Fuzzy Sets and Systems 56 (1993) 253–271
North-Holland

A fuzzy controller for construction activities

M.H.M. Hassan and B.M. Ayyub
Civil Engineering Department, University of Maryland, College Park, MD 20742, USA

Received September 1992
Revised November 1992

Abstract: Construction activities are highly uncertain systems. They do not lend themselves to the application of traditional control theory because of their highly subjective nature. A fuzzy-based self-learning control strategy is utilized in this paper. The system comprises a rule-base, a self-learning unit and a conflict resolution unit. A new three dimensional implication function was developed and integrated in the system to model the individual rules. A condition assessment procedure is also integrated in the system in order to evaluate the output levels given a set of input states. The condition assessment procedure acts as a transfer function in traditional control theory.

Keywords: Activity; assessment; construction; control; fuzzy; likelihood; probability; process; reliability; safety; system; uncertainty.

1. Introduction

Fuzzy control is considered the best control strategy for ill-defined systems. Construction activities are highly complex and uncertain systems. They involve a large number of subjectively defined variables and descriptors and they cannot be modelled mathematically. Thus, a fuzzy control strategy provides the necessary means for controlling such systems. In this paper, a fuzzy controller for construction activities is developed. The proposed controller monitors one or more attributes of interest real-time. The control strategy utilizes a rule-base, a self-learning unit and a conflict resolution unit. The rule-base comprises a collection of rules that summarizes the experiences and knowledge of human controllers. The self-learning unit is utilized in updating and expanding the rule-base in order to accommodate newly acquired experiences and information. The self-learning algorithm serves a very essential role becasue of the difficulty of developing a complete rule-base that handles all potential situations. The conflict resolution unit is utilized when several attributes are being monitored. In this case, parallel single-attribute controllers are set-up each of which results in a single control action. For two or more of these controllers, if the same critical control variable is selected while the control actions are not identical, a conflict arises. The conflict resolution unit should be triggered in such a case in order to define a compromise solution that satisfies all control criteria as close as possible.

Fuzzy control was first introduced by Zadeh [30], and then further developed and used by Mamdani and Assilian [8], Tang [24, 25], and Zimmermann [31]. Fuzzy control was utilized in several applications [e.g., 18, 24–26, 31]. Most recent developments in fuzzy control are provided by van Altrock et al. [26], Yager [28], and Gupt and Qi [9]. Herein fuzzy logic is used for construction safety assessment and control. Fuzzy if-then rules are utilized in this application. The developed implication function satisfied special criteria to ensure its adequate performance.

2. Condition assessment

In any control strategy for construction activities, a condition assessment procedure is considered an important and integral component of the system. For construction activities, the condition assessment procedure could be viewed as the transfer function in traditional control theory. Although no mathematical model could

Correspondence to: Prof. B.M. Ayyub, Civil Engineering Department, University of Maryland, College Park, MD 20742, USA.

be developed for the controlled system, i.e., construction activity, to represent the input/output relationship, a procedure could be developed whereby an output level is evaluated given a set of input states. In this study, the condition assessment procedure transfers the states of the input variables into some attribute measures, e.g. safety assessment, of the controlled system which represent the output of the system. The condition assessment procedure is applied at two main hierarchical levels. This hierarchical structure fits the nature of construction activities very well.

2.1. Condition assessment at the process level

At the process level, for each potential state of a given variable, a failure likelihood impact is assigned. The states of the variables as well as the failure likelihood levels are defined using linguistic measures which are quantified using fuzzy set theory [29]. The assigned likelihood level depends on the state of the variable rather than the variable itself. Thus, a sensitivity factor that reflects the importance of each variable to the attribute of interest, is applied to the state fuzzy set of each variable. The adjusted state fuzzy set is defined as

$$\mu_A^{\mathrm{a}}(x) = m\mu_A(x) \tag{1}$$

where $\mu_A^{\mathrm{a}}(x)$ = adjusted membership function of element x in fuzzy set A; m = importance factor; and $\mu_A(x)$ = original membership function of element x in fuzzy set A. The relationship between each individual potential state and its failure likelihood impact is defined by the Cartesian produce of fuzzy relations. Figure 1 shows a block diagram that summarizes the safety assessment procedure at the process level. Effects of potential combinations of states are then evaluated using the union of fuzzy relations as shown in Figure 1 and defined as follows;

$$\mu_{R_1 \cup R_2}(x) = \mathrm{MAX}\{\mu_{R_1}(x), \mu_{R_2}(x)\} \tag{2}$$

where $\mu_{R_1 \cup R_2}(x)$ = membership function for the union; MAX = maximum operator; $\mu_{R_1}(x)$ = membership function of element x in fuzzy relation R_1; and $\mu_{R_2}(x)$ = membership function of the same element x in fuzzy relation R_2. An aggregation and defuzzyfication procedures are applied to the resulting combined relation matrix in order to reduce the matrix into a single point estimate. The maximum operator is utilized as an aggregation tool operating on the columns of the combined relation matrix which results in a failure likelihood fuzzy set. The defuzzyfication procedure is then applied to the failure likelihood fuzzy set to evaluate the single point estimate. The utilized defuzzyfication procedure is defined as

$$\log_{10}(P_{f_j}) = \frac{\sum_{i=1}^{N_p} \mu(P_{f_i})\log_{10}(P_{f_i})}{\sum_{i=1}^{N_p} \mu(P_{f_i})} \tag{3}$$

for $j = 1, 2, \ldots, 9$,

where P_{f_i} = ith element in the failure likelihood fuzzy set; and N_p = number of elements in the failure likelihood fuzzy set.

If the exact states of the involved variables are known at the time of the analysis, i.e., real-time analysis, the failure likelihood evaluated in (3) represents a measure of the process safety level. However, if a what-if type of analysis is being performed, different variables may take several potential states at any point in time. Thus, the developed procedure should incorporate a scheme by which the uncertainties in the variables states could be preserved and accounted for in the evaluated safety level. This could be accomplished by using the behavior function of the process. This function assigns a probability measure based on the frequency of occurrence of each possible combination of states of the variables. Based on the previous discussion, a failure likelihood level was calculated for each possible combination of states. However, the resulting failure likelihood level has a frequency of occurrence which is equal to that of the combined state causing it. In other words, the resulting failure likelihood at the process level is distributed over all possible combinations of states. It is however important to determine a single point estimate of the process failure likelihood level, taking into account all possible combinations of states. This could be accomplished by applying the mathematical expectation which is defined as

