Proceedings of the International MultiConference of Engineers and Computer Scientists 2014 Vol I,
IMECS 2014, March 12 - 14, 2014, Hong Kong

An Application of Fuzzy Inference System Composed of Double-Input Rule Modules to Control Problems

Hirofumi Miyajima†1, Fumihiro Kawai†2, Noritaka Shigei†3, and Hiromi Miyajima†4

***Abstract*—The automatic construction of fuzzy system with a large number of input variables involves many difficulties such as large time complexity and getting stuck in a shallow and local minimum. As models to overcome them, the SIRMs (Single-Input Rule Modules) and DIRMs(Double-Input Rule Modules) models have been proposed. In some numerical simulations such as EX-OR problem, it was 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 control problem such as obstacle avoidance. As a result, it is shown that DIRMs model is also more effective than SIRMs model about control problem. Further, we propose a constructive DIRMs model with the reduced number of modules 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[1], [2] have been made. The aim of these studies is to construct automatically fuzzy reasoning rules from input and output data based on the steepest descend method. Obvious drawbacks of the steepest descend method are its large computational complexity and getting stuck in a shallow local minimum. In order to overcome them, some novel methods have been developed as shown in the references[3], [4], [5], [6], [7]. The SIRMs (Single-Input Rule Modules) model aims to obtain a better solution by using fuzzy inference system composed of SIRMs[8], where output is determined as the weighted sum of all modules. 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[9]. 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[10]. Further, there exists the difference of the ability between DIRMs and SIRMs models as shown later in the paper. 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 does the agent (or robot) avoid the obstacle and reach the specified point. We show that DIRMs model can simulate the problem but SIRMs model can not. The simulation results show that the proposed methods are also superior in control problem to the conventional SIRMs model.

Affiliation: Graduate School of Science and Engineering, Kagoshima University, 1-21-40 Korimoto, Kagoshima 890-0065, Japan
corresponding author to provide email: miya@eee.kagoshima-u-ac.jp
†1 email: k3768085@kadai.jp
†2 email: k8005442@kadai.jp
†3 email: shigei@eee.kagoshima-u-ac.jp
†4 email: miya@eee.kagoshima-u-ac.jp

II. Fuzzy Inference Model and its Learning

A. Fuzzy Inference Model

The conventional fuzzy inference model using the steepest descend method is described[1]. 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

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

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_i for input $\boldsymbol{x}$ is expressed as

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

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

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

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

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

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^*.

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

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

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[1].

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

where t is iteration times and K_α is a constant. In the following, the case of the triangular membership function is explained. From the Eqs.(2) to (6), $\frac{\partial E}{\partial \alpha}$'s are calculated as follows:

$$\frac{\partial E}{\partial c_{ij}} = \frac{\mu_j}{\sum_{j=1}^{n} \mu_j} \cdot (y^* - y^r) \cdot (w_j - y^*) \cdot \frac{2\mathrm{sgn}(x_i - c_{ij})}{b_{ij} \cdot M_{ij}(x_i)}, \tag{8}$$

$$\frac{\partial E}{\partial b_{ij}} = \frac{\mu_j}{\sum_{j=1}^{n} \mu_j} \cdot (y^* - y^r) \cdot (w_j - y^*) \cdot \frac{1 - M_{ij}(x_i)}{M_{ij}(x_i) \cdot b_{ij}}, \quad \text{and} \tag{9}$$

$$\frac{\partial E}{\partial w_j} = \frac{\mu_j}{\sum_{j=1}^{n} \mu_j} \cdot (y^* - y^r), \tag{10}$$

where

$$\mathrm{sgn}(z) = \begin{cases} -1 & ; \ z < 0 \\ 0 & ; \ z = 0 \\ 1 & ; \ z > 0. \end{cases} \tag{11}$$

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:

$$E = \frac{1}{P} \sum_{p=1}^{P} (y_p^* - y_p^r)^2. \tag{12}$$

The conventional learning algorithm is shown below[7].

Learning Algorithm A

Step 1: The initial number of rules, c_{ij}, b_{ij} and w_j are set. 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).

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

Step 7: Real number w_j is updated by Eq.(10).

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

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.(12). If $E(t) \le \theta_1$ then learning is terminated.

