Instance-Based Reinforcement Learning Technique with a Meta-learning Mechanism for Robust Multi-Robot Systems

Toshiyuki Yasuda, Motohiro Wada, and Kazuhiro Ohkura

Hiroshima University, Higashi-Hiroshima, Japan
{yasuda,wada,ohkura}@ohk.hiroshima-u.ac.jp

Abstract. In recent years, the subject of learning autonomous robots has been widely discussed. Reinforcement learning (RL) is a popular method in this domain. However, its performance is quite sensitive to the discretization of state and action spaces. To overcome this problem, we have developed a new technique called Bayesian-discrimination-function-based RL (BRL). BRL has proven to be more effective than other standard RL algorithms in dealing with multi-robot system (MRS) problems. However, similar to most learning systems, BRL occasionally suffers from overfitting. This paper introduces an extension of BRL for improving the robustness of MRSs. Meta-learning based on the information entropy of firing rules is adopted for adaptively modifying its learning parameters. Physical experiments are conducted to verify the effectiveness of our proposed method.

Keywords: multi-robot system, cooperation, robustness, reinforcement learning, meta-learning.

1 Introduction

Multi robot systems (MRSs) have recently attracted considerable attention from roboticists as these offer the possibility of accomplishing a task that a single robot can not. A robot team may provide redundancy and perform assigned tasks in a more reliable, faster, or cheaper way.

Reinforcement learning (RL) [1] is a frequently used method in the problem domain of learning autonomous robots. Because of the progress in RL research, a mobile robot can acquire appropriate behaviour such as obstacle-avoidance, wall-following or goal-reaching by interaction with an embedded environment. However, the results obtained from single robot systems are not directly applicable to MRSs, mainly because of the following two reasons. First, although RL is quite sensitive to how discretization is performed, its simple form assumes that learning space should be discretized before learning starts. Second, RL assumes a static environment, whereas a robot in an MRS is surrounded by an intrinsically nonstationary environment because of other robots that are learning simultaneously. Nevertheless, RL is often applied to MRS problems successfully because it allows for dynamics up to a certain level in an embedded environment.

R. Groß et al. (Eds.): TAROS 2011, LNAI 6856, pp. 161–172, 2011.

Several approaches for solving these problems and for learning in a continuous space have been discussed. A popular method applies function approximation techniques such as artificial neural networks to Q-function. Sutton [2] used Cerebellar Model Articulatory Controller (CMAC) and Morimoto and Doya [3] used Gaussian softmax basis functions for function approximation. Lin represented the Q-function by using multi-layer neural networks called *Q-net* [4]. However, these techniques have an inherent difficulty that a human designer must properly design their neural networks before executing RL. The idea of dimension reduction has been adopted[5,6]. The basic idea of this approach is to explicitly use a simpler representation of data by projecting it to lower dimensional spaces.

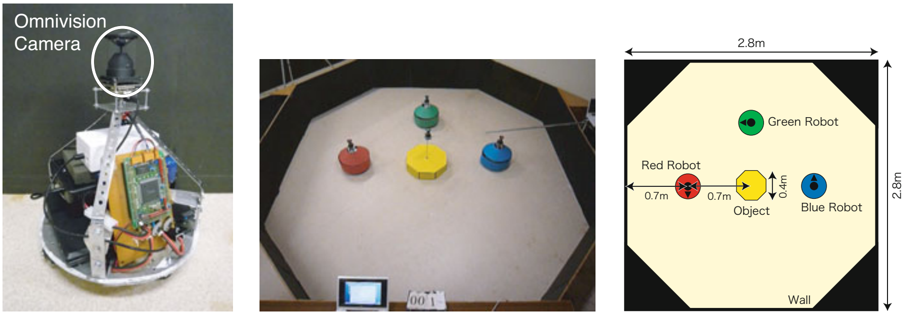
Other methods involve the adaptive segmentation of the continuous state space according to the robots' experiences. Asada *et al.* proposed a state clustering method based on a Mahalanobis distance [7]. Takahashi *et al.* used a nearest neighbour method [8]. However, these methods generally require large learning costs for tasks such as the continuous update of data classifications every time new data arrives. Actor-critic algorithms built with function approximators have a continuous learning space and adaptively modify actions [9,10]. These algorithms modify policies based on a temporal difference (TD) error at each time step.

