Appl. Math. and Comp. Sci., 1998, Vol.8, No.3, 633–645

GENERATING A SELF-ORGANIZING FUZZY CONTROLLER FROM HUMAN SKILL PERFORMANCE FOR MULTIVARIABLE SYSTEMS

ANNA LEKOVA*, D. BOYADJIEV*

The paper presents a learning concept for creating of a self-organizing fuzzy controller for multivariable systems by explicit use of human skill performance during the control of a complex technological process. When such processes cannot be entirely controlled automatically, a natural way to achieve flexible and adaptive control is a combination of human resources and information technologies. The main idea is the decomposition of a multivariable control system into several subsystems with two inputs and multiple outputs. The number of sub-systems corresponds to the number of all orthogonal projections of an N-dimensional input vector in the two-dimensional plane. Thus the method for generating fuzzy if-then rules from numerical data in the two-dimensional plane which is already available, can be used. Fuzzy rules with variable fuzzy regions are generated automatically. An aggregation operator for calculating the connectivity degree of membership functions which connects the whole set of the generated fuzzy rules in all decomposed subsystems is found.

1. Introduction

A fuzzy controller which is able to develop and improve fuzzy rules and its structure automatically as a result of monitoring the performance of the process so as to obtain a prespecified quality, is called a self-organizing fuzzy controller (SOC) (Shihuang, 1988). There are complex control systems in which a human controller is an essential part of control because the processes are so complicated that no mathematical models exist for them. Yet skilled human operators can control such systems quite successfully without having in mind any quantitative models. Several approaches have recently been proposed for automatically generating fuzzy if-then rules from numerical data (Takagi and Sugeno, 1985; Wang and Mendel, 1992). Self-learning methods have also been proposed for the evaluation of membership functions of fuzzy sets (Chaudhuri and Majumder, 1982; Lekova and Boyadjiev, 1996). Genetic algorithms (Ishibuchi *et al.*, 1995) have been employed for selecting fuzzy if-then rules in classification tasks. Unfortunately, most of the solutions can be implemented only for single-input-single-output (SISO) control systems. It is known that real control systems are multidimensional and the analysis and design procedures for such systems

* Institute of Control and System Research, Bulgarian Academy of Sciences, Bulgaria, Sofia 1113, Acad. G. Bontchev str., bl. 2, P.O. Box 79, e-mail: sylvia@acad.bg.

are very difficult. The process of generating fuzzy rules for multivariable systems has not been cleared yet. Usually such rules are obtained from human experts but knowledge acquisition is problem-dependent and in general this linguistic information is not sufficient. A survey of multivariable structures for fuzzy control systems is made in (Gupta *et al.*, 1986). The decisions are based on the decomposition of control rules through the intersection coefficients or the decomposition of a multivariable fuzzy system into a set of one-dimensional systems by multivariable fuzzy equations.

While various methods have been proposed for generating fuzzy rules with constant fuzzy regions, only a few approaches (Abe and Lan, 1995a; 1995b; Mikhailov *et al.*, 1996) have dealt with variable fuzzy regions. The fuzzy rules are composed recursively from activation hyperboxes (Abe and Lan, 1995b) which describe the input regions corresponding to given output intervals and inhibition hyperboxes where there is an overlapping of input regions for several output intervals. The overlapping among the output intervals is recursively resolved by defining additional activation and inhibition hyperboxes, until non-overlapped input regions are obtained. It is not necessary to preliminary divide the input range and to assign membership functions to each input variable. When there is no overlapping among the activation hyperboxes at a given level of resolution, the recursive process could not be resolved and a new method (Mikhailov *et al.*, 1996) for extraction of fuzzy rules from the set of input-output data has been proposed. There are no inhibition boxes, so the rules are simpler without the OR operator, which additionally simplifies the fuzzification procedure and allows a very fast reasoning.

The paper presents a learning concept for creating a self-organizing fuzzy controller for multivariable systems by explicit use of the human skill performance during the control of a complex technological process. The main idea is the decomposition of a multivariable control system into several two-inputs-multiple-output (TIMO) subsystems. The number of TIMO sub-systems corresponds to the number of all orthogonal projections of an N-dimensional input vector in the two-dimensional plane. Thus the method for generating fuzzy if-then rules from numerical data in the two-dimensional plane which is already available, can be used (Mikhailov *et al.*, 1996). It deals with fuzzy rules with variable fuzzy regions. Only the range of the output variables is divided into intervals and thus the input data corresponding to each output interval are grouped into appropriate areas. The fuzzy rules are composed recursively from activation hyperboxes (Mikhailov *et al.*, 1996) which describe the input regions corresponding to given output intervals. An aggregation operator for calculating the connectivity degree of membership functions for an input vector to all given output intervals is found. It connects the whole set of the fuzzy rules generated in all decomposed subsystems.

