Fuzzy Inference Systems Composed of Double-Input Rule Modules for Obstacle Avoidance Problems

Hirofumi Miyajima, Takehiro Kawai, Noritaka Shigei, and Hiromi Miyajima

Abstract—**The purpose of self-tuning algorithm for fuzzy inference system is to construct automatically fuzzy inference rules from learning data based on the steepest descend method. Obvious drawbacks of the method are its large computational complexity and getting stuck in a shallow local minimum. Further, it is difficult to apply for the conventional method to the problem with a large number of variables. In order to overcome them, the SIRMs (Single-Input Rule Modules) and DIRMs (Double-Input Rule Modules) models have been proposed. In some numerical simulations, it is shown that there exists the difference of the ability between DIRMs and SIRMs models. In this paper, we will apply DIRMs and SIRMs models to the control problem of obstacle avoidance. As a result, it is shown that DIRMs model is more effective than SIRMs model in this problem. Further, we propose a learning method to reduce the number of modules of DIRMs model and show the effectiveness in numerical simulations.**

Index Terms—**Fuzzy inference model, Single-input rule module, Small number of input rule module, Double input rule module, obstacle avoidance.**

I. Introduction

MANY studies on self-tuning fuzzy systems have been made [1], [2]. The aim of these studies is to construct automatically fuzzy inference rules from input and output data based on the steepest descend method. Obvious drawbacks of the method are its large computational complexity and getting stuck in a shallow local minimum. Further, there is a problem that the number of fuzzy rules increases with increasing of input variables [3]–[5]. In order to overcome them, some novel methods have been developed which 1) create fuzzy rules one by one starting from any number of rules [6], 2) delete fuzzy rules one by one starting from a sufficiently large number of rules [7] , 3) use GA and PSO to determine the structure of the fuzzy model [4], [15], 4) use a self-organization or a vector quantization technique to determine the initial assignment of fuzzy rules [8], and 5) use generalized objective functions [9]. However, there are little studies on effective learning methods of fuzzy inference systems dealing with a large number of input variables; in most of the conventional methods, fuzzy inference systems deal with a small number of input variables. Therefore, some methods have been proposed as shown in the references [3]–[5]. The SIRMs (Single-Input Rule Modules) model aims to obtain a better solution by using fuzzy inference system composed of SIRMs [10], [11]. Further, with SIRMs model there is the advantage to be able to apply easily to the problems with a large number of variables. However, it is known that the SIRMs model does not always achieve good performance in non-linear problems. Therefore, we have proposed the SNIRMs (Small Number of Input Rule Modules) model as a generalized SIRMs model, in which each module is composed of small number of input variables [12]–[14]. DIRMs (Double-Input Rule Modules) model is an example of such models and each module of DIRMs model is composed of two input variables. It is well known that EX-OR problem with two input variables can be approximated by DIRMs model but not by SIRMs model [13]. Further, there exists the difference of the ability between DIRMs and SIRMs models [13], [14]. Then, does there exist such example in control problems? In this paper, we consider the obstacle avoidance problem as an example of such problems. The problem is how the agent (or robot) avoids the obstacle and arrives at the specified point. We will show that DIRMs and its reduced models are also superior in control problem to the conventional SIRMs model. In section 2, the conventional fuzzy inference model and its learning method are introduced. In section 3, SIRMs, DIRMS and SNIRMs models are explained and the variable increase method for DIRMs model is proposed. In section 4, in order to compare the capability between SIRMs and DIRMs models, implementing EX-OR problem with two variables and numerical simulation of two category problems are discussed. Further, numerical simulations of obstacle avoidance for SIRMS and DIRMs models and the proposed method are performed, and the generalization capability of them are clarified by test simulations of obstacle avoidance.

Manuscript received July 8, 2014

Hirofumi Miyajima is with Graduate School of Science and Engineering, Kagoshima University, 1-21-40 Korimoto, Kagoshima 890-0065, Japan (e-mail: k3768085@kadai.jp).

T. Kawai was with Graduate School of Science and Engineering, Kagoshima University, 1-21-40 Korimoto, Kagoshima 890-0065, Japan (e-mail: k8005442@kadai.jp).

N. Shigei is with Graduate School of Science and Engineering, Kagoshima University, 1-21-40 Korimoto, Kagoshima 890-0065, Japan (e-mail: shigei@eee.kagoshima-u.ac.jp).

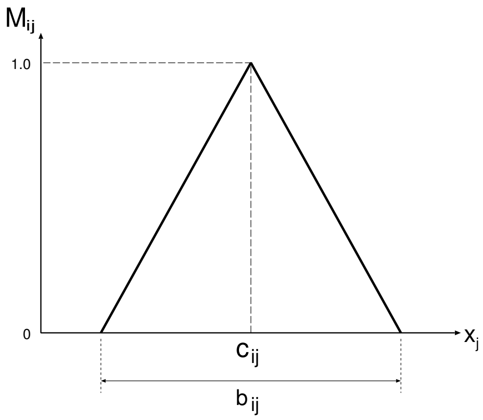
Hiromi Miyajima is with Graduate School of Science and Engineering, Kagoshima University, 1-21-40 Korimoto, Kagoshima 890-0065, Japan (corresponding author to provide e-mail: miya@eee.kagoshima-u.ac.jp).

II. Fuzzy Inference Model and its Learning

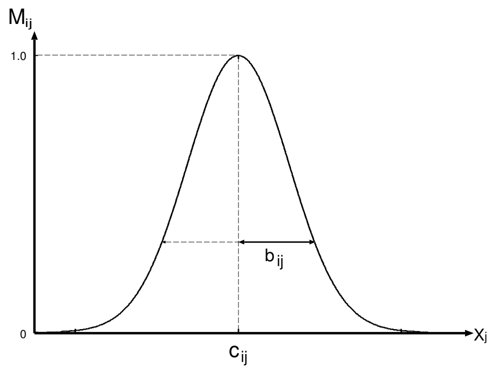
A. Fuzzy Inference Model

The conventional fuzzy reasoning model using the delta rule is described [1], [3], [4]. Let $Z_j = \{1, \cdots, j\}$ for the positive integer j. Let $\boldsymbol{x} = (x_1, \cdots, x_m)$ and y be input and output data, respectively, where x_i for $i \in Z_m$ and y are real number. Then the rule of simplified fuzzy inference model is expressed as

$$R_j \ : \text{ if } x_1 \text{ is } M_{1j} \text{ and } \cdots \ x_m \text{ is } M_{mj} \text{ then } y \text{ is } w_j, \quad (1)$$



(a) Triangular membership function



(b) Gaussian membership function

Fig. 1. Membership functions

where $j \in Z_n$ is a rule number, $i \in Z_m$ is a variable number, M_{ij} is a membership function of the antecedent part, and w_j is the weight of the consequent part.

