Fuzzy Control and its Application to a pH Process

K. E. Huang

A Thesis Submitted to the University of Warwick for the degree of Doctor of Philosophy

Department of Engineering
University of Warwick

July 1997

Summary

In the chemical industry, the control of pH is a well-known problem that presents difficulties due to the large variations in its process dynamics and the static nonlinearity between pH and concentration. pH control requires the application of advanced control techniques such as linear or nonlinear adaptive control methods. Unfortunately, adaptive controllers rely on a mathematical model of the process being controlled, the parameters being determined or modified in real time. Because of its characteristics, the pH control process is extremely difficult to model accurately.

Fuzzy logic, which is derived from Zadeh's theory of fuzzy sets and algorithms, provides an effective means of capturing the approximate, inexact nature of the physical world. It can be used to convert a linguistic control strategy based on expert knowledge, into an automatic control strategy to control a system in the absence of an exact mathematical model. The work described in this thesis sets out to investigate the suitability of fuzzy techniques for the control of pH within a continuous flow titration process.

Initially, a simple fuzzy development system was designed and used to produce an experimental fuzzy control program. A detailed study was then performed on the relationship between fuzzy decision table scaling factors and the control constants of a digital PI controller. Equation derived from this study were then confirmed experimentally using an analogue simulation of a first order plant. As a result of this work a novel method of tuning a fuzzy controller by adjusting its scaling factors, was derived. This technique was then used for the remainder of the work described in this thesis.

The findings of the simulation studies were confirmed by an extensive series of experiments using a pH process pilot plant. The performance of the tunable fuzzy controller was compared with that of a conventional PI controller in response to step change in the set-point, at a number of pH levels. The results showed not only that the fuzzy controller could be easily adjusted to provided a wide range of

operating characteristics, but also that the fuzzy controller was much better at controlling the highly non-linear pH process, than a conventional digital PI controller. The fuzzy controller achieved a shorter settling time, produced less over-shoot, and was less affected by contamination than the digital PI controller.

One of the most important characteristics of the tunable fuzzy controller is its ability to implement a wide variety of control mechanisms simply by modifying one or two control variables. Thus the controller can be made to behave in a manner similar to that of a conventional PI controller, or with different parameter values, can imitate other forms of controller. One such mode of operation uses sliding mode control, with the fuzzy decision table main diagonal being used as the variable structure system (VSS) switching line. A theoretical explanation of this behavior, and its boundary conditions, are given within the text.

While the work described within this thesis has concentrated on the use of fuzzy techniques in the control of continuous flow pH plants, the flexibility of the fuzzy control strategy described here, make it of interest in other areas. It is likely to be particularly useful in situations where high degrees of non-linearity make more conventional control methods ineffective.

Contents

Acknowledgement

There are many people to whom grateful thanks are due, without them this research would not have been possible. Firstly, I would like to thank Dr. Neil Storey and Dr. Peter Jones for their supervision and encouragement at all stages of my study at Warwick University. They have always given me advice and support whenever I needed it. I would also like to thank Dr. J. F. Craine for his advice and technical support in constructing my fuzzy control program generator.

I would like to acknowledge the support of Kuang-chih Huang, President of the National Kaohsiung Institute of Technology in the early stages of my application for paid leave for one year and for his encouragement during my period of study.

Special thanks go to Dr. T. I. Kuo who helped me to build my pH process control model plant, and to Dr. M. R. Cheng who has inspired me and helped me in many ways.

Above all, my deepest appreciation goes to my family. Without their complete support and the love of my two lovely daughters and my wife, my study in England for a doctoral degree would not have been possible.

Chapter 1

Introduction

In the chemical industry and in waste water treatment, the control of pH (the concentration of hydrogen ions) is a well-known problem that presents difficulties due to large variations in process dynamics and the static nonlinearity between pH and concentration. Usually it requires the application of advanced control techniques such as linear adaptive control[1,2,3,4,5], nonlinear adaptive control[6] or nonlinear control using a nonlinear transformation method[7]. However, these design methods are often complicated, as will be seen later in the survey section of this thesis (section 2.4).

1.1. pH Process Model Equations and pH Control Problems

pH is a measure of $[H^+]$,which denotes the concentration of hydrogen ions, in a solution. It is defined by:

$$pH = -\log[H^+] \qquad (1\text{-}1)$$

A well established method for modeling the dynamics of pH in a stirred tank is that developed by McAvoy[8]. This method, for single acid/single base systems, uses material balances for the anion of the acid and the cation of the base, together with all

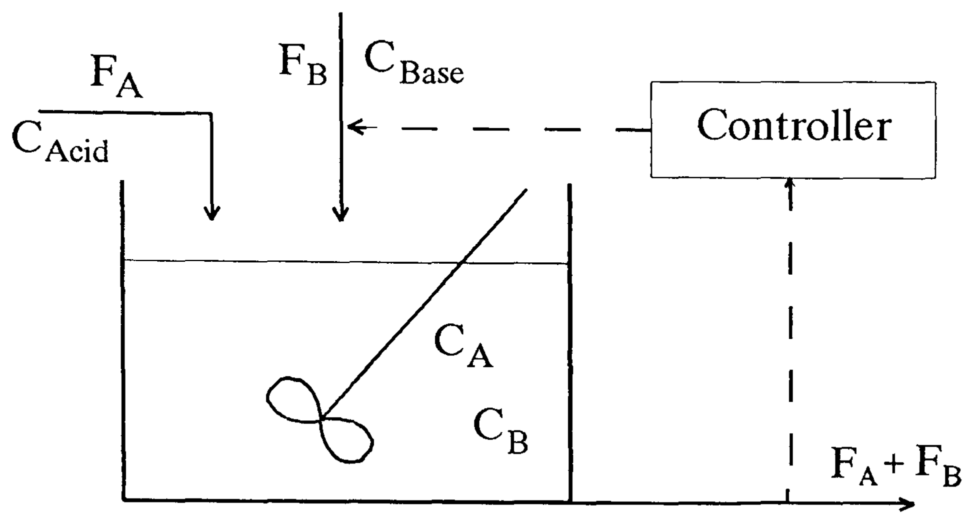


Figure 1-1 pH control system.

equilibrium equations and an electroneutrality restraint. For a typical pH control system as shown in Figure 1-1, there is a strong acid, HA, neutralized by a strong base, BOH, with the assumptions of constant volume, perfect mixing, and no species other than water present, the mole balances of the cation of the acid and the anion of the base are:

$$V\frac{dC_A}{dt} = C_{Acid}F_A - C_A(F_A + F_B) \qquad (1\text{-}2)$$

$$V\frac{dC_B}{dt} = C_{Base}F_B - C_B(F_A + F_B) \qquad (1\text{-}3)$$

here, C_A represents acid anion concentration and C_B represents base cation concentration in the effluent stream. C_{Acid} is the acid concentration in the acidic stream entering into this continuous stirred tank reactor (CSTR). C_{Base} is the base concentration in the basic stream entering CSTR. V is the volume of CSTR. F_A and F_B are flow rate of acidic and basic streams. These dynamic equations will be discussed in detail in chapter 6. To illustrate the difficulty of pH control, let $x = C_B - C_A$, the relation between the pH value and x which will be derived in chapter 6 is shown below

$$pH = f(x) = -\log[(\frac{x^2}{4} + K_w)^{\frac{1}{2}} - \frac{x}{2}]$$

The graph of the function f is called the *titration curve*. It is the fundamental nonlinearity for the neutralization problem. An example of the titration curve is shown in Fig 1-2. There is considerable variation in the slope of titration curves. In Figure 1-2, it also shows the titration curve for a weak acid and strong base. It can be seen that in this case the curve is not symmetrical about pH=7.

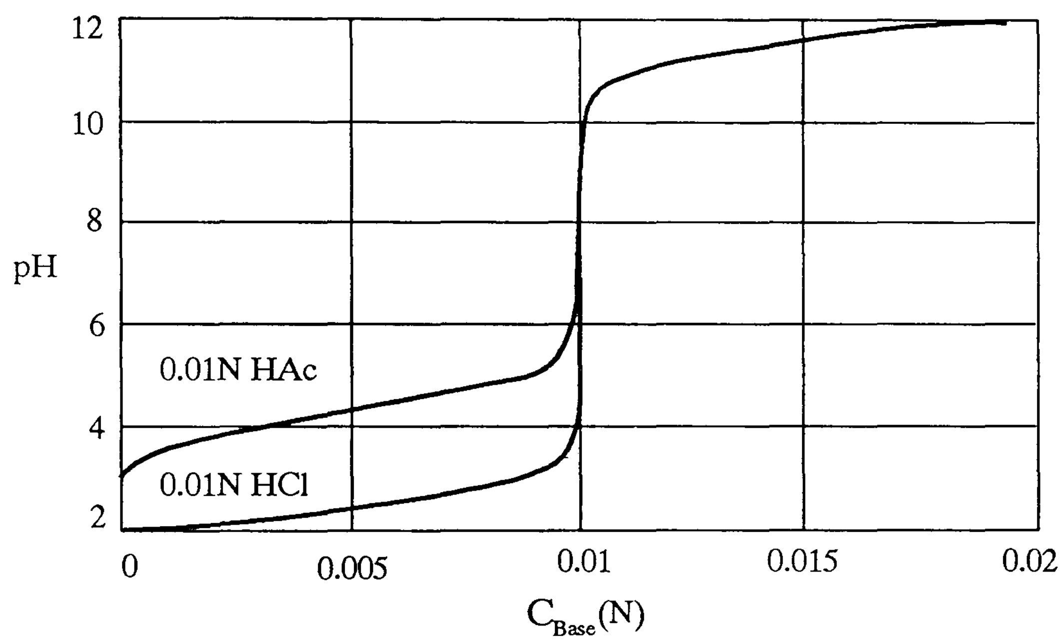


Figure 1-2 typical titration curves. (reproduced from[9])

Since the titration curve varies drastically with pH, if the pH process has been controlled by a conventional PI controller, the critical gain will vary accordingly. Some values are listed below for different values of the pH of the mixture[10].

pH	Critical gain
7	0.009
8	0.046
9	0.46
10	4.6

From these estimated critical gains, it showed that to make sure that the closed loop system is stable for small perturbations around an equilibrium of pH $=7$, the gain should thus be less than 0.009. A reasonable value of the gain for operation at pH $=8$ is $k=0.01$, but this gain will give an unstable system at pH $=7$ and is too low for a reasonable response at pH $=9$. Figure 1-3 shows a PI control with gain 0.01 to control the pH process at different set-points.

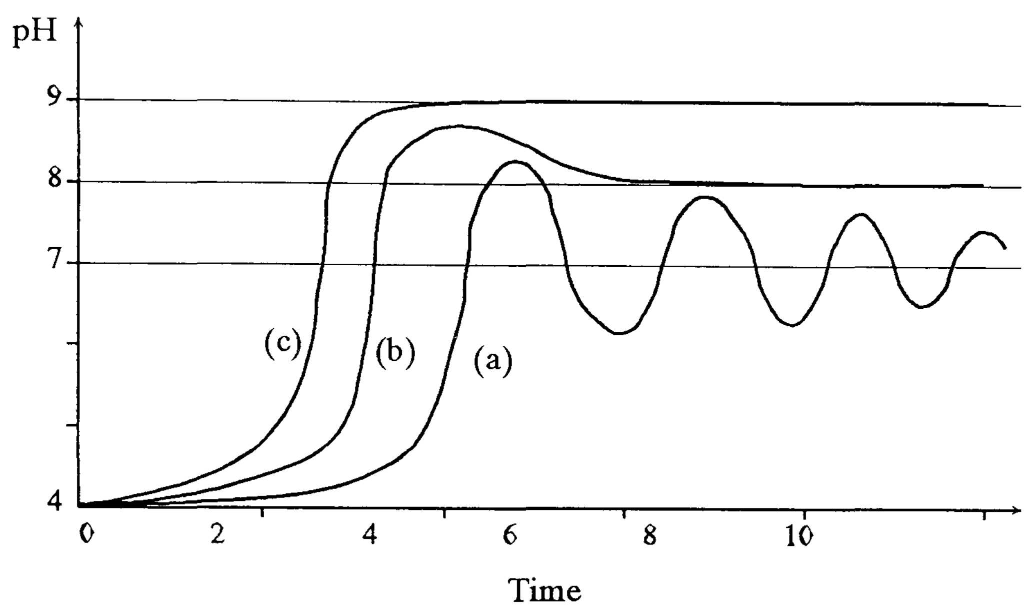


Figure 1-3. The pH outputs of a pH control process which controlled by a PI controller with set-point equal to (a)7; (b)8; (c)9.

1.2. Reasons for Using Fuzzy Logic in the Control of pH Process

Usually a control theory can be successfully applied only when the system under control can be sufficiently analyzed and a useful mathematical model has been found. When the process characteristics are known in advance, and are either constant or change predictably, a non- adaptive controller can be used to control it. On the other hand, if the characteristics of the process change with time in an unpredictable manner, an adaptive controller may be needed to control it. However, an adaptive controller still relies on a mathematical model of the process under control whose parameters are determined or modified in real time.

Difficulties arise in the control of the pH process due to the severe process non-linearity and frequent load changes. For example, changes in the influent composition or flow rate. The non-linearity can be understood from the S-shaped titration curve shown in Figure 1-2. Frequent and rapid load changes are common for most waste water treatment facilities since the influents come from the waste of a number of sources. It is therefore very difficult to analyze and derive the system model of a pH control process. However, experience shows that they are often controlled successfully by experienced human operator. Therefore, an alternative approach to the control of pH process is to investigate the control strategies employed by the human operators. A human operator can control a complex process effectively by simply attributing the difficulties he experienced to the rate or manner of information displayed or to the depth to which he may evaluate decisions. The operator's control strategy is based on intuition and experience, and can be considered as a set of heuristic decision rules. Usually these rules are expressed linguistically, and are often very difficult to convert into a quantitative control strategy.

The theory of fuzzy sets and algorithms developed by Zadeh[11,12] can be used to evaluated these imprecise linguistic statements directly. Fuzzy logic provides an effective means of capturing the approximate, inexact nature of the physical world. Therefore, it can be used to provide an algorithm which can convert a linguistic control strategy based on expert knowledge, into an automatic control strategy. The pioneering research of Mamdani and his colleagues on fuzzy control was started early in 1974 [13,14,15]. They developed a fuzzy controller for a boiler and steam engine. This showed that the fuzzy control system was less sensitive than conventional control systems to process parameter changes and that it gave good control at all operating points.

The work described in this thesis set out to investigate the suitability of fuzzy techniques for the control of pH within a continuous flow titration process.

1.3 Description of the Contents of this Thesis

The second chapter describes the basic concepts of fuzzy sets theory, fuzzy logic and approximate reasoning and also the design considerations for a fuzzy logic controller (FLC). This material will become the basis for later chapters. At the end of this chapter, current uses of fuzzy techniques and other non-linear methods in pH control are discussed briefly.

FLC industrial application techniques and FLC implementation methods are mentioned in the third chapter. After studying these implementation methods, a simple fuzzy developing system was designed and by which an experimental fuzzy control program was generated.

The fourth chapter gives a detailed study on the relationship between fuzzy decision table scaling factors and the control constants of a digital PI controller. Equations derived in this chapter are confirmed by experiments in the next chapter and become the basic tuning tools for all the fuzzy controllers used in later experiments.

Using an analog simulation of a plant, a fuzzy PI controller was compared with a conventional digital PI controller and described in beginning of the fifth chapter. It then describes a detailed study of the effects of the choice of scaling factors on the control actions of a fuzzy PI controller. It shows that, in some cases, a special sliding motion phenomenon will happen if the error change range of the decision table is too small. The boundary conditions of this sliding regime is also discussed. At the end of this chapter, by using these scaling factors, several other controllers are constructed which includes an integral controller and variable structure systems (VSS) formed by switching the scaling factors.

A discussion of the characteristics of the pH process and its experimental models provide the content of the sixth chapter. This includes the structure of the pH process model plant, the m del equations and static and dynamic behavior of the process.

Chapter seven contains the results and details of the pH control experiments performed. It shows that the fuzzy PI controller can make the system settle down easier and faster than the digital PI controller. The damping effect caused by the decision table near the set-point is also study in detail. Finally, the fuzzy PI control of pH process under varying set-points and noise are tested.

The last chapter provides the conclusions drawn from this study and some suggestions for further research.

Chapter 2

Theory and Design of a Fuzzy Logic Controller

After Mamdani's work on using a fuzzy controller to control an industrial process, reports of the application of fuzzy control techniques were widely spread through a number of fields, such as water quality control [16], automatic train operation systems [17], elevator control [18], nuclear reactor control [19], and roll and moment control for a flexible wing aircraft [20]. The literature on fuzzy control has grown rapidly, and in 1990 Lee[21] produced a comprehensive survey of control applications.

The following sections give a brief description of the basic concepts of fuzzy set theory and fuzzy logic, and includes elements from Lee's survey. This material will form a basis for later chapters.

2.1 Basic Definitions and the Fuzzy Set

A set U, denoted generally by $\{u\}$, is normally defined as a collection of elements or objects that can be discrete or continuous. U is called the universe of discourse and u are generic elements of U. Each element can either belong to or not belong to, a set A, $A \in U$. By defining a membership function for a set in which 1 indicates membership and 0 indicates nonmembership, a classical set can be represented by a set of ordered pairs (u,0) or (u,1). For a fuzzy set the membership function is allowed to have values between 0 and 1 to represent different degree of membership for the elements of a given set. Therefore, a fuzzy set F in a universe of discourse U is characterized by a membership function μ_F, which takes values in the interval [0,1] namely,

μ_F: U→[0,1].

Definition 1: Fuzzy set: A fuzzy set F in a universe of discourse U can be represented as a set of ordered pairs of a generic element u and its grade of membership function: $F = \{ (u, \mu_F(u)) \mid u \in U \}$. When U is discrete, a fuzzy set F is described by

$$F = \sum_{i=1}^{n} \mu_F(u_i) / u_i$$

where the symbol "/" is a separator and "Σ" means union. When U is continuous, a fuzzy set F can be written concisely as

$$F = \int_U \mu_F(u) / u$$

Definition 2: Support and Fuzzy Singleton: The support of a fuzzy set F, S(F), is the crisp set of all $u \in U$ such that $\mu_F(u) > 0$. In particular, a fuzzy set whose support is a single point in U with $\mu_F = 1.0$ is referred to as fuzzy singleton.

A fuzzy set is denoted in terms of its membership function. Therefore, the set theoretic operations of union, intersection and complement for fuzzy sets will be defined via their membership functions. Let A and B be two fuzzy sets in U with membership functions μ_A and μ_B, respectively. Some basic operations for fuzzy sets are shown below.

Definition 3: Intersection: The membership function $\mu_C(u)$ of the intersection $C=A\cap B$ is pointwise defined by

$$\mu_C(u) = \min\{\mu_A(u), \mu_B(u)\}, \qquad u \in U$$

Definition 4: Union: The membership function $\mu_C(u)$ of the union $C=A\cup B$ is pointwise defined by

$$\mu_C(u) = \max\{\mu_A(u), \mu_B(u)\} \qquad u \in U$$

Definition 5: Complement: The membership function $\mu_{\bar{A}}$ of the complement of a fuzzy set A is pointwise defined as

$$\mu_{\bar{A}}(u) = 1 - \mu_A(u)$$

Definition 6: Cartesian Product: If $A_1, \cdots, A_n$ are fuzzy sets in $U_1, \cdots, U_n$, respectively. Then, the Cartesian product of $A_1, \cdots, A_n$ is a fuzzy set in the product space $U_1 \times \cdots \times U_n$ with the membership function

$$\mu_{A_1 \times \cdots \times A_n}(u_1, u_2, \cdots, u_n) = \min\{\mu_{A_1}(u_1), \cdots, \mu_{A_n}(u_n)\}$$

or

$$\mu_{A_1 \times, \cdots, \times A_n}(u_1, u_2, \cdots, u_n) = \mu_{A_1}(u_1) \cdot \mu_{A_2}(u_2) \cdots \mu_{A_n}(u_n)$$

Definition 7: Fuzzy Relation: An n-ary fuzzy relation is a fuzzy set in $U_1 \times \cdots \times U_n$ and is expressed as

$$R_{U_1 \times \cdots \times U_n} = \{((u_1, \cdots, u_n), \mu_R(u_1, \cdots, u_n)) \mid (u_1, \cdots, u_n) \in U_1 \times \cdots \times U_n\}$$

Definition 8: Fuzzy relation: Sup-Star Composition: If R and S are fuzzy relations in $U \times V$ and $V \times W$, respectively. Then the composition of R and S is a fuzzy relation denoted by R o S and is defined by

$$R \circ S = \{[(u,w), \sup_{v}(\mu_R(u,v) * \mu_S(v,w))], u \in U, v \in V, w \in W\}$$

Definition 9: Fuzzy number: A fuzzy number F is a fuzzy set in the continuous universe U which is normal and convex, that is

$$\max_{u \in U} \mu_F(u) = 1 \qquad (normal)$$

$$\mu_F(\lambda u_1 + (1-\lambda)u_2) \geq \min(\mu_F(u_1), \mu_F(u_2)), \qquad (convex) \quad u_1, u_2 \in U, \ \lambda \in [0,1]$$

Definition 10: Linguistic Variables: A linguistic variable is defined by a quintuple (x,T(x),U,G,M) in which x is the name of a fuzzy variable; T(x) is the set of names of linguistic values of x with each value represented by a fuzzy number defined in U; G is a syntactic rule to generate the names for x; and M is a semantic rule to give the meaning of the x value. Usually, a linguistic variable can be regarded either as a variable whose value is a fuzzy number or as a variable whose value are defined in linguistic terms. For example, if there is a linguistic variable called "speed", then the term set T(speed) could be defined as

T(speed) = {slow, moderate, fast}

Each term in T(speed) is characterized by a fuzzy set in the universe of discourse U = [0,60] as shown in Figure 2-1. where "slow" could be interpreted as "a speed below about 15 mph", "medium" as "a speed close to 30 mph", and "fast" as "a speed more than 45 mph", etc.

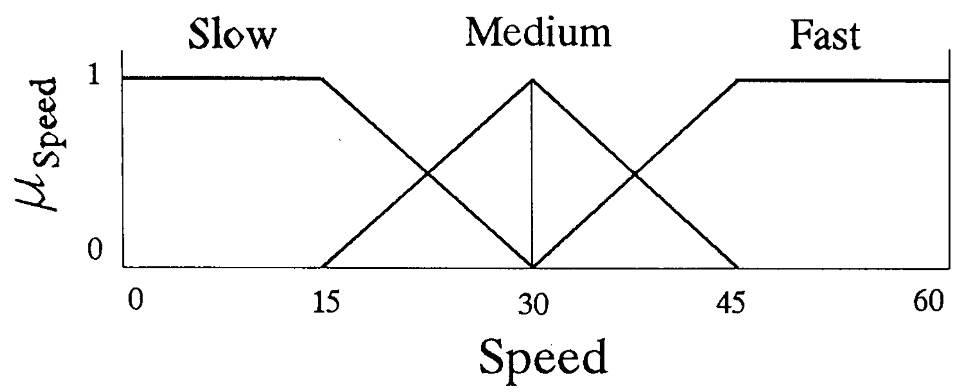


Figure 2-1. Diagrammatic representation of fuzzy "speed."

Definition 11: Sup-star Composition Rule of Inference: If R is a fuzzy relation in U×V, and u is a fuzzy set in U, then this rule asserts that the fuzzy set v in V induced by u is given by

$$v = u \circ R$$

where u o R is the sup-star composition of u and R.

This definition will become Zadeh's composition rule of inference if the star represents the minimum operator.

2.2 Structure of a Fuzzy Controller

The basic configuration of a Fuzzy logic controller (FLC) has been described by Lee in the survey paper discussed earlier[21]. He points out that usually there are four principal components in a FLC as shown in Figure 2-2. These components are a fuzzification interface, an inference engine (or decision-making logic), a knowledge base and a defuzzification interface. The functions of these are discussed briefly below

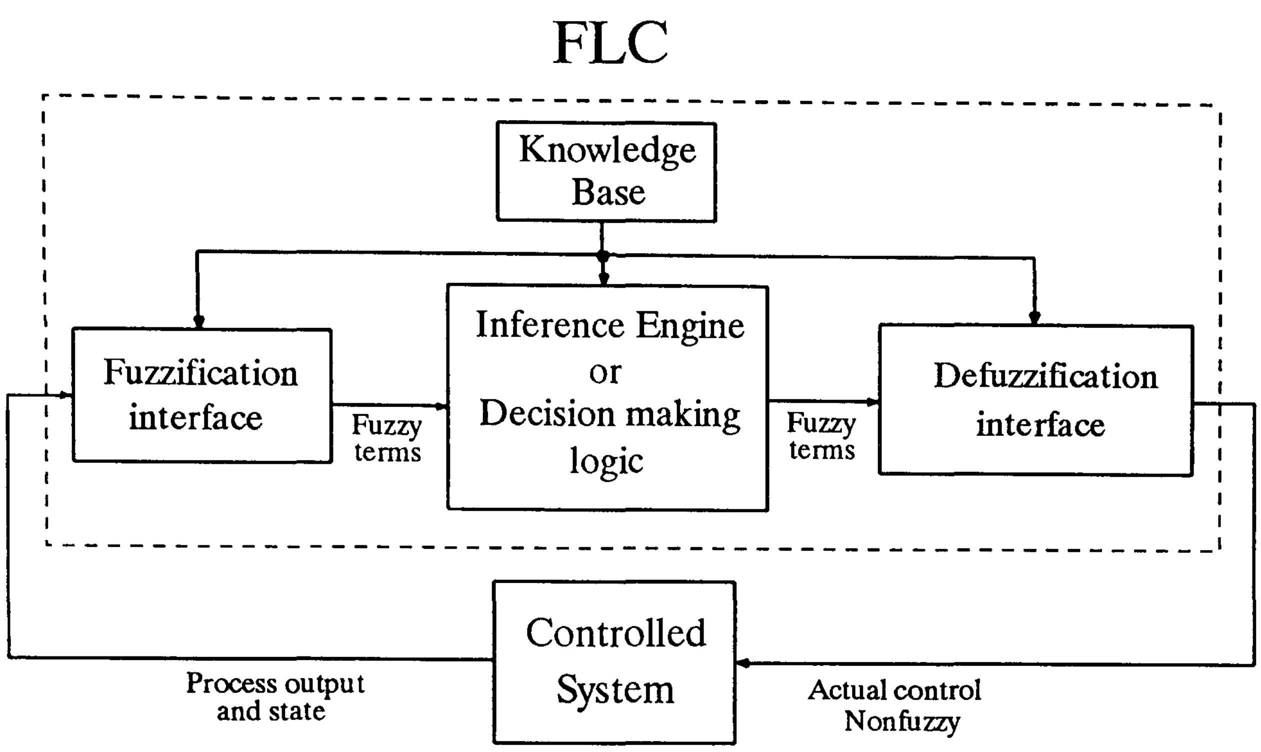


Figure 2-2. Basic configuration of a fuzzy logic controller (FLC).

(1) The fuzzification interface performs the following functions:

(a) it measures the values of the input variables (or the system state variables);

(b) it transfers the range of the values of the input variables into the corresponding universe of discourse by scale mapping. Usually this is done by normalization.

(c) it converts the input data into proper linguistic values which will be viewed as labels of fuzzy set.

(2) The knowledge base consists of a database and a linguistic control rule base. Its main functions are:

(a) to provide the necessary definitions, which are used to define the linguistic control rules and fuzzy data manipulation information.

(b) to provide the control goals and control policy of the domain experts which are written using a set of linguistic control rules.

(3) The inference engine is the kernel of the fuzzy logic controller.

It provides decision making using fuzzy concepts and infers fuzzy control action by fuzzy implication and the rules of inference of fuzzy logic.

(4) The defuzzification interface performs the following functions:

(a) it converts the range of values of output variables into the corresponding universe of discourse.

(b) it generates non-fuzzy control actions from the inferred fuzzy control action outputs.

A fuzzy system is characterized by a set of linguistic statements based on expert knowledge, which is usually in the form of "if-then" rules of the form:

IF (a set of conditions are satisfied)
THEN (a set of consequences will be inferred).

where both the set of conditions and the set of consequences are fuzzy terms. They are directly associated with fuzzy concepts and are referred to as "fuzzy conditional statements". Therefore, a fuzzy control rule is a fuzzy conditional statement in which the antecedent is a condition in its application domain and the consequent is a control action for the system under control.

The collection of these fuzzy control rules forms the rule base or the rule set of an fuzzy logic controller (FLC). For instance, if x_1 and x_2 denote the state variables of the target plant, and y is the output of the FLC, then the IF (antecedent)-THEN (consequence) control rule might be expressed as a control algorithms of the form:

if x_1 is small and x_2 is big then y is medium

if x_1 is big and x_2 is medium then y is big

The antecedent (A and B) is interpreted as a fuzzy set $A \times B$ in the production space $U \times V$ with membership functions $\mu_{A\times B}(u,v)$ and will be discussed latter in section 2.3.5(2). Therefore, a fuzzy control rule, such as "if (x is A_i and y is B_i) then (z is C_i)" can be implemented by a fuzzy implication (relation) R_i as follows:

$$\mu_{R_i} \triangleq \mu_{(A_i \wedge B_i \rightarrow C_i)}(u, v, w)$$
$$= [\mu_{A_i}(u) \wedge \mu_{B_i}(v)] \rightarrow \mu_{C_i}(w)$$

Where A_i and B_i are fuzzy sets, $A_i \times B_i$ in $U \times V$. The fuzzy implication $R_i \triangleq (A_i$ and $B_i) \rightarrow C_i$ is in $U \times V \times W$; "$\rightarrow$" denotes a fuzzy implication (relation) function which will also be discussed later in section 2.3.5(1)

2.3 Design considerations for an FLC

As mentioned earlier, the main function of the FLC is to provide an algorithm which can convert the linguistic control strategy based on expert knowledge into an automatic control strategy. Then, the principle design considerations for an FLC may be divided into five parts:

These are:

(1) fuzzification strategies;

(2) defuzzification strategies;

(3) data base considerations:

(a) discretization/normalization methods for the universes of discourse,

(b) the choice of the membership functions for the primary sets,

(c) the fuzzy partition of the input and output spaces.

(4) rule base considerations:

(a) the choice of the state and control variables for the fuzzy control rules,

(b) the derivation of fuzzy control rules,

(c) the justification of fuzzy control rules,

(d) the choice of fuzzy control rule types.

(5) decision making logic considerations:

(a) the definition of fuzzy implications,

(b) the interpretation of sentence connectives "and" and "also",

(c) the definition of compositional operators,

(d) the inference mechanisms;

These five parts will be discussed in the following sections.

2.3.1 Fuzzification strategies

Fuzzification could be defined as a mapping from an observed input space to fuzzy sets in a certain input universe of discourse. The data manipulation in an FLC is based on the fuzzy set theory, however, in control applications the observed data are usually crisp values. Therefore, fuzzification is necessary at an early stage. Fuzzification is normally achieved in one of two ways. Its results are shown in Figure 2-3(a) and (b). Both cases show an ordinary fuzzy set B intersecting with a fuzzy set A. A is a fuzzy set coming from the input signal x_0 after fuzzification. In Figure 2-3(a), A is a singleton but in (b) A is a triangular fuzzy set. The two fuzzification methods are

(1) A crisp value of some input variable can be viewed as a fuzzy singleton within its universe of discourse. In this case a crisp input x_0 will look like a fuzzy set A, with the membership function $\mu_A(x)$ that is equal to zero everywhere but at the point x_0, where $\mu_A(x_0)$ is equal to one as shown in Figure 2-3(a). The advantages of this approach

is that no processing is required for fuzzification in this case. This strategy has been widely used in fuzzy control applications, since it is very easy to implement.

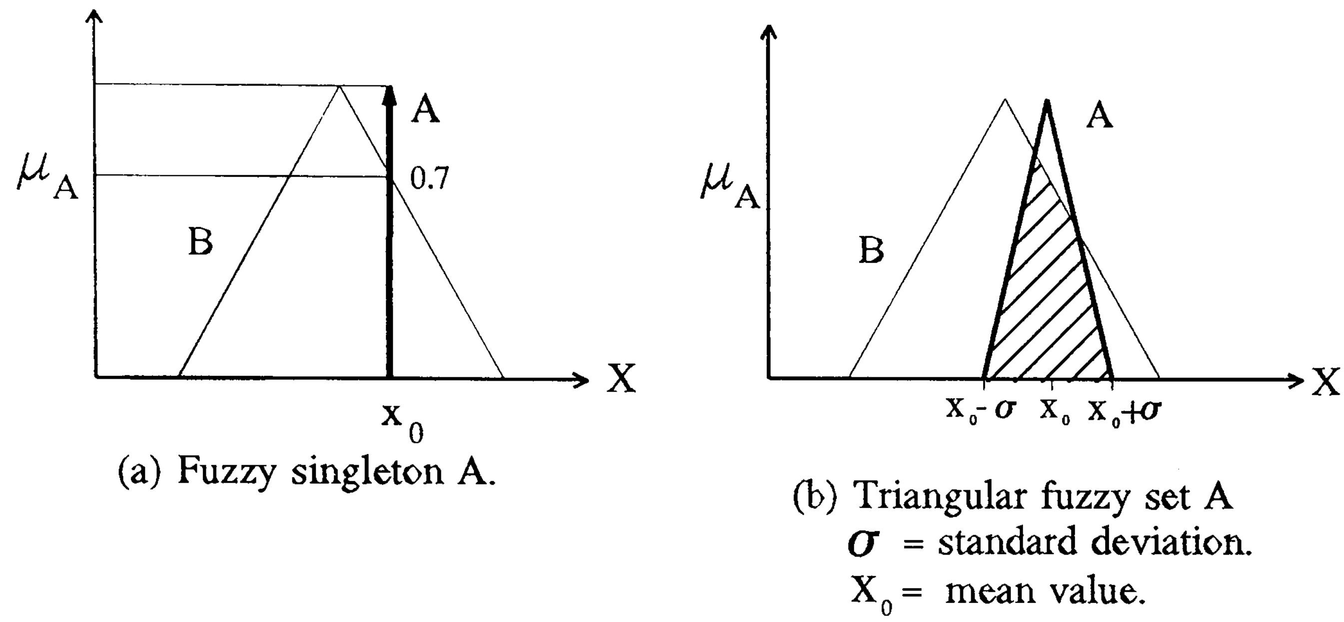


Figure 2-3.methods of fuzzification: The measured data are (a) converted to a fuzzy singleton. (b) converted to a triangular fuzzy set.

(2) In many applications the observed data are disturbed by random noise. In this case, it is more appropriate to choose an isosceles triangle as the fuzzification function. The vertex of this triangle corresponds to the mean value of a data set, while the base is twice the standard deviation of the data set. Then, we can use this fuzzy number for control manipulation. Figure 2-3(b) shows the second method of fuzzification. The fuzzy set A with its vertex at x_0 and having the base width equals to 2σ is the resulting triangular set. The point x_0 is the mean value and σ is the standard deviation of the measured data set.

2.3.2 <u>Defuzzification strategies</u>

In many practical applications a crisp control action is required to drive the system. Therefore, defuzzification is needed to produce a non-fuzzy control action which best represents the possibility distribution of the inferred fuzzy control action. In other words, the defuzzification process is a mapping between a space of fuzzy control actions defined over an output universe of discourse, and a space of non-fuzzy control actions. There are several kinds of strategies being commonly used. These include the maximum criteria, the mean of maximum and the center of area methods.

(a) The maximum criteria method

This method produces the point at which the possibility distribution of the control action reaches a maximum value.

(b) The Mean of Maximum Method (MOM)

The MOM strategy generates a control action which represents the mean value of all local control actions whose membership functions reach the maximum. In the case of a discrete universe, the control action will be expressed as

$$z_0 = \sum_{j=1}^{n} \frac{w_j}{n}$$

where w_j is the support value at which the membership function value $\mu_j(w_j)$ reaches the maximum, and n is the total number of such support values.

(c) The Center of Area Method

The COA method is the most popular method currently being used by control engineers. This strategy generates the center of gravity of the possibility distribution of a control action. The formula is shown below

$$z_0 = \frac{\int_z \mu_C(z) * z dz}{\int_z \mu_C(z) dz}$$

In the case of a discrete universe, the method yields

$$z_0 = \frac{\sum_{j=1}^{n} \mu_z(w_j) \cdot w_j}{\sum_{j=1}^{n} \mu_z(w_j)}$$

(d) The Center of Sums

This method is the same as the COA method except that the overlapping areas of the membership functions must be counted twice during the calculation. The calculation formula is adjusted as

$$z_0 = \frac{\int_z z \cdot \int \mu_C(z) dz}{\int_z \int \mu_C(z) dz}$$

(e) The Height method

This method is simpler than the COA approach, since there is no integration during calculation. The calculation formula is

$$z_0 = \frac{\sum z_i * h_i}{\sum h_i}$$

where h_i is the strength of the inferred antecedent output of the *i*th rule, and z_i is the center of the corresponding output fuzzy set of the ith rule.

Braae and Rutherford[22] have performed detailed analysis of various defuzzification strategies and observed that the COA strategy yields better results which are similar to those obtained with a conventional PI controller. However, the MOM strategy yields a better transient performance, while the COA method is better in its steady state performance.

2.3.3 Data Base consideration

The concepts associated with a data base are used to characterize fuzzy control rules and fuzzy data manipulation in an FLC. Since these concepts are basically defined according to experience and engineering judgement, a "good" choice of the membership functions of a fuzzy set will play an essential role in the success of the fuzzy controller application. In this section some of the important considerations relating to the construction of the data base will be discussed.

2.3.3.1 Discretization / normalization methods for the universes of discourse

The method used for the representation of linguistically described information using fuzzy sets usually depends on the nature of the universe of discourse. A universe of discourse in a FLC may be either discrete or continuous. If it is continuous then some discretization and normalization process will be needed before the primary fuzzy sets are applied on it.

(a) Discretization: The process of discretizing a universe of discourse is frequently referred to as the quantitization process. This process divides a universe of discourse into a number of segments.

Each segment (or range) is numerically labeled as a generic element and forms a discrete universe. Then a fuzzy set will be defined by assigning grade of membership values to each generic element of this discrete universe. Table 2-1 shows an example of a discretization look-up table, where a universe of discourse is discretized into 9

Table 2-1

Quantization and Primary Fuzzy sets Using a Numerical Definition

Level No.	Range	NB	NM	NS	ZE	PS	PM	PB
-4	$x \leq -4$	1	0.7	0	0	0	0	0
-3	$-4 \leq x \leq -2.5$	0.7	1	0.5	0	0	0	0
-2	$-2.5 \leq x \leq -1.5$	0	0.7	0.7	0	0	0	0
-1	$-1.5 \leq x \leq -0.5$	0	0.3	1	0.3	0	0	0
0	$-0.5 \leq x \leq 0.5$	0	0	0.3	1	0.3	0	0
1	$0.5 \leq x \leq 1.5$	0	0	0	0.3	1	0.3	0
2	$1.5 \leq x \leq 2.5$	0	0	0	0	0.7	0.7	0
3	$2.5 \leq x \leq 4$	0	0	0	0	0.5	1	0.7
4	$4 \leq x$	0	0	0	0	0	0.7	1

levels with 7 primary fuzzy sets defined on it. Note that the scale mapping of the measured variable values x into the discretized universe can be uniform or nonuniform. Looking into Table 2-1 we find that it is not a uniform mapping, since the width of the ranges for all the quantitization levels are equal to 1 except level 3. The shape of the membership functions of primary fuzzy sets, such as ZE and PM, will be discussed in next section.

(b) Normalization: The normalization of a continuous universe requires a discretization of the universe of discourse into a finite number of segments, with each segment mapped into a segment of the normalized universe. Then, using an explicit function, a fuzzy set will be defined to its membership function. One example is shown in Table 2-2. where the universe of discourse, [-6, +4.5] is mapped into the normalized interval [-1, +1];

Table 2-2

Quantization and Primary Fuzzy sets Using a Numerical Definition

Normalized Universe	Normalized Segments	Range	μ	σ	Primary Fuzzy Sets
	[-1.0,-0.5]	[-6.9,-4.1]	-1.0	0.4	NB
	[-0.5,-0.3]	[-4.1,-2.2]	-0.5	0.2	NM
	[-0.3,-0.0]	[-2.2,-0.0]	-0.2	0.2	NS
[-1.0,+1.0]	[+0.0,+0.2]	[+0.0,+1.0]	0.0	0.2	ZE
	[+0.2,+0.6]	[+1.0,+2.5]	0.2	0.2	PS
	[+0.6,+1.0]	[+2.5,+4.5]	0.5	0.2	PM
			1.0	0.4	PB

2.3.3.2 The choice of the membership functions for the primary fuzzy sets

There are two kinds of fuzzy set definitions, depending on whether the universe of discourse is discrete or continuous.

(a) Numerical Definition: In this case, the grade of membership function of a fuzzy set is represented by a vector of numbers. The dimension of the vector depends on the degree of discretization. If the membership function of a primary fuzzy set of this kind has the form

$$\mu_f(u) = \sum_{i=1}^{7} \frac{a_i}{u_i}$$

where a = [0.1, 0.3, 0.7, 1.0, 0.7, 0.3, 0.1] is the vector of grades, its dimension is equal to 7. Therefore, the primary fuzzy set "ZE" in Table 2-1 can be represented by its membership function together with its corresponding u_i as following

$$ZE = [\frac{0}{-4}, \frac{0}{-3}, \frac{0}{-2}, \frac{0.3}{-1}, \frac{1}{0}, \frac{0.3}{1}, \frac{0}{2}, \frac{0}{3}, \frac{0}{4}]$$

Therefore, under this definition a fuzzy singleton will have the form of s=[0,0,0,0,0,1,0] without showing its value of quentitization level.

(b) Functional Definition: The membership function of a fuzzy set can sometimes be expressed in a functional form, typically using triangle or bell shaped functions. It is very easy to describe a triangle shape by function, for example, one could define a triangle fuzzy set named "slow", which has vertex at 20 km and two base points at 0 km and 40 km. The following function can then be used to calculate its grade:

$$\mu_{slow}(x) = \frac{20 - |x - 20|}{20} \qquad , if \;\; 0 \le x \le 40$$

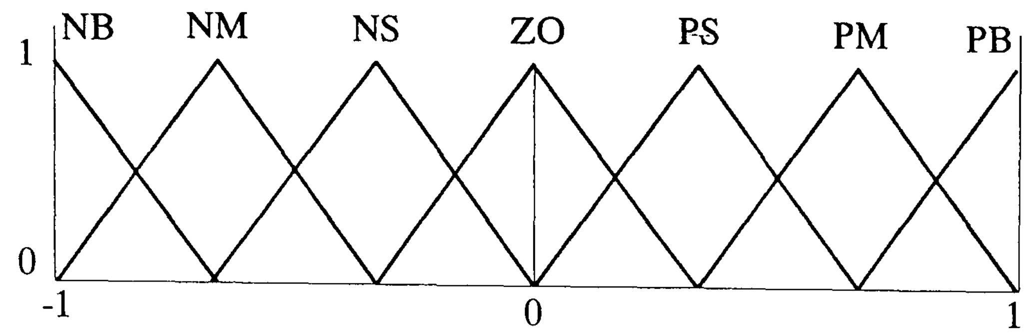
When using bell shaped fuzzy sets, a Gaussian normal distribution curve is commonly used. This may be described by the equation:

$$f(x) = \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

where μ denotes the mean value and σ denotes the standard deviation.

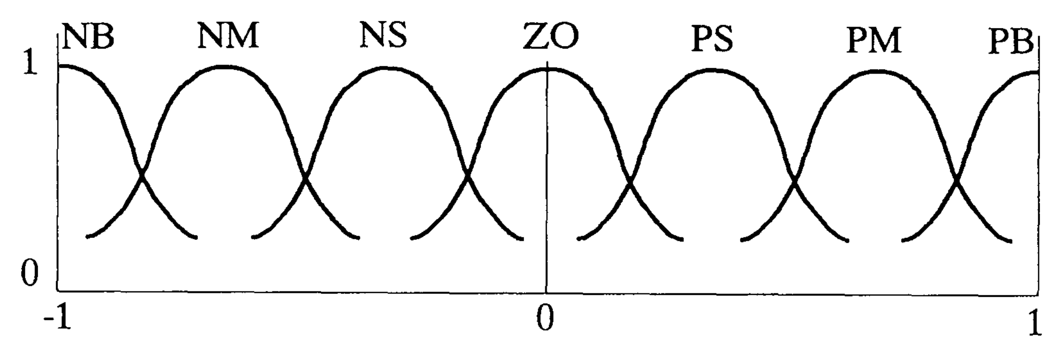
The method used to assign the grades of membership to the primary fuzzy set using these functions is very important. If the incoming signals are disturbed by noise, then the membership functions should be wide enough to reduce the sensitivity to this noise.

Methods of storing these functional fuzzy sets within computer memory will be discussed later. Figure 2-4 and 2-5 shows the triangle and the Gaussian curve shaped fuzzy sets.



$$\mu(x) = \frac{1}{a}(a - |x-b|) \vee 0, \quad a>0$$

Figure 2-4. The triangle shaped fuzzy sets.



$$\mu_c(x) = \exp\left[\frac{-(x-m)^2}{\sigma^2}\right]$$

Figure 2-5. The bell shaped fuzzy sets

2.3.3.3 <u>The fuzzy partition of the input and output space</u>

Within fuzzy systems, both the input (antecedent) and output (consequence) linguistic variables form fuzzy spaces with respect to their own universes of discourse. In general, a linguistic variable is associated with a term set, with each term in the term set being defined on the same universe of discourse. A fuzzy partition determines how many terms exist within the term set. A fine partition produces more terms than a coarse one. Figure 2-6 shows two typical examples of fuzzy partitions in the same normalized universe [-1,+1].

The cardinality (total number of terms) of a term set in a fuzzy input space determines the maximum number of fuzzy control rules that can be constructed. For instance, in the case of two-input, one-output fuzzy controller, if the cardinalities of the term sets of these two inputs are all 7, then, the maximum number of control rules is 7x7=49.The fuzzy partition of the fuzzy input/output space,is normally non-deterministic and has no unique solution. A heuristic cut and trial procedure is usually needed to get the appropriate number of fuzzy partitions.

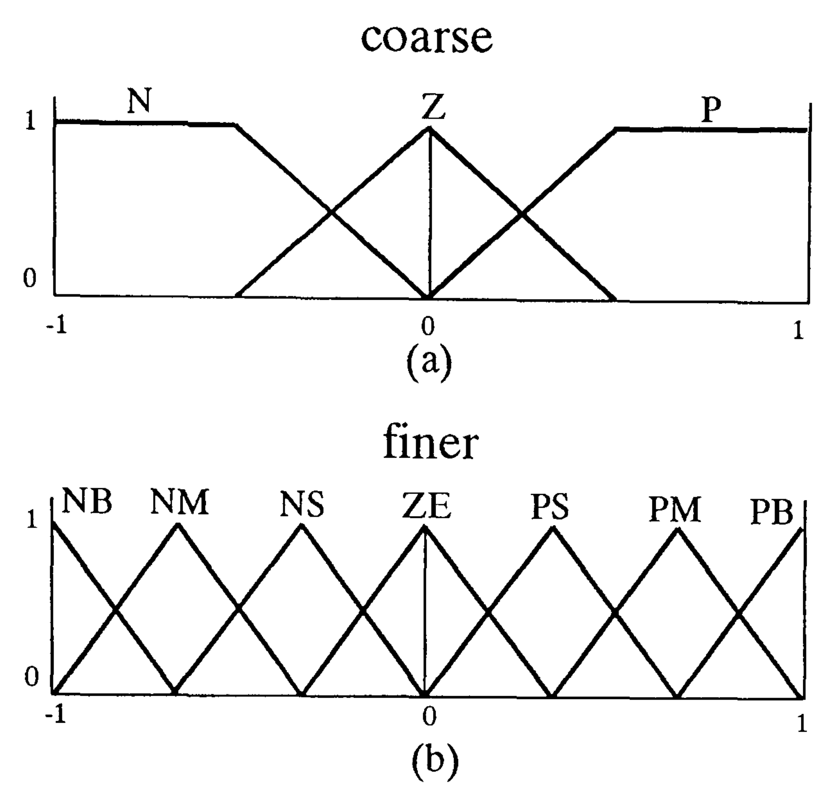


Figure 2-6. Diagrammatic representation of fuzzy partition. (a) 3 terms coarse partition(b) 7 terms finer partition.

2.3.4 Rule Base consideration

A primary task in the design of a fuzzy controller is the construction of the control rule base. Topics to be considered include: the choice of process state and control (output) variables; the derivation and justification of control rules; the type of fuzzy control rules to be used; and issues of consistency, interactivity, and completeness [21] of control rules.

2.3.4.1 The choice of the state and control variables for the fuzzy control rules

The selection of the linguistic variables greatly influences the performance of an FLC. As mentioned earlier, experience and engineering knowledge play a major role in their selection. Typically, the linguistic variables in an FLC are the state, the state error, the state error integral, the state error derivative, etc.

2.3.4.2 The derivation of fuzzy control rules

Tagaki and Sugeno [23] concluded that there are four ways to generate a set of fuzzy control rules. These four ways are not mutually exclusive, and may be combined to form an effective way to construct the fuzzy control rules.

(a) *Using Expert Experience and Control Engineering Knowledge :*

In everyday live most of the information on which our decisions are based is linguistic rather than numerical in nature. In this respect, fuzzy control rules provide a natural framework for the characterization of human behavior and for analyzing the decision making process. It was realized that fuzzy control rules could provide a convenient way to express domain knowledge. Therefore, most FLCs are based on the knowledge and experience which are expressed in the language of fuzzy if-then rules [14],[24],[25].

The formulation of fuzzy control rules can be done in two ways. The first one involves an introspective verbalization of human expertise. Examples of such verbalization are those operating manuals for the production plant. The second includes an interrogation of experienced experts or operators using a carefully organized questionnaire. By these two methods a prototype set of fuzzy control rules can be produced for a particular application. However, it will generally be necessary to optimize the performance of the system using a trial and error approach.

(b) *Using Information on Operator's Control Actions:*

In many man-machine control systems, the relationship between the input-output are not known sufficiently clearly, to make it possible to employ classical control theory for modeling and simulation. However, human operation can perform quite complex tasks in the absence of such models. For example, a skilled driver can park a car successfully without being aware of a quantitative model of the car. In fact, a set of fuzzy if-then control rules is being employed consciously or subconsciously by the car operator. Therefore, rules deduced from the observation of a human controller's actions, are a useful source of information of a rule base. Sugeno has presented a series of papers discussing the automation of this process [26][27][28].

(c) *Using a Fuzzy Model of a Process:*

By analogy with classical control system modeling, a linguistic description of the dynamic characteristics of a control system can be viewed as a fuzzy model of the system. Based on this fuzzy model, it is possible to obtain a set of fuzzy control rules for achieving near optimal control of the system. This set of control rules can then be used as the rule base of an FLC.

Some fuzzy modeling techniques have been reported [29][30], but this approach to the design of an FLC has not been fully developed [21].

(d) *Using Technique Based on Learning:*

Early in 1979 Procyk and Mamdani presented the first self-organizing controller (SOC)[31]. Such controllers are very attractive in applications as they automatically tune, and retune, themselfs on line, and require no special expertise from the operator once installed[45]. The SOC has a hierarchical structure which consists of two rule bases. The first is the general fuzzy rule base of an FLC. However, the second is constructed by "meta-rules" which have the ability to create and modify the general rule base, based on the desired overall performance of the system. In 1991, Lee presented a new method of SOC [32]. He employed an approximation reasoning and neural net concept for a self-learning rule-based controller. In Lee's controller there are two very important units: the Associative Critic Neuron (ANC), and the Associative Learning Neuron (ALN). The former evaluates the output response produced by present control action and creates an evaluation signal. The latter adjusts the break point of the membership functions in the rule base based on signals received from the ACN. Combining neural network theory with fuzzy control theory is seen as an area of great potential in the design of FLCs.

2.3.4.3 The justification of fuzzy control rules

The most common method applied to the justification of fuzzy control rules is called "scale mapping", which was first presented by King and Mamdani [33]. Rule justification is done by considering the nature of the response of the system when plotted on a phase plane diagram. This process is best understood through the use of an example phase plane analysis.

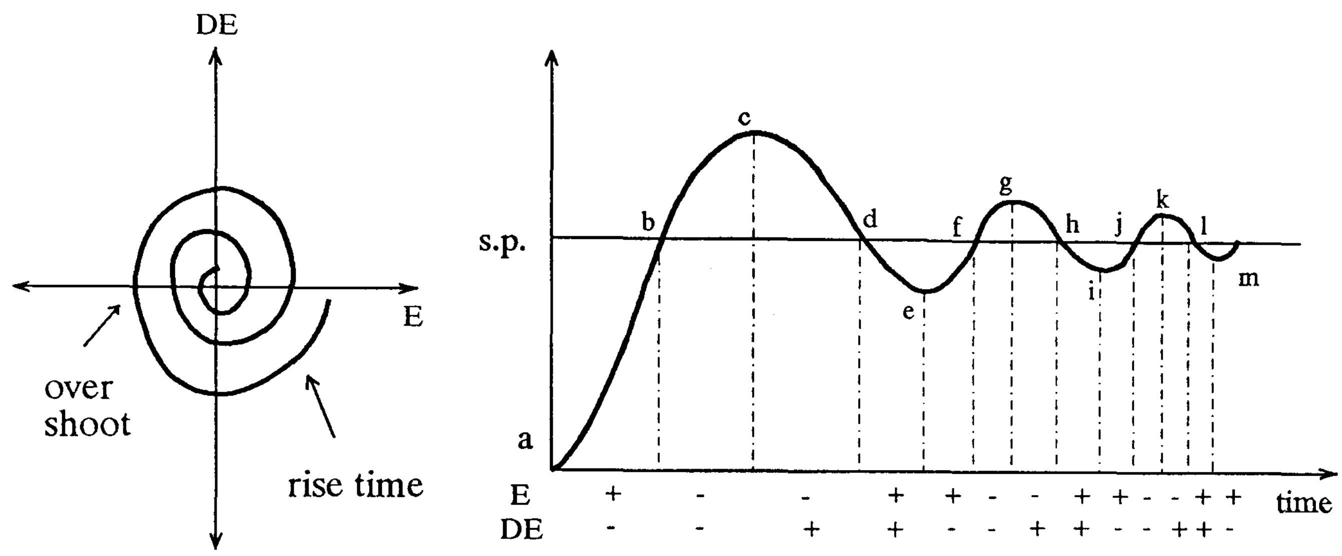
Consider a system using the error E and change of error DE as input variables and a rule linguistically described as follows:

IF E (error) =PB and DE (change of error) =PS
THEN y (output) =NM

(PB: positive big, PS: positive small, NM: negative medium.)

A possible response of such a system is shown in Figure 2-7. This shows the phase-plane trajectory of the system and its step response. If from the system step response it was found that the overshoot between point (b) and (c) was too big, then, the output fuzzy set NM in the decision table corresponding to the position (b)-(c), could be changed to NB. This would increase the retarding force to reduce the overshoot or adjust the value of membership function for NM to improve it.

A similar method was suggested by Braae and Rutherford [34]. They tracked the linguistic trajectory of the closed loop system in a 'linguistic phase plane'. The prnciples involved in this approach is similar to those described above.



(a) phase plane trajectory (b) system step response

Figure 2-7. Rule justification by using phase plane.

2.3.4.4 <u>The choice of the fuzzy control rule types</u>

Two types of fuzzy control rules are currently in use in the design of FLCs. These are "state evaluation" fuzzy control rules and "object evaluation" fuzzy control rules. A third type, slightly modified from the first of these, is also quite popular in the industrial application. Its consequent instead of using linguistic values is represention as a function of the process variables. These three types of fuzzy control rules are discussed briefly below

(a) *State Evaluation Fuzzy Control Rules:*

State evaluation fuzzy control rules are the most commonly used form of rules in FLC design. In the case of multi-input, single-output (MISO) systems, these may be characterized by the following form

$$R_i\text{: if x is } A_i, \cdots, \text{and y is } B_i \text{ then z is } C_i$$

where x,$\cdots$,yand z are linguistic variables representing the process state variables and the fuzzy control output variable; A_i,$\cdots$,B_i and C_i are the linguistic values of the linguistic variables x,$\cdots$,y, and z.

(b) *State Evaluation Fuzzy control Rules with Functional Output :*

In such rules, the consequent is represented as a function of the process state variables x,$\cdots$,y.i.e.,

$$R_i\text{: if x is } A_i, \cdots, \text{and y is } B_i \text{ then } z = f_i(x, \cdots, y)$$

Such type of rules were first proposed by Sugeno[23], and are now widely used in industry.

(c) *Object Evaluation Fuzzy Control Rules*:

Fuzzy control using objective evaluation control rules is also called 'predictive fuzzy control', and was first proposed by Yasunobu, Miyamoto, and Ihara[35]. A control command is inferred from an objective evaluation of a fuzzy control result that satisfies the desired states and objectives. The rule will be described as

$$R_i\text{: if(u is } C_i \text{ ->(x is } A_i \text{ and y is } B_i)) \text{ then u is } C_i.$$

A control command u takes a crisp set as a value, such as "changed" or "not changed", and x, y are performance indices for the evaluation of the *i*th rule, taking values such as "good" or "bad". Then the most likely control rules will be selected through predicting the results (x,y) corresponding to every control command C_i.

This rule can be interpreted linguistically as: "if the performance index x is A_i and index y is B_i when a control command u is chosen to be C_i, then this rule is selected and the control command C_i will be taken to be the output of the controller."

In practical, it is possible that more than one system state will give the same result and prediction becomes more complicated.

2.3.5 Decision making logic consideration

The use of an FLC can be viewed as trying to emulate human decision making within the conceptual framework of fuzzy logic and approximate reasoning. In this context, the forward data-driven inference (generalized modus ponens), this term will be explained later, plays a very important role. In this section, the properties of fuzzy implication functions, sentences connectives, composition operators and related concepts will be introduced.

(1) *Fuzzy Implication Functions*

A fuzzy control rule, is essentially a fuzzy relation which is expressed as a fuzzy implication. In fuzzy logic, a fuzzy implication may be defined using some kind of fuzzy implication functions.

(a) *Basic properties of a fuzzy implication function:*

As is well known, there are two important fuzzy implication inference rules in approximate reasoning. They are the generalized modus ponens (GMP) and the generalized modus tollens (GMT), both of them come from tautologies such that modus ponens: $(A \wedge (A \Rightarrow B)) \Rightarrow B$, and modus tollens: $((A \Rightarrow B) \wedge (\text{not } B)) \Rightarrow \text{not } A$. Here, "$\Rightarrow$" is the fuzzy implication or relation usually denoted by relation matrix "R". If A, A', B, and B' are fuzzy predicates, then these two rules can be described as following,

premise 1: x is A'
premise 2: if x is A then y is B (GMP)
consequence: y is B'

premise 1: y is B'
premise 2: if x is A then y is B (GMT)
consequence: x is A'

For the GMP case, "y is B'" results from matrix operation $B' = A' \circ R$. It has forward data-driven characteristics, thus which is more suitable for ordinary control applications.

(b) *Families of Fuzzy Implication Functions:*

There are at least 40 different kinds of fuzzy implication functions, in which the antecedents and consequences contain fuzzy variables. Before the inference mechanisms are discussed, in this section, several definitions must be briefly introduced.

Definition 1: Triangular Norms: The triangular norm $*$ is an operation represented by $* : [0,1] \times [0,1] \rightarrow [0,1]$, which includes the following operations

For all $x,y \in [0,1]$:

intersection $x \wedge y = \min\{x,y\}$

algebraic product $x \cdot y = xy$

bounded product $x \odot y = \max\{0, x+y-1\}$

drastic product $x \otimes y = \begin{cases} x & y = 1 \\ y & x = 1 \\ 0 & x,y < 1 \end{cases}$

Definition 2: Triangular Co-Norms: The triangular co-norm $\dotplus$ is also a two-place function denoted by $\dotplus:[0,1]\times[0,1]\rightarrow[0,1]$, which includes

union	$x \vee y = \max\{x,y\}$
algebraic sum	$x \uplus y = x + y - xy$
bounded sum	$x \oplus y = \min\{1,x+y\}$

drastic sum

$$x \mathbin{\dot\oplus} y = \begin{cases} x & y = 1 \\ y & x = 1 \\ 0 & x, y < 1 \end{cases}$$

Definition 3: Fuzzy conjunction: The fuzzy conjunction is defined for all $u \in U$ and $v \in V$ by

$$\begin{aligned} A \rightarrow B &= A \times B \\ &= \int_{U\times V} \mu_A(u) * \mu_B(v) / (u, v) \end{aligned}$$

where $*$ is an operator representing a triangular norm.

Definition 4: Fuzzy disjunction: The fuzzy disjunction is defined for all $u \in U$ and $v \in V$ by

$$\begin{aligned} A \rightarrow B &= A \times B \\ &= \int_{U\times V} \mu_A(u) \dotplus \mu_B(v) / (u, v) \end{aligned}$$

where $\dotplus$ is an operator representing a triangular co-norm.

Definition 5: Fuzzy Implication: There are five families of fuzzy implication functions in use. Here, as before, $*$ denotes a triangular norm and $\dotplus$ is a triangular co-norm.

a) material implication

$$A \rightarrow B = (\text{not } A) \dotplus B$$

b) Propositional calculus:

$$A \rightarrow B = (\text{not } A) \dotplus (A * B)$$

c) Extended propositional calculus:

$$A \rightarrow B = (\text{not } A \times \text{not } B) \dotplus B$$

d) Generalization of modus ponens:

$$A \rightarrow B = \sup \{ c \in [0,1],\ A * C \leq B \}$$

e) Generalization of modus tollens:

$$A \rightarrow B = \inf \{ t \in [0,1],\ B \dotplus t \leq A \}$$

Based on these definitions, many fuzzy implication functions may be generated by employing triangular norms and co-norms. Examples include the following fuzzy implications, which are called the mini-operation rule of fuzzy implications and the product operation rule of fuzzy implication. These were developed by Mamdani[13] and Larsen[65] respectively, and are often used in FLCs.

mini-operation rule:

$$\begin{aligned} R_c &= A \rightarrow B \\ &= A \times B \\ &= \int_{U \times V} \mu_A(u) \wedge \mu_B(v) / (u, v) \end{aligned} \qquad (2\text{-}1)$$

product-operation rule:

$$\begin{aligned} R_p &= A \times B \\ &= \int_{U \times V} \mu_A(u) \mu_B(v) / (u, v) \end{aligned} \qquad (2\text{-}2)$$

(2) *Interpretation of Sentence Connectives "and" and 'also"*

Commonly, the sentence connective "and" is implemented as a fuzzy conjunction in a Cartesian product space in which variables are taking values from different universes of discourse. For instance, in " if (A and B) then C," the antecedent (A and B) is interpreted as a fuzzy set $A \times B$ in the product space $U \times V$, with the membership function given by

$$\mu_{A\times B}(u,v) = \min\{\mu_A(u), \mu_B(v)\} \tag{2-3}$$

or

$$\mu_{A\times B}(u,v) = \mu_A(u) \cdot \mu_B(v) \tag{2-4}$$

Where U and V are the universe of discourse associated with A and B respectively.

When a fuzzy system is characterized by a set of fuzzy control rules, the ordering of the rules is immaterial. This implies that the sentence connective "also" should have the properties of commutativity and associativity. Therefore, the operators in triangular norms and co-norms possessing these properties are suitable to be used to interpret the connective "also".

Finally from the practical point of view, the computational aspect of FLC require a simplification of the fuzzy control algorithm. Therefore, Mamdani's R_c and Larsen's R_p with the connective "also" as the union operator " $\vee$ " seems to be more suitable for constructing fuzzy models.

(3) *Compositional operators*

Four kinds of compositional operators have been reported which can be used in the compositional rule of inference:

sup-min operation (Zadeh,[36])

sup-product operation (Kaufmann,[37])

sup-bounded-product operation (Mizumoto,[38])

sup-drastic-product operation (Mizumoto,[38])

In practical FLC applications, the sup-min and sup-product compositional operators are the most frequently used owing to their simplicity of computation.

(4) *Inference Mechanisms*

In an FLC, the consequence of a rule is not applied to the antecedent of another. This means that there is no chaining inference mechanism involved, and that the control actions are simply based on one-level forward data-driven inference (GMP). Combining all the implication functions, definitions and relations mentioned in this chapter, the inference mechanism can be explained by the following four types of fuzzy reasoning method which are currently employed in FLC application.

(a) Fuzzy Reasoning of the First Type - Mamdani's Minimum Operation Rule as a Fuzzy Implication Function:

This type is the most popular reasoning method, and was first introduced by Mamdani. Let us consider the following general form of MISO fuzzy control rules for the case of two-input/single-output fuzzy systems:

$$
\begin{array}{ll}
R_1: & \text{if } x_1 \text{ is } A_{11} \text{ and } x_2 \text{ is } A_{12} \text{ then } y \text{ is } B_1 \\
 & \text{or} \\
R_2: & \text{if } x_1 \text{ is } A_{21} \text{ and } x_2 \text{ is } A_{22} \text{ then } y \text{ is } B_2 \\
 & \text{or} \\
\vdots & \vdots \\
R_i: & \text{if } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \text{ then } y \text{ is } B_i \\
\vdots & \vdots \\
R_n: & \text{if } x_1 \text{ is } A_{n1} \text{ and } x_2 \text{ is } A_{n2} \text{ then } y \text{ is } B_n
\end{array}
$$

If the universe of discourse of x_1 ,x_2 and y are represented by X_1, X_2 and Y. Then, using equation (2.1) and (2.3), the i's control rule can be expressed by a fuzzy relation matrix:

$$R_i = (A_{i1} \times A_{i2}) \times B_i \qquad (2.5)$$

This system consists of n rules, $R_1, R_2, \ldots, R_n$, therefore, we may combine them by " $\cup$ " as connective "also" which has been discussed in section 2.3.5(2). Equation (2.5) becomes

$$R = R_1 \cup R_2 \cup \cdots \cup R_n = \bigcup_{i=1}^{n} R_i \qquad (2.6)$$

When the inputs are A_{01} and A_{02} and the output is denoted by B_0, we will have the following operation:

$$B_0 = R \circ (A_{01} \times A_{02}) \qquad (2.7)$$

or

$$B_0(y) = \max_{x_1, x_2} [R(x_1, x_2, y) \wedge A_{01}(x_1) \wedge A_{02}(x_2)] \qquad (2.8)$$

Normally, the control inputs are crisp values, which are called fuzzy singletons. In this case, we can replace the antecedent variables x_1, x_2 by x_{01} and x_{02}, thus

$$A_{01}(x_1) = \begin{cases} 1 & x_1 = x_{01} \\ 0 & x_1 \neq x_{01} \end{cases} \qquad (2.9)$$

$$A_{02}(x_2) = \begin{cases} 1 & x_2 = x_{02} \\ 0 & x_2 \neq x_{02} \end{cases} \qquad (2.10)$$

and equation (2.7) become

$$B_0(y) = R(x_{01}, x_{02}, y) \qquad (2.11)$$

combining (2.5) and (2.6), the right hand side of (2.11) will become

$$R(x_{01}, x_{02}, y) = R_1(x_{01}, x_{02}, y) \vee R_2(x_{01}, x_{02}, y) \vee \cdots \vee R_n(x_{01}, x_{02}, y) \qquad (2.12)$$

$$R_i(x_{01}, x_{02}, y) = A_{i1}(x_{01}) \wedge A_{i2}(x_{02}) \wedge B_i(y) \quad (2.13)$$

if we define

$$\omega_i = A_{i1}(x_{01}) \wedge A_{i2}(x_{02}) \quad (2.14)$$

then (2.11) will becomes:

$$B_0(y) = \bigvee_{i=1}^{n} [\omega_i \wedge B_i(y)] \quad (2.15)$$

Now, if we use the center of area method (COA) to get the defuzzification output, then we will have a crisp output value. As described in section 2.3.2(c) the defuzzification formula is

$$y_0 = \frac{\int B_0(y) \cdot y dy}{\int B_0(y)\, dy} \quad (2.16)$$

The following figure shows the reasoning method of this type.

