1994

Advanced fuzzy logic controllers and self-tuning strategy

Shou-Heng Huang
Iowa State University

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 9503566

Advanced fuzzy logic controllers and self-tuning strategy

Huang, Shou-Heng, Ph.D.

Iowa State University, 1994

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

Advanced fuzzy logic controllers and self-tuning strategy

by

Shou-Heng Huang

A Dissertation Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Department: Mechanical Engineering
Major: Mechanical Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa
1994

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

The basic concepts of fuzzy logic and fuzzy set theory were introduced by Lofti Zadeh in his excellent papers [Zadeh 1965 and 1968] in the mid-'60s. The pioneering research work on fuzzy logic controllers (FLCs) was done by E.H. Mamdani and his colleagues [Mamdani 1976 and Kickert 1978] in the mid-'70s. In recent years, the literature on fuzzy control has been growing rapidly and a wide variety of applications have been reported. Fuzzy control is an active and fruitful area for research in the application of fuzzy set theory.

All these applications, such as controls for automatic train operation system [Yasunobu 1985], robot arm [Mandic 1984], diesel engine [Murayama 1985], combustion [Ono 1989], and heat pumps [Meijer 1992] have indicated that fuzzy controls appear useful when the processes are too complex for analysis using conventional control algorithms or when the available information is qualitative, inexact, or uncertain.

Fuzzy Set Theory

In the real world, things do not always fall into the neat, crisp categories defined by traditional set theory, like the set of even numbers or human gender. In traditional set theory, membership in a class or set is not a matter of degree. Either a number is even, or it is not. But this on-or-off, yes-or-no, 0-or-1 approach is not suitable when

applied to many everyday classifications, like the set of old man or the set of cold day. To deal with such cases, Zadeh proposed that membership in a set be measured not as a 0 or 1, but a value between 0 and 1. Thus, in the set of old man, for example, a man of 72 year old might have a membership value of 0.8.

In fuzzy set theory, the concept "temperature" is seen as a "linguistic variable". Its "value" could be "very high", "high", "medium", "low", and "very low", which are referred as "fuzzy subsets". All these fuzzy subsets compose the fuzzy set from which the linguistic variable, temperature, takes values. Fuzzy mathematics deals with fuzzy subsets, or fuzzy values, which makes it possible to simulate human thinking and reasoning. A natural language statement, "if room temperature is high then close the steam valve in the air-conditioning system a little bit", could be expressed and calculated by using fuzzy mathematics.

Analytical Control Theory and Non-Analytical Control Technique

Many dynamical systems can be analytically modeled using linear differential equations and this forms the foundation of the classical control theory. With the addition of Laplace transformation and Z transformation methods [Ogata 1970], classical control theory provided a powerful means of analyzing and designing control systems. The mathematical model for the whole control system including controlled plant and controller is set up and the controller's parameters are calculated to compensate the controlled plant characteristic to get the desired behavior.

There are some real systems whose mathematical models cannot be derived from basic physical principles and described as differential equations. Even if the

exact models could be derived, they are too complex for analysis and calculation. Sometimes systems have highly nonlinear operating characteristics and time-varying transport delay, such as most thermal systems. When faced with these difficulties, control engineers use a trial and error method to adjust the controller's parameters, giving up the analytical procedure. In most real applications, this is the way to "design" the commonly-used PID controllers. Here PID indicates that the control output is proportional to the weighted combination of tracking error, integrative error, and derivative error. In most cases, this does not produce satisfactory results. Control engineers found that the control systems governed by humans who interact with the control systems to make them work often have better performance. Experienced operators can often deal with complex control problems which are difficult to manage with classical control theory. Then a reasonable question is how to create a controller which works based on humans' experience.

Fuzzy set theory provides the way to incorporate the experience of human operators into the design of controllers. From a set of linguistic rules which describe the operator's control strategy, a control algorithm is constructed where the words are defined as fuzzy sets. This new approach makes implementing "rule of thumb" experience possible. Mathematical models of plants are not needed. This is first successful application of an Artificial Intelligence (AI) technique to control engineering.

The problem of control is one of decision-making. Given the operating conditions and observation of a controlled process, it is necessary to decide what control actions to take. Knowledge-based systems, in particular rule-based systems, are ideally suited for such a decision-making task. The knowledge used to set up control rules for

a fuzzy controller is derived from expert operators and the designer of the fuzzy logic system. Some of the knowledge can be based on the understanding of the behavior of the process dynamics which cannot be expressed mathematically. The rules can be used in a rule-based fuzzy logic controller (FLC) but not in conventional analytic controllers. Thus, rule-based fuzzy systems can produce better controllers than analytic control theory.

FLCs are a kind of non-linear controller. They are more suitable for processes whose dynamics present strong non-linear characteristics than conventional linear controllers, such as PID controllers. Many studies show that the fuzzy controllers perform superior to conventional control algorithm [Daley 1985, Gupta 1980, Leigh 1983, Ollero 1989, Ralston 1985, and Togai 1991].

At present, there is no systematic procedure for the design of FLCs. Compared with classic control theory, fuzzy control theory is still being developed. Many researcher have attempted to find a systematic method for analyzing fuzzy system dynamics. Cumani considered system quantities such as states, controllability, and observability [Cumani 1982], Tong used the approach of defining system operators as fuzzy relations on the state space [Tong 1980], and Kiszka et al. formulated an energy function of a fuzzy set to investigate stability of a fuzzy system [Kiszka 1985]. However, these researchers failed to present significant descriptions of the dynamics of fuzzy systems. Fuzzy control still lacks a systematic method for analysis and design. The general method used to design fuzzy controllers is a trial and observation approach, which heavily depends on the knowledge of operators and designers of fuzzy systems. This limits widespread use of fuzzy controllers.

About This Work

FLCs sometimes fail to obtain satisfactory results with the initial rule set drawn from operators' and control engineers' experiences. This is because there still are some differences between the way a plant is operated by an experienced operator and by a FLC using the rules based directly on his experience. It is often difficult to express human experiences exactly using linguistic rules in a simple form. Sometimes there is no previous experience to be used to construct control rules for FLCs. In these cases, it is necessary to develop and modify the control rules for FLCs to obtain optimal performance.

In previous articles, there have been few discussions about rule development and adjustment strategies [Scharf 1985; Ollero and Williams 1989; Sheridah 1984; and Wakileh and Gill 1988]. This work presents a method to modify control rules and choose membership functions, scale factors and output gain for FLCs. The dynamics of FLCs are analyzed on a linguistic plane by using their performance trajectories. This approach still uses trial and observation and depends on the intuition and heuristics of designers. It provides a systematic method to design FLCs. Computer simulation and experimental identification indicated that FLCs can work much better than conventional controllers, such as PID controllers. Before dealing with rule development and adjustment strategies, the basic concepts of fuzzy set theory and fuzzy mathematics used for analyzing and designing FLCs are introduced, and the multilevel relay property, the intrinsic feature of FLCs, is illustrated.

In the adjustment procedure for FLCs, people found that it is difficult to obtain good fuzzy rules, especially when certain complicated dynamic processes are con-

cerned. It is time-consuming and needs deep knowledge of fuzzy logic. To avoid this difficulty, a self-tuning strategy has been developed. A novel Self-Tuning Fuzzy Logic Controller (STFLC) is proposed in this study which uses a model-based self-tuning strategy. A predetermined optimal performance trajectory serves as a control model. An initial rule set derived from a fuzzy model of the controlled plant is modified on a look-up-table basis. This self-tuning strategy has been experimentally verified and is effective, but there are some restrictions.

CHAPTER 2. FUZZY LOGIC AND FUZZY LOGIC CONTROL

A fuzzy control algorithm, being compatible with human thought, has two main aspects: rule base and fuzzy reasoning. The rule base is a set of linguistic rules which qualitatively describe the control strategies to be performed on the system. These rules are gleaned from an experienced process operator or designed by a fuzzy control engineer. Because humans think in imprecise terms, the rules are expressed in qualitative (fuzzy) terms. Numerical meaning is given to the qualitative terms of the rule via fuzzy sets. The rules and fuzzy sets are operated through fuzzy reasoning to infer control actions or decisions.

The Basic Concepts of Fuzzy Set Theory

Definition 1: Fuzzy subset: A fuzzy subset F in a universe of discourse U is a collection of elements:

$$u_1,\ u_2,\ .....,\ u_i,\ .....,\ u_l \qquad\qquad u \in U,$$

and is characterized by a membership function μ_F which takes values in the interval [0,1]:

$$\mu_F(u_1),\ \mu_F(u_2),\ .....,\ \mu_F(u_i),\ .....,\ \mu_F(u_l), \qquad\qquad \mu \in [0,1].$$

Thus, a fuzzy subset F may be represented as a set of ordered pairs of a generic element u and its grade of membership function:

$$F = (u_i, \mu_F(u_i))|u_i \in U \tag{1}$$

Definition 2: Fuzzy matrix: A fuzzy matrix M is defined in a universe of discourse V with two dimensions. It is a collection of two-dimensional elements characterized by a membership function μ_M which takes values in the interval [0,1]:

$$\begin{pmatrix} \mu_M(1,1) & \dots & \dots & \dots & \mu_M(1,m) \\ \dots & \dots & \mu_M(i,j) & \dots & \dots \\ \mu_M(l,1) & \dots & \dots & \dots & \mu_M(l,m) \end{pmatrix}$$

A fuzzy matrix M may be represented as follows:

$$M = (v_{i,j}, \mu_M(v_{i,j}))|v_{i,j} \in V \tag{2}$$

Fuzzy subsets and fuzzy matrices are referred to as fuzzy values or fuzzy numbers. A fuzzy subset is a one-dimensional fuzzy number and a fuzzy matrix is a two-dimensional fuzzy number.

Definition 3: Membership function: A relation giving grades of membership for each element of a fuzzy number is known as a membership function. "Fuzzy" refers to the fact that the fuzzy number does not have a sharp boundary between non-membership and membership. It takes values from 0.0 (no membership) to 1.0 (full membership).

Definition 4: Linguistic variables: A linguistic variable is defined by a quadruple (X, U, T(X), R) in which X is the name of variable; U is the universe of discourse of

Table 2.1: The Rule for Associating T(X) with U through Membership Functions

R	U	Range	$0°C$												$40°C$
		$°C$	2	5	8	11	14	17	20	23	26	29	32	35	38
		degrees	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
T(X)	Very high		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.7	1.0
	High		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.7	1.0	0.7	0.2	0.0
	Medium		0.0	0.0	0.0	0.0	0.2	0.7	1.0	0.7	0.2	0.0	0.0	0.0	0.0
	Low		0.0	0.2	0.7	1.0	0.7	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Very low		1.0	0.7	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

X; and T(X) is the term set of X, that is, the subsets of X with each value being a fuzzy number defined on U. For example, if *Temperature* is interpreted as a linguistic variable, then its term set T (temperature) could be:

T(temperature)={very high, high, medium, low, very low}

where each term in T(temperature) is characterized by a fuzzy subset in a universe of discourse U=[0, 40 degree C]. R is the rule for associating these subsets, T(X), with the universe of discourse, U, through membership functions. It can be explained by Table 2.1. For example, if the temperature is $35°C$, it belongs the set of Very high with membership value of 0.7. The linguistic variables may also be referred to as fuzzy variables.

Definition 5: Aggregation: An aggregation C of two fuzzy subsets A and B is a two-dimensional fuzzy variable, fuzzy matrix, which can be defined as follows:

$$C = A \otimes B = \begin{pmatrix} \mu_C(1,1) & ... & ... & ... & \mu_C(1,N) \\ ... & ... & ... & ... & ... \\ ... & ... & \mu_C(i,j) & ... & ... \\ ... & ... & ... & ... & ... \\ \mu_C(M,1) & ... & ... & ... & \mu_C(M,N) \end{pmatrix} \tag{3}$$

where the symbol ($\otimes$) denotes an aggregation operator, and

$$\mu_C(i,j) = \min(\mu_A(i), \mu_B(j))$$

$$A = (e_i, \mu_A(e_i))$$

$$B = (d_j, \mu_B(d_j))$$

e_i is the i-th element of the subset A(e) and $\mu_A(i)$ is its membership function. d_j is the j-th element of the subset B(d) and $\mu_B(j)$ is its membership function.

Definition 6: Align turning: An align-turning S of a fuzzy matrix is a one-dimensional fuzzy variable, which has a set of membership functions aligned according to a certain order, as follows:

$$S = [A \otimes B]^{\ominus} \tag{4}$$

where the symbol ($\ominus$) denotes an align-turning operator. The membership function μ_S is pointwise defined as:

$$\mu_S = (\mu_S(1,1), \mu_S(1,2), ..., \mu_S(i,j), ..., \mu_S(m,n))$$

where j varies from 1 to n first and i varies from 1 to m.

Definition 7: Union: A union R of two fuzzy matrices R_1 and R_2 is defined as:

$$R = R_1 \cup R_2 \tag{5}$$

where the symbol ($\cup$) denotes an union operator. The membership function μ_R is pointwise defined as:

$$\mu_R(i,j) = \max(\mu_{R_1}(i,j), \mu_{R_2}(i,j))$$

Definition 8: Intersection: A intersection R of two fuzzy matrices R_1 and R_2 is defined as:

$$R = R_1 \cap R_2 \tag{6}$$

where the symbol ($\cap$) denotes an intersection operator. The membership function μ_R is pointwise defined as:

$$\mu_R(i,j) = \min(\mu_{R_1}(i,j), \mu_{R_2}(i,j))$$

Definition 9: Max-min composition: A max-min composition of a m×n fuzzy matrix M_1 and a n×p fuzzy matrix M_2 is a m×p fuzzy matrix M defined as:

$$M = M_1 \circ M_2 \tag{7}$$

where the symbol ($\circ$) denotes a max-min composition operator. The membership function μ_M is pointwise defined as:

$$\mu_M(i,j) = \max_k(\min(\mu_{M_1}(i,k), \mu_{M_2}(k,j)))$$

where:

$$M_1 = \{m_1, \mu_{M_1}(i,k)\},$$
$$M_2 = \{m_2, \mu_{M_2}(k,j)\},$$
$$M = \{m, \mu_M(i,j)\},$$
$$i \in [1,l], \quad j \in [1,m], \quad \text{and} \quad k \in [1,n].$$

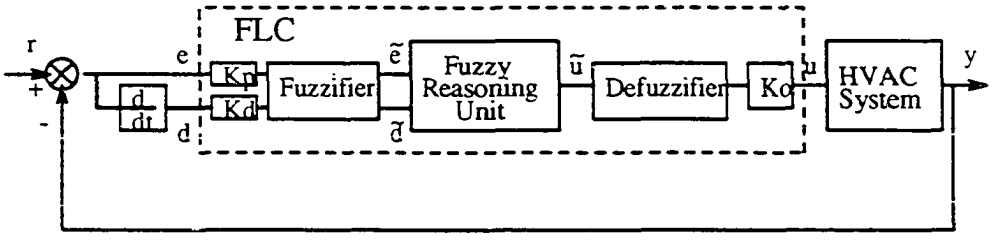


Figure 2.1: The Block Diagram of a Fuzzy Control System

Definition 10: Complement: $\bar{A}$ is referred the complement of a fuzzy subset A. Its membership function is pointwise defined for all $u \in U$ by

$$\mu_{\bar{A}(u)} = 1 - \mu_A(u) \tag{8}$$

The membership function of the complement matrix $\bar{R}$ is pointwise defined by

$$\mu_{\bar{R}(i,j)} = 1 - \mu_R(i,j) \tag{9}$$

Fuzzy Reasoning and Fuzzy Logic Control

A fuzzy logic controller (FLC) includes three parts: fuzzifier, fuzzy reasoning unit, and defuzzifier. The fuzzifier converts ordinary inputs into their fuzzy counterparts, the fuzzy reasoning unit creates fuzzy control signals based on these fuzzy variables, and the defuzzifier converts the fuzzy control signals into the real control outputs. The block diagram of a fuzzy logic controller is shown in Figure 2.1, where e, d, and u are tracking error, derivative error and output control action, $\tilde{e}$, $\tilde{d}$, and $\tilde{u}$ are their fuzzy counterparts respectively, y is the controlled parameter, and r is the set point for y. K_p is the scale factor for e, K_d is the scale factor for d, and K_o is the output gain.

The input universe of discourse for tracking error e and derivative error d is divided into several degrees connected with a number of fuzzy subsets by membership functions. In this study, e and d can each range from -6 to +6, and 13 degrees are used:

$$-6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6.$$

Seven fuzzy subsets are defined for fuzzy variables, $\tilde{e}$ and $\tilde{d}$, as:

$$\text{T(e) or T(d)} = \{\text{NL, NM, NS, ZZ, PS, PM, PL.}\}$$

where the first letters "N" and "P" mean negative and positive, the second letters "L", "M", and "S" mean large, middle and small, and "ZZ" means zero. These degrees and fuzzy subsets are shown in Table 2.2 which uses a 1.0-0.8-0.5-0.1 distribution. For example, if e=3, then its membership in PL is 0.1, its membership in PM is 0.8, etc.

A similar analysis is given to the output universe of discourse for the control action indicated in Table 2.3 which uses a 1.0-0.7-0.2 distribution, where the abbreviations mean that the output control actions are Large Increasing (Level 7), Middle Increasing (Level 6), Small Increasing (Level 5), No Change (Level 4), Small Decreasing (Level 3), Middle Decreasing (Level 2), and Large Decreasing (Level 1). This distribution, combined with the 1.0-0.8-0.5-0.1 distribution for inputs, gave a good performance in the experiments.

In this study, a fuzzy singleton is used as a fuzzification strategy, which interprets an input, e (or d), into a fuzzy value, $\tilde{e}$ (or $\tilde{d}$), with membership function (μ) equal to zero except at the element nearest to the real input, where μ=1.0. For example, if e=3.2, the nearest element is 3, then the fuzzy singleton will be:

Table 2.2: The Membership Function of Input of FLC

A(e), B(d)	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
PL	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.5	0.8	1.0
PM	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.5	0.8	1.0	0.8	0.5
PS	0.0	0.0	0.0	0.0	0.0	0.1	0.5	0.8	1.0	0.8	0.5	0.1	0.0
ZZ	0.0	0.0	0.0	0.1	0.5	0.8	1.0	0.8	0.5	0.1	0.0	0.0	0.0
NS	0.0	0.1	0.5	0.8	1.0	0.8	0.5	0.1	0.0	0.0	0.0	0.0	0.0
NM	0.5	0.8	1.0	0.8	0.5	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NL	1.0	0.8	0.5	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 2.3: The Membership Function of Output of FLC

C(u)	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
LI (Level 7)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.7	1.0
MI (Level 6)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.7	1.0	0.7	0.2
SI (Level 5)	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.7	1.0	0.7	0.2	0.0	0.0
NC (Level 4)	0.0	0.0	0.0	0.0	0.2	0.7	1.0	0.7	0.2	0.0	0.0	0.0	0.0
SD (Level 3)	0.0	0.0	0.2	0.7	1.0	0.7	0.2	0.0	0.0	0.0	0.0	0.0	0.0
MD (Level 2)	0.2	0.7	1.0	0.7	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LD (Level 1)	1.0	0.7	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

$$\tilde{e} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0.)$$

This fuzzy singleton has membership function μ=1.0 at the point of element e=3.

The control rules expressed in natural language can be simplified to have the following form:

IF (e is A) AND (d is B) THEN (u is C)

where A, B, and C are fuzzy subsets defined on the universes of discourse of e, d, and u, respectively. Every rule is interpreted into a fuzzy reasoning matrix:

$$R_k = [A_k(e) \otimes B_k(d)]^{\ominus} \otimes C_k(u) \qquad \text{k = (1,N)} \tag{10}$$

where N is the number of rules. The general fuzzy relation matrix R_g can be constructed as the union of the individual rules:

$$R_g = \cup_{k=1}^{N} R_k \tag{11}$$

This matrix represents the relationship between the fuzzy inputs and the fuzzy control output. The fuzzy control output can then be calculated from the known fuzzy input $\tilde{e}$ and $\tilde{d}$ by:

$$\tilde{u} = \left[\tilde{e} \otimes \tilde{d}\right]^{\ominus} \circ R_g \tag{12}$$

The defuzzifier converts the fuzzy control output created by the rule-based fuzzy reasoning unit into a real control action. In this study, a weighted combination method is used as a defuzzification strategy, which can be explained by the following example, if:

$$\tilde{u} = (0, 0, 0, 0, 0, 0, 0, 0.2, 0.4, 0.8, 0.7, 0.5, 0.1.)$$

then

$$u = \frac{0.2(1) + 0.4(2) + 0.8(3) + 0.7(4) + 0.5(5) + 0.1(6)}{0.2 + 0.4 + 0.8 + 0.7 + 0.5 + 0.1} = 3.4.$$

CHAPTER 3. MULTILEVEL RELAY PROPERTY

Fuzzy logic controllers have much better performance than the conventional controllers, such as PID controllers. PID controllers are linear regulators whose control output is proportional to the tracking error (e), the integral error (i), and the derivative error (d). The PID controllers' intrinsic limitation can be illustrated in Figure 3.1 where the horizontal axis represents time and the vertical axis represents the controlled parameter (CP). When a PID controller is adjusted to respond quickly to a step input, it will have large overshoot (curve 1), and if it is adjusted to eliminate the overshoot, a long rise time will occur (curve 3). So, in most applications, PID controllers are usually adjusted to have the form of curve 2: there is a little overshoot while the response is moderate.

