FUZZY LOGIC CONTROL

MARZUKI KHALID*
Fakulti Kejuruteraan Elektrik
Universiti Teknologi Malaysia
Jalan Semarak
54100 Kuala Lumpur

SIGERU OMATU
Dept. of Information Science and Intelligent Systems
Faculty of Engineering
University of Tokushima
Tokushima 770, Japan.

Abstract

Although fuzzy logic theory was proposed about three decades ago, it is only until recently that fuzzy control technology has been successfully applied to many industrial systems and domestic appliances. Much of these developments is mainly due to a better understanding of the concept of fuzzy logic, its simplicity of implementation, and its feasibility in hardware development. This article intends to provide a simple but clear understanding on the concept of fuzzy logic and its application to control systems (therefore not intended for those who have already understood the basic principles of fuzzy logic control). A simulation example is given on the development and operation of a fuzzy logic controller for a multi-variable water bath temperature control system. C-pseudocodes are given in the Appendices to clarify the water bath fuzzy control algorithms.

1.   Introduction

In the past few years, the term "fuzzy" has dominated a wide variety of electrical and electronic products. From household appliances such as rice cookers, washing machines, video cameras, etc., to industrial systems such as automatic train operation and aircraft landing systems, the technology of fuzzy logic has been applied with success. Fuzzy logic technology has revolutionised the shape of the development of domestic and industrial products for many years to come. One reason for this is due to the creation of the "Institute of Fuzzy Control" in Japan where a wide variety of fuzzy logic based products have been successfully developed and commercialized.

What is fuzzy logic? The theory of fuzzy logic is not new and it was first conceived by Zadeh in the seminal paper [1] entitled "fuzzy sets" about three decades ago. The term "fuzzy set" was introduced to offer a classification for objects encountered in the real physical world which is both imprecise and uncertain. The key idea is to develop a framework to deal with such imprecision and uncertainty. Instead of using the ordinary concept of set inclusion, Zadeh introduced a function that expressed the degree of belonging to a given set as a function taking values in the range of 0 to 1. The relevant aspects of fuzzy logic as introduced by Zadeh is given in the next section. A more detail explanation can be found in [1, 2].

---

* *Currently attached to the Dept of Information Science and Intelligent System, Faculty of Engineering, University of Tokushima, Tokushima 770, Japan.*

When fuzzy logic was first conceived, it was expected that most of its applications would be in the realm of knowledge-based systems in which the resident information is both imprecise and uncertain. Contrary to this expectation, however, most of the successful applications of fuzzy logic at this juncture relate to control and systems analysis in which there is imprecision but no uncertainty. In the field of control, fuzzy logic technology has been applied with varying success to a wide variety of systems. One notable application is the auotmatic train operation (ATO) system in the city of Sendai, Japan since 1987 [3]. The Sendai ATO system's train automatic stop control (TASC) subsystem controls the speed of the train in view of safety, riding comfort, traceability of target velocity, accuracy of stop gap, running time and energy consumption. This ATO system performs as skillfully as human experts do and superior to an ordinary proportional-plus-integral-plus-derivative (PID) automatic train operation controller.

The idea of applying fuzzy logic to control systems was first conceived by Assilian and Mamdani [4]. Based on Zadeh's fuzzy set theory and the simple conventional PID controller, Assilian and Mamdani developed what is now referred to as the basic fuzzy logic controller, which is used to regulate the outputs of a process around a given set-point using a digital computer. Due to the limitations of memory space and speed of the sequential Von Neumann computers, fuzzy logic controllers were not very popular in the beginning and much of their applications centred around slow varying processes. Some of the earlier applications of fuzzy logic to process control can be found in [5] and [6].

There are a number of advantages of applying fuzzy logic to the control of industrial processes over traditional controllers. Perhaps one of the main advantages of applying fuzzy logic control is that a controller can be developed along linguistic lines which has close associations with the field of artificial intelligence (AI). One of the aims of AI is to replace\human beings carrying out precise tasks by machines and hence the link between AI and control theory is strong. The fuzzy controller consists of a set of linguistic conditional statements (fuzzy associations) or rules which define the individual control situations. These linguistic conditional statements or control rules can be easily developed from common sense or from engineering judgement of the process to be controlled.

Many industrial processes are difficult to be controlled accurately and it has been claimed that fuzzy logic control can deal successfully with such processes which are usually multi-variable in nature and have unknown dynamics, nonlinear behaviour, and time-varying parameters. Compared to traditional adaptive controllers, fuzzy controllers have a key advantage in that they do not require a priori mathematical model of the plant. Compared to neural network controllers, basic fuzzy controllers do not have to be trained before they can be used. Based on common sense and engineering judgement of the process, fuzzy logic controllers can be developed in minutes. Another clear advantage is that fuzzy logic controllers can now be feasibly implemented in digital [7] or analog [8] VLSI circuitry where sampled information can be encoded in a parallel-distributed framework.

The main objective of this article is to provide the reader with the understanding on how to apply the concept of fuzzy logic to control systems on a microcomputer. An overview of the theory of fuzzy sets and fuzzy systems from the viewpoint of mappings between hypercubes is first introduced. The concept of the basic fuzzy logic controller is next discussed. The rest of this paper deals with the development of a fuzzy logic water bath temperature control system. As it is the intention of the author to provide a clear viewpoint to the reader, assuming that the reader has no basic knowledge, a rather detail explanation is given on the implementation and simulated operation of the fuzzy logic controller for the multi-variable water bath process. For clarity, C-codes for the water bath fuzzy associations or rules, the computation of the fuzzy membership functions and the fuzzy centroid defuzzification procedures are given in the Appendices.

2. A Brief Overview of Fuzzy-Set and Fuzzy-System Theory

Fuzzy-set theory is an active research area and some of its aspects have been discussed in the following literatures [9-11]. In this section a brief overview of fuzzy-set theory is described from the viewpoint of between-cube theory [12].

A fuzzy set defines a point in a cube whereas a fuzzy system defines a mapping between these cubes. A fuzzy system S maps fuzzy sets to fuzzy sets. Thus a fuzzy system S is a transformation from one hypercube I^n to another hypercube I^p

$$S: I^n \rightarrow I^p \tag{1}$$

The n-dimensional unit hypercube I^n houses all the fuzzy subsets of the domain space or input universe of discourse and I^p houses all the fuzzy subsets of the range space or output universe of discourse. In general a fuzzy system S maps families of fuzzy sets to families of fuzzy sets, thus,

$$S: I^{n_1} \times ... \times I^{n_r} \rightarrow I^{p_1} \times ... \times I^{p_r} \tag{2}$$

These fuzzy systems behave as associative memories where they map close inputs to close outputs. They are referred to as fuzzy associative memories (FAM). The simplest FAM encodes the FAM rule or association (A_i, B_i), which associates the p-dimensional fuzzy set B_i with the n-dimensional fuzzy set A_i. These minimal FAMs essentially map one ball in I^n to one ball in I^p.

Three common associations in fuzzy set theory [1] are 'complement', 'union', and 'intersection' of fuzzy sets. The complement of a fuzzy set A is denoted by A' and is defined by

$$m_{A'}(x) = 1 - m_A(x), \quad x \in X \tag{3}$$

where X is the real line R^1 or universe of discourse and $m_A(x)$ is the membership function of the fuzzy set A between 0 and 1.