A membership value of the antecedent part μ_j for input $\boldsymbol{x}$ is expressed as follows:

$$\mu_j = \prod_{i=1}^{m} M_{ij}(x_i) \tag{2}$$

where M_{ij} is the membership function of the antecedent part. Let c_{ij} and b_{ij} denote the center and the width values of M_{ij}, respectively. If the triangular membership function is used, then M_{ij} is expressed as

$$M_{ij}(x_i) = \begin{cases} 1 - \frac{2 \cdot \left| x_i - c_{ij} \right|}{b_{ij}} & (c_{ij} - \frac{b_{ij}}{2} \leq x_j \leq c_{ij} + \frac{b_{ij}}{2}) \\ 0 & \text{(otherwise).} \end{cases} \tag{3}$$

Further, if Gaussian membership function is used, then M_{ij} is expressed as follow:

$$M_{ij} = \exp\left(-\frac{1}{2} \left(\frac{x_j - c_{ij}}{b_{ij}} \right)^2 \right) \tag{4}$$

See Fig.1(a) and (b) for Eqs.(3) and (4), respectively.

The output y^* of fuzzy inference is calculated by the following equation.

$$y^* = \frac{\sum_{j=1}^{n} \mu_j \cdot w_j}{\sum_{j=1}^{n} \mu_j} \tag{5}$$

The objective function E is defined to evaluate the inference error between the desirable output y^r and the inference output y^*.

$$E = \frac{1}{2} (y^* - y^r)^2 \tag{6}$$

In order to minimize the objective function E, the parameters $\alpha \in \{c_{ij}, b_{ij}, w_j\}$ are updated based on the descent method [3].

$$\alpha(t+1) = \alpha(t) - K_\alpha \frac{\partial E}{\partial \alpha} \tag{7}$$

where t is iteration times and K_α is a constant. When the Gaussian membership function is used, the following equations are obtained:

$$\frac{\partial E}{\partial w_j} = \frac{\mu_j}{\sum_{j=1}^{n} \mu_j} \cdot (y^* - y^r) \tag{8}$$

$$\frac{\partial E}{\partial c_{ij}} = \frac{\mu_j}{\sum_{j=1}^{n} \mu_j} \cdot (y^* - y^r) \cdot (w_j - y^*) \cdot \frac{x_j - c_{ij}}{b_{ij}^2} \tag{9}$$

$$\frac{\partial E}{\partial b_{ij}} = \frac{\mu_j}{\sum_{j=1}^{n} \mu_j} \cdot (y^* - y^r) \cdot (w_j - y^*) \cdot \frac{(x_j - c_{ij})^2}{b_{ij}^3} \tag{10}$$

When the triangular membership function is used, the following equations are obtained:

$$\frac{\partial E}{\partial c_{ij}} = \frac{\mu_j}{\sum_{j=1}^{n} \mu_j} \cdot (y^* - y^r) \cdot (w_j - y^*) \cdot \frac{2\mathrm{sgn}(x_i - c_{ij})}{b_{ij} \cdot M_{ij}(x_i)}, \tag{11}$$

$$\frac{\partial E}{\partial b_{ij}} = \frac{\mu_j}{\sum_{j=1}^{n} \mu_j} \cdot (y^* - y^r) \cdot (w_j - y^*) \cdot \frac{1 - M_{ij}(x_i)}{M_{ij}(x_i) \cdot b_{ij}} \tag{12}$$

where

$$\mathrm{sgn}(z) = \begin{cases} -1 & ; \ z < 0 \\ 0 & ; \ z = 0 \\ 1 & ; \ z > 0. \end{cases} \tag{13}$$

B. The conventional leaning method

In this section, we describe the detailed learning algorithm described in the previous section. A target data set $D = \{(x_1^p, \cdots, x_m^p, y_p^r) | p \in Z_P\}$ is given in advance. The objective of learning is minimizing the following error.

$$E = \frac{1}{P} \sum_{p=1}^{P} (y_p^* - y_p^r)^2. \tag{14}$$

The conventional learning algorithm is shown below [3].

Learning Algorithm A

Step 1: The initial number of rules, c_{ij}, b_{ij} and w_j are set randomly. The threshold Θ_1 for inference error is given. Let T_{max} be the maximum number of learning times. The learning coefficients K_c, K_b and K_w are set.

Step 2: Let $t = 1$.

Step 3: Let $p = 1$.

Step 4: An input and output data $(x_1^p, \cdots, x_m^p, y_p^r)$ is given.

Step 5: Membership value of each rule is calculated by Eqs.(2) and (3) or (4).

Step 6: Inference output y_p^* is calculated by Eq.(5).

Step 7: Real number w_j is updated by Eq. (8).

Step 8: Parameters c_{ij} and b_{ij} are updated by Eqs.(9) and (10) or Eqs.(11) and (12).

Step 9: If $p = P$ then go to the next step. If $p < P$ then $p \leftarrow p + 1$ and go to Step 4.

Step 10: Inference error $E(t)$ is calculated by Eq.(14). If $E(t) \leq \theta_1$ then learning is terminated.

Step 11: If $t \neq T_{max}$ then $t \leftarrow t + 1$ and go to Step 3. Otherwise learning is terminated.

III. The SNIRMs and DIRMs Models

The SNIRMs, SIRMs and DIRMs models are introduced [13], [14]. Let U_k^m be the set of all ordered k-tuples of Z_m, that is

$$U_k^m = \{l_1 \cdots l_k | l_i < l_j \text{ if } i < j\}. \tag{15}$$

Then, each rule of SNIRMs model for U_k^m is defined as follows:

$$\begin{aligned} &\text{SNIRM}-l_1 \cdots l_k : \\ &\quad \{R_i^{l_1 \cdots l_k} : \text{if } x_{l_1} \text{ is } M_i^{l_1} \text{ and } \cdots \text{ and } x_{l_k} \text{ is } M_i^{l_k} \\ &\qquad\qquad \text{then } y_{l_1 \cdots l_k} \text{ is } w_i^{l_1 \cdots l_k}\}_{i=1}^n \end{aligned} \tag{16}$$

