Task-Oriented Reinforcement Learning for Continuing Task in Dynamic Environment

Kamal, Md.Abdus Samad
Department of Electrical and Electronic Systems Engineering, Graduate School of Information Science and Electrical Engineering, Kyushu University : Graduate Student

Murata, Junichi
Department of Electrical and Electronic Systems Engineering, Faculty of Information Science and Electrical Engineering, Kyushu University

Hirasawa, Kotaro
Graduate School of Information, Production and Systems, Waseda University

https://doi.org/10.15017/1515919

出版情報：九州大学大学院システム情報科学紀要. 9 (1), pp.7-12, 2004-03-26. 九州大学大学院システム情報科学研究院
バージョン：
権利関係：

九州大学大学院
システム情報科学紀要
第9巻 第1号 平成16年3月

Research Reports on Information Science and
Electrical Engineering of Kyushu University
Vol.9, No.1, March 2004

Task-Oriented Reinforcement Learning for Continuing Task in Dynamic Environment

Md.Abdus Samad KAMAL*, Junichi MURATA ** and Kotaro HIRASAWA***

(Received December 3, 2003)

Abstract: This paper presents task-oriented reinforcement learning, a modified approach of reinforcement-learning to simplify continuing dynamic problems in a more realistic and human-like way of thinking from the viewpoint of the tasks. In this learning method an agent takes as input the 'state of task' instead of 'state of environment' and chooses appropriate action to achieve the goal of the corresponding task. The proposed system learns from the viewpoint of tasks that enables the system to find and follow a precise policy in a continuing-dynamic environment and offers simple implementation for a multiple agents system.

Keywords: Reinforcement learning, Multiagent system, Dynamic tile world

1. Introduction

In a reinforcement learning (RL) system an autonomous agent successively improves its policy through repeated process of interaction with the environment without any supervision. The episodic problem consists of a series of iterated subsequences, such as plays of a game, called 'episodes'. This is the simplest learning task, where the agent gets the opportunity to repeat its trial for the same initial conditions, resulting in faster convergence. Most real world problems are continuing tasks, or tasks consisting of a single ever-lasting episode only, in nature with high dynamics and it is hardly possible to model in an episodic manner. For such learning tasks, the conventional RL agent has to learn from a different task situation at each time, which often causes failure in convergence.

This paper presents a new RL scheme called task-oriented reinforcement learning (TORL)[1),2)] to solve complex dynamic problems. In this method a separate RL is carried out for each logical subtask from the viewpoints of the subtask considering the corresponding goal, instead of a single RL for the whole problem. The task-oriented learning offers precise action selection in the dynamic environment where the environment changes with time and can be applied successfully to both episodic[1)] and continuing tasks[2)]. The dynamics of a continuing world is more complex to an agent. For a usual complex problem, some researchers have proposed physical division of a large task in a modular form to reduce the complexity in learning, where they use a central mediator module to select an action among the modules[3)]. Others have proposed multiple agents in lieu of a single agent to make a complex learning task easier through combining outcomes of multiple agents. But none of them studied the scope of reducing complexity by making an agent learn from the viewpoint of the task considering only the related information needed to attain the goal. The proposed task-oriented system learns from the viewpoints of the task, which simplifies the learning process and enables the agent to choose an action more precisely. Beside this, it attempts to combine the advantages of both multiple agents and modular RL approaches without any central coordinating mechanism.

In this paper we investigate the performance of TORL for a test bed, dynamic tile world. Learning is carried out considering two main tasks. The first task treats, from the viewpoint of the agent, how it moves around the environment, and the second one is related to, from the viewpoint of the tile, how the tile can be pushed into the hole. The learning process does not depend on the initial state of an agent and the experience of a trial can be effectively applied to the different type of trials. The effectiveness of the proposed system is also verified for multi-agents. The use of separate lookup tables greatly reduces the dimensionality in state spaces hence ensures faster learning, and the task oriented policy helps in attaining precise policy in a dynamic environment.

* Department of Electrical and Electronic Systems Engineering, Graduate Student

** Department of Electrical and Electronic Systems Engineering

* * * Graduate School of Information, Production and Systems, Waseda University

Next section describes the basic notion of conventional reinforcement learning and the modified task-oriented approach. Section 3 introduces the test bed, dynamic tile world, and implementation of the task-oriented algorithm. Section 4 contains simulation results, and Section 5 contains discussions on the proposed method and obtained results, and finally conclusions are drawn in Section 6.

