ORIGINAL RESEARCH

System identifications by SIRMs models with linear transformation of input variables

Hirofumi Miyajima, Noritaka Shigei, Hiromi Miyajima*

Graduate School of Science and Engineering, Kagoshima University, Japan

Received: November 10, 2015 **Accepted:** March 3, 2016 **Online Published:** April 6, 2016
DOI: 10.5430/air.v5n2p55 **URL:** http://dx.doi.org/10.5430/air.v5n2p55

ABSTRACT

This paper shows the effectiveness of the model proposed in the previous paper for system identifications. In the first simulation, which is for EX-OR, the fundamental idea of the proposed model is explained. In the second simulation, which is for classification problems for dataset of Iris, Wine, Sonar and BCW known as benchmark problems, the capability of the model is evaluated for involving a large number of input variables. In the third simulation, as one of control problems, numerical simulation for obstacle avoidance problem is performed. In these simulations, it is shown that the proposed model outperforms conventional models in terms of system identifications.

Key Words: SIRMs models with LT, Classification problem, Obstacle avoidance problem

1. INTRODUCTION

There have been many studies on self-tuning fuzzy systems.[1–5] Their aim is to construct self-tuning fuzzy systems from learning data based on the steepest descent method (SDM). The obvious drawbacks of the method are its large computational complexity and getting stuck in a shallow local minimum. Therefore, ineffective systems with an inference error and the number of rules are constructed. In order to construct effective systems, some novel methods have been developed which 1) create fuzzy rules one by one starting from a small number of rules,[6] 2) delete fuzzy rules one by one starting from a sufficiently large number of rules,[7] 3) use a genetic algorithm to determine the structure of the fuzzy model,[5] 4) use a self-organization or a vector quantization technique to determine the initial assignment of fuzzy rules,[8,9] 5) use generalized objective functions.[10] However, there are little effective fuzzy inference systems. Therefore, the conventional learning methods with multi-objective fuzzy modeling and fuzzy modeling with constrained parameters of the ranges have become popular.[5] On the other hand, SIRMs (Single-Input Rule Modules) model aims to obtain a better solution by using fuzzy inference system composed of single-input rule modules.[11] Although it is easy to apply SIRMs model to the problems with many input variables, it is always difficult to obtain a good performance for nonlinear problems. Accordingly, SNIRMs (Small Number of Input Rule Modules) models, which is a generalized SIRMs, have been proposed.[12–14] However, the capability of SNIRMs models such as DIRMs (Double-Input Rule Modles) one is still low compared with conventional models. Therefore, we proposed SIRMs model with linear transformation (LT) of input variables and showed its performance in previous papers.[15,16]

*Correspondence: Hiromi Miyajima; Email: miya@eee.kagoshima-u.ac.jp; Address: Graduate School of Science and Engineering, Kagoshima University, Japan.

In this paper, we investigate the performance of the proposed model for various problems of system identifications. In the simulation of classification problems, we compare the proposed model and the conventional model in terms of accuracy and the number of parameters. Further, it is shown that the proposed model can be applied to an obstacle avoidance problem, which is an application of control problems. In the simulation results, we will show that the proposed model outperforms conventional models in terms of accuracy and the number of parameters.

2. PRELIMINARY

2.1 The conventional model

The conventional model using the SDM was previously presented.[1,2,6] Let $Z_k = \{1, \cdots, k\}$ for the positive integer k. Let $\boldsymbol{R}$ denote the set of all real numbers. Let $\boldsymbol{x} = (x_1, \cdots, x_m)$ and y denote input and output data, respectively, where $x_i, y \in \boldsymbol{R}$ for $i \in Z_m$. Then, for $j \in Z_n$, the j-th rule of the conventional model is expressed as

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

where M_{ij} and w_j denote a membership function and the weight of j-th rule, respectively.

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

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

For Gaussian membership function, M_{ij} is defined as follow:

$$M_{ij}(x_i) = \exp\left(-\frac{1}{2}\left(\frac{x_i - c_{ij}}{b_{ij}}\right)^2\right) \quad (3)$$

where c_{ij} and b_{ij} denote the center and width parameters, respectively. The output y^* of fuzzy inference is obtained as follows:

$$y^* = \frac{\sum_{j=1}^{n} \mu_j \cdot w_j}{\sum_{j=1}^{n} \mu_j} \quad (4)$$

The objective function E, which is the inference error between the desirable output y^r and the output y^*, is defined as follows:

$$E = \frac{1}{2}\left(y^* - y^r\right)^2 \quad (5)$$

In order to minimize the objective function E, the procedure based on SDM updates each parameter $\beta \in \{c_{ij}, b_{ij}, w_j\}$ as follows:

$$\beta(t+1) = \beta(t) - K_\beta \frac{\partial E}{\partial \beta} \quad (6)$$

where t is the learning time and K_β is the learning rate of β, which is a constant value.

