23

# Improving Search Efficiency in the Action Space of an Instance-Based Reinforcement Learning Technique for Multi-Robot Systems

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

## 1. Introduction

Recent years have witnessed growing interest in multi-robot system (MRS) research. To date, numerous research projects have been undertaken in various forms, such as robot soccer (Stone & Sutton, 2001), all-terrain operation (Mondada et al., 2003), box-pushing problems (Gerkey & Mataric, 2002), and many others. We can point out at least three advantages of an MRS over traditional single-robot systems (Stone & Veloso, 2000). The first is *parallel processing*, performed by autonomous and asynchronous robots in the system. The second is *robustness*, realized by redundancy: the system has more robots than required. The third is *scalability* in the sense that a robot can be added or removed from the system easily.

From the viewpoint of complex adaptive systems, it is important to coordinate cooperative behavior to solve a given task because a task is given simply to a robot group without sufficiently detailed specifications to solve it. The most popular approach to realize coordination is providing strategies for effective cooperation in advance in the form of behavior rules, roles, or communication protocols.

However, it is practically impossible to give hand-crafted behavior rules for all possible situations that a robot will encounter. This means that the performance is context-sensitive. One approach to this problem is giving the ability of acquiring cooperative behavior through experience to each robot by autonomous role development and assignment so that an MRS has the potential for system-level robustness.

We consider that a key factor would be how to give an *on-line* autonomous specialization mechanism to an MRS. This study introduces an approach that uses reinforcement learning (RL) to achieve autonomous specialization. To date, RL has not often been applied to an MRS because of the following two reasons. The first is that RL generates quite sensitive results for segmentation of the state space and the action space. When segmentation is inappropriate, RL often fails. Even if RL obtains a successful result, the achieved behavior might not be sufficiently robust. The second is that the RL theory is constructed on the assumption of a static environment (Sutton & Barto, 1998). Therefore, RL in a simple form can yield good results only when the environment is sufficiently static or stable for a robot to be able to assume that it is static.

We must therefore apply RL carefully to an MRS, so that learning robots cope with the dynamics in their environment resulting from the moves of other robots that are learning simultaneously.

To overcome these problems, we apply a novel RL algorithm that has a mechanism for segmenting continuous state space and continuous action space autonomously and simultaneously. We call this Bayesian-discrimination-function-based Reinforcement Learning (BRL). In addition, for supporting the stabilization of the dynamics in the learning problem for the RL, complementary information, *i.e.*, the prediction of the other robots' postures at the next time step is provided to the BRL by a learning neural network.
The remainder of this chapter is organized as follows: The target MRS is introduced in the second section. The third and fourth sections explain our design concept and our reinforcement learning controller details. The fourth section proposes an extended BRL for improving the robustness. The fifth section shows results of our experiments. Conclusions are given in the final section.

## 2. Task: cooperative carrying problem

Our target problem is a simple MRS composed of three autonomous robots, as shown in Fig. 1. 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

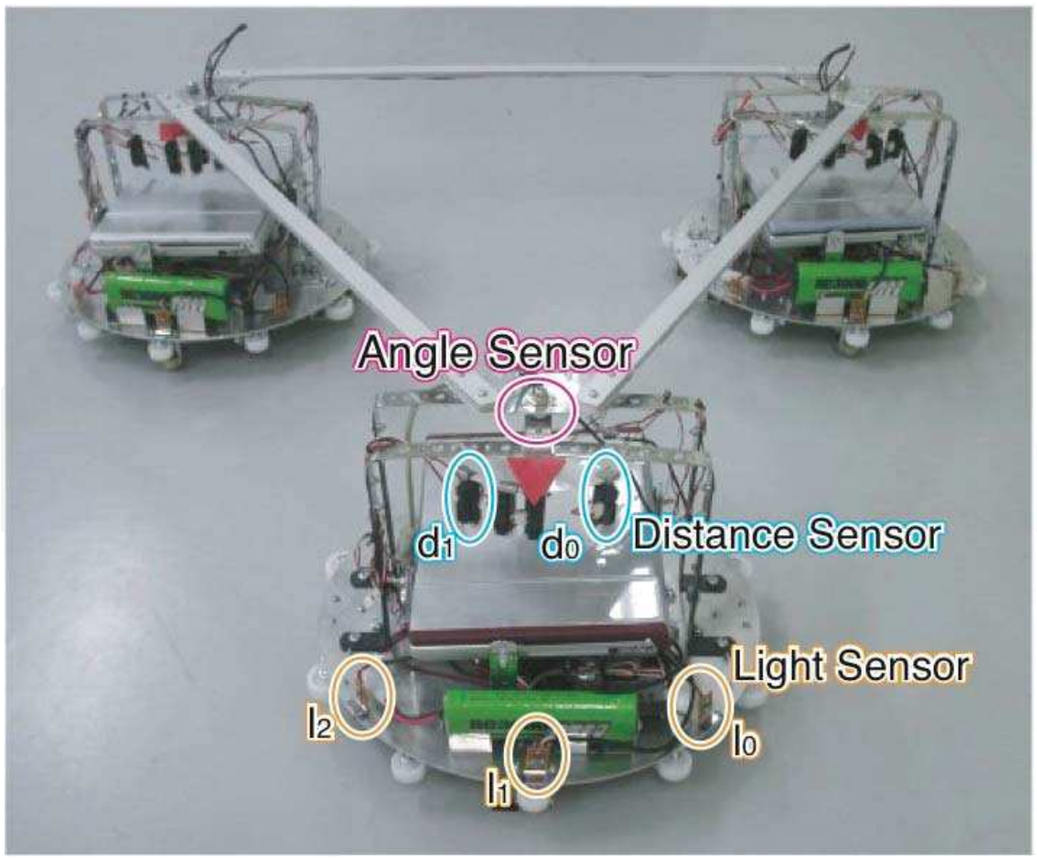



Fig. 1. Cooperative carrying problem, CCP

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. 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.
- 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.

