ROBUST CONTROL WITH FUZZY LOGIC ALGORITHMS

JIAN ZHOU WANG

Ph.D

DEPARTMENT OF ELECTRICAL ENGINEERING

THE UNIVERSITY OF EDINBURGH

EDINBURGH, UK

JULY, 1997

Declaration

This thesis, composed entirely by myself, reports work carried out solely by myself in the Department of Electrical Engineering, University of Edinburgh between October 1993 and September 1996.

Acknowledgement

I am grateful to my supervisors, Prof. J.R. Jordan and Dr. I.A.B. Lindsay, for their unsparing support and guidance during the course of this work. I would also like to thank the British ORS organization and the Faculty of the Science and Engineering, University of Edinburgh, for their financial support. Mr. D. Carmichael in the Power Lab and Mr. M. Gordon in the Computing Office provided able technical support and must be thanked particularly for their assistance with the design of the *FzySimu* package.

Finally, I wish to thank my wife for her unfailing encouragement and support over the years.

Jian Zhou Wang

January, 1997.

Abbreviations and symbols

$\mu(\bullet)$	Membership function
A/D	Analogue to digital.
COG	Center-of-gravity method in defuzzification.
CY	Number of ring cycles.
D/A	Digital to analogue.
$e(t)$, $\dot{e}(t)$	Error and derivative error of feedback control systems.
FLC	Fuzzy logic controller.
$G(s)$	Transfer function.
GA	Genetic algorithm.
GE, GC, GU	Scaling factor of FLCs.
H_∞	Hyper-infinity optimal control.
IES	Integral of square of error.
I_{rb}	Robust performance index.
I/O	Input-output.
ISE	Integral of the square of error.
K	Process gain.
MEUC	Microprocessor embedded universal controller.
MIMO	Multi-input-multi-output.

MISO	Multi-input-single-output.
MOM	Mean of maximum.
N_{gp}	Number of test points in specification.
OS	Overshoot.
PAM-I	First phase-advanced method for non-integrating processes.
PAM-II	Second phase-advanced method for integrating processes.
PID	Proportional, integral, and derivative controller.
P_{index}	Performance index.
R_s	Robustness of a control system.
R_{st}	Settling time ratio.
RT	Rise time.
S_a	Achievable robust space.
SISO	Single-input-single-output.
S_r	Required robust space.
ST	Settling time.
STPP	Similar tuning point performance.
T_Σ	Sum of all time constants in the controlled process.
T_1, T_2	Time constants of controlled processes.
t_s	Sample period of controllers.
u	Control input of processes under control.
US	Undershoot.
w	Step change of the setpoint of a control system.
y	Output of a control system.

Abstract

This thesis presents the results of an investigation of the robustness of the widely used Mandani-type fuzzy logic control systems under a wide variation of parameters of the controlled process.

The measurements of the dynamic performance and system robustness of a control system were firstly defined from the engineering point of view, and the concepts of the robust space and the robustness index were introduced. The robustness of the FLC systems was investigated by analyzing the structure of the fuzzy rule base and membership functions of the input-output variables. Based on the close relation of the fuzzy rule base and the system dynamic trajectory on the phase plane, a switching line method is introduced to qualitatively analyze the dynamic performance of the SISO FLC systems. This switching line method enables the qualitative prediction of the shape and position of the robust space of the FLC controlled first and second order processes. The effects of FLC parameters on system robustness were also investigated. The movements of the position and the shape of the switching line with the variation of the controller parameters were analyzed, and its relation with the system performance was reported.

Three methods were proposed to improve the robustness of the FLC system. The first design method proposed was based on the switching line characteristic of the FLC system. The second method, called phase advanced FLC, was introduced to handle the control of high order processes with fuzzy algorithms. The third method was an evolutionary method based on a genetic algorithm which was used to automatically design a robust fuzzy control system, assuming the availability of the controlled process model.

Finally, to overcome the tuning difficulty of FLC systems, an attempt was made to develop a fuzzy tuning algorithm. It was investigated for controlling a first order process and a second order process. Based on extensive simulation experiments, the fuzzy rules for tuning scaling factors and the sample rate were collected. A complete tuning procedure was designed.

CONTENTS

1

Introduction

An initial involvement with ABB Kent-Taylor while working as a visiting scholar to Edinburgh University in 1993 led to the design and prototype implementation of a fuzzy logic controller (FLC) for process control applications. This work clearly showed the difficulty of introducing a new controller to a market that had used and become very familiar with PID control concepts. Although the deficiencies of the PID algorithm were well recognised, very little work had been done to clarify the advantages and disadvantages of FLC.

Among the advantages claimed by many researchers, the robustness of FLC systems attracted special attention. Considerable anecdotal evidence, but very little hard experimental evidence, was available to confirm the robustness of fuzzy logic control. It is clear that a solid experimental base is required to support this robustness claim.

This *Chapter* presents the motivation for the program of research described in this thesis. The basic concepts of fuzzy control methods are discussed and the concepts of robustness are explained. The objectives of the research are defined and the scope of the thesis is presented.

1.1 Problems with control engineering

Control engineering, which is based on the foundations of feedback theory and system analysis, is concerned with understanding and controlling a process to provide a desired response. The block diagram of a typical classic feedback control system is shown in Fig.1-1a. Here C denotes the controller and P_1P_2 the process. The measurement device is symbolised by P_m and noise n corrupts the measured variable y. The controller determines the process input u on the basis of the error e. The objective of the feedback loop is to keep the output y close to the reference (setpoint) w and minimise the effect of the disturbance d on the process output y. Commonly the system is simplified into that shown in Fig.1-1b where P denotes the entire controlled process and exact knowledge of the output y is assumed (*i.e.* $n = 0$ and $P_m = 1$).

For commonly used design procedure to yield a control algorithm which works satisfactorily in a real environment, the process model or information about the dynamic behaviour of the process must be specified. Also tuning methods for the control system are centred around the process model. The process model can have the form of coupled partial differential equations or be simply the process gain and the settling time experienced by the plant operator.

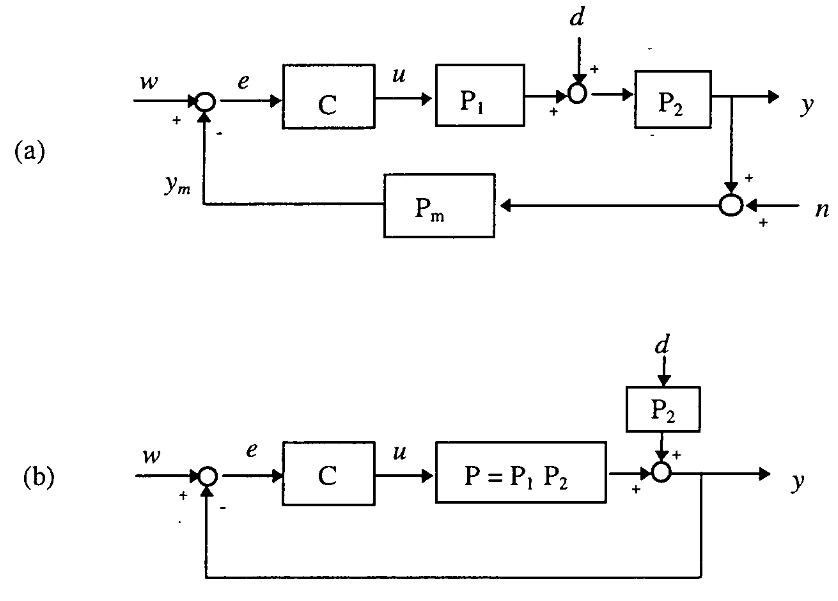


Fig. 1-1. General (a) and simplified (b) block diagram of feedback control system.

However, it is difficult to obtain an accurate process model in the real world. The accuracy of this model varies but is never perfect because of the high order, nonlinear characteristics of the practical plants. If the process model is obtained via *linearization,* then it is accurate only in the neighbourhood of the operating point chosen for the linearization. Even when the underlying process is essentially linear, the physical parameters are never known exactly and fast dynamic phenomena (e.g., valve dynamics) are usually neglected in the model.

Therefore, at high frequencies, even the model order is unknown. Moreover, the behaviour of the plant itself changes and this is rarely captured in the models. For example, increased throughput and flow rates usually result in smaller dead-time and time constants.

Neglecting these *model* uncertainties could lead to controllers which are too sensitive and which are likely to become unstable in the real operating environment. Therefore, a method for designing accurate systems in the presence of significant plant uncertainty is highly desirable, but it is a problem which is not solved in classical feedback control theory.

1.2 Robust control

There are several methods in modern control theory to cope with the uncertainties in control systems. In general, they can be classified into two types: robust control and adaptive control. The robust control method is based on the optimisation of a defined objective function of the designed system by, selecting proper controller parameters and controller structure. The system performance is entirely dependent on an understanding of the controlled process and the design of the controller. The adaptive control method is based on an on-line learning strategy to achieve the required system performance. The system performance mainly depends on the learning ability of the system and the controller used. In this thesis, the concept of robust control will be of particular interest.

A control system is *robust* when (1) it has low sensitivity to the model uncertainty, (2) it is stable over a family of process models, or (3) the performance continues to meet the specifications in the presence of a set of changes in system parameters [1] [2].

Clearly, the above definition of the system robustness implies three concepts: *low sensitivity, robust stability and robust performance*. If the effect of the process uncertainty on system output is the main concern, sensitivity analysis is applied. When the stability of a control system over a family of process models is addressed, robust stability is often analysed to investigate the design requirement. If a particular system performance is required no matter how the system changes, *robust performance* is studied.

From the point of view of practical use, robust performance is a more important characteristic than the other two. There are two reasons. The first one is that for a control system to be of practical use, stability guarantees alone are insufficient. Stable realisations may give totally inadequate performance. For a controller to have robust performance, the realisation must meet the desired performance specifications as well as the stability requirement. The second reason is that the sensitivity design objectives are different from those of robust performance design. Sensitivity looks for the dependence of closed-loop performance due to *small* parameter variations. While robust performance requires that the closed-loop system performance is acceptable for *all* possible parameter values in the prescribed range [64]. The third reason is that system performance is usually the major design requirement. Therefore, robust performance will be the focus of this research.

The design of a robust control system involves two tasks: determining the structure of the controller and adjusting the controller's parameters to give an "optimal" system performance. This design process is normally done with "assumed complete knowledge" of the plant. Furthermore, the plant is normally described by a linear time-invariant continuous model. The structure of the controller is chosen such that the system's response can meet certain performance criteria.

The most significant research activity in the robust control area is concerned with the so-called *Linear Quadratic (H_2) Optimal Control* and the *H_∞ Optimal Control* [2] methods. The H_2 optimal controller minimises the *Integral Squared Error (ISE)* for a particular input, *i.e.* the objective of H_2 optimal control is to find a controller c which meets the following requirement:

$$\min_c \int_0^\infty e^2 dt = \min_c \|e\|_2^2 \ , \tag{1-1}$$

where e is the system error. The *H_∞ Optimal Control* minimises the worst ISE which can result from a set of inputs W, which can be described as follows:

$$\min_c \sup_{w \in W} \int_0^\infty e^2 dt = \min_c \sup_{w \in W} \|e\|_2^2 = \min_c \|e\|_\infty \tag{1-2}$$

The required information for using these methods to design a robust control system is the process model, input type, performance specifications and uncertainty information. In general, the process model is in mathematical format.

There is one main challenge in using the robust control methods — the availability of a process model. In the real world, there are many plants which cannot be described mathematically because of the complexity of the plants or the high cost of measuring the model parameters. Multi-dimensional, hierarchical structures, mutual interactions, internal feedback mechanisms, and unpredictable dynamics are only part of the characteristics of such complex systems. This complexity accounts for some of the reasons for the difficulties in the attempts to apply the powerful modern system and control techniques. Such weakness of the quantitative techniques to adequately describe complex phenomena was summarised in the principle of incompatibility, formulated by L.Zadeh [3]. It states:

> *"As the complexity of a system increases, our ability to make precise and yet significant statement about its behaviour diminishes, until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics."*

Zadeh's proposal of modeling the mechanism of human thinking, with linguistic fuzzy values rather than numbers, led to the introduction of fuzziness into system theory and to the development of a new class of systems called *fuzzy control systems*.

1.3 Fuzzy algorithm

The main characteristic of fuzzy systems is that they are based on the concept of fuzzy partitioning of the system's information. Fuzzy systems operate with fuzzy sets instead of numbers. A fuzzy set has more power than a single number. The use of fuzzy sets permits using and operating on the imprecise information which is often found in the real world. With this characteristic of fuzzy systems, the problem of obtaining the mathematical model of the controlled process can be partially solved.

Moreover, fuzzy logic, a reasoning method based on fuzzy set theory, is used in fuzzy systems to operate on fuzzy variables and process imprecise information. It is much closer in spirit to human thinking and natural language. Basically, it provides an effective means of capturing the approximate, inexact nature of the real world. Thus the fuzzy controller can be used to imitate control experts or skillful operators as they control a process. It has been recognised that human operators can regulate practical plants satisfactorily based on their experience of controlling the plant. Control is maintained independent of model variations if the model belongs to the same class of plant family. With reference to this fact, therefore, the fuzzy controller should be expected to inherently possess some robustness.

However, a fuzzy control system is regarded as a nonlinear system. It is difficult to apply the analytical methods available for conventional systems to the fuzzy systems. In addition, there is no systematic method to design a fuzzy system and to analyse its performance. Thus, as in most applications of fuzzy systems, the experimental method has been used extensively in this research.

Note that the stability of the fuzzy control systems will not be studied in this work. The reason is that the main objective of this work is to investigate the robust performance of fuzzy control systems, and if system performance meets the performance specifications, the system must be stable.

1.4 Research objectives

Due to the novel linguistic aspects and the non-linearity of fuzzy systems, it is difficult to mathematically analyse the stability of a fuzzy control system. Consequently an analytical investigation of the robustness of a fuzzy system with respect to process parameter variation and the development of design procedures for a robust fuzzy logic controller has not, as yet, been satisfactorily completed. Only a few publications have reported attempts to investigate the robust capabilities of fuzzy logic controllers and give experimental evaluations of the robust performance of a FLC system. They found that in a certain range of process dynamics the conventional controller is difficult to adjust for good responses, while the fuzzy control system is less sensitive to process parameter changes and gives good control at all operating points. Moreover, many papers mentioned the shortage of analytical information on the robustness of fuzzy control systems and the difficulty of developing the systematic method for designing a robust fuzzy controller.

Therefore the following work related to the SISO systems was the focus of this research:

- Investigate the robustness of fuzzy logic control systems and compare with the widely used conventional PID (*Proportional, Integral, and Derivative controller*) control method.

- Investigate the effects of controller parameters on the system robustness.
- Investigate the practical design and tuning methods necessary to improve the robust performance of fuzzy logic control systems.

1.5 Achievements

This research program has required a full understanding of the theoretical concepts of fuzzy control systems and a wide range of system knowledge. To demonstrate the validity of the research results, system engineering experiences and computer related skills including computer simulation, software and hardware design have been employed. The major achievements of this work can be listed as follows:

- The robust space of the widely used fuzzy control system has been qualitatively analysed and experimentally tested. It is found that the robustness of the FLC system stems from its switching line feature which presents a low sensitivity of system performance to the process parameter variations.
- The effects of the FLC parameters (scaling factor, membership functions, or fuzzy rules) on the system robustness are demonstrated by both qualitative analysis and extensive simulation experiments. It is concluded that the robust space of a fuzzy control system can be optimised by changing the FLC parameters
- The robustness of the fuzzy control system has been compared with that of the widely used PID system by a detailed series of simulation experiments. Experiments performed as part of the research programme reported in this thesis have shown the advantages of fuzzy control systems over PID systems with respect to robustness.
- Design methods have been proposed for improving the robustness of fuzzy systems. These methods include the switching line method, the phase-advanced algorithms, and the genetic algorithm. Their practical value has been established.

- A fuzzy reasoning algorithm has been developed for the purpose of auto-tuning the scaling factors and the sampling rate when a FLC is used to control first order and second order processes. The fuzzy tuning rules for scaling factors and the sample rate have been generated based on extensive simulation experiments. A complete tuning procedure was designed. The adequacy of this tuning algorithm has been established.

- A simulation software package for the design and evaluation of fuzzy logic control systems has been developed under both MS-DOS and UNIX environments. The package includes system simulation, system analysis, robust space mapping, GA optimisation, and auto-tuning.

- An implementation of a fuzzy logic controller based on the Motorola 68HC11 microprocessor and a third-order process emulator has been carried out for confirmation of the simulation results. A real-time control interface between the practical FLC system and the simulation software stated above has been implemented for compiling, downloading, and debugging the program in real-time fuzzy control system applications as well as for displaying the system performance.

1.6 Scope of the thesis

Fuzzy logic algorithms for process control are reviewed in *Chapter* 2, beginning with the fundamental concepts of the fuzzy set theory. Methods for designing fuzzy control systems and studies on the robustness of fuzzy control systems are examined. The necessity for this research is discussed. In *Chapter* 3, the robustness of the widely used FLC systems has been investigated and the robust space of the FLC system is qualitatively predicted based on the switching line method. The effects of FLC parameters on system robustness are discussed and three methods are proposed to improve the robustness of a FLC system. Experimental investigations of the robustness of FLC systems are presented in *Chapter* 4 based on extensive simulations, and the feasibility of the proposed methods in analysing and designing the robustness of FLC systems is experimentally tested. The simulation software is also briefly introduced in *Chapter* 4. *Chapter* 5 introduces a novel tuning method for FLC systems based on the author's experiences. The fuzzy tuning rules for scaling factors and the sample rate are presented and the effectiveness of the tuning method is experimentally tested. Finally, the conclusions from the research reported in this thesis and recommendations for further work are presented in *Chapter* 6. The design methods and functions of the simulation package *FzySimu* are described in *Appendix A*. *Appendix B* presents the confirmation results of the computer simulations against the practical control systems.

2

Fuzzy Logic Control

2.1 Introduction

During the last decade, fuzzy logic control has emerged as one of the most active and fruitful areas for research in the application of fuzzy set theory [28], especially in the realm of industrial processes, which do not lend themselves to control by conventional methods because of a lack of quantitative data regarding the input-output relations. The pioneering research of Mamdani and his colleagues on fuzzy control [52], [78], [79], [29]-[31] was motivated by Zadeh's seminal papers on the linguistic approach and system analysis based on the fuzzy set theory [3] [26]. Since then, applications of fuzzy logic control covered numerous fields ranging from finance to earthquake engineering [80]. The notable applications include cement kiln control [81], automatic traffic operation [17], robot control [18] [19], arterial pressure control in surgery [20], chemical distillation control [21], anti-skid braking system[8], power system and nuclear reactor control [82] [83], and many consumer products [74] such as auto-focusing in cameras and camcorders, washing machines, rice-cookers and so on. Fuzzy logic microcontroller chips [22] and fuzzy computers [23] have also pointed the way to an effective utilisation of the fuzzy logic theory.

Surveying the vast literature on fuzzy systems [5] [24] [35]-[37] [69]-[83], and comparing with the traditional control methods, the fuzzy control algorithm offers some advantages and possesses certain weaknesses. They can be summarised as follows:

Advantages:

(1) It does not require a detailed mathematical model to formulate the control task.

(2) It provides an effective way to apply the expert experience and linguistic knowledge to the design of control systems.

(3) It is better able to handle nonlinear control problems.

(4) It is more robust with respect to changes in controlled process parameters.

Weaknesses:

(1) It is unable to verify system stability, performance and robustness by using mathematical analysis.

(2) There is no systematic design method.

Persons with different control experiences may have different opinions on the above summary. One of the main debating topics about fuzzy control is concerned with the stability issue. The weakness of the analytical capability of fuzzy systems has been criticised by the analytical control community because mathematical analysis is a traditional way to design a control system and it is able to provide assurance of adequate performance or stability of any mathematically modeled system [35, 36, 37]. However, some researchers in the fuzzy control community strongly disagree with this argument and insists that "prototyping test is more important than stability analysis; stability analysis by itself can never be considered a sufficient test" [5, 6].

Viewing fuzzy control as a new technique for control purposes, it is natural to find some problems with it because there are many facts unexplored. More effort is needed to investigate the new area and find solutions to the analysis problems. A solution will arise from a strong body of experimental data defining the performance of fuzzy logic control systems. This was the major consideration behind the research on the robustness of fuzzy control systems reported in this thesis.

This chapter reviews the basics of fuzzy set, fuzzy reasoning, and fuzzy control theory. It provides information necessary for understanding the rest of this thesis. A review of literature on the design of fuzzy control systems and the achievement of robust control with fuzzy algorithm is presented on the last part of this *Chapter*. More detailed discussions on fuzzy set theory can be found in [3] [16] [25] - [28].

2.2 Fuzzy sets

The idea of a fuzzy set, as introduced by Zadeh [3], allows imprecise and qualitative information to be expressed in an exact way; it is a generalisation of the ordinary notion of a set. Whilst this basic idea can be utilised in many ways, it will be discussed here in the form which seems most applicable to the control problem.

2.2.1 Fuzzy Set

A *fuzzy set* F in a universe of discourse U is characterised by a membership function μ_F which takes values in the interval [0, 1], namely, $\mu_F: U \rightarrow [0, 1]$. Thus a fuzzy set F in U may be represented as a set of ordered pairs of a generic element u and its grade of membership function: $F=\{(u, \mu_F(u)) | u \in U\}$.

The terminology *membership function* associated with the fuzzy set F stresses the idea that for each $u \in U$, $\mu_F(u)$ indicates the degree to which u is a member of the set F. In the case when the universe U is continuous, the membership function μ_F is expressed in some functional form. Fig.2-1 illustrates two widely used membership functions: the triangular function and trapezoidal function.

In situations when the universe U is discrete, the membership grade of each element in a fuzzy set will be explicitly expressed. For example, assume $F = \{x_1, x_2, x_3, x_4\}$ and $x_i \in U$; then $\mu_F = \{0.7/x_1, 0.3/x_2, 1.0/x_3, 0/x_4\}$provides a representation of a fuzzy set of U. In this representation of fuzzy set F, terms of the form a_i/x_i are understood to indicate that the element x_i has membership grade a_i in the fuzzy set F. Using this interpretation we see in the above example that x_2 has membership grade 0.3. It is noted that the larger the membership grade of an element, the more strongly it is the member of the fuzzy set.