$$E(P_f) = \sum_{j=1}^{9} P_{f_j} F_I(C_{I_j}) \tag{4}$$

where $E(P_f)$ = expected value of the probability

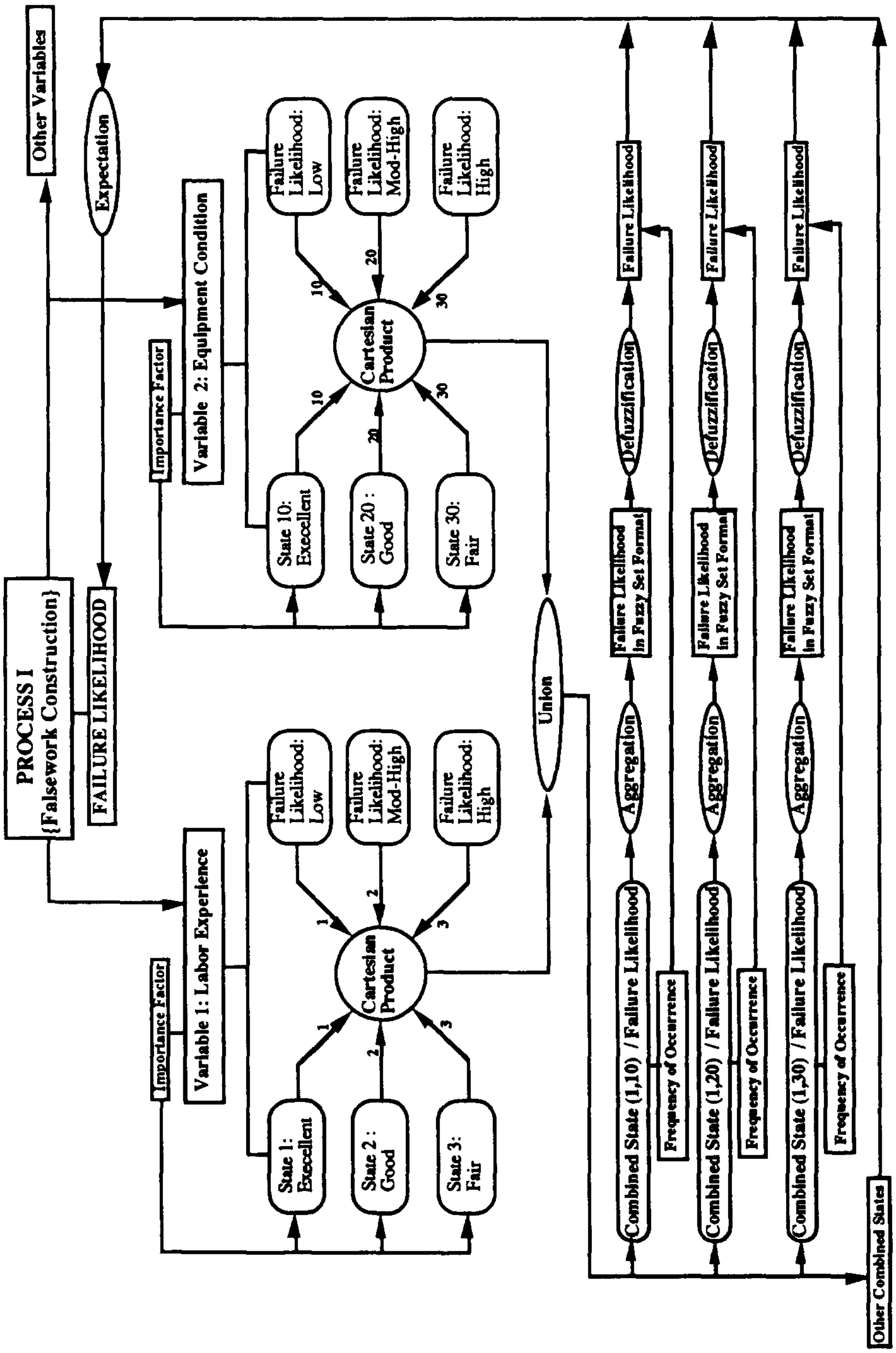

Fig. 1. Block diagram of attribute assessment at the process level.

of failure; P_{f_j} = the probability of the jth overall state as defined in (3); and $F_I(C_{I_i})$ = the frequency of occurrence of overall state C_{I_i} given by the behavior function of process I. Applying (4) to the probabilities of failure of all possible combinations, an expected value of the probability of failure at the process level could be evaluated.

2.2. Condition assessment at the activity level

The objective at this stage is to combine the resulting failure likelihood evaluations for all the involved processes into a failure likelihood assessment of the construction activity as a whole. Two issues need to be addressed at this tage, the first is the impact of the global overall states on the failure likelihood of the whole activity and the second is the overall behavior function that represents the entire activity.

2.2.1. Impacts of global overall states

In the previous section, combinations of states of variables, i.e., overall states, and their effect on a specific process were studied. The same rationale could be applied at the activity level. However, combinations of overall states, i.e., global overall states, and their effect on the entire activity should be studied. It is quite clear that the impact of overall states on the failure likelihood of the process is different from its impact on the construction activity as a whole. This difference arises from the fact that the impact of an overall state on the construction activity should include an importance factor of the process, it models, that reflects the significance of the process attribute condition on the failure likelihood of the construction activity. This importance factor is applied as a multiplier to the corresponding combined relation matrix at the process level. The combined effects of all the involved processes on the activity failure likelihood are then determined using the union function of fuzzy relations as defined in (2). The union function should be applied to the adjusted combined relation matrices of each potential global overall state. At this stage, all possible combinations should be considered at the process and activity levels. For example, if two variables were considered for each process with three potential states per variable and two processes were considered for the activity, this results in nine different combinations considered at the process level and eighty one potential combinations, i.e., global overall states, at the activity level. The resulting matrix of all potential combinations is defined herafter as the combined process matrix. Figure 2 shows a schematic representation of the combined process matrix, which contains the unions of the corresponding combined relation matrices as its entries. Applying the same aggregation and defuzzyfication procedures, which were developed at the process level, a failure likelihood level could be determined for each combined relation matrix. The final computational step is to determine a single point estimate of the activity failure likelihood level taking into account all possible combinations of the overall states. This could be accomplished by defining the overall behavior function of the whole activity, which is discussed in detail in the next section. Figure 3 summarizes the condition assessment methodology at the activity level in a block diagram format.

2.2.2. The overall behavior function

An overall behavior function of the construction activity should be defined, based on known behavior functions of its components, i.e., the different processes. The concepts and methods used to solve identification problems in system theory [4–6 and 11] were utilized in evaluating this behaviour function. Table 1 shows an example behavior function $F_{\mathrm{I}}(C_{\mathrm{I}_i})$ for process I, $i = 1, 2, \ldots, 9$. The overall states represent all possible combinations of the two considered variables. Table 2 shows an example behavior function $F_{\mathrm{II}}(C_{\mathrm{II}_j})$ for process II, $j = 1, 2, \ldots, 9$. Knowing these behavior functions, an overall behavior function that represents the overall system, i.e., the construction activity, needs to be evaluated. Several identification schemes are available in the literature [4–6 and 11]. These schemes could be utilized in the example under consideration after modifying them to suit the nature of the problem. Referring to Tables 1 and 2, in order to define the behavior function of the overall system, all possible combinations of the overall states of the two processes should be considered. Table 3 shows these possible

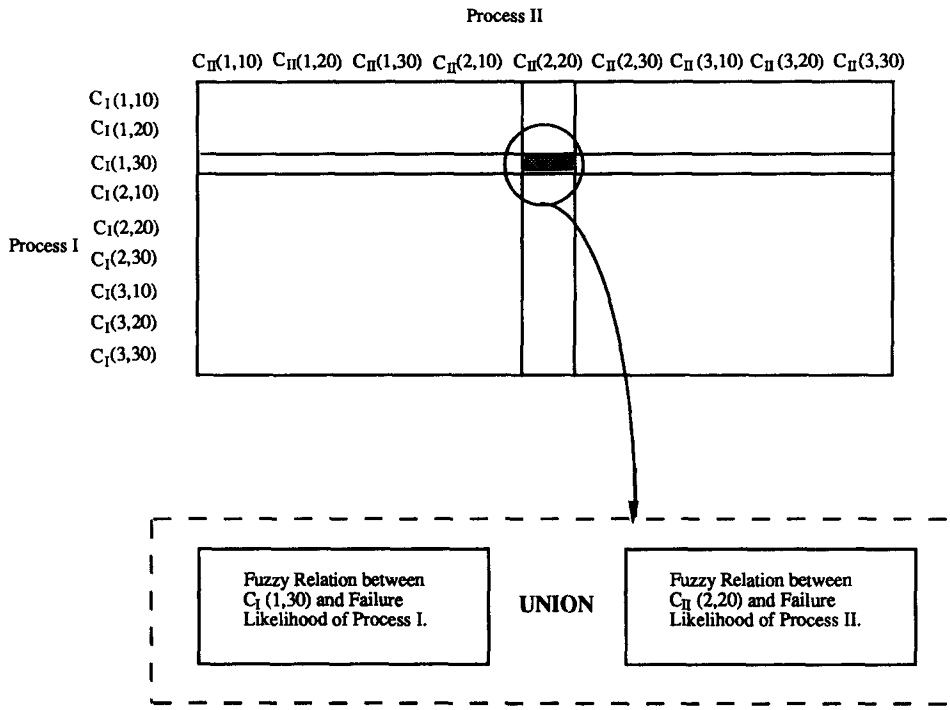

Fig. 2. Combined process matrix.

combinations together with the associated unknown frequencies of the combinations, i.e., $F_A(C_{\mathrm{I}_i}, C_{\mathrm{II}_j})$. In order to solve for the unknown frequencies, two conditions should be satisfied. The first condition requires that the frequency of an overall state of a process calculated from the unknown overall behavior function i.e., unknown frequencies in Table 3, should be compatible with the frequency of the same overall state calculated from the behavior function for the process. In other words, each process behavior function is considered as a marginal probability distribution which is known. The problem then reduces to determining the joint probability mass function that is compatible with all defined marginal distributions. This condition can be expressed as follows:

$$F_{\mathrm{I}}(C_{\mathrm{I}_i}) = \sum_{j=1}^{9} F_A(C_{\mathrm{I}_i}, C_{\mathrm{II}_j}) \quad \text{for } i = 1, 2, \ldots, 9, \tag{5-a}$$

$$F_{\mathrm{II}}(C_{\mathrm{II}_j}) = \sum_{i=1}^{9} F_A(C_{\mathrm{I}_i}, C_{\mathrm{II}_j}) \quad \text{for } j = 1, 2, \ldots, 9 \tag{5-b}$$

where $F_{\mathrm{I}}(C_{\mathrm{I}_i})$ = the value of the behavior function for process I for the overall state C_{I_i} as defined in Table 1; $F_{\mathrm{II}}(C_{\mathrm{II}_j})$ = the value of the behavior function for process II for the overall state C_{II_j} as defined in Table 2; $F_A(C_{\mathrm{I}_i}, C_{\mathrm{II}_j})$ = the value of the overall behavior function for the global overall state $(C_{\mathrm{I}_i}, C_{\mathrm{II}_j})$. Referring to Tables 1, 2 and 3, an example constraint equation could be written as

$$f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 + f_8 + f_9 = 0.1 \tag{6}$$