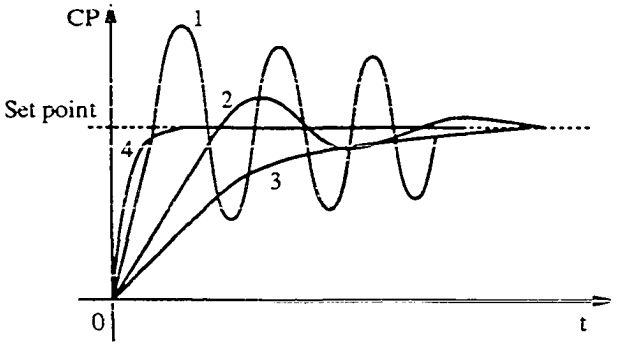


Figure 3.1: The Typical Behavior of PID Controllers

Table 3.1: A Simple Rule Set

u e / d	NL	NM	NS	ZZ	PS	PM	PL
PL	LD(1)	SD(3)	SD(3)	NC(4)	LI(7)	LI(7)	LI(7)
PM	LD(1)	SD(3)	SD(3)	NC(4)	MI(6)	LI(7)	LI(7)
PS	LD(1)	MD(2)	SD(3)	NC(4)	MI(6)	LI(7)	LI(7)
ZZ	LD(1)	MD(2)	SD(3)	NC(4)	SI(5)	MI(6)	LI(7)
NS	LD(1)	LD(1)	MD(2)	NC(4)	SI(5)	MI(6)	LI(7)
NM	LD(1)	LD(1)	MD(2)	NC(4)	SI(5)	SI(5)	LI(7)
NL	LD(1)	LD(1)	LD(1)	NC(4)	SI(5)	SI(5)	LI(7)

The behavior of an ideal controller to respond to a step input is shown by curve 4 in Figure 3.1: very fast response and no overshoot. FLCs can be used to provide a close ideal behavior. The intrinsic feature of a FLC is similar to that of a multilevel relay [Kickert 1978]. This property becomes apparent when the control rules are presented in the framework shown in Table 3.1 and Figure 3.2. Table 3.1 presents the rule set. There are two inputs, the tracking error (e) and derivative error (d), and one output, the control action level (u). Both inputs have 7 fuzzy values discussed in Chapter 2 and form 49 possible cells on the linguistic plane. The output of control actions also have 7 fuzzy values indicated simply by levels 1 to 7. The 49 possible cells are connected with different control action levels according to the rule set and form the control surface visually shown in Figure 3.2. More explicitly, if the derivative error (d) is fixed at ZZ, the control surface will be reduced to a ladder line shown in Figure 3.3.

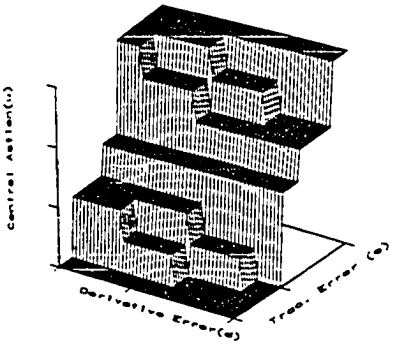


Figure 3.2: The Control Surface for The Rule Set of Table 3.1

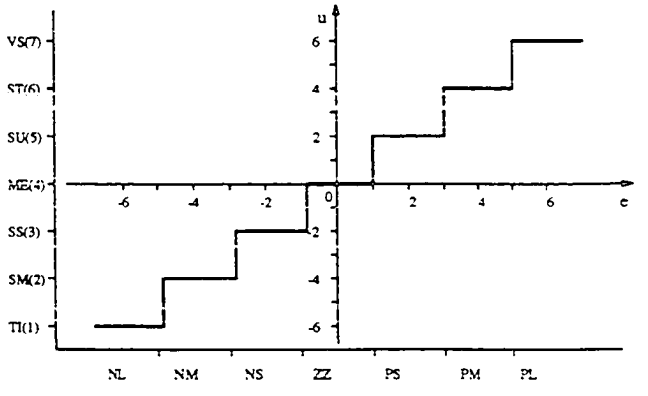


Figure 3.3: The Multilevel Relay Feature of FLCs

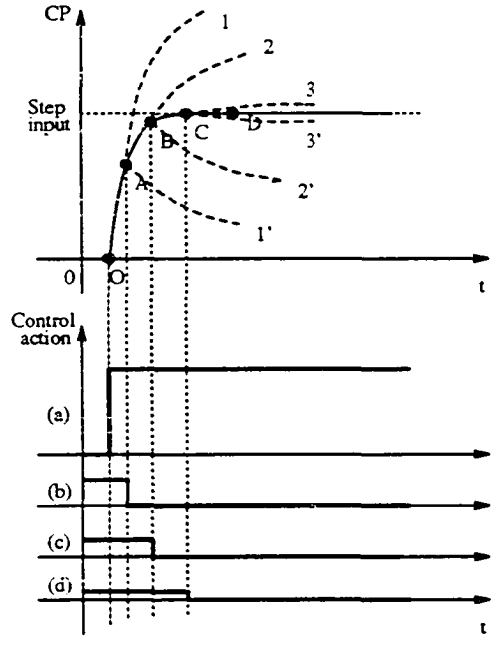


Figure 3.4: The Multilevel Relay Analysis of FLCs

The multilevel relay property of FLCs is explained in more detail in Figure 3.4 which expresses the response of a FLC to a step input. At the initial stage, the FLC responds strongly to the step input with a high control action level. This is fulfilled by creating a step output (a) which makes the controlled parameter (CP) rise quickly from the start point O to point A and will continue along with curve 1. But, at the point A, the control output of FLC jumps to a lower level, which is like adding an inverse step output (b) to the control action. The inverse step output will make the controlled parameter (CP) change along with curve 1'. Adding the curve 1' to the original curve 1, the new path of CP becomes the curve 2. Along this new path, CP changes from point A to point B. At this moment, another inverse step output (c) occurs and creates the control action which makes CP changes along the new curve 3. The controlled parameter then changes from point B to point C. Continuing

this analysis, CP changes from O, A, B, C, to D, and the FLC has performance in response to a step input that is much closer the ideal performance.

CHAPTER 4. RULE DEVELOPMENT AND ADJUSTMENT STRATEGIES

Rule Refinement

An Fuzzy Logic Controller (FLC) is characterized by a set of linguistic statements which are usually in the form of "IF-THEN" rules. The initial set of rules is usually constructed based on the operators' experience, or sometimes by analyzing the dynamic process of the controlled plant. Both approaches require modifying the initial set of rules to obtain an optimal rule set. This is called rule refinement.

Figure 4.1(a) shows an initial rule set analyzed on a linguistic plane. The initial rule set is expected to be adapted to the controlled plant as well as possible. The horizontal axis expresses the fuzzy subsets defined on the universe of discourse for the tracking error (e), and the vertical axis expresses the fuzzy subsets defined on the universe of discourse for the derivative error (d). Both have 7 fuzzy values: NL, NM, NS, ZZ, PS, PM, PL. On the cross points of these fuzzy values there are output control action levels which are also fuzzy subsets having 7 "values" from level 1 (LD) to level 7 (LI). For example, the cross point of e=NM and d=PM indicates u=Level 3. This corresponds to the rule:

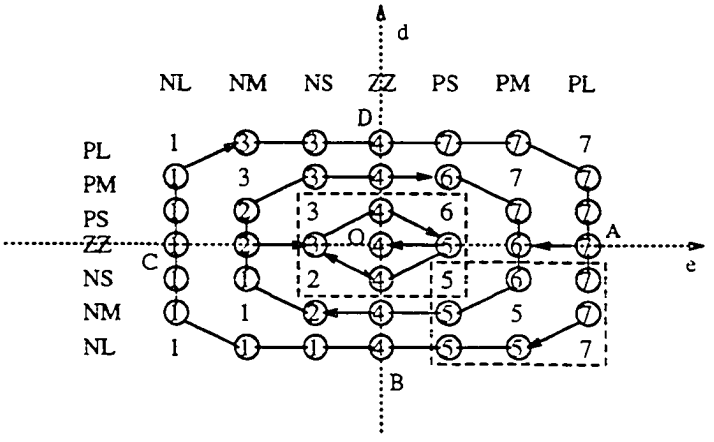
IF (e is NM) AND (d is PM) THEN (u is Level 3)

At this point, the initial rule set is based on the following control strategies. First, it tries to keep a proportional relationship between the control action (u) and the tracking error (e). Note that if the derivative error (d) is ZZ, then the output control action (u) increases from level 1 to level 7 when the tracking error (e) changes from NL to PL. Secondly, the influence of derivative error (d) is considered such that if it is positive then increase the control action (u) a little bit, and if it is negative then decrease the control action (u). For example, if the tracking error (e) keeps PM, the control action (u) increases from level 6 to level 7 when the derivative error (d) is positive, and it decreases from level 6 to level 5 when the derivative error (d) is negative.

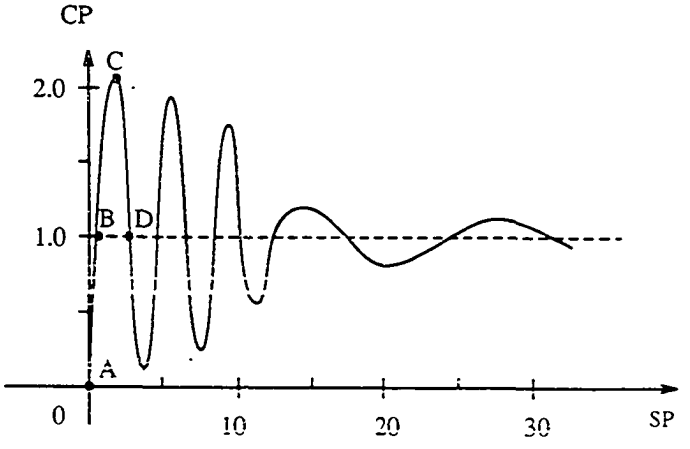
Consider a second order plant with a transfer function:

$$H(s) = \frac{1.0}{s^2 + 0.1s + 1.0} \tag{12}$$

that is controlled by FLC using the initial rule set to respond to a step input. The performance trajectory of the FLC for computer simulation is shown by the arrows in Figure 4.1(a) and the dynamic process of the normalized controlled parameter (CP) is shown in Figure 4.1(b) where the horizontal axis indicates the number of sample period (SP). Obviously, the dynamic process can be divided into two stages. At the first stage, there is a strong oscillation with a higher frequency, and at the second stage, there is a moderate swing with a smaller frequency. Looking at the performance trajectory in the linguistic plane, we can see that the stronger oscillation occurs at the out-cycle (points further from the center). As time increases, the state moves to the in-cycle near the center of the plane and becomes moderate. This shows that FLCs have the desirable property of a structure-variable controller.The rules at



(a) The initial rule set and performance trajectory on linguistic plane



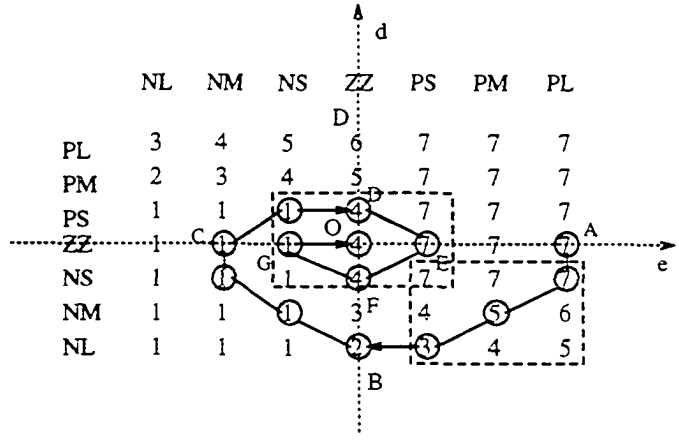
(b) The dynamic process

Figure 4.1: The Initial Rule Set Expressed on Linguistic Plane and Its Simulation of Step Response

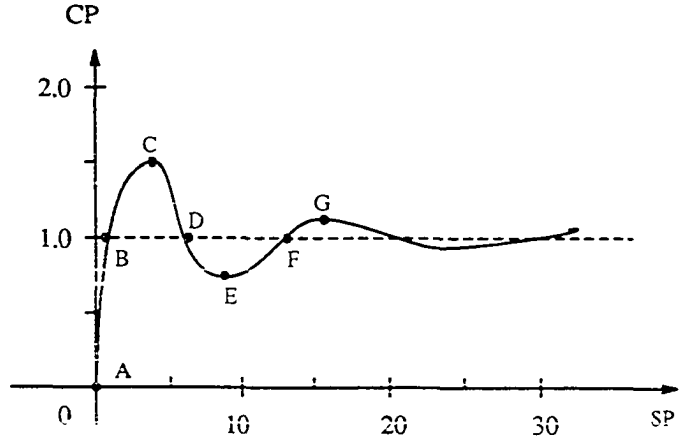
the out-cycle belong to one kind of structure for the first stage, and the rules at the in-cycle belong to another structure for the second stage.

Obviously, the initial rule set does not satisfy a good design for a controller. It can be modified by intuitive reasoning using "observation and error" method. A rule set is often symmetrically positioned about the central point which is the desired stable operating point where the tracking error (e) and the derivative error (d) both equal zero and the control action (u) does not change. When a positive step change is imposed to the set point, the tracking error (e) has the biggest value and the derivative error (d) is zero at the beginning time (point A in the linguistic plane). With the regulating action, the tracking error (e) will decrease, the derivative error (d) will be negative, and the performance trajectory will enter into the right-bottom block in the linguistic plane. So, the rules in this area have the most important effect on the behavior of the first stage of the dynamic process. The most important area responsible for the behavior of the second stage is the central block.

To avoid strong oscillations, it is apparent that the control actions in the right-bottom block should be decreased. The modified rule set and its simulation of response to a step input are shown in Figure 4.2. The performance trajectory expressed in the linguistic plane is just like spiral shown in Figure 4.2(a). We can see that the performance of the control system has been improved, but a small oscillation still exists and there is a little overshoot indicated by point C in Figure 4.2(b). Once again the rule set is modified and the final rule set and its simulation of response to a step input are shown in Figure 4.3. The final rule set gives good performance with a short rise time and a very small overshoot and it is considered satisfactory.

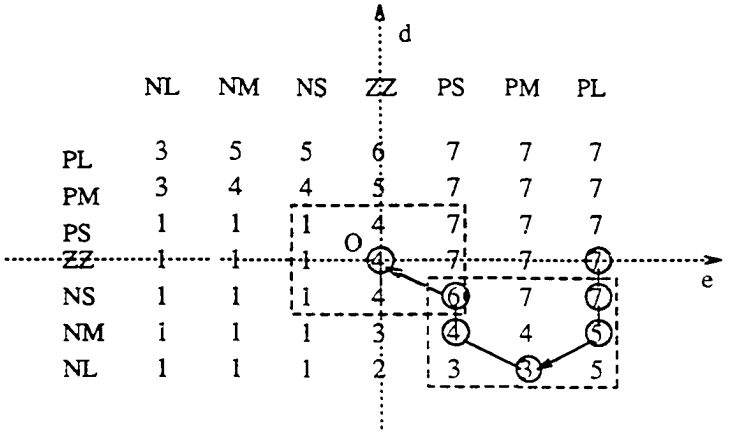


(a) The second rule set and performance trajectory on linguistic plane

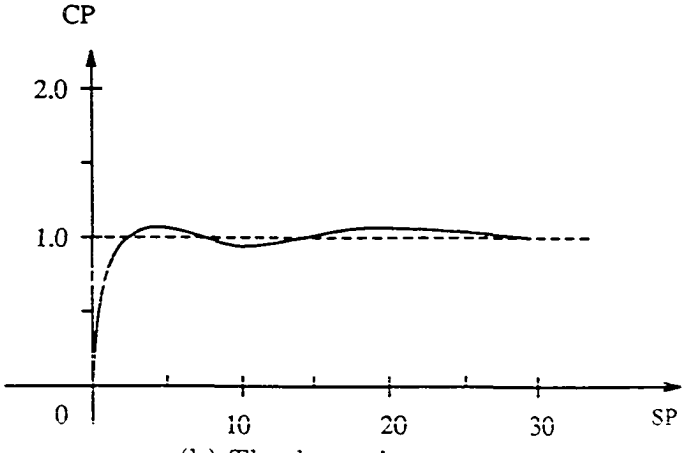


(b) The dynamic process

Figure 4.2: The Second Rule Set Expressed on Linguistic Plane and Its Simulation of Step Response



(a) The final rule set and performance trajectory on linguistic plane



(b) The dynamic process

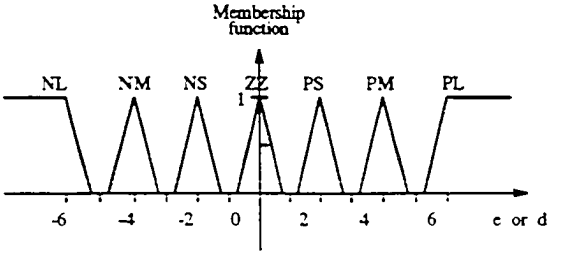
Figure 4.3: The Final Rule Set Expressed on Linguistic Plane and Its Simulation of Step Response

By analyzing the performance trajectory on the linguistic plane, a rule set is refined. It relies heavily on intuitive reasoning by comparing the dynamic process of the controlled parameter for the present rule set with the desired one.

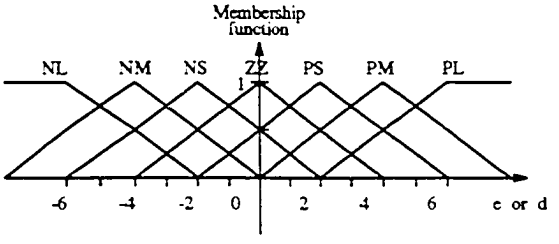
Completeness and Interaction of Rules and Selection of Membership Functions

The second significant influence on the behavior of an FLC is from the membership functions. They should be chosen carefully in the adjustment process. As mentioned above, the fuzzy subsets, linguistic values, NL, NM, NS, ZZ, PS, PM, and PL, are defined on the universe discourse of tracking error (e) and derivative error (d). Some possible membership functions are shown in Figure 4.4. The membership functions should be chosen to make these linguistic values have suitable coverage on the universe of discourse. For the case of Figure 4.4(a), the whole range is not covered by these linguistic values. There are some values of e or d, on which the membership functions of all linguistic values are zero. In this case, an empty output control action could be created. This means that the control actions are lost for those points which are not covered by any input fuzzy subset. This is referred as the non-completeness of control rules. FLCs should satisfy the condition of completeness for their membership functions. The membership function shown in Figure 4.4(a) can not be used for an effective fuzzy logic controllers. In other words, the union of all fuzzy subsets, X_i, $i = [1, 7]$, should be greater than zero for all $e \in E$, i.e.

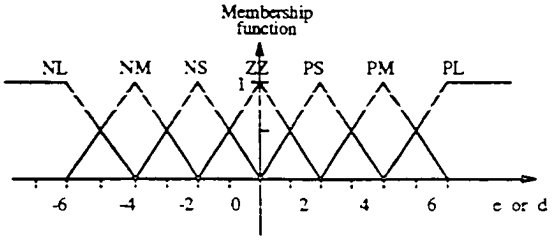
$$\forall\ e \in E \qquad \cup_{i=1}^{7} X_i(e) > 0 \tag{13}$$



(a) Non-completeness case



(b) Heavy overlap case



(c) Moderate overlap case

Figure 4.4: Three Membership Functions with Different Coverage

On the other hand, there can be interaction among the rules if the overlap of fuzzy subsets occurs on the range of the universe of discourse. In this case, the membership functions have the forms shown in Figure 4.4(b) and (c). The interaction tends to smooth out the set of control rules. Consider the single-input-single-output case for simplicity, the rule set is:

IF (e is A_i) THEN (u is C_i) i=[1,N]

where N is the number of rules in the set. These rules are incorporated into a fuzzy relation matrix as follows:

$$R = \cup_{i=1}^{N} R_i = \cup_{i=1}^{N} (A_i \otimes C_i) \tag{14}$$

If the fuzzy value of input e is known as $\tilde{e}$, the fuzzy output $\tilde{u}$ then can be calculated, mentioned in Chapter 2, as follows:

$$\tilde{u} = \tilde{e} \circ R \tag{15}$$

If $\tilde{e}$ is A_i, $\tilde{u}$ is expected to be C_i. But now the interaction of rules due to overlap results in:

$$C_i \subseteq A_i \circ R \tag{16}$$

The equality is established only when no overlap occurs. This analysis is based on the fuzzy logic scheme including max-min composition operator. A more detailed example of the numeric calculation is given in the Appendix.

