University of Jordan

Faculty of Graduate Studies

ADAPTIVE CONTROL USING FUZZY LOGIC

M.Sc. Thesis

By

Hashem Ezzat Abed Arabiat

عميد كلية الدراسات العليا

Supervisor

Dr. Yousef Al-Assaf

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science
in Industrial Engineering, Faculty of Graduate Studies, University of Jordan
Amman, Jordan

July, 1993

This thesis was defended successfully on July 14, 1993.

Dr. Yousef Al-Assaf
Industrial Engineering Department
University of Jordan

Chairman of the Committee

د. يوسف العساف

Dr. Sabah Al-Bermani
Industrial Engineering Department
University of Jordan

Member of the Committee

د. صباح عبود محمد البرماني
٢١/٧/٩٣

Dr. Bassam Al-Talhouni
Civil Engineering Department
University of Jordan

Member of the Committee

Bassam T. Talhouni

ACKNOWLEDGEMENTS

I would like to express my deep appreciation and sense of gratitude to my supervisor Dr. Yousef Al-Assaf for his help, advice and encouragement.

It is a pleasure to thank all who helped me during this research, special acknowledgement goes to the technicians in the control laboratory Mr. Mohammed Subieh and Mr. Husam Awad.

Finally, I am greatly indebted to my family and my friends for their encouragement and support.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

Page

Page

Page

Page

NOMENCLATURE

A : Cross-sectional area of tank 1 and tank 2.

a_1, a_2 : Cross-sectional area of the orifices.

Cd_1, Cd_2 : Discharge coefficient of the orifices, 1 and 2 respectively.

$e(t)$, E : Error between the output and the desired set point.

e_k, e_{k-1} : Error between the output and the desired set point at sample K and K-1.

ΔE : Change in error.

g : Gravitional constant.

H_1, H_2 : Height of fluid in tank 1 and tank 2.

H_3 : Height of drain tap (3 cm).

H_{1k}, H_{1k-1} : Height of fluid in tank 1 at sample K and K-1.

H_{2k}, H_{2k-1} : Height of fluid in tank 2 at sample K, and K-1.

J_1, J_2 : Performance criterias.

K_d : Derivative gain factor.

K_i : Integral gain factor.

K_p : Proportional gain factor.

NS : Number of the samples.

Q_i : Pump flow rate.

Q_1 : Flow rate of fluid from tank 1 to tank 2.

Q_o : Flow rate of fluid out of tank 2.

Q_{ik}, Q_{ik-1} : Pump flow rate at sample K, and K-1.

R : Desired water level in tank 2.

T : Sample time.

T_1 : Maximum estimated time constant of the process.

$u(t)$: Control signal.

U_k, U_{k-1} : Control signal at sample K and K-1.

V_1, V_2 : Volume of fluid in tank 1 and 2.

ABSTRACT

This thesis considers the application of Fuzzy Logic Controller (FLC) to a coupled tank process as an alternative to the well known Proportional, Integral and Derivative (PID) controller. The response of the FLC was comparable with that of the PID controller.

To modify the fuzzy rules obtained from operators to suit new process operating conditions an adaptive FLC was developed and applied to the coupled tank system. The practical results showed the usefulness of this technique in giving improved results.

Chapter 1

Introduction to Fuzzy Logic Controller

1.1 Introduction :

Mathematical modelling has been intensively used in most of classical and modern control design techniques such as Root Locus or even advanced techniques such as Predictive control and Pole-Placement [1].

These models have been obtained mostly either by analyzing the basic physical laws of the process to be controlled or by statistical regressions. These controllers produce adequate control as long as these mathematical models are a satisfactory representation of the process around the operating point.

However, for complex systems which are characterized by non-linear, long-time delay, and heavy interaction behaviors embarking on obtaining an adequate mathematical model is a fruitless exercise. An alternative approach to control such processes is to mimic the operator behavior who through experience have achieved adequate control for such plants.

However the "mental model" which an operator forms through experience is not based on mathematical formulation and consequently his actions when controlling a process are characterized by vagueness and are not precise. One method of implementing the bahavior of the operator in a computer is based on Fuzzy Theory. This method has been applied with a certain degree of success to various practical applications as will be discussed later.

1.2 The Basis of FLC :

1.2.1 Human Knowledge representation :

The experience of the operator can be represented in the Logic Form (IF-THEN).

For example :

IF < LEVEL IS LOW > THEN < OPEN VALVE BY LARGE AMOUNT >.

IF < LEVEL IS LOW AND CHANGE IN LEVEL IS MEDIUM > THEN < OPEN VALVE BY MEDIUM AMOUNT >.

IF < LEVEL IS HIGH OR CHANGE IN LEVEL IS HIGH > THEN < CLOSE VALVE BY MEDIUM AMOUNT >.

In general this logic form can be written on a linguistic form :

IF < SITUATION > THEN < ACTION >.

Each IF-THEN form is called a RULE. To adequately control a process a number of RULES are required.

The terms low, medium, large, and high in the above rules are imprecise terms and a method of interpreting such rules is the fuzzy set theory.

Several books discuss the fuzzy set theory [2,3], Fuzzy set theory [4] is a theory about vagueness, uncertainty and enables us to use nonprecise, ill defined concepts and yet to work with these in a mathematically strict sense.

The central concept of Zadeh's Fuzzy set theory [5] is the membership function which represents numerically the degree to which an element belongs to a set. This function takes on values between 0 and 1 . The membership function is assessed subjectively in any instance, small values representing a low degree of membership and high values representing a high degree of membership.

For example [6], if X is a space of all possible points (Universe of Discourse), then a fuzzy set A in X is characterized by a membership function μ_A (x) which associates with each point $x \in X$ a real number in the interval [0,1] that represents the degree of membership of $x \in A$ is :

$$A = ((x, \mu_A (x)) / x \in X) \qquad (1.1)$$

To illustrate the above basic fuzzy set concepts assume that the variable X takes only the integer values (without loss of generality) between 0 and 14, and assume X is divided into three fuzzy sets (small, medium, large), furthermore assume, as shows in Fig. 1.1, that the three fuzzy sets will be described by :

SMALL = 1/0 + 0.7/1 + 0.5/2 + 0.3/3 + 0/4.

MEDIUM = 0/3 + 0.3/4 + 0.5/5 + 0.7/6 + 1/7 + 0.7/8
0.5/9 + 0.3/10 + 0/11.

LARGE = 0/10 + 0.3/11 + 0.5/12 + 0.7/13 + 1/14.

The (+) denotes the union and a fuzzy singelton of the form 0.7/8 signifies the compatibility of 8 with MEDIUM is 0.7.

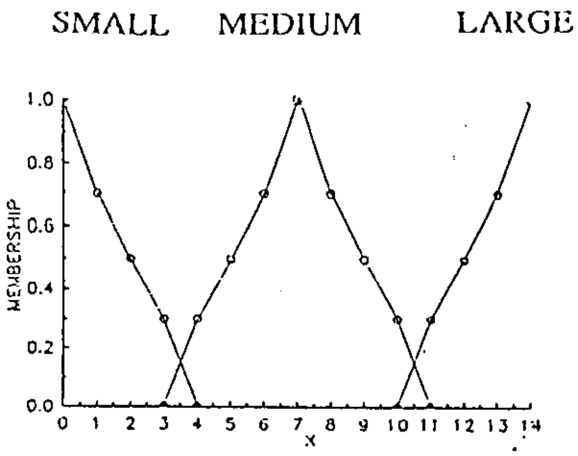


Fig. 1.1 : The three fuzzy sets (small, medium, large)

The question of how to assess these degrees of membership [5] has received surprisingly little attention in the literature, but as Zadeh points out, it is not in keeping with the spirit of the fuzzy-set approach to be too concerned about the precision of these numbers.

1.2.2 Basic Fuzzy Operations :

To illustrate the basic fuzzy operations, they will be applied to three fuzzy sets which are defined on the previous example.

1- Complement :

The complement of a fuzzy set A is defined by $\overline{A}$ with a membership function defined by :

$$\mu_{\overline{A}} = 1 - \mu_A(x) \qquad (1.2)$$

This corresponds to the negation NOT.

For instance, in the above example :

"NOT LARGE" = 1/10 + 0.7/11 + 0.5/12 + 0.3/13 + 0/14.

Fig. 1.2 shows the "NOT LARGE" fuzzy set.

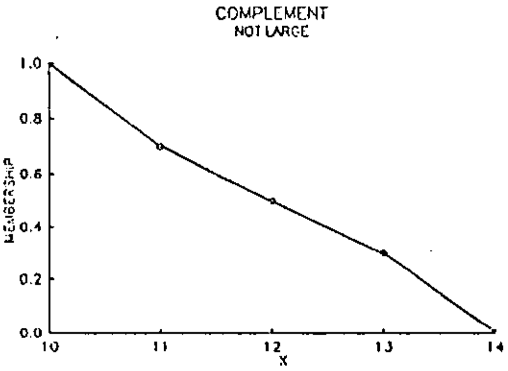


Fig. 1.2 : "Not large" fuzzy set

2. Union :

The union of the fuzzy sets A and B is a fuzzy set C written as C = A U B or A+ B, whose membership function is defined by :

$$\mu_C(x) = \max(\mu_A(x), \mu_B(x)) \qquad (1.3)$$

This corresponds to the connective OR operator.

For example :

"SMALL OR MEDIUM" = 1/0 + 0.7/1 + 0.5/2 + 0.3/3 + 0.3/4 + 0.5/5 + 0.7/6 + 1/7 + 0.7/8 + 0.5/9 + 0.3/10 + 0/11.

Fig. 1.3 shows the "SMALL OR MEDIUM" fuzzy set.

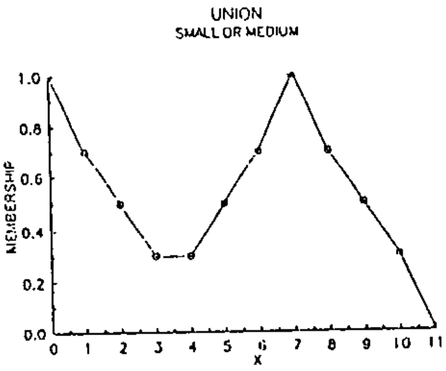


Fig. 1.3 : "Small or medium" fuzzy set

3. Intersection :

The intersection of two fuzzy sets A and B is a fuzzy set C, written as C = A $\cap$ B, the membership function for C is defined by :

$$\mu_C = \min(\mu_A(x), \mu_B(x)) \qquad (1.4)$$

This corresponds to the connective AND operator.

For example :

"SMALL AND MEDIUM" = 0/0 + 0/1 + 0/2 + 0/3 + 0/4 + 0/5 + 0/6 + 0/7 + 0/8 + 0/9 + 0/10 + 0/11.

Fig. 1.4 shows the "SMALL AND MEDIUM" fuzzy set.

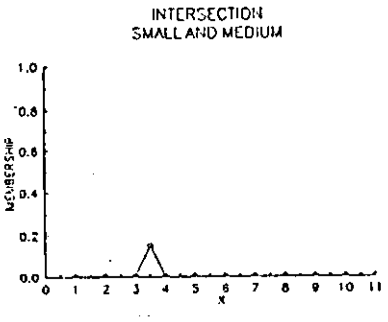


Fig. 1.4 : "Small and medium" fuzzy set.

1.2.3 Fuzzy Implication :

The fuzzy logic control strategy [7] is an automatic implementation of the routine control strategy of an experienced operator. A control strategy is defined by sets of control RULES of the following general form :

IF < CONDITION > THEN < CONTROL ACTION >

where < CONDITION > defines the state of the process for which the specified < CONTROL ACTION > should be executed.

If Y and U are fuzzy terms about the conditions and the control actions of the process respectively, then the rules will be in the form :

IF < Y > THEN < U >

The rule is defined by the cartesian product of the two sets, thus :

$$\mu_{R_i}(y,u) = \min(\mu_Y(y), \mu_U(u)) \qquad (1.5)$$

where y and u denote points in the appropriate multi-dimensional coordinate space. This space illustrates the membership function for one rule for a given y and u, the membership function distribution for a number of rules is defined by using the union operator :

$$\mu_R(y, u) = \max(\mu_{R_i}(y,u)). \qquad (1.6)$$

At a single input value "y*" :

$$\mu(u) = \mu_R(y^*, u) \qquad (1.7)$$

The following example will illustrate the concepts discussed above. Assume a FLC is used to regulate a process. The input to the fuzzy controller is X, and its output is Y, the universe of discourse for X is [0,14] and for Y is

[-7,7], without the loss of generality assume X and Y are integers.

Assume that the universe of discourse X consists of three fuzzy sets Small. Medium, and Large described by :

SMALL = 1/0 + 0.7/1 + 0.5/2 + 0.3/3 + 0/4.

MEDIUM = 0/3 + 0.3/4 + 0.5/5 + 0.7/6 + 1/7
+ 0.7/8 + 0.5/9 + 0.3/10 + 0/11

LARGE = 0/10 + 0.3/11 + 0.5/12 + 0.7/13 + 1/14

And the universe of discourse Y consists of three fuzzy sets Small, Medium, and Large described by :

SMALL = 1/-7 + 0.7/-6 + 0.5/-5 + 0.3/-4 + 0/-3.

MEDIUM = 0/-4 + 0.3/-3 + 0.5/-2 + 0.7/-1 + 1/0
+ 0.7/1 + 0.5/2 + 0.3/3 + 0/4

LARGE = 0/3 + 0.3/4 + 0.5/5 + 0.7/6 + 1/7

Suppose that the system could be controlled by three linguistic rules defined as :

RULE 1 :

If < X is Small > then < Y is Small >.

RULE 2 :

If < X is Medium > then < Y is Medium >.

RULE 3 :

If < X is Large > then < Y is Large >.

The membership values for a given values of X and Y can be found using the following formula :

$\mu_{Ri}(x,y) = \min(\mu_X(x), \mu_Y(y))$.

A sample is taken to illustrate the calculations. If x = 10 and y = 3 then :

$\mu_{R1}(10,3) = \min(\mu_{small}(10), \mu_{small}(3))$.

$= \min(0, 0)$.

$= 0$.

$\mu_{R2}(10,3) = \min(\mu_{medium}(10), \mu_{medium}(3))$.

$= \min(0.3, 0.3)$.

$= 0.3$.

$\mu_{R3}(10,3) = \min(\mu_{large}(10), \mu_{large}(3))$.

$= \min(0, 0)$.

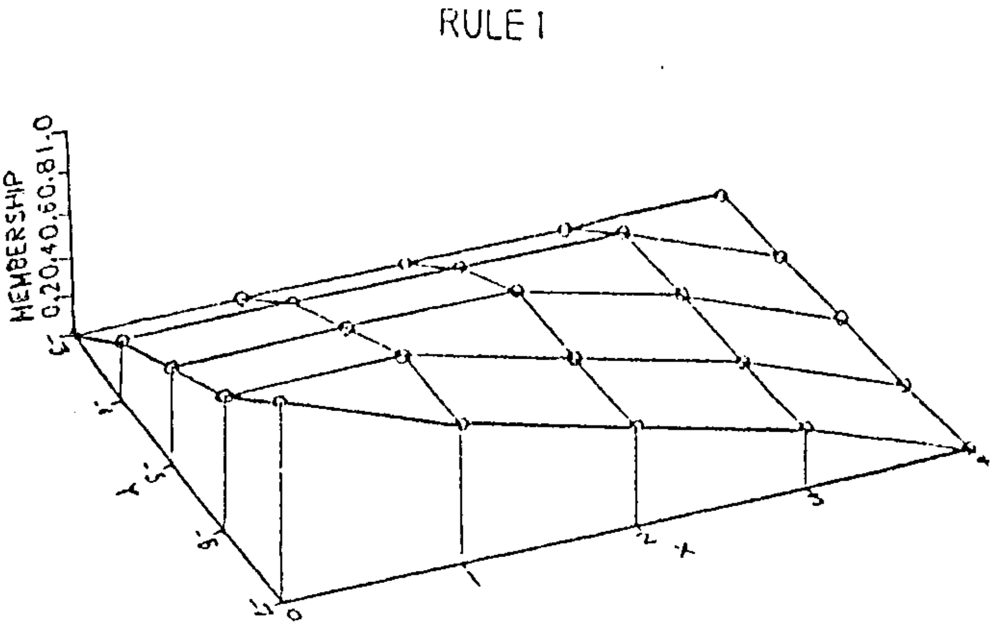
$= 0$.

$\mu_R(10.3) = \max(\mu_{R1}, \mu_{R2}, \mu_{R3})$

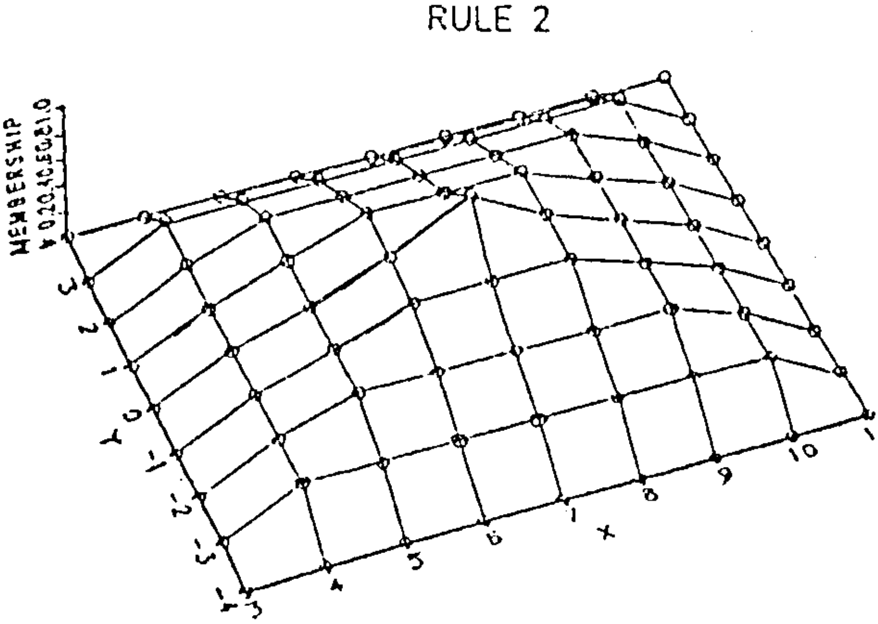
$= \max(0, 0.3, 0)$.

$= 0.3$.

Fig. 1.5 illustrates the results for all values of X and Y.

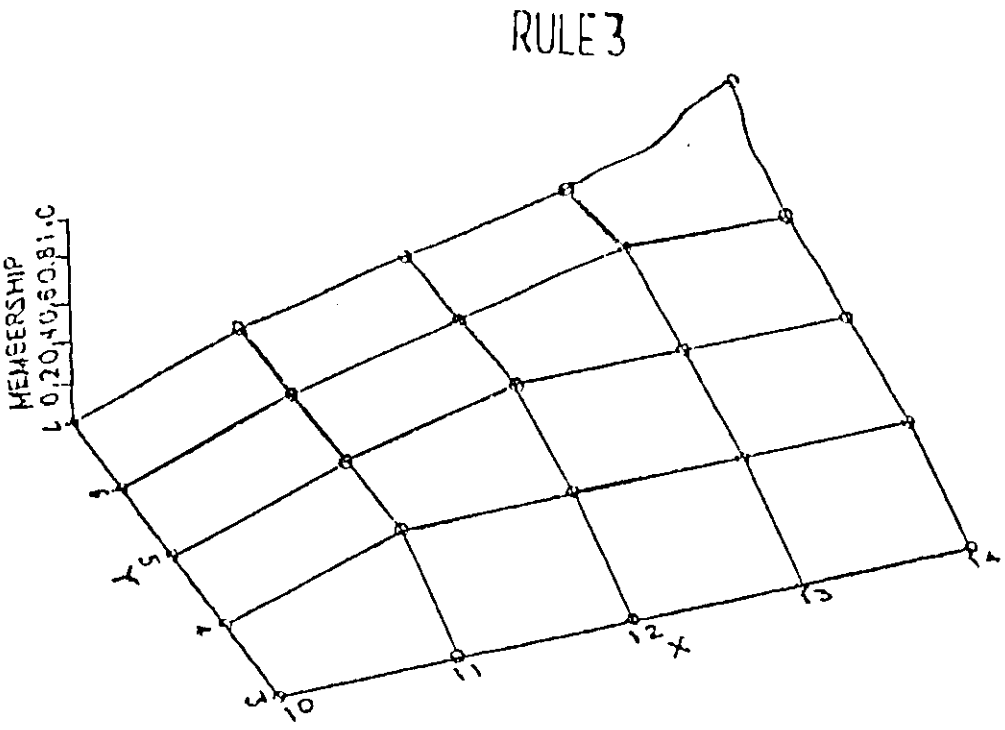


a- Fuzzy relation membership values of rule 1.

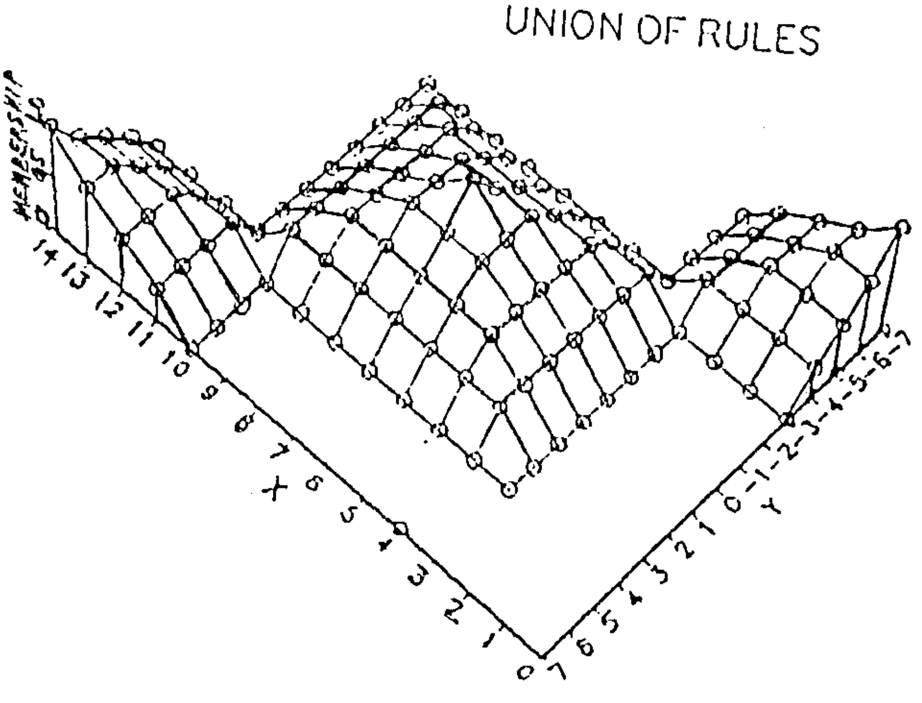


b- Fuzzy relation membership values of rule 2.

Fig. 1.5 Fuzzy membership values of the rules and their overall membership function



c- Fuzzy relation membership values for rule 3.



d- Overall membership function of three rules.

Fig. 1.5 Continued

For the controller to interact with the process it has to apply a single deterministic control action i.e a non-fuzzy action. The process of finding this value is called defuzzification.

There are two methods to find this value :

1- Mean-Maxima Method :

In this method the change in control action which corresponds to the maximum membership value is applied.

To illustrate this method of defuzzification assume in the previous example that X is equal to 12; Fig. 1.6 illustrates the membership for the output values for the above three rules. since we have more than one value of Y which has the maximum value of the membership, then we take the average of them.

$$Y = \frac{5+6+7}{3}$$

$$Y = 6.$$

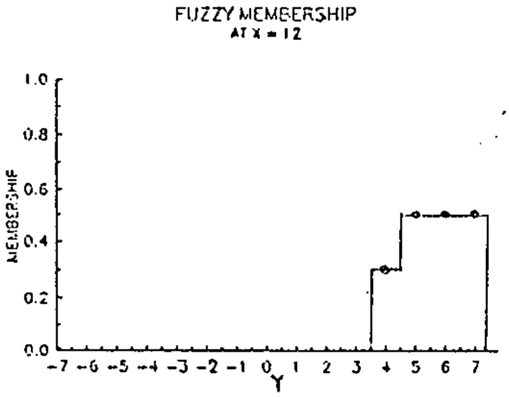


Fig. 1.6 : Fuzzy membership function at X = 12

2- *Center of Area :*

In this method the average of the area included under the membership function is found.

Using center of area method to find the output value at X equal to 12 gives :

$$Y = \frac{(0.3x4) + (0.5x5) + (0.5x6) + (0.5x7)}{(0.3+0.5+0.5+0.5)}$$

Y = 5.66.

Table 1.1 shows the output values using mean-maxima method and center of area method for all values of X.

Table 1.1 : Control actions resulting from the three rules.

Value of X	Action	
	Mean-Maxima Method	Centre of Area Method
0	-7.00	-5.96
1	-6.50	-5.82
2	-6.00	-5.67
3	-5.50	-5.50
4	0.00	0.00
5	0.00	0.00
6	0.00	0.00
7	0.00	0.00
8	0.00	0.00
9	0.00	0.00
10	0.00	0.00
11	5.50	5.50
12	6.00	5.67
13	6.50	5.82
14	7.00	5.96

Since the linguistic rules are fixed and to reduce the computer processing time, some researchers form this look-up table. This table is obtained off-line from the linguistic rules before closing the loop on the FLC, the only job of the controller is to read X and implement Y from the table.

It has been shown that mean-maxima method and center of area method has no major effect on the FLC performance [1], but as shown in table 1.1, mean-maxima method is more active than the center of area method.

1.3 Literature Survey :

ZADEH (1965) (whom is considered to be the father of fuzzy logic) solved the problem of representing the imprecise variables. He introduced the concept of fuzzy set and showed that this concept may be very useful in analyzing Economic, Urban, Social, Biological, and other human centered systems.

Fuzzy logic which was introduced by ZADEH was applied on several areas where there are imprecise information about the variables and the statistical theory is not appropriate such as Decision Making Analysis [8,9,10], control complex systems, Bibliographic Search [4], Power Demand Forcasting [11], Maintenance Scheduling in Transportation Systems [12], Question-Answering Systems for Information Retrieval [13], Medical Diagnosis [14], and on Expert Systems [15].

Our concern in the literature survey here is the application the fuzzy set theory to control industrial processes.

The first reported application of fuzzy set theory to the control of a dynamic process was introduced by Mamdani and Assilian [16,17]. They were concerned with the control of small laboratory steam engine, in which it is required to regulate the engine speed and boiler steam pressure by means of the heat applied to the boiler and the throttle setting on the engine. It is found that the process is highly non-linear and interaction effects neglected in the mathematical approximation model are significant. Operating experience and technical knowledge of the process was used to specify fuzzy control rules. The input to the FLC are the error (difference between the required set-point and the actual output) and the change in error. The number of quantisation levels used was 14 for error (i.e the error is assumed to take 14 integer values), 13 for change in error, 15 for heat input change and 5 for throttle input change.

The outcome of the research is that the fuzzy control system was much less sensitive to process parameter changes and gave good control at all operating points.

Some researchers used a precomputed look-up table to calculate the control action from prespecified linguistic rules in the same way table 1.1 is calculated [1].

A typical look-up table is shown in Fig. 1.7.

	p (K_1 x error)												
	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
d (K_2 x error change)	m (output change)												
-6	-6	-6	-6	-6	-6	-6	-6	-5	-4	-3	-2	-1	0
-5	-6	-6	-6	-6	-6	-6	-5	-4	-3	-2	-1	0	1
-4	-6	-6	-6	-6	-6	-5	-4	-3	-2	-1	0	1	2
-3	-6	-6	-6	-6	-5	-4	-3	-2	-1	0	1	2	3
-2	-6	-6	-6	-5	-4	-3	-2	-1	0	1	2	3	4
-1	-6	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
1	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	6
2	-4	-3	-2	-1	0	1	2	3	4	5	6	6	6
3	-3	-2	-1	0	1	2	3	4	5	6	6	6	6
4	-2	-1	0	1	2	3	4	5	6	6	6	6	6
5	-1	0	1	2	3	4	5	6	6	6	6	6	6
6	0	1	2	3	4	5	6	6	6	6	6	6	6

Fig. 1.7 A typical look-up table [1]

In this look-up table the input variables are the error and the change in error, the action is defined by these two inputs. The error takes 13 values, and the change in error takes 13 values.