2. Extraction of Fuzzy Rules from Numerical Data in a Two-Dimensional System

Let us consider a control system with two inputs and one output. It could be decomposed into a TIMO subsystem. The universe of discourse of Y is divided into n output intervals Y_i, $i = 1, 2, \ldots, n$. By considering the set of all input vectors $x \in X_i$

which produce an output in the i-th output interval Y_i, $i = 1, 2, \dots, n$, starting from the value $(0, 0)$ of the input space and moving from the left to the right and from the bottom to the top, the activation rectangle with the maximum point concentration is produced in the first step, which contains only the input data from X_i. This rectangle is denoted by $A_i(1)$, where the values of its minimum and maximum border points are $v_{ik}(1)$ and $V_{ik}(1)$, respectively. Activation rectangles with smaller areas exist in the unclassified input space $B(1) = X - A(1)$ and, in the same way, new rectangles are formed in the second step, and so on. Since the activation rectangle $A_i(l)$ does not overlap any other rectangle $A_j(l)$, $j \neq i$, $j = 1, 2, \dots, n$, fuzzy rules $R_i(l)$ from the l-th step and for the i-th output interval are generated forming the fuzzy rule base

$$\text{IF} \quad x \quad \text{is in} \quad A_i(l) \quad \text{THEN} \quad y \quad \text{is in} \quad Y_i$$

where

$$A_i(l) = \{x | v_{ik}(l) \leq x_k \leq V_{ik}(l),\ k = 1, 2\} \tag{1}$$

The coressponding geometrical representation in the two-dimensional plane is shown in Fig. 1(a), where the SOC is fully trained, i.e. the human skills are involved entirely. The control actions are $Y1$, $Y2$, $Y3$ and $Y4$. Each input variable has different influence on the control actions and this fact is expressed by the priority vector $DI = \langle 0.2\ 0.1 \rangle$. The deviated variables in positive direction $(\uparrow)$ are more important than those in negative one $(\downarrow)$. For this reason the activation rectangles in Fig. 1(b) are extended and translated in a certain direction when compared with those in Fig. 1(a). The surfaces which denote the control actions in a geometrical representation and correspond to the negative deviations or to smaller priorities, occupy a narrower space. The sensitivity parameter γ defines the slope of the trapezoid or the triangle membership function shape and could be different on the two axes in all the two-dimensional planes. These parameters describe the generalization region of the corresponding fuzzy rule and the overlapping of the fuzzy regions.

The learning concept aims at obtaining the geometrical representation shown in Fig. 1(b), which is unknown at the beginning, i.e. the human skills are not involved entirely. The design of the SOC allows us to improve iteratively the system as a modification based upon the results of the training approach. The geometrical representation for the available rules after the initial training is shown in Fig. 1(c). The method for extraction of fuzzy rules from numerical data of two dimensional systems is proposed in detail in (Mikhailov *et al.*, 1996).

3. Extraction of Fuzzy Rules from Numerical Data in a Multi-dimensional System

Let us suppose that a set of input-output data pairs

$$\big(x_1^{(1)}, \dots, x_m^{(1)}, y_1^{(1)}, \dots, y_n^{(1)}\big),\ \big(x_1^{(2)}, \dots, x_m^{(2)}, y_1^{(2)}, \dots, y_n^{(2)}\big), \dots$$