If the overlap is heavy as shown in Figure 4.4(b), there will be large deformation and the control rules will lose their original shape. In the limit, as the membership functions become unity for all values, the output of the FLC will always be the same

fuzzy quantity. This means that the fuzzy reasoning system conveys no valuable information and the FLC has lost its efficacy.

A moderate overlap, shown in Figure 4.4(c), is desirable to allow for fuzzy reasoning with uncertainty and the need for completeness of the control rules. How does one determine the "size" of overlap? At present, we use intuitive judgment to choose membership functions when adjusting an FLC. There appears to be some latitude in choosing the amount of overlap, on which the performance of an FLC does not change significantly. The quantitative analysis is the further research topic for FLCs.

When we modify the control rules in the linguistic plane, the overlapping membership functions let the rules near the performance trajectory have an effect on the output control actions. This is because the interactions occur among the neighboring rules.

Scale Factors and Output Gain

The scale factors, K_p and K_d, and the output gain K_o, shown in Figure 2.1, also have significant influence on the behavior of an FLC. Their influence is not as complicated as those of rules and membership functions. The adjustment for the scale factors and output gain is comparatively simple.

The scale factor K_p relates the actual range of tracking error (e) to the universe of discourse (E) defined in the fuzzy logic system. In this work, E consists of 13 degrees. Then K_p is determined as the ratio of the range of E to the range of the

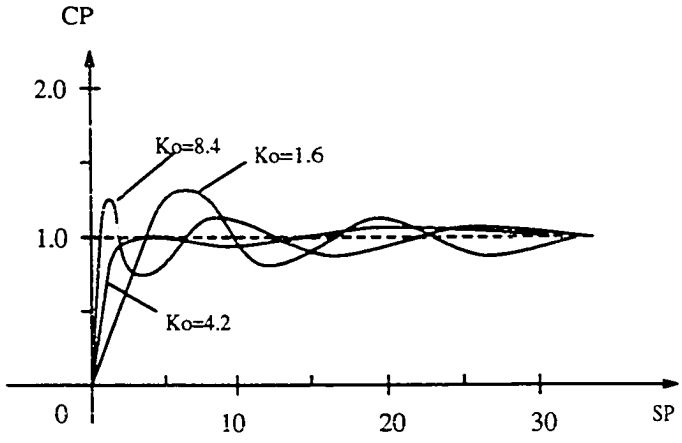


Figure 4.5: The Influence of K_o on the Behavior of FLCs

real variable:

$$K_p = \frac{E_{max} - E_{min}}{e_{max} - e_{min}} \tag{17}$$

For scale factor K_d, there is the similar analysis leading to:

$$K_d = \frac{D_{max} - D_{min}}{d_{max} - d_{min}} \tag{18}$$

where D is the universe of discourse for derivative error (d) defined in the fuzzy logic system. Small K_p or K_d will narrow the control band, while large K_p or K_d will lead to loss of control for large inputs.

The output gain K_o is defined as follows:

$$K_o = \frac{u_{max} - u_{min}}{U_{max} - U_{min}} \tag{19}$$

It is the ratio of range of real output control action (u) to the range of its universe of discourse (U) defined in the fuzzy logic system. K_o acts as an amplification factor of the whole FLC. Figure 4.5 shows the influence of K_o on the step response simulation of an FLC with the final rule set shown in Figure 4.3. Increasing K_o results in a

shorter rise time. The performance trajectory in the linguistic plane will become steeper for the first stage and oscillation occurs. Decreasing K_o results in a longer rise time and the performance trajectory in the linguistic plane will become moderate during the first stage. But, in the computer simulation, oscillation still occurred. This is because different values of K_o result in a new route of the performance trajectory which will activate the different rules causing possible oscillations. So the influence of output gain, K_o, should be considered together with the change of the activated rules.

CHAPTER 5. EXPERIMENTAL IDENTIFICATION OF RULE DEVELOPMENT AND ADJUSTMENT STRATEGIES

Fuzzy reasoning is much closer to the way humans think and the fuzzy logic controller (FLC) algorithm is based on a set of linguistic rules which describe control strategies where the words are defined as fuzzy subsets. As mentioned in Chapter 4, the optimal rule set is developed from an initial rule set. In this chapter, rule development and adjustment strategies of an FLC for heating mode is experimentally identified.

Experiment

An air flow test loop [Maxwell et al. 1986] was used in this study as the experimental setup to develop the rule sets and verify the adjustment strategies for an FLC. This facility, shown in Figure 5.1, consists of a supply section and a load section. The supply section uses a commercially available air handling unit and the load section includes heating, cooling, humidifying, and dehumidifying capabilities. A pneumatic control system is used for normal control tasks, and electro-pneumatic transducers are used to allow computerized control of the test loop.

In this experiment, the outdoor air was heated by the hot water coil in the supply section and then cooled by the chilled water coil in the load section. The fan speed

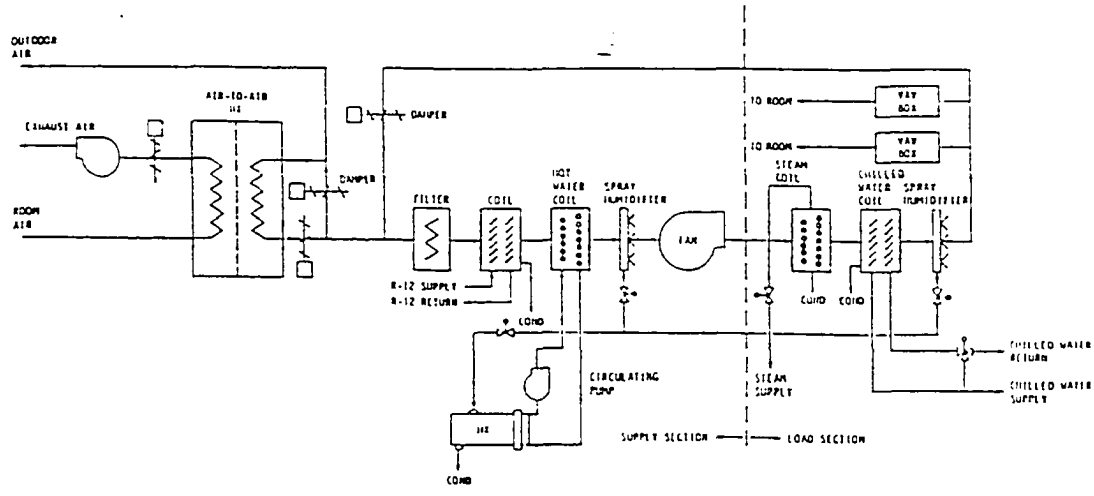


Figure 5.1: The Air Flow Test Loop Used in the Experiment

was fixed to keep a certain air flow rate. At the start of each test, the temperature of the return air was stable at 18.0 degree C. A step increase of 5.0 degree C was then imposed on the set point. The FLC controlled a pneumatic valve which regulated the steam flow rate to a heat exchanger (HX) which controlled the temperature of the hot water in the hot water coil.

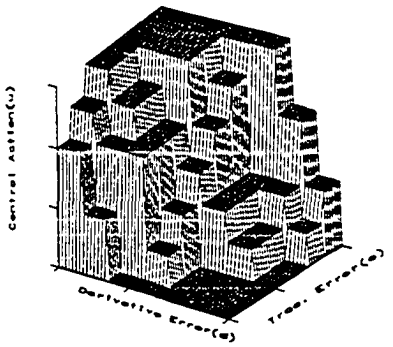
The range of tracking error (e) of the return air temperature was from -5 to +5 degree C and the range of derivative error (d) was from -0.10 to +0.10 degree C per sampling period (C/SP). The sample period was 3.41 seconds, and K_p and K_d were calculated to be 1.2 and 60 by equations (17) and (18). In the adjustment process, K_d was refined to 75 and K_p was not changed for the optimal behavior of the FLC. The range of the control output change was from -0.15 to +0.15 volt per sampling period (V/SP) and K_o was calculated to be 0.025 by Equation (19). This value was updated to 0.035 at the end of the adjustment process.

The membership functions were chosen to have moderate overlap with a 1.0-0.8-0.5-0.1 distribution for input fuzzy subsets and a 1.0-0.7-0.2 distribution for output fuzzy subsets. In the adjustment process, the "shapes" of the membership functions were not changed.

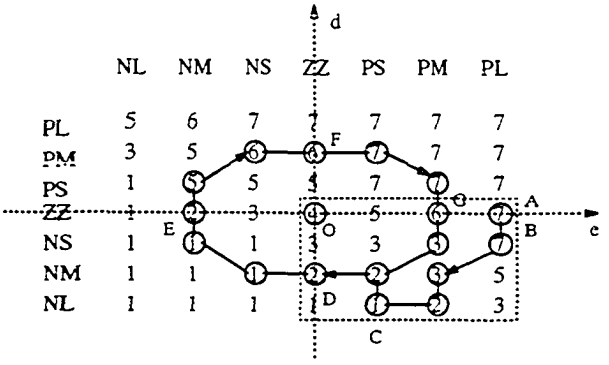
The initial rule set chosen in this study was based on the result of simulations and an intuitive understanding of the dynamic properties of the air flow test loop. Its control surface is shown in Figure 5.2(a) and its performance trajectory on the linguistic plane is shown in Figure 5.2(b). The system has a large delay time which delays the change in the controlled parameter (T) from point A to point B. This results in strong heating because point A has the highest output (level 7). For this reason, the output levels on the right-bottom block on the linguistic plane were decreased. The resulting dynamic curve of the return air temperature is shown in Figure 5.2(c).

The oscillation of dynamic process indicates that the output of control action is still too strong in the first stage and makes the performance trajectory, passing through the point D, reach point E, and not the central point O in Figure 5.2(b), the stable state point. The rule set is improved by adjusting the control output to reduce the control action levels along the performance trajectory. The control action levels of the points on and near the performance trajectory are updated because of the reaction between the neighboring rules. The improvement was experimentally checked step by step using the dynamic curve of the controlled parameter.

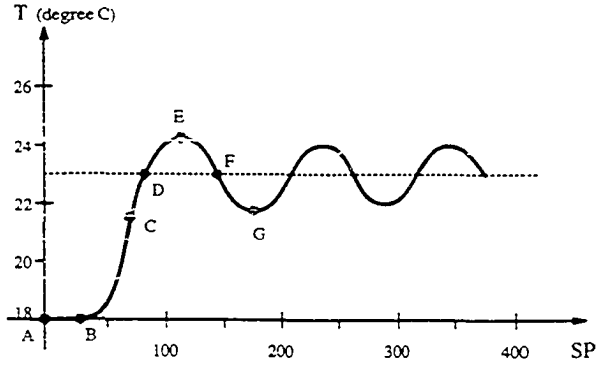
The optimal rule set, shown in Figure 5.3, was finally established. Comparing it with the initial rule set, the output control levels of most rules located in the right-



(a) Control surface

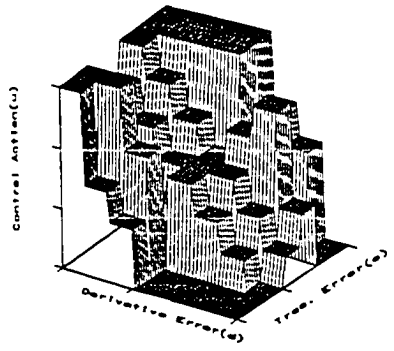


(b) Performance trajectory

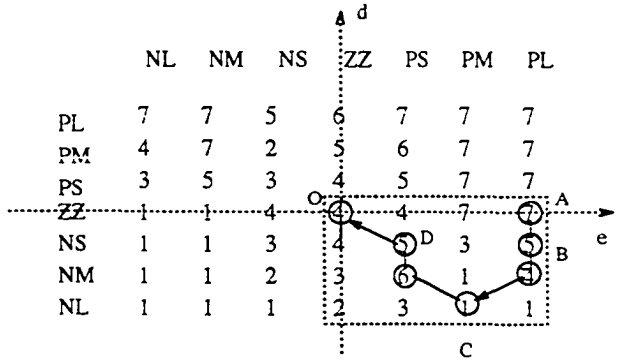


(c) Dynamic process

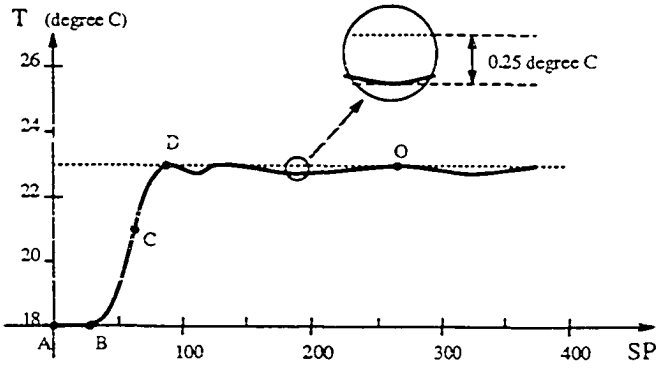
Figure 5.2: The Explanation and Analysis of Initial Rule Set



(a) Control surface



(b) Performance trajectory



(c) Dynamic process

Figure 5.3: The Explanation and Analysis of Final Rule Set

bottom box on the linguistic plane were decreased. At point A, the control action is still kept at level 7 and lasts a long time, due to the time-delay of the system. This results in the controlled parameter (T) rising quickly from point B to point D in Figure 5.3(C). For the cases where e=PL, and d=NS and NM, the control action levels are reduced from 7 to 5, and 5 to 4 respectively, to slow down the increase of the controlled parameter (T) in the section from point C to point D. When the performance trajectory reaches point C on the linguistic plane, where the control action level has been reduced from 2 to 1, the lower control action level lets the dynamic process curve have a sharp turning near the point D. Thus it does not pass the set value (23 degree C in this experiment) with overshoot. Note that, at the point C, the controlled parameter is rapidly increasing with a large rate, but the control action produced at this time will affect its behavior near point D due to the time-delay of the system. The next point of the performance trajectory, where e=PS and d=NM, has an output control action of level 6 which keeps the controlled parameter (T) from going down too much. This higher level will not result in an overshoot because its duration is not long enough. The performance trajectory reaches the stable point O with the medium control action of level 4. The final rule set has good performance with a short rise time and no overshoot. The steady-state error is 0.25 degree C.

Figure 5.4 shows the control action output of the FLC, where the horizontal axis indicates the number of sample periods (SP) and the vertical axis indicates the change of the control action with the unit of volts. It is interesting to identify the multilevel relay property of FLCs by making a "shape line" for the FLC's output and putting it together with the controlled parameter's curve, shown in Figure 5.5.

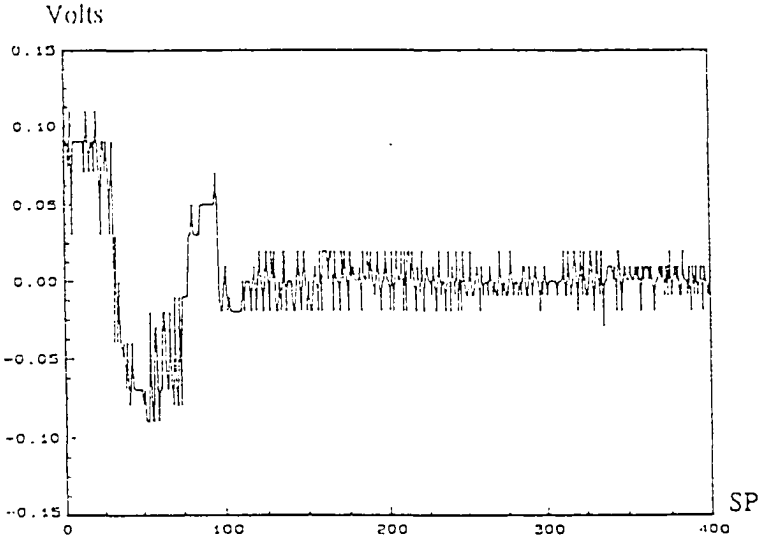


Figure 5.4: The Control Action Output of the FLC

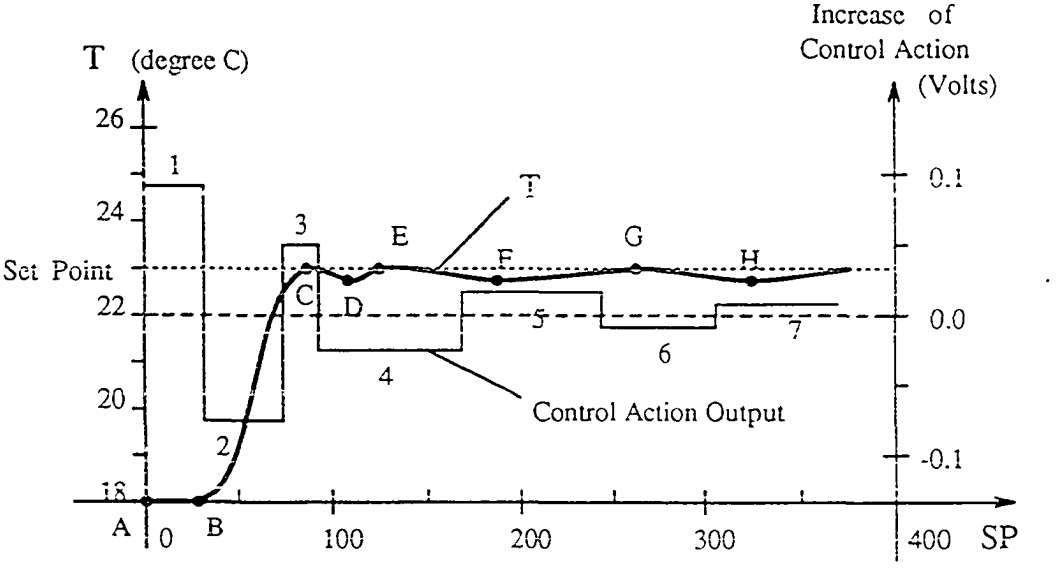


Figure 5.5: The Multilevel Relay Analysis for the Experimental Result

Initially, the FLC creates a high level control action, marked with number 1, and the system has no immediate response because of the time-delay. After about 25 sample periods (SP) or 85 seconds, the controlled parameter (T) starts to rise from point B. Then the FLC's output jumps to a low level marked with number 2 which results in a sharp turning at point C. Next, the control action goes up to a medium high level, number 3, which makes the controlled parameter (T) rise again from point D to point E. After point E, the dynamic process comes to the area where the controlled parameter (T) is stable with only small deviations from the set point (23 degree C). The output control action marked with number 4 lets the controlled parameter change from point E to point F and the output control action marked with number 5 corresponds to the change from point F to point G and so on.

Experimental Comparison of FLC with PID Controller

An experiment using a conventional PID algorithm was performed for the same operating conditions as using the FLC. The results are shown in Figure 5.6 where the horizontal axis indicates time and the vertical axis indicates the normalized controlled parameter (Y). Curves 2, 3, and 4 express three typical dynamic responses of a PID controller to a unit-step input. Curve 2 has a short rise time and large overshoot, which is for the case where the proportional coefficient (K_p) is 6.0, the integral coefficient (K_i) is 0.1, and the derivative coefficient (K_d) is 13.0. Curve 4 has no overshoot, but its rise time is long. It is for the case where K_p was decreased to 3.0 and K_d was increased to 18.0. With medium K_p of 5.0 and K_d of 14.0, curve 3 has moderate overshoot and rise time. Compared to the PID controller, the dynamic process of the FLC, curve 1, has short rise time and no overshoot.

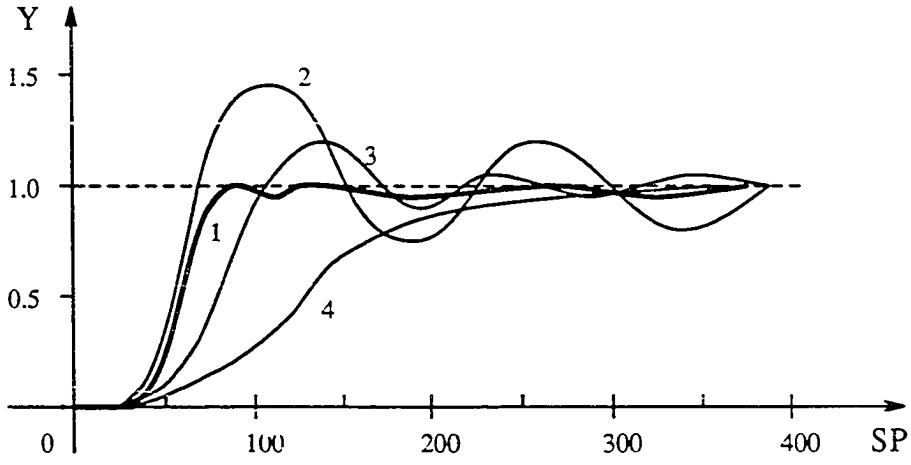


Figure 5.6: Comparison of the FLC with a PID Controller

(Curve 1 is the FLC, and curve 2, 3, and 4 are with PID control.)