The controller here only takes the values of the error and the change in error every sampling period and then read the action from the look-up table. The actual values of the error and change in error are scaled by factors to have an appropriate entry in the table. To improve the performance of the controller the scaling factors for the error, change in error, and control action are modified by the user.

This way for controlling the process has some drawbacks, which make it not preferable to be used. since modifying the scaling factors may give us adequate performance at a given situation, but if the operating conditions are changed, then the performance of this controller may not be acceptable and again the scaling factors must be modified by the user.

Another drawback of using scaling factors and limited number of input/output entries (13 values for error, change in error, and control action) is the possibility of producing a controller with unacceptable degree of quantisation. For example if we assume that the error can take values between -60 and 60 and the associated scaling factor is 10, then the action applied when the error is equal to 56 is the same as if the error is 60. Hence applying the FLC in this way would yield to approximations and imprecise actions which not only do not completely reflect the FLC linguistic rules but also may produce poor performance.

The preferable way to control the process by a FLC is to compute the control action from the rules, every sampling period. This would be a better imitation of the operator.

Kickert with Van Nauta Lemke [18] examined the performance of a fuzzy control algorithm for an experimental warm water plant. They used a FLC to regulate the temperature of the water leaving a tank at a constant flow rate by varying the flow of the hot water in a heat exchanger contained in the tank. It is needed to have a fast response to step changes in outlet water temperature set point. The process is nonlinear, noisy, and has a symmetric gain characteristics and has a pure time delay.

They compared the performance of the FLC with an optimal proportional plus Integral (PI) controller and it was found that the FLC have a faster step response, settling in approximately 0.3 minutes compared with 0.7 minutes for a 10 ℃ set point change obtained by the PI controller. An improved version of the algorithm also had a steady state performance as that given by the PI controller.

The success of Mamdani and Assilian work led King and Mamdani [19] to attempt the control of temperature in a pilot scale batch chemical reactor. The control input to this process is the heating/cooling applied to the batch kettle. The main difficulties in controlling that process is the present nonlinearity, time varying gain and a pure time delay. Good results were achieved.

Even in some cases, FLC performed better than human operators as was illustrated by Rutherford and Carter [20] when they used a fuzzy algorithm for controlling the raw mix permeability of a full seal sinter strand. This process has a single input and a single output, the control variable is the flow of the water of the process. The process is nonlinear, time varying gain, and has a pure time delay. They found that a fuzzy algorithm performed better than human operators. The use of a fuzzy algorithm reduced the standard deviation of permeability by over 40%, and just as well as a conventional PI controller.

FLC is also used for controlling multi-input/ multi-output system. Ostergaard [21] used a fuzzy algorithm to control a multi-input / multi-output small heat exchanger. This process has two inputs and two outputs, the control problem is to regulate the cold water outlet temperature and the hot water inlet temperature by means of the hot water flow and the power used to heat it. This process have some difficulties that the process is highly nonlinear and strongly coupled. It is concluded that the fuzzy algorithm did just as well as a two loop PID controller.

Another application which used the FLC for controlling a multi-input / multi-output system is conducted by Tong [22]. He accomplished experiments for control pressurized tank containing liquid.

A fuzzy algorithm is used to regulate the total pressure and the level of liquid inside the tank by changing the rates of flow of the liquid into the tank and the pressurizing air. This process is nonlinear, strongly coupled and has two very different time constants of approximately 3 and 3000 seconds. As a result adequate control of the process was achieved.

FLC is used to a new area where the control goals are not always clearly stated, the use of qualitative information in decision making, and there is a lack of relevant instrumentation. Tong and Latten [23] conducted this application. Here it is required to control of the activated Sludge wast water treatment process by using fuzzy controller. The activated sludge process is a commonly used method for treating sewage and waste water.

An activated sludge waste water treatment process must produce an effluent that meets some standards. They used 20 rules which are obtained from the operators to the process so that it is conformed with standards. They concluded that a fuzzy algorithm based on practical experience can be made to work on this difficult processes.

FLC is used for improving the performance, Sheridan and Skjoth [7] conducted the fuzzy set theory on automatic Kiln control to improve the performance. Before the fuzzy logic was applied the control of the process was accomplished manually. Acceptable control was achieved by the operators due to experience.

The fuzzy controller is implemented on a Supervision, Dialogue, and Reporting (SDR) system. The hardware consists of a minicomputer system including graphic CRT, operator keyboard, graphic printer, and an interface and process control unit.

The control strategy was defined by sets of control rules, these rules are obtained from the operators. Finally they concluded that automatic kiln control is possible with fuzzy system. The controller is more consistent and efficient than human operator.

A recent application of fuzzy logic control has implemented in Japan by Shoji Miyamoto and Seiji Yasunoby [24] of the System Development Laboratory of Hitashi Ltd. for Automatic Operation of Train (ATO) by means of predictive FLC using criteria of safety, riding, comfort, accurate, stopping, minimal running time, and minimal energy consumption. Field tests and simulations have shown that the "fuzzy ATO" can run a train as skillfully as an experienced human operator. What is interesting in that research is the adaptivity of the controller in which the predicted process behavior is compensated for before it occures.

A further research is conducted for developing a self learning FLC. One of the applications which is conducted on this area of research is applied in Japan was accomplished by Surgeno and others [24]. They developed a self learning FLC to control the motion of the car. The motion of the fuzzy logic controlled car is determined by 20 control rules, each rule recommends a specific change in direction, based on the car's distance from the side and back walls of the turn and its current heading. The car's change in direction is a weighted average of the results of all the rules. The car's position and heading are sensed by a small ultrasonic transducer, and an 8-bit microprocessor adjusts the heading to minimize the difference between the observed position and that predicted by the controlled rules.

From these applications it can be concluded that :

1- Using FLC is best to be used with processes where the controller based on a mathematical representation can not be obtained due to the complex nature of the process.

2- On most of the reported complex applications the fuzzy logic controllers performance can be comparable with PI controller. It is worth while mentioning at this stage that the PI itself based on mathematical modelling.

3- The FLC gave a consistent performance than a human operator if the experience of the operator is well taken.

1.4 Objective of the Research :

As mentioned before fuzzy logic control is an attractive method to be applied to complex processes. However implementing FLC through a look-up table has many drawbacks and the final controller diverges to a certain extent from the original linguistic rules obtained from operators. Furthermore in order to improve the performance of the FLC either the linguistic rules have to be modified or the membership functions have to be assessed again. However, most of the researchers improved the performance by manipulating three scaling factors and this may produce a local improvement but not in any way a comprehensive solution to the problem.

The objective of this research is to study and modify the FLC to overcome the above problems. This will be accomplished by :

1- Writing a software package to implement the basic fuzzy logic rules, exactly as how the operator would process his knowledge and applying it.

2- Apply this FLC, and the PID controller to a real practical application, namely, the coupled tank system and make a comparison between their performance.

3- Produce a technique which automatically modifies the fuzzy logic rules. This is equivalent to adaptive control technique used in mathematical modelling where the model is continuously modified to represent the process adequately at all times and operating conditions. However, a mathematical model is not required. It is proposed to have a reference trajectory where the controller has to achieve and the rules which give a response not close to that reference are automatically modified, and this learning process is repeated until the controller gives the closest response to that reference trajectory.

To accomplish this, Chapter two discusses the computer implementation of FLC, while Chapter three presents the application of FLC to the coupled tank system. Chapter four introduces the adaptive FLC, while the conclusions and the further future research are presented in Chapter five.

Chapter 2

Computer Implementation of Fuzzy Logic Controller

Computer Implementation of Fuzzy Logic Controller :

This chapter discusses a software package written in Quick-Basic to resemble a FLC. Detailed manipulation of linguistic rules is also discussed. To compare the performance of the FLC, a PID controller is also implemented for that purpose.

2.1 Programming Language :

Quick-Basic is used for implementing the FLC, because it is an easy language for programming, it has facilities that facilitates the programming process, and for its fast speed of execution.

2.2 FLC Software :

Fig. 2.1 shows a flow chart describing the requirements for FLC and its operation. Each item will be discussed in details in the following sections :

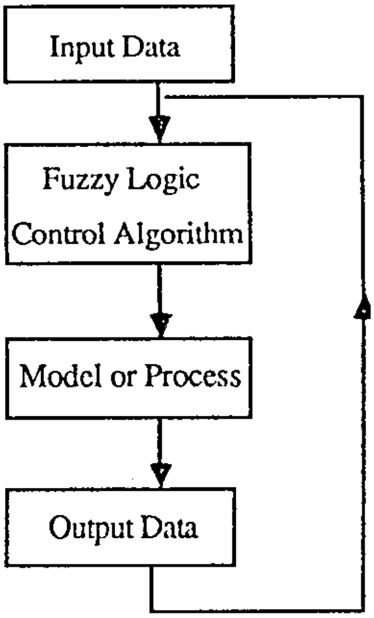


Fig. 2.1 FLC requirements and operation

2.2.1 Input Data :

As shown in Fig. 2.2, the input data required to be specified by the user includes :

- Controller input and output variables.
- Fuzzy subsets for each variable.
- Linguistic rules to control the process.

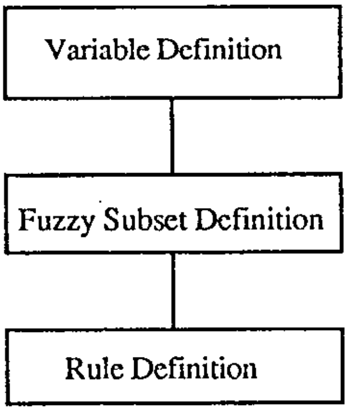


Fig. 2.2 Required input data

Two methods are available to give the input data, either loading data from file or the user inputs data interactively with software.

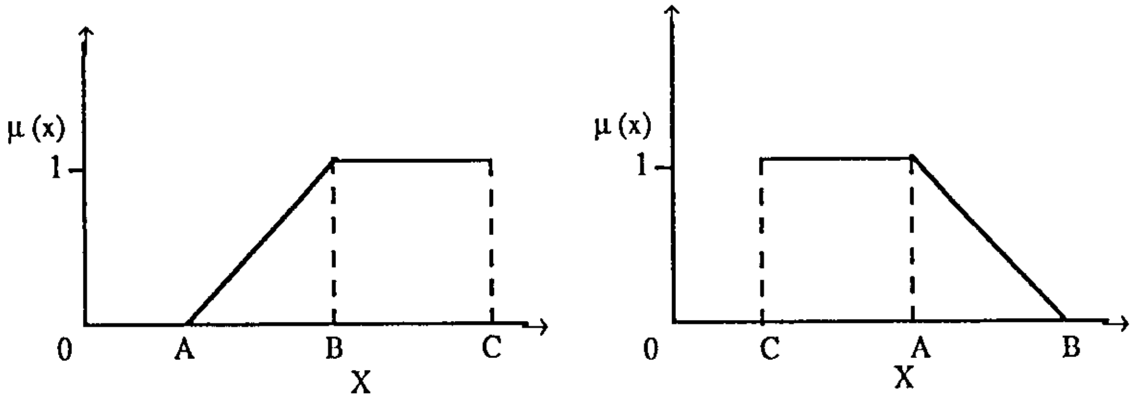
For variable definition the number of the variables, their names and the type of each variable (either condition or action) have to be given. The type of each variable is classified according to the general form of any rule :

IF < Condition > Then < Action >

After specifying the variable, the number and type of its fuzzy subsets like high, medium, and low are to be defined.

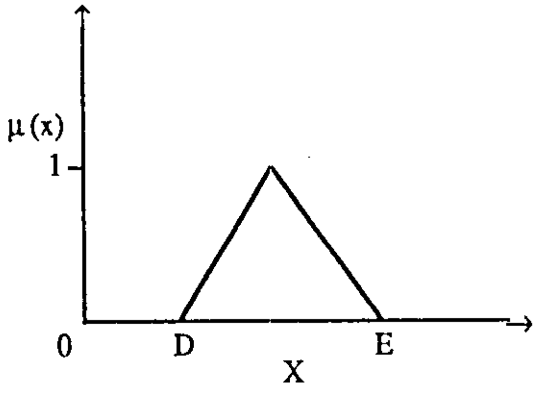
Since the shape of the membership functions for each subset is not critical

[1], the shapes shown in Fig. 2.3 are used for the ease of computation in manipulating of operations in the subsets. It is seen from Fig. 2.3 (a) and Fig. 2.3 (b) that the two shapes represent type 1 and they can be distinguished from each other by definning the points A,B,C, and the slope of the ramp part. Fig. 2.3 (C) shows type 2 which can be defined by the two points D and E.



a- Fuzzy subset type 1.

b- Fuzzy subset type 1.



c- Fuzzy subset type 2.

Fig. 2.3 Types of the fuzzy subset

After giving the fuzzy variables and their subsets, the rules which are based on these variables can be given. Rules definition consists of specifying their number, and types.

Type one :

On this type the action depends only on one input for the condition, it is usually written on the following form :

If < Condition > Then < Action >

It can be written as :

If <	Condition variable name	> is <	Fuzzy subset name	> Then <	Action variable name	> is <	Fuzzy subset name	>

For example :

If < Error > is < high > Then < Response > is < high >

Type two :

On this type the action depends on two inputs for the condition term. It is written on the following form :

If < Condition 1 > Relation < Condition 2 > Then < Action >

It can be written as :

If <	condition variable 1 name	> is <	Fuzzy subset name	> Relation <	Condition variable 2 name	> is <	Fuzzy subset name	>

Then < Action variable name > is < Fuzzy subset name > .

Relation has two choices, it may be chosen to be (OR) or (AND).

The following example illustates this type :

If < Error > is < high > AND < Change in error > is < high >

Then < Response > is < high >

After definning the type of a rule the condition variable name, Action variable name, and fuzzy subsets are defined to make the rule fully defined.

Finally, to simply modification of input data for the user, they can be saved on a file for later use.

2.2.2 : Fuzzy Logic Control Algorithm Manipulation :

As shown in Fig. 2.4, the major parts of fuzzy logic control algorithm manipulation are :

1. Partitioning the action variable and calculating its membership value at each division.
2. Calculating the membership function for each rule.
3. Calculating the membership function for the union of the rules.
4. Calculating the controller outputs.

Each item mentioned above will be discussed in details as follows.

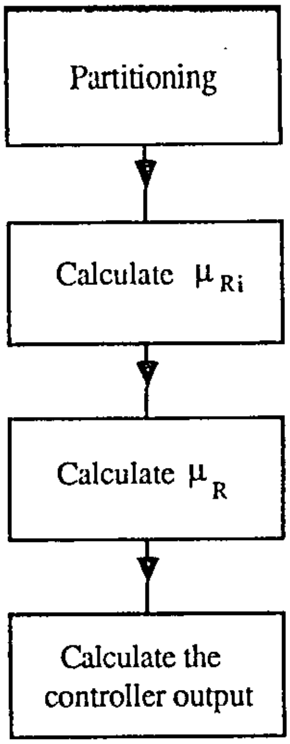


Fig. 2.4 Major parts of fuzzy logic control algorithm

1- Partitioning the action variable :

To illustrate this part, assume that the action variable (x) has two fuzzy subsets high and low as shown in Fig. 2.5.

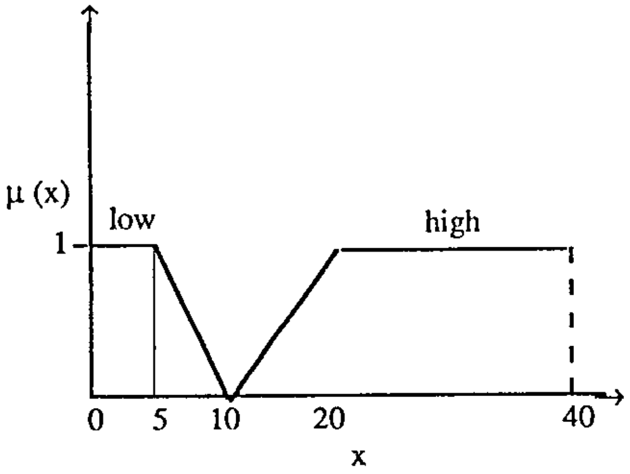


Fig. 2.5 Low and high fuzzy subsets

The membership function for the two fuzzy subset can be defined by :

For low fuzzy subset :

$\mu_{low}(x) = 1; \quad 0 \leq x \leq 5$

$\mu_{low}(x) = 2\text{-}0.2\ x; \quad 5 \leq x \leq 10$

$\mu_{low}(x) = 0$; otherwise

for high fuzzy subset :

$\mu_{high}(x) = 1\ ; 20 \leq x \leq 40$

$\mu_{high}(x) = -1 + 0.1\ x; \ 10 \leq x \leq 20$

$\mu_{high}(x) = 0$; otherwise

Assume it is found that dividing the action variable into 10 divisions will be accurate in calculating the output of the controller, then there are 10 divisions which fall between 0 and 40 with increment equal to :

$$\text{Increment} = \frac{40\text{-}0}{9} = 4.44$$

and the divisions are :

0, 4.44, 8.88, 13.33, 17.77, 22.22, 26.66, 31.11, 35.55, 40

The next step is to calculate the membership values at each division for each fuzzy subset.

For low fuzzy subset the following expression gives the value of the division and its membership value on the form of pairs.

$$(X, \mu\,(x)) = \{(0,1),\ (4.44,1),\ (8.88,\ 0.224),\ (13.33,0),\ (17.77,0),\ \ldots,\ (40,0)\}$$

For high fuzzy subset :

$(X, \mu(x)) = \{(0,0), (4.44,0), (8.88, 0), (13.33,0.333), (17.77,0.777), (22.22,1), ..., (40,1)\}$

On the software the action variable is divided into 100 divisions to have accurate results, (note that only 13 divisions are taken in most previous researchs). The choice of this number depends on the required resolution and accuracy of the controller. Furthermore the number of divisions can be changed easily in our software since no look-up table is formed. The membership value is calculated in the sofware using the values of the points A,B, and C when the fuzzy subset type is 1, or using the values of the points D and E when the fuzzy subset is type 2. The points (A,B,C,D, and E) that define the zone of the fuzzy subset are used to calculate the equations of the lines that connect them, from these equations the membership values are calculated.

2- Calculating the membership function for each rule :

Type of the rule gives the procedure for calculating its membership function :

Type one :

General form of the rule :

If < Y > Then < U >

The membership for this type of rule is defined as :

$$\mu_{R_i}(y,u) = \min(\mu_Y(y), \mu_U(u))$$

Notice that $\mu_Y(y)$ for a given value of y is a single number. Hence the 100 values of μ_{R_i} is the minimum between this single value and membership value of the action variable at each division.

For type two :

General form of the rule :

If < Y_1 > Relation < Y_2 > Then < U .

For OR relation :

If < Y_1 > OR < Y_2 > Then < U >

The membership for this type of rule is defined as :

$$\mu_{Ri}(y_1,y_2,u) = \min(\mathrm{Max}(\mu_{Y1}(y_1), \mu_{Y2}(y_2)), \mu_U(u))$$

Notice that $\mu_{Y1}(y_1)$, $\mu_{Y2}(y_2)$ for given values of y_1, y_2 are a single numbers. First the maximum of the two values of $\mu_{Y1}(y_1)$ and $\mu_{Y2}(y_2)$ is calculated then the minimum between this value and membership value at each division of the 100 divisions of the action variable is determined.

For AND relation :

If < Y_1 > AND < Y_2 > THEN < U >

the procedure for computing the membership for this type of rule is the same as in the OR relation case but instead of calculating the maximum value of $\mu_{Y1}(y_1)$ and $\mu_{Y2}(y_2)$ the minimum value is calculated.

3- Calculating the membership function of the union of the rules.

As discussed before the membership function of the union of the rules is calculated by taking at each division of the action variable the maximum membership value given by all rules.

4- Calculating the controller output :

Centre of area method for calculating the controller output is used in the software.

2.2.3 : Model or Process :

The FLC is either used to control a real time software simulation of the coupled tank process, or to control the actual process through an interface unit which has been designed at the Industrial Engineering Department by previous workers. A schematic diagram of the computer interface to the process is shown in Fig. 2.6. The program which is developed to be used to control a real time software simulation of the process is presented in Appendix A.

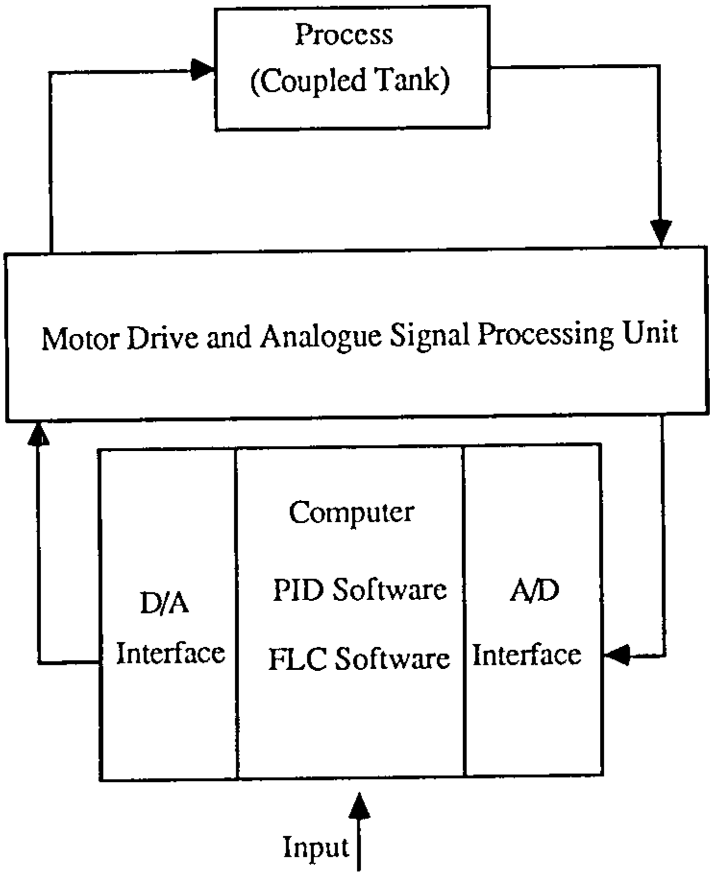


Fig. 2.6 Computer interface to the coupled tank process

2.2.4 : Output Data :

The output results (controller action and process output) are plotted on-line at each control sample, on the screen, so that the user can monitor the controller behaviour while its running. The data is also saved on a file for further usage.

To compare the results of FLC with the PID controller a program for the later is written. The PID controller in the time domain is given by [25] :

$$u(t) = Kp\, e(t) + Ki \int e(t)\, dt + Kd \frac{de(t)}{dt} . \qquad (2.1)$$

Where e (t) is the error

u (t) is the control signal

Kp : proportional gain factor

Ki : integral gain factor

Kd : derivative gain factor

Since a digital computer is used, a digital form of the PID controller is required. Hence the control signal can be calculated from [25] :

$$U_k = U_{k-1} + a\, e_k + b\, e_{k-1} + c\, e_{k-2} \qquad (2.2)$$

where : $a = K_p + 0.5\, K_i T + Kd/T.$

$b = 0.5\, Ki\, T - Kp - 2\, Kd/T.$

$c = Kd/T.$

U_k : control signal at sample K.

U_{k-1} : Control signal at sample K-1.

e_k : error at sample K.

e_{k-1} : error at sample K-1.

T : sample time.

eq. (2.2) can be written as :

$$\Delta U_k = a\, e_k + b\, e_{k-1} + c\, e_{k-2} \qquad (2.3)$$

where : $\Delta U_k = U_k - U_{k-1}$

This equation is used in the software to give the change in the control signal.

The user has to give values for the controller parameters. The program of the PID controller which is written to be used in simulation is presented, in Appendix B.

Chapter 3

Application of FLC to the Coupled Tank System

Application of FLC to the Coupled Tank System :

To demonstrate the ability of the FLC it is applied to a real-time process, namely, the coupled tank system. This process is characterised as a non-linear process. Firstly, software simulation for the coupled tank is developed and the FLC is applied to the simulation in order to gain confidence on the controller ability, troubleshoot the software and design suitable parameters for the FLC and the PID controller, so that it can be applied successfuly to the real system.

Finally, the two controllers are applied practically to the coupled tank system. This chapter discuss the simulation and the practical results of the FLC and the PID controller to the coupled tank system.

3.1 Coupled Tank System :

As shown in Fig. 3.1 the coupled tank [26] consists of a transparent plexi-glass container measuring (20 cm length by 10 width by 30 cm high). A center partition is used to divide the container into two tanks. Flow between the tanks is by means of a series of holes drilled near the base of partition. Three holes, having diameters of 1.27 cm, 0.95 cm and 0.635 cm are situated 3cm above the base of the tank, A smaller (bleed) hole of 0.317 cm diameter is situated at a height of 1.5 cm. With all bungs removed the container can be considered as one big tank.

The depth of fluid is measured using parallel track depth sensors which are stationed in tank 1 and tank 2. This device performs as an electrical resistance which varies with the water level. The changes in resistance are detected and provide an electrical signal which is proportional to the height of water.

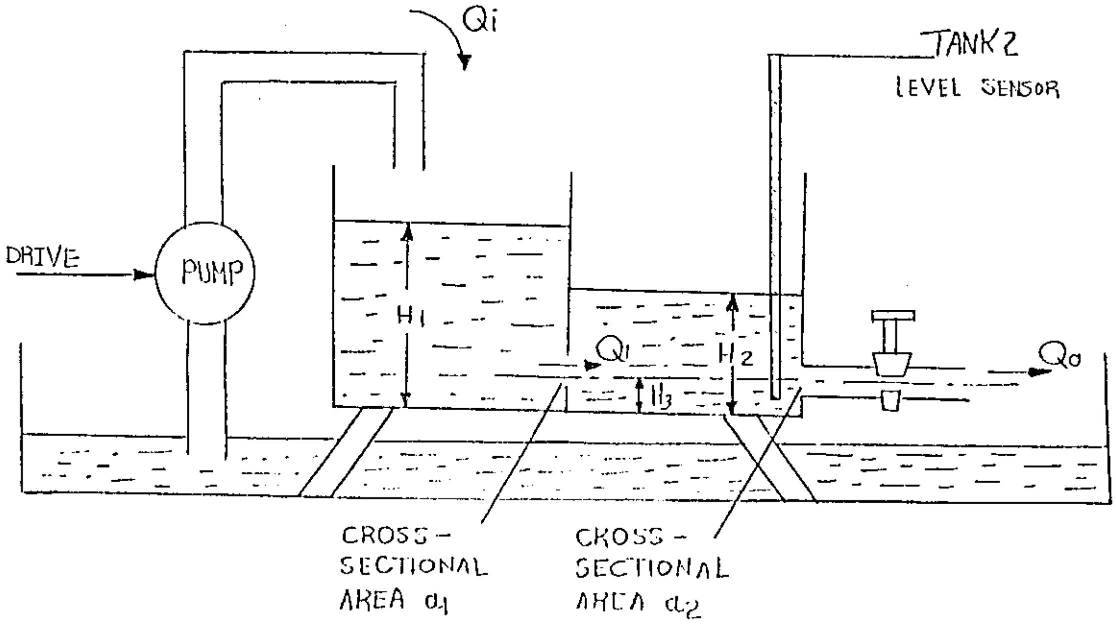


Fig. 3.1 Coupled Tank System

The water which flows into tank 2 is allowed to drain out via an adjustable tap, and the entire assembly is mounted in a large tray which also forms the supply reservoir for the pump. Fully open, the drain tap has a diameter of 0.70 cm.

3.2 Modelling :

The dynamic equations of the system [26] may be derived by taking flow balances about each tank. For the first tank :

$$Q_i - Q_1 = \frac{dV_1}{dt} = A\,\frac{dH_1}{dt} \qquad (3\text{-}1)$$

where V_1 = the volume of fluid in tank 1.

H_1 = height of fluid in tank 1.

A = cross-sectional area of tank 1 and tank 2 (i.e $A_1 = A_2 = A$).

Q_1 = flow rate of fluid from tank 1 to tank 2.

Q_i = pump flow rate.

For the second tank :

$$Q_1 - Q_o = \frac{dV_2}{dt} = A\,\frac{dH_2}{dt} \qquad (3\text{-}2)$$

where V_2 = the volume of fluid in tank 2.

