

Fuzzy Identification of Systems and Its Applications to Modeling and Control

TOMOHIRO TAKAGI AND MICHIO SUGENO

Abstract—A mathematical tool to build a fuzzy model of a system where fuzzy implications and reasoning are used is presented in this paper. The premise of an implication is the description of fuzzy subspace of inputs and its consequence is a linear input–output relation. The method of identification of a system using its input–output data is then shown. Two applications of the method to industrial processes are also discussed: a water cleaning process and a converter in a steel-making process.

I. INTRODUCTION

THE main purpose of this paper is to present a mathematical tool to build a fuzzy model of a system.

There has been a considerable number of studies [1]–[3] on fuzzy control where fuzzy implications are used to express control rules. Most of those implications contain fuzzy variables with unimodal membership functions since those are linguistically understandable and thus called linguistic variables. As for reasoning, the so-called compositional rule of inference or its simplified version is used. However, when we use this type of reasoning together with unimodal fuzzy variables for multivariable control, we have much difficulty since we need many fuzzy variables, i.e., many implications; it is usual to use five variables in each dimension of input space.

The authors have suggested multidimensional fuzzy reasoning [6] where we can surprisingly reduce the number of implications. The study in this paper is related to the above idea of reasoning, where a fuzzy implication is improved and reasoning is simplified.

Recently some studies [4], [5] have also been reported on fuzzy modeling of a system. Fuzzy modeling based on fuzzy implications and reasoning may be one of the most important fields in fuzzy systems theory. Here we have to deal with a multivariable system in general and so therefore have to consider multidimensional reasoning method.

Generally speaking, model building by input–output data is characterized by two things; one is a mathematical tool to express a system model and the other is the method of identification. A mathematical tool itself is required to have simplicity and generality. The fuzzy implication presented as a tool in the paper is quite simple. It is based on a fuzzy partition of input space. In each fuzzy subspace a linear input–output relation is formed. The output of fuzzy

reasoning is given by the aggregation of the values inferred by some implications that were applied to an input.

This paper also shows the method of identification of a system using its input–output data. As is well-known, identification is divided into two parts: structure identification and parameter identification.

In its nature structure identification is almost independent of a format of system description. We omit this part in the paper, so by identification we mean parameter identification in fuzzy implications. However, a kind of structure problem partly appears.

Finally this paper shows two applications to industrial processes. One is a water cleaning process where an operator's control actions are fuzzily modeled to design a fuzzy controller. The other is a converter in the steel-making process where the conversion process is fuzzily modeled and model-based fuzzy control is considered.

Most fuzzy controllers have been designed based on human operator experience and/or control engineer knowledge. It is, however, often the case that an operator cannot tell linguistically what kind of action he takes in a particular situation. In this respect it is quite useful to give a way to model his control actions using numerical data. Further, if there is no reason to believe that an operator's control is optimal, we have to develop model-based control just as in ordinary control theory. To this aim it is necessary to consider a means for fuzzy modeling of a system.

II. FORMAT OF FUZZY IMPLICATION AND REASONING ALGORITHM

In this paper we denote the membership function of a fuzzy set A as $A(x)$, $x \in X$. All the fuzzy sets are associated with linear membership functions. Thus, a membership function is characterized by two parameters giving the greatest grade 1 and the least grade 0. The truth value of a proposition " x is A and y is B " is expressed by

$$|x \text{ is } A \text{ and } y \text{ is } B| = A(x) \wedge B(y).$$

A. Format of Implications

We suggest that a fuzzy implication R is of the format

$$R: \text{If } f(x_1 \text{ is } A_1, \dots, x_k \text{ is } A_k) \text{ then } y = g(x_1, \dots, x_k) \quad (1)$$

Manuscript received September 30, 1983; revised April 6, 1984.

The authors are with the Department of Systems Science, Tokyo Institute of Technology, 4259 Nagatsuta, Midori-ku, Yokohama 227, Japan.

where

y	Variable of the consequence whose value is inferred.
$x_1 - x_k$	Variables of the premise that appear also in the part of the consequence.
$A_1 - A_k$	Fuzzy sets with linear membership functions representing a fuzzy subspace in which the implication R can be applied for reasoning.
f	Logical function connects the propositions in the premise.
g	Function that implies the value of y when $x_1 - x_k$ satisfies the premise.

In the premise if A_i is equal to X_i for some i where X_i is the universe of discourse of x_i , this term is omitted; x_i is unconditioned.

Example 1:

R : If x_1 is small and x_2 is big then $y = x_1 + x_2 + 2x_3$.

This implication states that if x_1 is small and x_2 is big, then the value of y would be equal to the sum of x_1 , x_2 and $2x_3$, where x_3 is unconditioned in the premise.

In the sequel we shall only use “and” connectives in the premise and adopt a linear function in the consequence as is seen in the above example. So an implication is written as

$$\begin{aligned} R: \text{ If } x_1 \text{ is } A_1 \text{ and } \cdots \text{ and } x_k \text{ is } A_k \\ \text{ then } y = p_0 + p_1x_1 + \cdots + p_kx_k. \end{aligned} \quad (2)$$

B. Algorithm of Reasoning

Suppose that we have implications R^i ($i = 1, \dots, n$) of the above format. When we are given

$$(x_1 = x_1^0, \dots, x_k = x_k^0)$$

where $x_1^0 - x_k^0$ are singletons, the value of y is inferred in the following steps.

1) For each implication R^i , y^i is calculated by the function g^i in the consequence

$$\begin{aligned} y^i &= g^i(x_1^0, \dots, x_k^0) \\ &= p_0^i + p_1^i x_1^0 + \cdots + p_k^i x_k^0. \end{aligned} \quad (3)$$

2) The truth value of the proposition $y = y^i$ is calculated by the equation

$$\begin{aligned} |y = y^i| &= |x_1^0 \text{ is } A_1^i \text{ and } \cdots \text{ and } x_k^0 \text{ is } A_k^i| \wedge |R^i| \\ &= (A_1^i(x_1^0) \wedge \cdots \wedge A_k^i(x_k^0)) \wedge |R^i| \end{aligned} \quad (4)$$

where $|*|$ means the truth value of proposition $*$ and \wedge stands for min operation, and $|x^0 \text{ is } A| = A(x^0)$, i.e., the grade of the membership of x^0 .

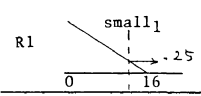
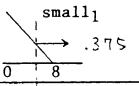
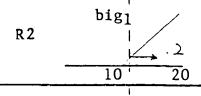
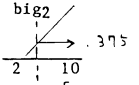
For simplicity we assume

$$|R^i| = 1 \quad (5)$$

so the truth value of the consequence obtained is

$$|y = y^i| = A_1^i(x_1^0) \wedge \cdots \wedge A_k^i(x_k^0). \quad (6)$$

TABLE I

Implication	Premise	Consequence	Tv
R1			$y = 12 + 5$ $= 17$ $.25 \wedge .375$ $= .25$
R2			$y = 2 \times 12$ $= 24$.2
R3			$y = 3 \times 5$ $= 15$.375

3) The final output y inferred from n implications is given as the average of all y^i with the weights $|y = y^i|$:

$$y = \frac{\sum |y = y^i| \times y^i}{\sum |y = y^i|}. \quad (7)$$

Example 2: Suppose that we have the following three implications:

R^1 : If x_1 is small₁ and x_2 is small₂ then $y = x_1 + x_2$

R^2 : If x_1 is big₁ then $y = 2 \times x_1$

R^3 : If x_2 is big₂ then $y = 3 \times x_2$

Table I shows the reasoning process by each implication when we are given $x_1 = 12$, $x_2 = 5$. The column “Premise” in Table I shows the membership functions of the fuzzy sets “small” and “big” in the premises. The column “Consequence” shows the value of y^i calculated by the function g^i of each consequence and “Tv” shows the truth value of $|y = y_i|$. For example, we have

$$\begin{aligned} |y = y^1| &= |x_1^0 = \text{small}_1| \wedge |x_2^0 = \text{small}_2| \\ &= \text{small}_1(x_1^0) \wedge \text{small}_2(x_2^0) \\ &= 0.25. \end{aligned} \quad (8)$$

The value inferred by the implications is obtained by referring to Table I

$$\begin{aligned} y &= \frac{0.25 \times 17 + 0.2 \times 24 + 0.375 \times 15}{0.25 + 0.2 + 0.375} \\ &\approx 17.8. \end{aligned} \quad (9)$$

C. Properties of Reasoning

We show two illustrative examples to find the performance of the presented reasoning algorithm.

Example 3: Suppose we have two implications.

$$R^1: \text{ If } x \text{ is } \begin{array}{c} \text{graph of } \mu_{R^1}(x) \\ \text{from } (4, 0) \text{ to } (10, 1) \end{array} \text{ then } y = 0.2x + 9$$

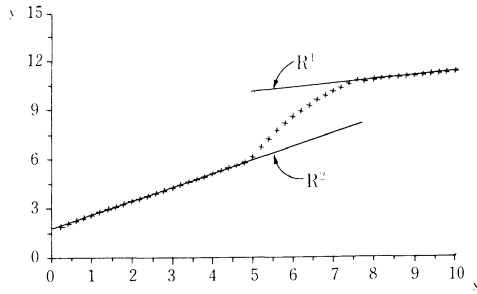


Fig. 1. Results of fuzzy reasoning.

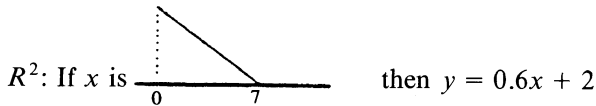


Fig. 2. Results of fuzzy reasoning in example 2.

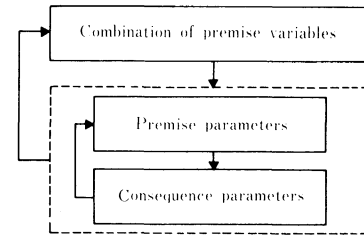


Fig. 3. Outline of the algorithm.