CHAPTER 6. DELAY TIME DETERMINATION USING AN ARTIFICIAL NEURAL NETWORK

Fuzzy model identification is the base to establish the initial rule set for fuzzy logic controllers. It involves delay time determination and fuzzy parameter estimation. This chapter is focused on the delay time determination and the fuzzy parameter estimation is the topic of the next chapter.

Problem Formulation

A general HVAC feedback control system is depicted by a block diagram shown in Figure 6.1 where u(t), y(t), r(t), and e(t) are control action, controlled parameter, set-point value, and tracking error respectively. When the set point has a step change, the system will give a response as shown in Figure 6.2. The time for which y(t) travels from 5% to 95% of whole distance is referred as the settling time (T_s) indicated by the period BC. In the period AB, the system is silent and has no response until reaching point B. This period is referred as the delay time (T_d) of the system. The whole period from A to C is referred as the system response time (T_r). The settling time is related to both the system and the controller, but the delay time is only a function of the system.

The delay time can also be defined by using a Dirac pulse [Ogata 1970]. When a

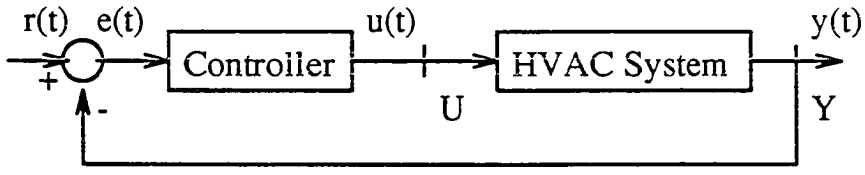


Figure 6.1: The Block Diagram of Feedback Control System.

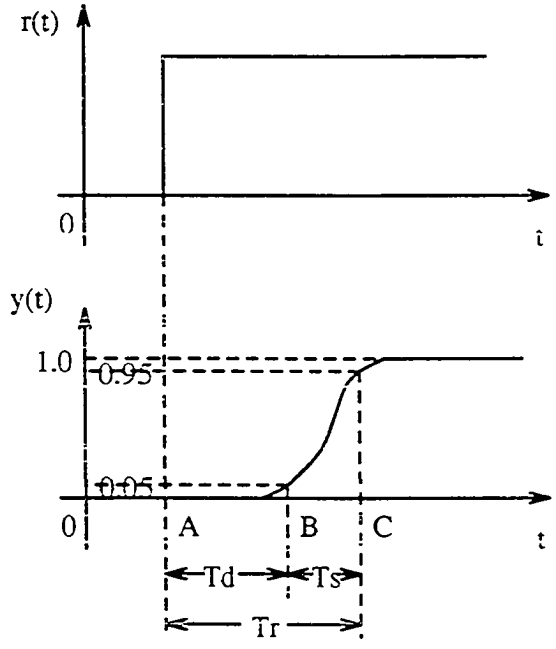


Figure 6.2: The System Response to a Step Input.

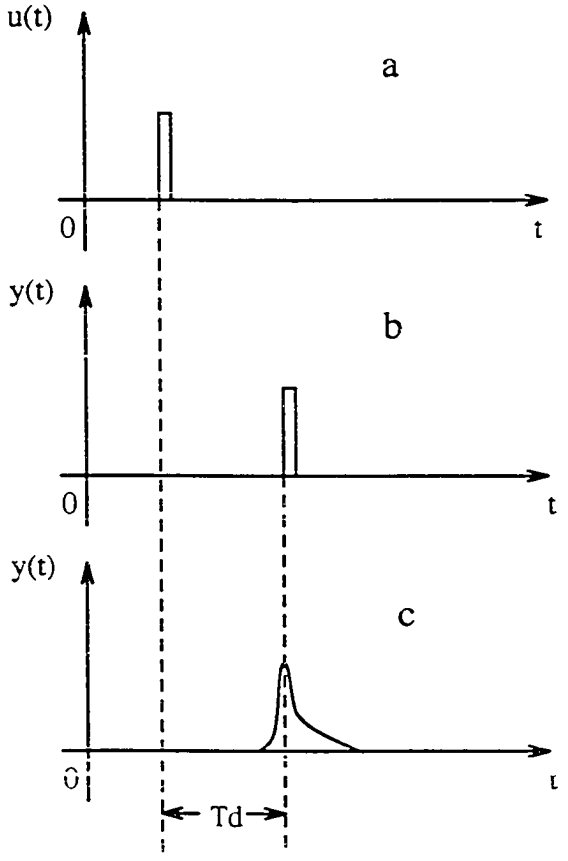


Figure 6.3: The Dirac Input and Its Response.

(a) The Dirac input.
(b) Its response for the ideal case.
(c) Its response for the real case.

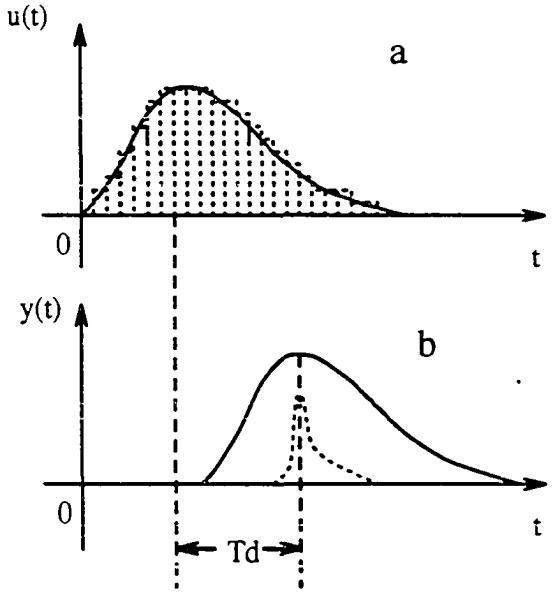


Figure 6.4: The Decomposition of Real Control Action and Its Response.

Dirac pulse, shown in Figure 6.3(a), is forced at point U in Figure 6.1, after some time the plant will give a Dirac response at point Y shown in Figure 6.3(b) for the ideal case, or a response of the shape shown in Figure 6.3(c) for the real case. The period for which the pulse is 'transferred' through the system is the delay time. In the real control environment, the controller outputs a continuous control action which can be decomposed into a series of Dirac pulses shown in Figure 6.4(a). The system will give the response curve for y(t) of the similar shape with u(t) shown in Figure 6.4(b). Every element response has a delay time related to its own input pulse. (Only one element response is depicted in the figure.) The comprehensive response for a linear system then has the same delay time related to the continuous control action. Thus the delay time can be determined by measuring the difference between the control action, u(t), and the controlled parameter, y(t), in the time domain.

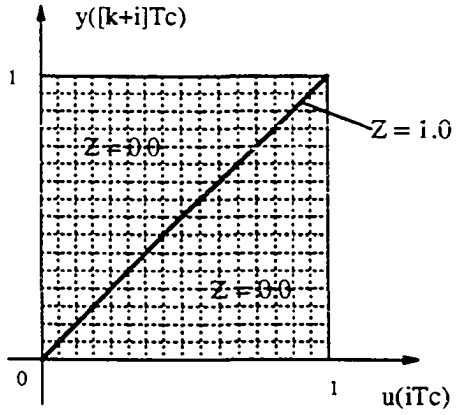


Figure 6.5: The Mapping Function.

An artificial neural network (ANN) can be used to recognize the system delay time. A computerized control system receives input and outputs control actions once in every sampling period. This results in a series of control signals, u(0), $u(T_c)$, $u(2T_c)$, ..., $u(nT_c)$, ..., and a series of the plant responses, y(0), $y(T_c)$, $y(2T_c)$, ..., $y(nT_c)$, $y([n+1]T_c)$, $y([n+2]T_c)$, ..., $y([n+k]T_c)$, ..., where T_c is the sampling period. If the control signal, $u(nT_c)$, is responsible for the plant response, $y([n+k]T_c)$, then the delay time, T_d, is the number of samples, k, times the sampling period, T_c:

$$T_d = k * T_c \tag{20}$$

Based on this analysis, the series of $u(T_c)$, $u(2T_c)$, ..., $u(nT_c)$, ..., and the series of $y(kT_c)$, $y([k+1]T_c)$, $y([k+2]T_c)$, ..., $y([k+n]T_c)$, ..., have similar shape. More over, after normalization, they have the similar values between 0 and 1 for every pair. Thus, we can design a method to find the delay time by checking if the normalized $u(iT_c)$ and $y([k+i]T_c)$ have similar values for an appropriate period, i=1, 2, ..., n. This method serves as a mapping function shown in Figure 6.5. When $u(iT_c)$ and $y([k+i]T_c)$ have the same values, the output, z, is 1.0, otherwise, z is less than 1.0.

The closer the normalized $u(iT_c)$ is to $y([k+i]T_c)$, the closer z is to 1.0. The output, z, is referred to as the meeting factor. In this study, the mapping function is completed by an ANN which has two inputs, one for $u(iT_c)$ and the other for $y([k+i]T_c)$, and one output giving the meeting factor. Let the series of $u(iT_c)$ and the series of $y([k+i]T_c)$ with different k's go though the neural network. The k which makes the ANN produce the largest average meeting factor for a series of sample periods (n is sufficiently large), is used to calculate the delay time with Equation 20..

ANN Construction

46¡ A fully-connected four-layer ANN, shown in Figure 6.6, was used for this work. The ANN has a feed-forward data flow mode and uses a back-propagation learning algorithm. The first layer is the input layer with 2 nodes and the fourth layer is output layer with only 1 neuron. The second and the third layers are hidden layers with 5 neurons in each layer. The neural network literature indicates that networks with 2 hidden layers can approximate any real-valued function and solve arbitrary classification problems [Simpson 1987; Lippmann 1987]. The nodes in the input layer receive input signals from the outside world and directly pass the signals to the next layer. In this work the controller's output and the measured response (temperature) are taken as the inputs for the ANN. They are both normalized to values ranging from 0 to 1. For the hidden layers and the output layer, each neuron behaves as a sigmoid function. The neuron i's output is expressed as:

$$Y_i = F(X_i) = \frac{1}{1 + e^{-C*(X_i+B)}} \tag{21}$$

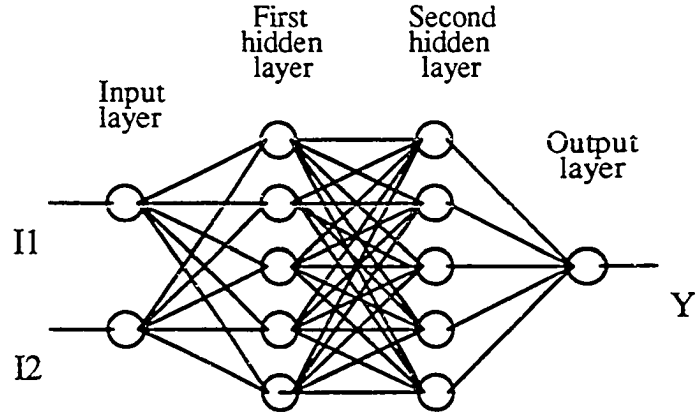


Figure 6.6: The Structure of the Proposed Neural Network.

where B is a bias, C is a positive scaling constant, and X_i is the neuron's activation which is given by

$$X_i = \sum_j (W_{ij} * Y_j) \tag{22}$$

The summation in the Equation 22 is made over all neurons (j's) in the previous layer. Here W_{ij} is the connection weight from neuron i to neuron j, and Y_j is the output of neuron j in the previous layer. Figure 6.7 shows the hidden and output neuron's mathematical model and Figure 6.8 shows their sigmoid function. The neuron in the output layer gives the ANN's output which is between 0 and 1.

Learning Algorithm

Before the ANN can be used to recognize the delay time of an HVAC system, the connection weights must be determined. The process of determining the connection weights is referred to as the learning process. For the learning process, we need a set of input-output pairs to train the proposed ANN. Training the network means establishing the values of these weights between neurons. Thus the ANN learns by

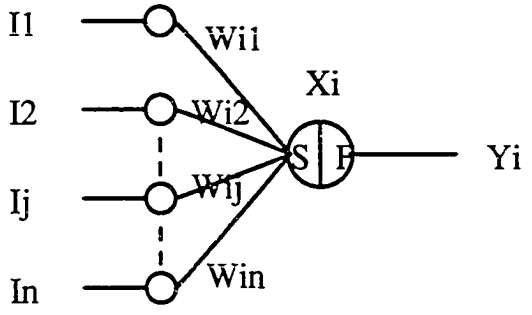


Figure 6.7: The Mathematical Model of Hidden and Output Neurons.

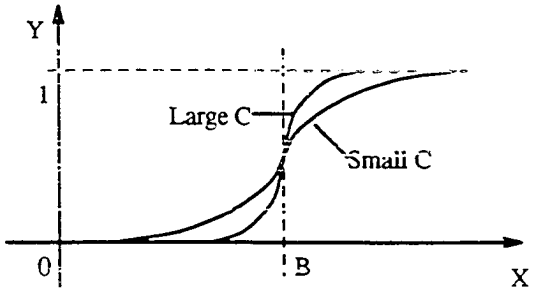


Figure 6.8: The Sigmoid Function of Hidden and Output Neurons.

being trained. In the present work, this is achieved by the known, desired mapping shown in Figure 6.5. An output, between 0 and 1, is produced for every input set by the untrained network. The difference between the calculated output and the desired output is the error. The neuron weights are then updated using backpropagation [Widrow and Lehr 1990] to reduce the error. The error propagates first from the output layer backward to the hidden layers and then to the input layer.

The output square error is described by Equation 23:

$$E = \frac{1}{2}(D_o - Y_o)^2 \tag{23}$$

where D_o is the desired output and Y_o is the output calculated by the ANN. The constant 1/2 is only for mathematical convenience. The method of gradient descent called the Generalized Delta Rule [Kosko 1992] is used to minimize E over all patterns in the training set. By this method the changes of the connection weights are proportional to the negative partial derivative of E with respect to the concerned weight W:

$$\Delta W = -\eta * \frac{\partial E}{\partial W} \tag{24}$$

where η is the learning rate. Then the weights between an output neuron and a neuron k in the second hidden layer are updated:

$$\Delta W_{ko} = \eta * C * Y_k * Y_o * (1 - Y_o) * (D_o - Y_o) \tag{25}$$

where Y_k is the calculated output of neuron k in the second hidden layer. The connection weights between a neuron k in the second hidden layer and a neuron j in the first hidden layer are updated using the equation:

$$\Delta W_{jk} = \eta * C * Y_j * (1 - Y_k) * W_{ko} * \Delta W_{ko} \tag{26}$$

where Y_j is the calculated output of neuron j in the first hidden layer. The connection weights between a neuron j in the first hidden layer and a node i in the input layer are updated using the following equation:

$$\Delta W_{ij} = \eta * C * Y_i * (1 - Y_j) * (\sum_{k=1}^{K} (W_{jk} * \Delta W_{jk})) \tag{27}$$

where Y_i is the calculated output of neuron i in the input layer.

For one batch iteration, the learning procedure is repeated for all 289 training points in the 17 by 17 input space shown in Figure 6.5. The learning process continues by repeating batch iterations until the difference between the desired output and the calculated output is smaller than a specified tolerance.

Improvement of the Neural Network

The neural network performance is mainly related to three elements including the neuron characteristics, the net topology, and the learning method. In this work, the sigmoid function is specified for the neuron dynamics and its parameters, scaling constant (C) and bias (B), determine the neuron characteristics. Large scaling constants result in a steep activation curve and small scaling constants result in a more spread out curve (Figure 6.8). The neuron bias moves the curve right or left. These parameters have different effects on neural network performance. Detailed discussions are presented in many published articles [Kong and Kosko 1992; Yamada and Yabuta 1992; Park 1991]. In this study, the scaling constant, bias, and the learning rate (η) are chosen to be 1.0, -0.5, and 1.2, respectively, based on the results of computer simulations.

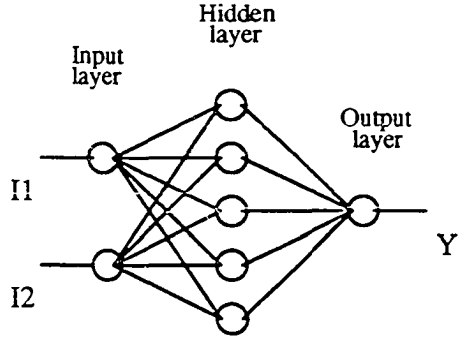


Figure 6.9: The Structure of the Three-Layer Neural Network.

The net topology is characterized by the connective structure of neurons, the number of layers, the number of hidden neurons, and the data flow mode. The proposed ANN is a fully-connected, four-layer net which has 5 neurons in each hidden layer and uses a feedforward flow mode. In this work, two networks have been studied, one is a three-layer net, shown in Figure 6.9, the other is the four-layer net shown previously in Figure 6.6. Five-layer nets are more complex and require more calculations for each training cycle, so they are not commonly used in real applications. The hidden layers for each net have 5 neurons. The computer simulation results of the learning process for both networks are shown in Figure 6.10 and Figure 6.11. In this work, two criteria are used to judge if the correct connection weights are set up in the learning process. One is the average square error (E_a) expressed by Equation 28 and the other is the maximal absolute error (E_m) expressed by Equation 29.

$$E_a = \frac{1}{N_b}\sum_{b=1}^{N_b}(D_b - Y_b)^2 \tag{28}$$

$$E_m = \max_{b\in(1,N_b)}\{abs(D_b - Y_b)\} \tag{29}$$

where D_b is the desired output for input case b of a batch learning cycle, which is 0 or 1, and Y_b is the calculated output, which is between 0 and 1. N_b is the number

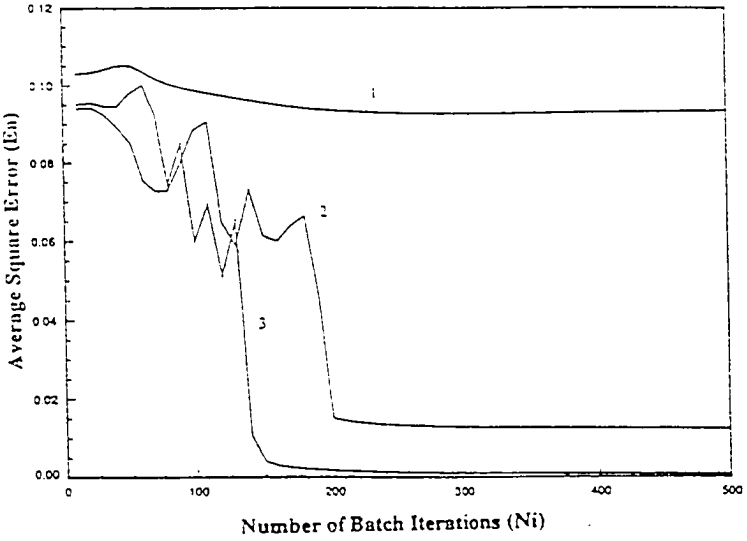


Figure 6.10: The Relationship of Average Square Error (E_a) to Batch Iterations (N_i). (1 = 3 layer, 2 = 4 layer, 3 = 4 layer w/acceleration)

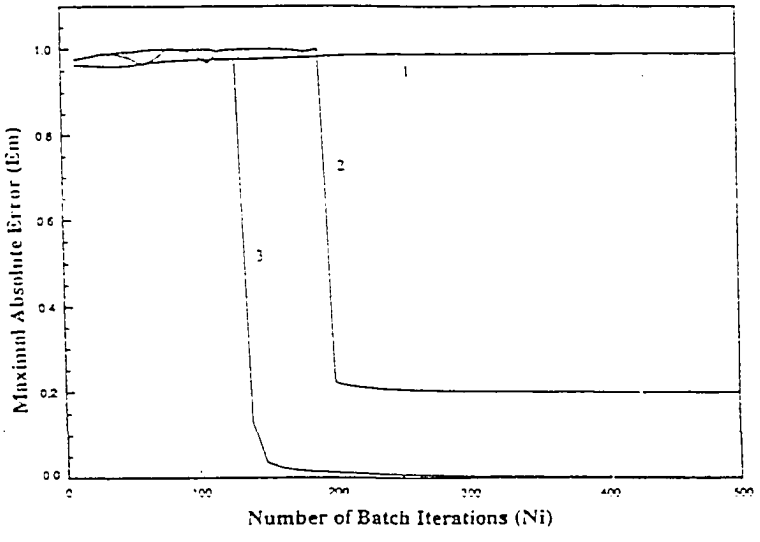


Figure 6.11: The Relationship of Maximal Absolute Error (E_m) to Batch Iterations (N_i). (1 = 3 layer, 2 = 4 layer, 3 = 4 layer w/acceleration)

of input cases in a batch learning cycle, which is 289 in this work. Figure 6.10 and Figure 6.11 show the average square error (E_a) and maximal absolute error (E_m) with respect to the batch iterations of the learning process. On these two figures, curve 1 is for the three-layer net and curves 2 and 3 are for the four-layer net. The difference between the curves 2 and 3 will be mentioned later. It is clear that the three-layer network has not converged after 500 batch iterations, E_a is around 0.1 and E_m is near 1.0, which means that at least one input case is not correctly mapped. However the four-layer network converges before 200 batch iterations.