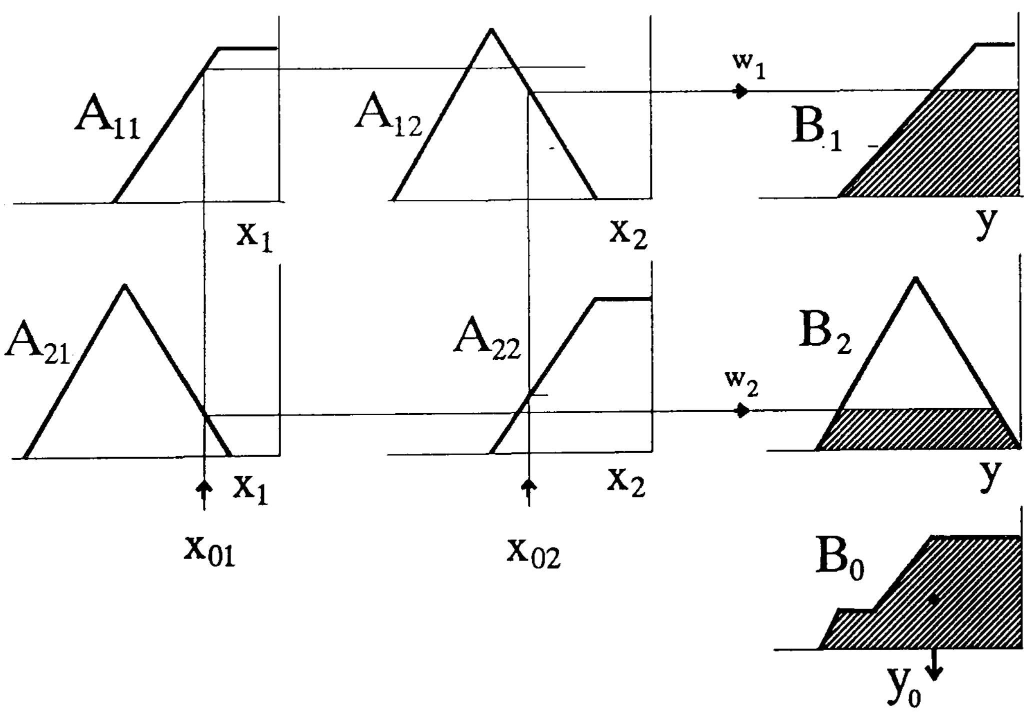


Figure 2-8. Diagrammatic representation of fuzzy reasoning of type 1.

(b) Fuzzy Reasoning of the Second Type - Larsen's Product Operation Rule as a Fuzzy Implication Function.

Fuzzy reasoning of the second type is based on the use of Larsen's product operation rule R_p as a fuzzy implication function. In this case, the ith rule leads to the control decision

$$\omega_i = A_{i1}(x_{01})A_{i2}(x_{02}) \qquad (2.17)$$

consequently, the result of reasoning will become

$$B_0(y) = \bigvee_{i=1}^{n} \omega_i B_i(y) \qquad (2.18)$$

The fuzzy reasoning process is illustrated in Fig.2-9 as shown below

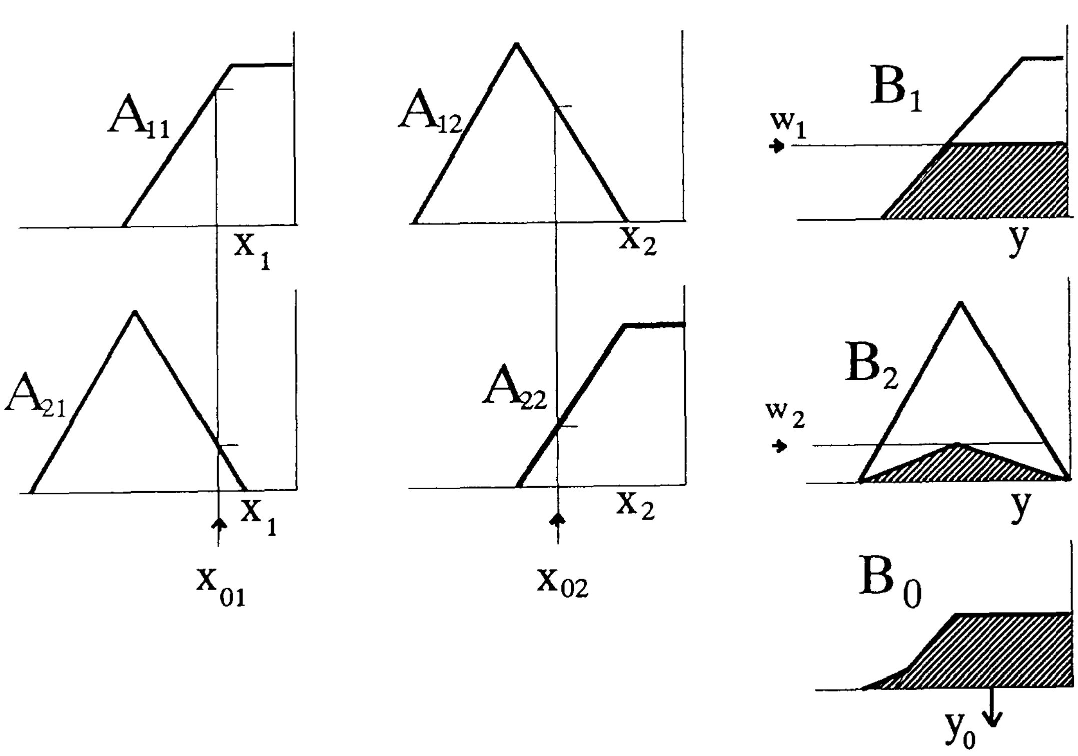
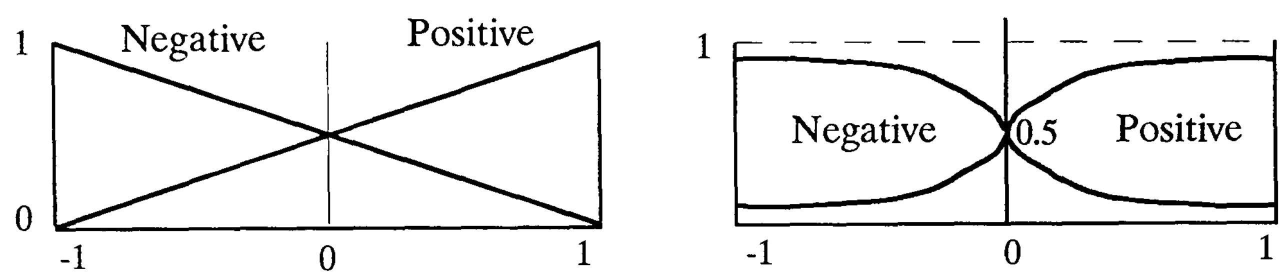


Figure 2-9. Diagrammatic representation of fuzzy reasoning of type 2.

(c) <u>Fuzzy Reasoning of the Third Type - Tsukamoto's Method with Linguistic Terms as Monotonic Membership Functions:</u>

This method was proposed by Tsukamoto[39]. It is a simplified method based on the fuzzy reasoning of the first type in which the membership functions of fuzzy sets X_1, X_2 and Y are monotonic, as shown in Figure 2-10(a) and 2-10(b).



(a) (b)

Figure 2-10(a) monotonic linear type fuzzy membership function.
(b) monotonic arctan type membership function.

The P(Positive), N(Negative) fuzzy variables can be described as

$$P(x) = \frac{1}{2}x + \frac{1}{2} \tag{2.19}$$

$$N(x) = P(-x) \tag{2.20}$$

for Figure 2-10(a), the monotonic linear type variables. and

$$P(x) = \frac{1}{\pi}\tan^{-1}ax + \frac{1}{2} \tag{2.21}$$

$$N(x) = P(-x)$$

for Figure 2-10(b) the arctan type variables.

If using Tsukamoto's method, the result inferred from the first rule is ω_1 such that $\omega_1 = N(y_1)$,and the result interred from the second rule is ω_2 such that $\omega_2 = P(y_2)$,then, a crisp control action may be expressed by equation (2.22), the height method, which has been described in section 2.3.2(e). This type of fuzzy reasoning is illustrated in Figure 2-11.

$$y_0 = \frac{\omega_1 y_1 + \omega_2 y_2}{\omega_1 + \omega_2} \qquad (2.22)$$

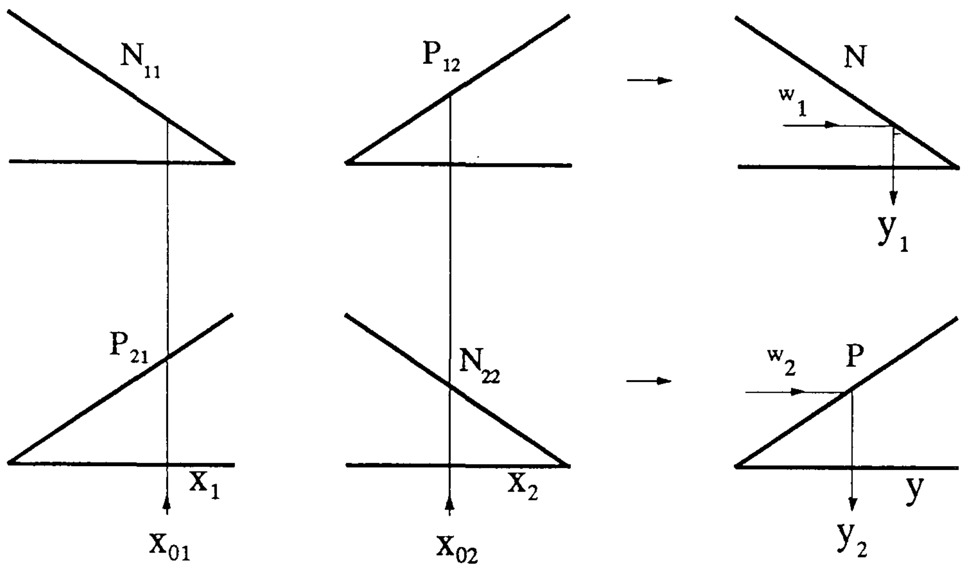


Figure 2-11. Diagrammatic representation of fuzzy reasoning type 3.

(d) Fuzzy Reasoning of the Fourth Type - The Consequence of a Rule is a Function of Input Variables.

Fuzzy reasoning of the fourth type employs a modified version of the state evaluation function. In this mode of reasoning, the ith fuzzy control rule is of the form

$$R_i : \text{if } (x_1 \text{ is } A_{i1},\ldots,x_n \text{ is } A_{in}) \text{ then } y = f_i(x_1,\ldots,x_n)$$

For simplicity, assume that we have two fuzzy control rules as follows

R_1 : if (x_1 is A_{11},x_2 is A_{12}) then $y=f_1(x_1,x_2)$

R_2 : if (x_1 is A_{21},x_2 is A_{22}) then $y=f_2(x_1,x_2)$

here, we may consider both f_1 and f_2 as linear equations of the form

$$y = ax_1 + bx_2 + c$$

although these are not usual in many applications. If the input variables are x_{01} and x_{02}, then the inferred value of the first and second rules will become

$$\omega_1 = A_{11}(x_{01})A_{12}(x_{02})$$

$$\omega_2 = A_{21}(x_{01})A_{22}(x_{02})$$

Thus, applying these two values, we can calculate the consequence result directly as

$$y_1 = f_1(x_{01},x_{02})$$

$$y_2 = f_2(x_{01},x_{02})$$

Finally, using the COA defuzzification formula, a crisp control action is given by

$$y_0 = \frac{\omega_1 f_1(x_{01},x_{02}) + \omega_2 f_2(x_{01},x_{02})}{\omega_1 + \omega_2} \tag{2.23}$$

This method was proposed by Takagi and Sugeno and has been applied to guide a model car smoothly along a crank-shaped track[26] and to park a car in a garage [27]. However, when the number of the input variables which representing the system states increases, the number of rules will increase also. Usually, it will be equal to the number of input space partitioned. Figure 2-12. shows an example of the fuzzy rules used when two input variables are involved, and Figure 2-13 shows the corresponding partitioning of the input space. Note that, the break points of the membership functions for the input variable x_1 space are 3, 9, 11 and 18; for x_2 spaces are 4 and 13. The shaded area are those membership function overlapping zones.

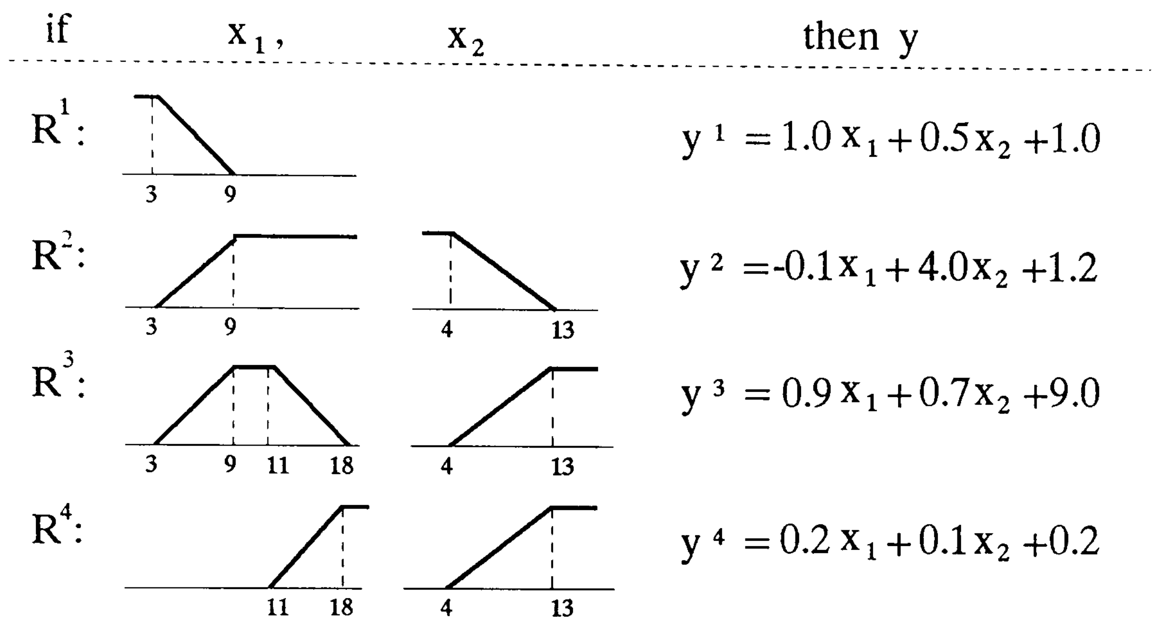


Figure 2-12. The fuzzy rules for reasoning of type 4.

Deciding on an appropriate method of partitioning is a serious problem from the fuzzy system identification point of view[40,41].

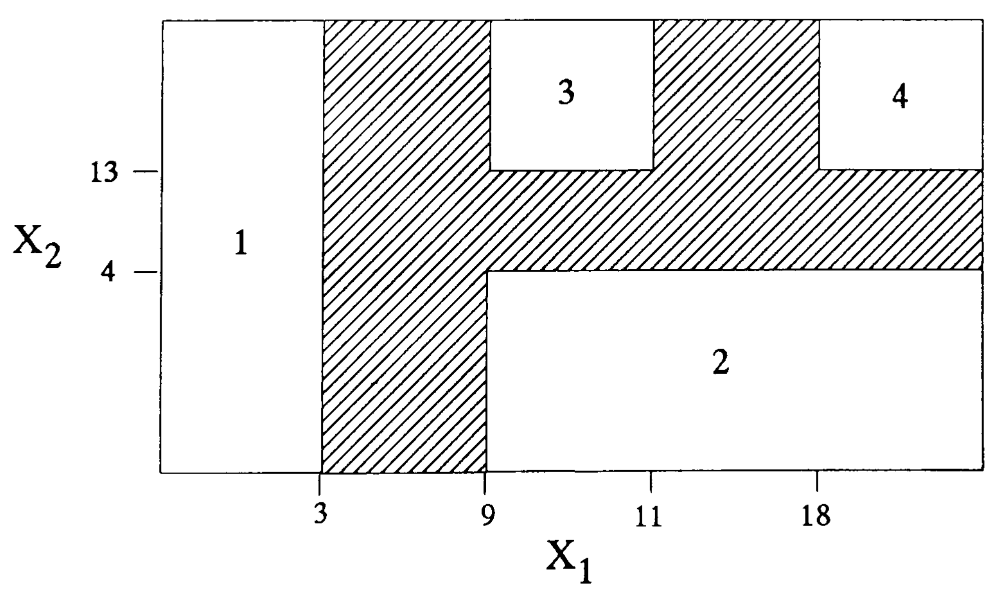


Figure 2-13. Partition of the fuzzy input space.

2.4 Current uses of Fuzzy Techniques and other Non-linear methods in Control of the pH Process

As mentioned in Chapter 1, control of the pH process is difficult due to the non-linearity and time varying nature of the titration curve, and because of its strong sensitivity to disturbances near the point of neutrality. For these reasons, over the years several new technologies have been applied in an attempt to solve these problems. In this section, several different approaches, including the use of the fuzzy logic are listed chronologically and briefly discussed.

2.4.1 The Digital Parameter Adaptive Control Technique (1980)

Bergmann & Lachmann's paper[3] published in 1980, describes the application of two digital parameter adaptive controllers to a pH process. The controllers use combinations of recursive least squares (RLS) parameter estimation methods with a extended minimal variance controller and a deadbeat controller of increased order. These two adaptive control algorithms have been applied to control a pH pilot plant. Experimental results are promising. However, selecting appropiate parameters to achieve good stability is very difficult.

2.4.2 Dynamic Modeling and Reaction Invariant Control of pH(1983)

In 1983, Gusstafsson and Waller presented a paper[5] on a general and systematic approach to the derivation of dynamic models for fast acid-base reactions. The treatment is based on a consistent use of chemical reaction invariants and variants, the former describing the physical properties of the reactor system independent of chemical reactions, the later describing the chemical reactions. For fast acid-base reactions the invariant part of the model is sufficient to define the thermodynamic state of the system.

The reaction rate vector is thus eliminated from the model. The reaction variant part of the model then, use a static equation to relate pH to the reaction invariant state variables. The model obtained provides a sound basis for the design of control loops for a pH control process. A computer simulation example is given at the end of this paper.

2.4.3 An Experimental Study of a Class of Algorithms for Adaptive pH Control (1985)

This paper, which was published by Gustafsson in 1985[4], describes models suited to the design of controllers of pH in fast acid-base reaction processes with varying buffer concentrations. The set of models includes a non-linear model resulting in a controller with linearized feedback of a reaction invariant state vector and linearized models resulting in adaptive linear feedback of pH. Experiments showed that the reaction invariant feedback control needs a *priori* knowledge of the reaction invariant dynamics of the process, which includes the time delay n and the model orders n_A and n_B. These models can be obtained by on-line identification performed by a self-tuning regulator. However, to ensure a correct result when identifying a process in closed-loop, feedback polynomials of a high enough order and extra perturbing inputs are necessary. In this paper the emphasis is put only on the adaptive properties of the controllers.

2.4.4 Non-linear State Feedback Synthesis for pH Control (1986)

In this paper, presented by Wright and Kravaris[7], a novel non-linear state feedback design methodology is described. This is based on a non-linear transformation of the output that linearizes the state/output map. A linear PID control law for the transformed output generates a non-linear control law for the pH process. Computer simulations were used to compare the performance of this non-linear control law to classical linear PID methods. Results showed that transformed control, with a minimum settling time of approximately 10 controller actions, is clearly superior than the classical contro at all sampling periods.

2.4.5 Non-linear Controller For a pH Process (1990)

Jayadeva and Rao, et al [42] synthesized a non-linear feedback-feedforward controller in which a model reference non-linear controller design technique is applied to control a pH process. The performance of this non-linear controller is evaluated by simulation both for regulatory and servo problems. The simulation result is very good. This control system is also shown to be robust to significant parameter variations and to small disturbances.

2.4.6 Non-linear Control of pH Process Using the Strong acid Equivalents (1991)

A novel approach was developed by Wright and Kravaris[43] for the design of nonlinear controllers for pH processes. It consists of defining an alternative equivalent control objective which is linear in the states and using a linear nonadaptive control law in terms of this new control objective. The new control objective is interpreted physically as the strong acid equivalent of the system.

A minimal order realization of the full-order model has been rigorously derived, in which the titration curve of the inlet stream appears explicitly. The strong acid equivalent is the state in the reduced model which can be calculated on line from pH measurements given a nominal titration curve of the process stream. However, this requires the use of additional hardware, such as an automatic titrator or ion-selective eletrodes to obtain a direct or indirect measurement of the strong acid equivalent value. Computer simulations have been used to evaluate the performance of this new methodology.

2.4.7 Fuzzy Control of pH Using Genetic Algorithms (1993)

Researchers at the U.S. Bureau of Mines have developed a new technique for producing adaptive FLCs that are capable of effectively managing systems such as the pH process. In this technique, a genetic algorithm(GA) is used to adjust the membership functions employed by a conventional FLC. GA's are search algorithms based on the mechanics of natural genetics that can rapidly locate near-optimal solutions to difficult problems. Karr and Gentry reported [44] that they used this technique to produce an adaptive GA FLC for a laboratory acid-base system. In the experiment, they change the process dynamics by altering system parameters such as the desired set point and the concentration and buffering capacity of the input solution. The experiment demonstrated the effectiveness of this approach in developing an adaptive FLC for handling those nonlinear systems such as those associated with pH control.

2.4.8 Summary

These papers are some of the recent technologies which have been applied in an attempt to solve the pH control problems. Only three of them[3][4][44] have tested their controllers on a real process. The latest one [44] is the first to try using the fuzzy logic control techniques in pH control process. This technique offers a powerful alternative to conventional process control techniques in non-linear, rapidly changing pH systems. However, their experiments were performed in a $1l$ beaker with a magnetic stirring bar in it, not a continuous stirred reaction tank (CSTR). Therefore, further study is required in the control of pH in a continuous flowing system.

Chapter 3

Implementation of a Real-Time Fuzzy Controller

A fuzzy controller is characterized by its ability to provide an algorithm which can convert a linguistic control strategy, based on expert knowledge, into an automatic control strategy, without having a precise mathematical model of the target plant. Lacking a knowledge of the system model, it is very difficult to employ the systematic design techniques associated with modern control methods, in the design of a fuzzy logic control system. Usually, trial and error are necessary components of all FLC applications. In the last chapter we discussed the design parameters of a fuzzy controller. In t is chapter, we look at current FLC implementation methods and consider the construction of an FLC for experimental use.

3.1 Industrial Application of FLC Techniques

It was noted earlier that the Fuzzy inference engine, or decision making logic, is merely an algorithm. This can be implemented in software within a personal computer or in hardware using a fuzzy microchips. When using a general purpose fuzzy inference engine, different control strategies can be achieved by simply changing the contents of the stored knowledge base.

The first fuzzy logic chip was designed by Togai and Watanabe at AT&T Bell Laboratories in 1985[46]. The fuzzy inference chip, which can process 16 rules in parallel, consists of four major parts: a rule-set memory; an inference processing unit; a controller; and an input/output unit.

Recently, in order to allow dynamic changes in the rule set, the rule-set memory has been implemented by a static random access memory (SRAM). Many new fuzzy microchips are being introduced each year, but these show little variation in the basic techniques used. The basic operation of these devices will be discussed later.

3.1.1 The Structure of a Fuzzy Control System

Usually, an industrial control system is composed of four main parts, as shown in Figure 3-1.

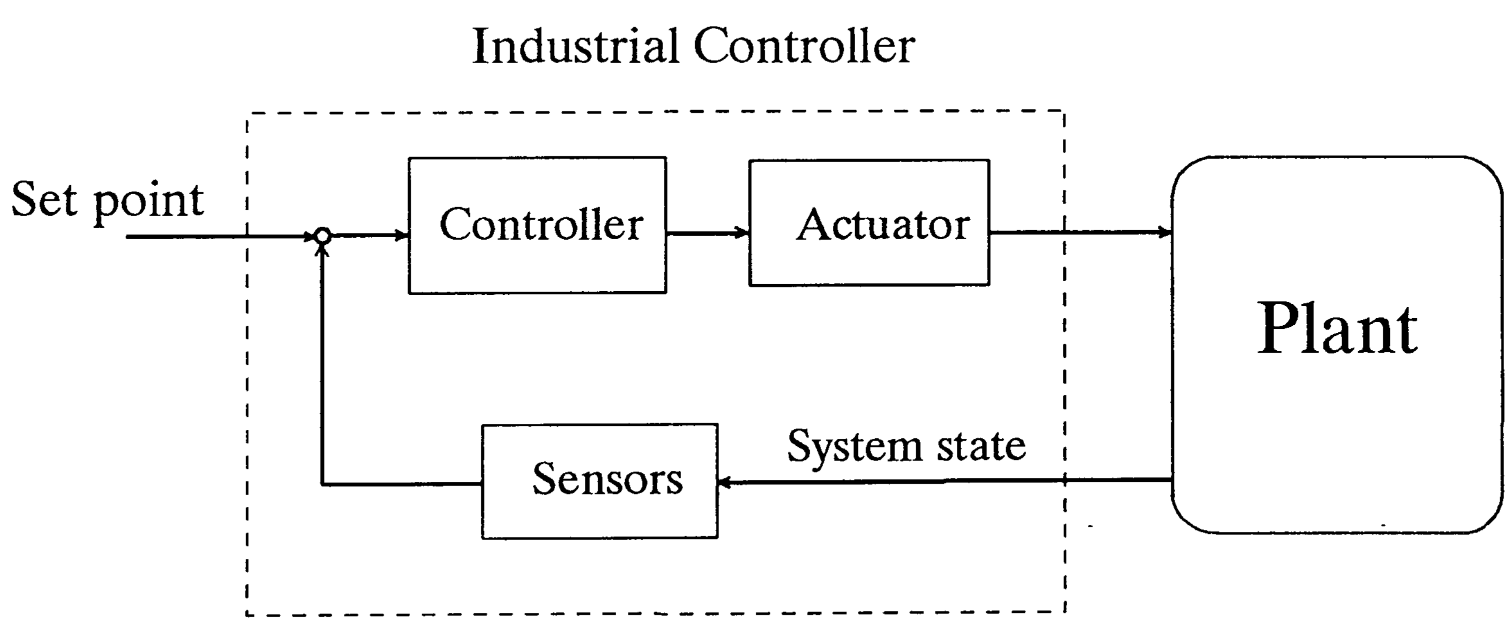


Figure 3-1 A block diagram of a typical industrial control system.

The state of the plant is first sensed by sensors in the form of physically measurable quantities. Then, after transforming to some appropriate signal, they are sent to the controller. The controller, after comparing the signals, which represents the state variables of the plant, with the previously decided value of set point, will adjust the output to the actuator. Finally, the plant will respond to the new driving force delivered by the actuator.

In fact, a fuzzy controller plays the same role in an industrial control system as classical controller. A fuzzy control system has the same structure as in Figure 3-1, the only difference is that the controller becomes an FLC.

FLC's will have different forms depending on the application, requirements and environment. Commonly used types are

(1) *Fuzzy controllers based on single-chip microcontrollers*

In, consumer products, such as washing machines and other electrical equipments, cost will be an important consideration. When fuzzy control is required, such system often use conventional 4-bit or 8-bit microcontrollers to form an FLC.

(2) *Fuzzy controllers based on personal computers*

If a fuzzy controller is formed by constructing a fuzzy knowledge base in a personal computer, then, the controller can utilize all the hardware and software resources of the PC to improve its performance. For instance, a data base can improve its data searching and handling ability, and process monitoring software can help it to perform intelligent diagnosis and decision making.

(3) *Fuzzy controllers based on programmable logic controllers(PLC).*

PLC s are sequential controllers that are widely used in industry. Some PLC manufacturers such as OMRON have also developed fuzzy controller modules which can be installed in a PLC to perform the FLC functions, thus directly upgrading a PLC into an FLC.

3.1.2 Technical Considerations When Developing a Fuzzy Control System

There are several technical issues that must be considered when developing a product that incorporates fuzzy control.

(1) *general purpose FLC*

A general purpose FLC can meet the requirements of a wide range of products without changes to its hardware structure. If such an FLC could be produced, all the development and maintenance costs would be reduced significantly. Therefore, the development, or using such an FLC is an important consideration for any FLC application.

(2) *The FLC development system*

A powerful fuzzy development system can help the FLC designer to build a useful fuzzy knowledge base quickly and shorten the time required for product research and development (R & D). Usually, such a fuzzy development system should have the following characteristics:

(a) It should make it easy to construct, edit and adjust the fuzzy knowledge base.

(b) It should be capable of displaying the control system input and output relationships within a control surface chart, and be ale to performing system model simulation.

(c) It should be very easy to implant it into all kinds of controller.

(3) *Self-learning and adjustment of the fuzzy knowledge base*

The fuzzy knowledge base is the kernel of a fuzzy controller, and this can be built from the knowledge of an expert or the experience of an operator. As the knowledge base increases in complexity, more and more time will be required to adjust it through a process of try and error. Therefore, in complex system, self-learning techniques and adjustment strategies become very important. This area is currently the subject of much research.

(4) *Performance of the fuzzy inference engine*

Another important part of an FLC is the inference engine. The inference speed will greatly affect system performance. Usually, the speed, or efficiency, of an inference engine is expressed in FLIPS (Fuzzy logical inferences per second). The speed of a particular system is likely to be greatly influenced by the methods used to implement the inference engine with arrangements based on VLSI techniques being much faster then those that are software based. Commercial products normally use low-end devices with speeds of several thousand FLIPS. More demanding applications, such as those in the aerospace or military section, often use high-end devices with speeds of about a quarter of a million FLIPS. Software inference engines have speeds of only a few hundred FLIPS, but are sufficiently fast for many industrial applications. Software-based arrangements offer considerable flexibility making them a very good choice for many FLC applications.

3.2 FLC Implementation methods

FLC design strategies and procedures have already been discussed in earlier sections of this chapter and in chapter two. However, when the target system has been investigated and the fuzzy knowledge base (FKB) has been created, it is necessary to consider the choice of implementation techniques. Implementation here refers to the process of implanting a FKB into a computer or some special controller. In other words, through implementation we convert the FKB into an executable form.

The strategies and techniques mentioned earlier directly affect the implementation of the FLC. However, as discussed in chapter 2, a fuzzy inference engine that is suitable for fuzzy control is composed of three parts: a fuzzification interface, a forward-driven fuzzy inference engine and a defuzzification interface. Usually, the fuzzy inference engine has a fixed operation which will not change once it is implemented. Therefore, the main task in designing an FLC is the design of an FKB.

There are two methods commonly used in the production of fuzzy knowledge bases... the FKB compilation method and the FKB interpretation method.

3.2.1 Implementation Using the FKB Compilation Method.

The FKB compilation method is a technique widely used in the early days of fuzzy control. It is an off-line implementation method, where the FKB is transformed into a data format that can be used to perform the fuzzy control operation. The following three forms are used most commonly.

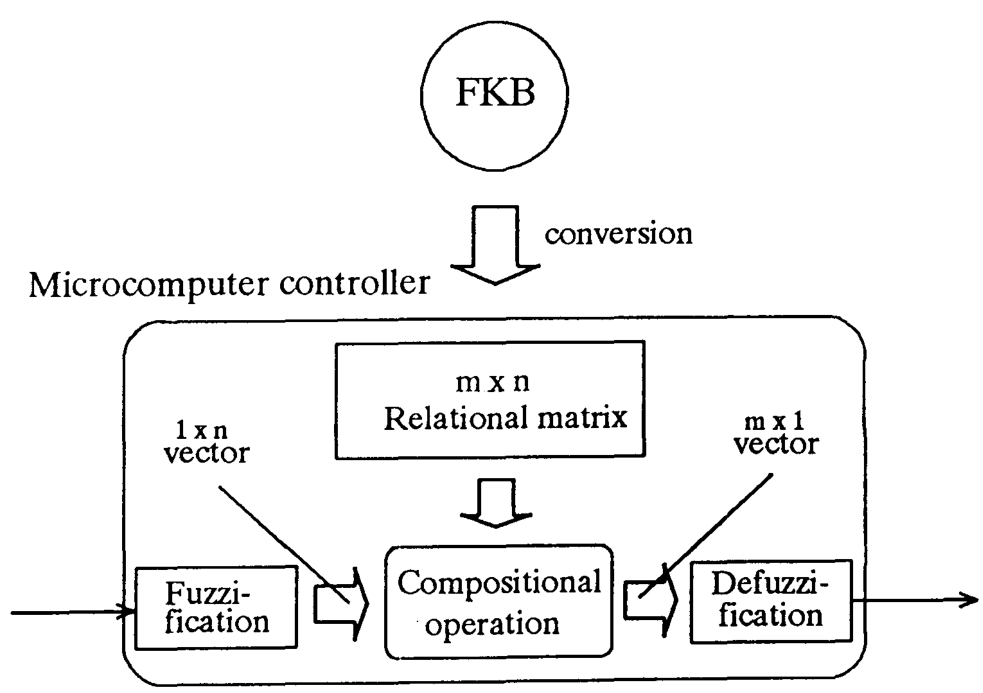


Figure 3-2. The block diagram of a FKB relation matrix implementation method.

(1) *To Convert the FKB into a Fuzzy Relation Matrix [47]*

The fuzzy control rules in a fuzzy control system may be characterized by the IF-THEN construct. From a mathematical point of view, these rules represents the relationships between the inputs and outputs. Therefore, if the variables of these linguistic rules can be discretized by applying some proper implication operation, the control rules can be converted into a discrete fuzzy relation matrix. After combining all the relation matrices of each rule, a numerical matrix is found which represents the control system FKB and can be loaded into a computer memory. This kind of fuzzy control system is shown in Figure 3-2, where the input signal is fuzzified and converted into a vector $(1\times n)$.Then, this vector will be used in the composition operation with the relation matrix $(m\times n)$and to produce a result vector $(m\times 1)$.Finally, this output vector will be defuzzified and become a crisp value before it is sent to the plant.

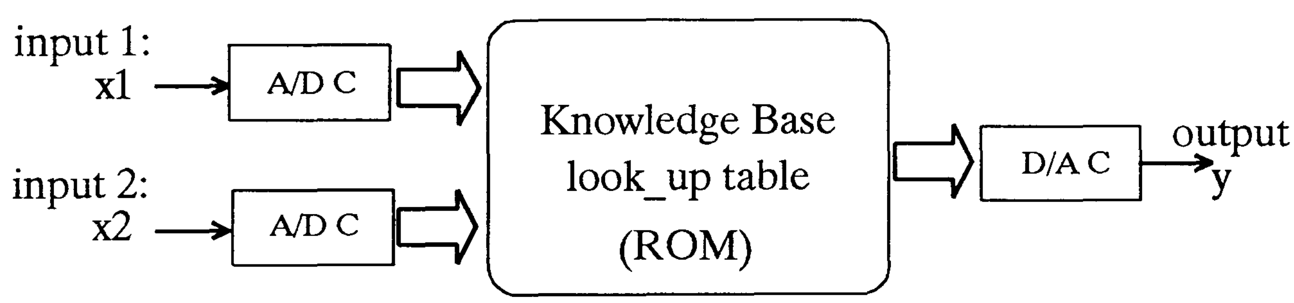


Figure 3-3. method of Implementation using look-up table.

(2) *Implementation by Look-up Table [48]*

While the use of a relation matrix is straight forward, it takes a long time to perform the matrix composition operations. In many real-time application this is unacceptable and a faster method is required. One approach is to compute all the possible states of the relation matrix off-time, and to store theses in a look-up table as shown in Figure 3-3.

In Figure 3-3, input signals x1 and x2 are digitized by the A/D converters and form the address of a ROM, which contains the input-output look-up table for the FKB. Resultant output is then applied to a D/A converter.

(3) *Implementation by compiling an FKB into executable real-time code.*

We have seen that the fuzzy control rules in a fuzzy control system are expressed by IF-THEN statements. Clearly, similar IF-THEN expressions are found in the ordinary computer language; the difference being that fuzzy control rules use linguistic fuzzy concepts to determine their action, while conventional programming languages make decisions based on crisp values.

Therefore, if the fuzzy terms of a linguistic variable are well defined by its membership function, then, it should be possible to convert the fuzzy control rules into the form of an ordinary program. This concept was first presented by Togai[46]. He developed a Fuzzy C Compiler which can convert an FKB into IF-THEN program statements in the form of ANSI C. Then, using an ordinary C compiler (such as Microsoft C or Borland C) it can be compiled to form executable code. This is shown in Figure 3-4.

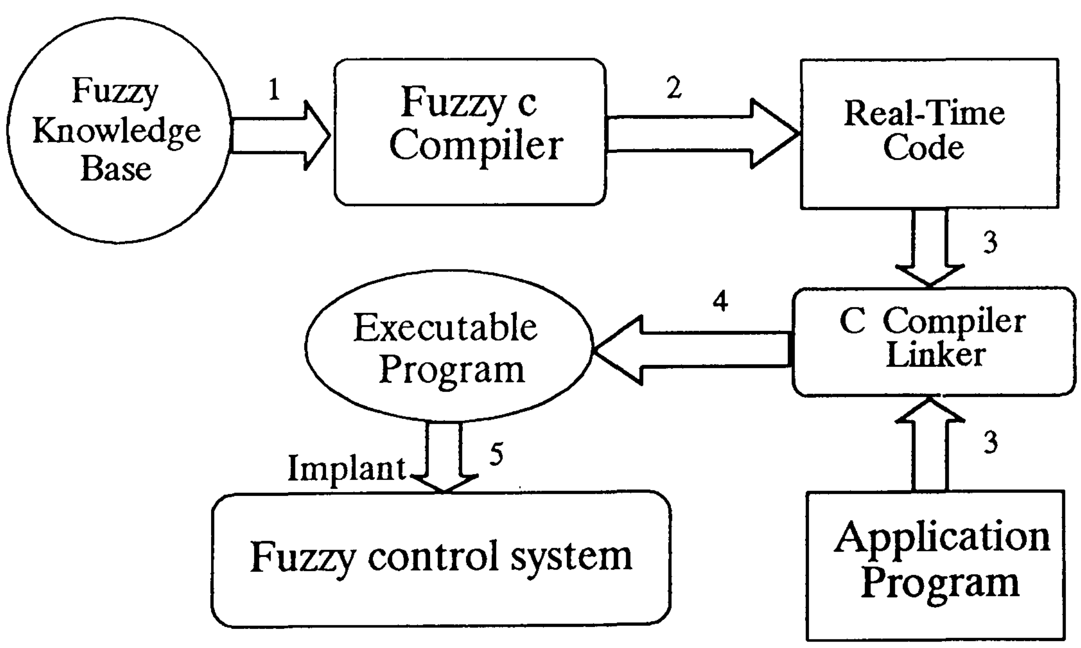


Figure 3-4 Implementation using FKB real-time code.

3.2.2 Implementation using the FKB Interpretation Method

FKB interpretation is a method in which the fuzzy inference engine and the FKB are separated. An independent fuzzy inference engine is installed in the fuzzy controller to do the real-time inference work on the FKB. This is shown in Figure 3-5. This kind of fuzzy control system need not apply any FKB format transformation techniques, and the inference engine works inside the controller. Therefore, sometimes this approach is called a real-time, on-line implementation.

As mentioned earlier, fuzzy inference is normally fixed operation and there are several ways to perform it. For instance, it can be done using a software inference program [49], special inference chips [46][50] or firmware within a microcontroller. Examples of inference engine implementation methods are given below.

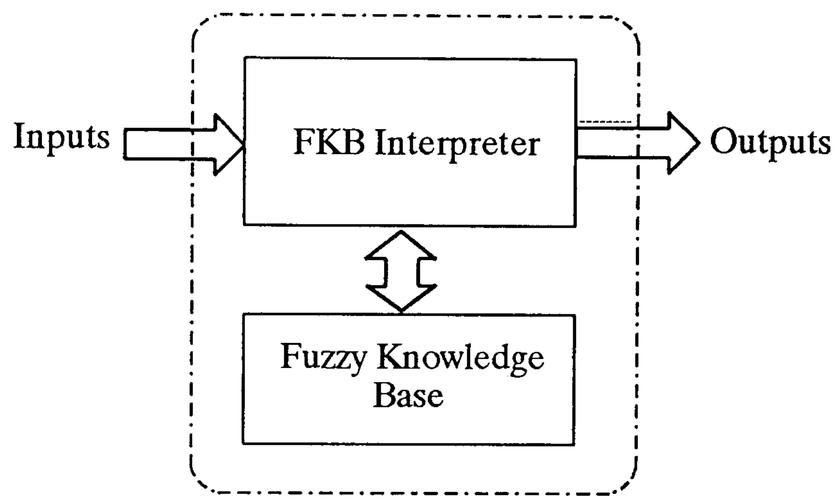


Figure 3-5 The basic configuration of an FKB interpreter.

(1) *A general purpose inference program working with a separate FKB [49]*

First, using the program design techniques described above, the FKB is organized into an efficient data structure. Then, a general purpose real-time function is designed to represent the fuzzy inference operation. As shown in Figure 3-6, the action of the fuzzy controller is written as a program and stored in the computer's memory. The main program is responsible for calling the fuzzy inference function to process incoming data.

(2) *Implementation using an independent fuzzy microcontroller*

Several commercial fuzzy microcontroller (FMC) are now available. An example is that developed by NeuraLogix [50]. This general purpose inference engine (NLX-230) contains a neural network inside the chips to improve its performance. The rule base and membership functions are written in the mask programmed ROM. The chip can be operated independently or as a slave device. Its performance is equivalent to an 8-bit microcontroller performing the inference of 30 million rules per second.

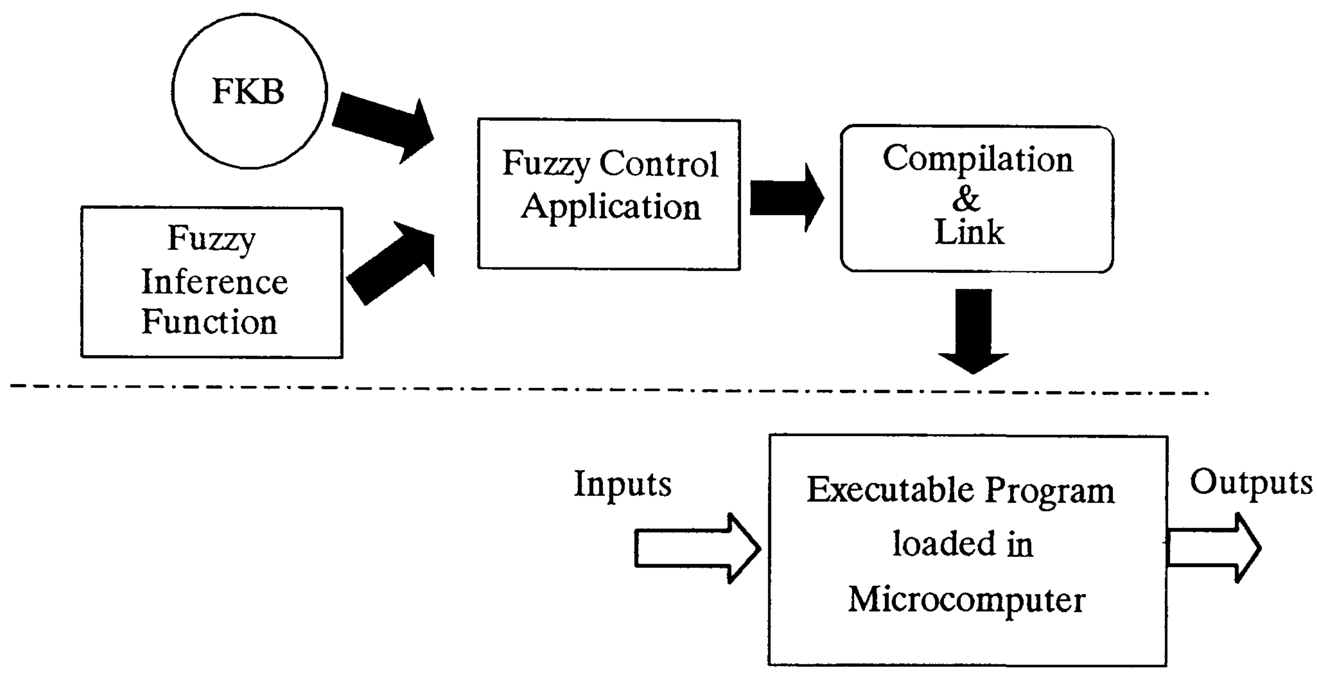


Figure 3-6 Implementation by real-time inference program.

3.2.3 A Comparison of Implementation Methods

The main feature of the FKB compilation method is that the inference engine and the FKB are put together to form a simple structure for implementation. However, there are several disadvantages with this approach. For instance: it will be very difficult to construct a multi-input, multi-output (MIMO) system [47]; the memory size and the number of memory accesses increase exponentially as the number of rules is increased [48]; the length of the executable code is directly related to the number of control rules [51]; and it is very difficult to perform on-line tuning.

The FKB interpretation method, in contrast, produces a separate inference engine and FKB. This kind of structure is more like the human brain; the inference engine resembles the unchanging structure of approximate reasoning, while the fuzzy knowledge base resembles human experience which can be changed or modified through learning. Therefore, it could be possible to develop the FKB on-line tuning and learning technique by using the FKB interpretation method.

For the FKB interpretation method, it seems that we need a general purpose fuzzy inference engine (FIE) to meet our requirement, and we also have to convert the FKB information to a specified data format which will satisfy this FIE. At present, a lot of successful FIE hardware chips have been reported[46][50]. These FIE hardware products are all characterized by their very high speed inference. However, they also have several structural deficiencies due to a compromise between designing complexity and product performance. For example: some have a limited number of input-output lines and a limited number of control rules; while others have only 4-bit membership function, with linguistic variables discretized to 64 levels. These characteristics result in a reduction of the FKB data resolution.

3.2.4 The Fuzzy Development System

The design of a fuzzy control system, usually, involves not only the construction and adjustment of the knowledge base , but also the task of transferring this to some operational hardware platform. As mentioned in section 3.1.2, a powerful fuzzy development system can greatly reduce the problems encountered during the designing stage.

There are several fuzzy development systems in use. Most of them use the FKB compilation method of implementation to generate the real-time C code for some special microprocessor. Some of these are listed below;

(a) TILShell- a fuzzy development system produced by Togai InfraLogic co., This operates within the Windows environment and has the following functions: to define and adjust membership functions; to construct or change a rule base; to perform static or dynamic system debugging and simulation; and to generate executable code in C for a FKB.

TILShell 3.0 uses a special Fuzzy Program Language (FPL) to define a fuzzy system, and also offers a graphic editor.

For supporting single chip microcontrollers, this company offers another product called the Microcontroller development package. It provides the feature of TILShell and can be used to convert the FKB to assembler code of a range of microcontrollers. At present, this package supports 8051, 68Hc11 and Hitachi H8/300, H8/500.

(b)FIDE - The Fuzzy Inference Development Environment is a product of Aptronix USA, which is composed of four parts: Editor, Debugger, Composer and RTC (Real Time Code). It uses a special Fuzzy Inference Language (FIL) to describe the target's fuzzy system. The RTC is used to generate the machine code for certain microprocessors. At present, FIDE support a range of Motorala devices, and is capable of generating machine code for MC6805, MC68H05,MC68HC11, MC68HC08, MC68HC16 and MC68342.

(c)FLD - The Fuzzy Logic Designer is a fuzzy development system designed by Byte Dynamics co.,USA. It is a low cost, user friendly FDS. It provides a very good 2D/3D display for the control surface.

Although FLD will generate an ANSI C source program for the FKB, it does not provide any interface for the application systems. Therefore, linking its FKB with a particular microcontroller may be a problem.

(d)FST - The Fuzzy System Toolbox is an application specific routine for use with MATLAB®. MATLAB® (derived from matrix laboratory) is a product of The MathWorks, Inc., which is a technical computing environment whose basic data element is a self-dimensioning matrix. It combines fast numerical capabilities with excellent graphics useful for developing and modifying algorithms, particularly those which depend heavily on matrix operations.

The Fuzzy System Toolbox(FST) by PWS Publishing Company is a useful tool for creating and and using fuzzy inference systems. This toolbox provides a command line approach to building fuzzy sets and fuzzy rule-based systems. It also provides a gradient descent approach to learning which is useful for the beginner. Several effective demonstration applications are included for pattern recognition, control, model fitting, and decision making.

(e)FLT - The Fuzzy Logic Toolbox is also a product of The Mathworks, Inc. The major difference between FLT and FST is the user interface. FST must be used from the command line, while FLT can be used through a graphical interface. This graphical interface grately simplifies the building and manipulation of fuzzy systems. The Takagi-Sugeno model of fuzzy inference is also provided in FLT. This allows quite sophisticated fuzzy models to be developed. In FLT the set of demonstration applications, which accessible through menu and button options, contains examples from control, time series, signal processing, and clustering applications. There is a procedure included in this toolbox which allows the user to build stand-alone code to do fuzzy inference. Therefore, FLT is very useful for the application and for the study of fuzzy control theories. However, this toolbox was not available when this research started.

3.3 Implementing an Experimental FLC

Although the work described in this thesis is of relevance to all forms of industrial pH control system, the experimental work was performed using a laboratory model of a pH plant. The relatively long time constants associated with this model (usually from 1 to 20 second) meant that the speed requirements of the control system were undemanding. It was therefore decided to adopt a software-based implementation of an FLC using a personal computer. This approach simplifies the task of formatting the data for the FKB and allows the construction of a user- friendly interface.

A 486-based PC can achieve a performance of about 300 FLIPS which is quite adequate for such an application. It was therefore decided to implement an experimental FLC using the FKB interpretation approach on such a PC.

Early attempts to design a fuzzy logic controller involved the setting up of a Fuzzy Data Processing System. This system could handle different kinds of fuzzy variables and perform the various fuzzy data operations needed for an FLC. This was followed by the construction of a simple fuzzy control program generator to generate different kinds of real-time FLC.

3.3.1 System Description

The main functions of the system include: (1) Fuzzy data base creation. (2) Fuzzy relation matrix manipulation. (3) Fuzzy logic inference. (4) Fuzzy data display. (5)

Controller program generation. Part (1),(2) and (3) are broadly based on the subroutines and files published by Mitzura and Tanaka[52]. Although these required considerable modification, because of differences in the data set and the algorithm used. An additional task was to write an interpreter for the complete system. This includes: (a) an interactive screen editor, for fuzzy data base entry; (b) a fuzzy data screen display program; and (c) an FLC program generator.

The main functions of this system are to perform the necessary procedures for recording, deleting and executing fuzzy logic operations on fuzzy data files which consist of: universal sets; fuzzy sets; fuzzy relation matrices and fuzzy IF-THEN rules.

(1) Data Sets.

There are 4 basic data sets in this system, each with user defined file name for use when recording and retrieving. Those sets are

- *Universal set*: The universal set is the universe of discourse of some fuzzy sets. Usually it is equally separated in the Real domain. The number of the elements in a universal set is defined by a constant MXEL in an included file in C. If the system is using the break points of membership functions to represent a continuous fuzzy set, then we have to store only the minimum and maximum value of the universal set to represent the domain of the universe of discourse.

- *Fuzzy set*: A fuzzy set X in a universe of discourse U is characterized by a membership function μ_X which takes values in the interval [0,1]. For the discrete fuzzy set case, we need to store all the μ_X and the corresponding points of the universal of discourse in memory. However, for the breakpoint representation, we have to specify only the membership values corresponding to each break points, and store it together with the breakpoint positions of the universe of discourse. An example of a continuous fuzzy set is shown in Figure 3-13, where the continuous fuzzy set can be represented by the seven breakpoints at -20, -18, -10, -2, 4, 12 and 20 of the universe of discourse. When written in the C language, the fuzzy set is represent by a structure which contains:

(i) the name of the universal set (char)

(ii) the name of the fuzzy set (char)

(iii) the number of breakpoints (int)

(iv) a data array of the value of the membership function at each breakpoint (float)

(v) a data array of the breakpoint positions in the universe of discourse

(vi) an address pointer for constructing a structure list.

Only those fuzzy sets with the same universal set can perform fuzzy set operations.

- *Fuzzy relation*: Binary relations are manipulated by matrix operations. The size of the matrix is defined by the universe of discourse appeared in the relation matrix row and column. Usually, this is used for discrete fuzzy sets, but not for the continuous fuzzy set operation.

- *Type 2 fuzzy set*: When the membership values of the elements of a fuzzy set are also a fuzzy set, it is called a type 2 fuzzy set (or simply T2 fuzzy set). In this system the universal set of a T2 fuzzy set membership value is also limited by the constant MXEL for the universal set.

- *Fuzzy IF-THEN Rules*: The following m IF-THEN rules are recorded under a file name:

R_1: if x_1 is A_{11} and ..... x_n is A_{1n} then y is B_1

or

R_2: if x_1 is A_{21} and ..... x_n is A_{2n} then y is B_2

or

: : :

: : :

R_m: if x_1 is A_{m1} and ..... x_n is A_{mn} then y is B_m

where $m \leq MXEL$, and $n \leq MXRC$ $(=10)$.n is the number of input variables defined in the system head file, which is also limited by a constant MXRC.

Wellcome to
Fuzzy theory system

○ Please input the command name and parameters,

Example: < Command label 1, label 2

○ Or command only:

Example: < MK_FS

○ Please type "END" to close the system.

Please hit any key to start !!

Figure 3-7 The opening menu for the fuzzy system.

(2) Data Entry and the Screen Editor

After entering the system, an opening menu as shown in Figure 3-7 will be displayed. Then, the system will start and respond with a < on the screen as a result of hitting any key on the key board. If we type a command such as <MK_FS↵ after the prompt,

the system will start the interactive program for the user to build a fuzzy set on the screen (MK_FS stands for "Making a Fuzzy Set). However, before trying to build any fuzzy set, we must construct its universal set first by using the command MK_UD (Making an Universe of Discourse). If the corresponding universal set already exists (stored as a record in an structured list with the name 'UDLIST'), the screen editor will find it and each time a point position is entered, the system will check to see if this point exceeds the maximum or minimum values of the universe of discourse. If it does, the screen will respond with a warning message and wait for correction. The screen editor for this command is shown below:

```
                              EDIT
                                                   command : MK_FS

        NAME OF THE FUZZY SET =                    Note: press ^q to quit;
    NAME OF THE UNIVERSAL SET =                          press ^s to save input,
                                                         and press ^z to ON/OFF
                                                         linearization.
                                                   ** Linearization = OFF **
     1( . )/    ,    11( . )/    ,    21( . )/    ,
     2( . )/    ,    12( . )/    ,    22( . )/    ,
     3( . )/    ,    13( . )/    ,    23( . )/    ,
     4( . )/    ,    14( . )/    ,    24( . )/    ,
     5( . )/    ,    15( . )/    ,    25( . )/    ,
     6( . )/    ,    16( . )/    ,    26( . )/    ,
     7( . )/    ,    17( . )/    ,    27( . )/    ,
     8( . )/    ,    18( . )/    ,    28( . )/    ,
     9( . )/    ,    19( . )/    ,    29( . )/    ,
    10( . )/    ,    20( . )/    ,    30( . )/    ,
                       INPUT OK? (Y/N)
```

Figure 3-8. screen editor for MK_FS command.

If the number of input elements exceeds the size of the frame allowed on screen, the contents of the screen will scroll to the left or right, up or down to cover the whole set. Some other examples are shown below

- MK_FM : command to construct the fuzzy binary relation matrix, see Figure 3-9.
- MK_FR : command to form the rule base for the controller. (see Figure 3-10.)

- INFR_D : command to perform fuzzy sets direct reasoning by chosen implication method. (see Figure 3-11.)

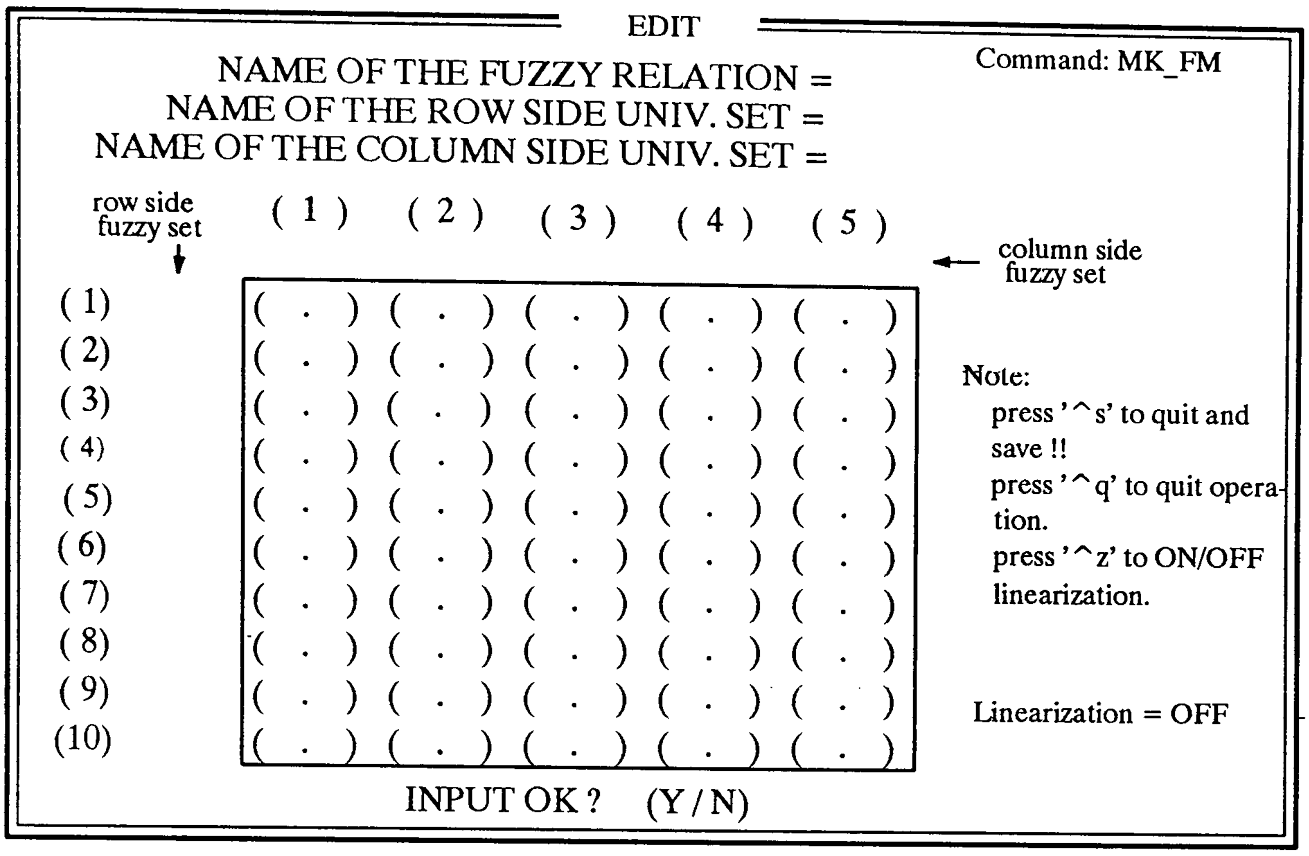


Figure 3-9. screen editor for command MK_FM.

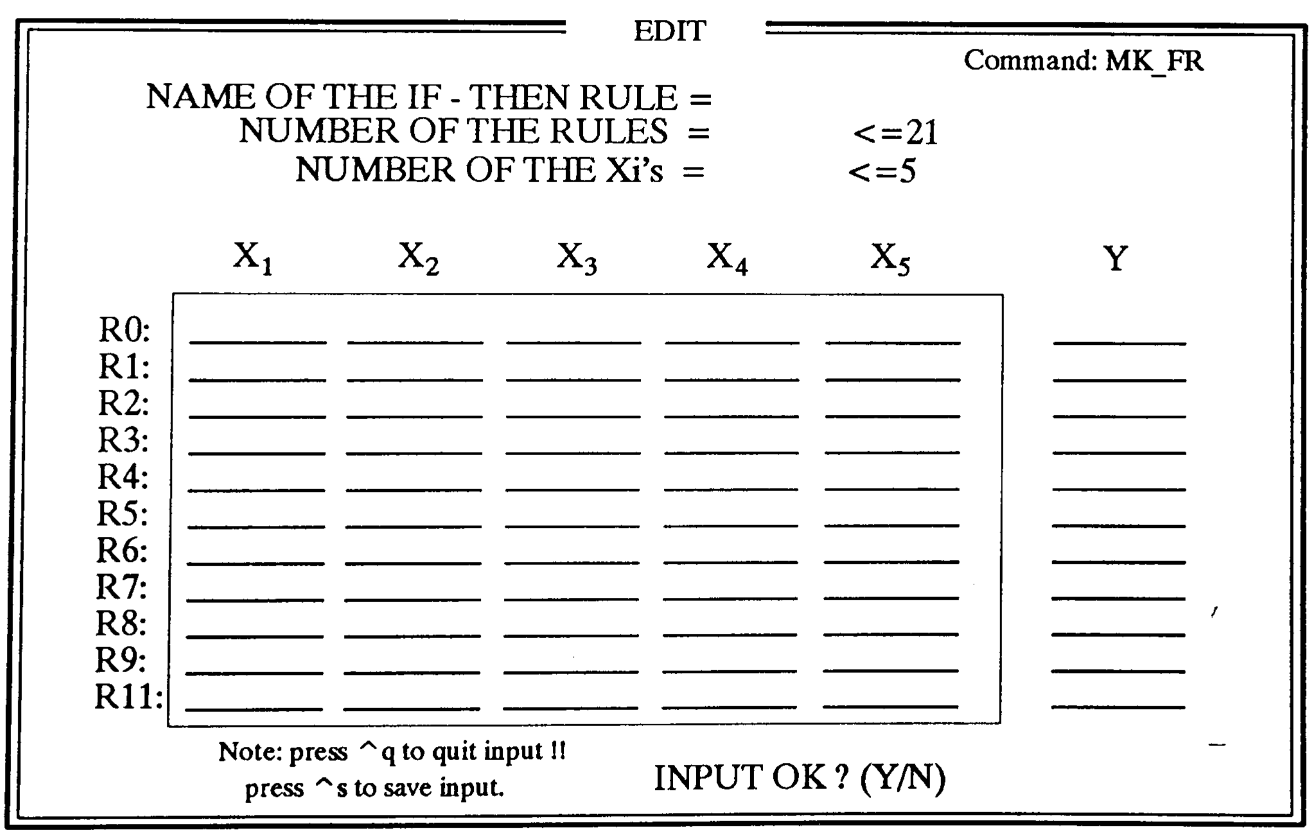


Figure 3-10. the input screen for the IF-THEN rules.

EDIT

Command: INFR_D

NAME OF THE IF-THEN RULES = ________

NAME OF THE INPUT
FUZZY SET : X1 = ________
X2 = ________
X3 = ________
X4 = ________
X5 = ________

Method to calculate the weighting factor
of the Antecedent ? (1 , 2) [ ]
Method of implication ? (1 , 2) [ ]
[1] = A ∧ B, [2] = 1 ∧ (1-A+B)
Name of the fussy set
to store the result = ________

Note: press ^q to quit input !
press ^s to save and run.

INPUT OK ? (Y / N)

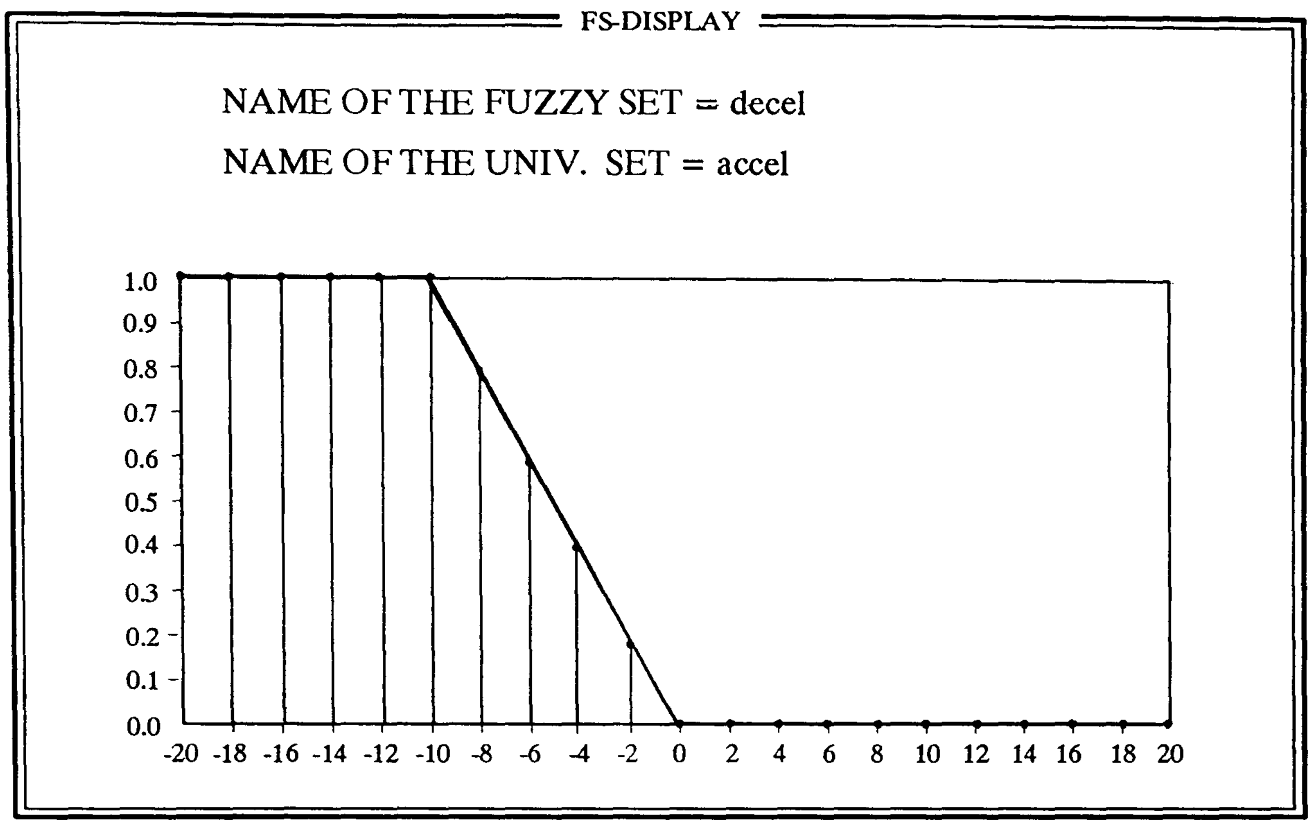
Figure 3-11. fuzzy sets direct reasoning.