H_2 = height of fluid in tank 2.

Q_o = flow rate of fluid out of tank 2.

If the inter-tank holes and the drain are assumed to behave like orifices, then the following equations describe the characterstic relations for orifices:

$$Q_1 = Cd_1\, a_1 \sqrt{2g\,(H_1 - H_2)} \qquad (3.3)$$

$$Q_o = Cd_2\, a_2 \sqrt{2g\,(H_2 - H_3)} \qquad (3.4)$$

where : a_1 = cross sectional area of orifice 1.

a_2 = cross sectional area of orifice 2.

Cd_1, Cd_2 = discharge coefficient of the orifices.

H_3 = height of drain tap (3cm).

g = gravitional constant.

Equations (3.1) to (3.4) describe the system dynamics in its nonlinear form.

Adding eq. (3.1) to eq. (3.2), gives :

$$Q_i - Q_o = A\left[\frac{dH_1}{dt} + \frac{dH_2}{dt}\right] \tag{3.5}$$

subsituting eq. (3.3) in eq (3.2), gives :

$$Cd_1\, a_1 \sqrt{2g\,(H_1 - H_2)} - Q_o = A\,\frac{dH_2}{dt} \tag{3.6}$$

by manipulating eq. (3.6), gives :

$$Cd_1\, a_1 \sqrt{2g\,(H_1 - H_2)} = A\,\frac{dH_2}{dt} + Q_o \tag{3.7}$$

squaring eq (3.7), gives :

$$H_1 = \frac{1}{2g\, Cd_1^2\, a_1^2}\left[A\frac{dH_2}{dt} + Q_o\right]^2 + H_2 \tag{3.8}$$

In order to simulate the coupled tank on the computer, the FLC would produce a value for Q_i then a resulting value for H_2 should be fedback to it. For proper simulation this value should be very close to the value produced by the real coupled tank. To achieve this a linear approximation for the equations (3.1) to (3.4) presented in [26] is not adapted, instead a digital non-linear solver is adapted. This solver uses the following feedback approximation for the derivative as :

$$\frac{dH_1}{dt} = \frac{H_{1k} - H_{1k-1}}{T} \qquad (3.9)$$

$$\frac{dH_2}{dt} = \frac{H_{2k} - H_{2k-1}}{T} \qquad (3.10)$$

where : H_{1k}, H_{1k-1} : Height of fluid in tank 1 at sample K and K-1.

H_{2k}, H_{2k-1} : Height of fluid in tank 2 at sample K and K-1.

The digital non-linear solver operation can be described by the following steps :

1. FLC or PID controller produce a value for Q_i.
2. Through experiments it was found that even for a maximum Q_i the change in H_2 for each sample (see the next section) would not exceed 1cm, hence the value of H_{2k} falls between $H_{2k-1}-1$ and $H_{2k-1}+1$, the solver examines all values of H_{2k} between this range with a step of 0.01 and determines the best value to solve the non-linear equations as described below.
3. Calculate H_{1k} using eq. (3.8).
4. Calculate the difference between the left side and the right side of eq. (3.5).

5. Calculate the value of H_{2k} which gives the minimum difference.

6. Take the values of H_{1k}, and H_{2k} that minimize the difference as the solution of the non-linear equations.

3.3 The Choice of Sampling Time :

The selection of the sampling time remains very much an art, rather than a science. Sampling too slowly can reduce the effectiveness of the feed back system. In such cases a disturbance could affect the process and disappear before the controller takes any corrective action. In the extreme where the sampling time is greater than the process response time, the controller is then performing only steady state control. On the other hand sampling too fast may activates the input actuator frequently while the process is slow and no benifit is gained and the actuator life-time is reduced.

Goodwin [27] suggests to select the sample time (T) such that :

$$T_1/15 \leq T \leq T_1/5 \qquad (3.11)$$

where T_1 is the maximum estimated time constant of the process. Although this time constant varies because of the non-linear nature of the process, an approximate value was used to suit the desired application. In our work the sampling time is chosen to be :

$$T = T_1/15$$

to have a reasonably fast sampling time.

Two open-loop experiments were conducted on the coupled tank to estimate the maximum time constant, one for the coupled tank with two holes open (smallest two holes with diameters 0.635 cm and 0.317 cm), while the other experiment is with all holes open.

The system in the first experiment is considered as a second order system, while in the second case it is considered as a first order system.

Fig. 3.2 shows the water level in the second tank with two holes closed in response to a step change in the input flow from 0 to 34 cm^3/sec.

A simple approximated procedure to find the time constant of the process is to find the time taken by its output to reach 63% of its steady-state value. It can be deduced from Fig. 3.2 that T_1 = 150 sec.

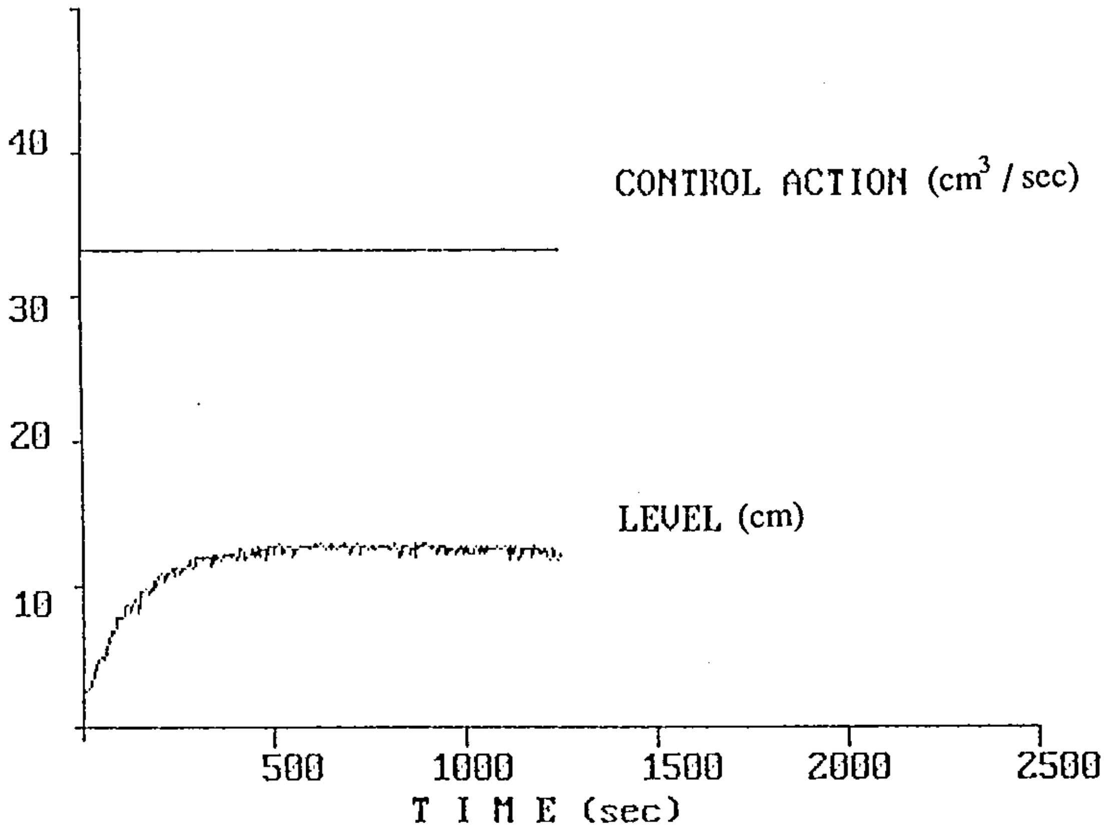


Fig. 3.2 Open loop response with second order system

It can be deduced from Fig. 3.3, that for the second experiment (all holes open). T_1 = 80 see. It can seen that,the time constant of the process in the second case is less than the first one and this confirms with the physical behavior of the coupled tank, since water increases in the second case faster than the first case. Furthermore the tank in both experiments is an overdamped process.

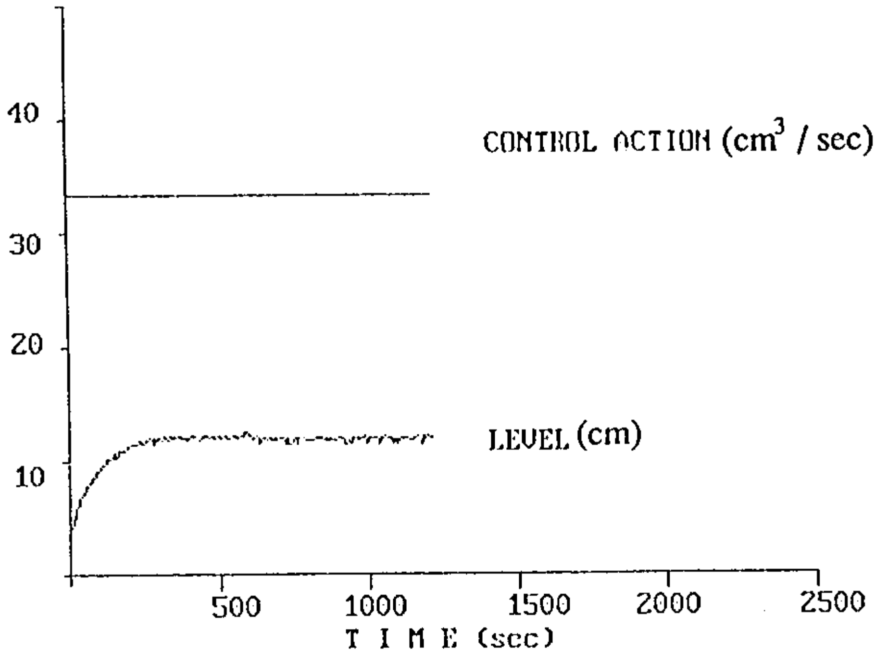


Fig. 3.3 open loop response with first order system

Using eq. (3.12), two values for the sample time are found :

1. For the coupled tank with two holes open :

   $T = T_1/15 = 150/15 = 10$ sec.

2. For the coupled tank with all holes open :

   $T = T_1/15 = 80/15 = 5.33$ sec.

Since it is designed to conduct experiments for the two cases (when two holes are open and when all holes are open) in the same run, it is decided to choose a 5 seconds sampling time.

As it is seen from Fig. 3.2 and Fig. 3.3 at steady state the level is not fixed but it varies about the steady state value, this is due to noise. It is found during the two experiments that the level deviates from the steady state value by ± 0.15 cm.

The noise is represented in simulation by a random variable with a mean equal to zero and a standard deviation equal to 0.05 cm.

3.4 Forming the PID controller :

The coupled tank simulation using the non-linear solver was used to design suitable parameters for Kp, Ki, and Kd that give a small overshoot, fast rising time, and good regulation at steady state. Using the simulation program in the computer for the coupled tank process, several experiments are conducted on different values of Kp, Ki, and Kd, so that the controller would increase the water level in the second tank from 8 cm to 12 cm. The following values for Kp,Ki, and Kd are found to be suitable Kp=5, Ki=0.2 and Kd=0.05.

These values will be used later in simulation and experimental work.

3.5 Forming the FLC :

As it has been done with the PID, the simulation was used to obtain reasonable fuzzy subsets and rules. The error and the change in error were used to manipulate the action signal (flow rate), the error and the change in error were defined as :

$$E = R - H_{2k} \tag{3.13}$$

$$\Delta E = H_{2k-1} - H_{2k} \tag{3.14}$$

where : E is the error

ΔE is the change in error

R is the desired water level in the second tank.

The error, change in error, and action were divided into the following fuzzy subsets Positive High (PH), Positive Medium (PM), Positive Low (PL), Negative Low (NL), Negative Medium (NM), and Negative High (NH). The fuzzy subsets for each fuzzy variable are shown in Fig. 3.4.

The general form for the rules which were used in the FLC can be written as :

If < error > is < fuzzy subset for the error > And < Change in error > is < fuzzy subset for the change > Then < action > is < fuzzy subset for the action >.

The fuzzy linguistic rules are shown in Fig. 3.5. From this figure, it can be shown that when the error is PH and change in error is PH, the action is PH.

This FLC was formed such that it would increase the water level in the second tank from 8 cm to 12 cm. Fig. 3.6 shows a block diagram illustration for the FLC operation in the coupled tank process.

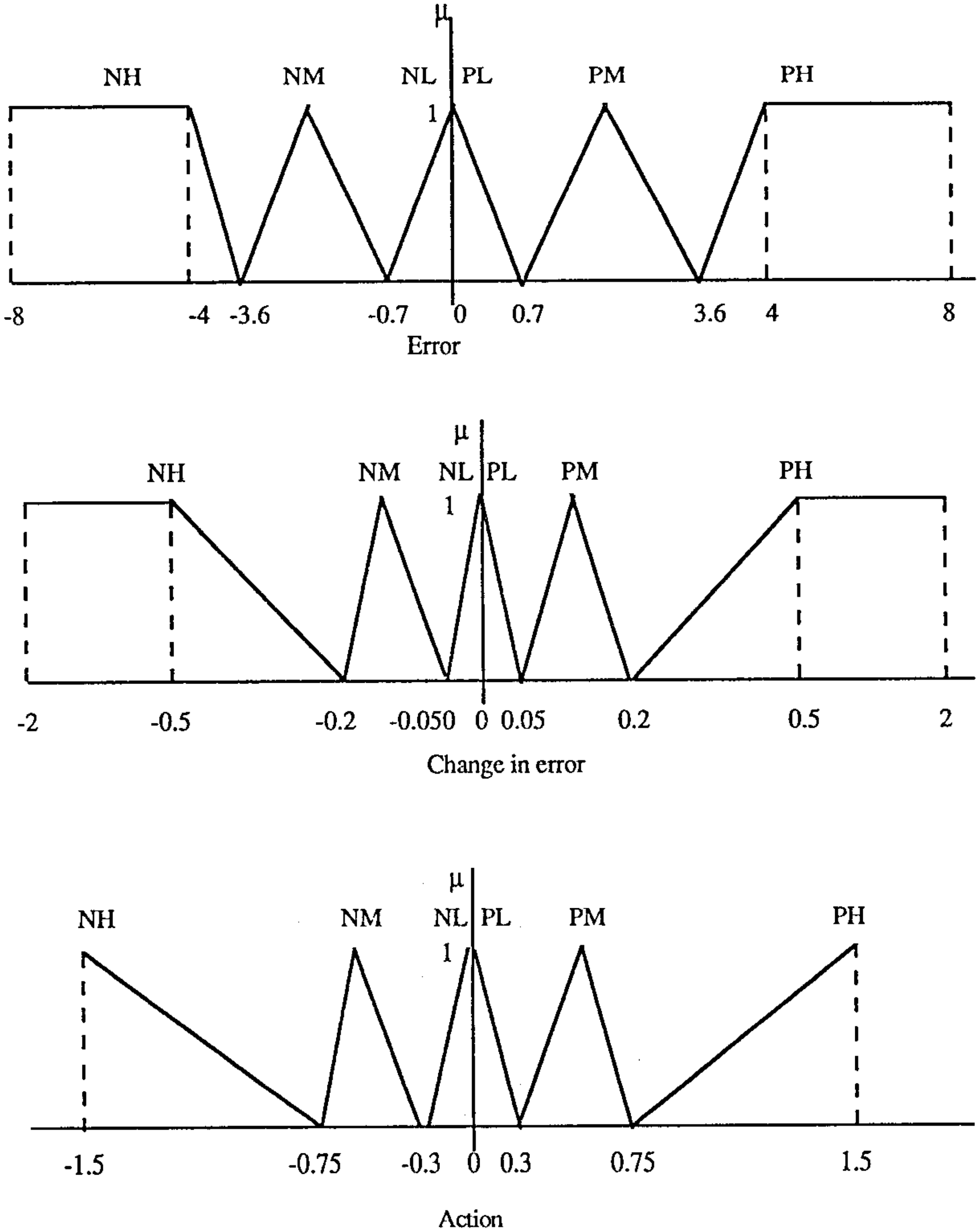


Fig. 3.4 Fuzzy subsets as designed by simulation

(Note : the x-axis is not to scale)

Error Action Change in error	PH	PM	PL	NL	NM	NH
PH	PH	PH	PM	PM	PM	NM
PM	PH	PM	PM	PL	NL	NM
PL	PH	PM	PL	NL	NM	NH
NL	PH	PM	PL	NL	NM	NH
NM	PH	PL	NL	NM	NM	NH
NH	PH	NM	NM	NM	NH	NH

Fig. 3.5 FLC rules as designed by simulation.

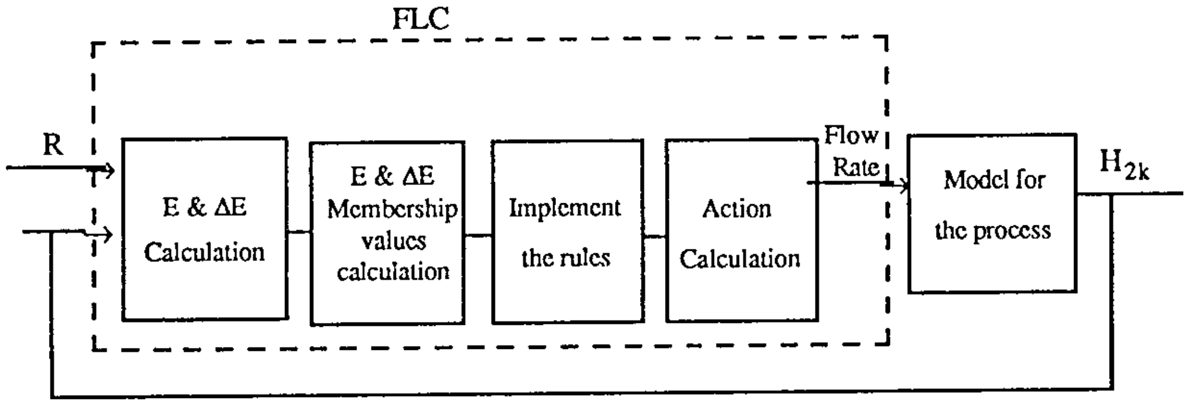


Fig. 3.6 FLC operation for controlling the level

One should mention at this stage that obtaining suitable initial setting for the FLC is not only harder in comparision with those of the PID controller but also time consuming.

3.6 Application Experiments :

The formed FLC and PID controller from simulation are used to increase the water level in the second tank from 8cm to 12 cm under several conditions to test the performance of these controllers. These experiments are introduced in the following sections.

3.6.1 Pure Second Order System :

As an initial experiment to compare the performance of the two controllers on a simple process, they are applied to the coupled tank with only two holes are open (smallest two).

Table 3.1 Coupled tank parameters for the second order system.

a_1 (cm^2)	0.3956
a_2 (cm^2)	0.3848
A (cm^2)	100
$Cd_1 \simeq$	0.80
$Cd_2 \simeq$	0.75

Table 3.1 shows the values of the coupled tank parameters for the second order system.

The process response using FLC and PID in simulation are shown respectively in Fig. 3.7 and Fig. 3.8. These figures indicate that the PID controller is faster than the FLC, and the two responses have a small overshoot. Practical responses for the two controllers are shown in Fig. 3.9

and Fig. 3.10. As with simulation results the PID is faster than the FLC. However there is a small overshoot, and the PID controller is seen clearly more active than the FLC. By comparing the simulation responses with the practical responses, it is seen that they confirm with each other. The difference in the activity of the action signal is due to noise.

The performance of the FLC and the PID controller is required to be analysed quantitively. To do that two performance criterias J_1 and J_2 are used such that :

$$J_1 = \sqrt{\sum_{k=1}^{NS} (H_{2k} - R)^2} \qquad (3\text{-}15)$$

$$J_2 = \sqrt{\sum_{K=1}^{NS} (Q_{ik} - Q_{ik-1})^2} \qquad (3\text{-}16)$$

where : NS is number of the samples

Q_{ik} is the pump flow rate at sample K

Q_{ik-1} is the pump flow rate at sample K-1

The first criteria (J_1) gives an indication of the deviation of the water level in tank 2 from the desired set-point. The more close the level to the set-point (i.e less J_1) the better is the controller. J_2 indicates the amount of activity of the actuater, the less the value of J_2 the better performance achieved by the controller. The best controller is the one which gives less values for both J_1 and J_2. However if this is not the case, then the application determine which is better. In some applications minimizing J_1 is more important than minimizing J_2 while in other it is vice versa.

Table 3.2 shows the values of J_1 and J_2 for the practical results (In fact for all future experiments, J_1 and J_2 are calculated for practical cases only).

Table 3.2 Analysis table for pure second order system.

Controller Type	NS	J_1	J_2	Average at steady state	Variance at steady state
FLC	244	16.35	7.14	12.036	0.058
PID	244	11.44	35	12.087	0.076

From table 3.2, according to the first criteria (J_1) the PID controller is better than the FLC since the PID has a fast response although that the PID variance at steady state is more than that for the FLC. Further more it can be deduced that the FLC is less active than the PID.

Both controllers reach the set-point and try to keep the water level close to it, this is seen from the average at steady state for both controllers. One could reduce the activity of the PID controller at the expense of reducing its speed by using smaller proportional gain. However this will make it slower.

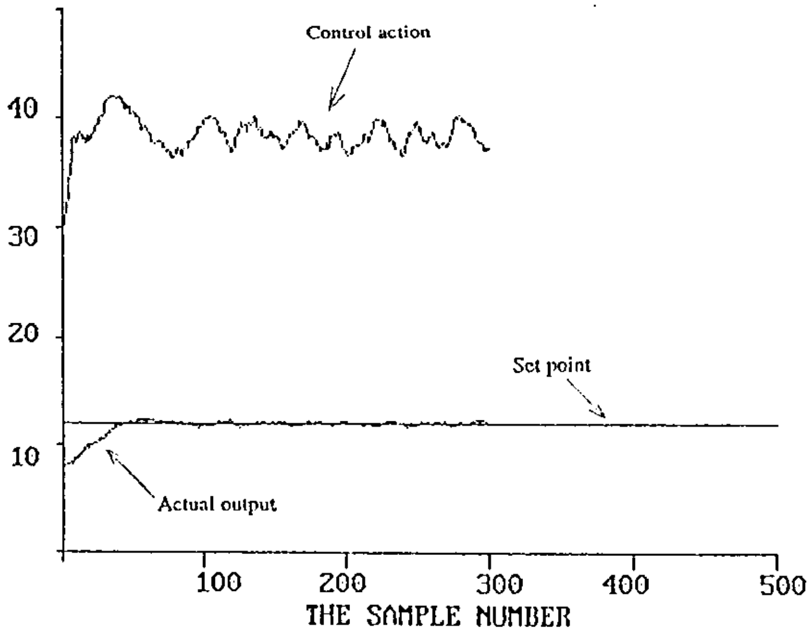


Fig. 3.7 Simulation process response using FLC with pure second order system

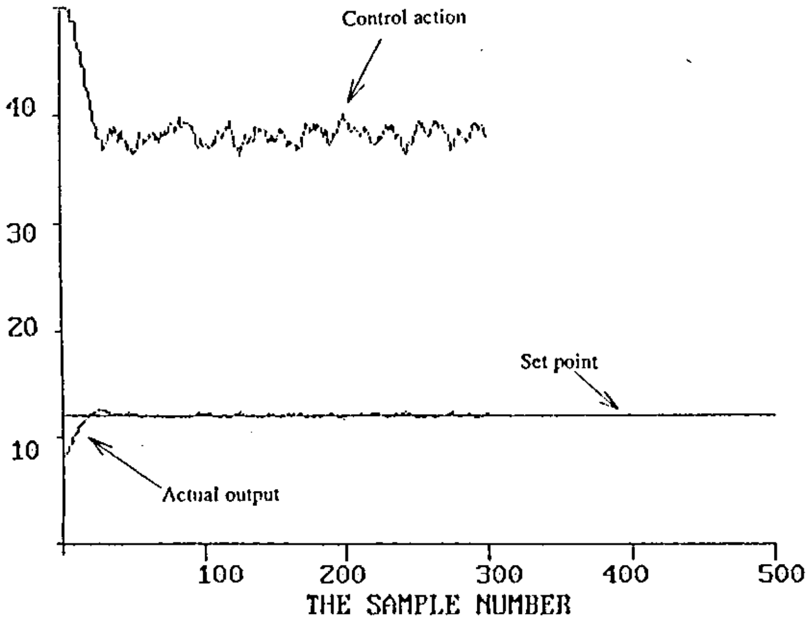


Fig. 3.8 Simulation process response using PID controller with pure second order system

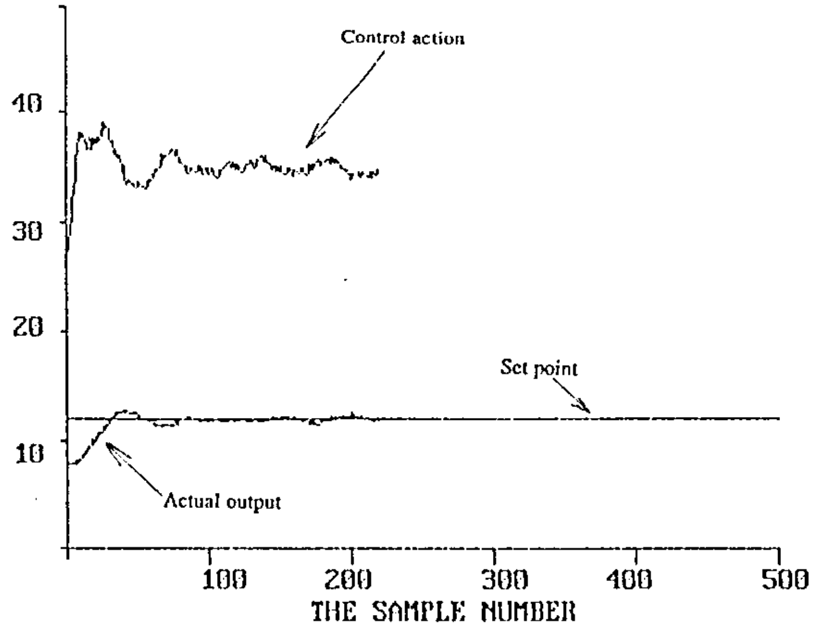


Fig. 3.9 Practical process response using FLC with pure second order system

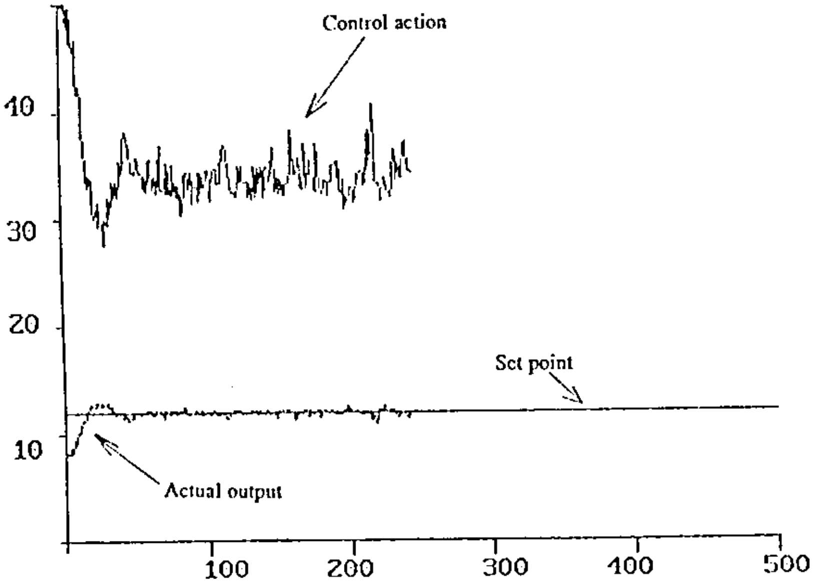


Fig. 3.10 Practical process response using PID controller with pure second order system

3.6.2 Disturbance (closing the drain tap) :

Since disturbances are a common behavior in most practical cases. The controller should reject its effect very quickly with less deviation from the set point to see the performance of the two controllers in response to disturbance in the process, the following experiment was conducted. This experiment is the same as the previous experiment except that at steady state the drain tap is closed for a period of 10 sec. This is simulated by making Q_0=zero at sample number 200 and 201.The simulation process responses using the FLC and PID controller are shown in Fig. 3.11 and Fig. 3.12 respectively. The other coupled tank parameters were not changed. While the practical responses are shown respectively in Fig. 3.13 and Fig. 3.14. From the simulation responses, it appears that the two controllers react to closing the drain tap by reducing the input flow Q_i differently. The FLC reacts slower than the PID controller, this causes a large undershoot in the case of the PID controller. A similar consequences are concluded from the practical responses. Analysing the practical responses, quantitively (presented in table 3.3) show that PID controller is more active than the FLC and the PID controller keeps the water level in the second tank closer to the set point than the FLC. However by comparing the difference between J_1 for the PID controller and FLC in this experiment with that of the previous experiment we can see that the difference is reduced.

