A Module based Robust Learning System to Environmental Change for Robot Brain

Masahiro Ono, Mamoru Sasaki and Atsushi Iwata

Graduate School of Advanced Sciences of Matter, Hiroshima University,
1-3-1 Kagamiyama, Higashi-Hiroshima-shi 739-8527, Japan
Phone and Fax: +81-824-22-7358, Email: {masa, sasaki, iwa}@dsl.hiroshima-u.ac.jp

1. Introduction

Recently, the progress of robots is remarkable in the field of high speed operation, humanoid, imitation of behavior and entertainment. Nevertheless, few autonomous robots having behavior-learning capability are developed. The main reason is that learning is insecure to environmental change and the learning speed is slow. Doya et al.[1] proposed a module based learning system, for solving this problem. It has many modules, each of which is composed of a situation predictor and learning controller. The situation predictor estimates the current environment dynamics and outputs the expected situation of next time. The system selects one module which has the best estimation of a situation transition sequence. The learning controller in the selected module chooses the suitable action, based on the output of the predictor. Thus, the system can quickly respond to environmental change. However, it needs more storage capacities for a predictor as the number of situations in an environment increases. Besides, one module can't be utilized to over two environments. Increasing the number of environments in a task, a required memory space increases. Especially, in autonomous robot applications, memory capacity is limited with power dissipation and physical size. It is difficult to apply to the complex real environments. Then, we propose an advanced module based learning system which reduces memory capacities by removing predictors without reducing adaptive learning capability to environmental change.

2. Proposed Learning System

Processing procedure in the proposed system is shown in Figure 1. The system works as follows**:** (1) detection of the current environment, (2) selection of a policy from memorized policies which can be suitable for multiple environments. We call it representation policy, "RP", (3) construction of a new policy by learning, and (4) addition of a new RP or elimination of it. The policy means the map from a situation of an environment to an action. We also regard a policy as a module.

2.1. Detection of the environmental change and selection of a RP

We define the rate of reach achievement (R). It means the rate how often robots with the system have achieved the goal. Hence, R is large if the current policy works effectively. If it is lower than a threshold Rg during some periods, the system selects another RP which is added more newly than the current one.

2.2 Learning of RP

We use reinforcement learning (RL) based on partial policy correction [2]. RL is an algorism which learns effective policies to the environment, based on rewards "**r**" given from the environment when the robots achieve the goal situation. RL based on partial policy correction uses Q-Learning, which is a method of RL. Q-Learning's purpose is to calculate action value function Q(**s**, **a**) ("**s**" is a situation, and "**a**" is an action). Q(**s**, **a**) expresses the quality of **a** in **s** (larger Q means that an action is better.). The system constructs a policy based on Q(**s**, **a**). The update rule is as follows. $Q(\mathbf{s}, \mathbf{a}) \leftarrow Q(\mathbf{s}, \mathbf{a}) + \alpha(\mathbf{r} + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}', \mathbf{a}') - Q(\mathbf{s}, \mathbf{a}))$, where **s**, **a**, **s**', **a**' means the last situation, the last action, the current situation and the next action to select, respectively. α and $\gamma (0< \alpha, \gamma<1)$ are discount parameter and step size parameter. RL based on partial policy correction [2] has only one policy and applies the policy constructed in the last environment to the current environment. When R is lower than Rg, some Q(**s**, **a**) including the main factor is corrected partially. Therefore, if both of the last and the current environments are alike, learning is very fast and moreover, the system can have the suitable policy to both environments. Consequently, the memory is also saved because one policy can be utilized to at least two environments. In contrast, when they are very different, learning is slow. In order to solve this problem, the proposed system has multiple policies. Thus, even if there are large differences between the current and the last environments, it is expected that learning is fast by selecting a proper policy.

2.3. Addition of a new RP or elimination of it

A correlation is calculated between the new policy and the memorized one. If the correlation is lower than a threshold, the new one is added to the memory. On the other hand, if the correlation is higher, the new one is eliminated.

3. Experiment

In order to evaluate the ability of the proposed system, we applied the proposed system to Maze Problem with some simulation experiments. Maze Problem is often used as an exercise for RL. We assumed a 5x5 maze in Figure 2. The purpose of the robot is to learn the shortest path from the start to the goal. The robot can select one of four actions: {up, down, left, right} (selecting one action is defined as one step.). If the robot takes a hundred steps or arrives at the goal, the robot is returned to the start. The reward is +1, or 0 if the robot reach the goal or not.

In order to evaluate the learning performance of the system for environmental changes, six kinds of mazes (a-f) are used. The maze changes randomly after Nc steps are taken. In order to compare the learning capability of the proposed system with conventional methods, we applied Q-Learning and RL based on partial policy correction to this maze problem.