We have previously proposed an instance-based RL method called Bayesian-discrimination-function-based RL (BRL) [11,12,13]. Our preliminary experiments illustrated that BRL exhibits better performance compared with CSCG through the adaptive discretization of state and action spaces. BRL has also proved to be robust against the dynamics in an environment that contains multiple robots. However, similar to other learning systems, we have occasionally observed overfitting problems in BRL. Overfitting is a problem such that a learning robot gradually becomes less robust after stable behaviour is acquired. We consider that this brittleness is critical for MRSs because robots must be essentially situated in a nonstationary environment, which is generally unpredictable. In this paper, an extension of BRL is proposed to overcome the abovementioned problem. The basic idea is that meta-parameters such as learning rate are adaptively coordinated so that each robot modifies its behaviour on the basis of the stability of its own actions.

The rest of this paper is organised as follows. The target problem is introduced in Section 2. The details of BRL and its extensions are explained in Section 3. The results of our experiments are described in Section 4. The conclusions are provided in Section 5.

2 Learning Task

Our target task is a simple MRS consisting of three autonomous mobile robots shown in Fig. 1. This problem is called the *object-orbiting task* and involves requiring the MRS to avoid collision with a wall and to move counterclockwise in a field.



(a) Robot

(b) Initial settings of the experimental environment

Fig. 1. Object-orbiting task

All the robots have the same specifications; each robot is 35 cm in height and 28 cm in length (diameter). They have an omnivision camera at the centre of their body. A robot can detect an object, a wall and the nearest robot. Each robot has three motors for rotating two omnidirectional wheels. A wheel simultaneously provides powered drive in the direction in which it is pointing and passive coasting in an orthogonal direction.

The difficulties in this task can be summarised as follows:

- The robots must cooperate with each other to achieve the given task.
- They begin with no predefined behaviour rule sets or roles.
- They have no explicit communication functions.

3 Extended BRL

3.1 BRL: RL in Continuous Learning Space

Overview. Our approach, called BRL, adaptively updates classifications on the basis of interval estimation, only when such an update is required. In BRL, the state space is covered by multivariate normal distributions, each of which represents a rule cluster C_i. A set of production rules is defined by Bayesian discrimination. This method can assign an input $\boldsymbol{x}$ 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, by using this technique, a robot can select a rule that is most similar to the current sensory input. In BRL, production rules are associated with clusters segmented by Bayes boundaries. Each rule contains a state vector $\boldsymbol{v}$, an action vector $\boldsymbol{a}$, a utility u and parameters for calculating the posterior probability, *i.e.* a prior probability f, a covariance matrix $\boldsymbol{\Sigma}$ and a sample set Φ.

The learning procedure is as follows:

(1) A robot perceives the current sensory input $\boldsymbol{x}$.
(2) By using Bayesian discrimination, the robot selects the most similar rule from a rule set R. If a rule is selected, the robot executes the corresponding action $\boldsymbol{a}$, otherwise, it performs a new action.
(3) The robot transfers to the next state and receives a reward r.
(4) All the rule utilities are updated according to r. The rules with utility below a certain threshold are removed.
(5) When the robot performs a new action, it produces a new rule by combining the current sensory input and the executed action. This executed new rule is memorised in the rule set R.
(6) If the robot receives no penalty, an interval estimation technique updates the parameters of all the rules. Otherwise, the robot updates only the parameters of the selected rule.
(7) Go to (1).