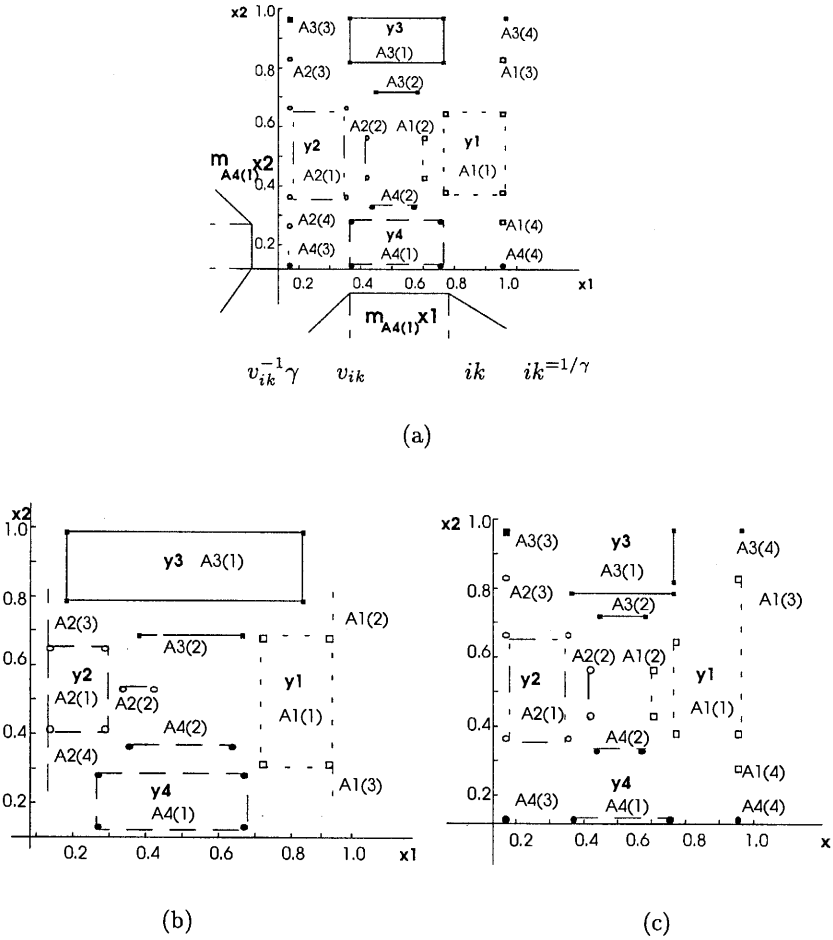


Fig. 1. Separation of the input space: a fully trained SOC (a), a fully trained SOC with priorities (b), and an initially trained SOC (c).

is given where $x_1^j, \ldots, x_m^j$ are the components of the m-dimensional input vector x and y_i, $i = 1, \ldots, n$ are the outputs. When $m > 4$, even if the output is single, it is difficult to create an m-dimensional fuzzy map. Similarly, the method presented in Section 2 could not be applied by analogy because of the difficult geometric representation in the m-dimensional space. For this reason, the MIMO control system is decomposed into several subsystems with two inputs and multiple outputs whose number $N_{prj}(k)$ corresponds to all the orthogonal projections in the two-dimensional plane:

$$N_{prj}(k) = \frac{1}{2}k(k-1), \quad k = 1, \ldots, m \tag{2}$$

When the input x_k is inside the rectangle $A_i(l)p$, the degree of the membership of x_k for the rule $R_i(l)p$, denoted by $m_{A_i(l)p}(x_k)$, is always equal to 1 and decreases when x_k moves away from the rectangle. This means that the membership function $m_{A_i(l)p}(x_k)$ could be trapezoidal, as shown in Fig. 1(a) The membership functions in all the orthogonal projections for each output interval could be generated as

$$\begin{aligned} m_{A_i(l)p}(x_k) &= \Big[l - \max\Big(0, \min\big(1, \gamma(v_{ikp}(l) - x_k)\big)\Big)\Big] \\ &\quad \times \Big[l - \max\Big(0, \min\big(1, \gamma(x_k - x_{ikp}(l))\big)\Big)\Big], \\ &\quad k = 1, \dots, m, \quad p = 1, \dots, N_{prj}(k), \quad l = 1, 2, \dots \end{aligned} \tag{3}$$

For separated points $(v_{ikp}(l) = x_k = V_{ikp}(1))$ or rectangles with zero height or length, the above membership function is triangular.

The degree of the membership for a given input vector x is 1 when its two components are inside the rectangle or on its border. Hence the degree of the membership function of the fuzzy rule for x has to be calculated as the minimum of the values for all the membership functions in all the orthogonal projections for all the i-th output intervals:

$$d_{R_i(l)p}(x) = \min\big(m_{A_i(l)p}(x_k)\big), \quad p = 1, \dots, N_{prj}(k), \quad k = 1, \dots, m \tag{4}$$

Thus the degree of the membership of x in the i-th output interval for the whole set of the generated fuzzy rules, giving an output in Y_i is as follows:

$$d_{ip}(x) = \max\big(d_{R_i(l)p}(x), \quad p = 1, \dots, N_{prj}(k), \quad k = 1, \dots, m \tag{5}$$

The connectivity degree of the membership of x to the i-th output interval for the whole set of generated fuzzy rules in all the orthogonal projections, i.e. $\mathrm{cd}_i(x)$, has to be found. N factors participate here which correspond to the degrees of the membership of x to the i-th output interval for the whole set of the generated fuzzy rules, see eqn. (5). All the variables have to be in certain intervals, which allows using the records of human-operator control actions:

$$\mathrm{cd}_i(x) = \mathrm{aggr\_oprt}\big(d_{ip}(x)\big), \quad p = 1, \dots, N_{prj}(k), \quad k = 1, \dots, m \tag{6}$$

The aggregation operator *aggr_ oprt* to handle this calculation from its own solutions in all TIMO subsystems could be: min, max, an average value, an algebraic product or a new one. The crisp value of the output variable for a given input vector x is obtained by defuzzification with the maximum type for a discrete system and defuzzification with bell-shaped membership functions or the centre of gravity for a continuous system (Mikhailov *et al.*, 1996).

