

## Multi-Dimensional Reinforcement Learning Using a Vector Q-Net – Application to Mobile Robots

K.Kiguchi\*, T.Nanayakkara\*\*, K.Watanabe\*, and T.Fukuda\*\*\*

\* Department of Advanced Systems Control Engineering, Saga University, 1 Honjomachi, Saga-shi, Saga 840-8502, Japan  
(Tel : 81-952-28-8702; Fax : 81-952-28-8587 ; E-mail: kiguchi@ieee.org)

\*\* Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD 21205, USA  
(E-mail: thrish@bme.jhu.edu)

\*\*\* Center of Cooperative Research in Advanced Science and Technology, Nagoya University, 1 Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan  
(Tel : 81-52-789-4478; Fax : 81-52-789-3909 ; E-mail: fukuda@mein.nagoya-u.ac.jp)

**Abstract:** Reinforcement learning is considered as an important tool for robotic learning in unknown/uncertain environments. In this paper, we propose an evaluation function expressed in a vector form in order to realize multi-dimensional reinforcement learning. The novel feature of the proposed method is that learning one behavior induces parallel learning of other behaviors though the objectives of each behavior are different. In brief, all behaviors watch other behaviors from a critical point of view. Therefore, in the proposed method there is cross-criticism and parallel learning that make the multi-dimensional learning process more efficient. By applying the proposed learning method, multi-dimensional evaluation (reward) and multi-dimensional learning can be carried out simultaneously in one trial. A special neural network (Q-net), in which the weights and the output are represented by vectors, is proposed to realize a critic network for Q-learning. The proposed learning method is applied for behavior planning of mobile robots.

**Keywords:** Reinforcement learning, Q-learning, Multi-dimensional evaluation, Neural networks, Intelligent robot

### 1. Introduction

Learning algorithms based on evaluative feedback signals is generally referred to as reinforcement learning algorithms. In a reinforcement learning paradigm, a system called *agent* senses the environment and produces control actions. The environment responds to these control actions. Based on these responses a *reward function* will evaluate the control actions. The *agent* tries to optimize the control policy to maximize the total expected *reward* over a finite time-span. Learning may occur using the prediction error of expected rewards. Such a learning mechanism can be found in the basal ganglia of the mammalian brain also [1]. In [1], it is experimentally shown that the activity of dopamine neurons in the ventral tegmental area and the substantia nigra of rats reflect the prediction of temporal difference or the prediction error of the expected rewards.

Reinforcement learning [2][3] plays an important role in robot learning under unknown/uncertain environments. In a reinforcement learning paradigm, the optimum control policy can be obtained based on interactive explorations in the environment. Therefore, reinforcement learning is effective for intelligent robots to realize intelligence such as making a game strategy [4] or skillful motions [5] based on their experience. Many studies on reinforcement learning have been performed to make the robots work intelligently in an unknown/uncertain environment [4]-[13]. In those studies, the optimal or desired behavior of the

robot is assumed to be only one, and evaluated with a single evaluation function or a weighted sum of evaluation functions. For some sophisticated systems such as intelligent robots, however, it is sometimes difficult to evaluate their performance with only one evaluation (*reward*). The desired behavior sometimes depends on the circumstances while there can be contradicting objectives that have special importance in certain circumstances. For example, the behavior of less energy consumption is usually preferred. However, time efficiency is more important than energy efficiency when the robot is in a rush. Usually, the best behavior with respect to energy consumption is not the same as that with respect to time efficiency. Furthermore, safety is the most important when the robot carry out important tasks. Thus the desired behavior should be changed according to the situation. This kind of idea is similar to the idea of multiple reward criterion proposed by Uchibe and Asada [13].