The union of two fuzzy sets A and B with respective membership functions $m_A(x)$ and $m_B(x)$ is a fuzzy set C, written as $C = A \cup B$, whose membership function is related to those of A and B by

$$m_C(x) = Max[m_A(x), m_B(x)], \quad x \in X \tag{4}$$

or in abbreviated form

$$m_C = m_A \vee m_B. \tag{5}$$

The intersection of two fuzzy sets A and B with respective membership functions $m_A(x)$ and $m_B(x)$ is a fuzzy set C, written as $C = A \cap B$, whose membership function is related to A and B by

$$m_C(x) = Min[m_A(x), m_B(x)], \quad x \in X \tag{6}$$

or in abbreviated form

$$m_C = m_A \wedge m_B \qquad (7)$$

In simple words, the union of A and B is the smallest fuzzy set containing both A and B, whereas the intersection of A and B is the largest fuzzy set which is contained in both A and B.

In general, a FAM system $F: I^n \rightarrow I^p$ encodes and processes in parallel a FAM bank of m FAM rules $(A_1, B_1), ..., (A_m, B_m)$ as shown in Fig. 1. Each input A to the FAM system activates each stored FAM rule to different degree. The minimal FAM that stores the ith association or rule (A_i, B_i) maps input A to B_i', which is a partially activated version of B_i. The more A resembles A_i, the more B_i' resembles B_i. The corresponding output fuzzy set B combines these partially activated fuzzy sets $B_i' ... B_m'$ where it equals a weighted average of the partially activated sets:

$$B = \sum_{i=1}^{m} w_i B_i' \qquad (8)$$

The term w_i reflects the frequency or strength of the fuzzy association (A_i, B_i). In practice the output waveform B can be "defuzzified" to a single numerical value y_j in Y by computing the fuzzy centroid of B with respect to the output universe of discourse Y.
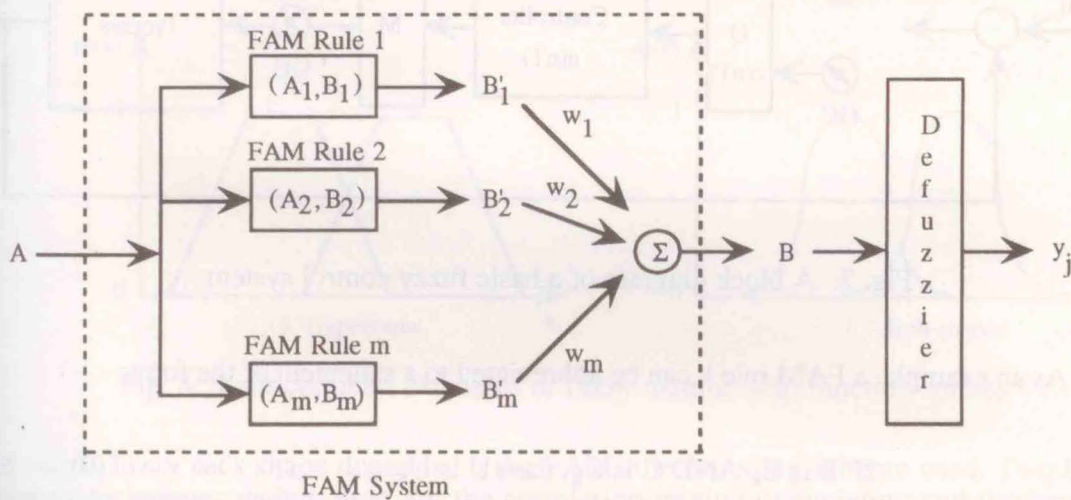

Fig. 1: A FAM system architecture. The FAM associations or rules (1 to m) are fired in parallel to produce the output fuzzy set B. B is "defuzzified" to a numerical value y_j.

3. The Concept of the Basic Fuzzy Logic Controller

Fuzzy logic control systems are proposed where they can be developed along linguistic lines which has some close associations with the field of artificial intelligence. Based on Zadeh's theory of fuzzy sets as discussed the concept of fuzzy logic has been applied successfully to the control of industrial processes particularly those which are ill-defined and those capable of successful human-operator regulation.

A simple fuzzy logic control system as shown in Fig. 2, originally conceived by Assilian and Mamdani [4], consists of a set of linguistic conditional statements or rules (referred to FAM rules) which define individual control situations. In many applications, it is sufficient to relate

the error and the change of error in the process to apply the correct amount of change in the control input to satisfactorily control the system. In this respect simple linguistic rules can be formulated based on observation or a simple study about the process. An example of a linguistic natural language rule that an expert might use to describe a control system action is as follows:

If error is positive large and change in error is negative small,
then change in the process input is negative large.

These three variables, error (E), change in error (C), and change in control input (U) are defined by fixed universes of discourse which defined the range of measurements they can take in a particular control system. In some applications [12], another variable such as the past control input value may be added as the antecedents in the FAM rule to improve the system performance, however, at the expense of added complexity in the development of the fuzzy control system.

Each of the fuzzy variables can be quantized into fuzzy subsets according to the complexity of the process. Linguistic terms such as positive large, negative small, zero, etc. in which an expert might use to describe the control system actions are used to label these fuzzy subsets which in reality represent numerical values. Each of these fuzzy subsets contain elements with degrees of membership. A fuzzy membership function assigns a real number generally between 0 and 1 to every element in its universe of discourse which indicates the degree to which the object or data belongs to that fuzzy set.
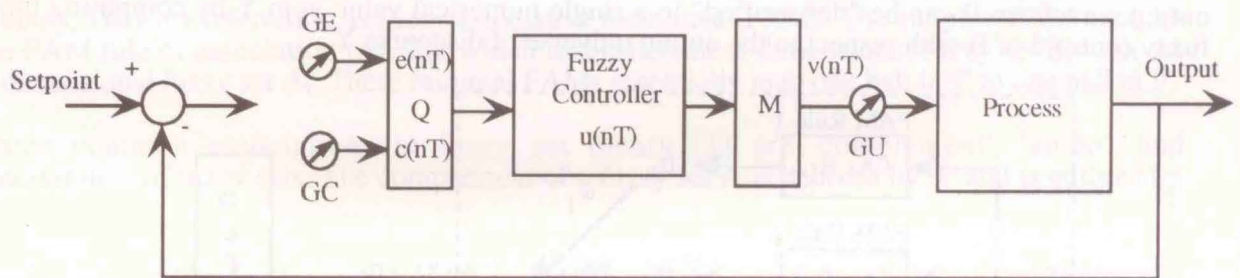

Fig. 2: A block diagram of a basic fuzzy control system

As an example, a FAM rule k can be abbreviated to a statement of the form:

$$\text{IF } E \text{ is } E_k \text{ AND } C \text{ is } C_k, \text{ then } U \text{ is } U_k \qquad (9)$$

These fuzzy subsets E_k, C_k and U_k are defined by a set of ordered pairs as follows:

$$E_k = \{(e, m_{e_k}(e))\} \subset E$$
$$C_k = \{(c, m_{c_k}(c))\} \subset C$$
$$U_k = \{(u, m_{u_k}(u))\} \subset U \qquad (10)$$

where e, c, and u are elements of the discrete universes and m_{e_k}, m_{c_k} and m_{u_k} are the corresponding membership values which give the degree to which the element is a member of the fuzzy subset.