As an illustration, suppose it is necessary to specify a linguistic measure of temperature, for example, a statement such as "temperature is about 150°C". An ordinary set or crisp set which defines this can be expressed in terms of a membership function μ, which can take values of either 0 or 1. Graphically, this might be the rectangular function shown in Fig.2-2. A fuzzy set which expresses the same idea has a membership function which takes all values between 0 and 1. It might thus be as shown in Fig.2-3. The ordinary set is thus precise in this meaning, having a definite transition from membership to non-membership. The fuzzy set, on the other hand, allows the qualitativeness of the measure to be reflected in a gradual membership transition. Using this idea, qualitative information can be represented mathematically and handled in a completely rigorous manner.

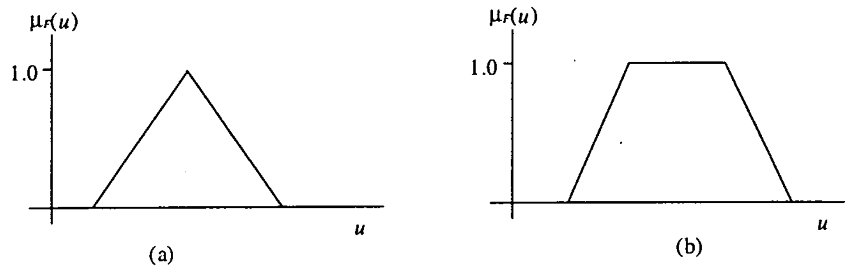


Fig.2-1 Triangular shape (a) and trapezoidal shape (b) membership functions

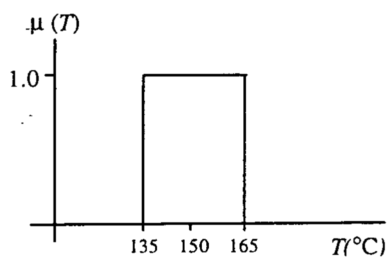


Fig.2-2 Non-fuzzy set of "temperature is about 150 °C"

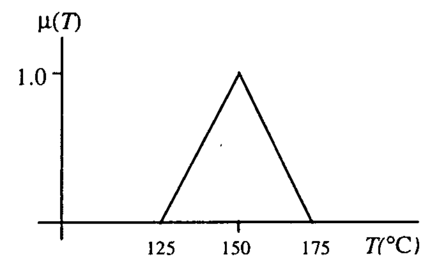


Fig.2-3 Fuzzy set of "temperature is about 150 °C"

Following are some of the terminology related to a fuzzy set.

Normal fuzzy set: A fuzzy set F is called normal if there exists at least one element in $u \in U$ such that $\mu(u)=1$. A fuzzy set that is not normal is called subnormal.

Support: The support of a fuzzy set F, denoted Sup(F), is the crisp set of all points u in U such that $\mu(u)>0$.

Core: The core of a fuzzy set F, denoted Core(F), is the crisp set of all points in u at which $\mu(u)$ achieves its maximum membership grade.

Fuzzy singleton: A fuzzy set whose support is a single point in U with $\mu(u) = 1.0$ is referred to as a fuzzy singleton.

It should be noted that in many cases the value of the membership grade for a fuzzy set is a subjective context dependent value, just as is the idea of belonging and not belonging to a crisp set. Furthermore, in many instances it is the shape of the membership function that is of significance rather than the actual grade values.

2.2.2 Fuzzy Set Operations

Many of the definitions of operations associated with fuzzy sets are straightforward extensions of corresponding definitions from the crisp set theory. These definitions usually return to their crisp counterpart when the fuzzy sets are restricted to membership grade of zero or one. In some cases, because of the fact that membership is drawn from the unit interval [0, 1] rather than simply {0, 1}, many definitions of fuzzy sets have been used. The following review will concentrate on the basic operations of fuzzy sets.

The theoretic operations on fuzzy sets are defined via their membership functions. The operation result is also a fuzzy set. Let A and B be two fuzzy sets in U with membership functions μ_A and μ_B, respectively. For all $u \in U$, the basic operations of *union, intersection* and *complement* on A and B are defined as:

Union $A \cup B$: $$\mu_{A\cup B}(u) = \max\{\mu_A(u), \mu_b(u)\} \tag{2.1}$$

Intersection $A \cap B$: $$\mu_{A\cap B}(u) = \min\{\mu_A(u),\ \mu_B(u)\} \tag{2.2}$$

Complement $\overline{A}$: $$\mu_{\overline{A}}(u) = 1 - \mu_A(u) \tag{2.3}$$

The union and intersection operations on the fuzzy membership functions are usually denoted by $\vee$ and $\wedge$ respectively, *i.e.*

$$\mu_{A\cup B}(u) = \mu_A(u) \vee \mu_B(u) \tag{2.4}$$

$$\mu_{A\cap B}(u) = \mu_A(u)\ \wedge\ \mu_B(u) \tag{2.5}$$

An illustration of union, intersection, and complement is given in Fig.2-4.

It is important to note that operations such as minimum and maximum indicate no interaction, *i.e.* as soon as one of the arguments (a grade of membership function) is greater (smaller, respectively) than the second one, it has no influence on the operation result. This property leads to a certain advantage, since no accurate value of the membership function is required. Thus the membership function can be estimated roughly, and the error in the measurement procedure can be largely ignored. This type of robust performance allows us to tolerate some imprecision in the real fuzzy control systems.

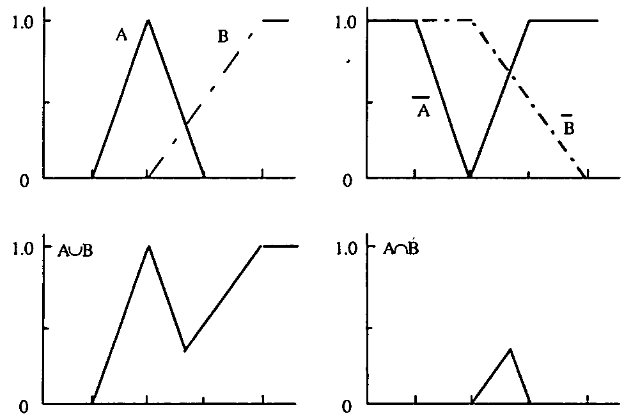


Fig.2-4 Operations on fuzzy sets

It should be noted that the union and intersection operations are related fuzzy sets in the same universe of discourse U. If the fuzzy sets concerned are not in the same universe, the operation on these fuzzy sets can be stated as the *Cartesian Product* which is defined as follows.

Cartesian Product: If $A_1, \cdots, A_n$ are fuzzy sets in $U_1, \cdots, U_n$ with their membership functions $\mu_{A_1}(u_1), \cdots, \mu_{A_n}(u_n)$, respectively, the Cartesian product of $A_1, \cdots, A_n$ is a fuzzy set in the product space $U_1 \times \cdots \times U_n$ with the membership function

$$\mu_{A_1\times\cdots\times A_n}(u_1, \cdots, u_n) = \min\{\mu_{A_1}(u_1), \cdots, \mu_{A_n}(u_n)\} \tag{2-6}$$

or

$$\mu_{A_1\times\cdots\times A_n}(u_1, \cdots, u_n) = \mu_{A_1}(u_1) \bullet \mu_{A_2}(u_2) \bullet \cdots \bullet \mu_{A_n}(u_n) \tag{2-7}$$

2.2.3 Linguistic Variable and Fuzzy Variable

In fuzzy control systems, a linguistic representation is used to convey knowledge about the control system and control strategies. Thus it is necessary to define linguistic variables in the system's design and analysis. A definition of a linguistic variable is as follows;

Linguistic Variable: If a variable can take words in natural languages (for example, small, fast, and so on) as its values, this variable is defined as a linguistic variable. These words are usually labels of fuzzy sets. A linguistic variable can take either words or numbers as its values.

For example, the linguistic variable *speed* can take "slow", "medium", and "fast" as its values. Term "slow" may be interpreted as "a speed below about 40mph", "moderate" as "a speed close to 55mph", and "fast" as "a sped above about 70mph". These values can be characterised by fuzzy sets in a universe of discourse U = [0, 100] with their membership functions shown in Fig.2-5.

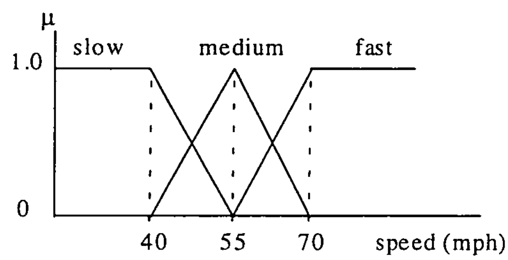


Fig.2-5. Membership functions of the fuzzy variable *speed*

The important feature of this representation of knowledge is the use of the fuzzy set to represent the meaning of a *word*. It provides a formal way to quantify a system's uncertainties which often exist in the practical systems.

A variable is called *fuzzy variable* if the variable uses fuzzy sets as its value. A linguistic variable is a fuzzy variable whose values are fuzzy sets labeled with words. In this thesis, the term *fuzzy variable* will be used for indicating the variable transferred from or to a non-fuzzy variable or a crisp variable, and *fuzzy sets* for the value of the fuzzy variable, rather than the terms *fuzzy set* and *fuzzy subset* respectively.

2.3 Fuzzy logic

In general, fuzzy logic is considered as a reasoning method based on fuzzy set theory. In fuzzy logic control, *fuzzy relation, fuzzy implication* and *fuzzy composition* are the most important concepts. In the following, the definitions of these concepts will be presented. It should be realised, though, that this can be done in several ways; the most commonly used definitions are given below.

2.3.1 Fuzzy relation

Assume X and Y are two fuzzy sets with their membership functions $\mu_X(x)$, $\mu_Y(y)$ respectively. A *fuzzy relation* R from a set X to set Y is a fuzzy set of the Cartesian product $X \times Y$ and is expressed as

$$R_{X \times Y} = \{((x, y), \mu_R(x, y)) \mid x \in X,\ y \in Y\}. \tag{2-8}$$

Its membership function $\mu_R(x, y)$ can be expressed by

$$\mu_R(x, y) = \mu_{X \times Y}(x, y) = \min\{ \mu_X(x),\ \mu_Y(y) \}. \tag{2-9}$$

More generally, if X_1, X_2, ... X_n are a collection of fuzzy sets, an n-array fuzzy relation is a fuzzy set over their Cartesian product $X_1 \times X_2 \times \ldots \times X_n$ with the membership function

$$\mu_{X_1 \times \cdots X_n}(x_1, x_2, \cdots, x_n) = \min\{\mu_{X_1}(x_1), \mu_{X_2}(x_2), \cdots, \mu_{X_n}(x_n)\} \tag{2-10}$$

An alternative expression of the fuzzy relation can be obtained by replacing the above min-operation by the product operation.

It is noted that the classical concept of relation is simply a special case of a fuzzy one where the membership grades are restricted to be zero or one. Since a fuzzy relation is simply a fuzzy set defined by the Cartesian product, then all the mechanisms developed for handling fuzzy sets can be used to manipulate the fuzzy relations.

2.3.2 Fuzzy implication

A fuzzy implication is a special type of fuzzy relation. Assume X and Y are two fuzzy sets with their membership functions $\mu_X(x)$, $\mu_Y(y)$ respectively. A fuzzy implication $X \rightarrow Y$ is a fuzzy set with the membership function;

$$\mu_{x \rightarrow y}(x, y) = \mu_X(x) * \mu_Y(y) \tag{2-11}$$

where * can by either the Min or the product operator.

In fuzzy logic control systems, a fuzzy implication A → B, is a fuzzy IF-THEN rule:

IF x is A, THEN y is B, where $x \in X$ and $y \in Y$ are linguistic variables.

As an illustration, a linguistic implication between electrical current input I, say, and temperature error ΔT might be

IF { ΔT is much great than 100°C } THEN { I is high }.

So if the fuzzy sets I = {current I is high } and ΔT_1 ={ temperature error ΔT is much higher than 100°C } are given by Fig.2-6 and Fig.2-7 respectively, the fuzzy set of this implication is a two dimensional fuzzy set R in the product space of $\Delta T \times I$. as shown in Fig.2-8.

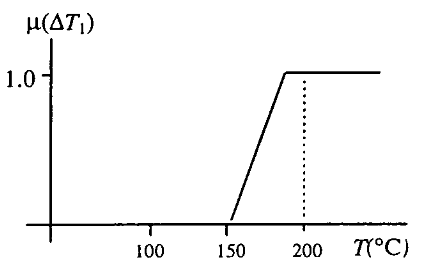


Fig.2-6 Fuzzy set ΔT_1 = "temperature error is much higher than 100 °C"

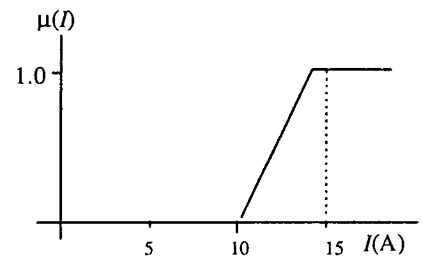


Fig.2-7 Fuzzy set of "current is high"

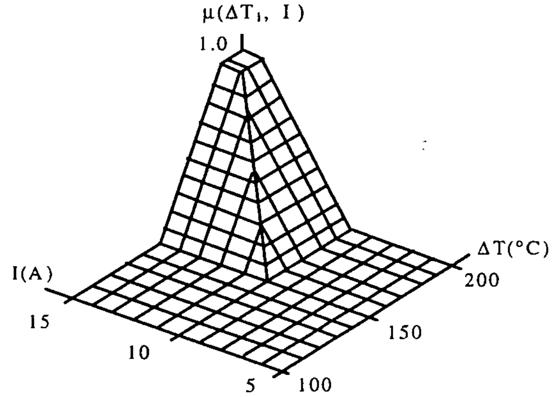


Fig.2-8 Fuzzy implication of *Temperature Error* ΔT and *Current* I

2.3.3 Fuzzy composition

Fuzzy relations in different product spaces can be combined with each other by the operation of *composition*. Assume X, Y and Z are fuzzy sets with their membership functions $\mu_X(x)$, $\mu_Y(y)$ and $\mu_Z(z)$ respectively. If R is a fuzzy relation from X to Y and S is a fuzzy relation from Y to Z, then the *composition* of R and S is a fuzzy relation denoted by $R \circ S$ with the membership function $\mu_{R \circ S}(x, z)$ as

$$\mu_{R \circ S}(x,z) = \sup_{y \in Y}[\mu_R(x,y) * \mu_S(y,z)] \tag{2-12}$$

where * is a fuzzy implication operator, $\sup_{y \in Y}$ is a pointwise operator of maximum in Y. Clearly, $R \circ S$ is a fuzzy set in $X \times Z$.

In the fuzzy logic control systems, the most commonly used compositions are the max-min and max-product because of the computational simplicity. The max-min and max-product can be expressed as

$$\text{max-min:} \quad \mu_{R \circ S}(x,z) = \max_y \{ \min \{ \mu_R(x,y), \mu_S(y,z) \} \}; \tag{2-13}$$

or

$$\text{max-product:} \quad \mu_{R \circ S}(x,z) = \max_y \{ \mu_R(x,y)\mu_S(y,z) \}. \tag{2-14}$$

A important case of the compositions in the fuzzy logic control is to join a fuzzy set X with a fuzzy relation R, $R \in (X \times Y)$. The fuzzy relation R presents the whole rule base and the fuzzy set X indicates the present system status. The composition results of X and R is a fuzzy set Y denoting the output fuzzy value of the FLC, *i.e.* $Y = X \circ R$. For example, the implication linking electrical current and temperature error in last sub-section, which is an algorithm with only one rule, can be used to make a statement about electrical current I when "temperature error ΔT is about 150°C". If fuzzy set ΔT_2 = { temperature error ΔT is about 150°C} is given in Fig.2-9, then the electrical current $\mu(I)$ can be estimated as follows by using max-min composition operation on ΔT_2 and $R(\Delta T_1, I)$.

$$\mu(I) = \underset{\Delta T}{Max} \; \{ \; \underset{\Delta T}{Min} \; [\, \mu(\Delta T_2), \; \mu_R(\Delta T_1, I) \,] \, \}.$$

Graphical illustrations on above Min-operation is shown in Fig.2-10(a) and Max-operation is shown in Fig.2-10(b). Note that the resulting fuzzy set of temperature in Fig.2-10(b) has a maximum membership function value much less than 1. This is because of the inadequacy of the algorithm (rule) described above. The resultant set should be interpreted as a "best estimate" given by this inadequate information.

It is clear that the Min-operation finds the minimum membership grade in the $\Delta T \times I$ space, and the Max-operation finds the maximum membership grade at every point in I space. Careful examination of the composition result can lead to the following general facts:

(1) The Max-operation is a projection of the result of Min-operation on I.

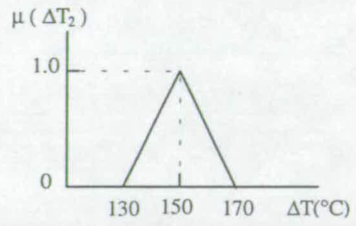


Fig.2-9 Fuzzy set of "temperature error ΔT is about 150°C"

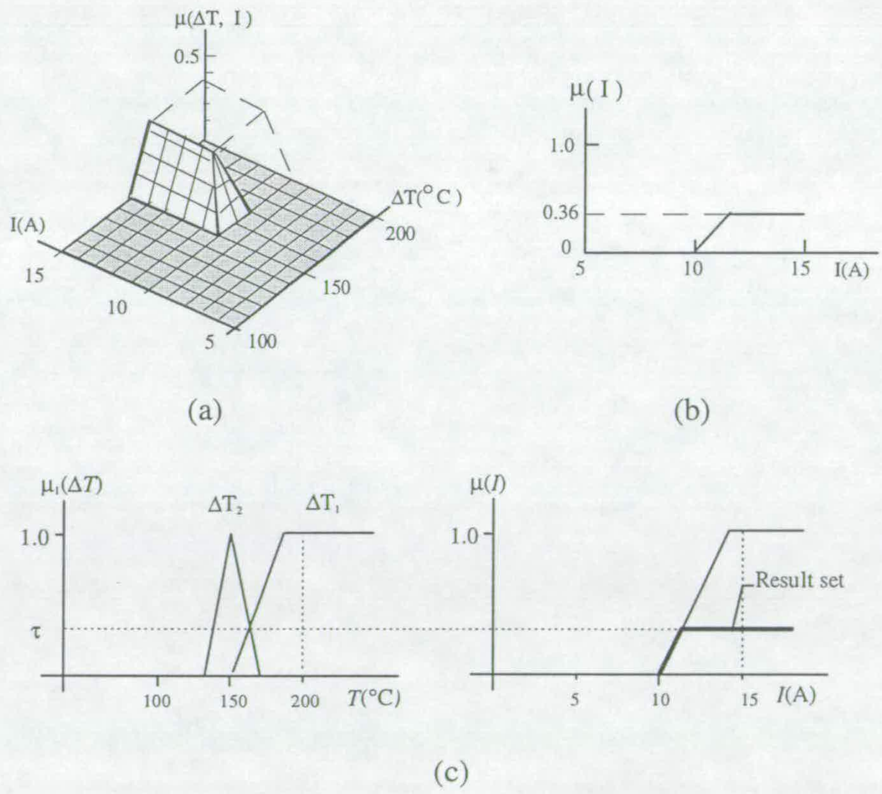


Fig.2-10 Illustration on the composition operation (a) Min-operation result; (b) Max-operation result; (c) simplified method

(2) The maximum membership grade of the resultant fuzzy set is equal to the maximum membership grade τ of fuzzy set $\Delta T_1 \cap \Delta T_2$.

The second fact is very important in fuzzy control applications because it significantly simplifies the operation procedure as shown in Fig.2-10 (c).

2.3.4. Fuzzy Connectives: AND and ALSO

It is necessary to interpret fuzzy connective AND and ALSO in the concept of fuzzy sets because these connectives are often used in fuzzy rules in an fuzzy control system, though ALSO is commonly omitted in practice. For example,

R_1: IF x is A_1 **AND** y is B_1 THEN z is C_1;

ALSO (2-15)

R_2: IF x is A_2 **AND** y is B_2 THEN z is C_2.

In most of the FLCs, the connective AND is implemented as a fuzzy intersection in a Cartesian product space in which the underlying variables take values in the different universe of discourse. As an illustration, the antecedent part of R_1 of example (2-15) is interpreted as a fuzzy set in the product space $X \times Y$ with membership function given by

$$\mu_{X\times Y}(x,\ y) = \text{Min}\{\ \mu_{A_1}(x),\ \mu_{B_1}(y)\ \}$$

or

$$\mu_{X\times Y}(x,\ y) = \ \mu_{A_1}(x)\ \cdot\ \mu_{B_1}(y)$$

where X and Y are the universe of discourse associated with A_1 and B_1 , respectively.

Several interpretations of the connective ALSO can be found in the published literature [24]. The interpretation of ALSO as a fuzzy union operator appears to be better suited for FLC applications because it is easier to implement. For example, the resultant fuzzy set C of example (2-15) can be given by

$$C = C_1^* \cup C_2^*$$

where C_1^* and C_2^* are the output fuzzy sets from R_1 and R_2, respectively.

2.4 Fuzzy logic controller (FLC)

The elementary basic diagram of a feedback control system is presented in Fig.2-11. The object to be controlled is called the plant, or the process, denoted as P. The purpose of the feedback controller, denoted as C, is to keep the output y close to the setpoint w, despite the presence of disturbances, noise and fluctuation of the plant parameters. The law governing corrective action of the controller is called the control algorithm. The output of the controller u is called the control action.

A common feature of conventional control is that the control algorithm is analytically described by equations; algebraic, difference, differential, and so on. In general, the synthesis of such control algorithms requires a formalised analytical description of the controlled process by a mathematical model. The concept of mathematical analysis is one of the main paradigms of conventional control theory.