4. Illustrative Example

To illustrate the procedure of fuzzy rule extraction, the synthesis of an SOC for control of a chemical technological process is considered (Boyadjiev *et al.*, 1995). The process is simulated in different conditions: first—with four inputs and one output, and second—with nine inputs and one output. In general, the Human Operator has the information that the control actions must reduce the deviations of the controlled variable from their normal values. The control action is a piece of imprecise knowledge, e.g. "If the temperature is high or medium_high deviated you might regulate a little", where "high or ... " is fuzzy, rather than precise, not clear cut and, which is more important, context-dependent. A self-organizing system has to scan the process variables and the Human Operator's control actions in appropriate form. The values are transformed in compressed form and the system must be trained to accomplish at least a part of the operator's skilled performance. This process must be repeated to iteratively improve the system.

Let us consider a multivariable control system with four inputs and one output ($m = 4$ and $n = 1$). The output is divided into eight output intervals: $Y1 = 1 \downarrow$, $Y2 = 1 \uparrow, \ldots, Y7 = 4 \downarrow$, $Y8 = 4 \uparrow$. The input space could be decomposed into six TIMO subsystems, see eqn. (2), and the geometrical representations in all the orthogonal projections are shown in Fig. 2. There are hyperboxes, lines and points for eight output intervals in each two-dimensional plane. For graphical simplification, geometrical representations for four output intervals are shown: $Y1 = 1 \downarrow$, $Y2 = 1 \uparrow$, $Y5 = 3 \downarrow$ and $Y6 = 3 \uparrow$. The priorities are as follows:

1. a more deviated variable has a maximum priority;
2. the deviated variables in a positive direction are more important than those in a negative one;
3. the third variable has a higher priority than the first, second and fourth ones.

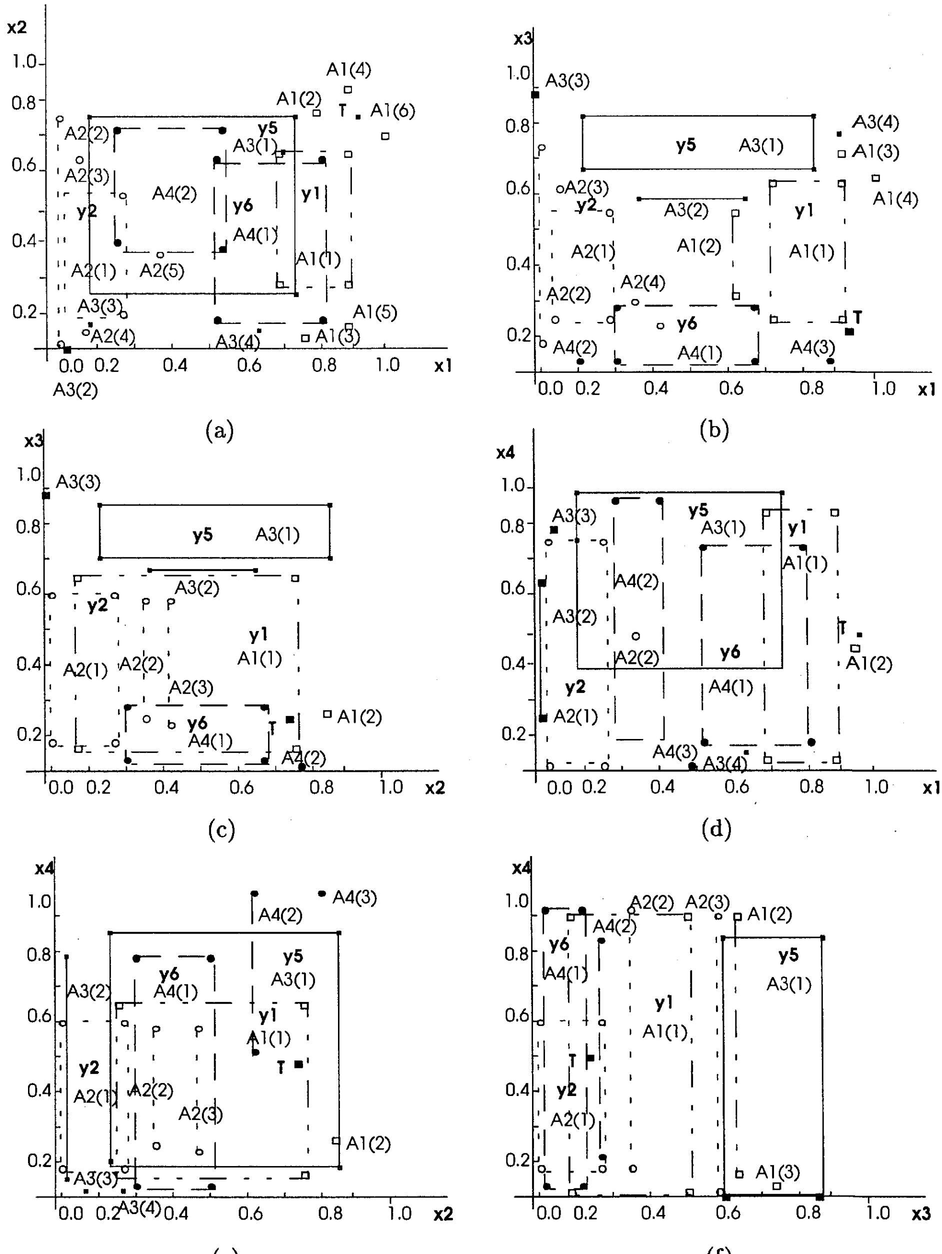
Hence the priority vector DI is $\langle 0.3, 0.2, 0.4, 0.1\rangle$.