Action Selection and Rule Production. In BRL, a rule in the rule set R is selected to minimise a discrimination function g. We obtain g on the basis of the posterior probability $\Pr(C_i|\boldsymbol{x})$, which is calculated as an indicator of the classification for each cluster by using Bayes' Theorem:

$$\Pr(C_i|\boldsymbol{x}) = \frac{\Pr(C_i)\Pr(\boldsymbol{x}|C_i)}{\Pr(\boldsymbol{x})}. \tag{1}$$

A rule cluster of the ith rule, C_i,is represented by a $\boldsymbol{v}_i$-centred Gaussian with covariance $\boldsymbol{\Sigma}_i$. Therefore, the probability density function of the ith rule's cluster is represented by:

$$\Pr(\boldsymbol{x}|C_i) = \frac{1}{(2\pi)^{\frac{n_s}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} \cdot \exp\left\{\frac{-1}{2}(\boldsymbol{x}-\boldsymbol{v}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{v}_i)\right\}. \tag{2}$$

A robot requires g_i, instead of calculating $\Pr(C_i|\boldsymbol{x})$[1] by omitting $\Pr(\boldsymbol{x})$ in Eq.(1) as a common factor for all clusters. A robot must select a rule on the basis of only the numerator. The value of g_i is calculated as follows:

$$\begin{aligned} g_i &= -\log(f_i \cdot \Pr(\boldsymbol{x}|C_i)) \\ &= \frac{1}{2}(\boldsymbol{x}-\boldsymbol{v}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{v}_i) - \log\left\{\frac{1}{(2\pi)^{\frac{n_s}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}}\right\} - \log f_i, \end{aligned} \tag{3}$$

where f_i is synonymous with $\Pr(C_i)$.

After calculating g for all the rules, the winner rl_w with the minimal value of g_i is selected. As mentioned in the learning procedure in Sec. 3.1, the action in rl_w is performed if g_w is lower than a threshold $g_{th} = -\log(f_0 \cdot P_{th})$, where f_0 and P_{th} are predefined positive constants. Otherwise, a new action is produced. This new action is given by comparison with another threshold[2], $g'_{th} = -\log(f_0 \cdot P'_{th})$ in one of the following two ways:

[1] The higher the value of $\Pr(C_i|\boldsymbol{x})$, the lower is the value of g_i.

[2] $P'_{th} < P_{th}$

- $g_{th} \leq g_w < g'_{th}$: The robot executes an action with parameters determined on the basis of rl_w and other rules with g in this range as follows:

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

  where n_r denotes the number of referred rules and $N(0, \sigma)$ is a zero-centred Gaussian noise with variance σ. This utility-weighted average action is regarded as an interpolation of previously acquired knowledge.
- $g'_{th} \leq g_w$: The robot generates a random action.

Updating Rule Set. The update phase is performed except when an action by rl_w results in punishment. If a new action is taken ($i.e.$ $g_w > g_{th}$), a new rule that is composed of the current sensory input and the executed action is added to R. The parameters for the new rule are defined as follows:

$$\boldsymbol{v}_c = \boldsymbol{x}, \boldsymbol{\Sigma}_c = \sigma_0^2 \boldsymbol{I}, \boldsymbol{a}_c = \boldsymbol{a}_w, u_c = u_0, f_c = f_0. \tag{5}$$

In these equations, σ_0, u_0 and f_0 are constants and 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 Φ_w is updated by adding the current sensory input to $\boldsymbol{x}$. Then, the sample mean $\bar{x} = \{\bar{x}_1, \ldots, \bar{x}_{n_s}\}^T$ and the sample variance $s^2 = \{s_1^2, \ldots, s_{n_s}^2\}^T$ are estimated from the updated set Φ_w. The confidence intervals for $\bar{x}$ and s^2 are also updated. In subsequence, BRL determines whether any component of $\boldsymbol{v}$ and $\boldsymbol{\Sigma}$ is outside the range of the confidence intervals. If any component is outside that range, the updates are conducted:

$$v_i \leftarrow v_i + \alpha(\bar{x}_i - v_i), \tag{6}$$

$$\sigma_i^2 \leftarrow \sigma_i^2 + \alpha^2 [s_i^2 - \sigma_i^2], \tag{7}$$

$$f_w \leftarrow f_w + \beta(1 - f_w), \tag{8}$$

where α and β are constants. For all other rules, the prior probabilities f_i are updated as follows:

$$f_i \leftarrow (1 - \beta) f_i. \tag{9}$$

3.2 BRL with a Meta-learning Mechanism

BRL has a mechanism for adaptively updating rule parameters described in Step (6) in Sec. 3.1. BRL collects input-output data during an experiment. When learning advances, as mentioned in Sec. 3.1, BRL improves the precision of the acquired rules by reducing the value of the $\boldsymbol{\Sigma}$ component of the rules. However, overfitting occasionally causes the problem of excessively adjusting to the collected input-output data (Fig. 2).