Fuzzy membership functions can have different shapes, such as monotonic, triangular, trapezoidal, and gaussian, according to the designer's preference or experience. Figure 3 shows

24

four common examples of membership function shapes or splines. Selection of the type of membership functions affects the type of reasoning to be performed [13]. In practice trapezoidal or triangular shapes are widely used which simplify computation.

The antecedents of all the FAM rules which describe the process control action are activated in parallel (as shown in Fig. 1) where the pulse nature of the inputs picks off single fit values of the fuzzy variables. The antecedent fit values can be combined with either minimum or maximum depending on whether the antecedent fuzzy sets are combined with the conjunctive AND or disjunctive OR. The example as described above, Eq. (9), combined the antecedents with the conjunctive AND which activates the consequent fuzzy set U_k to degree w_k, therefore using Zadeh's fuzzy intersection association of Eq.(7), we have

$$w_k = m_{e_k}(e) \wedge m_{c_k}(c). \tag{11}$$

(a). Monotonic

(b). Triangular
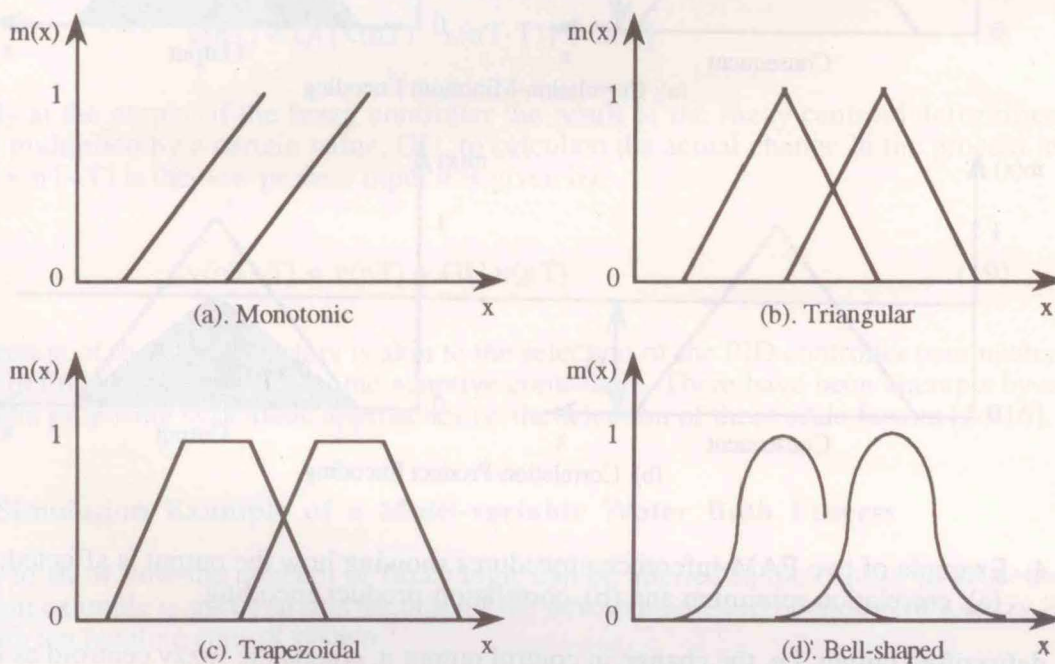
(c). Trapezoidal

(d). Bell-shaped

Fig. 3: Four common examples of fuzzy membership function splines

The output fuzzy set's shape depends on the FAM-rule encoding scheme used. Two types of encoding techniques widely used are the correlation-minimum encoding and thecorrelation-product encoding. With correlation-minimum encoding, the consequent fuzzy set U_k in the library of the output fuzzy set values is clipped to degree w_k with pointwise minimum with :

$$m_{O_k}(y) = w_k \wedge m_{u_k}(y) \tag{12}$$

and with correlation-product encoding, we multiply U_k by w_k:

$$m_{O_k}(y) = w_k \, m_{u_k}(y) \tag{13}$$

where $m_{O_k}(y)$ is the degree of membership of the activated output fuzzy set at interval y in the output universe of discourse. Figure 4 shows an example of how the output is affected when (a) correlation-minimum and (b) correlation-product encoding techniques are used.

In practice, $m_{O_k}(y)$ is not unique, the system then combines all of these functions as their union to yield the actual output fuzzy membership function, $m_O(y)$, following Zadeh's fuzzy union association of Eq. (5) such that:

25

$$m_O(y) = m_{O_1}(y) \vee m_{O_2}(y) \vee \ \ldots\ldots\ \vee m_{O_k}(y) \qquad (14)$$

Note that k in this case is not the number of rules, it is however, the quantization of the output fuzzy set into its fuzzy subsets.

(a). Correlation-Minimum Encoding

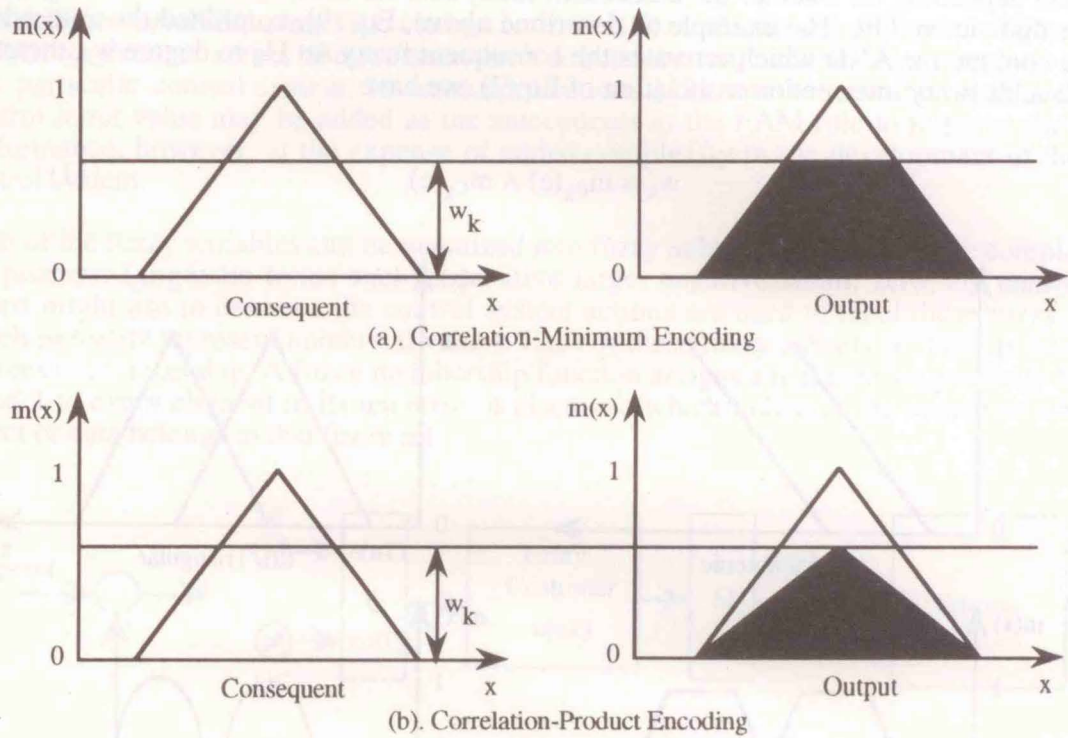

(b). Correlation-Product Encoding

Fig. 4: Example of two FAM-inference procedures showing how the output is affected: (a). correlation-minimum and (b). correlation-product encoding