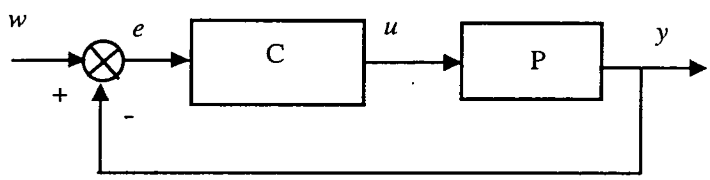


Fig.2-11 Block diagram of a basic feedback control system

The seminal work by L. Zadeh on fuzzy algorithms [3] [28] introduced the idea of formulating the control algorithm by logic rules. Mamdani and others [29] - [34] developed Zadeh's concept and demonstrated that logic rules with vague predicates can be used to derive inference from vaguely formulated data. They concluded that linguistic control algorithms can be used for control of complex systems, both human and technical.

The main paradigm of fuzzy control is that the control algorithm can be a knowledge-based algorithm, described by the methods of fuzzy logic. The knowledge encoded in the rule base is derived from human experience and intuition, and from theoretical and practical understanding of the dynamics of the controlled object. The machinery of approximate reasoning converts the knowledge embedded in the rule base into a crisp (non-fuzzy) control algorithm. What makes the fuzzy control special and conceptually different from conventional control is the lack of an analytical description.

A collection of implication statements (or rules) on the input and output fuzzy sets of a controlled process is called a fuzzy control algorithm or a *rule base*. A controller with this kind of control algorithm and the fuzzy logic inference method is called a *fuzzy logic controller*. Fig.2-12 shows the basic configuration of a practical FLC, which comprises five principal components: input and output scale mapping, a fuzzification operation, a rule base, an inference engine, and a defuzzification operation. It should be noted that, in most cases, the inference engine is designed to deliver change of control output, thus a serial accumulator is used for obtaining the actual control output. Of course, if the plant naturally includes an integrator, then the serial accumulator is not required. In the following, we shall discuss these basic elements of a fuzzy control system from the viewpoints of control system implementation.

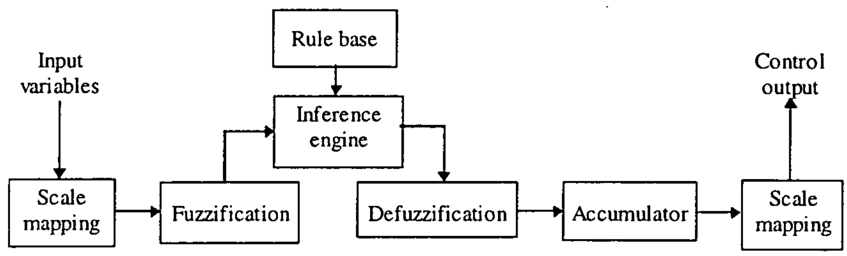


Fig.2-12 Basic configuration of fuzzy logic controller

2.4.1 Input and output scale mapping

The input scale mapping transfers the range of values of the practical input variables into a corresponding universe of discourse on which the input fuzzy sets (or linguistic variables) are designed. The output scale mapping transfers the universe of the output fuzzy variable of a FLC into the corresponding range of practical values of control output variable. These mappings can be uniform or non-uniform depending on the a priori knowledge of the concerned variable space.

A scale mapping of a variable in a FLC is done by multiplying a scaling factor with the variable. For example, in a three-input-one-output fuzzy controller, a control rule may read as

R_i : IF error (e) is A_i , integral of error (ie) is B_i , and change of error ($\dot{e}$) is C_i , THEN output control u is D_i .

A representation of the control rule can be written as

$$u(k) = K_4 F[\ K_1 e(k),\ K_2 ie(k),\ K_3 de(k)\ ], \tag{2-16}$$

where F denotes the fuzzy relation defined by the control rule and K_i , i=1,2,3,4, represents an appropriate scaling factor. In this relation, we see an analogy to the parameters of a conventional PID controller, in which as a special case F is a linear function of its arguments.

The scale mappings are needed when the universes of the fuzzy variables are normalised. Even when the maximum value of the input variables defines the normalised variables, it is usually necessary to tune scaling factors for obtaining the desired performance [25].

2.4.2. Fuzzification

In fuzzy control applications, the observed input data are usually crisp. Since the data manipulation in a FLC is based on fuzzy set theory, a transformation which converts a scaled measurement into a fuzzy value is necessary during the earlier stage. Fuzzification can be defined as a mapping from an observed input space to fuzzy sets in the input universe of discourse. Basically, fuzzification consists of the following two operations on the input variables.

A. Representation of sampled data

To manipulate sampled data with fuzzy sets by using fuzzy set theory, it is necessary to represent the sampled data as a fuzzy set, called *fuzzy sampled data*. Two principal ways to represent the sampled data are as follows.

1) If a sampled data x_0 is precise or the measurement noise can be ignored, it can be represented as a fuzzy singleton A with the membership function $\mu_A(x)$ equal to zero except at the point x_0 , at which $\mu_A(x_0)$ equals one. This representation has been widely used in fuzzy control applications since it is natural and easy to implement.

2) If sampled data is contaminated by random noise, the noisy measurement can be interpreted as fuzzy sets with a triangular membership function. This membership function is constructed with respect to the probability density function of the noise. The vertex of the triangle is associated with the mean value and the base is a function of the standard deviation [25, 33]. This method is much more complicated than the fuzzy singleton method, and requires further exploration.

B. Conversion

The conversion operation converts fuzzy sampled data into suitable linguistic values, which may be viewed as the labels of fuzzy sets, with a degree of membership to each of these fuzzy sets. The conversion is based on the membership functions defined on the universe of each input fuzzy variable. Assume that membership function $\mu_X(x)$ is defined in the universe of input fuzzy variable X, and fuzzy singleton A and fuzzy set B are two representations of sampled data, $x \in X$. Then the calculation of the membership degree τ of x to fuzzy set X can be illustrated as shown in Fig.2-13.

Primary fuzzy sets are usually labeled with linguistic terms, such as *NB*: negative big; *NM*: negative medium; *NS*: negative small; *ZE*: zero; *PS*: positive small; *PM*: positive medium; *PB*: positive big. A typical example was shown in Fig.2-14.

Membership functions can be different shape, asymmetrical or evenly distributed in the universe. Furthermore, the number of the primary fuzzy sets determines the maximum number of fuzzy rules. For example, if each fuzzy input variable has 7 primary fuzzy sets (7 membership functions), a two-input-one-output fuzzy system then has a maximum number of 49 rules.

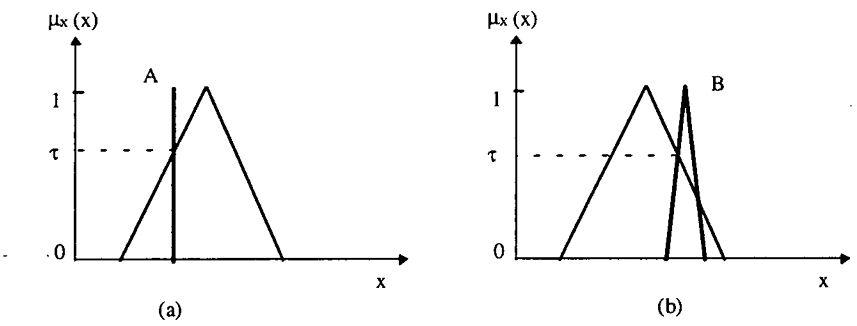


Fig.2-13 Calculation of membership degree τ of a sampled data in the form of (a) fuzzy singleton A and (b) fuzzy set B.

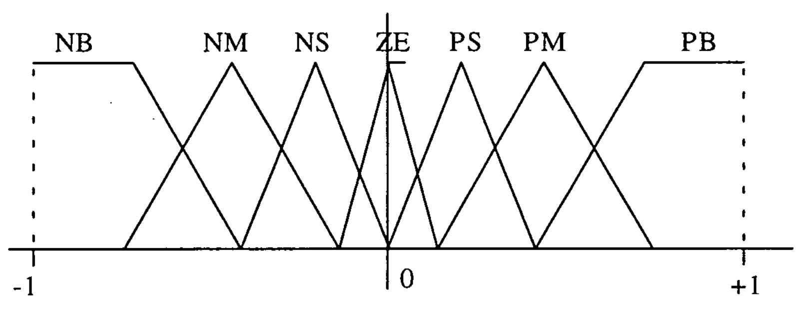


Fig.2-14 Typical membership functions

It should be noted that the membership function can be defined numerically or functionally, depending on whether the universe of discourse is discrete or continuous. Also, the fuzzy partition of the fuzzy input space has no unique solution. The concepts associated with the membership functions are used to characterise fuzzy control rules and the manipulation of sampled data in a FLC. These concepts are subjectively designed and they are based on experience and engineering judgement.

2.4.3. Rule base

A fuzzy system is characterised by a set of linguistic statements based on expert knowledge. The expert knowledge is in the form of "if-then" rules. The collection of these fuzzy control rules that are expressed as fuzzy conditional statements forms the rule base in a FLC. Generally, the rule base has the form of a MIMO system. However, it is proved in [24] that a MIMO fuzzy system can be represented as a collection of MISO fuzzy systems. Therefore, the MISO fuzzy system will only be consider in this research.

Most FLCs have a fuzzy rule base which, in the case of MISO systems, is a collection of rules of the form

$$R_i\text{: IF } x_1 \text{ is } A_{i1}, \dots, \text{ and } x_n \text{ is } A_{in} \text{ THEN } y \text{ is } B_i\,; \tag{2-17}$$

where $x_1, \dots, x_n$ and y are linguistic variables representing the process states; $A_{i1}, \dots A_{in}$ and B_i (i = 1, 2, ..., m, m is the rule number) are the linguistic values of the fuzzy variables x_1, ... , x_n and y in the universe of discourse X_1 , ..., X_n and Y, respectively. Rule R_i is interpreted as a fuzzy implication $X_1 \times \dots \times X_n \rightarrow Y$.

Typically, system error e and error derivative $\dot{e}$ are widely used as the input fuzzy variables, and change of control output Δu is used as the output fuzzy variable of fuzzy rules in fuzzy control applications. In this case, let E, DE and DU as the fuzzy variables of e, $\dot{e}$ and Δu respectively. Then a typical rule may be written as:

If E is A and DE is B, then DU is C,

where A, B *and* C are one of the linguistic labels (fuzzy sets): NB, NM, NS, ZE, PS, PM and PB. These fuzzy control rules can be represented as a matrix as shown in Fig.2-15, where the linguistic labels of input variables work as index of the matrix, and labels of the consequent part in the linguistic rules take the main part of the matrix.

		DE						
		NB	NM	NS	ZE	PS	PM	PB
	NB	NB	NB	NB	NB	NM	NS	ZE
	NM	NB	NB	NB	NM	NS	ZE	PS
	NS	NB	NB	NM	NS	ZE	PS	PM
E	ZE	NB	NM	NS	ZE	PS	PM	PB
	PS	NM	NS	ZE	PS	PM	PB	PB
	PM	NS	ZE	PS	PM	PB	PB	PB
	PB	ZE	PS	PM	PB	PB	PB	PB

Fig.2-15 A typical fuzzy rule base

Suppose the membership functions of *E, DE* and *DU* are defined in the format as shown in Fig.2-14, and e and $\dot{e}$ are defined as

$$e(k) = w(k) - y(k) \tag{2-18}$$

$$\dot{e}(k) = \frac{e(k) - e(k-1)}{t_s} \tag{2-19}$$

where *w(k)* and *y(k)* are the setpoint and the output of the control system respectively, t_s is the sampling period. Then the meaning of rules in the Fig.2-15 can be interpreted with reference to the phase plane as illustrated in Fig.2-16. Different consequent fuzzy labels were indicated by the areas with different darkness in the phase plane (overlaps are not shown). The upper right part and the lower left part in the phase plane can be interpreted as areas where the system output is changing toward a smaller system error, where the control rules (control action) are sensitive to the system status, indicating a very careful control action. While in the upper left part and lower right part of the phase plane, the interpretation is that the system output is changing toward a larger system error, where the control rules (control action) are less sensitive to the system status, indicating a very tough control action to drive the output back to the system setpoint. Clearly the typical fuzzy rule base implements a general control heuristic called 4-zone strategy [86].

However, it cannot be expected that the above fuzzy rule base has the ability to solve all control problems. Obviously, for complicated control issues, we need more knowledge, and therefore a more detailed rule base, that will supply a more sophisticated control strategy. It should not be overstated [35] [36] [37] that a designer of a fuzzy control system does not need any information about the controlled process.

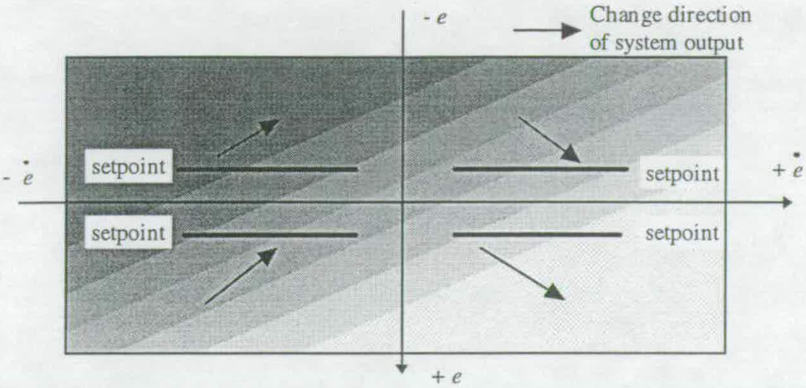


Fig.2-16 A explanation of the typical fuzzy rule base shown in Fig.2-15

Clearly, this straightforward formulation of the control algorithm allows the implementation of heuristic strategies, defined by linguistic statements. The fuzzy control algorithm reflects the mechanism of control, implemented by people, without using any formalised knowledge about the controlled process in the form of mathematical models, and without an analytical description of the control algorithm. These characteristics explicitly demonstrate the inherent robustness of the fuzzy control systems.

2.4.4. Inference engine — fuzzy reasoning

The inference engine is the kernel of a FLC. It has the capability to simulate human decision making based on fuzzy set concepts and it infers fuzzy control actions by employing fuzzy implication and compositional rules of inference in fuzzy logic. Basically, it evaluates the state of the process under control based on the fuzzy inputs, and then the fuzzy input/output relations defined in the rule base are used to compute the new control output.

A fuzzy inference engine with fuzzy rules in (2-17) can be treated as a system that maps fuzzy sets in $U = X_1 \times ... \times X_n$ into fuzzy sets in Y via fuzzy implication R_i defined by its membership function $\mu_{U\rightarrow Y}(u, y)$. This mapping is often called *fuzzy reasoning*. Let a fuzzy set A in U be the input to the fuzzy inference engine; then each fuzzy IF-THEN rule (2-17) determines a fuzzy set Y_i in Y by using fuzzy composition operation. That is

$$Y_i = A \circ R_i\,,\ \mu_{Y_i}(y) = \sup_{u\in U}[\mu_A(u)\ *\mu_{U\rightarrow B_i}(u,\ y)]. \qquad (2\text{-}20)$$

The final fuzzy set B in Y is determined by all rules R_1 , R_2 , $\cdots$, R_n in the fuzzy rule base.

Its membership function, $\mu_B(y)$, is obtained by combining $\mu_{Y_i}(y)$ for all i = 1, 2, ... , m using the fuzzy UNION operation because of the ALSO connectives. That is

$$B = A \circ (R_1 , R_2 , \cdots, R_n), \qquad (2\text{-}21)$$

$$\mu_B(y) = \mu_{Y_1}(y) \vee \mu_{Y_2}(y) \vee \cdots \vee \mu_{Y_m}(y). \qquad (2\text{-}22)$$

Because there are two types of fuzzy composition operations: max-min composition and max-product composition, equation (2-20) can be written as;

$$\text{min-inference:} \quad \mu_{Y_i}(y) = \text{Max}_{u \in U}\{\text{Min}[\mu_A(u),\ \mu_{U \to B_i}(u,\ y)]\} \qquad (2\text{-}23)$$

$$\text{product-inference:} \ \mu_{Y_i}(y) = \text{Max}_{u \in U}[\mu_A(u) \cdot \mu_{U \to B_i}(u,\ y)] \qquad (2\text{-}24)$$

It is obvious that the rule inference is just a series of fuzzy composition operations on the input fuzzy sets and the implication R_i. However, there is a major difference in the number of input fuzzy sets between the rule inference here and the composition operation in *Section* 2.3.3. The number of fuzzy sets in the composition operation will significantly affect the applicability of the FLC algorithms. To simplify this multi-input composition operation, C.C. Lee proved in [24] the following lemma.

Lemma 2-1: Consider the following fuzzy implication

IF x is A and y is B THEN z is C.

Let A′ be the input fuzzy set of x and B′ be the input fuzzy set of y. Suppose membership functions of A, B and A×B are μ_A, μ_B, and $\mu_{A\times B}$, respectively. Then the output fuzzy set C′ can be written as

$$(A', B') \circ (A \text{ and } B \to C) = [A' \circ (A \to C)] \cap [B' \circ (B \to C)], \quad \text{if } \mu_{A\times B} = \mu_A \wedge \mu_B ;$$

$$(A', B') \circ (A \text{ and } B \to C) = [A' \circ (A \to C)] \cdot [B' \circ (B \to C)], \quad \text{if } \mu_{A\times B} = \mu_A \cdot \mu_B .$$

The above lemma can be interpreted as that a multi-input fuzzy composition is equal to the Cartesian product of all compositions of each input and the implication from the input to the consequent. This fuzzy inference algorithm has been widely used in fuzzy control applications.

By applying the above lemma to equation (2-21) and assuming that the input fuzzy sets are X_1', ..., X_n', *i.e.* $A = X_1' \times ... \times X_n'$, the final output fuzzy set B can be further developed as follows:

$$
\begin{aligned}
B &= A \circ (R_1, R_2, \cdots, R_n) \\
&= \bigcup_{i=1}^{m} (A \circ R_i) \\
&= \bigcup_{i=1}^{m} \{(X_1' \times \cdots \times X_n') \circ \mathrm{R_i}\} \\
&= \bigcup_{i=1}^{m} \bigcap_{j=1}^{n} \{X_j' \circ (A_{ij} \rightarrow B_i)\}
\end{aligned}
$$

Applying the fact 2 in *Section* 2.3.3 to the above function and assuming τ_{ij} is the membership grade of X_j' to A_{ij} (i = 1, ..., m, j = i, ... n), the membership function of the consequent fuzzy set B_i in the ith rule (2-17) can be written as

$$
\begin{aligned}
\mu_B(y) &= \bigvee_{i=1}^{m} \bigwedge_{j=1}^{n} \{\tau_{ij} * \mu_{B_i}(y)\} \\
&= \bigvee_{i=1}^{m} \{\tau_i * \mu_{B_i}(y)\},
\end{aligned}
\tag{2-25}
$$

where * denotes the Cartesian product, τ_i is called the *degree of firing* (DOF) of the ith rule and is calculated as

$$
\tau_i = \begin{cases} Min\{\tau_{i1}, \tau_{i2}, \cdots, \tau_{in}\} & \text{if min - inference ;} \\ (\tau_{i1} \bullet \tau_{i2} \bullet \cdots \bullet \tau_{in}) & \text{if product - inference.} \end{cases} \tag{2-26}
$$

In summary, the procedure of the fuzzy reasoning is based on the following three steps:

(1) Determine the DOF, τ_i, of the ith rule (i = 1, ..., m).

(2) Calculate the Cartesian product, I_i, of τ_i and the membership function $\mu_{B_i}(y)$ of consequent fuzzy set B_i in the ith rule (i = 1, ..., m).

(3) Calculate the union of all I_i to obtain the final fuzzy output set B.

The following is an illustration of the above procedure. Assume that there are two fuzzy control rules:

R_1: if x is A_1 and y is B_1 then z is C_1;

R_2: if x is A_2 and y is B_2 then z is C_2;

where x, y and z are fuzzy variables representing the process states; A_i, B_i and C_i (i = 1, 2) are the fuzzy values of the fuzzy variables x, y and z in the universe of discourse X, Y and

Z, respectively. Suppose that the membership functions of A_i, B_i and C_i ($i = 1, 2$) are as shown in Fig2-17, and the input fuzzy sets are such that x is x_0 and y is y_0, where x_0 and y_0 are fuzzy singletons as is the common situation in practical fuzzy logic control systems. Two types of fuzzy reasoning are applied as follows.

A. Max-Min inference

In this model of reasoning, each rule leads to the control decision

$$\mu_{R1} = \mathrm{Min}\{\ \alpha_1,\ \mu_{C1}(z)\ \}$$
$$\mu_{R2} = \mathrm{Min}\ \{\ \alpha_2,\ \mu_{C2}(z)\ \}$$

where $\alpha_1 = \mathrm{Min}\{\ \mu_{A1}(x_0), \mu_{B1}(y_0)\ \}$, and $\alpha_2 = \mathrm{Min}\{\ \mu_{A2}(x_0), \mu_{B2}(y_0)\ \}$. The membership function μ_Z of the inferred consequence is pointwise given by

$$\mu_Z(z) = \mu_{R1} \vee \mu_{R2}\ .$$

The fuzzy reasoning process is illustrated in Fig.2-17.

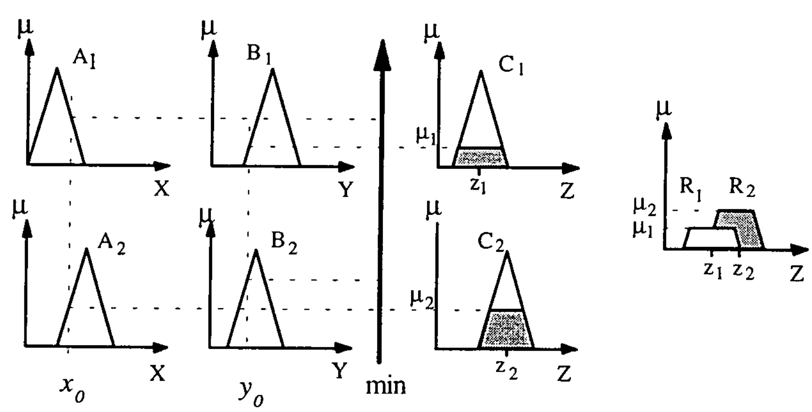


Fig.2-17 Max-Min inference method

B. Max-Product inference

In this model of reasoning, each rule leads to the control decision

$$\mu_{R1} = \alpha_1 \cdot \mu_{C1}(z)$$
$$\mu_{R2} = \alpha_2 \cdot \mu_{C2}(z)$$

where $\alpha_1 = \mathrm{Min}\{\ \mu_{A1}(x_0), \mu_{B1}(y_0)\ \}$, and $\alpha_2 = \mathrm{Min}\{\ \mu_{A2}(x_0), \mu_{B2}(y_0)\ \}$. The membership

Z, respectively. Suppose that the membership functions of A_i, B_i and C_i (i = 1, 2) are as shown in Fig2-17, and the input fuzzy sets are such that x is x_0 and y is y_0, where x_0 and y_0 are fuzzy singletons as is the common situation in practical fuzzy logic control systems. Two types of fuzzy reasoning are applied as follows.