(3) Fuzzy Data display

The fuzzy data display commands include

- DSP_UD: Display all the existing universal sets on the same screen by use of MK_UD command.
- DSP_FS: Display all the existing fuzzy sets in the structure list FSLIST graphically.
- DSP_T2: Display all the existing Type 2 fuzzy sets graphically.
- DSP_FR: Display all the existing groups of IF-THEN rules on the same screen by use of MK_FR command.
- DSP_FM: Display all the existing relation matrices on the same screen by use of MK_FM command.

When the fuzzy set membership values are displayed by graph, if these fuzzy set are the result of some controller output files, then, their centroid will also be displayed on the screen.



Figur 3-12. using the DSP_FS command to display a fuzzy set.

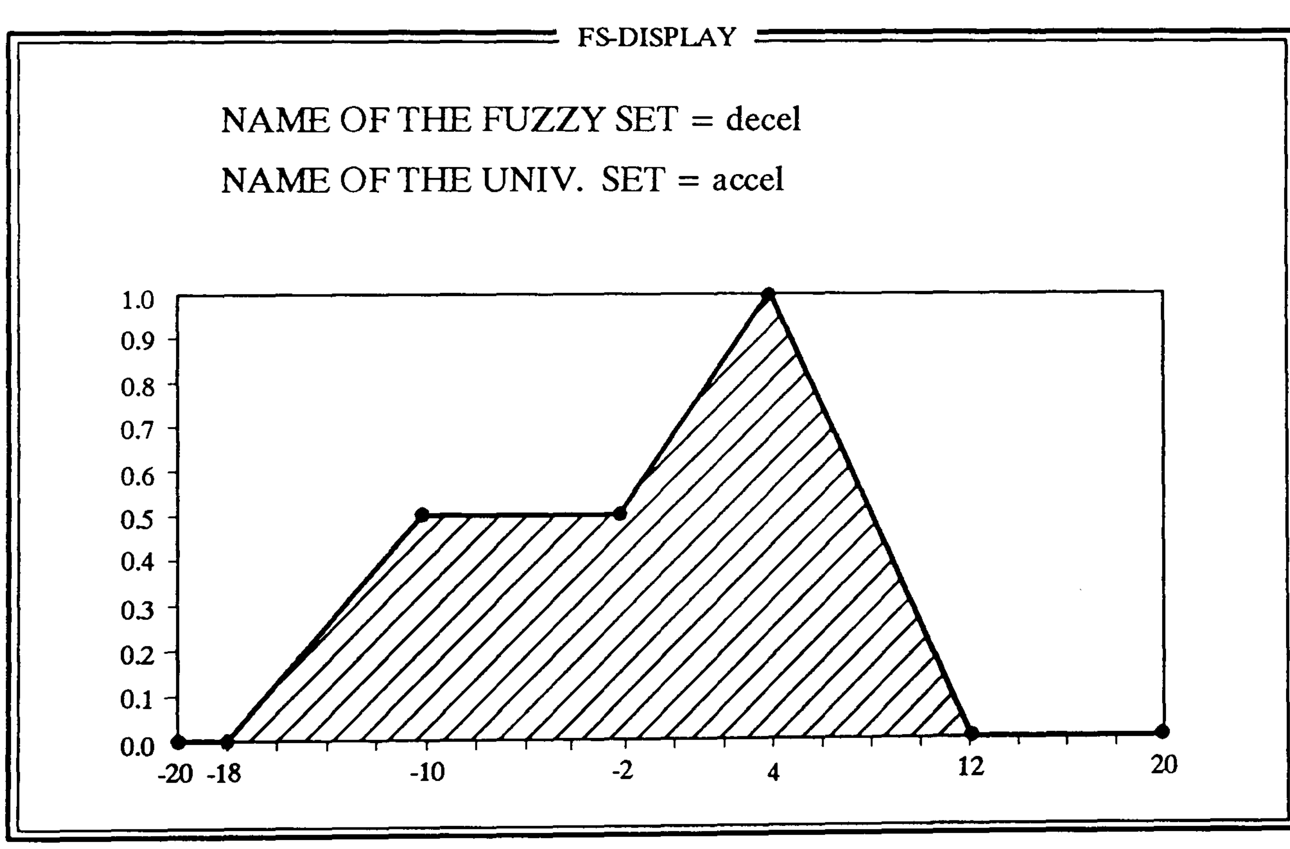


Figure 3-13 example of continuous fuzzy set structure, with 7 breakpoints

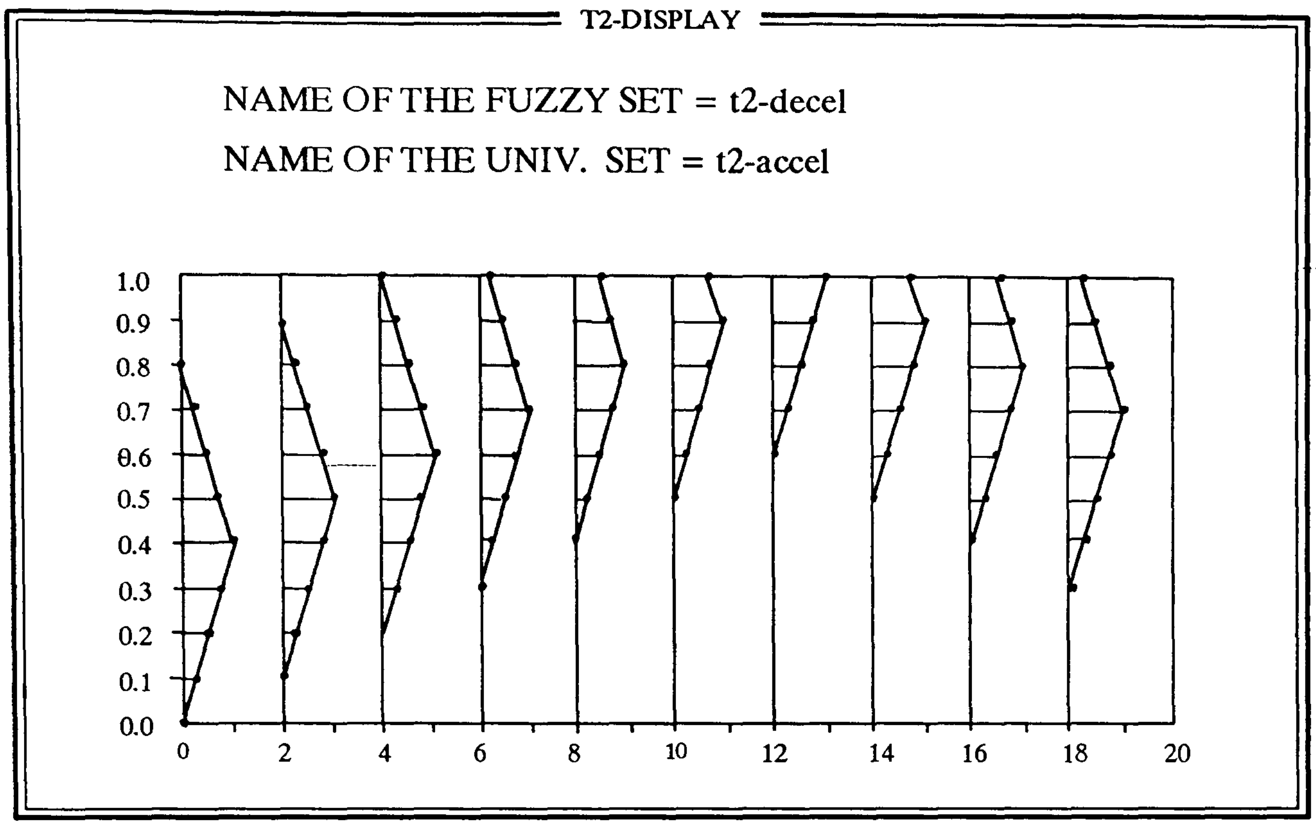


Figure 3-14 A T2 fuzzy set is displayed by the command DSP_T2.

Centroid positions are calculated by the COA defuzzification method. The page up or page down keys can be used to examine all the existing fuzzy sets in the list. Figure 3-12 and Figure 3-13 shows an discrete fuzzy set and a continuous fuzzy set having 7 breakpoints are displayed by commands DSP_FS and DSP_CFS respectively. Figure 3-14 shows the structure of a Type 2 fuzzy set by the command DSP_T2.

(4) Controller Program Generation

The command MK_CTR is used to organize those fuzzy data sets, which are involved in fuzzy reasoning of a group of IF-THEN rules, to form a C program data file. This data file is composed of fuzzy set names in ASCII forms. By adding to it head and tail parts of a controller program (CTRHEAD.C,CTRTAIL.C), a complete source program will be constructed. Using this command, we can construct a fuzzy controller for the experiment set that will be discussed in chapter 4.

(5) List of the System Functions and Commands

As mentioned earlier, a lot of computer memories will be saved if the fuzzy set is represented by breakpoints. Therefore, several commands are modified for handling the continuous fuzzy set in this system. They are specified by adding a capital "C" in the command.

- *Fuzzy set functions and commands*

1. MK_FS : generation of a fuzzy set.

1a. MK_CFS: generation of a continuous fuzzy set.

2. MK_UD : generation of a universal set.

2a. MK_CUD: generation of a continuous universal set.

3. uni_fs : find the union of two fuzzy sets.

4. int_fs : find the intersection of two fuzzy sets.

5. cpl_fs : find the complement of a fuzzy set.

6. l_cpl_fs : find the λ complement of a fuzzy set.

7. t_norm : T-norm operation.

8. t_cnorm : T-conorm operation.

9. center : centroid calculation.

10. mode : find the element of a fuzzy set with largest membership in the set.

11. cardnl : find the cardinal of a fuzzy set.

12. rl_cardnl : find the relative cardnl.

13. height : find the highest membership value.

14. alph_cut : proceed the α-cut operation.

15. nrmlze : normalization of a fuzzy set.

16. concent : concentration operation.

17. dilatn : dilation operation.

18. cntrst : contrast operation.

19. scl_mlt : scale multiplication.

20. pow_mlt : power multiplication.

21. convp : convex check.

22. nmlp : normalization check.

23. eqp : equivalent check.

24. incp_fs : fuzzy inclusion check.

- *linguistic variables and Type-2 fuzzy set functions.*

25. very : VERY operation of a fuzzy set.

26. mr_or_ls : MORE OR LESS operation.

27. slightly : SLIGHTLY operation.

28. sort_of : SORT_OF operation.

29. pretty : PRETTY operation.

30. rather : RATHER operation.

31. lng_tr_v : generation of the linguistic truth value.

32. MK_T2 : construct a T2 fuzzy set.

33. uni_t2 : find the union of two T2.

34. int_t2 : find the intersection of two T2.

35. cpl_t2 : find the complement of a T2.

- *Fuzzy set relations.*

36. MK_FM : generation of fuzzy relations.

37. dom : find the domain of the relation.

38. range : find the range of the relation.

39. prj : find the projection.

40. c_ext : find the cylindrical extension.

41. inv : find the inverse relation.

42. uni_fm : union operation of two fuzzy relations.

43. int_fm : intersection operation of two fuzzy relations.

44. cpl_fm : complement operation of fuzzy relations.

45. cp_fs_fm : composition of the fuzzy set with fuzzy relation.

46. cp_fm_fm : composition of two fuzzy relations.

47. incp_fm : inclusion check of two fuzzy relations.

48. rfrp : reflection check.

49. symp : symmetrical check.

50. asymp : asymmetrical check.

51. trnsp : transition check.

52. eqevp : equivalent check.

53. prordp : proper order check.

- *Fuzzy reasoning.*

54. MK_FR : generation of fuzzy IF-THEN rules.

54a. MK_CFR : generation of IF-THEN rules for the continuous fuzzy sets.

55. INFR_D : inference by direct method.

55a. INCFR_D: direct method of inference for the continuous fuzzy IF-THEN rules.

56. INFR_ID : inference by indirect method.

57. tr_ql : truth qualification operation.

58. cn_tr_ql : converse of truth qualification.

59. MK_CTR : fuzzy logic controller program generation.

- *Fuzzy set display and deletion.*

60. DSP_UD : screen display of the universal set.

60a. DSP_CUD : screen display of the continuous universal set.

61. DSP_FS : screen display of the fuzzy set.

61a. DSP_CFS : screen display of the continuous fuzzy set.

62. DSP_T2 : screen display of the type- 2 fuzzy set.

63. DSP_FM : screen display of the fuzzy relation matrix.

64. DSP_FR : screen display of the IF-THEN rules.

64a. DSP_CFR : screen display of the IF-THEN rules of the continuous fuzzy sets.

65. DLT_UD : deletion of a universal set from the structure list.

65. DLT_CUD : deletion of a continuous universal set from the structure list.

66. DLT_FS : deletion of a fuzzy set from the structure list.

66a. DLT_CFS : deletion of a continuous fuzzy set from the structure list.

67.·7XDLT_T2deletion of a type 2 fuzzy set.

The functions listed above in capital letters are executable programs, and can be used directly as commands. The size of the complete software suite is about 2M bytes. This forms a complete development system for the construction of fuzzy logic controllers.

3.3.2 Fuzzy Control Program Design

Using the development system described in the previous section, a fuzzy controller program can be made. The main function of the development system is to construct an FKB to meet specific requirements. The process of control program generation involves assembling various program segments written in C. These program segments include:

a header file, the main program, and all the subroutines. The FKB takes the form of a two dimensional array and is declared at the beginning of the "main()" program . The program "main()" is not only capable of handling the fuzzy control data, but can also perform the process control functions. It therefore has to perform tasks such as input and output signal manipulation, control data formating, controller parameter setting and the display of status information. The main algorithm of this fuzzy control program is shown in the flow chart in Figure 3-15. In the main control loop, as shown in Figure 3-15(a), the signal input from A/D converter is transformed to a fuzzy singleton by scale mapping, and then it is used to find the array index of the corresponding antecedent fuzzy sets. There are seven quantization levels for the inputs, therefore, the indexes are labeled from 0 to 6. After the antecedent fuzzy sets are found the approximate reasoning procedure subroutine will be called. The flow chart of this subroutine is shown in Figure 3-15(b). The fuzzy reasoning of the first type, that is using Mamdani's minimum operation rule as fuzzy implication function, is applied in this thesis. Therefore, the main task for this subroutine is to find the weighting factors for each input rules involved by fuzzy minimum operation, and then get the minimum one by comparison. For each rule, the consequent fuzzy set will be found by cutting the corresponding output fuzzy set with this minimum weighting factor. Before the subroutine returns, the output fuzzy set is constructed by maximum operation of these consequent fuzzy sets. Finally, the controller output value is calculated by applying the COA method of defuzzification to the resulting output fuzzy set.

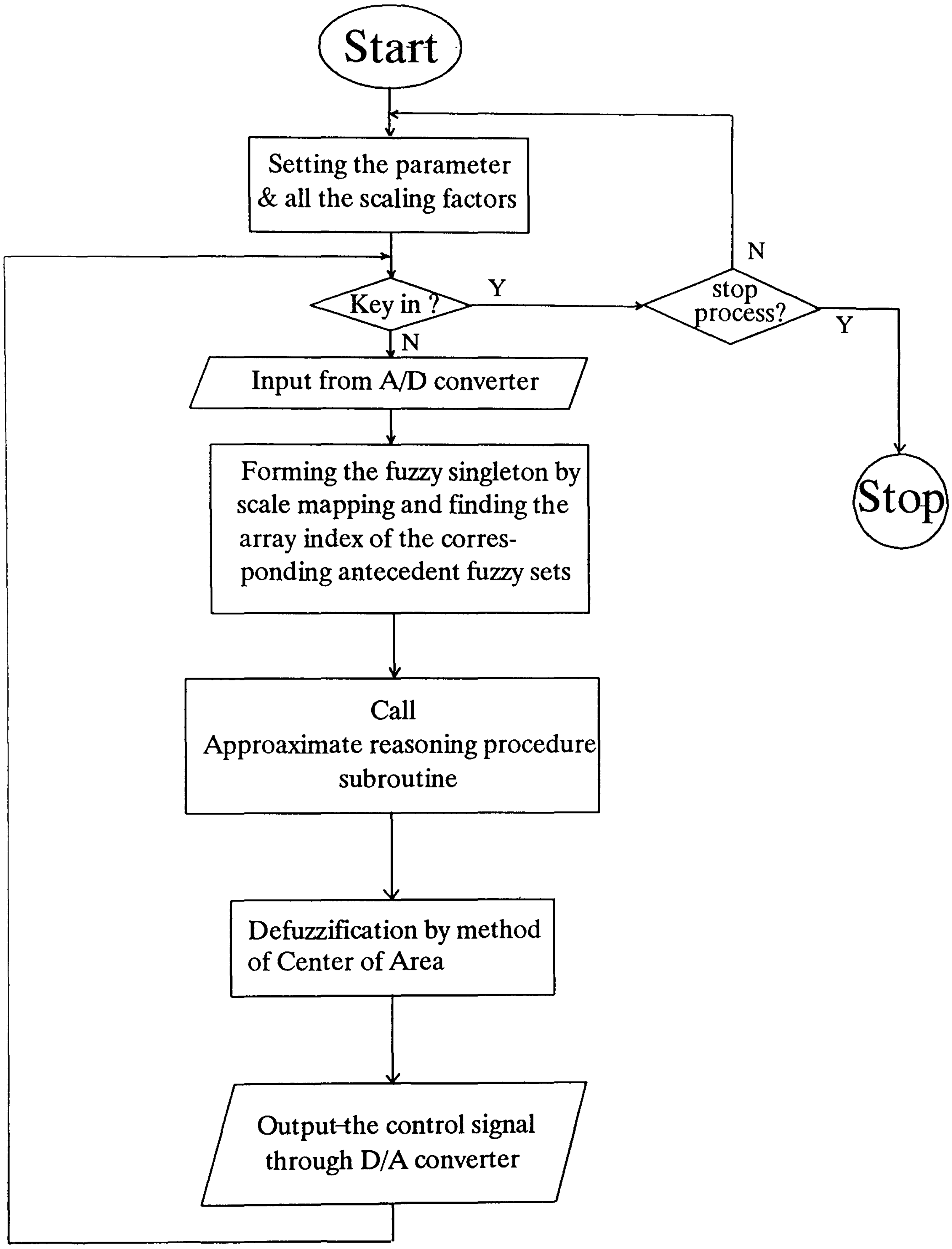


Figure 3-15(a) Flow chart of the fuzzy control main program.

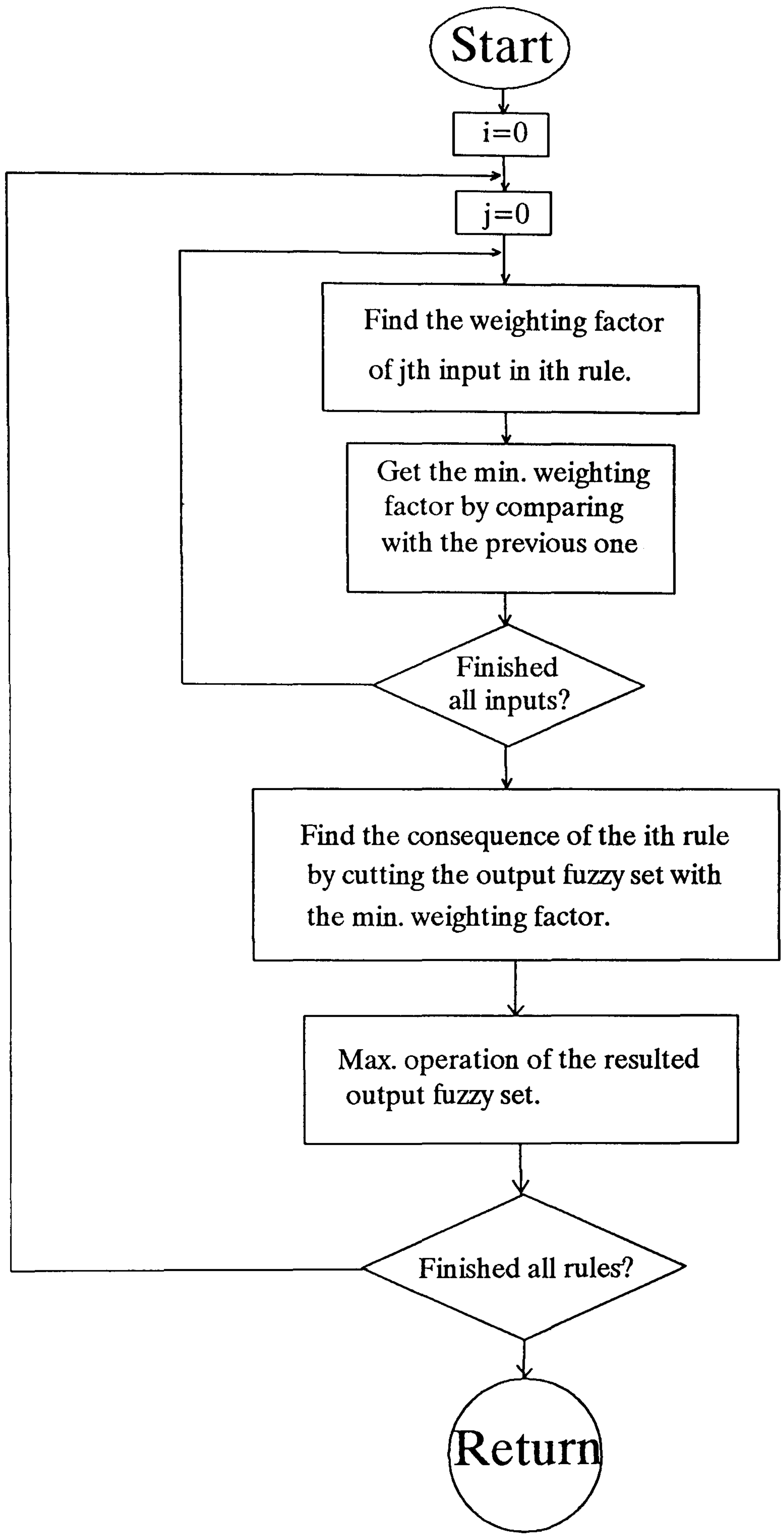


Figure 3-15(b) flow chart of the approximate reasoning procedure.

3.4 Experimental Setup for the Fuzzy Controller

The fuzzy controller was implemented using a 486-based IBM compatible PC. This was fitted with a high speed ADC/DAC board (PCL-818H high performance data acquisition card by Advantech Co., Ltd.) The main feature of this interface control card includes:

(1) 16 analog input channels.

(2) A standard 12 bit successive approximation A/D converter is used to convert the analog inputs. The highest A/D sampling rate is 100 KHz in DMA mode.

(3) Software selectable analog input ranges.

Bipolar : $\pm$0.625V, $\pm$1.25V, $\pm$2.5V, $\pm$5V, $\pm$10V.

Unipolar: 0 to +1.25V, 0 to +2.5V, 0 to +5V, 0 to +10V.

(4) Three A/D trigger modes: Software trigger, programmable pacer trigger and external pulse trigger.

(5) An Intel 8254 Programmable Timer/Counter provides pacer(trigger pulses) at rates from 0.00023 Hz to 2.5 MHz.

(6) One 12 bit monolithic multiplying D/A output channel with an output range of 0V to +10V.

In this experiment, the Bipolar $\pm$5V analog input and Software trigger were selected. However, a bipolar output voltage was required, a level translator and amplifier circuit was constructed using operational amplifiers to generate $\pm$18V from the D/A output. This output is carefully calibrated to give a linear change of 7.4mV/bit.

In many cases, differentiation or its numerical equivalent, differencing is needed in a process, however, which is a "roughing" process. It is sensitive to and accentuates data errors. Hence, a sixth-order Butterworth filter is applied for anti-aliasing before the A/D converter. Another sixth-order Butterworth filter was connected after the D/A converter to act as reconstruction filter. For convenience, a square wave of 3.1V pk-pk approximately was chosen as the input signal for the experiments.

Chapter 4

The Relationship between Fuzzy Decision Table Scaling Factors and the Control Constants of a Digital PI Controller

Since Mamdani's first success, a lot of applications of his pioneering efforts in fuzzy controller design have been reported [53]-[55]. However, many practical fuzzy PI controllers suffer from oscillation problems. Liaw and Wang[56] tried to overcome these problems in an induction motor control task, by switching an integral controller into their system when the fuzzy control action reached a certain limit cycle zone. Jihong Lee[57] tried to tackle the problems by using fuzzy control input resetting techniques. However, neither of these approaches was totally successful.

Based on Tang and Mulholland's[58] study of the relationship between the scaling factors of the MacVicar-Whelan rule-based fuzzy logic controller[59] and the equivalent linear PI controller coefficients, a detailed study was made of the decision table structures and the relationship between fuzzy PI and classical PI controller. The results of this study are described below.

4.1 Comparing Various Control Decision Tables

The decision tables used by Tang, Liaw & Wang, and Jihong Lee are very different. We shall see later that, from a study of the performance of the various controllers, it is concluded that the decision table from Jihong Lee's paper are the most appropriate for the construction of an FLC.

4.1.1 A comparison of digital PI and fuzzy controllers

In many cases, a discrete-time set-point PI controller will be adequate to control a particular process. If the error e(k) at step k is the difference between the process output and the set-point value, then the error and its incremental change are related by the expression

$$de(k) = e(k) - e(k-1) \qquad (4\text{-}1)$$

and e(k) and de(k) provide inputs to the controller. The controller output u(k) is given by

$$u(k) = u(k-1) + du(k) \qquad (4\text{-}2)$$

and is used to drive the plant. The well known incremental or velocity form of the ideal PI controller can be written as

0W

$$du(k) = K_i \cdot T_s e(k) + K_p de(k) \qquad (4\text{-}3)$$

where K_p is the proportional control gain, K_i is the integral control constant and T_s is the sampling interval.

With the same inputs and outputs, however, a fuzzy controller has the internal structure of an expert system. The knowledge base of a fuzzy logic controller usually consists of rules of the form: "if e(k) is 'positive large'(+L)and de(k) is 'negative medium'(-M), then du(k) must be made 'positive small'(+S)." These linguistic variables +L,-M and -S are represented by fuzzy subsets of discrete universes of discourse of the controller variables.

Inference is given by a relationship defined as a fuzzy subset of the Cartesian product formed by the three universes of discourse for the controller variables. Then, using the compositional rule of inference, a fuzzy subset for du(k) can be found even when the controller inputs are non-fuzzy.

4.1.2 Comparing various decision tables

The decision table for the MacVicar-Whelan[59] generalized fuzzy PI controller used by Tang and Mulholland is shown in Figure 4-1. The universes of discourse of the error e(k) and error change de(k) are quantized into fuzzy sets of seven levels, with L=large,M=medium,S=small,'+'standsfor positive and '-' stands for negative. Liaw & Wang used a different linguistic control rules table in their experiments[56]

e(k) \ de(k)	-L	-M	-S	-0	+0	+S	+M	+L
+L	-0	-S	-M	-L	-L	-L	-L	-L
+M	+S	-0	-S	-M	-M	-M	-L	-L
+S	+M	+S	-0	-S	-S	-S	-M	-L
+0	+M	+M	+S	+0	-0	-S	-M	-M
-0	+M	+M	+S	+0	-0	-S	-M	-M
-S	+L	+M	+S	+S	+S	+0	-S	-M
-M	+L	+L	+M	+M	+M	+S	+0	-S
-L	+L	+L	+L	+L	+L	+M	+S	+0

Figure 4-1. Decision table for MacVicar- Whelan generalized PI controller.

e(k) \ de(k)	-L	-M	-S	0	+S	+M	+L
+L	+L	+L	+M	+M	+S	+S	0
+M	+L	+M	+M	+S	+S	0	-S
+S	+M	+M	+S	+S	0	-S	-S
0	+M	+S	+S	0	-S	-S	-M
-S	+S	+S	0	-S	-S	-M	-M
-M	+S	0	-S	-S	-M	-M	-L
-L	0	-S	-S	-M	-M	-L	-L

Figure 4-2. The decision table derived from Liaw & Wang's linguistic control rule table.

as shown in Figure 4-2. For easy comparison, their symbols are somewhat changed here to be compatible with Figure 4-1. Clearly these two tables are very different. To investigate the characteristics of these two approaches, it is useful to look at the corresponding system responses.

Figure 4-3 shows the typical output response of a stable closed loop system. According to the magnitude of the error 'e' and the sign of the change of error 'de', the response plane is roughly divided into four areas. The index used for identifying these areas is defined as

$$a_1 : e > 0 \text{ and } de < 0, \quad a_2 : e < 0 \text{ and } de < 0$$

$$a_3 : e < 0 \text{ and } de > 0, \quad a_4 : e > 0 \text{ and } de > 0$$

A and C are the set point cross over points. B and D are the maximum and minimum points. Consider initially, the situation when the decision table of Figure 4-1 is used. In this case, at the cross over point A, which corresponds to the situation where the e=±0 and de=-L(negative large), the control output is set to +M.

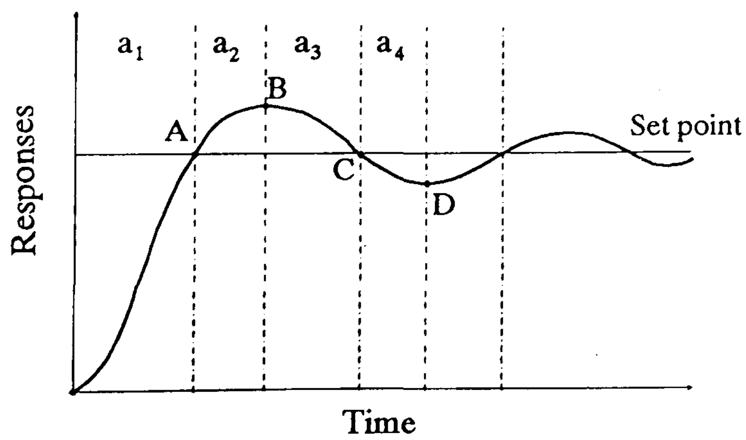


Figure 4-3 The general waveform of a system response.

e(k) \ de(k)	-L	-M	-S	0	+S	+M	+L
+L	0	+S	+S	+M	+M	+L	+L
+M	-S	0	+S	+S	+M	+M	+L
+S	-S	-S	0	+S	+S	+M	+M
0	-M	-S	-S	0	+S	+S	+M
-S	-M	-M	-S	-S	0	+S	+S
-M	-L	-M	-M	-S	-S	0	+S
-L	-L	-L	-M	-M	-S	-S	0

Figure 4-4. Lee's control rule table.

If we assume the conventional closed loop situation where the feedback signal is subtracted from the input reference signal to produce the error signal, then, at this point the controller increases its output to the plant positively, which will further increase its overshoot, thus drive the plant response in the wrong direction. Similarly, at the points B, C and D there is a similar problem with the rules selecting an output of the wrong sign. Thus, for the conventional closed loop situation the polarity of the MacVicar-Whelan table must be reversed to ensure stability.

If we now consider the decision table of Figure 4-2, for the Liaw & Wang's case, the maximum and minimum points B and D are correct but the cross over points A and C produce an output of the wrong sign. This represent 90° rotation of the required control action and it is clearly wrong.

The control rule table of Figure 4-4 is that due to Jihong Lee[57]. Study of this table shows that it generates outputs of the correct polarity under all conditions. For this reason, a table of this form is used in the remainder of this work. The table due to Jihong Lee corresponds to that of MacVicar-Whelan with its polarity reversed.

4.2 Comparing Fuzzy Logic with Classical Controller Design

Tang and Mulholland[58] have proposed a unified approach for comparing the performance of fuzzy and non-fuzzy controller designs. They point out that the fuzzy control rule tables represented by Figure 4-1 and 4-4 show a definite pattern of symmetry about their main diagonal. Projecting the entries onto an axis perpendicular to the main diagonal demonstrates the fuzzy segments for control policy in terms of the resultant overlap among the various values of du(k). This means that, as shown in Figure 4-5, those regions formed by grouping the areas with the same output strength is the fuzzy segments having the same control policy.

e(k) \ de(k)	-L	-M	-S	0	+S	+M	+L
+L	0	+S	+S	+M	+M	+L	+L
+M	-S	0	+S	+S	+M	+M	+L
+S	-S	-S	0	+S	+S	+M	+M
0	-M	-S	-S	0	+S	+S	+M
-S	-M	-M	-S	-S	0	+S	+S
-M	-L	-M	-M	-S	-S	0	+S
-L	-L	-L	-M	-M	-S	-S	0

Figure 4-5. The boundary lines formed by regions with the same output strengths in a decision table.

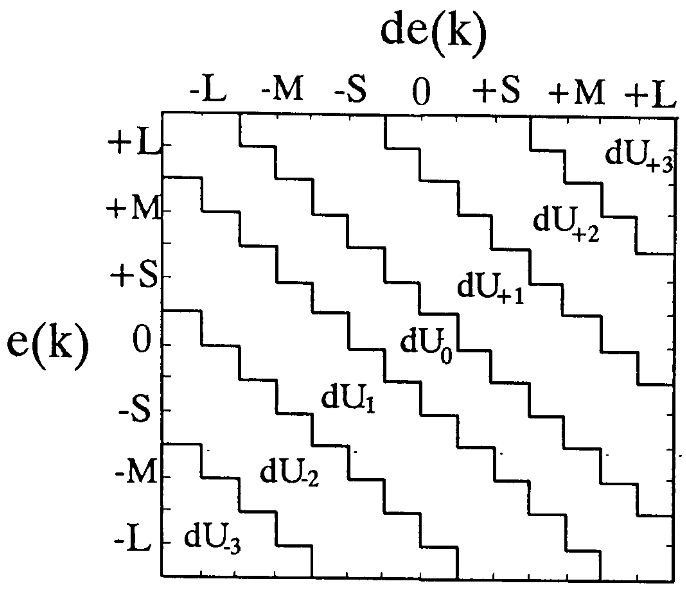


Figure 4-6. Decision table obtained by halving the quantization levels.

The overlap is clearly less fuzzy when the quantization of the independent variables is halved as shown in Fig. 4-6. We can continue this process of reducing the quantization levels until the frontiers of the control regions no longer are fuzzy.

Then, the control policy that results from an infinitely fine quantization is, as shown in Fig.4-7, where du_i ($i=-3,-2,\ldots,+3$) denotes the quantized controller outputs.

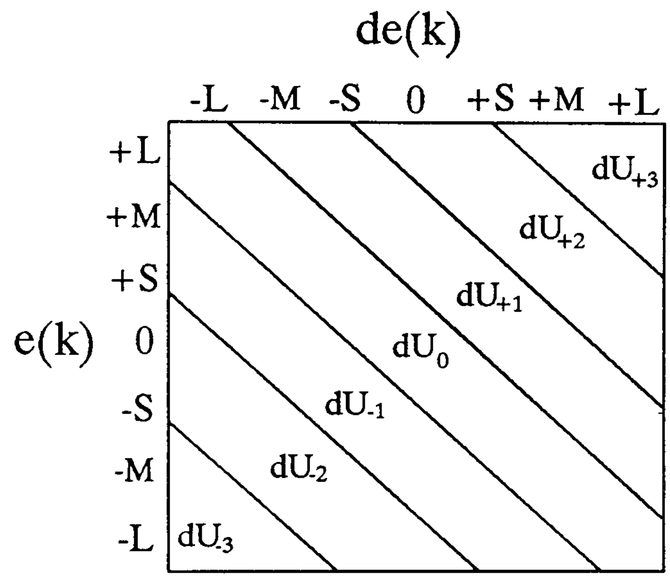


Figure 4-7 Decision table with infinitely fine quantization levels and seven control output bands.

If the regularized control frontier lines have a slope -C, as shown in Fig. 4-8, then, the control action du_x of a particular operating point (e_x, de_x) can be determined. The straight line with slope -C through the operating point is:

$$\frac{(e - e_x)}{(de - de_x)} = -C \qquad (4\text{-}4)$$

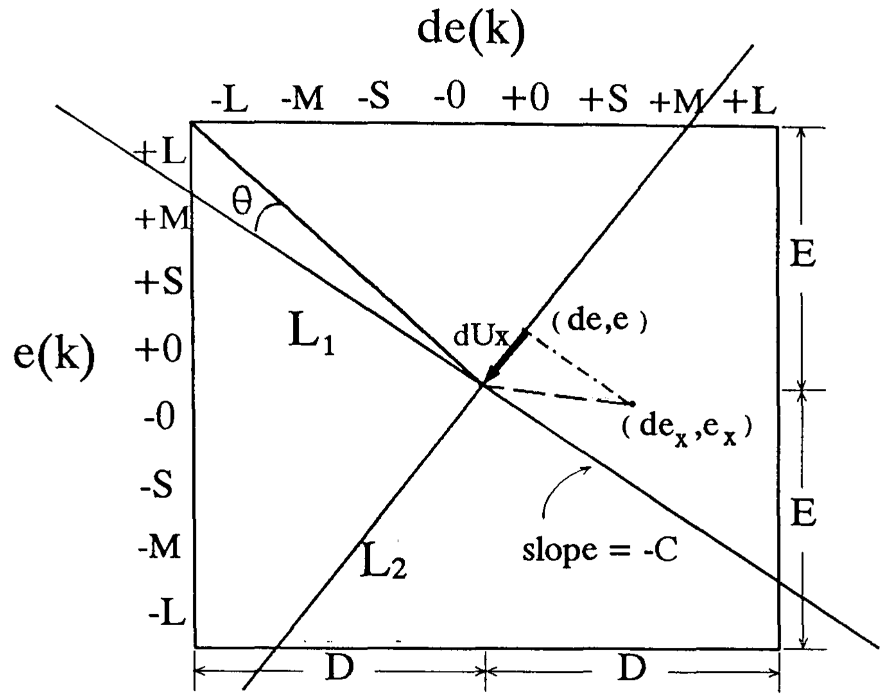


Figure 4-8. Idealized decision table for a linear PI controller where the line L_1 represents the regularized control frontier and the line L_2 (slope 1/C) represents the range of control actions.

where, C depends on the scaling. Therefore, if the error range E and the change of error range D of the decision table are introduced for scaling, then equation (4-4) will become

$$\frac{\frac{e}{E} - \frac{e_x}{E}}{\frac{de}{D} - \frac{de_x}{D}} = -C \qquad (4\text{-}5)$$

Note that the constant *C* here is not equal to the constant *C* in equation (4-4). The line(L_2), perpendicular to this line, that passes through the origin is

$$\frac{\frac{e}{E}}{\frac{de}{D}} = \frac{1}{C} \qquad (4\text{-}6)$$

Solving (4-5) and (4-6) gives

$$\frac{e}{E} = \frac{1}{(1 + C^2)} \left[\frac{e_x}{E} + C\frac{de_x}{D} \right] \tag{4-7}$$

and

$$\frac{de}{D} = \frac{C}{1 + C^2} \left[\frac{e_x}{E} + C\frac{de_x}{D} \right] \tag{4-8}$$

Thus, combining equation (4-7) and (4-8), the projected distance from the point (de,e) to the origin which is

$$du_x = \left[\left(\frac{e}{E}\right)^2 + \left(\frac{de}{D}\right)^2 \right]^{\frac{1}{2}} \tag{4-9}$$

This distance is linearly proportional to the control action, and therefore we can rearrange this to give

$$du_x = \frac{p}{\sqrt{(1 + C^2)}} \left[\frac{e_x}{E} + C\frac{de_x}{D} \right] \tag{4-10}$$

where P is the constant of proportionality to produce the real output du_x.

Now, without out lose of generality, let C=1.Then this equation become

$$du_x = \frac{p}{\sqrt{2}} \left[\frac{e_x}{E} + \frac{de_x}{D} \right] \tag{4-11}$$

Comparing this with equation (4-3), we see that equation (4-11) represents the characteristic equation of a classical PI controller with the control constants

$$K_p = \frac{p}{\sqrt{2}D} \tag{4-12}$$

$$K_i \cdot T_s = \frac{P}{\sqrt{2}E} \tag{4-13}$$

Therefore, if D and E are the maximum ranges of de_x and e_x in the fuzzy decision table, the smaller the value of D the more de_x affects the control result. Thus, since T_s is a constant, varying D and E will have the same effect as varying the value of K_p and K_i.

Usually after inference, the scaling factor 'U' has to be applied, which is the maximum range of the support element in the universe of discourse for du_x. This means that the true controller output values will be limited by the range $\pm$U before it is actually fed to the system. At the extreme ends of the control table, where e_x=E and de_x=D the following relationship is found from equation (4-11)

$$U = \sqrt{2}p \qquad (4\text{-}14)$$

This gives $p=U/(2)^{1/2}$ and equation (4-12) and (4-13) can be rearranged as

$$K_i \cdot T_s = \frac{U}{2E} \qquad (4\text{-}15)$$

$$K_p = \frac{U}{2D} \qquad (4\text{-}16)$$

Thus, K_i and K_p are directly proportion to the value of range U. Therefore, just as the design of a classical PI controller involves choosing the right values for the constant K_p and K_i, the design of a fuzzy PI controller involves the task of choosing the proper values for U, D and E. In later chapters we will look at experiments which illustrate the equivalence of the ranges E, D and U in a fuzzy controller, to the variables K_p and K_i in a PI controller. Some range modification techniques which can be applied to improve the control response are also introduced in the following chapters.

Chapter 5

The Effect of Scaling Factors on a Fuzzy PI Controller

In this chapter, we first look at the design of a fuzzy logic PI controller, and at methods used to test it, using an analog simulation of a plant, which allowed a comparison with a conventional digital PI controller. Following this, the chapter then describes a detailed study of the effects of the choice of scaling factors on the control actions of a fuzzy PI controller. This shows that there are several limitations on choosing the decision table ranges in order to avoid saturation or overdamping effects. In some cases, if choosing the range of error change D is too small or error range E is too large, it will result in a special sliding motion phenomenon. The boundary conditions of this sliding regime will be discussed latter in detail. Moreover, a different way of fine tuning a fuzzy controller is possible by varying the decision table ranges and by changing the fuzzy controller itself. If necessary, the fuzzy PI controller can be changed easily to become an integral controller without the addition of a special integral controller. At the end of this section, a variable structure systems(VSS) formed by fuzzy PI and PD controllers are also tried to control a second order system.

5.1 Experimental setup

For this series of experiments, a set of analog control modules based on operational amplifiers was used to model the target plant. By varying two amplifier gains K_1 and K_2, and adding on integrator 1/s, any desired plant transfer function of the form $K_1/(s+K_2)$ could be produced. The experimental arrangement used is shown in Figure 5-1. This set up can be used to investigate the performance of either a digital PI controller or a fuzzy PI controller. As mentioned in chapter three(section 3.4), the fuzzy PI controller is implemented using an IBM compatible PC. The digital

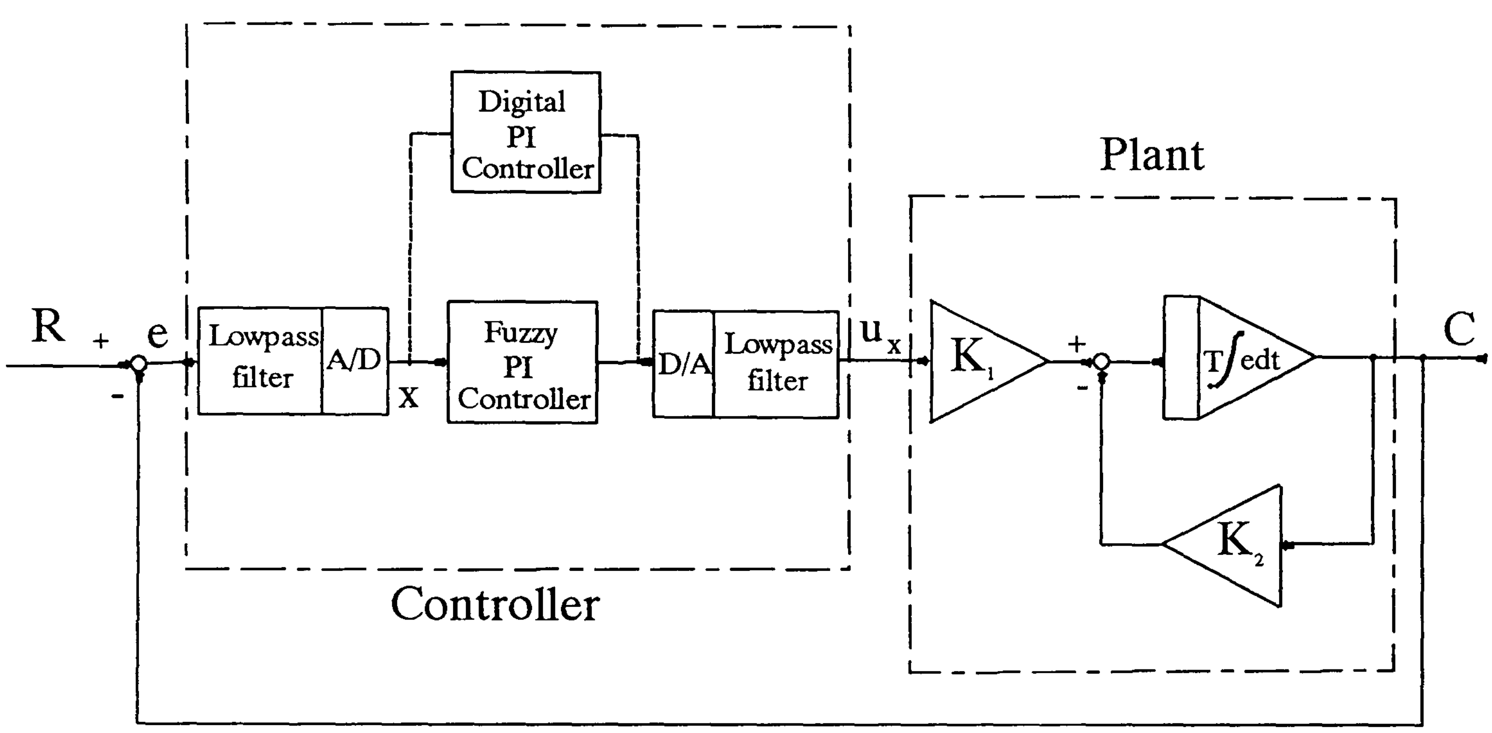


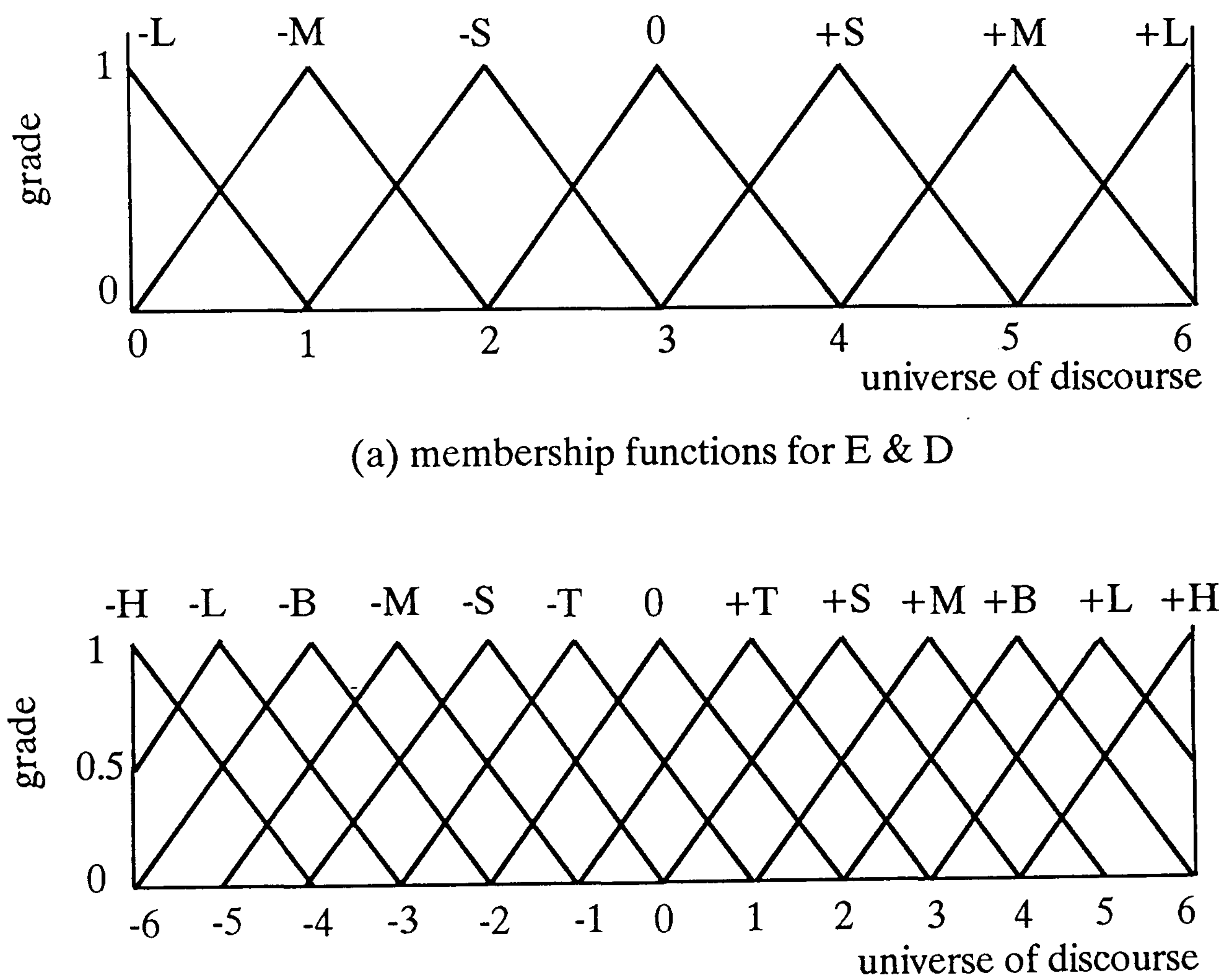
Figure 5-1. A block diagram of the experimental setup.

PI controller uses the same hardware, for which the velocity algorithm described in equation (4-3) is applied. The design of the fuzzy PI controller is described in the next section.

5.2 The Fuzzy PI Controller Designed for this Experiment

In common with most other researchers', the fuzzy reasoning method selected for this application is Mamdani's minimum operation method. In this experiment, the error and error change are used as the two control inputs. These are crisp values, which are referred to as fuzzy singletons.

To define the fuzzy linguistic control rules as shown in Figure. 5-3, the membership functions corresponding to each element in the linguistic set must be defined. Although many types of membership function can be defined, the equal triangular type as shown in Fig. 2-4 is applied here, for simplicity. The universe of discourse of the error (with range -E to +E)and error change (with range -D to +D)are quantized into 7 levels,



(a) membership functions for E & D

(b) membership functions for U

Figure 5-2. The membership functions.

from 0 to 6. The linguistic sets used to represent these levels are

$$\{-L\ ,\ -M,\ -S\ ,\ 0\ ,\ +S\ ,\ +M,\ +L\ \} \qquad (5\text{-}1)$$

where '+'is positive, '-' is negative, S is small, M is medium, L is large and 0 is zero.

However, in order to send an inferred result to the output with more fine quantization, the controller output du (with range -U to +U)is quantized into 13 levels from -6 to 6. The linguistic sets used are shown below

$$\{-H\ ,\ -L\ ,\ -B\ ,\ -M,\ -S\ ,\ -T\ ,0\ ,\ +T\ ,\ +S\ ,\ +M,\ +B\ ,\ +L\ ,\ +H\} \qquad (5\text{-}2)$$

where T is tiny, L is large and H is huge. The membership functions of these quantization levels are also expressed in Figure 5-2. Figure 5-2a shows its simplified single character sets which was used in the decision table. The linguistic control rules considered earlier and defined by Figure 4-4, but with thirteen quatization levels for the controller output, are shown in Figure 5-3. This table implies the conditional rules, for example, that the element of the first row and seventh column is +Himplies that

e(k) \ de(k)	-L	-M	-S	0	+S	+M	+L
+L	0	+T	+S	+M	+B	+L	+H
+M	-T	0	+T	+S	+M	+B	+L
+S	-S	-T	0	+T	+S	+M	+B
0	-M	-S	-T	0	+T	+S	+M
-S	-B	-M	-S	-T	0	+T	+S
-M	-L	-B	-M	-S	-T	0	+T
-L	-H	-L	-B	-M	-S	-T	0

Figure 5-3. The linguistic control rules used in this experiment.

IF e(k) is +Land de(k) is +L

THEN the control input to the plant is +H.

The scaling factors E, D and U are not fixed as is usually the case, because they will be used as varying factors to see their effects on K_p and K_i, and also will be used as tuning factors.

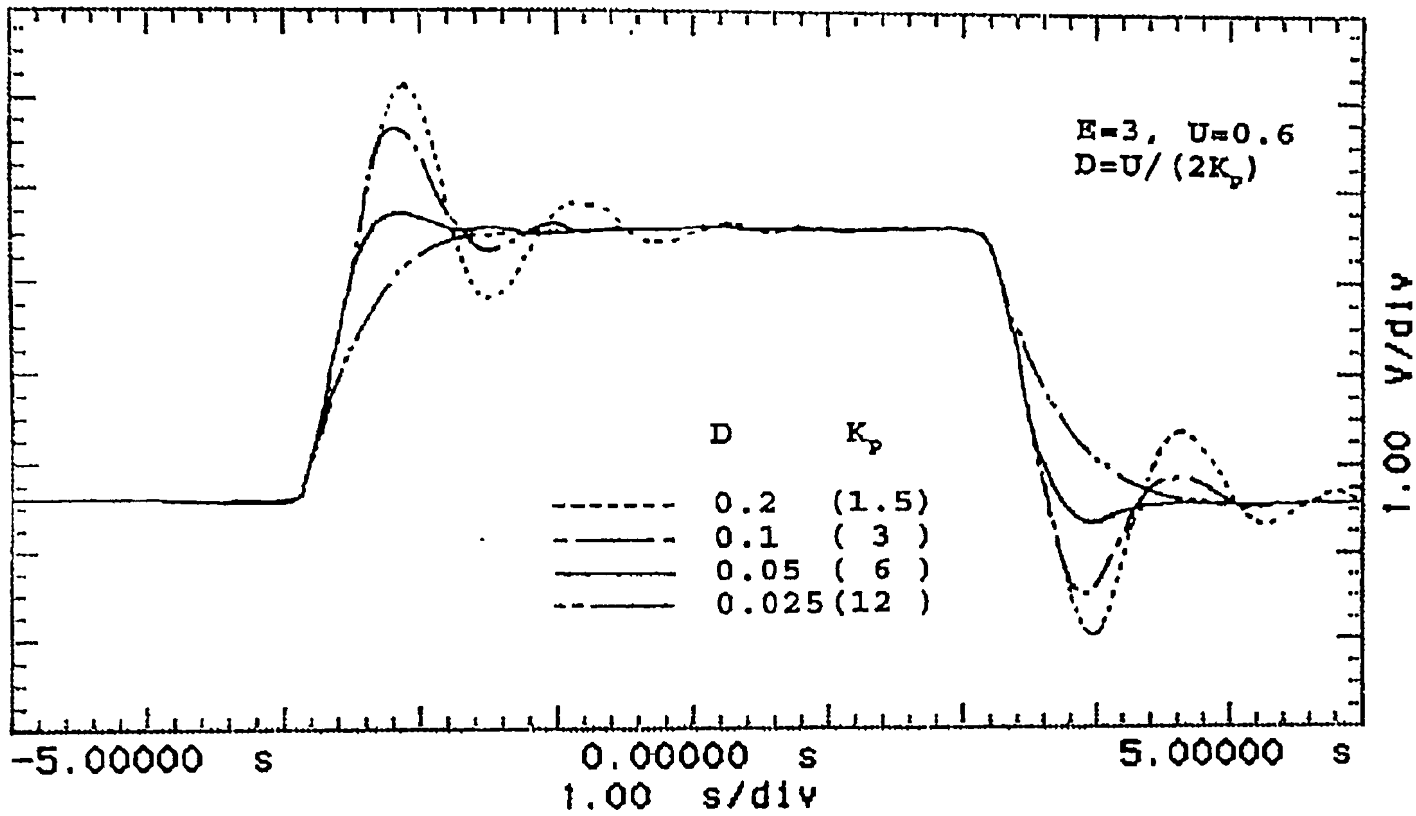
5.3 The fuzzy PI controller experiments performed

Using the experimental setup which has been described in section 5.1 as the target plant and a step input change, a series of fuzzy PI controller experiments were performed as described in this section.

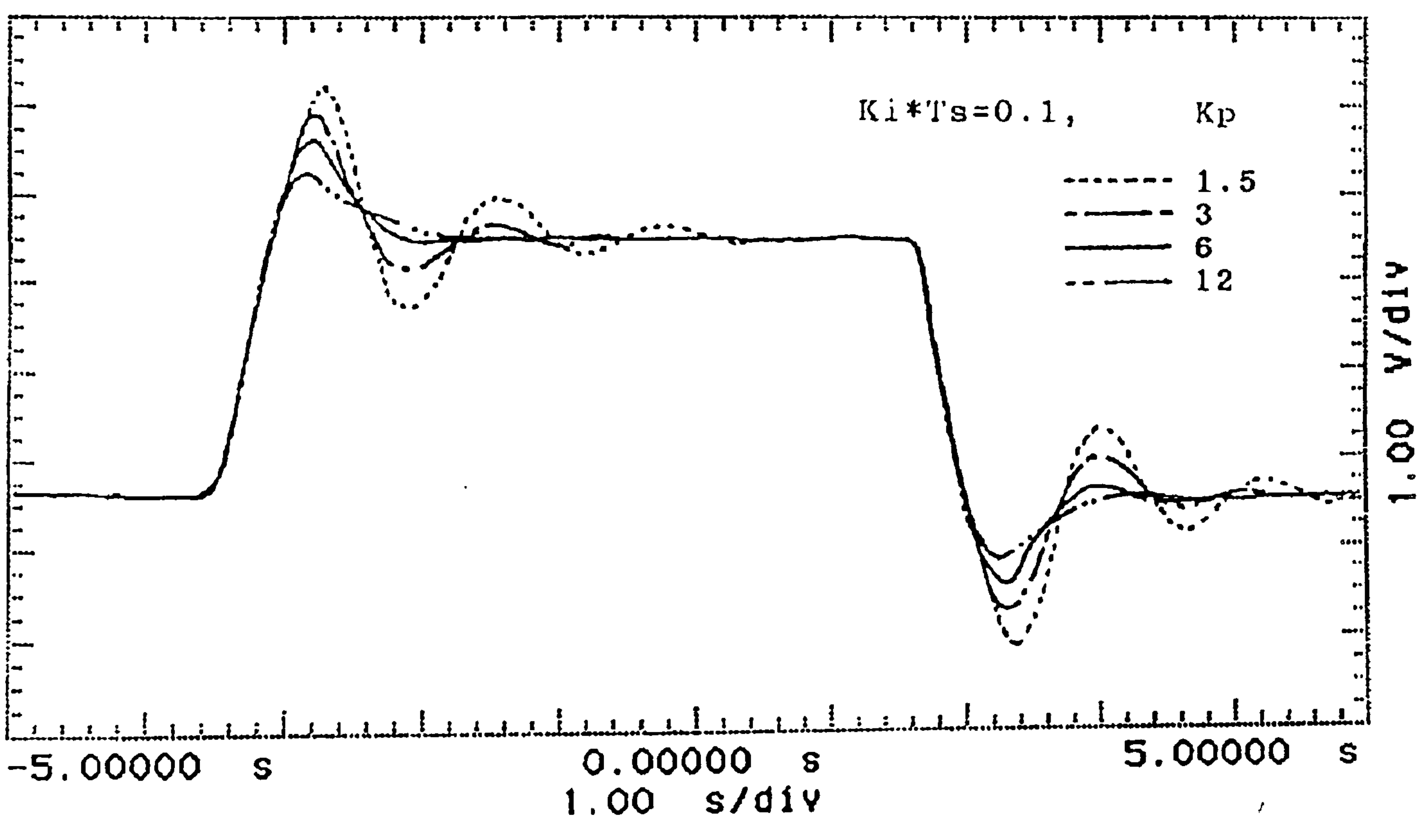
5.3.1 The effect of varying the decision table ranges on the behavior of a Fuzzy PI controller and its comparison with the control constant of a digital PI controller

Some experiments were performed to see if the ranges D, E and U of a fuzzy PI controller could be varied to achieve similar characteristics as those obtained by setting the constants K_p and K_i for a digital PI controller. First, an experiment was performed by setting the digital PI controller constants $K_i{*}T_s=0.1$(T_s is the sampling rate) and varying its K_p value while controlling a plant with a transfer function of $1/(s+1)$.In the fuzzy controller case, for each value of K_p used by the digital PI controller, equations (4-15) and (4-16) were applied to calculate the corresponding values of U, E and D. Initially, the value of E was set to 3V, and D was varied by varying K_p. The results obtained are shown in Figure 5-4(a) and 5-4(b). The fuzzy PI controller was similar to the digital PI controller in the way the overshoot and oscillations decreased with increasing K_p values. However, the output waveform of the fuzzy PI controller looks similar to the digital PI controller only when K_p was less than 3. Above this value it tended to be heavily overdamped. The reason for this effect will be discussed in detail in the next section.

Next, an experiment was performed by setting $K_p=6$ and varying K_i in the digital PI controller and selecting equivalent values for the fuzzy PI controller by E, to drive the same plant $1/(s+1)$.The results are shown in Figure 5-5. In both cases, the ripple and overshoot increased with increasing K_i. In the case of the fuzzy PI controller, once the value of K_i became less than 0.16 the damping of the system response became very significant. The reason for this effect will be discussed in the next section.