The defuzzified output, i.e. the change in control output u, equals the fuzzy centroid as follows:

$$u = \frac{\int y m_O(y)\,dy}{\int m_O(y)\,dy} \qquad (15)$$

where the limits of the integration correspond to the entire universe of discourse Y. Alternatively, in order to reduce computation, Y can be discretized to p values at regular interval, Δy, such that, $Y = \{y_1, y_2, \ldots y_p\}$ which gives the discrete fuzzy centroid:

$$u = \frac{\sum\limits_{j=1}^{p} y_j m_O(y_j)}{\sum\limits_{j=1}^{p} m_O(y_j)} \qquad (16)$$

26

Another important aspect of fuzzy logic control systems, which cannot be overlooked, is that of scale factors selection. Like all controllers, a fuzzy logic controller has a number of parameters· which must be chosen by the designer in prior. In the fuzzy controller, there are three scale factors, GE, GC, and GU for the process error (E), change in error (C), and the controller's output (U), respectively. For clarity, the block diagram shown of Fig. 2 provides some indication on the execution of these scale factors in a fuzzy control system.

The values of error and change in error used by the controller are elements of the corresponding universes of discourse which can be represented in time by e(nT) and c(nT) where T is the sampling period and n is the sample number. As these values have to be quantized to the closest element of the universe of discourse, they first have to be scaled by multiplying them by a suitable value. If the process output and the set-point at the nth sample are x(nT) and S(nT), respectively, then e(nT) and c(nT) are obtained from the following equations:

$$e(nT) = Q[\{S(nT) - x(nT)\}.(-GE)] \qquad (17)$$
$$c(nT) = Q[\{x(nT) - x(nT-T)\}.(-GC)] \qquad (18)$$

Similarly at the output of the fuzzy controller the result of the fuzzy centroid defuzzification must be multiplied by a certain value, GU, to calculate the actual change in the process input. Thus, if v(nT+T) is the new process input it is given by:

$$v(nT+T) = v(nT) + GU.u(nT) \qquad (19)$$

The selection of these scale factors is akin to the selection of the PID controller parameters and the user defined polynomials of some adaptive controllers. There have been attempts by some researchers proposing systematic approaches on the selection of these scale factors [14-16].

4.   Simulation Example of a Multi-variable Water Bath Process

In order to show how the concept of fuzzy logic can be applied to the control of processes, a simulation example is given in this section on the development and operation of a fuzzy logic water bath temperature control system.

4.1   Development of the Fuzzy Logic Water Bath Control System

The water bath is an example of an important component in many industrial chemical process control systems and a simple schematic diagram is shown in Fig. 5. It is used to mix liquid compounds in the bath and to automatically control the temperature of the mixture at different levels as accurately as possible to ensure that the final product is well formed. Solid state relays are used to regulate the signals between 0V to 5V to the heaters. The temperatures of the mixture in the bath are measured using thermocouples in the range of 0°C to 100°C which are transformed into corresponding voltage signals in the range of 0V to 10V. The sampling time of the process is 30 seconds. The process can be modelled in discrete-time as follows:

$$A(z^{-1}) y(k) = B(z^{-1}) u(k-1) + \xi(k), \qquad k=nT \qquad (20)$$

where

$$A(z^{-1}) = I + A_1 z^{-1} \text{ and } A_1 = \begin{bmatrix} -0.411 & -0.634 \\ -0.103 & -0.885 \end{bmatrix}$$

$$B(z^{-1}) = B_0 = \begin{bmatrix} 0.492 & 0.085 \\ 0.041 & 0.237 \end{bmatrix}$$

27

$y(k)$ is the process output, $u(k-1)$ is the control input and $\xi(k)$ is the uncorrelated Gaussian noise with zero mean and covariance r_ξ. It can be observed that the dynamics of the process is coupled in nature.
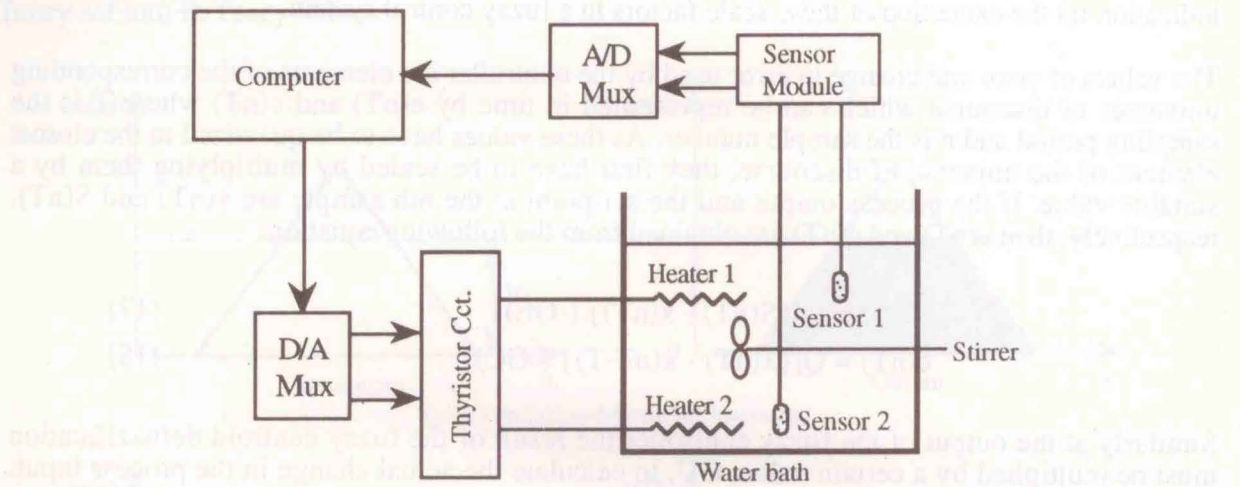

Fig. 5: A schematic diagram of the multivariable waterbath temperature control system

In order to develop the fuzzy controller, the input and output variables of the water bath must first be specified. The input variables are the performance error, θ, i.e. the error between the desired output and the actual plant output, and the rate of change of the performance error $\dot\theta$ However, in the experiments, the instantaneous rate of change of error can be approximated as the difference between the present error θ_k and the previous error θ_{k-1}, such that

$$\Delta\theta = \theta_k - \theta_{k-1}$$

The output or the control fuzzy variable is the voltage signal to the heater, v, where the range of the voltage is between 0V and 5V as described.