The another important element making a notable impact on the net behavior is the learning method. The General Delta Rule is widely used in the learning process for network applications. It is a gradient descent method based on minimizing the average square error over all patterns of the training set. However, in most applications the maximal absolute error (E_m) is a more important criteria than the average square error (E_a) to judge if the correct connections are set up for the network, because a qualified network must produce a good mapping for any possible input case. This work proposed a method, referred as the acceleration technique, to improve the General Delta Rule. It was successfully used in this study. The basic idea of the acceleration technique is that the connection weights are updated only for those input cases which have large output errors. The small errors, say less than 0.001, are ignored. This makes the learning procedure concentrate on the largest output errors and results in acceleration of convergence for the learning process. On Figure 6.10 and Figure 6.11, curve 2 is the result of using the General Delta Rule without the acceleration technique and curve 3 is the result with the acceleration technique. Curve 2, after about 200 batch iterations, converges at a local minimum point with E_a of

about 0.015 and E_m of about 0.2. Curve 3, after about 130 batch iterations, converges at the general minimum point with E_a of less than 0.001 and E_m of less than 0.01. The momentum technique [Lippmann, R.P. 1987] is also a method to improve the General Delta Rule performance, but it did not help for the cases in this study.

Experiment

The proposed ANN was used in experiments to recognize the delay time of the air flow test loop. The outdoor air was heated by the hot water coil in the supply section as the heating mode, and then cooled by the chilled water coil in the load section as the cooling mode. For different operating modes, the equipment presents different delay times which are then determined by the ANN.

The control signal, which is the controller's output, and the return air temperature, which is the system response, were normalized to values between 0.0 and 1.0. These two signals were then sent to the ANN which outputs a series of meeting factors for different values of the number of samples, k, between the time when the control signal is output and the time when the system response is detected. As mentioned above, the largest average meeting factor indicates the correct delay time.

Figure 6.12 presents a set of experimental data for the heating mode and Figure 6.13 presents a set of data for the cooling mode. The horizontal axis indicates the time ($t = N * T_c$) and the vertical axis indicates the normalized control signals (u(t)) and the normalized return air temperatures (y(t)). Using these data with the ANN, we get curve 1 in Figure 6.14 for the heating mode data and another curve 1 in Figure 6.15 for the cooling mode data. Figure 6.14 indicates that the control action

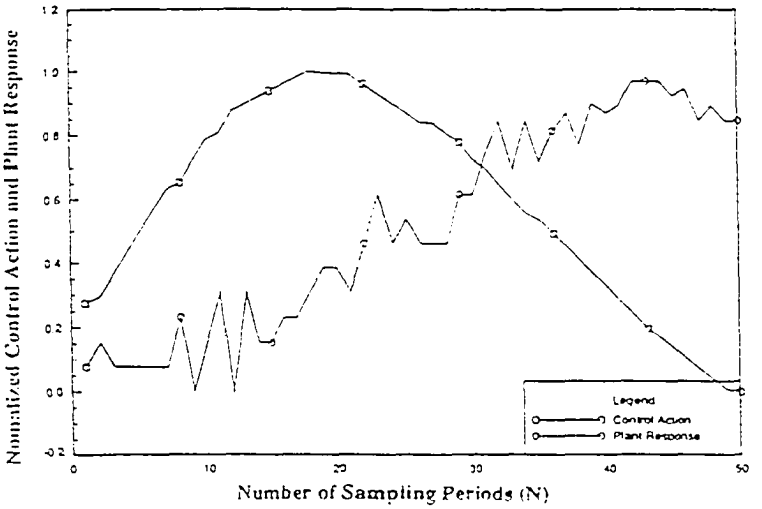


Figure 6.12: The Control Action and Plant Response Data for Heating Mode.

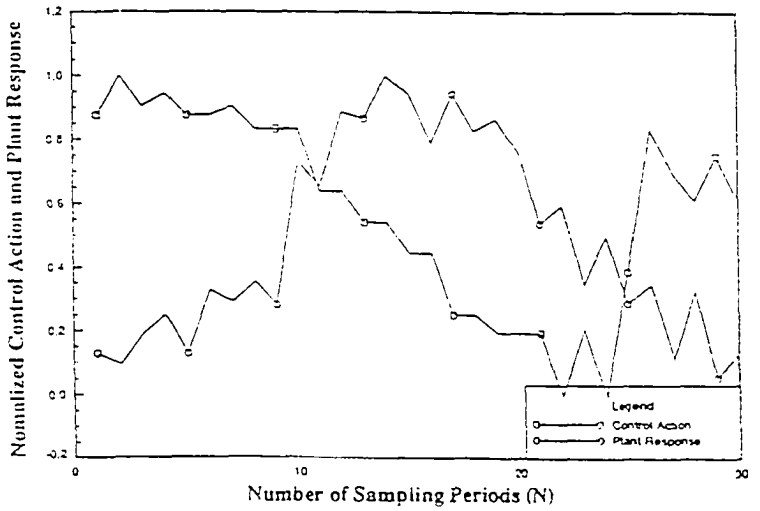


Figure 6.13: The Control Action and Plant Response Data for Cooling Mode.

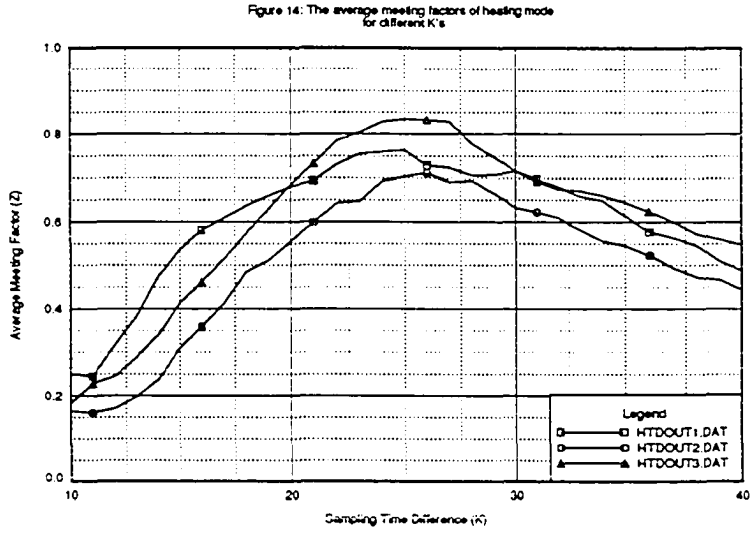


Figure 6.14: The Average Meeting Factors of Heating Mode for Different k's.

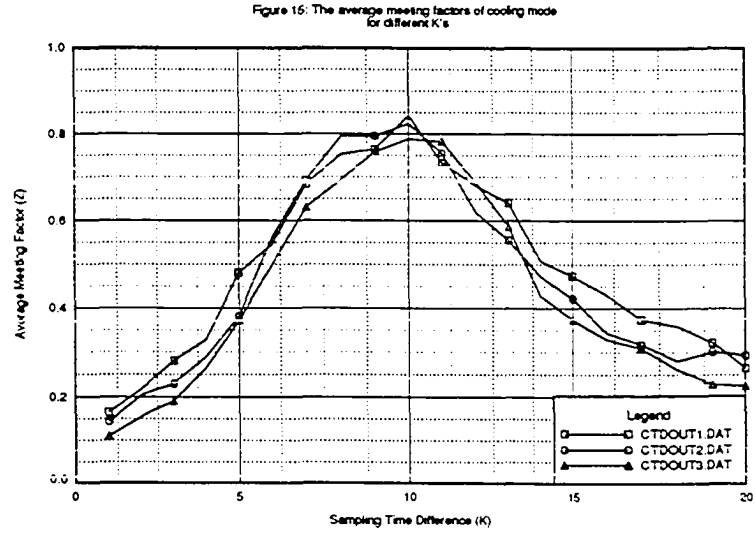


Figure 6.15: The Average Meeting Factors of Cooling Mode for Different k's.

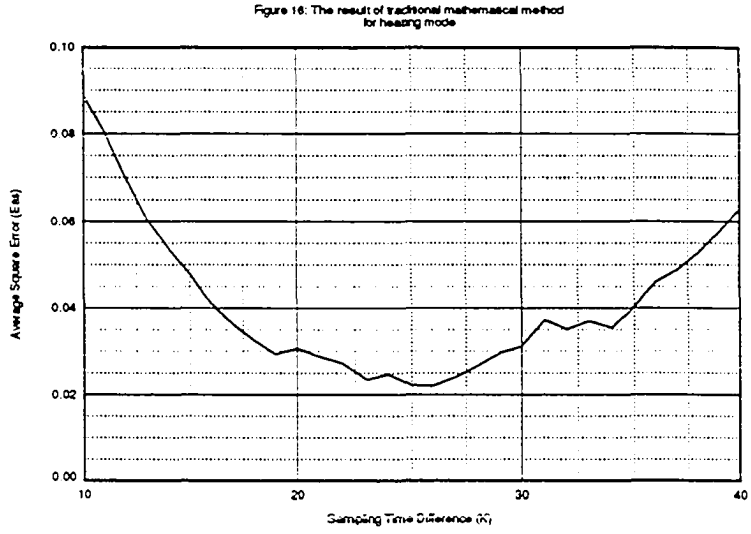


Figure 6.16: The Result of Traditional Mathematical Method for Heating Mode.

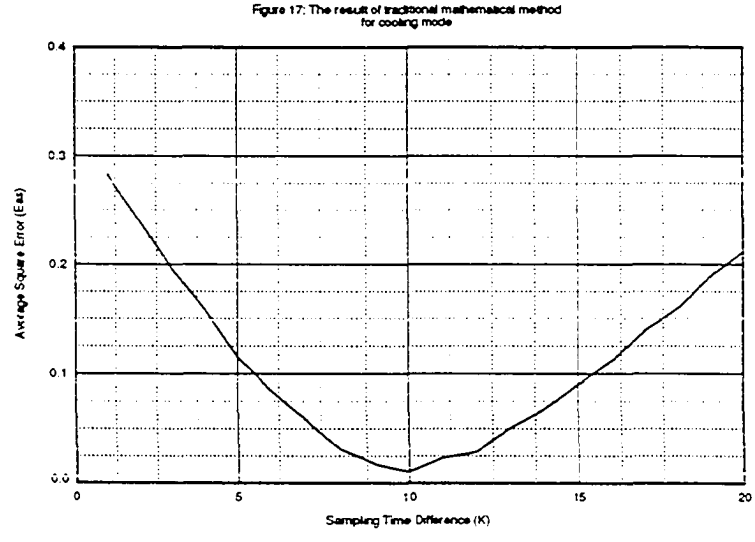


Figure 6.17: The Result of Traditional Mathematical Method for Cooling Mode.

and the plant response have the largest average meeting factor of 0.76 at k=25. The sampling period is 3.41 seconds, so the delay time is 85.2 seconds for the heating mode. Figure 6.15 indicates that the largest average meeting factor is 0.84 at k=10, then the delay time is 34.1 seconds for cooling mode. Curves 2 and 3 in Figures 6.14 and 6.15 come from two other sets of data and they indicate the same results as curve 1.

The experimental results using the ANN are compared with results using a traditional mathematical method. The average square error (E_{as}) between the normalized control action and the normalized plant response can be expressed as:

$$E_{as}(k) = \frac{1}{N_i} \sum_{i=1}^{N_i} (u(iT_c) - y([k+i]T_c))^2 \tag{30}$$

The minimum $E_{as}(k^*)$ indicates that the delay time is:

$$T_d = k^* * T_c \tag{31}$$

Figure 6.16 gives the average square error for different k's ranging from 10 to 40 for a heating mode case using the same experimental data as in Figure 6.12. The minimum average square error is reached for k=26 which is very close to the ANN result of k=25. Figure 6.17 gives the average square error for different k's ranging from 1 to 20 for a cooling mode case using the same experimental data as in Figure 6.13. In this case, the minimum average square error is reached for k=10 which is the same as the ANN result.

CHAPTER 7. FUZZY MODEL IDENTIFICATION

The model identification problem usually involves both structure identification and parameter estimation. A real single-input-single-output (SISO) system can always be described by a high-order differential equation:

$$C_0 y(t) + C_1 y^{'}(t) + C_2 y^{''}(t) + ... + C_n y^{(n)}(t) = u(t - T_d) \tag{32}$$

Obviously, the structure identification for the SISO system is to determine the highest order n, and the parameter estimation is to find the coefficients of the left-hand side terms in the equation and the delay time T_d. The delay time was estimated in the last chapter by using a neural network. In this chapter, the work is focused on how to determine the dynamical order and how to estimate the coefficients of the differential equation. For this study, the delay time is assumed to be zero for simplicity of analysis.

Structure Identification

Using Laplace transformations, we can get the system transfer function:

$$R(s) = \frac{1}{C_0 + C_1 s + C_2 s^2 + ... + C_n s^n} \tag{33}$$

where time delay has been assumed to be zero. Once the denominator polynomial has been factored, R(s) can be written as:

$$R(s) = \sum_{k=1}^{n} \frac{a_i}{s + p_i} \tag{34}$$

where a_i is the residue of the pole at $s = -p_i$. According to classical control theory [Ogata 1970], the relative magnitudes of the residues determine the relative importance of the components in the expanded form of R(s). If the residue at a pole is small, then the coefficient of the transient response term corresponding to this pole becomes small. So terms in the expanded form of R(s) having small residues contribute little to the transient response, and these terms may be neglected. If this is done, the higher-order system may be approximated by a lower-order one.

The poles of R(s) includes real poles and pairs of complex-conjugate poles. A pair of complex-conjugate poles yields a second-order term in s. Classical control theory states that there is a pair of dominant complex-conjugate poles in most real systems, and these poles play the most important role in the transient response behavior. Therefore, in most applications, equation (32) is then simplified into a second-order differential equation:

$$C_0 y(t) + C_1 y'(t) + C_2 y''(t) = u(t) \tag{35}$$

System Property Analysis

As mentioned in Chapter 4, different initial rule sets have to be set up to adapt the systems with different properties. The characteristic analysis of transient response for a controlled system is the basis for designing a fuzzy logic controller.

Table 7.1: The Relationship Between ω_n, ς, and C's

C_2		Large		Small	
C_1		Large	Small	Large	Small
C_0	Large	Below normal (4)	Normal (3)	Above normal (2)	Fast (1)
	Small	Slow (5)	Below normal (4)	Normal (3)	Above normal (2)

The transfer function of a system described by a second-order differential equation (35) is:

$$R(s) = \frac{1}{C_2 s^2 + C_1 s + C_0} = \frac{k}{s^2 + 2\varsigma w_n s + w_n^2} \tag{36}$$

where w_n is the undamped natural frequency and ς is the damping ratio:

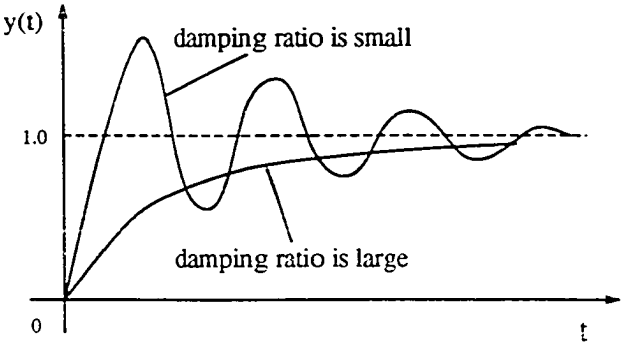
$$w_n^2 = \frac{C_0}{C_2} \tag{37}$$

$$\varsigma = \frac{C_1}{2\sqrt[2]{C_0 C_2}} \tag{38}$$

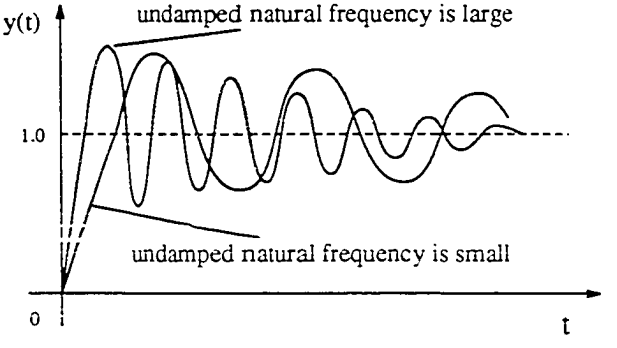
and k is a constant proportional coefficient:

$$k = \frac{1}{C_2} \tag{39}$$

Figure 7.1 shows the different characteristics of systems with different undamped natural frequencies w_n's and damping ratios ς's responding a typical step input. Obviously, the small ς or large w_n result in the fast response, and the large ς or small w_n result in the slow response. The relationship between undamped natural frequency w_n, damping ratio ς, and the coefficients of the second-order differential equation (23), C_0, C_1, and C_2 is listed in Table 7.1. The table classifies the controlled



(a) System transient response with different damping ratios



(b) System transient response with different undamped natural frequencies

Figure 7.1: System Transient Response to a Typical Step Input

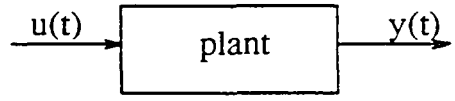


Figure 7.2: The Signal Flow Chart of a Plant

systems into 5 levels according their response characteristics to a step input: fast response (level 1), above normal response (level 2), normal response (level 3), below normal response (level 4), and slow response (level 5). A system described by a second-order differential equation with small C_2, small C_1, and large C_0 will response fast, and a system with large C_2, large C_1, and small C_0 will response slowly. These results can be visualized if the second-order system is seen as a spring-mass-dashpot system where C_0 represents a spring constant, C_1 represents a viscous-friction, and C_2 represents a mass.

Parameter Estimation

In fuzzy model identification, the model parameters are qualitatively estimated. The coefficients C_0, C_1, and C_2 have linguistic values, "Large" or "Small", not the conventional numeral values. Parameter estimation can be done by using experimental data from a plant. Figure 7.2 shows an input-output signal flow chart for a plant. A data acquisition system receives the control signal, u(t), which is the input to the plant, and the response signal of the plant, y(t), once in every sampling period. This results in a series of input signals:

$$u(t_0),\ u(t_1),\ u(t_2),\ ...,\ u(t_k),\ ...$$

and a series of output signals:

$$y(t_0),\ y(t_1),\ y(t_2),\ ....,\ y(t_k),\ ...$$

where the sampling period is:

$$dt = t_i - t_{i-1} \qquad\qquad i = 1, 2, ..., k, ... \tag{40}$$

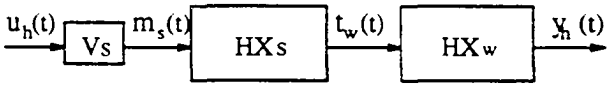
Using the central difference representation of the second-order differential equation (35), a system of equations is then constructed at the five serial sampling points, t_{i-1}, t_i, t_{i+1}, t_{i+2}, and t_{i+3}:

$$\begin{cases} C_2\frac{y(t_{i+1})-2y(t_i)+y(t_{i-1})}{dt^2} + C_1\frac{y(t_{i+1})-y(t_{i-1})}{2dt} + C_0 y(t_i) = u(t_i) \\ C_2\frac{y(t_{i+2})-2y(t_{i+1})+y(t_i)}{dt^2} + C_1\frac{y(t_{i+2})-y(t_i)}{2dt} + C_0 y(t_{i+1}) = u(t_{i+1}) \\ C_2\frac{y(t_{i+3})-2y(t_{i+2})+y(t_{i+1})}{dt^2} + C_1\frac{y(t_{i+3})-y(t_{i+1})}{2dt} + C_0 y(t_{i+2}) = u(t_{i+2}) \end{cases} \tag{41}$$

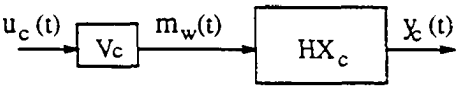
The coefficients of the second-order differential equation, C_0, C_1, and C_2, can be calculated by solving this system of equations. However, in real applications, the series of signals, u(t)'s and y(t)'s are always noisy. The relationship between u(t)'s and y(t)'s is correct only on a longer-term statistical basis, but not for every input-output pair. In this study, fuzzy set theory was used to determine the linguistic values for C_0, C_1, and C_2. An experiment was performed to test this theory.