## 3. Reinforcement learning approach to CCP

### 3.1 Reinforcement learning in continuous space:

**BRL: Overview**

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).

**Rule Representation**

The BRL operates on a set of rules $R$. A rule $rl \in R$ is defined as $rl$:=< $\boldsymbol{v}$, $u$, $\boldsymbol{a}$, $f$, $\boldsymbol{\Sigma}$, $\boldsymbol{\Phi}$>. In this expression, the state vector associated with $rl$ is $\boldsymbol{v} = \{v_1,\ldots,v_{nd}\}^{\mathrm{T}}$, where $n_d$ is the number of inputs. The utility of $rl$ is represented as $u$. The action vector is $\boldsymbol{a} = \{a_1,\ldots,a_{na}\}^{T}$, where $n_a$ is the number of actuators. The prior probability is denoted as $f$. The covariance matrix is $\boldsymbol{\Sigma}$ = diag $\{\sigma_1,\ldots,\sigma_{\mathrm{nd}}\}$. The sample set associated with $rl$ is $\Phi = \{\phi_1,\ldots, \phi_{ns}\}^{\mathrm{T}}$, where $n_s$ is the number of samples.

**Action Selection**

A rule in R is selected to minimize the risk of misclassification of the current input. The posterior probability $\Pr(C_i \mid \boldsymbol{x})$ is calculated as the risk of misclassification for each cluster; it is calculated by Bayes' Theorem:

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

For finding the minimal risk, it is sufficient to calculate the posterior probability because all clusters have a common factor of $1/\Pr(\boldsymbol{x})$. The probability density function of the $i$-th rule's cluster is represented as the following.

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

The estimated value of $g_i$, the risk of misclassification of the input data $\boldsymbol{x}$ into the other clusters, is calculated as the following:

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

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 $g_{th}$. Otherwise, a random action is performed.

**Temporal Credit Assignment**

The respective utilities of the rules are updated using the following four strategies after the action is performed.

1. *Direct payoff distribution:* The direct payoff $P$ is given to the winner rule. Two types of payoff are obtainable: reward ($P$>0) and punishment ($P$<0). The payoff is spread back along the sequence of the rules that triggered its actions with the discount rate γ.
2. *"Bucket brigade" like strategy:* The current winner rule, $rl_w$, hands over part of its utility $\Delta u$ to the previous winner only when $\Delta u$ is positive.
3. *Taxation:* A firing rule reduces its utility as $u_w \leftarrow (1 - c_f)\, u_w$.
4. *Evaporation:* All rules reduce their utilities at the evaporation rate $\eta < 1$ when the robot reaches the goal: $u_w \leftarrow \eta u_w$. A rule that has smaller utility than the threshold $u_{min}$ is removed from the rule set $R$.

**Updating Rule Set**

The update phase is performed except when action by $rl_w$ results in punishment. If a random action is taken (*i.e.* $g_w > g_{th}$), a new rule that is composed of the current sensory input, $\boldsymbol{v}_c$, and the executed action, $\boldsymbol{a}_c$, is added to $R$. Parameters for the new rule are defined as follows.

$$\boldsymbol{v}_c = \boldsymbol{x},\ \boldsymbol{\Sigma}_c = \sigma_0^2\mathbf{I},\ \boldsymbol{a}_c = \boldsymbol{a}_w,\ u_c = u_0,\ f_c = f_0 \tag{4}$$

In those equations, $\sigma_0$, $u_0$ and $f_0$ are constants, **I** is a unit matrix. When the action in $rl_w$ is performed as (*i.e.* $g_w \leq g_{th}$), all of its parameters are updated as follows. First, the sample set $\mathbf{\Phi}_w$ is updated by adding the current sensory input to $\boldsymbol{x}$. Then, the sample mean $\boldsymbol{x} = \{x_1,..., x_{ns}\}^T$ and the sample variance $\boldsymbol{s}^2 = \{s_1^2,..., s_{ns}^2\}^T$ are estimated from the updated set $\mathbf{\Phi}_w$. The confidence intervals for X and $s^2$ are also updated. Subsequently, BRL determines whether any component of $\boldsymbol{v}$ and $\mathbf{\Sigma}$ is out of the range of the confidence intervals. If any component is outside of that range, the updates are conducted:

$$v_i \leftarrow v_i + \alpha(x_i - v_i) \tag{5}$$

$$\sigma_i^2 \leftarrow \sigma_i^2 + \alpha[s_i^2 - \sigma_i^2] \tag{6}$$

$$f_w \leftarrow f_w + \beta(1 - f_w) \tag{7}$$

where $\alpha$ and $\beta$ are constants. For all other rules, the prior probabilities $f_i$ are updated as follows:

$$f_i \leftarrow (1 - \beta)f_i \tag{8}$$

### 3.2 Reducing the dynamics in an environment

**Related Work**

To date, numerous reports that are related to the RL approach and that are applied to an MRS have been published. For instance, Tan (Tan, 1993), who examined the effects of sharing information, described that shared information is beneficial if it can be used efficiently. Asada et al. (Asada et al., 1999) and Ikenoue et al. (Ikenoue et al. 2002) proposed a vision-based RL method for acquiring cooperative behavior in a soccer-like game that includes two mobile robots: a shooter and a passer. To stabilize the learning process, Asada et al. introduced a method of global scheduling by limiting the number of learning agents to one and allowing the remaining agents to execute fixed policies that were acquired in the previous learning stage. Ikenoue et al. proposed a method of asynchronous policy renewal with one policy and one action value function. Elfwing et al. (Elfwing et al., 2004) added macro actions. Macro actions force an agent to execute the same primitive action for more than one time step to thereby stabilize learning and make action selection more predictable for other agents. Several studies have specifically addressed the internal model of other learning agents (Littman, 1994; Hu & Wellman, 1998; Nagayuki et al., 2000). In those models, agents learn through estimating others' actions, Q values, or policies.
To the best of our knowledge, no RL approaches have displayed autonomous specialization. Therefore, robots need well-designed states, actions, strategies, or roles for acquiring cooperative behavior. Achieving all of these goals simultaneously is a practical impossibility.