where $f_1, f_2, \ldots, f_9$ = values of the overall behavior function for the combinations $C_{\mathrm{II}}(1, 10)$ and all possible overall states of process II; and 0.1 = value of the behavior function of process I for the overall state $C_{\mathrm{II}}(1, 10)$. The total number of constraints is equal to the total number of overall states in both processes. In the example under consideration eighteen equations similar to (6) were developed. The second condition requires that all resulting probabilities should be positive. This condition could be stated as follows:

$$f_i \geqslant 0 \quad \text{for } i = 1, 2, \ldots, 81. \tag{7}$$

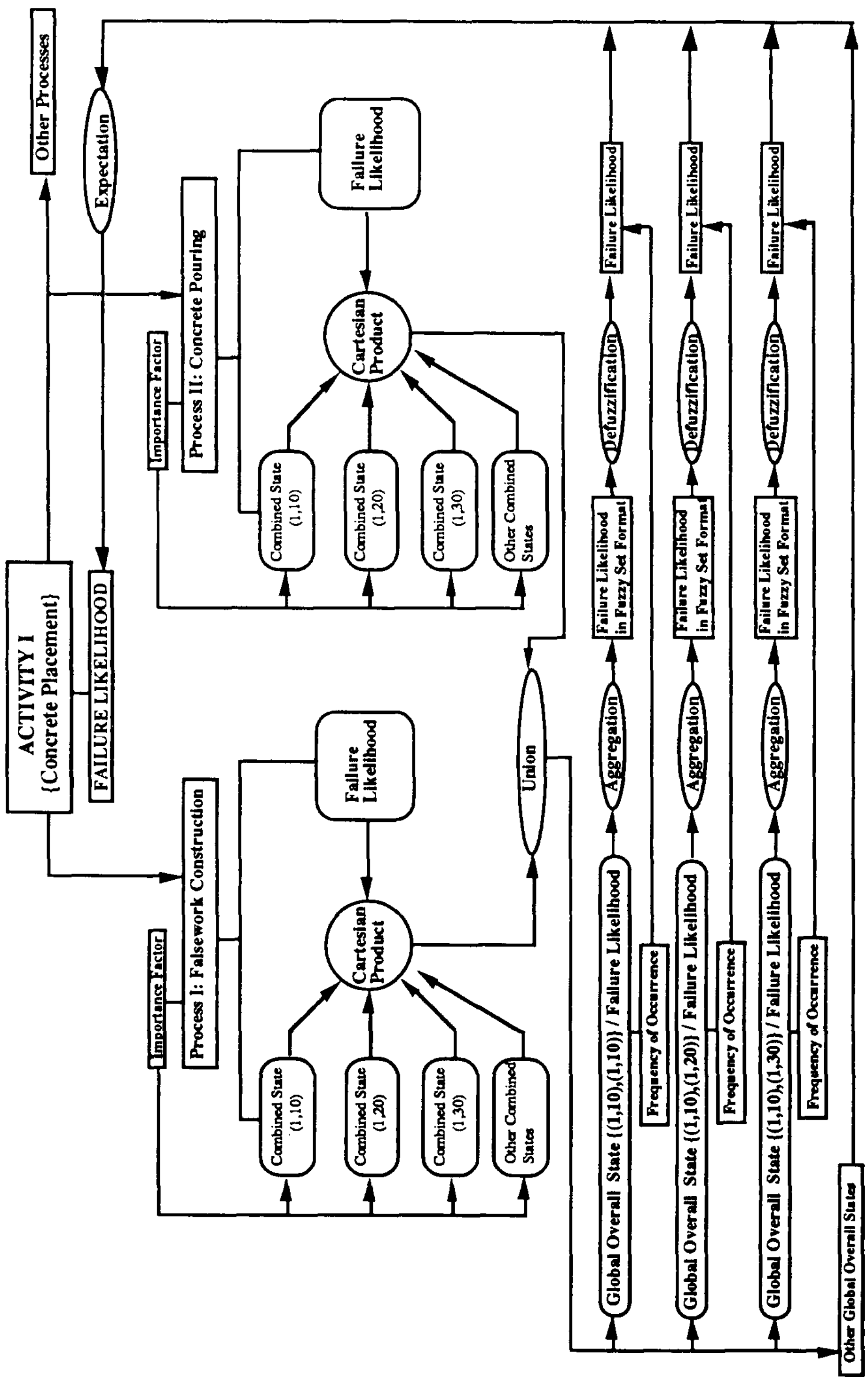

Fig. 3. Block diagram of attribute assessment at the activity level.

Table 1. Example behavior function for process I

Labor experience	Equipment condition: Excellent (10)	Good (20)	Fair (30)
Excellent (1)	$F_{\text{I}}(C_{11}) = 0.10$	$F_{\text{I}}(C_{12}) = 0.05$	$F_{\text{I}}(C_{13}) = 0.10$
Good (2)	$F_{\text{I}}(C_{14}) = 0.10$	$F_{\text{I}}(C_{15}) = 0.25$	$F_{\text{I}}(C_{16}) = 0.20$
Fair (3)	$F_{\text{I}}(C_{17}) = 0.05$	$F_{\text{I}}(C_{18}) = 0.05$	$F_{\text{I}}(C_{19}) = 0.10$

Table 2. Example behavior function for process II

Labor experience	Equipment condition: Excellent (10)	Good (20)	Fair (30)
Excellent (1)	$F_{\text{II}}(C_{\text{II}1}) = 0.10$	$F_{\text{II}}(C_{\text{II}2}) = 0.15$	$F_{\text{II}}(C_{\text{II}3}) = 0.05$
Good (2)	$F_{\text{II}}(C_{\text{II}4}) = 0.05$	$F_{\text{II}}(C_{\text{II}5}) = 0.20$	$F_{\text{II}}(C_{\text{II}6}) = 0.10$
Fair (3)	$F_{\text{II}}(C_{\text{II}7}) = 0.15$	$F_{\text{II}}(C_{\text{II}8}) = 0.10$	$F_{\text{II}}(C_{\text{II}9}) = 0.10$

Table 3. Behavior function for the activity

Process I	Process II: $C_{\text{II}1}(1, 10)$	$C_{\text{II}2}(1, 20)$	$C_{\text{II}3}(1, 30)$	$C_{\text{II}4}(2, 10)$	$C_{\text{II}5}(2, 20)$	$C_{\text{II}6}(2, 30)$	$C_{\text{II}7}(3, 10)$	$C_{\text{II}8}(3.20)$	$C_{\text{II}9}(3, 30)$
$C_{11}(1, 10)$	$\underline{f_1}$	$\underline{f_2}$	f_3	f_4	f_5	f_6	f_7	f_8	f_9
$C_{12}(1, 20)$	f_{10}	$\underline{f_{11}}$	$\underline{f_{12}}$	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
$C_{13}(1, 30)$	f_{19}	f_{20}	$\underline{f_{21}}$	$\underline{f_{22}}$	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}
$C_{14}(2, 10)$	f_{28}	f_{29}	f_{30}	$\underline{f_{31}}$	$\underline{f_{32}}$	f_{33}	f_{34}	f_{35}	f_{36}
$C_{15}(2, 20)$	f_{37}	f_{38}	f_{39}	f_{40}	$\underline{f_{41}}$	$\underline{f_{42}}$	f_{43}	f_{44}	f_{45}
$C_{16}(2, 30)$	f_{46}	f_{47}	f_{48}	f_{49}	f_{50}	$\underline{f_{51}}$	$\underline{f_{52}}$	f_{53}	f_{54}
$C_{17}(3, 10)$	f_{55}	f_{56}	f_{57}	f_{58}	f_{59}	f_{60}	$\underline{f_{61}}$	$\underline{f_{62}}$	f_{63}
$C_{18}(3, 20)$	f_{64}	f_{65}	f_{66}	f_{67}	f_{68}	f_{69}	f_{70}	$\underline{f_{71}}$	$\underline{f_{72}}$
$C_{19}(3, 30)$	f_{73}	f_{74}	f_{75}	f_{76}	f_{77}	f_{78}	f_{79}	$\underline{f_{80}}$	$\underline{f_{81}}$

Solving the set of constraints and the inequalities defined in (7), a solution for the overall behavior function results. However, it is obvious that this solution is not unique. In other words, the solution defines a range within which the predefined conditions are satisfied. In order to select the optimal overall behavior of the overall system, the principle of maximum entropy was utilized [12 and 13]. The principle of maximum entroy as introduced by [10] states that the selection of a probability distribution should be based on the maximization of the entropy, subject to all additional constraints required by the available information. This means that an optimization problem should be solved in order to obtain the required solution. In addition, all additional constraints should be linearly independent in order to be able to get a unique solution. Thus, the solution of the optimization problem should comprise two main phases. In the first phase, the set of constraints should be examined to develop a linearly independent subset. In the second phase, the optimization procedure should be performed. The principle of maximum entropy results in selecting a distribution with the least amount of information or maximum uncertainty. Using the Shannon entropy as a measure of uncertainty, the objective then reduces to maximizing the following equation:

$$H = -\sum_{i=1}^{81} f_i \, \text{Ln}(f_i), \tag{8}$$

where H = Shannon entropy. Therefore, the objective is to define the frequency distribution (or probability mass function) that maximizes the Shannon entropy with linearly independent constraints, which are a subset of the constraints defined by (4). The optimization scheme utilized in this paper is a combination of the Jacobian method [27], and the Lagrange multipliers method [23]. The Jacobian method was utilized to determine the linearly independent constraints, while the Lagrange multipliers method was utilized to solve the optimization problem.

