# 24

# A Reinforcement Learning Technique with an Adaptive Action Generator for a Multi-Robot System

Toshiyuki Yasuda and Kazuhiro Ohkura
*Hiroshima University*
*Japan*

## 1. Introduction

A robust instance-based reinforcement learning (RL) approach for controlling autonomous multi-robot systems (MRS) is introduced in this chapter. Although RL has been proven to be an effective approach for behavior acquisition for an autonomous robot, it generates considerably sensitive results for the segmentation of the state and action spaces. This problem can yield severe results with increase in the complexity of the system. When segmentation is inappropriate, RL often fails. Even if RL obtains successful results, the achieved behavior might not be sufficiently robust. In conventional RL, human designers segment the state and action spaces by using implicit knowledge based on their personal experience, because there are no guidelines for segmenting the state and action spaces.

Two main approaches for solving the abovementioned problem and for learning in a continuous space have been discussed. One of the methods applies function-approximation techniques such as artificial neural networks to the Q-function. Sutton (Sutton, 1996) used CMAC and Morimoto and Doya (Morimoto & Doya, 2000) used Gaussian softmax basis functions for function approximation. Lin represented the Q-function by using multi-layer neural networks called Q-net (Lin, 1993). However, these techniques have the inherent difficulty that a human designer must properly design their neural networks before executing RL. The other method involves the adaptive segmentation of the continuous state space according to the robots' experiences. Asada et al. proposed a state clustering method based on the Mahalanobis distance (Asada et al., 1996). Takahashi et al. used the nearest-neighbor method (Takahashi et al., 1996). However, these methods generally require large learning costs for tasks such as the continuous update of data classifications every time new data arrives.

Our research group has proposed an instance-based RL method called the continuous space classifier generator (CSCG), which proves to be effective for behavior acquisition (Svinin et al., 2000). We have also developed a second instance-based RL method called Bayesian-discrimination-function-based reinforcement learning (BRL) (Yasuda et al., 2005). Our preliminary experiments proved that BRL, by means of adaptive segmentation of state and action spaces, exhibits better performance as compared to CSCG.

As we mentioned in the previous chapter, BRL has an extended form that accelerates the learning speed (Yasuda & Ohkura, 2010). Our focal point for the extension is the process of action searching. In a standard BRL, a robot performs a random action and stores an input-

output pair as a new rule when it encounters a new situation. This random action sometimes produces one novel situation after another, which results in unstable behavior. In order to overcome this problem, we added a function that performs an action on the basis of acquired experiences. Our previous study demonstrated that MRSs that employ the extended BRL learn behaviors faster as compared to those that employ the standard BRL. In this chapter, we conduct further experiments in which a robot in an MRS is initialized after successful learning, and thus we investigate the robustness and relearning ability of the extended BRL.

The remainder of this chapter is organized as follows. In the next section, the target problem in this chapter and our concept of cooperative MRSs are introduced. Our design concept and the controller details are explained in Section 3. The results of our experiments are provided in Section 4. The conclusions are provided in Section 5.

## 2. Cooperative multi-robot systems

### 2.1 Cooperative transportation task

One of the challenging tasks in multi-robotics is object transportation. In this task, several robots move cooperatively to transport an object to a goal area in a static or dynamic environment. The object is sufficiently heavy and/or large so that no single robot can handle it.

Kosuge et al. adopted the feed back control method using the force sensors to achieve effective leading and following (Kosuge et al., 1997). Huntsberger et al. proposed a layered control architecture called CAMPOUT (Control Architecture for Multi-robot Planetary Outposts) that was tailored for extraterrestrial multi-robot systems (Huntsberger et al., 2004). CAMPOUT employs a leader-follower distributed control. Robots transport an object by lifting it in these approaches.

An alternative method of transporting object is pushing. Sen et al. used reinforcement learning techniques on a block pushing problem to show agents could learn coordinated behavior without any knowledge about each other (Sen et al., 1994). Kube and Zhang described a box-pushing task as a sequence of sub-tasks with separate controller designed for each step using finite state automata theory (Kube & Zhang, 1996). Here, 10 physical homogeneous robots achieved box-pushing without explicit communication. Parker proposed a behavior-based multi-robot architecture termed ALLIANCE that uses concepts of impatience/acquiescence to motivate behavior (Parker, 1998). ALLIANCE was validated in a pushing task by heterogeneous robots. Mataric et al. demonstrated box-pushing by two six-legged robots equipped with hand-coded sensing and behavior (Mataric et al., 1995). They demonstrated communication can produce performance improvements. They also developed that an auction-based task allocation system by using a publish/subscribe communication protocol (Gerkey et al., 2002). The system was validated in physical manipulation tasks by a watcher robot and two pusher robots. Wang and de Silva developed a controller comprised of reinforcement learning and genetic algorithms for object transportation by two robots (Wang & de Silva, 2008). A probabilistic arbitrator was used to select the winning output between reinforcement learning and genetic algorithms.

### 2.2 Autonomous specialization

MRSs aim to achieve *effective* cooperation by exploiting roles, behavior rules, or communication functions that are useful for the desired cooperation. However, it is

practically impossible to give the hand-crafted factors for all possible situations that a robot will encounter. This means that the performance of conventional human scripted manipulation is restricted in a given condition.

One approach to this problem is giving an ability to acquire cooperative behavior through experience to each robot by autonomous role development and assignment. This provides an MRS with the potential for system-level robustness, i.e., *generalization capability*. We call this particular ability *autonomous specialization* in this chapter. Recently, based on evolutionary robotics (Harvey et al., 1997; Nolfi & Floreano, 2000), Quinn et al. (Quinn et al., 2002) and Baldassarre et al. (Baldassarre et al., 2003) applied artificial evolution to realize this ideal function. There has been also significant progress in the field of swarm robotics (Sahin & Spears, 2005). Here, a swarm is as a kind of MRSs where many simple physical autonomous robots perform a task without any global central controller. The collective behavior emerges due to interactions between simple autonomous robots and an environment. The concept of swarm robotics however does not include the behavior-learning mechanisms. To the best of our knowledge, generally effective behavior-learning mechanisms for swarm robotics have not been proposed yet.