2. Task-Oriented Reinforcement Learning

Reinforcement learning[4] is a process of trial-and-error whereby an agent seeks to find the combination of actions that maximizes the rewards as its performance feed back. One of the most commonly used reinforcement learning methods is Q-learning. This algorithm does not need a model of the environment and directly computes the approximate function of optimal action-value independently of the policy followed. The updating rule of Q-learning is as follows:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)], \quad (1)$$

where, α is the learning rate, γ is the discount factor, r_t is the achieved reward at time t, and $Q(s_t, a_t)$ is the value of action a_t at state s_t.

The agent uses all perceivable information of the environment to constitute the state and tries to choose a better action maintaining a balance between exploration and exploitation according to the certain policy. The convergence of the Q-learning is proven under the assumption that each state-action pair is visited infinitely often. Unfortunately, in continuing-dynamic problems an agent has to deal with new working situation in the environemnet and it is hardly possible to visit all states repeatedly within a short interval to meet convergence.

The task-oriented approach of RL reduces the complexity of such problems by learning from the viewpoint of the task considering only related information after decomposing the whole problem into some logical subtasks according to the types of actions. For each subtask a separate lookup table is used to store Q values, where the corresponding agent uses only the specified information of the environment to understand the present condition of the task in term of 'task-state', and chooses an action. The goal of each subtask may be different apparently, but it helps attaining the global goal of the system. This method provides one lookup table for each subtask, therefore, the same agent may deal with all lookup tables[2] or a separate agent can be used for each subtask[5] depending on the involvement of the agent. The main objective of this method is to simplify the learning process considering less information related to corresponding task only, which ignores the information beyond the task, hence less affected by environement dynamics. At any environement instance, an agent can proceed with its task as it needs only task related information, which enables the TORL method work well in both episodic and continuous manner.

Figure 1 shows a comparative notion of task-oriented and conventional RL: in a conventional RL system, the agent learns to update the policy by choosing an action considering the state of the environment without specific idea on task; in task-oriented RL, the agent learns to update the policy by choosing an action considering the state of task from the viewpoint of the task. In the task-oriented system, policies belong to tasks instead of agent, which provides an opportunity of sharing a policy among the agents handling similar task in multiagent systems. Agents' cooperation by updating common policy makes a system achieving faster convergence with minimum memory requirement[6].

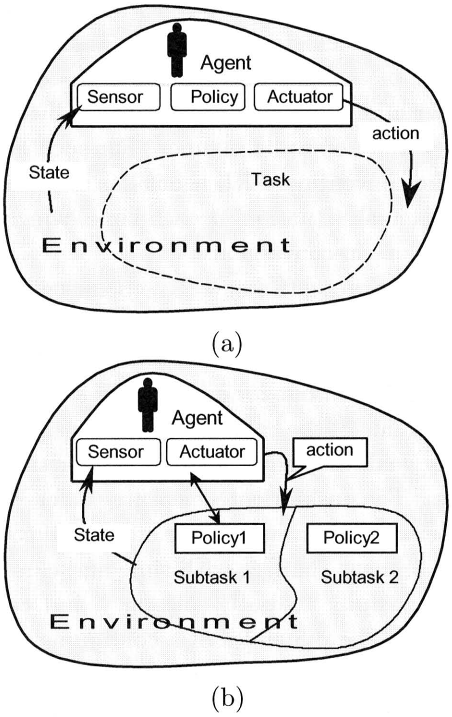


Fig. 1 Comparative notion of (a) conventional reinforcement learning system, and (b) task-oriented reinforcement learning system.

This mechanism of cooperation by sharing the policy of same task is introduced in task-oriented multiagent RL system to boost up its performance.

3. The Tile World Test Bed

3.1 Domain description

A pseudo-realistic tile world of 10 by 10 grids that evolves in discrete time steps is considered as our test bed for TORL, **Fig. 2**. Each cell of the world may contain an agent, an obstacle, a tile or a hole. In this continuing and dynamic environment, a tile and a hole appear in random location of the environment stochastically and disappear after a certain time interval. Although the shape and obstacle structure of the environement are fixed in learning, it becomes a dynamic environemnt to an agent due to different tile and hole locations at each time. The agent is permitted only to push the tile but not to pull, and the movement in diagonal directions and moving off the environment are considered to be illegal. More than one agent, or an agent and the tile cannot be in the same cell. The agent's task is to discover the tile-hole pair and then fill up the hole by putting the tile into it at each trial, then another tile-hole pair appears in the world at diffrent location and the agent has to continue the same process forever.