Example 1. For $U_1^3 = \{1, 2, 3\}$, the obtained system is as follows:

$$\begin{aligned} \text{SNIRM} \quad &- \quad 1 : \{R_i^1 : \text{if } x_1 \text{ is } M_i^1 \text{ then } y_1 \text{ is } w_i^1\}_{i=1}^n \\ \text{SNIRM} \quad &- \quad 2 : \{R_i^2 : \text{if } x_2 \text{ is } M_i^2 \text{ then } y_2 \text{ is } w_i^2\}_{i=1}^n \\ \text{SNIRM} \quad &- \quad 3 : \{R_i^3 : \text{if } x_3 \text{ is } M_i^3 \text{ then } y_3 \text{ is } w_i^3\}_{i=1}^n \end{aligned}$$

Example 2. For $U_2^3 = \{12, 13, 23\}$, the obtained system is as follows:

$$\begin{aligned} &\text{SNIRM} - 12 : \\ &\{R_i^{12} : \text{if } x_1 \text{ is } M_i^1 \text{ and } x_2 \text{ is } M_i^2 \text{ then } y_{12} \text{ is } w_i^{12}\}_{i=1}^n \\ &\text{SNIRM} - 13 : \\ &\{R_i^{13} : \text{if } x_1 \text{ is } M_i^1 \text{ and } x_3 \text{ is } M_i^3 \text{ then } y_{13} \text{ is } w_i^{13}\}_{i=1}^n \\ &\text{SNIRM} - 23 : \\ &\{R_i^{23} : \text{if } x_2 \text{ is } M_i^2 \text{ and } x_3 \text{ is } M_i^3 \text{ then } y_{23} \text{ is } w_i^{23}\}_{i=1}^n \end{aligned}$$

Let $\boldsymbol{x} = (x_1, \cdots, x_m)$. The fitness of the i-th rule and the output of SNIRM$-l_1 \cdots l_k$ are as follows:

$$\mu_i^{l_1 \cdots l_k} = M_i^{l_1}(x_{l_1}) M_i^{l_2}(x_{l_2}) \cdots M_i^{l_k}(x_{l_k}), \tag{17}$$

$$y_{l_1 \cdots l_k}^0 = \frac{\sum_{i=1}^n \mu_i^{l_1 \cdots l_k} w_i^{l_1 \cdots l_k}}{\sum_{i=1}^n \mu_i^{l_1 \cdots l_k}}. \tag{18}$$

In this model, in addition to the conventional parameters c, b and w, the importance degree h is introduced. Let h_L be the importance degree of each module L.

$$y^* = \sum_{L \in U_k^m} h_L \cdot y_L^0 \tag{19}$$

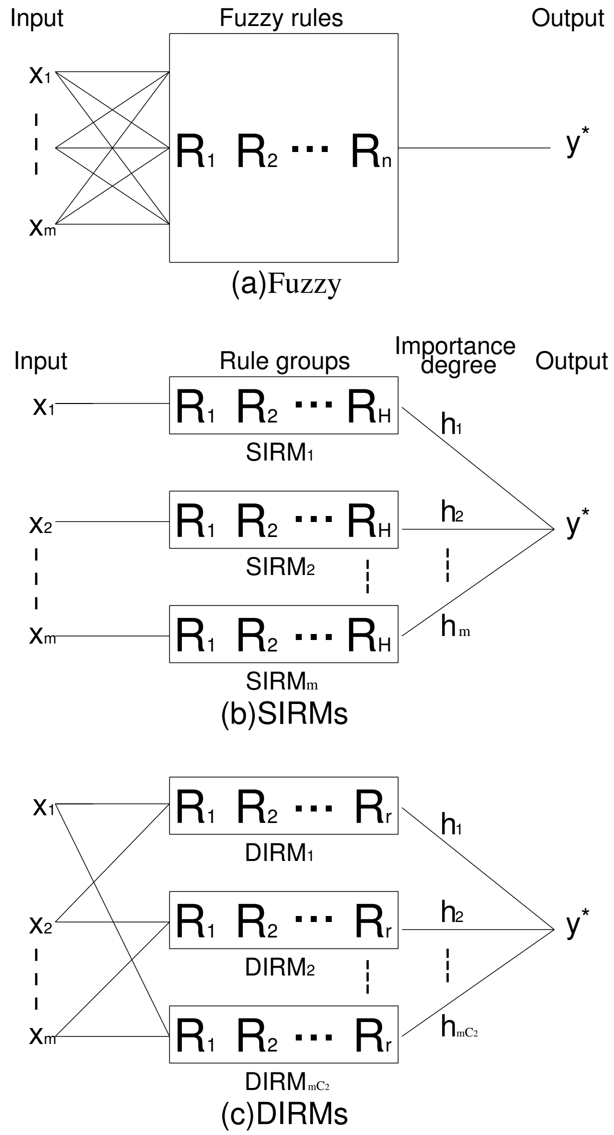


Fig. 2. The relation between the conventional fuzzy , SIRMs and DIRMs models

From the Eqs.(2) to (6), $\frac{\partial E}{\partial \alpha}$'s are calculated as follows:

$$\frac{\partial E}{\partial h_L} = (y^* - y^r) y_L^0, \tag{20}$$

$$\frac{\partial E}{\partial w_i^L} = h_L \cdot \frac{\mu_i^L}{\sum_{i=1}^n \mu_i^L}(y^* - y^r), \tag{21}$$

$$\frac{\partial E}{\partial c_i^L} = h_L \cdot (y^* - y^r) \frac{w_i^L - y_L^0}{\sum_{i=1}^n \mu_i^L} \frac{2\mathrm{sgn}(x_i - c_i^L)}{b_i^L \cdot M_i^L(x_i)} \tag{22}$$

$$\frac{\partial E}{\partial b_i^L} = h_L \cdot (y^* - y^r) \frac{w_i^L - y_L^0}{\sum_{i=1}^n \mu_i^L} \frac{1 - M_i^L(x_i)}{b_i^L \cdot M_i^L(x_i)} \tag{23}$$

$$\frac{\partial E}{\partial c_i^L} = h_L \cdot (y^* - y^r) \frac{w_i^L - y_L^0}{\sum_{i=1}^n \mu_i^L} \frac{x_i - c_i^L}{(b_i^L)^2} \tag{24}$$

$$\frac{\partial E}{\partial b_i^L} = h_L \cdot (y^* - y^r) \frac{w_i^L - y_L^0}{\sum_{i=1}^n \mu_i^L} \frac{(x_i - c_i^L)^2}{(b_i^L)^3} \tag{25}$$

, where Eqs.(20), (21), (22) and (23), and Eqs.(20), (21), (24) and (25) are the results for the triangular and the Gaussian membership functions, respectively.