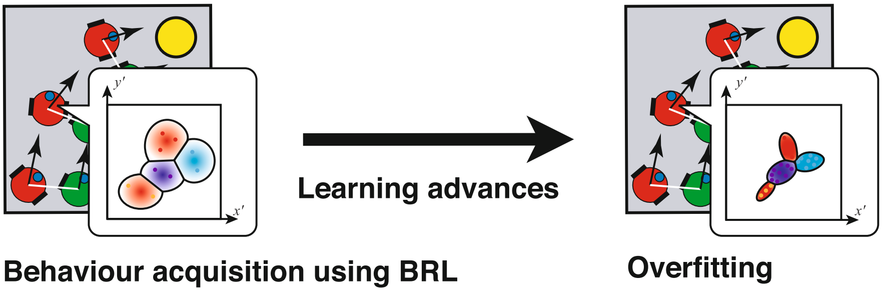


Fig. 2. Overfitting in BRL

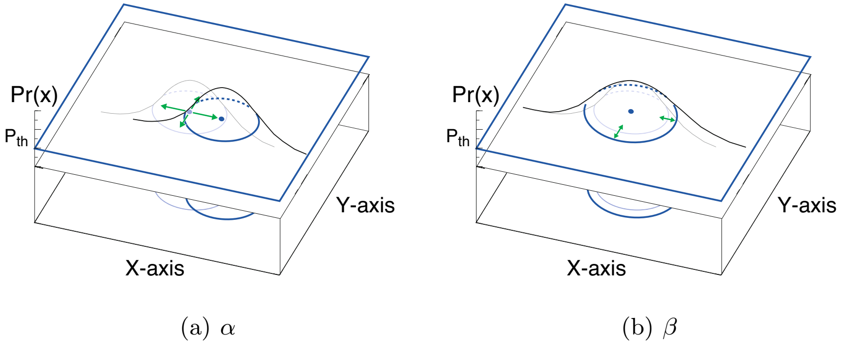


Fig. 3. Learning rates of BRL

To overcome this problem, the present paper proposes another extended BRL that focusses on meta-parameters that modulate the learning and are of crucial importance for RL. In this study, meta-learning [14,15], which is the capability of the learning algorithm to dynamically adjust its meta-parameters, is employed. As for RL robots, Elfwing *et al.* proposed an evolutionary approach to optimise meta-parameters [16].

In this study, the learning rates α and β (Fig. 3), which modulate the centre position and the range of rule clusters, respectively, are coordinated in response to the stability of behaviour S as follows:

$$\alpha \leftarrow (1-w)\alpha + \frac{w\alpha_{max}}{1+\exp[-\gamma(S-\delta)]}, \tag{10}$$

$$\beta \leftarrow (1-w)\beta + \frac{w\beta_{max}}{1+\exp[-\gamma(S-\delta)]}, \tag{11}$$

where α_{max} and β_{max} denote the maximum values of α and β in the range [0,1], respectively. γ and δ are positive constants and w is an inertia weight in the range [0,1]. Here, S is given on the basis of the transition of the information entropy of the fired rules E:

$$S = |E_t - E_{t-1}|, \tag{12}$$

$$E_t = -\sum Q(i) \log Q(i), \tag{13}$$

where $Q(i)$ is the probability that the ith rule is fired in the tth episode.