The cells just inside the boundary region are restricted only for the agent's movement, and the tile is not allowed to be pushed into them. This restriction is only to avoid the permanent dead lock situation and ensuring the continuity of the environment. At each time step the agent has four possible actions to choose from: pushing the tile if it is available or moving in to North, South, East or West. Before making any action, the agent searches its field of vision of limited depth 2 for the tile, hole, other agent, etc. The environment is partially observable and there is no special marks or coordinate to distinguish one cell from another.

3.2 Implementation

In this dynamic tile world, the agent needs to find out the tile and the hole at first, so it moves randomly throughout the environment until it finds out both. Then the agent needs moving to desired location (to a certain side of the tile to push it or after pushing once it may need to move to another side of the tile to push it again) in the environment, and this process continues at each trial until the tile is being pushed into the hole. In applying task oriented RL, here the whole task is decomposed as follows: (a) agent movement throughout the environment needed for finding out the tile-hole pair, or for moving to any particular location, and (b) transition of the tile towards the hole.

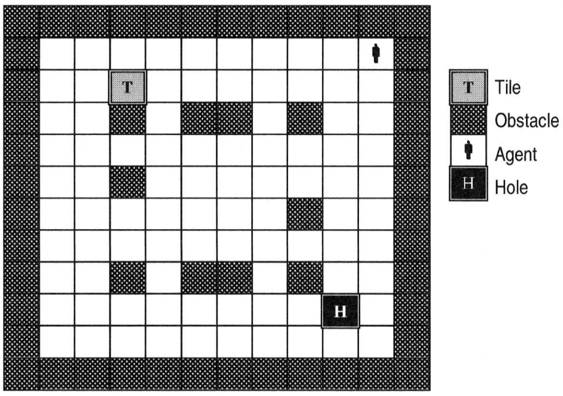


Fig. 2 Tile world.

The first subtasks fully concentrates on the agent's movement only. To search the environment the agent decides a relative location (sub-goal) randomly and looks for the tile and the hole during its trip towards that location. This process is repeated several times until the tile-hole pair is found out. The agent remembers the location of the tile and the hole by relative Cartesian coordinates and updates this value at each transition of state. After finding out both or after pushing the tile once the agent may need moving to a certain cell (sub-goal) beside the tile, which is determined by the Q-value of the subtask related to the tile movement. This subtask ends when the agent reaches the appropriate cell beside the tile to push it. For this subtasks of agent movement the Q_A-table is proposed. The state of Q_A-table is constituted by the relative directional information of the sub-goal state and status of the neighbor cells. The actions are either to move in North, South, East or West. The action that makes the agent to reach the target location is

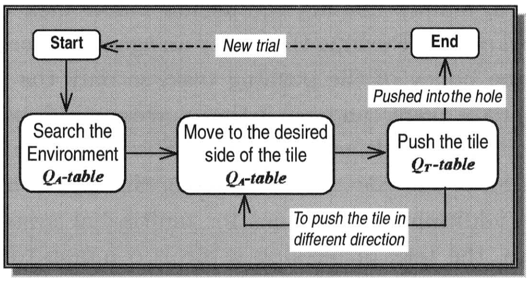

Start
New trial
End
Pushed into the hole
Search the Environment Q_A-table
Move to the desired side of the tile Q_A-table
Push the tile Q_T-table
To push the tile in different direction

Fig. 3 Typical way of action selection, reward obtaining and updating Q-table by the idle elevator agent.

given a reward of 1 and all other actions receive a reward of 0.

The other subtask deals with transition of the tile, how the agent should handle it. For this subtask a separate lookup table, Q_T-table, is proposed that contains the information of the tile itself instead of the agent, but the table is availed and updated by the agents. The state space of Q_T-table is constituted by the relative directional information of the hole (goal) and the status of the tile's neighbor cells ignoring the presence and activity of the agent, and the action space is the indication to the agent, in which direction the tile should be pushed. The agent acquires the information from it to handle the tile and update it after each transition of the tile. Pushing the tile once, if the agent needs to move in another location of the tile, it uses Q_A-table. When the tile is pushed into the hole, which is the final goal, it receives a reward 1.0, while all other transitions of the tile receive a reward of 0. The block diagram of proposed learning system with corresponding Q-tables is shown in **Fig. 3**.