Figure 3 shows the result during 20000 steps in case that Nc is 1000. The horizontal axis expresses steps (time steps) and the vertical axis expresses the following: (a):

environmental change, (b)-(d): number of steps required from the start to the current situation with the proposed system, RL with partial policy correction and Q Learning, respectively. In (b)-(d), as the value of the vertical axis is smaller, the policy used in the environment is more effective. While Q Learning and RL based on partial policy correction constructed the effective policy only for the maze "a", the proposed system learned the multiple effective policies for almost all mazes except maze "b".

Figure 4 shows simulation results of total rewards vs. Nc in 20000 steps. From Figure 4, the performance of the proposed system is better than the other methods over all of Nc. Additionally, the system constructed only two RP over all of Nc. Doya's system [1] probably requires more than six modules because one module is applied to one environment. One predictor needs at least 2500 data: 25(situation) x 4(action) x 25(next situation). Doya's system needs more than 15000 data (2500 x 6). On the other hand, in this experiment, the proposed system needs 150 data: Q(**s**, **a**) needs 100 (25(situation) x 4(action)) and two RP needs 50(25(situation) x 2(the number of RPs)). Therefore, the data size can reduce to about 1/100 comparing with the Doya's system.

4. Conclusion

We proposed the learning system for robot brain in order to realize autonomous robots with behavior learning. It was confirmed by the simulation experiments that the system is robust to environmental change and uses its memory more effectively. As next step, we'll improve selection of RP and implement this system into LSI.

References

[1] K. Doya et al., Technical report, Kawato Dynamic Brain Project Technical Report, KDB-TR-08, Japan Science and Technology Corporation, June 2000.

[2] T. Minato et al., Journal of the Robotics Society of Japan, Vol. 18, No. 5, pp. 706-712, 2000.