A. Max-Min inference

In this model of reasoning, each rule leads to the control decision

$$\mu_{R1} = \text{Min}\{\ \alpha_1,\ \mu_{C1}(z)\ \}$$
$$\mu_{R2} = \text{Min}\ \{\ \alpha_2,\ \mu_{C2}(z)\ \}$$

where $\alpha_1 = \text{Min}\{\ \mu_{A1}(x_0), \mu_{B1}(y_0)\ \}$, and $\alpha_2 = \text{Min}\{\ \mu_{A2}(x_0), \mu_{B2}(y_0)\ \}$. The membership function μ_Z of the inferred consequence is pointwise given by

$$\mu_Z(z) = \mu_{R1} \vee \mu_{R2}\ .$$

The fuzzy reasoning process is illustrated in Fig.2-17.

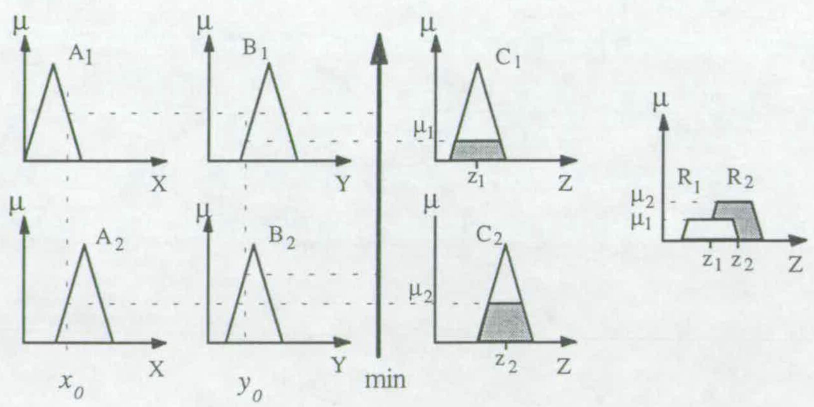


Fig.2-17 Max-Min inference method

B. Max-Product inference

In this model of reasoning, each rule leads to the control decision

$$\mu_{R1} = \alpha_1\ .\ \mu_{C1}(z)$$
$$\mu_{R2} = \alpha_2\ .\ \mu_{C2}(z)$$

where $\alpha_1 = \text{Min}\{\ \mu_{A1}(x_0), \mu_{B1}(y_0)\ \}$, and $\alpha_2 = \text{Min}\{\ \mu_{A2}(x_0), \mu_{B2}(y_0)\ \}$. The membership function μ_Z of the inferred consequence is pointwise given by

$$\mu_Z(z) = \mu_{R1} \vee \mu_{R2} \ .$$

This fuzzy reasoning process is illustrated in Fig.2-18.

2.4.5. Defuzzification

Basically, defuzzification is a mapping from a space of control output fuzzy set defined over an output universe of discourse into a space of non-fuzzy (crisp) control actions. This is because, in most practical applications, a crisp control action is required. At present, the commonly used strategies are the mean of maximum (MOM), and the centre of gravity (COG).

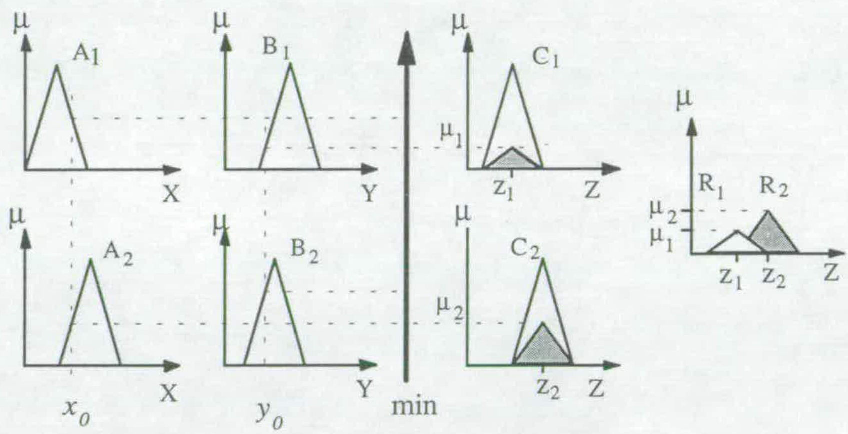


Fig.2-18 Max-Product inference method

MOM : The MOM strategy generates a control action which represents the mean value of all local control actions whose membership functions reach the maximum. More specifically, in the case of a discrete universe with finite elements, the control action may be expressed as;

$$y^{MOM} = \frac{1}{l} \sum_{y_i \in G} y_i \tag{2-27}$$

where G is the set of elements in the output universe Y which attains the maximum value of membership grade, and l is the number of such elements in G.

COG: This widely used strategy produces the centre of gravity of the possibility distribution of a control action. In the case of a discrete universe with finite elements, it yields;

$$y^{COG} = \frac{\sum_{i=1}^{m} \mu_B(y_i) y_i}{\sum_{j=1}^{m} \mu_B(y_i)} \tag{2-28}$$

where B is the fuzzy output set in the output universe Y, $\mu_B(y)$ is the membership function of B, and m is the number of elements in B.

If the output universe Y is continuous, the sum operations in(2-27) and (2-28) become integration in the corresponding space.

It should be noted that the defuzzification problem not only exists in fuzzy control area, but also in other decision making areas using fuzzy set theory. A number of different defuzzifiers have been proposed in fuzzy logic literature, and their use is quite subjective [38]. Some comparative studies of the defuzzification problem were presented in [34] [39] - [41]. An overview of the defuzzification methods was given by Lee in [24], indicating that the COG method yields a better steady-state performance while the MOM yields a better transient performance.

2.5 Design of fuzzy logic control systems

The control engineer, when attempting fuzzy control, is generally faced with three main problems: the rule derivation, determining the parameters (linguistic labels and membership functions) to be used together with the rule base, and tuning of a FLC to meet the performance requirements. It is important to realise that the solution to one of these problems is highly dependent on the solution of the other two. As indicated in many publications, use is made of *a priori* knowledge of the process.

There are two principal approaches to designing a FLC. The first is a heuristic method in which a collection of fuzzy rules and the parameters of the controller are formed by utilising expert experience, or by analysing the behaviour of a process operator. The controller is designed in such a way that the rule derivation from a desired state can be corrected and the control objective can be achieved. This method is purely heuristic in nature and relies on the qualitative knowledge of process behaviour. The second approach is basically a deterministic method which can systematically determine the rule base and/or parameters (membership functions and scaling factors) of the FLC. In this case, several design methods have been studied based on, for example, the self-organised principle, the neural network technique and the genetic algorithm. A brief review of these methods is given in the following sections.

2.5.1 Based on expert experience

The formulation of fuzzy control rules is achieved by means of two heuristic approaches. The most common one involves an introspective verbalisation of human expertise. Typical examples of such method are the application of fuzzy control to a cement kiln [43], and the early application of fuzzy logic to industrial control [30]. Many experts have found that the fuzzy control rules provide a convenient way to express their domain of knowledge [24]. This explains why most FLCs are based on the knowledge and experience which are expressed in the language of fuzzy if-then rules.

Another approach derives fuzzy rules from the working principle of the widely used PID, or PI controllers. Example of this method can be found in [47] and [50] which uses input and output mapping factors to design a three term controller (PID like) without any information about the plant to be controlled. The rule base and membership functions are symmetrically selected without any consideration of the plant's features. The performance of the closed-loop system is justified by tuning the input and output mapping factors which affect the membership functions of each input or output fuzzy variable. The weakness of this method stems from the assumption that the plant to be controlled is not highly nonlinear which enables the membership functions and rule base to be designed symmetrically and to be tuned evenly.

2.5.2 Based on measurements

Wang and Mendel [42] proposed an algorithm for deriving the fuzzy rules and choosing controller parameters from numerical data obtained from measurements. A simple and one-pass build-up procedure has been introduced and proved to be a success by applying it to the truck backer-upper control problem. This method provides a general method to combine measured numerical information and human linguistic information into a common fuzzy rule base. It is specially useful if only partial linguistic fuzzy rules and partial input-output data pairs are available in designing the fuzzy logic controller. However, resolving the conflict rules derived from the numerical data and the linguistic information is heuristic because no systematic method is given to tell the quality of each data pair which is used to generate the rule base. It is also difficult, or impossible in some case, for using this method to build-up a complete rule base to cope with all system situations in case of the time-varying process.

2.5.3 Based on fuzzy model

An approach to design a FLC, which is analogous to the conventional controller design by pole placement, is introduced by Braae and Rutherford in [13] and [51]. Braae and Rutherford assumed that the linguistic rules (model) of a process and a desired closed-loop system were initially given. The purpose is to synthesise a linguistic control element (FLC) based on the fuzzy models assumed above. The main idea is to invert the low order linguistic model of the process to find maps from linguistic system states to linguistic process control action. However, linguistic inversion mappings are usually incomplete or multi-valued. So, an "approximate" strategy, which is somewhat heuristic and subjective, is necessary to complete the inverse mapping which has a reasonable single-valued solution. This method is restricted to relatively low order system but it provides an explicit solution for the design of the FLC, assuming that fuzzy models of the open and closed systems are available.

2.5.4 Based on operator's control actions

Takagi and Sugeno [44, 45] proposed a fuzzy identification algorithm for modeling human operator's control actions. In this case, a suitable linguistic structure is easy to find since one can observe and/or ask for the information which the operator needs, such as process state variables. The design problem is reduced to parameter estimation, which is done by optimising a least-square performance index via a weighted linear regression method. This method provides a more systematic approach to the design of a FLC, and the experimental results are quite remarkable. However, some design steps of this algorithm, such as the choice of process state variables, the fuzzy partition of input spaces, and the choice of membership functions of primary set, depend on trial-and-error.

2.5.5 Based on optimisation

In the past few years, more and more research work has been directed towards the design of fuzzy control systems by optimisation methods because of the multi-parameter tuning problem. Isaka and Sebald [46] proposed an optimisation approach to design the membership functions of a FLC. This method is based on high-dimensional simulated annealing that produces a set of optimal membership function values through repeated simulations. In [52], a genetic algorithm was used to simultaneously design the membership functions and rule base in fuzzy controllers. The advantage of the optimisation method is that it significantly reduces time and effort to find appropriate parameters for the membership functions and, especially, to tune the FLC. However, a simulation model of the process is usually required to carry out the simulations.

2.5.6 Based on learning

A phase-plane self-organising FLC (SOFLC) was proposed by King and Mamdani [30]. It involves tracking the closed-loop system trajectory across the domain of the fuzzy controller. By relating each rule to an area in this domain or phase-plane, it is possible to update and create the relevant rules by applying some meta-rules for rule derivation. This technique was modified by Braae and Rutherford [33] by tracking the system trajectories through the linguistic space instead of the real space. The advantage of this method is that design of a FLC can be performed without any knowledge of the controlled process and it reduces the time to tune the FLC. However the selection of the improved rules relies heavily on an intuitive feel (meta-rules) for the behaviour of the closed-loop system, and the convergence of the algorithm may become a problem in practical applications.

Recently, considerable research effort has been devoted to the neural fuzzy area, and neural network methods have been proposed to tackle the design problems of FLC systems. This research area has already become a new branch of science but it is out of the research scope of this thesis.

2.6 Robustness of fuzzy control systems

As discussed in the introductory chapter, system robustness is one of the most important features of fuzzy control systems when system uncertainty exists. There are two important issues related to the implementation of robust fuzzy control: how to measure system robustness and how to design a robust fuzzy control system..

2.6.1 Measurement of system robustness

The proper measurement of robustness depends on the situation at hand. Basically, the robustness is the amount of uncertainty the system can tolerate and still function satisfactorily. So selection of the measure of robustness is according to what uncertainty is present in the system.

Different branches of control theory have different opinions about what "uncertainty" is. In adaptive control, people talk about "robustness towards measurement noise", *i.e.* how much measurement noise can be added to the measurements still allow the adaptation algorithm to converge "fast enough".

The more common usage, however, is that uncertainty means model errors. In classical control, model errors are modeled as unknown gain blocks or phase delays situated somewhere in the control loop. Furthermore, that the system functions satisfactorily means that it is stable. Robustness towards such uncertainty is then measured by gain and phase margins within which the changes of the unknown gain or phase delays will not make the system unstable.

Many control theorists simply equate robustness with the H_∞ framework [2]. Uncertainty is then modeled as MIMO transfer functions which are unknown but have an H_∞ norm less than one [2, 64]. The robustness is then measured by the H_∞ of some closed-loop transfer function. An extension of the H_∞ robustness is the μ-analysis, where several unknown transfer functions are present.

Another widely used form of robustness is that of parametric uncertainty. *I.e.* how much variation in a physical parameter of the controlled process can the system tolerate before the system goes unstable or the performance becomes too poor. The robustness is then measured in terms of a parameter range or, in the case of many uncertain parameters, of the radius of a ball in parameter space in which the system functions satisfactorily.

In this research work, parameter uncertainty only was considered, and system performance under this parameter uncertainty was emphasised. Therefore, the variation range of parameter(s), for which system performances meet the predefined specifications, was used as the measurement of system robustness.

2.6.2 Studies on robust control with fuzzy logic algorithms

Since the first introduction of the FLC in the 1970's by Mamdani [53] there has been a considerable world wide interest in this subject. Though the robust property of a FLC was claimed in many publications [5, 7, 8, 11, 24, 27, 30, 54 - 56, 59, 74, 75, 97], the robust performance of a FLC was not the main objective of any research in the fuzzy control field except for a few publications which reported initial investigations [16] and experimental assessments [30][54]. It was only recently that investigations on the robust capabilities of FLC systems have been reported [9, 12, 61, 63, 65 - 67, 103].

King and Mamdani [30] compared the sensitivities of a FLC with that of a conventional controller by changing the parameters of a process. They found that in a certain range of process dynamics the conventional controller is difficult to adjust for good responses, while the fuzzy control system is less sensitive to process parameter changes and gives good control at all operating points. A different method to test the robust performance of a FLC was proposed by Kosko [54] by replacing some rules with destructive rules or by removing some rules in the truck backing up system. A comparison was also made of the robust performances of a FLC system with a neural network system. The results show that the rule perturbations did not significantly affect the fuzzy controller's performance, and the FLC system produced a better performance than the neural network system when the prior knowledge about the controlled process was reduced.

Tanaka and Sano [9] discussed the robust stability of Takagi-Sugeno type fuzzy systems [45] using the Lyapunov Direct Method. In this paper, a fuzzy system was described by the fuzzy model containing parameter matrix (**A**, **B**, **C**), and four conditions for ensuring stability of fuzzy system and the concept of stability margin were introduced in the sense of the Lyapunov theorem. However, it is difficult to obtain the matrix information in practice, and the mathematical operation on the fuzzy sets in the format of a matrix is so complicated that the practical application of this method becomes difficult. In addition, the effect of the parameters of both the controlled process and the FLC on the robust performance was not studied, and no design procedure for a robust FLC was proposed.

The fuzzy sliding mode method was proposed by Palm in [10], followed by Kawaji [102], Hwang [101], Palm [61] and Wu [103]. In this method, a class of FLCs designed with respect to the phase plane $e \times \dot{e}$ was viewed as a sliding mode controller (SMC). In this class of FLCs, the general approach to the controller design is to partition the phase plane into two semi-planes where positive and negative control actions are produced respectively. The similarities between the FLC and the SMC were emphasised and the system stability and robustness were discussed, based on the sliding mode control principle in [10] and [61]. It is concluded that the robustness of FLC systems designed with a two-dimensional phase plane stemmed from their property of driving the system into the sliding mode, in which the control system is invariant to system parameter fluctuations and disturbances. However, this study was restricted to FLCs with a symmetrical rule base and 2-overlapped membership functions (the maximum number of overlapped membership functions is 2). The effect of the membership functions and the rule base on system robustness was not investigated, and a design method for improving system robustness was not provided.

Wang and Jordan, [65], presented the results of an experimental study on the effect of parameters of both the controlled process and the FLC on the system robust performance by exhaustive simulation in the parameter space. Results showed that, by choosing proper scaling factors of a FLC designed by common sense, a very good performance can be obtained if the variation range of the plant parameters is not wide. Alternatively a wide robust range can be achieved if the performance required at the tuning point is degraded from the best step response obtainable by tuning the scaling factors. A guideline to tuning the scaling factors was given.

Nguyen, [12, 63], and Wang, [67], presented experimental investigations of the robust step performance of SISO systems controlled by FLC and PID algorithms. A large number of simulation results quantified the claim that fuzzy control leads to an improved robust performance with respect to the use of PID control. The effects of tuning-point selection and tuning-point performance requirement on the system robustness were also presented.

Jordan and Wang, [66], introduced a phase advanced fuzzy logic control algorithm, which uses a proportional path in parallel with the fuzzy control output accumulator. This parallel path was introduced to minimise the very strong destabilising influence of the accumulator (integrator) required at the output of the fuzzy controller. The phase advance provided by the modified controller clearly leads to a better step performance, especially when high order processes are controlled. Experimental results were presented showing that the modified controller provides a more robust performance with respect to variation of the parameters of the controlled process.

Overall, then, what conclusions may be drawn from the literature survey? Firstly and most importantly, the fuzzy logic controller was claimed to be very robust by many researchers. It appears to tolerate process parameter changes well and has reasonable noise rejection capabilities. Secondly, there is no systematic method to analyse the stability of FLC systems. This makes it difficult to investigate the robust performance of the FLC system. Thirdly, research on the topic of FLC robustness is still in its initial stage and a more complete investigation of the robust performance of FLC systems is not yet available.

2.7 Summary

In this *Chapter*, the advantages and disadvantages of FLC systems were summarised first. Then the basic concepts of fuzzy set and fuzzy logic were presented to provide a theoretical background of this research work. The principle of FLC was illustrated in five functional blocks: mapping, fuzzification, reasoning, rule base, and defuzzification. Different design methods for FLC systems were reviewed from the point of view of parameter (membership functions and scaling factors) selection and rule derivation. A literature review of the studies on the robustness of FLC systems indicated that the research on the topic of FLC robustness is still in the initial stage and work is required to provide a design method for robust fuzzy control systems.

3

Robust Performance of Fuzzy Logic Control

3.1 Introduction

Fuzzy control, as one of the most successful applications of fuzzy set theory, has been intensively developed during the last twenty years. A large number of publications have covered theoretical aspects of fuzzy control systems and various application fields ranging from engineering to financial systems. Fuzzy logic control has been claimed in many papers to exhibit a very good robust performance under system uncertainties. However, no work has been found to give a thorough investigation of robustness of FLCs. The main reason for this may be the lack of a generally applicable analytical capability to explain the dynamic performance of fuzzy systems.

As the most desirable characteristic of a control system with system uncertainties, the robustness property of fuzzy control systems deserves a complete investigation, especially the most widely used fuzzy control system [85], namely the Mamdani type FLC system using Mamdani reasoning method (or Max-Min reasoning method) described in *Chapter* 2. In this chapter, the aim is to provide a qualitative analysis of the robustness of Mamdani-type FLC systems, supposing that this fuzzy logic controller is used to control several kinds of processes with parameters varying over a relatively wide range. Also, the effects of controller parameters on the system robustness will be examined, so as to pave the way to a systematic design method for robustness. Note that in this thesis, the Mamdani type FLC will be simply called a FLC.

As indicated in *Chapter* 1, robust performance is a more important characteristic because system performance is usually the major design requirement. Therefore, robust performance will be the focus of this research; the term *robustness* will be used to indicate *robust performance* and specifically attention will be restricted to step performances in the time domain.

This chapter is organised as follows. *Section* 3.2 will define the measurements necessary to characterise the dynamic performance and system robustness of a control system from the engineering point of view. *Section* 3.3 will describe the FLC system studied in this work. In section 3.4, the robustness of the FLC system will be investigated, a qualitative analysis method will be developed and applied to first-order and second order systems. *Section* 3.5 will discuss the effects of controller parameters on system robustness. *Section* 3.6 presents design methods for FLC systems with improved robustness. Conclusions will be drawn in *Section* 3.7.

3.2 Performance measure and robustness measure

It is important that a suitable measurement method for system robustness is defined, especially when performing a comparison between two systems. In this section, some basic concepts related to the robustness measure will be presented.

3.2.1 Performance measure

There are two methods often used for evaluating the performance of a control system: the time domain method and the frequency domain method. In practice, time-domain performance measures are usually preferred because control systems are inherently time-domain systems and transient responses to step inputs can be easily obtained. Therefore, a time domain measure for system performance will be used in this work.

In the time domain, system performance is often evaluated by following standard measures: *overshoot (OS), undershoot (US), settle time (ST),* and *rise time* (*RT*). These are defined from the step response of a system as shown in Fig.3-1 [1]. In Fig.3-1, w is the setpoint or reference input of the control system (w is also the magnitude of the input step), y the output, Δy_1 the difference between the maximum output and the setpoint, Δy_2 the difference between the minimum downward output and the setpoint, RT the time required for the system output to rise to the setpoint, and *ST* the time required for the system output to settle within the steady state zone (w ± δ), where δ is a percentage of the setpoint change, *w*. Normally, the overshoot and undershoot are presented as percentages of the setpoint as

$$OS = \frac{\Delta y_1}{w} \cdot 100\% \tag{3-1}$$

$$US = \frac{\Delta y_2}{w} \cdot 100\% \tag{3-2}$$

Rise time will not be used in this work because settle time can indicate not only the swiftness of the system response as rise time does, but also the closeness of the response to the desired value. Both overshoot and undershoot will be used in this work due to the non-linearity of fuzzy control systems where the undershoot feature may not be reflected by the commonly used overshoot specification.