Then, the following relation for Eq.(3) hold:

$$\frac{\partial E}{\partial w_j} = \frac{\mu_j}{\sum_{j=1}^{n} \mu_j} \cdot (y^* - y^r) \quad (7)$$

$$\frac{\partial E}{\partial c_{ij}} = \frac{\mu_j}{\sum_{j=1}^{n} \mu_j} \cdot (y^* - y^r) \cdot (w_j - y^*) \cdot \frac{x_j - c_{ij}}{b_{ij}^2} \quad (8)$$

$$\frac{\partial E}{\partial b_{ij}} = \frac{\mu_j}{\sum_{j=1}^{n} \mu_j} \cdot (y^* - y^r) \cdot (w_j - y^*) \cdot \frac{(x_j - c_{ij})^2}{b_{ij}^3} \quad (9)$$

In the following, Gaussian function is used as a membership function.

2.2 The leaning algorithm

A typical learning algorithm is introduced as the conventional one.[2] Let $D = \{(x_1^p, \cdots, x_m^p, y_p^r) | p \in Z_P\}$ be a given set of learning data. Figure 1 shows the learning algorithm, which minimizes the following objective function

$$E = \frac{1}{P}\sum_{p=1}^{P}(y_p^* - y_p^r)^2 \quad (10)$$

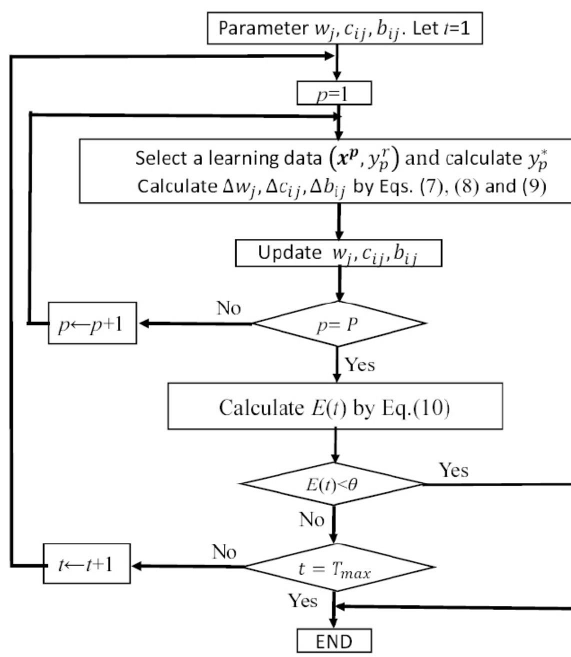

Figure 1. The flowchart of the conventional learning algorithm

In the algorithm, θ and T_{max} are the threshold for inference error and the maximum number of learning time, respectively.

3. SIRMs AND DIRMs MODELS

The SIRMs and DIRMs models are introduced.[12–14]

Each rule of SIRMs model is expressed as

$$\begin{aligned}
\text{SIRM}-1 &: \{R_i^1 : \text{if } x_1 \text{ is } M_i^1 \text{ then } y_1 \text{ is } w_i^1\}_{i=1}^n \\
&\vdots \\
\text{SIRM}-l &: \{R_i^l : \text{if } x_l \text{ is } M_i^l \text{ then } y_l \text{ is } w_i^l\}_{i=1}^n \\
&\vdots \\
\text{SIRM}-m &: \{R_i^m : \text{if } x_m \text{ is } M_i^m \text{ then } y_m \text{ is } w_i^m\}_{i=1}^n
\end{aligned}$$

A membership value for input $\boldsymbol{x}$ and output for the l-th rule are expressed as follows:

$$\mu_i^l = M_i^l(x_l) \tag{11}$$

$$y_l^0 = \frac{\sum_{i=1}^n \mu_i^l w_i^l}{\sum_{i=1}^n \mu_i^l} \tag{12}$$

Then the output of fuzzy inference is obtained as follow:

$$y^* = \sum_{l=1}^m h_l y_l^0 \tag{13}$$

Each parameter for the objective function E is updated as follow:

$$\frac{\partial E}{\partial h_l} = (y^* - y^r) y_l^0 \tag{14}$$

$$\frac{\partial E}{\partial w_i^l} = h_l \frac{\mu_i^l}{\sum_{i=1}^l \mu_i^l}(y^* - y^r) \tag{15}$$

$$\frac{\partial E}{\partial c_i^l} = h_l(y^* - y^r)\frac{(w_i^l - y_l^0)}{\sum_{i=1}^n \mu_i^l}\frac{x_i - c_i^l}{(b_i^l)^2} \tag{16}$$

$$\frac{\partial E}{\partial b_i^l} = h_l(y^* - y^r)\frac{(w_i^l - y_l^0)}{\sum_{i=1}^n \mu_i^l}\frac{(x_i - c_i^l)^2}{(b_i^l)^3} \tag{17}$$

where h_l, w_i^l, c_i^l and b_i^l are parameters.

Likewise, the rule of DIRMs model is expressed as

DIRMs-12

$$\{R_i^{12} : \text{if } x_1 \text{ is } M_i^1 \text{ and } x_2 \text{ is } M_i^2 \text{ then } y_{12} \text{ is } w_i^{12}\}_{i=1}^n$$

$\vdots$

DIRMs-$l_1 l_2$

$$\{R_i^{l_1 l_2} : \text{if } x_{l_1} \text{ is } M_i^{l_1} \text{ and } x_{l_2} \text{ is } M_i^{l_2} \text{ then } y_{l_1 l_2} \text{ is } w_i^{l_1 l_2}\}_{i=1}^n$$

$\vdots$

DIRMs-$m-1, m$

$$\{R_i^{m-1,m} : \text{if } x_{m-1} \text{ is } M_i^{m-1} \text{ and } x_m \text{ is } M_i^m \text{ then } y_{m-1,m} \text{ is } w_i^{m-1,m}\}_{i=1}^n$$

where $l_1 < l_2$.