We quantized the three fuzzy variables into seven fuzzy subsets as shown in Table 1 from negative large (LN) to zero (ZE) to positive large (LP). In practice these quantizations can be expanded according to the complexity of the control problem. For clarity the FAM-rules i.e. the conditional statements which describe the control actions of the process can be specified in matrix form as shown in Fig. 6. The antecedents are the conditions of the error and change in error in the process (given on the two axes) whereas the consequent is the amount of change in the control input to be applied (given in each box in the matrix). Common sense and engineering judgement dictate the entries in the water bath FAM bank matrix where a total of 24 rules are formulated (see Appendix I).

LN	Large Negative
MN	Medium Negative
SN	Small Negative
ZE	Zero
SP	Small Positive
MP	Medium Positive
LP	Large Positive

Table 1: Fuzzy set values for the three fuzzy variables
of the water bath control system

Each bank in the matrix constitute one FAM association or rule. Each rule describes how to modify the control variable v for observed values of the water bath input fuzzy variables θ and $\Delta\theta$. As an example, we can interpret FAM rule no. 7 in natural language as follows:

If the error θ in the water temperature is negative and large, and the rate of change of the error $\Delta\theta$ is about zero, then the heater v should be on positive and large.

or it can be expressed symbolically as follows:

$$IF\ \theta = LN\ AND\ \Delta\theta = ZE\ ,\ THEN\ v = LP$$

or more compactly, the triple:

$$(LN, ZE; LP).$$

Note that it is not necessary to fill up all the FAM banks. In general, some rules can be omitted and added according to the complexity of the control problem.

Error, θ

Δθ \ θ	LN	MN	SN	ZE	SP	MP	LP
LN	1 LP	2 LP	3 LP	4 LP			
MN	5 LP	8 MP	9 MP	10 MP			
SN	6 LP	11 MP	13 SP	14 SP	17 SN		
ZE	7 LP	12 MP	15 SP	16 ZE	18 SN	21 MN	23 LN
SP			19 SN	20 SN			
MP				22 MN			
LP				24 LN			

Change in Error, Δθ

Fig. 6: The FAM bank matrix for the multivariable water bath system. The antecedents are the error and change in error and the consequent of each rule is given in the box.

We chose triangular and trapezoidal membership functions for the fuzzy subsets of the three fuzzy variables as shown in Fig. 7. It can be observed that the ZE (zero) fuzzy subset of the three fuzzy variables are narrower to permit fine control near the setpoints. For simplicity, we may use the same shape for all the membership functions of the fuzzy subsets where they correspond to shifted versions of a single fuzzy set ZE. Using heuristic rules, the contiguous fuzzy subsets in each library overlapped about 25 per cent. Too much overlap blurs the distinction between the fuzzy-set values. Too little overlap tends to resemble bivalent control, producing excessive overshoot and undershoot. In reality, the overlap helps smooth the transition of the control action during the operation of the controller.

Correlation-minimum encoding technique is used for the inference procedure of the fuzzy logic water bath control system.The discrete fuzzy centroid defuzzification procedure is used to defuzzify the fuzzy control outputs into real values.

4.2   Fuzzy Controller Implementation

In this simulation, we wish to control the temperature of the water at two different levels in the bath: upper level and lower level. The set-points for the 100-sample duration experiment are given as follows:

Water Bath Level	Sample Number	Set-points
Upper level	$0 \leq n \leq 50$	50°C
	$50 < n \leq 100$	70°C
Lower level	$0 \leq n \leq 50$	45°C
	$50 < n \leq 100$	65°C

30

By simple trial and error procedure, we select the scaling parameters for the levels as follows:

Upper level:	GE = 0.14	GC = 0.62	GU = 4.0
Lower level:	GE = 0.12	GC = 0.66	GU = 4.0

These values can be fined tuned by performing several experiments to obtain the optimum performance of the system.
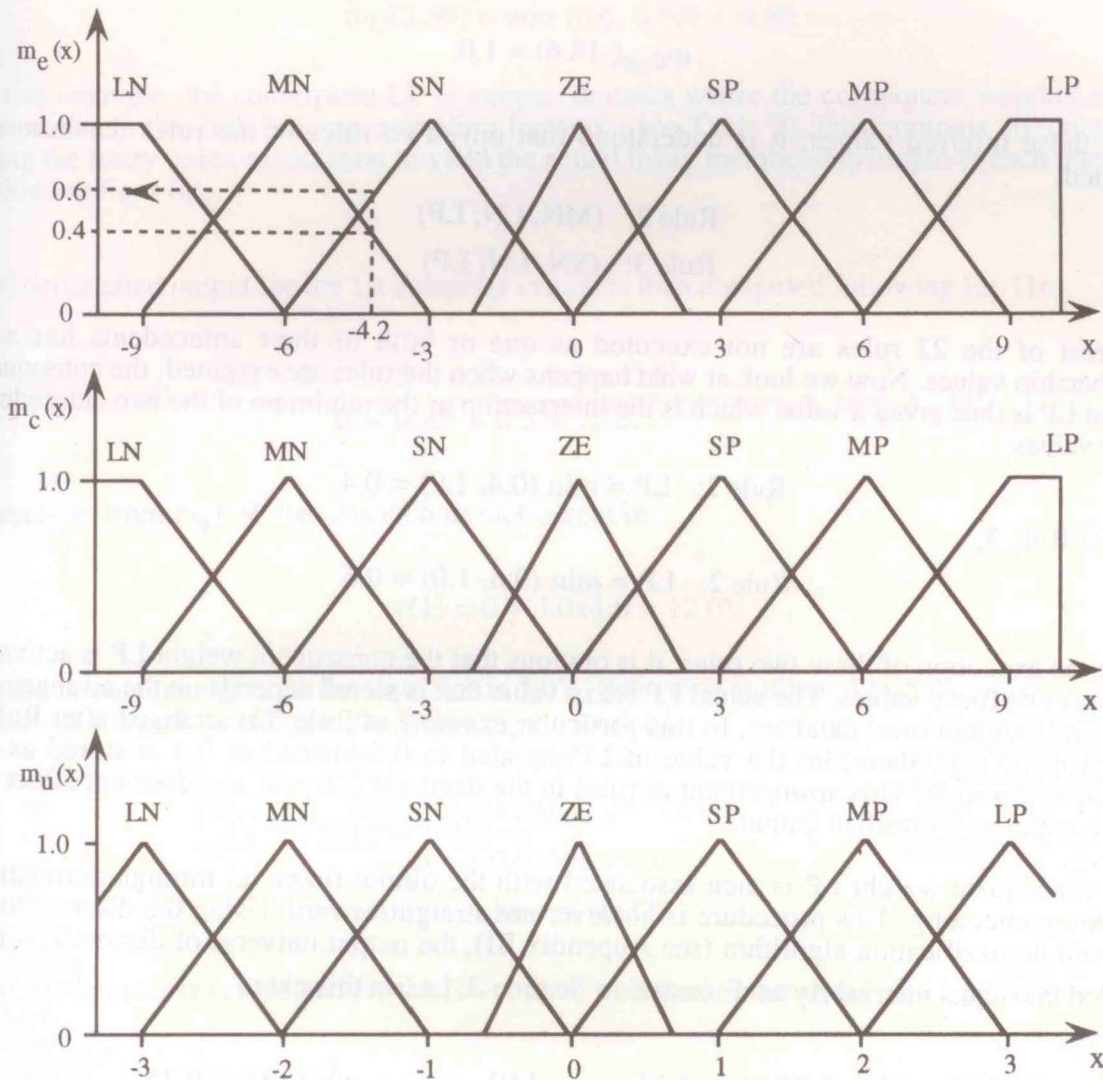

Fig.7: Membership functions of the three fuzzy variables of the water bath system. It can be observed that when the quantized error equals -4.2 on the universe of discourse, it is a member of two fuzzy subsets, i.e. MN and SN.