Phase I: *Linear independence*. Linear independence is a very essential condition that should be satisfied in any set of additional constraints for the optimization problem. The Jacobian method of optimization [27] was utilized to develop a procedure for detecting and deleting any linearly dependent equations in the set of additional constraints. According to this method the unknowns were divided into two groups, namely, state and decision variables. There are no specific criteria that govern the choice of the different variables in either of the two groups. However, the only needed limitation is that the number of state variables should be the same as the number of constraints. Once the state variables were defined, a Jacobian matrix was then developed. The Jacobian matrix is defined as follows:

$$\frac{\partial(T_1,\ldots,T_{18})}{\partial(e_1,\ldots,e_{18})}=\begin{bmatrix}\frac{\partial T_1}{\partial e_1} & \frac{\partial T_1}{\partial e_2} & \cdots & \frac{\partial T_1}{\partial e_{18}}\\ \frac{\partial T_2}{\partial e_1} & \ddots & & \vdots\\ \vdots & & & \\ \frac{\partial T_{18}}{\partial e_1} & \cdots & & \frac{\partial T_{18}}{\partial e_{18}}\end{bmatrix}, \tag{9}$$

where $T_i = i$th constraint; and $e_i = i$th state variable. For the example under consideration, the state variables were chosen to be the elements of the leading diagonal and the first subdiagonal of the behavior function matrix defined in Table 3. The choice of the state variables was based on an intuitive requirement of using at least one state variable in each constraint equation. The rational behind this requirement was based on the definition of the Jacobian matrix. If this rule was not satisfied, the Jacobian matrix would include zero rows corresponding to the missing constraints. Thus, these constraints would not be examined for linear dependency. The rank of the Jacobian matrix was then calculated. The rank is defined as the largest number of linearly independent equations in a given set [20 and 21]. If the rank is equal to the number of constraints, the set of constraints is a linearly independent one. However, if the rank is less than the number of constraints, the difference represents the number of linearly dependent equations. An iterative scheme was then developed by which each equation was examined for linear independence. As a result, this approach detected the linearly dependent equations within the given set of constraints.

Phase II: *The Lagrange multipliers method*. This phase deals with the calculation of the optimum solution based on the Lagrange multipliers method. According to [23], the maximized function (H) as defined in (8) should satisfy the following set of equations

$$\frac{\partial H}{\partial f_i}+\lambda_1\frac{\partial T_1}{\partial f_i}+\cdots+\lambda_{17}\frac{\partial T_{17}}{\partial f_i}=0 \tag{10}$$

where f_i = represents the variables involved in the maximized function, i.e., $f_1, f_2, \ldots, f_{81}$; λ_1 = constant multiplier which should be evaluated, i.e., Lagrange multiplier, and j varies from 1 to the number of linearly independent constraints which is 17 in this example; and T_j = set of linearly independent constraints as defined in phase I. Equation (10) results in a number of equations that is equal to the number of variables. Using these equations, expressions for the unknowns, i.e., f_i, were developed in terms of the Lagrange multipliers, i.e., λ_j. These expressions were then substituted in the original set of linearly independent constraints. These substitutions resulted in a set of nonlinear simutaneous equations in the Lagrange multipliers. The Newton–Raphson numerical method was then utilized in solving this set of nonlinear equations. The Lagrange multipliers were then substituted in the developed expressions for the optimum solution.

3. Fuzzy-based control strategy

A fuzzy controller can be considered as an abstracted collection of rules that summarize the experiences of a human controller. This collection of rules form what is known as the rule-base. Each rule is in the form of an IF THEN implication rule where if the antecedent, i.e., the input of the rule, is satisfied then the consequent, i.e., the output of the rule, is implied. In the control of construction activities, it is necessary to use two antecedents rather than just one. The two antecedents are the normalized amount of deviation of the attribute from its standard value, i.e., the error, and the rate of deviation from the standard value, i.e., the change in error. The two antecedents result in a three dimensional rule rather than the usual two dimensional rule. In order to infer actions from the rule-base, a mathematical model that represents the rule should be developed. This model is usually referred to as the implication function [14–17]. An inferring mechanism should then be developed in order to evaluate a single representative control action for a given set of rules and input values. Figure 4 shows the control strategy in a block diagram format.

3.1. Implication function

The implication function is a mathematical model that represents the rule. In this study a three dimensional implication function was developed such that some specified criteria are satisfied [19]. The implication function utilized in this study is defined as

$$(A \times B) \rightarrow C \tag{11}$$

and could be interpreted as

$$\text{IF}(A \text{ AND } B)\text{THEN } C. \tag{12}$$

This function results in a three dimensional relation matrix with a membership function defined as

$$\mu_{3\mathrm{R}}(l, y, z) = \mu_{A\times B}(l, y) \rightarrow \mu_{C}(z) \tag{13}$$

where $\mu_{3\mathrm{R}}(l, y, z)$ = membership value of error element l, change in error element y and the resulting control action element z in the three dimensional implication function defined in (1); $\mu_{A\times B}(l, y)$ = membership value of error element l and change in error element y in the Cartesian product of the error fuzzy set A and change in error fuzzy set B; $\rightarrow = 1$ if $\mu_{A\times B}(l, y) \leqslant \mu_C(z)$, and 0 if $\mu_{A\times B}(l, y) \geqslant \mu_C(z)$; and $\mu_C(z)$ = membership value of element z in the control action fuzzy set C.

3.2. Inference mechanism

An inference mechanism is a procedure by which a control action could be inferred given a rule-base and a set of input values. For a single rule, the compositional rule of inference developed by [29] was found satisfactory according to the reviewed literature [14–17]. For the three dimensional relation matrix defined by (3), the compositional rule used for evaluating the initial Control Action in this study is defined as

$$\mu_{(IA)_i}(z) = \underset{\text{all } l \in L}{\text{MAX}}\,\text{MIN}\left[\mu_{\text{Error}}(l)\left\{\underset{\text{all } y \in Y}{\text{MAX}}\,\text{MIN}(\mu_{\text{Change in Error}}(y), \mu_{3\mathrm{R}}(l, y, z))\right\}\right] \tag{14}$$

where $\mu_{(\mathrm{IA})_i}(z)$ = membership value of element z in the ith rule initial control action fuzzy set; $\mu_{\text{Error}}(l)$ = membership value of element l in the fuzzyfied input Error level; $\mu_{\text{Change in Error}}(y)$ = membership value of element y in the fuzzyfield input Change in Error level; and $\mu_{3\mathrm{R}}(l, y, z)$ = membership function of the three dimensional fuzzy relation defined by the implication function in (13). Equation (14) results in a fuzzy subset of the Control Action universe that represents the initial control action for each rule. If the rule-base contains a number of rules that apply, a procedure needs to be defined such that all the applicable rules contribute to the final control action [14–17]. Moreover, the levels of applicability of the individual rules are not the same. In this study, a non-negative value that is less than or equal to one is evaluated, for each rule, for a given set of input values. This value which is referred to as the applicability factor is evaluated based on how close the input values are to the rule antecedents. The applicability factor could be expressed mathematically as

$$F_i = \text{MIN}\{\mu_{(A)_i}(e), \mu_{(B)_i}(\text{ce})\} \tag{15}$$

where F_i = ith rule applicability factor and i ranges from 1 to the total number of rules n; MIN = minimum operator; the rule's first antecedent $\mu_{(A)_i}(e)$ = membership value of the