Experiment

The same air flow test loop that was used for the previous experiments was used to study fuzzy model identification. The signal flow diagrams for both heating and cooling modes are shown in Figure 7.3.



(a) Heating mode



(b) Cooling mode

Figure 7.3: The Signal Flow Chart of Both Heating and Cooling Modes

Heating Mode

The Figure 7.3(a) shows that, for the heating mode, the steam inlet valve (V_s) is adjusted by the control signal (u_h) to change the steam mass flow rate (m_s) which affects the temperature of the hot water (t_w) in the steam-water heat exchanger (HX_s). The hot water circulating in the hot water coil (HX_w) and the hot water temperature affect the air flow temperature (y_h) through the hot water coil. Thus the output signal, the air flow temperature (y_h), is controlled by the input signal, (u_h). The signal flow diagram indicates that two heat exchanges, steam-water heat exchanger (HX_s) and water-air heat exchange (HX_w) are serially connected. Therefore there is a large lag time for the output signal, $y_h(t)$, to respond to the input signal, $u_h(t)$.

The fan speed was fixed to keep the air flow rate at about 1000 cfm and the hot water mass flow rate was kept at about 0.3 kg/s in this experiment. 25 data sets were acquired and every data set includes 34 input-output pair of $u_h(t)$ and $y_h(t)$. With these original data, 30 triples of C_0, C_1, and C_2, were calculated by solving

equation (41). Eliminating 5 larger values and 5 smaller values, the averages of the remaining 20 values were then used for C_0, C_1, and C_2. The 25 average values are listed in Table 7.2. The average C_0s range from 0.14 to 0.20 (volt/deg. C); the average C_1s range from 30 to 42 (volt-sec/deg C), and the average C_2s range from 43 to 63 (volt-sec^2/deg C).

Cooling Mode

For the cooling mode (See Figure 7.3(b)), a three-way pneumatic valve (V_c) is adjusted by the control signal (u_c) to change the bypass proportion resulting in a change of the chilled water mass flow rate (m_w) in the chilled water coil (HX_c). The change of the chilled water flow rate will affect the temperature of air (y_c) passing through the chilled water coil. The signal flow diagram indicates that the control input signal (u_c) directly controls the output signal (y_c). Compared with the heating mode, it is expected that the response of $y_c(t)$ to the $u_c(t)$ would be quicker for the cooling mode.

25 data sets were acquired with every data set including 34 input-output pairs of $u_h(t)$ and $y_h(t)$. Using the same procedure as with the heating mode, C_0 ranges from 0.15 to 0.21 (volt/deg C), C_1 from 5.5 to 11.5 (volt-sec/deg C), and C_2 from 6.0 to 18.0 (volt-sec^2/deg C). The experimental data for the cooling mode is also listed in the Table 7.2.

The values of the coefficients C_0, C_1, and C_2 for both heating and cooling modes have the shape of a normal probability distribution. The C_0 of heating mode has the distribution with mean m=0.17 and variance σ=0.02, the C_1 of heating mode has the

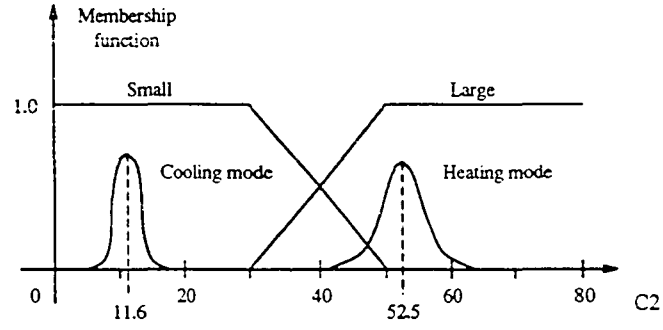
Table 7.2: The Experimental C_0, C_1, and C_2 for Heating and Cooling Modes

N	Heating Mode			Cooling Mode		
	C_2	C_1	C_0	C_2	C_1	C_0
1	55.1	35.9	0.159	7.33	9.18	0.183
2	51.0	32.6	0.178	13.14	8.13	0.179
3	60.2	40.3	0.143	11.41	10.86	0.195
4	42.9	35.8	0.192	11.87	8.47	0.168
5	59.1	39.1	0.140	16.22	9.85	0.149
6	49.7	34.8	0.180	12.66	7.24	0.152
7	43.5	31.5	0.201	11.02	8.01	0.173
8	54.6	39.3	0.175	14.36	5.51	0.186
9	61.3	30.1	0.183	8.89	5.91	0.180
10	47.6	35.9	0.170	6.02	8.42	0.207
11	57.2	30.8	0.160	17.35	6.14	0.193
12	52.4	35.3	0.173	10.88	7.12	0.181
13	53.2	40.0	0.175	6.91	8.51	0.186
14	58.4	37.3	0.165	11.62	9.18	0.190
15	52.0	38.1	0.156	10.54	7.76	0.191
16	50.9	35.7	0.201	9.12	6.22	0.169
17	45.3	41.2	0.152	11.90	8.59	0.178
18	56.0	38.4	0.168	15.17	7.87	0.158
19	62.8	36.7	0.194	11.93	6.12	0.177
20	45.1	35.2	0.162	17.98	8.73	0.212
21	55.1	32.7	0.181	8.41	6.43	0.182
22	46.8	41.8	0.161	12.38	11.52	0.184
23	53.8	31.5	0.174	16.78	9.56	0.180
24	44.1	35.5	0.158	11.22	8.55	0.174
25	57.3	33.2	0.198	10.25	10.33	0.201

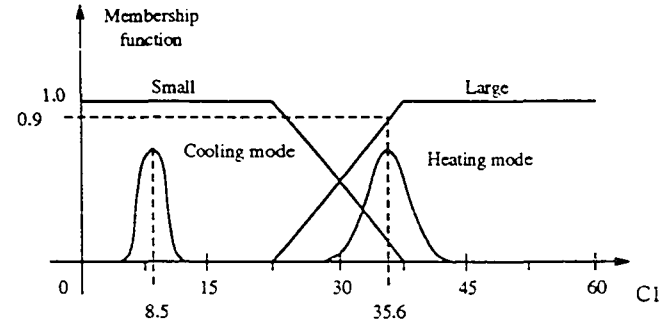
distribution with mean m=35.6 and variance σ=3.0, and the C_2 has the distribution with mean m=52.5 and variance σ=6.2. For the cooling mode, C_0, C_1, and C_2 have the means of distribution of 0.18, 8.5, and 11.6 and the variances of 0.02, 2.1, and 2.9 respectively.

Figure 7.4 shows the membership functions of the linguistic values, "Large" and "Small", for the coefficients C_0, C_1, and C_2 in both heating and cooling mode. The figure also indicates the probability distributions of their real values. Note that the membership functions of a linguistic value indicate the possibility of every element belonging to the linguistic value, mentioned in Chapter 2. Determining the membership functions and their distribution and shape depends on the applications. In most case, this is done by a "rule of thumb" method. For this study, the membership functions of the fuzzy subsets "Large"and "Small" are chosen as the shapes shown in the figure.

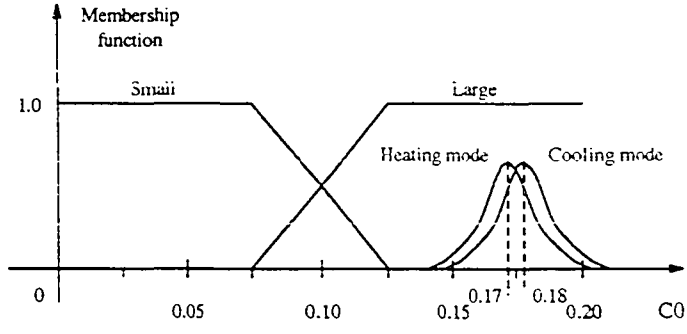
In the Figure 7.4(a), C_2's probability distribution for heating mode is wholly located in the area of the linguistic value "Large" at its mean value 52.6. This means that the membership function of C_2 for heating mode in "Large" is 1.0. For the cooling mode, C_2's probability distribution is wholly located in the area of the linguistic value "Small" at its mean value 11.6, and the membership function of C_2 for cooling mode in "Small" is 1.0. Figure 7.4(b) shows that the membership function of C_1 for heating mode in "Large" is 0.9 at its mean value of 35.6 and the membership function of C_1 for cooling mode in "Small" is 1.0 at its mean value of 8.5. In the same way, Figure7.4(c) shows that C_0 has a membership function of 1.0 in "Large" for both heating mode and cooling modes. According the Table 7.1, C_0, C_1, and C_2 all are large for the heating mode, which results a slower response characteristic (level 4).



(a)The membership function and probability distribution of C_2



(b) The membership function and probability distribution of C_1



(c) The membership function and probability distribution of C_0

Figure 7.4: The Membership Functions and Probability Distributions of C_2, C_1, and C_0 for Both Heating Mode and Cooling Mode.

For the cooling mode, C_0 is large and C_1 and C_2 are small, which results a fast response characteristic (level 1). For the two different modes, different original rule sets should be set up for the adjustment procedure of a fuzzy logic control system.

Conclusion

The model identification developed in this chapter is a linguistic identification method where the parameters have linguistic values. Two numerical examples (cooling mode and heating mode) have shown that the proposed method can result in fuzzy models. Based on the fuzzy model identification, an initial control rules are set up for self-tuning procedure discussed in next two chapters.

CHAPTER 8. SELF-TUNING STRATEGY

Introduction

Fuzzy logic controllers are based on a set of fuzzy control rules which make use of people's common sense and experience. But it is often difficult to obtain adequate fuzzy rules, especially when complicated dynamic processes are concerned. The initial rule set is always rough and inaccurate. It has to be refined using rule development and adjustment strategies introduced in Chapter 4. Typically, the fuzzy controller designer has to laboriously modify fuzzy rules by a "trial and error" method to achieve the performance improvement. To avoid this difficulty, Self-Tuning Fuzzy Logic Controllers (STFLC) have been developed. Their control actions will improve as they adjust to the controlled process and the environment. These controllers can also be called Adaptive Fuzzy Logic Controllers (AFLC) or Self-Learning Fuzzy Logic Controllers (SLFLC). The basic aspect of a STFLC is that its operation relies on past experience, i.e. a suitable combination of control strategies (control rules, membership functions, and scale factors) and the effects they produce. A particular feature of STFLCs is that they improve their performance until they converge to a predetermined optimal condition.

The first STFLC suggested evaluating and modifying the control rules by a self-organizing algorithm [Mamdani 1979]. Several other types of STFLCs have been

proposed by researchers [Xu 1987, Shao 1988, Tansheit 1988, Acosta 1992, and Lee 1992]. Most of these STFLCs use the tracking error (e) and derivative error (d) at every sampling instant as the basis not only for the control algorithm but also for the self-tuning algorithm. Thus, an interesting problem arises: the control action output is created and improved at the same stage. The control action is improved as it is created, without feedback information. The control output usually becomes bang-bang signals. Furthermore, fast convergence of the modifying process can not be obtained since it does not consider how to best modify the control rules. As an alternative, the values of the input scaling factors and output gain for FLCs are modified, instead of directly modifying the control rules.

In this study, a novel STFLC is developed and experimentally verified. A desired performance trajectory is used as a control model. Performance measurements are analyzed in the linguistic plane and modifications are performed using a look-up-table. The deviation of the actual performance trajectory of the FLC from the desired performance trajectory is used to adjust the control rules and other FLC parameters. Using fuzzy model identification methods discussed in the last chapter, an initial rule set is set up for a controlled plant with assumed properties (quick response or slow response). Then the STFLC will improve itself to obtain optimal behavior. The proposed STFLC makes sense and overcomes shortcomings of the previous STFLCs.

Performance Measure

In general, two performance measures are used to evaluate control systems: a global criterion which measures the overall performance and a local one which measures the performance over a small set of system states. Most FLC researchers prefer

to use the local criterion which makes the problem of assigning values to individual control outputs easier. It is hoped that an improvement in the local performance will also improve the global criterion. The difficulty with using a global criterion like integral square error is in choosing an appropriate figure of merit and relating a change in this figure of merit to a set of control action outputs that caused it. In the applications with local criteria, the tracking error (e) and the derivative error (d) are not only used to create the control action output but also function as a basis of improvement for the control action.

As discussed in Chapter 4, the behavior of a FLC with a set of control rules can be analyzed in a linguistic plane with a performance trajectory. For heating mode temperature control, the performance trajectory starts from a disturbed condition at point A and finally reaches the controlled condition at point O, at the center of the linguistic plane (See Figures 4.1, 4.2, and 4.3). A performance trajectory with a spiral shape is not good since it represents oscillation and overshoot in the dynamic process. There is an optimal performance trajectory shown in Figure 8.1 with curves from point A to point O in the right-bottom part of the linguistic plane for the heating mode and from point A' to point O in the left-up part for the cooling mode. Point A indicates a state of the control system with a large positive tracking error (e) and no derivative error (d) when a positive step input is imposed on the control system for the heating mode. Point A' indicates the state of the control system when a negative step input is imposed for the cooling mode. Here the tracking error (e) is defined to equal the set point temperature minus the actual temperature. Point O represents a desired stable state where both the tracking error and derivative error are zero.

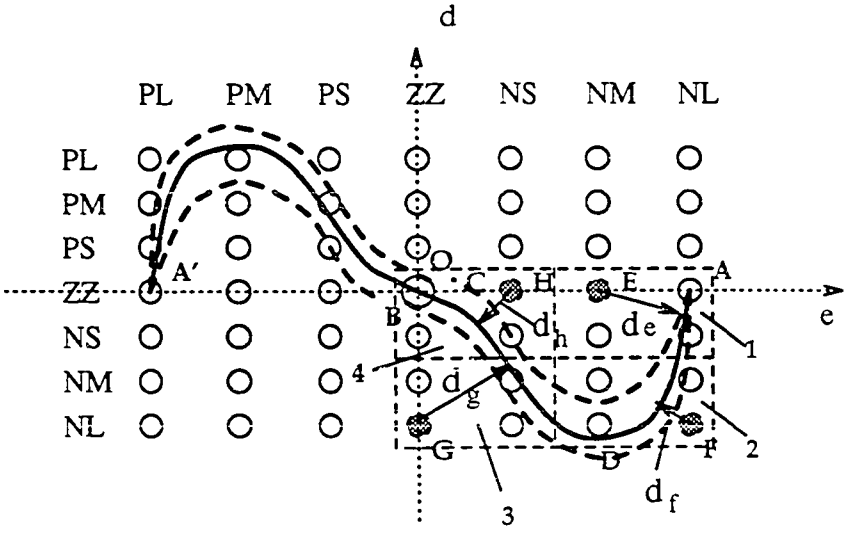


Figure 8.1: The Optimal Performance Trajectory Expressed on Linguistic Plane

The self-tuning strategy of a STFLC is now analyzed in the linguistic plane and its performance in relation to the dynamic process is measured by the deviation of the actual response from the desired one. A dashed curve AB in Figure 8.1 indicates a performance trajectory deviating a little bit from the desired path such that it will enter into the left-half part of the linguistic plane resulting oscillation and overshoot. The dashed curve AC indicates a performance trajectory deviating to the other side of the desired path. The key objectives of the self-tuning strategy proposed in this study are to measure the deviation of the performance trajectory from the desired one and to improve the relative rules or scale factors to eliminate the deviation.

In real applications, the desired performance trajectory is established based on operating experience and knowledge about fuzzy logic systems. In this work, the desired trajectory for the heating mode, shown in Figure 8.1 with AO, goes from

Table 8.1: The performance measurement strategies for self-tuning FLCs

Stage	Reference	Criterion	smaller value	larger value
1	E	d_e	slow response	possible overshoot
2	F	d_f	possible overshoot	slow response
3	G	d_g	overshoot	static error
4	H	d_h	static error	overshoot

point A to point D where the derivative error (d) has a large value which results the controlled parameter (T) responding quickly. Then the system approaches the stable point O along a smooth path.

The whole performance trajectory can be divided into 4 stages located in 4 areas marked 1, 2, 3, and 4. In area 1, point E is chosen as a reference point and the performance is then measured by the distance d_e of point E from the trajectory. A smaller d_e results in a slower dynamic response while a larger d_e results in a faster response in the first stage. In area 2, point F is chosen as a reference point and the performance in the second stage then can be measured by the distance d_f between point F and the trajectory. In this area, a smaller d_f results in a faster dynamic response while a larger d_f results in a slower response. Very small d_f may result in overshoot in the dynamic process. Using the same analysis, points G and H are chosen as the reference points for areas 3 and 4 respectively, and d_g and d_h are used as the criterion to measure the performances of the FLC in the third and fourth stages. The performance measurement strategies are summarized in Table 8.1.

Modification Procedure

There are two types of modification strategies, modifying the control rules and modifying the look-up-table. Replacing an old control rule with a new one results in revising the fuzzy relation matrix. This involves complicated fuzzy mathematical calculations.

If an old control rule expressed as a fuzzy matrix $R_{k,old}$ is replaced by a new rule expressed as a fuzzy matrix $R_{k,new}$, the general relation matrix R_g can be revised as follows:

$$R_{g,new} = [R_{g,old} \cap \bar{R}_{k,old}] \cup R_{k,new} \tag{42}$$

where $\bar{R}_{k,old}$ is the fuzzy complementary matrix of $R_{k,old}$. The complementary operation on $R_{k,old}$ and the intersection operation with $R_{g,old}$ signify "elimination" of the obsolete rule from the general relation matrix. The union operation with $R_{k,new}$ signifies "appending" the new rule to the general relation matrix.

Modifying control rules is time-consuming and demands substantial computer storage. In this study, the work focuses on a strategy to modify the look-up-table. As introduced in Chapter 2, the universes of discourse for tracking error (e), derivative error (d), and control action (u) are all included in 13 elements connected with 7 fuzzy subsets. For a given set of control rules, the control policy is usually expressed by a control surface. As discussed in Chapter 5, Figure 5.2(a) presents the control surface for an initial set of rules and Figure 5.3(a) presents the control surface for a final set of rules. These two control surfaces are expressed in terms of fuzzy subsets, which means that the level of each cell in the control surface indicates the fuzzy value of the control action. A control surface can also be expressed in terms of the

elements of the universe of discourse. This accounts for the effects of the distribution of membership functions and the interaction of the neighboring rules. Note that the fuzzy relation matrix represents the control policy with membership functions. Each element indicates the membership degrees for an exact control action. The look-up-table is obtained from the general relation matrix through defuzification. The elements in the look-up-table represent the exact non-fuzzy control outputs for relevant tracking errors (e) and derivative errors (d).

Tables 8.2 and 8.3 present the look-up-tables for the initial rule set and final rule set for the experiment of chapter 5. Figure 8.2(a) and (b) present the control surfaces in terms of elements corresponding to those shown in Figures 5.2(a) and 5.3(a). Note that the new control surfaces are formed from the old ones using the interactions between the neighbor rules. They can be changed for different distributions of the membership function. Modifying the control rules is the modification strategy in terms of fuzzy subsets while modifying the look-up-table is the strategy in terms of elements in universe of discourse. The look-up-table is more detailed and should provide better control. A possible disadvantage is that the initial rule set may be changed beyond recognition during the modification process.