Let us suppose that an SOC is not fully trained and take the following test state in the normalized form $T = (0.95, 0.75, 0.25, 0.5)$. After the deffuzification a right control action could be found.

The membership functions (3) are formed for all the rules. For the first activation rectangle (the first step) in Fig. 2(a) the membership functions are as follows:

$$m_{A1(1)1}(x_1) = \begin{cases} 1 & \text{for } 0.7 \le x_1 \le 0.9 \\ 1 - \max\left(0, \min\left(1, \gamma(0.7 - x_1)\right)\right) & \text{for } x_1 \le 0.7 \\ 1 - \max\left(0, \min\left(1, \gamma(x_1 - 0.9)\right)\right) & \text{for } x_1 \ge 0.9 \end{cases}$$

$$m_{A1(1)1}(x_2) = \begin{cases} 1 & \text{for } 0.3 \le x_2 \le 0.65 \\ 1 - \max\left(0, \min\left(1, \gamma(0.3 - x_2)\right)\right) & \text{for } x_2 \le 0.3 \\ 1 - \max\left(0, \min\left(1, \gamma(x_2 - 0.65)\right)\right) & \text{for } x_2 \ge 0.65 \end{cases}$$



□ – $y1$ (A1) – data in the first output interval (1↓);

○ – $y2$ (A2) – data in the second output interval (1↑);

■ – $y5$ (A3) – data in the fifth output interval (3↓);

● – $y6$ (A4) – data in the sixth output interval (3↑).

Fig. 2. Input space decomposition into orthogonal projections.

From (4) the degrees of the membership of the fuzzy rules for test points in all the orthogonal projections could be obtained:

- from the first orthogonal projection (Fig. 2(a))

$$d_{R1(1)1}(0.95,\ 0.75) = \min(0.95,\ 0.85) = 0.85$$
$$d_{R1(2)1}(0.95,\ 0.75) = \min(0.8,\ 1.0) = 0.8$$
$$d_{R1(3)1}(0.95,\ 0.75) = \min(0.8,\ 0.25) = 0.25$$
$$d_{R1(4)1}(0.95,\ 0.75) = \min(0.95,\ 0.9) = 0.9$$
$$d_{R1(5)1}(0.95,\ 0.75) = \min(1.0,\ 0.3) = 0.3$$
$$d_{R1(6)1}(0.95,\ 0.75) = \min(0.9,\ 0.9) = 0.9$$

$$d_{R2(1)1}(0.95,\ 0.75) = \min(0.35,\ 0.75) = 0.35$$
$$d_{R2(2)1}(0.95,\ 0.75) = \min(0.05,\ 1.0) = 0.05$$
$$d_{R2(3)1}(0.95,\ 0.75) = \min(0.15,\ 0.95) = 0.15$$
$$d_{R2(4)1}(0.95,\ 0.75) = \min(0.15,\ 0.15) = 0.15$$
$$d_{R2(5)1}(0.95,\ 0.75) = \min(0.45,\ 0.6) = 0.45$$

$$d_{R3(1)1}(0.95,\ 0.75) = \min(0.8,\ 1.0) = 0.8$$
$$d_{R3(2)1}(0.95,\ 0.75) = \min(0.0,\ 0.25) = 0.25$$
$$d_{R3(3)1}(0.95,\ 0.75) = \min(0.1,\ 0.3) = 0.1$$
$$d_{R3(4)1}(0.95,\ 0.75) = \min(0.7,\ 0.3) = 0.3$$

$$d_{R4(1)1}(0.95,\ 0.75) = \min(0.85,\ 0.85) = 0.85$$
$$d_{R4(2)1}(0.95,\ 0.75) = \min(0.6,\ 0.95) = 0.6$$