In this case, a membership value, output for the $l_1 l_2$-th rule and output for fuzzy inference system are shown as follows:

$$\mu_i^{l_1 l_2} = M_i^{l_1}(x_{l_1}) M_i^{l_2}(x_{l_2}) \tag{18}$$

$$y_{l_1 l_2}^0 = \frac{\sum_{i=1}^n \mu_i^{l_1 l_2} w_i^{l_1 l_2}}{\sum_{i=1}^n \mu_i^{l_1 l_2}} \tag{19}$$

$$y^* = \sum_{l_1 l_2 \in Z_m^2} h_{l_1 l_2} y_{l_1 l_2}^0 \tag{20}$$

where $Z_m^2 = Z_m \times Z_m$, and $l_1 < l_2$.

Each parameter for the objective function E is updated as follow:

$$\frac{\partial E}{\partial h_{l_1 l_2}} = (y^* - y^r) y_{l_1 l_2}^0 \tag{21}$$

$$\frac{\partial E}{\partial w_i^{l_1 l_2}} = h_{l_1 l_2} \frac{\mu_i^{l_1 l_2}}{\sum_{i=1}^n \mu_i^{l_1 l_2}}(y^* - y^r) \tag{22}$$

$$\frac{\partial E}{\partial c_i^{l_1 l_2}} = h_{l_1 l_2}(y^* - y^r) \cdot \gamma \cdot \frac{x_i - c_i^{l_1 l_2}}{(b_i^{l_1 l_2})^2} \tag{23}$$

$$\frac{\partial E}{\partial b_i^{l_1 l_2}} = h_{l_1 l_2}(y^* - y^r) \cdot \gamma \cdot \frac{(x_i - c_i^{l_1 l_2})^2}{(b_i^{l_1 l_2})^3} \tag{24}$$

where $\gamma = \frac{(w_i^{l_1 l_2} - y_{l_1 l_2}^0)}{\sum_{i=1}^n \mu_i^{l_1 l_2}}$, and $h_{l_1 l_2}$, $w_i^{l_1 l_2}$, $c_i^{l_1 l_2}$ and $b_i^{l_1 l_2}$ are parameters.

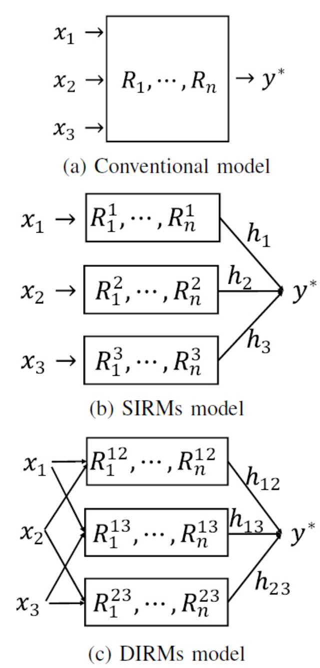

Figure 2. The block diagrams of three types of models for m = 3

Figure 2 shows the relation among conventional, SIRMs and DIRMs models for $m = 3$. Note that each order of the numbers of rules for three models is $O(h^m)$, $O(mh)$ and $O(m^2h^2)$, respectively, where h denotes the number of partitions in a membership function. Further, the numbers of parameters for them are $(2m+1)h^m$, $(3h+1)m$ and $(5h^2+1)_m C_2$, respectively.

Learning methods for SIRMs and DIRMs models are shown as the same methods as Figure 1.

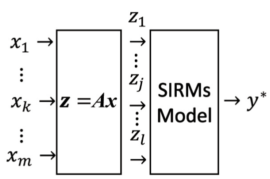

Figure 3. The proposed model: Input ***x*** is transformed into intermediate ***z*** by the matrix A and SIRMs model with variable ***z*** is formed

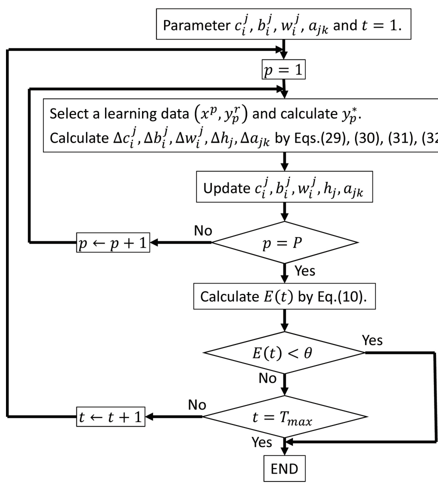

Figure 4. The flowchart of the proposed learning algorithm

4. PROPOSED FUZZY INFERENCE MODEL

The proposed model is shown in Figure 3.[15, 16]

