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

Toshiyuki Yasuda and Kazuhiro Ohkura

Graduate School of Engineering, Hiroshima University
Kagamiyama 1-4-1, Higashi-Hiroshima, Hiroshima 739-8527, Japan
{yasu,kohkura}@hiroshima-u.ac.jp
http://www.ohk.hiroshima-u.ac.jp

Abstract. We have developed a new reinforcement learning (RL) technique called Bayesian-discrimination-function-based reinforcement learning (BRL). BRL is unique, in that it does not have state and action spaces designed by a human designer, but adaptively segments them through the learning process. Compared to other standard RL algorithms, BRL has been proven to be more effective in handling problems encountered by multi-robot systems (MRS), which operate in a learning environment that is naturally dynamic. Furthermore, we have developed an extended form of BRL in order to improve the learning efficiency. Instead of generating a random action when a robot functioning within the framework of the standard BRL encounters an unknown situation, the extended BRL generates an action determined by linear interpolation among the rules that have high similarity to the current sensory input. In this study, we investigate the robustness of the extended BRL through further experiments. In both physical experiments and computer simulations, the extended BRL shows higher robustness and relearning ability against an environmental change as compared to the standard BRL.

Keywords: Multi-Robot System, Reinforcement Learning, Autonomous Specialization, Action Search.

1 Introduction

A robust instance-based reinforcement learning (RL) approach for controlling autonomous multi-robot systems (MRS) is introduced in this paper. 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.

M. Asada et al. (Eds.): SAB 2008, LNAI 5040, pp. 250–259, 2008.

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 [1] used CMAC and Morimoto and Doya [2] used Gaussian softmax basis functions for function approximation. Lin represented the Q-function by using multi-layer neural networks called *Q-net* [3]. 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 [4]. Takahashi *et al.* used the nearest-neighbor method [5]. 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 [6]. We have also developed a second instance-based RL method called Bayesian-discrimination-function-based reinforcement learning (BRL) [7]. Our preliminary experiments proved that BRL, by means of adaptive segmentation of state and action spaces, exhibits better performance as compared to CSCG.

BRL has an extended form that accelerates the learning speed [8]. 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 MRS that employ the extended BRL learn behaviors faster as compared to those that employ the standard BRL. In this study, 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 paper is organized as follows. The target problem is introduced in Section 2. 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 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 it involves requiring the MRS to carry a triangular board from the start position to the goal position. 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 θ. A robot can perceive the potentiometer measurements of the other robots as well as its own. All three robots have

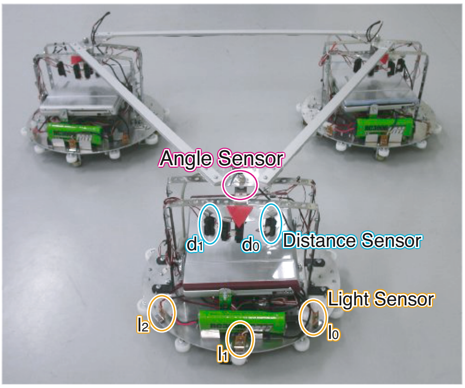


Fig. 1. Cooperative carrying problem

the same specifications; each robot possesses two distance sensors d and three light sensors l. The larger the value of d / l, the shorter is the distance to an obstacle or a light source. The resolutions of the distance sensor, light sensor, and potentiometer are 350, 400, and 100, respectively. Each robot possesses two motors for rotating two omnidirectional wheels. The resolution of the motor signal is 16. A wheel provides a powered drive along the direction in which it points and a passive coasting along an orthogonal direction, simultaneously.

The difficulties involved in executing 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 causes a dynamic and uncertain state transition.

3 APPROACH

3.1 BRL: RL in Continuous Learning Space

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 this RL, 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 means of Bayesian discrimination, the robot selects the most similar rule from a rule set. If a rule is selected, the robot executes the corresponding action $\boldsymbol{a}$; otherwise, it performs a random action.
(3) The robot transfers to the next state and receives a reward r.
(4) All rule utilities are updated according to r. Rules with utility below a certain threshold are removed.
(5) When the robot performs a random action, the robot produces a new rule by combining the current sensory input and the executed action. This executed new rule is memorized in the rule table.
(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 is selected to minimize g, *i.e.*, the risk of misclassification of the current input. We obtain g on the basis of 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:

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

A rule cluster of i-th rule, C_i, is represented by a $\boldsymbol{v}_i$-centered Gaussian with covariance $\boldsymbol{\Sigma}_i$. Therefore, the probability density function of the i-th 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 $\Pr(C_i|\boldsymbol{x})^1$, because no one can correctly estimate $\Pr(\boldsymbol{x})$ in Eq.(1). A robot must select a rule on the basis of only the numerator. The value of g_i is calculated as

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

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

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

After calculating g for all the rules, the winner rl_w is selected as that with the minimal value of g_i. 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 constants. Otherwise, a random action is performed.