From the view point of autonomous specialization, a homogeneous MRS for a task that require appropriate cooperation is explained in this chapter. Reinforcement learning with some extensions is adopted as a behavior-learning mechanism.

## 3. Approach

### 3.1 BRL: RL in continuous learning space

Our approach, called BRL, updates the classification only when such an update is required.

A set of production rules is defined using Bayesian discrimination method, which is a well-known method of pattern classification (Dura & Hart, 1972). This method can assign an input, $X$, to the cluster, $C_i$, which has the largest posterior probability, max $\Pr(C_i|\boldsymbol{x})$. Here, $\Pr(C_i|\boldsymbol{x})$ indicates the probability calculated by Bayes' formula that a cluster, $C_i$, holds the observed input $\boldsymbol{x}$. Therefore, using this technique, a robot can select the most similar rule to the current sensory input. The learning procedure is overviewed as follows:

1. A robot perceives the current input data $\boldsymbol{x}$.
2. A robot selects the most similar rule from a rule set by using the Bayesian discrimination method. If a robot selects a rule, it executes the corresponding action $\boldsymbol{a}$. Otherwise, a robot executes an action randomly.
3. A robot is transferred to the next state and receives a reward $r$.
4. The utilities of all rules are updated according to $r$. The rules for which the utilities are below a certain threshold are removed.
5. The robot produces a new rule as the combination of the current input data and the executed action if a robot executed an action randomly. This executed rule is stored in the rule set.
6. Parameters of all the rules are updated by the interval estimation technique if a robot receives no penalty. Otherwise, a robot only updates the parameters of the selected rule.
7. Go to (1).

#### Action Selection and Rule Production

In BRL, a rule in the rule set is selected to minimize $g_i$, i.e. the risk of misclassification of the current input. We obtain $g$ on the basis of the the posterior probability $\Pr(C_i|\boldsymbol{x})$. $\Pr(C_i|\boldsymbol{x})$ is calculated as an indicator of classification for each cluster by using Bayes' Theorem:

$$\Pr(C_i \mid x) = \frac{\Pr(C_i)\Pr(x \mid C_i)}{\Pr(x)} \tag{1}$$

A rule cluster of $i$-th rule, $C_i$, is represented by a $v_i$-centered Gaussian distribution with covariance $\mathbf{\Sigma}_i$. Therefore, the probability density function of the $i$-th rule's cluster is represented by

$$\Pr(C_i \mid x) = \frac{1}{(2\pi)^{\frac{n_s}{2}} \mid \Sigma_i \mid^{\frac{1}{2}}} \cdot \exp\left\{\frac{-1}{2}(x - v_i)^{\mathrm{T}} \Sigma_i^{-1}(x - v_i)\right\} \tag{2}$$

A robot requires $g_i$ instead of calculating Pr $(C_i \mid x)$, because no one can correctly estimate $\Pr(x)$ in Eq.(1) (the higher the value of $\Pr(C_i \mid x)$, the lower is the value of $g_i$). A robot must select a rule on the basis of only the numerator. The value of $g_i$ is calculated as

$$\begin{aligned} g_i &= -\log\left\{f_i \cdot \Pr(C_i \mid x)\right\} \\ &= \frac{1}{2}(x - v_i)^{\mathrm{T}} \Sigma_i^{-1}(x - v_i) - \log\left\{\frac{1}{(2\pi)^{\frac{n_s}{2}} \mid \Sigma_i \mid^{\frac{1}{2}}}\right\} - \log f_i \end{aligned} \tag{3}$$

After calculating $g_i$ for all the rules, the winner rule, $rl_w$, is selected as that which has the minimal value of $g_i$. As mentioned in the learning procedure, the action in the $rl_w$ is performed if $g_i$ is lower than a threshold $g_{th}$ as shown in Fig. 1. Otherwise, a random action is performed.

### 3.2 Extended BRL with an adaptive action generator

#### Basic Concept

We have some RL approaches that provide learning in continuous action spaces. An actor-critic algorithm built with function approximators has a continuous learning space and

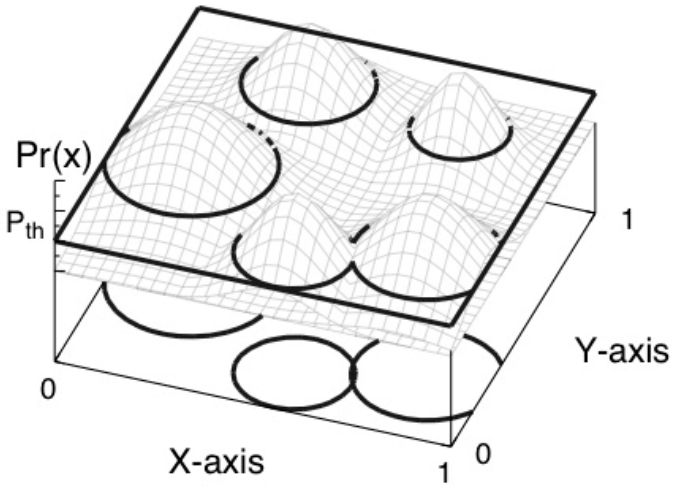



Fig. 1. State Space of the Standard BRL

modifies actions adaptively (Doya, 2000; Peters & Schaal, 2008). This algorithm modifies policies based on TD-error at every time step. Theoretically, the REINFORCE algorithm requires immediate rewards (Williams, 1992). These approaches are not useful for tasks such as transport tasks if a robot gets a reward only when the goal is achieved. However, BRL proves to be robust against a delayed reward.

In the standard BRL, a robot performs a random search in its action space; such random actions often resulted in instability in the global behavior of MRS in our preliminary experiments (Fig. 2). Therefore, reducing the chance of random actions may accelerate behavior acquisition and provide a more robust behavior. Instead of performing a random action, BRL requires a function that determines actions on the basis of acquired knowledge (Fig. 3).