The cases of $k = 1$ and $k = 2$ are called SIRMs and DIRMs models, respectively. Fig.2 shows the relation between the simplified fuzzy inference , SIRMs and DIRMs models. Examples 1 and 2 are modules for SIRMs and DIRMs models for m=3, respectively. It is known that the SIRMs model does not always achieve good performance in non-linear systems [12], [13]. On the other hand, when the number of input variables is large, Algorithm A requires a large time complexity and tends to easily get stuck into a shallow local minimum. The DIRMs model can achieve good performance in non-linear systems compared to the SIRMs model and is simpler than the conventional fuzzy model.

A learning algorithm for SNIRMs(including SIRMs and DIRMs) model is given as follows:

Learning Algorithm B
Step 1: The initial parameters, c_i^L, b_i^L, w_i^L, Θ_1, T_{max}, K_c, K_b and K_w are set.
Step 2: Let $t=1$.
Step 3: Let $p=1$.
Step 4: An input and output data $(x_1^p,\cdots,x_m^p,y_p^r)$ is given.
Step 5: Membership value of each rule is calculated by Eq.(17).
Step 6: Inference output y_p is calculated by Eq.(19).
Step 7: Importance degree h_L is updated by Eq.(20).
Step 8: Real number w_i^L is updated by Eq.(21).
Step 9: Parameters c_i^L and b_i^L are updated by Eqs.(22) and (23) or Eqs.(24) and (25).
Step 10: If $p=P$ then go to the next step. If $p<P$ then $p\leftarrow p+1$ and go to Step 4.
Step 11: Inference error $E(t)$ is calculated by Eq.(14). If $E(t)<\Theta_1$ then learning is terminated.
Step 12: If $t\neq T_{max}$, $t\leftarrow t+1$ and go to Step 3. Otherwise learning is terminated.

Note that the numbers of rules for the conventional model by Algorithm A, DIRMs and SIRMs models are $O(H^m)$, $O(m^2H^2)$ and $O(mH)$, respectively, where H is the number of partitions for fuzzy inference rules. In order to reduce the number of rule for DIRMs model, we propose the variable increase method for DIRMs model with $O(mH^2)$ rules. The model is composed of SIRMs model and $O(mH^2)$ rules of DIRMs model. The algorithm is as follows:

Learning Algorithm C (The variable increase method for DIRMs model)
Step 1: Algorithm B for k=1 is performed. SIRMs model is constructed.
Step 2: Select a variable x_0 with highest importance degree in step1 and add all new modules composed of two input variables including the variable x_0 to the system obtained in step1.
Step 3: In order to adjust the parameters of the system, algorithm B is performed.

IV. Numerical Simulations

In the section IV.A, we give a proposition to show the theoretical difference of capability between SIRMs and DIRMs models. In the section IV.B, numerical simulations to general features for SIRMs and DIRMs models using two-category problems are presented. Further, numerical simulations for obstacle avoidance as one of control problems are performed in the section IV.C.

A. The EX-OR problem with two variables

The EX-OR problem with two variables is defined as follows:

$$y = x_1 \oplus x_2 \tag{26}$$

, where x_1, x_2 and $y\in\{0,1\}$ and $\oplus$ means the Exclusive OR operation [3]. Then the following result holds.
[Proposition] The EX-OR problem with two variables cannot be implemented by any SIRMs model.
(proof) Assume that there exists SIRMs model implementing the EX-OR problem with two variables. Then, the output y^* of SIRMs model is defined as follows:

$$y^* = \sum_{j=1}^{2} h_j \frac{\sum_{i=1}^n w_{ij}M_{ij}(x_j)}{\sum_{i=1}^n M_{ij}(x_j)} \tag{27}$$

From the relation between input and output of EX-OR operation, the following relation holds.

$$h_1\frac{\sum_{i=1}^n w_{i1}M_{i1}(0)}{\sum_{i=1}^n M_{i1}(0)} + h_2\frac{\sum_{i=1}^n w_{i2}M_{i2}(0)}{\sum_{i=1}^n M_{i2}(0)} = 0 \tag{28}$$
$$h_1\frac{\sum_{i=1}^n w_{i1}M_{i1}(0)}{\sum_{i=1}^n M_{i1}(0)} + h_2\frac{\sum_{i=1}^n w_{i2}M_{i2}(1)}{\sum_{i=1}^n M_{i2}(1)} = 1 \tag{29}$$
$$h_1\frac{\sum_{i=1}^n w_{i1}M_{i1}(1)}{\sum_{i=1}^n M_{i1}(1)} + h_2\frac{\sum_{i=1}^n w_{i2}M_{i2}(0)}{\sum_{i=1}^n M_{i2}(0)} = 1 \tag{30}$$
$$h_1\frac{\sum_{i=1}^n w_{i1}M_{i1}(1)}{\sum_{i=1}^n M_{i1}(1)} + h_2\frac{\sum_{i=1}^n w_{i2}M_{i2}(1)}{\sum_{i=1}^n M_{i2}(1)} = 0 \tag{31}$$

Let us define f_1, f_2, f_3 and f_4 as follows:

$$f_1 = \frac{\sum_{i=1}^n w_{i1}M_{i1}(0)}{\sum_{i=1}^n M_{i1}(0)},\ f_2 = \frac{\sum_{i=1}^n w_{i2}M_{i2}(0)}{\sum_{i=1}^n M_{i2}(0)}$$
$$f_3 = \frac{\sum_{i=1}^n w_{i1}M_{i1}(1)}{\sum_{i=1}^n M_{i1}(1)},\ f_4 = \frac{\sum_{i=1}^n w_{i2}M_{i2}(1)}{\sum_{i=1}^n M_{i2}(1)}$$

From Eqs.(28) and (31), the following holds:

$$h_1(f_1+f_3)+h_2(f_2+f_4)=0 \tag{32}$$

From Eqs.(29) and (30), the following holds:

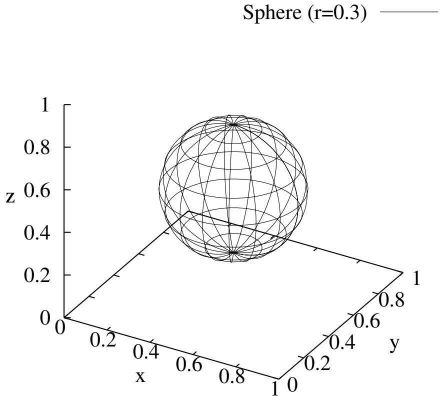
$$h_1(f_1+f_3)+h_2(f_2+f_4)=2 \tag{33}$$

This is contradiction. Therefore, there does not exist such SIRMs model. □

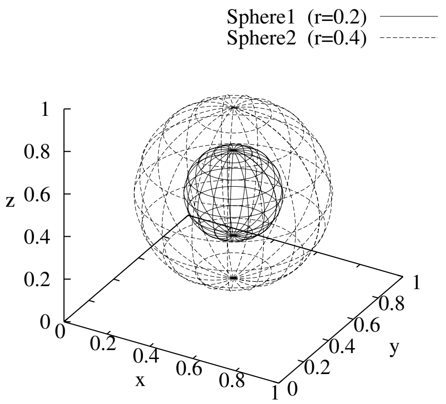
Remark that the proposition is the theoretical result. In [13], we have already conjectured that the same result holds in numerical simulation. On the other hand, there exists DIRMs model implementing the EX-OR problem with two variables.