In both subtasks, the corresponding goal position defined by relative Cartesian coordinate is scaled logarithmically in constituting the state. This logarithmic scaling makes the agent to know the goal and its surrounding in more details when it is nearer, and have a brief idea of goal when it is far. The representation of the state using relative information of the goal rather than exact location makes the system deviate from pure Markov Decision Process (MDP). The Q-learning system is guaranteed to converge for MDPs, but the performance degrades for non-Markov tasks. The $Q(\lambda)$-learning is the first line defense against both long-delayed rewards and non-Markov tasks[4], but it requires huge computation to update the Q-table at each time. For these reasons, we considered truncated $Q(\lambda)$-learning[7] algorithm for all subtasks.

The system can be converted into a multiagent system simply making the agents follow task oriented policy. Pushing the tile by an agent is directed by the policy of tile pushing task, so only the appropriate agent can push it that makes an indirectly coordinated multiagent systems. Sharing the information of the tile and hole among the agents may give additional advantages for multiagent system. Since, the task of pushing a tile is common to all agent, all agent can use and update corresponding policy of the task, which gives faster convergence in learning.

4. Simulation Results

Simulations were performed for the above problem with truncated $Q(\lambda)$ and ε-greedy policy with decreasing value of ϵ, the learning rate $\alpha = 0.1$, the discount factor $\gamma = 0.90$ and $\lambda = 0.75$. The life time of a tile-hole pair is set at 3000 steps.

At first we have tested the above problem in an episodic way making the agent, tile, and hole at the same initial positions in each episode. **Figure 4(a)** shows the success rate, percentage of trials with achieved goal within the fixed interval, and **Fig. 4(b)** shows the average time required to finish a trial. For this test one trial is considered as an episode. The results show high success rate of 99.34% with satisfactorily less elapsed time of 155 steps to finish an episode after only a few thousands episodes of learning. This test has been done only to indicate its capability to conduct episodic tasks as well as continuing tasks.

Figure 5 shows the performance of the proposed task-oriented RL system in handling the continuing-

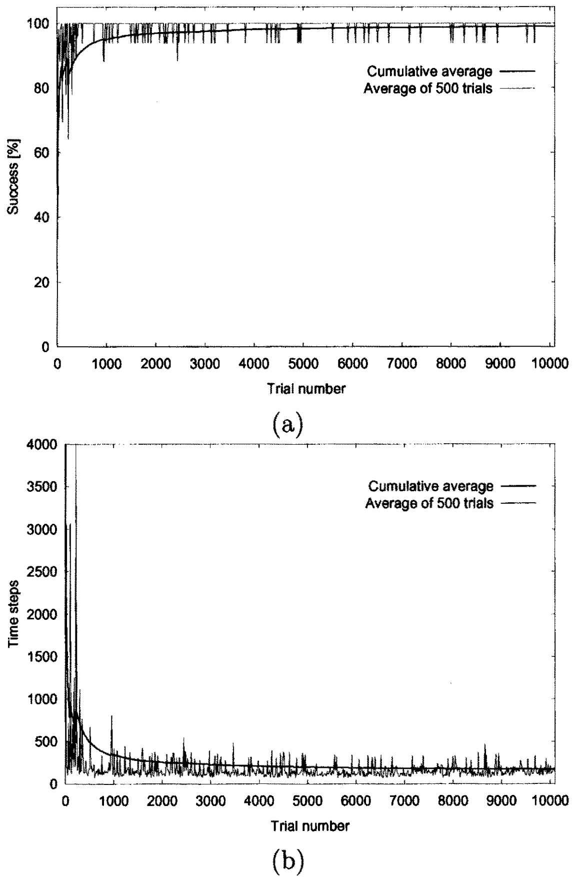
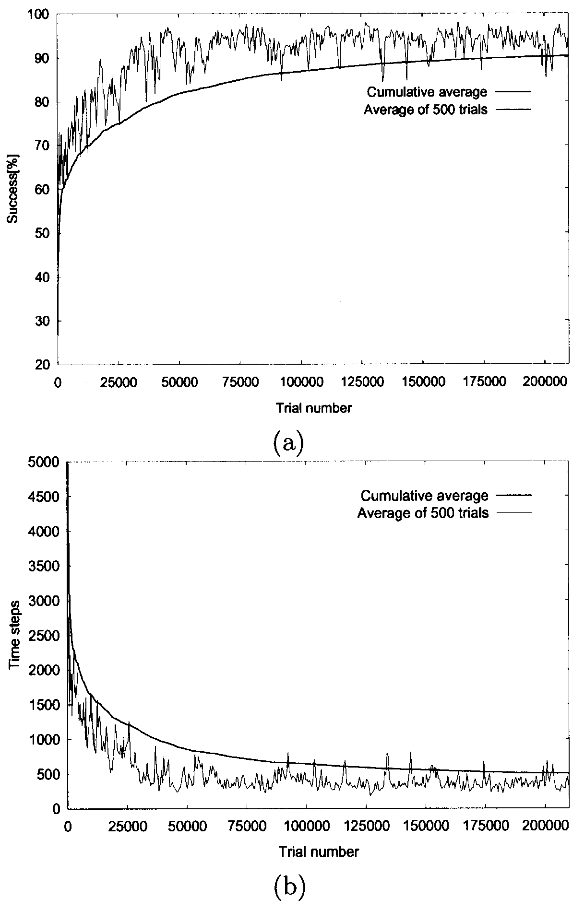