In this paper, we propose an evaluation function expressed in a vector form in order to realize multi-dimensional reinforcement learning. Q-learning [3], one of the basic reinforcement learning methods, has been applied in this study. A special neural network (Q-net), in which the weights and the output are represented by vectors, is proposed to realize critic networks for Q-learning. Each parallel network in the Q-net works as an element of the vector Q-net. The novel feature in the proposed learning algorithm is that learning occurs in all the networks

while implementing any given behavior. This is realized through cross-criticism by reward functions at any given time. When a certain behavior is performed, reward or punishment with respect to the performed behavior is evaluated by all the elements in the vector evaluation function. At the same time, all the networks in the Q-net try to predict the expected sum of future rewards from each network's point of view, even though the actual behavior corresponds to only one of the objectives in the vector of objectives. This kind of cross evaluations can be found in the learning process of humans through social interactions also. Sometimes we observe the behavior of another person in a given situation and try to subconsciously predict future results based on a self-centered internal model. While observing we continuously criticize the internal model of prediction, leading to cross learning. Therefore we learn not only from our own behavior but also by observing other's behaviors. The proposed learning method is based on a similar phenomenon.

In this study, we have assumed that there are obstacle regions, slippery regions, and danger regions in the working environment of the mobile robot. The robot is supposed to waste some energy and time for the slip in the slippery regions, and waste a lot of energy and time for struggling to move in the danger regions. The dynamics of the mobile robot is taken into account. The energy minimum behavior, the hasty behavior, and the safe behavior are efficiently explored using the proposed reinforcement learning in this environment. Consequently, each weight vector and the output vector of the Q-net consist of three components (1st component: for energy minimum behavior, 2nd component: for hasty behavior, and 3rd component: for safe behavior) in this case. The robot is able to change the optimal behavior according to the situation after the proposed learning. The effectiveness of the proposed reinforcement learning has been evaluated in simulation.

## 2. Dynamic Model of the Mobile Robot

The schematic diagram of the mobile robot is shown on the left side of Fig. 1, where  $I_v$  is the moment of inertia around the c.g. of robot,  $v$  is the velocity of robot,  $\mathbf{f}$  is the azimuth of robot, and  $l$  is the distance between the left or right wheel and the c.g. of the robot.

Let

$$\mathbf{x}(t) = [v(t) \ \mathbf{f}(t) \ \dot{\mathbf{f}}(t)]^T$$

be the state variable vector and

$$\mathbf{u}(t) = [u_r \ u_l]^T$$

be the manipulated variable vector. Then the state space model for the mobile robot can be written as:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (1)$$

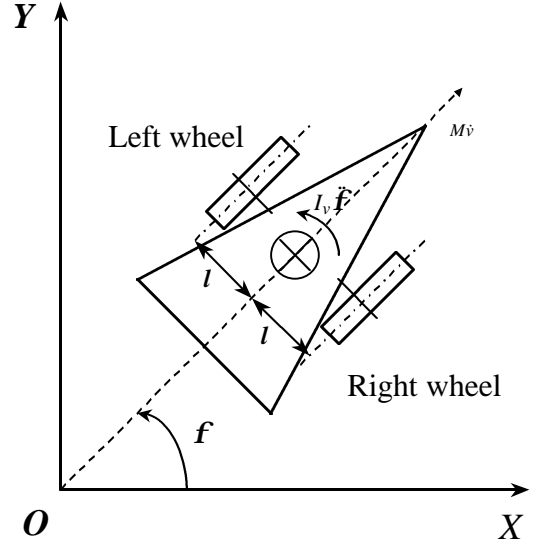


Fig. 1 Schematic diagram of the mobile robot.

with

$$\mathbf{A} = \begin{bmatrix} -2c/(Mr^2 + 2I_w) & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -2cl^2/(I_v r^2 + 2I_w l^2) \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} -kr/(Mr^2 + 2I_w) & -kr/(Mr^2 + 2I_w) \\ 0 & 0 \\ krl/(I_v r^2 + 2I_w l^2) & krl/(I_v r^2 + 2I_w l^2) \end{bmatrix}$$

where  $M$  represents the mass of robot,  $I_w$  is the moment of inertia of wheel,  $c$  is the viscous friction factor of wheel,  $k$  is the driving gain factor,  $r$  represents the radius of wheel, and  $u_r$  and  $u_l$  are the right and left driving input torque, respectively.