**Adaptation Based on Acquired Knowledge**

To improve the search efficiency in a action space, in this paper, we introduce an extended BRL by modifying the learning procedure, Step (2) in the previous sub-section. In this extension, instead of a random action, the robot performs a knowledge-based action when it encounters a new environment. To do this, we set a new threshold, $P'_{th}$ ($< P_{th}$) as shown in Fig. 4, and provide three cases for rule selection in Step (2) as follows:

- $g_w < g_{th}$: The robot selects the rule with $g_w$ and executes its corresponding action $\boldsymbol{a}_w$.
- $g_{th} \leq g_w < g'_{th}$: The robot executes an action with parameters determined based on $rl_w$ and other rules with misclassification risks within this range as follows:

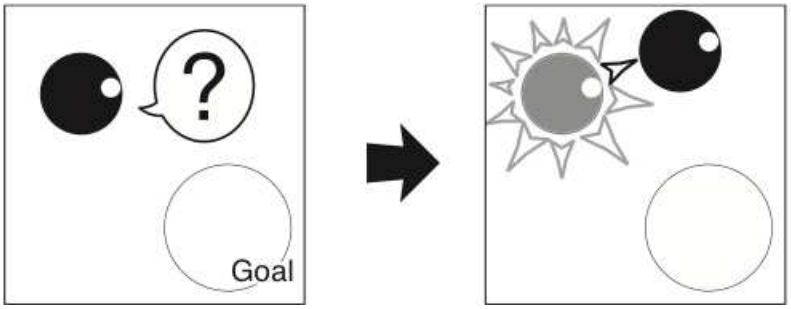



Fig. 2. BRL Robot that Executes a Random Action When it Encounters an Unknown Situation

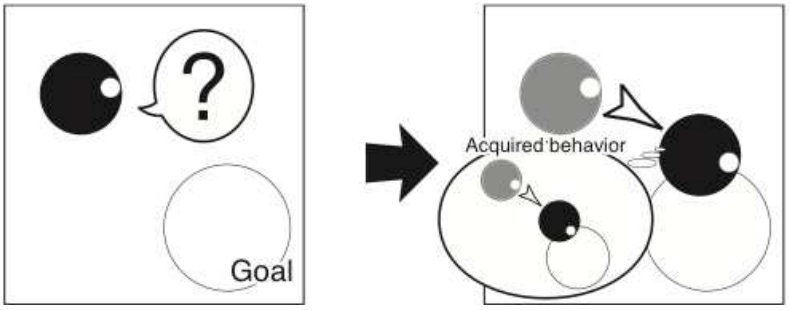



Fig. 3. Basic Idea of the Extended BRL; Generating an Action on the basis of Acquired Behavior

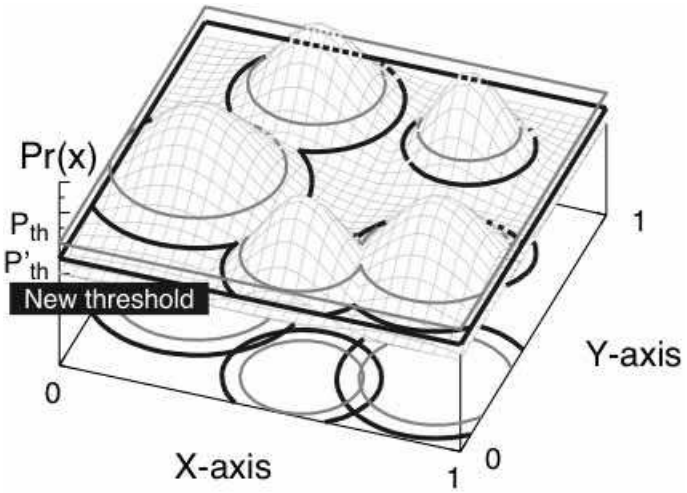



Fig. 4. State Space of the Extended BRL

$$a' = \sum_{l=1}^{n_r} \left( \frac{u_l}{\sum_{k=1}^{n_r} u_k} \cdot a_l \right) + N(0, \sigma), \tag{4}$$

where $n_r$ is the number of referred rules, and $N(0, \sigma)$ is a zero-centred Gaussian noise with variance σ. This action is regarded as an interpolation of previously-acquired knowledge.

- $g'_{th} \leq g_w$: The robot generates a random action.

In this rule selection, the first and third cases are the same as the standard BRL.

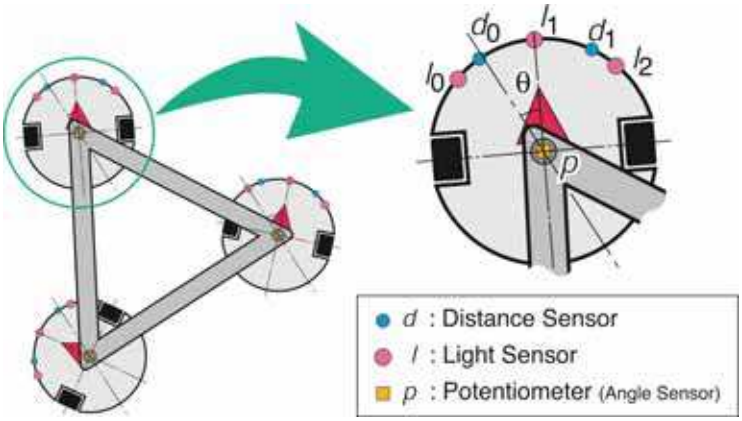
## 4. Experiments

### 4.1 Problem settings

Our target problem is a simple MRS composed of three autonomous robots, as shown in Fig. 5. This problem is called the *cooperative carrying problem* (CCP), and involves requiring the MRS to carry a triangular board from the start to the goal. A robot is connected to the different corners of the load so that it can rotate freely. A potentiometer measures the angle between the load and the robot's direction $\theta$. A robot can perceive the potentiometer measurements of the other robots, as well as its own. All three robots have the same specifications. Each robot has two distance sensors $d$ and three light sensors $l$. The greater $d / l$ becomes, the nearer the distance to an obstacle or a light source. Each robot has two motors for rotating two omnidirectional wheels (Fig. 6). A wheel provides powered drive in the direction it is pointing and passive coasting in an orthogonal direction at the same time.