B. Two-category Classification Problems

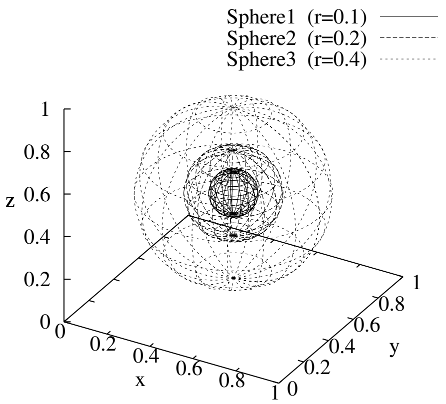
In the next, we perform two-category classification problems as in Fig. 3 to investigate the basic feature of SIRMs and DIRMs models. In the classification problems, points on $[0,1]\times[0,1]\times[0,1]$ are classified into two classes: class 0 and class 1. The class boundaries are given as spheres centered at $(0.5,0.5,0.5)$. For Sphere, the inside of sphere is associated with class 1 and the outside with class 0. For Double-Sphere, the area between Spheres 1 and 2 is associated with class 1 and the other area with class 0. For triple-Sphere, the inside of Sphere1 and the area between Sphere2 and Sphere3 is associated with class 1 and the other area with class 0.



(a) Sphere



(b) Double-Sphere



(c) Triple-Sphere

Fig. 3. Two-category Classification Problems

The desired output y_p^r is set as follows: if $\boldsymbol{x_p}$ belongs to class 0, then $y_p^r = 0.0$. Otherwise $y_p^r = 1.0$. The simulation condition is shown in Table I and the numbers of partitions is 3. Gaussian function is used as the membership function. The results on the rate of misclassification are shown in Table II. In TableII, A, B, and C mean Learning Algorithms A, B, and C, respectively, and the numbers in parenthesis mean the numbers of parameters. Further, the upper and lower values in each box mean the error rates for learning and test, respectively.

TABLE I
INITIAL CONDITION FOR SIMULATION OF TWO-CATEGORY CLASSIFICATION PROBLEMS.

	A	B ($k = 1$)	B ($k = 2$)	C
T_{max}	10000	100	3000	3000
K_w	0.05	0.01	0.01	0.01
K_h	-	0.05	0.05	0.05
K_c	0.00001	0.001	0.0001	0.0001
K_b	0.00001	0.001	0.0001	0.0001
Initial c_{ij}	equal intervals			
Initial b_{ij}	$\frac{1}{2(H-1)}\times$(the domain of input)			
Initial w_{ij}	random on $[0, 1]$			
Initial h_i	random on $[0, 1]$			

TABLE II
SIMULATION RESULT FOR TWO-CATEGORY CLASSIFICATION PROBLEM.

H=3	Sphere	Double-Sphere	Triple-Sphere
A	1.699	1.562	2.753
(189)	2.210	4.320	5.412
B(k=1)	11.230	16.835	16.328
(30)	11.237	16.789	16.371
B(k=2)	1.484	2.128	3.476
(138)	2.179	5.095	6.307
C	1.660	4.550	5.019
(122)	3.317	8.582	8.789

C. Obstacle avoidance

1) Obstacle avoidance: From (operation) data to avoid obstacle given by an examine, fuzzy inference rules for each model are constructed. As shown in Fig.4, the distance d and the angle θ between mobile object and obstacle are selected as 2 input variables. The mobile object moves with the vector $\mathbf{A}$=(A_x, A_y) at each step, where the element A_x of $\mathbf{A}$ is constant and the element A_y of $\mathbf{A}$ is only determined as an output from fuzzy inference. Learning data to avoid obstacle given by an examine are shown as 100 points in Fig.5. From the data, fuzzy inference rules to perform the trace of Fig.5 are constructed for each model, where the simulation condition is shown in Table III. The number of partitions for each model is 5. Let us perform the test simulation after learning. Fig. 6 shows the results for the moves of mobile object from the starting places at $(0.1, 0), (0.2, 0), \cdots, (0.8, 0), (0.9, 0)$. In both SIRMs and DIRMS models, obstacle avoidance is successful as shown in Fig.6. Further, test simulations with the place of obstacle different from the place in learning are performed with the same fuzzy inference rule for each model. As shown in Fig.7, the results are successful for both models.

Furthermore, let us perform simulations to avoid the obstacle moving with the vector (0.012, 0.02) at each step, from the initial place (0.9, 0) and arrives at the place (0.3, 1.0) at step $T = 50$ as shown in Fig.8. As shown Fig.9, the test simulation is successful for both models.

TABLE III
INITIAL CONDITION FOR SIMULATION OF OBSTACLE AVOIDANCE.

	A	B ($k = 1$)	B ($k = 2$)	C
T_{max}	10000	100	1000	1000
K_w	0.01	0.01	0.01	0.01
K_h	-	0.05	0.05	0.05
K_c	0.001	0.001	0.001	0.001
K_b	0.001	0.001	0.001	0.001