The physical parameters of the mobile robot used in this study are given by  $I_v = 0.6541[\text{kgm}^2]$ ,  $M = 25.5 [\text{kg}]$ ,  $l = 0.165 [\text{m}]$ ,  $r = 0.05 [\text{m}]$ ,  $I_w = 0.4419 \cdot 10^{-3} [\text{kgm}^2]$ ,  $k = 90$ , and  $c = 0.0479 [\text{kgm}^2/\text{s}]$ .

## 3. Multi-Dimensional Reinforcement Learning

In order to make the basic concept of the proposed learning clear, Q-learning method, one of the basic reinforcement learning methods, has been selected in this study. The proposed learning method is applied for behavior planning of the mobile robot. A special neural network (Q-net) is proposed to realize critic networks. In the proposed Q-net, the weights and the output are represented by vectors, although those are usually represented by scalars. Each component of vectors is in charge of each item of the evaluation (reward). In this study, evaluation is carried out with respect to energy consumption (energy minimum behavior), time efficiency (hasty behavior), and safety (safe

behavior) assuming that there are obstacle regions, slippery regions, and danger regions in the working environment of the mobile robot. In this case, each weight vector and output vector of the Q-net consist of three components (i.e., 1<sup>st</sup> component: for energy consumption, 2<sup>nd</sup> component: for time efficiency, and 3<sup>rd</sup> component: for safety). After a certain behavior is performed, each component of the weight vectors and the output vector of the Q-net is adjusted based on reward or punishment for energy minimum behavior, hasty behavior, and safe behavior.

### 3.1 Q-net Architecture

The proposed Q-net consists of three layers (input layer, hidden layer, and output layer). There are 16 input variables (1: distance to target, 2: angle to target, 3: distance to obstacle, 4: angle to obstacle, 5: distance to the first slippery area, 6: angle to the first slippery area, 7: distance to the second slippery area, 8: angle to the second slippery area, 9: position of robot in x-direction, 10: position of robot in y-direction, 11: velocity of robot in x-direction, 12: velocity of robot in y-direction, 13: azimuth of robot, 14: azimuth change rate, 15: left wheel torque, 16: right wheel torque).

There are 50 neurons in the hidden layer. The activation function used in the neurons is written as:

$$y_i = \frac{1}{1 + e^{-s_i}}, \quad i = 1, \dots, 50 \quad (2)$$

$$s_i = w_{oi} + \sum_{j=1}^{16} w_{ij} x_j \quad (3)$$

$$w_{oi} = [w_{1oi} \quad w_{2oi} \quad w_{3oi}]$$

$$w_{ij} = [w_{1ij} \quad w_{2ij} \quad w_{3ij}]$$

where  $w_{oi}$  is the bias weight vector of the  $i$ th activation function,  $w_{ji}$  represents the connecting weight vectors between the  $i$ th activation function and the  $j$ th input given by  $x_j$ .

The output of the Q-net is the Q values of the current control input combination given the situation. The Q values are calculated by:

$$Q_v = \sum_{i=1}^{50} w_{oi} y_i \quad (4)$$

where  $w_{oi}$  is the output weight vectors of the Q-net that connect the activation function and the output node.

### 3.2 Definition

Let the right and left side control torque inputs to the mobile robot by a conventional controller based on a potential field method be denoted by  $u_{cr}$  and  $u_{cl}$ , respectively. Denote the right and left side control torque inputs given by the Q-net be  $u_{qr} \in U_{qr}$  and  $u_{ql} \in U_{ql}$ , respectively, where  $U_{qr}$  and  $U_{ql}$  are real bounded spaces within which the right and left hand torques are defined.