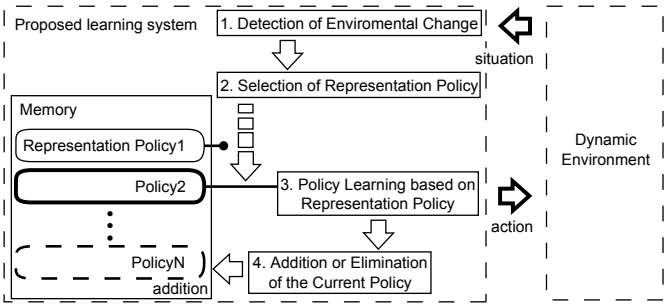


Figure 1: Processing procedure in the proposed learning system

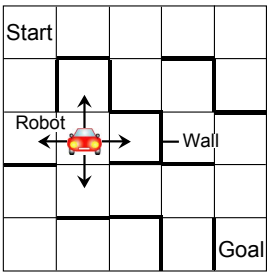


Figure 2: Maze problem used in this experiment

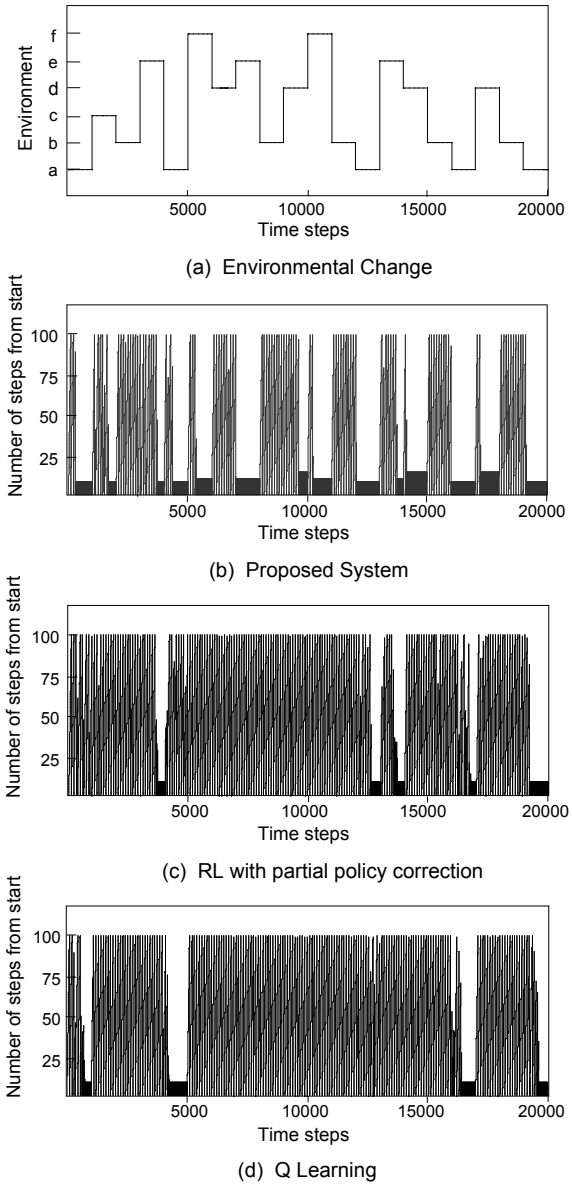


Figure 3: Result of the simulation experiment during 20000 steps (the environment changes randomly after the robot takes 1000 steps.)

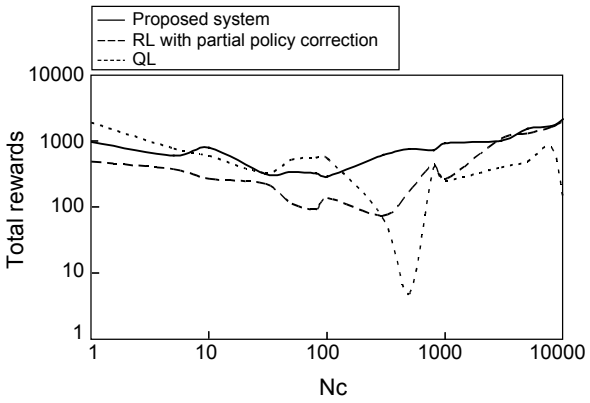


Figure 4: Total rewards vs. Nc in 20000 steps (the environment changes randomly after Nc steps are taken.)

A Module based Robust Learning System to Environmental Change for Robot Brain

Masahiro Ono, Mamoru Sasaki and Atsushi Iwata

Graduate school of Advanced Sciences of Matter, Hiroshima University

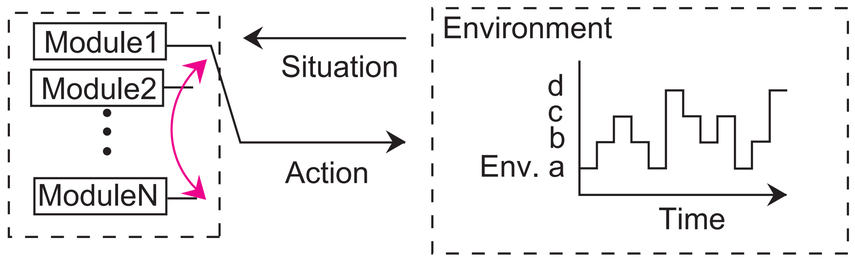
Introduction

Few autonomous robots having behavior- learning capability are developed.

The main reason: 1. **Instability of learning by environmental change**
2. Learning speed is slow.

In order to solve this problem

A module based learning model is proposed by Doya et al.

