An Algorithm of Policy Gradient Reinforcement Learning with a Fuzzy Controller in Policies

Harukazu Igarashi *arashi50@sic.shibaura-it.ac.jp*
Department of Information Science and Engineering
Shibaura Institute of Technology
Tokyo, 135-8548, Japan

Seiji Ishihara *ishihara_s@mail.dendai.ac.jp*
Division of Science, Department of Science and Engineering
Tokyo Denki University
Saitama, 350-0394, Japan

Abstract

Typical fuzzy reinforcement learning algorithms take value-function based approaches, such as fuzzy Q-learning in Markov Decision Processes (MDPs), and use constant or linear functions in the consequent parts of fuzzy rules. Instead of taking such approaches, we propose a fuzzy reinforcement learning algorithm in another approach. That is the policy gradient approach. Our method can handle fuzzy sets even in the consequent part and also learn the rule weights of fuzzy rules. Specifically, we derived learning rules of membership functions and rule weights for both cases when input/output variables to/from the control system are discrete and continuous.

Keywords: Reinforcement Learning, Policy Gradient Method, Fuzzy Inference, Membership Function.

1. INTRODUCTION

Much work [3-7] has been done combining fuzzy control [1] and reinforcement learning algorithms [2]. Combining benefits fuzzy control systems in that parameters included in membership functions in fuzzy control rules can be learned by reinforcement learning even if there is no teacher data in the input and output of the control system. For reinforcement learning, fuzzy rules expressed by linguistic terms are very convenient for experts to introduce a priori knowledge in the rule database and for system users to understand the if-then rules. In particular, they are appropriate for building control systems with continuous and layered system states [8].

Most combining methods proposed thus far have taken approaches using value-based reinforcement learning, such as *Q*-learning that assumes Markov Decision Processes (MDPs) for the environments and the policies of agents [3-8]. The fuzzy rules in those works usually describe system states in the antecedent part and parameter values [3,4] or functions [5] corresponding to *Q* values in the consequent part. However, fuzzy sets were not allowed to describe output variables in the consequent part. The system calculated *Q* values only for discrete actions of agents. Moreover, there were no weight parameters representing the confidence or importance of the rules that can reinforce suitable rules, suppress unsuitable rules, generate new rules, and remove unnecessary rules.

In reinforcement learning, there is another approach other than the value-based approach. The policy gradient method, which originates from Williams' REINFORCE algorithm [9], is an approach that computes the policy gradient with respect to parameters in the policy function and improves the policy by adjusting the parameters in the gradient direction [9-11]. A combining method is proposed by Wang et al. [12] using a policy gradient method called the GPOMDP algorithm that was proposed by Baxter and Bartlett [10]. However, agent actions were restricted

to be discrete, fuzzy sets were not allowed to be used in the consequent part of the control rules, and the weight parameters of rules were not considered at all.

To compensate for these imperfections, this paper proposes a combining method of fuzzy control and reinforcement learning based on the policy gradient method described in Ref. [11]. Our combining method allows fuzzy sets for describing agent actions in the consequent part, and can also learn the weight parameters of the rules.

This paper is organized as follows: Section 2 describes a policy gradient method to be extended to the fuzzy control system in later sections. Details of the extension are described in Section 3. Learning rules of discrete and continuous membership functions are derived in Sections 4 and 5, respectively. Section 4 also describes the learning rules of the weight parameters of the fuzzy control rules. Section 6 discusses management of the rules and Section 7 is a summary of this paper and our future work.

2. POLICY GRADIENT METHOD

2.1 Policy and Learning Rules

A policy gradient method is a kind of reinforcement learning scheme originated from Williams' REINFORCE algorithm [9]. The method locally increases and maximizes the expected reward by calculating the derivatives of the expected reward function of the parameters included in a stochastic policy function. This method has a firm mathematical basis and is easily applied to many learning problems. It can be used for learning problems even in the non-MDPs by Igarashi et al. [11][13]. In their work, they proposed a policy gradient method that calculates the derivatives of the expected reward function per episode—not per unit time—to maximize the expected reward that does not have Markovian property. The reward r given at time t can depend on not only the current state $s(t)$, but also history $h(t)$, which is a set of all the past states and actions of an agent in an episode. Moreover, this method can be applied to cases with non-Markovian state transition probabilities of the environment and non-Markovian policies of an agent. It has been applied to pursuit problems, where the policy function consists of state-action rules with weight coefficients and potential functions of agent states [13].