Then Fig. 1 shows the relation of x and y , which is marked by the symbol $+$. The line R^i shows the function in the consequence of R^i . The equation in a consequence can be interpreted to represent a law that holds in the fuzzy subspace defined in a premise.

Let us consider the difference between ordinary piecewise linear approximation method and the presented method. If we take piecewise linear approximation, we first divide input space into crisp subspaces and next build a linear relation in each subspace. For example, in the case shown in Fig. 1, we need another linear relation connecting R^1 and R^2 . It is easily seen that those three straight lines are not smoothly connected. On the other hand, the presented method enables us to reduce the number of piecewise linear relations and also to connect them smoothly. It is of crucial importance to reduce the number of linear relations in a multidimensional case.

Further, with the fuzzy partition of input space, we can put linguistic conditions to linear relations such as " x_1 is small and x_2 is big." Thus, for example, we can use the variable that is observed only by man (see Section IV-B).

Example 4: Fig. 2 shows the input-output relation expressed by the implications of Example 2. In this case the premises are two-dimensional. In the figure the curved surface shows a highly nonlinear input-output relation whose shape reflects the dominance of each implication in its essentially applicable area and also the conflict of implications in an overlapped area.

III. ALGORITHM OF IDENTIFICATION

As has been stated, we consider a fuzzy model consisting of some number of implications that are of the format

$$\text{If } x_1 \text{ is } A_1 \text{ and } \cdots \text{ and } x_k \text{ is } A_k \\ \text{then } y = p_0 + p_1 \cdot x_1 + \cdots + p_k \cdot x_k$$

characterized by "and" connective and a linear equation.

For identification we have to determine the following three items by using the input-output data of an objective system.

- 1) x_1, \cdots, x_k Variables composing the premises of implications.

- 2) A_1, \cdots, A_k Membership functions of the fuzzy sets in the premises, abbreviated as premise parameters.
- 3) p_0, \cdots, p_k Parameters in the consequences.

Notice that all the variables in a premise may not always appear. The items 1) and 2) are related to the partition of space of input variables into some fuzzy subspaces. The item 3) is related to describing an input-output relation in each fuzzy subspace.

We can consider the relation among three items hierarchically from 1) down to 3). The algorithm of the identification of implications is divided into three steps corresponding to the above three items. We first give a brief explanation of the algorithm at each step.

1) **Choice of Premise Variables:** First a combination of premise variables is chosen out of possible input variables we can consider. Next the optimum premise and consequence parameters are identified according to the steps 2) and 3), and also the errors between the output values of the model and the output data of the objective system are calculated. We then improve the choice of the premise variables so that the performance index is decreased, which is defined as the root mean square of the output errors.

2) **Premise Parameters Identification:** In this step the optimum premise parameters are searched for the premise variables chosen at step 1). Assuming the values of premise parameters, we can obtain the optimum consequence parameters together with the performance index according to step 3). So the problem of finding the optimum premise parameters is reduced to a nonlinear programming problem minimizing the performance index.

3) **Consequence Parameters Identification:** The consequence parameters that give the least performance index are searched by the least squares method for the given

premise variables in the step 1) and parameters in step 2).

The outline of the algorithm is shown in Fig. 3. From the next section we shall discuss the method in detail with an illustrative example at each step.

A. Consequence Parameters Identification

In this section we show how to determine the optimum consequence parameters to minimize the performance index, provided that both the premise variables and parameters are given. The performance index has been defined above as a root mean square of the output errors, which means the differences between the output data of an original system and those of a model.

Let a system be represented by the following implications:

$$\begin{aligned} R^1 \quad & \text{If } x_1 \text{ is } A_1^1, \dots, \text{ and } x_k \text{ is } A_k^1 \\ & \text{then } y = p_0^1 + p_1^1 \cdot x_1 + \dots + p_k^1 \cdot x_k \\ & \quad \vdots \\ R^n \quad & \text{If } x_1 \text{ is } A_1^n, \dots, \text{ and } x_k \text{ is } A_k^n \\ & \text{then } y = p_0^n + p_1^n \cdot x_1 + \dots + p_k^n \cdot x_k. \end{aligned}$$

Then the output y for the input (x_1, \dots, x_k) is obtained as

$$y = \frac{\sum_{i=1}^n (A_1^i(x_1) \wedge \dots \wedge A_n^i(x_n)) \cdot (p_0^i + p_1^i \cdot x_1 + \dots + p_k^i \cdot x_k)}{\sum_{i=1}^n (A_1^i(x_1) \wedge \dots \wedge A_n^i(x_n))}. \quad (10)$$

Let β_i be

$$\beta_i = \frac{A_1^i(x_1) \wedge \dots \wedge A_n^i(x_n)}{\sum_{i=1}^n (A_1^i(x_1) \wedge \dots \wedge A_n^i(x_n))} \quad (11)$$

then

$$\begin{aligned} y &= \sum_{i=1}^n \beta_i (p_0^i + p_1^i \cdot x_1 + \dots + p_k^i \cdot x_k) \\ &= \sum_{i=1}^n (p_0^i \cdot \beta_i + p_1^i \cdot x_1 \cdot \beta_i + \dots + p_k^i \cdot x_k \cdot \beta_i). \end{aligned} \quad (12)$$

When a set of input-output data $x_{1j}, x_{2j}, \dots, x_{kj} \rightarrow y_j$ ($j = 1 \dots m$) is given, we can obtain the consequence parameters $p_0^i, p_1^i, \dots, p_k^i$ ($i = 1 \dots n$) by the least squares method using (12).

Let X ($m \times n(k+1)$ matrix), Y (m vector) and P ($n(k+1)$ vector) be

$$X = \begin{bmatrix} \beta_{11}, \dots, & \beta_{n1}, x_{11} \cdot \beta_{11}, \dots, & x_{11} \cdot \beta_{n1}, \dots \\ & \dots, & x_{k1} \cdot \beta_{11}, \dots, & x_{k1} \cdot \beta_n \\ \vdots & & & \vdots \\ \beta_{1m}, \dots, & \beta_{nm}, x_{1m} \cdot \beta_{1m}, \dots, & x_{1m} \cdot \beta_{1m}, \dots \\ & \dots, & x_{k1} \cdot \beta_{1m}, \dots, & x_{k1} \cdot \beta_{nm} \end{bmatrix} \quad (13)$$

where

$$\beta_{ij} = \frac{A_{i1}(x_{1j}) \wedge \dots \wedge A_{ik}(x_{kj})}{\sum_j A_{i1}(x_{1j}) \wedge \dots \wedge A_{ik}(x_{kj})} \quad (14)$$

$$Y = [y_1, \dots, y_m]^T \quad (15)$$

$$P = [p_0^1, \dots, p_0^n, p_1^1, \dots, p_1^n, \dots, p_k^1, \dots, p_k^n]^T. \quad (16)$$

Then the parameter vector P is calculated by

$$P = (X^T X)^{-1} X^T Y. \quad (17)$$

It is noted that the proposed method is consistent with the reasoning method. In other words, this method of identification enables us to obtain just the same parameters as the original system, if we have a sufficient number of noiseless output data for the identification.

In this paper the parameter vector P is calculated by a stable-state Kalman filter. The so-called stable-state Kalman filter is an algorithm to calculate the parameters of a linear algebraic equation that gives the least squares of errors. Here we apply it to calculate the parameter vector P in (17).

Let the i th row vector of matrix X defined in (13) be x_i and the i th element of Y be y_i . Then P is recursively calculated by (18) and (19) where S_i is $(n \cdot (k+1)) \times (n \cdot (k+1))$ matrix.

$$P_{i+1} = P_i + S_{i+1} \cdot x_{i+1} \cdot (y_{i+1} - x_{i+1} \cdot P_i) \quad (18)$$

$$S_{i+1} = S_i - \frac{S_i \cdot x_i + x_{i+1} \cdot S_i}{1 + x_{i+1} \cdot S_i \cdot x_{i+1}^T}, \quad i = 0, 1, \dots, m-1 \quad (19)$$

$$P = P_m \quad (20)$$

where the initial values of P_0 and S_0 are set as follows.

$$P_0 = 0 \quad (21)$$

$$S_0 = \alpha \cdot I \quad (\alpha = \text{big number}) \quad (22)$$

where I is the identity matrix.

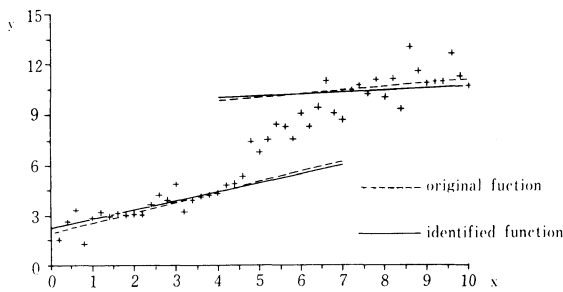


Fig. 4. Results of identification.

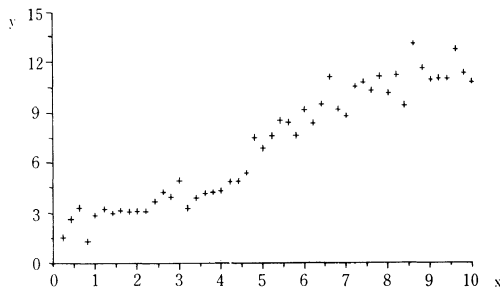
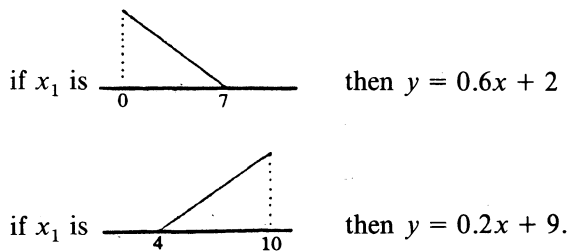


Fig. 5. Input-output data.

Example 5: Suppose in a system



Under the condition that the premises of the model are fixed to those of the original system, the consequences are identified from input-output data as follows, where the noises are added to the data.