Fig. 4 Learning for the episodic task: (a) the percentage of average success, (b) average time steps.



(a)

(b)

Fig. 5 Learning for the continuous task: (a) the percentage of average success (b) average time steps.

dynamic task. The system learned slowly but ultimately it achieved satisfactory performance. The ultimate success rate is more than 94% and the average time to finish a trial remains within reasonable limits of about 450 steps. The elapsed time of a trial depends on how much obstacles exist in the environment. it is found that for fewer obstacles the system shows better performance. The conventional RL method cannot solve this dynamic problem since the agent has no chance to repeat its trial for same situations.

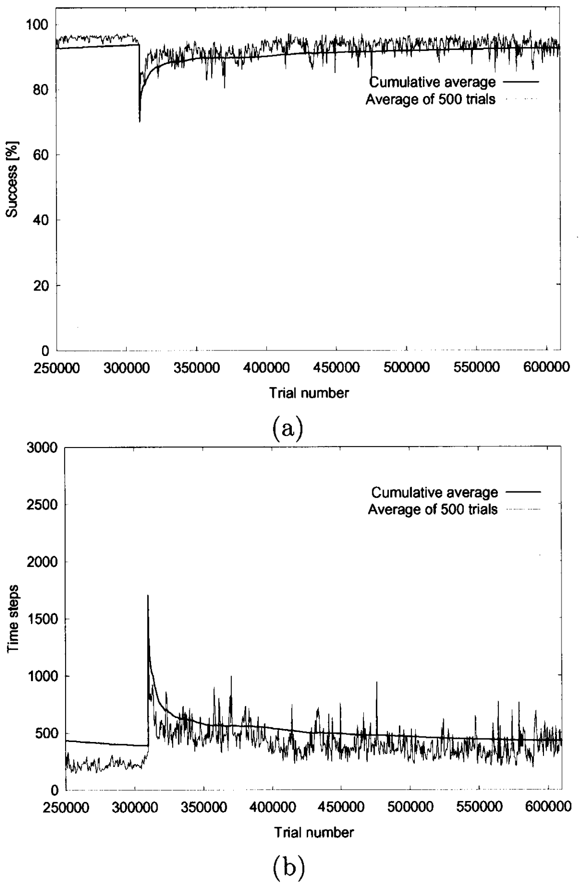
Keeping the same difficulty level, the positions of the obstacles in the environment are changed once in on going learning process. The success level suddenly falls from 93% to 70% in **Fig. 6**, but after an interval of time it re-achieved it. This small deviation with faster recovering capability shows the robustness of the system and capability of applying the learned policy for different environment. The small deviation would not occur, if it were possible to use 90° rotationally symmetry of directions.

The performance of a multi-agent system has also been investigated, introducing another agent into the system with its own lookup table and making them to share the lookup table related to tile pushing. **Figure 7** shows the comparison for 1-agent and 2-agent system. The system suited well with multi-agent and shows better performance both in the success rate and the average time to finish a trial. This simplest way to convert the system for multiagent in achieving better performance shows an additional feature of the proposed method and recommending it for multiagent systems.

5. Discussions

In the proposed TORL method, human effort is needed to decompose the task into some subproblems, separating related information for each task, and to define their relations obtaining the global goal. Apart from this additional requirement, this method offers a number of advantages, which cannot be achieved using monolithic reinforcement learning methods for continuous dynamic tasks.