The model consists of two stages. The first stage performs a linear transformation (LT) $\boldsymbol{A}$ from input $\boldsymbol{x}$ into intermediate variables $\boldsymbol{z} = (z_1, \cdots, z_l)^T$ as follows:

$$\begin{pmatrix} z_1 \\ \vdots \\ z_j \\ \vdots \\ z_l \end{pmatrix} = \begin{pmatrix} a_{10} & \cdots & a_{1k} & \cdots & a_{1m} \\ & & \vdots & & \\ a_{j0} & \cdots & a_{jk} & \cdots & a_{jm} \\ & & \vdots & & \\ a_{l0} & \cdots & a_{lk} & \cdots & a_{lm} \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_k \\ \vdots \\ x_m \end{pmatrix} \tag{25}$$

where $\boldsymbol{A} = (a_{jk})$ for $j \in Z_l$ and $k \in Z_m \cup \{0\}$ and $z_j = \sum_{k=0}^{m} a_{jk} x_k$ for $x_0 = 1$. The second stage is performed by SIRMs model with $z_1, \cdots, z_l$. The output y^* is calculated as follows:

$$\mu_i^j = M_i^j(z_j) \tag{26}$$

$$y_j^0 = \frac{\sum_{i=1}^{n} \mu_i^j w_i^j}{\sum_{i=1}^{n} \mu_i^j} \tag{27}$$

$$y^* = \sum_{j=1}^{l} h_j y_j^0 \tag{28}$$

where h_i for $i \in Z_l$ is the weight for the i-th module.

Then, the following relation hold:

$$\frac{\partial E}{\partial h_j} = (y^* - y^r) y_j^0 \tag{29}$$

$$\frac{\partial E}{\partial w_i^j} = h_j \frac{\mu_i^j}{\sum_{i=1}^{n} \mu_i^j} (y^* - y^r) \tag{30}$$

$$\frac{\partial E}{\partial c_i^j} = (y^* - y^r) h_j \frac{(w_i^j - y_j^0)\mu_i^j}{\sum_{i=1}^{n} \mu_i^j} \frac{z_j - c_i^j}{(b_i^j)^2} \tag{31}$$

$$\frac{\partial E}{\partial b_i^j} = (y^* - y^r) h_j \frac{(w_i^j - y_j^0)\mu_i^j}{\sum_{i=1}^{n} \mu_i^j} \frac{(z_j - c_i^j)^2}{(b_i^j)^3} \tag{32}$$

Further, $\frac{\partial E}{\partial a_{jk}}$ is computed as follows:

$$\frac{\partial E}{\partial a_{jk}} = (y^r - y^*) \frac{h_j}{\sum_{i=1}^{n} \mu_i^j} \sum_{i=1}^{n} \mu_i^j (w_i^j - y_j^0) \frac{z_j - c_i^j}{(b_i^j)^2} \tag{33}$$

The flowchart for learning algorithm of the proposed model is shown in Figure 4.

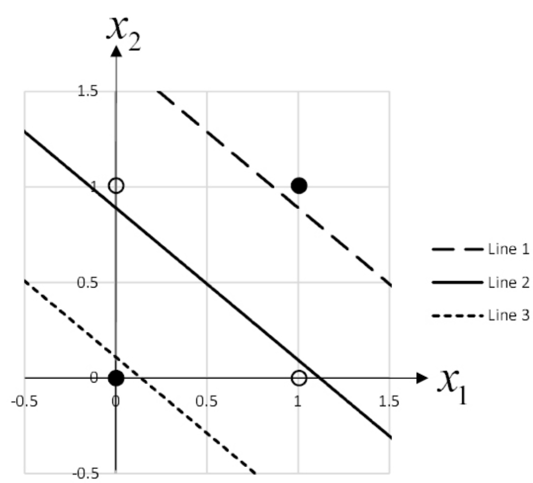

Figure 5. A schematic explanation of the proposed model: Three lines mean the centers of three membership functions. Two input of (0; 1) and (1; 0) are nearly on Line 2, where ○ and ● mean output 1 and 0, respectively

In this case, let θ and T_{max} be the threshold for inference error and the maximum number of learning time.

We will determine the number of intermediate variables l according to the threshold of inference error θ and the maximum number of learning time T_{max}. Note that the number of parameters for the proposed model is $(m+3h+2)l$.

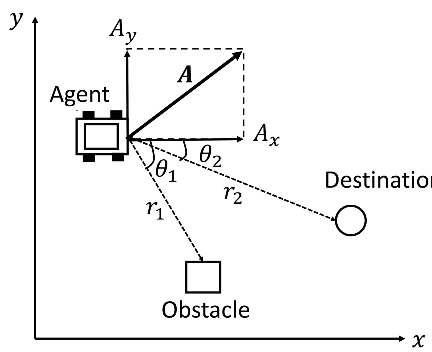

Figure 6. Simulation on obstacle avoidance, where r_1 and θ_1 are the distance and the angle between the agent and the obstacle, and r_2 and θ_2 are the distance and the angle between the agent and the destination.

5. Numerical simulations

In this section, the effectiveness of the proposed model is investigated for EX-OR, classification and obstacle avoidance problems.