**Our Approach**

In this study, we adopt a mechanism for predicting the near-future state based on time-series sensory information. As related work, a memory-based method (Moore & Atkenson, 1993) and a decision-tree (Suzuki et al., 1999) have been proposed for dealing with non-Markovian characteristics in an MRS environment. However, the state space is expanded according to the length of the time series information; in the worst case, it is expanded indefinitely.

We consider that the state space expansion should be as little as possible.
Our research group has demonstrated that merely the nearest future state prediction is sufficient for stabilizing the dynamics in an RL space (Kawakami et al., 1999). In this study, although a continuous learning space is assumed, an identical approach is examined with a feed-forward neural network for predicting the average of the other robots' postures at the next time step. As shown in Fig. 2, BRL uses the output of the neural network as a sensory information input.

## 4. Extended BRL

### 4.1 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 modifies actions adaptively (Doya, 2000; Peters & Schaal, 2008). This algorithm modifies policies based on TD-error at every time step. The REINFORCE algorithm theoretically also needs immediate reward (Williams, 1992). These approaches are not useful for tasks such as the navigation problem shown in Sec. 2, because the robot gets a reward only when it reaches the goal. BRL, however, proves to be robust against a delayed reward.
In the standard BRL, a robot performs a random search in its action space, and these random actions can produce unstable behavior. Therefore, reducing the chance of random actions may accelerate behavior acquisition and provide more robust behavior. Instead of performing a random action, BRL needs a function that determines action based on acquired knowledge.

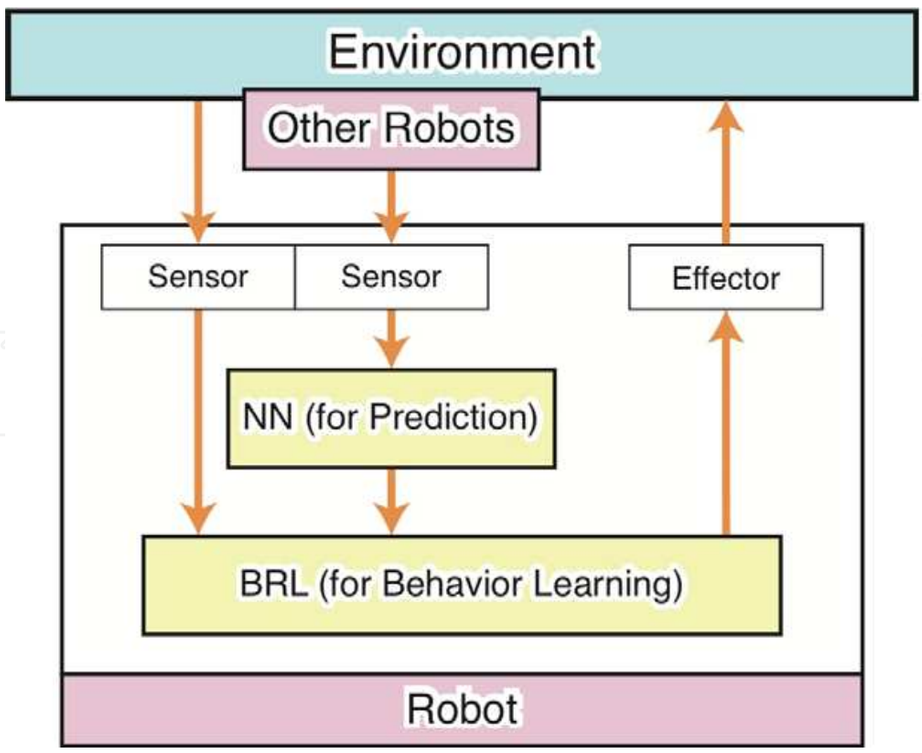



Fig. 2. Robot Controller

### 4.2 BRL with an adaptive action generator

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 Sec. 3. 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, $P'_{th}$ ($< P_{th}$), 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:

$$a' = \sum_{l=1}^{n_r} \left( \frac{u_l}{\sum_{k=1}^{n_r} u_k} \cdot a_l \right) + N(0, \sigma), \tag{9}$$

  where $n_r$ is the number of referred rules, and $N(0, \sigma)$ is a zero-centred Gaussian noise with variance $\sigma$. 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.

## 5. Experiments

### 5.1 Settings

Fig. 3 and 4 show the general views of the experimental environments for simulation and physical experiments, respectively. 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 settings of the robot controller 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. $\sigma$ 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 parameters are shown in Table. 1. These values are the same as the recommended values in our journal (Yasuda et al., 2005).

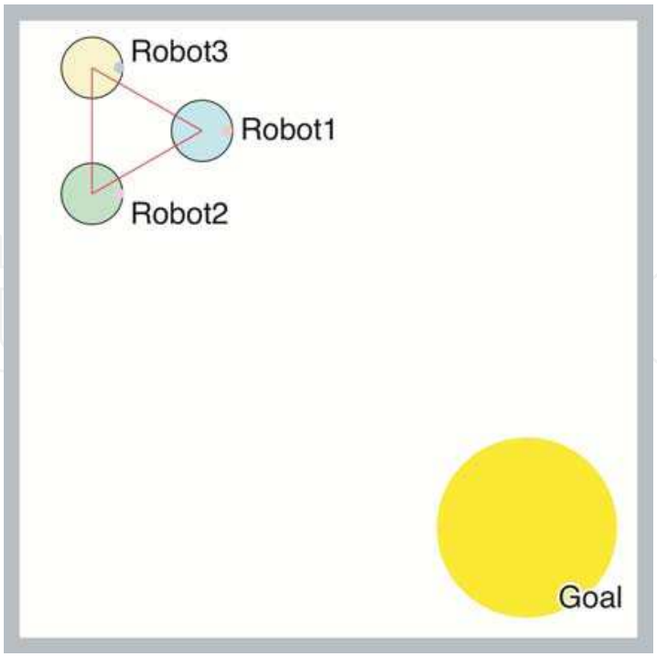



Fig. 3. Experimental environment (simulation)