Following Ref. [11], let us assume discrete time t, agent state $s(t)$, and agent action $a(t)$. A stochastic policy given by a Boltzmann distribution function,

$$\pi\left(a(t);s(t),h(t),\omega\right) \equiv \frac{e^{-E\left(a(t);s(t),h(t),\omega\right)/T}}{\sum_{a\in A} e^{-E\left(a;s(t),h(t),\omega\right)/T}} \tag{1}$$

controls an agent's action. ω are parameters to be learned and T is a parameter called *temperature*. The learning rule to maximize the expected reward per episode,

$$\Delta\omega = \varepsilon \cdot r \cdot \sum_{t=0}^{L-1} e_{\omega}(t), \tag{2}$$

is shown in Refs. [9] and [11], where L is a time-step size, called the episode length, and ε is a small positive number called the learning rate. $e_{\omega}(t)$ are characteristic eligibilities [9] defined by

$$e_{\omega}(t) \equiv \partial \ln \pi\left(a(t);s(t),h(t),\omega\right)/\partial\omega. \tag{3}$$

Parameters ω are updated at the end of each episode by the learning rules in (2).

2.2 Policy Expressed by If-Then Rules

Let policy π, which determines an agent's action at each time, be expressed by the data base of if-then type rules as "if an agent's state is *s*, then it takes action *a*." In this paper, we deal with only π that does not depend on history $h(t)$. However, discussion in Section 3 and later can be easily extended to non-Markovian policies.

Rule *i* is discriminated by discrete state *s* and discrete action *a* as $i=(s,a)$, and has weight parameter $\theta_i = \theta(s,a)$. If more than one rule matches the current agent state $s(t)$, their firing probabilities depend on the rule weight parameters. Such stochastic policy can be given if objective function $E(a(t);s(t),\theta)$ is defined by

$$E\left(a(t);s(t),\theta\right) = -\theta\left(s(t),a(t)\right). \tag{4}$$

This objective function can be written as [13]

$$E\left(a(t);s(t),\theta\right) = -\sum_{s}\sum_{a}\theta(s,a)\delta_{s,s(t)}\delta_{a,a(t)}, \tag{5}$$

where $\delta_{x,y}$ takes 1 if $x=y$ and 0 otherwise. The right-hand side is considered as an integration method of results of the inference rules. The learning rule of weight parameter $\theta(s,a)$ was derived in [13] as

$$\Delta\theta(s,a) = \varepsilon \cdot r \cdot \sum_{t=0}^{L-1} e_{\theta(s,a)}(t), \tag{6}$$

where

$$e_{\theta(s,a)}(t) = \delta_{s,s(t)}\left[\delta_{a,a(t)} - \pi\left(a;s(t),\theta\right)\right] / T. \tag{7}$$

3. POLICY GRADIENT METHOD EXTENDED TO FUZZY CONTROL SYSTEMS

3.1 Basic Principles

Much work [3-7] has been done combining fuzzy control and reinforcement learning algorithms. A typical method is Fuzzy Q-Learning proposed by Jouffe [3]. However, fuzzy sets were not allowed to describe output variables in the consequent part. Actions must be discretized and Q parameters must be prepared for all actions.

In this paper, we propose an inference system for combining fuzzy control and reinforcement learning based on the following four characteristics:

i) allowing fuzzy-set expressions in both the antecedent part and the consequent part in system control rules;

ii) selecting the output of the system by a stochastic policy;

iii) learning the membership functions of fuzzy sets in both the antecedent part and the consequent part; and

iv) taking account of rule weight parameters and learning them.

The stochastic selection in ii) has been already introduced in [4] rather than determining the output value by the centroid computation that is frequently used, but sometimes reported to bring incorrect and undesirable results. Learning membership functions in the antecedent part is not dealt with in Refs. [3] and [12]. The introduction of rule weights in iv) is for growing or removing control rules by learning.