- from the second orthogonal projection (Fig. 2(b))

$$d_{R1(1)2}(0.95,\ 0.25) = \min(1.0,\ 0.95) = 0.95, \quad \cdots$$
$$d_{R1(2)2}(0.95,\ 0.25) = \min(0.3,\ 0.9) = 0.3, \quad \cdots$$
$$d_{R1(3)2}(0.95,\ 0.25) = \min(1.0,\ 0.5) = 0.5, \quad \cdots$$
$$d_{R1(4)2}(0.95,\ 0.25) = \min(0.9,\ 0.6) = 0.6, \quad \cdots$$

Using eqn. (5), we have

$$d_{11} = \max(0.85,\ 0.8,\ 0.25,\ 0.9,\ 0.3,\ 0.9) = 0.9, \quad d_{51} = \max(0.8,\ 0.0,\ 0.1,\ 0.3) = 0.8$$
$$d_{21} = \max(0.35,\ 0.05,\ 0.15,\ 0.15,\ 0.45) = 0.45, \quad d_{61} = \max(0.85,\ 0.6) = 0.85$$

$$
\begin{array}{lllll}
d_{12} = 0.95, & d_{13} = 1.0, & d_{14} = 0.95, & d_{15} = 1.0, & d_{16} = 1.0 \\
d_{22} = 0.45, & d_{23} = 0.75, & d_{24} = 0.4, & d_{25} = 1.0, & d_{26} = 1.0 \\
d_{52} = 0.65, & d_{53} = 0.6, & d_{44} = 0.8, & d_{55} = 1.0, & d_{56} = 0.6 \\
d_{62} = 0.85, & d_{63} = 0.95, & d_{64} = 0.85, & d_{65} = 0.9, & d_{66} = 1.0
\end{array}
$$

Compared to crisp logic implication operations, a number of fuzzy operation methods such a t-norm and s-norm have been introduced. Even under the same conditions (rules, etc.), when the operation method is changed, the inference result will be somewhat different. In other words, it is necessary to support as many operation methods as possible, even to evaluate which fuzzy operation method is best suited to the target system. Consequently, in this system it has been made possible to choose between two representative operation methods with two aggregation operators: the average value and the algebraic product.

By using eqn. (6) and an aggregation operator (the algebraic product), we obtained that the degrees of the membership of x to the $Y1 = 1 \downarrow$, $Y2 = 1 \uparrow$, $Y5 = 3 \downarrow$ and $Y6 = 3 \uparrow$ output intervals from the whole set of the generated fuzzy rules are as follows:

$$\mathrm{cd}_1(0.95, 0.75, 0.25, 0.5) = 0.9 * 0.95 * 1.0 * 0.95 * 1.0 * 1.0 = 0.81$$

$$\mathrm{cd}_2(0.95, 0.75, 0.25, 0.5) = 0.45 * 0.45 * 0.75 * 0.4 * 1.0 * 1.0 = 0.06$$

$$\mathrm{cd}_5(0.95, 0.75, 0.25, 0.5) = 0.8 * 0.65 * 0.6 * 0.8 * 1.0 * 0.6 = 0.15$$

$$\mathrm{cd}_6(0.95, 0.75, 0.25, 0.5) = 0.85 * 0.85 * 0.95 * 0.85 * 0.9 * 1.0 = 0.53$$

As can be seen, after the maximum-type deffuzification, the test state is classified into the first control interval. The second alternative is $Y6$. If the Human Mental Model for the control actions were used, the same result could be obtained: $x1$ is the maximum deviated variable in a positive direction.

5. Implementations and Results

The control problem for a chemical technological process is described with nine inputs and one output divided into eighteen output intervals (the control actions): $Y1 = 1 \downarrow$, $Y2 = 1 \uparrow$, $Y3 = 2 \downarrow$, $Y4 = 2 \uparrow$, $Y5 = 3 \downarrow$, $Y6 = 3 \uparrow$, $\dots$, $Y17 = 9 \downarrow$, $Y18 = 9 \uparrow$. For the present technological process DI is $\langle 0.5, 0.4, 0.9, 0.3, 0.6, 0.2, 0.7, 0.1, 0.8 \rangle$. The design of an SOC allows us to iteratively improve the system as a modification based upon the human skills involved after each control action. The test patterns are randomly chosen numbers in the interval $(-1 \div +1)$, which corresponds to control actions which deviate the controlled variable from its normal values. According to 1000 test patterns in the present example, the SOC is approximately fully trained and makes a good generalization, despite the fact that a geometrical representation for a fully trained system is not yet achieved.