First, thinking from the viewpoint of the task



(a)

(b)

Fig. 6 Sudden change in obstacles arrangement: (a) the percentage of average success, (b) average time steps.

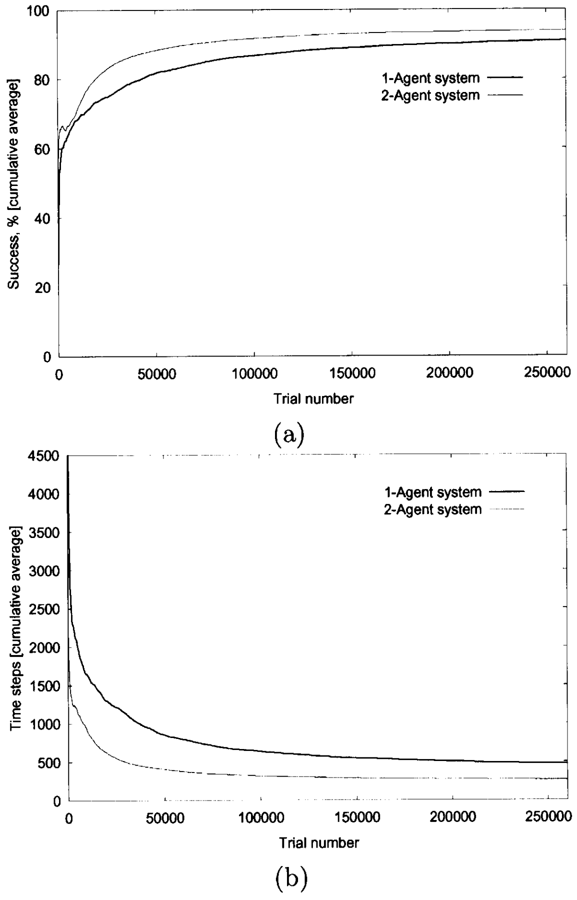


Fig. 7 Comparison with single agent and two-agents system: (a) the percentage of average success, (b) average time steps.

leads the agent to choose an action more precisely. This feature makes the system robust since the agent gets a gist of the task from state information, which remains same in all environment conditions. So learning process goes well in a dynamic environment, and the agent can restart its trial at any instance in the continuous world. In multiagent systems agents can share their policy, and no additional coordinating mechanism is required as only the suitable agent handle the task. Second, the task decomposition limits the size of state-spaces hence ensures less memory and less computation requirements, which leads the system achieving faster convergence.

This kind of dynamic task cannot be handled using conventional RL, whereas in TORL method agents fulfill the goal of a trial and from that point restart for the next trial, and follows the continuing process successfully.

6. Conclusions

A novel method, TORL has been presented here, which generalized states representation in terms of status of task that simplified the continuous dynamic task into an easily learnable problem, reduced the size of state-spaces and fast convergence were achieved. It provides an indirect coordination for multiagent systems without any change in state size or learning structure but gives remarkable performance compared to the single agent systems. These promising results reveal the versatility of the task-oriented system and it can be extended to design a complete multi-objective intelligent society of autonomous agents.

References

1) M.A.S. Kamal, J. Murata and K. Hirasawa: Task-Oriented Reinforcement Learning in Cooperative Multiagent System, Procedings of the 20th SICE Kyushu Branch Annual Conference, pp.477-480 (2001).

2) M.A.S. Kamal, J. Murata and K. Hirasawa: Task-Oriented Reinforcement Learning for Continuous Tasks in Dynamic Environment, Proceeding of the SICE annual conference, pp.932-935 (2002).

3) Kui-Hong Park, Yong-Jae Kim, Jong-Hwan Kim: Modular Q-learning based multi-agent cooperation for robot soccer, Robotics and Autonomous Systems Vol.35, pp.109-122 (2001).

4) R. S. Sutton, and A. G Barto: Reinforcement Learning: An Introduction, MIT press (1998).

5) M.A.S. Kamal, J. Murata and K. Hirasawa: Task-Oriented Multiagent Reinforcement Learning Control for a Real Time High-Dimensional Problem, Proceedings of the eighth International Symposium on Artificial Life and Robotics, Vol.2 pp.353-356 (2003).

6) M. Tan: Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents, Proceedings of the Tenth International Conference on Machine Learning, pp.330-337 (1993).

7) P. Cichosz: Reinforcement learning by truncating temporal differences,Ph.D dissertation, Dept. Electron. Inform. Technol., Warsaw Univ. Technol. Warsaw, Poland (1997).