Step 11: If $t \ne T_{max}$ 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[9]. Let U_k^m be the set of all ordered k-tuples of Z_m, that is

$$U_k^m = \{l_1 \cdots l_k | l_i < l_j \text{ if } i < j\}. \tag{13}$$

Example 1. $U_2^4 = \{12, 13, 14, 23, 24, 34\}$, $U_1^4 = \{1, 2, 3, 4\}$.

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

$$\begin{aligned} &\mathrm{SNIRM}{-}l_1 \cdots l_k : \\ &\quad \{R_i^{l_1 \cdots l_k} : \text{if } x_{l_1} \text{ is } M_i^{l_1} \text{ and } \cdots \text{ and } x_{l_k} \text{ is } M_i^{l_k} \\ &\qquad\qquad \text{then } y_{l_1 \cdots l_k} \text{ is } w_i^{l_1 \cdots l_k}\}_{i=1}^n \end{aligned} \tag{14}$$

Example 2. For U_2^4, the obtained system is as follows:

$\mathrm{SNIRM} - 12$:
$\{R_i^{12}$: if x_1 is M_i^1 and x_2 is M_i^2 then y_{12} is $w_i^{12}\}_{i=1}^n$
$\mathrm{SNIRM} - 13$:
$\{R_i^{13}$: if x_1 is M_i^1 and x_3 is M_i^3 then y_{13} is $w_i^{13}\}_{i=1}^n$
$\mathrm{SNIRM} - 14$:
$\{R_i^{14}$: if x_1 is M_i^1 and x_4 is M_i^4 then y_{14} is $w_i^{14}\}_{i=1}^n$
$\mathrm{SNIRM} - 23$:
$\{R_i^{23}$: if x_2 is M_i^2 and x_3 is M_i^3 then y_{23} is $w_i^{23}\}_{i=1}^n$
$\mathrm{SNIRM} - 24$:
$\{R_i^{24}$: if x_2 is M_i^2 and x_4 is M_i^4 then y_{24} is $w_i^{24}\}_{i=1}^n$
$\mathrm{SNIRM} - 34$:
$\{R_i^{34}$: if x_3 is M_i^3 and x_4 is M_i^4 then y_{34} is $w_i^{34}\}_{i=1}^n$

Note that the number of modules in the obtained system is 6.

Example 3. For U_1^4, the obtained system is as follows:

$$\begin{aligned} &\mathrm{SIRM} - 1 : \{R_i^1 : \text{if } x_1 \text{ is } M_i^1 \text{ then } y_1 \text{ is } w_i^1\}_{i=1}^n \\ &\mathrm{SIRM} - 2 : \{R_i^2 : \text{if } x_2 \text{ is } M_i^2 \text{ then } y_2 \text{ is } w_i^2\}_{i=1}^n \\ &\mathrm{SIRM} - 3 : \{R_i^3 : \text{if } x_3 \text{ is } M_i^3 \text{ then } y_3 \text{ is } w_i^3\}_{i=1}^n \\ &\mathrm{SIRM} - 4 : \{R_i^4 : \text{if } x_4 \text{ is } M_i^4 \text{ then } y_4 \text{ is } w_i^4\}_{i=1}^n \end{aligned}$$

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:

$$\mu_i^{l_1 \cdots l_k} = M_i^{l_1}(x_{l_1}) M_i^{l_2}(x_{l_2}) \cdots M_i^{l_k}(x_{l_k}), \tag{15}$$

$$y_{l_1 \cdots l_k}^o = \frac{\sum_{i=1}^{n} \mu_i^{l_1 \cdots l_k} w_i^{l_1 \cdots l_k}}{\sum_{i=1}^{n} \mu_i^{l_1 \cdots l_k}}. \tag{16}$$

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.

$$y^* = \sum_{L \in U_k^m} h_L \cdot y_L^o \tag{17}$$

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

$$\frac{\partial E}{\partial h_L} = (y^* - y^r) y_L^O, \tag{18}$$

$$\frac{\partial E}{\partial w_i^L} = h_L \cdot \frac{\mu_i^L}{\sum_{i=1}^{n} \mu_i^L} (y^* - y^r), \tag{19}$$

$$\frac{\partial E}{\partial c_i^L} = h_L \cdot (y^* - y^r) \frac{w_i^L - y_L^o}{\sum_{i=1}^{n} \mu_i^L} \frac{2\mathrm{sgn}(x_i - c_i^L)}{b_i^L \cdot M_i^L(x_i)}, \tag{20}$$

$$\frac{\partial E}{\partial b_i^L} = h_L \cdot (y^* - y^r) \frac{w_i^L - y_L^o}{\sum_{i=1}^{n} \mu_i^L} \frac{x_i - c_i^L}{(b_i^L)^2}. \tag{21}$$