We consider the following control rules of the system:

Rule i:

if (x_1 is A_1^i) and …and (x_M is A_M^i) then (y_1 is B_1^i) and…and (y_N is B_N^i) with θ_i, (8)

where $x=(x_1,\ldots,x_M)/y=(y_1,\ldots,y_N)$ is the input/output of the control system and corresponds to state *s*/action *a* in reinforcement learning. This paper deals with the case where membership functions $\{A_j^i\}$ and $\{B_j^i\}$ do not depend on each other. Rules do not share an identical membership function. However, the same formalization in this paper is possible and easily extended to cases where multiple rules share an identical membership function with each other.

3.2 Objective Function and Policy

Instead of (4), we propose the following objective function:

$$E\left(y(t);x(t),\theta,A,B\right)=-\sum_{i=1}^{n_R}\theta_i A^i\left(x(t)\right)B^i\left(y(t)\right), \tag{9}$$

where n_R is the number of rules in the rule database and $x(t)/y(t)$ is the input/output of the control system at time *t*. $A^i(x)/B^i(y)$ is the degree of truth value of the antecedent/consequent part in the *i*-th rule and is defined by the products of membership functions $\{A_j^i\}/\{B_j^i\}$ as

$$A^i\left(x\right)\equiv\prod_{j=1}^{M}A_j^i\left(x_j\right) \tag{10}$$

and

$$B^i\left(y\right)\equiv\prod_{j=1}^{N}B_j^i\left(y_j\right). \tag{11}$$

The product in (10)/(11) means that the truth value of the antecedent/consequent part is calculated by the product of the degrees of inclusion in fuzzy sets A_j^i/B_j^i of input/output variable x_j/y_j.

The objective function in (9) indicates how much all rules support output value *y* when *x* is input to the control system. If you compare (9) with (5), θ_i, $A^i(x)$, and $B^i(y)$ in (9) correspond to $\theta(s,a)$, $\delta_{s,s(t)}$, and $\delta_{a,a(t)}$ in (5), respectively. This means that the objective function in (9) is a natural extension of (5) to fuzzy inference systems. Table 1 shows a correspondence relation between the proposed combining method and the policy gradient method described in Refs. [11] and [13].

	Combining method	**Policy gradient[11][13]**
label of variables	fuzzy	non-fuzzy
(a) antecedent part	input $x(t)$	state $s(t)$
(b) consequent part	output $y(t)$	action $a(t)$
rule identifier	i	(s,a)
rule weight	θ_i	$\theta(s,a)$
truth value of (a)	$A^i(x(t))$	$\delta_{s,s(t)}$
truth value of (b)	$B^i(y(t))$	$\delta_{a,a(t)}$

TABLE 1: Correspondence relation between the proposed combining method and the policy gradient method described in Refs. [11] and [13].

The control system determines output $y(t)$ for input $x(t)$ stochastically. The policy for the system is given by a Boltzmann distribution function with the objective function in (9), i.e.,

$$\pi\left(y\left(t\right);x\left(t\right),\theta,A,B\right)\equiv\frac{e^{-E\left(y(t);x(t),\theta,A,B\right)/T}}{\sum_{y}e^{-E\left(y;x(t),\theta,A,B\right)/T}}. \tag{12}$$

Imai et al. applied the policy gradient method in Ref. [13] to pursuit problems [14]. They combined state-action rules from several knowledge resources to speed up learning. In their work, the grid world, where hunter and prey agents move around, was divided into crisp coarse regions. They used the regions as states in state-action rules that control the hunters' actions. A set of state-action rules defined on a different set of regions produces a different knowledge source. A state in the original state space activates rules on the knowledge sources and the inference results of the rules are combined to determine the hunter agents' actions. This work is a special case in our combining method. If $A^i(x)$ in (9) is a binary function that identifies whether an agent's position x is included in the i-th region and B^i(y) in (9) is a binary function that identifies whether an agent's action y is included in an action set {left, right, up, down}, objective function $E(y;x,\theta,A,B)$ in (9) corresponds exactly to the objective function combining rules described by the multiple knowledge sources proposed in Ref. [14].