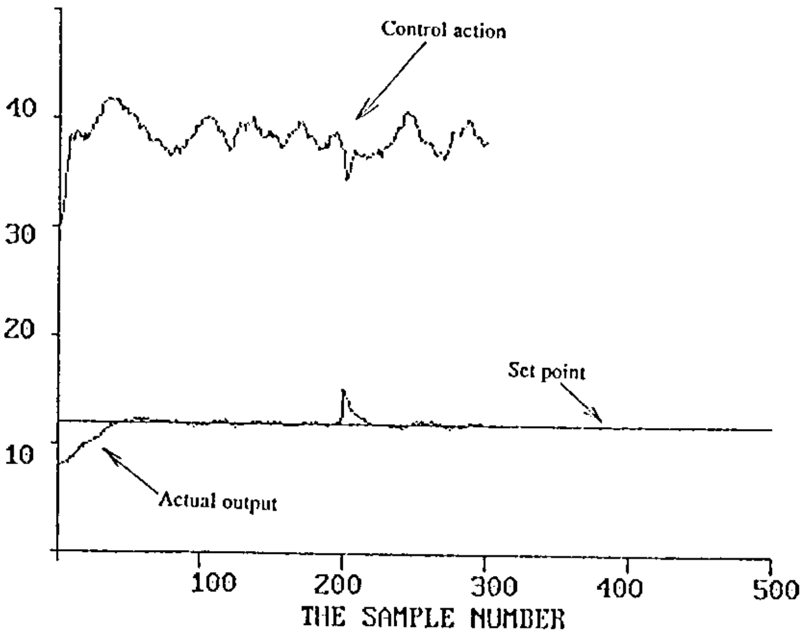


Fig. 3.11 Simulation process response using FLC with closing the drain tap (10 sec)

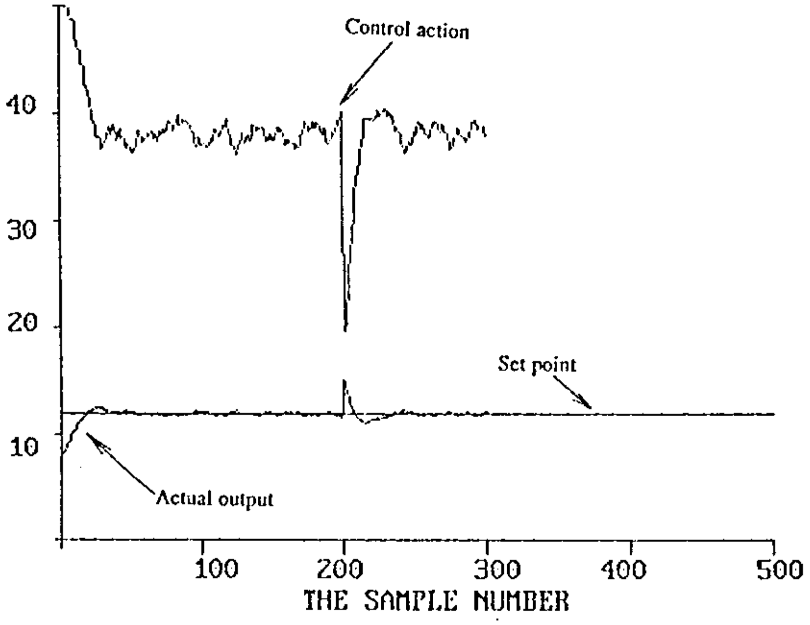


Fig. 3.12 Simulation process response using PID controller with closing the drain tap (10 sec)

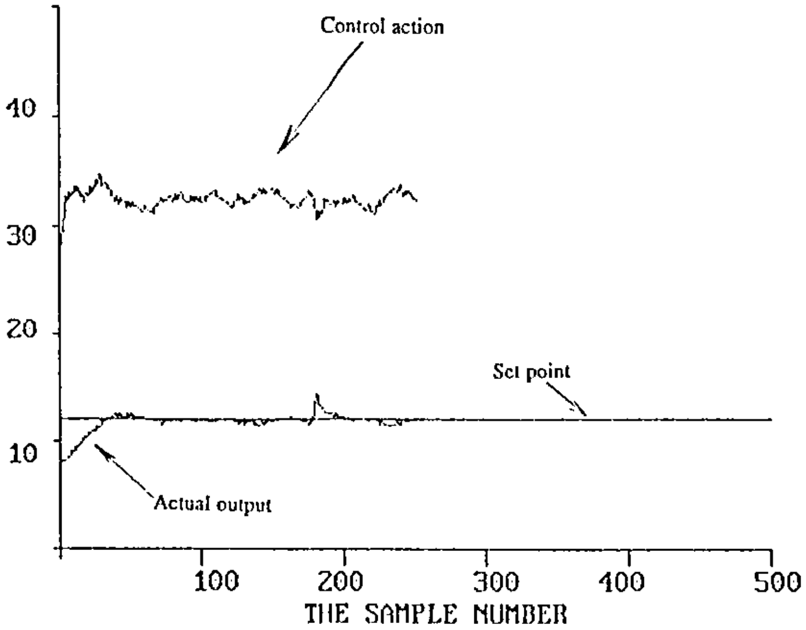


Fig. 3.13 Practical process response using FLC with closing the drain tap (10 sec)

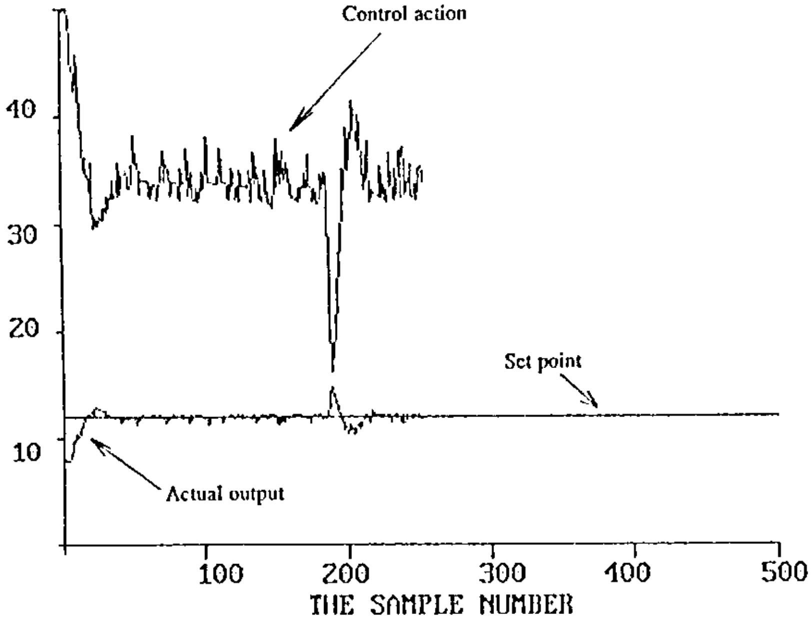


Fig. 3.14 Practical process response using PID controller with closing the drain tap (10 sec)

Table 3.3 Analysis table for second order system with disturbance

(close the drain tap).

Controller Type	NS	J_1	J_2
FLC	252	14.69	6.42
PID	252	13.38	39.60

3.6.3 Disturbance (adding water) :

In this experiment, it is required to test the performance of the two controllers for a different kind of disturbance. An amount of water that increases the water level by 4cm is added to the second tank. From the simulation process responses for the FLC and PID controller, shown respectively in Fig. 3.15 and Fig. 3.16, it is deduced that the PID controller is very active to adding water and an undershoot which is larger than that in the previous experiment is apparent. While the FLC responds to adding water slower than the PID controller and there is a very small undershoot. From the practical process responses, shown in Fig. 3.17 and Fig. 3.18, it can be seen that the FLC and PID controller are more active than in the simulation experiments, this is due to noise. In general, the conclusions for the simulation and practical experiments are the same.

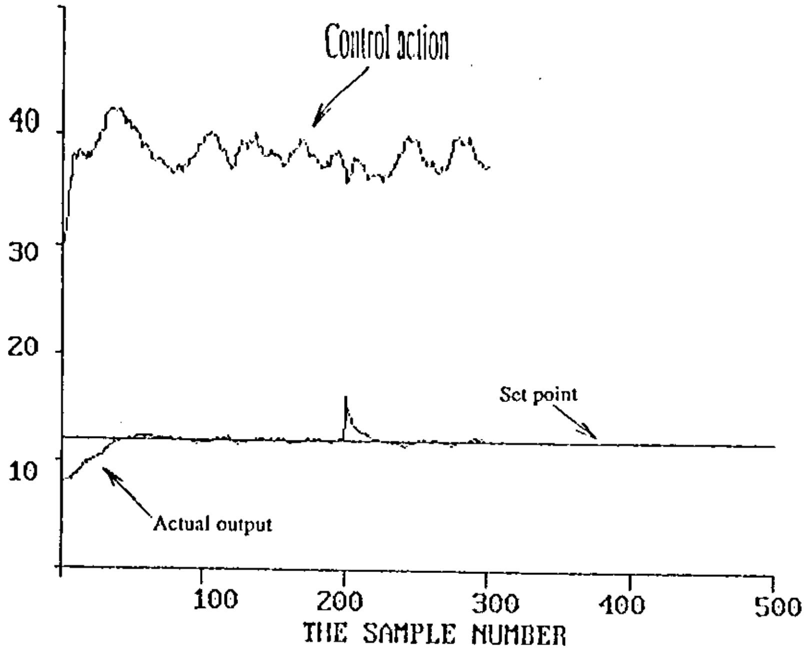


Fig. 3.15 Simulation process response using FLC in response to water addition (4cm)

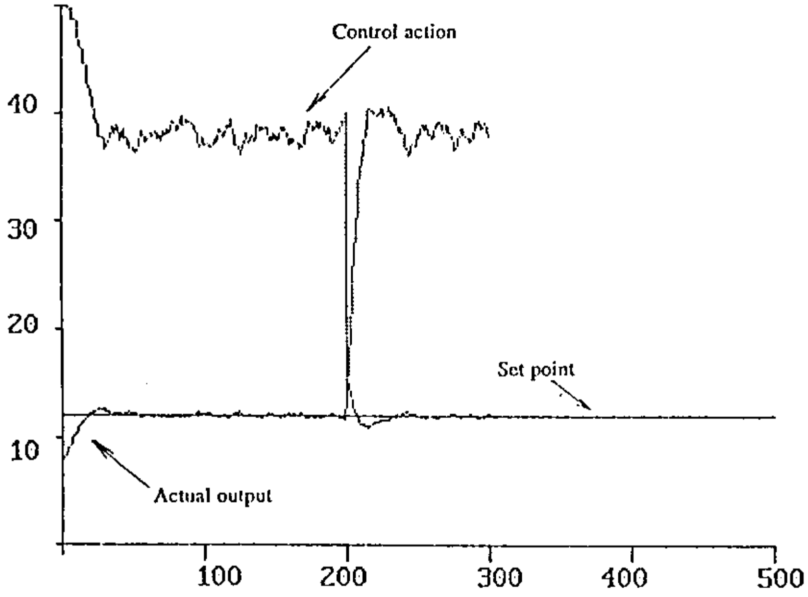


Fig. 3.16 Simulation process response using PID controller in response to water addition (4cm)

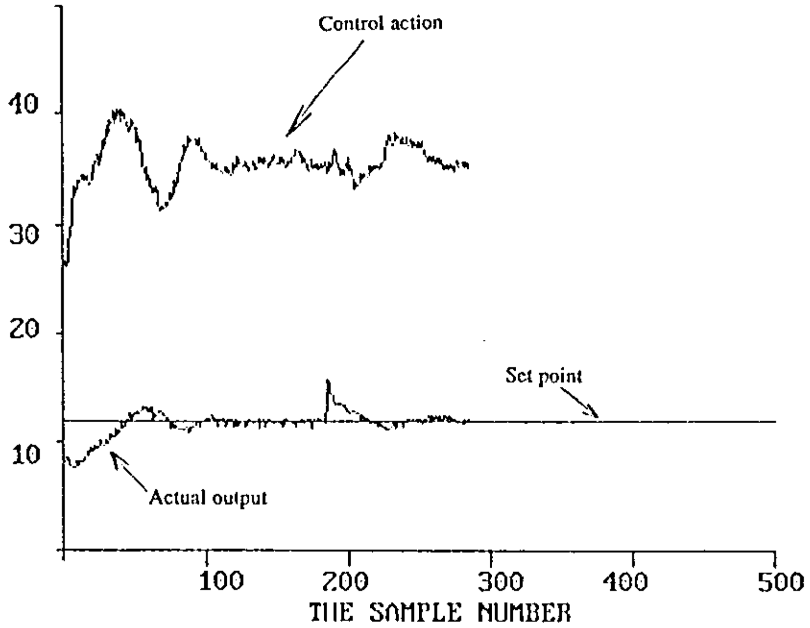


Fig. 3.17 Practical process response using FLC in response to water addition (4cm)

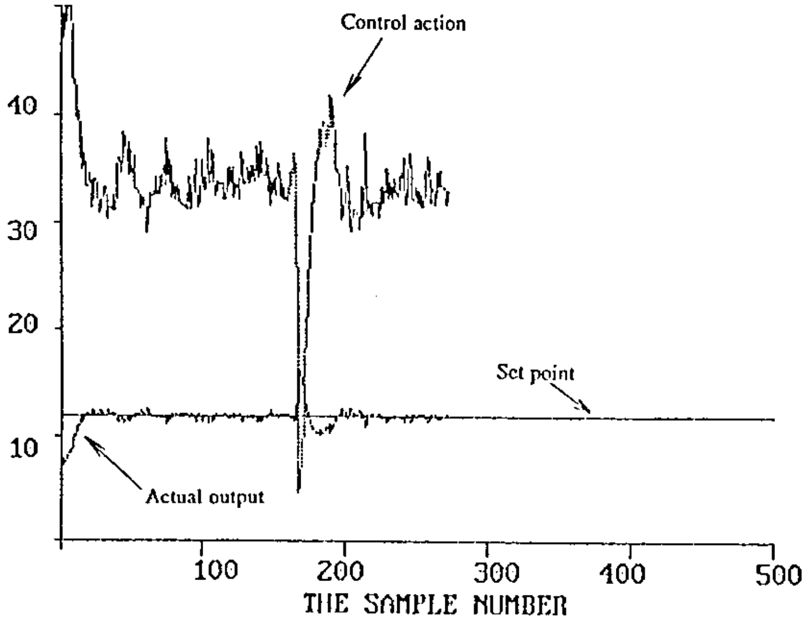


Fig. 3.18 Practical process response using PID controller in response to water addition (4cm)

3.6.4 First Order System :

In this experiment, it is required to study the performance of the two controllers in controlling the water level when all the bungs are removed (the process becomes a first order system), the coupled tank process parameters for this set up are shown in table 3.5.

Table 3.5 coupled tank system parameters for the first order system.

a_1 (cm^2)	2.3711
a_2 (cm^2)	0.3848
A (cm^2)	100
Cd_1 $\simeq$	0.5
Cd_2 $\simeq$	0.75

Fig. 3.19 and Fig. 3.20 show, respectively the process response using the FLC and PID controller in simulation. By comparing these responses with the process responses when the coupled tank is arranged as a second order system, shown in Fig. 3.7 and Fig. 3.8, it is deduced that the overshoot is reduced for both controllers.

The practical process responses using the two controllers are shown in Fig. 3.21 and Fig. 3.22. The process response using FLC seems to be slower than the process response using PID controller and there is no overshoot in the process response of the two controllers.

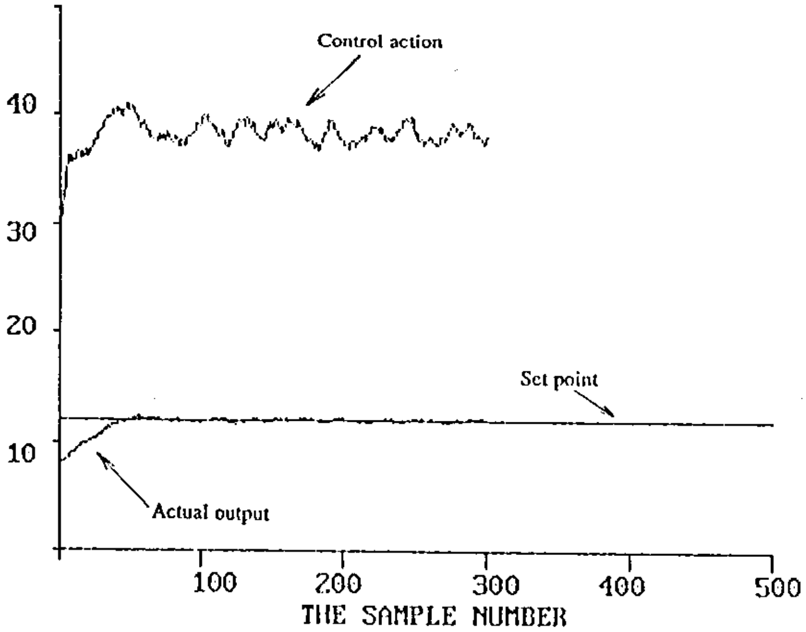


Fig. 3.19 Simulation process response using FLC with first order system.

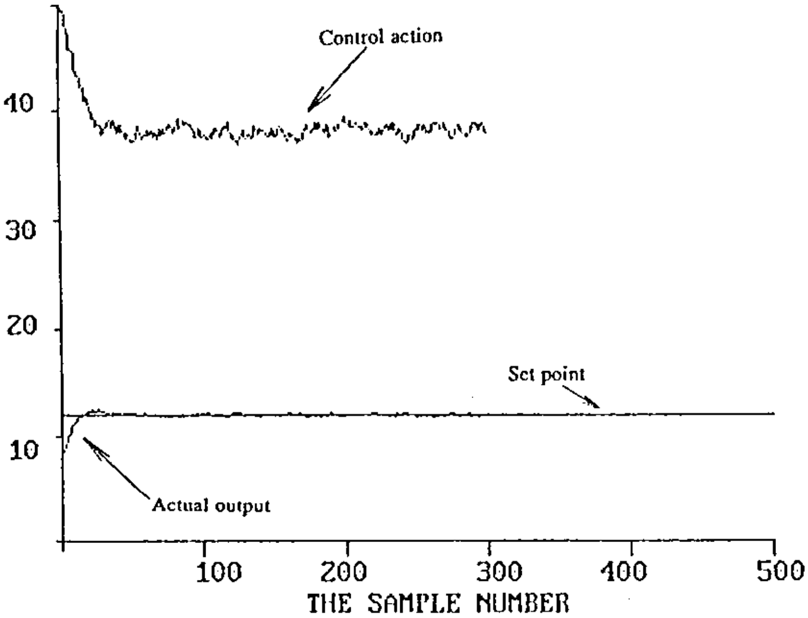


Fig. 3.20 Simulation process response using PID controller with first order system

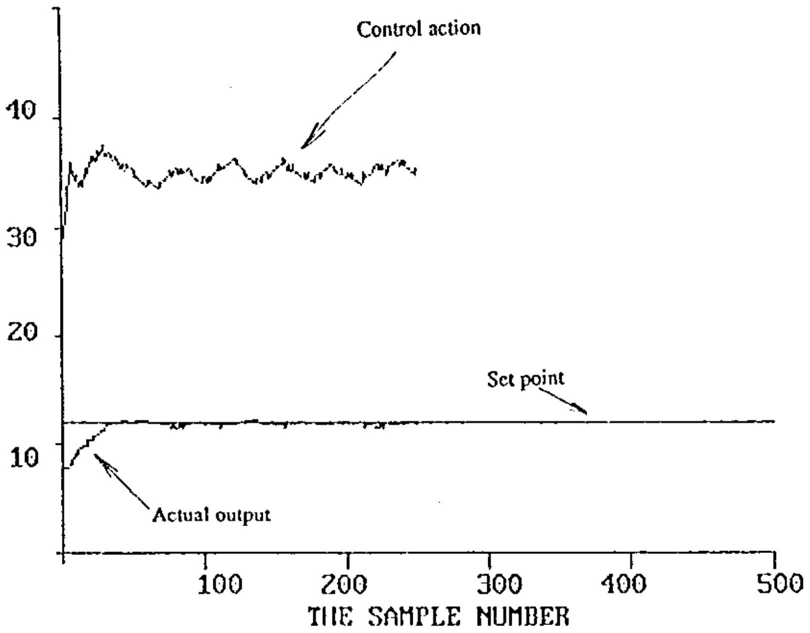


Fig. 3.21 Practical process response using FLC with first order system

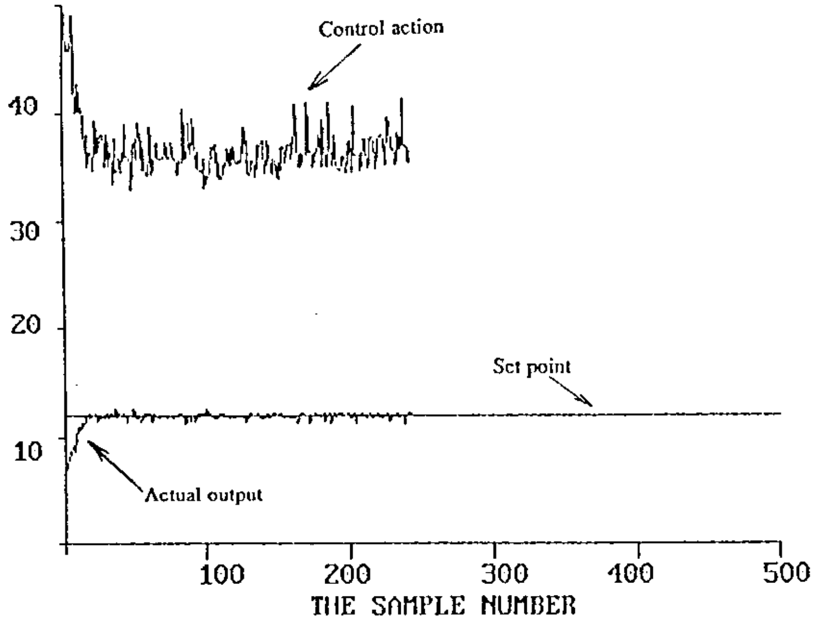


Fig. 3.22 Practical process response using PID controller with first order system

Table 3.6 Analysis table for first order system.

Controller Type	NS	J_1	J_2	Average at steady state	Variance at steady state
FLC	243	14.71	6.22	11.998	0.0424
PID	243	11.52	37.71	12.076	0.0610

From table 3.6, the PID is more active than the FLC while the PID keeps the water level closer to the set point than the FLC, this is due to the fact that the PID controller is faster than the FLC. At steady state the two controllers keep the water level close to the set point but the FLC response have a smaller variance at steady state.

3.6.5 Time-variant process :

Many practical processes are time-variant and any employed controller must take this variation into consideration. In this study the order of the system is changed at the beginning before the water level in the second tank reaches the set point to see the response of the two controllers under such conditions. The system is changed from second to first order one by removing the bungs at the required time.

Fig. 3.23 and Fig. 3.24 show the process response using the two controllers in simulation. From these figures, it is seen that at the moment of changing the order of the system, the water level in the second tank is increased suddenly (transient stage) until the water level in the two tanks,

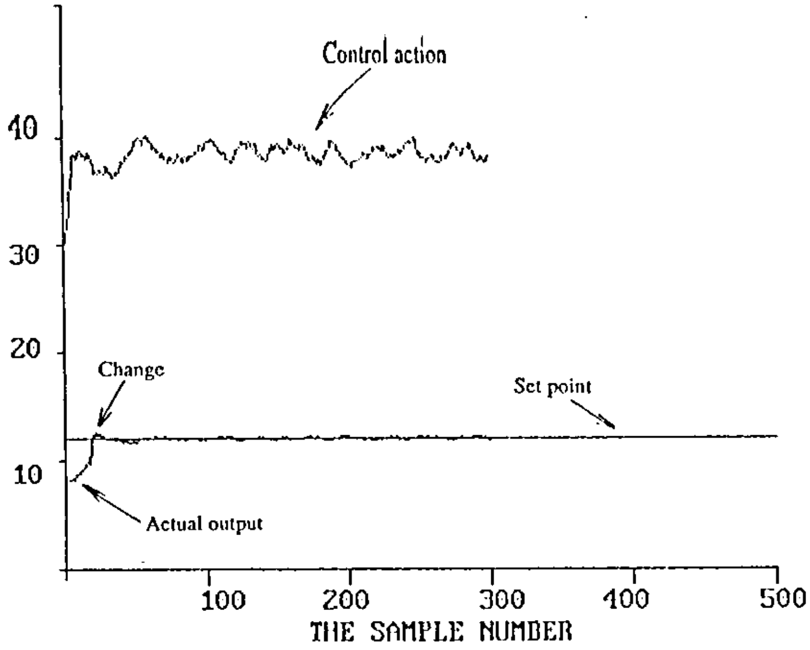


Fig. 3.23 Simulation process response using FLC with changing the order of the system at the beginning

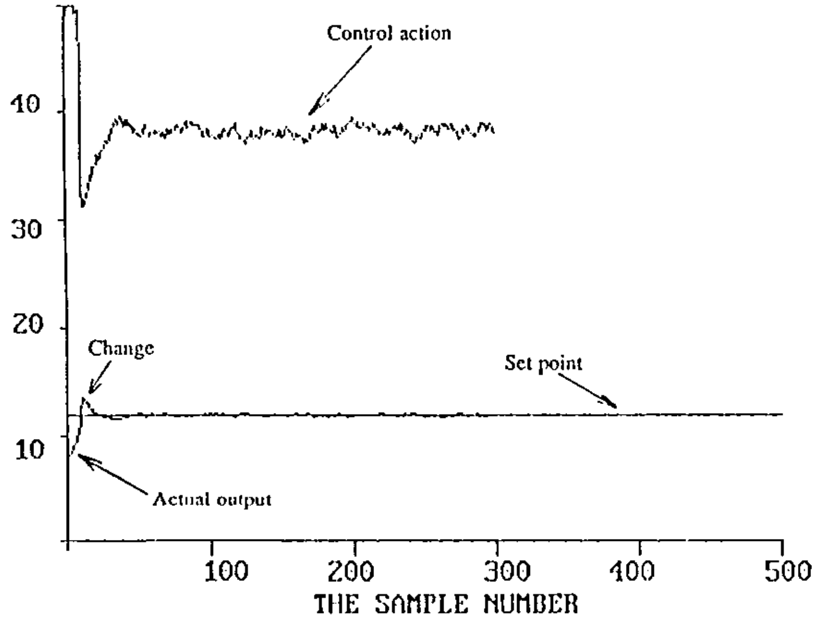


Fig. 3.24 Simulation process response using PID controller with changing the order of the system at the beginning

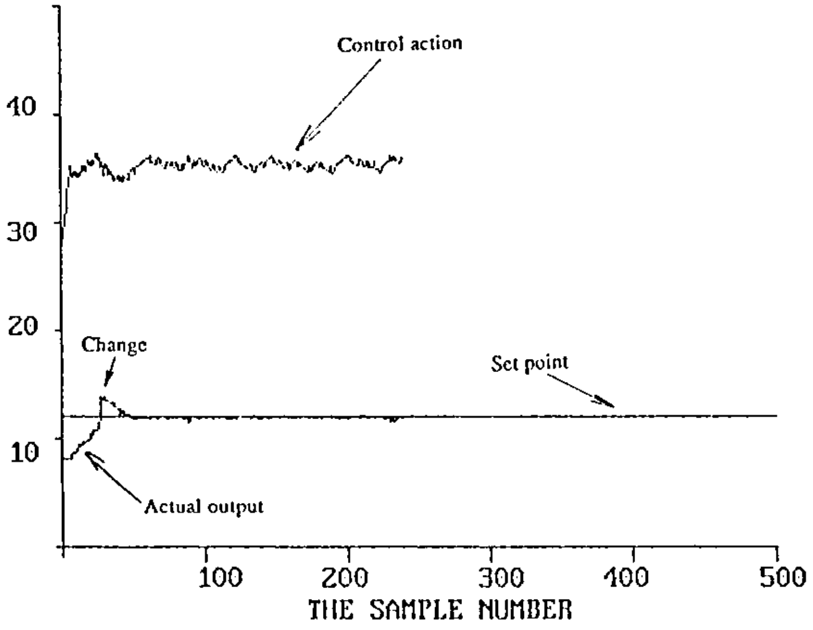


Fig. 3.25 Practical process response using FLC with changing the order of the system at the beginning

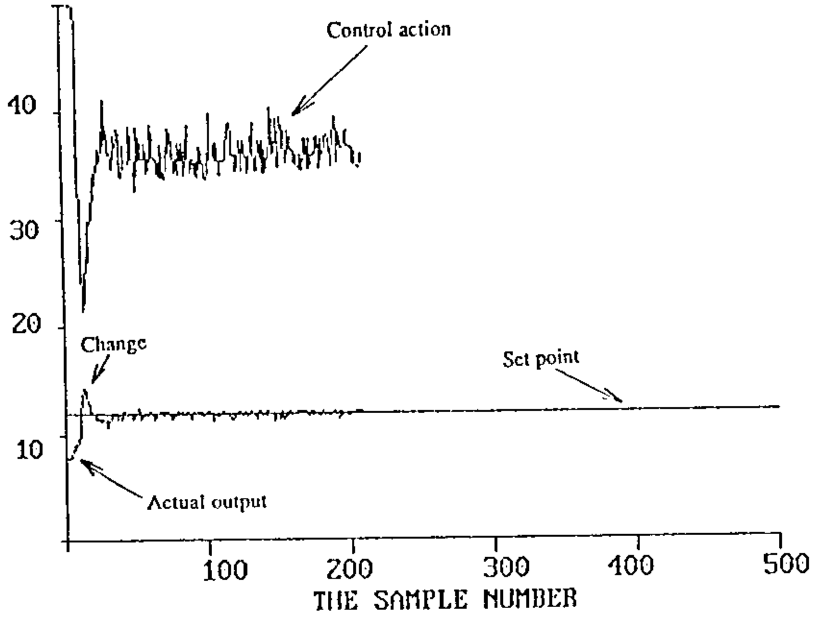


Fig. 3.26 Practical process response using PID controller with changing the order of the system at the beginning

become equal (steady state stage). Since at this stage the system becomes first order system (one big tank). The two controllers respond to changing the order of the system by decreasing the flow rate by different amounts.