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

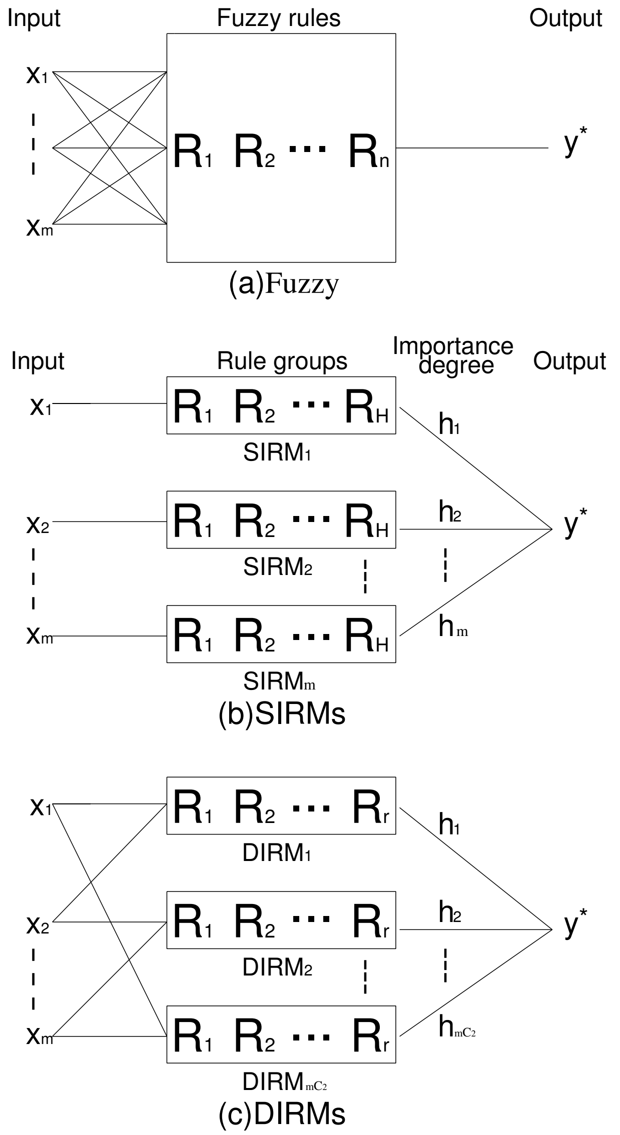


Fig. 1. The relation among the conventional fuzzy , SIRMs and DIRMs models

The cases of $k = 1$ and $k = 2$ are called SIRMs and DIRMs models, respectively. Fig.1 shows the relation among the conventional fuzzy inference , SIRMs and DIRMs models. Example 2 and Example 3 are DIRMs and SIRMs models for m=4, respectively. It is known that the SIRMs model does not always achieve good performance in non-linear problems[10]. 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 problems compared to the SIRMs model and is simpler than the conventional fuzzy model.

A learning algorithm for SNIRMs (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.(15).
Step 6: Inference output y_p^* is calculated by Eqs.(16) and (17).
Step 7: Importance degree h_L is updated by Eq.(18).
Step 8: Real number w_i^L is updated by Eq.(19).
Step 9: Parameters c_i^L and b_i^L are updated by Eqs.(20) and (21).
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.(12). 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, DIRMs and SIRMs models are $O(H^m)$, $O(m^2H^2)$ and $O(mH)$, respectively, where H is the number of fuzzy partitions. In order to reduce the number of rule for DIRMs model, we propose the constructive DIRMs model with $O(mH^2)$ rules. The model is composed of SIRMs model and $O(mH^2)$ rules of DIRMs models. The algorithm is as follows:

Learning Algorithm C (The constructive 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 of step1.
Step 3: In order to adjust the parameters of the system, algorithm B is performed.