4 Real Robot Experiments

4.1 Experimental Settings

The robot makes decisions on the basis of the position (distance r and direction θ) of the object, the nearest robot and the wall by using its omnidirectional camera (Fig. 4). The input to the controller is given by $\boldsymbol{x} = \{r_0, \cos\theta_0, \sin\theta_0, r_1, \cos\theta_1, \sin\theta_1, r_3, \cos\theta_2, \sin\theta_2\}$, where the suffixes 0, 1 and 2 denote the object, the nearest robot and the wall, respectively. The output to the robot is $\boldsymbol{a} = \{m_{rud}, m_{th}\}$, where m_{rud} and m_{th} are the motor commands for the rudder and the throttle, respectively.

We represent a unit of time as a *step*. A *step* is a sequence that allows the robot to obtain its own input information, make decisions by itself, and execute its action. An episode of learning continues until two or three robots turn 60° in the counterclockwise direction by maintaining a distance of 1.0 m or less from the nearest robot or until the 100th time-step arrives. The robots are manually transported to their initial position after every 30 episodes. The robots are rewarded when they turn 60° around the object, whereas they are penalised when they collide with the wall or object. If robots complete a lap, the experimental run is regarded as successful. The BRL parameters are the same as the values

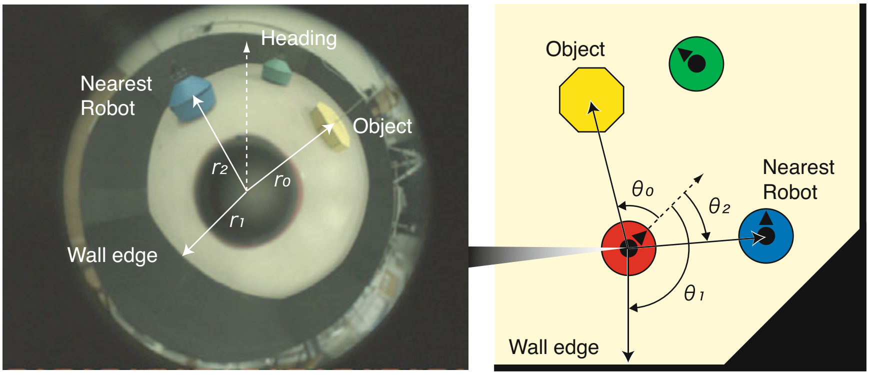


Fig. 4. Input through the omnidirectional camera

proposed in our previous study. The parameters designed for the extended BRL are $\alpha_{max} = 0.001$, $\beta_{max} = 0.01$, $\gamma = 30$, $\delta = 0.7$ and $w = 0.1$.

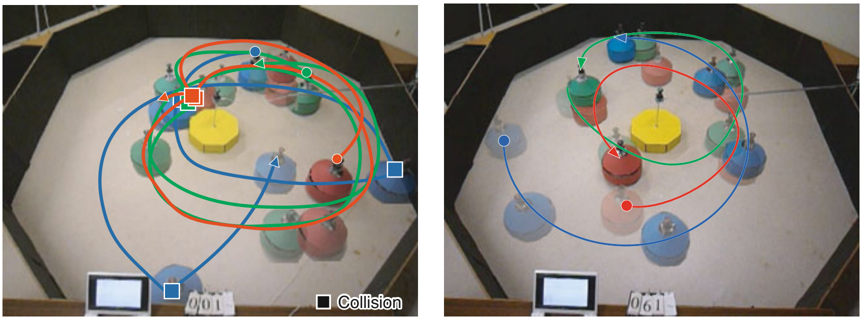
In addition, Q-learning is adopted for comparison. It has a normalised radial-basis function network to approximate an action value function. The input is the same as BRL, and six primitive actions, such as slowly/quickly moving straight ahead, slowly/quickly moving diagonally forward left and slowly/quickly turning left, are available.

4.2 Results

Five experimental runs were performed for each controller. All the experiments yielded successful results by using Q-learning and BRLs. Figure 5 shows an example of the behaviour during a run for the extended BRL. In this figure, trajectories are depicted from the circles to the triangles. In the early stages, the robots have no knowledge and function by trial and error. During this process, the robots often collide with the wall or object. Then, the robots maintain a regulation distance between each other and completely orbit the object.