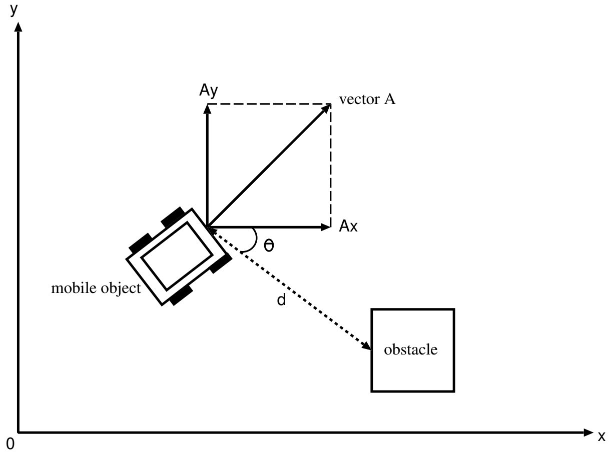


Fig. 4. Simulation on obstacle avoidance

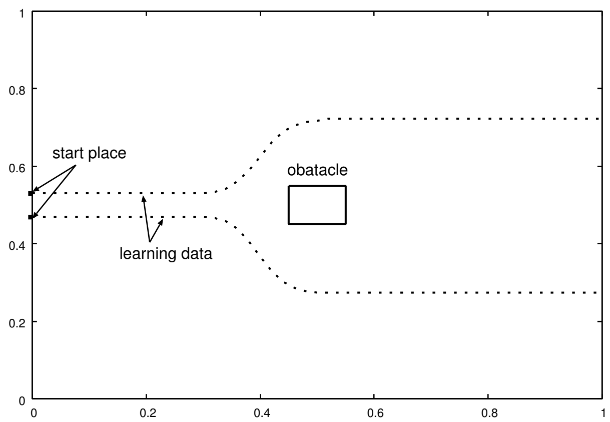
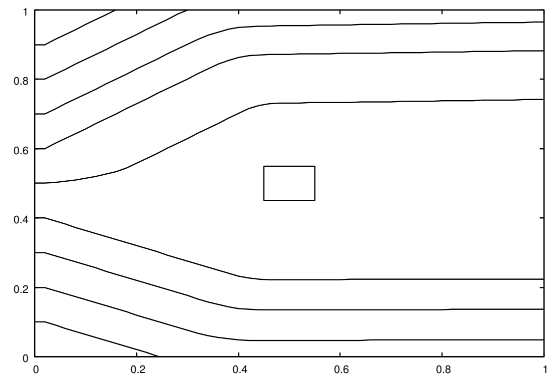
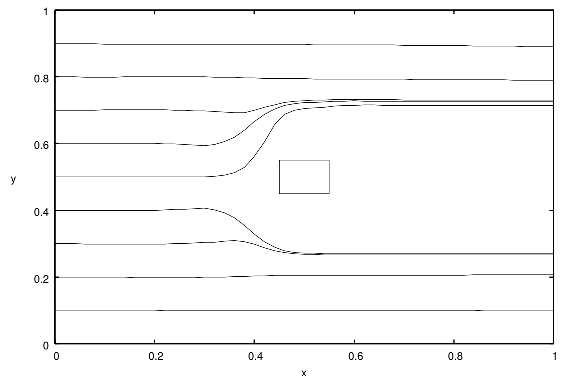


Fig. 5. Learning data denoted by dots to avoid obstacle.

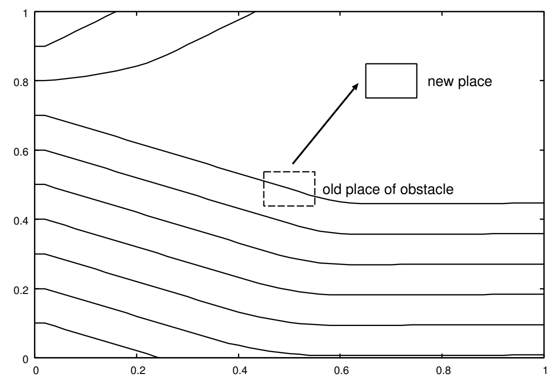


(a) SIRMs model

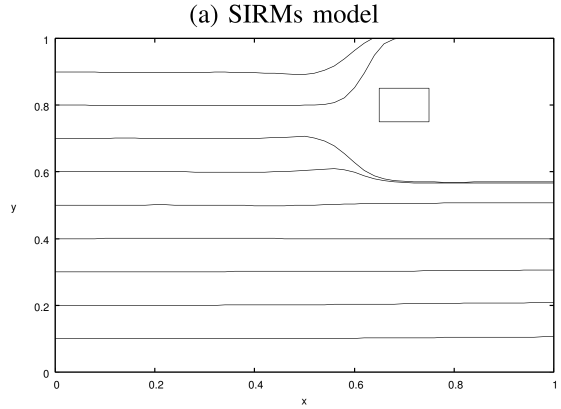


(b) DIRMs model

Fig. 6. Simulation result for obstacle avoidance starting from various places learning



(a) SIRMs model



(b) DIRMs model

Fig. 7. Simulation for obstacle avoidance placed at different place after learning.

2) Obstacle avoidance and arriving at the designated place: As shown in Fig.10, the distance d_1 and the angle θ_1 between mobile object and obstacle and the distance d_2 and the angle θ_2 between mobile object and the designated place are selected as input variables. The problem is to construct fuzzy inference system that mobile object avoids obstacle and arrives at the designated place. From (operation) data, Fuzzy inference rules for each model are constructed from learning of data (200 points shown in Fig. 11). The number of partitions for each model is 5. As the same method as the above, the mobile object moves with the vector $\mathbf{A}$ at each step, where A_y of $\mathbf{A}$ is output variable. The simulation condition is shown in Table III.

Four tests after learning are performed as follows:
(1)Test 1 is simulation for obstacle avoidance and arriving at the designated place when the mobile object stars from various places (See Fig.12). Fig.12 shows the results of moves of mobile object for starting places at $(0.1, 0), (0.2, 0), \cdots, (0.8, 0), (0.9, 0)$ after learning. As shown in Fig.12, the test simulations are unsuccessful and successful for SIRMs and DIRMs models, respectively.
(2)Test 2 is simulation for the case where the mobile object arrives at different designated place. Simulations arriving at the place (1, 0.35) different from the designated place (1, 0.5) in learning are performed for DIRMs model (See Fig.13). The results are successful as shown in Fig.13.

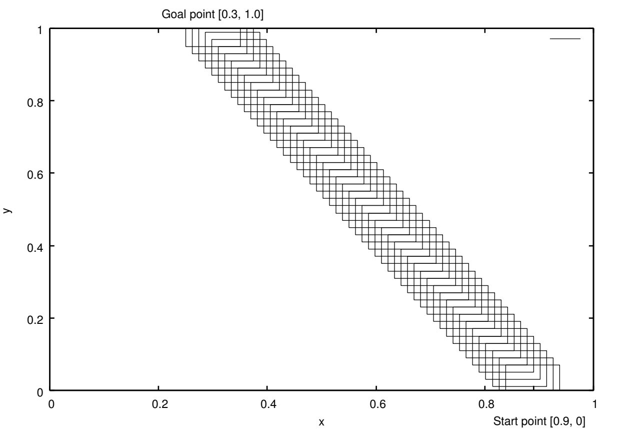
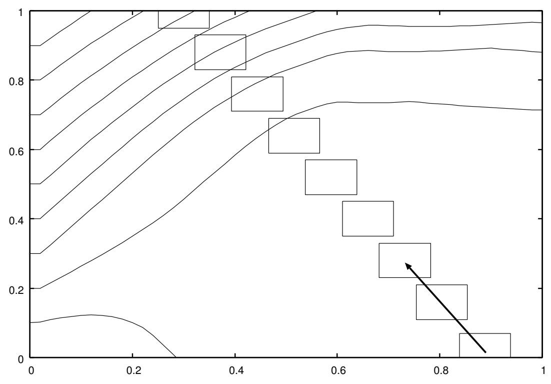


Fig. 8. The obstacle moves with the vector (0.012, 0.02) at each step from starting point (0.9, 0) to arriving point (0.3, 1.0).