### 3.3 External Reward Function

The external reward function is a vector of functions each rewarding distinct behaviors. In this case, the reward function vector consisted of three component functions for 1: hasty behavior, 2: Energy conscious behavior, 3: safety conscious behavior. Therefore the vector of functions were given by:

$$r(t) = [r_1(t) \quad r_2(t) \quad r_3(t)]^T \quad (5)$$

$$r_1(t) = \frac{4}{1 + 100e^{-(|u_r| + |u_l|)}} + r_{obs} + e^{-D} + P \quad (6)$$

where  $D$  is the distance to the target,  $P$  is a punishment given by  $P = -10$  if  $|u_r| > 0.04$  or  $|u_l| > 0.04$ , and  $r_{obs}$  is the reward or penalty for avoiding or colliding with the obstacle, which is calculated by  $r_{obs} = -100e^{-5|d_{obs}-0.5|}$  if close to the obstacle region, and  $r_{obs} = 1$  if sufficient distance is kept, in which  $d_{obs}$  is the distance to the obstacle.

$$r_2(t) = 4(\dot{v}_{tar} + e^{-D}) + r_{obs} \quad (7)$$

where  $\dot{v}_{tar}$  is the target reaching velocity.

$$r_3(t) = -100e^{-5|d_{da}-0.5|} + r_{obs} + e^{-D} \quad (8)$$

where  $d_{da}$  is the distance to the danger region. Input to the right and left wheels are given by:

$$u_r = u_{cr} + u_{rl}$$

$$u_l = u_{cl} + u_{ql} \quad (9)$$

Let the output of the Q-net for a given vector of environmental sensor information and a chosen control input be denoted by:

$$Q_v(t) = [Q_{1v}(t) \quad Q_{2v}(t) \quad Q_{3v}(t)]^T,$$

the maximum  $Q_v(t)$  that can be obtained by changing the right and left wheel torques in  $U_{qr}$  and  $U_{ql}$  for a given environmental situation be denoted by  $Q_{v,max}(t)$ , the reward obtained from an external reward function be given by:

$$r(t) = [r_1(t) \quad r_2(t) \quad r_3(t)]^T.$$

Then the following algorithm can be applied to obtain the optimum behaviors of the robot.

### 3.4 Reinforcement Algorithm

The algorithm of the proposed reinforcement learning is expressed as follows:

- Step 1: Initialize the weights of the Q-net, and set time  $t = 0$ .
- Step 2: Sense the state of the robot and calculate  $u_r$  and  $u_l$ .
- Step 3: Given the current control input and the environmental information, evaluate the Q-net and obtain a vector  $Q_v(t)$ .
- Step 4: Run the robot for one sampling time duration

and obtain a reward vector  $\mathbf{r}(t+1)$  from a set of external reward functions.

Step 5: For a given behavioral objective, i.e., energy optimization, hasty movement, or safe movement, Evaluate the Q-net and obtain  $\mathbf{Q}_{v,max}(t+1)$ , and the pair of control inputs  $u_{r,opt}$  and  $u_{l,opt}$  that renders  $\mathbf{Q}_{v,max}(t+1)$ .

Step 6: Calculate the temporal difference  $D(t+1) = [\Delta_1(t+1) \ \Delta_2(t+1) \ \Delta_3(t+1)]^T$  given by

$$D(t+1) = \mathbf{r}(t+1) + \mathbf{g}\mathbf{Q}_{v,max}(t+1) - \mathbf{Q}_v(t), \quad 0 < \mathbf{g} < 1 \quad (6)$$

Step 7: Use this  $D(t+1)$  vector to update the respective weight vectors of the Q-net.

Step 8: Set  $u_r = u_{r,opt} + N(0, \mathbf{S})$  so that  $\mathbf{S} = 1/(1 + e^{r_p(t+1)})$ , where  $p$  is the counter of the behavior type that decides the control inputs at time  $t+1$ . Go to Step 3, and set time  $t = t + 1$ ;

Continue these steps until a predetermined level of performance is achieved by all the vectors of weights in the Q-net. Note that a vector of Q values given by  $\mathbf{Q}_v(t)$  and reward values given by  $\mathbf{r}(t+1)$  are evaluated at any given time, eventhough only one behavior is executed at any given time. This ability of parale learning while executing a single behavior is the main advantage of the proposed method. This results from the mechanism of cross-criticism found in the proposed method.