Simulation results obtained for two kinds of aggregation operators are summarized in Tables 1 and 2 and are shown in Figs. 3. and 4. The learning abilities of the SOC are tested in the following way: the system learns each 100 training patterns and after that it is tested by means of 100 unknown patterns. The number of classified states (the classification ability) and the best distance between the connectivity degrees of the membership of x to the i-th output interval are explored. The first, second and third alternative control actions are proposed. Taking into account the results obtained using the different aggregation operators, we conclude that the most appropriate one is the algebraic product. The corresponding results for $\mathrm{cd}_i(x)$ are given in Table 3. For the aggregation operator (the average value), all the values are in the range $(0.7 \div 0.95)$ and after the sixth trial they fall into the interval $(0.960 \div 0.996)$. All the three alternatives have virtually equal values for $\mathrm{cd}_i(x)$.

Table 1. The classification ability of the SOC under consideration for the aggregation operator being the average value.

% Test	First pred.	Second pred.	Third pred.
1	21	32	47
2	30	55	67
3	40	65	76
4	51	69	82
5	51	72	84
6	65	79	90
7	69	84	94
8	71	87	95
9	74	90	95
10	80	88	94

Table 2. The classification ability of the SOC under consideration for the aggregation operator being the algebraic product.

% Test	First pred.	Second pred.	Third pred.
1	20	30	48
2	34	56	69
3	40	64	76
4	50	69	82
5	55	72	83
6	65	79	90
7	69	84	94
8	71	88	95
9	75	89	94
10	78	88	94

A comparison of these results with similar ones from a self-organizing neural-network system is also drawn. The classification ability after each series of 100 patterns is calculated. These computer simulations show that a good generalization is obtained after 1000 test patterns and both the methods have a high classification ability. Further, the quality of the test patterns produces a significant effect on the classification ability. Because of the transparent nature of the fuzzy system, the method described in the present work allows us to accelerate the learning process. The results for $\mathrm{cd}_i(x)$ obtained in each test make it possible to analyse to what extent the SOC is trained and, in this way, to plan the training. If they are in the interval $(0.7 \div 0.95)$, the learning process can be stopped. From Table 3 it can be seen that even if the interval is $(0.4 \div 0.6)$, the generalization is good. This is obtained in the fourth test trial.

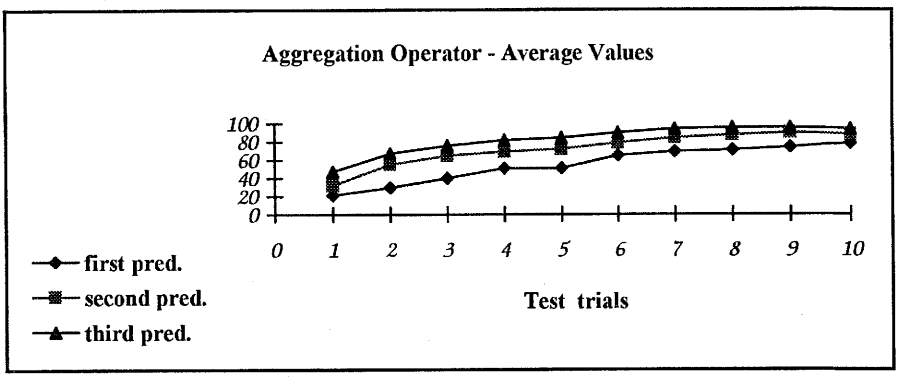


Fig. 3. Graphical representation of Table 1.

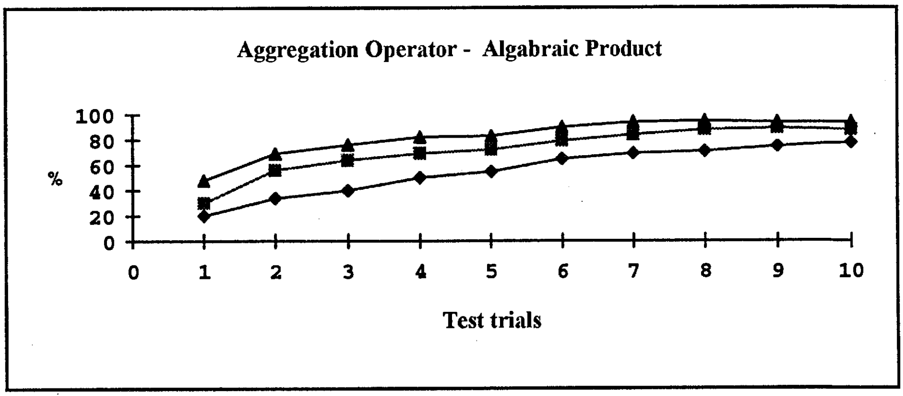


Fig. 4. Graphical representation of Table 2.