The average number of consecutive complete laps and punishments are illustrated in Fig. 6. It is found that a larger number of rewards and a smaller number of punishments are obtained using our extension. These results indicate that for this cooperative task, robots with the extended BRL develop more stable object-orbiting behaviour than those with Q-learning or standard BRL.

Figure 7 illustrates the transition of the learning rate α of the extended BRL in a typical experimental run[3]. learning rate fluctuates and gradually reduces during the experiment. Our proposed meta-learning mechanism enables the learning rates to be modified based on the stability of the behaviour of MRS. This mechanism would provide better, more robust performance in MRSs.

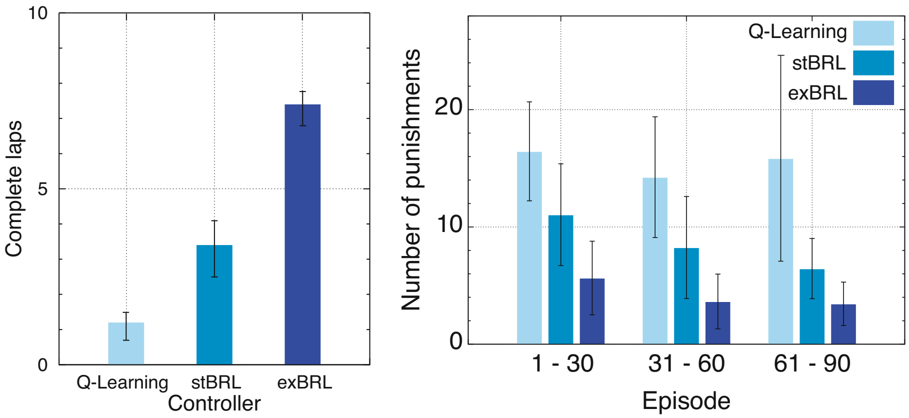


(a) At an early stage

(b) After successful learning

Fig. 5. Examples of behaviour

[3] The transition of β is exactly the same as α.