(a) SIRMs model

(b) DIRMs model

Fig. 9. Simulation for moving-obstacle avoidance.

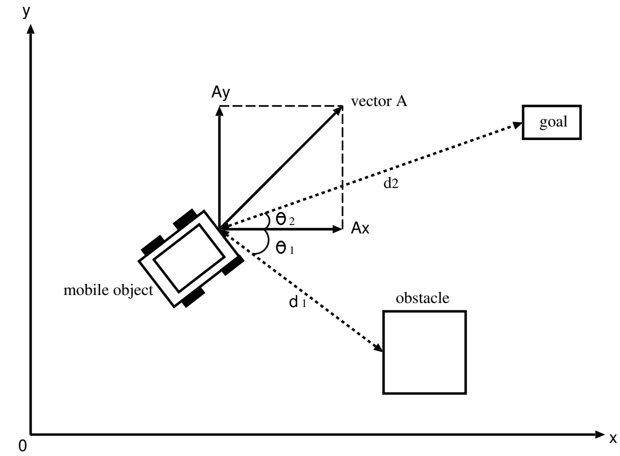


Fig. 10. Simulation on obstacle avoidance and arriving at the goal.

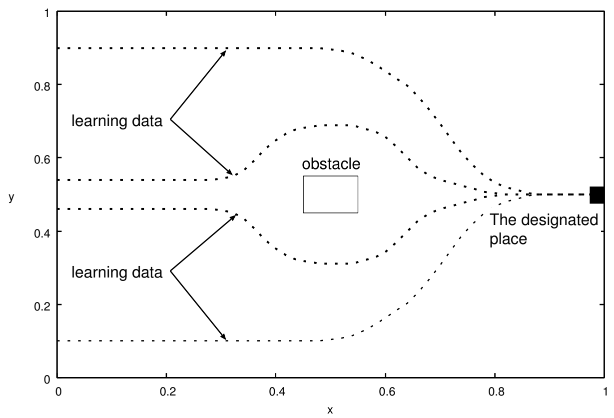


Fig. 11. Learning data to avoid obstacle and arrive at the designated place (1, 0.35).

(3)Test 3 is simulation for the case where the mobile object avoids obstacle placed at different place and arrives at the different designated place. Simulations with obstacle placed at the place (0.4, 0.4) and arriving at the designated place (1, 0.6) are performed for DIRMs model (See Fig.14). The results are successful as shown in Fig.14.

(4)Test 4 is simulation for the case where obstacle moves with the fixed speed. Simulations with obstacle moving as Fig. 8 and arriving at the place (1, 0.5) are performed. The results for the steps T=27, 28 and 50 are shown in Fig.15. It means that simulations for obstacle avoidance are successful.

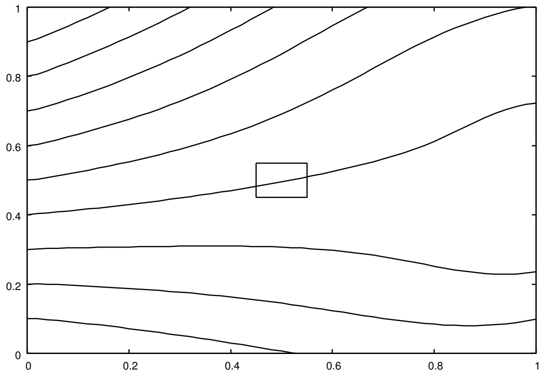
Lastly, we performed the same simulations for the variable increase method for DIRMs model. As a result, all test simulations are also successful in the variable increase method for DIRMs model. Therefore, the number 6 of modules for DIRMs model can be reduced to the model composed of 3 modules.

V. Conclusion

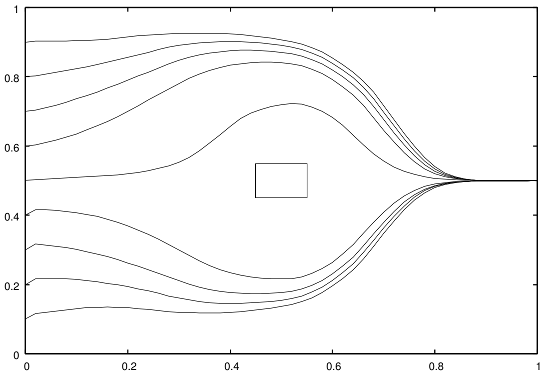
In this paper, a theoretical result and some numerical simulations including obstacle avoidance are presented in order to compare DIRMs model with SIRMs model. It is shown that there exists the difference of capability in the theoretical means with EX-OR problem with two variables and in numerical simulation with two-category problems. Further, two types of obstacle avoidance problems are performed: The first problem is simply to avoid obstacle and the second one is to avoid obstacle and to arrive at the designated place. In the first problem, both SIRMs and DIRMs models are successful in all test simulations. In the second problem, there exists the difference of capability between DIRMs and SIRMs models in simulations. Further, DIRMs model with the variable increase method is also successful in all simulations. In the future works, the application to the other control problems and a proposal of the new generalized SIRMs model are considered.

References

[1] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice Hall, Englewood Cliffs, NJ, 1992.



(a) SIRMs model



(b) DIRMs model

Fig. 12. Simulation for obstacle avoidance and arriving at the different designated place after learning.