Table 8.1 indicates that the global behavior of a STFLC can be measured by the criteria d_e, d_f, d_g, and d_h. The deviations from their desired values:

$$\triangle d_i = d_i - d_{i,o} \qquad\qquad i \in [e, f, g, h] \tag{43}$$

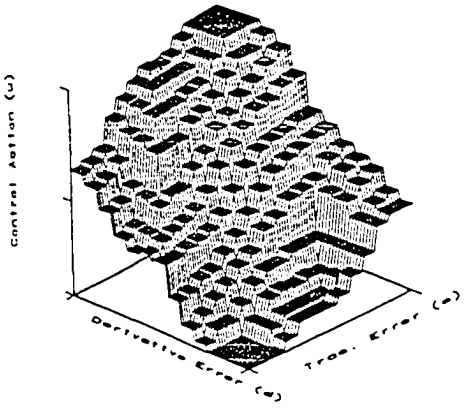
are then used to modify the look-up-table. In equation (43), $d_{e,o}$, $d_{f,o}$, $d_{g,o}$, and $d_{h,o}$ are the distances from the reference points E, F, G, and H to the desired performance trajectory respectively. Positive $\triangle d_e$ indicates possible overshoot in the dynamic

Table 8.2: The Look-Up-Table for the Initial Rule Set of Experiment in Chapter 5

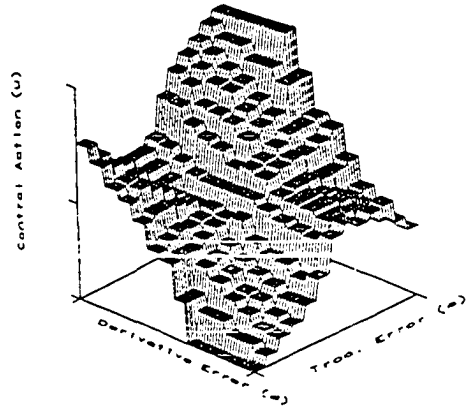
u	e												
d	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	-5.2	-5.4	-5.4	-5.1	-4.6	-3.5	-1.9	-1.7	-1.9	-1.4	-0.3	0.7	1.2
-5	-5.2	-5.4	-5.4	-4.6	-4.2	-3.4	-1.9	-1.3	-1.1	-1.1	-0.1	0.9	1.4
-4	-5.2	-5.4	-5.4	-4.6	-3.7	-3.0	-1.7	-1.1	-0.1	0.1	0.4	1.4	1.7
-3	-4.9	-4.6	-4.6	-4.0	-3.3	-2.9	-1.6	-0.9	0.2	2.1	2.4	2.6	3.0
-2	-4.3	-4.2	-3.7	-3.3	-2.7	-2.3	-1.4	-0.6	0.2	2.1	3.4	3.6	3.7
-1	-4.3	-4.2	-3.6	-2.9	-2.3	-1.8	-0.9	0.0	0.9	2.1	3.4	4.2	4.2
0	-4.3	-4.2	-3.7	-2.9	-1.4	-0.9	0.0	0.9	1.4	2.7	3.4	4.2	4.6
1	-4.0	-4.2	-3.6	-2.6	-1.1	0.0	0.9	1.8	2.3	2.9	3.6	4.2	4.5
2	-3.3	-3.6	-3.7	-2.6	-0.5	0.6	1.4	2.3	2.7	3.3	3.7	4.2	4.6
3	-2.6	-2.6	-2.6	-2.6	-0.4	0.9	1.6	2.9	3.3	4.0	4.6	4.6	5.1
4	-1.4	-1.4	-0.7	-0.6	-0.2	1.1	1.7	3.0	3.7	4.6	5.4	5.4	5.4
5	-1.1	-0.9	-0.2	0.2	0.5	1.3	1.9	3.4	4.2	4.6	5.4	5.4	5.4
6	-1.1	-0.7	0.1	0.4	0.8	1.4	1.9	3.5	4.6	5.1	5.4	5.4	5.4

Table 8.3: The Look-Up-Table for the Final Rule Set of Experiment in Chapter 5

u	e												
d	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	-5.2	-5.4	-5.4	-5.4	-5.4	-3.4	-1.7	-1.4	-0.2	0.6	1.0	1.9	2.6
-5	-4.9	-4.6	-4.6	-4.0	-4.0	-3.4	-1.6	-1.1	0.0	1.0	1.1	1.6	1.8
-4	-4.3	-4.2	-3.7	-3.3	-2.7	-2.2	-1.6	-0.9	0.0	0.6	0.6	0.6	0.8
-3	-4.0	-3.8	-3.3	-3.2	-2.5	-2.0	-1.3	-1.0	-0.2	0.2	0.1	0.5	0.8
-2	-3.3	-3.3	-2.9	-2.6	-2.2	-1.7	-1.0	-0.9	-0.4	-0.3	-0.1	0.3	0.7
-1	-2.3	-2.6	-2.1	-2.0	-1.6	-0.7	-0.9	-0.7	-0.8	-0.8	-0.1	0.3	0.7
0	-0.6	-0.9	-0.8	-0.6	-0.3	0.0	0.0	0.0	0.0	0.1	0.5	0.9	1.0
1	-0.2	-0.3	-0.1	0.0	0.1	0.7	0.9	0.7	1.3	1.5	1.9	2.6	2.7
2	-0.6	-0.3	-0.1	0.0	0.2	0.9	1.0	1.7	2.2	2.4	2.5	3.3	3.7
3	-0.3	-0.5	-0.3	-0.2	0.2	1.0	1.3	2.0	2.5	3.2	3.3	3.8	4.2
4	-0.2	-0.6	-0.6	-0.6	0.0	0.9	1.6	2.2	2.7	3.3	3.7	4.2	4.6
5	-1.3	-1.6	-1.1	-1.0	0.0	1.1	1.6	3.4	4.0	4.0	4.6	4.6	5.1
6	-2.0	-1.9	-1.5	-0.9	0.2	1.4	1.7	3.4	5.4	5.4	5.4	5.4	5.4



(a) Control surface in terms of elements corresponded to Figure 5.2(a)



(b) Control surface in terms of elements corresponded to Figure 5.3(a)

Figure 8.2: The Control Surfaces in Terms of Elements in the Universe of Discourse

response process. To avoid the possible overshoot, the control actions in the first area should be reduced. Negative Δd_e indicates slow response in the dynamic process and the control actions in this area should be increased to accelerate the dynamic process. The equation:

$$\Delta u_1 = -f_1 \Delta d_e \tag{44}$$

is used to determine the correction value for control actions in the first area. Here f_1 is a modification factor which is an adjustable parameter of the STFLC. For the second area, the equation is:

$$\Delta u_2 = f_2 \Delta d_f \tag{45}$$

Positive Δd_f indicates slow response and an increase of control actions in the area should be issued to accelerate the dynamic response. Negative Δd_f indicates a possible overshoot and a decrease of control actions should be issued to avoid this. Equations:

$$\Delta u_3 = f_3 \Delta d_g \tag{46}$$

$$\Delta u_4 = -f_4 \Delta d_h \tag{47}$$

are used to calculate the corrections for control actions in the third and fourth areas. Positive Δd_g or negative Δd_h indicates that a static error will result, then an increase of control actions in the third area or fourth area should be issued to eliminate the static error. Negative Δd_g or positive Δd_h indicates that an overshoot will result, and a decrease of control actions in the third area or fourth area should be issued.

The corrections Δu_1, Δu_2, Δu_3, and Δu_4 are then added to the control actions expressed in the look-up-table. Correspondingly, the control surface in terms of the elements in universe of discourse will raise or drop by the values of the Δu's.

The algorithm for implementing the self-tuning strategy can be represented as follows:

Step 1: Run the STFLC algorithm with a look-up-table based on the initial rule set and obtain the system response a step input. Determine d_e, d_f, d_g, and d_h.

Step 2: Calculate the corrections Δu_1, Δu_2, Δu_3, and Δu_4 using equations (43-47).

Step 3: If all these Δs are smaller than a predetermined value, then the self-tuning procedure is completed. Otherwise go to Step 4.

Step 4: Modify the relevant elements in the look-up-table.

Step 5: With the refined look-up-table, run the STFLC algorithm again. Determine new values for d_e, d_f, d_g, and d_h and go to step 2.

CHAPTER 9. EXPERIMENT OF SELF-TUNING STRATEGY

Introduction

The self-tuning strategy presented in the last chapter was experimentally verified using the air flow test loop described in Chapter 5. The experiment was done for the heating mode. The range of tracking error (e) was from -5 to +5 degree C and the range of derivative error (d) was from -0.10 to +0,10 degree C per sampling period. The scale factors K_p and K_d were 1.2 and 75 and the output gain was 0.012. The membership functions were a 1.0-0.8-0.5-0.1 distribution for the input fuzzy subsets and a 1.0-0.7-0.2 distribution for the output fuzzy subset, the same as those used in the experiment described in Chapter 5.

It is indicated in Chapter 7, that the equipment has slowly-responding characteristics with long delay time (level 4 in Table 7.1) for the heating mode. The initial control rule set was chosen as Table 9.1. As mentioned earlier, the rules in the right-bottom block function for the case when the set-point temperature is higher than the current real temperature. The rules in the left-top block function in the case when the set-point temperature is lower than the current real temperature. The rules in central block mostly affect the process dynamic characters when the system is near the set-point. The rest of the rule set is not important in most cases and can be filled with the highest control output (Level 7) in the right-top corner and the

Table 9.1: The Initial Rule Set

u e / d	NL	NM	NS	ZZ	PS	PM	PL
PL	LI(7)	LI(7)	MI(6)	MI(6)	LI(7)	LI(7)	LI(7)
PM	SD(3)	MI(6)	SD(3)	MI(6)	MI(6)	LI(7)	LI(7)
PS	MD(2)	MD(2)	SI(5)	SI(5)	SI(5)	LI(7)	LI(7)
ZZ	LD(1)	LD(1)	SD(3)	NC(4)	SI(5)	LI(7)	LI(7)
NS	LD(1)	LD(1)	SD(3)	SD(3)	SD(3)	MI(6)	MI(6)
NM	LD(1)	LD(1)	MD(2)	MD(2)	SI(5)	MD(2)	SI(5)
NL	LD(1)	LD(1)	LD(1)	MD(2)	MD(2)	LD(1)	LD(1)

lowest control output (Level 1) in the left-bottom corner. The whole control rule set is symmetrical about the central point.

Optimal Mode and Criterion Measurement

A good performance trajectory for the heating mode was obtained in Chapter 5. This is used as the desired "optimal model" for the experiment to verify self-tuning strategy. Its discrete form in the linguistic plane was shown in Figure 5.3(b), and its continuous form in the e-d plane is the curve AO shown in Figure 8.1. To calculate the distances of these special points (E, F, G, and H) to the optimal trajectory, a smooth curve is needed. This can be obtained from the original experimental data collected in the experiment mentioned in Chapter 5 by using a curve fitting technique such as the least square error method. In this work, and in most real application cases for simplicity, the distances (d_{eo}, d_{fo}, d_{go}, and d_{ho}) can be calculated as the average of the distances from the reference points to the discrete experimental points

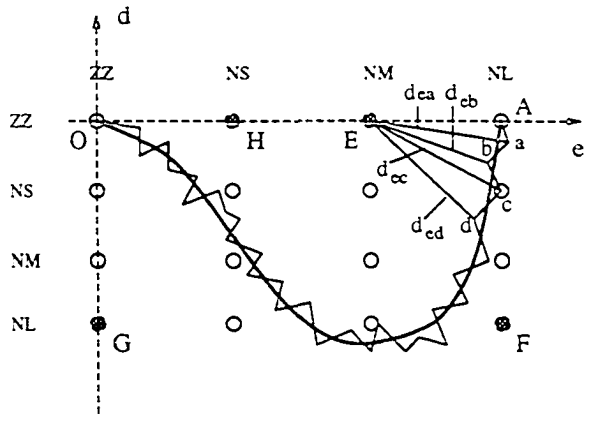


Figure 9.1: The Experimental Performance Trajectory Expressed in the e-d Plane

along the real trajectory. Figure 9.1 shows the experimental performance trajectory and equation 48 shows how to calculate d_{eo}:

$$d_{eo} = \frac{1}{5}(d_{ea} + d_{eb} + d_{ec} + d_{ed}) \tag{48}$$

The deviations, $\triangle d_e$, $\triangle d_f$, $\triangle d_g$, and $\triangle d_h$, are calculated using equation 43, and the correction values $\triangle u_i$ (i=1, 2, 3, and 4) are calculated using equations 44-47. In this experiment, the modification factors f_i (i=1, 2, 3, and 4) have the same value which was initially set to 2.4. Note that the heating mode has time delay of 85.3 seconds measured in Chapter 6 and the sample period for this experiment is 3.08 seconds. The delay time is 28 times the sample period which means that the plant response is affected by the control signal given 28 sample periods previously. This was taken into consideration for calculating the correction values $\triangle us$.

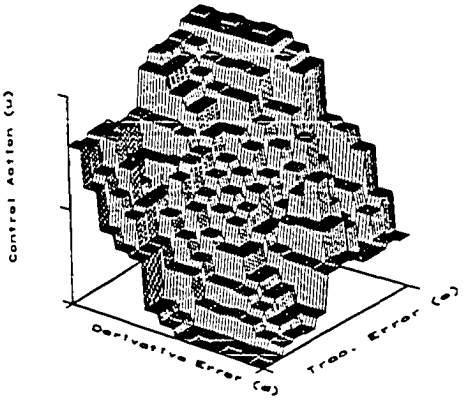
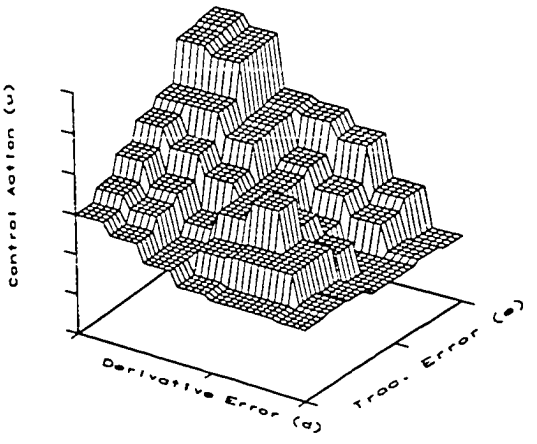


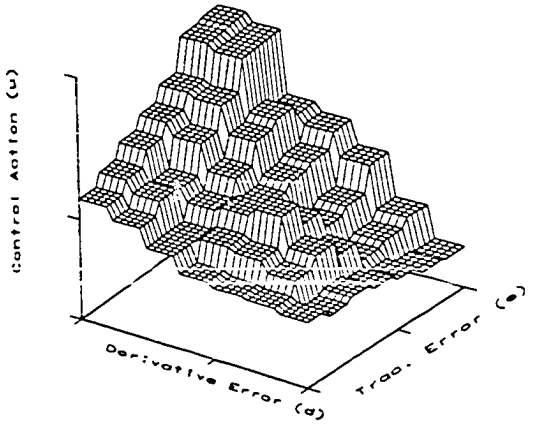
Figure 9.2: The Control Surface in Terms of Elements of the Initial Rule Set

Experiment Procedure Analysis

The initial rule set has a control surface in terms of elements in universe of discourse, shown in Figure 9.2. More clearly, the right-bottom part is shown in Figure 9.3(a). With this control policy, the control action output of the FLC is shown in Figure 9.4(a). Its dynamical response is shown in Figure 9.5 with curve 1 which has a short rising time and a large overshoot. This indicates that the performance trajectory is like a spiral curve on the linguistic plane. After the first self-tuning procedure, the look-up-table is modified and expressed using the control surface in terms of elements of the look-up-table. The control surface has the shape shown in Figure 9.3(b) where the control action output levels have slightly lower values compared with Figure 9.3(a). The actual control signals are shown in Figure 9.4(b). Compared with Figure 9.4(a), between the 40th SP and 80th SP, the average control action obviously goes down, which results decreasing the overshoot. Curve 2 in

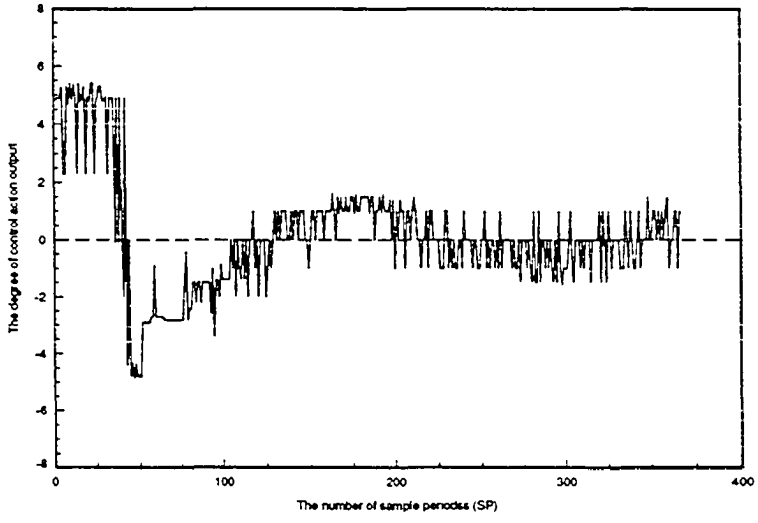


(a) The control surface of the initial rule set

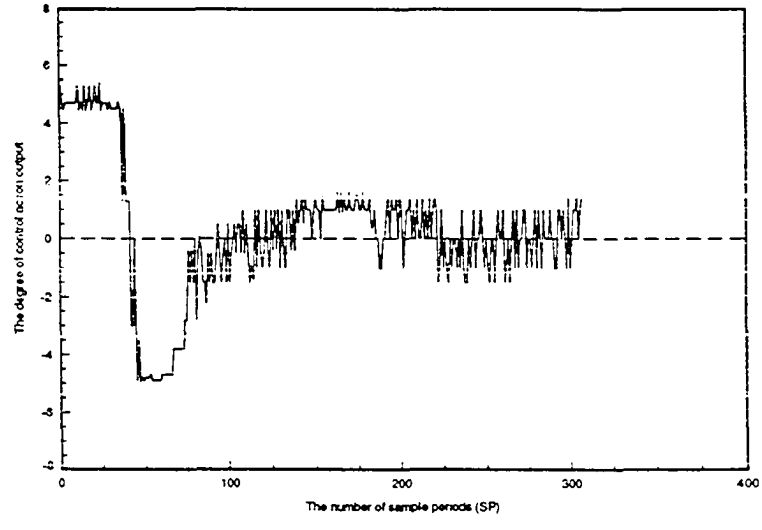


(b) The control surface after the first modification with f=2.4

Figure 9.3: The Control Surface of the Right-Bottom Part (1)



(a) The control output created by the initial rule set



(b) The control output after the first modification with f=2.4

Figure 9.4: The Control Output Signal (1)

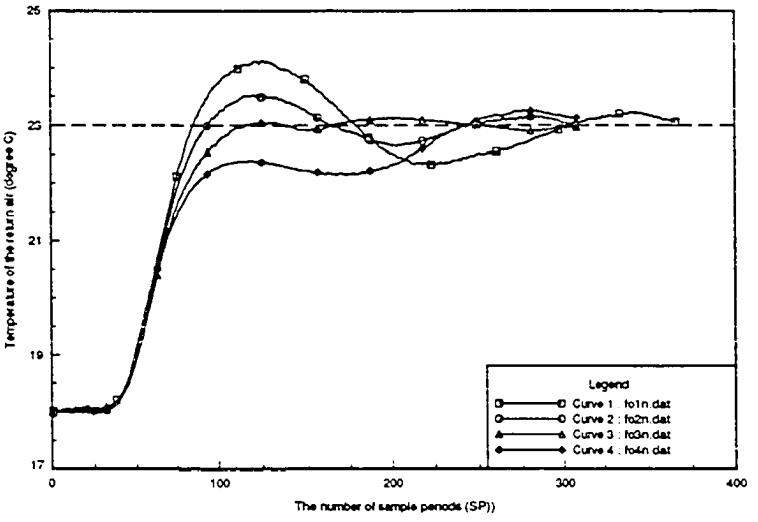
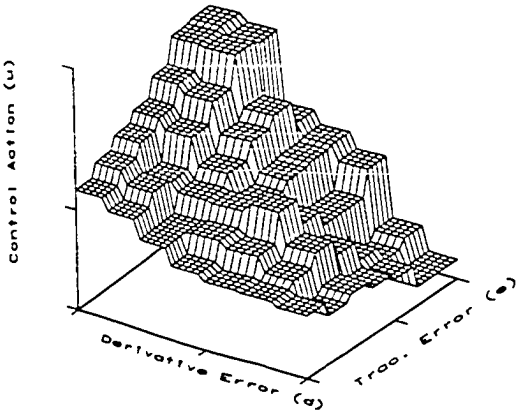


Figure 9.5: The Dynamic Process of the Self-Tuning FLC

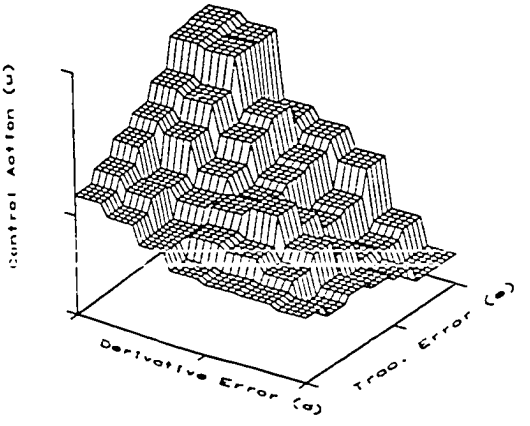
Figure 9.5 presents the dynamic process created by the modified control surface, which still has a little overshoot and some oscillation.