The practical process responses using the two controllers are shown in Fig. 3.25 and Fig. 3.26. However the PID controller reacts sharply to changing the order of the system at the beginning, and there is a large undershoot as shown in Fig. 3.26. While the FLC responds smoothly to changing the order of the system at the beginning.

Table 3.7 Analysis table for time-variant process (changing the order of the system at the begining)

Controller Type	NS	J_1	J_2
FLC	210	15.47	6.30
PID	210	12.79	38.81

The values of J_1 and J_2 are shown in table 3.7, the PID controller is found to be more active the FLC and that the water level in the second tank is kept closer to the set point when the PID controller is used. Generally speaking, the two controllers were able to adapt to process time-variation.

3.6.6 Time variant process (change the order of the system at steady state) :

This experiment is the same as the previous experiment, but here the order of the system is changed at steady state. The PID controller fastly responds to changing the order of the system at steady state as shown in Fig. 3.28 and Fig. 3.30 while the FLC slowly responds as shown in Fig. 3.27 and Fig. 3.29. A large undershoot is produced in the case of the PID controller while in the case of the FLC there is a very small undershoot.

Table 3.8 Analysis table for time-variant process (change the order of the system at steady state).

Controller Type	NS	J_1	J_2
FLC	275	19.53	6.97
PID	275	13.25	39.54

The process responses using the two controllers are reasonable when the order of the system is changed at steady state. The PID controller is more sensitive and active than the FLC as deduced from table 3.8, but keeps the water level in the second tank closer to the set point.

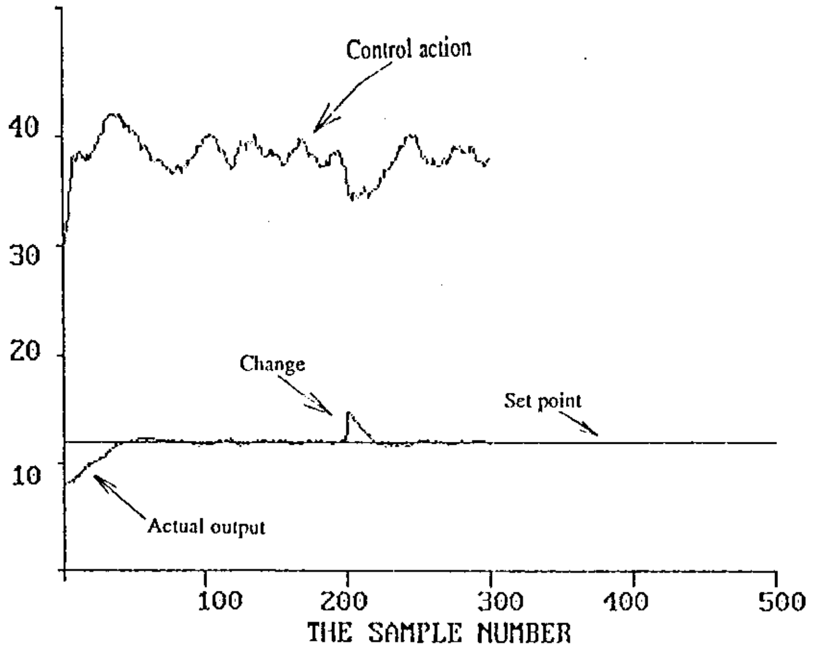


Fig 3.27 Simulation process response using FLC with changing the order of the system at steady state.

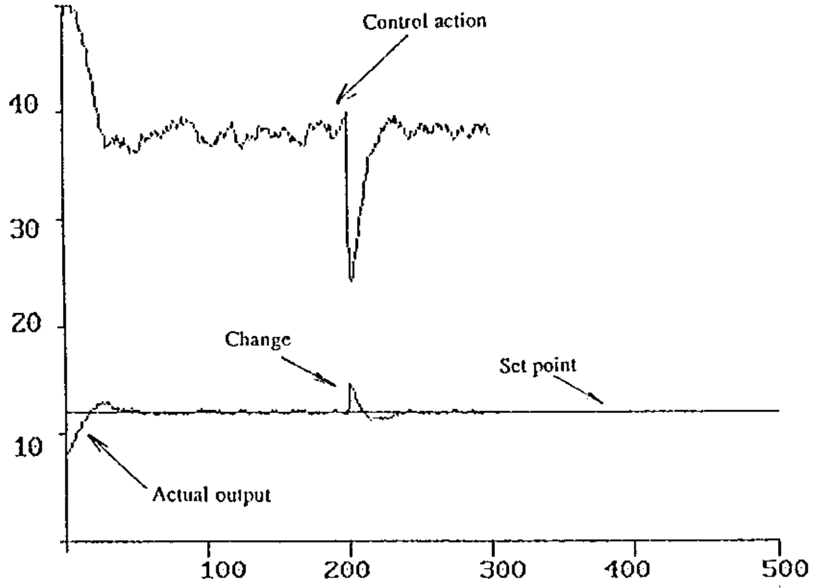


Fig. 3.28 Simulation process response using PID controller with changing the order of the system at steady state

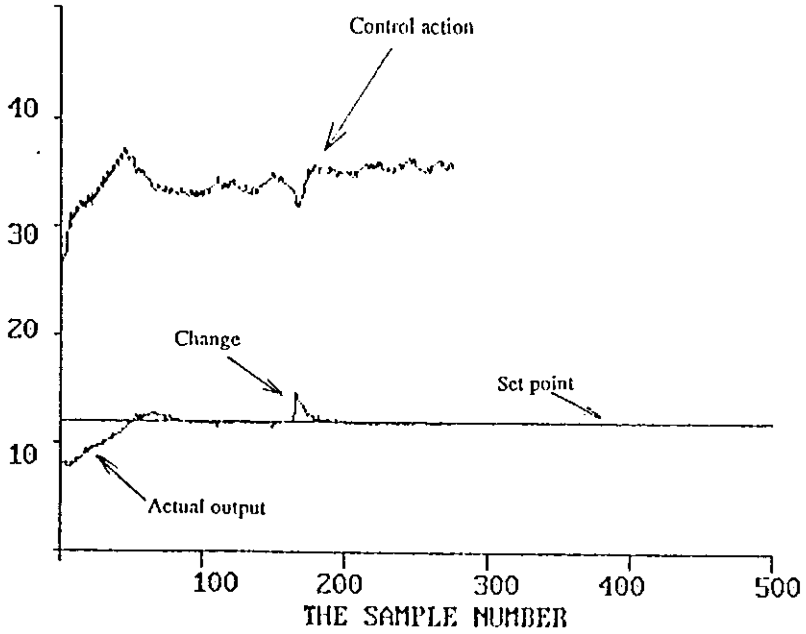


Fig. 3.29 Practical process response using FLC with changing the order of the system at steady state

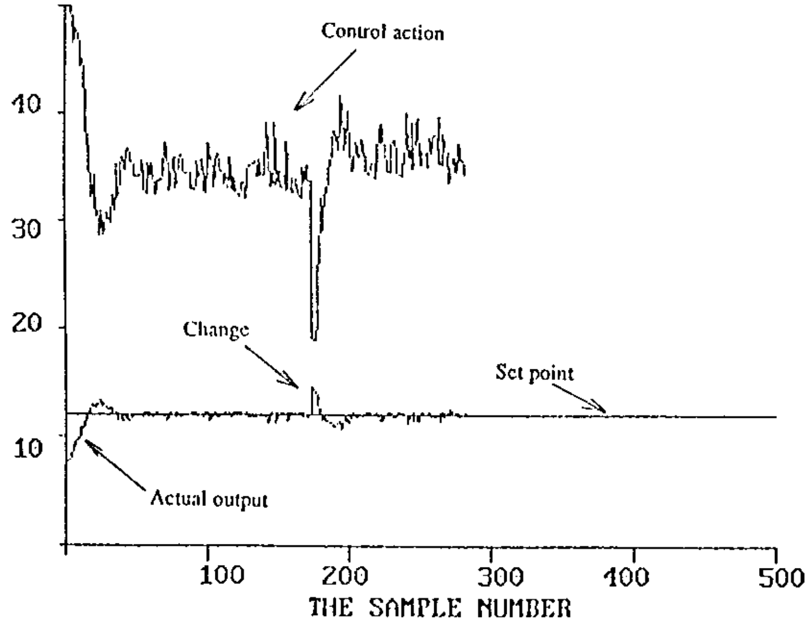


Fig. 3.30 Practical process response using PID controller with changing the order of the system at steady state

3.6.7 Non-linearity (Rising the water level in the second tank from 5cm to 10 cm) :

The FLC rules and subsets were chosen such that they are suitable to increase the water level from 8cm to 12cm. However to demonstrate the capabilities of the two controllers to respond to the non-linear behavior of the coupled tank, the controllers (with the previous parameters) were applied to increase the water level from 5cm to 10cm (i.e different conditions). Fig. 3.31 and Fig. 3.32 shows the simulation process responses using the FLC and the PID controller. It is seen that the process response using FLC has a slow response while in the case of PID controller it has a fast response but the process response using PID controller has an overshoot larger than that for the FLC. By comparing the flow rate at steady state for the second order system when the water level in the second tank is increased from 8 cm to 12 cm and the steady state flow rate for the same system when the water level in the second tank is increased from 5cm to 10 cm, it is noticed that the flow rate in the first case is larger than that in the second case.

The practical process responses for the two controllers are show in Fig. 3.33 and Fig. 3.34. From these figures, it is seen that the response of the process using the PID controller is faster than the response in the case of FLC and the PID controller is more active than the FLC. These qualitative conclusions confirm with the quantitative conclusions which are deduced from table 3.9.

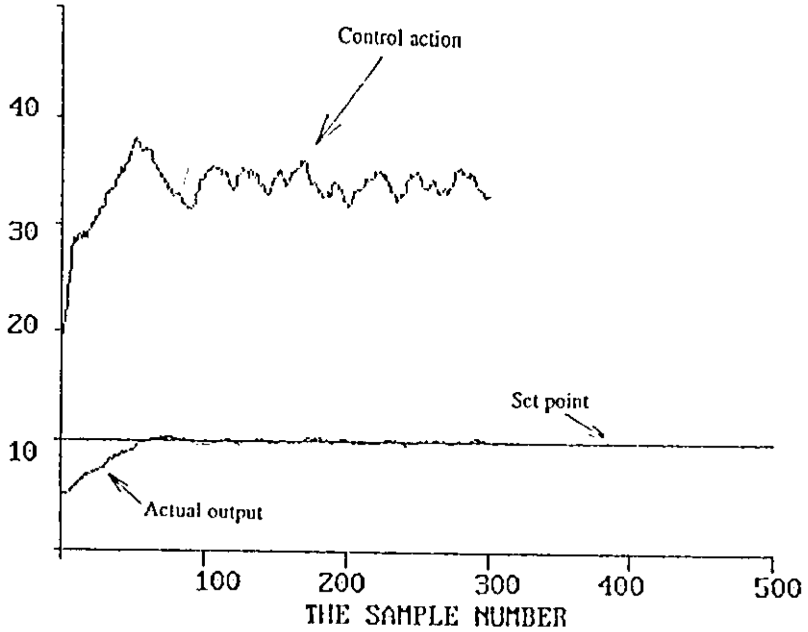


Fig. 3.31 Simulation process response using FLC with second order system for rising the level from 5cm to 10 cm

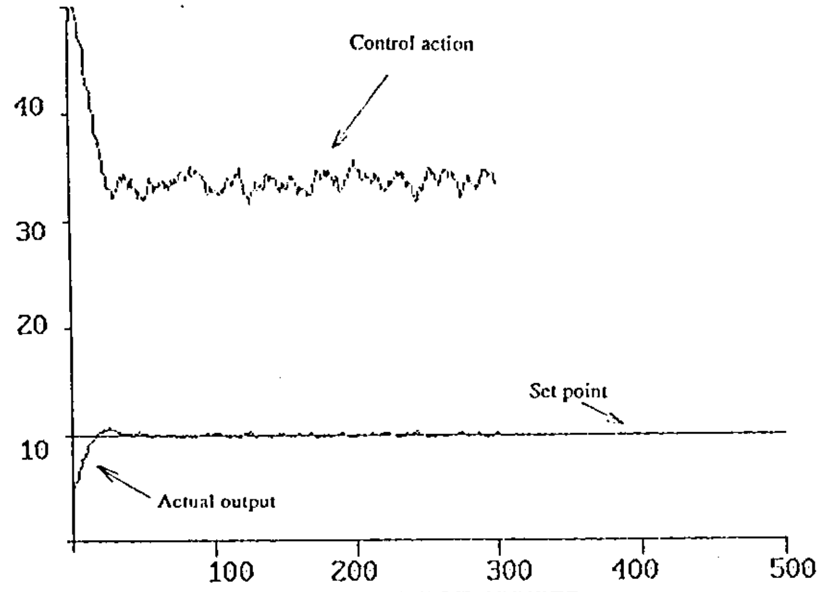


Fig. 3.32 Simulation process response using PID controller with second order system for rising the level from 5cm to 10 cm

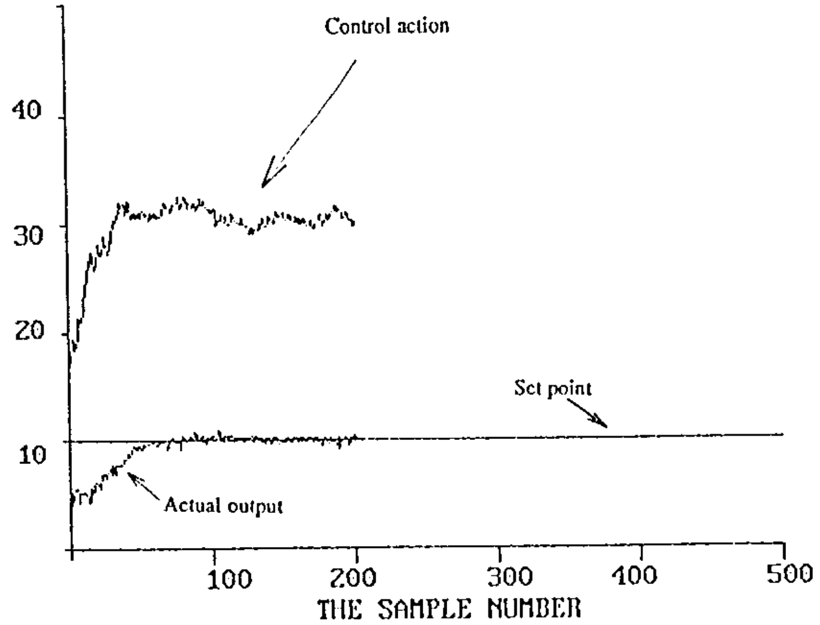


Fig. 3.33 Practical process response using FLC with second order system for rising the level from 5cm to 10 cm

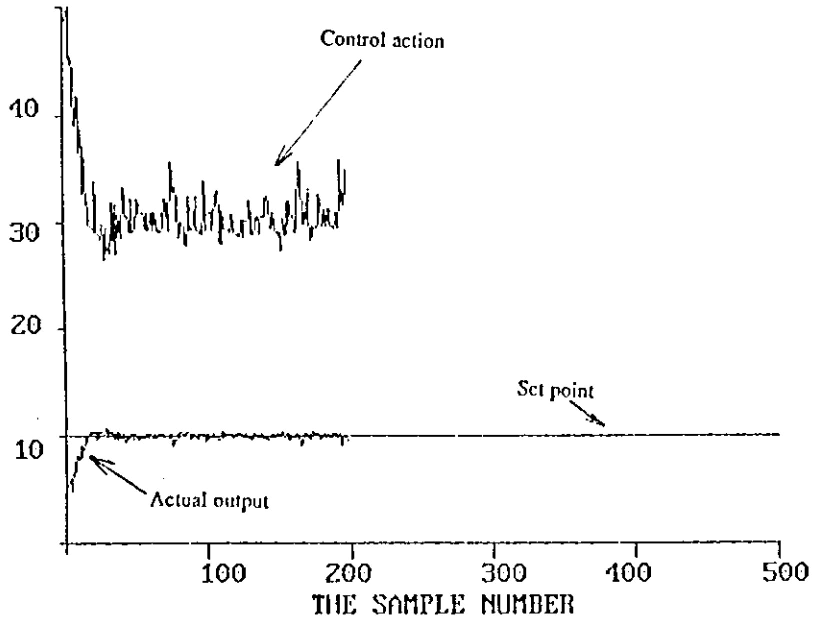


Fig. 3.34 Practical process response using PID controller with second order system for rising the level from 5cm to 10 cm

Table 3.9 Analysis table for the second order system (increasing the water level from 5cm to 10 cm).

Controller Type	NS	J_1	J_2	Average at steady state	Variance at steady state
FLC	197	26.71	7.68	10.126	0.0631
PID	197	12.60	40.50	10.096	0.0641

It is seen from this table that the two controllers variance at steady state are close to each other and the average at steady state is approximately the same and equal 10.10 cm. Although the two controller parameters are designed to increase the water level from 8cm to 12cm it can be deduced that the two controller have the capability to give adequate response at different operating conditions.

3.6.8 Non-linearity (Rising the water level in the second tank from 10 cm to 15cm) :

In this experiment, it is also required to study the effect of changing the operating conditions. The water level in the second tank is increased from 10 cm to 15 cm. The process responses for the FLC controller and PID controller in simulation are shown in Fig. 3.35 and Fig. 3.36. The flow rate at steady is shown to be larger than the flow rate in the previous two cases (i.e increasing the water level in the second tank from 8cm to 12 cm and from 5 cm to 10 cm). From the simulation responses, the process response for the

PID controller is faster than the response in the case of FLC, and the response using FLC has an overshoot while there is no overshoot in the case of PID controller. Fig. 3.37 and Fig. 3.38 shows the practical process responses for the two controllers. By comparing the practical responses for the FLC shown in Fig. 3.37 and Fig. 3.33, it is seen that the response for increasing the water level in the second tank from 10 cm to 15 cm is faster than the response for increasing the water level in the second tank from 5cm to 10cm, and while in the first case there is an overshoot, in the second case there is no overshoot.

However, comparing the practical responses for the PID controller shown in Fig. 3.38 and Fig. 3.34, shows that the response in the first case is slower than the response in the second case, and while in the first case there is no overshoot, in the second case there is a small overshoot.

The simulation and practical responses for the two controllers indicate that the response of the process using PID controller is faster and more active than the FLC which can be deduced quantitively from table 3.10.

Table 3.10 Analysis table for the second order system (increasing the water level from 10 cm to 15 cm).

Controller Type	NS	J_1	J_2	Average at steady state	Variance at steady state
FLC	185	19.62	6.94	15.19	0.0756
PID	185	16.38	25.44	15.22	0.0247

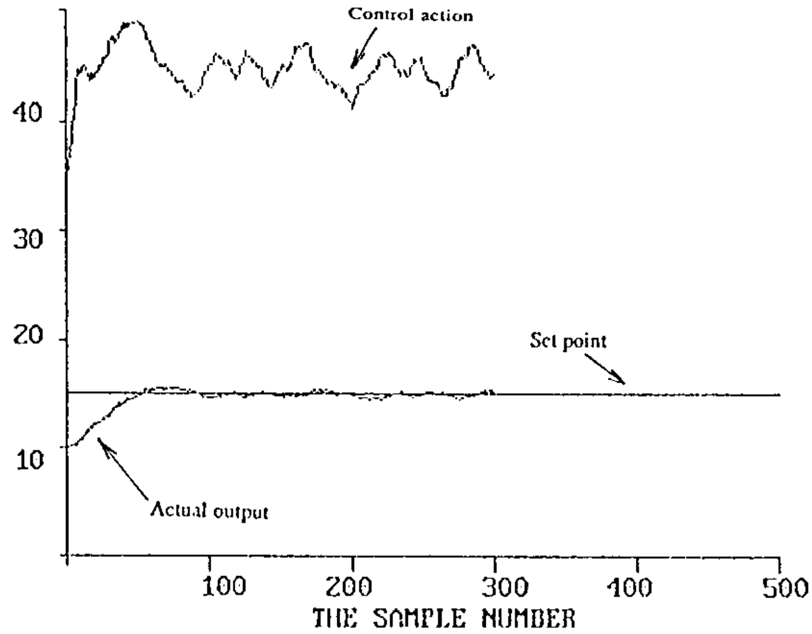


Fig. 3.35 Simulation process response using FLC with second order system for rising the level from 10 cm to 15 cm.

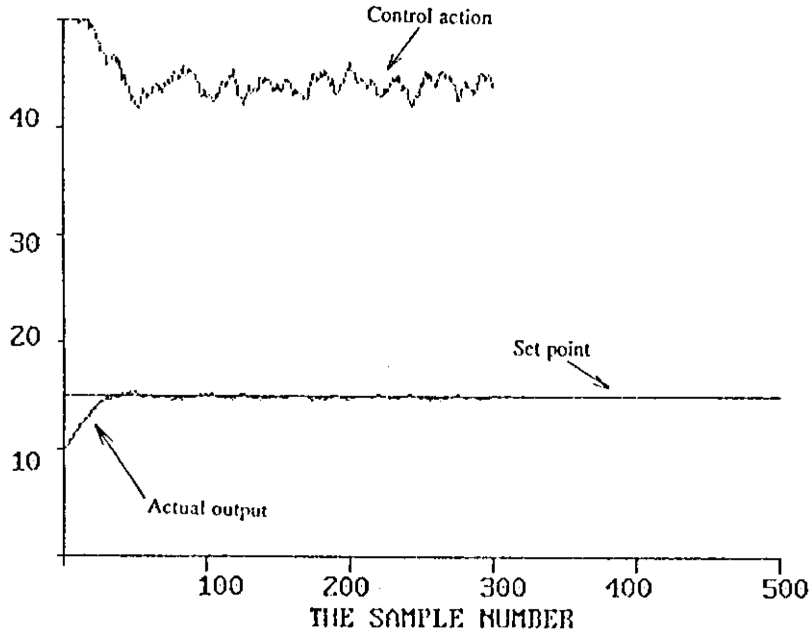


Fig. 3.36 Simulation process response using PID controller with second order system for rising the level from 10 cm to 15 cm.

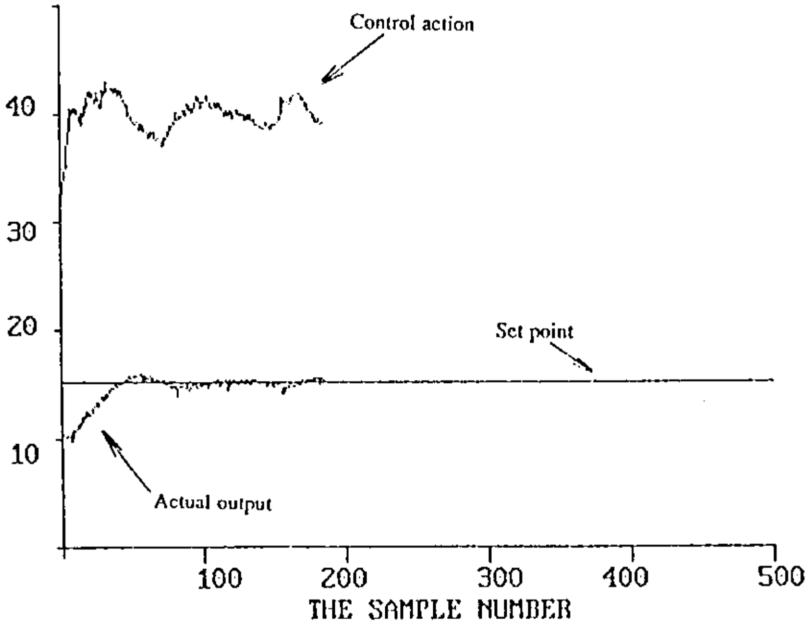


Fig. 3.37 Practical process response using FLC with second order system for rising the level from 10 cm to 15 cm

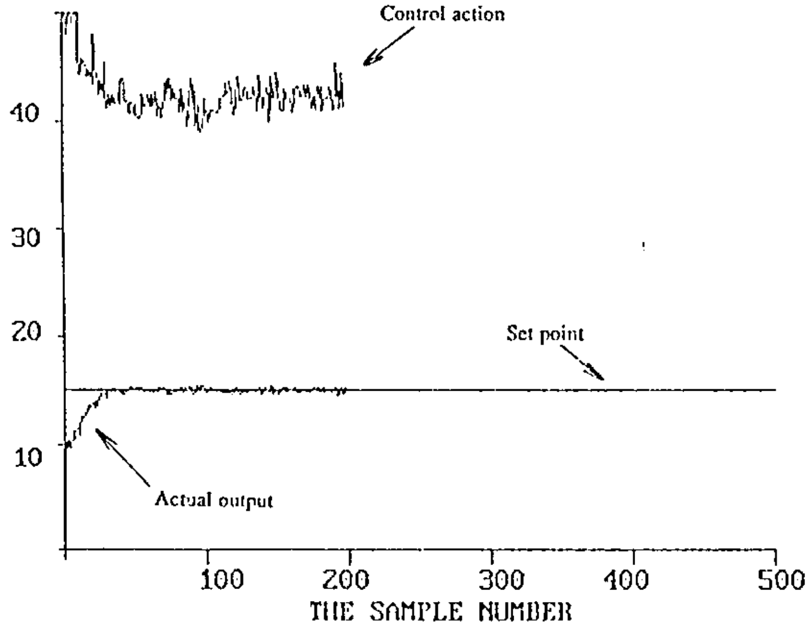


Fig. 3.38 Practical process response using FLC controller with second order system for rising the level from 10 cm to 15 cm

3.7 Conclusion :

From the results of the previous experiments, it was deduced that the two controllers give adequate control for different process characteristics. The performance of each controller can be modified by redesigning their parameters, for example, the PID controller can be made slower and less active while the FLC can be made faster and more active. However for the parameters chosen in this chapter for both controllers it can be deduced that the PID controller performs better than the FLC with faster response to set-point changes and disturbances.