3.3 Characteristic Eligibilities

Policy π in (12) includes weight θ_i and membership function $A^i(x)/B^i(y)$ in the antecedent/consequent part of the i-th rule. These parameters are denoted by ω. The learning rule of ω is given by (2) and (3) in the policy gradient method. Substituting (12) into (3), characteristic eligibility $e_\omega(t)$ at time t is written as

$$e_{\omega}\left(t\right)=-\frac{1}{T}\left[\frac{\partial E\left(y\left(t\right);x\left(t\right),\theta,A,B\right)}{\partial\omega}-\left\langle\frac{\partial E\left(y;x\left(t\right),\theta,A,B\right)}{\partial\omega}\right\rangle\right], \tag{13}$$

where

$$\left\langle\frac{\partial E\left(y;x\left(t\right),\theta,A,B\right)}{\partial\omega}\right\rangle\equiv\sum_{y}\frac{\partial E\left(y;x\left(t\right),\theta,A,B\right)}{\partial\omega}\pi\left(y;x\left(t\right),\theta,A,B\right). \tag{14}$$

We derive the detailed expression of $e_\omega(t)$ in Sections 4 and 5. In Section 4, input x and output y are discrete variables and, in Section 5, they are continuous.

4. LEARNING RULES

4.1 Rule Weight Parameter θ_i

Characteristic eligibilities $e_\omega(t)$ are obtained by substituting objective function (9) into (13). Substituting the expression of $e_\omega(t)$ into learning rule (2), a learning rule of rule weight parameter θ_i is given as

$$\Delta\theta_i=\varepsilon\cdot r\cdot\sum_{t=0}^{L-1}e_{\theta_i}\left(t\right), \tag{15}$$

where

$$e_{\theta_i}\left(t\right)=A^i\left(x\left(t\right)\right)\left[B^i\left(y\left(t\right)\right)-\left\langle B^i\right\rangle\left(t\right)\right]/T \tag{16}$$

and

$$\left\langle B^i \right\rangle(t) \equiv \sum_y B^i(y)\pi\left(y; x(t), \theta, A, B\right). \tag{17}$$

The meanings of learning rules (15)-(17) are as follows: The degrees of reward r and the truth value in the antecedent part, $A^i(x(t))$, control the amount of update of θ_i. Rule weights that match $x(t)$ very well that were actually input to the control system in an episode are strongly reinforced. The degree of truth value in the consequent part, $B^i(y)$, determines the direction of the update. If the truth value $B^i(y)$ with respect to $y=y(t)$ is stronger than the expectation $\langle B^i \rangle(t)$ defined by (17), the i-th rule's weight θ_i is reinforced. If not, θ_i is suppressed.

This means that rules with large truth values both in matching input $x(t)$ and output $y(t)$ selected by the control system are largely reinforced in episodes where high reward values are given to the system. However, rules that do not match output $y(t)$ actually selected in the episodes are suppressed as the rules produce undesirable output y even if the rules match input $x(t)$ very well.

Let us note that one can easily confirm that Eqs. (16) and (17) lead to Eqs. (6) and (7) by using the correspondences in Table 1. This affirms that the objective function proposed in (9) is a very natural extension from the objective function (4) in the non-fuzzy policy gradient method to one expressed by fuzzy sets.

4.2 Membership Functions in the Antecedent Part

Substituting (9) and (10) into (13), the gradient of the objective function with respect to $A^i_j(x)$, which is a value of a membership function at input x in the antecedent part of fuzzy control rules, and its expectation value are given as

$$\frac{\partial E\left(y(t); x(t), \theta, A, B\right)}{\partial A^i_j(x)} = \frac{\partial}{\partial A^i_j(x)}\left[-\sum_{k=1}^{N}\theta_k A^k\left(x(t)\right)B^k\left(y(t)\right)\right] = -\theta_i \cdot \delta_{x(t),x}\prod_{l\neq j} A^i_l(x)\cdot B^i\left(y(t)\right) \tag{18}$$

and

$$\left\langle \frac{\partial E\left(y; x(t), \theta, A, B\right)}{\partial A^i_j(x)} \right\rangle = -\theta_i \cdot \delta_{x(t),x}\prod_{l\neq j} A^i_l(x)\cdot \left\langle B^i \right\rangle(t). \tag{19}$$

Substituting (18) and (19) into (2) and (13), a learning rule for $A^i_j(x)$ and its characteristic eligibilities are given as

$$\Delta A^i{}_j(x) = \varepsilon \cdot r \cdot \sum_{t=0}^{L-1} e_{A^i{}_j(x)}(t) \tag{20}$$

and

$$e_{A^i{}_j(x)}(t) = \theta_i \cdot \delta_{x(t),x}\prod_{l\neq j} A^i_l(x)\cdot\left[B^i\left(y(t)\right) - \left\langle B^i \right\rangle(t)\right]\Big/T. \tag{21}$$