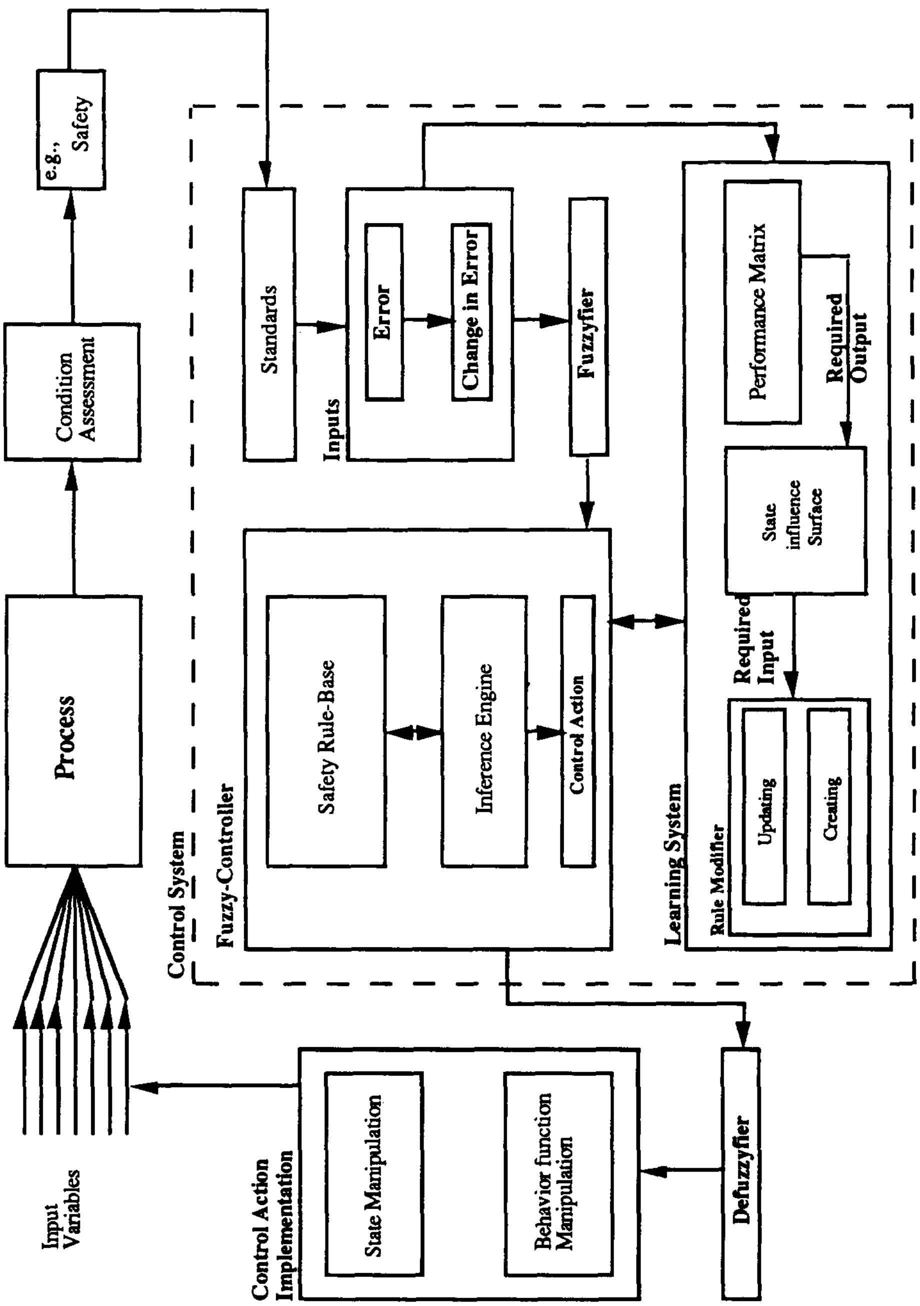

Fig. 4. Block diagram of the control system of construction activities.

input Error level e; and the rule's second antecedent $\mu_{(B)_i}(\mathrm{ce})$ = membership value of the input Change in Error level ce. The applicability factor is then introduced as a multiplier to the rule action which was developed in (14). Thus, if the rule is applicable with some degree, its final action is scaled based on its applicability factor. This step could be expressed mathematically as

$$\mu_{(\mathrm{FA})_i} = F_i \mu_{(\mathrm{IA})_i} \tag{16}$$

where $\mu_{(\mathrm{FA})_i}$ = membership function of the final ith rule action; F_i = ith rule applicability factor and i ranges from l to the total number of rules n; and $\mu_{(\mathrm{IA})_i}$ = membership function of the initial ith rule action. The final step is to evaluate an overall action based on all applicable rules. The overall action is defined using the union function of fuzzy relations which is defined as

$$\mu_{(\mathrm{OA})}(z) = \underset{i=1}{\overset{n}{\mathrm{MAX}}}\ \mu_{(\mathrm{FA})_i}(z) \tag{17}$$

where $\mu_{(\mathrm{OA})}(z)$ = membership value of element z in the overall action fuzzy set; $\mu_{(\mathrm{FA})_i}(z)$ = membership function of element z in the overall action of the ith rule; and MAX = maximum opeator applied over all the applicable rules, where i ranges from l to the number of applicable rules n. The resulting overall action is a fuzzy subset of the universe of discourse of the fuzzy variable Control Action. In order to be able to apply the control action, it should be in the form of a crisp number. A crisp control action could be defined as follows:

$$\mathbf{OA} = \frac{\sum_{i=1}^{n} z\mu_{(\mathrm{OA})_i}(z)}{\sum_{i=1}^{n} \mu_{(\mathrm{OA})_i}(z)} \tag{18}$$

where $\mathbf{OA}$ = crisp overall action; $\mu_{(\mathrm{OA})}(z)$ = membership value of element z in the overall action where z ranges from 0 to +7; and n = total number of applicable rules. The crisp overall action is rounded off to the nearest integer for the purpose of implementation.

3.3. Self learning system

A fuzzy controlle depends on a rule-base where all previous experiences and information are stored. However, it is difficult to construct a complete rule-base that is capable of handling all potential situations. Thus, it is necessary to include a self learning unit within the control system that is responsible for expanding and updating the current rule-base. The self learning unit should identify situations or cases that are not considered in the current rule-base. It should also extract necessary information and construct new rules to handle these situations. In general, the self learning unit monitors the performance of the control system. It compares the performance with an ideal performance that is built in the learing unit and stored in what is defined as the performance matrix [22]. If the control system deviates from this ideal track, the learing system should be able to identify the underlying causes and improve the control system's performance. Unsatisfactory performances may be due to one of two main causes. The first is due to missing rules, i.e., the current rule-base cannot handle the current situation because none of its rules is applicable. Accordingly, the control system cannot suggest any corrective action. The second is due to untuned rules. In other words, the rules that apply to the current case needs to be adjusted in order to result in the expected ideal performance. Figure 5 summarizes the logic behind the operation of the learing system in a flow chart format.

3.3.1. Performance matrix

The performance matrix is a two dimensional matrix that summarizes the required output correction for the system, knowing the Error and the Change in Error levels [22]. The performance of the control system is measured using the already known Error and the Change in Error to look-up the required output correction according to the performance matrix. The performance matrix should be developed based on previous experiences and knowledge of experienced controllers. This matrix represents a generalized view of the required adjustment regardless of the type of process being controlled. This means that a single performance matrix could be used for the different controlled processes. The performance matrix entries range from 0 to +7 where these entries represent a power order. For example, if the entry is 3 then the required output correction is 10^3. The required process output is then evaluated as

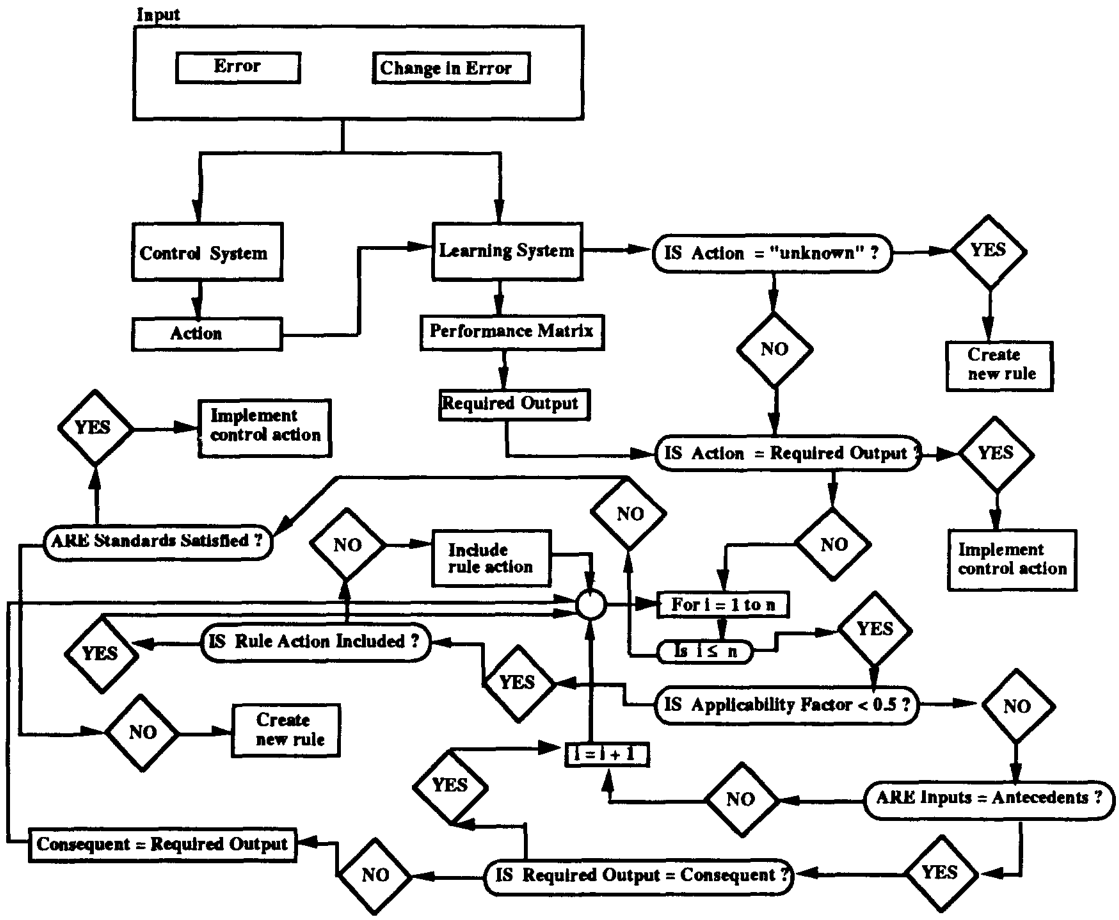

Fig. 5. Flow chart of the learning system.