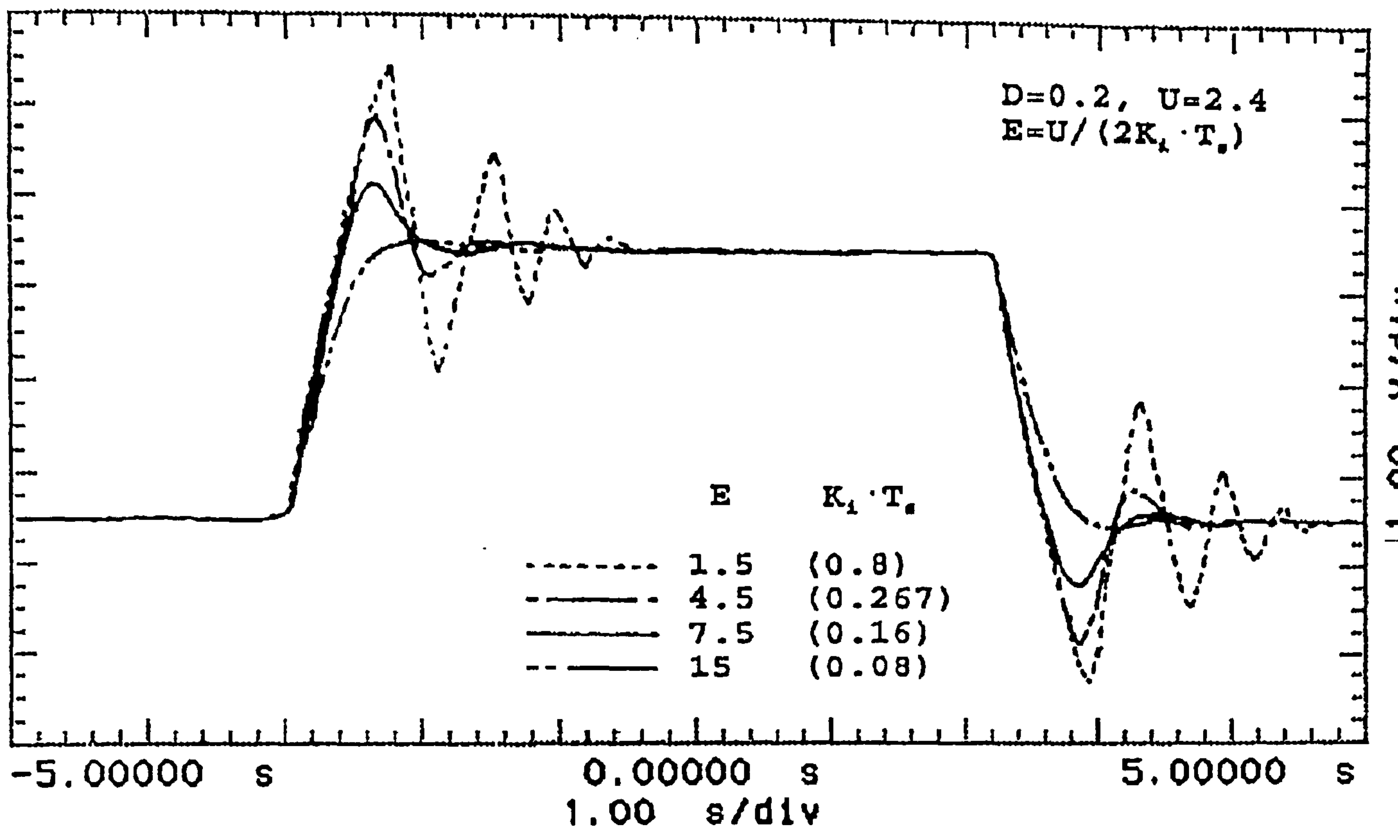


(a)

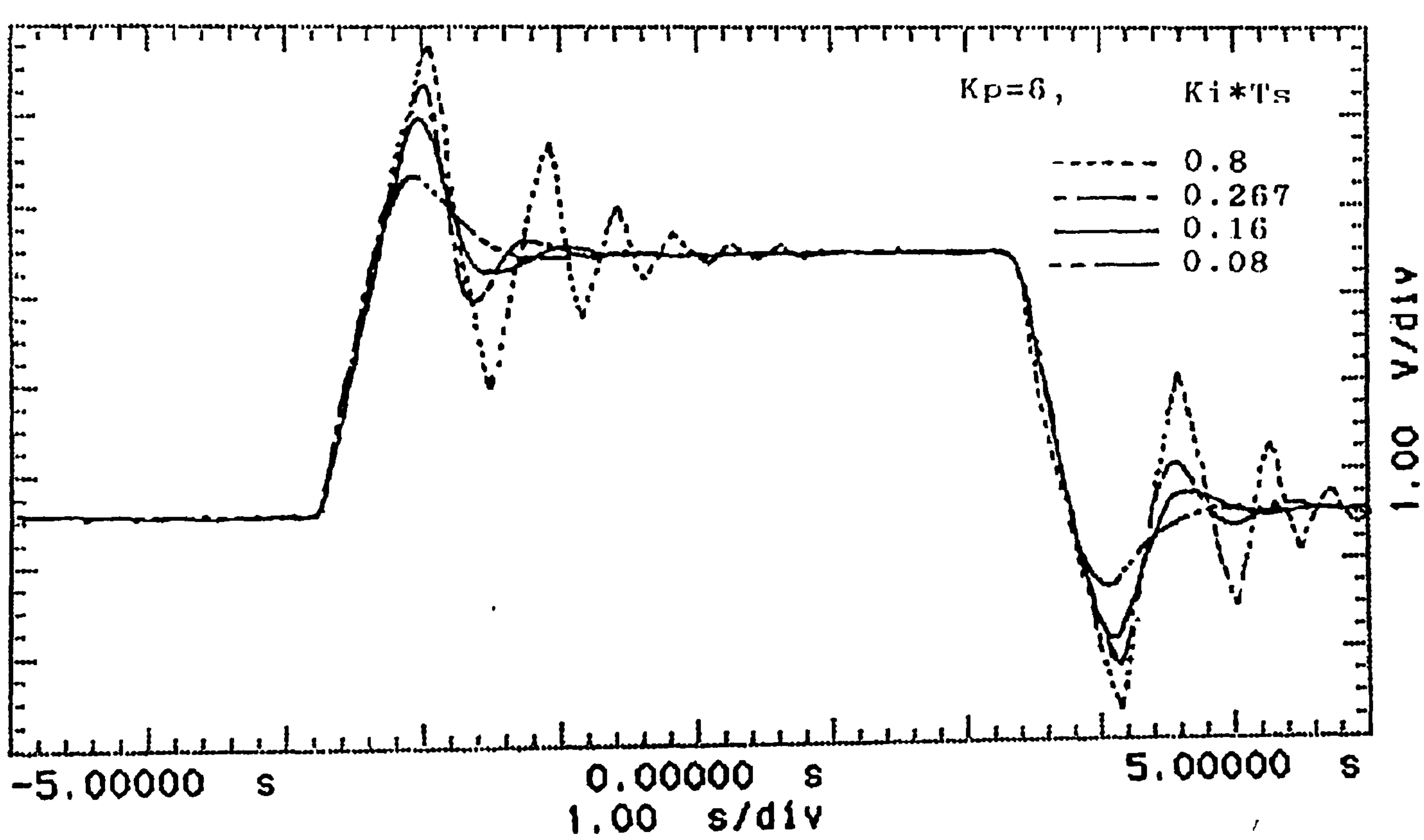


(b)

Figure 5-4. The output of a plant 1/(s+1)which was driven by (a) a fuzzy PI controller with ranges E=3V,U=0.6V and by (b) a digital PI controller with constant $K_i \cdot T_s = 0.1 K_p$ is varying in both cases.



(a)



(b)

Figure 5-5. The output of a plant 1/(s+1)was driven by (a) a fuzzy PI controller with ranges D=0.2Vand U=2.4V,(b) a digital PI controller with constants K_p=6,K_i was varied in both cases.

5.3.2 Limitation of the decision table ranges.

As mentioned in the previous section, the behavior of a fuzzy PI controller is largely determined by its control decision table. Unlike an ordinary digital PI controller, where the control trajectory can move freely in the phase plane, the fuzzy controller decision table ranges will impose limits on the changes to its variable. When these limits are reached the system output can exhibit the characteristics of a sliding control or show an overdamped response. To investigate this effect experiments will be conducted on the same target plant, 1/(s+1),with a 3.1V step input change, when controlled by a fuzzy PI controller. The limits on D, E, and U coud then be studied.

(1) The saturation phenomena caused by decreasing the values of the E, D and U ranges together with the same ratio.

To see the influence of an inappropriate choice of the control decision table ranges D, E and U, an experiment was performed by keeping K_p and K_i constant while increasing or decreasing the E, D and U ranges by the same factor. Since K_p=U/2Dand $K_i \cdot T_s$=U/2E,if D, E and U are varied by the same factor, both the values of K_p and K_i remain the same. The effect of these changes are illustrated in Figure 5-6. This shows a control action trajectory drawn as a dashed curve within a control table having E and D set equal to "1". It also shows a second control table formed by halving the values of E and D. This results in a table of reduced size . Use of this second table will result in some of the control action trajectory points falling outside of the new control table. This will result in these points being limited at the extreme edges of the new control table. In other words, the control output du will be saturated at the maximum values of its output ranges.

When the actual inputs de(k) or e(k) are limited to their maximum E or D ranges, the output du will be saturated at this point by the output fuzzy set assigned to it, and this will result in an

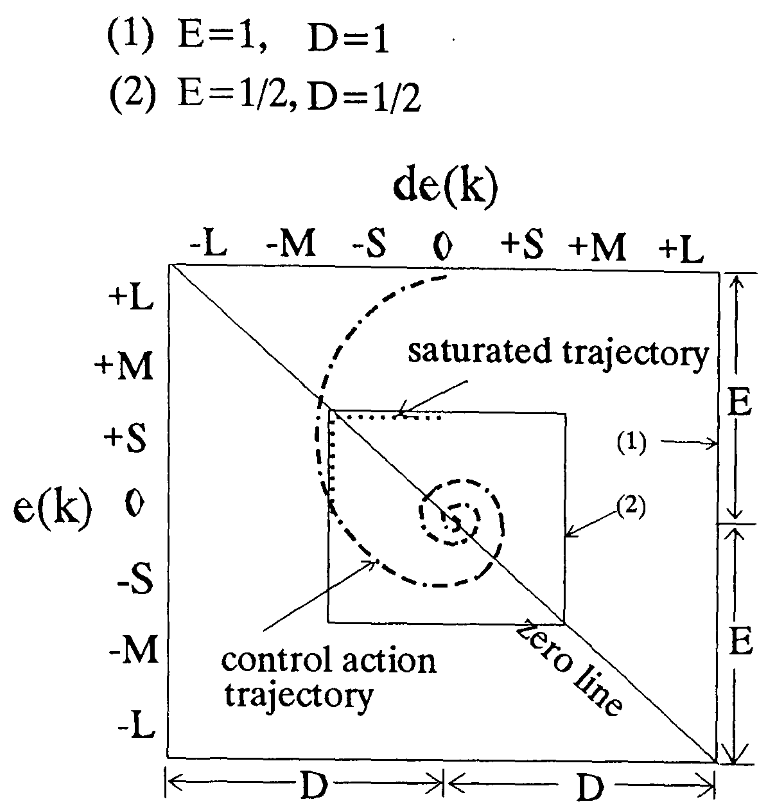


Figure 5-6. The effect of decreasing E, D and U together by the same factor. (K_p and K_i are constant.)

inadequate driving force being produced. This means that the system will respond more slowly if the values of E exceeds some range. This results in a longer rise time and an overdamped output response.

If all the E, D and U ranges are increased sufficiently in the same ratio, the complete trajectory will now be included in the new control table, and no output saturation will occur. Therefore, as long as the actual control trajectory can be included in the control table, increasing or decreasing the D, E and U by the same factor will not change its output value. The experimental results of using different values of D, E and U while keeping K_p and K_i constant are shown in Figure 5-7(a). The control trajectories and the total number of saturation points reached within 1000 samples are shown in (b), (c), (d) and (e). From (b) to (e), the ranges of E, D and U are decreased by the same factor. No saturation point was found in Figure 5-7(b), but the number of saturation points increased with decreasing decision table ranges. Figure 5-7(e) shows the worst case with 121 positive and 119 negative saturation points. In fact, although not shown in the Figure, when all the ranges are expanded by 50%, they became E=18V,D=0.6V and U=3.6Vand the output response keeps almost the same shape as Figure 5-7(b).

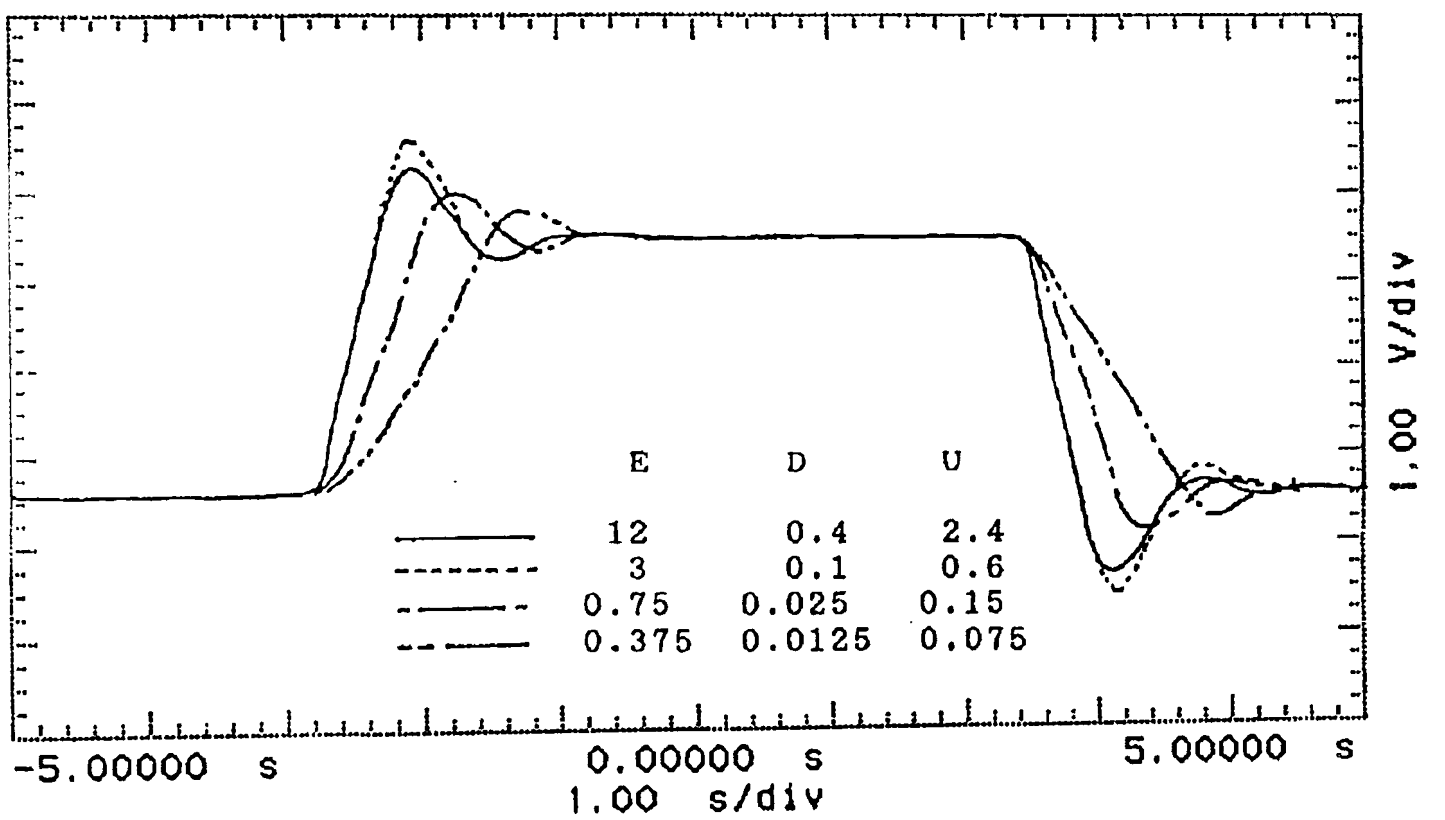
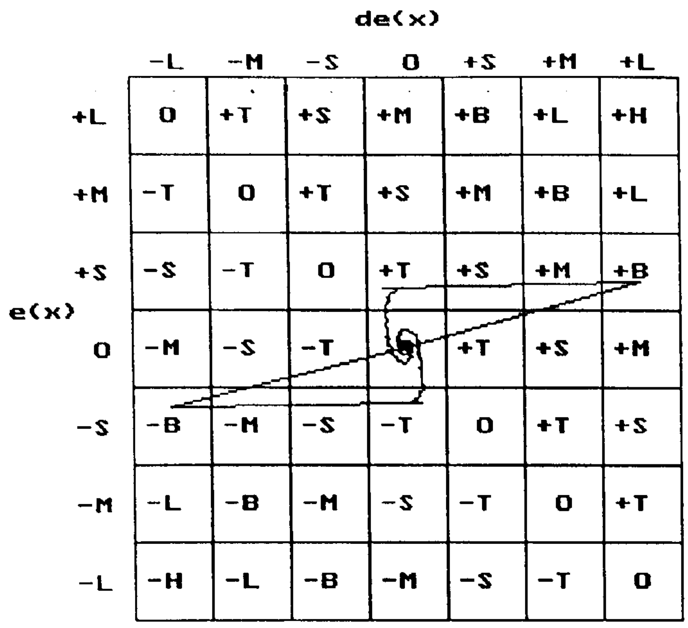
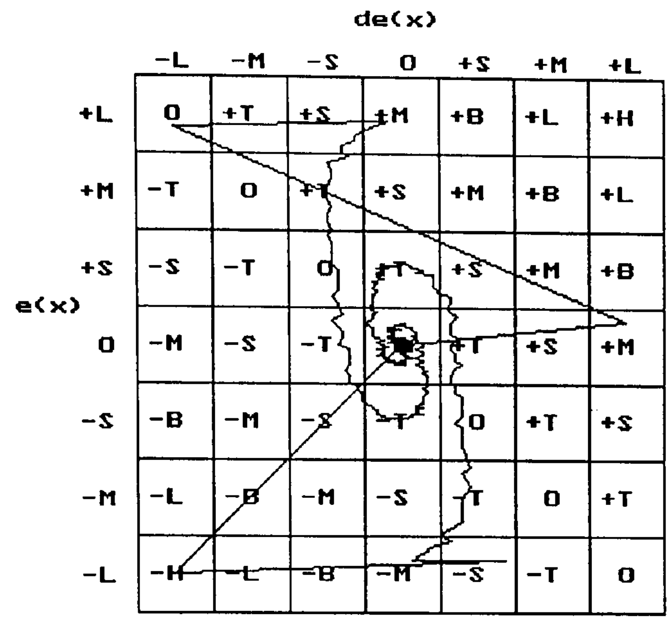


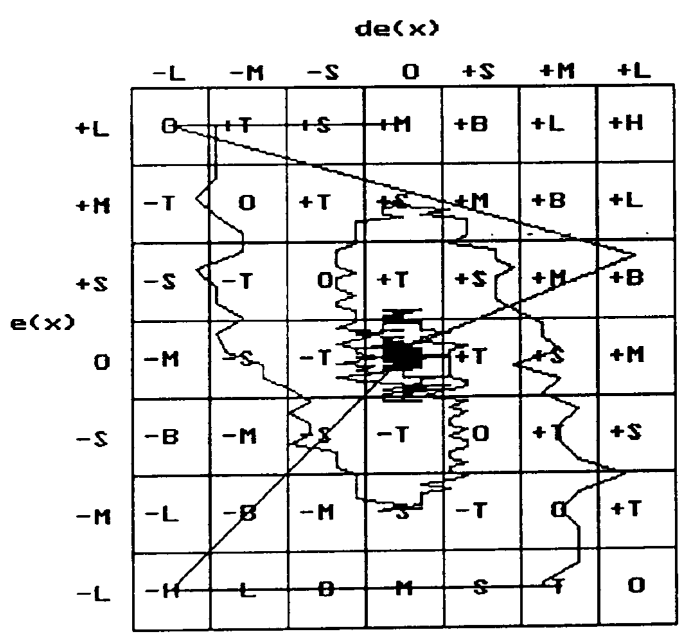
Figure 5-7(a). The output of a plant 1/(s+1)controlled by a fuzzy PI controller with different E, D and U values but keeping $K_p=3, K_i \cdot T_s=0.1$.



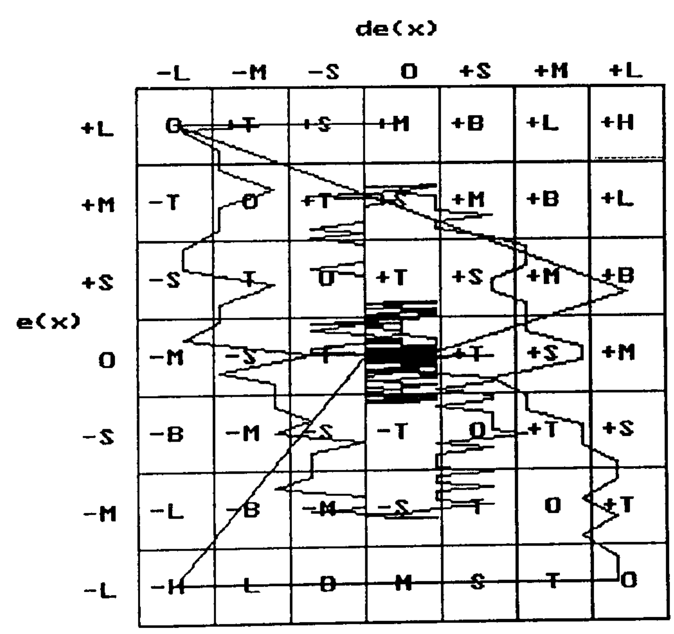
(b)The control trajectory of E=12V, D=0.4Vand U=2.4V.No error saturation points were recorded in this experiment.



(c) The control trajectory of E=3V, D=0.1V and U=0.6V.There were 3 positive and 2 negative error saturation points recorded.



(d)The control trajectory of E=0.75V, D=0.025Vand U=0.15V.There were 65 positive and 64 negative error saturation points recorded.



(e)The control trajectory of E=0.375V D=0.0125Vand U=0.075V.There were 121 positive and 119 negative saturation points recorded.

(2) The influence on the control action of decreasing the range D while keeping E sufficiently large.

The experimental results presented in Figure 5-4 show that the output waveforms of the fuzzy PI controller are similar to the output of a digital PI controller only when D is greater than 0.1V. When D is below this value the response tends to be heavily damped. This can be explained as follows. Assume that the maximum error signal e(k) caused by the step input change is at about the same value as the error range E. Then, decreasing the range D while keeping E constant will cause the starting point of the control trajectory on the control decision table to shift left into different output fuzzy sets. As shown in Figure 5-8, curve (1) represent the normal control trajectory produced by setting D=0.1Curve (2) and (3) are the results of decreasing the D value by 1/2 and 1/4. The starting position of these curves are different. Their corresponding control output fuzzy sets are +M, +Sand +T respectively. Therefore, curve (2) has less driving force

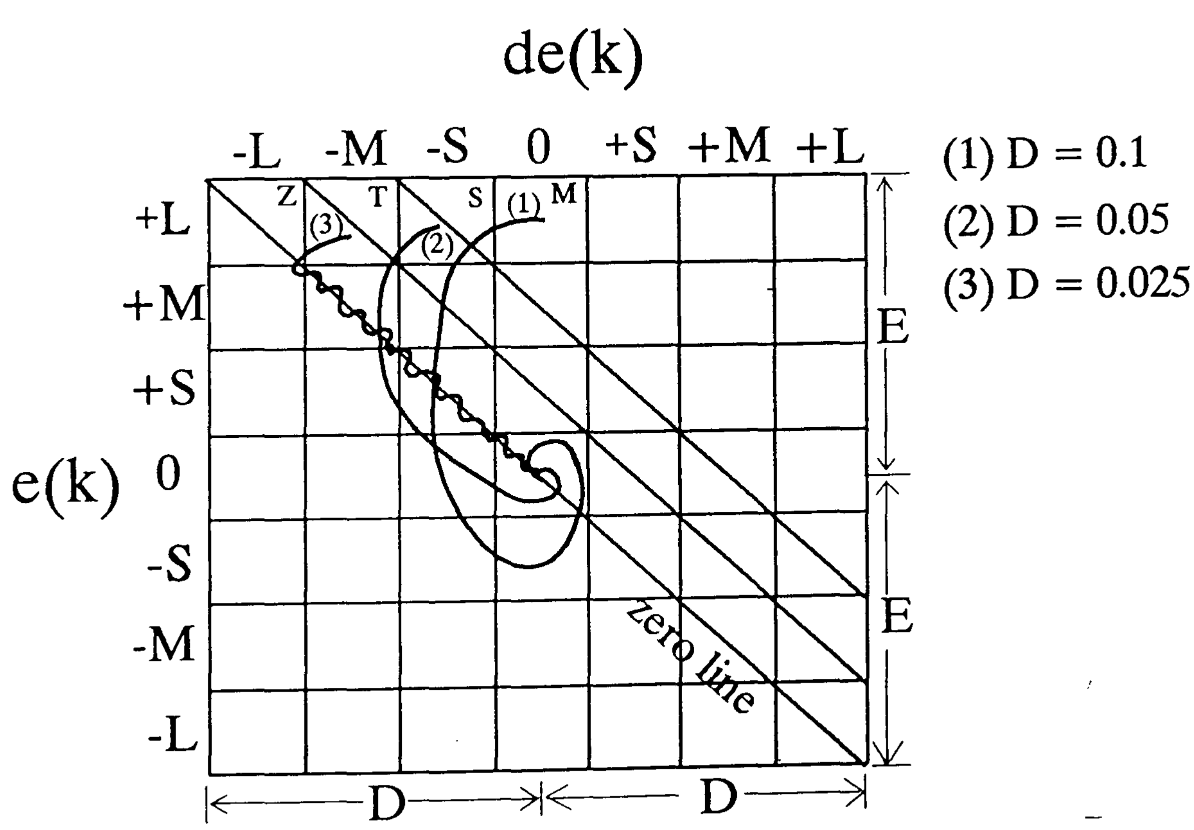
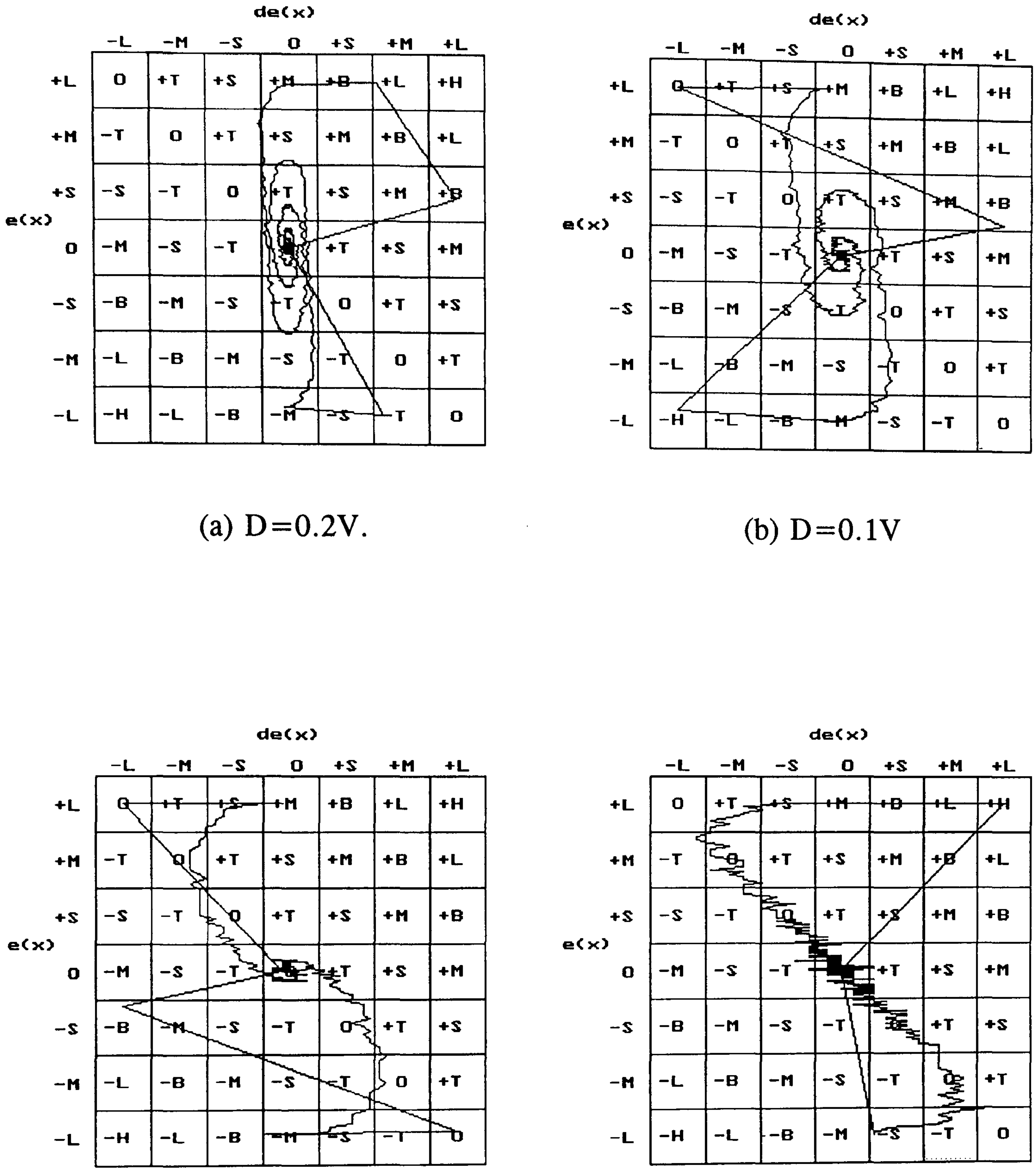


Figure 5-8. effect of decreasing the value of D range.



(a) D=0.2V. (b) D=0.1V

(c) D=0.05V (d) D=0.025V

Figure 5-9. The control trajectories of Figure 5-4.

than curve (1) and thus less overshoot. Moreover, the control output fuzzy set of curve (3) is +T, which is too weak to drive the system deep into the negative part of the control table. Then, it can only swing back and forth along the zero line and be driven to the origin in a manner similar to a sliding mode controller. Figure 5-9 shows the control trajectories corresponding to the experiment results of Figure 5-4, and confirms that the sliding control phenomenon does happen in Figure 5-9(d). This is analyzed further in section 5.4.

(3) <u>The damping effects caused by setting the error range E too large</u>

We have noticed that there is another damping effect occurring in Figure 5-5(a). The control trajectory of the third curve in Figure 5-5(a) whose error range E=7.5Vis shown in Figure 5-10(b). For comparison, the trajectory for E=3with the same D and U as in Figure 5-5(a) is also shown in Figure 5-10(a). Figure 5-10(b) shows that when E=7.5V,the largest e(k) which caused by the step input changes are about +3.1Vor -3.1V, therefore, it could only drive the control trajectory to the frame (0,+S)in the

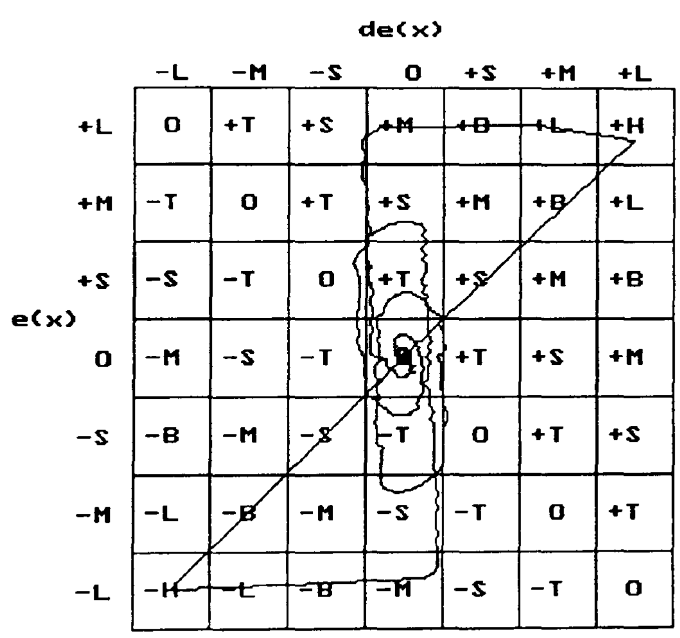


Figure 5-10(a) output fuzzy set is +Mwhen E=3V,D=0.2V and U=2.4.

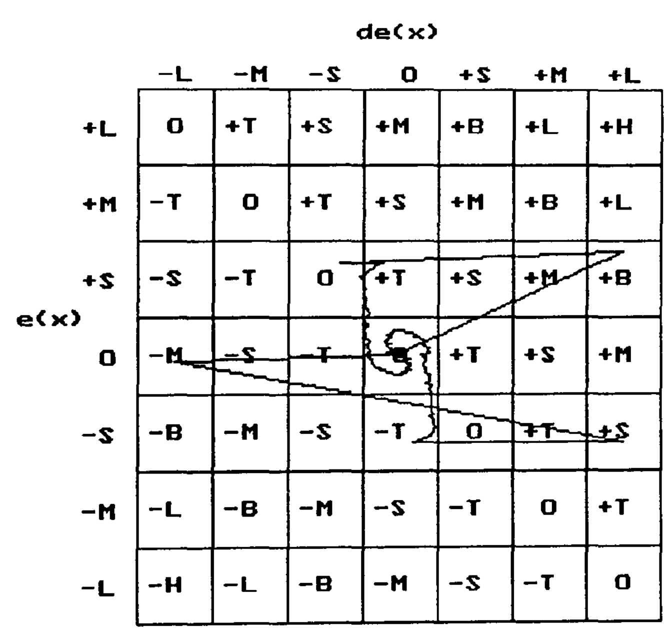


Figure 5-10(b) the output fuzzy set is +Twhen E=7.5V,D=0.2V and U=2.4.

second quadrant or (0,-S) in the fourth quadrant. In this case, the consequence fuzzy set for the positive output is +T(about 1/6 of +H)and -T for the negative output. However, in Figure 5-10(a), if E was set equal to 3V, the same input change will make the maximum positive frame reached in the second quadrant become (0,+L),for which the output fuzzy set is +M(about 1/2 of +H). Since the values of U are the same for both cases, E=7.5 will produce weaker controller outputs than E=3, hence, less overshoot or a more damped response is obtained.

The sliding control phenomenon could also happen in the case of the error range E having too large setting. This will be discussed in the next section.

5.4 Sliding control of a fuzzy PI controller caused by choosing too small a value of D or too large a value of E

5.4.1 The negative gain caused by choosing too small a value of D

As mentioned in section 5.3, when the range of error change D in a fuzzy PI controller is made too small, it results in a special sliding motion as shown in Figure 5-9d. The reason for this sliding characteristic and its theoretical background will now be discussed in more detail. The sliding motion phenomenon is often found in variable structure systems (VSSs). Usually, the structure of such a system is changed intentionally in accordance with a preset structure-control law. The control laws developed in the theory of VSS provide for changes in the structure of the system whenever the representative point (RP is a point which represent the position of a control trajectory in the phase plane) crosses certain surfaces (hypersurfaces) in the phase space of the system. We shall see later that the hypersurface is a straight line $x_1+a\cdot x_2=0$ in the phase plane where x_1 is the error and x_2 is the change of error. On each sides of this line the system structure is different and this will produce two phase plane trajectories with opposite directions in the neighborhood of the line. The system will move back and forth between the two regions divided by this line and gradually slide down along the line to the phase plane center.

Sliding motion control theories have been developed earlier by Itkis[60], Utkin[61] and several others[62,63,64]. Their concepts will be applied in the next section to study sliding motion control in a fuzzy controller. First, however, we shall consider the cause of the sudden change in the fuzzy PI control system structure.

Consider a fuzzy PI controller which has the control decision table shown in Figure 5-11. The error and error change ranges are set as E and D respectively. Then, the diagonal line L_1 with slope -c (c=E/D) will divide the table into two parts, the upper right part produces positive control output du while the lower left part generate negative du. If, at some instant, the control motion is located at the point (de_x,e_x), as shown in the figure, this will generate a positive du since, in this example, the point is in the upper right portion of the table. If e_x is held unchanged then the horizontal distance F from this point to the diagonal line L_1 can be written as

$$F = \left|\frac{e_x \cdot D}{E}\right| - |de_x| \qquad (5\text{-}3)$$

It can be seen that F>0 will generate positive du but F<0 will output a negative du. When F>0 the following condition exists

$$\left|\frac{de_x}{D}\right| < \left|\frac{e_x}{E}\right| \qquad (5\text{-}4)$$

Therefore, if we choose a new value D' which is too small to make the R.P moving into negative du part, then this inequality will no longer hold. This means that a point (de_x,e_x) in the fuzzy decision table above line L_1 should generate a positive du.

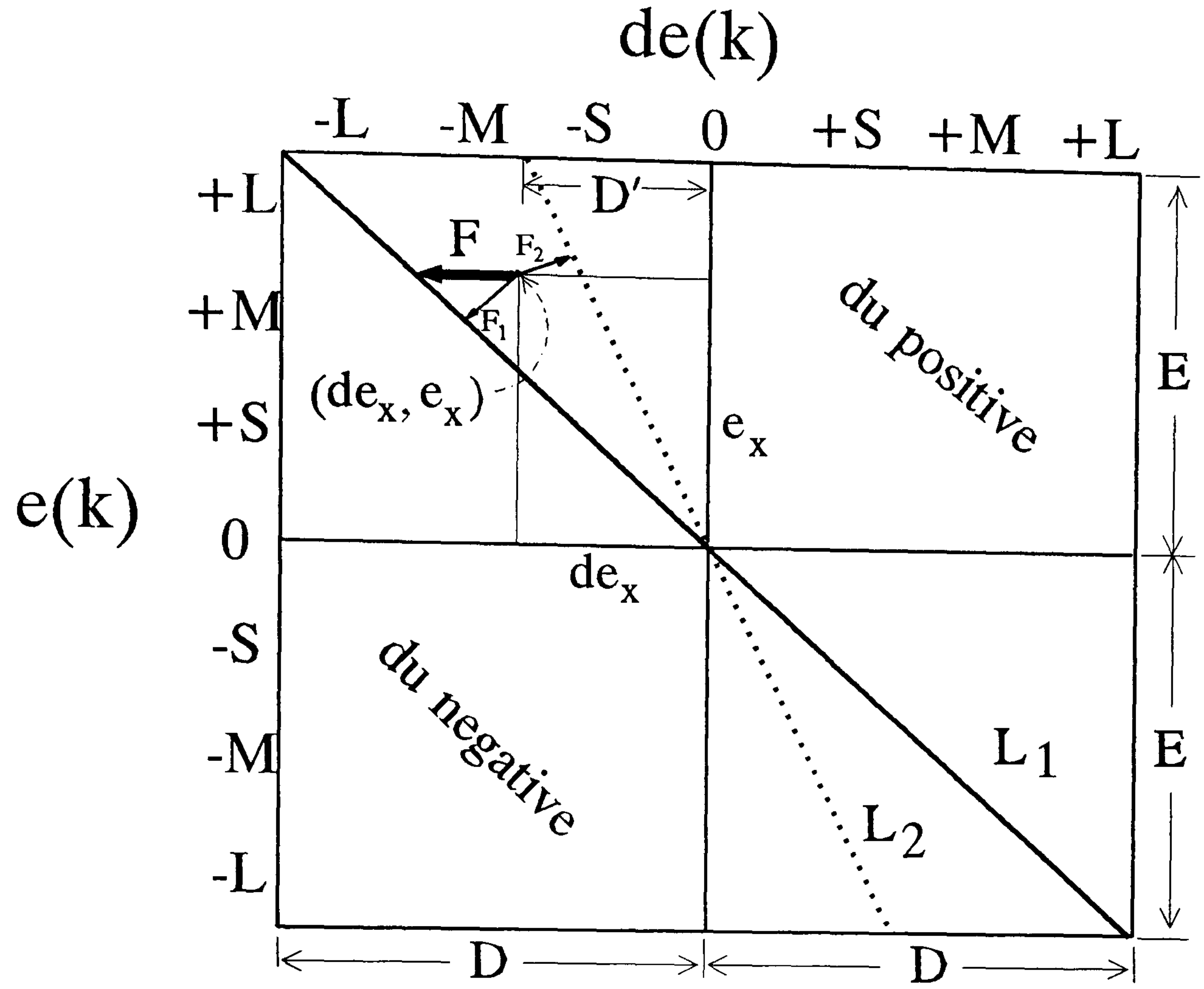
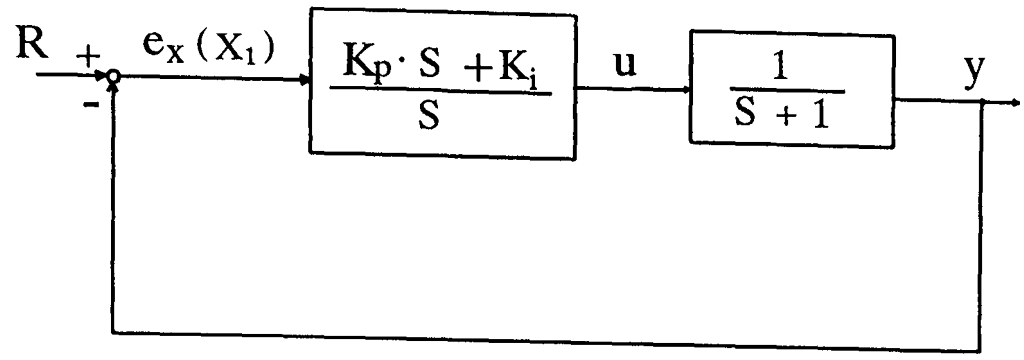


Figure 5-11. The negative sign caused by choosing too small value of D.

However, after scaling by this value D′ it falls into the negative part of the new table of range D′ as show in Figure 5-11 and thus generates a negative du. Consequently, before the control action moves across the L_2 line, the fuzzy controller behaves like a normal PI controller generating positive du. However, once it crosses line L_2 and moves into the region between line L_1 and L_2, it behaves as if a '-1' gain is instantaneously switched into the control loop and K_p is multiplied by K. Since the range E is unchanged and supposed to be a proper value, K_i remains the same. From chapter 4, the value of K is approximately equal to F_2/F_1 the ratio of the projected distance from line



Note : the error e_x is chosen as State variable X_1

Figure 5-12. The block diagram of a plant 1/(s+1)being controlled by a fuzzy PI controller with proper D range.

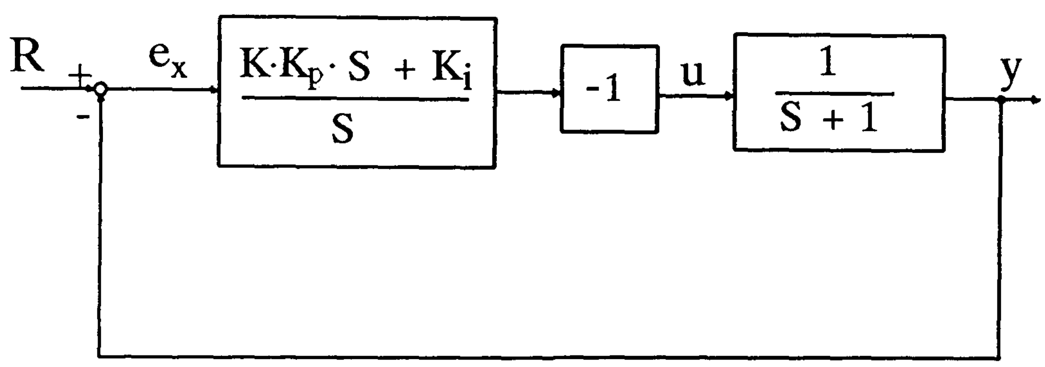


Figure 5-13. A plant 1/(s+1)is controlled by a fuzzy PI controller with too small value of D'.

L_2 and the distance from line L_1. Figure 5-12 and 5-13 represents the structural changes of this special phenomenon.

From equation (5-4) the range of D which makes the relation $K_p = \bar{U}/2D$ valid is given by

$$E \cdot \left|\frac{de_x}{e_x}\right| < D \qquad (5\text{-}5)$$

On the contrary, if D′ meets the following inequality, it is a range too small to locate the right control action point (de_x,e_x) on the decision table.

$$E \cdot \left|\frac{de_x}{e_x}\right| > D' \qquad (5\text{-}6)$$

Note that when F=0,no sliding regime is present and the slope of L_2 is equal to L_1. If the ratio of E:D is kept constant (that is the slope L_1 is kept constant) there will be no slide regime no matter how small D becomes. This system property was illustrated in Figure 5-7. The system is then only affected by the decision table saturation phenomenon which was discussed in the last section. Thus, instead of finding a minimum value for D or a maximum value for D′, it is more convenient to find the value of the E/D ratio that correspond to the start of the sliding regime, This ratio can be expressed by $K_p/(T_s \cdot K_i)$ from equation (4-15) and (4-16).

5.4.2 Theoretical explanation of the sliding mode control caused by the introduction of a negative gain in the controller to form a variable structure system (VSS).

Consider the plant of Figure 5-12, whose free motion is described by the following system of differential equations:

$$
\begin{cases}
\dfrac{dx_1}{dt} = x_2 & (5\text{-}7a) \\
\dfrac{dx_2}{dt} = -K_i x_1 - (K_p+1)x_2 & (5\text{-}7b)
\end{cases}
$$

where x_1 is the difference between the reference input R and the output signal y of the system, so, it is equal to the error e_x and x_2 is equal to the error change rate de_x/T_s. There are three phase-portrait types for this system, depending on the relative position of the roots of the characteristic equation:

$$p^2 + (K_p + 1)p + K_i = 0 \qquad (5\text{-}8)$$

The possible phase-portraits are as follows:

1. The roots are real, negative and distinct, say $\lambda_1 < \lambda_2 < 0$. The phase trajectories (Figure 5-14a) are parabolas, with two asymptotes:

$$\sigma_1 = x_1 + \frac{1}{\lambda_1} x_2 = 0 \qquad (5\text{-}9a)$$

$$\sigma_2 = x_1 + \frac{1}{\lambda_2} x_2 = 0 \qquad (5\text{-}9b)$$

The system is globally asymptotically stable (whatever the initial position of the representative point (RP) in the phase plane).

2. The roots are real, negative and equal, say $\lambda_1 = \lambda_2 = \lambda < 0$. The phase trajectories (Figure 5-14b) are convergent parabolas, but have only one asymptote:

$$\sigma = x_1 + \frac{1}{\lambda} x_2 = 0$$

3. The roots are complex with negative real parts: $Re\lambda_1$, $Re\ \lambda_2 < 0$. The phase trajectory as shown Figure 5-14(c), is a spiral. The system is globally asymptotically stable (oscillatory).

Each of the above portraits corresponds to a certain system structure that may be represented by the block diagram of Figure 5-14. Each portrait corresponds to a distinct set of values for K_p and K_i. All three systems are stable. However, if a negative gain -1 is included in the system loop, as shown in Figure 5-13, the whole system will become unstable and is described by the equations

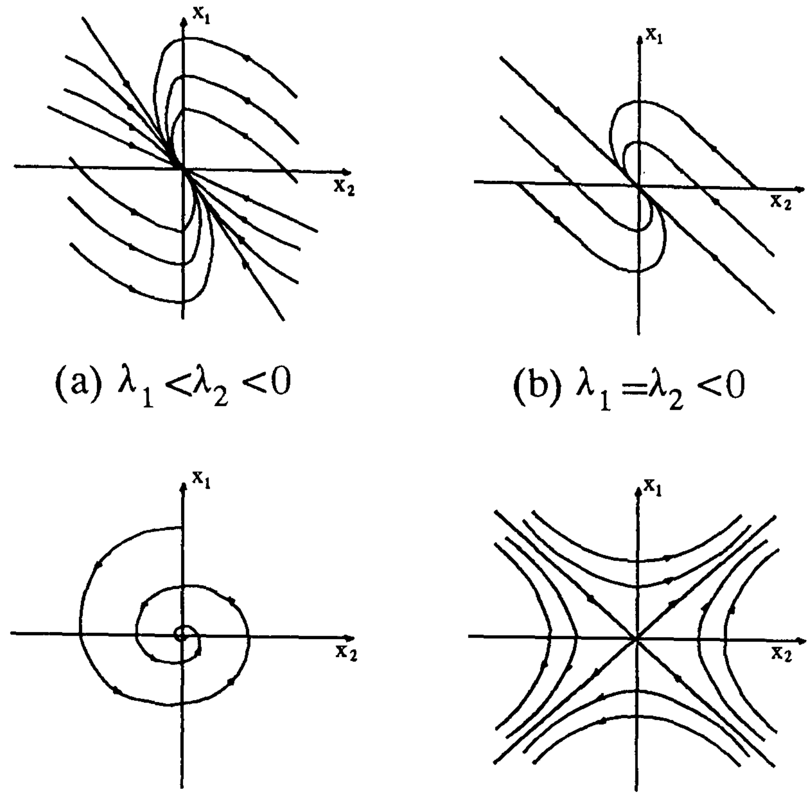
$$\begin{cases} \dfrac{dx_1}{dt} = x_2 & (5\text{-}10a) \\ \dfrac{dx_2}{dt} = K_i \cdot x_1 + (K_p \cdot K - 1)x_2 & (5\text{-}10b) \end{cases}$$

The characteristic equation becomes

$$p^2 + (1 - K_p \cdot K)p - K_i = 0 \qquad (5\text{-}11)$$

The roots are real and of unlike sign, $\lambda_1 < 0$ and $0 < \lambda_2$. The phase trajectories are hyperbolas (Figure 6-7d) with two asymptotes that are given by equation 5-9. The slopes of these are $(1/\lambda_1)$ and $(1/\lambda_2)$. The system is unstable, but there is a unique phase trajectory $(\sigma_1 = x_1 + (1/\lambda_1)x_2 = 0)$ along which the representative point (RP) moves asymptotically to the origin.

Now, consider the case where a plant $1/(s+1)$ is controlled by a fuzzy PI controller in which the D' (the range of the error change) and E (the range of the error) are chosen such that K_p and K_i produce two distinct negative real roots in its characteristic equation. This should have a spiral portrait in the phase plane as in Figure 5-14. However, if D' is reduced until it is too small, an additional gain of "-1" is included in the system loop whenever the representative point RP moves into the region



(a) $\lambda_1 < \lambda_2 < 0$ (b) $\lambda_1 = \lambda_2 < 0$

(c) $\mathrm{Re}\lambda_1, \mathrm{Re}\lambda_2 < 0$ (d) $\lambda_1 < 0, \lambda_2 > 0$

Figure 5-14 The phase portrait of the first order system (1/S+1) controlled by a fuzzy PI controller

between L_1 and L_2 (see Figure 5-11). Under these circumstances the phase trajectory will become hyperbolic since its characteristic equation (Equation 5-11) will have two real roots of different signs.

As the RP moves into and out of the region between L_1 and L_2 the plant structure will switch between these two modes of operation. L_2 acts as a switching line for the structure, since at the instant when the RP reaches this line the structure of the system switches instantaneously from spiral to hyperbolic. L_2 can be described by the equation

$$x_1 + \frac{T_s \cdot E}{D'} x_2 = 0 \qquad (5\text{-}12)$$

Note that T_s is introduced into this equation because it will be necessary for the calculation involving K_p and K_i.

It is obvious that L_1 is another switching line. Although we don't know the exact value of the proper range D, we do know that it lies it the same quadrant as L_2 and can be expressed as

$$x_1 + \frac{T_s \cdot E}{D} x_2 = 0 \qquad (5\text{-}13)$$

Since D' < D, the slopes of L_2 and L_1 have the following relation

$$\frac{T_s \cdot E}{D'} > \frac{T_s \cdot E}{D}$$

The phase plane is then divided by the two lines L_1 and L_2 into four regions (see Figure 5-15a). The structure being spiral in regions I and III, and hyperbolic in region II and IV. Then, the sliding control phenomenon will be decided by how the asymptotes $\sigma = x_1 + (1/\lambda_1)x_2 = 0$ lie in the phase plane. These four regions are specified by the following conditions

$$\begin{array}{lll}
\textit{Region I:} & x_1 + \frac{T_s \cdot E}{D'} x_2 > 0, & x_1 + \frac{T_s \cdot E}{D} x_2 > 0 \\
\textit{Region II:} & x_1 + \frac{T_s \cdot E}{D'} x_2 < 0, & x_1 + \frac{T_s \cdot E}{D} x_2 > 0 \\
\textit{Region III:} & x_1 + \frac{T_s \cdot E}{D'} x_2 < 0, & x_1 + \frac{T_s \cdot E}{D} x_2 < 0 \\
\textit{Region IV:} & x_1 + \frac{T_s \cdot E}{D'} x_2 > 0, & x_1 + \frac{T_s \cdot E}{D} x_2 < 0
\end{array}$$

We first consider the case $| T_s \cdot E/D' | < | (1/\lambda_1) |$. The asymptote $\sigma = x_1 + (1/\lambda_1)x_2 = 0$ lies entirely in the union of region I and III. Therefore, the RP, starting out from the x_1 axis of region I, passes through the line $\sigma = 0$ and reaches L_2. Then, at that instant the system structure switches from spiral to hyperbolic, the RP continues to move in region II along an arc of a hyperbola that deviates from the asymptote (Figure 5-15b). Consequently, at a certain time later the RP will reach the switching line L_1, the structure again switches back to a spiral behavior. The same process happens from region III to region IV, so the process repeats itself producing periodic oscillations.

Figure 5-15b also shows that in this region not only the spiral trajectory moves toward the origin but the hyperbola trajectory tends to approach the origin. Therefore, these oscillations will be damped and the system is globally asymptotically stable.

Now consider the second case when the asymptote $\sigma = x_1 + (1/\lambda_1)x_2 = 0$ lies between L_1 and L_2, i.e. $| T_s \cdot E/D | < | (1/\lambda_1) | < | T_s \cdot E/D' |$ and the asymptote lies entirely in the union of region II and IV. Therefore, the RP, starting out from the x_1 axis of region I, passes through the line L_2 before reaching the line $\sigma=0$. Then the motion

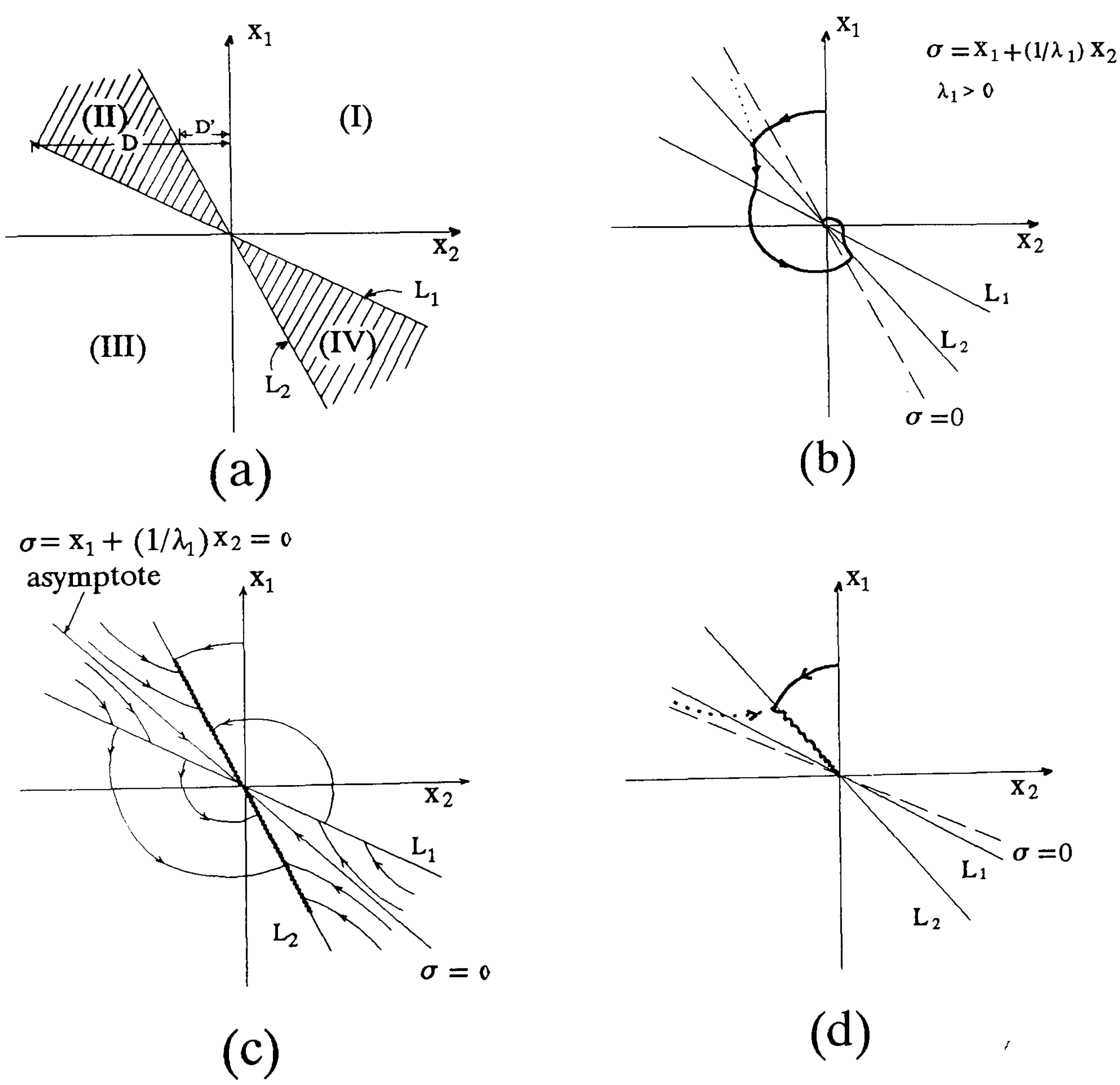


Figure 5-15. Switching lines and asymptotes in the phase plane.

of the VSS takes place in a special regime, called a sliding regime: the system structure switches back and forth at very high frequency and the RP performs infinitesimal oscillations about the switching line L_2 (see Figure 5-15(c)) and moves toward the origin. The reason that this regime appears is that the phase trajectories in spiral and hyperbolic structures have opposite directions in the neighborhood of the switching line L_2. The mathematical existence conditions for this sliding regime in a fuzzy PI controller case will be discussed in the next section.

Finally consider the case $|(1/\lambda_1)|<|T_s \cdot E/D|$. The asymptotes $\sigma=x_1+(1/\lambda_1)x_2=0$ lies entirely in the union of region III and I. As shown in Figure 5-15d, the sliding motion happens in the same way as when $|T_s \cdot E/D| < |(1/\lambda_1)| < |T_s \cdot E/D'|$.

5.4.3 Mathematical existence conditions for a sliding regime

We have explained the formation of a sliding regime in a fuzzy PI controller when D' is chosen as a too small value to control a plant with transfer function $1/(s+1)$. In this section, the mathematical existence conditions for it will be discussed in more detail.

The following inequality (5-14) was first suggested by Dolgolenko (1952)[58] as a necessary and sufficient condition for the existence of a sliding regime

$$\lim_{\sigma \to +0} \frac{d\sigma}{dt} \leq 0 \leq \lim_{\sigma \to -0} \frac{d\sigma}{dt} \qquad (5\text{-}14)$$

Its equivalent is

$$\lim_{\sigma \to 0} \sigma \frac{d\sigma}{dt} \leq 0 \qquad (5\text{-}15)$$

Now, let us apply equation (5-14) to verify the existence of a sliding regime for the special fuzzy PI controllers discussed in this chapter. Starting from region I, before reaching the switching line L_2 the system structure (Figure 5-12) is represented by the set of equations (5-7a) and (5-7b). The switching line L_2 is described by

$$\sigma = x_1 + \frac{T_s \cdot E}{D'} x_2 = 0 \qquad (5\text{-}16)$$

Then, the derivative of σ along the trajectories of system (5-7a) and (5-7b) can be found as

$$\begin{aligned} \frac{d\sigma}{dt} &= \frac{d}{dt}\left(x_1 + \frac{T_s \cdot E}{D'} x_2\right) \\ &= \frac{dx_1}{dt} + \frac{T_s \cdot E}{D'} \frac{dx_2}{dt} \\ &= x_2 + \frac{T_s \cdot E}{D'}[-K_i x_1 - (K_p + 1)x_2] \qquad (5\text{-}17) \end{aligned}$$

At a point on the switching line $\sigma=0$, from equation (5-16)

$$x_1 = -\frac{T_s \cdot E}{D'} x_2 \qquad (5\text{-}18)$$

and so equation (5-17) becomes

$$\begin{aligned} \lim_{\sigma \to +0} \frac{d\sigma}{dt} &= x_2 + \frac{T_s \cdot E}{D'}[K_i \frac{T_s \cdot E}{D'} x_2 - (K_p + 1)x_2] \\ &= x_2 \left\{ 1 + \frac{T_s \cdot E}{D'}[K_i \frac{T_s \cdot E}{D'} - (K_p + 1)]\right\} \qquad (5\text{-}19) \end{aligned}$$

For the sliding regime existence inequality (5-14) to hold, equation (5-19) must be equal to a negative value. Therefore, since x_2 is negative in region II, the following inequality should be maintained:

$$\frac{T_s \cdot E}{D'}[K_i \frac{T_s \cdot E}{D'} - (K_p + 1)] > -1 \qquad (5\text{-}20)$$

If $c' = (T_s \cdot E)/D'$ then this inequality is

$$c' \cdot K_i + \frac{1}{c'} > K_p + 1 \qquad (5\text{-}21)$$

For either of the phase-portrait types for the system of Figure 5-12, this inequality will guarantee that the RP will move toward the switching line $\sigma=0$ from the upper side of $\sigma>0$.

For the case of the system described by equations (5-10a) and (5-10b), the derivative of σ along its hyperbola trajectory in the region II is

$$\begin{aligned}
\lim_{\sigma\to -0} \frac{d\sigma}{dt} &= \frac{dx_1}{dt} + \frac{T_s \cdot E}{D'}\frac{dx_2}{dt} \\
&= x_2 + \frac{T_s \cdot E}{D'}[(K_pK - 1)x_2 + K_i x_1] \\
&= x_2 + \frac{T_s \cdot E}{D'}[(K_pK - 1)x_2 - K_i \frac{T_s \cdot E}{D'} x_2] \\
&= x_2 \{ 1 + \frac{T_s \cdot E}{D'}[(K_pK - 1) - K_i \frac{T_s \cdot E}{D'}]\} \qquad (5\text{-}22)
\end{aligned}$$

Since x_2 is negative, if we want equation (5-22) to be positive to meet the sliding regime existence condition, then, the following inequality must hold

$$\frac{T_s \cdot E}{D'}[(K_pK - 1) - K_i \frac{T_s \cdot E}{D'} < -1 \qquad (5\text{-}23)$$

Again, if $c' = (T_s \cdot E)/D'$ this inequality can be rewritten as

$$c' \cdot K_i - \frac{1}{c'} > K_pK - 1 \qquad (5\text{-}24)$$

Obviously, this inequality will not hold either for too small a value of c', if its right hand side is positive or greater than -1, or too large a value of K. In section 5.4.1 it has been mentioned that the value of K is approximately equal to F_2/F_1, the ratio of the projected distance from line L_2 and the distance from line L_1. Too large value of K that means the control action points may exceed the unknown proper line L_1 or simply the sliding regime existence condition is failed. The value of K is approximately

$$K = \frac{F_2}{F_1} = \frac{(|de_x| - |\frac{e_x D'}{E}|)\sqrt{D^2 + E^2}}{(|\frac{e_x D}{E}| - |de_x|)\sqrt{D'^2 + E^2}} \qquad (5\text{-}25)$$

here

$$|\frac{e_x D'}{E}| < |de_x| < |\frac{e_x D}{E}|$$

Since the switching of the system structure always happens in the neighborhood of the hypersurface, i.e. $de_x \approx |(e_x \cdot D')/E|$ and K is usually a small positive number.

Now, let us verify the inequality (5-21) and (5-24) by those sliding regime cases occurring in Figure 5-4, and see what value of K, described by equation (5-25), will insure that the inequality (5-24) always holds, i.e. $K < K_M$ where

$$K_M = \frac{1 + K_i c' - \frac{1}{c'}}{K_p} \qquad (5\text{-}26)$$

The following Table 5-1 shows the results of applying those values of E, D and the corresponding K_p and K_i of Figure 5-4a and 5-4b to the calculations of inequality (5-21) and (5-24). The value of K_M is also listed. Since sliding motion can only happen when $K_M > K > 0$, the table shows that when D=0.1and 0.2, the calculated K_M will be negative. Therefore, there are no sliding regimes in these cases as shown in Figure 5-9(b) and 5-9(c). However, according to the definition $C' = \bar{T}_s \cdot E/D'$, C' increases with decreasing D'. Thus, if we decrease the D range to 0.05 or 0.025, the corresponding C' are 0.21 and 0.42 and the calculated K_M now become positive which will make the sliding regime possible. Actually, Figure 5-9(e) shows that sliding motion did happen when D was decreased to 0.025.

$Ts = 0.0035 \quad K_i = 28.57$
$E = 3$

D	C'	$C' K_i + \frac{1}{C'}$	$1 + C' K_i - \frac{1}{C'}$	$K_p + 1$	K_M
0.025	0.42	14.381	10.620	12 + 1	0.885
0.05	0.21	10.762	2.238	6 + 1	0.373
0.1	0.105	12.524	-5.524	3 + 1	- 1.841
0.2	0.0525	20.548	-16.548	1.5 + 1	-11.032

Table 5-1. The C' and K_M calculated for Figure 5-9.

Now, applying the same C' and K_M calculation procedures for Figure 5-7, which was a series of experiments done under the condition maintaining $K_i{\cdot}T_s=0.1$ and $K_p=3$ as constant and decreasing the U, E and D ranges by the same factor, produces the results shown in Table 5-2. Since C' are the same, for these four cases, the calculated K_M are all equal to -1.841, i.e. no sliding regime exists. Figure 5-7 confirms this is true.

$Ts = 0.0035, \ K_i = 28.57, \ K_P = 3$

U	E	D	C'	$C' K_i + \frac{1}{C'}$	$1 + C' K_i - \frac{1}{C'}$	K_M
2.4	12	0.4	0.105	12.524	-5.524	-1.841
0.6	3	0.1				
0.15	0.75	0.025				
0.075	0.375	0.0125				

Table 5-2. C' and K_M calculations for Figure 5-7.

It has been pointed out in section 5.4.1 that the E:D ratio is convenient to represent the sliding regime boundary. Since $C'=Ts{\cdot}E/D$ and the sampling time interval T_s is a constant, the C' can also be used to represent this E:D ratio. From the results of Table 5-1 and Table 5-2, other than the value K_M, C' is indeed another suitable index to show the existence of a sliding regime. Consequently, we can draw the conclusion that no sliding regime will exist if $C' \leq 0.106$ and sliding will start when $C' \geq 0.21$.

5.4.4 The sliding regime caused by choosing too large a value of E

As mentioned in section 5.4.1,the slope of the switching line L_2 is different from the proper line L_1 when too small a value of D' is chosen while keeping the error range E constant. However, obviously we can get the same slope by choosing a very large E' while keeping D constant. A similar analysis has, as its starting point, the decision table shown in Figure 5-16.

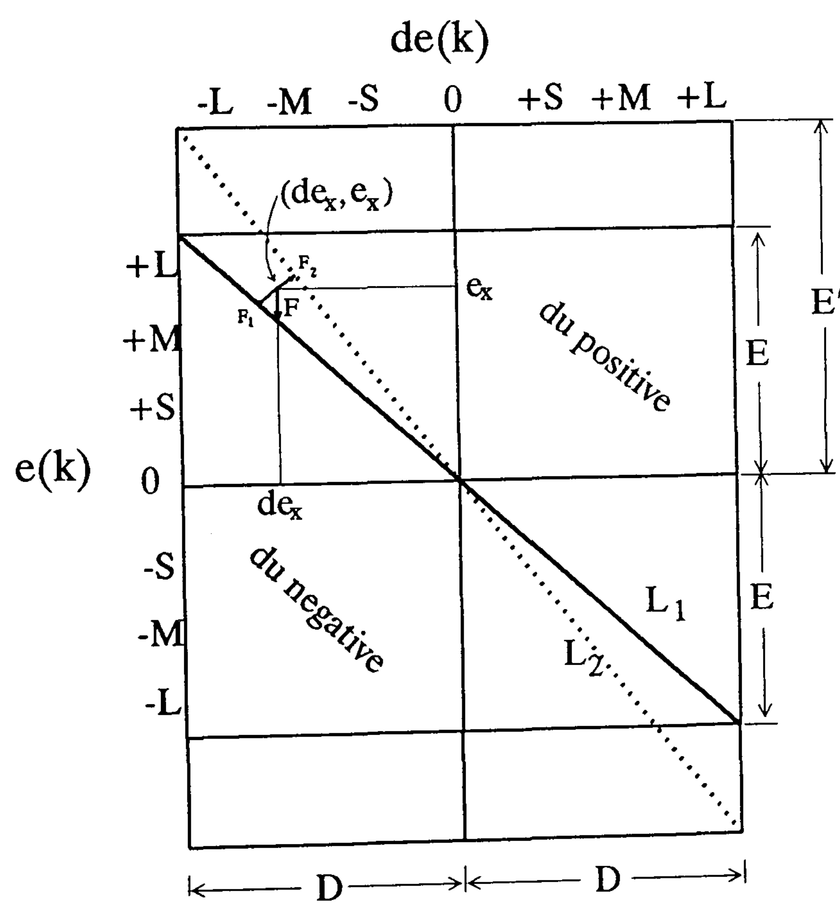


Figure 5-16. The negative sign caused by setting the error range E too large.