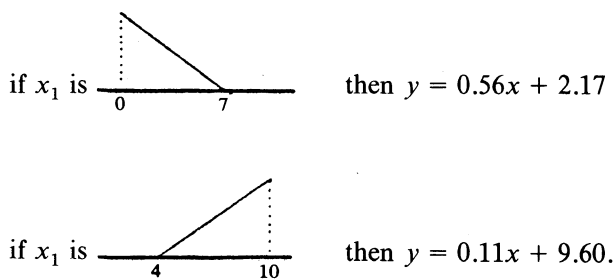


Fig. 4 shows the noised input-output data, the original consequences, and the identified consequences.

B. Premise Parameters Identification

In this section we show how to identify the fuzzy sets in the premises, that is, how to partition the space of premise variables into fuzzy subspaces, provided that the premise variables are chosen.

For example, looking at the input-output data shown in Fig. 5, we can see that the input-output characteristics change as the input x increases. So dividing the space of x into two fuzzy subspaces such that x is small or x is big, we have a model with the following two implications:

$$\text{if } x \text{ is small then } y = a_1x + b_1$$

$$\text{if } x \text{ is big then } y = a_2x + b_2.$$

We next have to determine the membership functions of “small” and “big” as well as the parameters a_1 , b_1 , a_2 and b_2 in the consequences.

As it is easily seen, to divide the spaces into some fuzzy subspaces is to determine the membership functions of the fuzzy sets in the premises. The problem is thus to find the optimum parameters of their membership functions by which the performance index is minimized.

We call this procedure “premise parameter identification.” The algorithm is as follows.

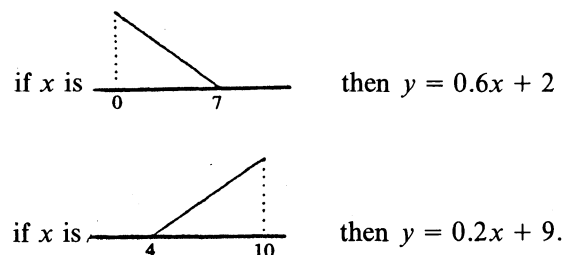
1) Assuming the parameters of the fuzzy sets in the premises, we can obtain the optimum parameters in the consequences that minimize the performance index as discussed in the previous section.

2) The problem of finding the optimum premise parameters minimizing the performance index is reduced to a nonlinear programming problem. In this study we use the well-known complex method for the minimization. Each fuzzy set in the premises is determined by two parameters that give the greatest grade 1 and the least grade 0, since a fuzzy set is assumed to have a linear membership function.

Example 6: This example shows the identification using the input-output data gathered from a preassumed system with noises. The standard deviation of the noises is five percent of that of the outputs. It has to be also noted that we can identify just the same parameters of the premises as the original system if noises do not exist.

It is of great importance to point out the above fact. If it is not the case, we cannot claim the validity of an identification algorithm together with a fuzzy system description language.

Suppose the original system exists with the following two implications:



The functions in the consequences of the implications and the noised input-output data are shown in Fig. 6.

The identified premise parameters are as follows. We can see that almost the same parameters have been derived.

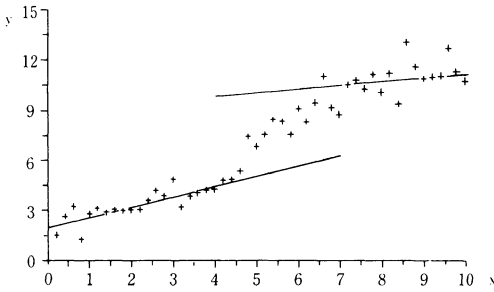


Fig. 6. Consequences and noised data.

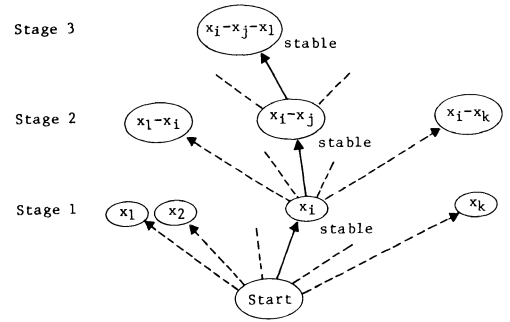


Fig. 7. Choice of premise variables.

if x is $\begin{array}{c} \text{---} \\ 0.0 \end{array} \begin{array}{c} \text{---} \\ 6.6 \end{array}$ then $y = 0.59x + 2.2$

if x is $\begin{array}{c} \text{---} \\ 4.0 \end{array} \begin{array}{c} \text{---} \\ 10.0 \end{array}$ then $y = 0.12x + 9.5$.

C. Choice of Premise Variables

In this section we suggest an algorithm to choose premise variables from the considerable input variables. As has been stated previously, all the variables of the consequences do not always appear in the premises. There are two problems concerned with the algorithm. One is the choice of variables: to choose a variable in the premises implies that its space is divided. The other is the number of divisions. The whole problem is a combinatorial one. So in general there seems no theoretical approach available. Here we just take a heuristic search method described in the following steps.

Suppose that we build a fuzzy model of a k -input x_1, \dots, x_k and single-output system.

Step 1: The range of x_1 is divided into two fuzzy subspaces "big" and "small," and the ranges of the other variables x_2, \dots, x_k are not divided, which means that only x_1 appears in the premises of the implications. This model consisting of two implications is thus

if x_1 is big₁ then ...

if x_1 is small₂ then ...

It is called model 1-1. Similarly, a model in which the range of x_2 is divided and the ranges of the other variables x_1, x_3, \dots, x_k are undivided is called model 1-2. In this way we have k -models, each of which is composed of two implications. In

general, the model 1 - i is of the form

if x_i is big _{i} then ...

if x_i is small _{i} then ...

Step 2: For each model the optimum premise parameters and consequence parameters are found by the algorithm described in the previous sections. The optimum model with the least performance index is adopted out of the k -models. It is called a stable state.

Step 3: Starting from a stable state at step 1, say model 1 - i , where only the variable x_i appears in the premises, take all the combinations of $x_i - x_j$ ($j = 1, 2, \dots, k$) and divide the range of each variable in two fuzzy subspaces. For the combination $x_i - x_i$, the range of x_i is divided into four subspaces, for example, "big," "medium big," "medium small," and "small." Thus we get k -models each of which is named model 2- j . Each model consists of 2×2 implications. Then find again a model with the least performance index just as in step 2 that is also called a stable state at this step.

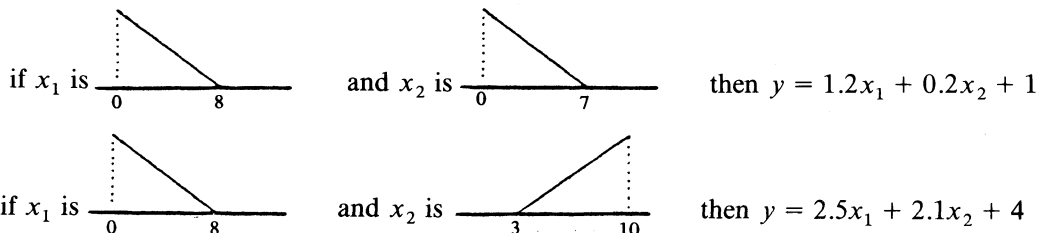
Step 4: Repeat step 3 in a similar way by putting another variable into the premise.

Step 5: The search is stopped if either of the following criteria is satisfied.

- 1) The performance index of a stable state becomes less than the predetermined value.
- 2) The number of implications of a stable state exceeds the predetermined number.

The choice of the variables in the premises proceeds as is shown in Fig. 7.

Example 7: We show an example of identification. The original system is also a fuzzy system with two inputs and single output expressed by the implications as are shown below.



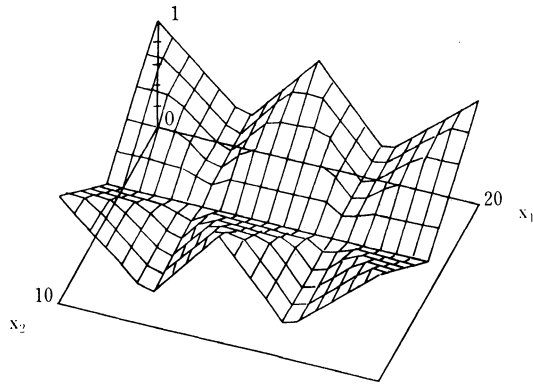


Fig. 8. Fuzzy partition of input space.

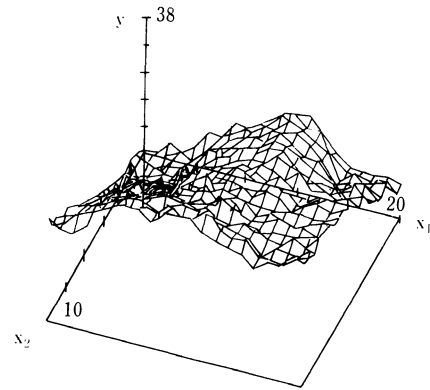


Fig. 10. Identification data with noises.