5.1 The EX-OR problems

The EX-OR problem with m input variables is defined by the following equation.

$$z = x_1 \oplus x_2 \oplus \cdots \oplus x_m \tag{34}$$

where $x_1, \cdots, x_m, z \in \{0,1\}$ and $\oplus$ is the Exclusive OR operator.[2]

Let K_c, K_b, K_w, K_h, K_a and T_{max} in numerical simulations be the learning rate of c_i^j, b_i^j, w_i^j, h_j and a_{jk} and the maximum number of learning time, respectively. Further, a_{jk} means each element of the matrix $\boldsymbol{A}$ (See Eq.(25)). In this simulation, $K_c = 0.001$, $K_b = 0.001$, $K_w = 0.05$, $K_h = 0.05$, $K_a = 0.01$ and $h = 3$ are used. Further, $T_{max} = 50000$, 100, 2000 and 50000 are set for the conventional SIRMs, DIRMs and proposed methods, respectively. Furthermore, initial parameters c_i^j, b_i^j, w_i^j, h_j and a_{jk} are as follows: c_i^j is set to equal intervals $\frac{1}{2h-1} \times$ (the domain of input), and w_i^j, h_j and a_{jk} are randomly selected from domains $[0,1]$, $[0,1]$ and $[-1,1]$, respectively. The simulation result is shown in Table 1, where the symbol "-" means that it is impossible to simulate the problem in the condition or the result is over 0.25. The MSE value is the average value from ten trials. In Table 1, MSE = 0 means that the correct output is obtained for any input case, and MSE = 0.25 mean that the correct output is not obtained for just 25% input cases. Therefore, the SIRMs model and the DIRMs one for $m = 3$ cannot implement the EX-OR problem. On the other hand, the proposed model can implement the EX-OR problem for any m.

Table 1. The simulation result for EX-OR problem with m input variables

	m			
	2	3	4	10
The conventional	0	0	0	-
SIRMs	0.25	-	-	-
DIRMs	0	0.25	-	-
The proposed (*l* = *m*)	0	0	0	0

In the following, the idea of the proposed model is explained. Let us consider the following fuzzy inference system for $l = 1$ and $n = 3$ obtained by learning.

$$\begin{aligned} z &= 1.07 - 0.51x_1 - 0.64x_2 \\ h_1 &= 0.93 \end{aligned} \tag{35}$$

$$SIRMs-1 \begin{cases} R_1^1 \ : \ if\ z\ is\ M_1^1\ then\ y\ is\ -0.06, \\ R_2^1 \ : \ if\ z\ is\ M_2^1\ then\ y\ is\ 1.37, \\ R_3^1 \ : \ if\ z\ is\ M_3^1\ then\ y\ is\ -0.06, \end{cases}$$

$$M_1^1 = \exp\left(-\frac{(z+0.01)^2}{0.12}\right) \tag{36}$$

$$M_2^1 = \exp\left(-\frac{(z-0.50)^2}{0.11}\right) \tag{37}$$

$$M_3^1 = \exp\left(-\frac{(z-1.00)^2}{0.12}\right) \tag{38}$$

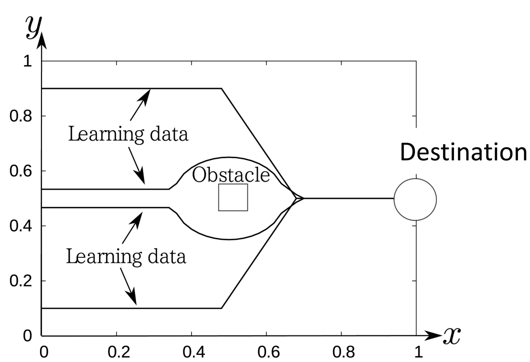

Figure 7. Learning data for simulation

In Figure 5, three lines of Lines 1, 2, and 3 mean the centers of three membership functions for Eqs.(36), (37) and (38), respectively. As two input of (0,1) and (1,0) are nearly on Line 2, the output is about 1. Further, output of (0,1) and (1,0) for Lines 1 and 3 nearly equal to 0. Therefore, output for (0,1) and (1,0) obtained from all rules nearly equals to 1. Likewise, we can consider about the cases of (0,0) and (1,1). On the other hand, we have already shown that EX-OR problem with two variables cannot be implemented by any SIRMs model.[16]

5.2 Classification problems

For classification problems, the benchmark datasets Iris, Wine, Sonar and BCW from UCI database are used.[18] The numbers of data, variables and classes are 150, 4 and 3 for Iris, 178, 13 and 3 for Wine, 208, 60 and 2 for Sonar and 683, 9 and 2 for BCW, respectively. In order to evaluate the model, 5-fold cross-validation is used. As the initial condition, $K_c = 0.001$, $K_b = 0.001$, $K_w = 0.05$, $K_h = 0.05$, $K_a = 0.01$, $h = 3$ and $T_{max} = 50000$ are used. Further, the initial value of c_i^j, b_i^j, w_i^j, h_j and a_{jk} are as follows: c_i^j is set to equal intervals $\frac{1}{2h-1} \times (\text{the domain of input})$, and b_i^j, w_i^j, h_j and a_{jk} are randomly selected from domains $[0, 1]$, $[0, 1]$ and $[-1, 1]$, respectively.

Table 2 shows the simulation result for the classification problems. In each box, two numbers from the top show misclassification rate for training and test data sets, respectively, and the bottom number shows the number of parameters. The simulation result is obtained as the average value from twenty trials.