Chapter 4

Adaptive Fuzzy Logic Controller

Adaptive Fuzzy Logic Controller :

As mentioned before the FLC is formed usually from operator information, observations and experimental work. As shown in Fig. 4.1, to form this controller the following stages are required :

1. Collect information from operators and observation.
2. Compile and organize data for implementation.
3. Implement in computer.
4. Apply and examine algorithm.
5. Check and modify.

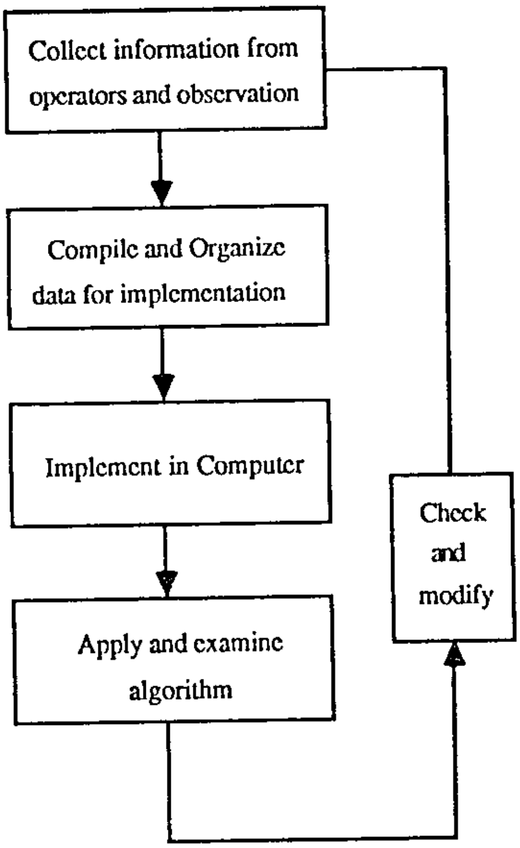


Fig. 4.1 Forming a FLC [1]

The final version of a FLC used requires the developer to interactively trying the controller and by monitoring the results modify the rules of the FLC. However, this requires a continuous intervention of the user in the controller design which may be not possible from practical point of view. Furthermore, it is time consuming and not an easy task. Hence, an automatic way of modifying the basic rules given by the operator is required.

4.1 FLC Modification :

The FLC can be modified by either modifying the definitions of the fuzzy subsets or modifying the linguistic rules. It was decided to obtain an adaptive algorithm which modifies the linguistic rules and keeps the definition of the fuzzy subsets fixed. This is done since the operator modifies his actions without redefinning the fuzzy subsets which don't have a clear limits anyway. Furthermore modifying a rule have a significant effect on the performance and easier to change in comparison with changing the fuzzy subsets.

4.2 Adaptive Algorithm :

The basis of this adaptive algorithm is to have a desired trajectory zone which is defined through experience, this trajectory zone is used through this algorithm as a reference trajectory. A zone was chosen rather than a single path since rule modification shouldn't depend on noise. Furthermore due to the nature of the FLC (i.e A rule could be applied to various conditions) it is not possible to have rules which give a single desired path.. It is required to make the response of the process to be within this trajectory zone, if this is not the case modification of the rules is necessary. Fig. 4.2 illustrates the operation of this adaptive algorithm.

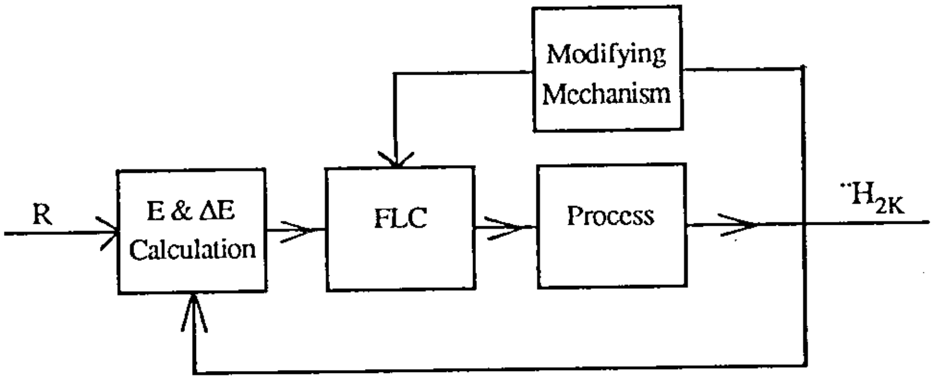


Fig. 4.2 Adaptive algorithm operation

As shown in Fig. 4.3 the basic idea used in the adaptive FLC is that if the actual output is outside the trajectory zone, then only the level is used to modify rules. However if the actual output is inside the trajectory zone then only the change in level (i.e $H_{2k} - H_{2k-1}$) is used. The level is used to

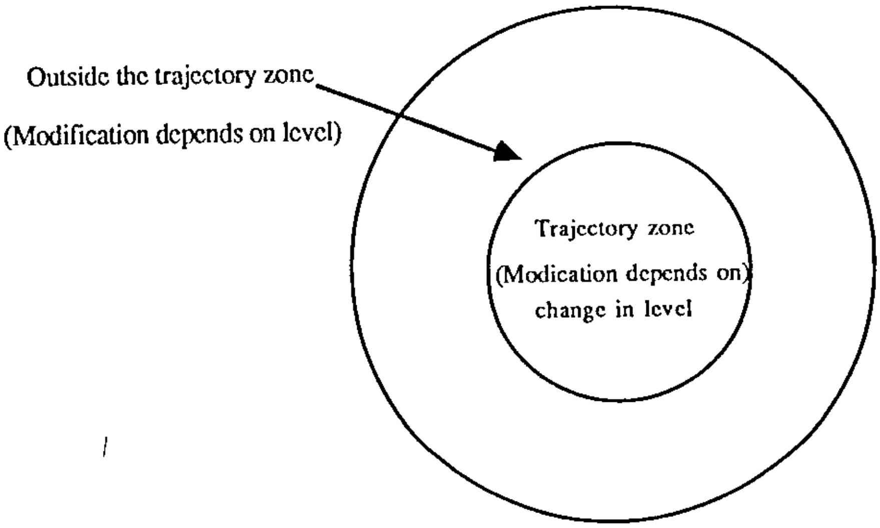


Fig. 4.3 Working mechanism

determine if the control action is required to be increased or decreased in order to force the actual output to be inside the trajectory zone, while the change in level is used to prevent the actual output from going outside this zone by increasing or decreasing the control action.

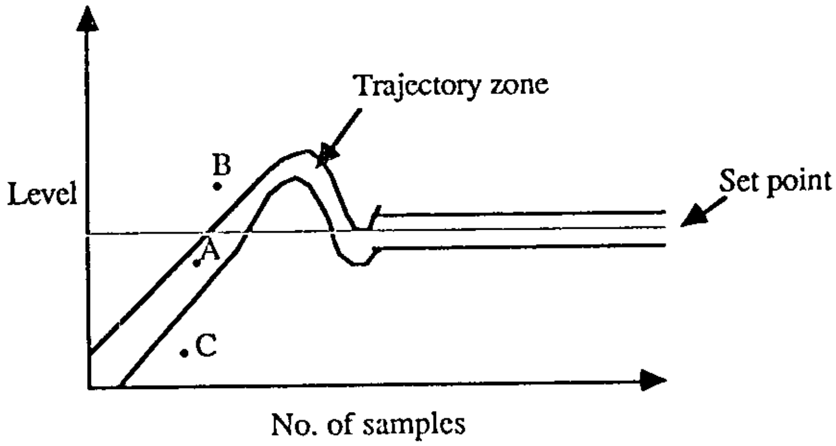


Fig. 4.4 Modification mechanism illustration figure

The trajectory zone is defined by specifying the desired single path through experience, then the zone is defined by specifying an acceptable deviation around this path. Using Fig. 4.4 there are five cases to be studied in order to see if modifications are needed on the rules.

Case 1 :

If the actual output is inside the trajectory zone, point A, and the change in level is within the desired change in level ± 0.1, (this allowance number is obtained through experience) there is no need to modify the rules.

Case 2 :

In this case, the actual output represented by point B is outside the trajectory zone and above it. The rules which are used to give the control action at this sample are tested to find the rule of the maximum action fuzzy subset (Note, the fuzzy subset are arranged on a descending order, for example the fuzzy subsets for the action which are used in simulation are arranged on a descending order as PH, PM, PL, NL, NM, and NH). The fuzzy subset of this rule is decreased one fuzzy subset, for example, if the action fuzzy subset is PM, it is decreased one fuzzy subset and becomes PL.

Case 3 :

When the actual output (point C) is outside and below the trajectory zone, the rule of the minimum action fuzzy subset is obtained, then the action fuzzy subset for this rule is increased one fuzzy subset, for example, if the action fuzzy subset is PM, it is increased one fuzzy subset and becomes PH.

Case 4 :

In this case, the actual output represented by point A is inside the trajectory zone and the actual change in level is greater than the desired change in level +0.1 cm. The modification procedure used in case 2 is applied for this case.

Case 5 :

If the actual output like point A is inside the trajectory zone and the actual change in level is less than the desired change in level -0.1 cm, the same modification procedure used in case 3 can be applied for this case.

A flow chart of the adaptive algorithm is shown in Fig. 4.5. At the end of each run the algorithm produces new linguistic rules which are used in the next run. Each rule is allowed to be changed one time through each run, this means that the rules may be changed several times but in different runs. The final version of the linguistic rules is reached through several runs of the adaptive algorithm.

It should be noted that the old and new rules are saved in separate files.

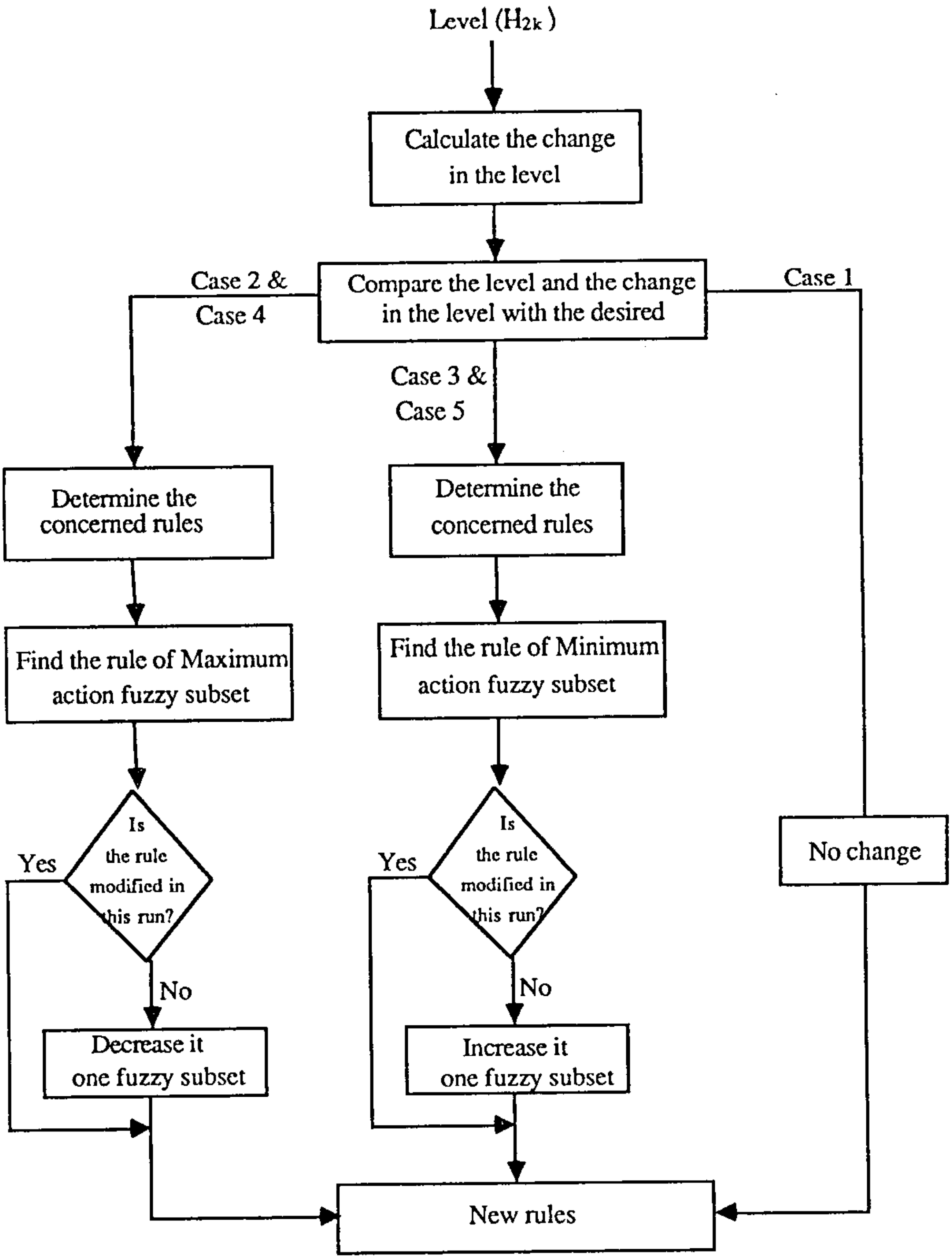


Fig. 4.5 Adaptive aglorithm

4.3 Adaptive Algorithm Application :

The adaptive algorithm is applied to the second order coupled tank system initially in simulation, the fuzzy subset definitions are the same as those shown in Fig. 3.4. In order to test the adaptive FLC, the action of the initial rules was put to be totally unacceptable and would give a very bad response at the first run. To achieve that the control action fuzzy subsets was chosen to be PL for all the rules, the initial rules are shown in Fig. 4.6 (a). The initial response of the process is shown in Fig. 4.7 (a). From this figure, it can be seen that the response is totally unacceptable and far away from the desired trajectory zone.

It can be deduced from Fig. 4.7 (a)-(k) that the process response is improved continuously from the first run until the sixth run. At the sixth run, (Fig. 4.7 (f)), the process response seems to be worse than the previous run (Fig. 4.7 (e)). This is due to the fact that the FLC tries to reduce the overshoot by reducing the control action, this will decrease the water level less than the desired set point and the rules responsible in this region until now are not changed and not appropriate. After this run the adaptive algorithm tries to improve the performance by changing these rules. A continuous improvement in the performance of the FLC can be seen from the seventh run, (Fig. 4.7 (g)), until the tenth run, (Fig. 4.7 (j)).

The final version of the rules are shown in Fig. 4.6 (b), the numbers inside the brackets are the number of modifications for each rule, it is noticed from this figure that some of the rules are not changed, while other rules are changed more than one time.

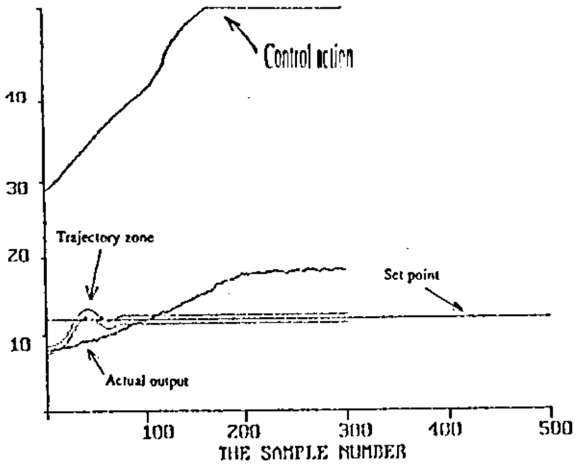
Error Action Change in error	PH	PM	PL	NL	NM	NH
PH	PL	PL	PL	PL	PL	PL
PM	PL	PL	PL	PL	PL	PL
PL	PL	PL	PL	PL	PL	PL
NL	PL	PL	PL	PL	PL	PL
NM	PL	PL	PL	PL	PL	PL
NH	PL	PL	PL	PL	PL	PL

a- Initial rules

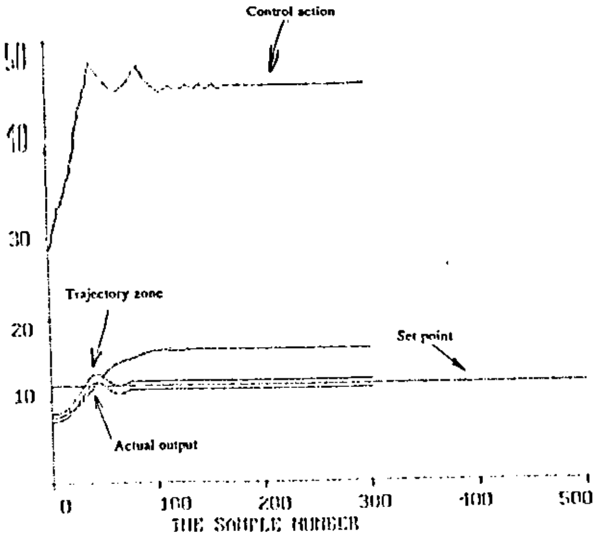
Error Action Change in error	PH	PM	PL	NL	NM	NH
PH	PL (0)	PM (1)	PH (2)	PH (2)	PL (0)	PL (0)
PM	PL (0)	PH (2)	PH (2)	NM (4)	NH (3)	PL (0)
PL	PL (0)	PH (2)	PM (1)	NL (1)	NH (3)	NL (1)
NL	PM (1)	PH (2)	PL (0)	NM (2)	NH (3)	NM (2)
NM	PL (0)	PH (2)	NL (5)	NH (5)	NH (3)	NM (2)
NH	PL (0)	PL (0)	NL (1)	NL (1)	PL (0)	PL (0)

b- Final version of rules

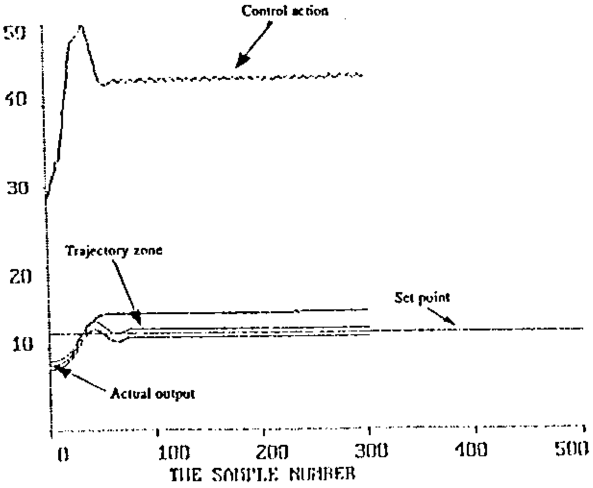
Fig. 4.6 The initial and final rules produced in simulation



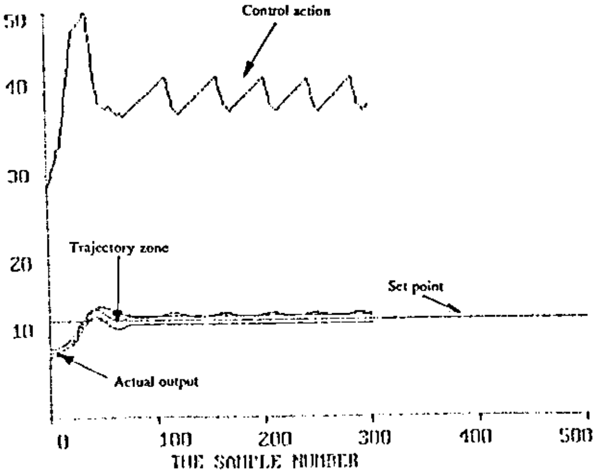
a- Run No. 1



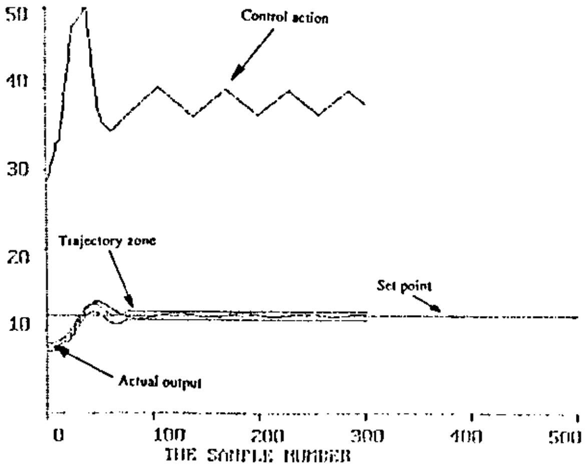
b- Run No. 2



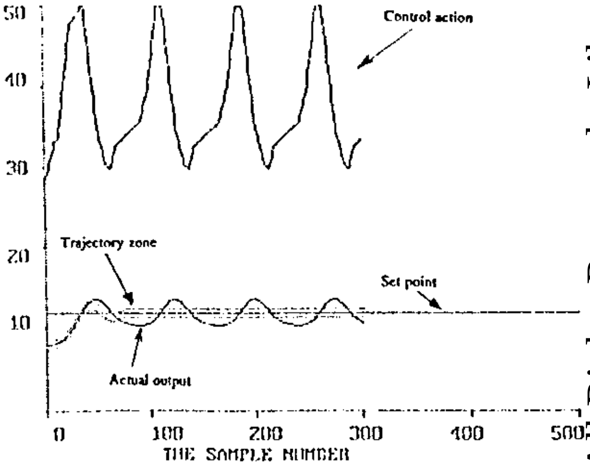
c- Run No. 3



d- Run No. 4

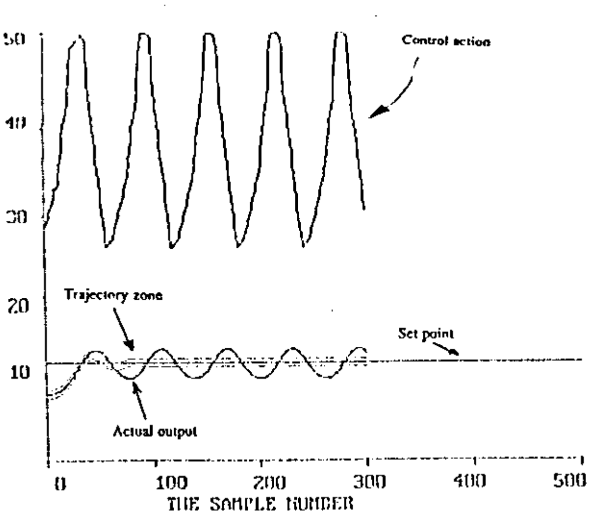


e- Run No. 5

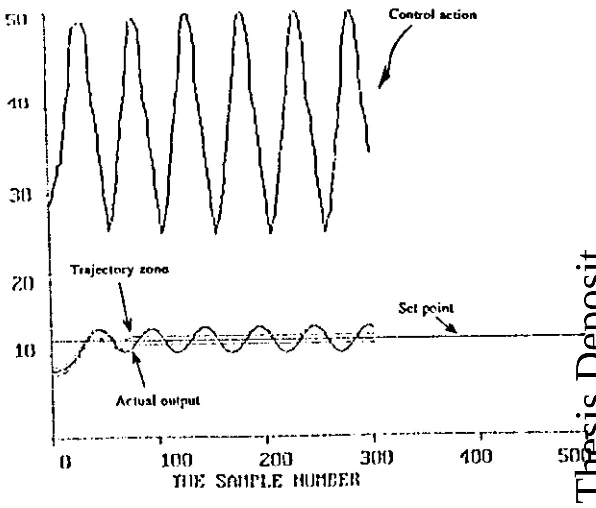


f- Run No. 6

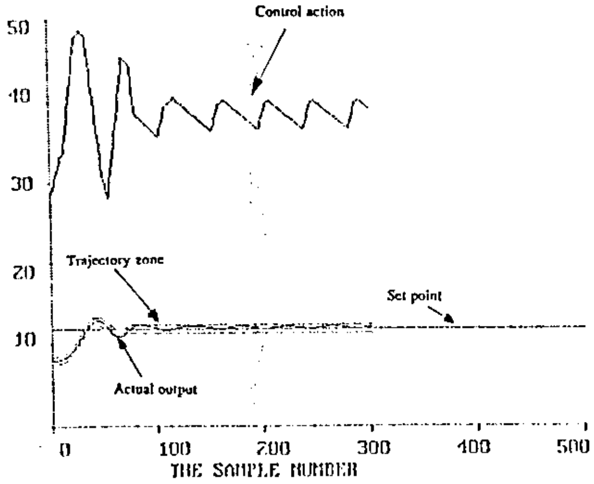
Fig. 4.7 Process response for each run (simulation)



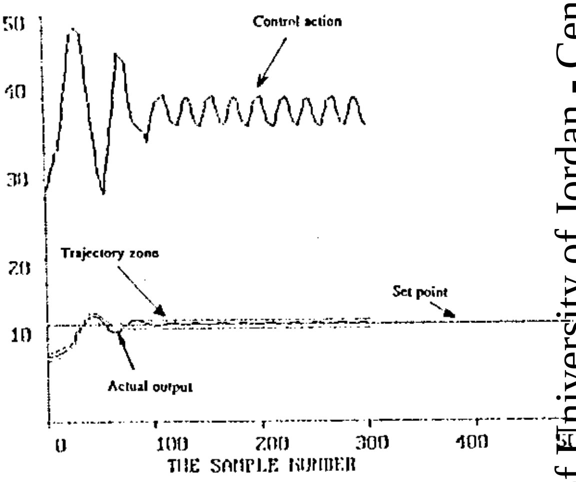
g- Run No. 7



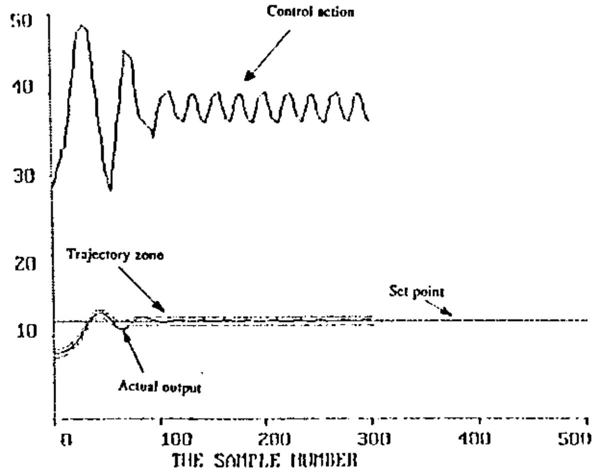
h- Run No. 8



i- Run No. 9



j- Run No. 10



k- Run No. 11

Fig. 4.7 continued

The adaptive aglorithm is also applied practically to the second order coupled tank system. The fuzzy subsets definitions used previously are not changed, shown in Fig. 3.4. The initial rules are chosen in simulation to give unacceptable performance but not far away from the desired performance for safety reasons. These initial rules are shown in Fig. 4.8 (a). The initial process response due to these rules is shown in Fig. 4.9 (a). It can be seen that it is slower than the desired response and have an oscillatory behavior.

Using the adaptive algorithm improves the performance of the FLC continuously as deduced by comparing the process responses, shown in Fig. 4.9 (a)-(i). The final version of the rules is shown in Fig. 4.8 (b).