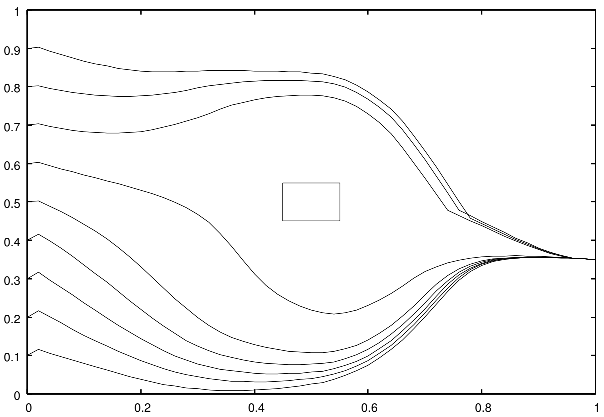
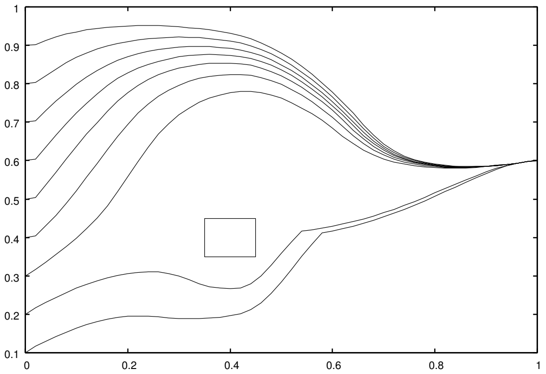
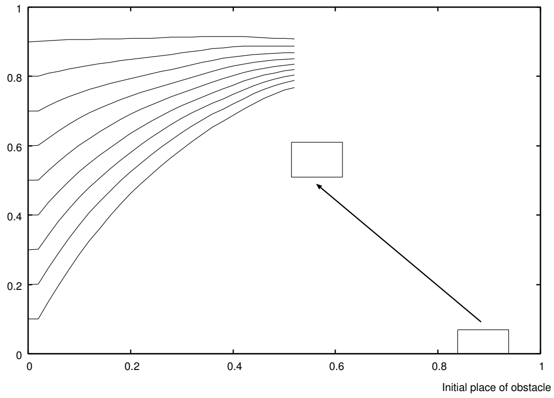


Fig. 13. Simulation for obstacle avoidance with the different designated place (1, 0.35) from learning.

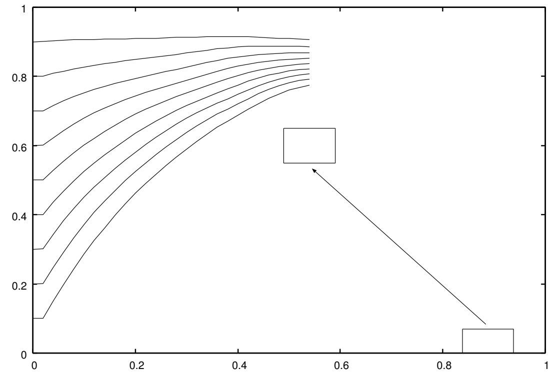


DIRMs.

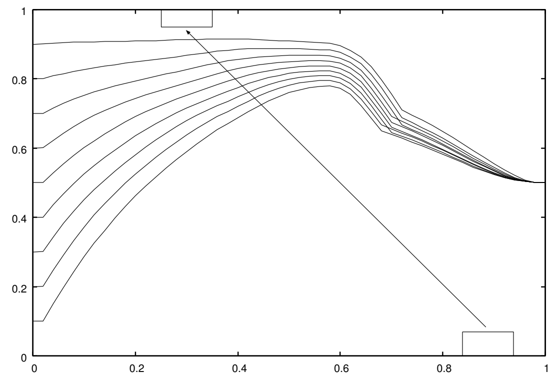
Fig. 14. Simulation for obstacle avoidance with the different designated place (1, 0.35) from learning.



(a) DIRMs model(T=27).



(b) DIRMs model(T=28).



(c) DIRMs model(T=50).

Fig. 15. Simulation for obstacle avoidance with the different designated place (1, 0.35) from learning.

[2] C. Lin and C. Lee, *Neural Fuzzy Systems*, Prentice Hall, PTR, 1996.

[3] M.M. Gupta, L. Jin and N. Homma, *Static and Dynamic Neural Networks*, IEEE Press, 2003.

[4] J. Casillas, O. Cordon, F. Herrera and L. Magdalena, *Accuracy Improvements in Linguistic Fuzzy Modeling*, Studies in Fuzziness and Soft Computing, Vol.129, Springer, 2003.

[5] B. Liu, *Theory and Practice of Uncertain Programming*, Studies in Fuzziness and Soft Computing, Vol.239, Springer, 2009.

[6] S. Araki, H. Nomura, I. Hayashi and N. Wakami, "A Fuzzy Modeling with Iterative Generation Mechanism of Fuzzy Inference Rules," *Journal of Japan Society for Fuzzy Theory and Systems*, vol.4, no.4, pp.722–732, 1992.

[7] S. Fukumoto, H. Miyajima, K. Kishida and Y. Nagasawa, "A Destructive Learning Method of Fuzzy Inference Rules," *Proc. of IEEE on Fuzzy Systems*, pp.687–694, 1995.

[8] K. Kishida, H. Miya jima, M. Maeda and S. Murashima, "A Self-tuning Method of Fuzzy Modeling using Vector Quantization," *Proceedings of FUZZ-IEEE ' 97*, pp.397–402, 1997.

[9] S. Fukumoto and H. Miyajima, "Learning Algorithms with Regularization Criteria for Fuzzy Reasoning Model," *Journal of Innovative Computing, Information and Control*, vol.1, no.1, pp.249–263, 2006.

[10] J. Yi, N. Yubazaki and K. Hirota, "A Proposal of SIRMs Dynamically

Connected Fuzzy Inference Model for Plural Input Fuzzy Control," *Fuzzy Sets and Systems*, 125, pp.79–92, 2002.
[11] N. Yubazaki, J. Yi and K. Hirota, "SIRMS (Single Input Rule Modules) Connected Fuzzy Inference Model," *J. Advanced Computational Intelligence*, 1, 1, pp.23–30, 1997.
[12] N. Shigei, H. Miyajima and S. Nagamine, "A Proposal of Fuzzy Inference Model Composed of Small-Number-of-Input Rule Modules", *Proc. of Int. Symp. on Neural Networks: Advances in Neural Networks - Part II*, pp.118–126, 2009.
[13] S. Miike, H. Miyajima, N. Shigei and K. Noo, "Fuzzy Reasoning Model with Deletion of Rules Consisting of Small-Number-of-Input Rule Modules," *Journal of Japan Society for Fuzzy Theory and Intelligent Informatics*, pp.621–629, 2010 (in Japanese).
[14] H. Miyajima, N. Shigei, H. Miyajima, "An Application of Fuzzy Inference System Composed of Double-Input Rule Modules to Control Problems," *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2014*, *IMECS 2014*, 12-14 March, 2014, Hong Kong, pp.23–28.
[15] O. Cordon, "A Historical Review of Evolutionary Learning Methods for Mamdani-type Fuzzy Rule-based Systems: Designing Interpretable Genetic Fuzzy Systems," *Journal of Approximate Reasoning*, 52, pp.894–913, 2011.