To descibe in more detail the operation of the fuzzy controller, we observe at what happens during the first sampling instant. For brevity only the upper level control is described. The first setpoint of the upper level is given at 50°C therefore as the initial temperature of the water in the bath is 20°C, a large process error is present. The fuzzy controller operates by inferencing the quantized values of the process error and change in error to compute the change in the process input. The error and change in error are quantized into their corresponding universes of discourse by multiplying the pre-selected respective scale factors following Eqns. (17) and (18). Thus at the first sampling instant (for simplicity we omit the sampling period, T):

$$e(1) = (50-20) \times (-0.14) = -4.2 \qquad\qquad c(1) = (30-0) \times (-0.62) = -18.6$$

31

From Fig. 7 it can be observed that e(1) is a member of two fuzzy subsets, i.e. MN and SN and their membership values are:

$$me_{MN}(-4.2) = 0.4 \text{ and } me_{SN}(-4.2) = 0.6$$

whereas c(1) is clearly a member of the LN fuzzy subset:

$$mc_{LN}(-18.6) = 1.0.$$

From these inferred values, it is understood that only two rules in the rules database are executed:

Rule 2: (MN, LN; LP)

Rule 3: (SN, LN; LP)

The rest of the 22 rules are not executed as one or both of their antecedents has zero membership values. Now we look at what happens when the rules are executed; the consequent weight LP is thus given a value which is the intersection or the minimum of the two antecedents fuzzy values:

Rule 2: LP = min (0.4, 1.0) = 0.4

and for Rule 3,

Rule 2: LP = min (0.6, 1.0) = 0.6.

From the execution of these two rules, it is obvious that the consequent weight LP is activated but with two fuzzy values. The actual LP fuzzy value that is stored depends on the arrangement of the rules in the rules database. In this particular example as Rule 3 is arranged after Rule 2 (see Appendix I), therefore the value of LP equaled to 0.6 instead of 0.4 is stored as the consequent weight. This arrangement of rules in the database is trivial and does not affect the final computed defuzzified output.

This consequent weight LP is then associated with the output fuzzy set through correlation-minimum encoding. This procedure is however not straightforward. Using the discrete fuzzy centroid defuzzification algorithm (see Appendix III), the output universe of discourse is first divided into equal interval Δy as discussed in Section 3, i.e., in this case:

$$Y = \{4.0, 3.75, 3.5, 3.25, ..., -4.0\} \qquad \text{where } \Delta y = 0.25.$$

At each discrete interval along the output universe of discourse, correlation-minimum encoding clipped the output fuzzy set to degree 0.6 (LP) with pointwise minimum following Eq. (12). Several examples of these values are computed as follows:

$$m_O(4.00) = \min (0.6, 0.00) = 0.00$$
$$m_O(3.75) = \min (0.6, 0.25) = 0.25$$
$$m_O(3.50) = \min (0.6, 0.50) = 0.50$$
$$m_O(3.25) = \min (0.6, 0.75) = 0.60$$
$$m_O(3.00) = \min (0.6, 1.00) = 0.60$$
$$m_O(2.75) = \min (0.6, 0.75) = 0.60$$
$$m_O(2.50) = \min (0.6, 0.50) = 0.50$$

In this example, the consequent LP is unique. In cases where the consequent weights are not unique (as in the 2nd, 3rd, etc. sampling instants - see Table 2), these outputs are combined using the fuzzy union association to yield the actual fuzzy membership output at each interval y following Eq. (14).

The defuzzified output for the 1st sampling instant is then computed following Eq. (16):

$$U = \frac{4.0 \times 0 + 3.75 \times 0.25 + 3.5 \times 0.5 + \ldots\ldots}{0 + 0.25 + 0.5 + \ldots\ldots} = \frac{9.9}{3.3} = 3.0$$

Therefore, from Eq.(19) the actual controller output is:

$$v(1) = 0 + 3.0 \times 4.0 = 12.0V$$

As the maximum signal to the thyristor is 5.0V, this value is clipped to 5.0V maximum. The value v is clipped according to the following constraints:

$$\text{if } v \geq 5.0 \text{ then } v = 5.0V$$
$$\text{if } v \leq 0.0 \text{ then } v = 0.0V$$
$$\text{else } v = v(nT)$$

The whole process is repeated for the lower level waterbath control and also for the subsequent samples.

Table 2 shows an example of rules and consequent weights which are activated for the upper level waterbath control for several sampling instants. It can be observed that the consequent weights are activated at different degrees at each sampling instant over the entire experimental cycle. For example at the 8th sample, the consequent weight is more of SP and around steady state, at the 49th sample, the consequent weight is a strong ZE fuzzy subset. Figure 8 shows the performance of the fuzzy controller on the water bath over the entire cycle.

In many real-time applications this entire FAM¹inference procedure is repeated hundreds, perhaps thousands, of times per second. This may require fuzzy VLSI or optical processors for parallel processing where the fuzzy controller can operate at very fast speed. The application of fuzzy control using a microcomputer to a real-time water bath process with comparison to a conventional PI controller can be found in [17]. It was observed that the fuzzy logic based water bath system performed better than the conventional PI water bath system with respect to

set-point changes, load disturbance rejection, changes in deadtime and the dynamics of the process.

	Quantized error, e(nT)	Quantized change in error, c(nT)	Rules Executed	Consequents Activated	Scaled Defuzzified Output, v(nT)
1	-4.2	-18.0	2,3	LP=0.6	5.0
2	-4.074	0.558	12,15,19	MP=0.358 SP=0.642 SN=0.186	5.0
8	-2.189	1.006	15,16,19,20	SP=0.567 ZE=0.059 SN=0.058	5.0
16	-0.417	0.921	15,16,19,20	SP=0.139 ZE=0.604 SN=0.307	3.57
32	-0.003	-0.002	13,14,15,16	SP=0.001 ZE=0.999	1.33
49	0.0	0.0	15,16,19,20	ZE=1.0	1.33
50	-2.8	-12.4	3	LP=0.933	5.0

Table 2: Some examples of rules and consequent weights which are activated at various sampling instants for the upper level water bath control.

5.   Concluding Remarks

Fuzzy logic controllers offer a key advantage over many traditional controllers. They can be developed along linguistic lines where human intelligence and common sense engineering judgement can play an important role. Similar to human operators, fuzzy logic controllers do not need precise prior information for implementation as much as that desired of algebraic controllers. Many complex industrial processes are ill-defined and difficult to be modeled accurately. Fuzzy logic controllers do not need mathematical models of plants in order to be implemented where such information is a prior necessity for many traditional adaptive controllers (see [18]).