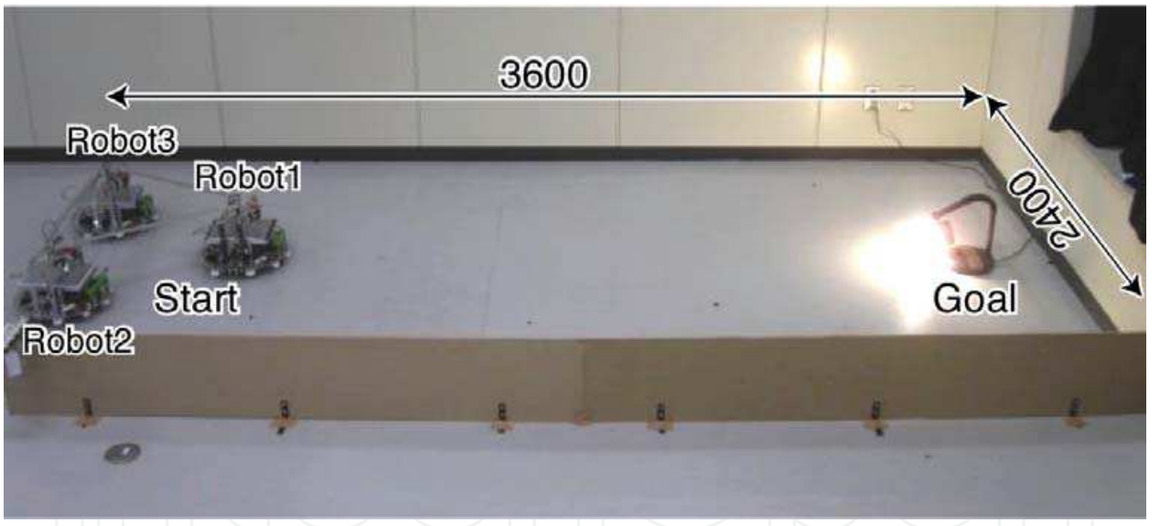



Fig. 4. Experimental environment (physical experiment)

### 5.2 Results: simulations

Fig. 5 shows the averages and the deviations of steps that the MRS takes by the end of each episode. In the early stages, the MRS requires a lot of trial and error and takes many steps to finish the episode. After such a trial and error process, the behavior of MRS becomes more stable and it takes fewer steps. An MRS with the standard BRL stably achieves the task within nearly constant steps after the 250th episode, and the extended BRL accomplishes this in 200 episodes. This means that, in terms of learning speed, the extended BRL outperforms the standard one.

| Parameter | | Value |
|---|---|---|
| $n_{rl}^{max}$ | maximum size of the rules | 100 |
| $n_s^{max}$ | maximum size of the samples | 50 |
| $P$ | payoff (reward) | 25.0 |
| $P$ | payoff (punishment) | -0.05$u$ |
| $u_0$ | initial utility | 10.0 |
| $u_{min}$ | threshold for extinction | 9.2 |
| $c_f$ | cost for an action | 0.01 |
| $\gamma$ | distribution rate of utility | 0.9 |
| $\kappa$ | utility spread rate | 0.15 |
| $\eta$ | evaporation rate | 0.98 |
| $f_0$ | initial prior probability | 0.001 |
| $\sigma_0$ | initial variance | 0.05 |
| $\alpha$ | in eq (6) | 0.001 |
| $\beta$ | in eqs (7) and (8) | 0.0001 |

Table 1. BRL parameters

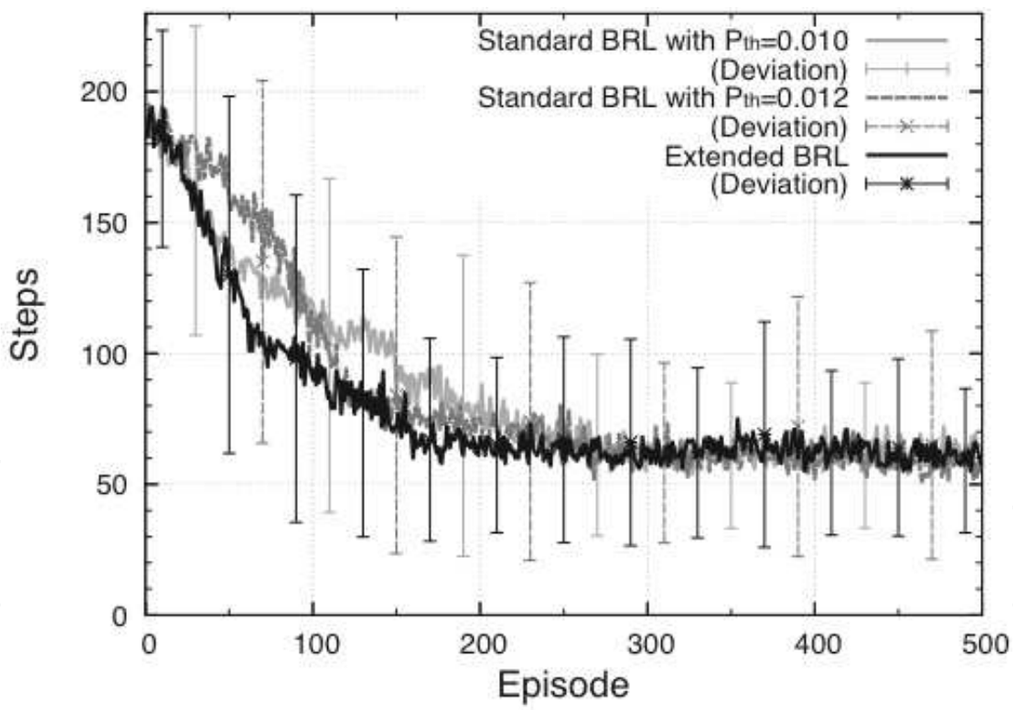



Fig. 5. Mean learning history for 50 simulations (three robots)

For the 50 independent runs, the MRS achieved different globally stable behavior as shown in Fig. 6. However, we found a common point that robots always achieved cooperative behavior by developing team play organised by a leader, a sub-leader and a follower. This implies that acquiring cooperative behavior always involved autonomous specialization. The extended BRL displayed higher adaptability, and yielded autonomous specialization faster than the standard BRL.