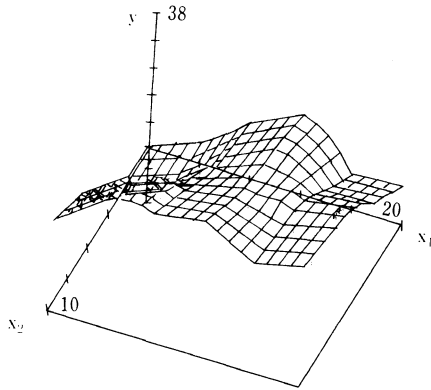


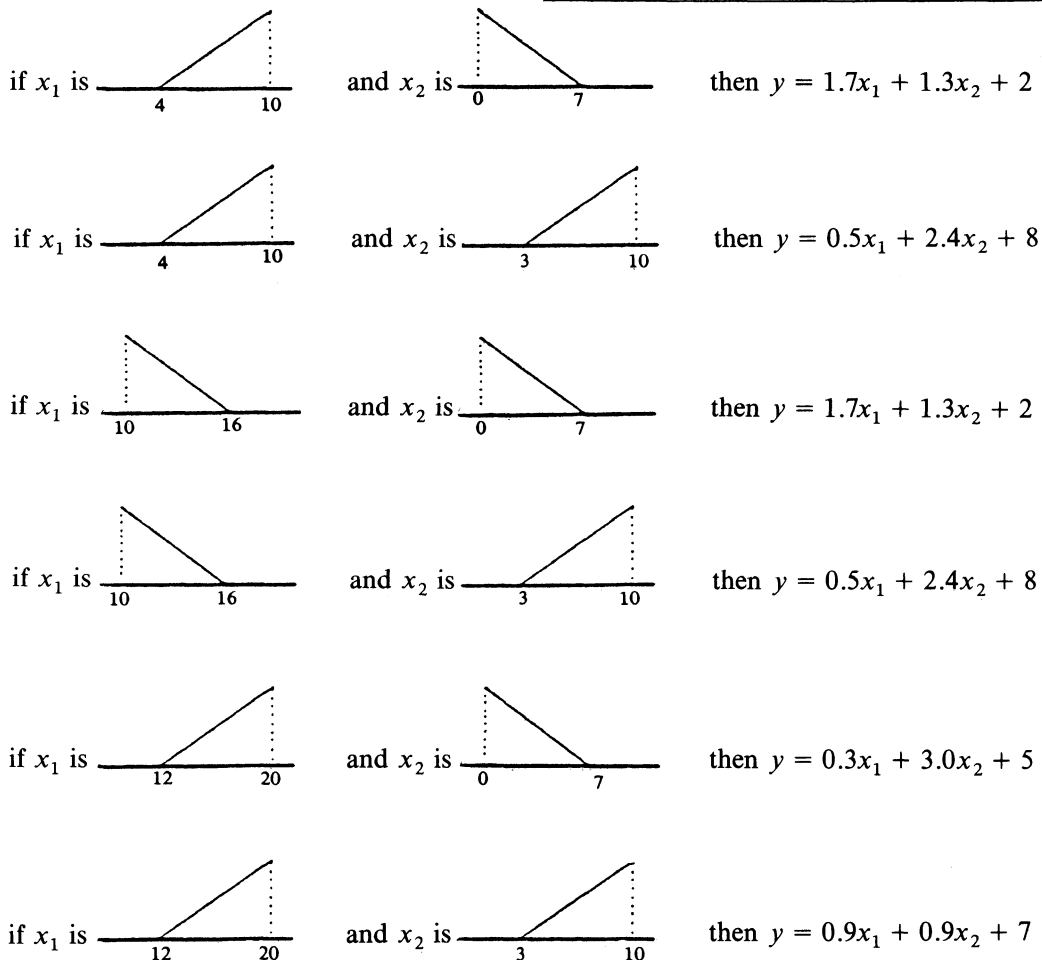
Fig. 9. Input-output relation of the original system.

In this system, the range of x_1 is divided into four fuzzy subspaces and that of x_2 into two fuzzy subspaces. So the number of implications is 4×2 altogether.

Fig. 8 shows a fuzzy partition of the input space in the original system, i.e., the membership functions of the fuzzy relation expressed in the premises.

Fig. 9 shows the input-output relation of the above system. Now 441 input-output data of this system are taken for the identification and noises are added to the outputs, where the standard deviation of the noises is two percent of that of the outputs. Fig. 10 shows the noised data. Now the system is identified using these data.

Stage 1: Let us start from two models, each of which consists of two implications. They are shown together with



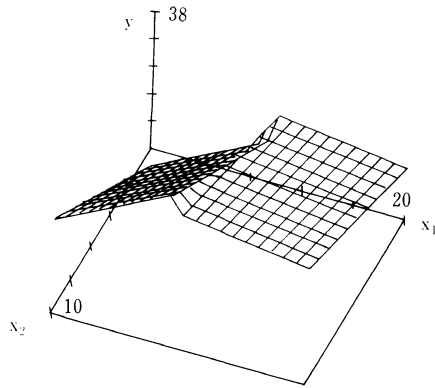


Fig. 11. Input-output relation of the model 1-1.

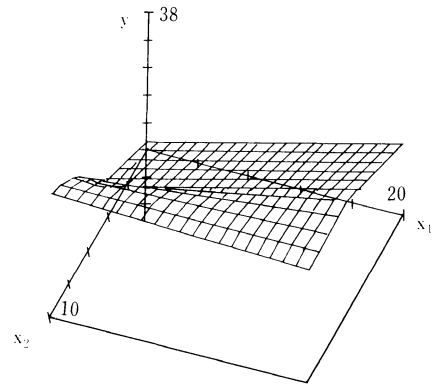


Fig. 12. Input-output relation of the model 1-2.

their performance indices. Figs. 11 and 12 show the input-output relation of the models.

Model 1-1: (the Range of x_1 is Divided)

Implications

$$\text{if } x_1 \text{ is } \begin{array}{|c|c|} \hline 10 & 20 \\ \hline \end{array} \quad \text{then } y = -0.45x_1 + 1.82x_2 + 23.2$$

$$\text{if } x_1 \text{ is } \begin{array}{|c|c|} \hline 0 & 9.5 \\ \hline \end{array} \quad \text{then } y = 1.71x_1 + 3.09x_2 - 4.9$$

performance index = 2.55.

Model 1-2: (the Range of x_2 is Divided)

Implications

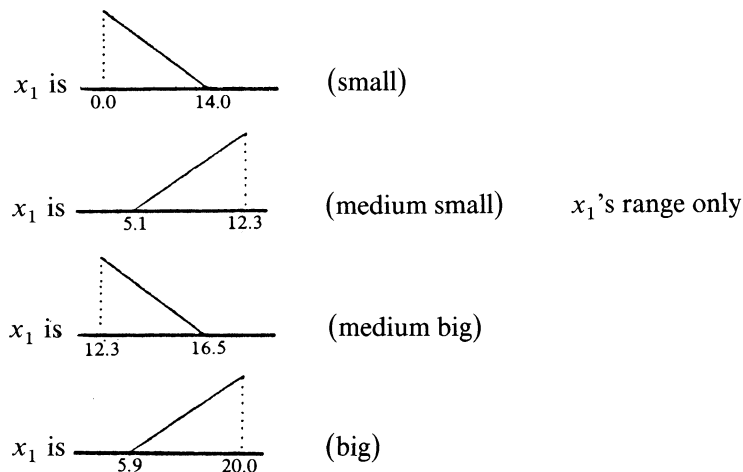
$$\text{if } x_2 \text{ is } \begin{array}{|c|c|} \hline 5.5 & 10 \\ \hline \end{array} \quad \text{then } y = -0.14x_1 + 0.65x_2 + 27.6$$

$$\text{if } x_2 \text{ is } \begin{array}{|c|c|} \hline 0 & 10 \\ \hline \end{array} \quad \text{then } y = 0.91x_1 + 2.06x_2 + 2.3$$

performance index = 3.73.

The model 1-1 is found to be a stable state at the stage 1 since its performance index is the minimum of the two. In fact Fig. 9 seems more similar to Fig. 11 than to Fig. 12. At the next stage we fix the variable x_1 in the premises.

Stage 2: In this stage the space of inputs is further divided. In the model 2-1 (Fig. 13), which is the extended one of the model 1-1, the range of x_1 is divided into four subspaces, leaving that of x_2 undivided.



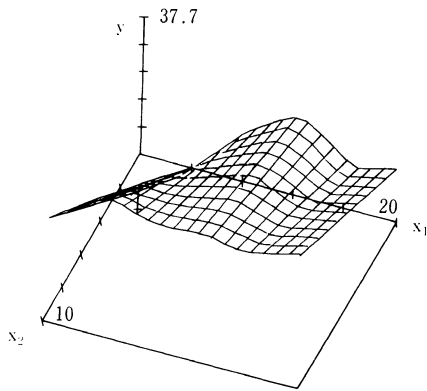


Fig. 13. Input-output relation of the model 2-1.

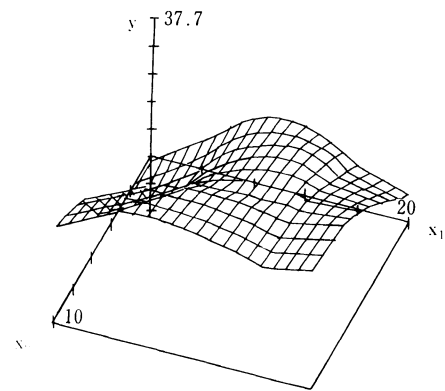
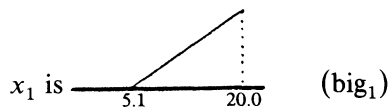
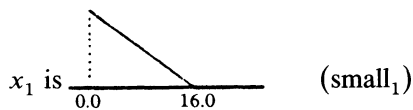
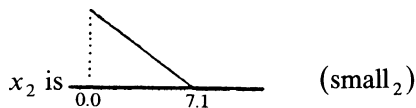


Fig. 14. Input-output relation of the model 2-2.

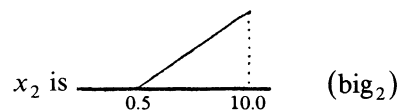
In the model 2-2 (Fig. 14), the ranges of x_1 and x_2 are newly divided in two fuzzy subspaces, respectively.



x_1 's range



x_2 's range

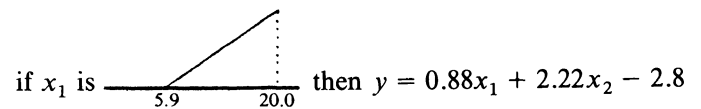
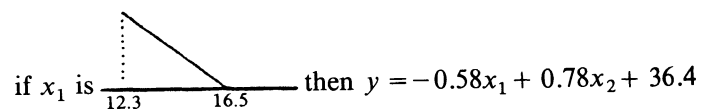
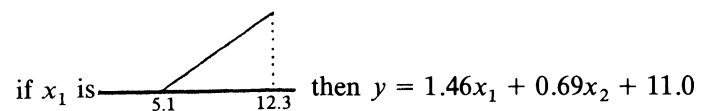
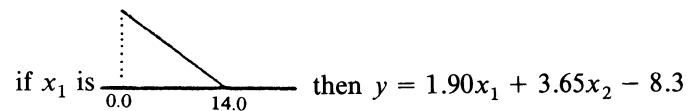


Notice that this partition of the range of x_1 is different from that of model 1-1. This is because the optimization of fuzzy sets concerned with x_1 is performed together with those concerned with x_2 .