In Figure 5-16, at point (de_x, e_x), this time if de_x is held fixed then the vertical distance F from this point to the diagonal line L_1 can be written as

$$F = |e_x| - \left|\frac{de_x \cdot E}{D}\right| \qquad (5\text{-}27)$$

Therefore, if it is in the normal condition, F>0 and the controller output du will have a positive value. But when E' is too large, as shown in Figure 5-16, du becomes negative and F<0. So, sliding control may exist if the decision table ranges meet the following condition.

$$D \cdot \left|\frac{e_x}{de_x}\right| < E' \qquad (5\text{-}28)$$

K_p remains the same value because the D range is constant,, but now K_i must be multiplied by a factor K according to the following relation

$$K = \frac{F_2}{F_1} = \frac{\left(\left|\frac{de_x E'}{D}\right| - |e_x|\right)\sqrt{D^2 + E^2}}{\left(|e_x| - \left|\frac{de_x E}{D}\right|\right)\sqrt{D^2 + E'^2}} \qquad (5\text{-}29)$$

here

$$\left|\frac{de_x E'}{D}\right| > |e_x| > \left|\frac{de_x E}{D}\right|$$

Therefore, similar to Figure 5-4 and 5-5, when a plant 1/(s+1) is controlled by a fuzzy PI controller of which the error range E is set too large, each time its control action moves across the line L_2, before reaching L_1 its structure will change to the one shown in Figure 5-17. Note that as in Figure 5-5, a gain " -1" is included in the system loop

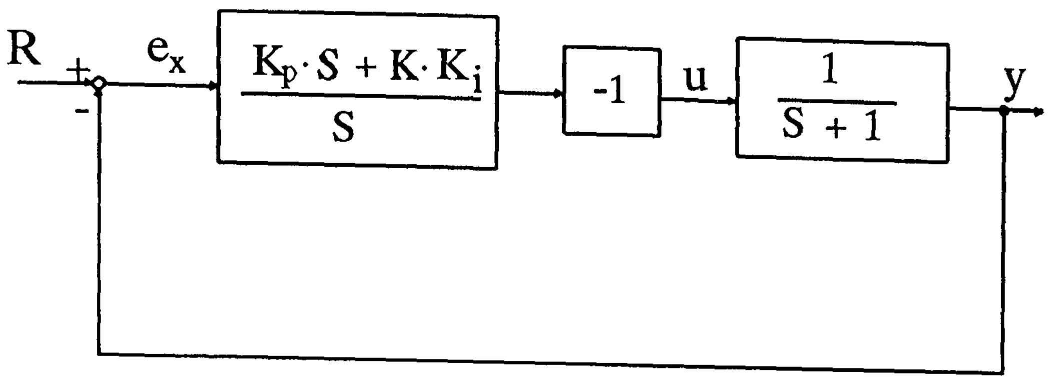


Figure 5-17. the new structure of a plant 1/(s+1)controlled by a fuzzy PI controller with too large E′.

to indicate that the controller output du is negative. The whole system thus become unstable and is described by the equations

$$\begin{cases} \dfrac{dx_1}{dt} = x_2 & (5\text{-}30a) \\ \dfrac{dx_2}{dt} = K \cdot K_i \cdot x_1 + (K_p - 1)x_2 & (5\text{-}30b) \end{cases}$$

The roots of its characteristic equation are real and of unlike sign. Therefore, its phase trajectories are hyperbolas (Figure 5-14(d)) with two asymptotes that are given by equation (5-9). Applying equation 5-30 (a) and (b) to the inequalities (5-14), which defined the existence conditions for a sliding regime, a boundary condition for K and C′ is found by produces similar procedures to those used to derive equation (5-26). The condition is

$$c' \cdot K_i \cdot K - \frac{1}{c'} > K_p - 1 \qquad (5\text{-}31)$$

Therefore, for the existence of a sliding regime, we need $K > K_m$,

$$K_m = \frac{K_p - 1 + \frac{1}{C'}}{C' \cdot K_i} \qquad (5\text{-}32)$$

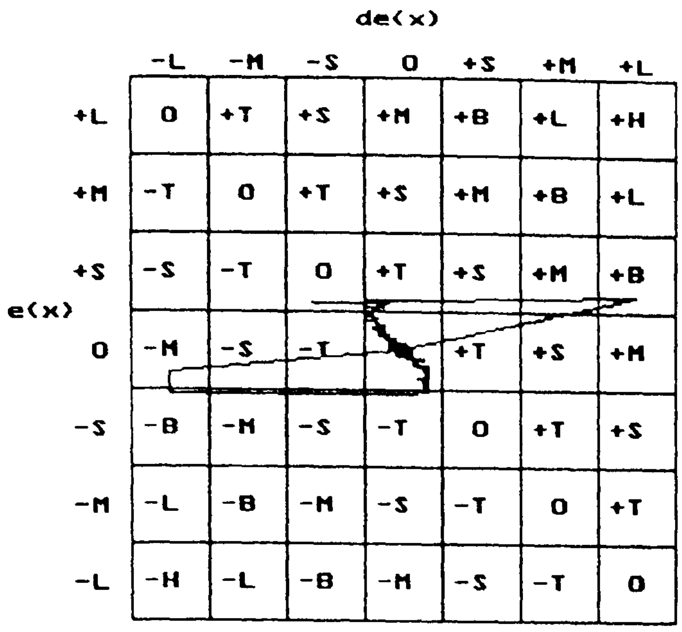
For some large value of error range E, table 5-3 shows the values of C' and K_m calculated by equation (5-32) and Figure 5-18 shows the corresponding control action trajectories. In the two cases where $E=15, K_m \leq 1.468$, i.e. K can exceed this small value, so, clearly they are in a sliding regime and the trajectories slide down to the origin along the diagonal line. However, in the cases where E=12 and E=7.5, $K_m \geq 2.103$, i.e. a value too large to be exceeded by K, or the control action points are too far from the line L_2 to meet the existence condition.

Other than comparing the K_m, if we look into the calculated C', in Table 5-3, then a result similar to the previous case of too small a value of D' is found. The sliding control started when C' is greater than approximately 0.26 and this phenomenon ceased when it is less than 0.132.

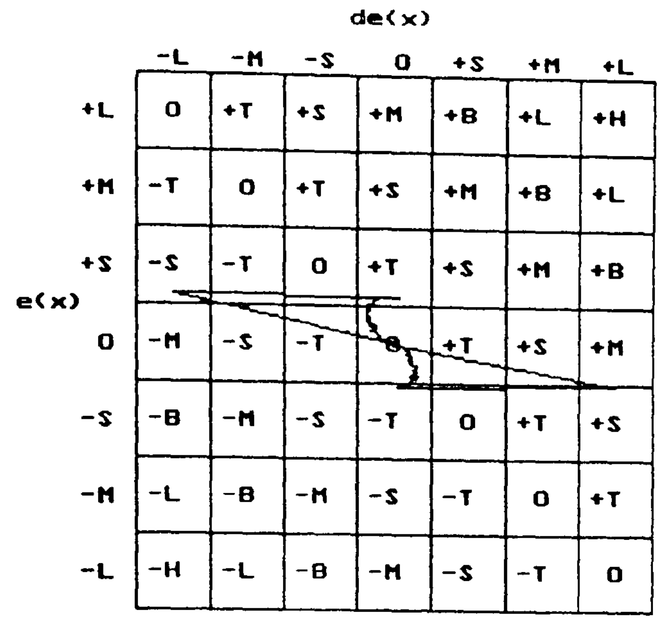
Ts = 0.0035 ,

E	D	U	K_p	K_i	C'	$K_p - 1 + \frac{1}{C'}$	K_m
15	0.1	1.2	6	11.43	0.525	6.905	1.151
15	0.2	2.4	6	22.86	0.263	8.81	1.468
7.5	0.2	2.4	6	45.71	0.131	12.619	2.103
12	0.4	2.4	3	28.57	0.105	11.524	3.841

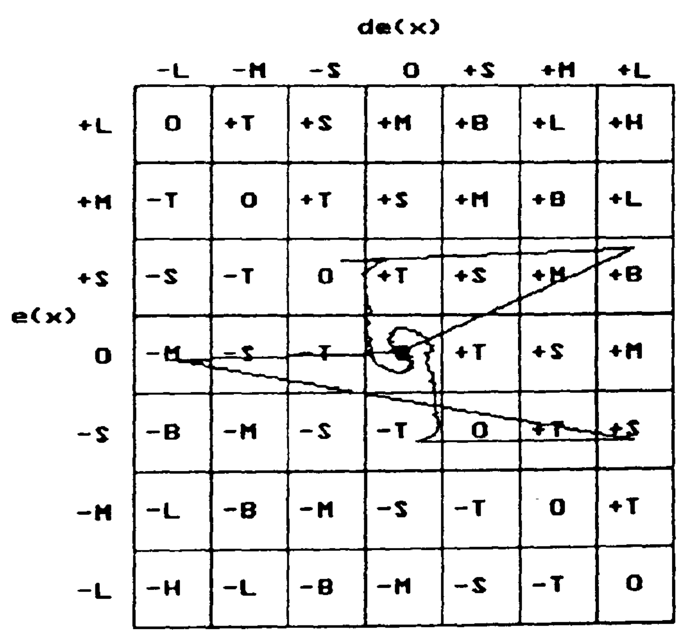
Table 3. calculation of K_m and C's for the cases of too large E'.



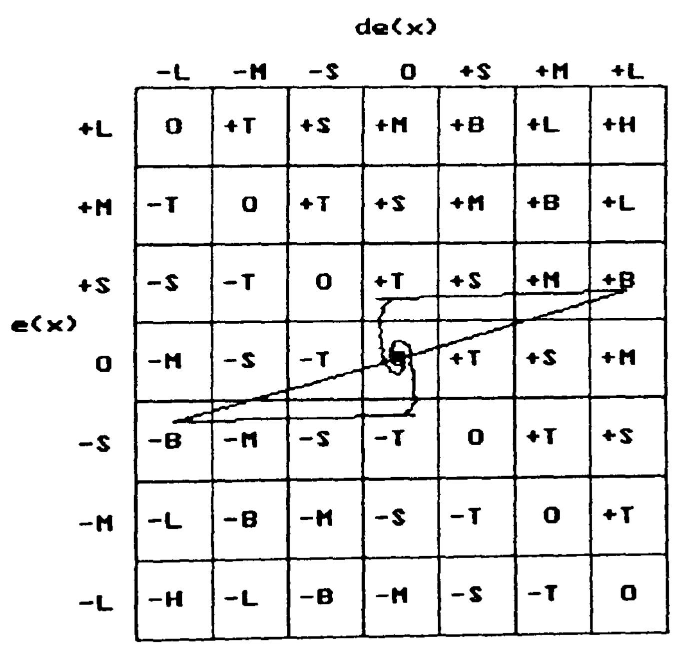
(a) The control trajectory of a plant 1/(s+1) controlled by a fuzzy PI controller with E=15,D=0.1 and U=1.2. (C'=0.525, K_m=1.151)



(b) the results of setting E=15, D=0.2 and U=2.4 for the fuzzy PI controller. (C'=0.2625, K_m=1.468)



(c) The result of setting E=7.5 D=0.2 and U=2.4 for the fuzzy PI controller. (C'=0.131, K_m=2.103)



(d) result of setting E=12, D=0.4 and U=2.4 for the PI controller. (C'=0.105, K_m=3.841)

Figure 5-18. The control action trajectories corresponding to Table 5-3.

5.4.5 The sliding regime for a general first order plant controlled by a fuzzy PI controller

Applying the same procedures that we used in section 5.4.3 and 5.4.4 to a general first order plant with transfer function a/(s+b), the existence conditions for a sliding regime can be found. The boundary conditions for K and C′ for the too small value of D range case are

$$a \cdot K_p \cdot K < b + a \cdot K_i \cdot c' - \frac{1}{c'} \tag{5-33}$$

$$K_M = \frac{b + a \cdot K_i \cdot c' - \frac{1}{c'}}{a \cdot K_p} \tag{5-34}$$

Therefore, for the existence of a sliding regime we need $0<K<K_M$. The K_M and C′ of a plant 10/(s+10) controlled by a fuzzy PI controller with E=3,D=0.03 and U=0.3 are calculated as shown in Table 5-4. Note that the value C′=0.35 is just a little smaller than the first C′ of Table 5-1. The sliding regime does exist since K_M is positive, and its control action trajectory is shown in Figure 5-21(b).

Ts = 0.0035
a = 10, b = 10,

E	D	U	K_P	K_i	C′	$b + a K_i C' - \frac{1}{C'}$	K_M
3	0.03	0.3	5	14.29	0.35	57.14	1.143

Table 5-4. K_M and C′ calculation for the plant 10/(s+10) controlled by a fuzzy PI controller.

5.5 A Fuzzy integral controller

In the problem of controlling an induction motor, Liaw & Wang[54] tried to surpress the tendency for a limit cycle to occur around the set-point by replacing the fuzzy controller with an integral controller. This integral controller was provided with the capability of error adaptation by setting

$$K_i = 15 - 20 \cdot |e|$$

However, in this section a further study about the decision table scaling ranges will show that any fuzzy PI controller can be easily changed to a fuzzy integral controller without any additional circuitry.

It is a special case for a fuzzy PI controller to set the ranges D or E to an extremely large value. If D -> ∞, after scaling, de_x/D will approach zero for all values of finite de_x, i.e. the control trajectory can move only along the vertical zero quantization level line in the center of the control decision table. This makes the system behave as if $de_x = 0$. When this happens, the inference result will only be decided by the error e_x; therefore, the fuzzy controller becomes an integral controller represented by half of equation (4-11) as

$$du_x = \frac{P}{\sqrt{2}} \left(\frac{e_x}{E} \right) \qquad (5\text{-}35)$$

At the extremes $e_x = \pm E$, the line $de_x = 0$ ends at the controller output quantization levels '$\pm$M', i.e. at the middle of the output range U. So, with e_x=Eand du_x set equal to U/2, from equation (5-35) the constant P will be

$$P = \frac{U}{\sqrt{2}} \qquad (5\text{-}36)$$

and the integral constant $T_s \cdot K_i$ is the same as in equation (4-15)

$$T_s \cdot K_i = \frac{U}{2 \cdot E} \qquad (5\text{-}37)$$

Figure 5-19 shows the results of a fuzzy PI controller in which D is set equal to 20,000,000V to control a plant with transfer function 10/(S+10).This plant is chosen because it is easier to see its overshooting and damping than for the plant 1/(s+1). Actually, from the relation (5-37), this fuzzy integral controller looks like a digital integral controller. The effects of the output scaling range U on system response is similar to the effects of the k_i constant of an digital I controller. However, the former had a longer rise time and smaller overshoot than the latter.

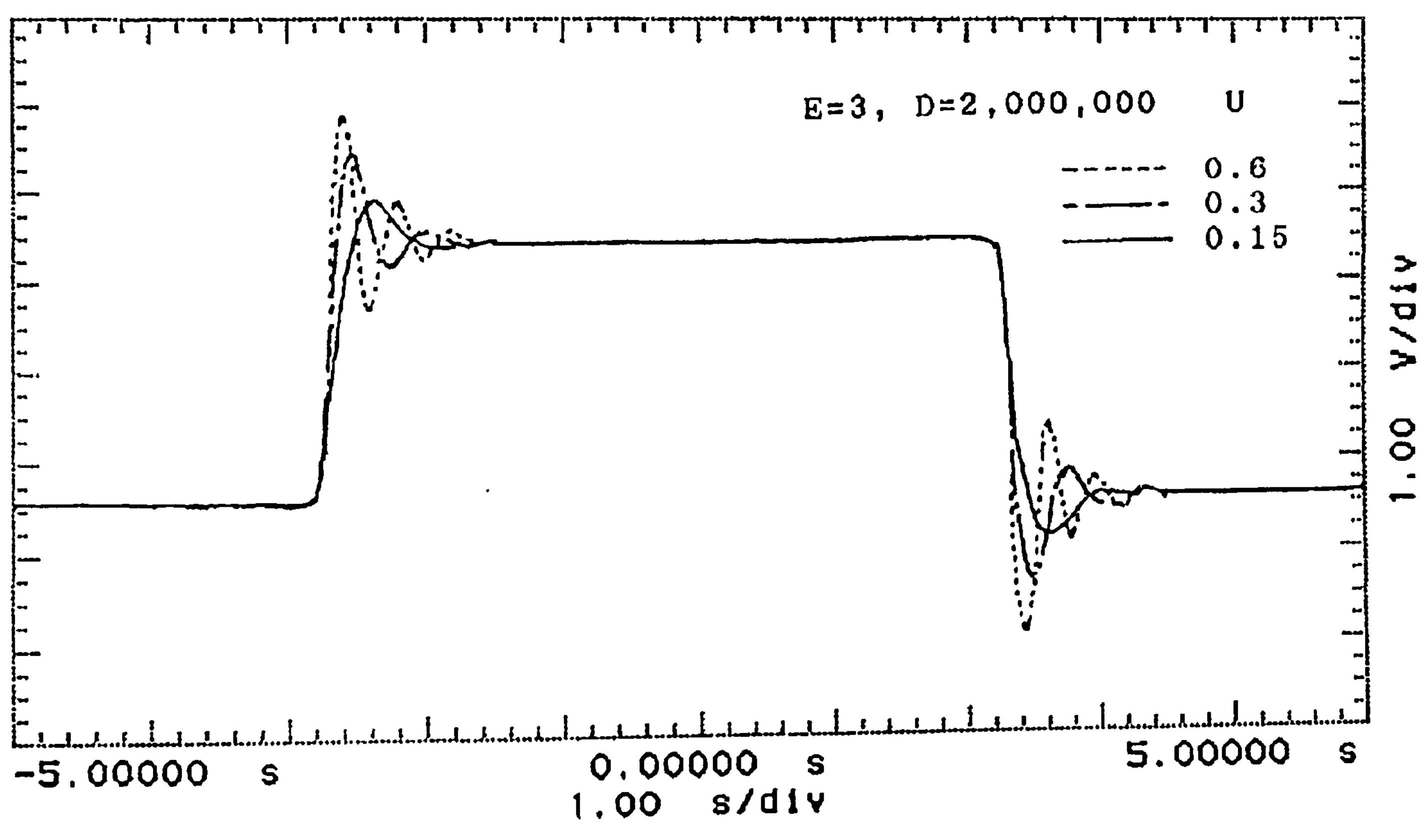


Figure 5-19(a) The outputs of a plant with transfer function 10/(s+10)is driven by a fuzzy I controller.

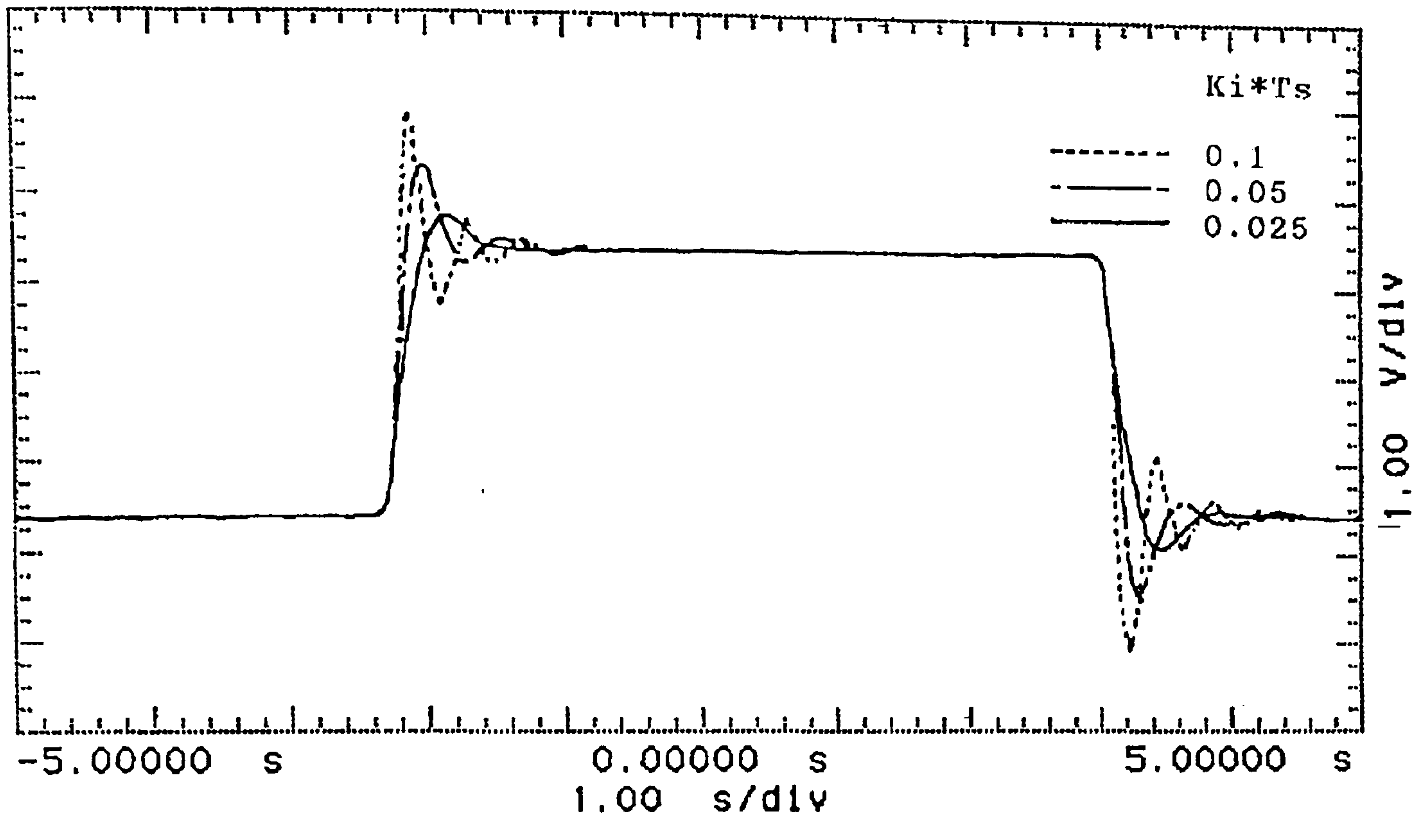


Figure 5-19(b) The output of a plant 10/(s+10)driven by a digital I controller.

5.6 <u>Modification of the fuzzy PI controller by temporarily switching the decision table range to a new value</u>.

(1) <u>Narrowing the membership functions near the set point</u>

When control action is close to the set point, sometimes a group of narrow membership functions are needed to enable fine control. In this experiment, however, the same triangular membership functions are used for all quantization levels, and the narrow membership function effect is obtained, temporarily by shrinking the decision table ranges. Figure 5-20 shows how this is

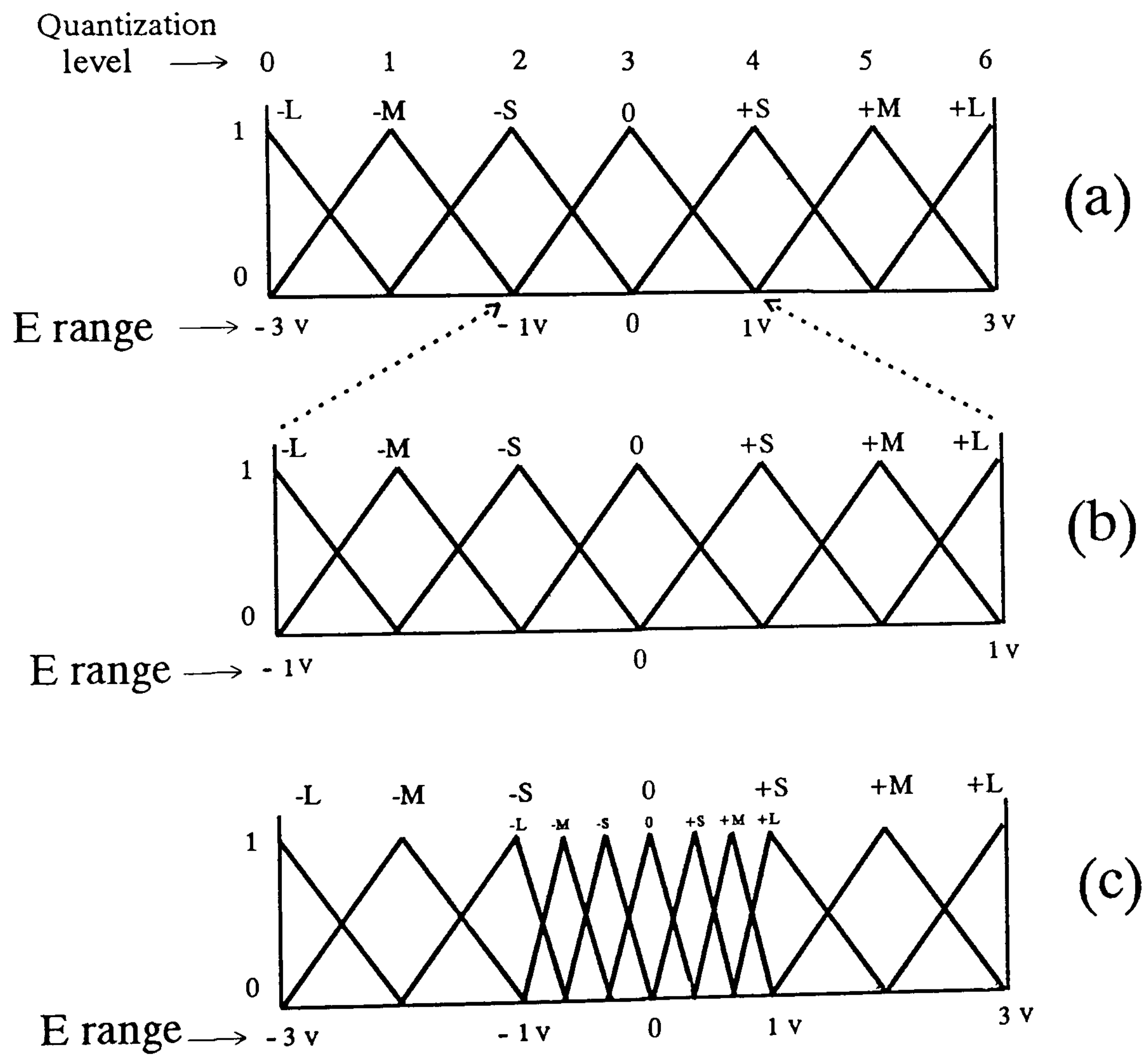


Figure 5-20. Fine control around the set point by shrinking the E range.

done by shrinking the error range E from ± 3Vto ± 1V,when control action moves into the quantization level between 2 and 4. At first, when the error value e_x is still very large, the membership functions are shown in Figure 5-20(a), where the E range is ± 3V.

The universe of discourse of the error is quantized from 0 to 6. If the error e_x becomes smaller, the control action will move into the area of the quantization level between 2 and 4, which means -1V $<e_x<$ 1V. Then, the decision table E range is intentionally switched to ± 1V as shown in Figure 5-20(b). The result of shrinking the E range is equivalent to constructing some narrow membership functions near the set point as shown in Figure 5-20(c).

Figure 5-21(a) shows the effect of shrinking E, D and U by the same ratio $r=15$ when a plant $1/(s+1)$ was controlled by a fuzzy PI controller with E=3V, D=0.1V and U=0.6V. The figure shows the effect of shrinking action on the system response. The original control trajectory is shown in Figure 5-21(b) and 5-21(c) shows the effect of shrinking on the control action. The shrinking effect was introduced when the control action moved into the region $2<e_x/E+3<4$ (within 1/3 of range E).

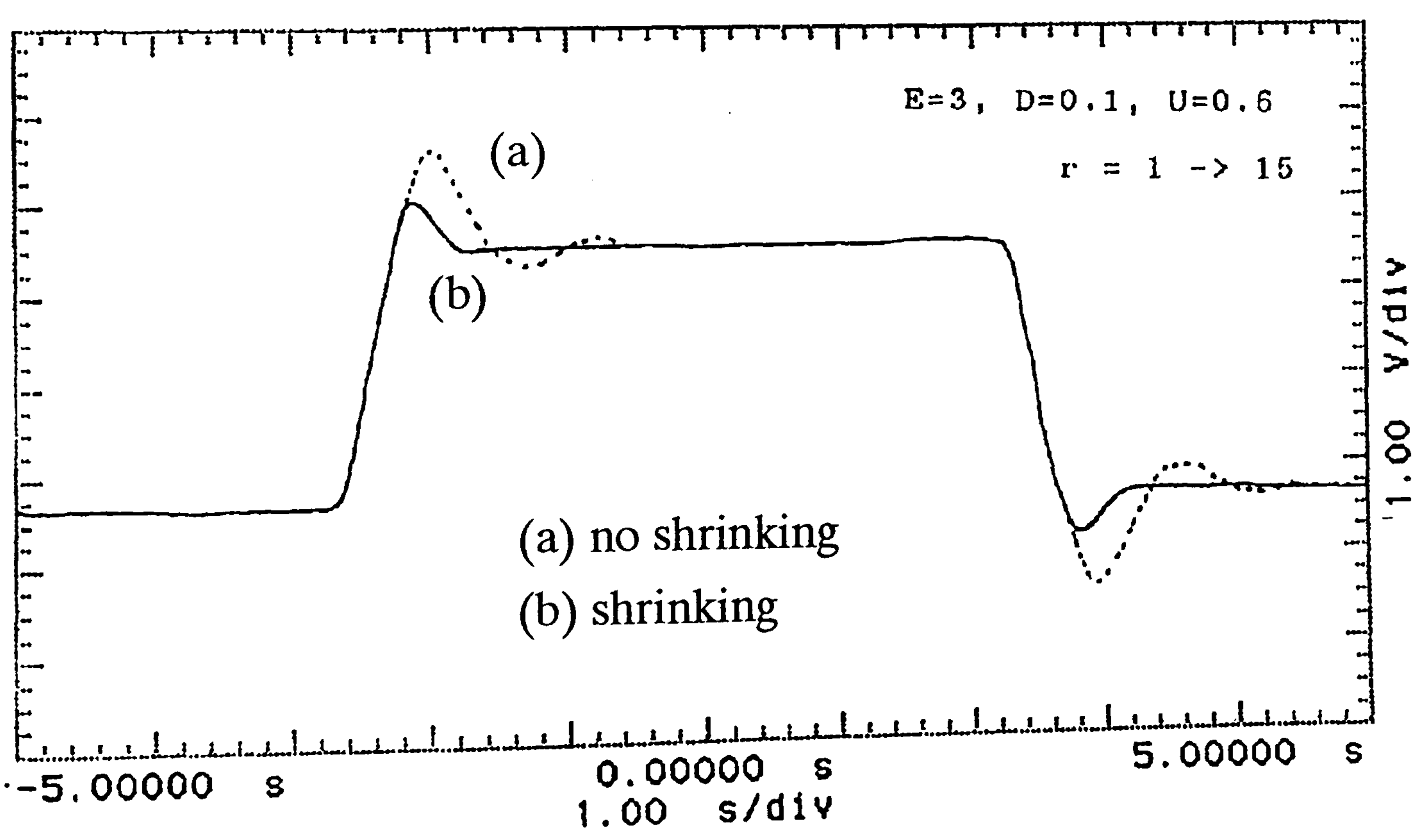


Figure 5-21. The effect of temporary decreasing the decision table ranges to a new value by dividing it a constant r.

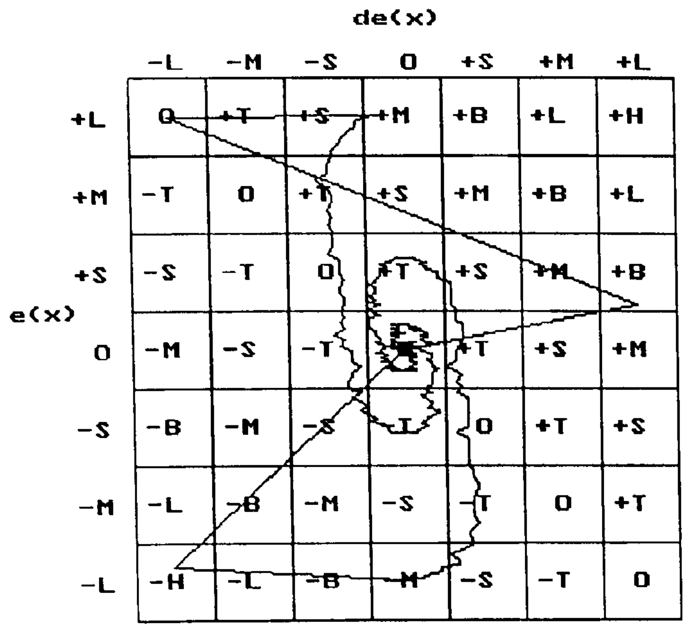


Figure 5-21(b). the trajectory before shrinking, r=1.

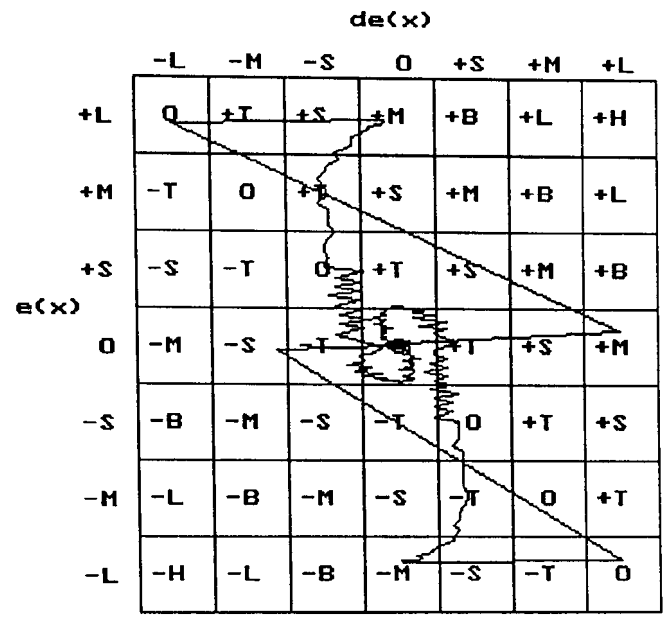


Figure 5-21(c). the control trajectory after shrinking by r=15.

(2) <u>Temporarily changing D to a very large value and switching the Fuzzy PI controller to an integral controller</u>

Liaw and Wang[54] have tried to overcome the limit cycle problem in an induction motor control task. By switching an integral controller into their system when the fuzzy control action was close to the set point. A similar technique was applied here. However, instead of introducing any analog I controller into the control system, the fuzzy controller itself was transformed to an integral controller by setting D=20,000,000V.

Curve (1) of Figure 5-22(a) was the output response of a plant 10/(S+10),which was controlled by a Fuzzy PI controller with E=3V,D=0.03and U=0.3V.Obviously, it is a sliding control response as shown in Figure 5-22(b), where the overdamped output has a longer rise time. Curve (2) of Figure 5-22(a) is the output response when this plant was controlled by a fuzzy I controller with E=3V,D=20000000Vānd U=0.3V. The system response increased rapidly but with a larger overshoot. Curve (3) shows the result when the fuzzy controller was modified such that it had the same decision table

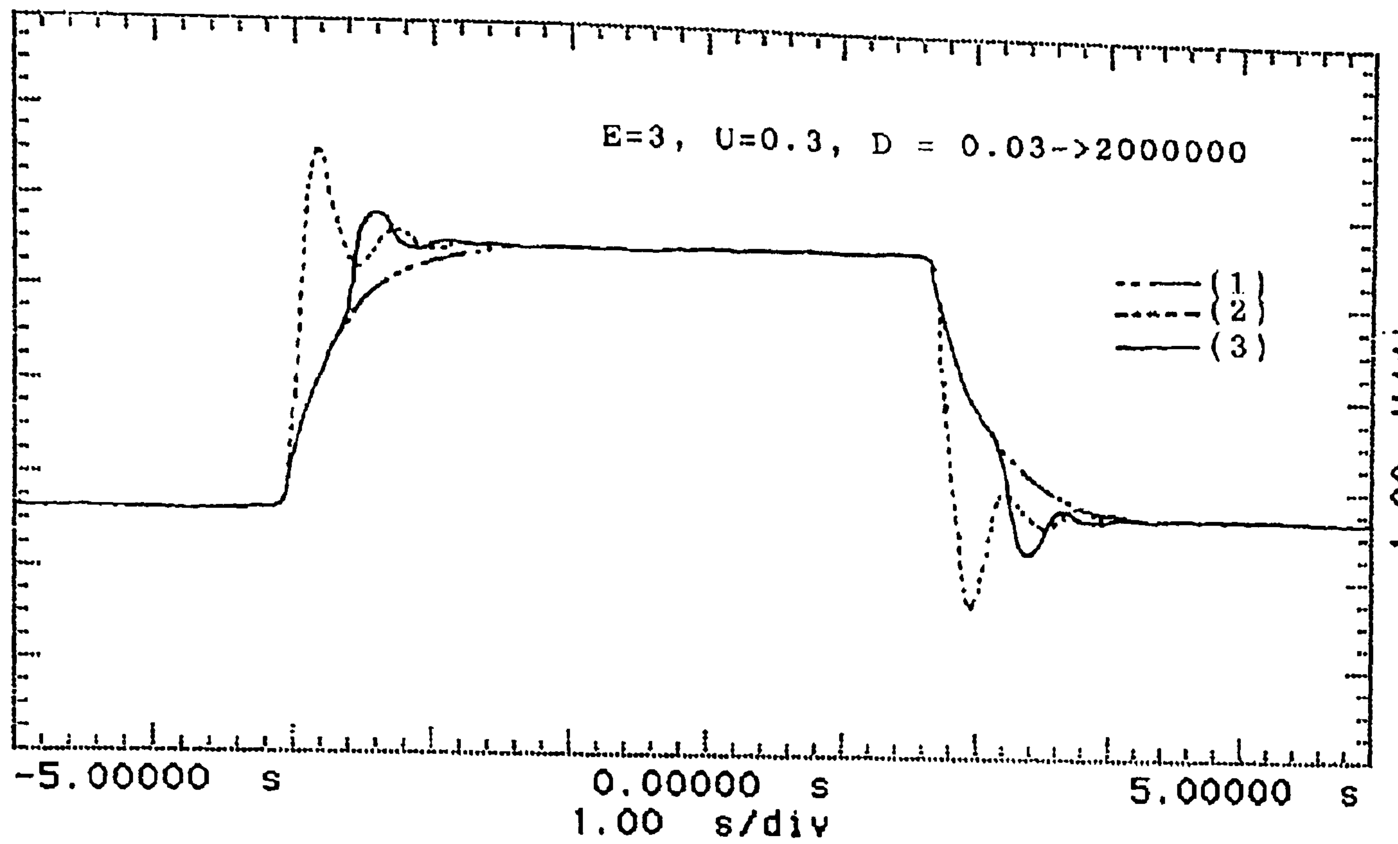


Figure 5-22(a). The output response of a plant 10/(s+10)when it is controlled first by a fuzzy PI controller then it is temporary switched to a fuzzy I controller.
(1) E=3V,D=0.03V and U=0.3V,
(2) E=3V,D=20000000V and U=0.3V,
(3) result of switching from (a) to (b) when -1V<e_x<+1V.

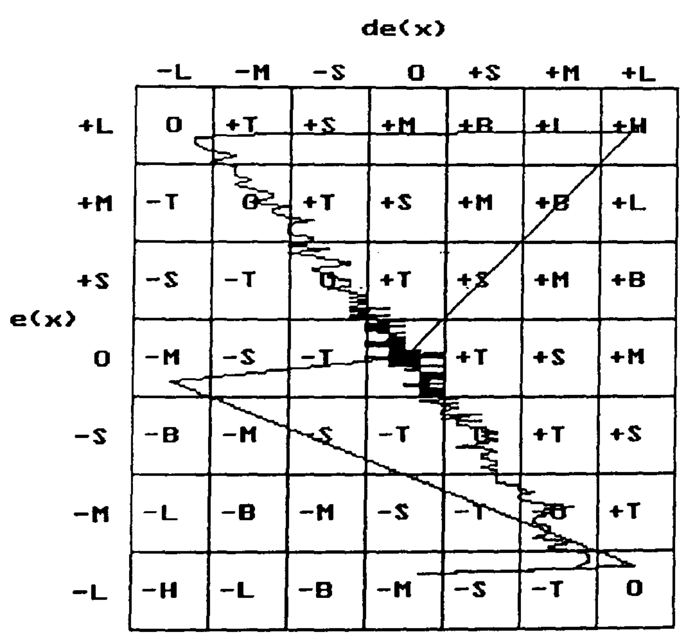


Figure 5-22(b).The trajectory of the original fuzzy PI sliding control with D=0.3V

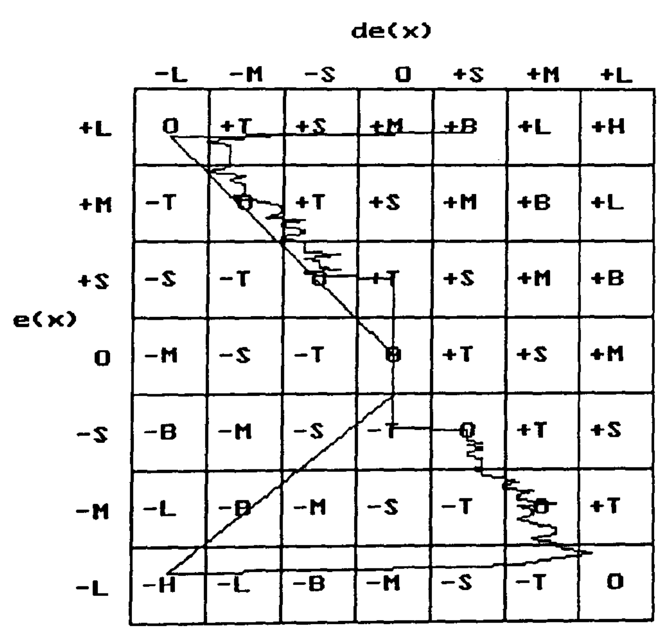


Figure 5-22(c) trajectory of the PI→I switching fuzzy controller.

ranges as curve (1) at first, and was switched to the settings of curve (2) whenever it was driven by e_x into the quantization level between 2 to 4. The output response has a rise time about the same as curve (1) in the beginning but it rises very fast when close to the set point. The overshoot is also decreased. The control action trajectory of curve (3) is also shown in Figure 5-22(c).

(3) Shrinking U to a small value around the set point

From equation (4-15) and (4-16) we noted that decreasing D, E and U together tends to keep both K_i and K_p constant. Figure 5-4 and Figure 5-5 also shows that increasing K_i and K_p will have different effects on the system response. The former will increase the overshoot and the later will decrease it. Since U appeared both in equation (4-15) and (4-16), its magnitude will affect K_i and K_p together. However, decreasing U has the

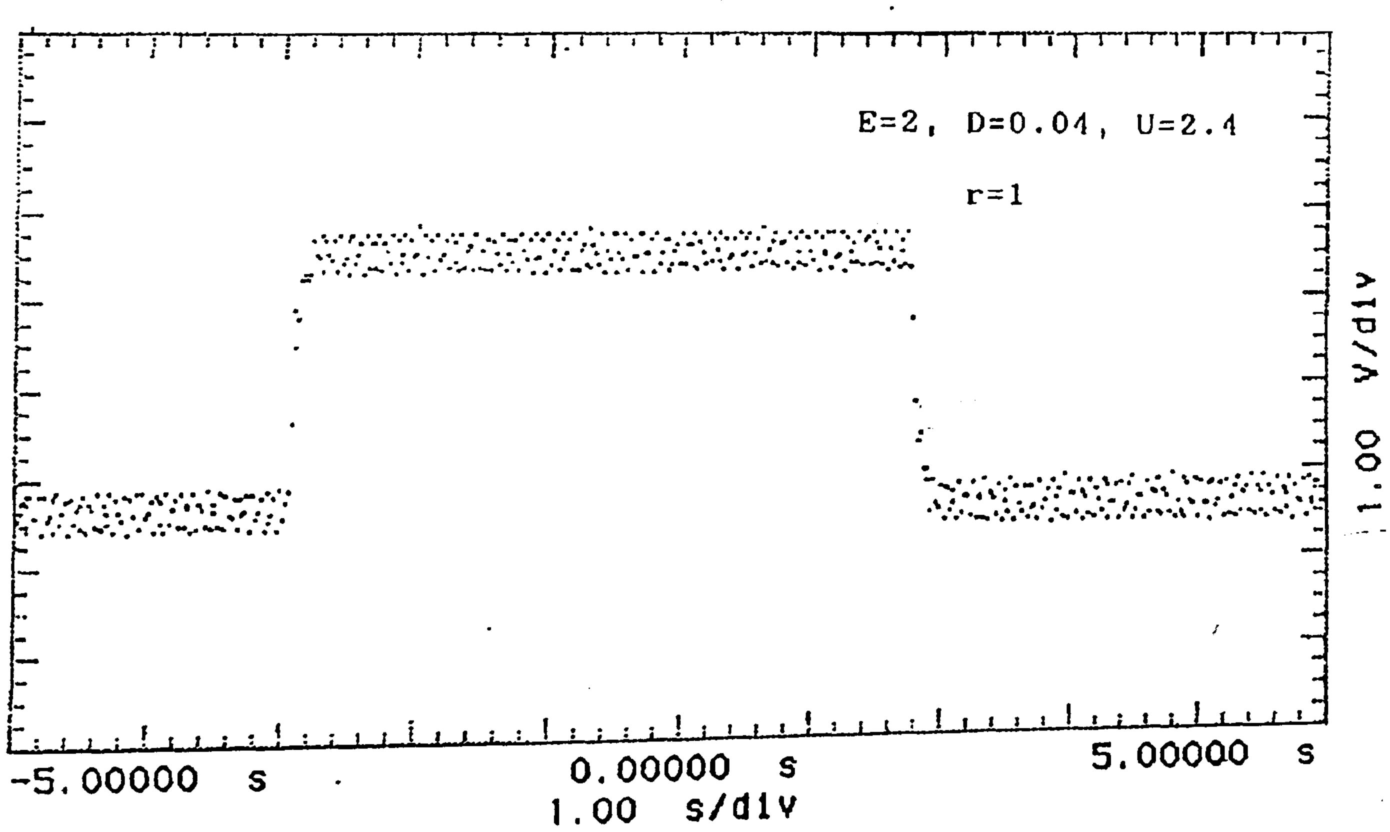
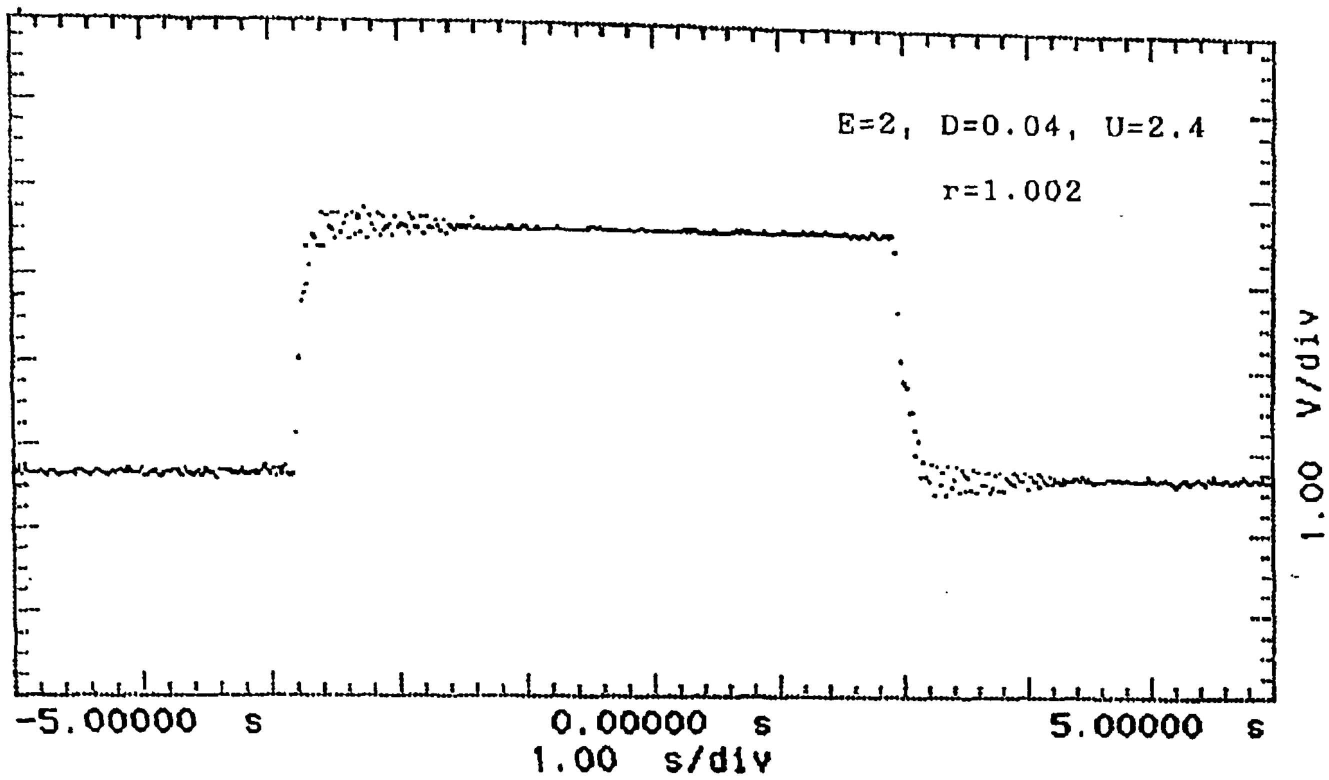
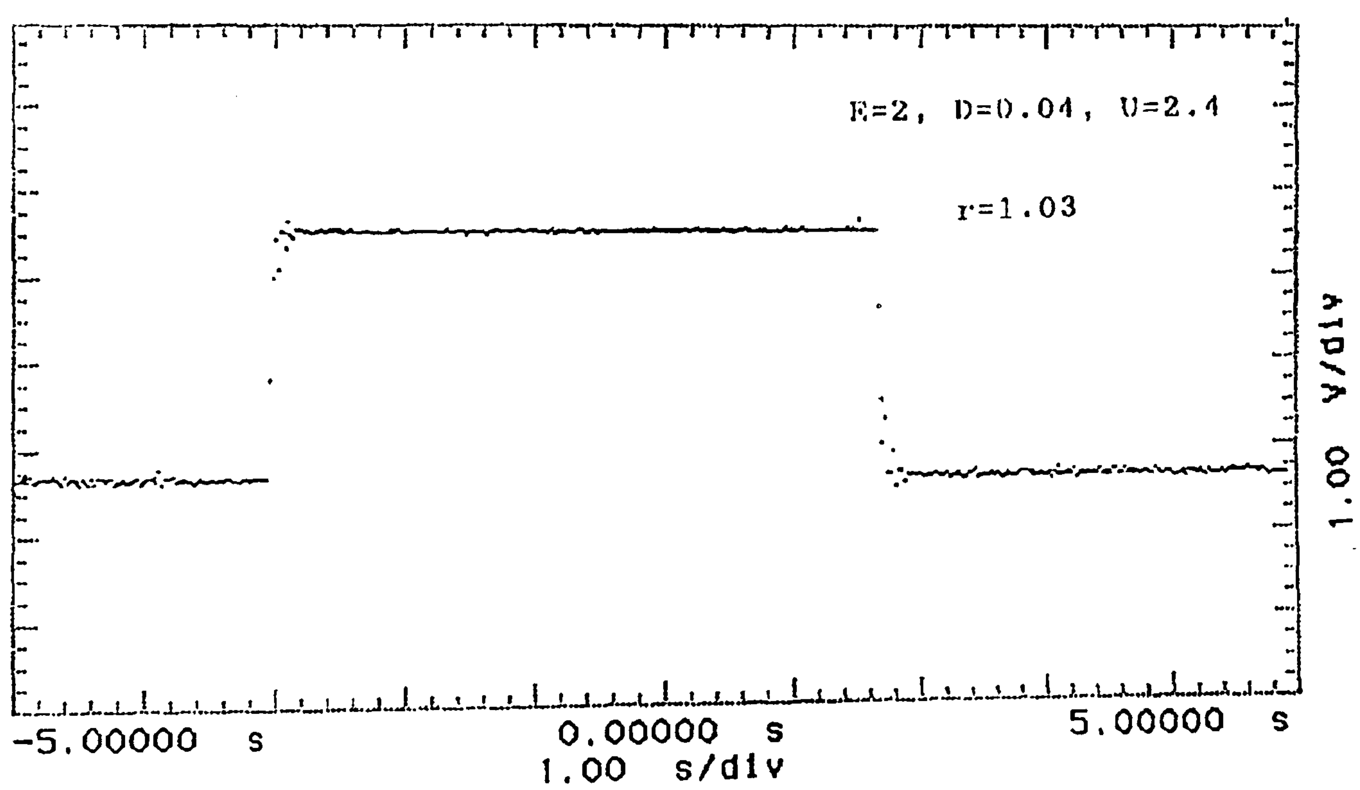


Figure 5-23(a) The output response without shrinking U.(r=1)



(b) result of shrinking U by r=1.002



(c) result of shrinking U by r=1.03

Figure 5-23. The effect of continuously decreasing the fuzzy PI controller output range U by a shrinking ratio r. The plant under control is 10/(s+10).

effect of reducing the open loop gain of the system and seems to decrease the energy fed to the plant. Therefore, it might help to make the control response settled down at the set point. The following experiment were done by continuously decreasing U when those control action moved into the error quantization level 2 to 4 of the decision table. Figure 5-23(a) shows that a plant 10/(s+10)was controlled by a fuzzy controller with E=2V,D=0.04Vand U=2Vat first, having a strong limit cycle around the set point. Then, in Figure 5-23(b), U was decreased by continuous shrinking with shrinking factor r=1.002, i.e. U was divided by r at each control sampling instant. Figure 5-23(c) is the result of shrinking by r=1.03,where the oscillation was suppressed quickly. To follow the input change, the controller had to be switched back to the previous U range, 2.4V, when e_x dropped below the quantization level 1 or rose above the quantization level 5.

Chapter 6

The characteristics of the pH process experimental pilot plant

As mentioned early in Chapter 1, the control of pH (the concentration of hydrogen ions) is a well-known control problem. Usually this problem demands the application of control method which take into account the non-linear and time varying titration curve and the strong sensitivity to disturbances near the point of neutrality. Several methods which have been tried in the past two decades are briefly discussed in section 2.4, and the motivation for applying fuzzy control in this thesis has also been explained in chapter 1. Now, before starting the fuzzy control experiments, the characteristics of the pH process experimental plant preferably should be understood through its model equations and measurements of the process component I/O relations.

6.1. pH Process Model Equations

The method for modeling the dynamics of pH in a stirred tank developed by McAvoy[8] has been mentioned in chapter 1 and its model equations (1-5) and (1-6) are rewritten here as

$$V\frac{dC_A}{dt} = C_{Acid}F_A - C_A(F_A+F_B) \qquad (6\text{-}1)$$

$$V\frac{dC_B}{dt} = C_{Base}F_B - C_B(F_A+F_B) \qquad (6\text{-}2)$$

In these equations, C_A represents acid anion concentration and C_B represents base cation concentration in the effluent stream. C_{Acid} and C_{Base} are the acid and the base concentrations of the acidic and basic streams entering the continuous stirred tank reactor (CSTR) with flow rate F_1 and F_2 respectively. V is the volume of the CSTR.

Water molecules are dissociated (split up into hydrogen and hydroxyl ions) according to the formula

$$H_2O \rightleftharpoons H^+ + OH^- \qquad (6\text{-}3)$$

In chemical equilibrium the concentration of hydrogen H^+ (or rather H_3O^+) and hydroxyl OH^- ions are given by the formula

$$\frac{[H^+][OH^-]}{[H_2O]} = constant \qquad (6\text{-}4)$$

Only a small fraction of the water molecules are dissociated, so, the water activity is practically unity, i.e. $[H_2O] \approx 1$ and we get

$$[H^+][OH^-] = K_w \qquad (6\text{-}5)$$

where the equilibrium constant K_w has the value $10^{-14}[(mole/l)^2]$ at 25^0C.

The constraint that the solution remains electrically neutral gives

$$C_A + [OH^-] = [H^+] + C_B \qquad (6\text{-}6)$$

The concentration of hydroxyl ions can be related to the hydrogen ion concentration by equation (6-5). Hence

$$C_B - C_A = [OH^-] - [H^+] = \frac{K_W}{[H^+]} - [H^+] \tag{6-7}$$

Then, letting $x = C_B - C_A$ and solving for $[H^+]$ gives

$$[H^+] = (\frac{x^2}{4} + K_W)^{\frac{1}{2}} - \frac{x}{2}$$

and hence

$$[OH^-] = (\frac{x^2}{4} + K_W)^{\frac{1}{2}} + \frac{x}{2}$$

Thus from the definition of pH given in chapter 1

$$pH = f(x) = -\log[(\frac{x^2}{4} + K_W)^{\frac{1}{2}} - \frac{x}{2}] \tag{6-8}$$

The graph of the function f is called the *titration curve*. It is the fundamental non-linearity for the pH control problem. An example of the titration curve has been shown in chapter Figure 1-2. There is considerable variation in the slope of titration curves. This can be seen from the derivative of the function f which is

$$f'(x) = \frac{\log_{10}e}{2(\frac{x^2}{4} + K_W)^{\frac{1}{2}}} = \frac{\log_{10}e}{10^{pH-14} + 10^{-pH}} \tag{6-9}$$

The largest slope occurs at pH $=7$ where $f' = 2.2 \times 10^6$. However, when pH increases to 10 or decreases to 4 the slope $f' = 4.3 \times 10^3$. Therefore, the gain can vary by a factor of order 3. Figure 1-2 also shows the titration curve for a weak acid and strong base. It can be seen that in this case the curve is not symmetrical about pH$=7$, and more importantly, its slope changes more gradually as the point pH 7 is approached from below. This makes the titration process more controllable, which is why weak acid is sometimes used as a buffer to slow down the strong acid reaction.

6.2 Experimental setup

The pH process model plant which has been used for the experiments conducted in the work reported in this thesis is shown in Figures 6-1 and 6-2.

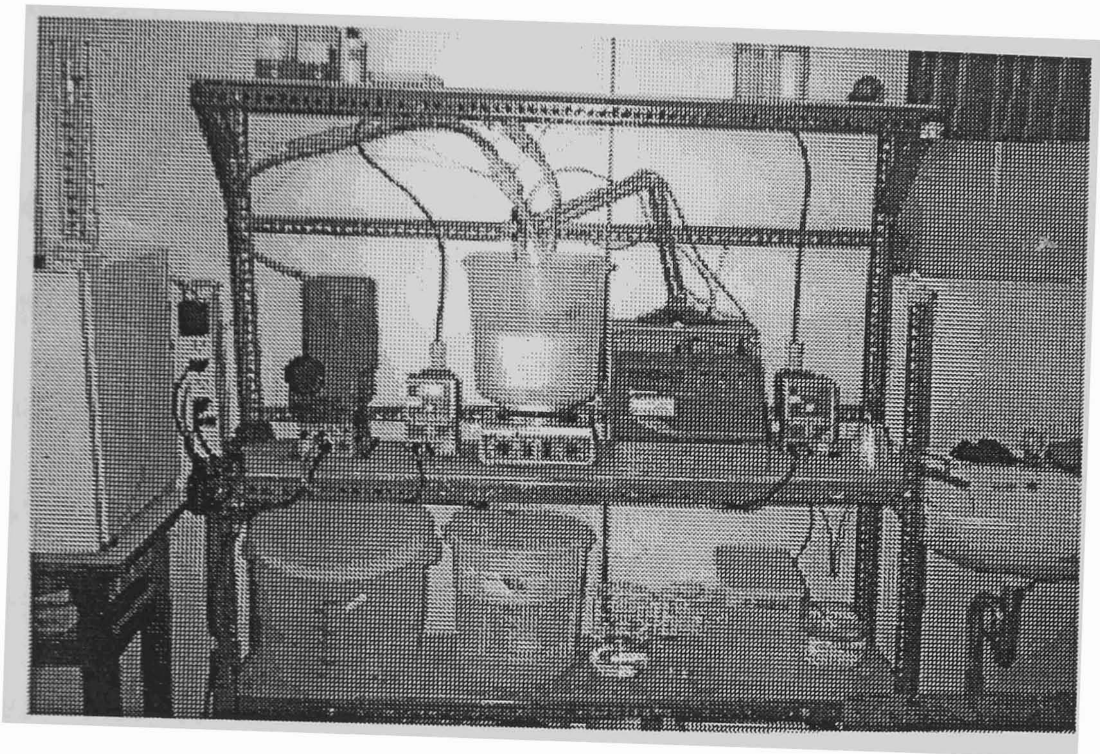


Figure 6-1. pH process plant model.

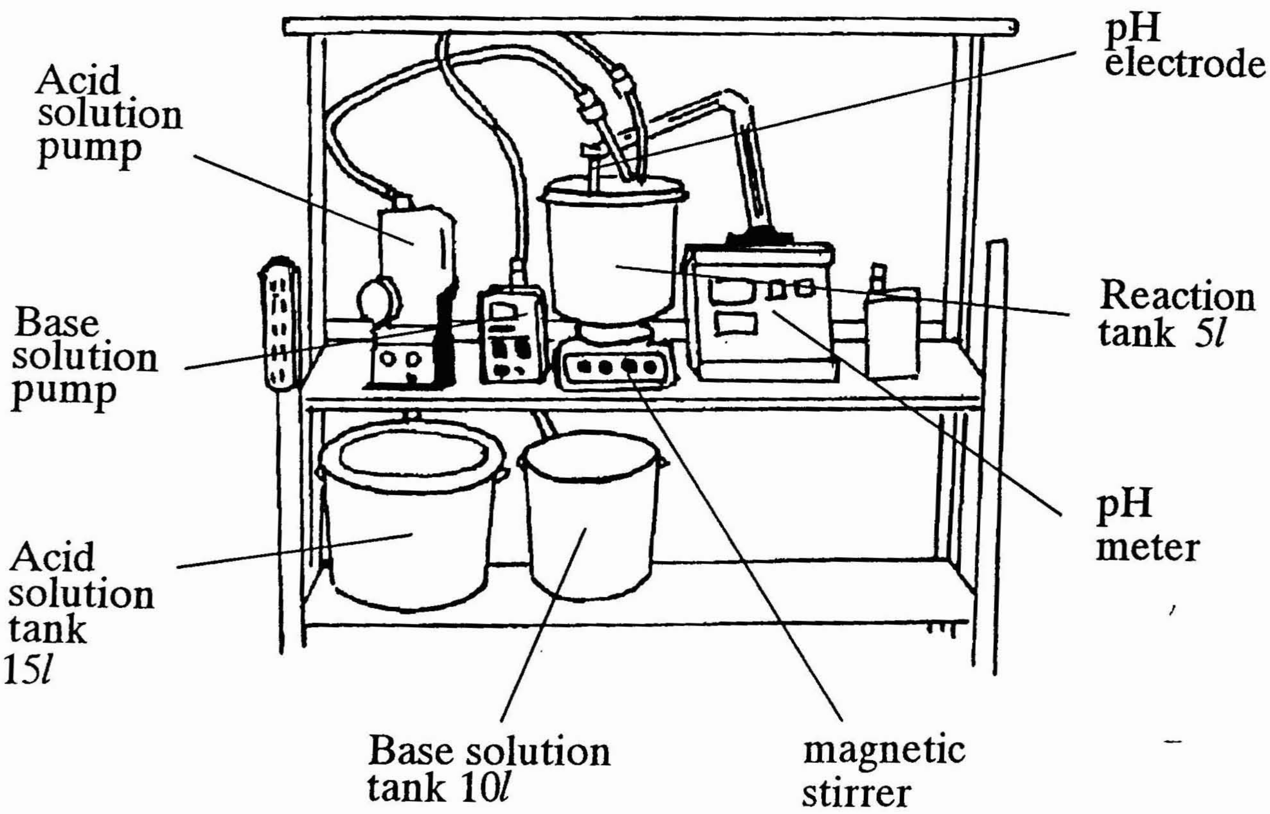


Figure 6-2. Schematic of pH process plant model.

The model plant consists of a reaction tank of volume 5l, a base solution tank of volume 10l and a acid solution tank of volume 15l. The acid tank contained a solution of 0.0032 mol/l of HCl acid and the base solution tank contained a solution of 0.025 mol/l NaOH. The acid and base solutions were fed to the reaction tank via metering pumps and mixed in the reaction tank by a magnetic stirrer. The acid metering pump could be varied manually between 0 ~ 45l/h, and was used to set the flow rate of the process stream for the CSTR. To keep the volume of the reaction tank constant, an overflow pipe was connected to the top of the reaction tank. For all the experiments conducted in this work a flow rate of 0.25 l/min was used. The base flow was delivered via a microprocessor-based solenoid driven metering pump set to operate at 100% stroke length, which enabled the flow rate to be controlled between 0 and 1.8 l/h by an analog current signal (4-20 mA). The pH value of the reaction tank was measured by a pH

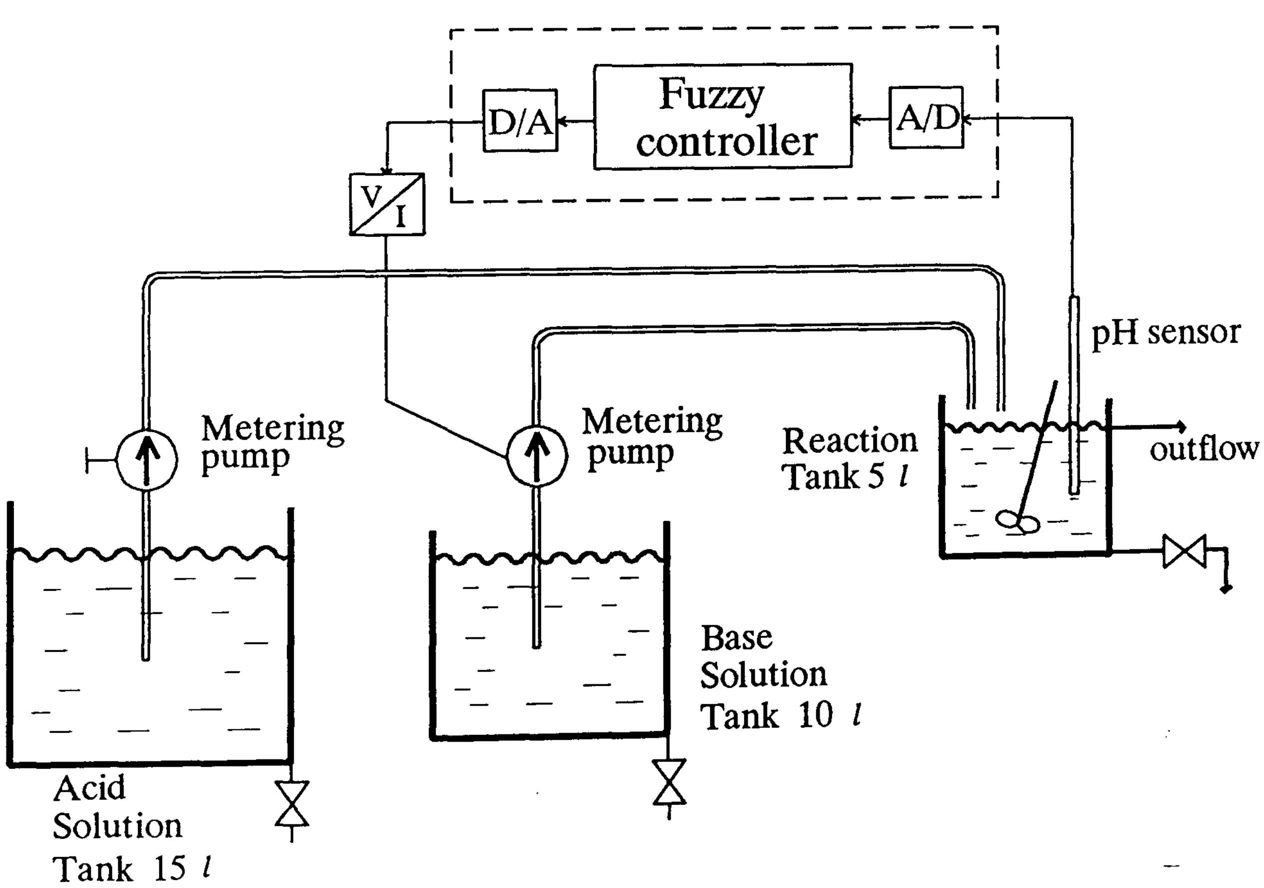


Figure 6-3 the pH process model plant.

sensor consisting of a pH electrode and associated meter. The PC based control system described in section 3.4 was coupled to the pH process plant model in order to conduct the pH control experiments. A schematic description of the pH process model plant and controller is given in Figure 6-3. The major component specifications are listed in the Appendix.