3.2 Extended BRL

Basic Concept. We have some RL approaches that provide learning in continuous action spaces. An actor-critic algorithm built with neural networks has a continuous learning space and modifies actions adaptively [9]. This algorithm modifies policies based on TD-error at every time step. Theoretically, the REINFORCE algorithm requires immediate rewards [10]. 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. 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. 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.

BRL with an Adaptive Action Generator. In order to accelerate learning, in this study, we introduce an extended BRL by modifying the learning procedure, Step (2) in Sec. 3.1. In this extension, instead of a random action, the robot performs a knowledge-based action when it encounters a new environment. Therefore, 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:

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

  where n_r denotes the number of referred rules, and $N(0, \sigma)$ is a zero-centered 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.

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

4 Experiments

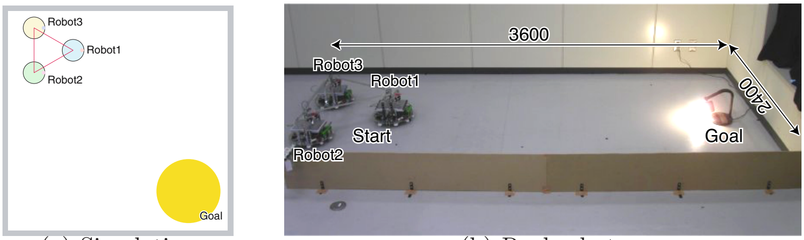
4.1 Settings

Figure 2 shows the general view of the experimental environments for the simulation and physical experiments. In the simulation runs, the field is a square surrounded by a wall. The real robots are situated in a pathway with length and width 3.6 m and 2.4 m, respectively. The task for the MRS is to move from the start position to the goal position (light source). All the robots get a 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 punishment when it collides with a wall ($d_0^i > thr_d \vee d_1^i > thr_d$). We represent a unit of time as a *step*. A *step* is a sequence that allows the three robots to obtain 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 the simulations and 100 steps in the physical experiments, it is returned to the start position. This time span is called an *episode*.

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

Prediction Mechanism (NN). In our previous study [7], we verified BRL to be a successful approach to CCP by introducing reformations such that the state space was constructed by using sensory information and predictions of the posture of the other robots in the subsequent time step in order to decrease the learning problem dynamics.

The prediction mechanism attached is a three-layered feed-forward neural network that performs back propagation. The hidden layer has eight nodes. The input of the i-th robot is a short history of sensory information, $I^i = \{\cos\theta^i_{t-2}, \sin\theta^i_{t-2}, \cos\psi^i_{t-2}, \sin\psi^i_{t-2}, \cos\theta^i_{t-1}, \sin\theta^i_{t-1}, \cos\psi^i_{t-1}, \sin\psi^i_{t-1}, \cos\theta^i_t, \sin\theta^i_t, \cos\psi^i_t, \sin\psi^i_t\}$, where $\psi^i_t = (\theta^j_t + \theta^k_t)\ /2 \quad (i \neq j \neq k$, j and k indicate the IDs of the neighboring robots). The output is a prediction of the posture of the other robots in the subsequent time step $O^i = \{\cos\psi^i_{t+1}, \sin\psi^i_{t+1}\}$. The behavior learning mechanism utilizes O^i as a part of sensory information input.



(a) Simulation (b) Real robots

Fig. 2. Experimental environment

Behavior Learning Mechanism (BRL). The input of the i-th robot is $\boldsymbol{x}^i = \{\ \cos\theta_t^i,\ \sin\theta_t^i,\ \cos\psi_{t+1}^i,\ \sin\psi_{t+1}^i,\ d_0^i,\ d_1^i,\ l_0^i,\ l_1^i,\ l_2^i\ \}$. The output is $\boldsymbol{a}^i = \{m_{rud}^i, m_{th}^i\}$, where m_{rud}^i and m_{th}^i are the motor commands for the rudder and the throttle respectively. The value of σ in Eq.(4) is 0.05. For the standard BRL, $P_{th} = 0.012$. For the extended BRL, $P_{th} = 0.012$ and $P'_{th} = 0.01$. The other parameters are the same as the values recommended in our journal [7].

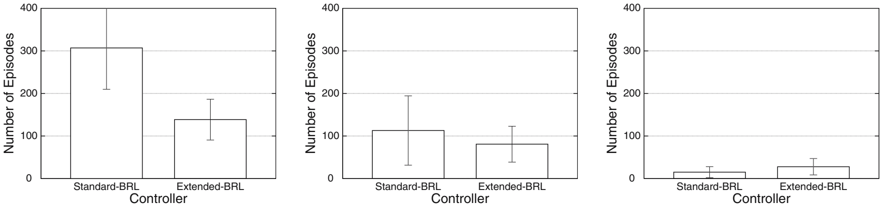
We introduce a change in an environment by initializing one of the three robots. This may correspond to a situation in which a robot is replaced with a new one. Such changes occur when the MRS continuously reaches the goal for 100 consecutive episodes in the simulations and for 25 consecutive episodes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

4.2 Result: 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 [8]. 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.