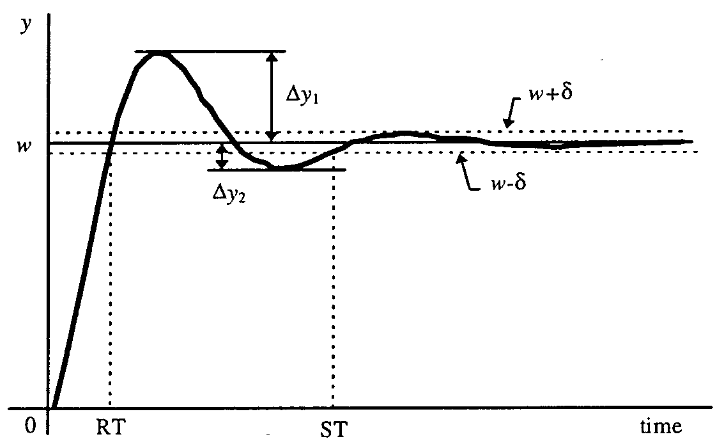


Fig.3-1 Definitions of standard time-domain measurements

It should be noted that a system performance satisfying the requirements of the above evaluations may not be acceptable because it is too oscillatory. This fact suggests that another evaluating term — *ring cycle number (CY)* should be adopted to characterise an oscillatory response. The ring cycle number *CY* is defined as the oscillation cycles of the output *y* around the setpoint when *y* is outside the steady state zone. In summary, *OS, US, ST,* and *CY* will be used as measures to evaluate a system's performance.

In addition, among these performance measures, *ST* is dependent on the controlled processes, *i.e.* the larger the time constant of the controlled process, the longer it takes for the control system to settle within the steady state zone. Care is required when specifying the settling time, especially when a large range of process parameter variation is expected.

In general control systems where robustness is not the main concern, the ratio $\frac{ST}{T_\Sigma}$, where T_Σ is the sum of all time constants in the process at the tuning point, is often used as the indicator of the swiftness of system response. This measure is suitable for non-time-delay systems whose parameter variation can be neglected. However, it is not adequate when process parameters, especially the time constant, vary in large ranges. For example, for a temperature control system in a poultry brooding incubator [86], the time constant of the process in the early brooding stage is much larger than in the late brooding stage [98], because the poultry eggs must receive heat in the early stage while they are capable of producing a certain amount of heat energy in the late brooding stage. A fixed settling time specification, say 30 minutes, will be too tight for the early brooding stage so that large overshoot can be produced, or it will be too loose in the late brooding stage so that slow response can be expected.

This research has used a settling time specification which takes the current process parameters, not necessarily those at the tuning point, into account. That is, whatever process parameter varies, the ratio of the settling time ST to the sum of all current time constants, T_Σ, will be used as the indicator of the swiftness of system response. Characterising the settling time by means of the above method, a fixed ratio of ST and T_Σ can be used as the settling time specification and it will be denoted as R_{st}. In this research, the system settling time has been characterised by this method.

3.2.2 Robustness measure

As indicated in *Chapter* 2, the investigation of robust performance of a control system is concerned with the parameter range of the system within which the system performance meets the required specifications. The parameters affecting system robustness can be either controller parameters, or the controlled process parameters, or both. Robust parameter range is defined as follows:

Definition 3.1: If a system's performance meets the required specifications when parameters A_i (i = 1, 2, ..., n) in the system vary in the range (a_i, b_i), then the parameter range (a_i, b_i) is called the *robust range* of parameter A_i.

The robust range can be measured by the width of this parameter range $\Delta A_i = b_i - a_i$. The product of all robust ranges is called the *robust space* denoted by S. That is,

$$S = \prod_{i=1}^{n} \Delta A_i \tag{3-3}$$

The robust space S can serve as a measure of system robustness. For example, for a first order system with the following transfer function,

$$G(s) = \frac{K}{1+Ts}\ , \tag{3-4}$$

the robustness can be characterised by the robust space $\Delta k \times \Delta t$, where Δk and Δt are the robust ranges of system parameters K and T respectively.

To compare robustness of two control algorithms for the same control task, the required robust space and the robust space achievable by each control algorithm should be considered. Hence the following definition is necessary.

Definition 3.2: Assuming S_a and S_r are the achievable robust space and required robust space of a control system respectively, then system robustness R_s can be expressed as

$$R_s = \frac{S_a}{S_r} \tag{3-5}$$

As an illustration, let the required variation range of K and T in the above example be ΔK and ΔT respectively; then we have

$$S_a = \Delta k \times \Delta t$$

$$S_r = \Delta K \times \Delta T$$

$$R_s = \frac{S_a}{S_r} = \frac{\Delta k \times \Delta t}{\Delta K \times \Delta T}$$

In practice, system performance is usually tested at a finite number of points within the process parameter space concerned. In this case, the achievable robust space can be related to the number of tested points at which the system performance meets the performance requirements. These test points can be simply called *good points,* and the number of these good points is denoted by N_{gp}. The required robust space can be replaced by the total number of tested points N_{tp} in the parameter space. Then the system robustness R_s becomes

$$R_s = \frac{N_{gp}}{N_{tp}} \tag{3-6}$$

Clearly the larger the S_a or N_{gp} value, the more robust the control system.

The advantages of the above method for presenting system robustness are that (1) it clearly shows the size of the robust space relative to the total concerned parameter space; (2) it can be easily implemented in computer simulations. However, this measurement cannot indicate the gradual variation of system performance when system parameters are changed, because it simply measures the size of the robust space and only counts the performance as *in* or *out of* specification.

To study the effect of a system parameter on robust performance, or to optimise system robustness by changing certain controller parameters, a suitable measure of system robustness is needed to reflect the gradual change of system performance with any system changes. The widely used performance index in control systems is the *integral of square of error* (*ISE*) which is defined as

$$ISE = \int_0^T e^2(t)dt \tag{3-7}$$

where $e(t)$ is the difference between setpoint and the output, T is a finite time chosen somewhat arbitrarily so that the integral approaches a steady-state value.

From the definition of *ISE*, it can be found that *ISE* is dependent on the setpoint w and process dynamics. In the step input case, this dependency can be removed by the following normalisation, where w is the setpoint of the control system;

$$ISE = \frac{1}{w^2 T}\int_0^T e^2(t)dt \tag{3-8}$$

However, *ISE* does not take performance specifications into account. For example, performance *A* and *B* in Fig.3-2 have the same *ISE* value, but performance *B* may not be acceptable in practice due to its high overshoot. For the purpose of studying system robustness, the performance index should include information related to the performance specification. Therefore, the following definitions are made.

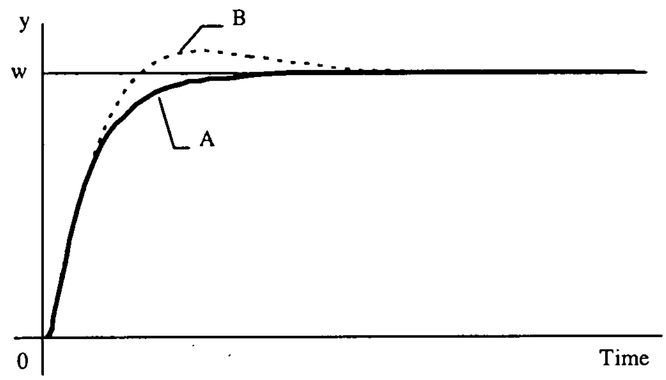


Fig. 3-2 Two performances with same *ISE* value

Definition 3.3: The *performance index* P_{index} of a control system was defined from experimental experiences as

$$P_{index} = OS + US + CY + ISE\times10 + R_{st} \tag{3-9}$$

where *OS, US, CY* and *ST* are the time-domain measurements described in section 3.2.1, and *ISE* is the normalised integral of square of error defined in (3-8). Note that using ten times the *ISE* in the P_{index} ensures that all items contained in the P_{index} will be of equal importance. Also, R_{st} reflects the swiftness of response with respect to the system dynamics. Clearly a smaller P_{index} reflects a better step performance. P_{index} provides a continuous measurement of the system performance.

Definition 3.4: Assume system parameters vary within a finite parameter space containing *n* test points with performance index $P_{index}^{(i)}$ at test point *i* (i = 0, 1, ..., n), and there are *m* test points at which the system performance meets the pre-defined specification. Then the *robustness index* I_{rb} of the system within that parameter space is defined as:

$$I_{rb} = n - m + \frac{10^{-3}}{n}\sum_{n} P_{index}^{(i)} . \tag{3-10}$$

From the definitions of I_{rb}, It can be found that $\frac{1}{n}\sum_{n} P_{index}^{(i)}$ in the definition is actually the average performance index, and it provides a continuous measure of the system robustness, which is important when attempting to obtain a fine tuning of the evolved system performance. (n - m) is the number of tests in which the system performance was not satisfied. In addition, the average performance index is made to play a small part in the robustness index by multiplying by 10^{-3}, so that the number of unsatisfied tests is addressed in the robustness index. Clearly the smaller I_{rb}, the more robust the system. I_{st} provides a continuous measurement of system robustness within the parameter space of interest.

3.3 Fuzzy control system

A block diagram of the fuzzy control system used in this research is shown in Fig.3-3, where w is the setpoint, y is the output of the process, u the control input, and P the process to be controlled. The FLC system was considered as a discrete system, and the accumulator, if required, was assumed to be included within the FLC. Error e and derivative error $\dot{e}$ were calculated at instance k as:

$$e(k) = w(k) - y(k) \tag{3-11}$$

$$\dot{e}(k) = \frac{e(k) - e(k-1)}{t_s} \tag{3-12}$$

where t_s is the sampling interval of the controller. The inputs and the output of the controller were limited to lie within the range (-10, 10) to simulate the saturation characteristic of practical systems. To conform with normal practice and to minimise overload of the controller input, the derivative error was obtained from the process output, *i.e.* $\dot{e} = -\dot{y}$, in all simulation experiments.

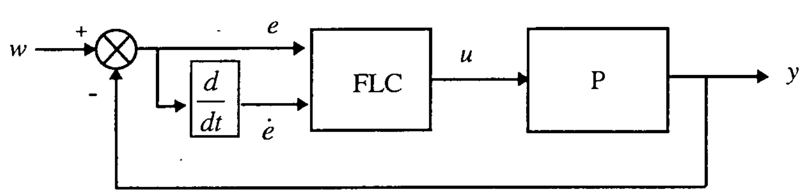


Fig.3-3 Structure of the fuzzy control system

It should be noted that the SISO control systems were the main consideration in the simulation. The fuzzy controller was chosen to be the Mamdani-type FLC. Three symbols, e, $\dot{e}$ and Δu, ware also used to denoted the corresponding fuzzy variables. Seven fuzzy sets were chosen for each fuzzy variable and named as NB, NM, NS, ZE, PS, PM, PB. Membership functions of each fuzzy variable were defined as the triangle shape as shown in Fig.3-4.

Before performing fuzzification, non-fuzzy values of two input variables were multiplied by the input scaling factors to adjust the sensitivity of the controller to the variables, and the results were limited to the designed interval (-10, 10). After defuzzification, the non-fuzzy control change Δu was re-scaled and accumulated if needed. The results were limited to the practical range of the control variable u. The scaling factors were gain of error, gain of change of error, and gain of control output, denoted by *GE, GC*, and *GU* respectively.

The fuzzy rule base provides an inference mapping from the input fuzzy universe $E \times C$ to the control output fuzzy universe U and takes the form of linguistic conditional statements:

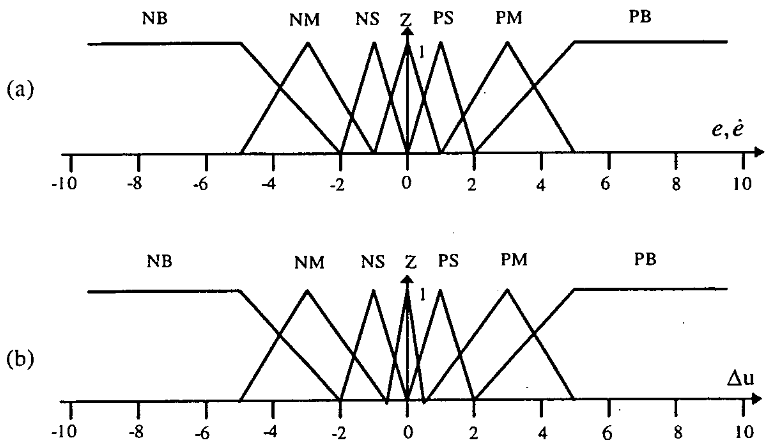
IF e is E_k and $\dot{e}$ is C_k , THEN Δu is U_k ,

where E_k, C_k and U_k are fuzzy sets, or linguistic labels, defined in the universe E, C and U respectively. Every rule expresses a single control action at the stated system situation. The entire fuzzy rule base is usually represented by a matrix called the fuzzy inference matrix (FIM). Fig.3-4(c) shows the FIM used in the simulation study.

The COG method was used to defuzzify the output fuzzy variable in the simulation. An accumulator is inserted after the FLC to calculate the control input to the process if there is no integrator in the process to be controlled. Fig.3-5 shows the basic structure of a FLC.

The membership functions and the FIM in Fig.3-4 were heuristically designed based on general experiences obtained in controlling linear SISO processes. The control heuristic can be illustrated in the phase plane as shown in Fig.3-6. In phase 2 (or phase 4) where the process output is moving away from the set-point, the control input to the process should be decreased (or increased) proportionally to the magnitude of e and $\dot{e}$ to drive the output back to the set-point. In phase 1 (or phase 3) where the output is moving toward the set-point, the control action should be carefully decided according to the change of the output so as to keep the output moving at a designed speed without a big overshoot (or undershoot).

It should be note that the FLC described above is widely used in fuzzy control applications, and the scaling factors are tuned in practice to obtain the required system performance. It must be realised that the results produced by this FLC are conservative; better results may be obtainable by using specific rules for a specific control task.



(c)

e \ ė	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	PM
NM	NB	NB	NB	NS	NS	ZE	PM
NS	NB	NB	NB	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PB	PB	PB
PM	NM	ZE	PS	PS	PB	PB	PB
PB	NM	PS	PM	PB	PB	PB	PB

Fig.3-4 Membership functions (a)(b) and FIM (c) used in the FLC

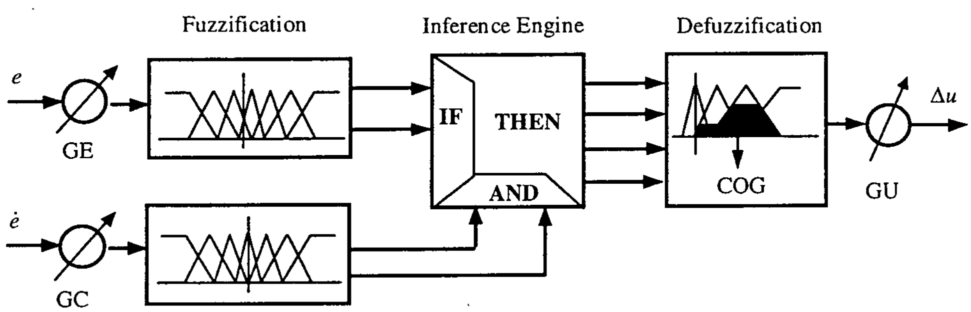


Fig.3-5 Basic structure of Mamdani-type FLC

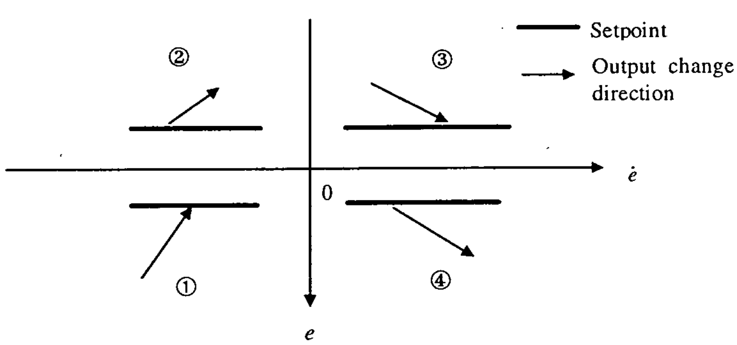


Fig.3-6 Phase plane of a control system

The basic principle of the operation of the controller is described as follows. For the rule base shown in Fig.3-4(c), the change of control input Δu is calculated as:

R_i : if e is A_i and $\dot{e}$ is B_i, then Δu_i is C_i ($i = 1, \cdots\cdots 49$)

$$\alpha_i = \mu_{A_i}(e) \wedge \mu_{B_i}(\dot{e}) \tag{3-13}$$

$$\Delta u_i = cog(\alpha_i \wedge \mu_{C_i}) \tag{3-14}$$

$$\Delta u = \frac{\sum_{i=1}^{49} \alpha_i \Delta u_i}{\sum_{i=1}^{49} \alpha_i} \tag{3-15}$$

where μ_{A_i}, μ_{B_i}, and μ_{C_i} are membership functions of A_i, B_i and C_i respectively, cog(·) is the defuzzification function with the centre-of-gravity method, α_i is called the *degree of firing* (DOF) of R_i. Note that cog(·) can be obtained by performing the following operation on the membership function of $(a_i \wedge \mu_{C_i})$, say $F(u)$:

$$cog(F(u)) = \frac{\int_u uF(u)du}{\int_u F(u)du}. \tag{3-16}$$

3.4 Robustness of fuzzy control systems

In this section, the robustness of the fuzzy control system will be investigated, a qualitative analysis method will be developed and then applied to analyse the robustness of FLC controlled first-order and second order systems.

3.4.1 Qualitative analysis

A. Switching line Characteristics

It has been recognised that human operators can regulate practical plants satisfactorily based on their experience of controlling the plant. Control is maintained independent of model variations if the model belongs to the same plant family. With reference to this fact, the fuzzy controller should be expected to inherently possess some robustness since its role is to imitate human control heuristics. Close examination of the input-output mapping, (*e*,

$\dot{e}$) $\Rightarrow \Delta u$, determined by the fuzzy control rule base in Fig.3-4c reveals the following characteristics:

(i) The *input-output mapping, (e, $\dot{e}$) $\Rightarrow \Delta u$, has an output fuzzy-zero line (not necessarily straight) called the* ***switching line****, separating identically signed output fuzzy variables. The larger the distance between a system state (e, $\dot{e}$) and the switching line, the greater will be the control output change.*

(ii) T*he fuzzy controller always tries to drive the system state (e, $\dot{e}$) in the phase plane to the equilibrium state (e = $\dot{e}$ = 0) along the switching line. On the switching line, the change of error, $\dot{e}$, is only dependent on the error e, and the bigger the error, the faster it changes.*

Character (ii) can also be interpreted as follows;

For any given system error e, the FLC will try to drive the system so that the system state (e, $\dot{e}$) is on the switching line.

The above characteristics are called the switching line characteristics. These characteristics cause a behaviour similar to that of a sliding mode controller (SMC) [88] [61] [101] [102] [103]. Because of this sliding mode behaviour, the robustness is inherent in the FLC systems according to the sliding mode theory. When the sliding mode occurs on the switching line, the system remains insensitive to external disturbances and plant uncertainty. The first characteristic enables the desired trajectory in the phase plane to reach the switching line in finite time. The second characteristic presents the ability of the FLC to drive the desired trajectory to asymptotically go to the origin of the phase plane along the switching line.

Compared with an ordinary SMC, however, there are two differences between fuzzy control and sliding mode control. First, the FLC can provide a flexible input-output mapping function which is more adaptable to the state space of the system to be controlled [27]. Second, control actions of SMC often discontinuously change whenever the phase plane trajectory crosses the switching line; while the variation of FLC's control action is continuous. Moreover, the FLC can make use of the linguistic information of the system by means of fuzzy control rules.

Suppose $f(e, \dot{e})$ is the function describing the control surface generated by the FLC. Then the switching line L can be expressed as;

$$L = \{ (e, \dot{e}) \mid f(e, \dot{e}) = 0 \} \tag{3-17}$$

If the membership functions of e and $\dot{e}$ are symmetrically distributed with respect to their origins ($e = 0$ and $\dot{e} = 0$) respectively, then the switching line corresponding to the rule base in Fig.3-4c is as shown in Fig.3-7. The system states on the switching line determine a system performance called the ***switching line performance***. The location of the switching line, and thus the performance corresponding to this switching line, is determined by the locations of all membership functions, the rule base, and the scaling factors.

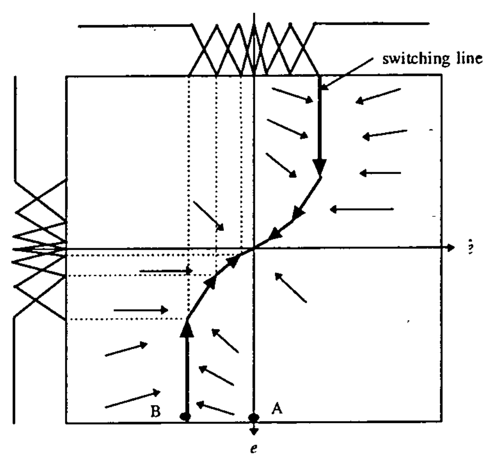


Fig.3-7 Switching line corresponding to the rule base in Fig.3-4c
(arrows indicate the movements of system states).

Obviously the switching line characteristic determines the phase trajectory of the fuzzy control system. The control objective of the FLC is to maintain the system states on this trajectory by regulating the error change rate, $\dot{e}$. *The switching line performance will dominate the dynamic behaviour of the system if the controller can drive the phase trajectory to follow the switching line to the origin of the phase plane*, *i.e.* the ***sliding mode*** occurs. That is, during the sliding mode, $f(e, \dot{e}) = 0$ effectively defines the **transfer function** of the FLC [10].

Note that due to the effect of the dynamics in the controlled process, $\dot{e}$ cannot change suddenly from one initial state, say point A in Fig.3-7, to point B on the switching line.

Practical performance will be different from the designed switching line performance, especially in the early stage of the response. If most parts of the trajectory of a system response can overlap the switching line, this response will be generally considered as the switching line performance. Fig.3-8 shows some experimental results of phase-plane trajectories of the system states, which relate to several first-order systems with different process parameters shown on the figure. In Fig.3-8, all trajectories are very close to each other except for the parts corresponding to large error signals, though the process parameters vary significantly. Also it can be found in the figure that two trajectories oscillate around the switching line at the early stage of the response and finally converge to the switching line. Clearly the FLC possesses an ability to control systems with different dynamics to give a performance similar to the switching line performance.

