. All links touch the ground and the friction of the vertical direction with respect to the robot

body is larger than that of the tangential direction. Owing to this difference of friction the snake-like robot can move.

6. 3 Implementation of proposed method

At first we describe the composition of the state-action space. We set the action $\Delta\theta_1$ as two values: {−5[deg], +5[deg]}. And time step is set as 0.5[s]. The state is composed of the distance from the head unit to the goal, the direction of the goal from the head unit, the relative angle of head joint as depicted in Fig. 5 and the past actions.

The distance is divided into 6 regions: from 0[m] to 2.5[m] every 0.5[m] and else. The direction of goal position is divided into 10 regions: from –90[deg] to +90[deg] every 20[deg] and else. The relative angle of head joint is divided into 6 regions: from –25[deg] to +25[deg] every 10[deg] and else. To satisfy the condition that is discussed in the section 5.3, we set p as 8 and the state of the past actions is composed of 8 actions which was used from 1 step past to 8 steps past.

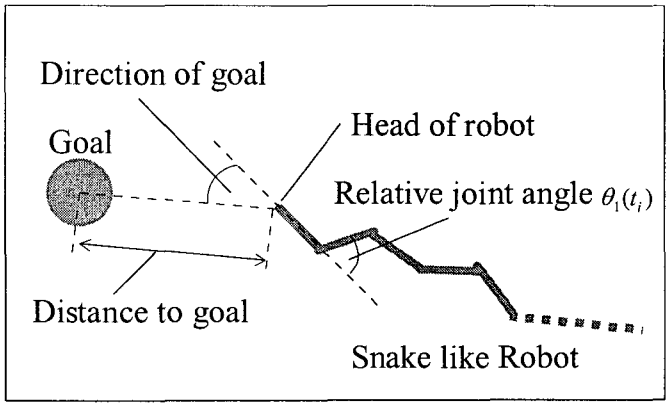

Fig. 5 State space

Next we describe the reward and penalty. When the distance from the goal to the head unit became less than 0.5[m], the reward 100 is given. And when the direction of the goal from the head exceeded the region from –90[deg] to +90[deg], the penalty –50 is given.

We assume that the movable region of each joint is from –25[deg] to +25[deg], and when a joint exceeds the movable limit, the penalty -50 is given.

When the reward or penalty is given, the robot is reset to the initial position and the next trial is restarted.

Next we describe the implementation of Q-learning. In this simulation we employ simple Q-learning as shown in (1) and to select an action, we employ the Boltzmann distribution

$$P(a \mid x) = \frac{\exp(Q(x,a)/T)}{\sum_{b \in actions} \exp(Q(x,b)/T)} \quad . \quad (12)$$

The other parameters are set as follows. The learning rate α is 0.5, the discount rate γ is 0.9, and the T in the Boltzmann distribution (12) is 1.

6. 4 Simulation results

Fig. 6 shows the learning history and Fig. 7 shows the acquired locomotion. In Fig. 6, the number of success increases as learning progresses and at the 20000th trial the learning is completed. The reason why the number of success can not converge to 50 (total number of trials) is that we employ the probabilistic method (12) to select the action. In Fig. 7, we can find that the winding motion is acquired and the head unit reaches the goal.