6.3 Static and Dynamic Behavior of the pH process model plant

A block diagram for the pH control process model plant described in Figures 6-1 and 6-2 is shown in Figure 6-4. As shown in the figure, the control signal from the control computer is sent to the metering pump unit to control the flow rate of the base solution into the reaction tank. A dynamic filling block and a non-linear titration characteristics block represents the accumulation and chemical reaction in the tank, respectively. The pH value in the reaction tank is measured using a pH meter and represented by a voltage

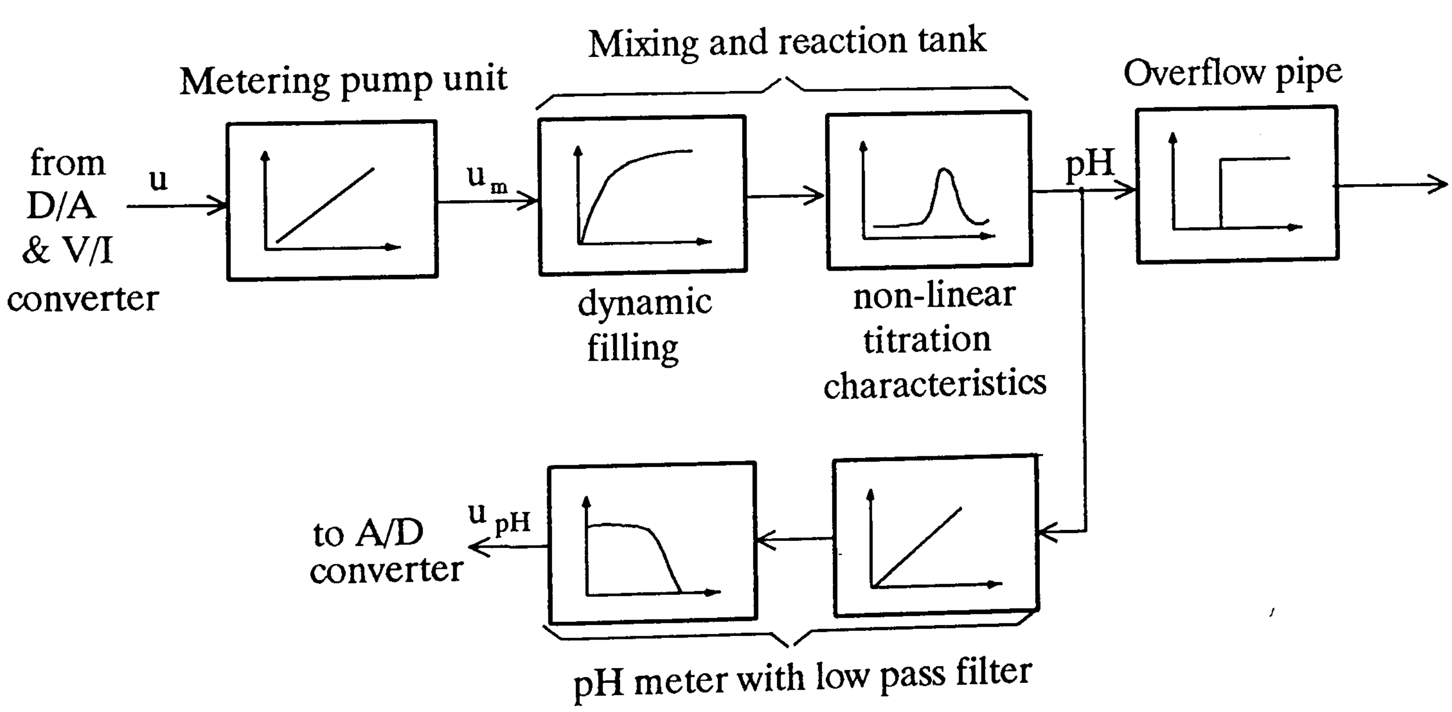


Figure 6-4. block diagram for a pH control model plant

signal between $0 \sim 1400$mV. This pH signal is fed back to the control computer via an anti-aliasing filter of bandwidth 20 Hz.

6.3.1 Dynamic characteristics

In this strong acid and base pH process, the chemical reaction time is small in comparison with the dynamics of flow into the tank. The flow dynamics are described by equation 6-1 and 6-2. When the inlet flows F_A and F_B are constant, these equations are linear and first order with time constant $T=V/(F_A+F_B)$. At values of pH close to 7, the titration curve dominates the process and masks the effect of the filling dynamics. However, for pH values below 5 and above 9 the titration curve effects are less dominant and the filling dynamics become evident in control experiments. Since the process pilot plant has a reaction tank with volume 5*l* and was operated with a process flow rate of 0.25 *l*/min, its theoretical filling time constant is 20min. Note that the base flow rate used n the experiments is much smaller than the acid (process) flow rate and was neglected in this calculation. This value compares well with the time constant observed in subsequent experiments.

6.3.2 Calibration of pH sensor and actuator

The $0 \sim 14$pH value of the reaction tank is measured by a pH sensor as a signal in the range $0 \sim 1400$ mV. This signal passes through an anti-aliasing filter of bandwidth 20 Hz and traverses a wire of 4.5m in length before reaching the A/D converter. Therefore, an experiment was conducted to measure the difference between the voltage at the pH meter and the voltage logged by the control computer. The result of this experiment is shown in Figure 6-5, where it can be clearly seen that a negative voltage offset has been introduced at high pH values. The voltage offset is attributed to the resistance in the wire. In order to correct for this, the measured voltage V_{x1} received by the control computer was corrected to $V_x = V_{x1} + (V_{x1} - 300)*23$.

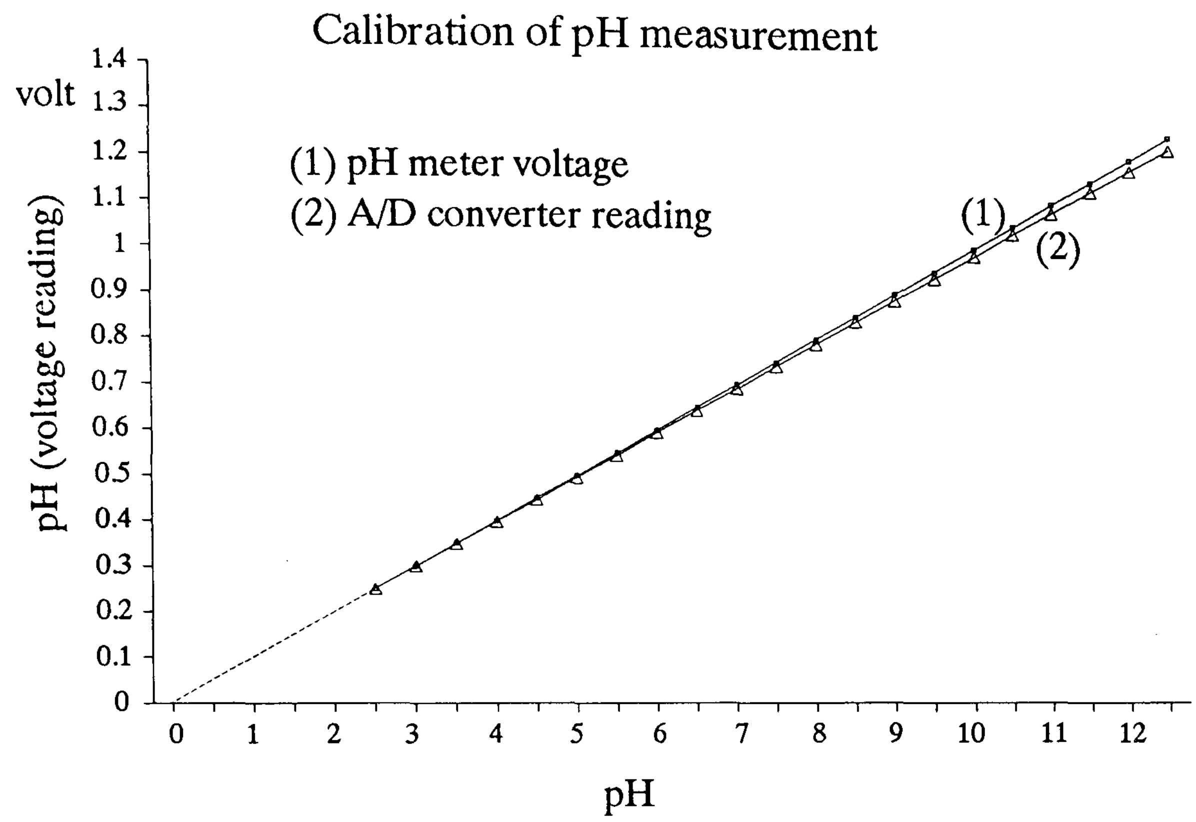


Figure 6-5. calibration of pH measurement.

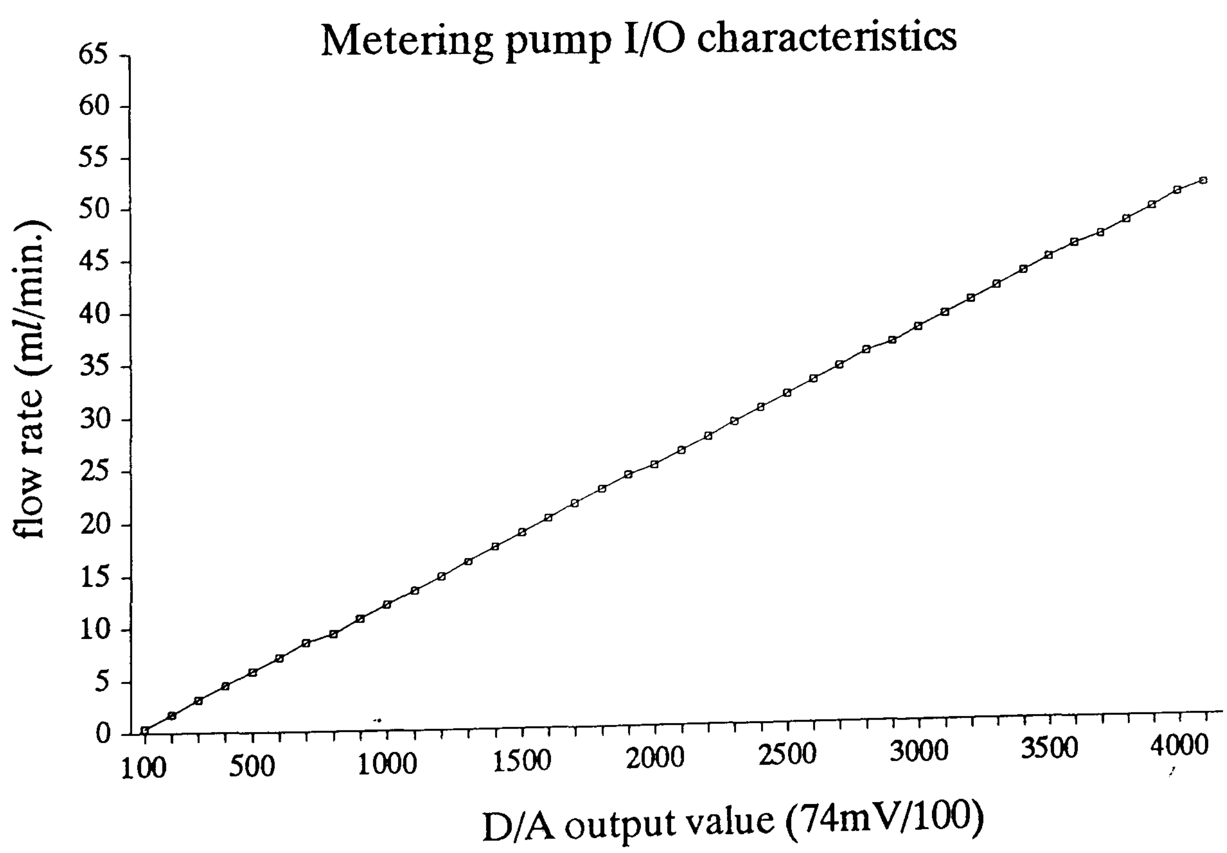


Figure 6-6 Metering pump characteristics.

In order to calibrate the base flow rate actuator, a measuring cylinder and stop watch was used to measure the flow rate from the metering pump corresponding to a demand flow rate set in the control computer. The results shown in Figure 6-6 indicate that the flow rate of the metering pump is linear between the settings 0 ~ 4000 at the D/A output. Above 4000 it was found that the pump saturated at its maximum flow rate.

6.3.3 Non-linear titration characteristics

The static behavior of the pH process is specified by its titration curve, which defines the variation in the pH value with the ratio(r=Mb/Ma) of the mass(mole) of base and the mass(mole) of acid contained in a vessel. Titration curves are normally obtained by adding a measured quantity of one reagent to a fixed quantity of a second reagent contained in a vessel. In this experiment, a measures amount of HCl solution (0.0032 mol/l) was add to 200 ml of NaOH solution (0.025 mol/l) contained in a vessel. The

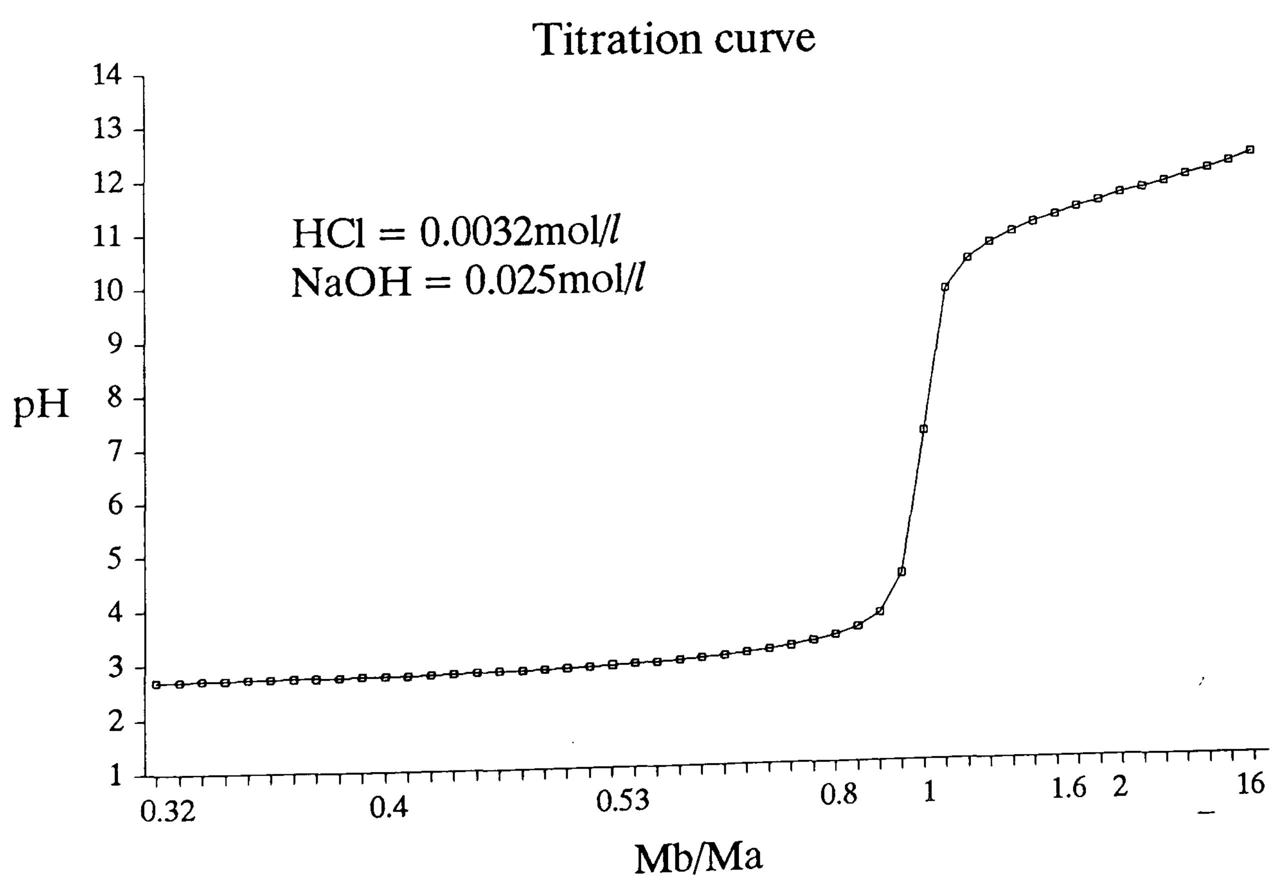


Figure 6-7 the titration curve of strong acid (HCl) and base (NaOH)

resultant titration curve is shown in Figure 6-7. To illustrate the non-linearity of the system, the static gain K (=ΔpH/ΔMb) is also calculated from the data in Figure 6-7 and shown in Figure 6-8.

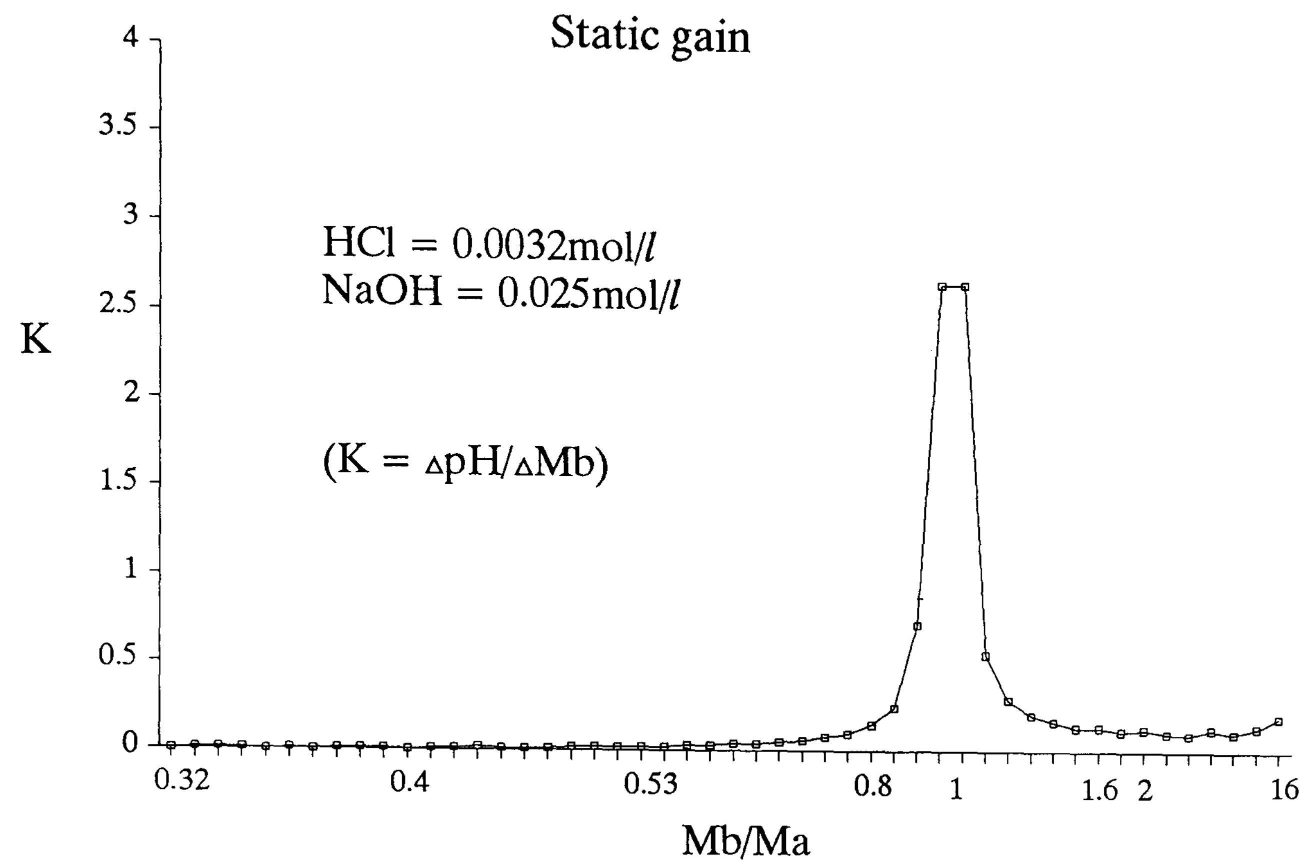


Figure 6-8. the static gain K =ΔpH/ΔMb

In the experiments of the following chapter, the NaOH solution must be used within two hours after it is made. This is because the NaOH solution will absorb CO_2 quickly and form the 1- and 2- weak acidic bases. Usually the weak acid will act as a buffer in the titration process, that is it will slow down the reaction when approaching pH 7 and largely increase the time constant. This weak acid buffering phenomenon has been discussed by McAvoy[8]. Figure 6-9 shows the buffering effect resulting from the NaOH solution having been left to stand for 5 hours before commencing the titration experiment.

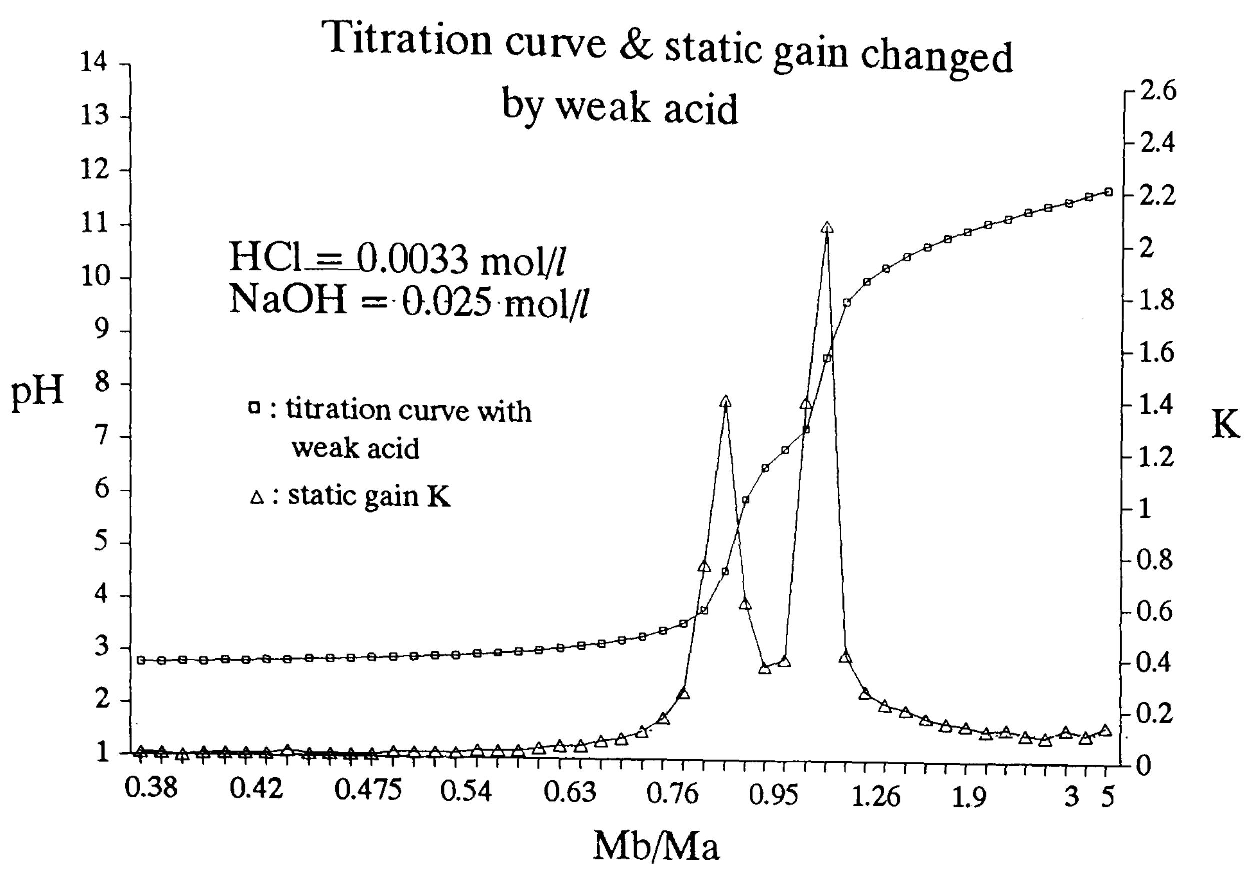


Figure 6-9. The titration curve and static gain K when weak acid is present.

The titration curves in Figure 6-8 and 6-9 were obtained by a conventional chemical bench experiment. The pH control problem investigated in this thesis is a continuous flow process where an acidic solution is pumped at a constant rate into a stirred tank into which a controlled amount of base solution is introduced to control the pH level.

In order to obtain the titration curve for this continuous flow pH process, an experiment was conducted with the reaction tank initially empty. The acid flow pump was then switch on at a fixed delivery rate of 0.25ml/min and the tank was allowed to full at which time the mass(mole) of acid in the tank

$$M_A = C_{Acid} \cdot V ,$$

Once the tank was full, the base flow pump was switched on at its maximum delivery rate of 0.05ml/min. Then, from equation 6-2, the subsequent increase in the mass of base in the tank followed the first order response

$$M_B(t) = \frac{C_{Base}\cdot V\cdot F_B}{F_A+F_B}\{1 - e^{\frac{F_A+F_B}{V}\cdot t}\}$$

Similarly, from equation 6-1, the corresponding decrease in the volume of acid in the tank followed the first order response

$$M_A(t) = \frac{C_{Acid}\cdot V}{F_A+F_B}\{F_A + F_B e^{-\frac{F_A+F_B}{V}\cdot t}\}$$

Hence, the ratio of base to acid in the tank was

$$r(t) = \frac{M_B(t)}{M_A(t)} = \frac{C_{Base}\cdot F_B\{1 - e^{-(\frac{F_A+F_B}{V})t}\}}{C_{Acid}\cdot F_A\{1 + \frac{F_B}{F_A}e^{-(\frac{F_A+F_B}{V})t}\}} \tag{6-10}$$

Clearly, at t=0,r(t)=0.Also as $t\to\infty$,

$$r(t) \to \frac{C_{Base}F_B}{C_{Acid}F_A} = \frac{0.033\cdot 0.05}{0.0032\cdot 0.25} = 2.06$$

Thus, this experiment effectively cause r(t) to vary from 0 to 2.06.

Figure 6-10 shows the variation in pH with time recorded in this experiment. The corresponding value of r(t) at each of the measured time instances was calculated from equation 6-10 and the resulting variation in pH with r(t) is given in figure 6-11. The continuous flow titration curve of Figure 6-11 compares well with the bench experiment titration curve of Figure 6-7, thereby confirming the validity of this continuous flow titration experiment.

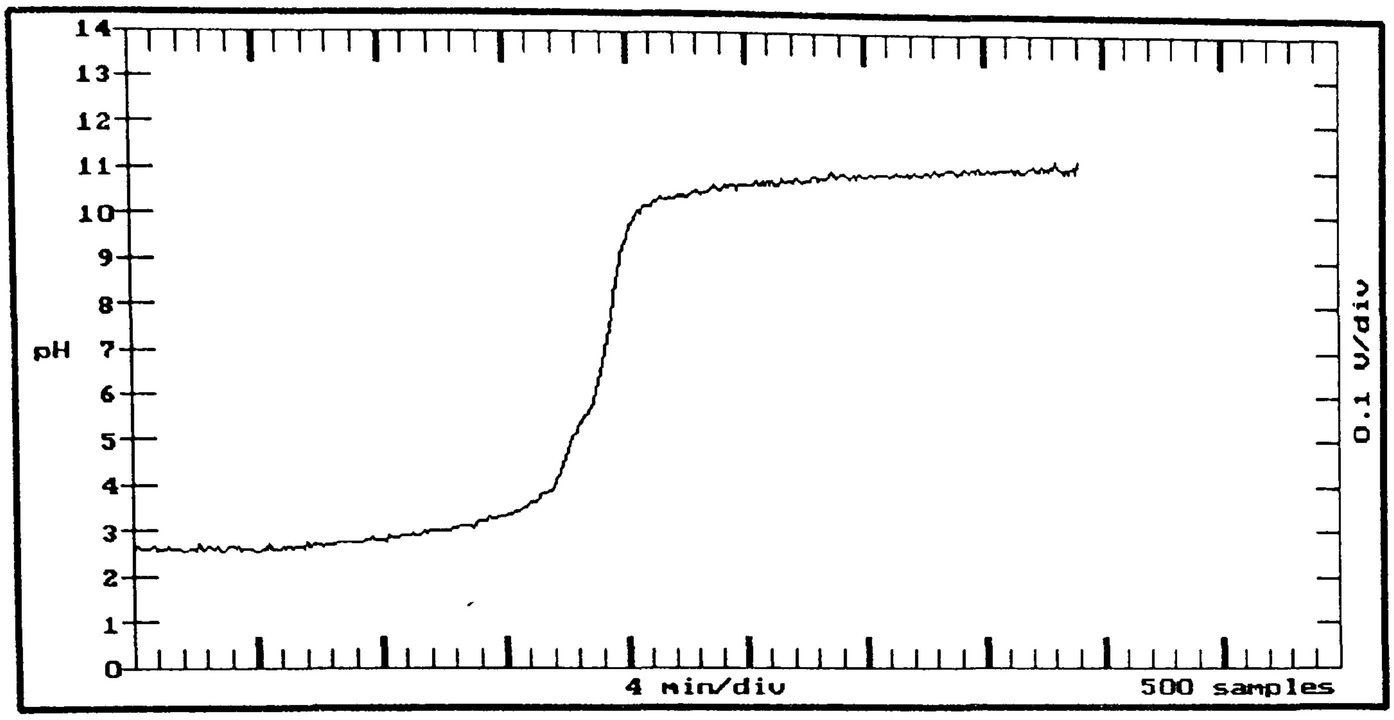


Figure 6-10 The flow titration experiment.

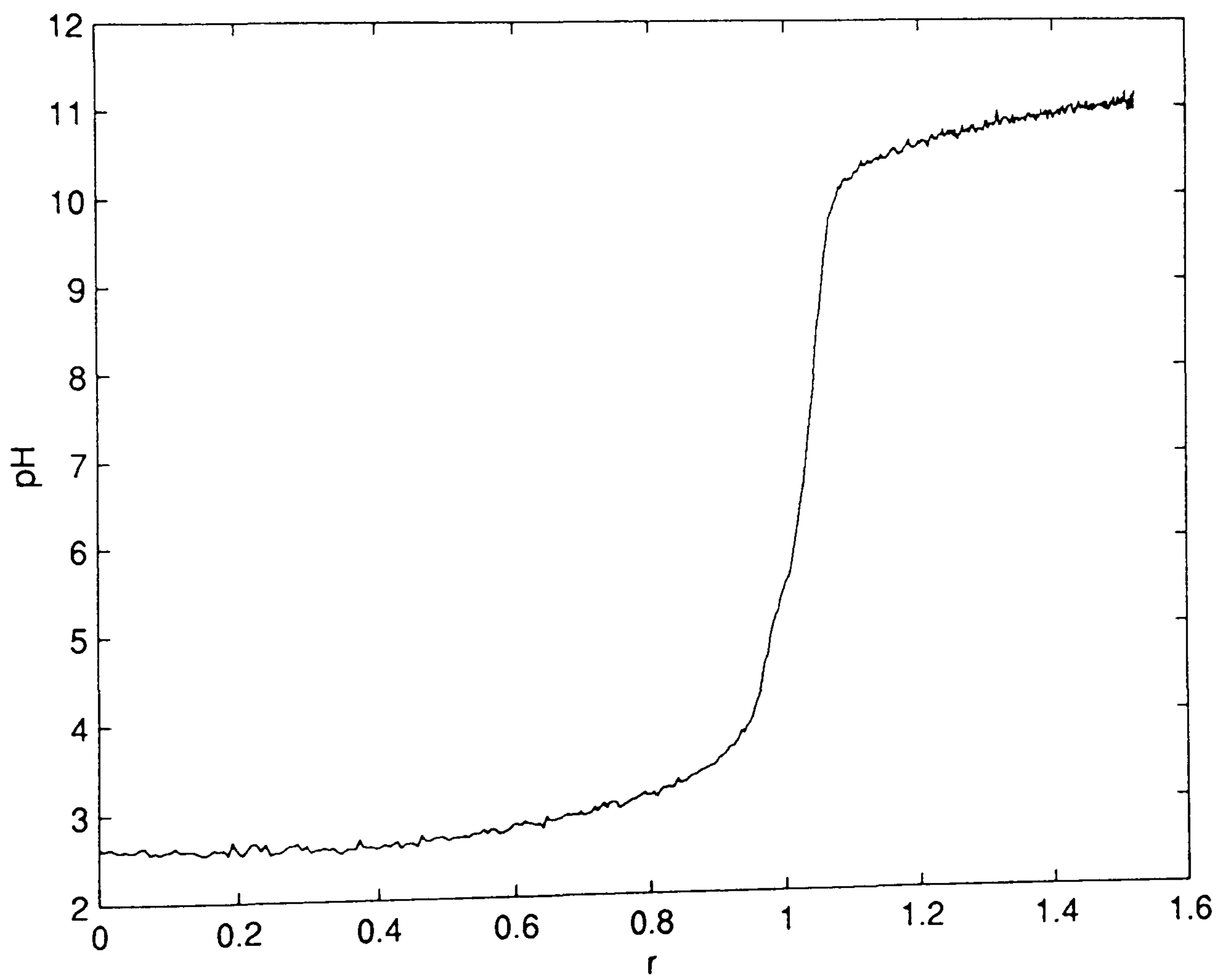


Figure 6-11 The resulting variation in pH value with r(t).

The flow titration experiments were repeated and the results are given in Figure 6-12 where curves (2) and (3) show the buffering effect of exposing the NaOH solution to air for 4 and 7 hours, respectively, before commencing the continuous flow experiment.

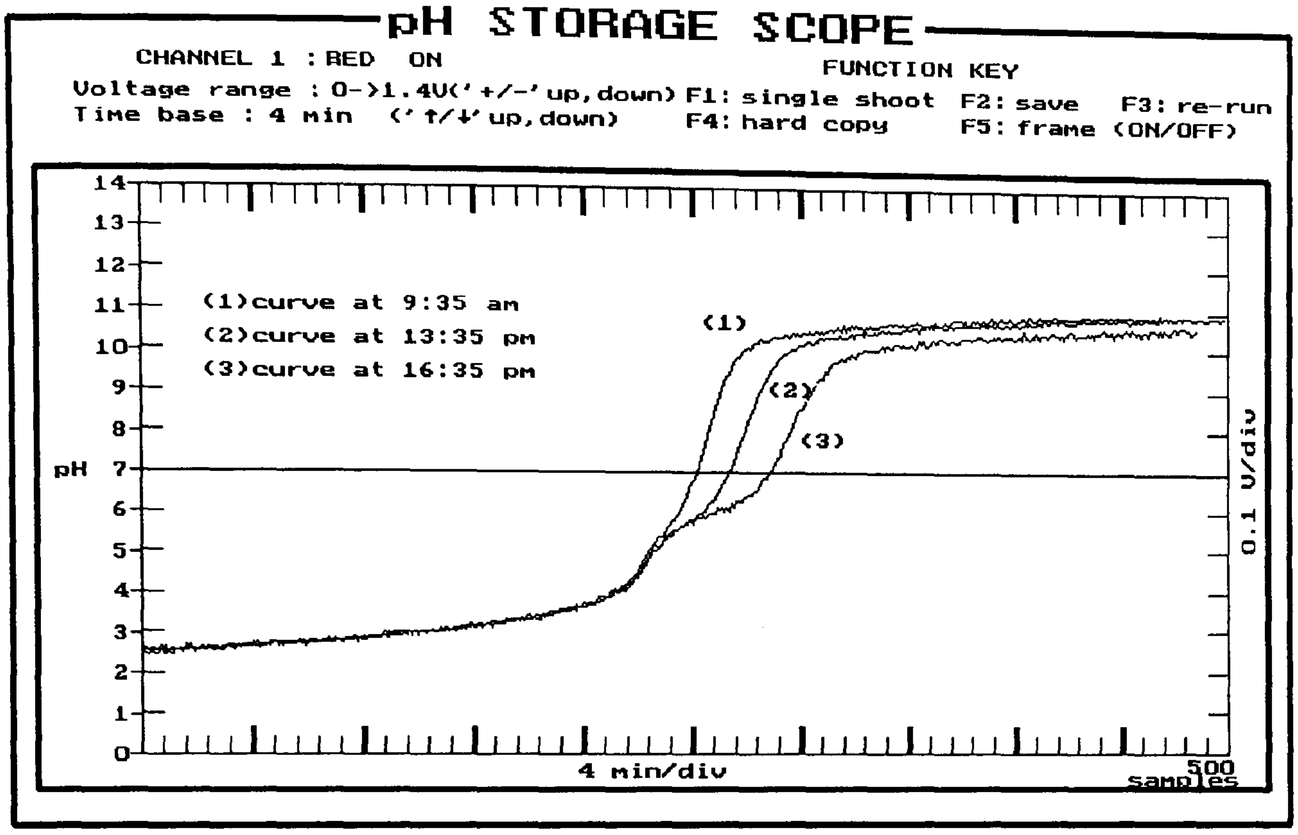


Figure 6-12. effect of weak acid buffering(absorbing CO_2) on titration curve.

6.4 Summary

In this chapter, the characteristics of the pH process experimental model plant have been described through its model equations and measurements of the process component I/O relations. The results show that firstly, it is a dynamic process with a time constant of about 20 min. This time constant is dominated by the volume of the mixing tank as well as the flow rate of the process stream. Secondly, the titration curve is highly non-linear, its static gain varying by a factor of order 3 between pH=4 and pH=7. Finally, it has been demonstrated it is also a time varying process due to the effect of CO_2 on the titration curve.

In summary, we have a time-varying and highly non-linear control problem in this pH process.

Chapter 7

Fuzzy control of the pH process pilot plant

Having discussed the pilot plant in the last chapter, we will now turn our attention to the experiments performed using this plant. It has been noted that pH control is made difficult by the nature of the signals involved. The measured value of pH is strongly affected by the stirring action, as well as by the non-uniform concentration due to the pulsed injection of the titrant by the metering pump. The result is that the signal representing the pH appears to be corrupted by a large amount of "noise". The effects of this noise are reduced by the use of a Butterworth filter, as described earlier, together with data averaging over one hundred samples. In order to allow time for the averaging process the sampling time T_s is increased to 14ms. To deal with the large time constant of the pH process, an anti-windup integrator technique is used by all the controller programs. This means that the controller stops integrating when its output reaches some saturation value.

Section 7.1 describes a direct comparison between the performance of a digital PI controller and a fuzzy PI controller in controlling the pH pilot plant. This section describes experiments to investigate the response of the plant to both a single step input, and to a set-point tracking task. In both cases the fuzzy controller performs consistently better than the digital PI controller. Having established the superiority of the fuzzy controller, section 7.2 looks in more detail at the differences between the two control methods when used to control the plant at pH 7, and 8. This shows that the fuzzy PI controller can make the system settle down at the set-point faster, and with less over-shoot than the digital PI controller when a strong acid and a strong base are reacting, with or without buffering. The effects of the U and D ranges on the fuzzy control response, and the damping effect which occurs when the control action trajectory is near the set-point, are studied in section 7.3, 7.4 and 7.5. Then, in section 7.6, the results of the experiments performed in previous sections are used to determine the best ranges of E, D and U for different value of pH. Finally, in section 7.7 the performance of the fuzzy controller is investigated for different load concentrations and in response to a perturbation of the load.

7.1 Comparing the performances of a digital PI controller and a fuzzy PI controller in the control of the pH process.

We noted in chapter one that the control of pH is very difficult due to the non-linear relationship between pH and concentration. When using a conventional digital PI controller to control the process at a pH of 7, the pH output often oscillates around the set-point with a large over-shoot and a very long settling time as shown in Figure 1-3. We shall see in section 7.2, that a series of experiments was performed to investigate the difficulties of handling this control problem using a conventional digital PI controller. The best result that could be obtained from these digital PI control experiments is shown in Figure 7-1(a). However, if the fuzzy controller developed in this thesis is applied to the same system, a better result is obtained as shown in Figure

7-1(b). The pH output settles down more quickly and with a smaller over-shoot. Therefore, it is better than the digital PI controller.

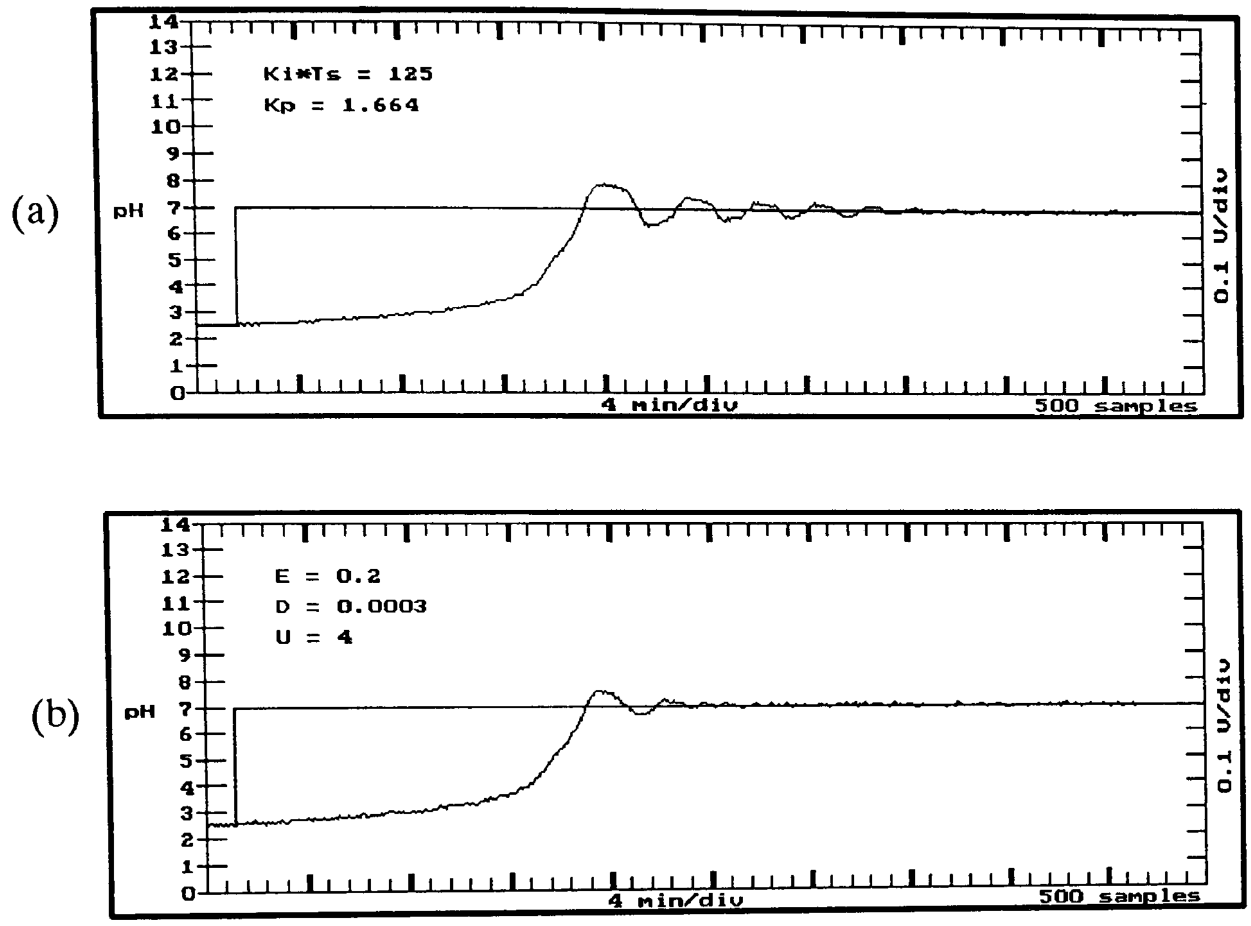
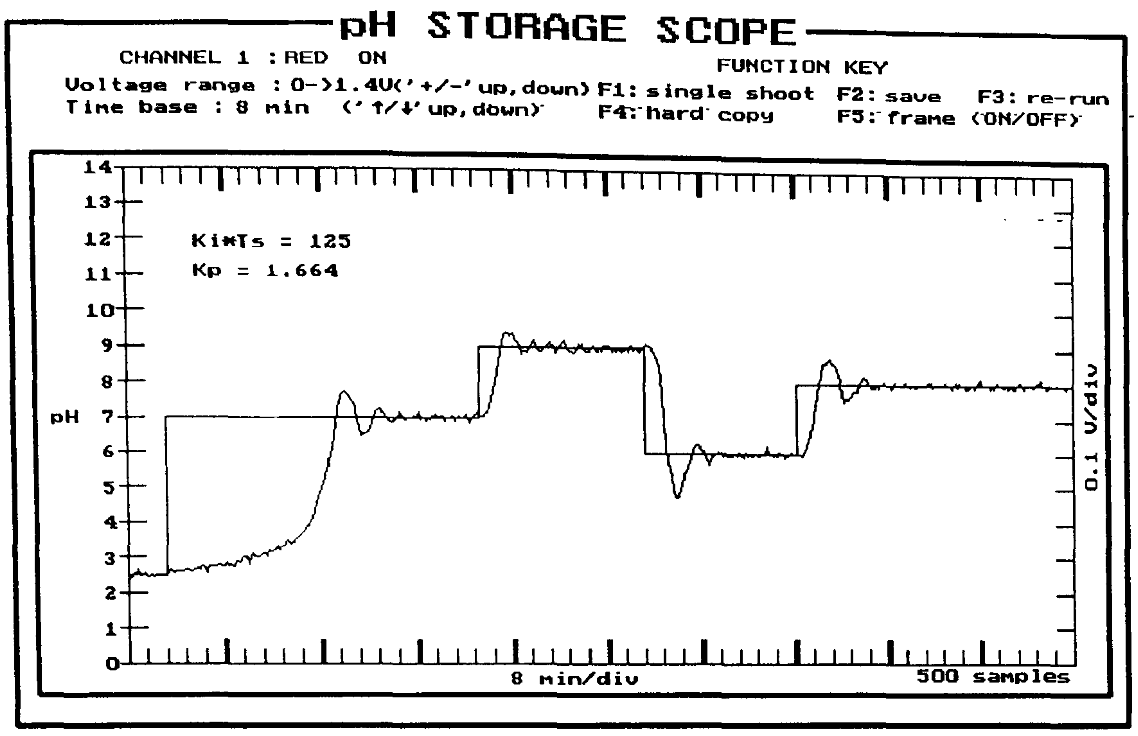


Figure 7-1. (a) The control of pH at pH=7 using a digital PI controller.
(b) The same plant controlled by the fuzzy controller developed in this thesis.

Set-point variation experiments were also performed for both the digital and the fuzzy controllers. The results of these experiment are shown in Figure 7-2. This again shows that the fuzzy controller has a better pH output response than that of the digital PI controller. The controller constants used in these experiments were $K_i \cdot T_s = 125$ and $K_p = 1.664$ for the digital PI controller, and E=0.2V, D=0.0003V and U=4V for the fuzzy controller. These values are the same as those used in Figure 7-1, although in section 7.6 it will be shown that these are not the optimum values for the fuzzy controller.

(a)



(b)

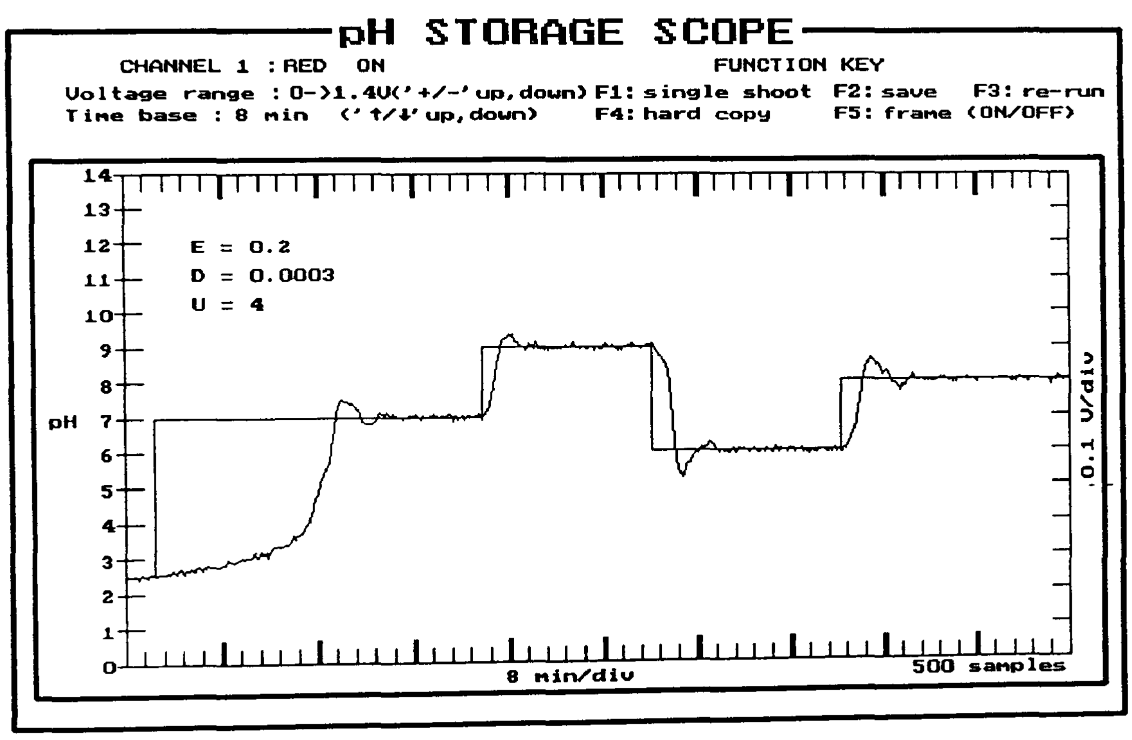
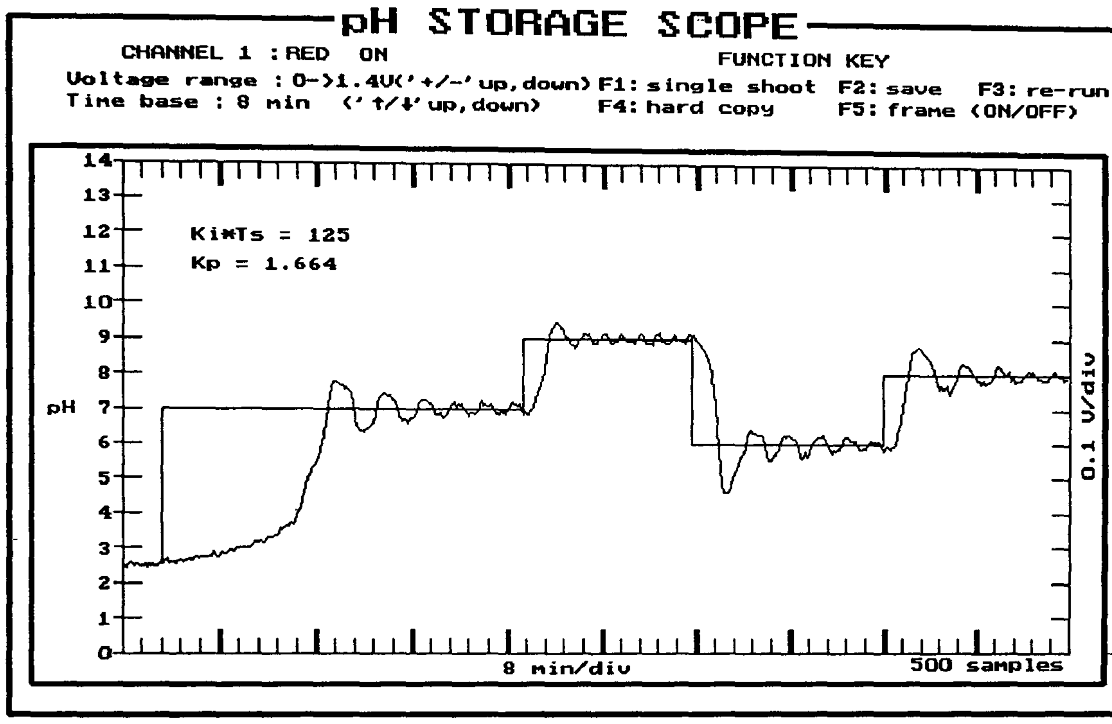


Figure 7-2. Response of a pH process under set-point variation when controlled by (a) a digital PI controller, (b) a fuzzy controller.

Although the response of Figure 7-2 shows the superiority of the fuzzy controller, if CO_2 contamination is considered, the improvement due to fuzzy control is even more marked. It was noted in chapter 6 that if a solution of NaOH solution is exposed to air for a period of time, CO_2 is absorbed by the solution and it becomes a weak acid. This was shown in Figure 6-7, in which it can be seen that the static gain changes drastically. If the experiment of Figure 7-2, is repeated for a plant with CO_2 contaminated, the resulting outputs are as shown in Figure 7-3. This shows that while the performance of the digital PI controller is degraded by the contamination, that of the fuzzy controller is little affected. Overall the fuzzy controller has a performance that is much better than the digital PI controller in both over-shoot and settling time, at each of the pH set-points. That shows that the fuzzy controller is more robust when the system changes.

(a)



(b)

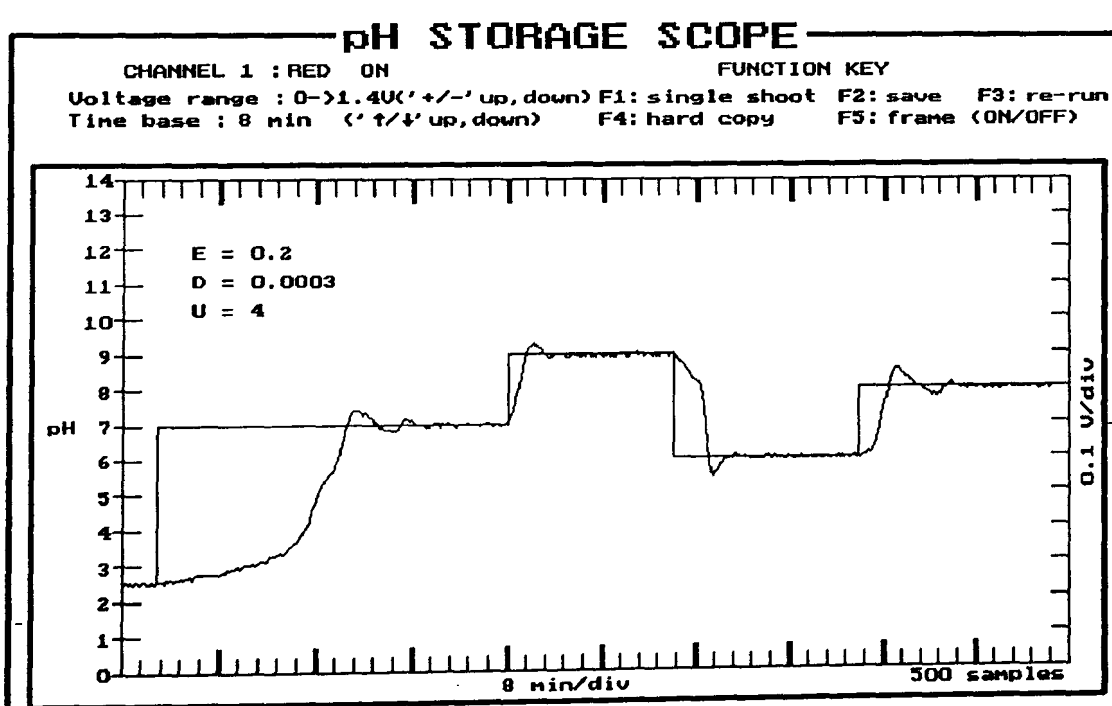


Figure 7-3 Results of an experiment similar to that of Figure 7-2 but after the NaOH had been exposed to air for seven hours. (a) for the digital PI controller, (b) for the fuzzy controller.

7.2 Comparing digital and fuzzy PI controllers for different value of K_i and K_p

Having demonstrated the overall superiority of the fuzzy controller we will now look in more detail at the performance of the digital and the fuzzy PI controllers for different values of their control parameters.

For pH=7, two experiments were performed, one with a large K_i and a small K_p, the other with a small K_i and a large K_p. Both the digital and the fuzzy PI controllers were used and, for comparison, the K_i and K_p values used were calculated from the corresponding E, D and U values of the fuzzy controller as described in Chapter 5. The position formula was applied by the digital PI controller algorithm to reduce the effects of noise. This was especially useful when K_p is very large.

An experiment was also performed to compare the two methods at pH=8. The experiments described in this section, and those described in section 7.3 to 7.5, are performed by pumping the HCl solution with pH $\approx$2.49 into the reaction tank at a constant rate (750 m*l*/min) and applying a step change to the set-point.

7.2.1 Digital and fuzzy PI control with large K_i and small K_p at pH=7

Figure 7-4 shows the performance of the conventional digital PI controller for value of $K_i \cdot T_s$ of 500, 250, 125 and 62.5, with value of K_p of 0.416, 0.832, 1.664 and 3.328. These values correspond to the E, D and U settings of the fuzzy PI controller shown in the corresponding graphs of Figure 7-5. From Figure 7-4, it is clear that a good choice for K_i and K_p is that shown in (c) with $Ki \cdot T_s$=125 and K_p=1.664. However, for the fuzzy controller, Figure 7-5 shows that the process pH output settles down more quickly at (b) and (d), where the settings are E=0.2, D=60, U=100, and E=0.2, D=3.75, U=25,which do not correspond to Figure 7-5 (c). Therefore, the relationship between the control constants of a digital PI controller and the fuzzy decision table ranges as

(a)
(b)
(c)
(d)

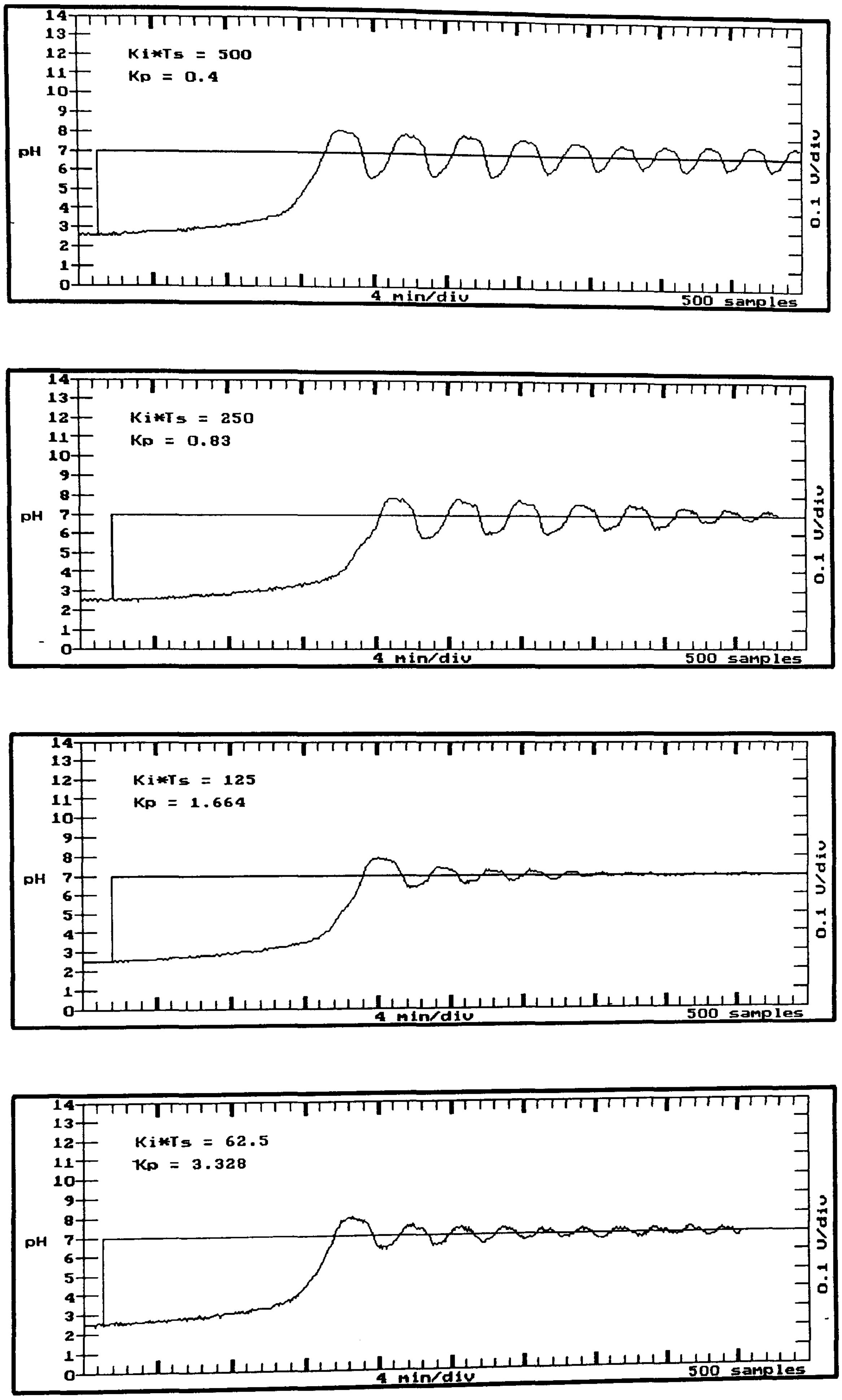


Figure 7-4. pH process controlled by the digital PI controller with large K_i and small K_p.

(a)
(b)
(c)
(d)

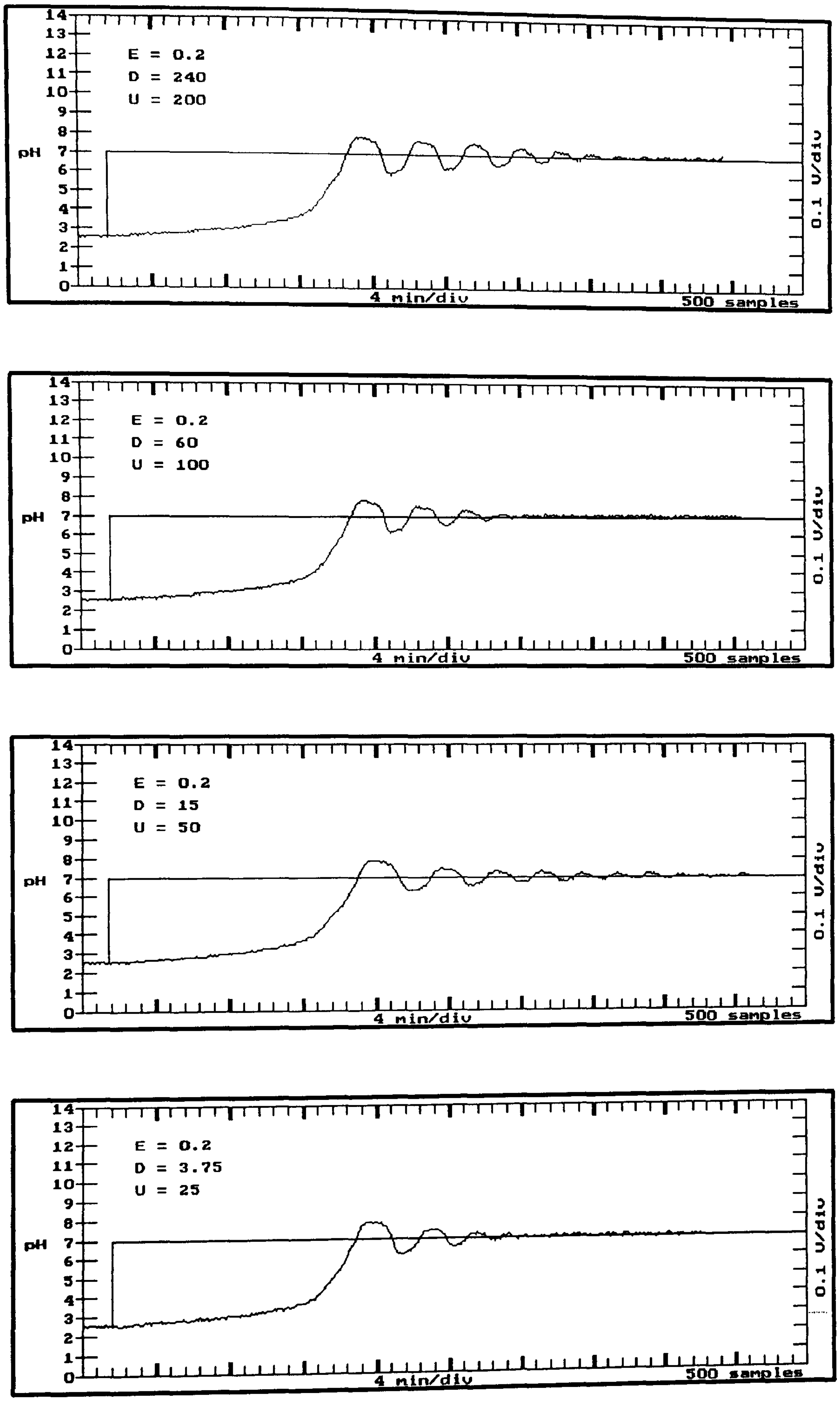


Figure 7-5. results of fuzzy pH process control with the value of the range E, D and U corresponding to the value of K_i and K_p in Figure 7-4.

described in chapter 5, are not valid in this pH control process. It is not clear why there are two similar pH responses as shown in Figure 7-5(b) and (d). The effect of the choice of range U on the control response will be discussed later in section 7.3. However, it was noted in chapter 5 that when D is too large, the fuzzy controller behaves like an integral controller. Therefore, these responses may be present different modes of operation of the controller.