The modification factor, f, is an important parameter for the self-tuning algorithm. If it is too small, the convergent process of modification reaching the "optimal mode" will take a long time. If it is too large, over-damping will occur which results in a large rising time. Figure 9.6(a) shows the control surface created with f=5.0 and Figure 9.7(a) shows its control output signal. The curve 4 in Figure 9.5 shows this dynamic process. Obviously, overdoing the modifications with a large modification factor resulted in a large rising time in this case because of the control output level goes down too much.

After 3 modifications with the proper value of correction factor, f=2.4, the control

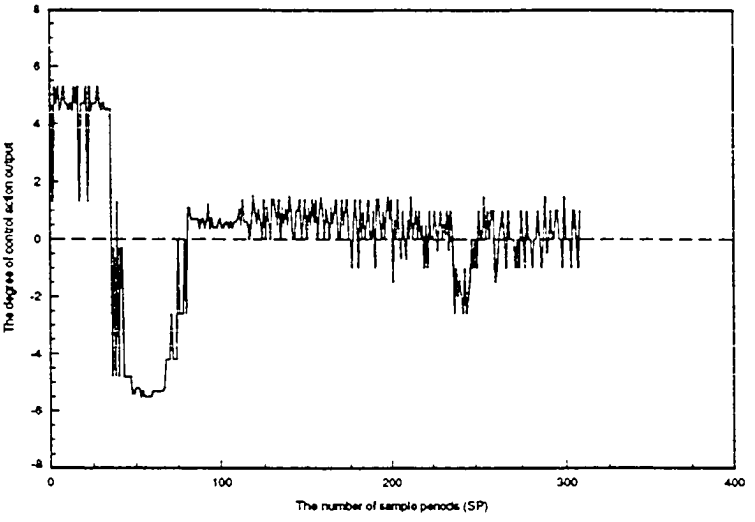


(a) The control surface after the first modification with f=5.0

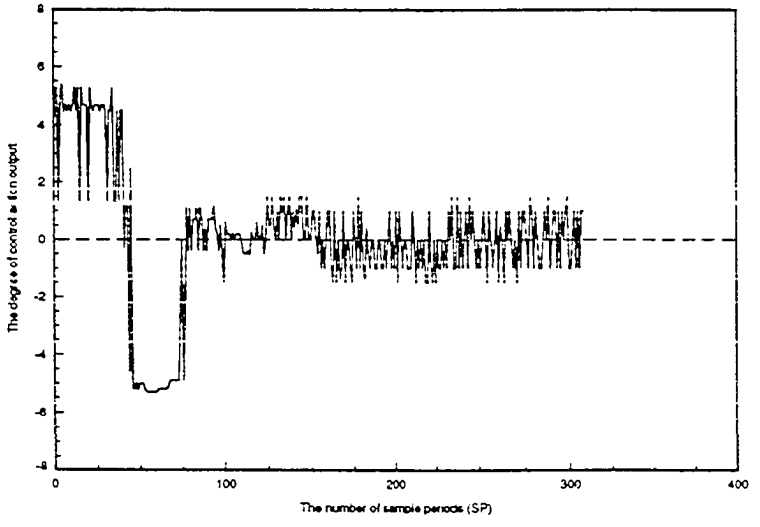


(b) The control surface after three modifications with f=2.4

Figure 9.6: The Control Surface of the Right-Bottom Part (2)



(a) The control output after the first modification with f=5.0



(b) The control output after three modifications with f=2.4

Figure 9.7: The Control Output Signal (2)

surface has the shape shown in Figure 9.6(b). Its control output signal is shown in Figure 9.7(b) and curve 3 in Figure 9.5 shows the dynamic process which has a fast rise time and almost no overshoot.

Discussion

The final dynamic process curve is very close to curve 1 in Figure 5.6 which was created by the optimal performance trajectory. This indicates that the actual performance trajectory has reached the optimal model. Note that the optimal curve quickly levels off when the controlled parameter (temperature) nears the set point (23 degree C), shown in Figure 5.6 with curve 1. But the final curve created by the self-tuning algorithm levels off at a more moderate rate, shown in Figure 9.5 with curve 3. This is because the self-tuning strategy is not as sophisticated as a human being. The general modification policy is fixed by the set-up of the self-tuning strategy. This includes choosing the reference points, dividing the correction area, and assigning values to the modification factors. These pre-determined parameters control the modification procedure and restrict its flexibility. In the modification process, the shape of the control surface doesn't change significantly. The self-tuning algorithm seems to modify the control output levels and doesn't modify the other parameters.

In this experiment, the self-tuning strategy was verified under certain conditions: constant air flow rate and constant steam pressure. Unstable air flow rates or steam pressures will severely affect the dynamic process. This is because information about air flow rates and steam pressures is not determined or used by the self-tuning

strategies. This is a limitation of this experiment.

It is interesting to compare the dynamic process characteristics of the STFLC with traditional PID controllers. When adjusting a PID controller by changing the proportional coefficient k_p, integral coefficient k_i, and derivative coefficient k_d; if the overshoot has been eliminated, there will be a long rise time (see Figure 3.1). It is possible for a STFLC to eliminate overshoot while keeping the rise time considerably shorter. Curve 1 in Figure 9.5 was improved by a self-tuning strategy to become curve 3. The overshoot was eliminated and the rise time only increased slightly. It is possible to get excellent dynamic process characteristics using STFLCs.

CHAPTER 10. CONCLUSION

Fuzzy logic control is different from the traditional control method. Traditional control method is based on the analytical control theory. Dynamical systems are analytically modeled using linear differential equations. This forms the foundation of the classical control theory. With addition of Laplace transformation and Z transformation methods, classical control theory provides a powerful means to design and analyze controllers. For some real dynamic systems, it is difficult to get precise mathematical model. Some systems are too complicated or have strong non-linear property. For these cases, We often fail to get satisfactory performance by using classical traditional control methods. We have found that the control systems governed by experienced operators often have better performance. Experienced operators can deal with complicated control problems which are difficult to be managed by the classical control theory.

The idea is how to create a new type of controller based on human experience. Fuzzy logic provides a very good technique for knowledge representation, which makes it possible to incorporate human experience with control strategies into the design of control systems. With this feature, knowledge-based fuzzy control is different from conventional mathematical-model-based controls. In recent 10 years, fuzzy logic controllers have found a wide range of applications in industry. This is because fuzzy

set theory has the ability to deal with uncertainty, fuzzy logic can be used to simulate human thinking, and fuzzy logic controllers have better behavior than traditional controllers for many cases.

The knowledge used for FLCs not only derives from expert operators, it can also come from the designer of the fuzzy logic systems based on the understanding of the dynamical behavior of the controlled plant. In most cases, human knowledge can not be described in mathematical form because it ultimately depends on humans who may change their behavior suddenly or in an unpredictable ways. However, it can be encoded in the form of rules which govern the control policy of FLCs. The multilevel relay feature of FLCs reveals why it is possible for FLCs to have good performance.

Classical control theory has been well developed and provides an effective tool for mathematical analysis and system design when a precise model is available. However, at present, there is no systematic procedure for the design of FLCs. Many researchers are engaged in the development of a theory for fuzzy dynamic systems. In this work, an experimental procedure for rule development and parameter adjustment for FLCs, based on the intuitive reasoning, has been summarized and identified.

This study has concentrated on fuzzy logic controllers from the basic aspects to an advanced self-tuning strategy. The basic concepts of fuzzy set theory, fundamental definitions of fuzzy logic, and basic structure of fuzzy logic controllers are introduced and a guideline for building the fuzzy rule-based system is developed. The rule development and adjustment strategies for fuzzy logic controllers are presented and experimentally identified. The computer simulations and laboratory experiments indicate that FLCs perform better than conventional PID controllers.

A self-tuning strategy was proposed in this work as an extension of simple FLCs to avoid the laborious task of adjusting FLCs. The actual performance trajectory was analyzed in the linguistic plane and compared with a desired model. The deviations were used to improve the look-up-table which represents the control policy.

The direction of recent research has been to incorporate neural networks into fuzzy logic systems. This combination of techniques drawn from both fuzzy logic and neural networks may provide a powerful tool for the design of control systems which can more effectively emulate human abilities. In this study, a neural network was used to determine the system delay time. Another study, which is not included in this paper, has been to use a neural network as a rule selector for FLCs suitable for different operating modes and environmental conditions.

A future topic for investigation of FLCs is to construct a fuzzy-neural system. A neural network is designed to learn the input-output relationships of FLCs and function as the fuzzy reasoning unit. This avoids complicated fuzzy mathematical calculations. This kind of fuzzy-neural controller should use less memory space and CPU time and thus be suitable for system controls with high-speed dynamics.

BIBLIOGRAPHY

Acosta, G. G. and Mayobky, M. A. 1992. Fuzzy Logic and Pattern Recognition in a Self-Tuning Controller. Proc. of the 1992 IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems, Raleigh, NC. pp. 759-765.

Cumani, S. 1982. Analysis of Fuzzy Dynamic Systems. Proc. 2nd IFSA Congress, Tokyo, Japan, pp. 143-152.

Daley, S. and Gill, K.F. 1985. The Fuzzy Logic Controller: an Alternative Design Scheme? Computers in Industry, Vol. 6, pp. 3-14.

Ding, Y. and Wong, K.V. 1990. Control of a simulated dual-temperature hydronic system using a neural network approach. ASHRAE Trans., Vol. 96, Pt. 2, pp. 727-732.

Ferrano, F.J. Jr. and Wong, K.V. 1990. Prediction of Thermal Storage Loads Using a Neural Network. ASHRAE Trans., Vol. 96, Pt. 2, pp. 723-726.

Gupta, M.M. and Tsukamoto, Y. 1980. Fuzzy Logic Controllers - a Perspective. Proc. Joint Automatic Control Conf., pp. FA10-C, August 1980, San Francisco.

Hsu, Yuan-Yih and Chen, Chao-Rong 1991. Tuning of Power System Stabilizers Using an Artificial Neural Network. IEEE Trans. on Energy Conversion, Vol. 6, No. 4 (Dec.), pp. 612-617.

Ichikawa, Yoshiaki and Sawa, Toshiyuki 1992. Neural Network Application for Direct Feedback Controllers. IEEE Trans. on Neural Networks, Vol. 3, No. 2 (March), pp. 224-231.

Kickert, W.J.M. and Mamdani, E.H. 1978. Analysis of a Fuzzy Logic Controller. Fuzzy Sets and Systems, Vol. 1, pp. 29-44.

Kiszka, J. B. 1985. Energetistic Stability of Fuzzy Dynamic Systems. IEEE Trans. Syst. Man Cybern., Vol. SMC-15, No. 5, pp. 783-792.

Kong, Seong-Gon and Kosko. Bart 1992. Adaptive Fuzzy Systems for Backing Up a Truck-and-Trailer. IEEE Trans. on Neural Networks, Vol. 3, No. 2 (March), pp. 211-223.

Kosko, B. 1992. Neural Networks and Fuzzy Systems: a Dynamical Systems Approach to Machine Intelligence. Englewood Cliffs, NJ: Prentice-Hall Inc..

Kreider, J.F. and Wang, X.A. 1991. Artificial Neural Networks Demonstration for Automated Generation of Energy Use Predictor for Commercial Buildings. ASHRAE Trans., Vol. 97, Pt. 2, pp. 775-779.

Lee, K.Y.; Cha, Y.T.; and Park, J,H. 1992. Short-Term Load Forecasting Using an Artificial Neural Network. IEEE Trans. on Power Systems, Vol. 7, No. 1 (Feb.), pp. 124-129.

Leigh, R. and Wetton, M. 1983. Thinking Clearly with Fuzzy Logic. Process Engineering, Vol. 64, pp. 36-37.

Lippmann, R.P. 1987. An Introduction to Computing with Neural Nets. IEEE ASSP Magazine, April 1987, pp. 4-22.

Mamdani, E.H. 1976. Advances in the Linguistic Synthesis of Fuzzy Controllers. Int. J. Man-Machine Studies, Vol. 8, pp. 669-678.

Mandic, N. J. 1984. Practical Application of a Heuristic Fuzzy Rule-Based Controller to The Dynamic Control of a Robot Arm. IEE Proceedings, Vol. 132, Pt. D, No. 4, pp. 190-203.

Maxwell, G.M.; Nelson, R.M.; Pate, M.B.; and Shapiro, H.N. 1986. Development of an HVAC Air Flow Test Loop. Air Movement Distribution Conference, pp. 246-252. West Lafaytte, Indiana. May 1986.

Meijer, G. 1992. Fuzzy Logic-Controlled A/Cs Heat Pumps. IEA Heat Pump Centre Newsletter, Vol. 10, No. 1, pp. 18-25.

Miller, R.C. and Seem, J.E. 1991. Comparison of Artificial Neural Networks with Traditional Methods of Predicting Return Time from Night or Weekend Setback. ASHRAE Trans., Vol. 97, Pt. 2, pp. 500-503.

Murayama, Y. and Terano, T. 1985. Optimizing Control of a Diesel Engine. Industrial Application of Fuzzy Control, M. Sugeno, Ed. Amsterdam: North-Holland, pp. 63-71.

Ogata, K. 1970. Modern Control Engineering. Englewood Cliff, NJ: Prentice-Hall Inc..

Ollero, A. and Williams, J. 1989. Direct Digital Control, Auto-Tuning, and Supervision Using Fuzzy Logic. Fuzzy Sets and Systems, Vol. 30, pp. 135-153.

Ono, H.; Ohnish, T.; and Terada, Y. 1989. Combustion Control of Refuse Incineration Plant by Fuzzy Logic. Fuzzy Sets and Systems, Vol. 32, pp. 193-206.

Park, D.C. 1991. Electric Load Forecasting Using an Artificial Neural Network. IEEE Trans. on Power Systems, Vol. 6, No. 2, pp. 442-447.

Procyk, T. J. and Mamdani, E. H. 1979. A Linguistic self-Organizing Process Controller. Automatica, Vol. 15, pp. 15-30.

Ralston, P.A. and Ward, T.L. 1985. Fuzzy Control of Industrial Process. Applications of Fuzzy Set Methodologies in Industrial Engineering, pp. 29-45. B.V., North-Holland. Elsevier Science Publishers.

Sakai, Y. and Ohkusa, K. 1985. A Fuzzy Controller in Turning Process Automation. Industrial Application of Fuzzy Control, pp. 139-151. B.V., North-Holland. Elsevier Science Publishers.

Scharf, E.M. and Mandic, N.J. 1985. The Application of a Fuzzy Controller to the Control of a Multi-Degree-of-Freedom Robot Arm. Industrial Application of Fuzzy Control, pp. 1-18. B.V., North-Holland. Elsevier Science Publishers.

Shao, S. 1987. Fuzzy Self-Organizing Controller and Its Application for Dynamic Process. Fuzzy Sets Systems, Vol. 26, pp. 151-164.

Sheridah, S.E. 1984. Automatic Kiln Control at Oregon Portland Cement Company's Durkee Plant Utilizing Fuzzy Logic. IEEE Trans. on Industry Applications, Vol. 20, pp. 562-568.

Simpson, Patrick K. 1987. Artificial Neural Systems. New York, Pergamon Press.

Tanscheit, R. and Scharf, E. M. 1988. Experiments with the Use of a Rule-Based Self-Organizing Controller for Robotics Applications. Fuzzy Sets Systems, Vol. 26, pp. 195-214.

Togai and Maski 1991. An Example of Fuzzy Logic Control. Computer Design, Vol. 30, pp. 93-103.

Tong, R. M. 1980. Some Properties of Fuzzy Feedback Systems. IEEE Transactions, Syst., Man, Cybern., Vol. SMC-10, No. 6, pp. 307-330.

Wakileh, B.A. and Gill, K.F. 1988. Use of Fuzzy Logic in Robotics. Computers in Industry, Vol. 10, pp. 35-46.

Widrow, Bernard and Lehr, Michael A. 1990. 30 Years of Adaptive Neural Networks: Perceptron, Madeline, and Backpropagation. Proceeding of The IEEE, Vol. 78, No. 9 (Sept.), pp. 1415-1441.

Xu, Chen-Wei and Lu, Yong-Zai 1987. Fuzzy Model Identification and Self-Learning for Dynamic Systems. IEEE Transactions on Systems, Man, and Cybernitics, Vol. SMC-17, No. 4, pp. 683-689.

Xu, Chen-Wei 1989. Fuzzy System Identification. IEEE Proceedings, Vol. 136, Pt. D, No. 4, pp 146-150.

Yamada, Takayiki and Yabuta, Tetsuro 1992. Neural Network Controller Using Autotuning Method for Nonlinear Function. IEEE Trans. on Neural Networks, Vol. 3, No. 4 (July), pp. 595-601.

Yasunobu, S. and Hasegawa, S. 1985. Automatic Train Operation by Predictive Fuzzy Control. Industrial Application of Fuzzy Control, M. Sugeno, Ed. Amsterdam: North-Holland, pp. 1-18.

Zadeh, L.A. 1965. Fuzzy Sets. Information and Control, Vol. 8, pp. 338-353.

Zadeh, L.A. 1968. Fuzzy Algorithm. Information and Control, Vol. 12, pp. 94-102.

APPENDIX AN EXAMPLE OF NUMERIC CALCULATION FOR INFLUENCE OF MEMBERSHIP FUNCTION

There are two rules:

R1: IF (e is PM) THEN (u is VS) and

R2: IF (e is PS) THEN (u is ST).

For the first rule,

$\tilde{e_1}$=(0, 0, 0, 0, 0, 0, 0, 0.1, 0.5, 0.8, 1.0, 0.8, 0.5)

$\tilde{u_1}$=(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.2, 0.7, 1.0)

Then the rule can be interpreted into a fuzzy reasoning matrix as follows:

$$
\begin{aligned}
R_1 &= \tilde{e_1} \otimes \tilde{u_1} \\
&= \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0.1 & 0.1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.5 & 0.5 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.7 & 0.8 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.7 & 1.0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.7 & 0.8 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.5 & 0.5
\end{pmatrix} \\
&\equiv (\mu_{R_1}(i,j))
\end{aligned}
$$

where the membership in R_1 for the element (i,j) of the matrix, $\mu_{R_1}(i,j)$, is

$$\mu_{R_1}(i,j) = \min(\mu_{\tilde{e_1}}(i), \mu_{\tilde{u_1}}(j))$$

For the second rule,

$\tilde{e_2}$=(0, 0, 0, 0, 0, 0.1, 0.5, 0.8, 1.0, 0.8, 0.5, 0.1, 0)

$\tilde{u_2}$=(0, 0, 0, 0, 0, 0, 0, 0, 0.2, 0.7, 1.0, 0.7, 0.2)

Then the second rule can be interpreted into a fuzzy reasoning matrix as follows:

$$R_2 = \tilde{e_2} \otimes \tilde{u_2}$$

$$= \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.5 & 0.5 & 0.5 & 0.2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.7 & 0.8 & 0.7 & 0.2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.7 & 1.0 & 0.7 & 0.2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.7 & 0.8 & 0.7 & 0.2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.5 & 0.5 & 0.5 & 0.2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

$$\equiv (\mu_{R_2}(i,j))$$

where the membership in R_2 for the element (i,j) of the matrix, $\mu_{R_2}(i,j)$, is

$$\mu_{R_2}(i,j) = \min(\mu_{\tilde{e_2}}(i), \mu_{\tilde{u_2}}(j))$$

The general fuzzy relation matrix R is then constructed as the union of these two rules:

$$R = R_1 \cup R_2$$

$$
= \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.5 & 0.5 & 0.5 & 0.2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.7 & 0.8 & 0.7 & 0.2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.7 & 1.0 & 0.7 & 0.5 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.7 & 0.8 & 0.7 & 0.8 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.5 & 0.5 & 0.7 & 1.0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0.1 & 0.2 & 0.7 & 0.8 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.5 & 0.5
\end{pmatrix}
$$

$$
\equiv (\mu_R(i,j))
$$

where

$$
\mu_R(i,j) = \max\left(\mu_{R_1}(i,j), \mu_{R_2}(i,j)\right)
$$

Assume that there is an input (e) and its fuzzy value $\tilde{e} = PM$, then the output is expected to be VS according to the first rule. But now the output is calculated through the fuzzy matrix R as follows:

$$
\begin{aligned}
\tilde{u} &= \tilde{e} \circ R \\
&= PM \circ R \\
&= (0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0.1,\ 0.5,\ 0.8,\ 1.0,\ 0.8,\ 0.5) \circ R \\
&= (0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0.2,\ 0.7,\ 0.8,\ 0.7,\ 1.0)
\end{aligned}
$$

where

$$
\mu_{\tilde{u}}(j) = \max_i(\min(\mu_{\tilde{e}}(i), \mu_R(i,j)))
$$

While

$$
\begin{aligned}
\tilde{u_1} &= VS \\
&= (0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0.2,\ 0.7,\ 1.0)
\end{aligned}
$$

So,

$$
\tilde{u_1} \subset \tilde{u}\ .
$$