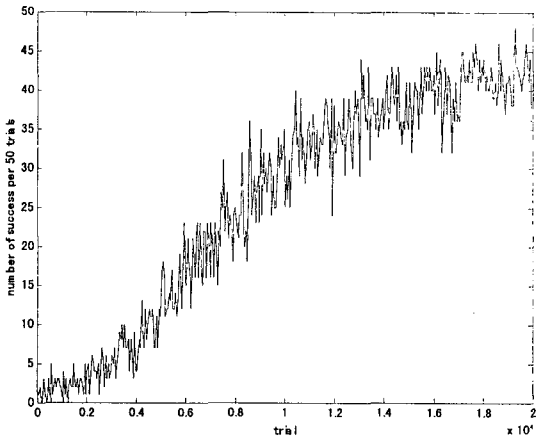

Fig. 6 Learning history

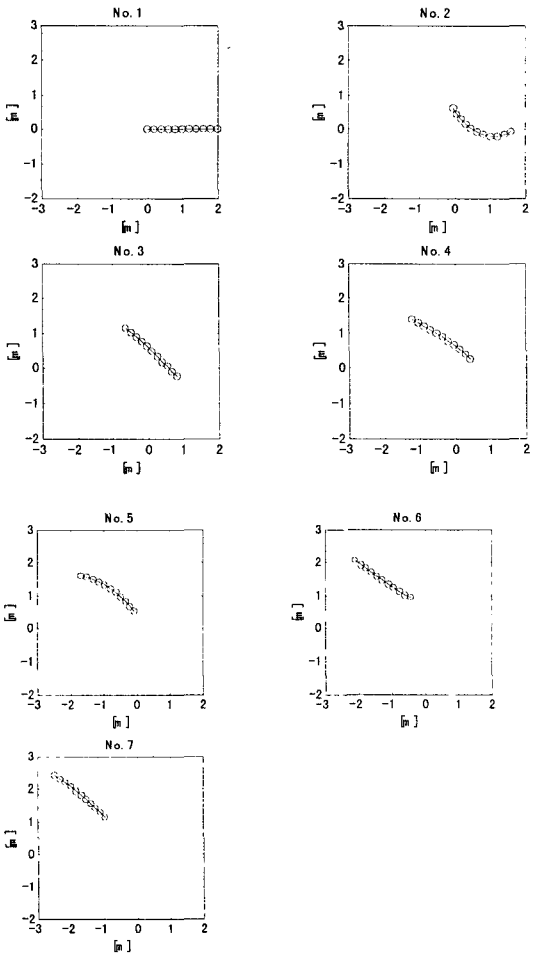

Fig. 7 Acquired locomotion

In the early trials, the robot body could not move efficiently. As the trails are repeated, the winding motion has been acquired and the task has been accomplished. Surely, the "propagation of motion" influences the locomotion, but the winding motion is not owing to the "propagation of motion" directly, the winding motion is acquired as a result of learning for the robot with the nature of "propagation of motion".

7. Conclusion

In this paper, we have discussed the effectiveness and the problems in applying the reinforcement learning to the hyper-redundant systems. We proposed "Q-learning with propagation" and applied it to the snake-like hyper-redundant robot. The condition that the system is the complete Markov decision process under the proposed algorithm was shown. The simulation of acquiring the locomotion of the snake-like robot was carried out. And the winding motion was acquired and the task that is moving the head of robot to the desired position was accomplished. We can conclude that our proposed algorithm is effective to the snake-like hyper-redundant robot system and our analysis of the condition, that the system is the Markov decision process, is valid.

Our proposed method is applicable to the multi-agent systems with the strong connections. To demonstrate the various applicability of the method we would consider the locomotion pattern generation for the multi-legged robots and the decision making strategy for multiple mobile robots and other examples.

Acknowledgment

This research is supported by The Grant-in Aid for COE Research Project of Super Mechano-Systems by The Ministry of Education, Science, Sport and Culture of Japan.

Reference

[1] S. Hirose, "Biologically Inspired Robots (Snake like Locomotion and Manipulator)", Oxford University Press, 1993

[2] H. Yamauchi, M. Fukaya, M. Saito, T. Iwasaki, "Locomotion Analysis of Hyper Redundant Systems", Proc. of 28th SICE Symposium on Control Theory, pp. 171-174, 1999 (in Japanese)

[3] R. S. Sutton, A. G. Barto, "Reinforcement Learning: An Introduction", A Bradford Book, The MIT Press, 1998

[4] R. S. Sutton, "Learning to predict by the Methods of Temporal Differences", Machine Learning 3, pp. 9-44, 1988

[5] C. J. C. H. Watkins, P. Dayan, "Technical note Q-Learning," Machine Learning, Vol. 8, pp 279-292, 1992

[6] M. Svinin, S. Ushio, K. Yamada, K. Ueda, "Emergent systems of motion patterns for locomotion robots", Proc. Int. Workshop on Emergent Sunthesis, pp. 119-126, 1999

[7] S. Ushio, M. Svinin, K. Ueda, and S. Hosoe, "An Evolutionary Approach to Decentralized Reinforcement Learning for Walking Robots", Proc. of the 6th Int. Symp on Artificial life and Robotics, pp. 176-179, 2001

[8] T. Jakkola, S. P. Singh, M. I. Jordan, "Reinforcement Learning Algorithm for Partially Observable Markov Decision Problems", Advances of Neural Information Processing Systems 7, pp. 345-352, 1994.

[9] S. P. Singh, T. Jakkola, M. I. Jordan, "Learning Without State-Estimation in Partially Observable Markov Decision Problems", Proceedings of the 11th International Conference on Machine Learning, pp. 284-292, 1994.