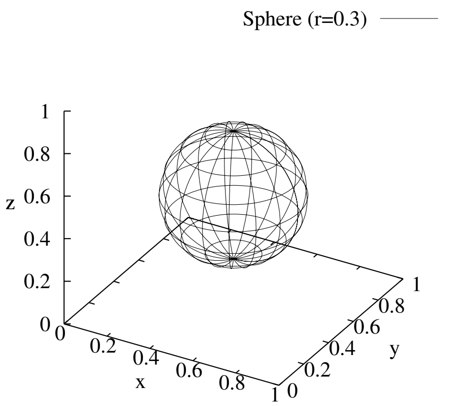
IV. Numerical Simulations

In order to show the effectiveness of DIRMs models, numerical simulations for function approximation and obstacle avoidance are performed.

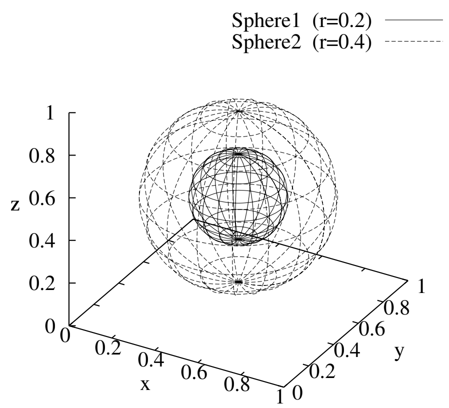
A. Two-category Classification Problems

First, we perform two-category classification problems as in Fig.2 to investigate the basic feature of the proposed method and to compare it with the SIRMs model. 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. 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 Gauss function is used as the membership functions. The number of partitions for each model is 3. The results of classification are shown in Table II. TableII, A, B, and C means Algorithm A, B, and C, respectively, and the number in parenthesis means the number of parameters. Further, the upper and lower values in each box mean the error rates for learning and test, respectively.

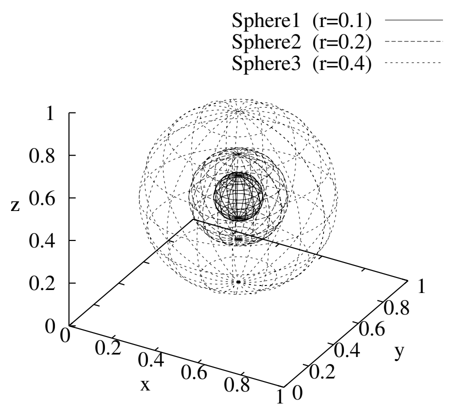
ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)



(a) Sphere



(b) Double-Sphere



(c) Triple-Sphere

Fig. 2. Two-category Classification Problems

TABLE I
INITIAL CONDITION FOR SIMULATION.

	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

TABLE II
SIMULATION RESULT FOR TWO-CATEGORY CLASSIFICATION PROBLEM.

	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

TABLE III
INITIAL CONDITION FOR SIMULATION.

	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

B. Obstacle avoidance

1) Obstacle avoidance: From (operation) data given by an examinee to avoid obstacle, fuzzy inference rule for each model is constructed. As shown in Fig.3, 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.4. From the data, fuzzy inference rule to perform the trace of Fig.4 is constructed for each model, where the simulation condition is shown in Table III. The number of partitions for each model is 5. Fig.5 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.5. 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.6, the results are successful for both models.

2) Obstacle avoidance and arriving at the designated place: As shown in Fig.7, the distance d_1 and the angle θ_1

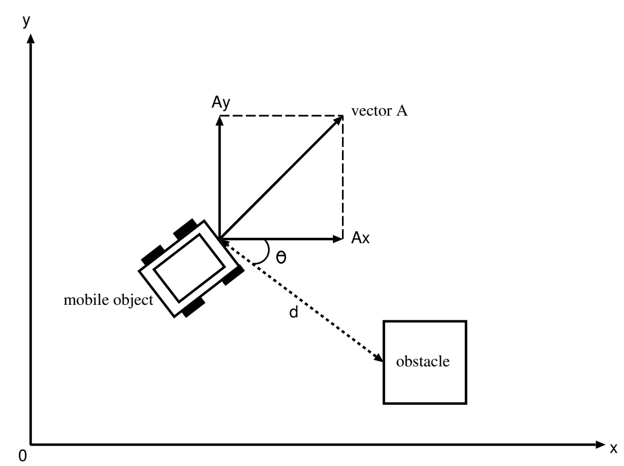


Fig. 3. Simulation on obstacle avoidance.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