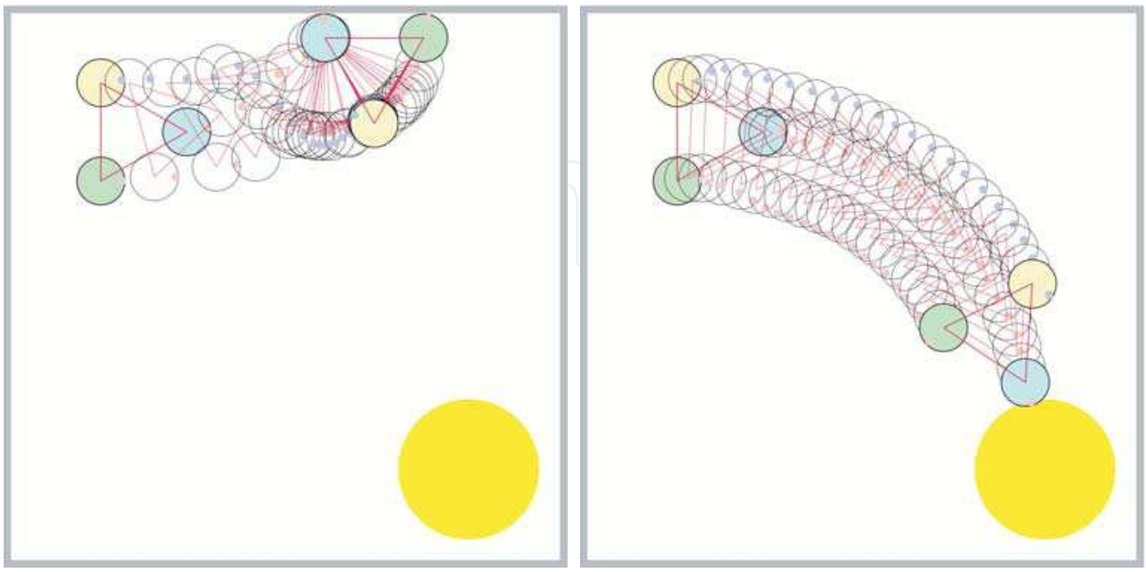


Fig. 6. Typical behavior in the early stage and acquired stable behavior (three robots)

**Discussion**

There is no significant difference in results in the learning performance of the BRLs for a three-robot CCP; therefore, we tested four- and five-robot CCP performance for more dynamic and complicated problems. The four robots use a square load, and the five robots have a pentagonal load. In these CCPs, $\psi$ is the average of the angles between two neighbouring robots and the load. The other controller settings are the same as those for the three-robot CCP.

Figs. 7 and 8 show the average and the deviations of steps an MRS takes by the end of each episode. As the number of robots increases, we can find that the extended BRL provides increasingly better results than the standard BRL, although it requires more episodes before obtaining stable behavior as shown in Figs. 9 and 10. The extended BRL has a function for coordinating behavior as well as reducing the number of random actions that can result in unstable behavior. These results show that the extended BRL has a higher learning ability and is less dependent on the number of robots in the MRS. This implies that the extended BRL might have more scalability, which is one of the advantages of MRS over single-robot systems.

Although parameters that are more refined might provide better performance, parameter tuning is outside the scope, because BRL is designed for acquiring reasonable behavior as quickly as possible, rather than 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) using the same parameter settings.

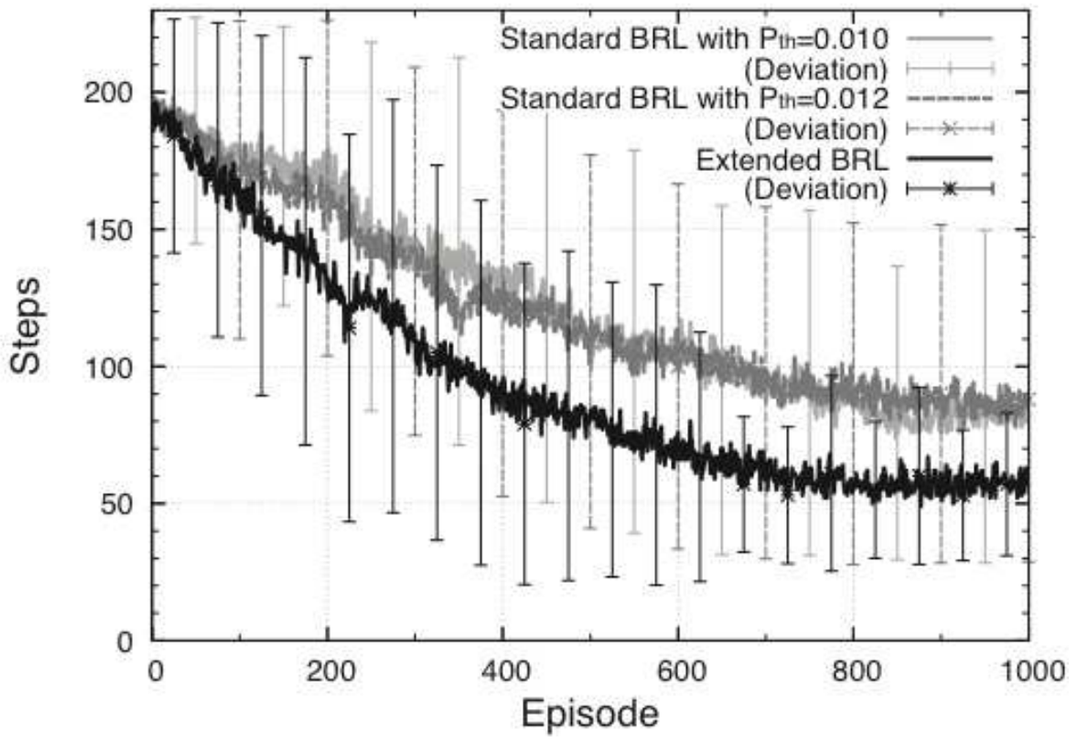



Fig. 7. Mean learning history for 50 simulations (four robots)