Figure 3 shows the average and the deviations in the number of episodes for 10 independent runs. 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. 3(a)). 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



(a) For the leader cases (b) For the sub-leader cases (c) For the follower cases

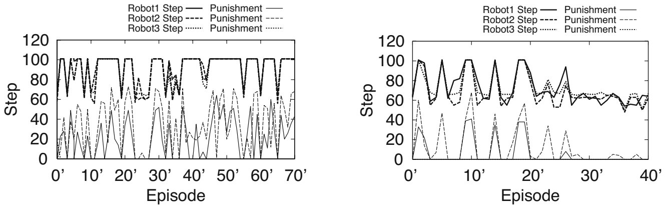
Fig. 3. Numbers of episodes required to relearn a behavior after an environmental change

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. 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.

4.3 Result: Physical Experiments

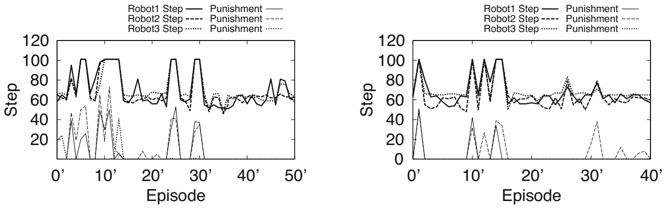
We conducted five independent experimental runs for each case employing the BRL. The standard BRL provided two successful results and the extended BRL provided four successful results from scratch [8].

Figures 4–6 illustrate the learning results after one of the robots is initialized by using the best results in [8] 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.



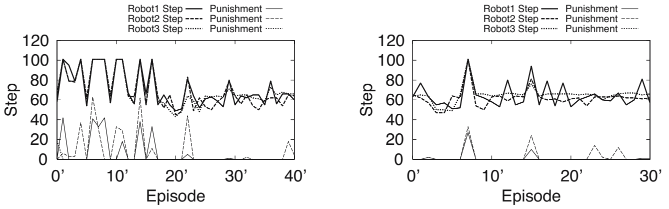
(a) Standard BRL (b) Extended BRL

Fig. 4. Learning history after a leader is initialized



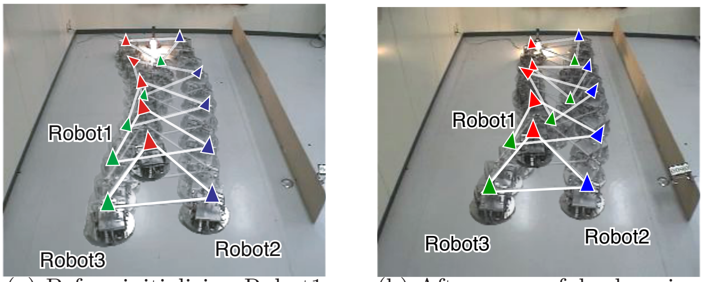
(a) Standard BRL (b) Extended BRL

Fig. 5. Learning history after a sub-leader is initialized



(a) Standard BRL (b) Extended BRL

Fig. 6. Learning history after a follower is initialized



(a) Before initializing Robot1 (b) After successful relearning

Fig. 7. Acquired behavior: extended BRL

Figure 7 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 [6], by using the same parameter settings.

5 Conclusions

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.

References

1. Sutton, R.S.: Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding. In: Advances in Neural Information Processing Systems, vol. 8, pp. 1038–1044. MIT Press, Cambridge (1996)
2. 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: Intl. Conf. on Machine Learning, pp. 623–630 (2000)
3. Lin, L.J.: Scaling Up Reinforcement Learning for Robot Control. In: the 10th Intl Conf. on Machine Learning, pp. 182–189 (1993)
4. Asada, M., Noda, S., Hosoda, K.: Action-Based Sensor Space Categorization for Robot Learning. In: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, pp. 1502–1509 (1996)
5. Takahashi, Y., Asada, M., Hosoda, K.: Reasonable Performance in Less Learning Time by Real Robot Based on Incremental State Space Segmentation. In: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, pp. 1518–1524 (1996)
6. Svinin, M., Kojima, F., Katada, Y., Ueda, K.: Initial Experiments on Reinforcement Learning Control of Cooperative Manipulations. In: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, pp. 416–422 (2000)
7. Yasuda, T., Ohkura, K.: Autonomous Role Assignment in Homogeneous Multi-Robot Systems. Journal of Robotics and Mechatronics 17(5), 596–604 (2005)
8. Yasuda, T., Ohkura, K.: Improving Search Efficiency in the Action Space of an Instance-Based Reinforcement Learning. In: e Costa, F.A., Rocha, L.M., Costa, E., Harvey, I., Colutinho, A. (eds.) ECAL 2007. LNCS (LNAI), vol. 4648, pp. 325–334. Springer, Heidelberg (2007)
9. Doya, K.: Reinforcement Learning in Continuous Time and Space. Neural Computation 12, 219–245 (2000)
10. Williams, R.J.: Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. Machine Learning 8, 229–256 (1992)