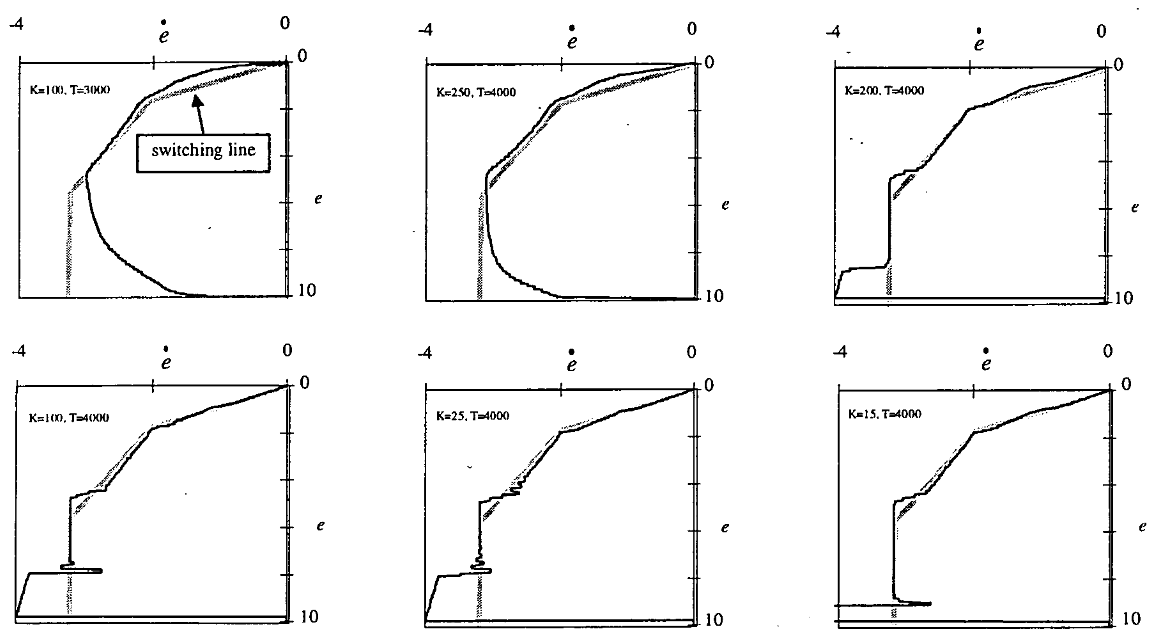


Fig.3-8 Experimental results of the phase-plane trajectories near the switching line.

B. Robust space

Like the SMC system, the transient dynamics of the FLC system consists of two conditions: a *reaching condition* and a *sliding condition*. Under the reaching condition, the desired response aims to reach the switching line in finite time. Under the sliding condition, the trajectory asymptotically goes to the origin of the phase plane along the switching line. When the parameters of the FLC are determined, the reaching condition of the phase trajectory sets a lower limit on the response dynamics of the controlled process, and the sliding condition sets an upper limit on the process response speed. Within these limited dynamics, the switching line performance can be obtained. On other hand, any change in the FLC parameters could lead to a different position or shape of the switching line, and thus change these dynamic limits on the controller process. Clearly, both conditions determine the robustness of the FLC system. Since they are closely related to the FLC's performance and the process' dynamics, further development of these conditions have to be based on the controlled process and the position of the switching line (see *Section* 3.4.2).

Considering that the performance specifications allow a range of system performance to be acceptable and the system performance at the tuning point, say P_0, is tuned to meet the performance specifications, there must be a process parameter space, say ϕ containing P_0, within which the system dynamic performance is in the performance specifications. Normally, the switching line performance is the designed system performance which will be realised at a selected tuning point P_0. System performance will deteriorate if the system operating point drifts away from the tuning point. Therefore, the size of the robust space ϕ indicates the ability of the FLC to cope with parameter variations.

Clearly, the robustness of this type of fuzzy control systems will be determined by the FLC's ability to control the system state trajectory to follow the designed switching line on the phase plane under the variation of system parameters. Fundamentally, there are two aspects which can affect this capability of the FLC. The first one is the switching line performance because it directly determines the quality of the system response. If the performance related to the switching line is just within the performance specification, then it can be expected that the system performance will quickly go out of specification when the system parameter changes. For example, if the switching line performance produces a 5% overshoot which is the maximum overshoot required by the specification, then increasing the gain in a first-order process will increase overshoot in the step response, thus the performance will go out of specification.

The second aspect influencing the robustness of the FLC system is the smoothness of the control action on both sides of the switching line. A smooth transform of the control action can improve the system's stability, but is less able to bring the phase plane trajectory to the switching line, since the smooth transform exhibits a relative low controller gain. On the other hand, a harsh jump in the control change Δu, when the phase plane trajectory crosses the switching line, will lead to close conformity of the phase plane trajectory to the switching line, but oscillations in the system output and control input might occur if the process gain is high.

With respect to the principle of the FLC, the switching line performance of the FLC system will be determined by the location of the switching line on the phase-plane, and the smoothness of the control action on both sides of the switching line will be determined by the rate of change of the control surface. The switching line and the control surface can be affected by the controller parameters, *i.e.* the membership functions, scaling factors and the rule base. These will be discussed in the next sections.

Unfortunately, the size of the robust space cannot be analytically determined because (1) it completely depends on the controller parameters; and (2) the high non-linearity of the fuzzy system makes the analysis very complicated. However, in some specific cases, it may be possible to qualitatively predict the shape of the robust space, as will be shown in the next section.

3.4.2 Robustness of the first order process

The first-order process

$$G(s) = \frac{K}{1+Ts} \tag{3-18}$$

is the widely used process model to approximate practical processes in analysing and designing a control system. Hence it will be used here to develop a qualitative analysis of the robustness of FLC systems.

Assume K and T in process G(s) are parameters that can vary over a relatively wide range. If the system is tuned at the operating point (K_0, T_0) in the K-T plane and produces the switching line performance at this tuning point, then, based on the argument in the last section, there must exists an area ϕ around (K_0, T_0) in the K-T plane within which the variation of K or T does not significantly affect the closed loop performance.

Suppose that at time t, process input and output are $u(t)$ and $y(t)$ respectively. For a very small time increase Δt, the output $y(t+\Delta t)$ of the process (3-18) can be expressed as

$$y(t+\Delta t) = y(t)e^{-\Delta t/T} + K(1-e^{-\Delta t/T})u(t) \tag{3-19}$$

Suppose $\Delta t \ll T$. Then (3-19) can be approximately written as

$$y(t+\Delta t) = y(t)(1-\frac{\Delta t}{T}) + K\frac{\Delta t}{T}u(t) \tag{3-20}$$

$$\frac{y(t+\Delta t)-y(t)}{\Delta t} = \frac{K}{T}[u(t)-\tilde{u}(t)], \tag{3-21}$$

where $\tilde{u}(t) = \frac{y(t)}{K}$ denotes the equivalent control input of $y(t)$. Let $\Delta u_{eq} = u(t) - \tilde{u}(t)$, called the equivalent control change. From the definition of the change of error $\dot{e}$ in (3-12), if Δt is sufficiently small, then we have

$$\dot{e}(t) = -\dot{y}(t) \approx -\frac{K}{T}\Delta u_{eq} \tag{3-22}$$

From (3-22), it is clear that if the FLC controlled process is of first order, the error change $\dot{e}$ at any time is proportional to the process parameter ratio (K/T) and the equivalent control change Δu_{eq}. If Δu_{eq} is constant, then all points on the line $K=aT$ (a is a constant) in the K-T plane will present a common phase plane trajectory for a given initial error. If (K/T) is constant, the equivalent control change Δu_{eq} can be regulated by the FLC to keep the system state $(e, \dot{e})$ on the trajectory determined by the switching line.

Because of the inevitable limitation of the FLC output, there will be a limited range of process parameters with which the system phase plane trajectories can be brought to or near the switching line. This problem becomes most critical at the beginning of a step input response, because the response at that time is required to be fast enough to meet the performance specifications. Suppose that $|\dot{e}|_{min}$ is the minimum response speed at the beginning of a step input response for an acceptable performance, and $|\Delta u|_{max}$ is the maximum control change from the FLC. Then from (3-22), for an acceptable performance, the process parameters should meet the condition

$$\frac{K}{T} \geq \frac{|\dot{e}|_{min}}{|\Delta u|_{max}}, \tag{3-23}$$

otherwise, the system performance will be slower than the specified response. Note that at the beginning of a step input response, $|\Delta u_{eq}|_{max} = |\Delta u|_{max}$.

On the other hand, because of the switching line feature of FLC systems, fast system response or large control change may lead to oscillations of the phase plane trajectory around the switching line, so that the system cannot reach the steady state. For a FLC system with all controller parameters fixed, the maximum dynamic response speed of the controlled plant will be limited. That is, the response speed of the plant should be so slow that any non-zero control change $|\Delta u|$ cannot drive the system to the unstable state. Therefore, there will be a limited range of process parameters, within which the motion speed of any phase plane trajectory maintains the above restriction.

In practice, a certain amount of fluctuation of the phase plane trajectory around the switching line will generally be accepted. That is, the controller's output may vary in a certain bounded range to force the system phase plane trajectory to slide on to the switching line. Let Δu_{sl} indicate this bounded control change. Normally $|\Delta u_{sl}|_{max} \leq |\Delta u|_{max}$. Suppose that the maximum speed change is $|\Delta \dot{e}|_{max}$. Then the upper limit of the process parameters can be expressed as

$$\frac{K}{T} \leq \frac{|\dot{e}|_{max}}{|\Delta u_{sl}|_{max}}, \tag{3-24}$$

The condition (3-23) provides a reaching condition for the system phase plane trajectory to reach the switching line; and (3-24) provides a sliding condition for the system phase plane trajectory to slide along the switching line. In practice, it is not necessary for the trajectory to be exactly on the switching line. These two conditions present a general definition of the robust space of a FLC controlled first order system and they are represented by a sector on the K-T plane, as shown in Fig.3-9.

As expected from conventional control theory, the sample rate of the FLC will affect system robustness. If the sample rate is fixed, its effect will becomes significant when the process time constant T is comparatively small or large. For example, if T is relatively small, the system response is fast, and the response of the FLC becomes relatively slow so that it may be too late for the controller to change its control action before the performance becomes poor. When T is large, the system response is slow, and the response of the controller becomes relatively fast. The fast response will lead to too much control action so that the system can be easily over-driven or saturated, thus large overshoot or undershoot can be produced. Therefore, for a FLC with fixed parameters, there will be a limited range of time constant (T_{min}, T_{max}) of the controlled plant, outside which the system performance will be significantly impaired. The parameter regions corresponding to this limitation are illustrated as area 3 and area 4 as shown in Fig.3-9.

In addition, because the response speed in the transient period is limited by the switching line performance, the transient performance may not meet the current settling time specification in some situations (see *Section 3.2.1* for R_{st} definition). For example, a small time constant process will expect a faster response to meet the settling time requirement. So, there will be a low time constant area (area 5 in Fig.3-9) within which system performance will be out of the performance specifications due to the switching line characteristic, though its step input performance is the same as the switching line performance.

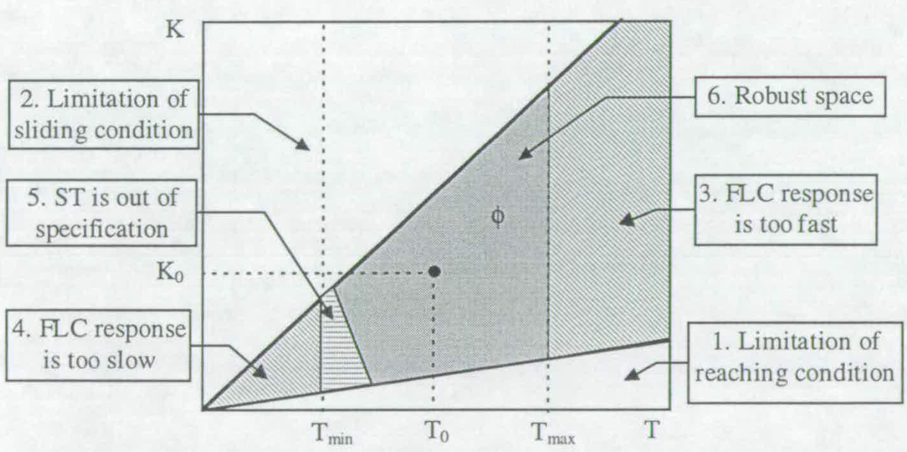


Fig.3-9 Illustration of the robust space of the FLC system with first-order process

If the system contains a time delay, or is of high order with one dominant time constant, the system robustness will possess the same general features as that of the first order system, but the size of the robust space will be significantly affected. As expected with conventional control theory, the system becomes unstable if high gain is used in both situations. This will result in a narrowed robust space in the gain K direction. In addition, the time delay amplifies the effect of saturation or over-drive in dynamically slow systems, and leads to big overshoots or undershoots. Thus the robust space will be reduced in low gain and large time constant situations. Although the effect of the time delay is expected to decrease when time constant is increased, it is also related to the sampling rate of the controller. For example, experiments show that if the time delay is more than four times the sample period, obvious increase of overshoot or undershoot in the system performance will be observed, no matter how large is the time constant.

Unfortunately, non-linear system theory cannot, as yet, offer techniques that could be used to quantitatively define the robust space. General comments can be made, such as, $|\Delta u|_{max}$ is related to the membership functions of Δu and GU, while $|\dot{e}|_{min}$, $|\dot{e}|_{max}$, T_{min} and T_{max} are mainly affected by location of the switching line on the phase plane. But, any change in the membership functions or rule base may affect these parameters.

3.4.3 Robustness of the second order process

The second order process is defined by the transfer function;

$$G(s) = \frac{\omega_n^2 K}{s^2 + 2\zeta\omega_n s + \omega_n^2}. \tag{3-25}$$

It has characteristic features different from that of the first-order process. It is known that changes in the parameters of a second order process can change the form of the response as well as the response speed, while parameter changes in the first-order process only affect the speed of the response. A second order process can display damped characteristics much like a first-order process or pure oscillations in its transient response, depending on the damping ratio ζ.

When $\zeta \geq 1$, a second order process has two real poles

$$r_{1,2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}, \tag{3-26}$$

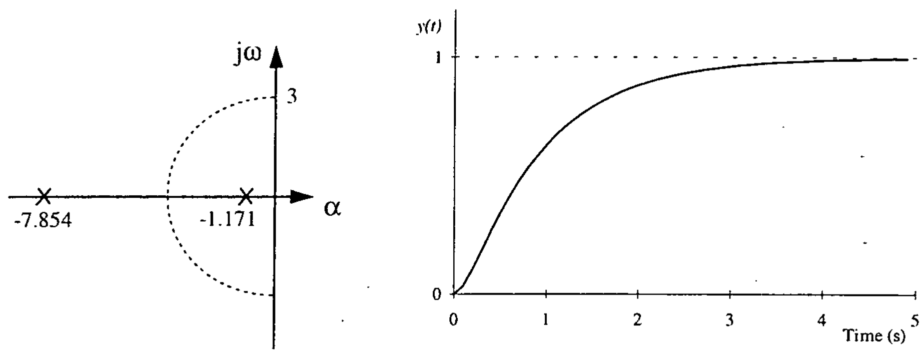
and its transient step response is formed from two exponentials with time constants equal to the reciprocal of the pole locations (see Fig.3-10).

As ζ increases, one pole moves away from the origin of the s-plane and another moves towards to the origin, and the exponential corresponding to the pole near the origin becomes more and more dominant in the response. Generally, when ζ >1.5, the second order process can be approximated by a first-order process with pole at

$$r = -\zeta\omega_n + \omega_n\sqrt{\zeta^2 - 1}\,. \tag{3-27}$$

The corresponding time constant is

$$T = \frac{1}{\omega_n(\zeta - \sqrt{\zeta^2 - 1})}\,. \tag{3-28}$$



(a) (b)

Fig.3-10 Second-order process (ζ = 1.5, ω_n = 3). (a) pole location; (b) step response.

It can be found that T is proportional to ζ and ($1/\omega_n$). Thus, the robustness of the FLC controlled second order process will be similar to that of the first order process in case of ζ > 1.5. That is, the robust space will approximately be a sector area in the K — ζ plane (with constant ω_n) or in the K —($1/\omega_n$) plane (with constant ζ).

When $\zeta \leq 1.0$, the effect of ζ on the robust space of the system is approximately equivalent to the effect of the time constant in the first-order process. As ζ decreases, both poles will move toward the imaginary axis along a circular route (with radial length equal to ω_n, see Fig.3-11); and thus the step response of the process becomes increasingly oscillatory (OS increases and rise time decreases) and its sinusoidal amplitude will decay more and more slowly. Under the control of a FLC, ζ should be large enough so that the limited control input can overcome the overshoot to bring the system trajectory to the switching line. On the other hand, for a range of possible control inputs, the damping ratio should be sufficiently small to allow the trajectory to reach the switching line. When ζ is small, the closed-loop performance is oscillatory, decreasing gain K can improve system performance. When ζ is large, the response is slow, increasing gain K can help to meet the reaching condition. Therefore, it can be predicted that the robust space of the second order process under variation of the damping ratio is approximately a sector area in the K-ζ plane similar to that of the first-order process.

The effect of ω_n on the system robustness when $\zeta \leq 1.0$ will be different from that of the damping ratio, because in the open-loop step response variation of ω_n leads to a change of response speed, but does not affect the overshoot. As ω_n increases, both poles of the second order process will move away from the origin of the s-plane in the radial direction (ζ constant, see Fig.3-12). Therefore, the frequency of the sinusoidal part of the step response becomes higher, and the response becomes faster (RT decreases).

In the closed-loop situation with a FLC as the controller, the FLC will try to bring the phase plane trajectory to the switching line. When ω_n increases, the FLC will limit the response speed to the speed determined by the switching line. Suppose that the settling time of the switching line performance is T_{sl} and the settling time specification of the second order process is $3/(\zeta\omega_n)$ for the system output to reach within 2% of the steady state value. Then the maximum ω_n to meet the settling time specification is

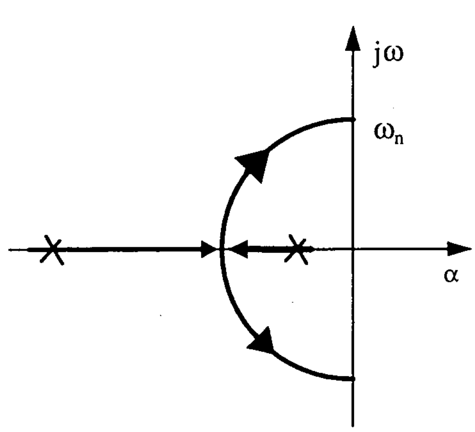


Fig.3-11 Pole trajectories of the second order process when ζ decreased

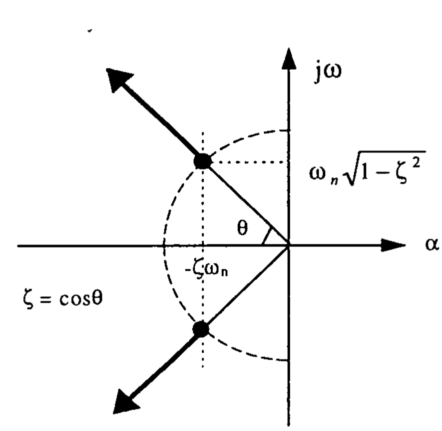


Fig.3-12 Pole trajectories of the second order process when ω_n increased

$$\omega_{n.\max} = \frac{3}{\zeta T_{sl}} \tag{3-29}$$

If ω_n is large, the controlled process produces a fast response to the control action. Because of the switching line feature, the FLC will try to bring the fast response speed down by decreasing the control actions. Hence the process will run at a slower speed in this situation than in the open-loop condition with the same setpoint. In addition, the equivalent time constant, $1/(\zeta\omega_n)$, of the second order process is relatively small if ω_n is large. Thus, the settling time specification for the FLC controlled second order process will be relatively small, because the settling time specification is normally defined as a time period proportional to the time constant of the controlled process. Due to the response speed limitation arising from the switching line feature of the FLC systems, the system cannot meet the settling time specification if ω_n becomes larger than a certain value. In this case, increasing gain *K* can help to increase the response speed without changing the settling time specification. The larger ω_n, the larger will be the gain needed to bring the system performance within the specification. Therefore, the minimum gain K values required to meet the settling time specification at different ω_n will form part of the boundary of the robust space of the second order system on the K— ω_n plane.

As ω_n decreases, both poles of the second order process will move toward the origin of the s-plane in the radial direction. The frequency of the sinusoidal part in the step response becomes slower, and the transient period becomes longer (RT increases). Under fuzzy logic control, the system produces a faster response speed than that of the open-loop second order process, because of the over-drive control action generated from the switching line feature (see Fig.3-13). Thus there is a tendency to generate a bigger overshoot than that of the open-loop system, and the phase plane trajectory will run away from the switching line when the phase plane trajectory crosses the setpoint (point A in Fig.3-13). If the correction control is applied at this time and the controller's dynamic speed is relatively slow, then big undershoot will be produced, as depicted in Fig.3-13. In addition, even if there is a small change in the overshoot or undershoot, the settling time may exhibit a large change because of the oscillatory performance, see illustration in Fig.3-14. In this case, intuitively, decreasing gain *K* can reduce the effect of the over-drive problem. The smaller the ω_n, the smaller will be the gain *K* needed to meet the performance specifications. Therefore, the maximum K values required to meet the specification for overshoot or undershoot at different ω_n will form another part of the boundary of the robust space of the second order system on the K— ω_n plane.

The sampling rate of the controller can also affect the robustness of a system. If the sampling rate is fixed, systems with high gain K or large frequency ω_n will encounter stability problems; and systems with low gain K or low frequency ω_n will have problems associated with accumulator saturation. So, from system theory, there must be limited parameter ranges of K and ω_n which satisfy the stability requirements (maximum K and ω_n) and the performance requirements (minimum K and ω_n).