(a) Consecutive complete laps (b) Collisions

Fig. 6. Performance comparison

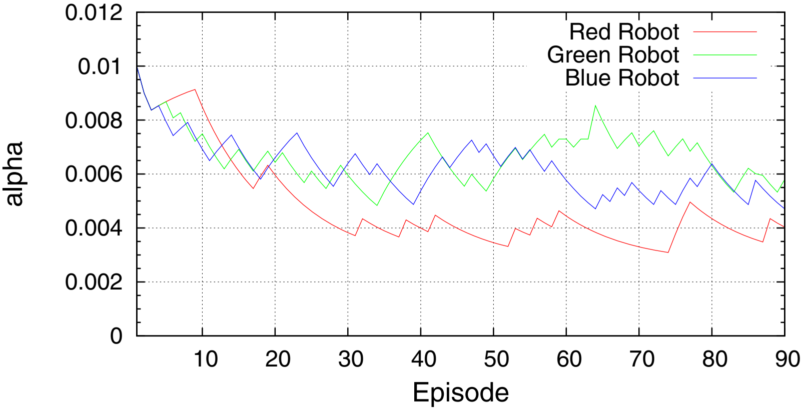
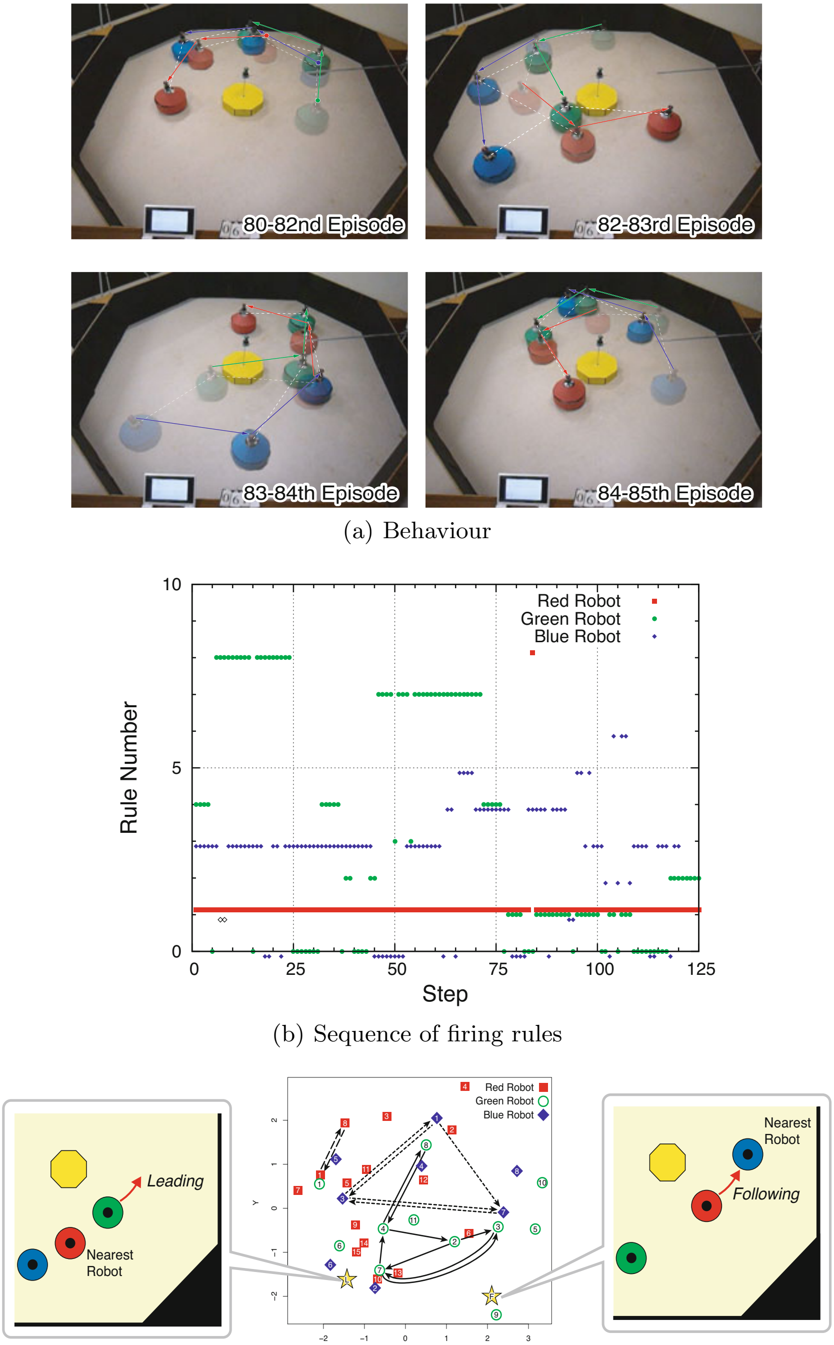


Fig. 7. Transition of learning rate α

4.3 Discussion

To visually grasp how the robots demonstrate their cooperative behaviour, the acquired rules are projected onto a plane using an Isomap [17]. The Isomap can reduce dimensionality on the basis of manifold structures in high-dimensional space and preserve local topological relationships among data.

Figure 8(a) shows an example of successful object-orbiting behaviours. First, the red robot leads the blue robot, which is followed by the green robot. In subsequence, the green robot catches up with the blue robot. Then, the green robot is the red robot's follower and leads the blue robot. Finally, the three



(a) Behaviour

(b) Sequence of firing rules

(c) Isomap

Fig. 8. Acquired rules

robots orbit the object once. The sequence of firing rules in this duration is shown in Fig. 8(b)[4].

Figure 8(c) shows the Isomap based on the input-output data of the three robots and the typical leading and following behaviours. The arrows in this figure indicate the transition of firing rules. The rules for leading and following are located on the left and right, respectively. The rules for the red robot, which always leads, are projected on the left , whereas the rules for the other two are distributed in the map. This indicates that green and blue robots dynamically change their roles of leader and follower.