The meanings of learning rules (20) and (21) are as follows: Only values of $A^i_j(x)$ at values of x actually input to the control system during an episode are updated by the learning rule in (20) and (21). As in the case of learning rule weight parameters discussed in Section 4.1., degrees of reward r, rule weight θ_i, and truth value in the antecedent part, $A^i(x(t))$, control the amount of

update of $A^i_j(x)$. That increases for a large reward, a large rule weight, and how well the rule matches $x(t)$ that appeared in an episode, except for the j-th component $A^i_j(x)$.

The degree of truth value in the consequent part, $B^i(y)$, determines the update direction. If the truth value $B^i(y)$ at $y=y(t)$ is larger than the expectation value $\langle B^i \rangle(t)$ defined by (17), the i-th rule's weight θ_i is reinforced. If not, $A^i_j(x)$ is suppressed.

4.3 Membership Functions in the Consequent Part

Substituting (9) and (11) into (13), the gradient of the objective function with respect to $B^i_j(y)$, which is the value of the membership function at input y in the consequent part of fuzzy control rules, and its expectation are given as

$$\frac{\partial E\left(y(t);x(t),\theta,A,B\right)}{\partial B^i_j(y)} = \frac{\partial}{\partial B^i_j(y)}\left[-\sum_{k=1}^{N}\theta_k A^k\left(x(t)\right)B^k\left(y(t)\right)\right] = -\theta_i A^i\left(x(t)\right)\delta_{y(t),y}\prod_{l\neq j}B^i_l(y) \quad (22)$$

and

$$\left\langle\frac{\partial E\left(y;x(t),\theta,A,B\right)}{\partial B^i_j(y)}\right\rangle = -\theta_i A^i\left(x(t)\right)\prod_{l\neq j}B^i_l(y)\pi\left(y;x(t),\theta,A,B\right). \quad (23)$$

Substituting (22) and (23) into (2) and (13), the learning rules for $B^i_j(y)$ and its characteristic eligibilities are given as

$$\Delta B^i_j(y) = \varepsilon\cdot r\cdot\sum_{t=0}^{L-1}e_{B^i_j(y)}(t) \quad (24)$$

and

$$e_{B^i_j(y)}(t) = \theta_i\cdot A^i\left(x(t)\right)\cdot\left[\delta_{y(t),y}-\pi\left(y;x(t),\theta,A,B\right)\right]\prod_{l\neq j}B^i_l(y)\Big/T \quad . \quad (25)$$

The meanings of learning rules (24) and (25) are as follows: As in the case of learning $A^i_j(x)$, degrees of reward r, rule weight θ_l, and the truth value in the antecedent part controls the amount of update of $B^i_j(y)$. In addition, the degree of truth value in the consequent part, except j-th component $B^i_j(y)$, also controls the amount. Therefore, it increases for a large reward, a large rule weight, and how well the rule matches $x(t)$ and $y(t)$, except the j-th component that appeared in an episode. Moreover, $B^i_j(y)$ is reinforced if $y=y(t)$, while $B^i_j(y)$'s at values of output y that competed against $y(t)$ are all suppressed.

5. LEARNING RULES OF PARAMETERS IN CONTINUOUS MEMBERSHIP FUNCTIONS

Learning rules in Section 4 are derived under an assumption that input x and output y are discrete variables. In this section, we derive learning rules when x and y are continuous. In such a case, membership functions $A^i_j(x)$ and $B^i_j(y)$ are continuous functions of x and y. For example, we consider the membership functions as

$$\mu(x;b,c,m) \equiv 1\Big/\left[1+b\left(x-c\right)^m\right], \quad (26)$$

which are frequently used in fuzzy control. In (26), m is an even integer and the targets of learning are parameters b (>0) and c.