#### 4. Simulation

In order to evaluate the effectiveness of the proposed learning method, computer simulation has been performed. In this simulation, the mobile robot is supposed to head toward the goal subjected to various performance criteria. There are one obstacle region, two slippery regions, and one danger region in

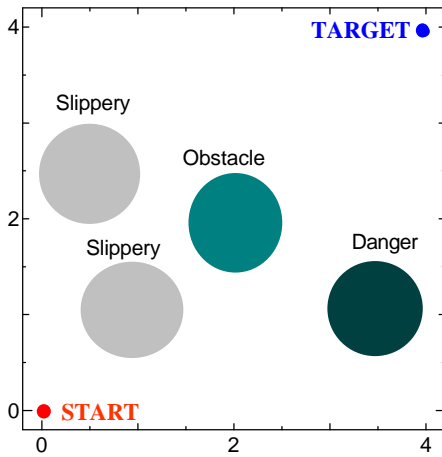


Fig. 2 Working environment of the mobile robot.

the working environment as shown in Fig. 2. In this simulation, the robot is supposed to waste 20% of driving torque for the slip in the slippery regions, and waste 80% of driving torque for struggling to move in the danger regions. The dynamics of the mobile robot explained in Section 2 is taken into account. The energy minimum behavior, the hasty behavior, and the safe behavior are considered in this simulation, although another behaviors can be considered.

Although multi-dimensional learning is carried out in each trial, one representative behavior is chosen in turn from among the three evaluating behaviors (energy minimum behavior, hasty behavior, and safe behavior). The random behavior is generated in certain range during the learning at every other trial of each evaluating behavior as shown in Fig. 3. Figure 4 and 5 show the simulation results after 1000 trials. The obtained energy minimum behavior, hasty behavior, and safe behavior are depicted in Fig. 4 (a), (b), and (c), respectively. The torque profiles of energy minimum behavior, hasty behavior, and safe behavior are shown in Fig. 5 (a), (b), and (c), respectively. One can see that the energy minimum behavior consumes less energy than the other behavior. In the hasty behavior, the robot quickly arrives at the target although a lot of energy is consumed. The safe behavior takes a lot of time to get to the target. These results show that the behavior of the robot can be changed depend on the situation.

#### 5. Conclusions

A novel multi-dimensional reinforcement learning method has been proposed and applied to Q-learning in this study. A special neural network (Q-net) is proposed to realize critic networks. In the proposed Q-net, the weights and the output are represented by vectors, although those are usually represented

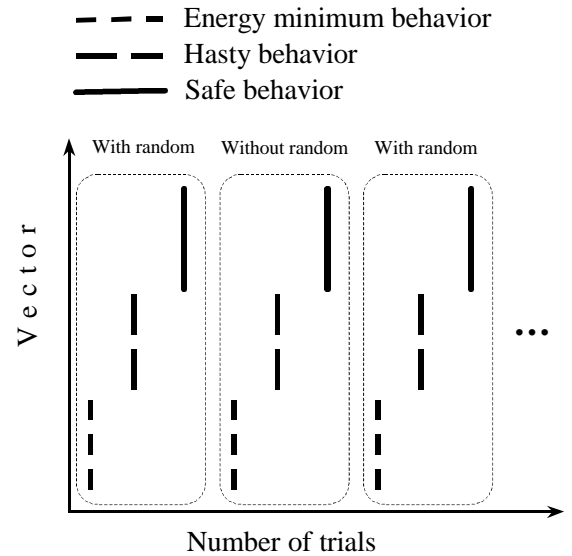
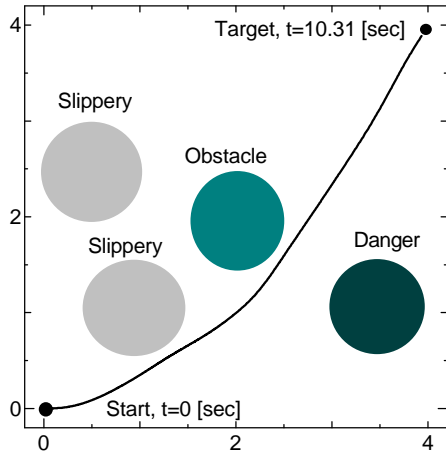
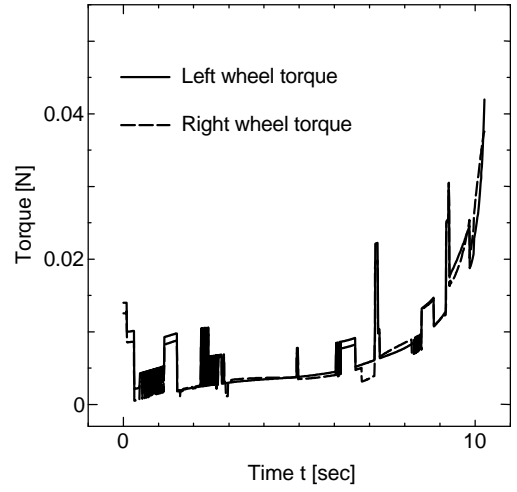