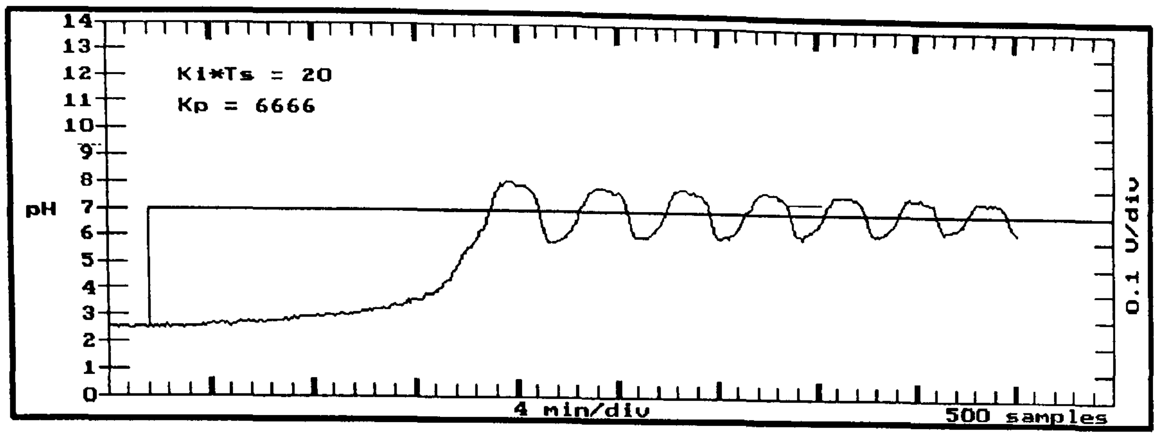
7.2.2 Digital and fuzzy PI control with small K_i and large K_p at pH=7

Experiments were next performed to investigate the performance of the controllers for small value of K_i and large values of K_p, again at pH=7. This experiment produced some rather unexpected results.

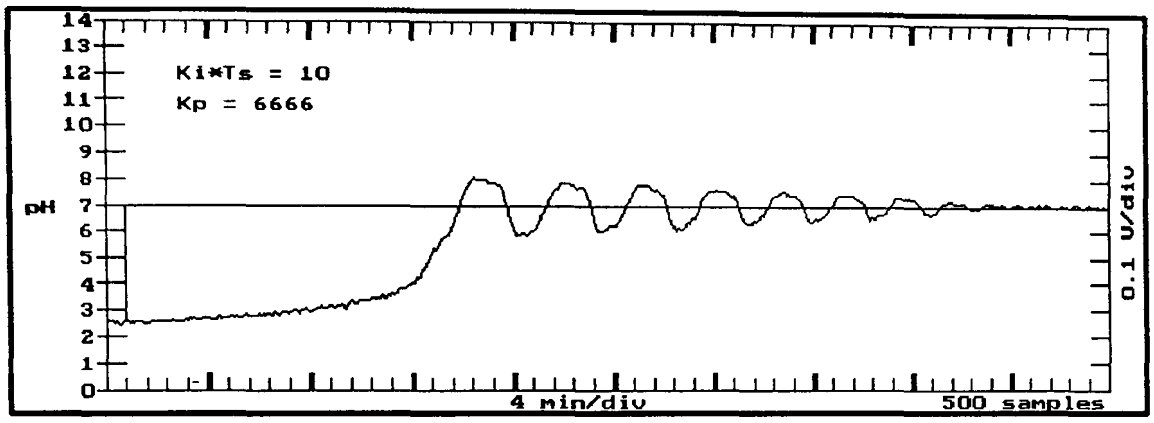
Figure 7-6 shows the results for the digital controller. It can be seen that the pH outputs oscillates around the set-point with a large limit circle if the constant $K_i \cdot T_s =$ 20 and K_p is set to 6666 (see Figure 7-6(a)). However, if the value of $K_i \cdot T_s$ is reduced to 10, the system response will settle down after 8 cycles as shown in Figure 7-6(b). However, this damping effect reaches its maximum when K_p is doubled, keeping $K_i \cdot T_s =$ 10. As shown in Figure 7-6(c) the settling time is reduced to about 4 cycles.

The experiments of Figure 7.6 were repeated using a fuzzy controller with corresponding values of E,D and U, and the results obtained are shown in Figure 7-7. This shows a much better response than that obtained by the corresponding digital PI controller. The reasons for this will be discussed in later sections.

(a)



(b)



(c)

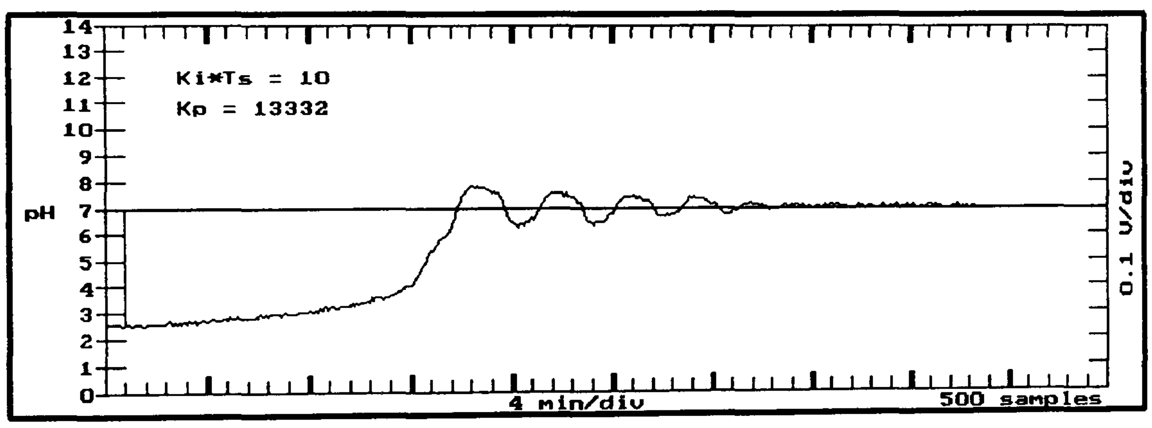


Figure 7-6. The pH process controlled by the digital controller with small K_i and large K_p.

(a)

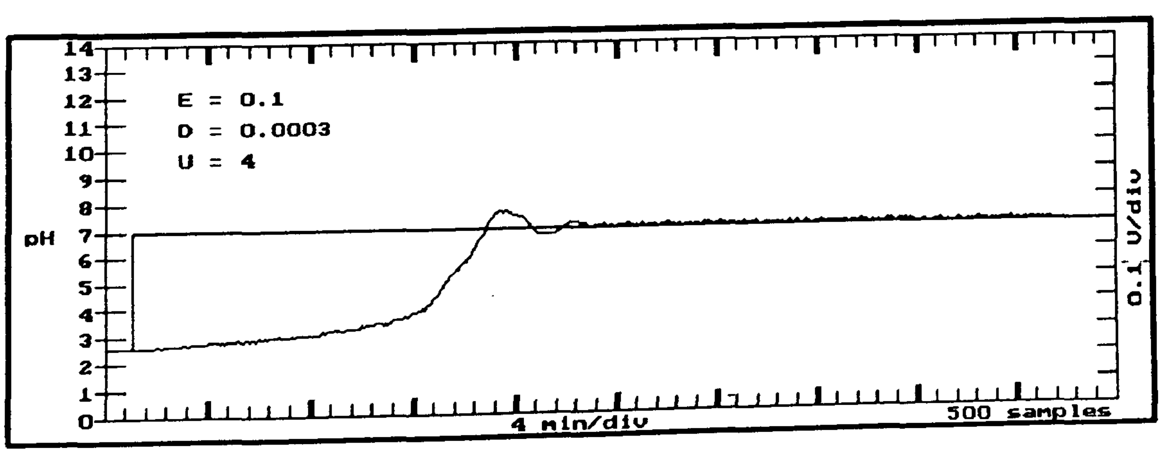
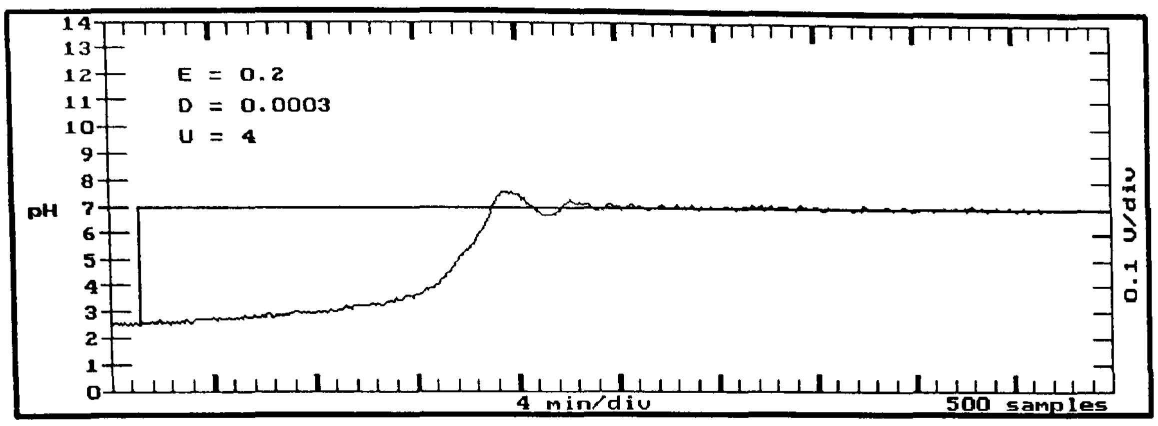


Figure 7-7 (a). Result of fuzzy pH control with parameters corresponding to K_i's and K_p's of Figure 7-6.

(b)



(c)

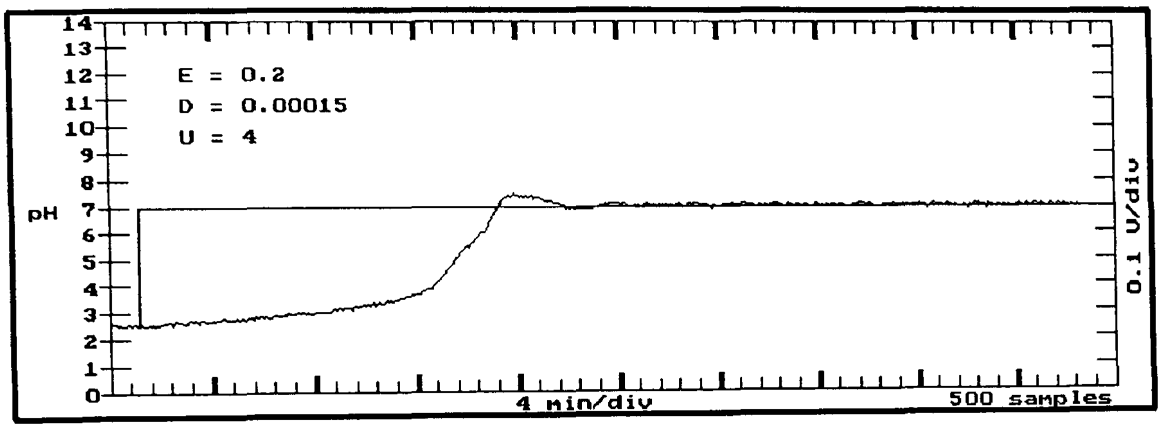
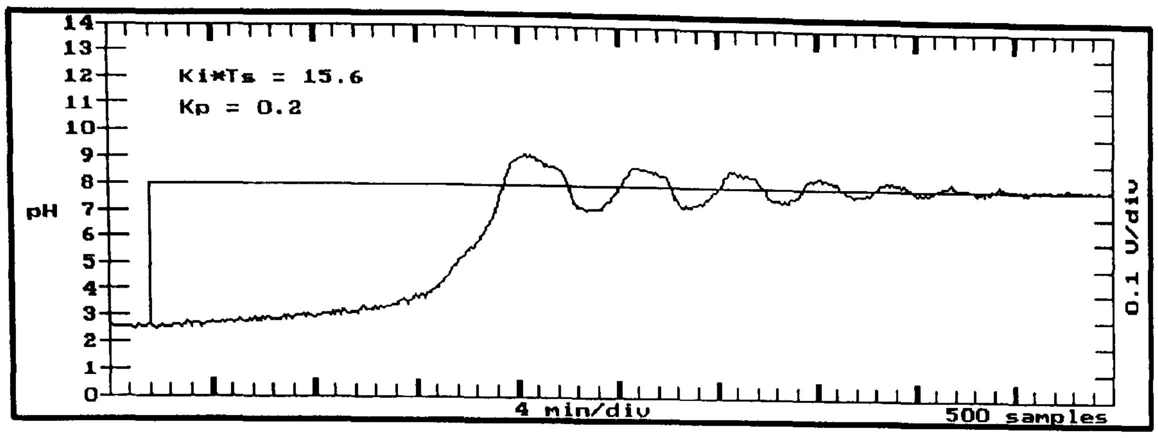


Figure 7-7. Results of fuzzy pH process control with parameters corresponding to the K_i's and K_p's of Figure 7-6.

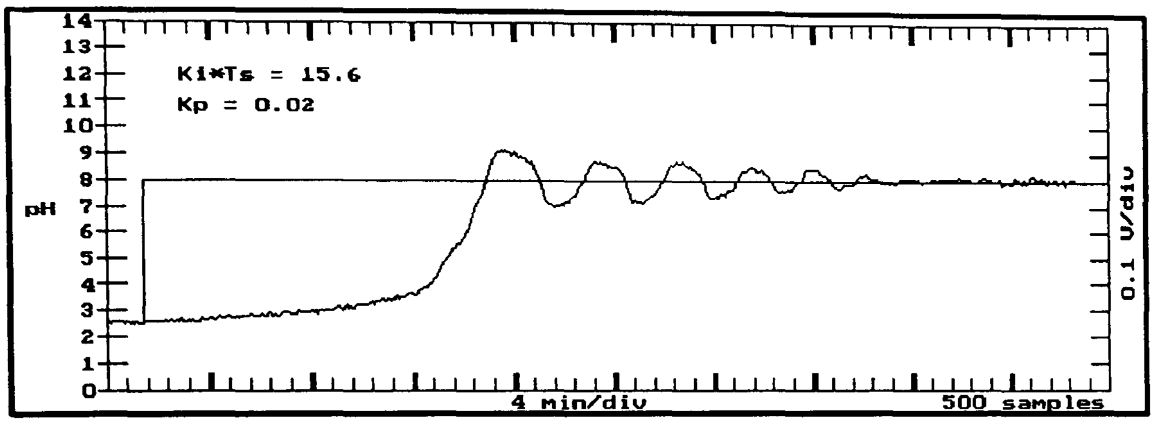
7.2.3 Comparing digital and fuzzy control at pH=8

For pH=8, several experiments were performed to compare digital and fuzzy control. For the digital controller these kept $K_i \cdot T_s$ equal to 15.61 and set K_p to 0.2, 0.02 and 0.002, while for the fuzzy controller used corresponding values of E,D and U. The results of these experiments are shown in Figures 7-8 and 7-9. It can be seen that when the digital controller is applied, damping increases with decreasing K_p as shown in Figure 7-8. However, for the fuzzy control case, as shown in Figure 7-9, a good performance is obtained at D=156, which corresponds to K_p=0.02. Therefore, the relationship between the control constants of these two controllers again fails to follow that suggested in Chapter 5, but again fuzzy controller proves superior to the digital controller.

(a)



(b)



(c)

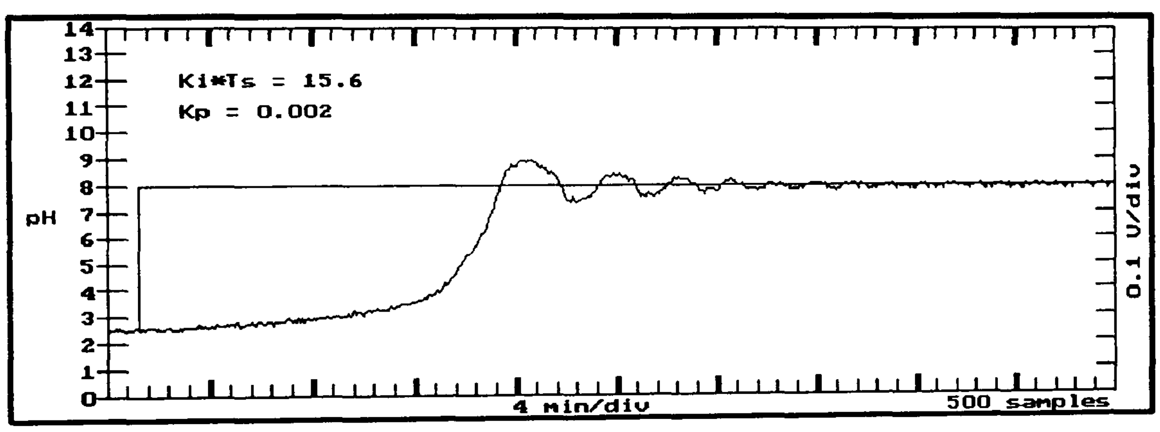


Figure 7-8. Digital control of the pH process at pH=8.

(a)

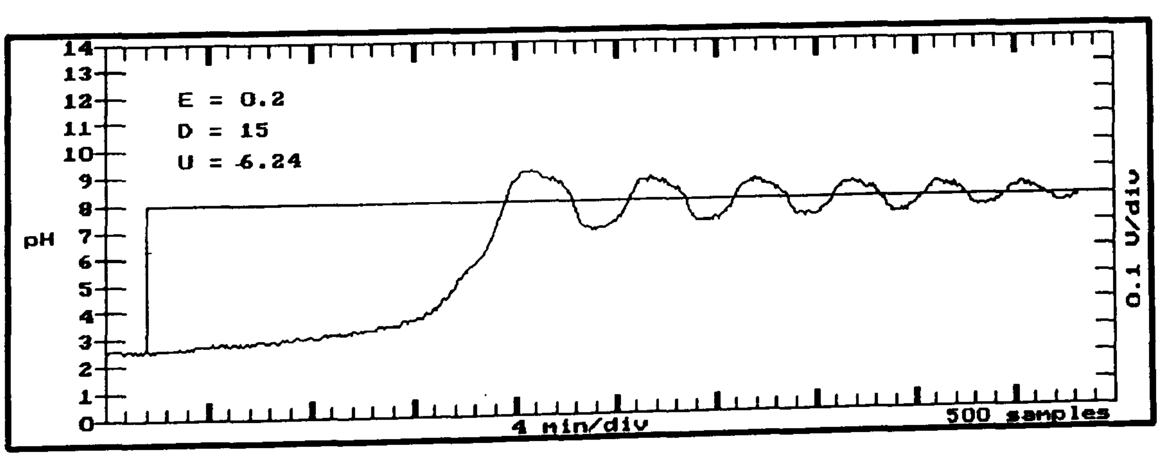
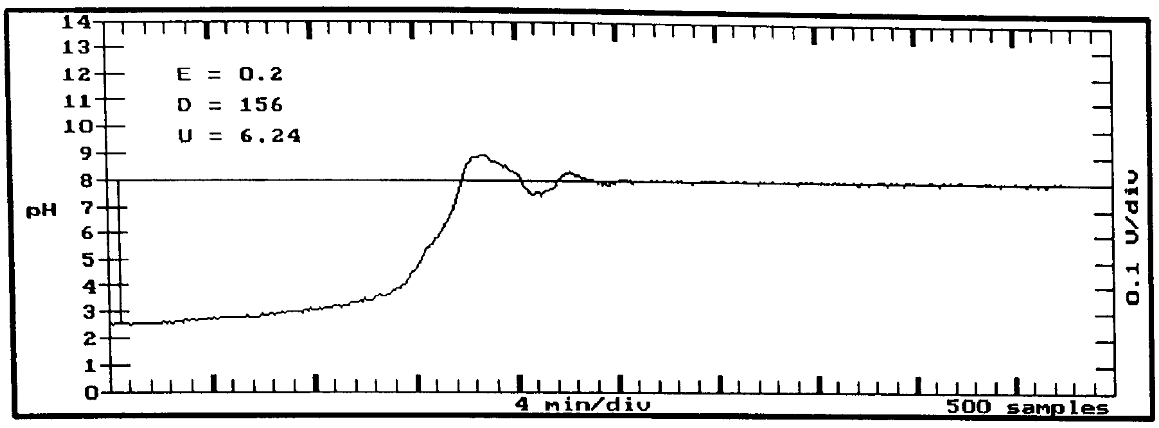


Figure 7-9(a) Result of the pH process at pH=8,with E,D and U chosen corresponds to K_is ad K_p's of Figure 7-8.

(b)



(c)

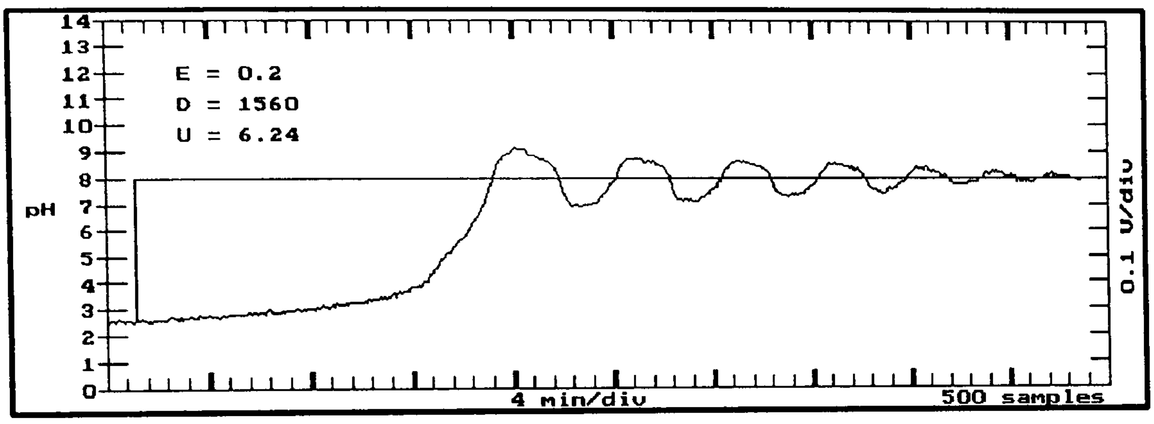


Figure 7-9. Fuzzy control of the pH process at pH=8,with E, D and U chosen to correspond to the K_i and K_p values of Figure 7-8.

7.3. The effects of the choice of U on the fuzzy pH process control response

To investigate the effects of the choice of U on the pH process response, an example from the previous section was chosen for further study. The response selected is that shown in Figure 7-5(b). This example uses values of E=0.2V, D=60V and E=100V, which correspond to $K_i \cdot T_s=250$ and $K_p=0.9$. From the figure it is clear that in response to a step change in the set-point to pH=7, the output settles down quickly.

The experiment was repeated keeping the value of E and D unchanged, but reducing U by a factor of 10 to 10V. The results obtained were quite different, with the output response oscillating about the set-point with a very large amplitude failing to settle down.

To investigate this behavior it is useful to consider the control trajectory on the fuzzy decision table, together with the fuzzy controller outputs and the error signal. These are all shown together in Figures 7-10 and 7-11 for these two experiments.

Note that the scales used for the control output are different in these two figures, It can be seen that although D is 60V in theses two figures the actual maximum error changes are only about 0.0002V, therefore, the control trajectory moving in the center of the decision table looks like an integral controller. However, since U is different by a factor of 10 between these two cases, their control outputs are not equal. The maximum accumulated output value for U=100V is 50V but it is about 14.4V when U=10V. This means that the system needs a greater damping force to settle down the process around the origin in the latter case. To see the differences between these two response more clearly, we can reduce the range of the change of error of the phase planes. Figure 7-12 and 7-13 show the phase plane trajectory redrawn with this range reduced to 0.0003V. This figure shows that when U=10V the trajectory moves vertically and horizontal around the origin. However, when U=100V, the trajectory moves back and forth rapidly along the error=0 line at the origin. This kind of oscillation probably results from the pH process having a very large static gain K near pH =7 (see figure 6-5).

The "kink" in the control output of Figure 7-14 is an anomaly introduced by the action of the anti-windup integrator used.

In Figures 7-14 and 7-15, the trajectories of Figure 7-12 and 7-13 are replotted but for clarity only the first 500 and 750 sampling points are shown. The region of interest in these figures is the center square in each of the phase plane plots. In the case where U=100V (Figure 7-14) the control action provides sufficient damping force from the negative part of the decision table to make the control trajectory pass through the origin at the error=0 point. This is shown in the figure by the trajectory going to the right hand side of the center square of the phase plane plot. When U=10 (Figure 7-15) there is insufficient damping to make the trajectory pass through the origin. In this case the control trajectory simply moves around the center of the decision table.

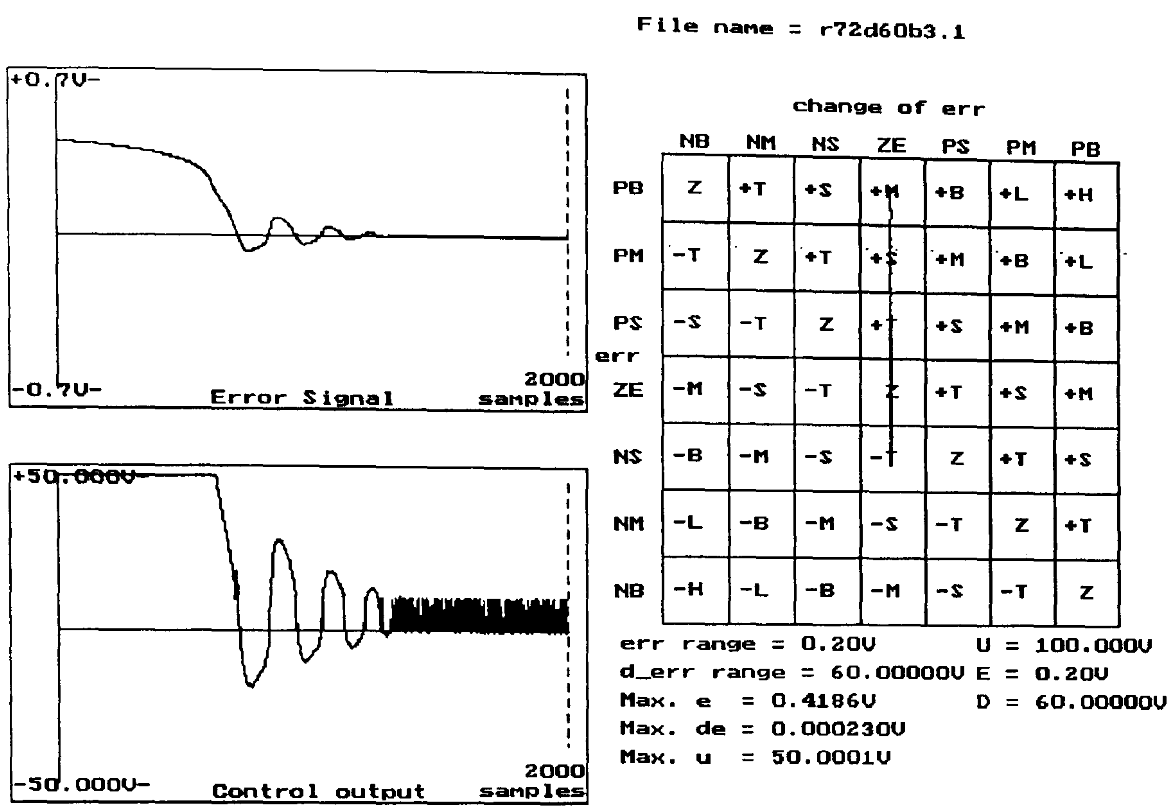


Figure 7-10 Fuzzy control of the pH process using E=0.2V,D=60V and U=100V.

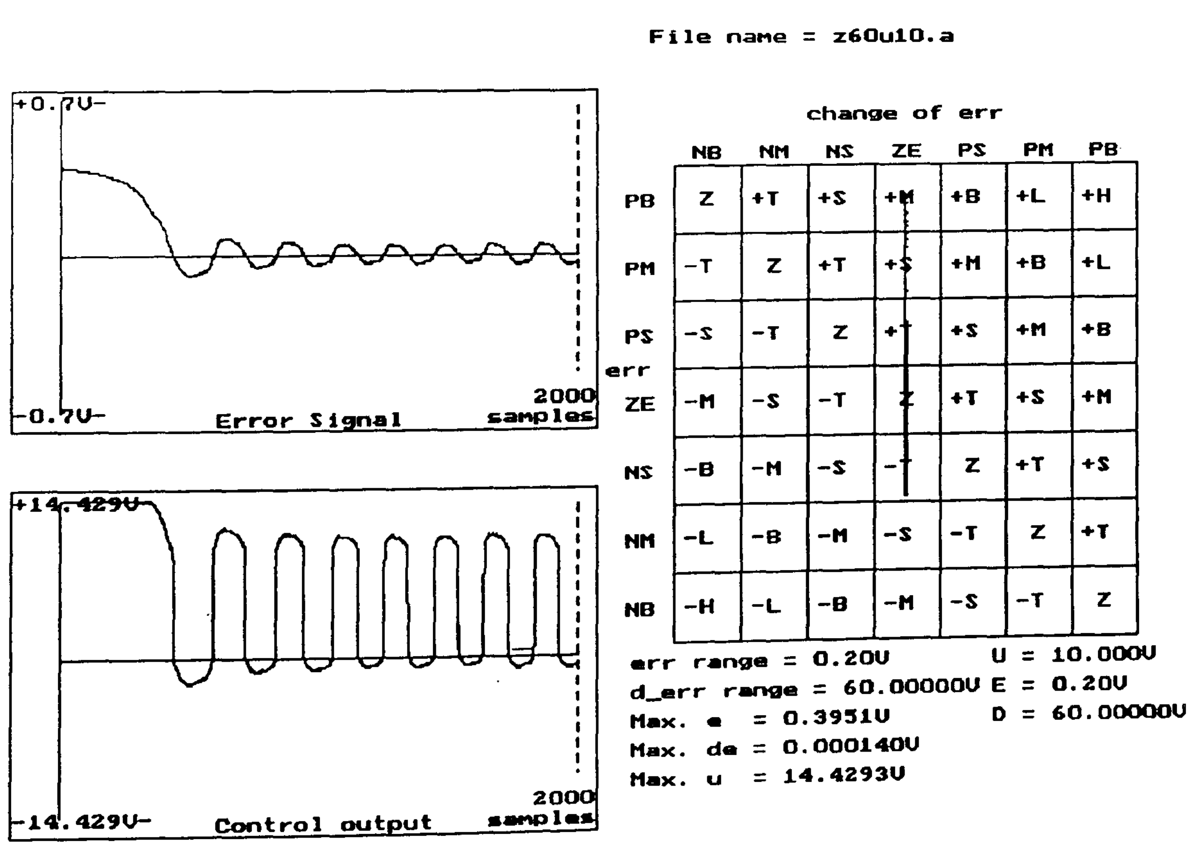


Figure 7-11. Fuzzy control of the pH process using E=0.2V,D=60V and U=10V.

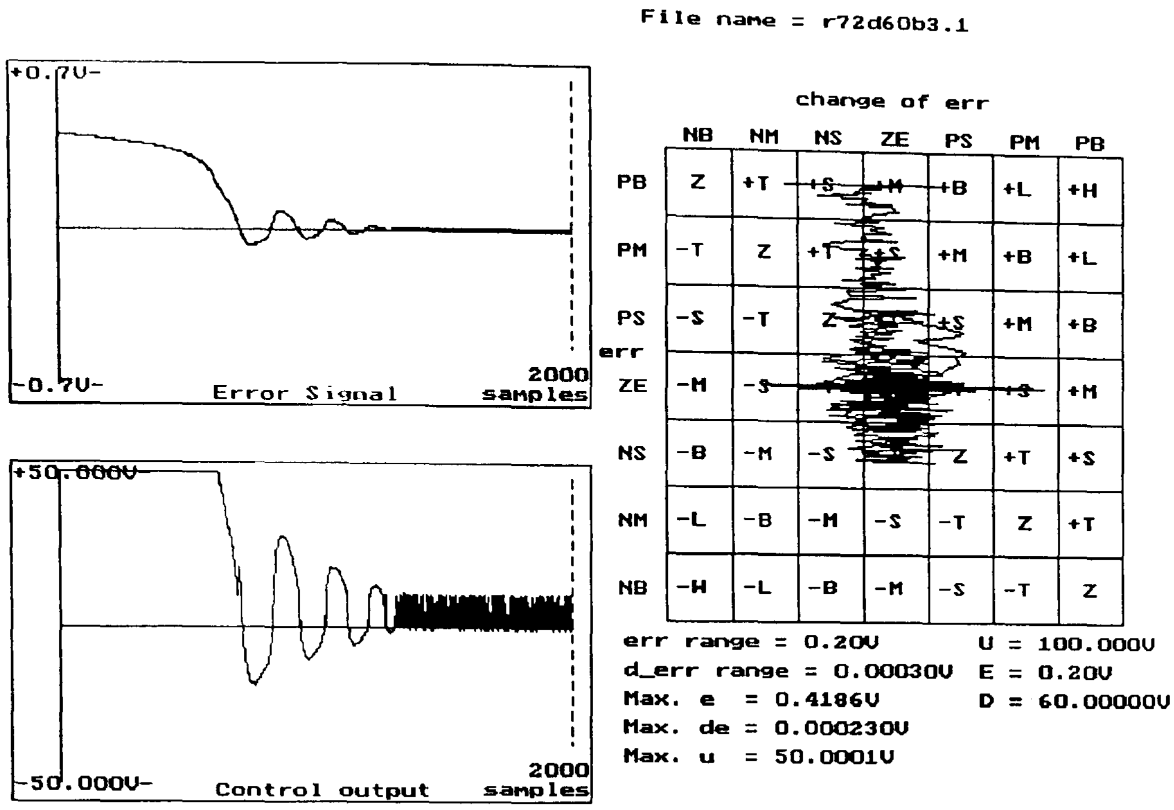


Figure 7-12. Figure 7-10 redrawn with the error change range reduced to 0.0003V (U=100V).

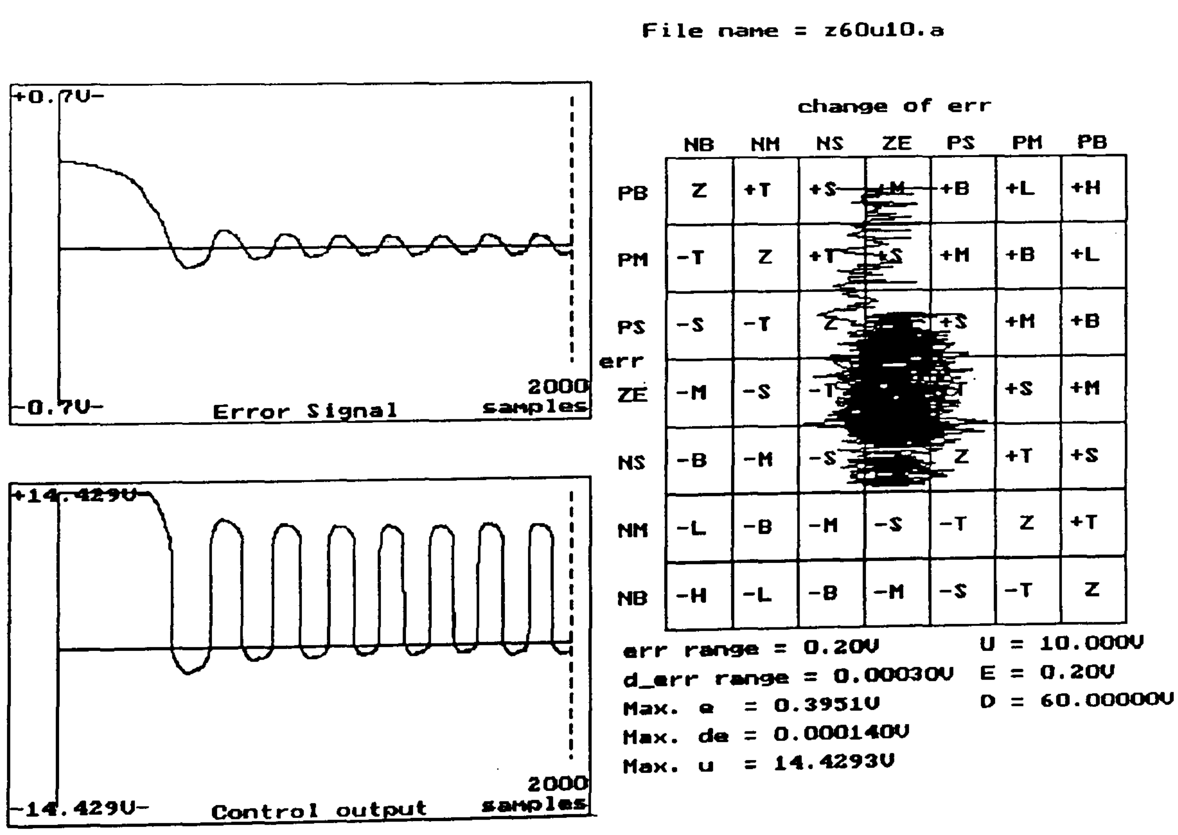


Figure 7-13. Figure 7-11 redrawn with the error change range reduced to 0.0003V (U=10V).

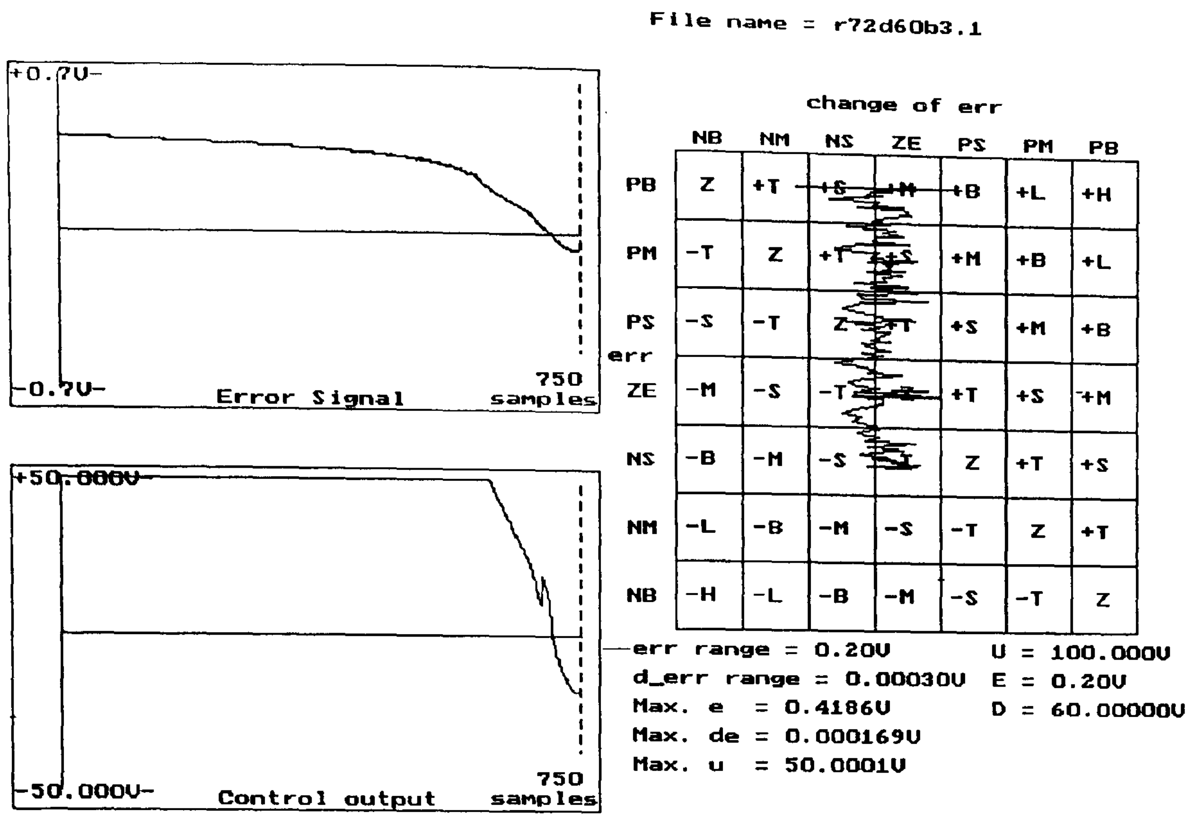


Figure 7-14. Figure 7-12 redrawn using less sample points (U=100).

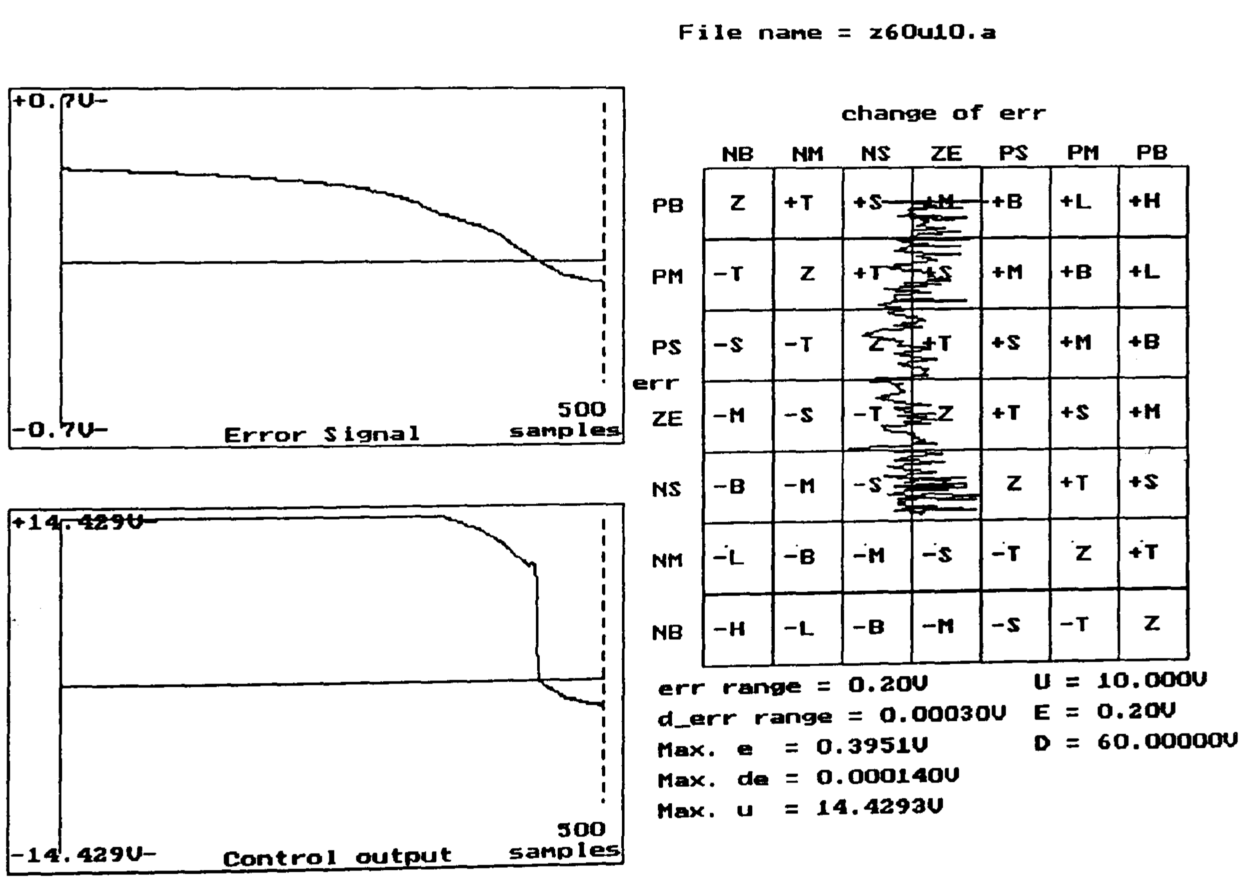


Figure 7-15. Figure 7-13 redrawn using less sample points (U=100).

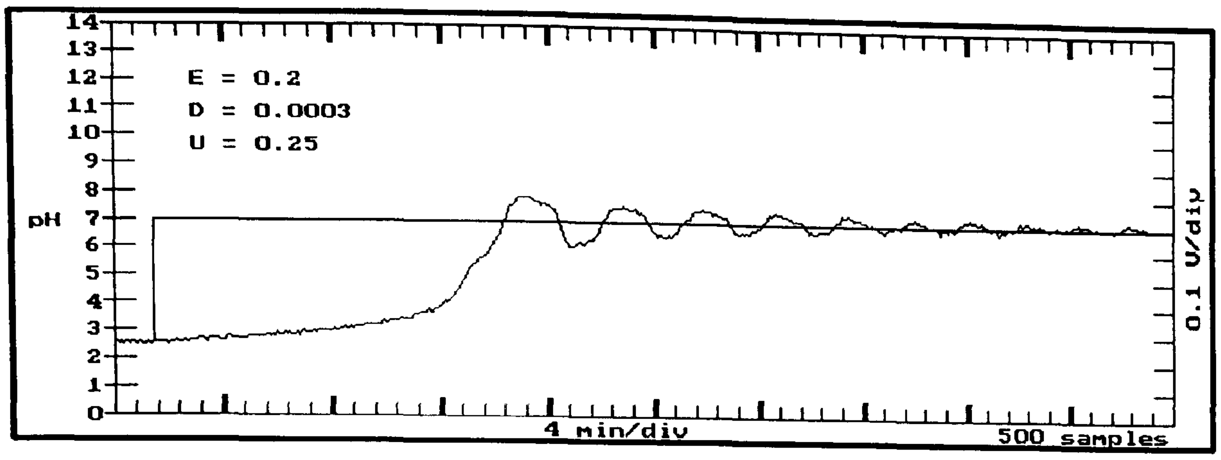
The noise like oscillation exhibited by the controller when U=100V causes the controller output to chatter at the origin with the average of its force just keeping the process pH at the set-point. This phenomenon suggests that there may be a way of controlling the pH process such that it settles down to its set-point. If the range D is set small enough, before reaching the error=0 line, the noise like oscillation will drive the control action trajectory into the negative part of the decision table, thus bringing a strong damping force to the system. In both Figures 7-10 and 7-11, the recorded maximum error change value is about 0.0002V. Thus, reducing D to 0.0003V would seem to be a good choice and will be discussed in the next section.

7.4 An investigation of the effects of setting the error change range D to 0.0003V or less

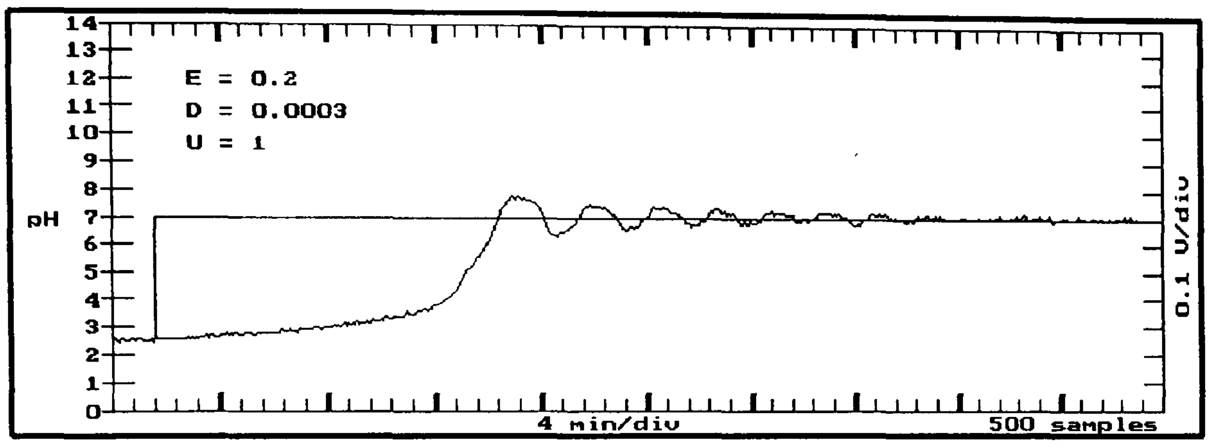
In this section, a series of experiments is described which show how the pH control process responds when the error change range D of the decision table is set to 0.0003V. Figure 7-16 shows the performance of the system when E=0.2V, D=0.0003V and U is varied. It can be seen that the system settles down quickly to the set-point pH=7 when U=0.2.The response is slightly worse when U is increased to 4, and is much worse if U is decreased to 1.

Figure 7-17 shows the results of reducing both the E and the D ranges to half, their value in the previous experiment. It was noted in chapter 5 that a similar system response will be obtained if E, D and U are reduced in the same ratio. This can be seen by comparing the response of Figure 7-17(a) for E=0.1V, D=0.00015V and U=2V, with that of Figure 7-16(d) where E=0.2V, D=0.0003V and U=4V.

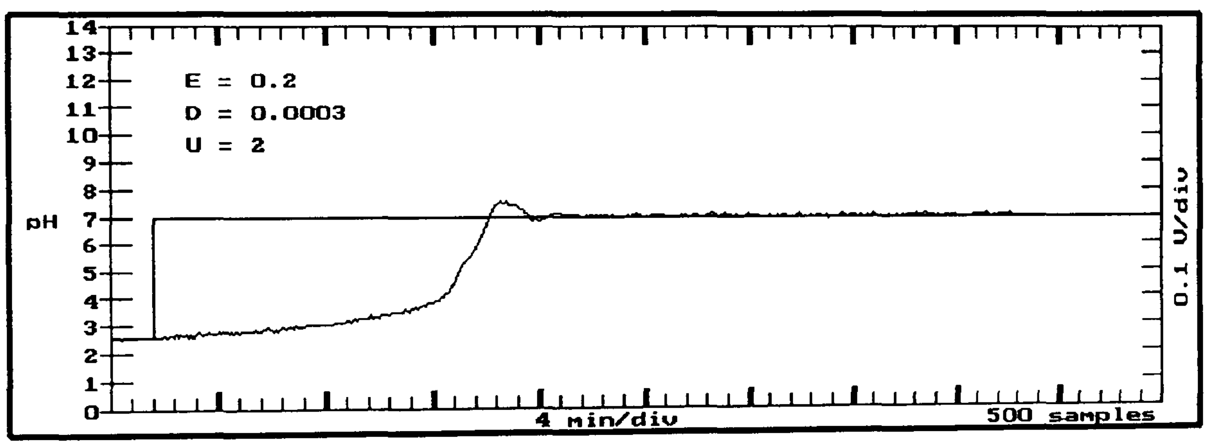
(a)



(b)



(c)



(d)

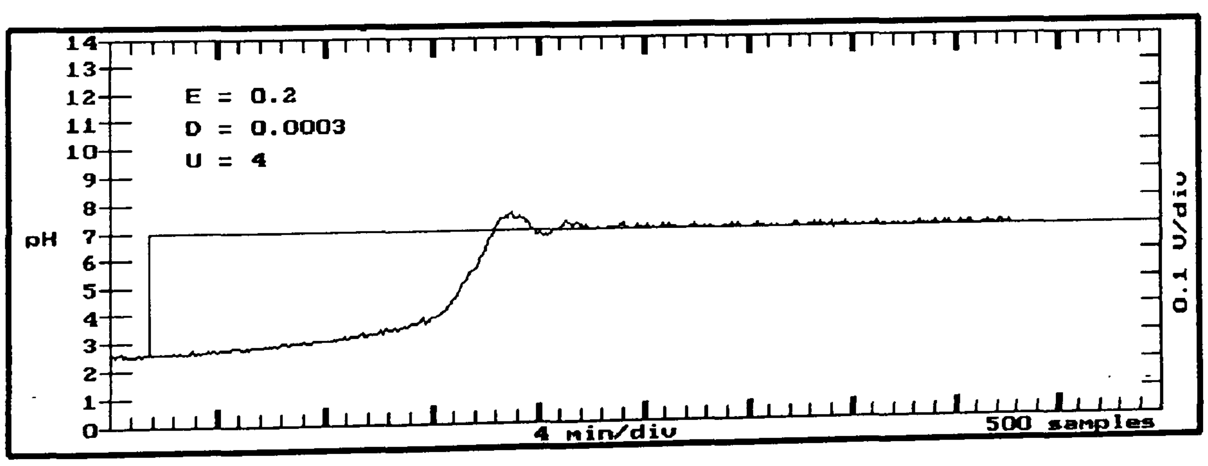
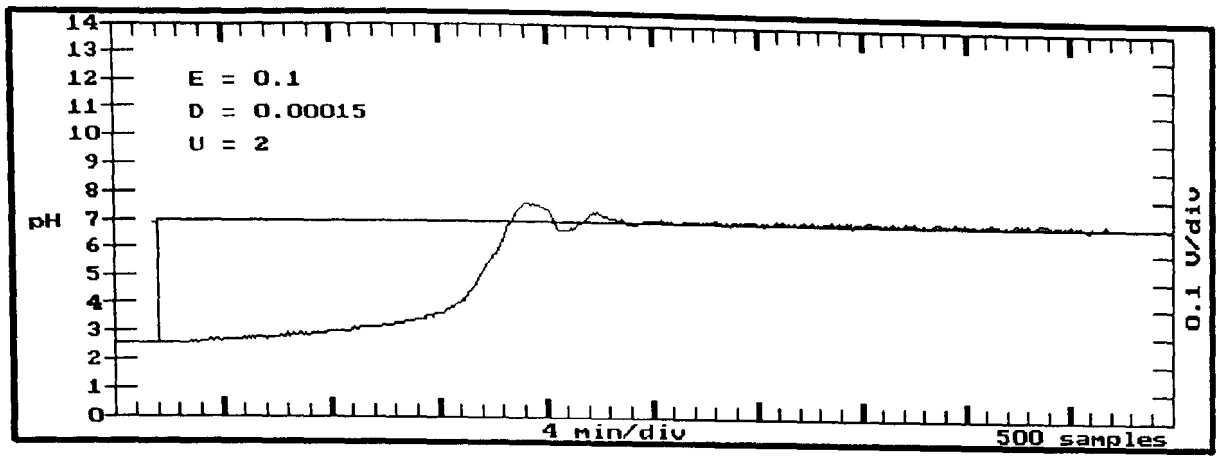
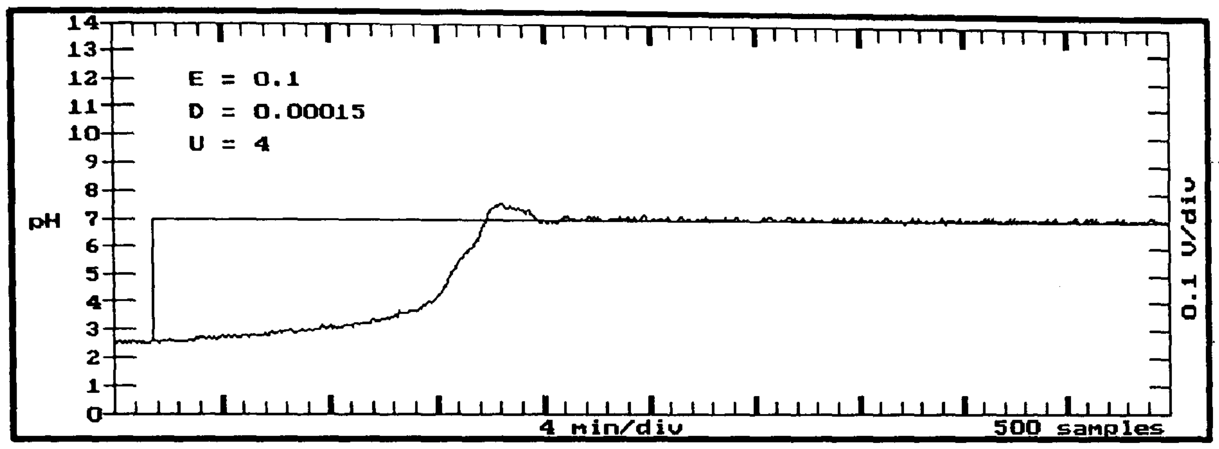


Figure 7-16. Responses of fuzzy pH process control when E=0.2Vand D=0.0003V.

(a)



(b)



(c)

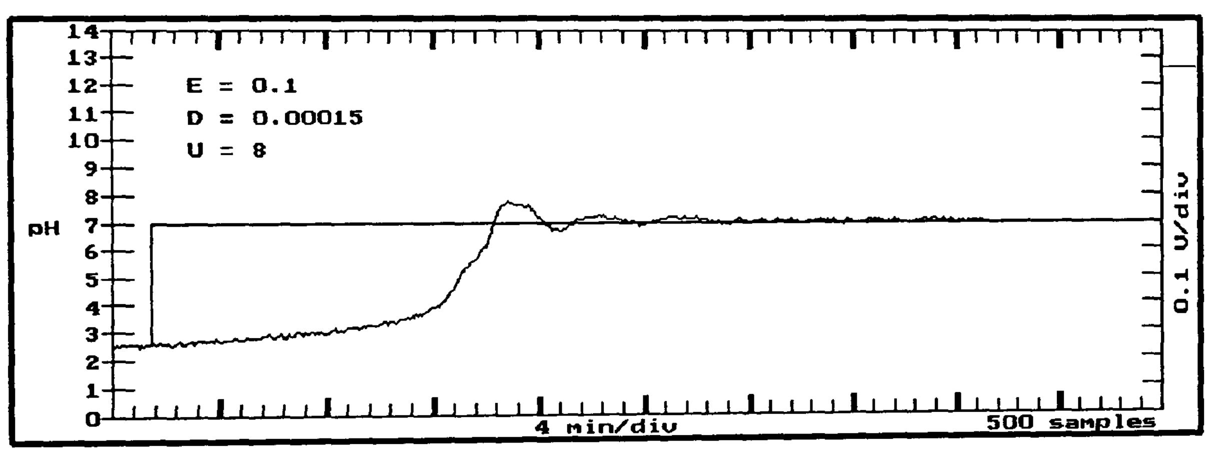


Figure 7-17. Responses of fuzzy control when E=0.1Vand D=0.00015V.

7.5 Enlarging the D range while keeping the error range E constant

For most of the experiments performed in previous sectionw the error range E was kept equal to 0.2V while the error change D was varied. This value of E was chosen since it provides a reasonable ±2pH range for the error change. In this section, the noisy oscillation area of the fuzzy control decision table near the set-point is studied, to see if there are any other values of D other than 0.0003V, that can be used to obtain similar good results.

7.5.1 The relationship between the strength of the fuzzy controller output and the sweeping area of the decision table

The strength of these fuzzy controller output is determined by the position of the control action trajectory on the fuzzy decision table. Fluctuations make the trajectory sweep out some area. Figure 7-18 shows two shaded boxes A and B which represent areas swept out by different fragment of the trajectory. The area to the right of the diagonal line in the decision table corresponds to a positive output while that below the line corresponds to a negative output. Since box A covers equal positive and negative area of the decision table, the accumulated output value (produced by integrating the output over the period corresponding to the fragment) will be equal to zero. However, in box B, the positive part is larger than the negative part, and the accumulated output will be positive. This output will act on the system so as to reduce the error.

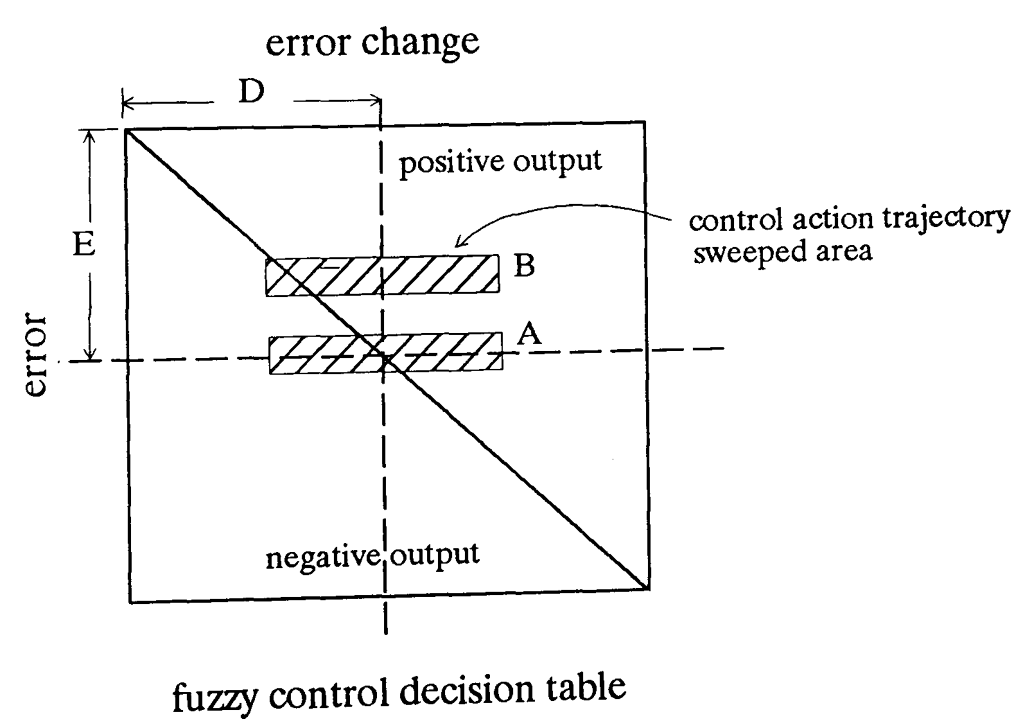
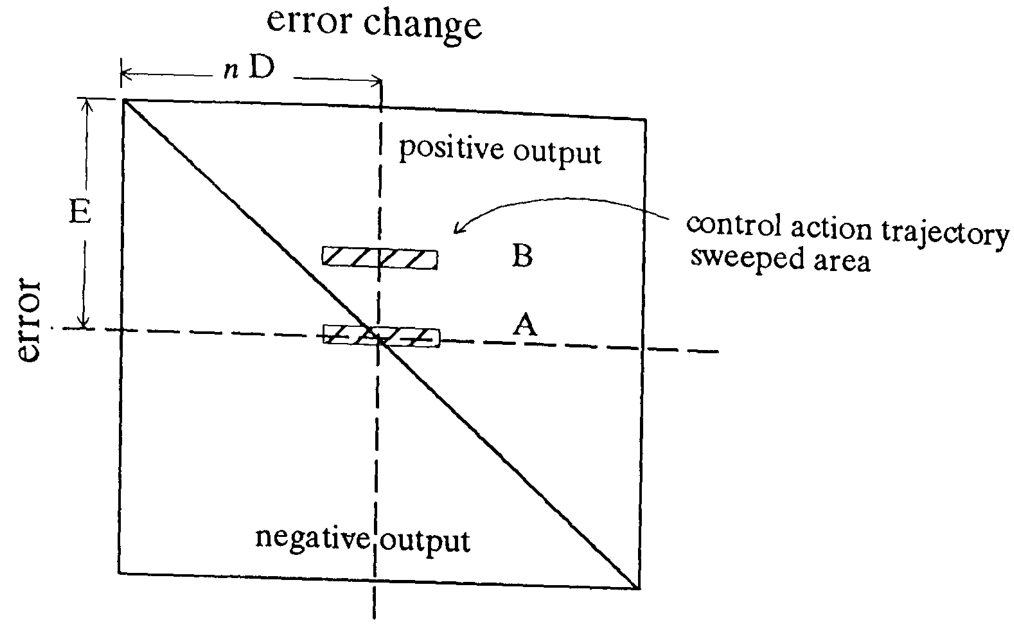


Figure 7-18. Sweeping areas within the fuzzy decision table.

Let us now consider the effect of expanding the range D by a factor n. Figure 7-19 shows the result decision table. The expansion of the D range of the table means that



fuzzy control decision table

Figure 7-19 Reducing the swept area by expanding D.

the area swept out by the fragment of the control trajectory are correspondingly smaller. Thus the accumulated output value is reduced as the D range is expanded. As the trajectory moves away from the diagonal of the decision table its effect on the control output increases (as shown in Figure 4-7). Therefore, the effect of expanding the D range of the table would be expected simply by considering the reduction of the swept area. It has been found that increasing D by a factor of n is equivalent to reducing U by a factor of n^3. This relationship is investigated experimentally in the next section.

7.5.2 An experimental study of the relationship between D and U

In Figure 7-16(c) we looked at the response of the system when E=0.2V, D=0.0003V and U=2V. Figure 7-20 shows a set of experiments obtained with the same value of E but with D increased by a factor of 4 to 0.012V. Figure 7-20 shows responses for value of U of 4, 16, 64, 128 and 256, and from these curves it is clear that a value of U=128V produces a response equivalent to that of Figure 7-16(c). This supports the assertion that increasing D by a factor of n will get the same result if U is increased in the same time by a factor of n^3. Thus increasing D by a factor 4 is equivalent to decreasing U by a factor of 64.