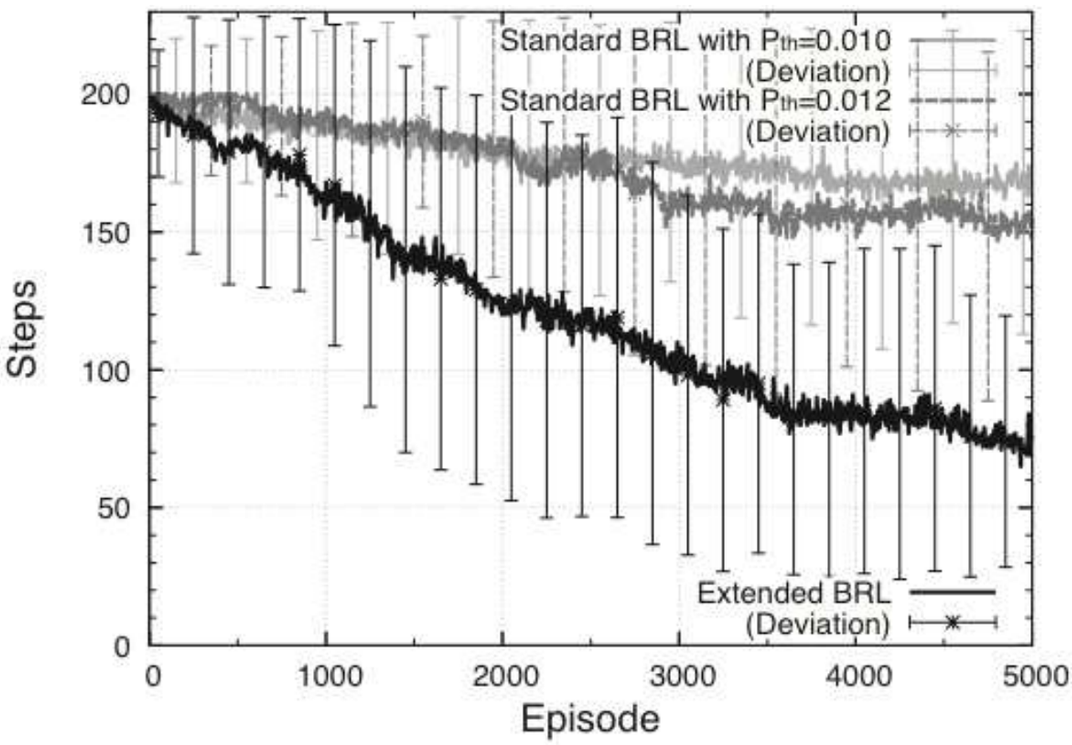



Fig. 8. Mean learning history for 50 simulations (five robots)

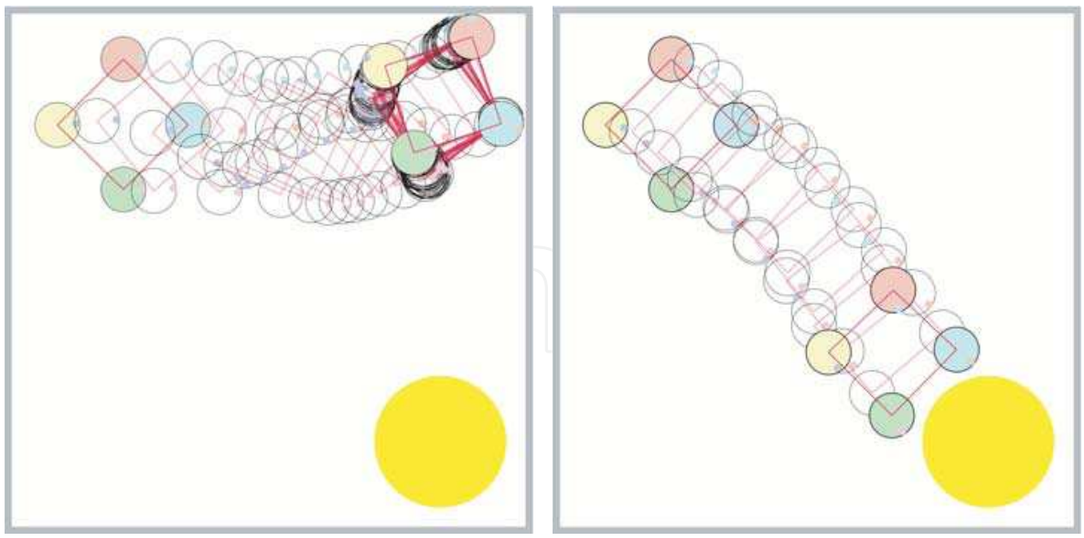


Fig. 9. Typical behavior in the early stage and acquired stable behavior (four robots)