The implications of the two models and their performance indices are obtained this time as follows.

Model 2-1:

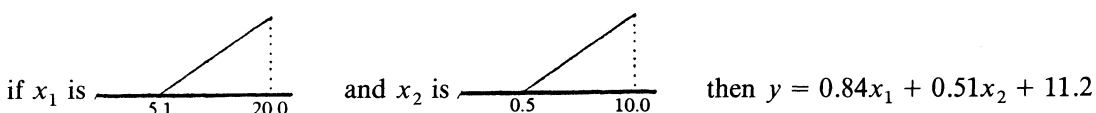
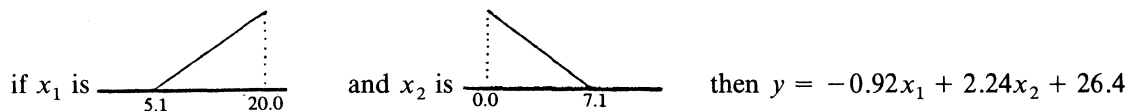
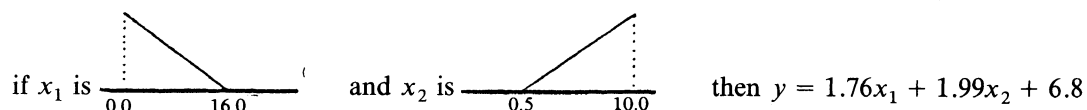
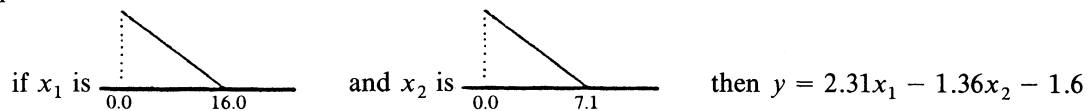
Implications



performance index = 1.91.

Model 2-2:

Implications

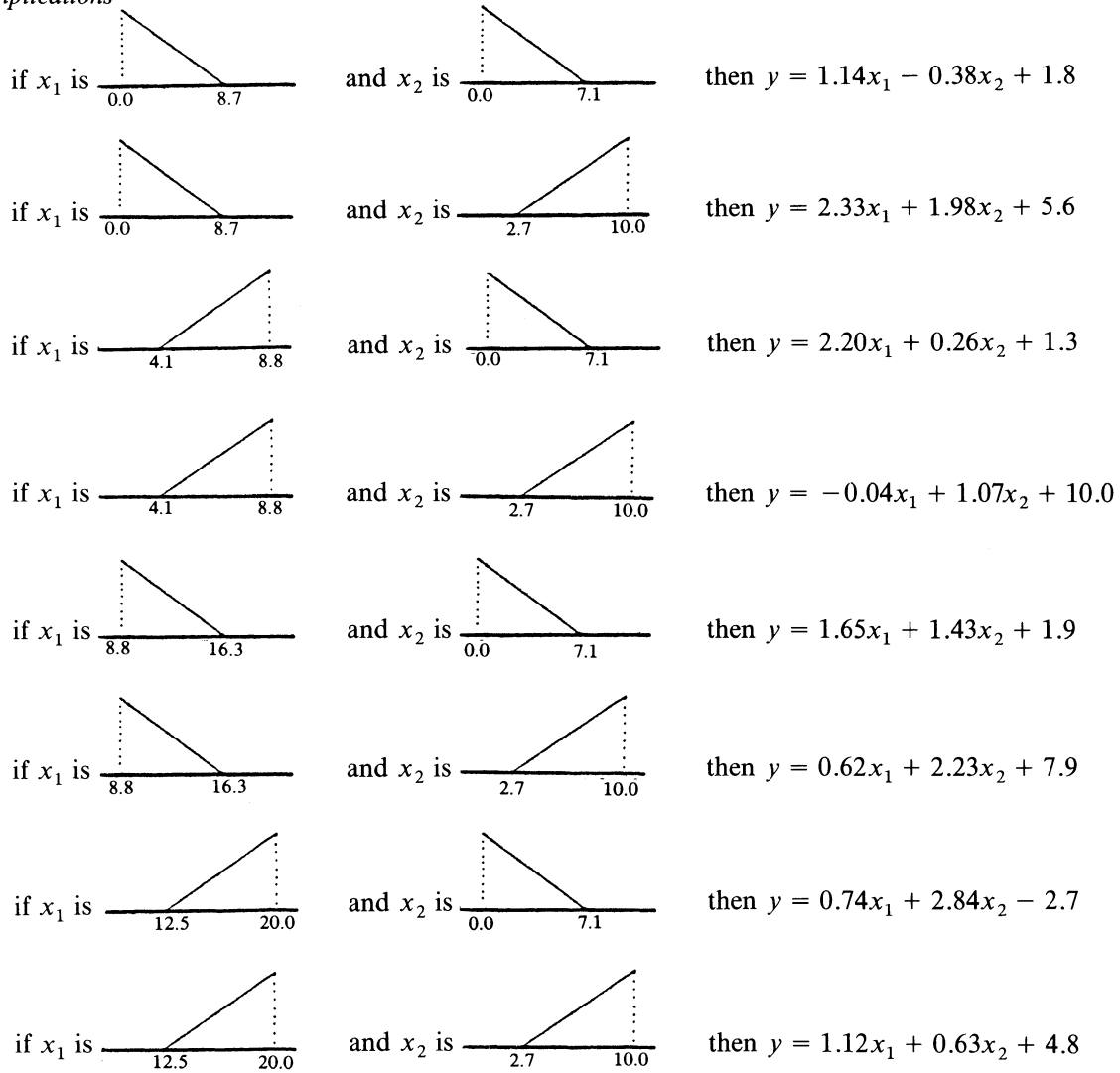


The model 2-2 is now found to be a stable state at stage 2.

Stage 3: We show the implications and the performance indices of two models (Figs. 15 and 16) extended from the model 2-2.

Model 3-1:

Implications

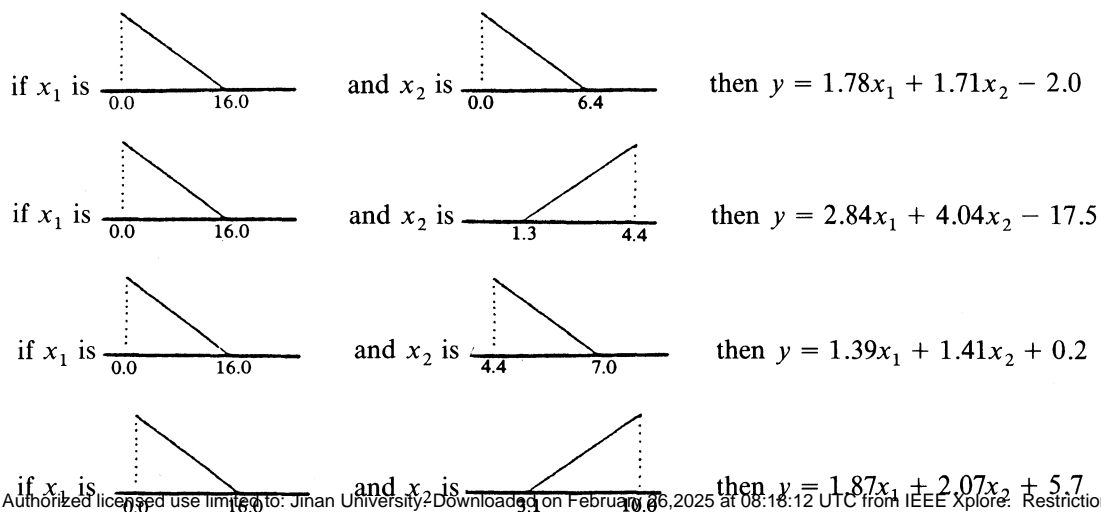


performance index = 1.08

where the partition of the range of x_1 is four and that of x_2 is two, as is seen.

Model 3-2:

Implications



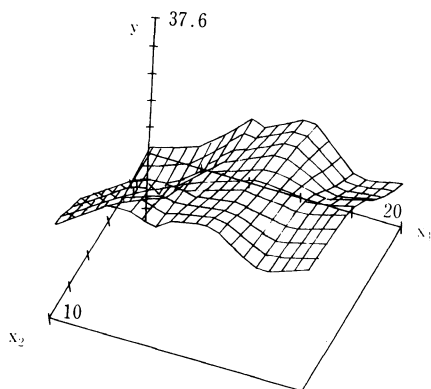


Fig. 15. Input-output relation of the model 3-1.

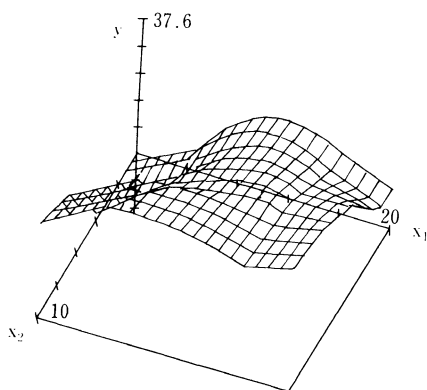


Fig. 16. Input-output relation of the model 3-2.