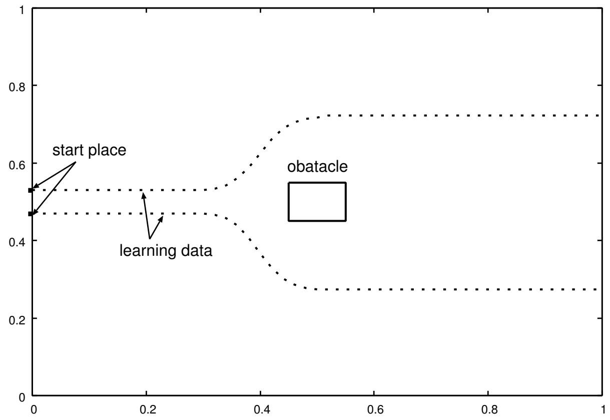
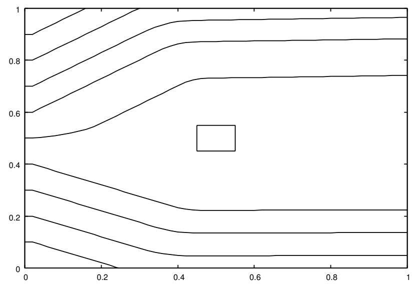
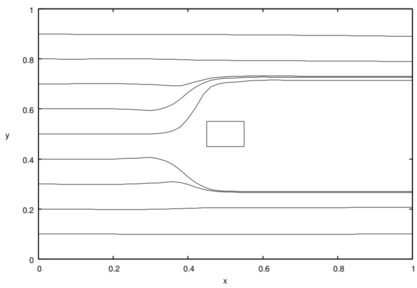


Fig. 4. Learning data denoted by dots, to avoid obstacle.

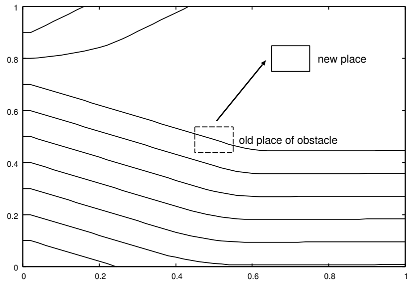


(a) SIRMs.

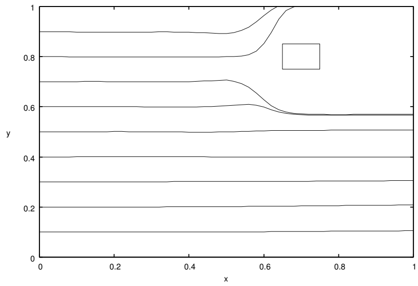


(b) DIRMs.

Fig. 5. Simulation result for obstacle avoidance starting from various places learning



(a) SIRMs.



(b) DIRMs.

Fig. 6. Simulation for obstacle avoidance placed at different place after learning.

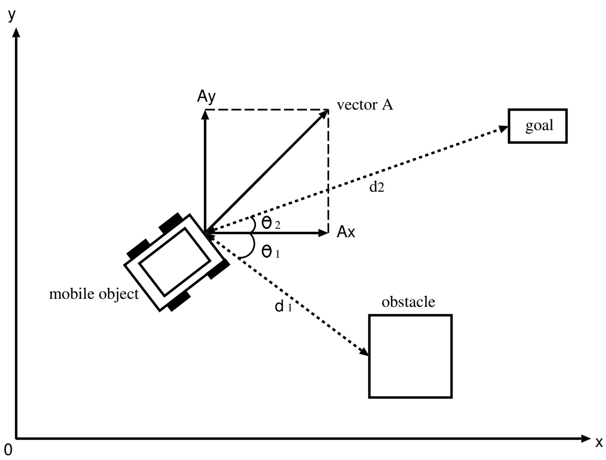


Fig. 7. Simulation on obstacle avoidance and arriving at the goal.

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 rule that mobile object avoids obstacle and arrives at the designated place. From (operation) data, fuzzy inference rule for each model is constructed as shown as 200 points in Fig.8. 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. The simulation results for SIRMs and DIRMs models are unsuccessful and successful, respectively, as shown in Fig.9. Fig.9 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. In Fig.9(a), mobile agent collides with obstacle in simulation of starting place at $(0.4, 0)$. Further, test simulations with the places, (1, 0.35), different from the designated place in learning are performed for DIRMs. The results are also successful in DIRMs model as shown in Fig.10. Lastly, we performed the same simulations for the constructive DIRMs model. As a result, all simulations are also successful in the constructive DIRMs. Therefore, the number 6 of modules for DIRMs model can be reduced to the model composed of 3 modules.