follows:

$$RO = \frac{CO}{PM} \tag{19}$$

where RO = required process output, i.e., the safety level for example; CO = current process output; and PM = look-up value from the performance matrix, i.e., performance measure. The required process output is rounded off to the nearest integer for the purpose of implementation. Knowing the required process output, the next step is to determine which combination of states could result in the required process output. This is accomplished using the state influence surface which summarizes the process output, i.e., the safety level, for each potential combination of the involved variables. This surface could be considered as the inverse of the condition assessment methodology developed by [2]. The state influence surface could be expressed mathematically as

$$RI = COND^{-1}\{RO\} \tag{20}$$

where RI = required process input; $COND^{-1}$ = look-up value from the state influence surface that corresponds to the required output RO. Knowing the original states of the variables, the difference in levels represent the ideal control system output. The actual control system output, evaluated in (8), should then be checked against this ideal output. Based on the result of the comparison, the learing system should perform one of three actions. The first is to recommend the resulting action, if both systems resulted in the same answer, the second is to update an existing rule, and the third is to create a new rule. In the following discussion both updating and creating rules are discussed in detail.

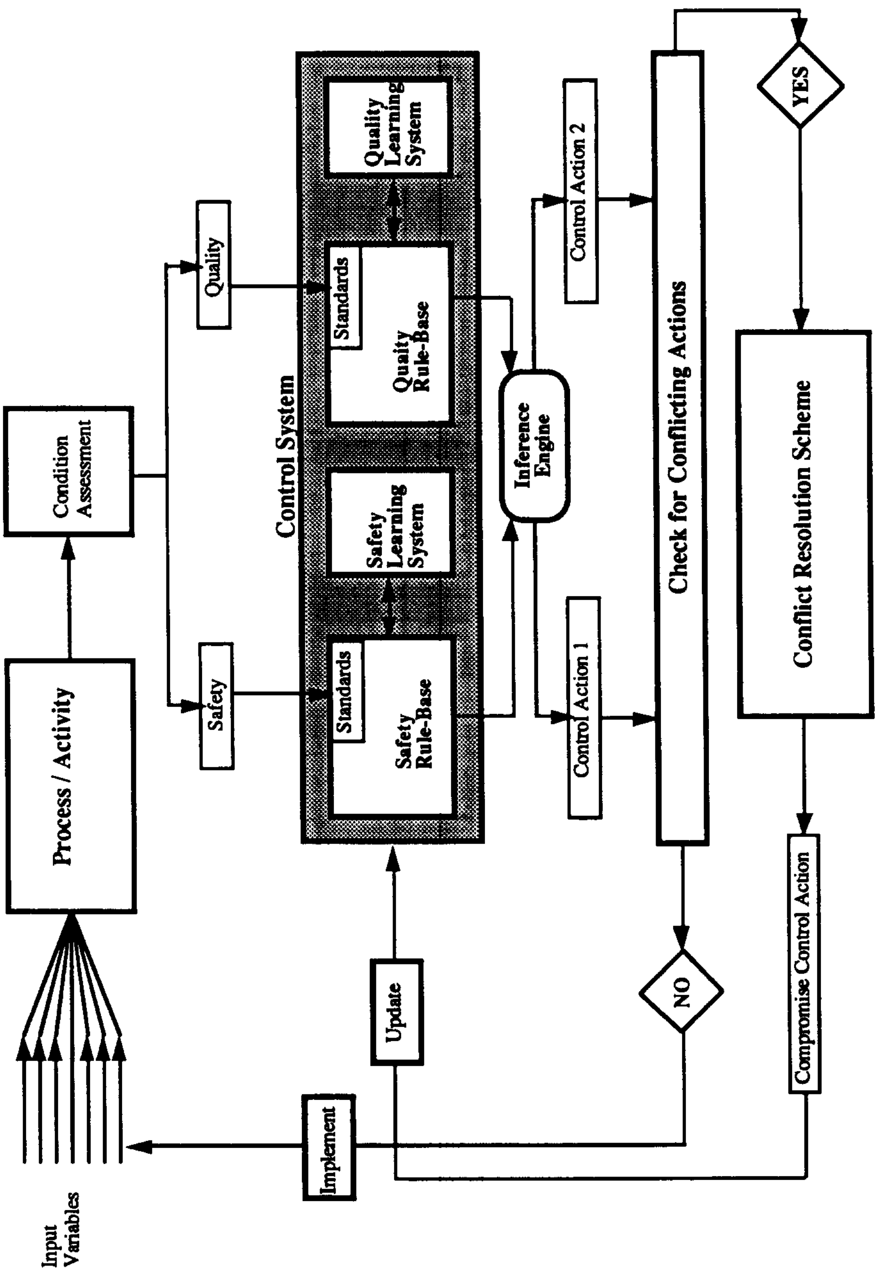

Fig. 6. Multi-attribute control strategy.

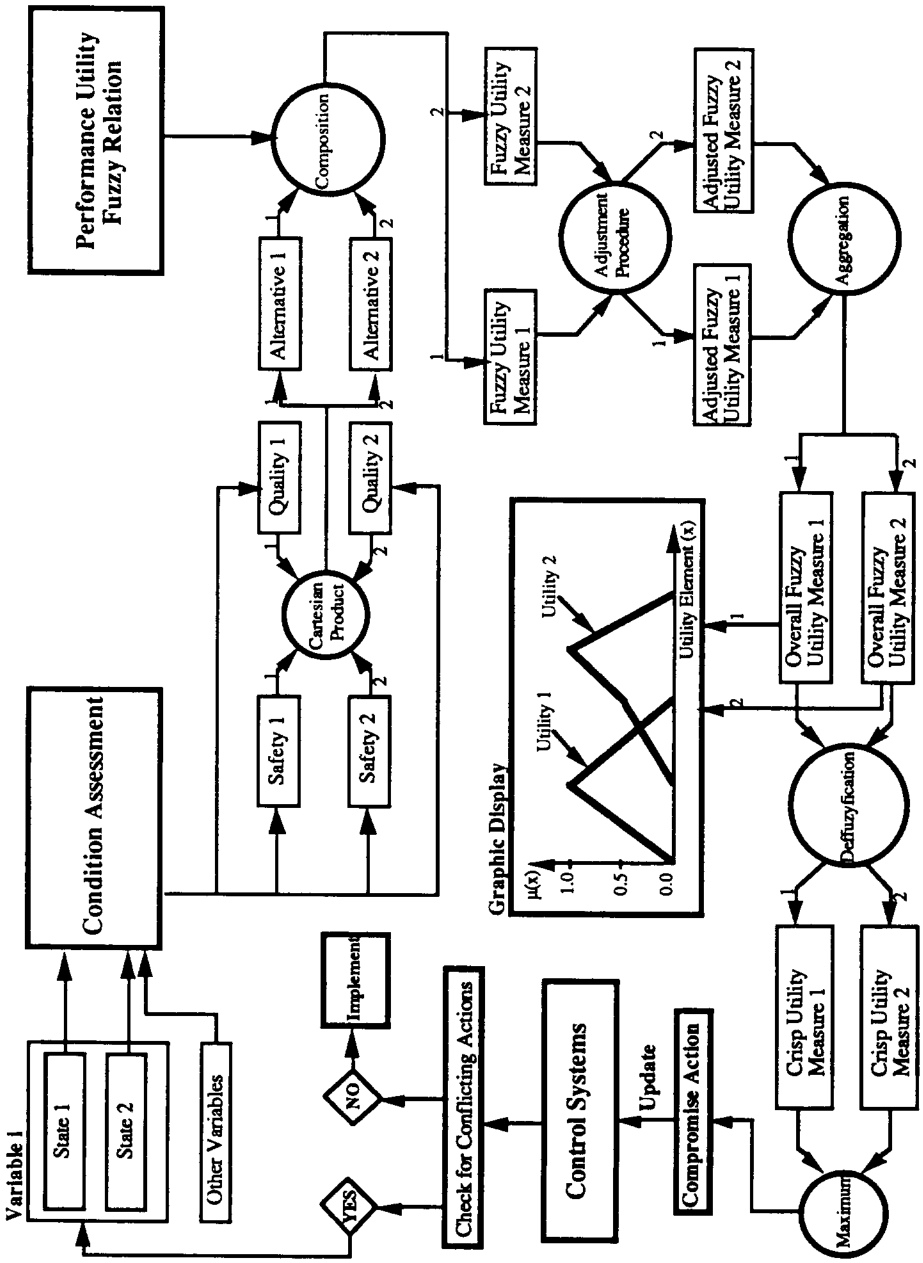

Fig. 7. Block diagram of the conflict resolution scheme.