Compared to neural network controllers, basic fuzzy logic controllers do not need to be trained in prior. We have also compared a neural network controller to that of a basic fuzzy logic controller with application to a real-time water bath temperature control system [19]. It was found that the fuzzy controller performed just as well and furthermore it offers the advantage of simpler implementation. In addition the fuzzy controller is computationally lighter than neural network and many traditional adaptive controllers. Most computational operations only involved comparing and adding two real numbers.

This paper has been written with the intention of providing detail explanation on the development and application of basic fuzzy logic control. Some basic concepts of fuzzy sets and

systems theory have also been discussed. The state of the art research in fuzzy logic control is the development of adaptive fuzzy control systems which can compose and decompose the rules automatically. Some examples of these adaptive fuzzy control systems can be found in [12,13, 15 and 20].

Is the technology of fuzzy logic more of just a fad which will soon disappear? The answer is no and Japan has made sure of that. Today, perhaps unconciously (concious enough to the intellectuals but maybe unconcious to the laymen) we are already making use of equipments and domestic products that use the technology of fuzzy logic.

References

[1]   Zadeh, L.A. Fuzzy sets. *Information and Control,* **8**, 338-352, (1965).

[2]   Zadeh, L. A., Outline of a new approach to the analysis of complex systems and decision processes. *IEEE trans on Systems, Man and Cybernetics,* **3**, pp.28-44, (1973).

[3]   Langari, R. and Berenji, H. R. Fuzzy logic in control engineering. *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches.* Eds. D.A. White and D.A. Sofge, Van Nostrand Reinhold, NY, USA, (1992).

[4]   Assilian, S. and Mamdani, E.H. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man-Mach. Stud.,* **7**, pp.1-13, (1974).

[5]   Tong, R.M., A control engineering review of fuzzy systems. *Automatica,* **13**. pp. 559-569, (1977).

[6]   King, P.J. and Mamdani, E.H. The application of fuzzy control systems to industrial processes. *Automatica,* **13**. pp. 235-242, (1977).

[7]   Togai, M. and Watanabe, H. Expert system on a chip: an engine for realtime approximate reasoning. IEEE Expert, **1**, 3, (1986).

[8]   Yamakawa, T. Fuzzy microprocessors - rule chip and defuzzification chip. *Proc. of Intl. Workshop on Fuzzy systems Applications.* Iizuka-88, Kyushu, 51-52, (1988).

[9]   Zimmermann, H. *Fuzzy Set Theory - and its Applications.* 2nd Ed., Kluwer, Boston, (1990).

[10]   Klir, G. and Folger, T. *Fuzzy Sets: Uncertainty and Information.* Prentice Hall, Englewood Cliffs, N.J., (1988).

[11]   Dubois, D. and Prade, H. *Fuzzy Sets and Systems: Theory and Applications.* Academic Press, N.Y., (1980).

[12]   Kosko, B. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence.* Prentice Hall, Englewood Cliffs, N.J., (1991).

[13]   Berenji, H.R. An architecture for designing fuzzy controllers using neural networks. *Int. J. of Approx. Reasoning.* **6**, (2), pp. 267-292, (1992).

[14]   Linkens, D.A. and Abbod, M.F. Self-organising fuzzy logic control and the selection of its scaling factors. *Trans Inst MC,* **14**, 3, pp. 114-125, (1992).

[15]   Procyk, T.J. and Mamdani, E.H. A linguistic self-organizing process controller. *Automatica,* **15**. pp. 15-30, (1979).

[16] Daley, S. and Gill, K. F. A design study of a self-organising fuzzy logic controller. *Proc. I. Mech E*, **200**, (CI), pp.59-69, (1986).

[17] Khalid, M. Fuzzy temperature control of a water bath process (submitted to *IEM Jurnal*, 1992).

[18] Yusof, R., Omatu, S. and Khalid, M. Temperature control of a water bath by self-tuning PID controller. (submitted to *Int. J. of Systems Science*, 1992).

[19] Khalid, M. and Omatu, S. Comparison of neural, fuzzy and conventional water bath temperature control systems. (submitted to *Int. J. of Eng. Appl*, 1993).

[20] Moore, C.G. and Harris, C.J. Indirect adaptive fuzzy control. *Int. J. Control*. **56**, (2), pp. 441-468, (1992).

Marzuki Khalid is a Lecturer at the Dept. of Control Engineering, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia. He received the B.Sc. (Hons) degree in Electrical Engineering in 1983 and M.Sc. in Control Systems in 1986 from Southampton University and Cranfield Institute of Technology, United Kingdom, respectively. Currently, he is completing his Ph.D at the Dept. of Information Science and Intelligent Systems, University of Tokushima, Japan concentrating in the area of neuro-control. His research interest include control theory and artificial intelligence. He was involved in consultancy works with Pallmilltech (M) Sdn. Bhd., Perkhidmatan & Pemborong Kejuruteraan (M) Sdn. Bhd. and Perwaja (M) Sdn. Bhd. in areas of microprocessor/microcomputer controlled system. He has authored and co-authored over forty papers in journals and conferences. In 1980, he received a silver medal from the National Electricity Board for the Best Student Award. He has held the post of Deputy Chairman of the IEEE Malaysian Chapter. He is a member of the IEEE and the Int. Neural Networks Society, USA.
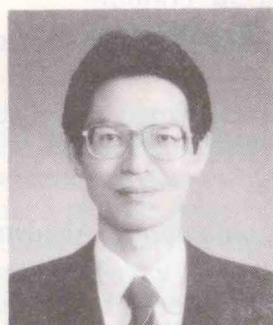
Sigeru Omatu was Born in Ehime, Japan, on December 16, 1946. He received the B.E. degree in Electrical Engineering from the University of Ehime, Japan, in 1969, and the M.E. and Ph.D. degrees in Electronic Engineering from the University of Osaka Prefecture, Japan, in 1971 and 1974, respectively.

From 1974 to 1975 he was a Research Associate, from 1975 to 1980 a Lecturer, from 1980 to 1988 an Associate Professor, and since 1988 a Professor at the University of Tokushima, Japan. From November 1980 to February 1981 and from June to September 1986, he was a visiting Associate in Chemical Engineering at the California Institute of Technology, Pasedana. From November 1984 to October 1985 he was a Visiting Researcher at the International Institute for Applied Systems Analysis, Austria. His current interests center on neural networks and distributed parameter system theory. He has authored an co-authored over three hundred papers in journals and conferences.

Dr. Omatu received an Excellent Young Researcher Award from the Society of Instrument and Control Engineers of Japan in 1972 and a Best Paper Award from the Institute of Electrical Engineers of Japan in 1991. He is an Associate Editor for the International Journal of Modelling and Simulation (U.S.) and IMA Journal of Mathematical Control and Information (U.K.) He is co-author of *Distributed Parameter systems; Theory and Applications* (Oxford University press).