Now, membership functions $A^i_j(x)$ and $B^i_j(y)$ are continuous functions, shown in (26), and have parameters α^i_j and β^i_j for *b* and *c* in (26). We define an objective function as

$$E\left(y(t);x(t),\theta,A,B\right)=-\sum_{i=1}^{n_R}\theta_i A^i\left(x(t);\alpha^i\right)B^i\left(y(t);\beta^i\right), \tag{27}$$

where

$$A^i\left(x;\alpha^i\right)\equiv\prod_{j=1}^{M}A^i_j\left(x_j;\alpha^i_j\right) \tag{28}$$

and

$$B^i\left(y;\beta^i\right)\equiv\prod_{j=1}^{N}B^i_j\left(y_j;\beta^i_j\right). \tag{29}$$

Substituting (27) into (13) and (14), characteristic eligibilities are obtained as

$$e_{\alpha^i_j}(t)=(1/T)\cdot\theta_i A^i\left(x(t);\alpha^i\right)\cdot\partial\ln A^i_j\left(x_j(t);\alpha^i_j\right)\Big/\partial\alpha^i_j\cdot\left[B^i\left(y(t);\beta^i\right)-\left\langle B^i\left(y;\beta^i\right)\right\rangle(t)\right] \tag{30}$$

and

$$e_{\beta^i_j}(t)=(1/T)\cdot\theta_i A^i\left(x(t);\alpha^i\right)\cdot \left[\prod_{l\neq j}B^i_l\left(y(t);\beta^i\right)\partial B^i_j\left(y_j(t);\beta^i_j\right)\Big/\partial\beta^i_j-\left\langle\prod_{l\neq j}B^i_l\left(y;\beta^i\right)\partial B^i_j\left(y_j;\beta^i_j\right)\Big/\partial\beta^i_j\right\rangle(t)\right], \tag{31}$$

where

$$\left\langle f(y)\right\rangle(t)\equiv\int_{-\infty}^{+\infty}f(y)\pi\left(y;x(t),\theta,A,B\right)dy. \tag{32}$$

Characteristic eligibilities of rule weight θ_i are given by (16), but (17) is replaced by (32). That is

$$e_{\theta_i}(t)=A^i\left(x(t)\right)\left[B^i\left(y(t)\right)-\left\langle B^i\right\rangle(t)\right]\Big/T \tag{33}$$

and

$$\left\langle B^i(y)\right\rangle(t)\equiv\int_{-\infty}^{+\infty}B^i(y)\pi\left(y;x(t),\theta,A,B\right)dy. \tag{34}$$

Learning rules of α^i_j, β^i_j, and θ_i are given by substituting (30), (31), and (33) into (2) as

$$\Delta\alpha^i_j=\varepsilon\cdot r\cdot\sum_{t=0}^{L-1}e_{\alpha^i_j}(t), \tag{35}$$

$$\Delta\beta_j^i = \varepsilon \cdot r \cdot \sum_{t=0}^{L-1} e_{\beta_j^i}(t) \tag{36}$$

and

$$\Delta\theta_i = \varepsilon \cdot r \cdot \sum_{t=0}^{L-1} e_{\theta_i}(t). \tag{37}$$

6. MANAGEMENT OF RULES

In this section, we consider the management of fuzzy rules in the policy. Rule management means how to remove unnecessary rules, generate new rules, and merge two rules.

Rules should be removed if their weight parameters are decreased to near zero by learning. Rules whose truth values in the antecedent part and the consequent part are constantly very small in all episodes are also to be removed from the database of fuzzy control rules in the policy. If all rules cannot match input x, then a good idea generates a fuzzy rule whose antecedent part includes a fuzzy set whose center locates at x [15].

There are two ideas for merging two rules into one rule. First, if there is a strong positive correlation in the truth value of the antecedent and the consequent part between two rules, then remove one of the two rules. Second, if there is a strong positive correlation in the truth value of the consequent part between two rules and their membership functions $A^i(x)$ are adjacent to each other, then one can merge their membership functions into a new $A^i(x)$ and replace the two rules with a new one.

7. CONCLUSION

This paper has extended a policy gradient method with weight parameters of *if-then* type controlling rules in the objective function to a case where the rules are described by fuzzy sets. Learning rules of membership functions and rule weights are derived for both cases when input/output variables to/from the control system are discrete and continuous.

In the future, we plan to verify the learning rules derived here by applying them to examples such as robot soccer games [16] and car velocity control problems. Moreover, we will try to extend the theory proposed here to multistage fuzzy inference, multilayer control systems, and multiagent systems.