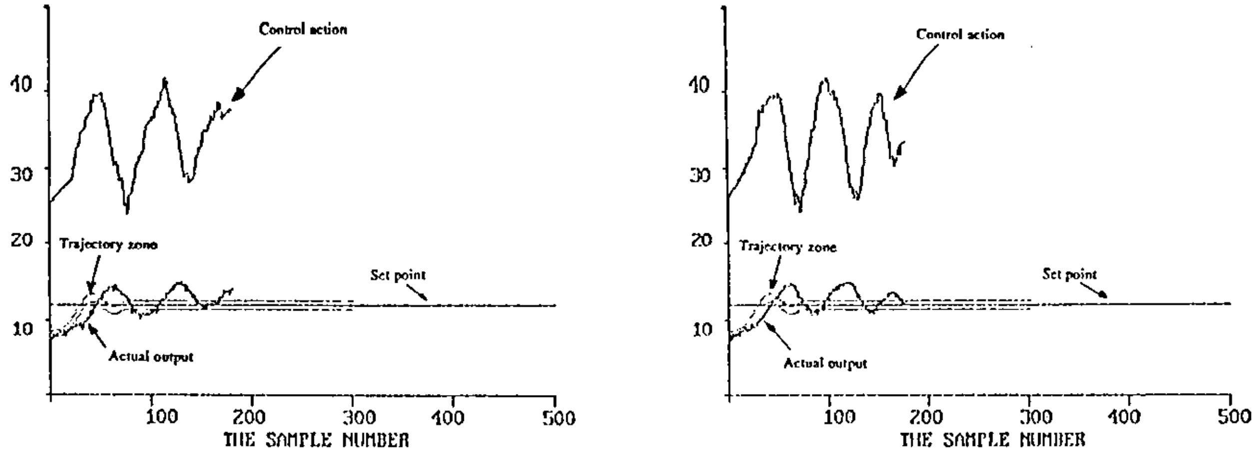
Error Action Change in error	PH	PM	PL	NL	NM	NH
PH	PL	PM	PH	PH	PL	PL
PM	PL	PH	PH	NM	NH	PL
PL	PL	PH	PM	NL	NH	NL
NL	PM	PH	PL	NL	NH	NM
NM	PL	PH	NL	NH	NH	NM
NH	PL	PL	NL	NL	PL	PL

a- Initial rules

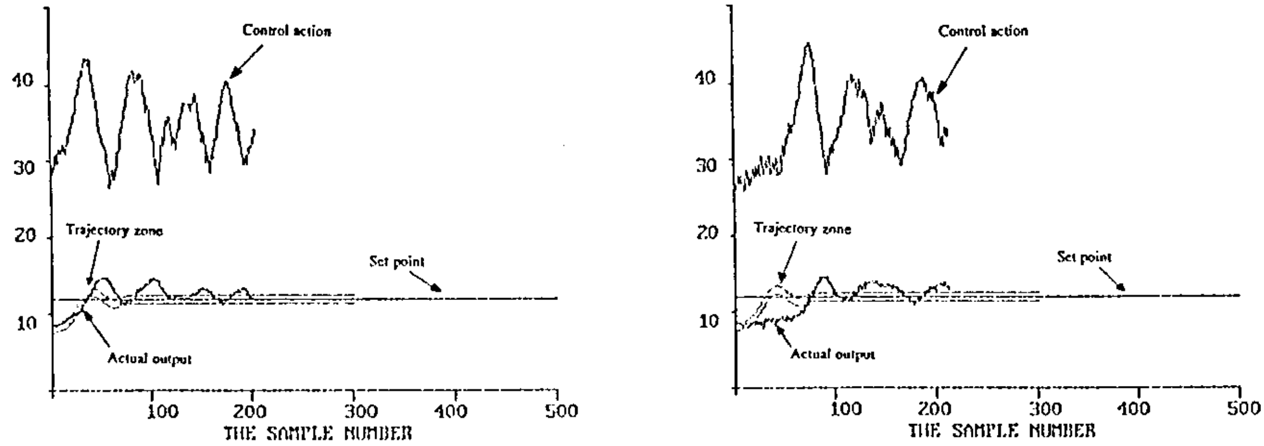
Error Action Change in error	PH	PM	PL	NL	NM	NH
PH	PH	PH	PH	PH	NH	PL
PM	PL	PH	PH	NM	NH	PL
PL	PH	PH	PM	NM	NH	NM
NL	PH	PM	PM	NL	NH	NM
NM	PL	PH	NL	NH	NH	NH
NH	NH	NM	NH	NH	NH	PL

b- Final version of rules

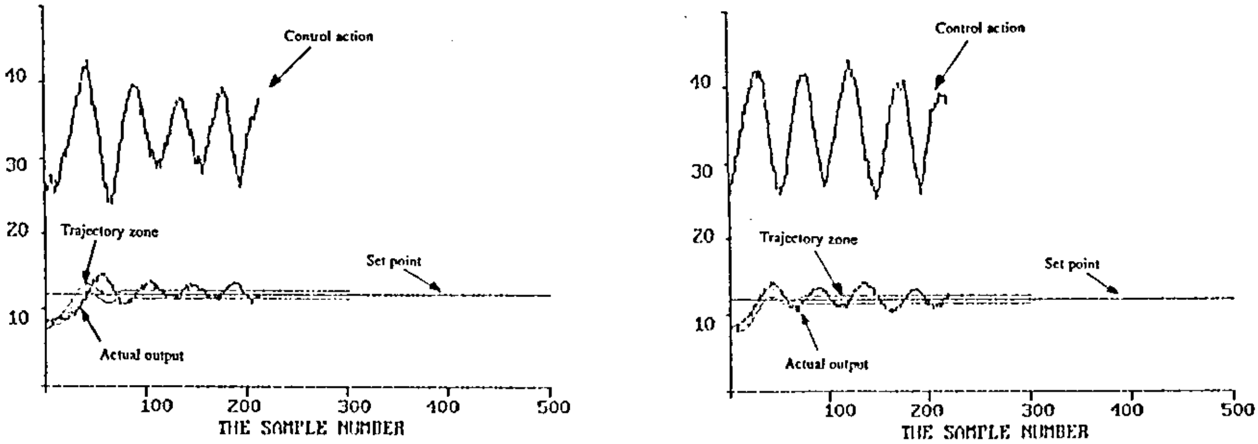
Fig. 4.8 The initial and final version of the rules produced practically.



a- Run No. 1 **b- Run No. 2**

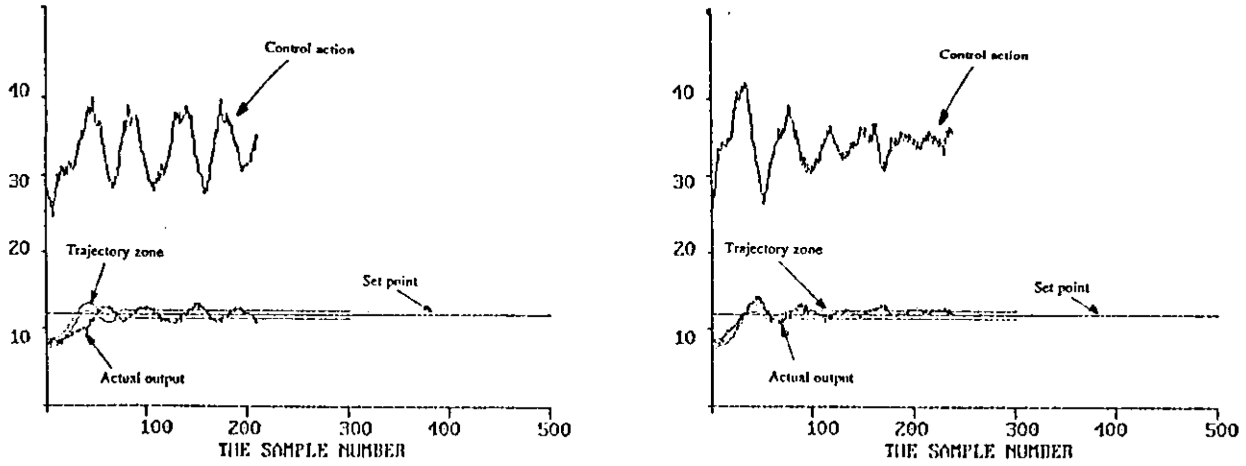


c- Run No. 3 **d- Run No. 4**

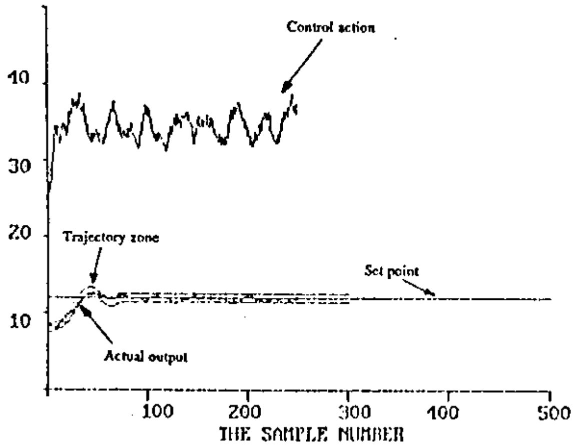


e- Run No. 5 **f- Run No. 6**

Fig. 4.9 process response for each run (practical)



g- Run No. 7 **h- Run No. 8**



i- Run No. 9

Fig. 4.9 continued

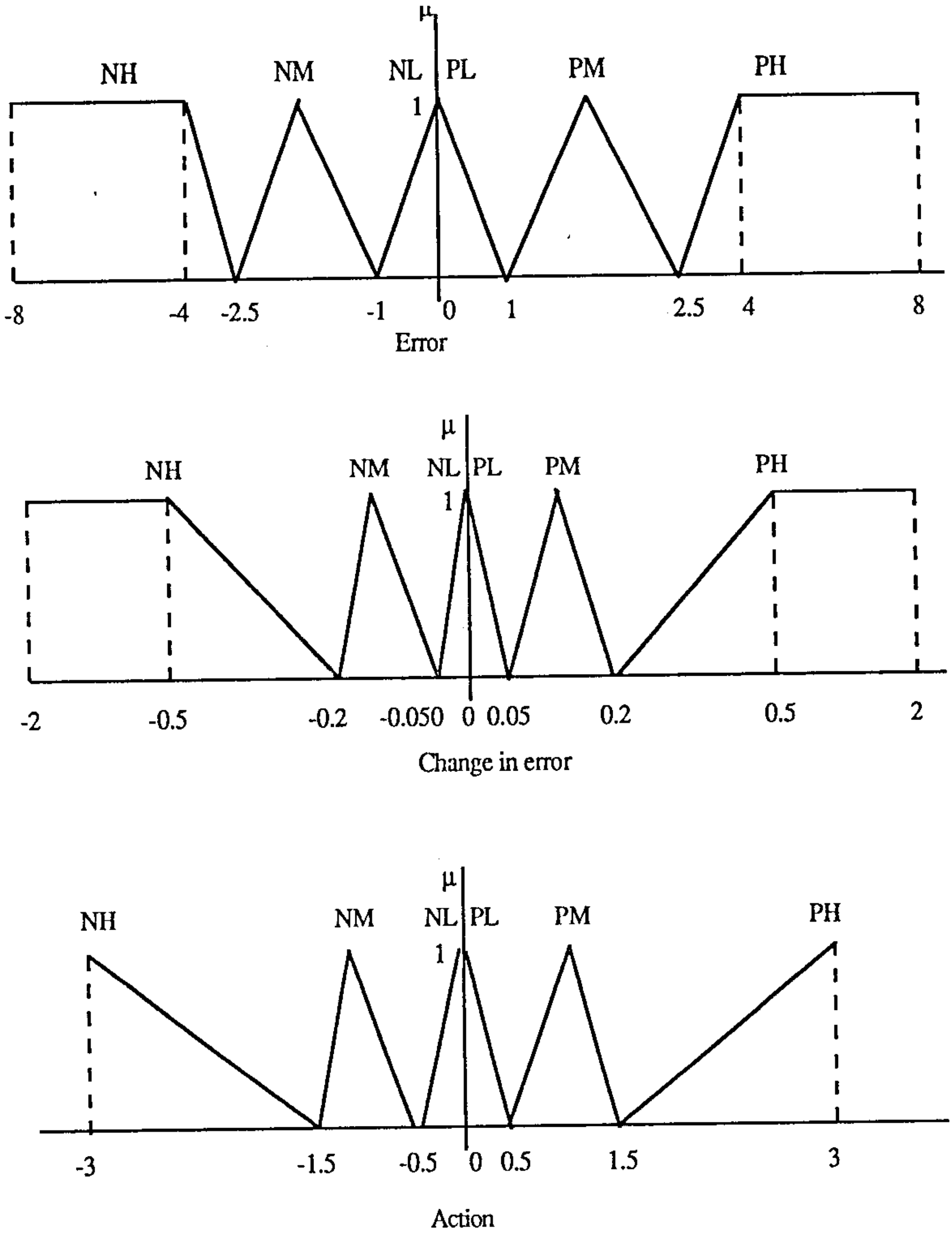


Fig. 4.10 Fuzzy subsets as defined by the operator

(Note : the x-axis is not to scale)

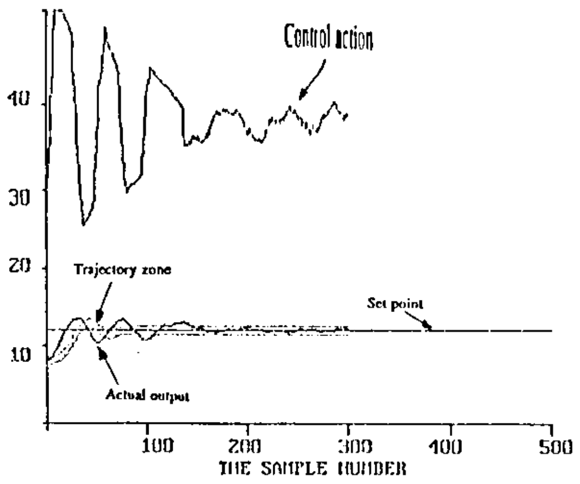
Error Action Change in error	PH	PM	PL	NL	NM	NH
PH	PH	PH	PL	PL	NM	NH
PM	PH	PH	PL	PL	NM	NH
PL	PH	PH	PL	NL	NM	NH
NL	PH	PM	PL	NL	NH	NH
NM	PH	PM	NL	NL	NH	NH
NH	PH	PM	NL	NL	NH	NH

a- Initial rules as defined by the operator.

Error Action Change in error	PH	PM	PL	NL	NM	NH
PH	PH	PH	PH	PL	NH	NH
PM	PM	PH	PH	PM	NM	NH
PL	PH	PH	NL	NL	NH	NH
NL	PL	PH	PM	NL	NH	NH
NM	PM	PL	PL	NH	NH	NH
NH	NL	NL	NM	NH	NH	NH

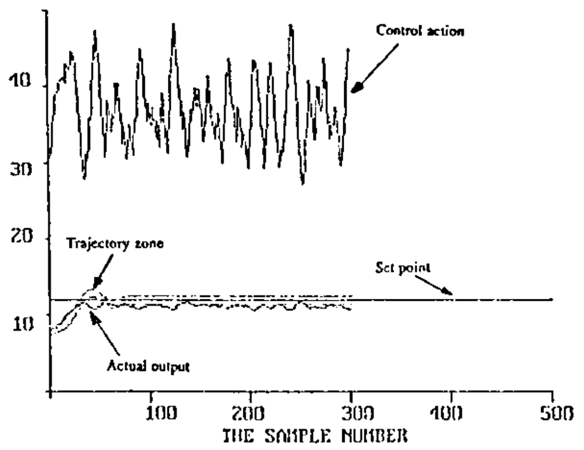
b- Final version of rules

Fig. 4.11 The operator initial rules and final rules



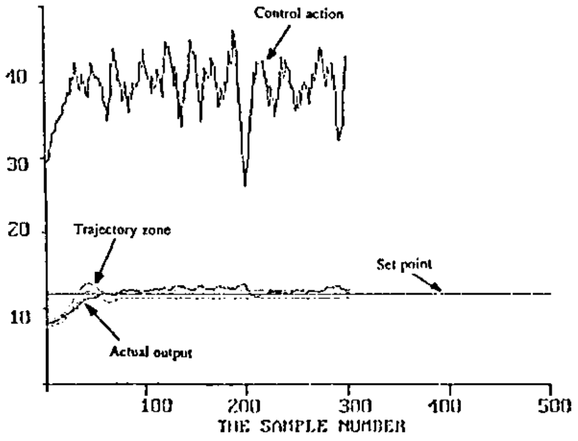
a- Run No. 1

b- Run No. 2



c- Run No. 3

d- Run No. 4



e- Run No. 5

f- Run No. 6

Fig. 4.12 process response for each run

Simulation and practical results presented previously show the importance of having an adaptive algorithm that gives a satisfactory linguistic rules.

Chapter 5

Conclusions

The complex nature of some processes make them difficult to control them adequately using most of classical and even modern controllers which need mathematical modelling of the system.

One method to control such processes is to mimic the operator behavior. By using the fuzzy theory, the behavior of the operator can be implemented in a computer to be used later to control the process.

In this study a software package was written, this software package implements the basic fuzzy logic rules, exactly as how the operator would process his knowledge and applying it.

The FLC and PID controller were applied to the coupled tank system through simulation and also practically. The two controllers were tested to analyze their performance in controlling the coupled tank with various practical problems such as the presence of disturbances, variation of system order and changing the operating region.

The two controllers gave a satisfactory performance, the PID controller is faster and more active than the FLC and the FLC was found to be comparable with the PID controller. It is should be noted here that the PID controller can be made slow and less active by changing its parameters. However by modifying the fuzzy logic rules the FLC can be made fast and more active.

To improve initial fuzzy rules and automatically modifying them to suit new process conditions an adaptive technique was developed which automatically modifies the fuzzy logic rules which give response not close to the desired one. This technique was applied to the coupled tank system and it was deduced it is useful and applicable. Even more the rules obtained from an experienced operator were improved by this technique.

An extention of this work could be in the following areas :

1. Apply FLC to more complex and non-linear systems.
2. Modify the membership functions automatically.
3. Switching technique between PID controller and FLC.

REFERENCES

1. Y. Al-Assaf, "Self-Tuning Control : Theory and Applications", Ph.D. thesis, Oxford University, 1988.

2. A. Kaufmann, "Introduction to The Theory of Fuzzy Subsets", 1st. ED. Academic Press, New York, 1975.

3. A. Jones, A. Kaufmann, and H.J. Zimmermann, "Fuzzy Sets Theory and Applications", 1st ED., D. Reidel Publishing Company, Dordrecht, Holland, 1986.

4. R.R. Yager, "A logical on-line Bibiographic Searcher : An Application of Fuzzy Sets", IEEE Trans. Syst. Man, Cybern., Vol. SMC-10, No. 1, 1980, pp. 51-53.

5. S.R. Watson, J.J. Weiss, and M.L. Donnel, "Fuzzy Decision Analysis", IEEE Trans. Syst., Man., Cybern., Vol. SMC-9, No.1, 1979, pp. 1-9.

6. G.S. Virk, Digital Computer Control Systems, 1st Ed., Macmillan, Hong Kong, 1991, pp. 136-175.

7. S.E. Sheridan and P. Skjoth, "Automatic Kiln Control at Oregon Pertland Cement Cempany's Durkee Plant Utilizing Fuzzy Logic", IEEE Trans. on Industry Applications, Vol. IA-20, No. 3, 1984, pp. 562-568.

8. R. Jain, "Decision Making in The Presence of Fuzzy Variables", IEEE Trans. Syst., Man, Cybern., October, 1976, pp. 698-702.

9. J. Efstathiou and V. Rajkovic, "Multiattribute Decision Making Using a Fuzzy Heuristic Approach", IEEE Trans. Syst., Man. Cybern., Vol.

SMC-9, No. 6, 1979, pp. 326-333.

10. R.M. Tong and P.P. Bonissune, "A linguistic Approach to Decision Making with Fuzzy Sets", IEEE Trans. Syst., Man, Cybern., Vol. SMC-10, No. 11, 1980, pp. 716-723.

11. E. Economakos, "Application of Fuzzy Concept To Power Demand Forcasting", IEEE Trans. Syst., Man, cybern., Vol. SMC-9, No. 10, 1979, pp. 651-657.

12. M.M. Etschmaier, "Fuzzy controls for Maintenance Scheduling in Transportation Systems", Automatica, Vol. 16, 1980, pp. 255-263.

13. K. Nakamura and S. Iwai, "Topological Fuzzy Sets as a Quantitative Description of Analogical Inference and It's Application to Question-Answering Systems for Information Retrieval", IEEE Trans. Syst., Man, Cybern., Vol. SMC-12, No. 2, 1982, pp. 193-203.

14. Y.Y. change and B. Mcinnis, "An Algorithm for Multiple Attribute, Multiple Alternative Decision problems Based on Fuzzy Sets with Application to Medical Diagnosis", IEEE Trans. Syst., Man, Cybern., Vol. SMC-10, No. 10, 1980, pp 645-650.

15. T.J. Schwartz, "Fuzzy Tools for Expert Systems", AI Expert, February, 1991, pp. 34-41.

16. E.H. Mamdani, "Application of Fuzzy Algorithms for Control of Simple Dynamic Plant", Proc. IEE, Vol. 212, 1974, pp. 1585-1588.

17. E.H. Mamdani and S. Assilian,"A Fuzzy Logic Controller for a Dynamic Plant", Int. J. Man-Machine Stud., Vol. 7, 1975, pp. 1-13.

18. M.J.M. Kickert and H.R.V.N. Lemke, "Application of a Fuzzy controller in a Warm Water Plant", Automatica, Vol. 12, 1976, pp. 301-308.

19. P.J. King and E.H. Mamdani, "The Application of Fuzzy Control Systems to Industrial Processes", Automatica, Vol. 13, 1977, pp. 235-242.

20. D. Rutherford and G.A. Carter, "A heuristic Adaptive Controller for a Sinter Plant", Proc. 2nd IFAC Symp. Automation in Mining, Mineral and Metal Processing, Johannesburg, 1976.

21. J.J. Ostergaard, "Fuzzy logic Control of a Heat Exchanger Process", Internal report 7601, Tech. University Denmark, Electric Power Eng. Dept., 1976.

22. R.M. Tong, "Some Problems with the Design and Implementation of Fuzzy Controller", Internal report CUED/F-CAMS/TR127, Cambridge University Engineering Department, 1976.

23. R.M. Tong and A. Latten, "Fuzzy Control of the Activated Sludge Wastewater Treatment Process", Automatica, Vol. 16, 1980, pp 659-701.