Table 2. The simulation result for classification problems

	Iris	Wine	Sonar	BCW
The conventional	0.004 0.055 (729)	-	-	-
SIRMs	0.021 0.052 (40)	0.022 0.102 (130)	0.024 0.301 (600)	0.055 0.063 (90)
DIRMs	0.001 0.057 (276)	0.011 0.092 (3588)	-	0.001 0.065 (1656)
The proposed	0.029 0.031 (45)	0.001 0.037 (72)	0.001 0.205 (213)	0.016 0.036 (60)

In Table 2, the symbol "-" means that it is impossible to simulate the problem. Table 2 shows that the proposed method is superior to SIRMs and DIRMs models in terms of accuracy and the proposed one is superior to the conventional model in terms of the number of parameters.

5.3 Obstacle avoidance problem

As an application to control problems, let us show simulations of an obstacle avoidance problem used in the previous paper.[14] See it for the detailed explanation of the simulations.[14] Until now, we have already performed some simulations by the conventional, SIRMs and DIRMs models. As a result, the conventional and DIRMs models were successful in learning and test simulations.[14] We will show that the proposed model is successful with a small number of parameters.

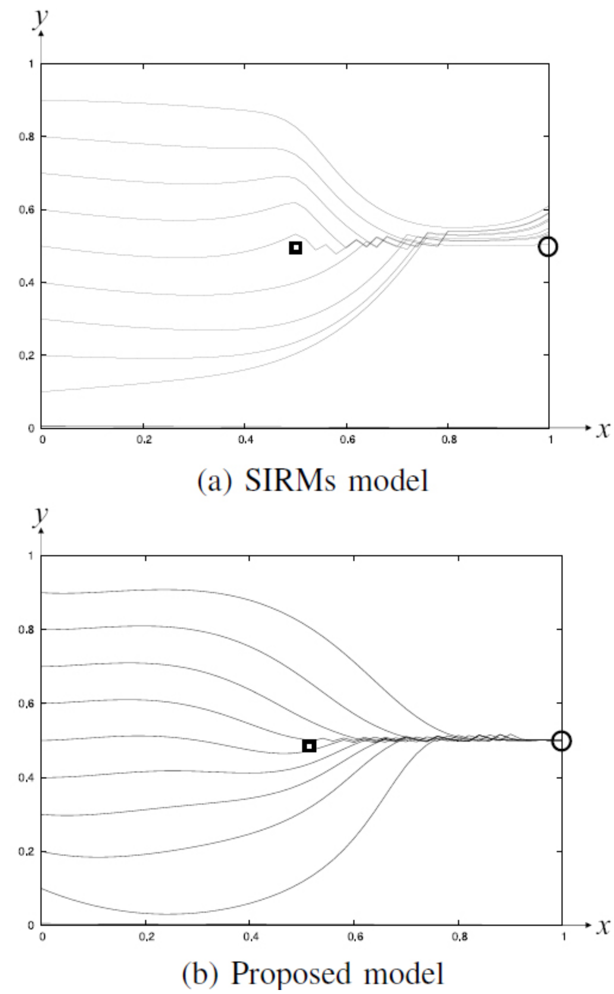

Figure 8. Simulation result for Test 1

As shown in Figure 6, four parameters are selected as input variables. The problem is to construct a fuzzy system by which the mobile agent avoids the obstacle and arrives at the destination. The mobile agent moves with the vector $\boldsymbol{A} = (A_x, A_y)$ at each step, where A_x is constant and A_y is only adjusted as an output from the fuzzy system. The learning data with 200 points are obtained from an examinee. Using the learning data, fuzzy inference rules are constructed for SIRMs and the proposed models. As simulation conditions for SIRMs and proposed models, $K_c = 0.001$, $K_b = 0.001$, $K_w = 0.05$, $K_h = 0.05$, $K_a = 0.05$, $h = 3$ and $T_{max} = 50000$ are selected. Initial values for c_i^j and b_i^j are set to equal intervals and $\frac{1}{2(h-1)}(\text{the domain of input})$, respectively. Initial values for w_i^j, h_j and a_{jk} are selected randomly from $[0, 1]$, $[0, 1]$ and $[-1, 1]$, respectively. For $h = 3$, the number of parameters of SIRMs is 40 and that of proposed model is 30, respectively.

In the simulations, only when the agent reaches the destination unless colliding with the obstacle, the trial is regarded as a successful one. Otherwise, the trial is regarded as a unsuccessful (failed) one. There are two types of evaluations: learning and test. For the learning evaluation, the positions of the agent's starting point, the obstacle and the destination are same as the ones of learning data. In the learning evaluation, SIRMs fails in the trials but the proposed model is successful. For the test evaluation, one or more of the three positions are different from the ones of learning data. The following four types of test evaluations are considered in the simulations.

(1) Test 1 uses some different agent's starting points (see Figure 8). Figure 8 shows the movements of the agent from starting points $(0.0, 0.1), (0.0, 0.2), \cdots, (0.0, 0.8), (0.0, 0.9)$ after learning. The test evaluation result is shown in Figure 8. Like the learning evaluation, SIRMs fails in the trials but the proposed model is successful.