8. REFERENCES

[1] R. R. Yager and L. A. Zadeh. *An Introduction to Fuzzy Logic Applications in Intelligent Systems*. Norwell, MA, USA: Kluwer Academic Publishers, 1992.

[2] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[3] L. Jouffe. “Fuzzy Inference System Learning by Reinforcement Methods.” *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, vol. 28, No. 3, pp. 338-355, 1998.

[4] C. Oh, T. Nakashima, and H. Ishibuchi. “Initialization of Q-values by Fuzzy Rules for Accelerating Q-learning.” in Proc. IEEE World Congress on Computational Intelligence, vol. 3, 1998, pp. 2051-2056.

[5] T. Horiuchi, A. Fujino, O. Katai, and T. Sawaragi. “Fuzzy Interpolation-based Q-learning with Continuous States and Actions,” in Proc. the Fifth Inter. Conf. on Fuzzy Systems, 1996, vol. 1, pp. 594-600.

[6] Y. Hoshino and K. Kamei. “A Proposal of Reinforcement Learning with Fuzzy Environment Evaluation Rules and Its Application to Chess.” *J. of Japan Society for Fuzzy Theory and Systems*, vol. 13, no. 6, pp. 626-632, 2001. (in Japanese)

[7] H. R. Berenji. “A Reinforcement Learning-based Architecture for Fuzzy Logic Control.” *Int. J. Approx. Reasoning*, vol. 6, pp. 267–292, 1992.

[8] H. R. Berenji and D. Vengerov. “Cooperation and Coordination Between Fuzzy Reinforcement Learning Agents in Continuous State Partially Observable Markov Decision Processes,” in 1999 IEEE Int. Fuzzy Systems Conf. Proc., 1999, vol. 2, pp. 621-627.

[9] R.J. Williams. “Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning.” *Machine Learning*, vol. 8, pp. 229-256, 1992.

[10] J. Baxter and P. L. Bartlett. “Infinite-Horizon Policy- Gradient Estimation,” *Journal of Artificial Intelligence Research*, vol. 15, pp. 319-350, 2001.

[11] H. Igarashi, S. Ishihara, and M. Kimura. “Reinforcement Learning in Non-Markov Decision Processes-Statistical Properties of Characteristic Eligibility.” *IEICE Transactions on Information and Systems*, vol. J90-D, no. 9, pp. 2271-2280, 2007. (in Japanese)

(This paper is translated into English and included in *The Research Reports of Shibaura Institute of Technology, Natural Sciences and Engineering*, vol. 52, no. 2, pp. 1-7, 2008.)

[12] X. Wang, X. Xu, and H. He. “Policy Gradient Fuzzy Reinforcement Learning,” in Proc. 3rd Inter. Conf. on Machine Learning and Cybernetics, 2004, pp. 992-995.

[13] S. Ishihara and H. Igarashi, “Applying the Policy Gradient Method to Behavior Learning in Multiagent Systems: The Pursuit Problem.” *Systems and Computers in Japan*, vol. 37, no. 10, pp. 101-109, 2006.

[14] S. Imai, H. Igarashi, and S. Ishihara. “Policy-Gradient Method Integrating Abstract Information in Policy Function and Its Application to Pursuit Games with a Tunnel of Static Obstacles.” *IEICE Transactions on Information and Systems*, vol. J94-D, no. 6, pp. 968-976, 2011. (in Japanese).

This paper is translated into English and included in *The Research Reports of Shibaura Institute of Technology, Natural Sciences and Engineering*, vol. 52, no. 2, pp. 7-12, 2011.

[15] Y. Hosoya, T. Yamamura, M. Umano, and K. Seta. “Reinforcement Learning Based on Dynamic Construction of the Fuzzy State Space-Adjustment of Fuzzy Sets of States-,“ in Proc. of the 22nd Fuzzy System Symposium (CD-ROM), vol. 22, 8D3-1, 2006. (in Japanese).

[16] M. Sugimoto, H. Igarashi, S. Ishihara, K. Tanaka. “Policy Gradient Reinforcement Learning with a Fuzzy Controller for Policy: Decision Making in RoboCup Soccer Small Size League,” presented at the 29th Fuzzy System Symposium, Osaka, Japan, 2013. (in Japanese).