By observing the acquired behaviour and investigating the rules, it is concluded that the robots developed cooperative behaviour on the basis of autonomous specialisation.

5 Conclusion

We investigated an RL approach for the behaviour acquisition of autonomous MRSs. Our proposed RL technique, BRL, has a mechanism for the adaptive discretization of the continuous learning space and has proven to be effective for an MRS. This paper introduced an extended BRL for improving the robustness of an MRS by providing a meta-learning mechanism that adaptively modifies its learning parameters on the basis of the information entropy of firing rules. The results of the physical experiments demonstrated that our proposed method would improve the robustness of an MRS.

In the future, we plan to investigate the robustness of the extended BRL against environmental change after successful learning. We also plan to conduct experiments employing larger number of robots, especially those with more sensors and actuators.

References

1. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
2. Sutton, R.S.: Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding. Advances in Neural Information Processing Systems 8, 1038–1044 (1996)
3. Morimoto, J., Doya, K.: Acquisition of Stand-Up Behavior by a Real Robot using Hierarchical Reinforcement Learning for Motion Learning: Learning 'Stand Up' Trajectories. In: Proc. of International Conference on Machine Learning, pp. 623–630 (2000)
4. Lin, L.J.: Scaling Up Reinforcement Learning for Robot Control. In: Proc. of the 10th International Conference on Machine Learning, pp. 182–189 (1993)
5. Kolter, J.Z., Ng, A.Y.: Regularization and Feature Selection in Least-Squares Temporal Difference Learning. In: Proc. of the 26th International Conference on Machine Learning (2009)

[4] Rules 0 denote newly produced rules. Although the robots have the same IDs, their components are completely different.

6. Nouri, A., Littman, M.L.: Dimension Reduction and Its Application to Model-Based Exploration in Continuous Spaces. Machine Learning 81(1), 85–98 (2010)
7. Asada, M., Noda, S., Hosoda, K.: Action-Based Sensor Space Categorization for Robot Learning. In: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1502–1509 (1996)
8. Takahashi, Y., Asada, M., Hosoda, K.: Reasonable Performance in Less Learning Time by Real Robot Based on Incremental State Space Segmentation. In: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1502–1524 (1996)
9. Doya, K.: Reinforcement Learning in Continuous Time and Space. Neural Computation 12, 219–245 (2000)
10. Peters, J., Schaal, S.: Natural actor critic. Neurocomputing 71(7-9), 1180–1190 (2008)
11. Yasuda, T., Ohkura, K.: Autonomous Role Assignment in Homogeneous Multi-Robot Systems. Journal of Robotics and Mechatronics 17(5), 596–604 (2005)
12. Yasuda, T., Ohkura, K.: Improving Search Efficiency in the Action Space of an Instance-Based Reinforcement Learning. In: Almeida e Costa, F., Rocha, L.M., Costa, E., Harvey, I., Coutinho, A. (eds.) ECAL 2007. LNCS (LNAI), vol. 4648, pp. 325–334. Springer, Heidelberg (2007)
13. Yasuda, T., Ohkura, K.: Reinforcement Learning Technique with an Adaptive Action Generator for a Multi-Robot System. In: Asada, M., Hallam, J.C.T., Meyer, J.-A., Tani, J. (eds.) SAB 2008. LNCS (LNAI), vol. 5040, pp. 250–259. Springer, Heidelberg (2008)
14. Doya, K.: Metalearning and neuromodulation. Neural Networks 15(4-6), 495–506 (2002)
15. Schweighofer, N., Doya, K.: Meta-learning in Reinforcement Learning. Neural Networks 16(1), 5–9 (2003)
16. Elfwing, S., Uchibe, E., Doya, K., Chiristensen, H.I.: Co-evolution of Shaping Rewards and Meta-Parameters in Reinforcement Learning. Adaptive Behavior 16, 400–412 (2008)
17. Tenenbaum, J.B., de Sliva, V., Lagford, J.C.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science 290(22), 2319–2323 (2000)