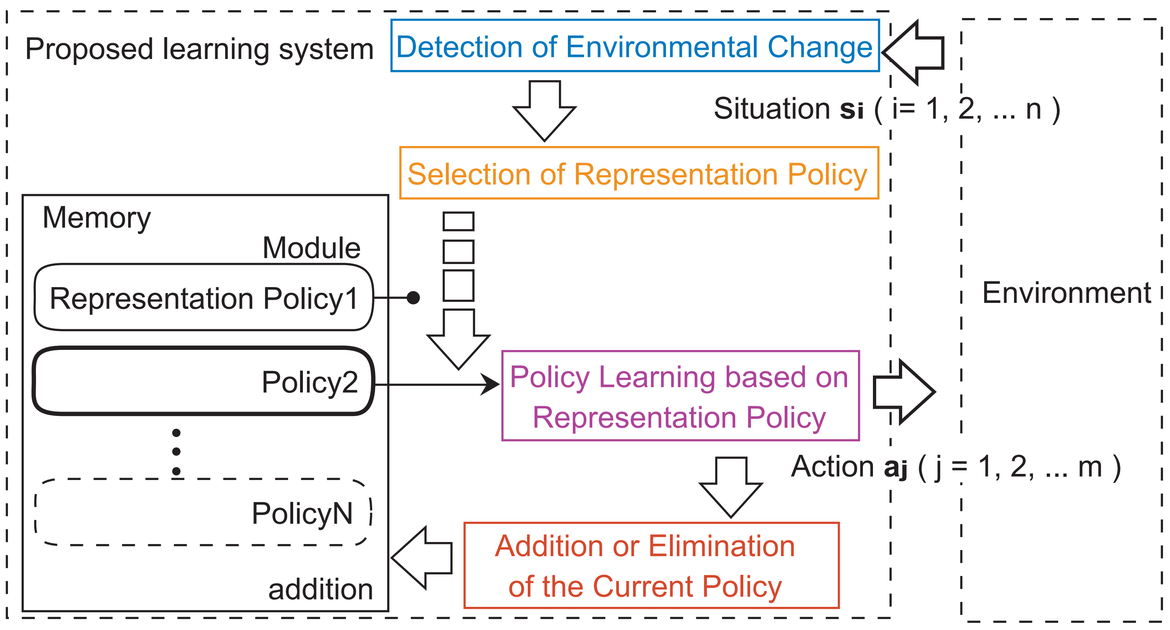


Effective module is selected for the current environment.

Advantage: Stability to environmental change (Quick adaptation)
Disadvantage: Large data size on condition that an environment is complex because of the necessity of modeling the environment

Purpose of research:
We propose an advanced module based learning system with stability to environmental change and **small data size**.

Proposed Learning System



Detection of Environmental Change

Definition The rate of goal attainment (R) = how often the robot have attained goal?

If **R > Rg (threshold)** during some period, the system **recognizes environmental change**.

Selection of Representation Policy

Next representation policy is selected until R > Rg.
(Example: If current RP* is RPi, then next one is RPi+1.) *RP: representation policy

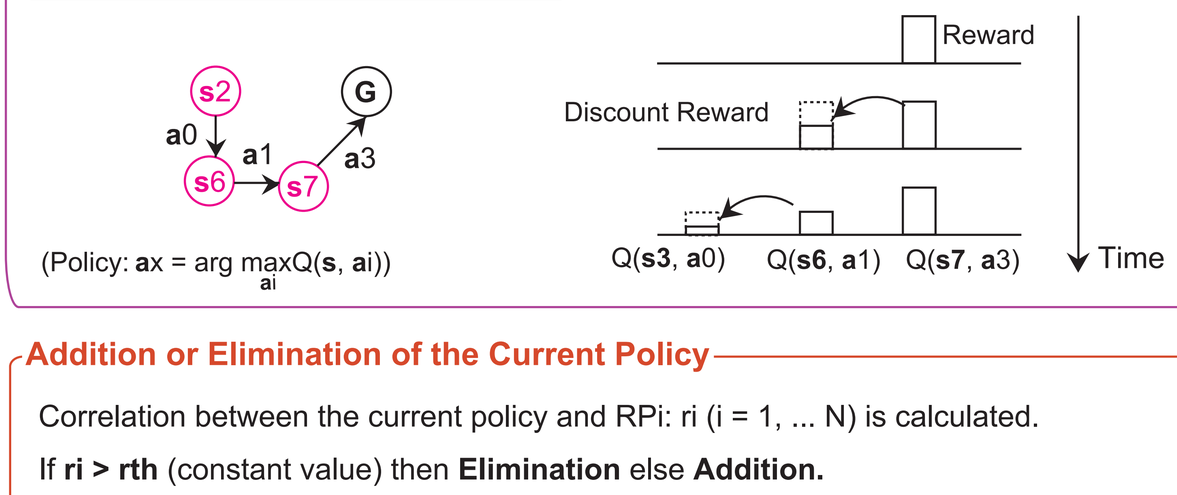
Policy Learning: Partial policy correction with Q Learning

Partial Policy correction

Policy: **a** = f(**s**)

Partial correction of policy: f(**s**3), f(**s**6), f(**s**7)

Q Learning (Reinforcement Learning)



Addition or Elimination of the Current Policy

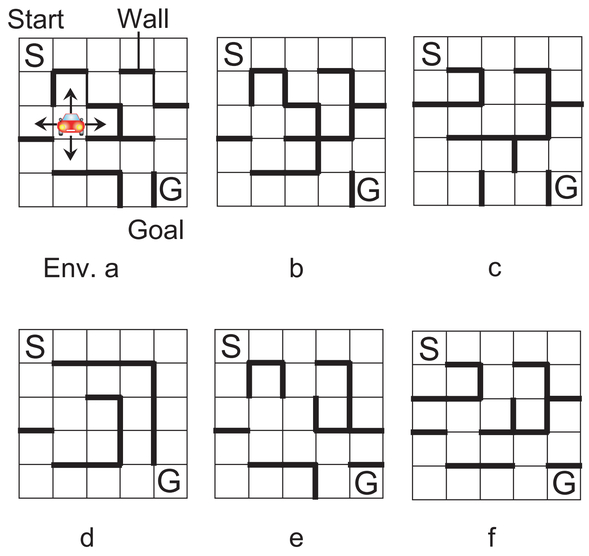
Correlation between the current policy and RPi: ri (i = 1, ... N) is calculated.