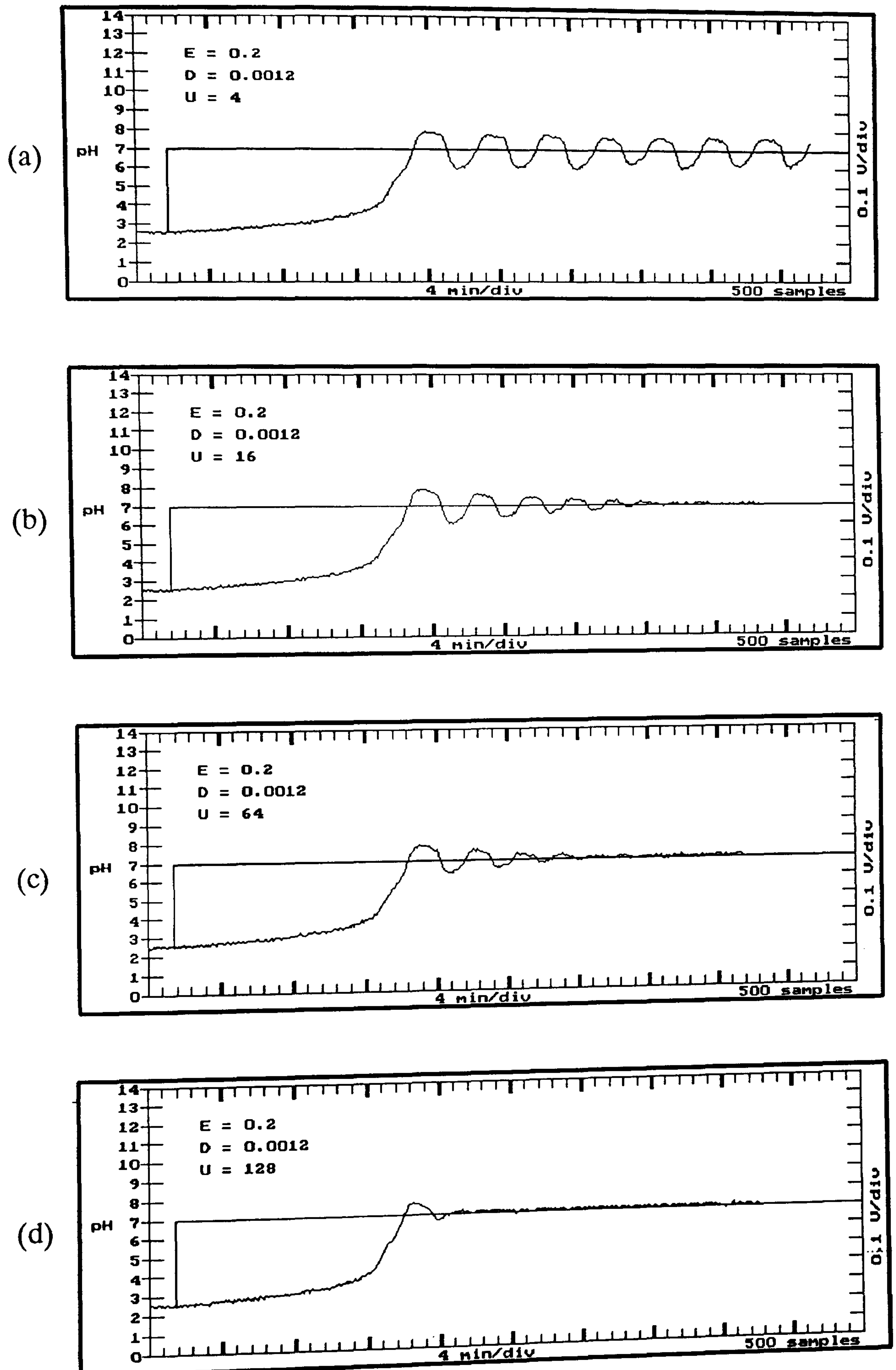


Figure 7-20. step responses when the range D expanded to 0.012V

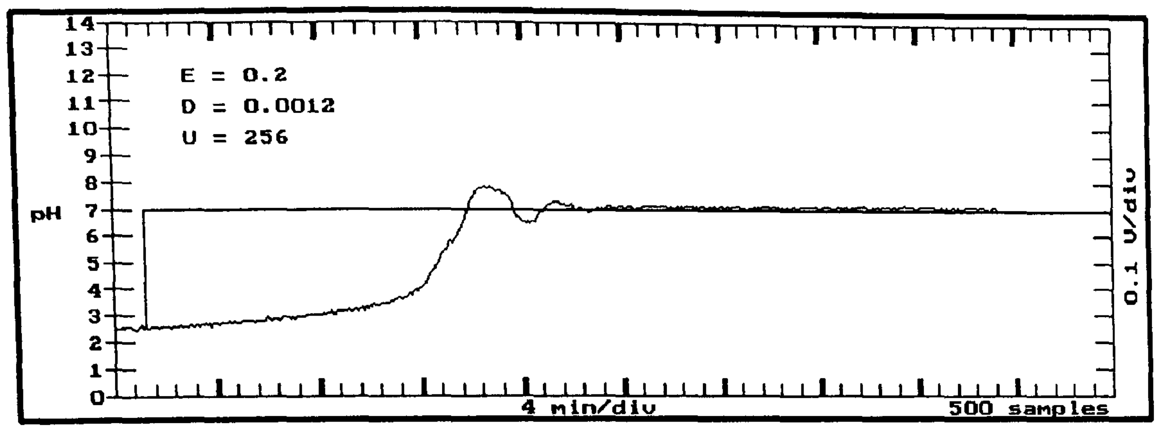
(e) 

Figure 7-20(e). response of the pH process control when D=0.0012V and U=256V

7.6 Fuzzy control at pH=5,9 and 10

In section 7.2.2it was shown that the pH process can be controlled very well at pH=7 by choosing a small value of D such as 0.0003V. Test were also carried out at pH=5,9 and 10, using a value of D of 0.0003V, keeping E constant at 0.2V and varying U. Experiment results are shown in Figures 7-21,7- 22 and 7-23.

Since the static gain of the pH process decreases as we move above or below a pH of 7, it follows that a higher value of U is required. Indeed, the results shows that at pH=5 the best choice for U is 24, while at a pH of 9 the best value for U is 12, and at pH =10 the best value for U is 36. Figure 7-21 shows the response of the pH plant at pH=5. The static gain is very low in this region and the system requires a high value of U to drive the plant effectively. Figure 7-21(a) shows the performance of the system when U=1. Here we see an obvious limit cycle around the set-point. Increasing the value of U decreases this effect as shown in Figure 7-21(b) and (c).

Figure 7-22 and 7-23 show the behavior of the fuzzy controlled plant for pH=9 and 10 respectively. For comparison, by choosing its control constant K_p and K_i the values corresponds to Figure 7-22(b), a digital PI controller experiment for set-point pH=10 is also shown in Figure 7-22(c). It is obvious that these two responses are almost the same.

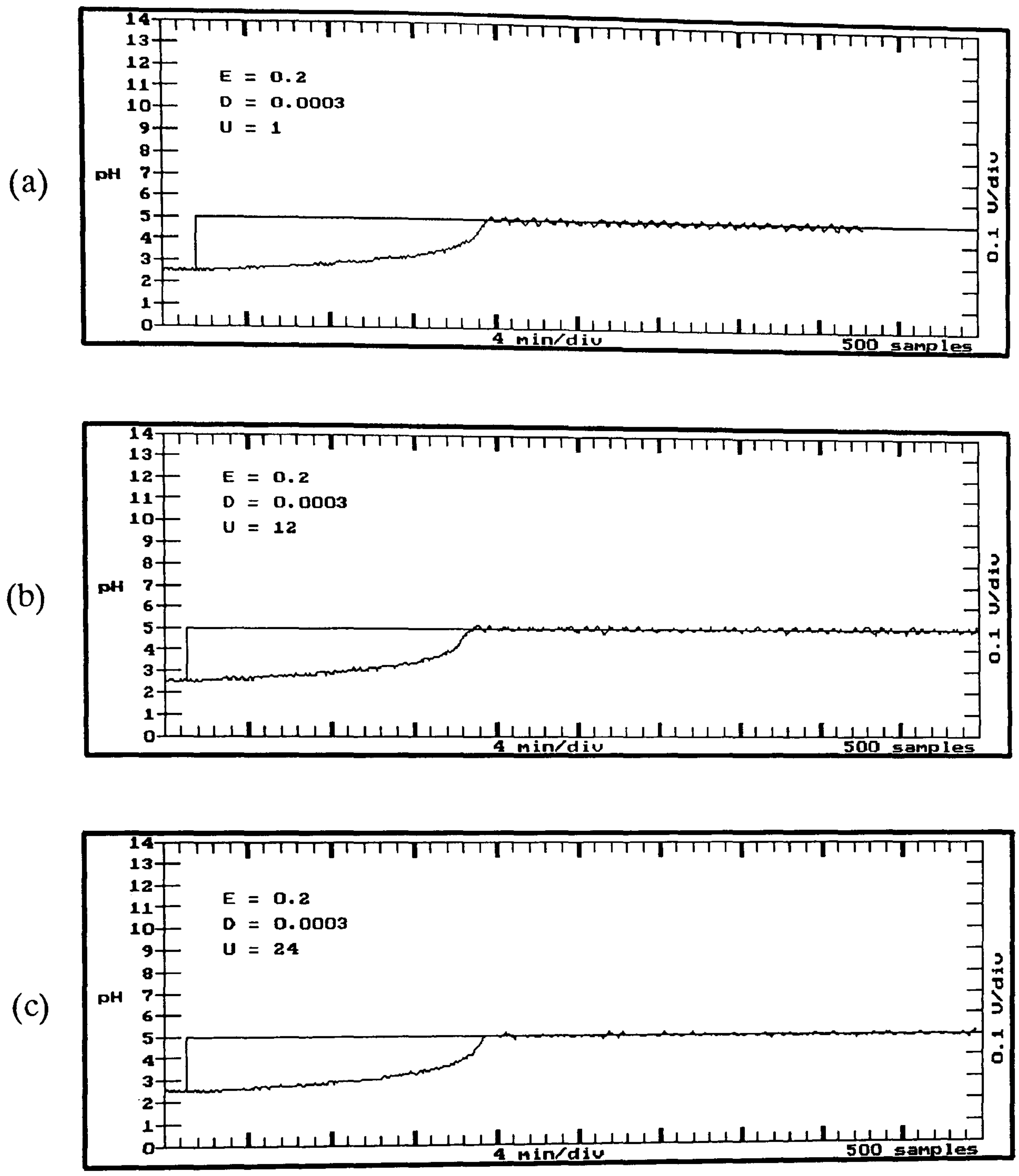


Figure 7-21. Using a fuzzy controller to control the process at pH=5.

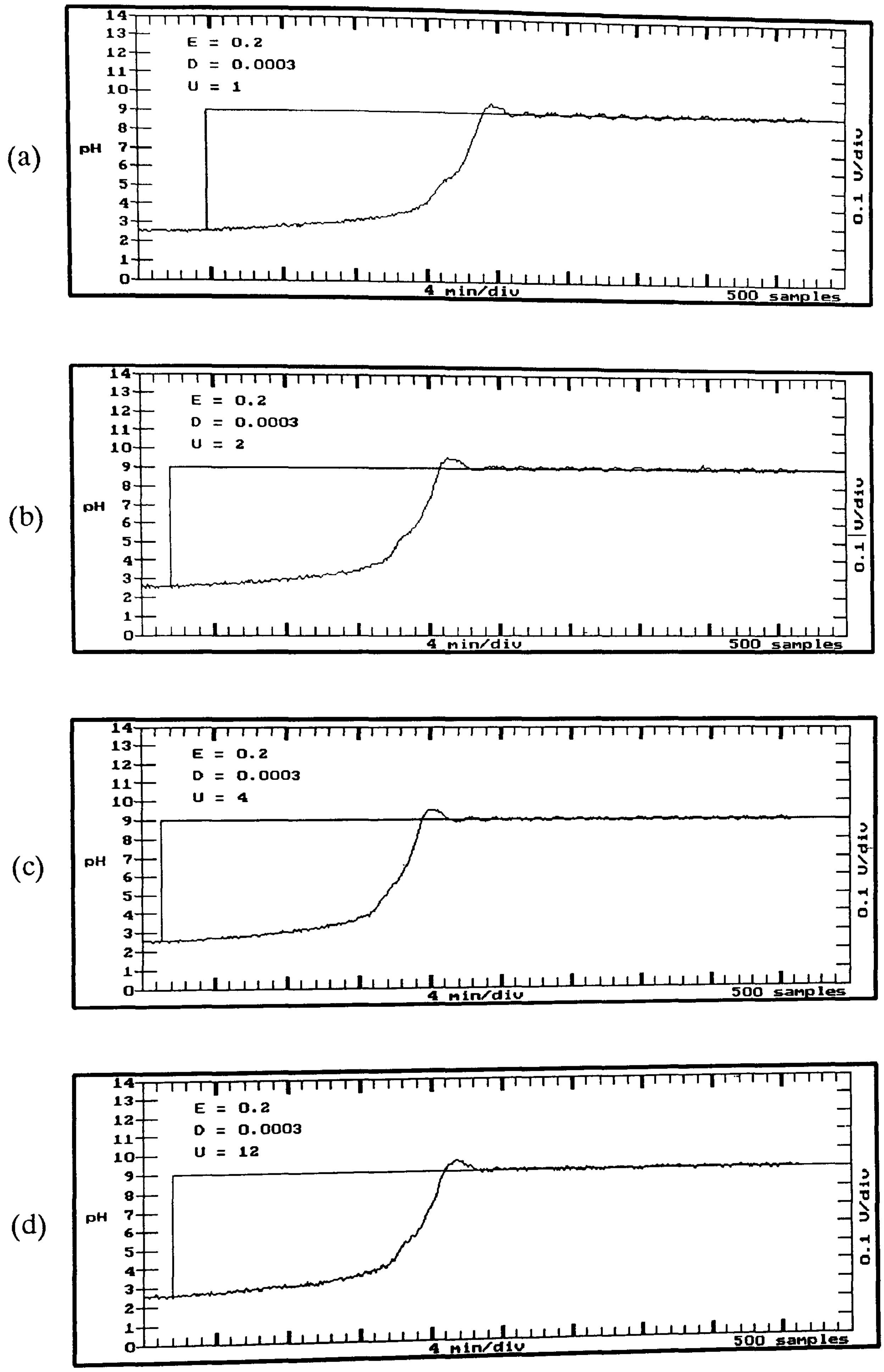


Figure 7-22. The fuzzy pH control response at pH=9.

(a)

(b)

(c)

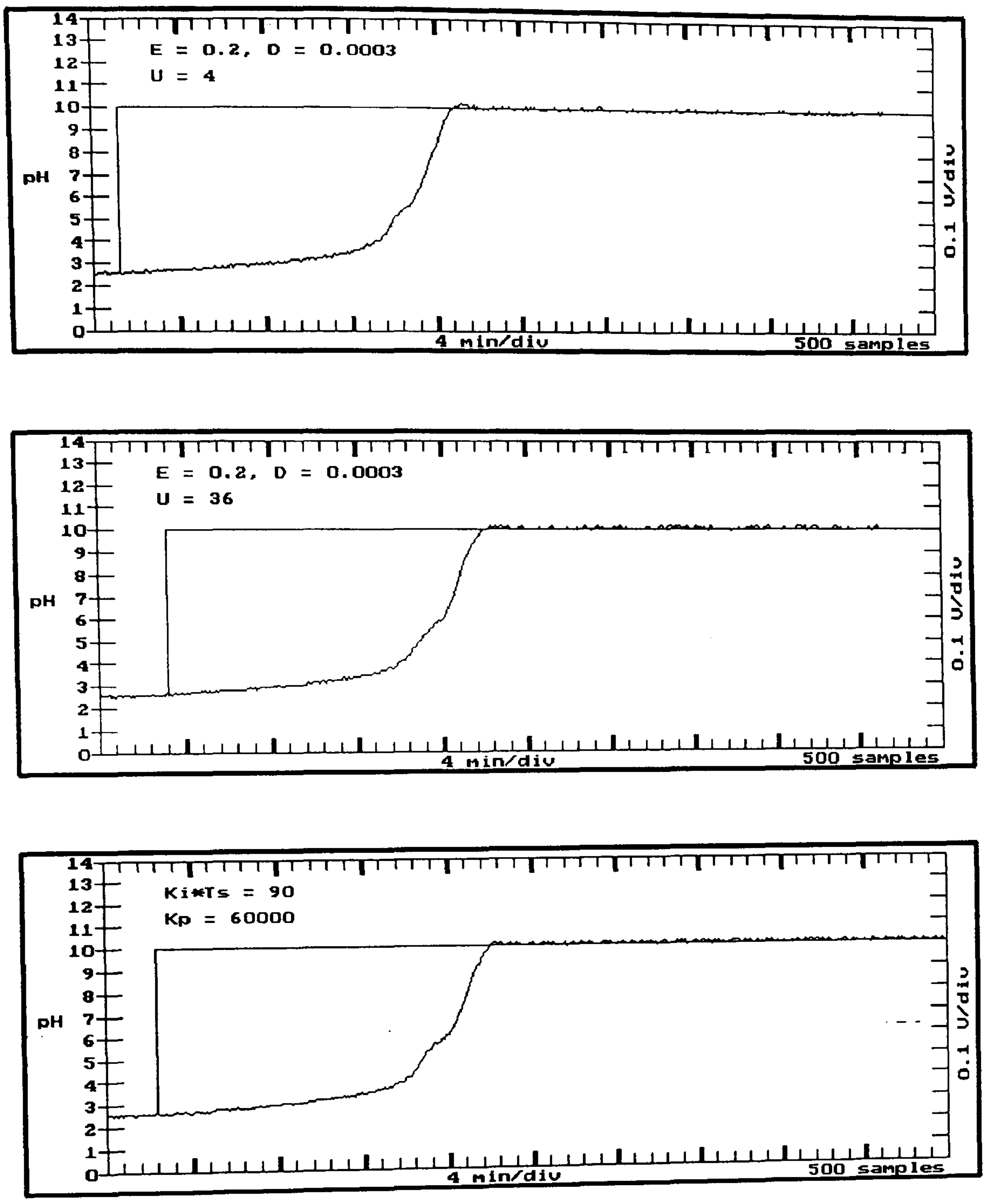
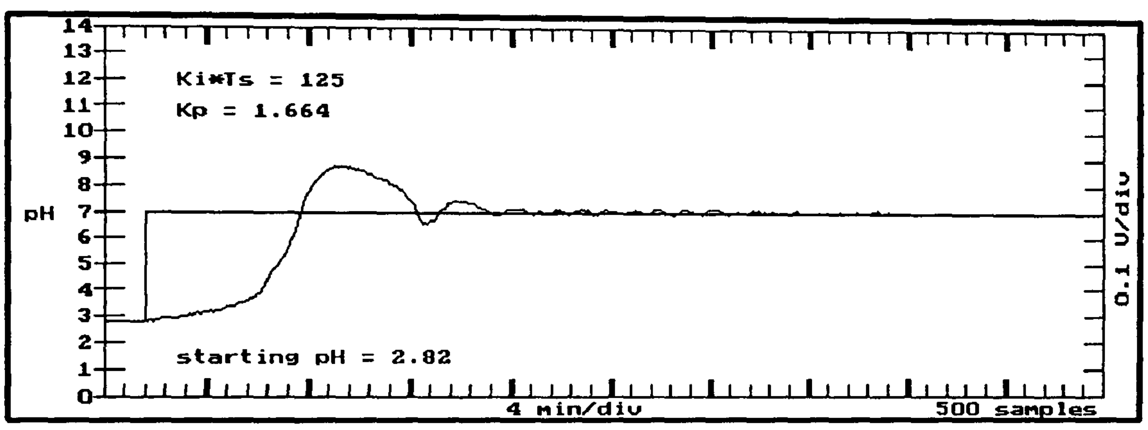


Figure 7-23. (a) and (b) shows the fuzzy pH control response at pH=10,(c) shows the digital pH control response with K_i and K_p corresponding to (b).

7.7 Fuzzy PI control for different load concentrations and with load perturbations

Finally, two experiments were performed to investigate the performance of the fuzzy controller in situations not previously investigated. The first of these looked at the effect of load concentration and the second of the effect of load perturbations. Figure 7-24(a) shows the response of a digital PI controller with $K_i \cdot T_s = 125$, $K_p = 1.664$ when controlling a pH process having a starting pH equal to 2.82. Figure 7-24 (b) shows the behavior of the same system when controlled by a fuzzy controller with E=0.2, D=0.0003 and U=4. Obviously, the latter shows less overshooting than the former.

(a)



(b)

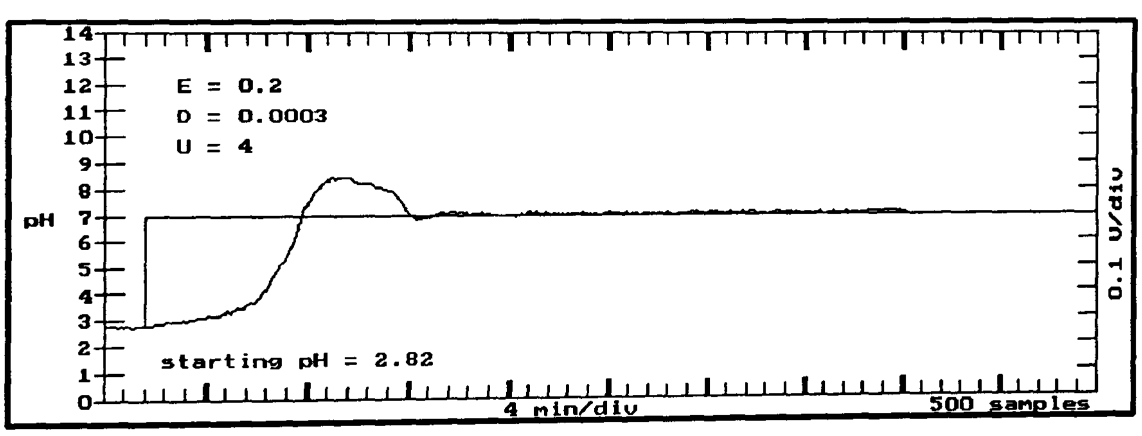
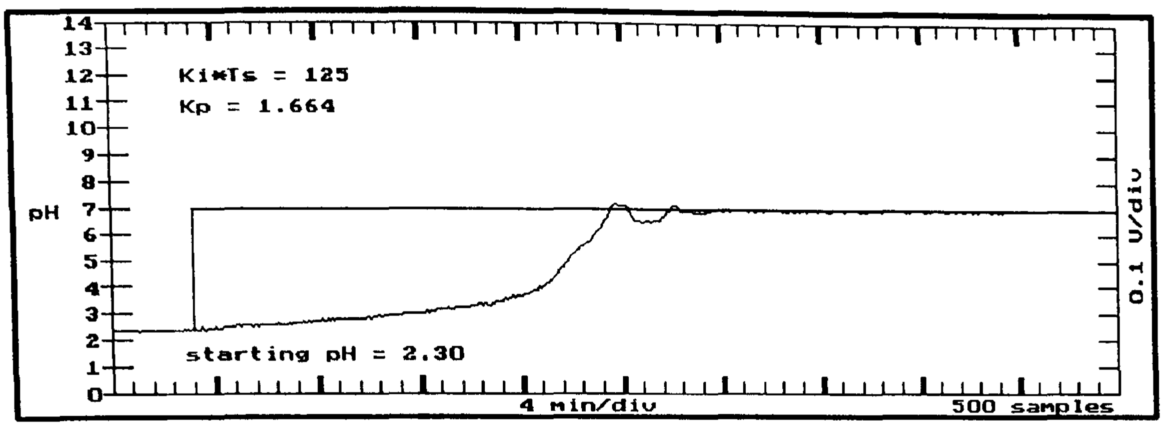


Figure 7-24 A process with a stream concentration pH=2.82controlled by (a) a digital controller, (b) a fuzzy controller.

A second test was performed using a starting pH of 2.30. The results of this test are shown in Figure 7-25. Again the fuzzy controller has a better performance than the digital controller.

Figure 7-26 shows the system response to a load perturbation. This was achieved by suddenly adding some strong base solution into the reaction tank. This qualitative

(a)



(b)

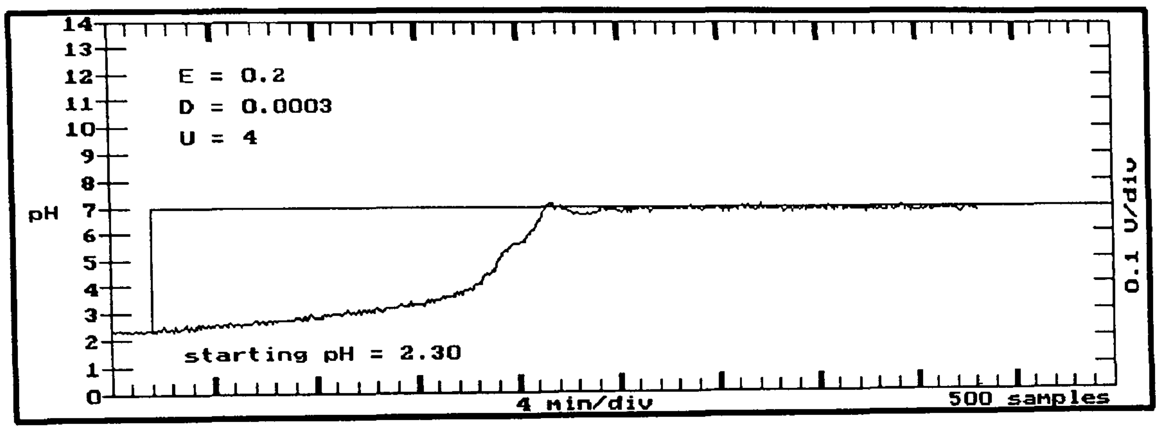


Figure 7-25. A similar experiment to that of Figure 7-28, but with a process stream concentration pH=2.3.

experiment indicate that the fuzzy controller can cope with such perturbations and will bring the pH back to the set-point.

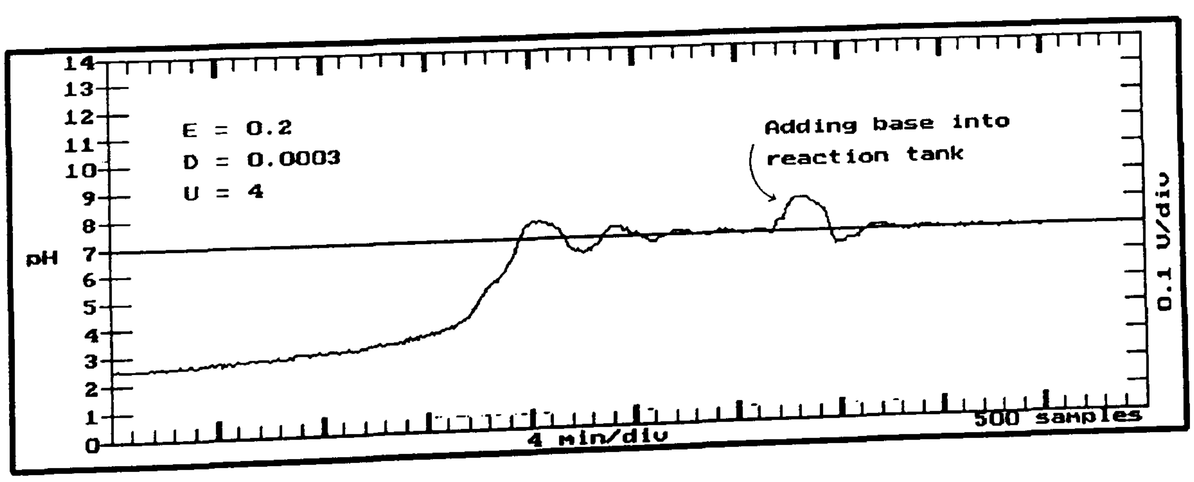


Figure 7-26. The load perturbation test of the fuzzy pH control

7.8 Conclusions from the experiments performed

At the beginning of this chapter, the results of several experiments were presented which demonstrated the superiority of the fuzzy controller described in this thesis over a conventional digital controller. These experiments compared the two methods of control in response to both a single step input change at pH=7 and to a series of different set points.

Two series of experiments were then described which compared the performances of a conventional digital PI controller and the fuzzy controller in controlling the nominal plant at different set-points. The first series of experiments used a large value of K_i and a small value of K_p for both controllers. In the second series, K_i was made small and K_p large. In the first case, it was very difficult to choose the right control constants for the two controllers to make the system settle down quickly at the set-point, especially for pH=7and 8. However, in the second case, when K_p was made very large and the corresponding value of D was very small, the fuzzy controller was found to be significantly better than the digital controller in reducing the settling time. To investigate the reason for this a detailed study of the control output and the control action trajectory around the set-point was made. The results of this study were described in section 7.3.

When using a fuzzy controller, very fast oscillations occur around the set-point. The control action trajectory for these oscillations resembles that of noise chattering around the set-point. Since the output of a fuzzy controller is determined by the position of its control action trajectory on the decision table, this chattering behavior produces very rapid fluctuations of the control output. The system is driven by the average of these fluctuations, and will settle down at the set-point if the average of this output drives the error input to the fuzzy controller to zero.

The interaction between the various parameters of the fuzzy controller were investigated and a simple relationship was found between the values of D and U. It was found that expanding the range D by a factor of n has a similar effect to reducing U by a factor of n^3. In most of the experiments described within this chapter E was set to a fixed value to give an appropriate pH error range. The relationship between D and U then allowed the characteristics of the controller to be adjusted by varying a single parameter (either D or U). The ability to adjust the system's characteristics using a single control parameter should simplify the application of adaptive techniques to such a controller.

The final experiments within this chapter looked at the effects of load variations and perturbations, on the performance of the fuzzy controller. These showed that the fussy controller performed better than a conventional digital controller for various static values of the load concentration. They also showed that the fuzzy controller could deal satisfactorily with perturbations of the load.

Chapter 8

Conclusion

This thesis has investigated the application of fuzzy logic techniques to the control of a continuous flow pH process. In the chemical industry, the control of pH is a well-known problem that presents difficulties due to the large variations in its process dynamics and the static nonlinearity between pH and concentration. pH control requires the application of advanced control techniques such as linear or non-linear adaptive control methods. Unfortunately, adaptive controller rely on a mathematical model of the process being controlled, the parameters being determined or modified in real time. Because of its characteristics, the pH control process is extremely difficult to model accurately. This has motivated the use of fuzzy logic methods in this thesis.

In chapter 2, the basic concepts of fuzzy set theory, fuzzy logic and approximate reasoning have been discussed. Various methods of incorporating fuzzy reasoning into a control system to create a fuzzy logic controller were then considered. Chapter 2 also presented a brief review of previous work on the subject of pH control. This review concluded that, to date, there has been little or no effect addressed at investigating the use of fuzzy logic methods in the control of a continuous flow pH process.

In chapter 3, the practical issues of building a fuzzy logic controller and implementing a fuzzy logic controller in real-time were considered. The chapter then described the development of a C language generator to automate the process of programming a real-time fuzzy logic controller.

Chapter 4 presented a detailed study on the relationship between fuzzy decision table scaling factors and the control constants of a digital PI controller. This study resulted in a novel method of tuning a fuzzy logic controller by adjusting its scaling factors. This tuning method forms the bases of the experiments described in the remainder of the thesis.

Chapter 5 described experiments conducted with the real-time fuzzy logic controller when used to control an analog first order plant. These experiments were carried out in order to understand the effect of the novel tuning method on the operation of the fuzzy logic controller prior to its use to control the continuous flow pH process. It was described that one of the most important characteristics of the tunable fuzzy controller is its ability to implement a wide variety of control mechanisms simply by modifying one or two control variables. Thus the controller could be made to behave in a manner similar to that of a conventional PI controller, or with different parameter values, could imitate other forms of controller. One such mode of operation uses sliding mode control, with the fuzzy decision table main diagonal being used as the variable structure system (VSS) switching line. A theoretical explanation of this behavior, and its boundary conditions, were given in the chapter.

In chapter 6, the continuous flow pH process used for the experiments in the remainder of the thesis was described. The Chapter presented the results of experiments to obtain the pH process titration curve. This curve, which dominate the pH process, is time varying and highly nonlinear.

Chapter 7 described an extensive series of experiments using the continuous flow pH process. The performance of the tunable fuzzy controller was compared with that of a conventional PI controller in response to step change in the set-point, at a number of pH levels. The results showed not only that the fuzzy controller could be easily adjusted to provided a wide range of operating characteristics, but also that the fuzzy controller was much better at controlling the highly non-linear pH process, than a conventional digital PI controller. The fuzzy controller achieved a shorter settling time, produced less over-shoot, and was less affected by contamination than the digital PI controller.

8.1 Assessment of the fuzzy PI controller

The main achievement of this research is that it provides an easy way to deal with a most difficult pH control process. As mentioned in the survey section of chapter 3, sophisticated control techniques are usually required to handle the non-linearity problems encountered in pH control processes. Most of the controllers have to change or adjust their control constants for different pH set-points. However, the simple fuzzy PI controller developed in this work shows very good performance when the set-point is changed to a different value or the load (solution concentration) is altered.

Perhaps the most outstanding feature of this PI controller is that it can be used as a regular PI controller or switched to another type of controller easily by simply changing the decision table ranges. As has been discussed in chapter 5, this fuzzy PI controller can even make the system perform as a sliding mode controller, with the decision table main diagonal line forming the variable structure system (VSS) switching line.

The fuzzy PI controller shows several saturation or damping effects, which result from choosing too large or too small a value for the error and the error change ranges. Sometimes this damping effect may be useful for reducing the overshoot of the system response or confining the amplitude of the control output.

Another aspect of this fuzzy PI controller is that it can be used to provide a VSS by switching the decision table ranges to new settings during the control action. For instance, chapter 5 shows that the fuzzy PI controller can be switched to an integral controller by setting D to an extremely large value or be used to diminish the limit cycle amplitude by continuously shrinking all three decision table ranges together.

8.2 Recommendations for future work

The controller described here is based on Mamdani's minimum operation method and used triangular membership functions to represent all the linguistic terms. But there are several other reasoning methods, such as Larsen's product operation method and the type 4 reasoning method. Also there are several other membership functions, for instance, bell shaped and trapezoid shaped membership functions which have not been tried. Therefore, it may prove useful in further studies to try using these membership functions and reasoning method in control of the pH process.

The work in testing the fuzzy PI controller was based on experiments with a first order analog simulation. Some success has been obtained in understanding the fuzzy PI control response via the corresponding values of conventional PI control constants K_p and K_i. It is worth studying the behavior of this fuzzy controller with plants of higher order and, especially, to see if any sliding regimes exist in the higher order fuzzy systems.

The controller described here was designed for use with a single-input single-output process. This is not a limitation of the fuzzy logic controllers. So, further work should consider the design and implement of multi-variable fuzzy controllers and to test their performance on multi-input and multi-output real processes.

REFERENCE

[1] Shinsky F. G., "A self-adjusting system for effluent pH control," *Spring joint conf. ISA*, St. Louis (1978).

[2] Buchholt F. and M. Kummel, "Self-tuning control of a pH-neutralization process," *Automatica*, vol. 15, pp.665-671, 1979.

[3] Bergmann S. and K. H. Lachmann, "Digital parameter adaptive control of a pH process," *Joint Auto. Control Conf.* San Francisco, Paper TA2-E, 1980.

[4] Gustaffson T. K. "An experimental study of a class of algorithms for adaptive pH control," *Chem. Engng. Sci.*, vol. 40, pp. 827-837, 1985.

[5] T. K. Gustaffson and K. V. Waller, "Dynamic modeling and reaction invariant control of pH," *Chem. Engng. Sci.* vol. 38, no.3, pp. 389-398, 1983.

[6] Goodwin G. C. and K. S. Sin, *Adaptive Filtering Prediction and Control*, pp.220-223, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

[7] Wright R. and C. Kravaris, "Nonlinear state feedback synthesisfor pH control," *World Congr. III of Chem. Engng.*, Tokyo, pp. 731-734, 1986.

[8] T. J. McAvoy, E. Hsu and S. Lowenthal, "Dynamics of pH in controlled stirred tank reactor," *Ind. Eng. Chem. Process Des. Develop.*, vol. 11, no. 1, 1972.

[9] J.O.Flowerand N Vijeh, "Design, Performance and Assessment of a Self-organising Fuzzy-logic controller of a Highly Non-linear Fluid Supply System", read at a meeting of The Institute of Marine Engineerings, London, March 1997.

[10] K. J. Åström, B. Wittenmark, ,*Adaptive control.* New York: Addison-Wesley, 1989.

[11] L. A. Zadeh, "Fuzzy algorithm," *Informat. Control.* vol. 12, pp. 94-102, 1968.

[12] L. A. Zadeh, "Outline of a new approach to the analysis complex systems and

[13] E. H. Mamdani, "Applications of fuzzy algorithms for simple dynamic plant," *Proc. IEE,* vol. 121, no. 12, pp. 1585-1588, 1974.

[14] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man Mach. Studies.* vol. 7, no. 1, pp. 1-13, 1975.

[15] E. H. Mamdani, "Advances in the linguistic synthesis of fuzzy controllers," *Int. J. Man Mach. Studies*, vol. 8, no. 6, pp. 669-678, 1976.

[16] O. Yagishita, O. Ito and M. Sugeno, "Application of fuzzy reasoning to the water purification process," in *Industrial applications of fuzzy control,* M. Sugeno Ed. Amsterdam: North-Holland, 1985, pp. 19-40.

[17] S. Yasunobu, S. Miyamoto, "Automatic train operation by predictive fuzzy control," in *Industrial applications of fuzzy control,*M. Sugeno Ed. Amsterdam: North-Holland, 1985, pp.1-18.

[18] F. Fujitec, "FLEX-8800 series elevator group control system," Fujitec Co., Ltd., Osaka, Japan, 1988.

[19] J. A. Bernard, "Use of rule-based system for process control," *IEEE Contr. Syst. Mag.*, vol.8, no.5, pp.3-13, 1988.

[20] S. Chiu, S. Chand, D. Moore, and A. Chaudhary, "Fuzzy logic for control of roll and moment for a flexible wing aircraft," *IEEE Control Syst. Mag.*, vol. 11, no. 4, pp. 42-48, 1991.

[21] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller-Part I & II," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-20, no. 2, pp. 404-435, 1990.

[22] M. Braae and D. A. Rutherford, "Fuzzy relations in a control setting," *Kybernete,* vol. 7, no. 3, pp. 185-188, 1978.

[23] T. Tagaki and M. Sugeno, "Derivation of fuzzy control rules from human operator's control actions," in *Proc. of the IFAC Symp. on Fuzzy Information,*

Knowledge Representation and Decision Analysis, Marseilles, France, July 1983, pp 55-60.

[24] C. P. Pappis and E. H. Mamdani,"A fuzzy logic controller for a traffic junction," *IEEE Trans. Syst. Man Cybern.*, vol. SMC7, no. 10, pp. 707-717, 1977.

[25] P. M. Larsen, "Industrial applications of fuzzy logic control." *Int. J. Man Mach. studies*. vol. 12, no. 1, pp. 3-10, 1980.

[26] M. Sugeno and M. Nishida,"Fuzzycontrol of model car," *Fuzzy Sets syst.,* vol. 16, pp. 103-113, 1985.

[27] M. sugeno and K. Murakami, "An experimental study on fuzzy parking control using a model car," in *Industrial Applications of Fuzzy Control,* M. Sugeno, Ed. Amsterdam: North-Holland, 1985, pp. 125-138.

[28] M. Sugeno and K. Murakami,"Fuzzyparking control of model car," in *23rd IEEE Conf. on Decision and Control.* Las Vegas, 1984.

[29] M. Sugeno and G. T. Kang, "Structure identification of fuzzy model," *Fuzzy Sets and Systems,* vol. 28, pp. 15-33, 1988.

[30] M. Sugeno and K. Tanaka,"Successive identification of a fuzzy model and its applications to prediction of a complex system," *Fuzzy Sets and Systems,* vol. 42, pp. 315-334, 1991.

[31] T. J. Procyk and E. H. Mamdani,"A linguistic self-organizing process controller," Automat., vol. 15, no. 1, pp. 15-30, 1979.

[32] C. C. Lee,"A Self-learning Rule-Based Controller Emplying Approximation Reasoning and Neural Net Concepts," *Int. J of Intel, Syst.*, vol., 6, pp. 71-93, 1991.

[33] P. J. King and E. H. Mamdani,"The application of fuzzy control systems to industrial process," in *IFAC World Congress*, MIT, Boston, 1975.

[34] M. Braae and D. A. Rutherford, "Sectionof parameters for a fuzzy logic controller," *Fuzzy sets Syst.*, vol. 2, no. 3, pp. 185-199, 1979.

[35] S. Yasunobu, S. Miyamoto, and H. Ihara, "Fuzzy control for automatic train operation system,", in *Proc. 4th IFAC/IFIP/IFORS Int. Congress on Control in Transportation Systems.* BadenBaden, April 1983.

[36] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision process." *IEEE Trans. Syst. Man Cybern.*, vol. SMC-3, no.1, pp. 28-44, Jan 1973.

[37] A. Kaufmann, *Introduction to Theory of Fuzzy Subsets*, New York: Academic 1975.

[38] M. Mizumoto, "Note on the arithmetic rule by Zadeh for fuzzy conditional inference," *Cybern. Syst.*, vol. 12, pp 247-306, 1981.

[39] Y. Tsukamoto, "An approach to fuzzy reasoning method." in *Advances in Fuzzy Set Theory and Applications,* M. M. Gupta, R. K. Ragade, and R. R. Yager, Eds. Amsterdam: North-Holland 71X1979.

[40] M. Sugeno and T. Takagi, "Multi-dimensional fuzzy reasoning," *Fuzzy Sets and Systems*, vol. 9, no. 2, 1983.

[41] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-15, no. 1, pp. 116- 132, 1985.

[42] B. Jayadeva, Y.S.N.M.Rao, M. Chidambaram and K.P. Madhavan," Nonlinear controller for a pH process," *Computers chem. Engng*, vol.14, no. 8, pp. 917-920, 1990.

[43] R. A. Wright and C. Kravaris, "Nonlinear control of pH process using the strong acid equivalent," *Ind. Eng. Chem. Res.* vol 30, pp. 1561-1572, 1991.

[44] C. L. Karr and E. J. Gentry,"Fuzzy Control of pH Using Genetic Algorithms," *IEEE trans. on Fuzzy Syst.*, vol. 1.no. 1 pp. 46-53, 1993.

[45] Masaki Togai,Horoyuki Watanabe."Expert System on a Chip: An Engine for approximate reasoning." *IEEE Expert,* vol 1, no.3,pp. 55-62, 1986.

[46] E. H. Mamdani, S. Assilian."An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller." *Fuzzy Reasoning and its Application.* London, 1981 NewYork.

[47] H. Arikawa, K. Hirota, and M. Sakawa."Virtual paging Fuzzy Chip & Fuzzy Workstation as its Design Evironment." *3rd IFSA World Congress,* Seattle, 1989.

[48] T. Y. Sun and Y. K. Yang,"A Real-Time Interpreted Fuzzy Inference Engine." *2nd Symposium on Fuzzy System,* 1994, Taipei, R.O.C.

[49] NeuraLogix Inc."*FMC NLX23x Fuzzy MicroController Family.*" 1991.

[50] J. Teichrow, E. Horstkotte, and Masaki Togaai."The Fuzzy-C Compiler: A Software Tool for Producing Portable Fuzzy Expert Systems." *Proc. Int. Workshop on Fuzzy Syst. Appl.,* 1989, Iizuka Japan.

[51] N. Mitzuya and K. Tanaka,"*Practical fuzzy book-by C language,*"Ra Se Ru co., Tokyo,1989. (in Japanese)

[52] R. M. Tong,"A control engineering review of fuzzy systems." *Automatica,* vol.13, pp. 559-569, Nov. 1977.

[53] M. Sugeno, "An introductory survey of fuzzy control," *Inform. Sci.*, vol. 36, pp. 59-83, Aug. 1985.

[54] R. J. Mulholland,"Designingexpert controllers,"presented at the *IEEE Region Five Conf.*, Tulsa, OK, Mar. 1987.

[55] C. M. Liaw and J. B. Wang,"Design and implementation of a fuzzy controller for a high performance induction motor drive,"*IEEE Trans. on Syst. Man. and Cybern.*, vol. 21, no. 4. pp. 921-929. 1991.

[56] Jihong Lee,"On Methods for Improving Performance of PI-Type Fuzzy Logic Controllers," *IEEE. Trans. on Fuzzy Syst.*, vol. 1,no 4, pp 298-301,1993.

[57] K. L. Tang and R. J. Mulholland,"Comparing fuzzy logic with classical controller designs," *IEEE. Trans. on Syst. Man. and Cybern.*, vol. SMC-17,no. 6, pp 1085-1087, 1987.

[58] P. J. MacVicar-Whelan,"Fussy sets for man-manchine interaction," *Int. J. Man-Machine Studies*, vol. 8,pp. 687-697, Nov. 1976.

[59] U. Itkis, *Control Systems of Variable Structure*, John Wiley & Sons, Inc. 1976.

[60] V. I. Utkin, "Variable structure systems with sliding modes," *IEEE. Trans. on Automatic Control*, vol. AC-22, no 2, pp 212-222, April 1977.

[61] DeCarlo, R. A., Zak, S. H., and Mattews, G. P.," Variable structure control of nonlinear and multivariable system: a tutorial," *Proc. IEEE,* vol. 76, pp 212-232, 1988.

[62] C. M. Dorling, A. S. I. Zinober, "Robust hyperplane design in multivariable structure control systems'" *Int. J. Control,* vol. 48, no. 5, pp 2043-2054, 1988.

[63] S. K. Spurgeon, "Hyperplane design techniques for discrete-time variable structure control systems," *Int. J. Control,* vol. 55, no. 2, pp 445-456, 1992.

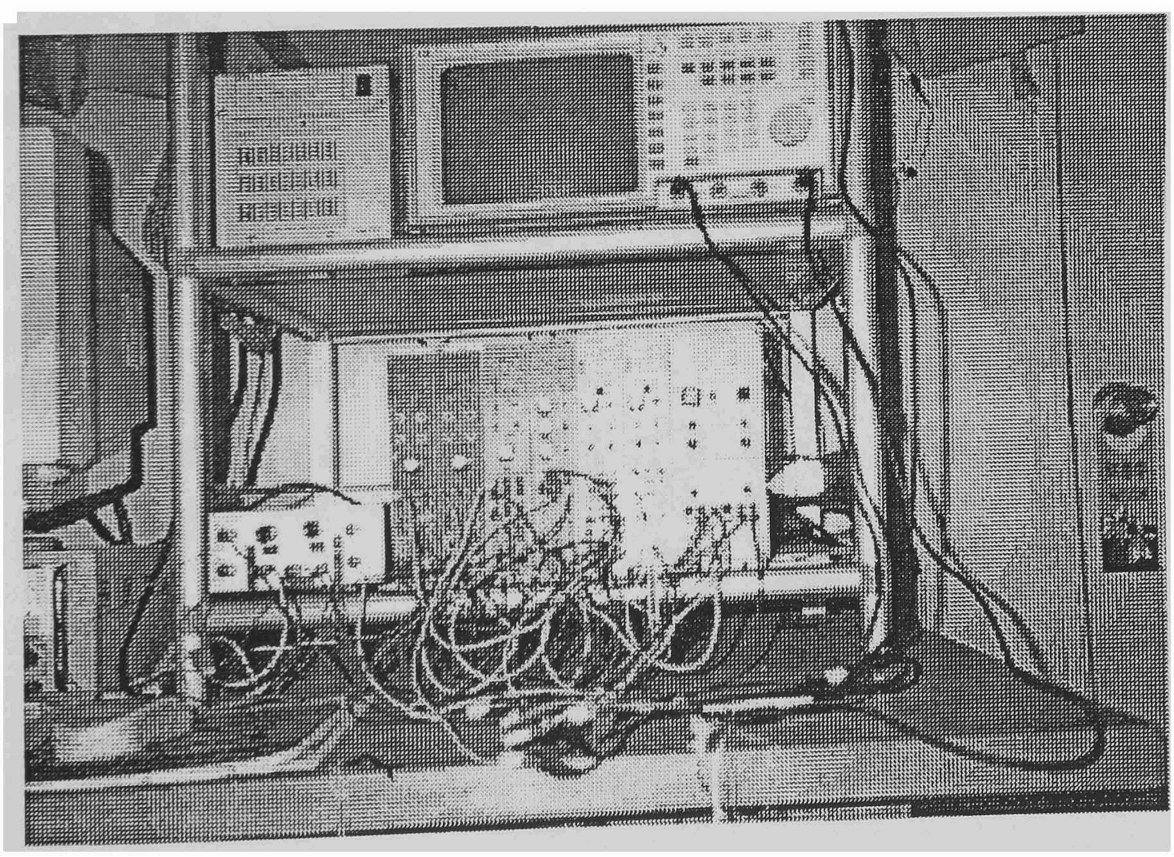
[64] Walker W. A. and Y. Liu, "Design Criteria for pH control systems," *World Congr. III of Chem. Engng.*, Tokyo, pp. 735-738, 1986.

[65] P. M. Larsen," Industrial applications of fuzzy logic control," *Int. J. Man Mach. Studies,* vol. 12, no. 1, pp. 3-10, 1980.

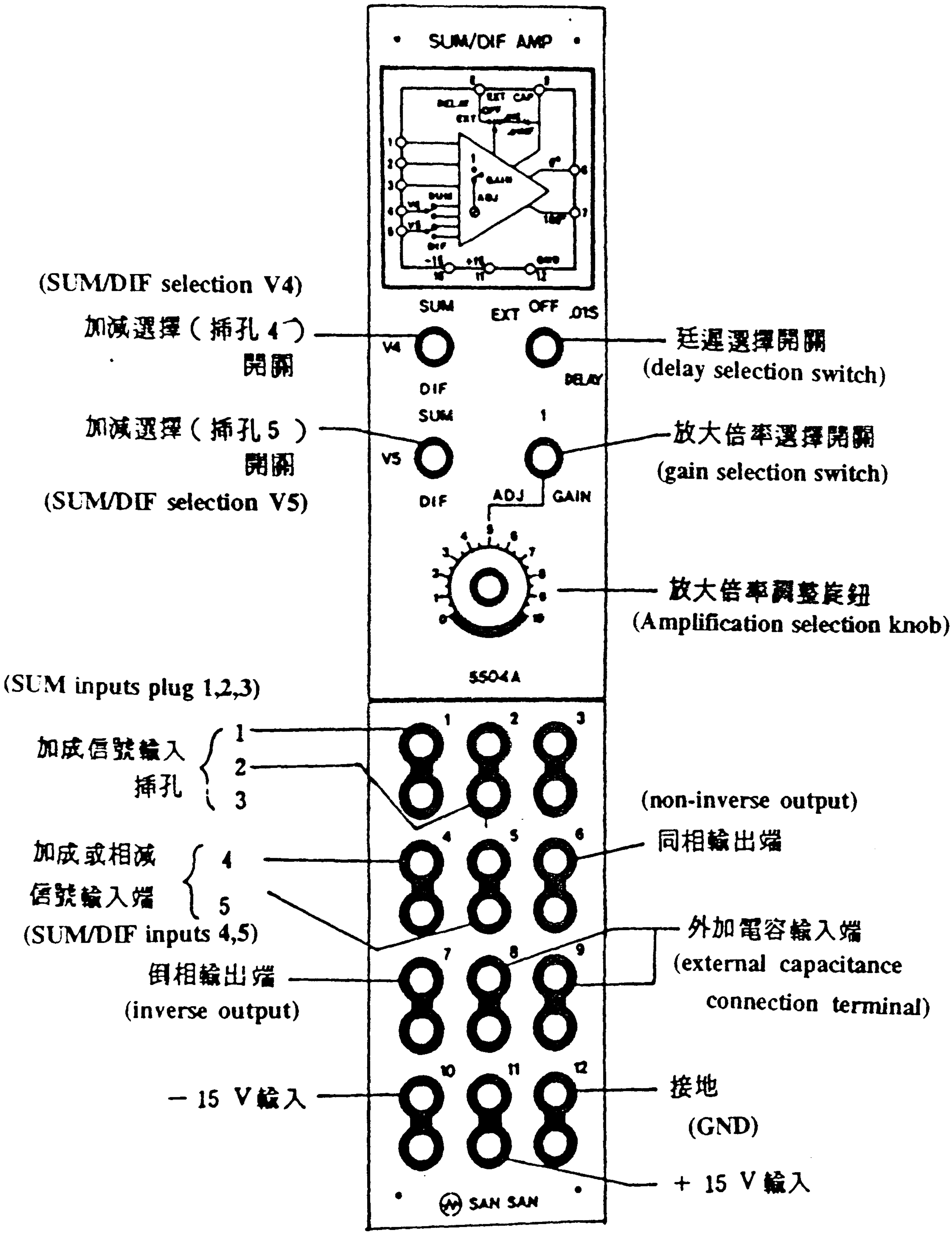
Appendix A

The Analog Modules Used for Simulation

This analog module set is a product of Sansonic Electronics Co. Ltd, Taiwan. It consists of several modules such as Lead/Lag compensator, Phase Mod/Demod, Servo Amp, Voltage Amp, Phase controller, PID controller, Sum/Dif Amp, Transducer Amp, Backlash, Saturation/Dead zone, Atenuator, Integrator, Synchro Transmitter, Synchro Receiver, SCR/TRIAC etc., which is specially designed for the school control experiment laboratory. In this thesis, only Integrator and Sum/Dif Amplifier modules were applied for constructing a first order plant as shown in the picture below. Their structures are shown in the next few pages.

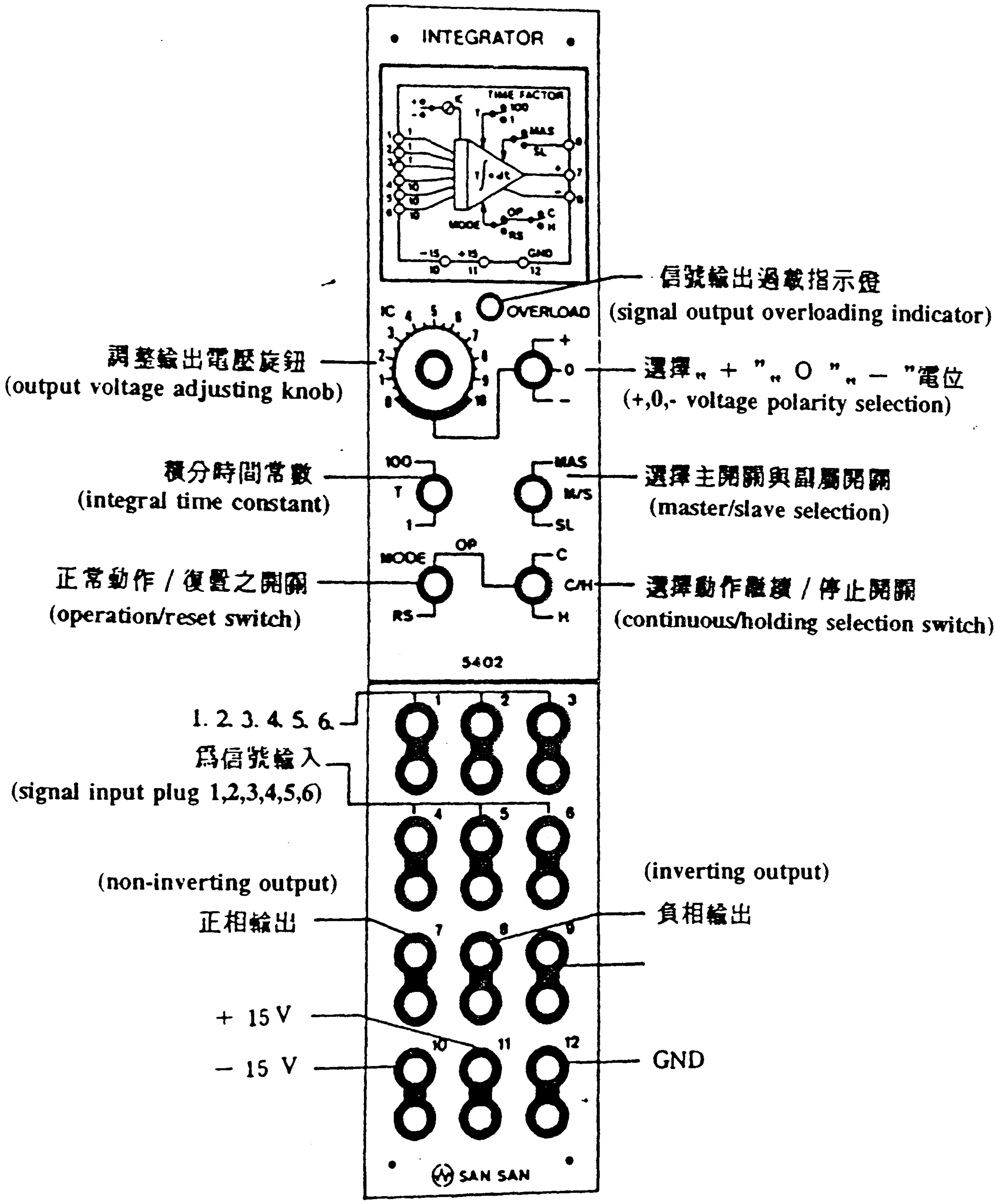


5504A SUM/DIF AMP



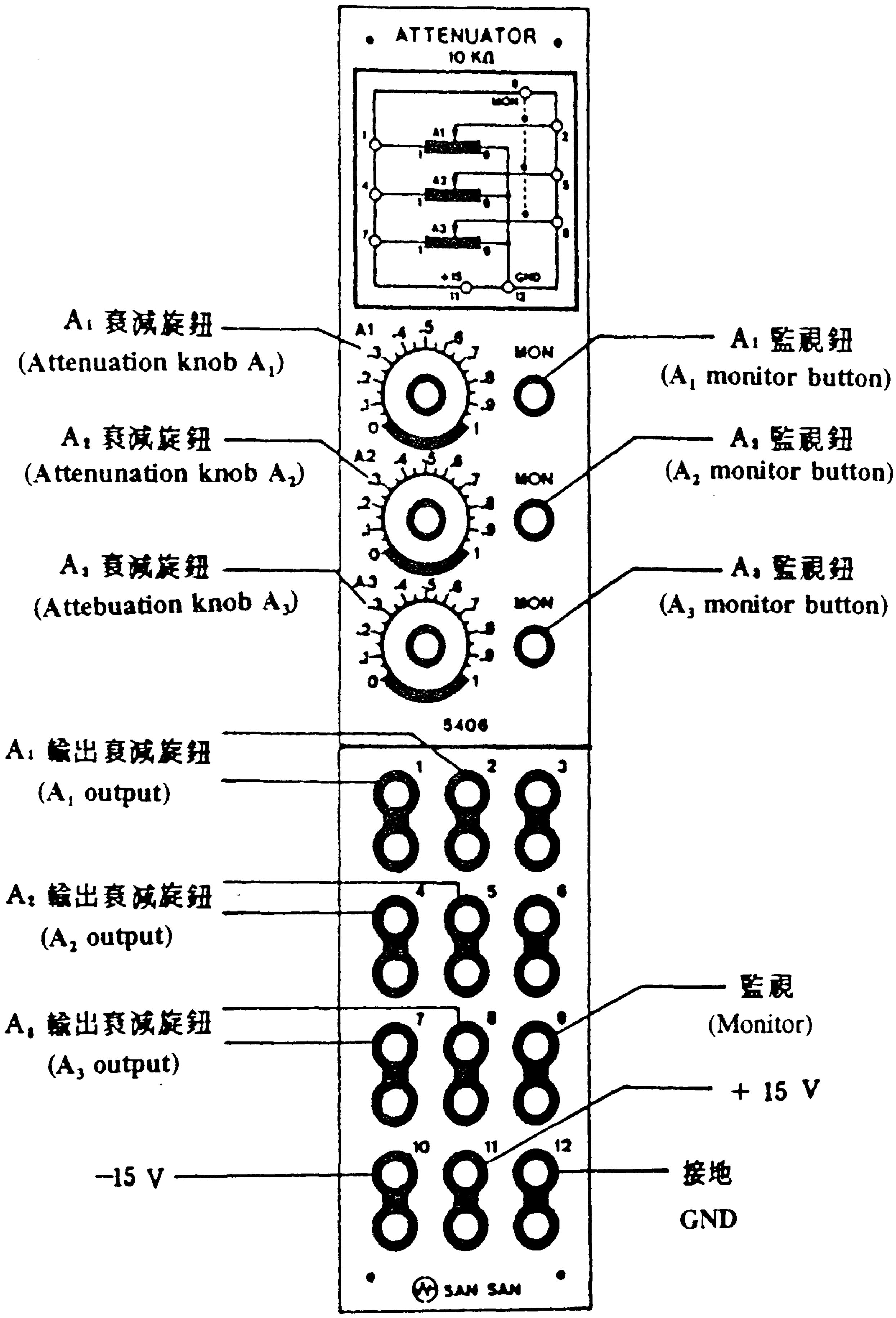
SANSONIC ELECTRONICS CO., LTD.

5402 INTEGRATOR (2EA)



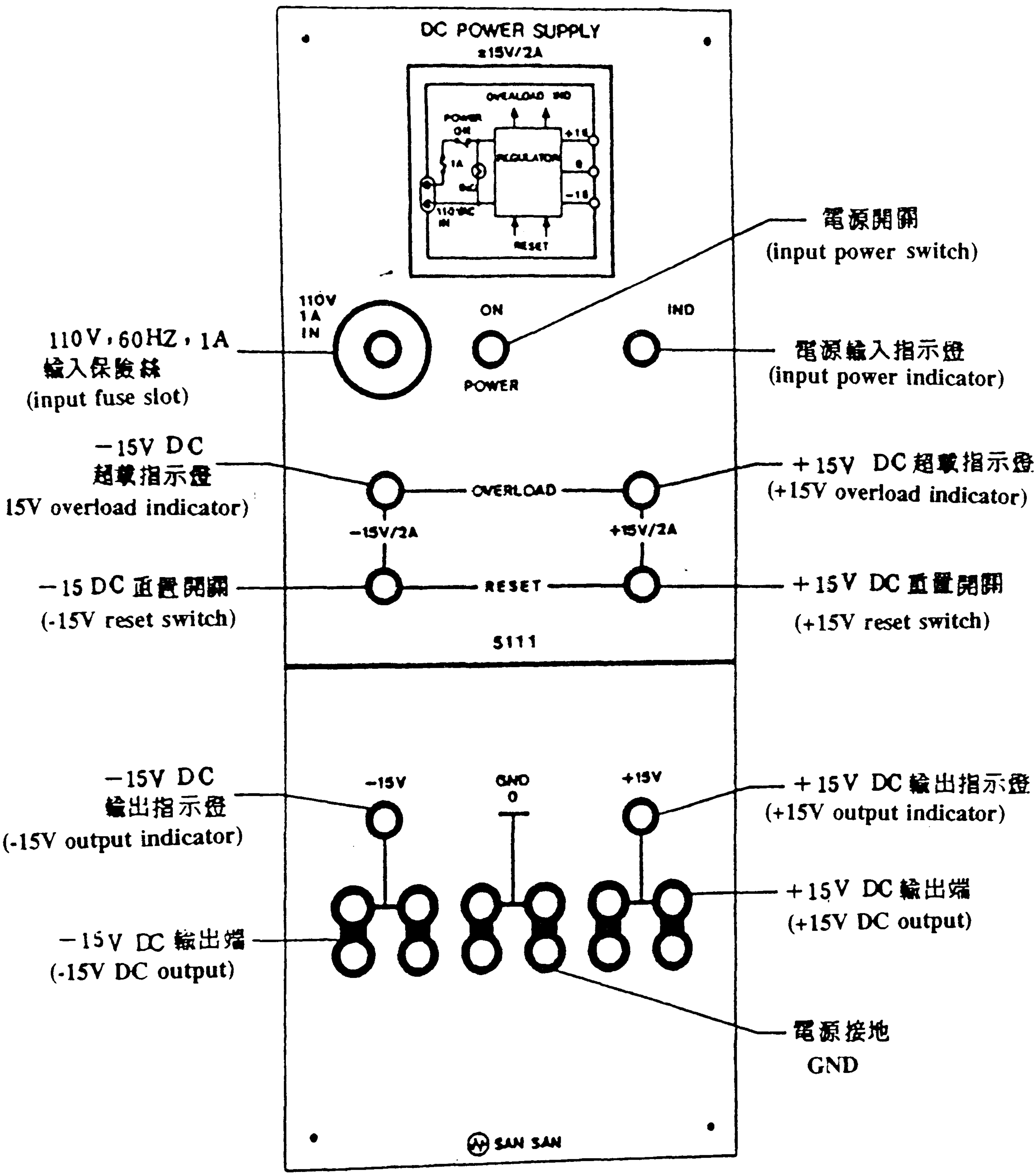
SANSONIC ELECTRONICS CO., LTD.

5406 ATTENUATOR (10KΩ)



SANSONIC ELECTRONICS CO., LTD.

5111 DC POWER SUPPLY (±15V/2A)



SANSONIC ELECTRONICS CO., LTD.

Appendix B

The Model Plant Components Specification

Specifications of the model plant major component

(1) The metering pump P-1 for the acid solution:

This pump containing a variable, interchangeable control unit which pace the pump continuously or directly by means of external pulse signals.

control type - 1. power switch +fuse.

2. two-stage float switch connector.

3. metering monitor input.

4. equipped with an analog input which can be used to control the capacity of the pump proportionally to a 0/4-20 mA input signal.

control version - 4000 pulses/h, 0/4-20mA.

Capacity at max. back pressure - 123 psi, at least 47.3 *l*/h.

Stroking rate at max. back pressure - at least 146 strokes/min.

manufacture and its type - ProMinent Vario, Germany

(2) The metering pump P-2 for the alkaline solution:

This is a microprocessor-based solenoid driven diaphragm-type metering pump for chemical feeding. The pump capacity can be varied by varying the stroke length between 100 and 10% and by the variation of the stroking rate between 120 to 1 strokes per minute in the 1:1200 range at least. This pump can monitor its own output. If certain amount of pump strokes are missing, the pump will stop and showing some error signal.

The control versions -

Analog control: The stroking rate can be varied between 0 ~ 100% according to the $0 \sim 10$Vinput analog signal.

Pulse control: This is used to tune the pump to generators of any kind. The pulse step-down and step-up ratio can be set via keyboard. The number of strokes predetermined once can be call up by some contact or some special key.

Capacity at average back pressure - 6 bar : 1.8 *l*/h, 0.25m*l*/stroke, 120 strokes/min.

manufacture and its type: ProMinent gamma G/4a, Germany

(3) The pH/mV/Temp. Meter:

range: pH: $0 \sim 14.00$pH, mV: -1999mV $\sim +1999$mV, Temp.: $0 \sim 100$°C.

resolution: pH: 0.01pH, mV: 1mV, Temp.: 0.1°C

accuracy: pH: 0.01pH$\pm$1digit, mV: 0.1%$\pm$1digit, Temp.: 0.4°C$\pm$1 digit.

Temperature compensation: both Manual and Auto are $0 \sim 100$°C.

Recorder output: pH $0 \sim 1400$mV

mV -2000mV $\sim +2000$mV

Temp. $0 \sim 1000$mV

Manufacture and its type: Suntex, SP-701, Taiwan

(4) 486 based pH recorder: Usually, the pH process reaction time takes more than 30 minutes. The storage scope used in chapter 5 can only record about 2 minutes, therefore, it is not suitable for recording. A 486 based microcomputer storage scope program is written for this purpose, as shown in figure 6-10. The voltage of its A/D card (PCL818H) is carefully calibrated and shown on the screen and timing is accuratly controlled through the interrupt procedure by the internal 10 MHz clock.