The difficulties in this task can be summarized as follows:

- The robots have to cooperate with each other to move around.
- They begin with no predefined behavior rule sets or roles.
- They have no explicit communication functions.
- They cannot perceive the other robots through the distance sensors because the sensors do not have sufficient range.



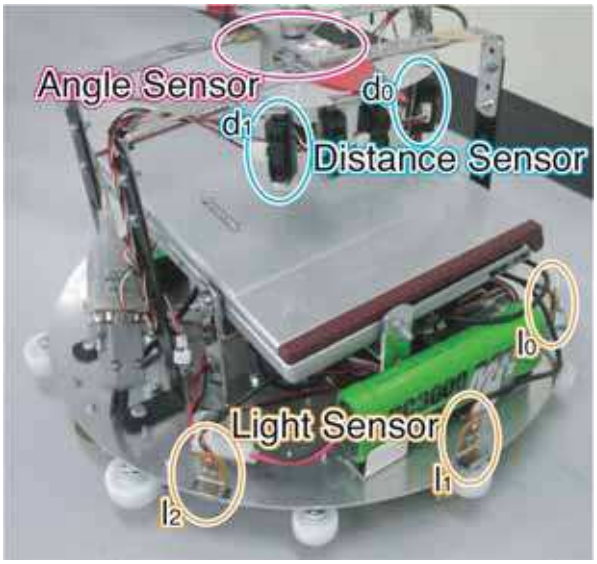




Fig. 5. Cooperative Carrying Task


Fig. 6. Ominidirectional Wheel

- Each robot can perceive the goal (the location of the light source) only when the light is within the range of its light sensors.

Passive coasting of the omnidirectional wheels brings a dynamic and uncertain state transition.

### 4.2 Experimental settings

Fig. 7 shows the general views of the experimental environments for simulation and physical experiments. In the simulation runs, the field is a square surrounded by a wall. The physical robots are situated in a 3.6-meter-long and 2.4-meter-wide pathway. The task for the MRS is to move from the start to the goal (light source). All robots get a positive reward when one of them reaches the goal ($l_0 > thr_{goal} \vee l_1 > thr_{goal} \vee l_2 > thr_{goal}$). A robot gets a negative reward when it collides with a wall ($d_0 > thr_d \vee d_1 > thr_d$). We represent a unit of time as a *step*. A step is a sequence that allows the three robots to get their own input information, make decisions by themselves, and execute their actions independently. When the MRS reaches the goal, or when it cannot reach the goal within 200 steps in simulations and 100 steps in physical experiments, it is put back to the start. This time span is called an *episode*.

The robot controller comprises a prediction mechanism and abehavior learning algorithm. The settings for these two mechanisms are as follows.

**Prediction Mechanism (NN)**

The prediction mechanism attached is a three-layered feed-forward neural network that performs back propagation. The input of *i*-th robot is a short history of sensory information, $\boldsymbol{I}^i = \{\cos\theta_{t-2}, \sin\theta_{t-2}, \cos\psi^i_{t-2}, \sin\psi^i_{t-2}, \cos\theta_{t-1}, \sin\theta_{t-1}, \cos\psi^i_{t-1}, \sin\psi^i_{t-1}, \cos\theta_t, \sin\theta_t, \cos\psi^i_t, \sin\psi^i_t\}$, where $\psi^i_t = (\theta_t + \theta^k_t)/2$ (i ≠ j ≠ k). The output is a prediction of the posture of the other robots at the next time step $\boldsymbol{O}^i = \{\cos\psi^i_{t+1}, \sin\psi^i_{t+1}\}$. The hidden layer has eight nodes.

**Behavior Learning Mechanism (BRL)**

The input is $\boldsymbol{x}^i = \{\cos\theta_t, \sin\theta_t, \cos\psi^i_{t+1}, \sin\psi^i_{t+1}, d^i_0, d^i_1, l^i_0, l^i_1, l^i_2\}$. The output is $\boldsymbol{a}^i = \{m^i_{rud}, m^i_{th}\}$, where $m^i_{rud}$ and $m^i_{th}$ are the motor commands for the rudder and the throttle respectively. σ in Eq.(9) is 0.05. For the standard BRL, $P_{th}$ = {0.012, 0.01}. For the extended BRL, $P_{th}$ = 0.012 and $P'_{th}$= 0.01. The other parameter values are the same as the recommended values in our previous chapter.

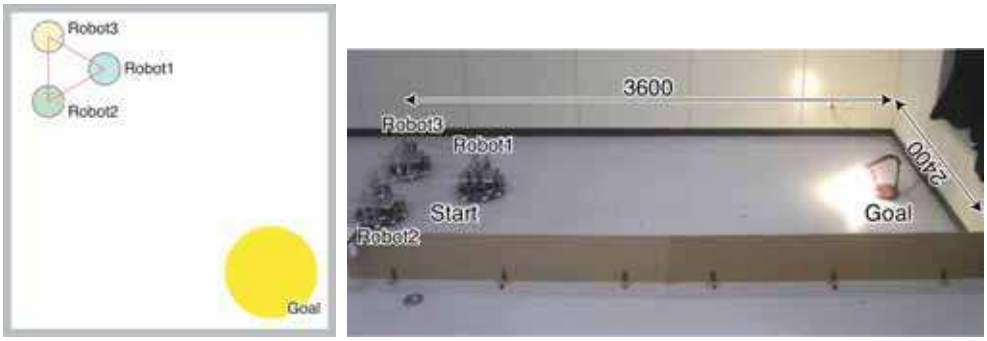



Fig. 7. Experimental Environment; (left) simulation, (right) physical experiment

We introduce a change in an environment by initializing one of the three robots. This may correspond to a situation in which a robot is replaced with a new one. Such changes occur when the MRS continuously reaches the goal for 100 consecutive episodes in the simulations and for 25 consecutive episodes in the physical experiments.