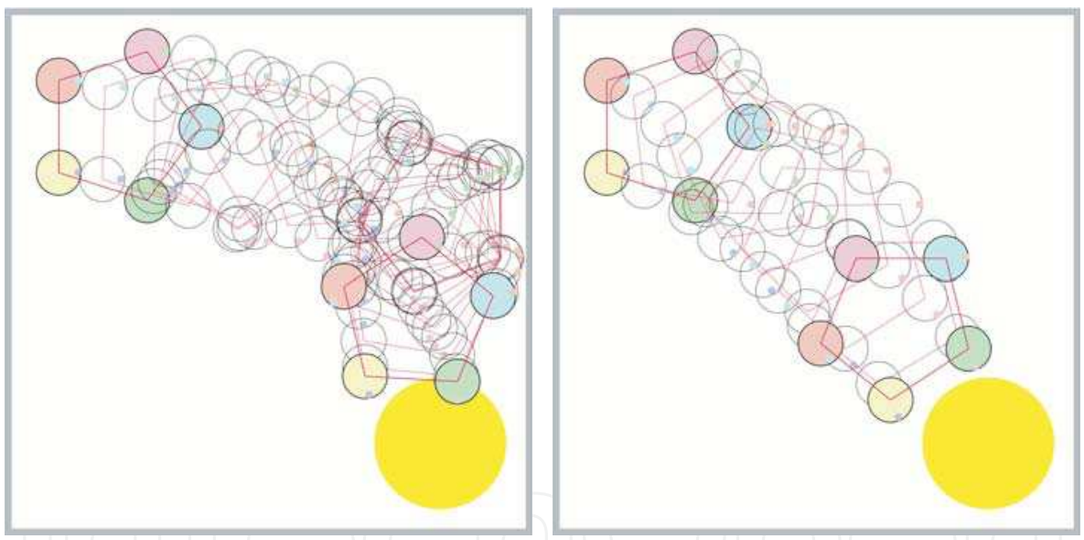


Fig. 10. Typical behavior in the early stage and acquired stable behavior (five robots)

### 5.3 Results: physical experiments

We conducted five independent experimental runs for each BRL. The standard BRL provided two successful results and the extended BRL provided four. Figs. 11 and 12 illustrate the best results of the physical experiments by the standard and the extende BRL, respectively. These figures illustrate the number of steps and punishments in each episode. Comparing these results shows that the extended BRL requires fewer episodes to learn behavior. The other successful results of the extended BRL show better performance than the best result of the standard BRL. The behavior of the extended BRL is also more stable than that of the standard, because the MRS with the standard BRL gets several punishments after learning goal-reaching behavior.

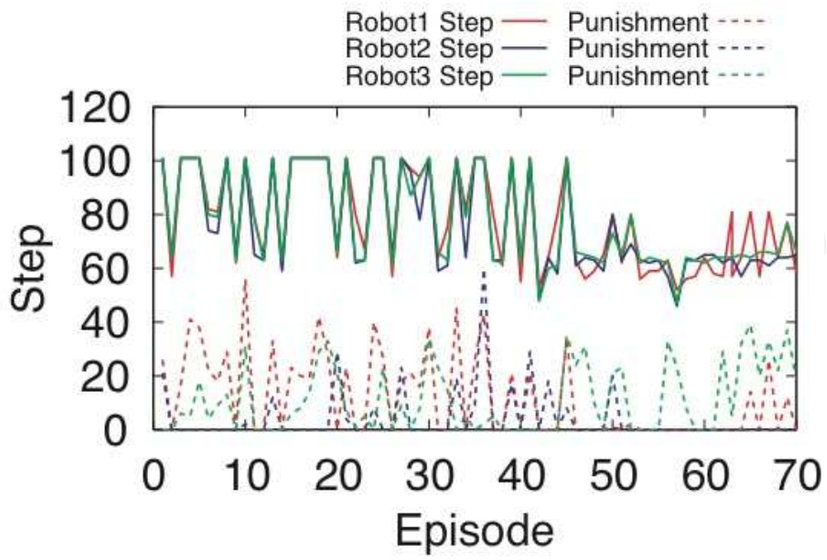



Fig. 11. Learning history: physical experiment (standard BRL)

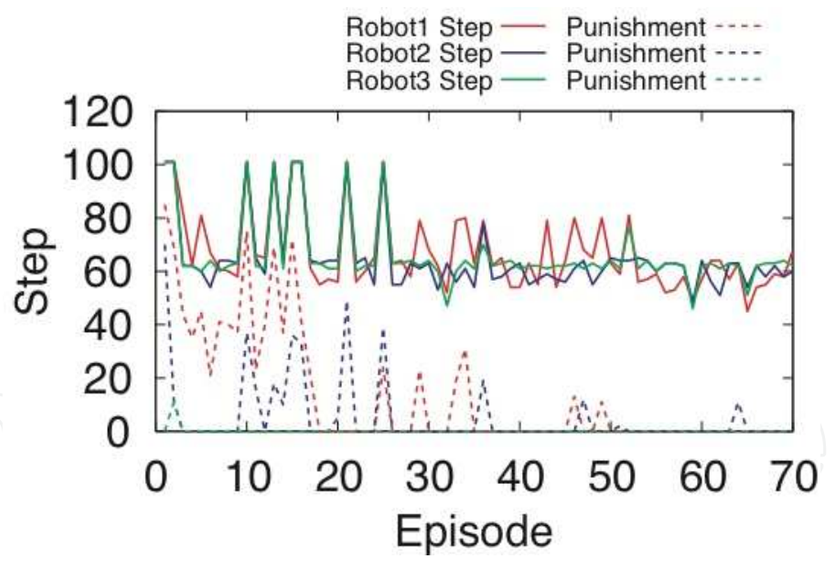



Fig. 12. Learning history: physical experiment (extended BRL)

Figs. 13 and 14 show examples of the behavior of the extended BRL. In the early stages, robots have no knowledge and function by trial and error. During this process, robots often collide with a wall and become immovable (Fig. 13). Then, some robots reach the goal and develop appropriate input-output mappings (Fig. 14). Observing the acquired behavior and investigating rule parameters, we found that the robots developed cooperative behavior, based on autonomous specialization.