(2) Test 2 assumes the scenario where the mobile agent avoids the obstacle placed at a different place and arrives at the different destination. The proposed model is applied for the obstacle at (0.4, 0.4) and the destination (1.0, 0.6). Every trial is successful as shown in Figure 9.

(3) Test 3 assumes the scenario where the obstacle moves with the fixed speed. As shown in Figure 10, every trial is successful.

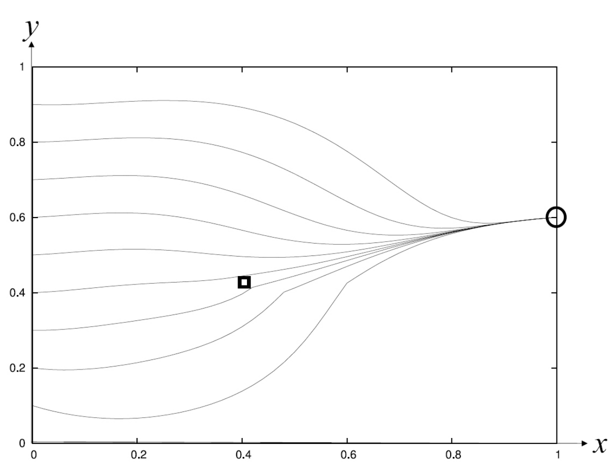

Figure 9. Simulation result for Test 2

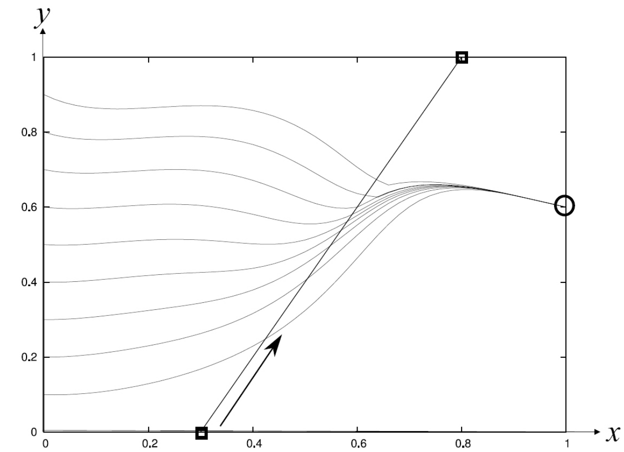

Figure 10. Simulation result for Test 3

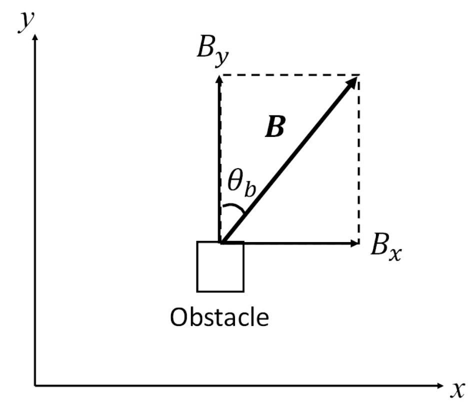

Figure 11. The movement of the mobile obstacle for Test 4

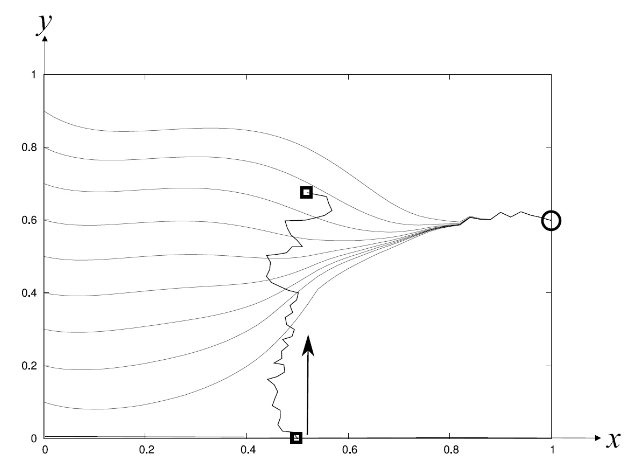

Figure 12. Simulation result for Test 4

(4) Test 4 assumes the scenario where the obstacle moves randomly with a random vector $\boldsymbol{B}$ as shown in Figure 11, where $|\boldsymbol{B}|$ is constant and the angle θ_b changes randomly at each step. As shown in Figure 12, the proposed model is successful for every trial.

In Ref,[14] we have already shown that the simulations are successful by DIRMs model in almost the same condition. In this paper, it is shown that they are successful by the proposed model with half of the parameters compared to DIRMs model.