Low gain can be expected to reduce the overshoot or undershoot when ω_n is small, but low gain will increase the saturation if ω_n is low. This conflict will lead to a narrow band of gain K in this case.

From the above discussions, the robust space of the FLC controlled second order process can be estimated as the shaded area on the K— ω_n plane as shown in Fig.3-15. The lines A and D on the figure form a narrow space in the low gain area. Lines B and F are the maximum limitations of K and ω_n determined by the sampling rate. Line C is the minimum K boundary determined by the settling time specification in the high ω_n situation. Line E is the maximum K boundary determined by the overshoot (undershoot) specification in the relatively low ω_n situation.

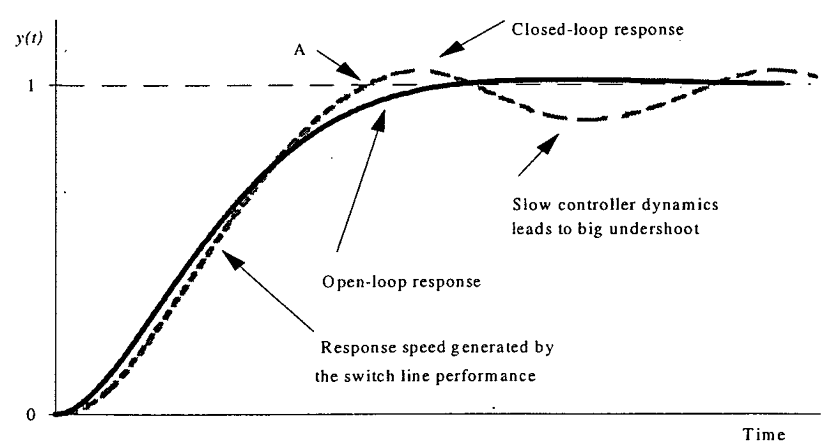


Fig.3-13 Illustration of the system performance when switching line performance is faster than the open-loop response of the second order process

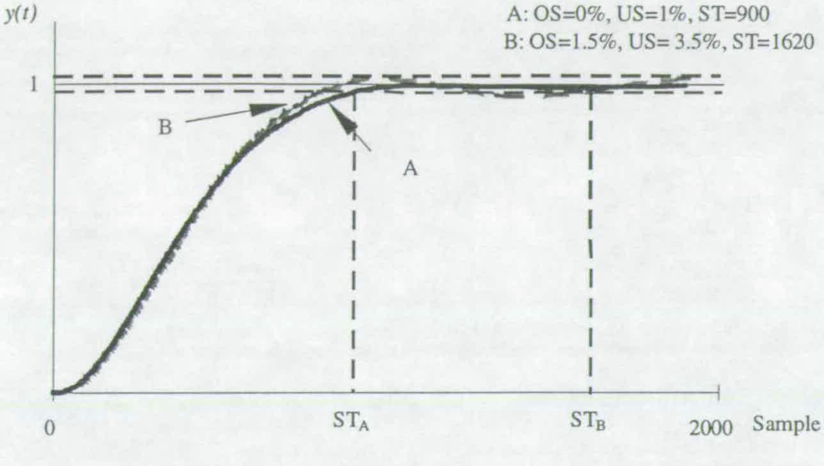


Fig.3-14 Big change in settling time with a small change in OS or US

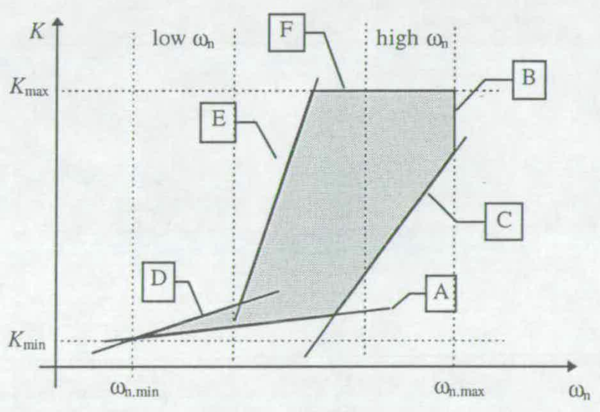


Fig.3-15 Estimated robust space of the fuzzy logic controlled second order process

Finally, it can be concluded that the robustness of the FLC system stems from the switching line feature of the controller. The switching line performance will dominate the dynamic behaviour of the system if the sliding mode occurs. The size of this robust space will be affected by the distribution of rules on both sides of the switching line. If the parameters of the controller are fixed, the switching line feature can tolerate some variation of the process parameters, provided that the reaching condition and the sliding condition are met.

Fundamentally, the reaching condition and the sliding condition are closely related to the sliding ability of the controller, which is in turn related to the maximum drive and the dynamic speed of the controller. Increasing the sliding ability of FLC is achieved by increasing the sampling rate (thus increasing the dynamic speed of the integrator after the

FLC), or by increasing the scaling factor GU (thus increasing the output of the FLC), or both. However, fast sampling or large GU can lead to big and fast oscillation in the control output *u* and thus increase the energy consumption, see illustration in Fig.3-16.

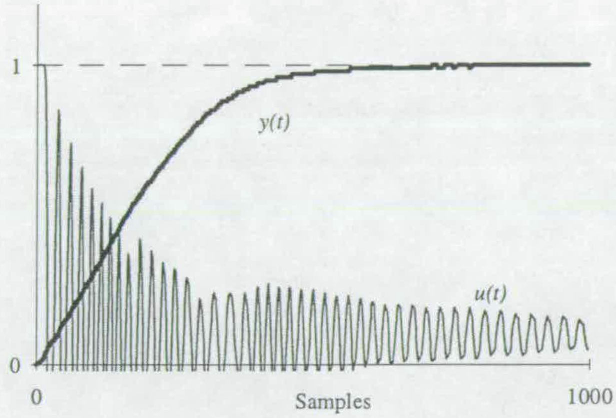


Fig.3-16 System performance of the FLC controlled second order process when sliding ability is increased. (T_s =1, GE =1, GC =100, GU =100. ω_n = 8.0, ζ = 0.8, K = 2.5).

Nevertheless, this qualitative discussion does explain that the robustness of FLC systems is closely related to the switching line performance, and the sliding ability should be increased in the case of a second order system if high robustness is required. However, because of the addition of the integrator (accumulator) in the forward path of the control loop, the robustness of FLC systems will decrease when the order of controlled processes is increased.

3.5 Effects of FLC parameters on system robustness

Since the parameters of the FLC determine the dynamic performance of the controller, they can affect the robustness of the control system. The FLC parameters include rule base, membership functions and scaling factors. In this section, the effects of these parameters on system robustness will be discussed and attention will be paid to the relation between the variation of the controller parameters and the switching line performance.

3.5.1 Effect of fuzzy rule base

The fuzzy rule base contains control strategies under most system states. In the SISO systems, the rule base can be represented as a matrix. To explore the relationship between the rule base and switching line, the rule base will be placed on the phase plane together with the membership functions. For readability, the membership functions will not be drawn to scale and only the dominant rule will be shown for each cell of the phase plane.

Let's first investigate the rule base in Fig.3-4 which is redrawn in Fig.3-17. From the position of the switching line in the phase plane, it can be seen that the error change rate $\dot{e}$ in the transient period is required to be no bigger than the fuzzy MEDIUM level. This structure of the rule base makes it relatively easy for slow dynamic processes to meet the reaching condition and relatively hard for fast dynamic processes to meet the sliding condition. Therefore, the fuzzy control systems with this rule base will be expected to produce a better robustness in the case of slow dynamic processes than in the case of fast dynamic processes.

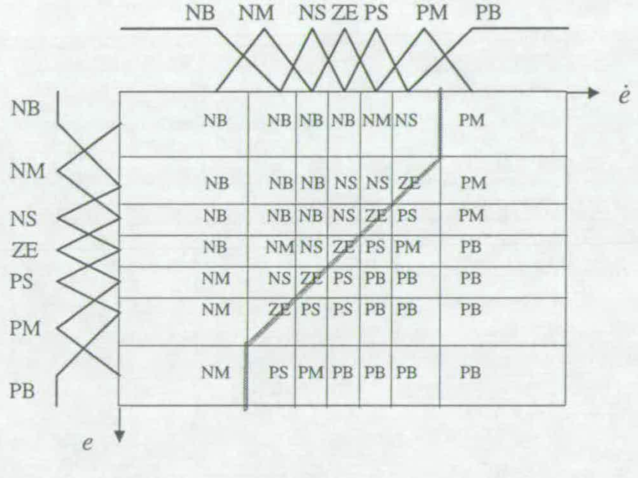


Fig.3-17 Switching line defined by the rule base.

If all membership functions and scaling factors are kept unchanged, the position of the switching line can easily be modified by positioning differently signed control actions in the matrix, *i.e.* changing the consequent part in the fuzzy rules. For example, a symmetrical rule base defines a diagonal switching line in the phase plane, as shown in Fig.3-18. This switching line, defined by the symmetrical rule base, allows a faster response speed in the transient period than that defined in Fig.3-17. On the gray areas in the figure, the controller output is maintained at a constant value and thus becomes insensitive to variations in error and error change.

By modifying the positions of ZERO control actions in the rule base, switching lines with different shapes can be obtained. Fig.3-19 gives two types of switching lines which require smaller response speed in the relatively small error situations than the switching lines defined in Fig.3-17 and Fig.3-18, and produce different responses when the error is relatively big. The rule base in Fig.3-19(a) will be suitable for dynamically slow systems, because the dynamically slow switching-line-performance can reduce system overshoot and undershoot. Similarly, the rule base in Fig.3-19(b) can satisfy fast response requirements with a small overshoot or undershoot.

It can be found that if the angle φ between the *e*-axis and the switching line in the phase plane (see Fig.3-20), called the *switching line angle*, is reduced, the response speed, the system overshoot and undershoot in the switching line performance will be decreased, but settling time will be increased, and *vice versa*. Moreover, small φ may lead to oscillation of the phase plane trajectory around the switching line, especially when the controlled process

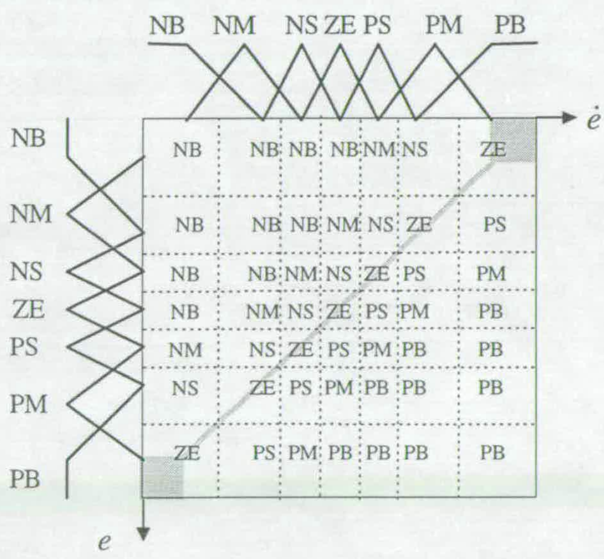


Fig.3-18 Switching line defined by a symmetrical rule base. Note that the gray areas on the switching line indicates $\Delta u = 0$.

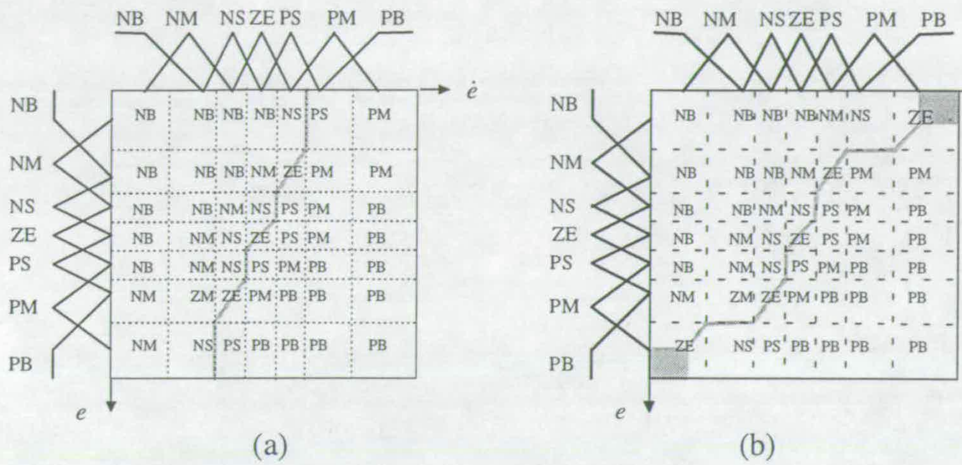


Fig.3-19 Switching lines defined by the modified rule bases which produce slow response (a) and (b) fast response in big error situations

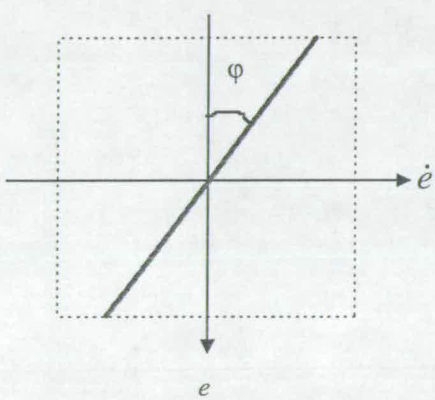


Fig.3-20 Definition of the *switching line angle*

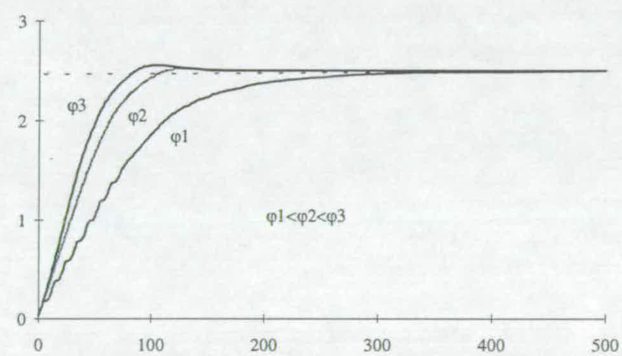


Fig.3-21 Different performances at different switching line angles

is dynamically fast; and big φ may lead to oscillation of system output around the setpoint, specially when the controlled process is dynamically slow. Therefore, unsuitable design of the switching line angle can reduce system stability. Fig.3-21 shows three performances of a first-order process controlled by a FLC with different switching line angles. Note that a switching line may consist of several segments with different switching line angles, which generate different dynamic performances at different system states.

In addition to the effect of the rule base on the position of the switching line, the rule base can also affect the smoothness of the control on both sides of the switching line. For example, if more SMALL control actions (PS and NS) are added to both sides of the switching line in the symmetrical rule base in Fig.3-18, as shown in Fig.3-22, the phase plane trajectory of the system will reach the switching line more slowly than that determined by the rule base in Fig.3-18. A smooth phase plane trajectory can reduce oscillation around the switching line, thus it is suitable for stabilizing high gain systems. However, increasing the smoothness of the phase plane trajectory will lead to a slow response.

In summary, the rule base can affect the position of the switching line and the smoothness of the phase plane trajectory. Because system robustness is closely related to the switching line performance, any changes in the rule base can affect the robust performance. Different control objectives may require different switching line performance. When the controlled process and the switching line performance are decided, system robustness is then determined.

However, the possibilities of changing the switching line performance by modifying the rule base are limited because of the limited number of output fuzzy variables. Due to the possibility of continuous change in the positions and shapes of membership functions, switching lines can be changed to any position with any shape. This will be discussed in the next section,

3.5.2 Effect of membership functions

To examine the effect of membership functions on system robustness, assume that the rule base is symmetrical as shown in Fig.3-18, and the scaling factors are set to 1.0. Fig.3-23(a) shows the switching line corresponding to the membership functions in Fig.3-4(b). Because

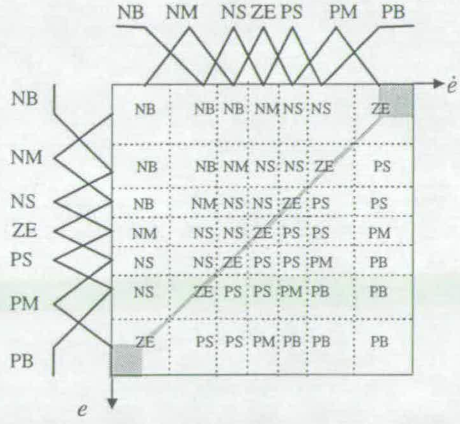


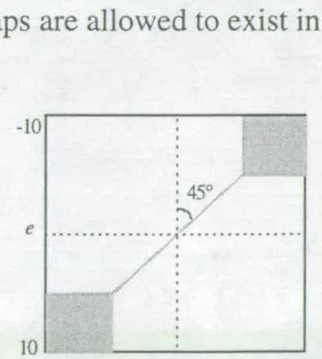
Fig.3-22 Increasing the smoothness of phase plane trajectory by adding more SMALL control actions by the side of ZE-line in the rule base

two fuzzy input variables, *error* and *change of error*, have the same definitions in both fuzzy values (or labels) and membership functions, the switching line determined by the FLC has a switching line angles of 45° at the origin of the phase plane, and two large gray areas on both ends which will lead to a less sensitive response to the error change $\dot{e}$ in big error situations.

If the coordinates of the membership functions of $\dot{e}$ are multiplied by 2, *i.e.* changing the density of the membership functions around the origin, the saturation part in the membership functions of PB and NB disappears, thus there will be no gray areas on the switching line. And the switching line angle will be increased, as shown in Fig.3-23(b). Clearly the switching line performance defined in Fig.3-23(b) will generate a faster response, thus bigger overshoot than the switching line performance defined in Fig.3-23(a). Similarly, if the coordinates of the membership functions of e instead of $\dot{e}$ are multiplied by 2, there will be no gray areas on the switching line either, and the switching line angle will be decreased to 30°. The switching line performance will become slow.

If the positive part of membership functions of $\dot{e}$ is multiplied by 2, and the negative part is kept unchanged, a combination of the switching lines in Fig.3-23(a) and (b) can be realized, as shown in Fig.3-24. This switching line produces fast response when *error* is negative and slow response when *error* is positive.

It should be noted that the switching line performance can also be altered by changing the overlap of membership functions. If trapezoidal membership functions are used and non-overlapped gaps are allowed to exist in the membership functions of fuzzy input variables,



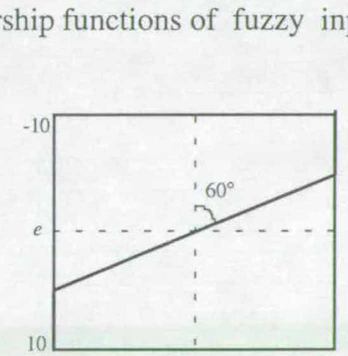


Fig.3-23 Switching line defined by a symmetric rule base, unit scaling factors, and the membership functions in Fig.3-4(b): (a) before (b) after multiplying the coordinates of the membership functions of $\dot{e}$ by 2

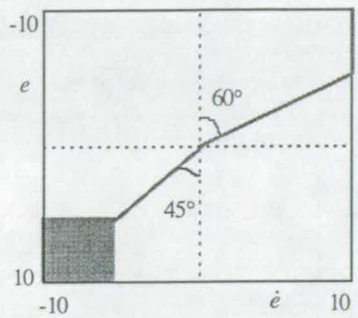


Fig.3-24 Combination of the switching lines in Fig.3-23(a) and (b)

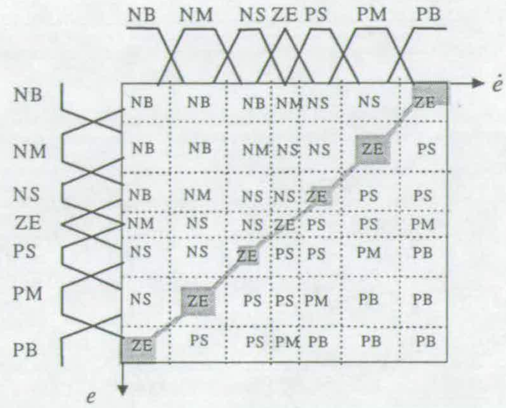


Fig.3-25 Creating insensitive-areas on the switching line by using trapezoidal membership functions

there will be several insensitive-blocks on the switching line, as shown in Fig.3-25. In the insensitive-area, control output *u* will not change however the input varies, since only one fuzzy rule with constant degree of firing (DOF) is fired. This robustness with respect to the input variation can be utilized to stabilize the system when the oscillation of phase plane trajectory around the switching line develops, because operation in the insensitive-area allows loose control of the phase plane trajectory. It can also be used to eliminate the effect of noise in the crisp inputs on the system performance. However, increasing the insensitive-area on the phase plane may deteriorate system performance due to the open-loop-like control characteristic in the insensitive-area.

On the other hand, increasing the overlap in both sides of membership functions of input fuzzy variables will not generate significant changes in the switching line performance compared with two-overlapped membership functions. The reason is that an increased overlap will increase the number of fired fuzzy rules; the extra rules sit around the previous rules; the average control change generated by the extra rules will not significantly differ from that generated by the previous rules. In addition, more overlapping requires more computational resource, and this may reduce the controller speed. That is why most fuzzy control applications use no more than two overlapped membership functions.

So far little attention has been given to the effect of output variable membership functions on system performance. However, they can directly affect the switching line performance of fuzzy systems in a manner similar to the effect of the process gain. The membership functions of the output variable enable the FLC to produce any non-linear control by different definitions of each fuzzy output set. Changing the position of the membership function of an output fuzzy set will alter its contribution to the controller output if this output fuzzy set is the consequence of a fired fuzzy rule. Changing the support of the membership function of an output fuzzy set will alter its weight in defuzzification if the COG defuzzification algorithm is used, this changes its effect on the controller output. But the shape of membership functions will not significantly affect the controller output. All these effects are equivalent to gain changes in the controller from a piecewise linear point of view.

As far as the reaching condition of the FLC, described in section 3.4, is concerned, high controller gain is required so that any start point on the phase plane can meet the reaching condition, especially for low gain processes. As far as the sliding condition of the FLC is concerned, low controller gain on both sides of the switching line is preferred, since low gain can help to stabilize the oscillation of phase plane trajectory around the switching line, especially in case of high gain processes. In addition, high order processes or time delay processes will not endure high gain controls because of the stability problem. Therefore, there is a trade-off between the system performance and stability, or the reaching condition and sliding condition.