### 4.3 Results: simulations

We have investigated the improved performance of the extended BRL by means of three-/four-/five-robot CCP simulations in which robots must learn cooperative behavior from scratch in our previous work. In these experiments, we observed that robots always achieve cooperative behavior by developing team play organized by a leader, a sub-leader, and a follower. This implies that acquiring cooperative behavior always involves *autonomous specialization*.

The experiments in this section are conducted to observe the robustness of BRLs against a change in an environment. The MRS is disturbed in such a manner that one of the three robots is initialized immediately after a globally stable behavior is observed. Then, we count the number of episodes required for the MRS to relearn a new, stable behavior.

Mean performance for all 30 independent runs are illustrated in Fig. 8. The extended BRL, needs about twice as small number of the episodes as the standard BRL. On the other hand, Figs. 9-11 show the averages and the deviations in the number of episodes for the three roles of the initialized robot. For each roles, 10 independent runs are conducted. The difficulty in relearning is apparently different for each case. The most difficult cases are those in which the initialized robot is the leader of the team (Fig.9). If a leader robot is initialized, the robots require a large number of episodes to relearn a new, stable behavior; however, such cases show the largest difference among those employing BRLs. The extended BRL generates 50% better results as compared to the standard BRL. Since the acquired cooperative behavior possesses slight instability and the robots must coordinate their behaviors, particularly in a case in which a follower is initialized, the extended BRL provides a slightly worse result (Fig. 11). The improvement can be observed from the graphs for our proposed extensions. This implies that in terms of learning speed, the extended BRL outperforms the standard BRL.

### 4.4 Results: physical experiments

We conducted five independent experimental runs for each case employing the BRL. The standard BRL provided two successful results and theextended BRL provided four successful results from scratch (Yasuda & Ohkura, 2007).

Figs. 12-14 illustrate the learning results after one of the robots is initialized by using the best results in our preliminary experiments (Yasuda & Ohkura, 2007) for the standard and extended BRL. Before an environmental change, Robot1, Robot2, and Robot3 are the leader, sub-leader, and follower, respectively, in the experiments for both the BRLs. These figures illustrate the number of steps and punishments in each episode. Comparing these results shows that the extended BRL requires fewer episodes to newly develop a globally stable behavior. Similar to the simulation results, the case where a leader robot is initialized demonstrates the most significant difference. In this case, the standard BRL could not achieve a globally stable behavior and hence resulted in failure. In the other cases, the extended BRL required smaller number of episodes to relearn cooperative behavior. Further, the extended BRL is more stable than the standard BRL because the MRS with the standard BRL gets several punishments.

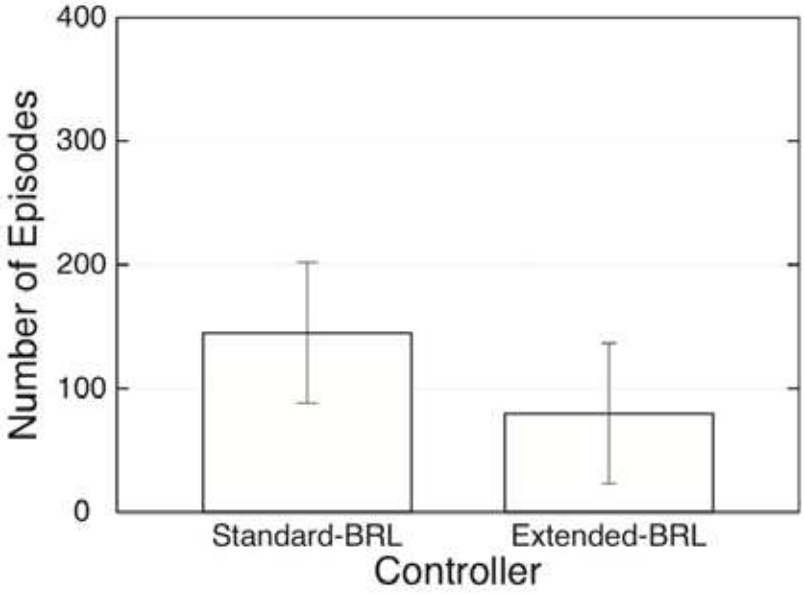



Fig. 8. Numbers of episodes required to relearn a behavior after an environmental change for all 30 independent runs.

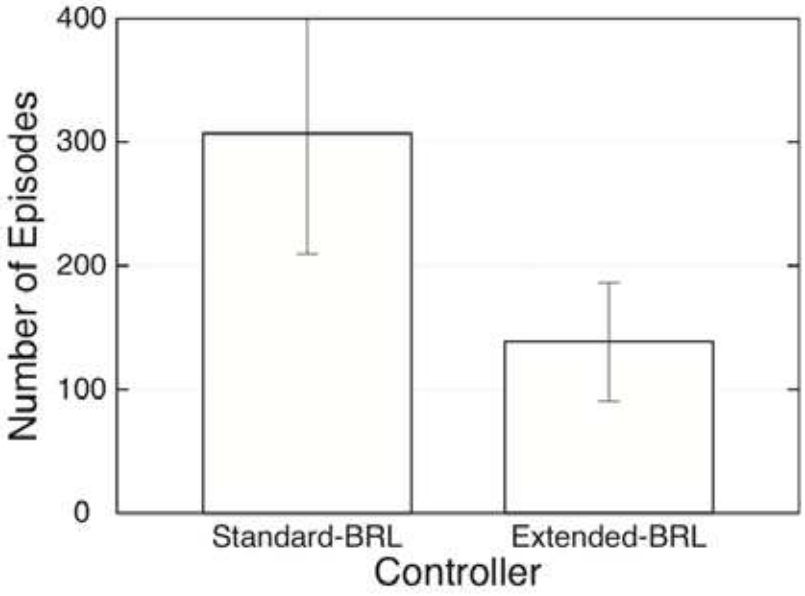



Fig. 9. Numbers of episodes required to relearn a behavior after an environmental change for 10 independent runs (a leader robot is initialized).