3.3.2. Updating rules

The learing system may detect a discrepancy between the control action evaluated by the fuzzy-controller and the ideal control action suggested by the learning system. The presence of untuned rules might be responsible for such discrepancy. Untuned rules are applicable rules that imply a consequent that is different from the ideal control action for the same antecedents. The learning system then examines all the rules in order to identify the untuned rule which is the only rule that is eligible for updating. In this study, the criterion for updating rules is defined as a perfect match between the input values for the rule and its antecedents. Once, the untuned rule is identified, the rule consequent is then replaced with the ideal process output evaluated by the learning system. If none of the applicable rules matched the input values, no updating could be performed and a new rule should be created. Figure 6, summarizes the logic used in updating rules in a flow chart format.

3.3.3. Creating rules

A second reason for having discrepancies between the control system and the learing system is due to a missing matching rule. In other words, if all the rules in the rule-base have an appliability factor of zero for the current combination of Error and Change in Error, i.e., the control system cannot evaluate a control action for the current combination, the control system output in this case is an 'unknown' message. Once the learning system detects the 'unknown' message, it triggers a procedure that is responsible for creating and linking new rules to the rule-base. The learning system then starts creating the new rule by simply copying an old rule and tuning it to match the antecedents of the case under consideration and the consequent evaluated by the learning system. The rule is then saved with a new name that matches its order in the rule-base. Before including the new rule in the rule-base and linking it to the control system, it should be tested to ensure its uniqueness. This test is performed by comparing the different rule actions in the control system and the one resulting from the new rule. If the test is successful, i.e., there is not a similar rule, the learning system then develops the new rule and links it to the control system. If the rule is rejected, the learning system conveys an appropriate message to explain its action. Once the new rule is accepted and linked to the control system, it is considered an integral part of the rule-base and it contributes to any future control actions based on its applicability. Figure 7 shows the logic used by the learning system in creating new rules.

4. Conflict resolution

In most construction projects, it is essential to monitor and control two or more attributes at the same time. The proposed system comprises several parallel single-attribute control systems. Each system is responsible for monitoring and controlling a single attribute, and comprises a suitable rule-base that is related to the attribute of interest and the corresponding control standard. In general, each attribute should utilize a separate rule-base. However, in some special cases certain attributes may end up sharing the same rule-base. A self-learning system should be established for each control system. Different learning systems utilize relevant performance matrices depending on the control attribute under consideration. Because of multiple control actions, conflicts may arise if the same variable is selected as a critical control variable for two or more attributes. In order to resolve any conflicting actions, a conflict resolution algorithm is needed. For the application under consideration an algorithm that is capable of handling uncertain decision problems is required. The proposed algorithm develops a compromise control action which satisfies all involved attributes as close as possible. This compromise solution is one of several alternatives developed by the algorithm. Each alternative is then rated using a fuzzy utility function. A rating procedure is then utilized in order to rank the individual alternative utilities for the purpose of selecting the solution that maximizes the utility rating. Each control action, when applied to the critical control results in a new state for that variable. This new state, together with the states of the other involved variables, renders a new condition for each controlled attribute. In other words, the attribute has different potential conditions resulting from the suggested control actions. Hence, several alternatives result, each represents a combination of the alternative conditions, evaluated in a fuzzy set format, that corresponds to a certain control action. The ith alternative is defined as the Cartesian product of the individual attribute condition fuzzy sets. This could be defined mathematically as

$$\mu_{A_i}(x, y) = \text{MIN}[\mu_{T_1}(x), \mu_{T_2}(y)] \tag{21}$$

where $\mu_{A_i}(x, y)$ = membership value of the combination of element x in the first attribute condition fuzzy set and element y in the second attribute condition fuzzy set of the ith alternative; MIN = minimum operator; $\mu_{T_1}(x)$ = membership value of element x in the first attribute condition fuzzy set; and $\mu_{T_2}(y)$ = membership value of element y in the second attribute condition fuzzy set. Knowing the individual alternatives resulting from the conflicting control actions, a rating system should be developed by which each alternative could be evaluated. Because of the use of linguistic measures and quantifiers in construction activities, a fuzzy-based utility function is a suitable approach for handling the rating problem. A fuzzy utility function subjectively assigns utility levels with certain degrees of belief for all potential fuzzy alternatives. Expert judgement should be utilized in the development of the fuzzy utility function where a performance utility matrix results. Each potential combination of attribute conditions is assigned a degree of belief that it results in a certain utility level. In this context, this utility function is modelled as a fuzzy relation between the universe of the Cartesian product of the attributes of interest and the universe of the utility function. Knowing the fuzzy utility function and the individual alternatives, the composition of fuzzy relations could be utilized in order to evaluate a fuzzy utility measure for each potential alternative [7]. Thus, the fuzzy utility measure for a certain alternative is defined as

$$\mu_{U_j}(u_i) = \underset{\text{for all } (x,y)}{\text{MAX}} \text{MIN}(\mu_{A_j}(x, y), \mu_{3R}(x, y, u_i)) \tag{22}$$

where $\mu_{U_j}(u_i)$ = membership value of the utility level u_i in the fuzzy measure of the jth alternative; MAX = maximum operator; MIN = minimum operator; $\mu_{A_j}(x, y)$ = membership value of element (x, y) in the jth alternative defined in (1); and $\mu_{3R}(x, y, u_i)$ = membership value of element (x, y) with a utility level u_i, defined by a three-dimensional fuzzy relation in the performance utility matrix. Because the elements of the universe of the fuzzy utility function are linguistic measures, defined by fuzzy sets themselves, an adjustment process is required in order to evaluate a fuzzy utility measure with numerical elements. The rational behind such adjustment is based on the fact that further processing and ranking of the individual utility measures is required. However, if the elements remained in a lingustic format, such processing would be difficult. A problem then arises where each numerical element in the fuzzy utility measure might have more than one degree of belief. Thus, an aggregation procedure is required in order to evaluate an overall fuzzy utility measure for each individual alternative. Utilizing the maximum operator as an aggregation tool, the aggregated fuzzy utility measure is defined as

$$\mu^{g}_{U_j}(x) = \underset{\text{for all } u_i}{\text{MAX}} \mu^{a}_{u_{ij}}(x) \tag{23}$$

where $\mu^{g}_{U_j}(x)$ = membership value of element x in the overall fuzzy utility measure of the jth alternative; MAX = maximum operator; and $\mu^{a}_{u_{ij}}(x)$ = adjusted membership value of element x in the linguistic utility element u_i in the fuzzy utility measure of the jth alternative. Knowing the overall fuzzy utility measure for each individual alternative, a ranking procedure is then utilized in order to select the best compromise solution. In this study, a defuzzyfication process is utilized as a ranking procedure. Since utility measures have been already evaluated in a fuzzy set format in order to preserve the underlying uncertainties, single point estimates could be evaluated, at this stage, without loosing any available information. The resulting single point estimate is used as a rank that represents the utility of the corresponding alternative. This rank is then used as a basis for selecting the best compromise alternative. In general, the best compromise alternative is the one resulting in the maximum utility measure. Thus, a maximization operator is utilized in selecting the final compromise alternative. The rank of the compromise utility is then defined as

$$R_c = \underset{\text{for all } j}{\mathrm{MAX}}\, R_j \tag{24}$$

where R_c = rank of the compromise utility; MAX = maximum operator applied for all j which varies from 1 to the number of potential alternatives; and R_j = rank of the overall fuzzy utility measure of the jth alternative.

5. Case study

In this case study, the construction of a real bridge was investigated. The bridge collapsed during construction, and its failure was reported in the literature [8]. The bridge was one of four bridges under construction at the Maryland 198 crossing the Baltimore–Washington Parkway. The bridge was designed as a simple span post-tensioned concrete box girder bridge with the east abutment providing a fixed support and the west abutment providing a roller support. The superstructure was supported by timber formwork on steel longitudinal support beams which were supported by metal shoring as schematically shown in Figure 8. The case study was used to demonstrate the applicability and validity of the control system. Several possible failure initiators were selected to test whether they could have initiated the failure of the modelled system. This analysis was performed in an effort to identify several potential scenarios of events that could have triggered the failure mechanism. The bridge–deck construction (called the activity) was assumed to consist of two construction processes (I and II). Seven factors were considered for modelling these processes. The factors are stability, alignment, falsework condition and tower arrangement for process I, and concrete-placing practice, pouring sequence and finishing practice for process II. Each factor has three potential states. Therefore, there are 2 187 possible states for the construction activity. These states were investigated in the control system. The controller was tested by starting at several scenarios that might lead to failure, then the fuzzy controller produced the necessary actions to achieve a specified level of failure likelihood. Five combined states were then identified as failure initiators. The failure likelihoods of processes I and II as well as the activity failure likelihood were evaluated for each combined state as shown, for example, in Figure 9. A critical limit of 0.001 for the failure likelihood was assigned for the purpose of illustration. Therefore, three zones were identified, failure, critical and safe zones. If the failure likelihood lies in the failure zone, then the combined state might lead to failure. On the other hand, if the failure likelihood lies in the safe zone, the combined state should lead to a safe construction. The critical zone identify the