24. L.A. Zadeh, "Making Computers Think Like People", IEEE Spectrum, August, 1984, pp. 26-32.

25. J.V.D. Vegte, "Feedback Control Systems", 1st ED., Prentice-Hall, New Jersey, 1986, pp. 232-278.

26. P.E. Wellstead, "Coupled Tanks Apparatus", Group of Companies, England, 1981.

27. G.C. Goodwin and K.S. Sin, "Adaptive Filtering, Prediction and Control", 1st ED., Prentice-Hall, 1984.

Appendix A

```
'*******************************************************
'*                                                     *
'*  THIS PROGRAM IMPLEMENTS THE FUZZY LOGIC CONTROLLER *
'*  IN COMPUTER  USED  IN  CONTROL  OF  THE SOFTWARE   *
'*  SIMULATION OF THE COUPLED TANK SYSTEM .            *
'*                                                     *
'*******************************************************

'-------------------------------------------------------
'                 N O M E N C L A T U R E
'-------------------------------------------------------

'A(1)               : VALUE OF THE ERROR
'A(2)               : VALUE OF THE CHANGE IN ERROR
'A$()               : VARIABLE  NAME
'A1$(),A2$()        : CONDITION VARIABLE NAME
'ACT()              : MEMBERSHIP VALUE FOR THE RULE
'                     AT A CERTAIN DIVISION
'B$()               : FUZZY SUBSET NAME
'B1$(),B2$()        : CONDITION FUZZY SUBSET NAME
'C1$()              : ACTION VARIABLE NAME
'D1$()              : ACTION FUZZY SUBSET NAME
'FLOW(1),FLOW(2)    : THE FLOW RATE AT THE CURRENT SAMPLE
'                     AND THE PREVIOUS SAMPLE,RESPECTIVELY
'FACT()             : MEMBERSHIP VALUE FOR THE UNION OF THE
'                     RULES  AT A CERTAIN  DIVISION
'H1(1),H1(2)        : THE WATER LEVEL IN THE FIRST TANK AT
'                     THE CURRENT AND THE  PREVIOUS SAMPLE,
'                     RESPECTIVELY
'H2(1),H2(2),H2(3): THE WATER LEVEL IN THE SECOND TANK AT THE
'                     CURRENT AND THE PREVIOUS TWO SAMPLES,
'                     RESPECTIVELY
'LIMIT()            : END LIMIT FOR THE CONSTANT PART OF TYPE ONE
'                     OF THE FUZZY SUBSET
'LLR()              : LOWER LIMIT FOR THE RAMP PART (TYPE ONE) OR
'                     THE LOWER LIMIT (TYPE TWO)
'LR()               : MINIMUM VALUE FOR THE ACTION VARIABLE
'M()                : NUMBER OF THE FUZZY SUBSETS FOR A CERTAIN
'                     VARIABLE
'MMB()              : THE MEMBERSHIP FUNCTION FOR THE CONDITION
'                     VARIABLE AT A CERTAIN FUZZY SUBSET
'N                  : NUMBER OF VARIABLES
'NRUL               : NUMBER OF RULES
'P1()               : VALUE OF THE ACTION VARIABLE AT A CERTAIN
'                     DIVISION
'P2()               : MEMBERSHIP VALUE FOR THE ACTION VARIABLE
'                     AT A CERTAIN DIVISION
'QOUT               : FLOW RATE OF WATER OUT OF THE SECOND TANK
'RESP()             : RESPONSE (CHANGE IN THE CONTROL SIGNAL)
'S()                : TYPE OF THE RULE
'SLOPE$()           : SLOPE OF THE RAMP PART OF TYPE ONE OF THE
'                     FUZZY SUBSET
'SS()               : TYPE OF THE FUZZY SUBSET
'ST.................: SAMPLE TIME
'T                  : DESIRED WATER LEVEL IN THE SECOND TANK
'TH1                : CALCULATED PROPOSED VALUE FOR THE WATER
```

```
'                    LEVEL IN THE FIRST TANK
'TH2                : PROPOSED VALUE FOR THE WATER LEVEL IN THE
'                     SECOND TANK
'ULR()              : UPPER LIMIT FOR THE RAMP PART(TYPE ONE) OR
'                     THE UPPER LIMIT(TYPE TWO)
'UR                 : MAXIMUM VALUE FOR THE ACTION VARIABLE
'
'------------------------------------------------------------------
'          END OF  NOMENCLATURE
'------------------------------------------------------------------

'==================================================================
'          DEFINNING  THE MATRICIES
'==================================================================

DIM M(10), A$(5), B$(5, 10), SS(5, 10), LLR(5, 10), A(5)
DIM ULR(5, 10), P1(5, 100), P2(5, 10, 100), S(40), A1$(40)
DIM B1$(40), C1$(40), D1$(40), RELAT$(40), A2$(40), B2$(40)
DIM MMB(5, 10), FACT(5, 100), YY(40), YY1(40), YY2(40), UR(5), LR(5)
DIM LIMIT(5, 10), SUM2(5), RESP(5), SLOPE$(5, 10), X(10), SUM1(5)
DIM H2(3), H1(2), FLOW(2), G(2), I(2), DIFF(2), ACT(100)
'==================================================================
'          END OF DEFINNING THE MATRICIES
'==================================================================

'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
'                        INPUT OF DATA
'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
CLS
SCREEN 8
COLOR 4
COLOR , 14
LOCATE 6, 10
PRINT "FUZZY  LOGIC  PROGRAM"
LOCATE 18, 5
100 INPUT "DO YOU WANT TO LOAD DATA FROM FILE(YES OR NO) ?  ", LD$
COLOR 2
IF (LD$ <> "YES") AND (LD$ <> "yes") AND (LD$ <> "NO")
AND (LD$ <> "no") THEN
CLS
GOTO 100
END IF
IF (LD$ = "NO") OR (LD$ = "no") THEN
200 INPUT "  NO. OF VARIABLES = ", N
IF N = X THEN
GOTO 200
END IF
FOR I = 1 TO N
PRINT " FOR THE VARIABLE NO.", I
PRINT "                        "
PRINT " *********************************************"
PRINT "                        "
PRINT " THE VARIABLES ARE CLASSIFIED INTO CONDITION"
PRINT " AND ACTION VARIABLES ACCORDING TO THE GENERAL"
PRINT " FORM OF THE RULE :"
PRINT "                        "
PRINT " IF < CONDITION >  THEN  < ACTION >"
```

```
PRINT "                          "
PRINT " ***********************************************"

300 PRINT "SELECT ( 0 ) FOR CONDITION  VARIABLE "
PRINT "SELECT ( 1 ) FOR ACTION  VARIABLE "
INPUT "     ", D(I)
IF (D(I) <> 1) AND (D(I) <> 0) THEN
CLS
GOTO 300
END IF

400 INPUT "  NO. OF THE FUZZY SUBSETS FOR THE VARIABLE  = ", M(I)
IF M(I) = X THEN
GOTO 400
END IF

INPUT "  VARIABLE NAME IS  ", A$(I)
IF D(I) = 1 THEN
INPUT " MAXIMUM VALUE FOR THE ACTION VARIABLE =", UR(I)
INPUT " MINIMUM VALUE FOR THE ACTION VARIABLE =", LR(I)
END IF

FOR J = 1 TO M(I)
INPUT "  FUZZY SUBSET  NAME IS  ", B$(I, J)
500 CLS
PRINT " ----------------------------"
PRINT " SELECT THE FUZZY SUBSET TYPE"
PRINT "                             "
PRINT " THERE ARE TWO TYPES         "
PRINT "                             "
PRINT " 1- EXTREAME FUNCTION        "
PRINT " IF SMALL OR LARGE .......etc"
PRINT "                             "
PRINT " 2- MID. FUNCTION            "
PRINT " IF A BOUT, AROUND .......etc"
PRINT "                             "
PRINT "                             "
PRINT "                             "
PRINT " ----------------------------"
INPUT " TYPE OF THE FUZZY SUBSET  = ", SS(I, J)
IF (SS(I, J) <> 1) AND (SS(I, J) <> 2) THEN
GOTO 500
END IF
CLS
IF SS(I, J) = 1 THEN
550 PRINT " INSERT [P] FOR POSITIVE SLOPE"
PRINT "INSERT [N] FOR NEGATIVE SLOPE"
INPUT "[P/N]", SLOPE$(I, J)
IF (SLOPE$(I, J) <> "P") AND (SLOPE$(I, J) <> "p")
AND (SLOPE$(I, J) <> "N") AND (SLOPE$(I, J) <> "n") THEN
CLS
GOTO 550
END IF
INPUT " LOWER LIMIT RANGE FOR RAMP PART= ", LLR(I, J)
INPUT " UPPER LIMIT RANGE FOR RAMP PART= ", ULR(I, J)
INPUT " END LIMIT FOR THE CONSTANT PART =", LIMIT(I, J)
```

```
ELSE
IF SS(I, J) = 2 THEN
INPUT " LOWER LIMIT = ", LLR(I, J)
INPUT " UPPER LIMIT = ", ULR(I, J)
END IF
END IF
NEXT J
NEXT I
CLS

INPUT " NO. OF RULES ", NRUL
FOR I = 1 TO NRUL
600 PRINT "--------------------------"
PRINT "SELECT TYPE OF RULES :     "
PRINT "                           "
PRINT "1- IF A THEN C             "
PRINT "                           "
PRINT "2- IF A RELATION B THEN C  "
PRINT "                           "
PRINT "--------------------------"
INPUT " SELECT TYPE OF RULE IS ", S(I)
IF (S(I) <> 1) AND (S(I) <> 2) THEN
CLS
GOTO 600
END IF
IF S(I) = 1 THEN
INPUT "IF ", A1$(I)
INPUT "      IS", B1$(I)
INPUT "               THEN", C1$(I)
INPUT "                                 IS", D1$(I)
ELSE
IF S(I) = 2 THEN
INPUT "IF ", A1$(I)
INPUT "IS", B1$(I)
INPUT "RELATE (OR , AND)", RELAT$(I)
INPUT "    ", A2$(I)
INPUT "IS", B2$(I)
INPUT "THEN", C1$(I)
INPUT "IS", D1$(I)
END IF
END IF
CLS
NEXT I
CLS
CLS
700 INPUT " DO YOU WANT TO SAVE DATA (YES OR NO)", SD$
IF (SD$ <> "YES") AND (SD$ <> "yes") AND (SD$ <> "NO")
AND (SD$ <> "no") THEN
CLS
GOTO 700
END IF
IF SD$ = "YES" THEN
INPUT " FILE NAME ", FLN$
OPEN "O", 1, FLN$
PRINT #1, N
FOR I = 1 TO N
```

```
PRINT #1, D(I)
PRINT #1, M(I)
PRINT #1, A$(I)
IF D(I) = 1 THEN
PRINT #1, UR(I)
PRINT #1, LR(I)
END IF

FOR J = 1 TO M(I)
PRINT #1, B$(I, J)
PRINT #1, SS(I, J)
IF SS(I, J) = 1 THEN
PRINT #1, SLOPE$(I, J)
PRINT #1, LLR(I, J)
PRINT #1, ULR(I, J)
PRINT #1, LIMIT(I, J)
ELSE
IF SS(I, J) = 2 THEN
PRINT #1, LLR(I, J)
PRINT #1, ULR(I, J)
END IF
END IF
NEXT J
NEXT I
PRINT #1, NRUL
FOR I = 1 TO NRUL
PRINT #1, S(I)
IF S(I) = 1 THEN
PRINT #1, A1$(I)
PRINT #1, B1$(I)
PRINT #1, C1$(I)
PRINT #1, D1$(I)
ELSE
IF S(I) = 2 THEN
PRINT #1, A1$(I)
PRINT #1, B1$(I)
PRINT #1, RELAT$(I)
PRINT #1, A2$(I)
PRINT #1, B2$(I)
PRINT #1, C1$(I)
PRINT #1, D1$(I)
END IF
END IF
NEXT I
CLOSE 1
END IF

ELSE
INPUT "    FILE NAME "; FLN$
OPEN "I", 1, FLN$
INPUT #1, N
FOR I = 1 TO N
INPUT #1, D(I)
INPUT #1, M(I)
INPUT #1, A$(I)
```

```
IF D(I) = 1 THEN
INPUT #1, UR(I)
INPUT #1, LR(I)
END IF
FOR J = 1 TO M(I)
INPUT #1, B$(I, J)
INPUT #1, SS(I, J)

IF SS(I, J) = 1 THEN
INPUT #1, SLOPE$(I, J)
INPUT #1, LLR(I, J)
INPUT #1, ULR(I, J)
INPUT #1, LIMIT(I, J)
ELSE
IF SS(I, J) = 2 THEN
INPUT #1, LLR(I, J)
INPUT #1, ULR(I, J)
END IF
END IF
NEXT J
NEXT I
INPUT #1, NRUL
FOR I = 1 TO NRUL
INPUT #1, S(I)
IF S(I) = 1 THEN
INPUT #1, A1$(I)
INPUT #1, B1$(I)
INPUT #1, C1$(I)
INPUT #1, D1$(I)
ELSE
IF S(I) = 2 THEN
INPUT #1, A1$(I)
INPUT #1, B1$(I)
INPUT #1, RELAT$(I)
INPUT #1, A2$(I)
INPUT #1, B2$(I)
INPUT #1, C1$(I)
INPUT #1, D1$(I)
END IF
END IF
NEXT I
CLOSE 1
END IF

'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
'                       END OF INPUT OF DATA
'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
'################################################################
'        PARTITIONING  THE ACTION  VARIABLE INTO 100 DIVISION
'        AND CALCULATE THE MEMBERSHIP AT EACH DIVISION
'################################################################

CLS
FOR I = 1 TO N
IF D(I) = 1 THEN
P = (UR(I) - LR(I)) / 99
FOR J = 1 TO M(I)
ICOUNT = 0
C = .0001

FOR R = LR(I) TO (UR(I) + C) STEP P
ICOUNT = ICOUNT + 1
P1(I, ICOUNT) = R

IF SS(I, J) = 1 THEN
IF (R <= (ULR(I, J) + C)) AND (R >= LLR(I, J)) THEN
IF (SLOPE$(I, J) = "P") OR (SLOPE$(I, J) = "p") THEN
COEF1 = LLR(I, J) / (LLR(I, J) - ULR(I, J))
COEF2 = 1 / (ULR(I, J) - LLR(I, J))
ELSE
COEF1 = ULR(I, J) / (ULR(I, J) - LLR(I, J))
COEF2 = 1 / (LLR(I, J) - ULR(I, J))
END IF
P2(I, J, ICOUNT) = COEF1 + COEF2 * R
ELSE
IF (R < LLR(I, J)) AND (R >= LIMIT(I, J)) AND (COEF2 < 0) THEN
P2(I, J, ICOUNT) = 1
ELSE
IF (R > (ULR(I, J) + C)) AND (R <= LIMIT(I, J)) AND (COEF2 > 0) THEN
P2(I, J, ICOUNT) = 1
ELSE
P2(I, J, ICOUNT) = 0
END IF
END IF
END IF
ELSE
MID = (ULR(I, J) + LLR(I, J)) / 2
IF (R <= (ULR(I, J) + C)) AND (R >= LLR(I, J)) THEN
IF (R >= LLR(I, J)) AND (R <= MID) THEN

COEF1 = LLR(I, J) / (LLR(I, J) - MID)
COEF2 = 1 / (MID - LLR(I, J))
P2(I, J, ICOUNT) = COEF1 + COEF2 * R
ELSE
COEF2 = 1 / (MID - ULR(I, J))
COEF1 = ULR(I, J) / (ULR(I, J) - MID)
P2(I, J, ICOUNT) = COEF1 + COEF2 * R
END IF
ELSE
P2(I, J, ICOUNT) = 0
END IF
END IF
```

```
NEXT R
NEXT J
END IF
NEXT I

'##################################################################
'        END  OF  PARTITIONING  THE ACTION VARIABLE
'##################################################################


'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
'                  INITIALIZATION
'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


H2(3) = 9
H2(2) = 9
H1(2) = 13.85
FLOW(2) = 31.2
SCREEN 12
WINDOW (-20, -10)-(1000, 50)
COLOR 0
PSET (0, 0)
LINE (0, 0)-(1000, 0), 2
LINE (0, -1)-(0, 50), 2
LINE (0, 12)-(1000, 12), 4
FOR I = -5 TO 5
LINE (0, 10 * I)-(-2, 10 * I), 1
NEXT I
FOR I = 1 TO 300
LINE (20 * I, 0)-(20 * I, -.05), 1
NEXT I
COLOR 2
S = 0
FOR W = 0 TO 300 STEP 40
IF W <> 0 THEN
S = S + 10
ELSE
S = 5
END IF
LOCATE 26, S
PRINT W
NEXT W
LOCATE 1, 2: PRINT 50
LOCATE 16, 2: PRINT 20
LOCATE 28, 26: PRINT "THE SAMPLE NUMBER"
LOCATE 10, 26: PRINT "SET  POINT"
OPEN "O", 2, "OUTPUT.OUT"
FOR IJ = 1 TO 1000
T = 12
A(1) = T - H2(2): 'ERROR CALCULATION
A(2) = H2(3) - H2(2): 'CHANGE IN ERROR CALCULATION
```

```
'$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
' DEFINNING  THE MEMBERSHIP FUNCTIONS FOR THE CONDITION VARIABLES
'$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

FOR I = 1 TO N
IF D(I) = 0 THEN
FOR J = 1 TO M(I)
IF SS(I, J) = 1 THEN
IF (SLOPE$(I, J) = "P") OR (SLOPE$(I, J) = "p") THEN
COEF1 = LLR(I, J) / (LLR(I, J) - ULR(I, J))
COEF2 = 1 / (ULR(I, J) - LLR(I, J))
ELSE
COEF1 = ULR(I, J) / (ULR(I, J) - LLR(I, J))
COEF2 = 1 / (LLR(I, J) - ULR(I, J))
END IF
IF (A(I) <= ULR(I, J)) AND (A(I) >= LLR(I, J)) THEN
MMB(I, J) = COEF1 + COEF2 * A(I)
ELSE
IF (A(I) < LLR(I, J)) AND (A(I) >= LIMIT(I, J)) AND (COEF2 < 0) THEN
MMB(I, J) = 1
ELSE
IF (A(I) > ULR(I, J)) AND (A(I) <= LIMIT(I, J)) AND (COEF2 > 0) THEN
MMB(I, J) = 1
ELSE
MMB(I, J) = 0
END IF
END IF
END IF
ELSE
MID = (ULR(I, J) + LLR(I, J)) / 2
IF (A(I) > ULR(I, J)) OR (A(I) < LLR(I, J)) THEN
MMB(I, J) = 0
ELSE
IF (A(I) >= LLR(I, J)) AND (A(I) <= MID) THEN
COEF1 = LLR(I, J) / (LLR(I, J) - MID)
COEF2 = 1 / (MID - LLR(I, J))
MMB(I, J) = COEF1 + COEF2 * A(I)
ELSE
COEF2 = 1 / (MID - ULR(I, J))
COEF1 = ULR(I, J) / (ULR(I, J) - MID)
MMB(I, J) = COEF1 + COEF2 * A(I)
END IF
END IF
END IF
NEXT J
END IF
NEXT I

'$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
'          END OF DEFINNING  THE MEMBERSHIP FUNCTIONS
'$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
'*****  INITIALIZE THE MEMBERSHIP FOR THE RULES *****
'*****  AND FOR THE UNION OF THE RULES          *****
FOR I2 = 1 TO N
IF D(I2) = 1 THEN
FOR K = 1 TO 100
FACT(I2, K) = 0
ACT(K) = 0
NEXT K
END IF
NEXT I2

'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
'   CALCULATION OF THE MEMBERSHIP  FOR ALL THE RULES
'   AND FOR THE UNION OF THE RULES
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&


FOR I1 = 1 TO NRUL
IF S(I1) = 1 THEN
FOR I2 = 1 TO N
IF D(I2) = 0 THEN
FOR J = 1 TO M(I2)
IF (A1$(I1) = A$(I2)) AND (B1$(I1) = B$(I2, J)) THEN
YY(I1) = MMB(I2, J)
IF YY(I1) = 0 THEN
GOTO 1000
END IF
END IF
NEXT J
END IF
NEXT I2


FOR I2 = 1 TO N
IF D(I2) = 1 THEN
FOR J = 1 TO M(I2)
IF (C1$(I1) = A$(I2)) AND (D1$(I1) = B$(I2, J)) THEN
FOR K = 1 TO 100
IF P2(I2, J, K) <> 0 THEN
IF YY(I1) < P2(I2, J, K) THEN
Z = YY(I1)
ELSE
Z = P2(I2, J, K)
END IF
ELSE
Z = 0
END IF
ACT(K) = Z
IF ACT(K) >= FACT(I2, K) THEN
FACT(I2, K) = ACT(K)
ELSE
FACT(I2, K) = FACT(I2, K)
```

```
END IF
NEXT K
END IF
NEXT J
END IF
NEXT I2
ELSE
IF S(I1) = 2 THEN
FOR I2 = 1 TO N
IF D(I2) = 0 THEN
FOR J = 1 TO M(I2)
IF (A1$(I1) = A$(I2)) AND (B1$(I1) = B$(I2, J)) THEN
YY1(I1) = MMB(I2, J)
ELSE
IF (A2$(I1) = A$(I2)) AND (B2$(I1) = B$(I2, J)) THEN
YY2(I1) = MMB(I2, J)
END IF
END IF
NEXT J
END IF
NEXT I2


FOR I2 = 1 TO N
IF D(I2) = 1 THEN
FOR J = 1 TO M(I2)
IF (C1$(I1) = A$(I2)) AND (D1$(I1) = B$(I2, J)) THEN
IF (RELAT$(I1) = "AND") OR (RELAT$(I1) = "and") THEN
IF (YY1(I1) = 0) OR (YY2(I1) = 0) THEN
GOTO 1000
END IF
FOR K = 1 TO 100
IF P2(I2, J, K) <> 0 THEN
IF (YY1(I1) < YY2(I1)) AND (YY1(I1) <= P2(I2, J, K)) THEN
Z = YY1(I1)
ELSE
IF (YY2(I1) <= YY1(I1)) AND (YY2(I1) <= P2(I2, J, K)) THEN
Z = YY2(I1)
ELSE
IF (P2(I2, J, K) < YY1(I1)) AND (P2(I2, J, K) < YY2(I1)) THEN
Z = P2(I2, J, K)
END IF
END IF
END IF
ELSE
Z = 0
END IF
ACT(K) = Z
IF ACT(K) >= FACT(I2, K) THEN
FACT(I2, K) = ACT(K)
```

```
ELSE
FACT(I2, K) = FACT(I2, K)
END IF
NEXT K
ELSE
IF (RELAT$(I1) = "OR") OR (RELAT$(I1) = "or") THEN
IF (YY1(I1) = 0) AND (YY2(I1) = 0) THEN
GOTO 1000
END IF
FOR K = 1 TO 100
IF P2(I2, J, K) <> 0 THEN
IF (YY1(I1) >= YY2(I1)) AND (YY1(I1) <= P2(I2, J, K)) THEN
Z = YY1(I1)
ELSE
IF (YY2(I1) > YY1(I1)) AND (YY2(I1) <= P2(I2, J, K)) THEN
Z = YY2(I1)
ELSE
Z = P2(I2, J, K)
END IF
END IF
ELSE
Z = 0
END IF
ACT(K) = Z
IF ACT(K) >= FACT(I2, K) THEN
FACT(I2, K) = ACT(K)
ELSE
FACT(I2, K) = FACT(I2, K)
END IF
NEXT K
END IF
END IF
END IF
NEXT J
END IF
NEXT I2
END IF
END IF
1000 NEXT I1

'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
' END OF CALCULATION OF THE MEMBERSHIP FOR THE RULES AND
' FOR THE UNION OF THE RULES
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&


'^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
' CALCULATION OF THE FUZZY LOGIC CONTROLLER RESPONSE
' USING CENTRE OF AREA METHOD
'
'^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

FOR I2 = 1 TO N
SUM1(I2) = 0
SUM2(I2) = 0
NEXT I2
```

```
FOR I2 = 1 TO N
IF D(I2) = 1 THEN
FOR K = 1 TO 100
SUM1(I2) = SUM1(I2) + FACT(I2, K)
SUM2(I2) = SUM2(I2) + (FACT(I2, K) * P1(I2, K))
NEXT K
IF SUM1(I2) <> 0 THEN
RESP(I2) = SUM2(I2) / SUM1(I2)
ELSE
RESP(I2) = 0
END IF
'PRINT " RESPONSE = ",RESP(I2)
END IF
NEXT I2

'^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
'        END OF CALCULATION OF THE RESPONSE
'
'^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

'----------------------------------------------------------------------
' ADJUSTMENT OF THE FLOW RATE TO BE WITHIN THE LIMITS(0 AND 50)
'----------------------------------------------------------------------


XXX = FLOW(2)
X(1) = RESP(3)
FLOW(1) = X(1) + FLOW(2)
IF FLOW(1) <= 0 THEN
FLOW(1) = 0
ELSE
IF FLOW(1) >= 50 THEN
FLOW(1) = 50
ELSE
END IF
END IF

'----------------------------------------------------------------------
'        END OF ADJUSTMENT OF THE FLOW RATE
'----------------------------------------------------------------------

FLOW(2) = FLOW(1)

NOISE = (RND - .5) / 3
RRR = H2(2)

'**************************************************
' **********   MODELLING   *********************
'**************************************************
'
ST = 5
DIFF(2) = 10
G(2) = 0
I(2) = 0
FOR TH2 = (H2(2) - 1) TO (H2(2) + 1) STEP .01
```

```
IF TH2 < 3 THEN
GOTO 1010
END IF
QOUT = .2886 * ((1962 * (TH2 - 3)) ^ .5)
TH1 = 4.9638 * .001 * ((100 * (TH2 - H2(2)) / ST) + QOUT) ^ 2 + TH2
DIFF(1) = FLOW(1) - QOUT - (100 / ST) * (TH1 - H1(2) + TH2 - H2(2))
G(1) = TH2
I(1) = TH1
IF DIFF(1) <= 0 THEN
DIFF(1) = -DIFF(1)
END IF
IF DIFF(1) <= DIFF(2) THEN
G(2) = G(1)
I(2) = I(1)
DIFF(2) = DIFF(1)
ELSE
G(2) = G(2)
I(2) = I(2)
DIFF(2) = DIFF(2)
END IF
1010 NEXT TH2
H2(1) = G(2)
H1(1) = I(2)
' *****************************************************************
'  ****************  END OF  MODELLING ***********************
' *****************************************************************

'-----------------------------------------------------------------
'        ADJUSTMENT OF THE WATER LEVELS TO BE HIGHER OR EQUAL
'        TO THE LOWER LIMIT (3CM)
'-----------------------------------------------------------------

IF H1(1) < 3 THEN
H1(1) = 3
END IF
IF H2(1) < 3 THEN
H2(1) = 3
END IF

'-----------------------------------------------------------------
'        END OF ADJUSTMENT OF THE WATER LEVELS
'-----------------------------------------------------------------
H2(3) = H2(2)
H2(2) = H2(1)
H1(2) = H1(1)

'.................................................................
' PLOTTING THE RESPONSE OF THE FUZZY LOGIC CONTROLLER
'.................................................................

PRINT #2, IJ, FLOW(1), H2(1), H1(1)
LINE (IJ - 1, RRR)-(IJ, H2(1)), 3
LINE (IJ - 1, XXX)-(IJ, FLOW(1)), 10
NEXT IJ
CLOSE 2
END
```

Appendix B

```
'**********************************************************
'*                                                        *
'*   THIS PROGRAM IMPLEMENTS THE PID CONTROLLER   IN      *
'*   COMPUTER USED IN CONTROL OF THE SOFTWARE SIMULATION  *
'*   OF THE COUPLED TANK SYSTEM                           *
'*                                                        *
'**********************************************************

'----------------------------------------------------------
'              N O M E N C L A T U R E
'----------------------------------------------------------

'A(1),A(2),A(3)     : ERROR AT THE CURRENT  SAMPLE  AND
'                     THE PREVIOUS  TWO SAMPLES,RESPECTIVELY
'FLOW(1),FLOW(2)    : FLOW RATE AT THE CURRENT SAMPLE AND
'                     THE PREVIOUS SAMPLE,RESPECTIVELY
'H1(1),H1(2)        : WATER LEVEL IN THE FIRST TANK AT
'                     THE CURRENT AND THE PREVIOUS SAMPLE,
'                     RESPECTIVELY
'H2(1),H2(2),H2(3)  : WATER LEVEL IN THE SECOND TANK  AT
'                     THE CURRENT SAMPLE AND THE PREVIOUS TWO,
'                     RESPECTIVELY
'KD                 : DERIVATIVE GAIN FACTOR
'KI                 : INTEGRAL GAIN FACTOR
'KP                 : PROPORTIONAL GAIN FACTOR
'QOUT               : FLOW RATE OF WATER OUT OF THE SECOND TANK
'ST                 : SAMPLE TIME
'T                  : DESIRED WATER LEVEL IN THE SECOND TANK
'TH1                : CALCULATED PROPOSED VALUE FOR THE WATER
'                     LEVEL IN THE FIRST TANK
'TH2                : PROPOSED VALUE FOR THE WATER LEVEL IN THE
'                     SECOND TANK
'U(1)               : THE RESPONSE (CHANGE IN THE CONTROL SIGNAL)

'----------------------------------------------------------
'              END OF NOMENCLATURE
'----------------------------------------------------------


'==========================================================
'        DEFINNING  THE MATRICIES
'==========================================================

DIM U(2), A(3), H2(3), H1(2)
DIM DIFF(2), FLOW(2), I(2), G(2)

'==========================================================
'        END OF DEFINNING  THE MATRICIES
'==========================================================
```

```
'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
'                        INPUT OF DATA
'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

INPUT " KP = ", KP
INPUT " KI = ", KI
INPUT " KD = ", KD
INPUT " THE DESIRED WATER LEVEL =", T
INPUT " THE SAMPLE TIME =", ST

'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
'        END OF INPUT OF DATA
'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
'                        INITIALIZATION
'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

A(3) = 0
A(2) = 0
H2(2) = 9
H1(2) = 13.85
FLOW(2) = 31

CLS
SCREEN 12
WINDOW (-20, -10)-(1000, 50)
COLOR 0
PSET (0, 0)
LINE (0, 0)-(1000, 0), 2
LINE (0, -1)-(0, 50), 2
LINE (0, T)-(1000, T), 4

FOR I = -5 TO 5
LINE (0, 10 * I)-(-4, 10 * I), 1
NEXT I

FOR I = 1 TO 1000
LINE (100 * I, 0)-(100 * I, -2), 1
NEXT I
COLOR 2
S = 0
FOR W = 0 TO 900 STEP 100
IF W <> 0 THEN
S = S + 8
ELSE
S = 1
END IF
LOCATE 26, S
PRINT W
NEXT W
LOCATE 1, 2: PRINT 50
LOCATE 16, 2: PRINT 20
LOCATE 28, 26: PRINT " THE SAMPLE NUMBER "
LOCATE 10, 26: PRINT " SET POINT "

'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
'        END OF INITIALIZATION
'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
'   CALCULATION OF THE PID RESPONSE
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&


OPEN "O", 2, "EXPID.OUT"
FOR IJ = 1 TO 1000
A(1) = T - H2(2)
APID = KP + KI / (2 * ST) + KD / ST
BPID = KI * ST / 2 - KP - (2 * KD / ST)
CPID = KD / ST

U(1) = APID * A(1) + BPID * A(2) + CPID * A(3)

FLOW(1) = FLOW(2) + U(1)

'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
'         END OF CALCULATION OF THE RESPONSE
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&


'-------------------------------------------------------------
'   ADJUSTMENT OF THE FLOW TO BE WITHIN THE LIMITS (0 AND 50)
'-------------------------------------------------------------

IF FLOW(1) <= 0 THEN
FLOW(1) = 0
END IF
IF FLOW(1) >= 50 THEN
FLOW(1) = 50
END IF
'-------------------------------------------------------------
'         END OF ADJUSTMENT OF THE FLOW
'-------------------------------------------------------------

XXX = FLOW(2)
RRR = H2(2)

'************************************************************
'***********     M O D E L L I N G     **********************
'************************************************************

DIFF(2) = 10
G(2) = 0
I(2) = 0

FOR TH2 = (H2(2) - 1) TO (H2(2) + 1) STEP .01
IF TH2 < 3 THEN
GOTO 1010
END IF

QOUT = .2886 * ((1962 * (TH2 - 3)) ^ .5)
TH1 = 4.9638 * .001 * ((100 * (TH2 - H2(2)) / ST) + QOUT) ^ 2 + TH2
DIFF(1) = FLOW(1) - QOUT - (100 / ST) * (TH1 - H1(2) + TH2 - H2(2))
G(1) = TH2
```

```
I(1) = TH1
IF DIFF(1) <= 0 THEN
DIFF(1) = -DIFF(1)
END IF

IF DIFF(1) <= DIFF(2) THEN
G(2) = G(1)
I(2) = I(1)
DIFF(2) = DIFF(1)
ELSE
G(2) = G(2)
I(2) = I(2)
DIFF(2) = DIFF(2)
END IF

1010 NEXT TH2

'************************************************************
'***********  END OF  MODELLING    ***********************
'************************************************************


A(3) = A(2)
A(2) = A(1)
H2(1) = G(2) + (RND - .5) / 2
H1(1) = I(2)


'-------------------------------------------------------------
'        ADJUSTMENT OF THE WATER LEVELS TO BE HIGHER OR EQUAL
'        TO THE LOWER LIMIT (3CM)
'-------------------------------------------------------------

IF H1(1) < 3 THEN
H1(1) = 3
END IF

IF H2(1) < 3 THEN
H2(1) = 3
END IF

'-------------------------------------------------------------
'        END OF ADJUSTMENT OF THE WATER LEVELS
'-------------------------------------------------------------

FLOW(2) = FLOW(1)
H2(2) = H2(1)
H1(2) = H1(1)

'.............................................................
'   PLOTTING  THE RESPONSE OF THE PID CONTROLLER
'   AND SAVING THE RESULTS ON FILE
'.............................................................

PRINT #2, IJ, FLOW(1), H1(1), H2(1)
LINE (IJ - 1, RRR)-(IJ, H2(1)), 3
LINE (IJ - 1, XXX)-(IJ, FLOW(1)), 10
NEXT IJ
CLOSE 2
END
```

الملخص

هذا البحث يدرس تطبيق المنطق المشوش للتحكم بالأنظمة الديناميكية المعقدة كطريقة بديلة للطرق التقليدية التي تتطلب نموذجاً رياضياً للنظام. تم تطبيق هذه الطريقة على نظام الخزان المزدوج وأعطت نتائج مماثلة لتلك الناتجة عن الطرق التقليدية .

إن من أسس الطريقة هي محاكاة ما يقوم به العامل للتحكم بالنظام وبما أن القواعد التي يعطيها لا تكون دقيقة فقد تم تطوير إسلوب لتعديل تلك القواعد لإعطاء تحكم أفضل بالأنظمة تلقائياً وبشكل مستمر . النتائج النظرية والعملية لذلك الاسلوب كانت جيدة .