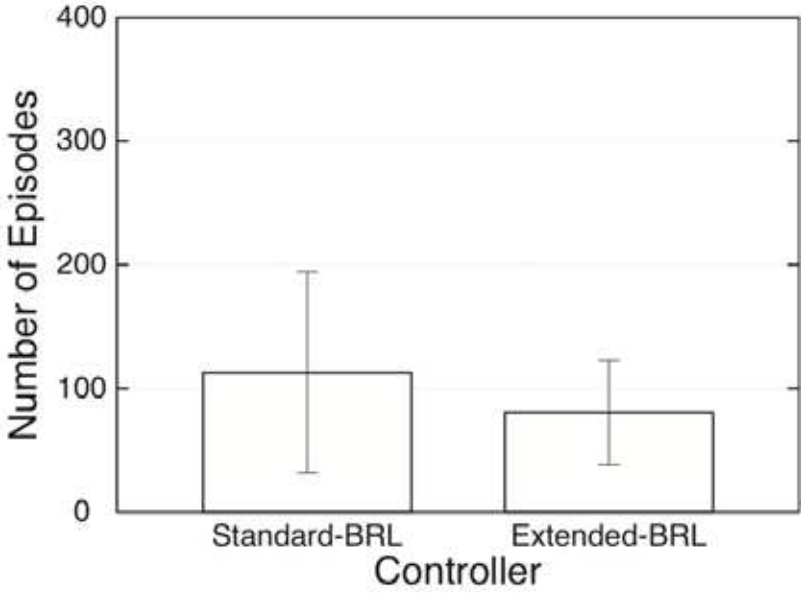



Fig. 10. Numbers of episodes required to relearn a behavior after an environmental change for 10 independent runs (a sub-leader robot is initialized).

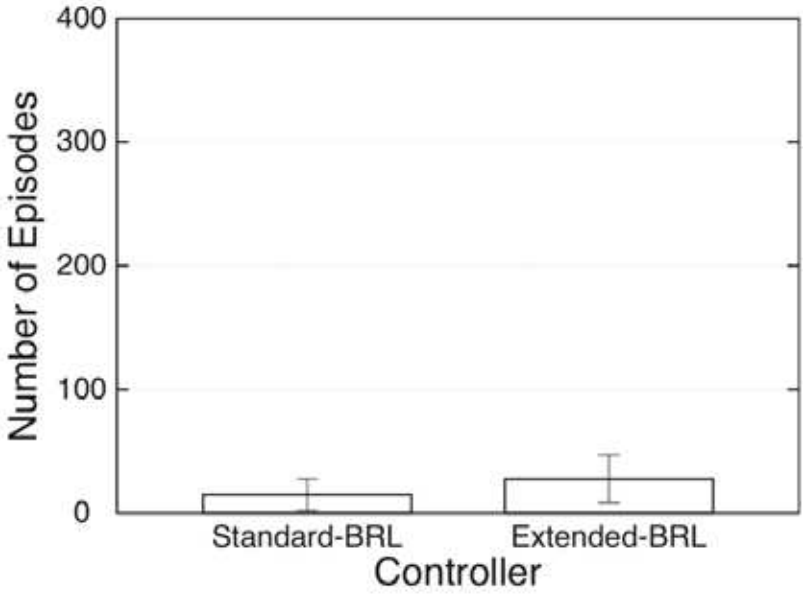
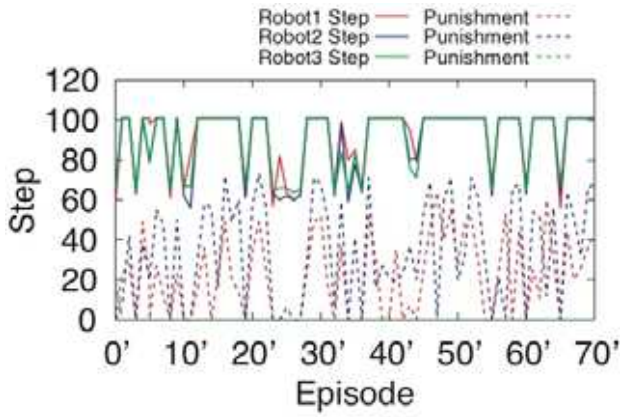
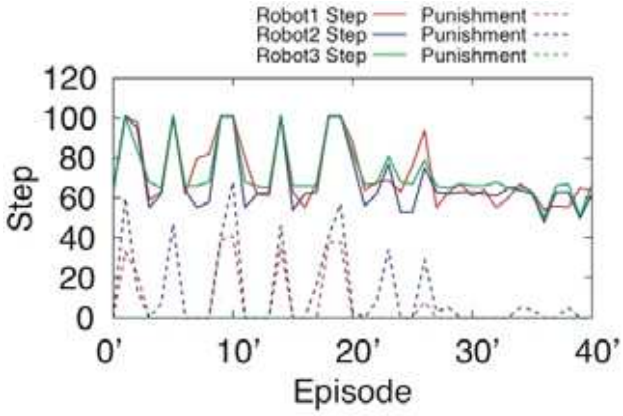



Fig. 11. Numbers of episodes required to relearn a behavior after an environmental change for 10 independent runs (a follower robot is initialized).