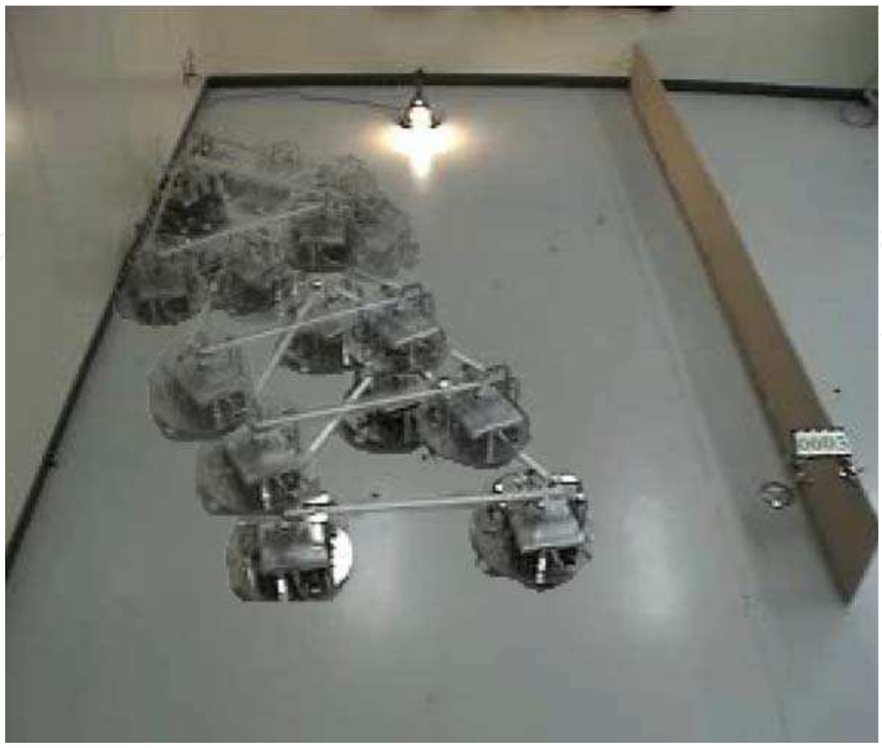


Fig. 13. An example of behavior in the early stage (extended BRL)

Fig. 14. An example of acquired behavior after successful learning (extended BRL)

## 6. Conclusions

We investigated the RL approach for the behavior acquisition of autonomous MRS. Our proposed RL technique, BRL, has a mechanism for autonomous segmentation of the continuous learning space, and proved effective for MRS through the emergence of autonomous specialization. For accelerated learning, we proposed an extension of BRL with a function to generate interpolated actions based on previously acquired rules. Results of the simulations and physical experiments showed that the MRS with an extended BRL did learn behavior faster than that with the standard BRL.

## 7. References

Stone, P. & Sutton, R.S. (2001). Scaling Reinforcement Learning toward RoboCup Soccer, *Proc. of the 18th International Conference on Machine Learning*, pp.537-544

Mondada, F., Guignard. A., Bonani, M., Floreano, D., Bar, D., & Lauria, M. (2003). SWARM-BOT: From Concept to Implementation, *Proc. of IEEE/RSJ International Conference on Intelligent Robot and Systems*, pp.1626-1631

Gerkey, B. & Mataric, M.J. (2002). Pusher-Watcher: An Approach to Fault-Tolerant Tightly Coupled Robot Coordination, *Proc. of IEEE International Conference on Robotics and Automation*, pp.464-469

Stone, P. & Veloso, M. (2000). Multiagent systems: survey from a machine learning perspective, *Autonomous Robots*, 8(3): pp. 345-383

Sutton R.S. & Barto, A.G. (1998). *Reinforcement Learning: An Introduction*, MIT Press

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

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

Asada, M., Uchibe, E. &Hosoda, K. (1999). Cooperative Behavior Acquisition for Mobile Robots in Dynamically Changing Real Worlds via Vision-Based Reinforcement Learning and Development, *Artificial Intelligence*, 110, pp.275-292

Ikenoue, S., Asada, M., & Hosoda, K. (2002). Cooperative Behavior Acquisition by Asynchronous Policy Renewal that Enables Simultaneous Learning in Multiagent Environment, *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.2728-2734

Elfwing, S., Uchibe, E., Doya, K. & Christensen, H.I. (2004). Multi-Agent Reinforcement Learning: Using Macro Actions to Learn a Mating Task'', *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.3164-3169

Littman, M.L. (1994). Markov Games as a Framework for Multi-Agent Reinforcement Learning, *Proc. of Eleventh International Conference on Machine Learning*, pp.157-163

Hu, J. & Wellman, M.P. (1998). Multiagent Reinforcement Learning: Theoretical Framework and an Algorithm, Proc. of Fifteenth International Conference on Machine Learning, pp.242-250

Nagayuki, Y., Ishii, S. & Doya, K. (2000). Multi-Agent Reinforcement Learning: An Approach Based on the Other Agent's Internal Model, *Proc. of Fourth International Conference on Multi-Agent Systems*, pp.215-221

Moore, A.W. & Atkeson, C.G. (1993). Memory-Based Reinforcement Learning: Converging with Less Data and Less Real Time, *Machine Learning*, 13, pp.103-130

Suzuki, S., Tamura, T., & Asada, M. (1999). Learning from conceptual aliasing caused by direct teaching, *Proc. of the IEEE International Conference on Systems, Man, and Cybernetics*, pp.698-703

Kawakami, K., Ohkura, K., & Ueda, K. (1999) Adaptive Role Development in a Homogeneous Connected Robot Group, *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, 3, pp. 251-256

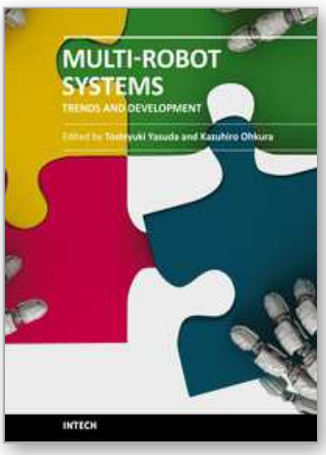
Doya, K. (2000). Reinforcement Learning in Continuous Time and Space, *Neural Computation*, 12, 219-245

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

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

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




**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). Improving Search Efficiency in the Action Space of an Instance-Based Reinforcement Learning Technique for Multi-Robot Systems, 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/improving-search-efficiency-in-the-action-space-of-an-instance-based-reinforcement-learning-techniqu

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

IntechOpen