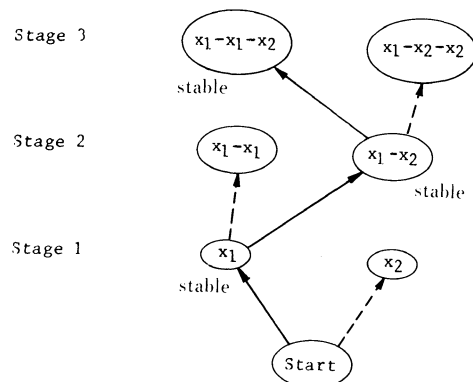
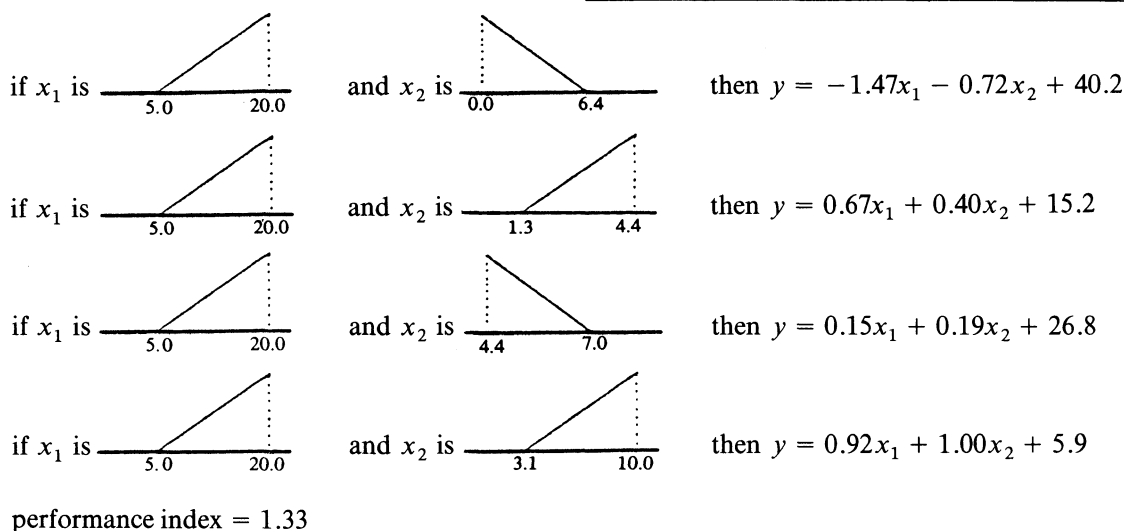


Fig. 17. Choice of the premise variables.

IV. APPLICATION TO FUZZY MODELING

This chapter shows two practical applications of the proposed method to real industrial processes. The first one is the fuzzy modeling of a human operator's control actions in a water cleaning process. The obtained model may be directly used in place of an operator to control the process.

The other one is the fuzzy modeling of a converter in a steel-making process. The relation between the input-output of the converter is so complex that an appropriate algebraic model has not been developed. The obtained fuzzy model is applied to the control of the converter, and the results are compared with the case when an operator controls it without a model.



At stage 3 the model 3-1 is found to be a model with the same structure as the original system, which is of course a stable state.

We can say that the presented method enables us to derive almost the same premise parameters as those of the original system. We can also recognize that Fig. 9 is almost the same as Fig. 15.

The choice of the premise variables has proceeded in this example as is shown in Fig. 17.

A. Fuzzy modeling of human operator's control actions

Water Cleaning Process: We shall now show an example where an operator's control actions are fuzzily modeled.

The control process is a water cleaning process for civil water supply as is illustrated in Fig. 18. In the process, turbid river water first comes into a mixing tank where chemical products called PAC and also chlorine are put and mixed in the water. Then the mixed water flows into a sedimentation tank where the turbid part of water is cohered with the aid of PAC and settled to the bottom.

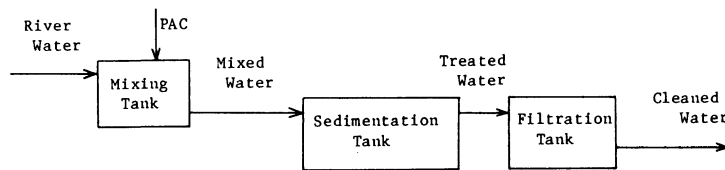


Fig. 18. Water cleaning process.

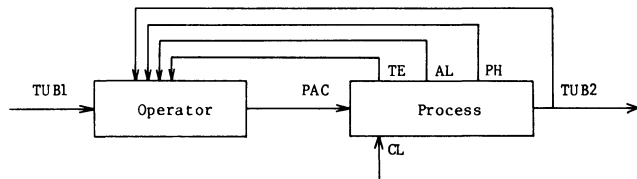


Fig. 19. Diagram of control process.

After sedimentation, which takes about 3–5 hours depending on the capacity of the tank, the treated water finally flows into a filtration tank producing clean water. Chlorine is added only for the sterilization of the water.

The main control problem of a human operator in this process is to determine the amount of PAC to be added so that the turbidity of the treated water is kept below a certain level. The optimal amount, not too little, nor too much, depends on the properties of the turbid water. The amount of PAC must be controlled also from an economical point of view.

The process is characterized by a lack of any physical model, significant variation of the turbidity of river water and the fact that turbidity itself is not clearly defined nor accurately measured. So an operator's experience is a key factor in this control process.

However, a number of variables influencing sedimentation process have been found so far that can be measured. Let us first list all the variables concerned.

TB1 Turbidity of the original water (ppm).

TB2 Turbidity of the treated water (ppm).

PAC Amount of PAC (ppm).

TE Temperature of water (°C).

PH PH.

AL Alkalinity.

CL Amount of chlorine (ppm).

For example, if TE is lower, then more PAC is necessary. Both PH and AL affect nonlinearly the necessary amount of PAC. The optimal PAC depends on these variables; the relation among them is not clear. There are some other variables influencing the process, e.g., plankton in the river water, which increases in springtime but cannot be measured at present.

In most water cleaning processes a statistical model has been built. However the models are not accurate. These cover only steady state, i.e., a small range of TB1. TB1 increases for example 100 times more when it rains. So an

TABLE II

TUB1	PH	TE	AL	PAC	TUB2
10.0	7.1	18.8	53.0	1300	1.0
17.0	7.0	18.6	50.0	1300	1.0
22.0	7.3	19.4	46.0	1400	2.0
50.0	7.1	19.5	40.0	1400	1.0
9.0	7.3	23.3	48.0	900	4.0
11.0	7.1	20.7	50.0	900	1.0
12.0	7.2	21.3	50.0	900	3.0
14.0	7.2	23.6	53.0	900	4.0
35.0	7.0	17.8	35.0	1200	1.0
20.0	7.0	16.6	40.0	1100	1.0
20.0	6.9	17.8	42.0	1100	1.0
18.0	7.1	17.3	40.0	1100	1.0
12.0	7.2	18.8	55.0	900	3.0
8.0	7.2	18.0	50.0	1000	1.5
11.0	7.1	19.2	49.0	1000	2.0
50.0	7.0	18.0	37.0	1200	1.5
35.0	7.0	17.7	42.0	1200	1.5
30.0	7.0	17.3	41.0	1100	1.5
16.0	7.1	19.3	42.0	1100	3.0

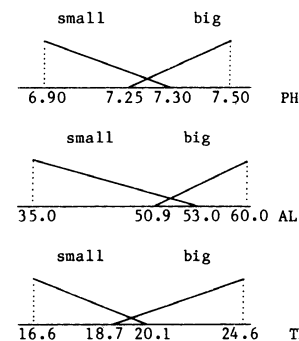


Fig. 20. Partition of the ranges of premise variables.

operator controls PAC taking into account of TB1, TE, PH, AL, and TB2. Now our process can be illustrated as in Fig. 19.

Derivation of Control Rules

We have a lot of operation data where all the variables are measured every hour for four months. That is, the number of data is 24 hours \times 30 days \times 4 months = 2880. Table II shows a part of these.

Among the data we have used about 600 for the identification taken in June and July. June in Japan is a rainy season and July is summer.

According to the identification algorithm discussed previously, eight control rules are derived that can be called a fuzzy model of operator's control as is stated, where a control rule is of the form

If (PH is *), (AL is *) and (TE is *)

then $PAC = p_0 + p_1 \cdot TB1 + p_2 \cdot TB2 + p_3 \cdot PH + p_4 \cdot AL + p_5 \cdot TE$

PH, AL, and TE are picked up as premise variables and their ranges divided into small and big, as shown in Fig. 20. The number of the control rules is thus $2^3 = 8$. Those are shown in Fig. 21.