If **ri > rth** (constant value) then **Elimination** else **Addition.**

Numerical Simulation

Maze Problem

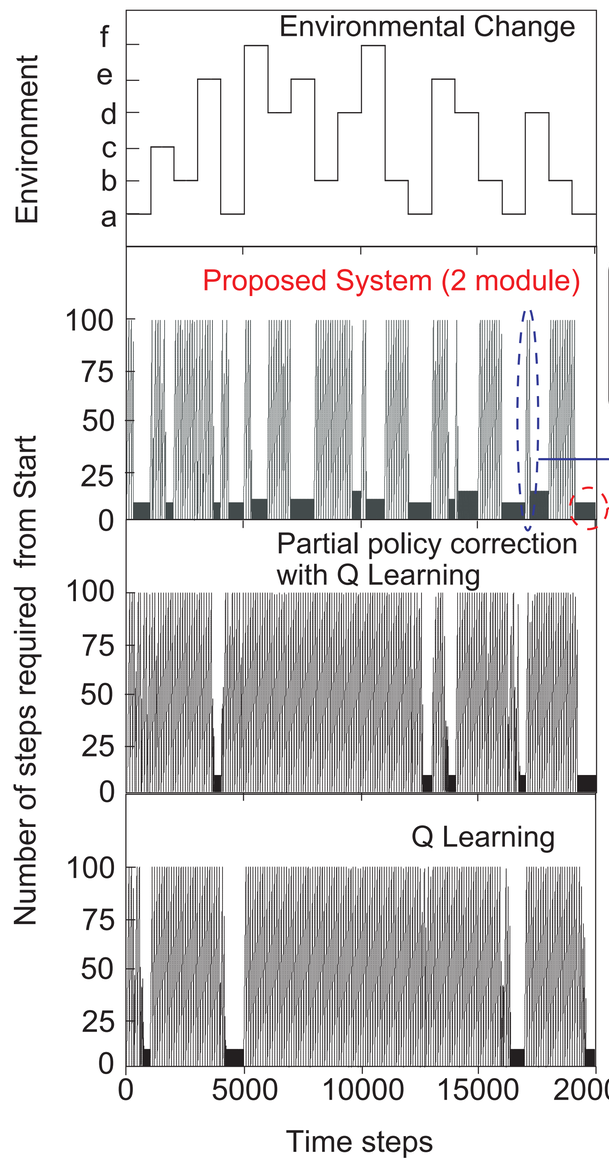
Restrictions:
- Situation:25 (5x5), Action:4(up, down, right, left)
- Environment changes randomly to one of six environments (Env.a - f) by **Nc** steps.
- If the robot does not reach to Goal in 100 steps, the trial is failure.
- If the robot reaches to Goal or the trial is failure, the robot is returned to Start.
- Reward: 1(Goal) 0 (except for Goal)



Result

1. Learning capability of the proposed system

■ Result of the simulation during 2000 steps
on condition that the environment changes by 1000 steps



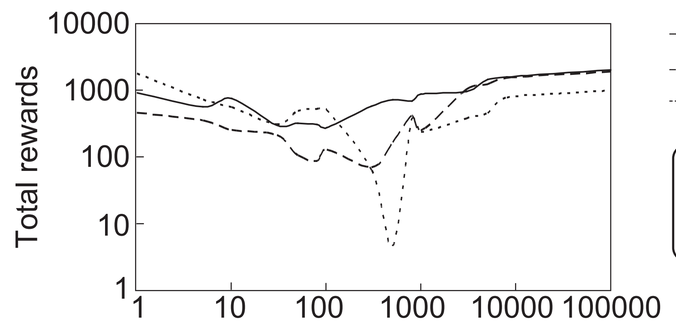
The proposed system constructed the multiple effective policies for almost all mazes except Env.b, with 2 modules(representation policies).

An effective policy is not constructed.

An effective policy is constructed.

Partial policy correction with Q Learning and Q Learning constructed the effective policy only for Env. a.

■ Result of the simulation on condition that the environment changes by Nc steps



The performance of the proposed system is better than the other methods by synthetic judgment.

2. Number of module and total data size

	Proposed system	Doya's system
Number of modules	2	6
Total data size	150	15000

Conclusions

We confirmed the effectiveness of the proposed system by the simulation experiment.
- **Learning is stable to environmental change.**
- **Memory is utilized effectively.**
  (Data size is **1/100** compared with conventional system.)

The future schedule: Implementation of this system to LSI