(a) Standard BRL

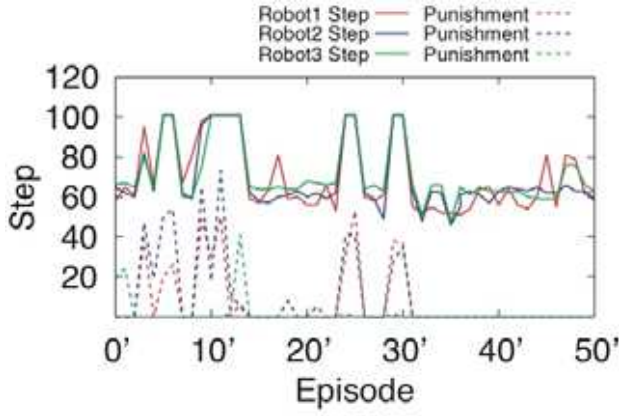



(b) Extended BRL

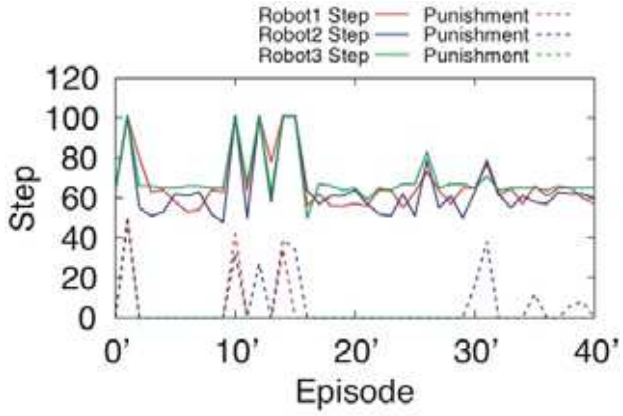
Fig. 12. Learning history after a leader is initialized

Fig. 15 shows examples of the stable behaviors acquired by the extended BRL, before and after Robot1 is initialized. Although an environmental change occurred for Robot2 and Robot3, the robots achieved a globally stable behavior similar to the behavior before initialization. The robots trooped right, left and right, and then reached the goal. By observing the rule parameters, we found that Robot1 learned to be another type of a leader and the other robots utilized some rules stored before initialization and the newly generated rules based on our extension.

Although parameters that are more refined might provide better performance, parameter tuning is outside the scope of this study because BRL is designed for acquiring a reasonable behavior as quickly as possible, rather than the optimal behavior. In other words, the focal point of our MRS controller is not optimality but versatility. In fact, we obtain similar experimental results through experiments with an arm-type MRS, similar to that in (Svinin et al., 2000), by using the same parameter settings.




(a) Standard BRL




(b) Extended BRL

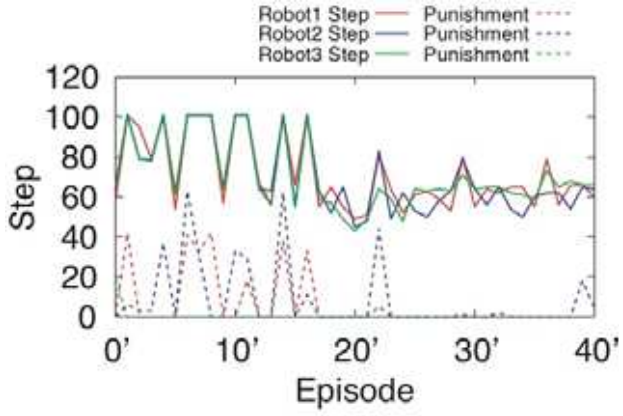
Fig. 13. Learning history after a sub-leader is initialized

## 5. Conclusion

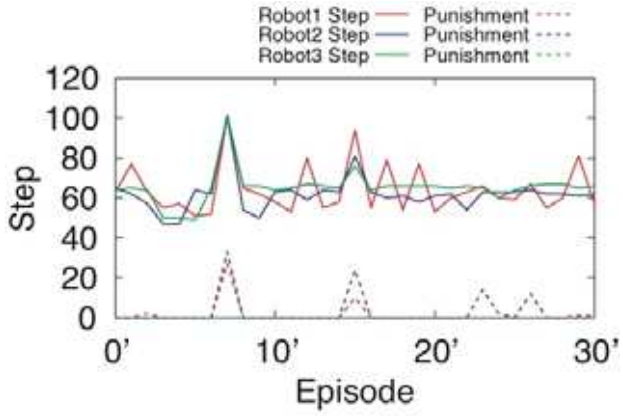
We investigated an RL approach for the behavior acquisition of an autonomous MRS. Our proposed RL technique, BRL, has a mechanism for the autonomous segmentation of the continuous learning space, and it proves to be effective for an MRS through autonomous specialization. For improving the robustness of an MRS, we proposed an extension of BRL by adding a function to generate interpolated actions based on previously acquired rules.

The results of the simulations and physical experiments demonstrated that the MRS with the extended BRL relearns behaviors faster than that with the standard BRL, after an environmental change.

In the future, we plan to analyze the learning process in detail. We also plan to increase the number of sensors and adopt other expensive sensors such as an omnidirectional camera that will allow a robot to incorporate a variety of information, and thereby acquire more sophisticated cooperative behavior in more complex environments.




(a) Standard BRL




(b) Extended BRL

Fig. 14. Learning history after a follower is initialized

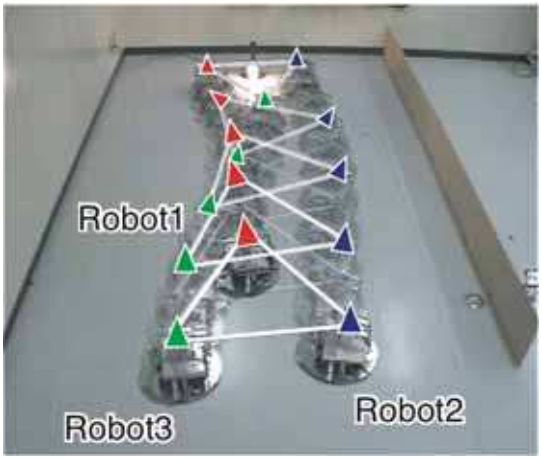
## 6. References

Sutton, R.S. (1996). Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding. *Advances in Neural Information Processing Systems*, 8, pp. 1038-1044, MIT Press