6. Conclusions

The paper presents a learning concept to create a self-organizing fuzzy controller by an explicit use of the human skill performance during the control of a complex technological process. The concept allows earlier and permanent model testing as the learned samples are changing in the process of control. The method could be applied for generation of fuzzy rules in a multivariable system where the human assessment and uncertainty of the control for a technological process are highly complicated and difficult to express. The SOC permits the analysis of the convergence of the learning process through the connectivity degree of the membership of x to the i-th output interval for the whole set of the generated fuzzy rules. Thus the training could be planned and the learning process could be accelerated because of the transparent nature of the fuzzy system.

Table 3. The range for the connectivity degree of the membership of x to the i-th output interval, $\mathrm{cd}_i(x)$.

Trials	min value	max value	Main interval
1 prd1	0.00	0.415	
prd2	0.00	0.103	0.050–0.220
prd3	0.00	0.058	
2	0.004	0.864	
	0.004	0.220	0.200–0.400
	0.002	0.130	
3	0.006	0.864	
	0.046	0.294	0.300–0.550
	0.003	0.150	
4	0.009	0.864	
	0.006	0.511	0.390–0.600
	0.014	0.280	
5	0.038	0.864	
	0.011	0.550	0.410–0.615
	0.127	0.324	
6	0.058	0.864	
	0.035	0.550	0.500–0.650
	0.015	0.280	
7	0.024	0.920	
	0.035	0.522	0.550–0.690
	0.035	0.427	
8	0.061	0.920	
	0.185	0.530	0.550–0.700
	0.052	0.397	
9	0.061	0.920	
	0.185	0.523	0.600–0.750
	0.060	0.530	
10	0.071	0.920	
	0.082	0.566	0.600–0.850
	0.287	0.513	

References

Abe Sh. and Lan M. (1995a): *A method for fuzzy rules extraction directly from numerical data and its application to pattern classification.* — IEEE Trans. Fuzzy Syst., Vol.3, pp.18–28.

Abe Sh. and Lan M. (1995b): *Fuzzy rules extraction directly from numerical data for function approximation.* — IEEE Trans. Syst. Man Cybern., Vol.25, No.1, pp.119–129.

Boyadjiev D., Ivanova P. and Lekova A. (1995): *Skill-based human-machine control of complex technological processes.* — Proc. 5th IFAC Symp. *Automated Systems Based on Human Skill*, Berlin, Germany, pp.15–19.

Chaudhuri D. and Majumder D. (1982): *On membership evaluation in fuzzy sets*, In: Approximate Reasoning in Decision Analysis (M.M. Gupta and E. Sanchez, Eds.). — Amsterdam: North-Holland, pp.3–11.

Gupta M., Kiszka J. and Trojan G. (1986): *Multivariable structure of fuzzy control systems.* — IEEE Trans. Syst. Man Cybern., Vol.16, No.5, pp.638–655.

Ishibuchi H., Nozaki K., Yamamoto N. and Tanaka H. (1995): *Selecting fuzzy if-then rules for classification problems using genetic algorithms.* — IEEE Trans. Fuzzy Syst., Vol.3, No.3, pp.260–270.

Lekova A. and Boyadjiev D. (1996): *Self-organizing fuzzy controller based on human skill performance.* — SAER, Varna, Bulgaria, pp.72–76.

Mikhailov L., Lekova A. and Nabout A. (1996): *Method for fuzzy rules extraction from numerical data.* — Complex Systems, No.1, pp.1–12.

Shihuang Sh. (1988): *Fuzzy self organizing controller and its application for dynamic processes.* — Fuzzy Sets Syst., No.26, pp.1678–1683.

Takagi M. and Sugeno M. (1985): *Fuzzy identification of systems and its applications to modelling and control.* — IEEE Trans. Syst. Man Cybern., Vol.15, No.1, pp.116–132.

Wang J. and Mendel J. (1992): *Generating fuzzy rules by learning from examples.* — Trans. Syst. Man and Cybern., Vol.22, No.6, pp.1414–1427.

Received: 19 February 1997
Revised: 4 November 1997
Re-revised: 24 February 1998