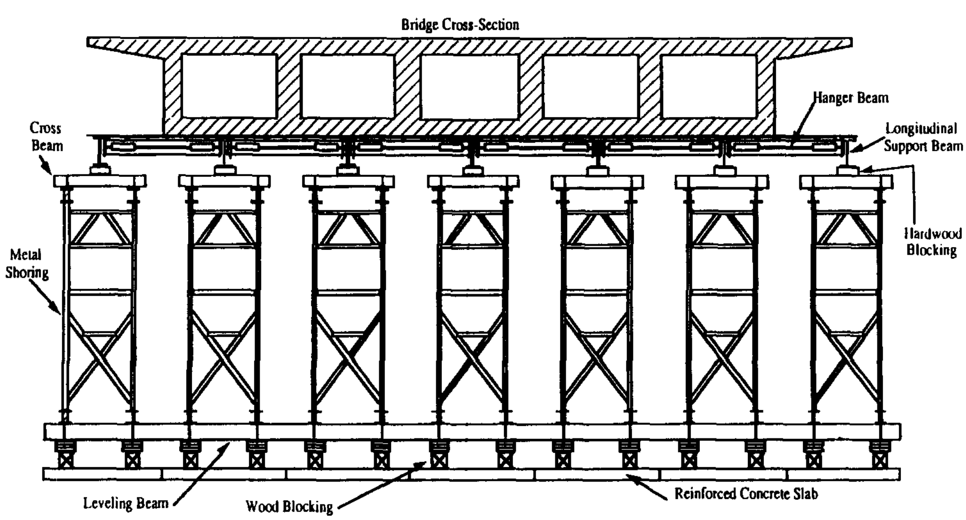

Fig. 8. Falsework system for the case study.

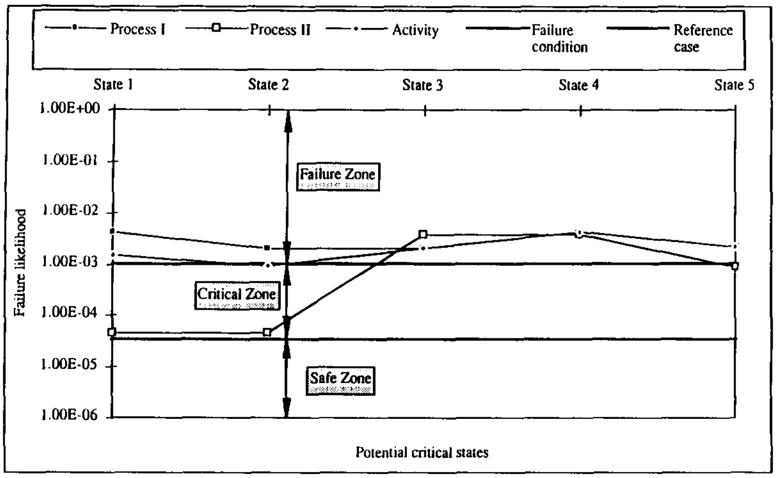

Fig. 9. Potential failure modes.

failure likelihood for states that might lead to failure due to any small variations in their conditions.

6. Summary and conclusions

In this paper, a fuzzy-based controller for construction activities was presented. The control system was developed for monitoring and controlling multiple attributes of interest at the same time. The system builds a model of the construction activity which emphasized critical properties. A condition assessment procedure that transfers a set of input states into output assessments was developed. The proposed procedure acts as an integral component of the control system. Fuzzy control essentially relies on subjective rules that are developed by experienced human controllers. This strategy proved very efficient for the safety control of construction activities because of their ill-defined nature. In other words, because these systems cannot by mathematically modelled, they do not lend themselves to the application of traditional control theory. However, fuzzy control strategy proved very successful in controlling this type of systems. If such a control system is operating on the activity during construction, it could detect a failure potential and adjust those variables that can be responsible for initiating a failure. The developed system is intended to act as a tool for human controlles where the controller is supposed to interactively operate the system using its suggestions and utilizing the information it provides in making other interactive decisions.

7. References

[1] B.M. Ayyub and M.H.M. Hassan, Control of construction activities: I. Systems identification, *Civil Engineering Systems* **9** (1992) 123–146.
[2] B.M. Ayyub and M.H.M. Hassan, Control of construction activities: II. Condition assessment of attributes, Civil Engineering Systems **9** (1992) 147–204.
[3] B.M. Ayyub and M.H.B. Hassan, Control of construction activities: III. Fuzzy-Based Controller, *Civil Engineering System* (1992) to be published.
[4] R.E. Cavallo and G.J. Klir, Reconstructability analysis: evaluation of reconstruction hypotheses, *Int. J. General Systems* **7** (1981) 7–32.
[5] R.E. Cavallo and G.J. Klir, Reconstructability analysis: overview and bibliography, *Int. J. General Systems* **7** (1981) 1–6.
[6] R.E. Cavallo and G.J. Klir, Decision making in reconstrucability analysis, *Int. J. General Systems* **8** (1982) 243–255.
[7] J. Efstathiou and V. Rajkovic, Multiattribute decision-making using fuzzy heuristic approach, *IEEE Transaction on Systems, Man, and Cybernetics,* vol. SMC-9 (6) (1979) 326–333.
[8] FHWA Borad or review, Report of the investigation into the collapse of the route 198 Baltimore–Washington parkway bridge, U.S. Department of Transportation, FHWA Publication No. PR-90-001 (1989).
[9] M.M. Gupta and J. Qi, Design of fuzzy logic controllers based on generalized t-operators, *Fuzzy Sets and Systems* **40** (1991) 473–489.

[10] E.T. Jaynes, Information theory and statistical mechanics, II., *Phys. Rev.* **108** (1957).
[11] B. Jones, Determination of reconstruction families, *Int. J. General Systems* **8** (1982) 225–228.
[12] G.J. Klir, *Architecture of Systems Problem Solving* (Plenum Press, New York, 1985).
[13] G.J. Klir and T.A. Folger, *Fuzzy Sets, Uncertainty, and Information* (Prentice Hall, Englewood cliffs, New Jersey, 1988).
[14] P.J. King and E.H. Mamdani, The application of fuzzy control systems to industrial processes, *Automatica* **13** (1977) 235–242.
[15] E.H. Mamdani, Perspective method for driving control policy in a fuzzy logic controller, *Electronics Letters* **11** (1975) 625–626.
[16] E.H. Mamdani, Advances in the linguistic synthesis of fuzzy controllers, *Int. J. Man–Machine Stud.* **8** (1976) 669–678.
[17] E.H. Mamdani, Application of fuzzy logic approximate reasoning using linguistic synthesis, *IEEE Transactions on Computers* C-**26** (12) (1977) 1182–1191.
[18] E.H. Mamdani and S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, *Internat. J. Man–Machine Stud.* **7** (1975) 1–13.
[19] M. Mizumoto, Note on the arithmetic rule by Zadeh for fuzzy conditional inference, *Cybernetics and Systems* **12** (1981) 247–306.
[20] E.D. Nering, *Linear Algebra and Matrix Theory* (John Wiley & Sons, New York, 1970).
[21] K. Nomizu, *Fundamentals of Linear Algebra* (McGraw Hill, Inc., New York, 1966).
[22] T.J. Procyk and E.H. Mamdani, A linguistic self-organizing process controller, *Automatica* **15** (1979) 15–30.
[23] J.S. Sokolnikoff and R.M. Redheffer, *Mathematics of Physics and Modern Engineering* (McGraw-Hill, New York, 1966).
[24] R.M. Tang, Analysis of fuzzy control algorithms using the relation matrix, *Internat. J. Man–Machine Stud.* **8** (1976) 679–686.
[25] R.M. Tang, An annotated bibliography of fuzzy control, in: M. Sugeno, Ed., *Industrial Applications of Fuzzy Control* (North-Holland, Amsterdam, 1985) 249–269.
[26] C. von Altrock, B. Karuse and H.-J. Zimmermann, Advanced fuzzy logic control of a model car in extreme situations, *Fuzzy Sets and Systems* **48** (1992) 41–52.
[27] D.J. Wilde and C.S. Beighler, *Foundations of Optimization* (Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1967).
[28] R.R. Yager, Implementing fuzzy logic controller using a neural network framework, *Fuzzy Sets and Systems* **48** (1992) 53–64.
[29] L.A. Zadeh, Fuzzy sets, *Information and Control* **8** (1965) 338–353.
[30] L.A. Zadeh, A rationale for fuzzy control, *Trans. ASME J. Dynamic Systems* **94** (1972) 3–4.
[31] H.-J. Zimmermann, *Fuzzy Set Theory – and its Applications,* second edition (Kluwer, Dordrecht-Boston, 1991).