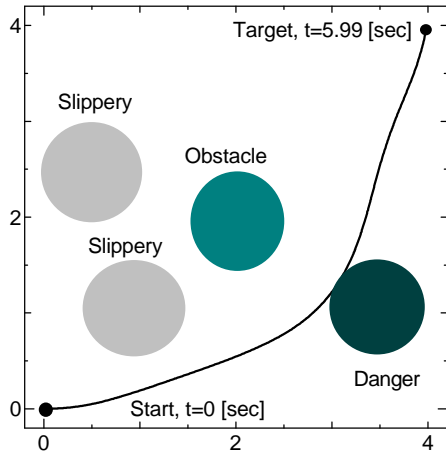
Fig. 3 Learning at each trial.



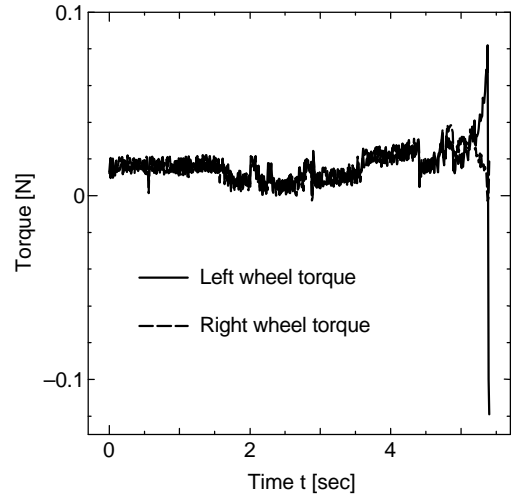
(a) Energy minimum behavior



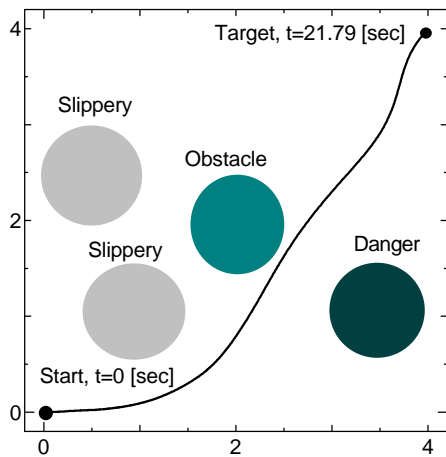
(a) Energy minimum behavior



(b) Hasty behavior

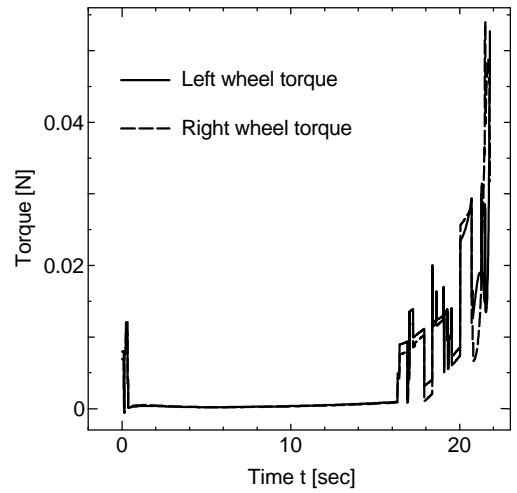


(b) Hasty behavior



(c) Safe behavior

Fig. 4 Simulation results.