V. Conclusion

It is well known that the construction of fuzzy system with a large number of input variables involves many difficulties such as large time complexities and getting stuck in a shallow local minimum. As one model to overcome them, SIRMs model has been proposed. However, such a simple model does not always achieve good performance in complex non-linear systems. Therefore, we have proposed the DIRMs model, in which each module is composed of two variables. DIRMs model is superior in pattern classification to SIRMs one as shown in the paper. Further, we have shown that DIRMs model is more effective than SIRMs model in the control problem as obstacle avoidance. It means that there exists the control problem with the need of cooperation of

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

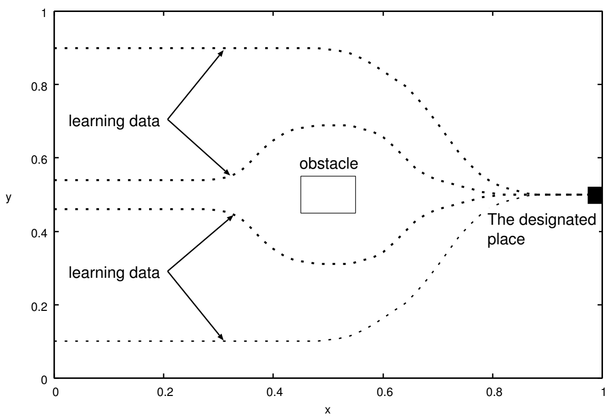
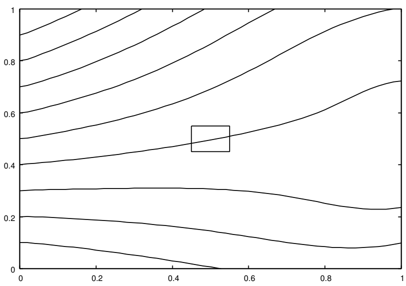
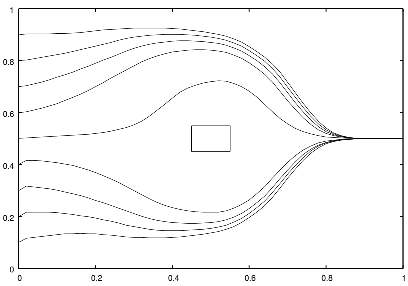


Fig. 8. Learning data to avoid obstacle and arrive at the designated place (1, 0.35).



(a) SIRMs.



(b) DIRMs.

Fig. 9. Simulation for obstacle avoidance and arriving at the different designated place after learning.

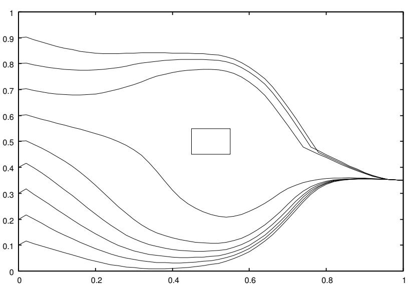


Fig. 10. Simulation for obstacle avoidance with the different designated place (1, 0.35) from learning.

two variables at a time as EX-OR problem. As a future work, we will consider theoretical characterization of SIRMs and DIRMs models.

REFERENCES

[1] H. Nomura, I. Hayashi and N. Wakami, A Self-Tuning Method of Fuzzy Reasoning by Delta Rule and Its Application to a Moving Obstacle Avoidance, *Journal of Japan Society for Fuzzy Theory & Systems*, vol.4, no.2, pp. 379–388, 1992 (in Japanese).

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

[3] 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 & Systems*, vol.4, no.4, pp.722–732, 1992.

[4] 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.

[5] H. Nomura, I. Hayashi and N. Wakami, A Learning Method of Simplified Fuzzy Reasoning by Genetic Algorithm, *Proc. of the Int. Fuzzy Systems and Intelligent Control Conference*, pp.236–245, 1992.

[6] L.X. Wang and J.M. Mendel, Fuzzy Basis Functions, Universal Approximation, and Orthogonal Least Square Learning, *IEEE Trans. Neural Networks*, vol.3, no.5, pp.807–814, 1992.

[7] 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.

[8] 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.

[9] 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.

[10] 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).