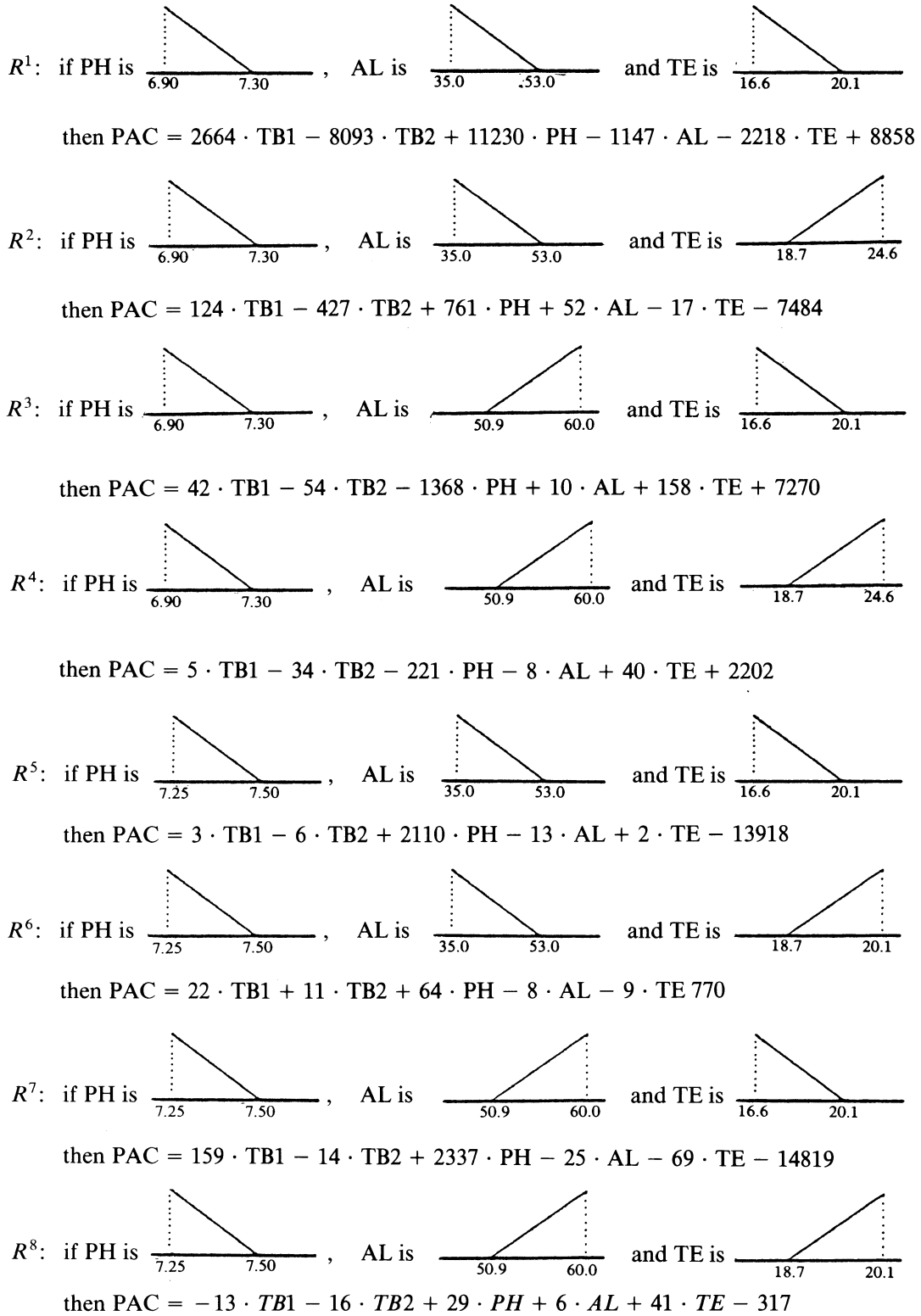


Fig. 21. Control rules.

Results of Fuzzy Control

The performance of the derived control rules is tested by using testing data. The results are shown in Table III as well as operator's control input and the results of a statistical model where we used 38 testing data. The statistical

model is represented in (23) that is usually used in a water cleaning process.

$$PAC = 9.11\sqrt{TB1} - 79.8 PH + 12.7CL + 1255.6.$$

(23)

TABLE III

Operator	Statistical Model	Fuzzy Model
1300	994.7	1308.6
1300	995.9	1027.4
1300	1119.6	1063.0
1400	1151.1	1386.2
1400	1409.4	1551.1
900	1066.4	923.9
900	1068.9	965.5
900	1012.3	875.4
1200	1286.8	1236.7
1200	1246.8	1172.6
1100	1151.4	1075.9
1100	1199.5	1115.1
1100	1159.4	1130.5
1000	985.7	934.1
1000	1009.3	973.8
1000	1038.2	984.6
1200	1398.3	1285.3
1200	1290.6	1160.8

It is seen that operator's control actions are well modeled in the form of fuzzy control rules. The average of the absolute differences between the results of the fuzzy model and the operator, and those between the results of a statistical model and the operator are, respectively,

fuzzy model	48.5
statistical model	128.0.

These results show the excellence of the fuzzy model.

B. Fuzzy Modeling of Converter in a Steel-Making Process and its Control

The Problem

The steel-making process consists of the following four steps.

- 1) Iron ore is melted in a blast furnace. The obtained molten iron called hot pig is removed by a torpedo car into a converter after desulfurization.
- 2) In a converter, scrap, iron ore, and burnt lime are first added to hot pig, and then decarbonization and dephosphor are performed by oxygen below. After that various alloys are added for adjusting the ingredients of produced steel.
- 3) Floating slag is taken away and the amount of ingredients is readjusted in a ladle refining process.
- 4) It is then cast and finally cut into appropriate figures.

Each step except the final half of the fourth step depends on a human operator's trained control because it is very difficult to build a process model.

A steel-making plant produces various kinds of steel according to its ingredients. Especially the manganese ratio in the products is required to be variously adjusted.

In this section we deal with the problem of determining the amount of manganese alloy to adjust the manganese ingredient of produced steel in step (2). This process is the most difficult to be controlled among the ingredients adjustments.

We now list all the variables possibly concerned with the process.

Mn1 Original ratio of manganese in input iron

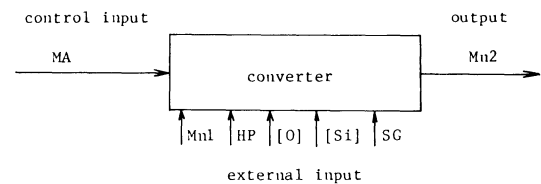


Fig. 22. Conversion process.

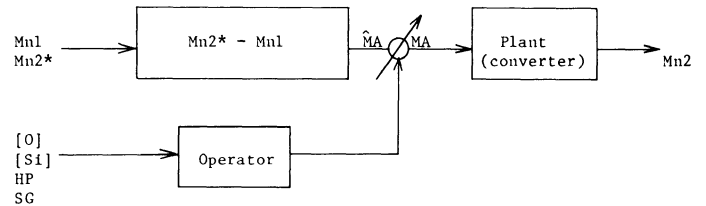


Fig. 23. Control of manganese ratio.

Mn2 Final ratio of manganese in produced steel.

Mn2* Required ratio of manganese in produced steel.

MA Ratio of manganese alloy put into input iron.

HP Hot pig ratio of input iron.

[O] Ratio of oxygen in input iron after oxygen blow.

[Si] Ratio of silicon of hot pig.

SG State of floating slag.

Here input iron consists of hot pig and scrap. Among those, SG is physically measured only after the conversion process is finished. But it can be evaluated by operator's observation before that. Fig. 22 shows the conversion process.

Manganese alloy is added into hot pig after the oxygen blow process to produce steel such that the percentage of manganese ingredient, Mn2, becomes a required value Mn2* and its ideal amount can be calculated by (24), based on physical analysis.

$$\hat{MA} = (Mn2^* - Mn1). \quad (24)$$

However, the actual ratio of manganese in the products is usually less than the physically estimated value from Mn1 and MA because of absorption by slag or other reasons due to, for example, various other ingredients in steel. So a human operator predicts Mn2 by referring to [O], [Si], HP, and observing the state of slag SG etc., and controls MA by correcting the guided value by (24) so that Mn2* is attained. Fig. 23 shows the present situation of manganese control.

The past trials to derive a mathematical model of the converter with respect to manganese control have not been successful where the inputs are the observable variables and the output is the manganese ratio of the produced steel.

For the purpose of control, we try to build a fuzzy model of the converter by finding some clues from an experienced operator's way of control. This approach has the following advantages.

- 1) His manner tells us how he recognizes the characteristics of the conversion process based on his experience. For example we can know important variables that should be put into premises of fuzzy implications.

TABLE IV

HP	ΔMn	SG	MA
93.90	1.11	0.30	14.00
94.00	14.52	-0.08	135.00
93.10	5.22	-0.05	54.00
93.60	5.36	-0.43	53.00
85.60	13.56	0.11	129.00
85.80	11.55	0.12	106.00
86.40	11.88	-0.06	113.00
93.10	13.85	0.16	119.00
88.50	12.27	-0.02	129.00
93.00	11.69	-0.03	127.00
93.00	9.86	-0.01	98.00
95.00	4.96	-0.13	46.00
94.70	2.19	0.01	25.00
85.00	1.89	-0.10	23.00
87.50	14.26	-0.09	121.00
90.50	11.52	-0.10	112.00
90.10	12.59	-0.06	116.00
90.20	5.67	-0.28	60.00

TABLE V

choice of premise variable		performance index	correlation coefficient
[Stage 1]	HP	5.20	0.98975
	MA*	4.87	0.99105
	SG	5.54	0.98840
[Stage 2]	MA - HP	4.17	0.99344
	MA - MA	4.23	0.99234
	MA - SG*	4.06	0.99378

* indicates a stable state at each stage

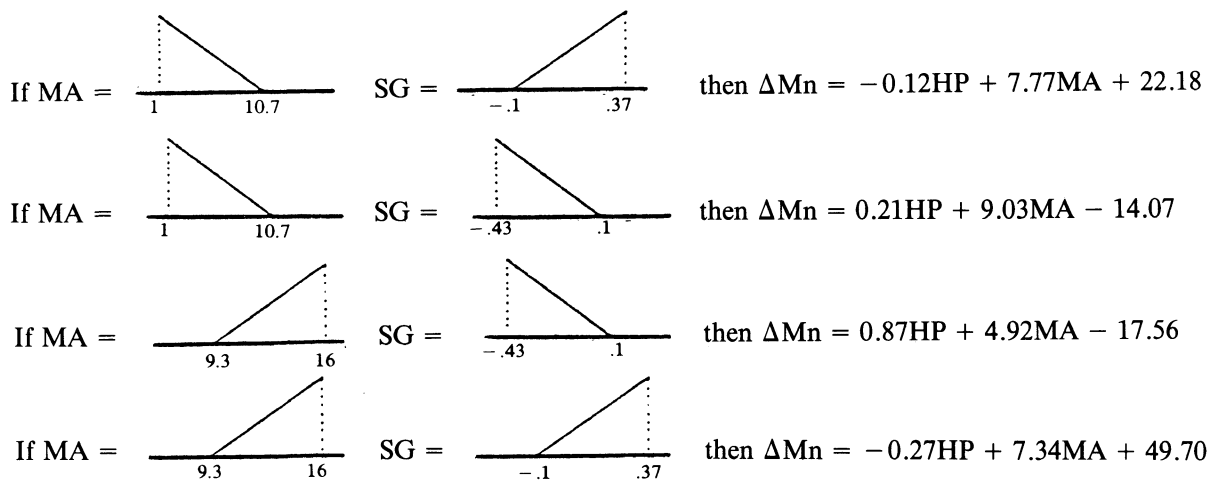


Fig. 24. Fuzzy model of converter.