(c) Safe behavior

Fig. 5 Torque profiles.

by scalars. Each component of vectors is in charge of each item of the evaluation (*reward*). Consequently, each component of the weight vectors and the output vector of the Q-net is adjusted based on reward or punishment for each item of the evaluation after a certain behavior is performed. The novelty of the proposed method is that the algorithm facilitated parallel learning of all behaviors while executing a single behavior. This novel feature is expected to accelerate the learning speed of multi-dimensional reinforcement learning algorithms. This kind of a cross-criticism is expected to be functioning in the human brain though there is no biological evidence is found so far. Yet, this phenomenon is seen in human learning through social interaction, where one updates its internal models by observing the behavior of others. In this method, the robot is able to change the optimal behavior according to the situation after the learning. Simulation results show the effectiveness of the proposed reinforcement learning.

### References

- [1] W. Schultz, P. Dayan, and P. R. Montague, "A Neural Substrate of Prediction and Reward," *Science*, vol. 275, pp. 1593-1599, 1997.
- [2] C. H. An, C. G. Atkeson, and J. M. Hollerbach, "Estimation of Internal Parameters of Rigid Body Links of Manipulators," *Artificial Intelligence Memo 887*, MIT Artificial Intelligence Laboratory, 1986.
- [3] R.S.Sutton and A.G.Barto, *Reinforcement Learning*, MIT Press, 1998.
- [4] C.J.C.H.Watkins, "Learning from Delayed Rewards", Ph.D. Dissertation, Cambridge University, 1989.
- [5] M.Asada, S.Noda, S.Tawaratumida, and K.Hosoda, "Positive Behavior Acquisition for a Real Robot by Vision-Based Reinforcement Learning", *Machine Learning*, vol.23, pp.279-303, 1996.
- [6] F.Saito and T.Fukuda, "Learning Architecture for Real Robotic Systems – Extension of Connectionist Q-Learning for Continuous Robot Control Domain", *Proc. of IEEE International Conference on Robotics and Automation*, pp.27-32, 1994.
- [7] S.Mahadevan and J.Connell, "Automatic Programming of Behavior-based Robots using Reinforcement Learning", *Proc. of 9<sup>th</sup> National Conf. on Artificial Intelligence*, pp.768-773, 1991.
- [8] L.J.Lin, "Reinforcement Learning for Robots Using Neural Networks", Ph.D. Dissertation, Carnegie Mellon University, 1992.
- [9] M.J.Mataric, "Interaction and Intelligent Behavior", Ph.D. Dissertation, MIT, 1994.
- [10] V.Gullapalli, J.A.Franklin, and H.Benbrahim, "Acquiring Robot Skills via Reinforcement Learning", *IEEE Control Systems Magazine*, vol.14, no.1, pp.13-24, 1994.
- [11] H.K.Beom and H.S.Cho, "A Sensor-Based Navigation for a Mobile Robot Using Fuzzy Logic and Reinforcement Learning", *IEEE Trans. on Systems, Man, and Cybernetics*, vol.25, pp.464-477, 1995.
- [12] Z.Kalmar, C.Szepesvari, and A.Lorincz, "Module-Based Reinforcement Learning: Experiments with a Real Robot", *Machine Learning*, vol.31, pp.55-85, 1998.
- [13] E.Uchibe and M.Asada, "Multiple Reward Criterion for Cooperative Behavior Acquisition in a Multiagent Environment", *Proc. of IEEE International Conf. on Systems, Man, and Cybernetics*, pp.VI 710-VI 715, 1999.