36

Appendix I

```c
/*Fuzzy Associations or Rules*/
void     cont_rule(error,change_error,consq)
float    error,change_error;
RULE     *consq;

/*Rule#1*/          if(rule1(error,1,LN1)!=0.0 && rule2(change_error,1,LN2)!=0.0)
                    consq->LP=min(rule1(error,1,LN1),rule2(change_error,1,LN2));

/*Rule#2*/          if(rule1(error,2,MN1)!=0.0 && rule2(change_error,1,LN2)!=0.0)
                    consq->LP=min(rule1(error,2,MN1),rule2(change_error,1,LN2));

/*Rule#3*/          if(rule1(error,2,SN1)!=0.0 && rule2(change_error,1,LN2)!=0.0)
                    consq->LP=min(rule1(error,2,SN1),rule2(change_error,1,LN2));

/*Rule#4*/          if(rule1(error,3,ZE1)!=0.0 && rule2(change_error,1,LN2)!=0.0)
                    consq->LP=min(rule1(error,3,ZE1),rule2(change_error,1,LN2));

/*Rule#5*/          if(rule1(error,1,LN1)!=0.0 && rule2(change_error,2,MN2)!=0.0)
                    consq->LP=min(rule1(error,1,LN1),rule2(change_error,2,MN2));

/*Rule#6*/          if(rule1(error,1,LN1)!=0.0 && rule2(change_error,2,SN2)!=0.0)
                    consq->LP=min(rule1(error,1,LN1),rule2(change_error,2,SN2));

/*Rule#7*/          if(rule1(error,1,LN1)!=0.0 && rule2(change_error,3,ZE2)!=0.0)
                    consq->LP=min(rule1(error,1,LN1),rule2(change_error,3,ZE2));

/*Rule#8*/          if(rule1(error,2,MN1)!=0.0 && rule2(change_error,2,MN2)!=0.0)
                    consq->MP=min(rule1(error,2,MN1),rule2(change_error,2,MN2));

/*Rule#9*/          if(rule1(error,2,SN1)!=0.0 && rule2(change_error,2,MN2)!=0.0)
                    consq->MP=min(rule1(error,2,SN1),rule2(change_error,2,MN2));

/*Rule#10*/         if(rule1(error,3,ZE1)!=0.0 && rule2(change_error,2,MN2)!=0.0)
                    consq->MP=min(rule1(error,3,ZE1),rule2(change_error,2,MN2));

/*Rule#11*/         if(rule1(error,2,MN1)!=0.0 && rule2(change_error,2,SN2)!=0.0)
                    consq->MP=min(rule1(error,2,MN1),rule2(change_error,2,SN2));

/*Rule#12*/         if(rule1(error,2,MN1)!=0.0 && rule2(change_error,3,ZE2)!=0.0)
                    consq->MP=min(rule1(error,2,MN1),rule2(change_error,3,ZE2));

/*Rule#13*/         if(rule1(error,2,SN1)!=0.0 && rule2(change_error,2,SN2)!=0.0)
                    consq->SP=min(rule1(error,2,SN1),rule2(change_error,2,SN2));

/*Rule#14*/         if(rule1(error,3,ZE1)!=0.0 && rule2(change_error,2,SN2)!=0.0)
                    consq->SP=min(rule1(error,3,ZE1),rule2(change_error,2,SN2));

/*Rule#15*/         if(rule1(error,2,SN1)!=0.0 && rule2(change_error,3,ZE2)!=0.0)
                    consq->SP=min(rule1(error,2,SN1),rule2(change_error,3,ZE2));

/*Rule#16*/         if(rule1(error,3,ZE1)!=0.0 && rule2(change_error,3,ZE2)!=0.0)
                    consq->ZE=min(rule1(error,3,ZE1),rule2(change_error,3,ZE2));

/*Rule#17*/         if(rule1(error,2,SP1)!=0.0 && rule2(change_error,2,SN2)!=0.0)
                    consq->SN=min(rule1(error,2,SP1),rule2(change_error,2,SN2));
```

```
/*Rule#18*/              if(rule1(error,2,SP1)!=0.0 && rule2(change_error,3,ZE2)!=0.0)
                         consq->SN=min(rule1(error,2,SP1),rule2(change_error,3,ZE2));

/*Rule#19*/              if(rule1(error,2,SN1)!=0.0 && rule2(change_error,2,SP2)!=0.0)
                         consq->SN=min(rule1(error,2,SN1),rule2(change_error,2,SP2));

/*Rule#20*/              if(rule1(error,3,ZE1)!=0.0 && rule2(change_error,2,SP2)!=0.0)
                         consq->SN=min(rule1(error,3,ZE1),rule2(change_error,2,SP2));

/*Rule#21*/              if(rule1(error,2,MP1)!=0.0 && rule2(change_error,3,ZE2)!=0.0)
                         consq->MN=min(rule1(error,2,MP1),rule2(change_error,3,ZE2));

/*Rule#22*/              if(rule1(error,3,ZE1)!=0.0 && rule2(change_error,2,MP2)!=0.0)
                         consq->MN=min(rule1(error,3,ZE1),rule2(change_error,2,MP2));

/*Rule#23*/              if(rule1(error,4,LP1)!=0.0 && rule2(change_error,3,ZE2)!=0.0)
                         consq->LN=min(rule1(error,4,LP1),rule2(change_error,3,ZE2));

/*Rule#24*/              if(rule1(error,3,ZE1)!=0.0 && rule2(change_error,1,LP2)!=0.0)
                         consq->LN=min(rule1(error,3,ZE1),rule2(change_error,1,LP2));
```

Appendix II

```
/*Membership Function of Error Fuzzy Variable*/
float      rule1(error,flag,centre)
float      error,centre;
int        flag;

       float              wgt;
       switch(flag)
       {
/*For LN1*/        case 1:      if(error<=LN1)wgt =1.0;
                                else wgt=max((2.0/6.0)*(-fabs(error-centre)+6.0/2.0),0);
                                return(wgt);

/*For MN1,SN1,SP1,MP1*/
                  case 2:       wgt=max((2.0/6.0)*(-fabs(error-centre)+6.0/2.0),0);
                                return(wgt);

/*ZE1*/           case 3:       wgt=max((2.0/4.5)*(-fabs(error-centre)+4.5/2.0),0);
                                return(wgt);

/*For LP1*/       case 4:       if(error>=LP1)wgt =1.0;
                                else wgt=max((2.0/6.0)*(-fabs(error-centre)+6.0/2.0),0);
                                return(wgt);
```

Apendix III

```
/*Fuzzy Centroid Defuzzification*/
float     centroid(consq)
RULE   *consq;

       float              w,x,mo_t,mo_b, rule3();
                          mo_t = mo_b = 0.0;

       for(x=4.0;x<=-4;x+=0.25)

           if(consq->LP!=0.0)        w=min(consq->LP,rule3(x,2,LP3));
           if(consq->MP!=0.0)        w=max(w,min(consq->MP,rule3(x,2,MP3));
           if(consq->SP!=0.0)        w=max(w,min(consq->SP,rule3(x,2,SP3));
           if(consq->ZE!=0.0)        w=max(w,min(consq->ZE,rule3(x,3,ZE3));
           if(consq->SN!=0.0)        w=max(w,min(consq->SN,rule3(x,2,SN3));
           if(consq->MN!=0.0)        w=max(w,min(consq->MN,rule3(x,2,MN3));
           if(consq->LN!=0.0)        w=max(w,min(consq->LN,rule3(x,2,LN3));

       mo_b+= w;
       mo_t += w * x;

       return(mo_t/mo_b);
```

40