2) We can even use input variables that only he can measure, for example, by just watching. Those variables are easily used as premise variables of our model. Since a model is of the form "if \dots then \dots ," the obtained model may be refined by his knowledge.

3) We can derive fuzzy control rules from the model, rather than from his control actions which may not be the best from a quantitative point of view. Needless to say, fuzzy control rules are easily understood qualitatively by him and we can adjust control rules also by his way of control. This is a very important point if an operator remains as a key essence in process control.

Modeling: We have taken 61 operation data from among the ones obtained in one month, and have used them for the identification of the conversion process. Further, we prepared 20 testing data different from the above identification data to check the validity of the obtained model. Table IV shows some of the data. The input and output variables of the converter model are as follows:

[input]

$$HP = \frac{\text{hot pig}}{\text{input iron}} = \frac{\text{hot pig}}{\text{hot pig} + \text{scrap}} \quad (\text{percent})$$

$$MA = \frac{\text{manganese alloy}}{\text{input iron}} \times 10^{-1} \quad (\text{percent})$$

SG = indication about softness of slag

[output]

$$\Delta Mn = Mn2 - Mn1 = \text{increment of manganese ratio.}$$

In this study, other variables [O] and [Si] are found not to seriously affect the process and so are deleted. For SG we conventionally use the measured values after the conversion is finished, which can be replaced by operator's observation. An experienced operator can measure SG rather qualitatively such as soft, medium, or hard. For this reason we put SG only into the premises for conditioning the input-output relation and do not use it in the consequences.

According to the proposed identification algorithm, each range of MA and SG is divided. The range of HP has remained undivided. Finally we have obtained four implications as are shown in Fig. 24.

The premise variables, the performance indices of the models, the correlation coefficients of the original output and models' output through the identification process are shown in Table V.

As is seen in the model, the space of each premise variable is divided only into two fuzzy subspaces. This is mainly because of the shortage of data. Notice that there

TABLE VI

Actual output	Fuzzy model	Statistical model
143.00	135.18	136.48
128.00	124.06	125.09
29.00	27.29	29.40
129.00	128.65	126.57
101.00	99.31	97.07
112.00	114.27	111.57
107.00	111.74	111.63
143.00	117.70	115.08
13.00	13.72	15.85
121.00	120.66	120.73
58.00	52.59	53.89
112.00	109.61	105.14
126.00	126.44	127.22
58.00	57.32	58.33
57.00	55.44	56.34
72.00	75.03	75.44
40.00	43.78	43.94
104.00	110.12	102.14
113.00	114.60	114.50
110.00	96.93	94.99

are five parameters in one implication: $4 \times 5 = 20$ altogether. On the other hand, the number of data is only 61.

Results of Fuzzy Model: Table VI shows the results of the fuzzy model, those of a statistical model and converter outputs, when the models are applied for the testing data.

The statistical model is represented in (25), whose parameters were obtained by linear regression using the identification data.

$$\Delta Mn = -0.24HP + 8.64MA + 45.60. \quad (25)$$

The performance indices of the results of the fuzzy model and the statistical model for 20 testing data are as follows:

fuzzy model 7.15

statistical model 7.77.

The results are better than those obtained by a statistical model.

It should be noted that the fuzzy partition of the state of slag, SG, derived from the data shows a good agreement with that by an operator: he usually recognizes SG according to a similar partition and uses this information in his control.

Control of Converter: We now try to control the converter by using its model. Given a desired output ΔMn^* , we can calculate a necessary input MA from a model. Here for simplicity we use this MA instead of designing a fuzzy controller.

The problem is how to compare the results of the model-based control with those of an operator, because we cannot make an experiment at present. Let us take as an index of control performance

$$AC = \text{average of } \frac{|\Delta Mn^* - \Delta Mn|}{\Delta Mn^*}$$

where

ΔMn^* desired output

ΔMn actual output.

As for an operator's control, this index is obtained from input-output data since, given ΔMn^* , he controls a converter. Let us denote it AC_{ope} . In case of a model-based control, we assume that the output ΔMn of a converter is a desired output. Then we get the optimal input MA^0 from a model, input MA^0 to a process and see its output $\Delta \tilde{Mn}$. This output $\Delta \tilde{Mn}$ can be estimated by taking into account of the accuracy of the model without experiments. So we can set $\Delta \tilde{Mn} = \Delta Mn \pm \epsilon$ where ϵ is the error of the model. Now we have

$$AC_{model} = \text{average of } \frac{|\Delta Mn - \Delta \tilde{Mn}|}{\Delta Mn}$$

= average of error of the model.

We obtain the following results:

$$AC_{model} = 4.7 \text{ percent}$$

$$AC_{ope} = 6.7 \text{ percent.}$$

From the results we can expect that the control based on the obtained fuzzy model gives us better results than the present control by the operator.

Apart from the above method to calculate the input MA, we can directly derive fuzzy control rules in this case from the data $(\Delta Mn, HP, SG) \rightarrow MA$. Those are shown in Fig. 25.

V. CONCLUSION

We have suggested a mathematical tool to fuzzily describe a system. It has a quite simple form, but it can represent highly nonlinear relations as has been shown in examples. An algorithm of identification has also been shown and two applications to industrial processes have been discussed. The applications as well as illustrative examples show that the proposed method is general and thus very useful. We can put the results of fuzzy measurements by man such as "temperature is high" into the premises of implications. Linear relations in the consequences enable us to easily deal with this mathematical tool, as we well know in linear systems theory.

However, to claim the validity of the method, more studies have to be performed. The system theoretic approach is especially important. For example we have minimal realization problems, decomposition problems, design problems of controller, etc. To solve these problems, it is required for us to deal with fuzzy system representation just as we do with a linear system.

In this paper, modeling of a human operator's control actions, rather than that of a process, has been mainly discussed. It is, however, possible to apply the method presented for the identification of a dynamical system. Modeling of a multilayer incineration furnace, a dynamic and distributed parameter system, is now under study.

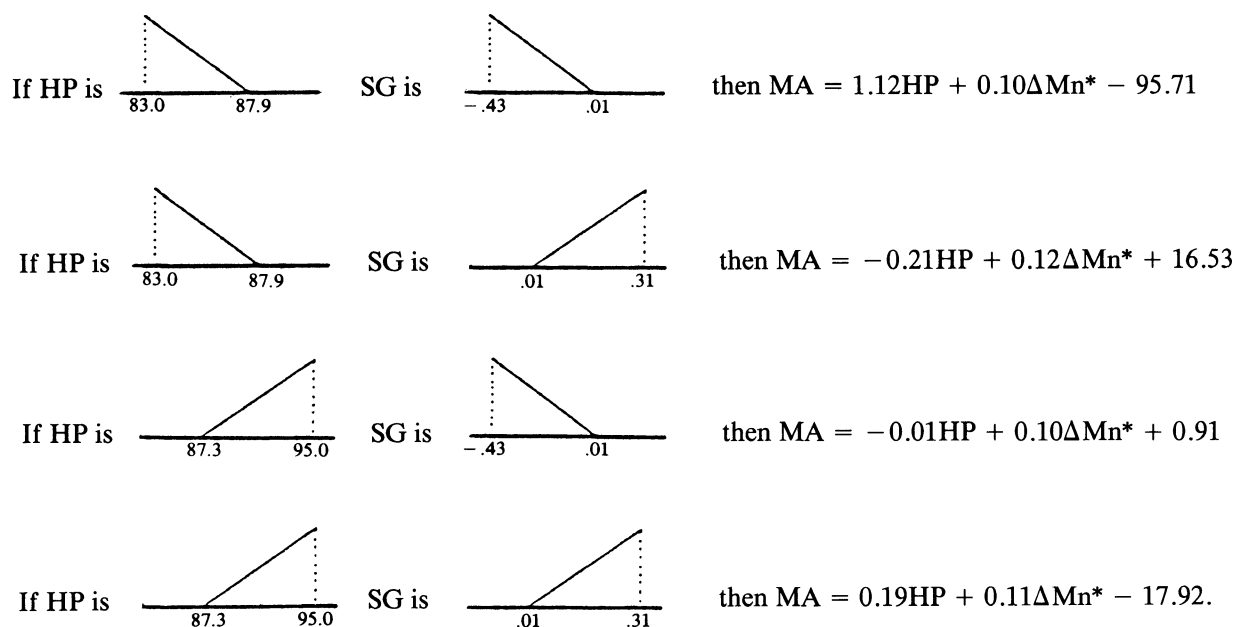


Fig. 25. Control rules.

REFERENCES

- [1] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Proc. IEEE*, vol. 121, no. 12, pp. 1585-1588, 1976.
- [2] R. M. Tong, M. B. Beck, and A. Latten, "Fuzzy control of the activated sludge wastewater treatment process," *Automatica*, vol. 16, pp. 659-701, 1980.
- [3] P. M. Larsen, "Industrial application of fuzzy logic control," *Int. J. Man-Machine Studies*, vol. 12, pp. 3-10, 1980.
- [4] R. M. Tong, "The construction and evaluation of fuzzy models," in *Advances to Fuzzy Set Theory and Applications*, M. M. Gupta Ed. New York: Plenum, 1979.
- [5] M. B. Gorzalkany, J. B. Kiszka, and M. S. Stachowicz, "Some problems of studying adequacy of fuzzy models," in *Fuzzy Set and Possibility Theory*, R. R. Yager, Ed. New York: Pergamon, 1982.
- [6] M. Sugeno and T. Takagi, "Multi-dimensional fuzzy reasoning," *Fuzzy Sets and Systems*, vol. 9, no. 2, 1983.