Lastly, we consider the interpretation for fuzzy rules of the proposed model. Let us consider fuzzy rules of Table 3 obtained by learning. Assume that there are three attributes "small", "middle" and "large" for r_1, r_2, θ_1 and θ_2, and that there are two attributes left ($A_y > 0$) and right ($A_y < 0$) for the direction of A_y. From this result, we can obtain the following fuzzy rules:

Z_1 : (r_1 is small) or (θ_1 is large) or (r_2 is large) or (θ_2 is small)
A_{11} : If Z_1 is small then move to the right
A_{12} : If Z_1 is middle then move to the right
A_{13} : If Z_1 is large then move to the left
Z_2 : (r_1 is large) or (θ_1 is small) or (r_2 is small) or (θ_2 is large)
A_{21} : If Z_2 is small then move to the left
A_{22} : If Z_2 is middle then move to the left
A_{23} : If Z_2 is large then move to the right

Let us consider the behavior for two cases on the places of object.
(I) The case that the mobile agent approaches to the obstacle:
In this case, "r_1 is small" and "r_2 is large" are valid.
(i) If θ_1 is large, the mobile agent move to the left based on A_{13}.
(ii) If θ_1 is not so large, the mobile agent move to the right based on A_{11} or A_{12}.
Likewise, we can also explain about the rule of Z_2.
(II) The case that the mobile agent approaches to the destination:
In this case, "r_1 is large" and "r_2 is small" are valid.
(i) If θ_2 is large, the mobile agent move to the right based on A_{23}.
(ii) If θ_2 is not so large, the mobile agent move to the left based on A_{21} or A_{22}.
Likewise, we can also explain about the rule of Z_1.

That is, it is shown that the mobile agent moves away from the obstacle when the mobile agent approaches to the obstacle and mobile agent moves to the direction of the destination when the mobile agent approaches to the destination.

6. Conclusions

In previous papers, we proposed the SIRMs model with LT of input variables. In this paper, we showed the effectiveness of the proposed method for system identifications. In the second simulation of classification problems, well-known benchmark datasets Iris, Wine, Sonar and BCW are used. According to the simulation result, it is difficult for the conventional model to implement classification problems with large number of parameters such as Wine, Sonar and BCW. However, the proposed model can implement them while keeping high accuracy. Further, as one of control problems, the simulation of obstacle avoidance problem is performed. The simulation result shows that the proposed model outperforms conventional models in terms of accuracy and the number of parameters. In our future work, we will consider to refine the proposed model.

References

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

[2] Gupta MM, Jin L, Homma N. Static and Dynamic Neural Networks. IEEE Press. 2003.

[3] Casillas J, Cordon O, Herrera F, *et al*, Accuracy Improvements in Linguistic Fuzzy Modeling. Studies in Fuzziness and Soft Computing. 2003; 129.

[4] Liu B. Theory and Practice of Uncertain Programming. Studies in Fuzziness and Soft Computing. 2009; 239.

[5] Cordon O. A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems. Journal of Approximate Reasoning. 2011; 52: 894913. http://dx.doi.org/10.1016/j.ijar.2011.03.004

[6] Araki S, Nomura H, Hayashi I, *et al*. A Fuzzy Modeling with Iterative Generation Mechanism of Fuzzy Inference Rules, Journal of Japan Society for Fuzzy Theory & Systems. 1992; 4(4): 722-32.

[7] Fukumoto S, Miyajima H, Kishida K, *et al*. A Destructive Learning Method of Fuzzy Inference Rules. Proc. of IEEE on Fuzzy Systems. 1995: 687-94.

[8] Kishida K, Miyajima H, Maeda M, *et al*. A Self-tuning Method of Fuzzy Modeling using Vector Quantization. Proceedings of FUZZ-IEEE' 97. 1997: 397-402.

[9] Pedrycz W, Izakian H. Cluster-Centric Fuzzy Modeling. IEEE Trans. on Fuzzy Systems. 2014; 22(6): 1585-97. http://dx.doi.org/10.1109/TFUZZ.2014.2300134

[10] Fukumoto S, Miyajima H. Learning Algorithms with Regularization Criteria for Fuzzy Reasoning Model. Journal of Innovative Computing, Information and Control. 2006; 1(1): 249-63.

[11] Yi J, Yubazaki N, Hirota K. A Proposal of SIRMs Dynamically Connected Fuzzy Inference Model for Plural Input Fuzzy Control. Fuzzy Sets and Systems. 2002; 125: 79-92. http://dx.doi.org/10.1016/S0165-0114(00)00135-4

[12] Shigei N, Miyajima H, Nagamine S. 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. 2009: 118-26.

[13] Miike S, Miyajima H, Shigei N, *et al*. 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. 2010: 621-629.

[14] Miyajima H, Shigei N, Miyajima H. Fuzzy Inference Systems Composed of Double-Input Rule Modules for Obstacle Avoidance Problems. IAENG International Journal of Computer Science. 2014; 41(4): 222-30.

[15] Miyajima H, Shigei N, Miyajima H. A Proposal of SIRMs Model with Linear Transformation of Input Variables. Proc. Joint 7th International Conference on Soft Computing and Intelligent Systems and 15th International Symposium on Advanced Intelligent Systems. 2014: 311-6.

[16] Miyajima H, Shigei N, Miyajima H. SIRMs Fuzzy Inference Model with Linear Transformation of Input Variables and Universal Approximation. Advances in Computational Intelligence: Proc. 13th International Work Conference on Artificial Neural Networks, Part I. 2015: 561-75.

[17] Miyajima H, Shigei N, Miyajima H. Some Properties on Fuzzy Inference Systems Composed of Small Number of Input Rule Modules. Advances in Fuzzy Sets and Systems. 2015; 20(2): 155-75. http://dx.doi.org/10.17654/AFSSDec2015_155_175

[18] UCI Repository of Machine Learning Databases and Domain Theories.