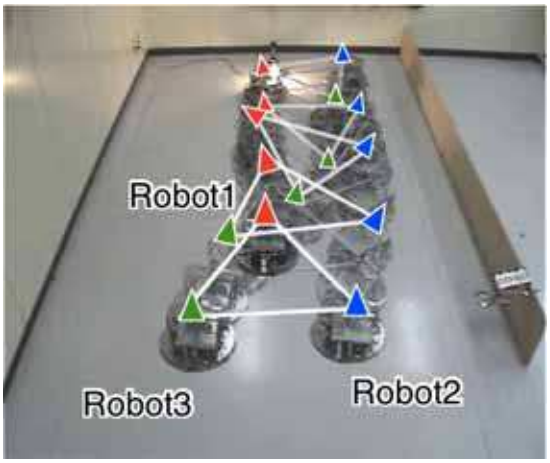
Morimoto, J. & Doya, K. (2000). Acquisition of Stand-Up Behavior by a Real Robot using Hierarchical Reinforcement Learning for Motion Learning: Learning ``Stand Up'' Trajectories, *Proceedings of International Conference on Machine learning*, pp. 623-630

Lin, L.J. (1993). Scaling Up Reinforcement Learning for Robot Control, *Proceedings of the 10th International Conference on Machine Learning*, pp. 182-189

Asada, M., Noda, S. & Hosoda, K. (1996). Action-Based Sensor Space Categorization for Robot Learning, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1502-1509




(a) Before Initializing Robot1




(b) After Successful Relearning

Fig. 15. Acquired Behavior by the Extended BRL

Takahashi, Y., Asada, M., & Hosoda, K. (1996). Reasonable Performance in Less Learning Time by Real Robot Based on Incremental State Space Segmentation, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1518-1524

Svinin, M., Kojima, F., Katada, Y., and Ueda, K. (2000) Initial Experiments on Reinforcement Learning Control of Cooperative Manipulations, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 416-422

Yasuda, T. and Ohkura, K. (2005) Autonomous Role Assignment in Homogeneous Multi-Robot Systems. Journal of Robotics and Mechatronics, 17, 5, pp. 596-604

Kosuge, K., Oosumi, T. & Chiba, K. (1997). Load Sharing of Decentralized-Controlled Multiple Mobile Robots Handling a Single Object, *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, pp.3373--3378

Hunstsburger, T.L., Trebi-Ollennu, A., Aghazarian, H. & Schenker, P.S. (2004). Distributed Control of Multi-Robot Systems Engaged in Tightly Coupled Tasks, *Autonomous Robotics*, 17, pp.79-92

Sen, I., Sekaran, M. & Hale, J. (1994). Learning to Coordinate without sharing information, *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pp.426-431

Kube, C., Zhang, H. (1996) The use of perceptual cues in multi-robot box-pushing, *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pp. 2085-2090

Parker, L. (1998) Alliance: an architecture for fault tolerant multirobot cooperation, *IEEE Transaction on Robotics and Automation*, 14, 2, pp.220-240

Mataric, M.J., Nilsson, M. & Simsarian, K. (1995). Cooperative Multi-Robot Box-Pushing, *Proceedings of International Conference on Intelligent Robots and Systems*, pp.556-561

Gerkey, B.P. & Mataric, M.J. (2002), Sold!: Auction methods for multi-robot coordination, *IEEE Transactions on Robotics and Automation, special issue on Advances in Multi-Robot Systems*, 18, 5, pp.758-786

Wang, Y. & de Silva, C.W. (2008). A machine-learning approach to multi-robot coordination, *Engineering Applications of Artificial Intelligence*, 21, pp.470-487

Harvey, I., Husbands, P., Cliff, D., Thompson, A., & Jakobi, N. (1997). Evolutionary Robotics: the Sussex Approach, *Robotics and Autonomous Systems*, 20, pp.205-224

Nolfi, S. & Floreano, D. (2000), Evolutionary Robotics, MIT Press

Quinn, M., Smith, L., Mayley, G. & Husbands, P. (2002). Evolving Team Behavior for Real Robots, *Proceedings of EPSRC/BBSRC International Workshop on Biologically-Inspired Robotics*, pp.217-224

Baldassarre, G, Nolfi, S. & Parisi, D. (2003), Evolution of collective behaviour in a team of physically linked robots, *Applications of Evolutionary Computing*, pp.581-592

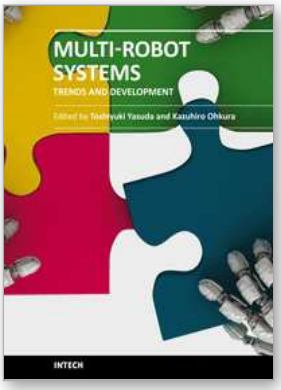
Sahin, E. & Spears, W.M. (eds.) (2005). *Swarm Robotics*, LNCS 3342, Springer

Duda, R.O. &Hart, P.E. (1972). *Pattern Classification and Scene Analysis*, Wiley-Interscience, N.Y.

Doya, K. (2000). Reinforcement Learning in Continuous Time and Space, *Neural Computation*, 12, 219-245

Peters, J. & Schaal, S. (2008). Natural actor critic, *Neurocomputing*, 71, 7-9, pp.1180-1190

Williams, R.J. (1992). Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning, *Machine Learning*, 8, pp. 229-256




**Multi-Robot Systems, Trends and Development**

Edited by Dr Toshiyuki Yasuda

ISBN 978-953-307-425-2
Hard cover, 586 pages
**Publisher** InTech
**Published online** 30, January, 2011
**Published in print edition** January, 2011

This book is a collection of 29 excellent works and comprised of three sections: task oriented approach, bio inspired approach, and modeling/design. In the first section, applications on formation, localization/mapping, and planning are introduced. The second section is on behavior-based approach by means of artificial intelligence techniques. The last section includes research articles on development of architectures and control systems.

**How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Kazuhiro Ohkura and Toshiyuki Yasuda (2011). A Reinforcement Learning Technique with an Adaptive Action Generator for a Multi-Robot System, Multi-Robot Systems, Trends and Development, Dr Toshiyuki Yasuda (Ed.), ISBN: 978-953-307-425-2, InTech, Available from: http://www.intechopen.com/books/multi-robot-systems-trends-and-development/a-reinforcement-learning-technique-with-an-adaptive-action-generator-for-a-multi-robot-system

INTECH
open science | open minds

**InTech Europe**
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

**InTech China**
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821