Generally, when a symmetrical rule base is used, any monotonic switching line in the phase plane can be realized by using different sized and positioned membership functions of the input variables. If both the rule base and membership functions are manipulated, any type of switching line can be implemented. This flexible switching line feature makes FLC systems extremely powerful, able to produce any required non-time-varying control law. Owing to this flexible switching line feature, system robustness can be optimized by properly designed rule base and membership functions.

3.5.3 Effect of scaling factors

Optimal scaling not only depends on the properties of e, $\dot{e}$ and Δu, but also on the shape and position of the membership functions used and the dynamics of the plant to be controlled.

For a given fuzzy controller, increasing the scaling factor is equivalent to compressing the corresponding membership functions towards the origin of the related axis, and thus decreasing the support of each fuzzy subset related to that fuzzy variable on both sides of the origin. For example, membership functions for error $\mu(e)$ in Fig.3-26(a) with scaling factor GE = 2, are equivalent to those shown in Fig.3-26(b) with GE = 1, because the same fuzzification can be produced for a crisp input in both cases. That is, if the practical input *error* = 1.0, the output of the fuzzification operation will be PM with membership grade 1.0 in both situations shown in Fig.3-26. It should be noted that the coordinates in Fig.3-26 are given after scaling.

If the membership functions are fixed, increasing the scaling factor of a variable results in a decrease of the practical region of that variable covered by the unsaturated membership functions, thus increasing the practical region covered by the saturated part of membership functions, *i.e.* NB and PB. Therefore, the bigger the scaling factor of a variable, the more sensitive will the controller be to the small values of this variable, and less sensitive to big values of this variable.

Due to the effect of the scaling factors on the position of membership functions, the

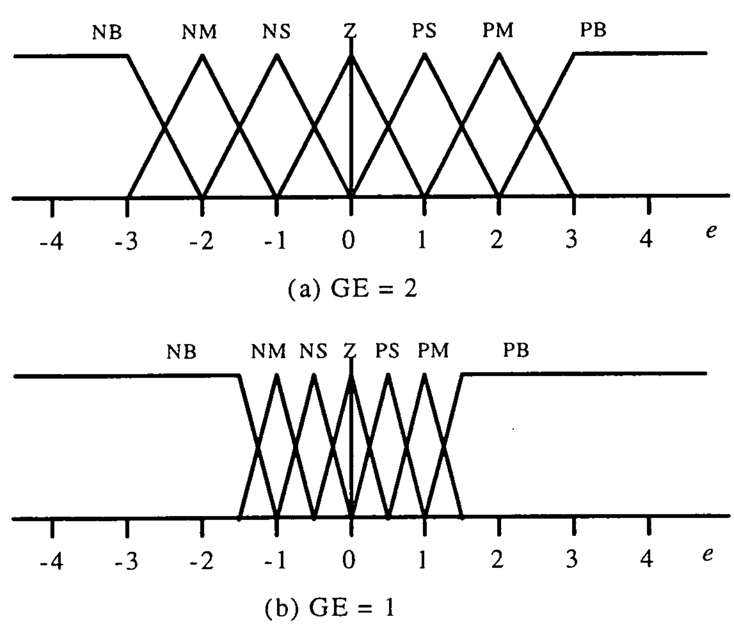
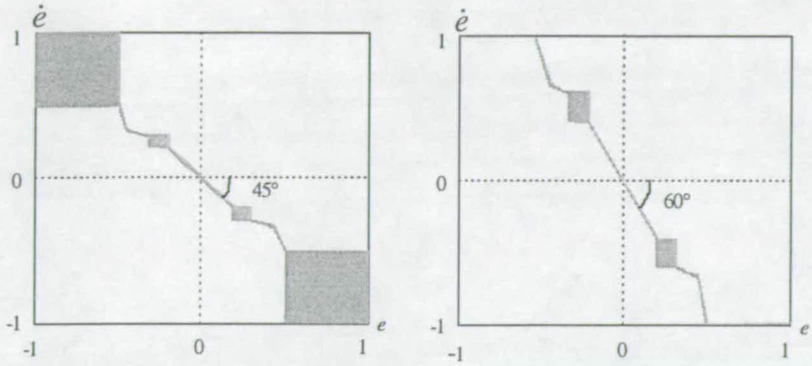


Fig.3-26 Same fuzzification output from different definition of membership functions and different scaling factor



(a) GC = 1.0; (b) GC = 0.5

Fig.3-27 Illustration of effect of scaling factor GC on the switching line

switching line angle and the size of the insensitive area on the switching line can be changed simply by tuning the input scaling factors. Fig.3-27 illustrates two switching lines at different input scaling factors. It can be seen that the two big insensitive-areas at both ends of the switching line in Fig.3-27(a) have disappeared and the switching line angle changes from 45° to 60° when GC changes from 1.0 to 0.5.

It is also known that the function of scaling factors is to map the practical universe R of a variable to its corresponding designed fuzzy universe F, *i.e.*

$$R = a\,F \tag{3-30}$$

where a is the scaling factor. From the mapping point of view, if the designed fuzzy universe is kept unchanged as it is in the practice, the corresponding practical universe will be decreased by increasing the scaling factor. Then a relatively small area in the universe R will be focused inside the controller, thus increasing the sensitivity of the controller to small input signals.

The effect of output scaling factor GU on system performance is the same as that of output membership functions. Any change in GU will directly affect the system loop gain. The difference between the effect of the output scaling factor and the output membership functions is that the scaling factor can only change the overall gain of the control, not a specific part of the output universe of the control change Δu, which can be done by modifying only part of the output membership function as discussed in the last section. Since GU can change the overall loop gain, it can be used to tune the controller to give the required response speed and/or to meet certain stability conditions. Intuitively, decreasing the output scaling factor can improve system robustness if the stability is the main reason for poor robust performance; increasing the output scaling factor will improve system response speed if there is no stability problem.

In summary, input scaling factors can affect the position of the corresponding membership functions and change the position of the switching line, the size of the insensitive-area, and the switching line angle. A big scaling factor results in a small input region around the origin, increasing the sensitivity of the controller to small signal changes and saturating large signal changes. The output scaling factor functions as a multiplier of the overall loop gain. It can directly affect system performance. Owing to the effects of scaling factors on system performance and their simplicity in tuning, they are often used to tune system performance in many practical applications of fuzzy control. *Chapter* 5 will focus on this issue.

3.6 Designing for robust performance

Although fuzzy control is very successful, especially for the control of nonlinear systems, there is a lack of a systematic design method for such controllers with respect to the robust performance and stability of FLC systems. The control rules are normally extracted from practical experience, which may make the design rather subjective. It is also difficult to design the membership functions of input-output fuzzy variables for a specific system performance because of the multi-parameter dependency. A detailed theoretical base has yet to be developed for non-linear systems in general and fuzzy systems in particular that will allow a systematic, mathematically based, design method to be established. Fortunately, the qualitative approach to predicting the performance of fuzzy logic controllers presented in this thesis can be applied to the design of fuzzy systems and has been found to lead to successful system operation. It is expected that this approach will lead to the formation of a knowledge base essential for the development of the theoretical framework.

In this section, a procedure will be prescribed that will lead to the to design of a FLC to achieve robust performance. First, a switching line method for a robust FLC system will be discussed. The position of the switching line will be selected and the distribution of rules on both sides of the switching line will be determined, based on the relationship between the system performance and the position of the switching line investigated in the previous sections. Second, the high order process control problem will be addressed by modifying the structure of the FLC. Finally, an evolutionary method based on a genetic algorithm will be proposed to automatically design a robust fuzzy control system.

3.6.1 Switching line method for a robust FLC system

The motivations for the design of a robust FLC system can be stated as follows. It is found in section 3.4.1 that the switching line performance will dominate the dynamic behaviour of the system if the sliding mode occurs. So the position of the switching line on the phase plane can be determined according to system performance requirements. It is also indicated in section 3.4.1 that the size of the robust space will be affected by the distribution of rules on both sides of the switching line, thus the locations of all membership functions and rules on the phase plane can be selected with respect to the robustness requirement.

In this section, the position of the switching line will first be determined, then the locations of the membership functions and the rules in the phase plane. The position of the switching line is determined to meet the requirements of the system performance at the selected tuning point. The positions of membership functions of the input variables and the ZERO-output rules will be decided by satisfying the reaching condition and sliding condition so that the system can operate in the sliding mode.

A. Design the switching line form the required system performance

Suppose that no more than two membership functions are overlapped, and normalized triangles are used except for the trapezoidal membership function at each end of the predetermined domain of the fuzzy variable. Also assume that the variation ranges of the system error e and error change $\dot{e}$ are known as $[-e_{max}, e_{max}]$ and $[-\dot{e}_{max}, \dot{e}_{max}]$. Based on the procedure used to draw the switching line on the phase plane from the input fuzzy membership functions, the following reverse procedure can be used to design the required switching line satisfying the performance specifications.

Step 1: Draw a normalized phase plane P, and scale the variation ranges of e and $\dot{e}$ to the unit interval, *i.e.*

e domain $[-e_{max}, e_{max}] \Rightarrow [-1, 1]$, $\dot{e}$ domain $[-\dot{e}_{max}, \dot{e}_{max}] \Rightarrow [-1, 1]$.

Step 2: Select a set of process parameters as the operating point O of the control system. Consideration should be given to the possible variation of process parameters, and selecting the middle point as the center of the parameter domains. In this operation, the equal variation ratio of each parameter on both sides of the tuning point is recommended, that is, the ratio of the upper limit of the operation range to the selected operating point value of a parameter should be equal to the ratio of the operating point value to the lower limit of that parameter range. For example, if the bounded variation range of parameter K is [1, 9], the operating point should be K = 3, because K has the same increase and decrease ratios.

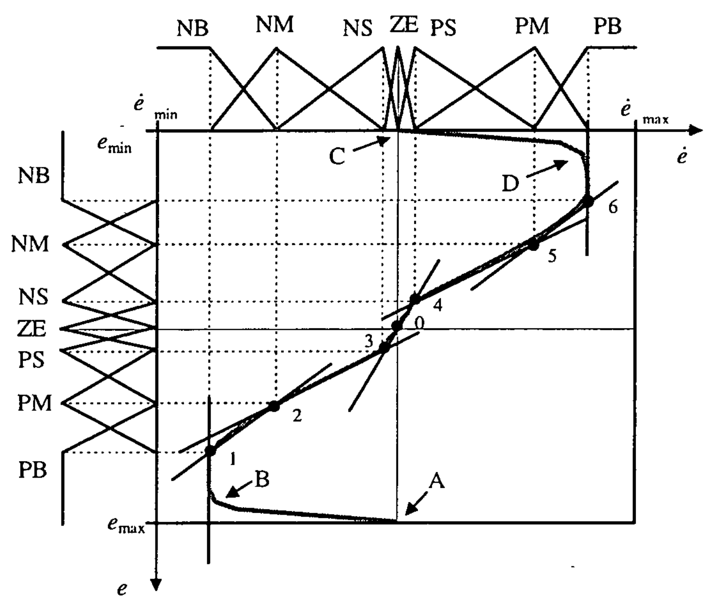
Step 3: On the normalized phase plane P, design a phase plane trajectory L, on which the system performance at the operating point O is satisfied. Except for the start-up part of the phase trajectory, L will be used as the switching line to determine the membership functions. In this step, help from computing tools is needed to calculate the settling time of the step response.

Step 4: Use N tangentially positioned lines to approximate the selected switching line L as close as possible (N is the number of membership functions to be used for fuzzy variables e and $\dot{e}$), as shown in Fig.3-28 (a). (N-1) crossing points are generated by these tangential lines. One of these lines must cross the origin of the phase plane.

Step 5: Together with the origin of the phase plane, N points determine the coordinates of the N apexes of the triangle membership functions for both input fuzzy variables, see Fig.3-28 (a).

There are many possible positions of L because normally the system performance specifications are set to accept a performance range rather than exact specifications. Attention has to be paid to determine the swiftness of the step response. A fast step response may create a stability problem when process gain K becomes high; a slow step response could lead to an unsatisfied settling time specification when the process is dynamically fast, for example when it has a small time constant T. Either situation could impair the system robustness. Adjustment of the switching line position may be necessary if the robustness does not meet the requirement for the whole or part of the parameter space.

It should be noted that the selection of the switching line is heavily dependent on the dynamics of the controlled process. It is impossible to arbitrarily select a switching line for the system with control saturation (as all practical processes are). From the author's experience, the phase plane trajectories of the open-loop unit step responses of the controlled process with different gains can be used to determine the switching line of a closed-loop system. However, a systematic design method for this task has not been developed yet. Further research work is needed.



(a)

e \ $\dot{e}$	NB	NM	NS	ZE	PS	PM	PB
NB						NS	PS
NM					NS	ZE	PS
NS				NS	ZE	PS	
ZE			NS	ZE	PS		
PS		NS	ZE	PS			
PM	NS	ZE	PS				
PB	NS	PS					

use PS or NS on the rules on both sides of the ZE rules

use ZE on the rules corresponding to the switching line

(b)

Fig.3-28 Determination of (a) membership functions (n=7) and (b) rules from the required switching line (A-B, C-D are the start-up region of the phase plane trajectory).

B. Design fuzzy rule base from the required robustness

After the determination of the switching line of the FLC system, system robustness becomes the main concern when we are going to design the output membership functions and the rule base. Because of the switching line feature of FLC systems, system robustness can be achieved by requiring the system to meet the reaching condition and the sliding condition.

For different controlled processes, a robustness specification may require different reaching conditions and sliding conditions. In general, a parameter whose variation can affect the dynamic performance of the process will influence these conditions. In the following, a first order process will be used to illustrate the design of the output membership functions and the rule base to meet these conditions. The method can be applied to other process cases, provided that the concerned parameter has the same effect on the system robustness as one of the parameters of a first-order process.

From section 3.4.2, the reaching condition and the sliding condition of the first-order process are respectively

$$\frac{K}{T} \geq \frac{|\dot{e}|_{min}}{|\Delta u|_{max}}, \quad (3\text{-}31)$$

$$\frac{K}{T} \leq \frac{|\Delta \dot{e}|_{max}}{|\Delta u_{sl}|_{max}}, \quad (3\text{-}32)$$

where K is the process gain, T is the process time constant, $|\Delta u|_{max}$ is the maximum control change from the FLC, $|\Delta u_{sl}|_{max}$ is the maximum acceptable control fluctuation, $|\dot{e}|_{min}$ is defined by the minimum error change to satisfy the reaching condition, and $|\Delta \dot{e}|_{max}$ is the maximum speed change allowed for the phase plane trajectory to slide on the switching line in the sliding period.

Suppose that the parameter domains of K and T are $[K_{min}, K_{max}]$ and $[T_{min}, T_{max}]$ respectively, within which the system performance is required to satisfy the design specifications. From (3-31) and (3-32), we have

$$\frac{K_{min}}{T_{max}} \geq \frac{|\dot{e}|_{min}}{|\Delta u|_{max}}, \quad (3\text{-}33)$$

$$\frac{K_{max}}{T_{min}} \leq \frac{|\Delta \dot{e}|_{max}}{|\Delta u_{sl}|_{max}}, \quad (3\text{-}34)$$

From the reaching condition (3-33), $|\dot{e}|_{min}$ is related to the settling time specification, denoted as ST_{spec}. That is, the system response should be fast enough to reach the switching line so that the settling time specification can be met. If the phase plane trajectory of the FLC system is partitioned into reaching part and sliding part, see Fig.3-29, the following condition can be derived from (3-33):

$$t_{reach} + t_{slide} \le ST_{spec} \tag{3-35}$$

where t_{reach} and t_{slide} are times spent in the reaching period and the sliding part of the step response respectively.

If the unequal sign in (3-35) is replace with the equal sign, then t_{reach} is the maximum time spent during the reaching period. Thus $|\dot{e}|_{min}$ can be approximately calculated as

$$|\dot{e}|_{min} = \frac{\Delta e}{t_{reach}} \tag{3-36}$$

where Δe is error change covered by the reaching period, as shown in Fig.3-29. Δe and t_{reach} can be obtained from the phase plane trajectory. From (3-33), the maximum control change $|\Delta u|_{max}$ should meet the condition

$$|\Delta u|_{max} \ge \frac{|\dot{e}|_{min} \cdot T_{max}}{K_{min}} = \frac{\Delta e \cdot T_{max}}{t_{reach} \cdot K_{min}} \tag{3-37}$$

Based on the defuzzification algorithm presented in *Chapter* 2, $|\Delta u|_{max}$ is the control output related to the fuzzy control rule corresponding to the start point on the phase plane where e is fuzzy BIG, and $\dot{e}$ is fuzzy ZERO. Suppose that at this system state, the consequent part

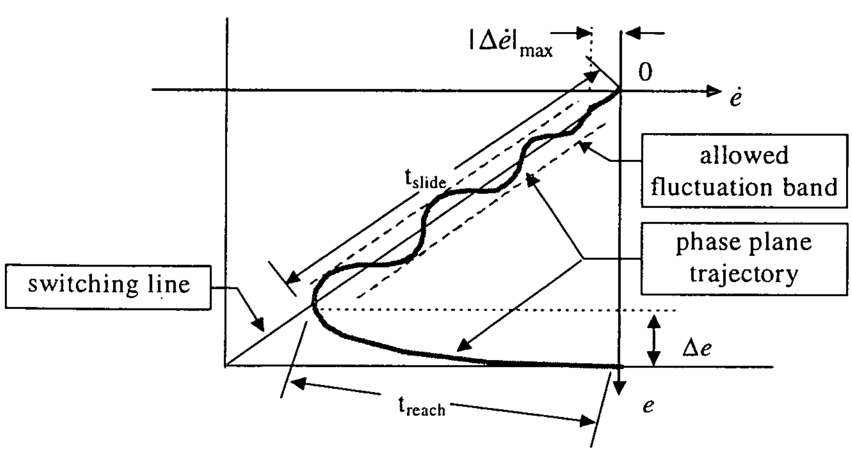


Fig.3-29 Diagram illustrating the reaching period and the sliding period

of the fuzzy rule is fuzzy PB as its output, and the center-of-gravity method is used. Then the center-of-gravity value of this output fuzzy set PB, cog($PB_{\Delta u}$), should be

$$cog(PB_{\Delta u}) = \frac{|\Delta u|_{max}}{GU} \tag{3-38}$$

where GU is the scaling factor of the control change Δu. If the isosceles triangle is used as the membership functions of Δu, then the core value of fuzzy set PB, Core($PB_{\Delta u}$), will be the same as cog($PB_{\Delta u}$) as shown in (3-38).

Similarly, from (3-34), the maximum acceptable control fluctuation, $|\Delta u_{sl}|_{max}$ should met the condition

$$|\Delta u_{sl}|_{max} \le \frac{|\Delta \dot{e}|_{max} \cdot T_{min}}{K_{max}} \tag{3-39}$$

where $|\Delta \dot{e}|_{max}$ can be obtained from the required switching line shown in Fig.3-29.

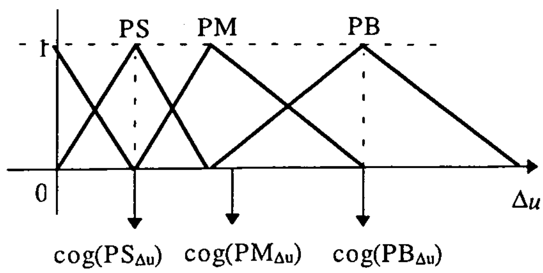
If fuzzy SMALL control changes are used in the rules on both sides of the switching line, then, for example, the center-of-gravity value of the output fuzzy set PS, cog($PS_{\Delta u}$), should be

$$cog(PS_{\Delta u}) = \frac{|\Delta u_{sl}|_{max}}{GU} \le \frac{|\Delta \dot{e}|_{max} \cdot T_{min}}{GU \cdot K_{max}}. \tag{3-40}$$

If the isosceles triangle is used as the membership functions of Δu, then the core value of the PS, Core($PS_{\Delta u}$), will be the same as cog($PS_{\Delta u}$) as shown in (3-40). Note that normally $|\Delta u_{sl}|_{max} < |\Delta u|_{max}$, cog($PS_{\Delta u}$)< cog($PB_{\Delta u}$).

If fuzzy PM is used in fuzzy variable Δu, its center-of-gravity value can be chosen at between the cog($PB_{\Delta u}$) and cog($PS_{\Delta u}$). The support of each fuzzy set can be decided by using a normal triangle shape membership function, or simply using fuzzy singletons at the positions of cog($PB_{\Delta u}$), cog($PM_{\Delta u}$) and cog($PS_{\Delta u}$), as shown in Fig.3-30. Similarly, the membership functions of NS, NM and NB can be determined by the same method.

During the determination of the output membership functions, rules on and besides the switching line (see Fig.3-28) are decided. Other rules can be determined by expert experience or methods introduced by Yager [25].



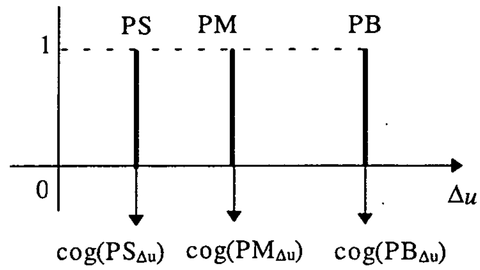


Fig.3-30 Determination of the output membership functions (a)triangles; (b) singletons.

Note that, if the parameter range of K or T is large, $|\Delta u_{sl}|_{max}$ may be too small and $|\Delta u|_{max}$ may be too large to implement in the practical system. In this case, modification should be made on either the performance specification or the parameter range. There will be a trade-off between the system performance and the robustness.

3.6.2 Phase advanced FLC

It is found in this research that FLC systems present a superior robustness in comparison with the traditional PID control systems with respect to the parameter changes in the lower order controlled processes. In case of high order controlled processes, FLC cannot handle the parameter change so well as in the low order case. Poor transient performance of fuzzy logic controlled high order systems has also been reported in [120]. Further investigation revealed that the degraded robustness results from the delayed response of the process to the control action from the controller. The initial response of a high order process to a step input cannot be set up quickly, but rises more slowly to its maximum rate of change than that of a first-order process, see Fig.3-31.

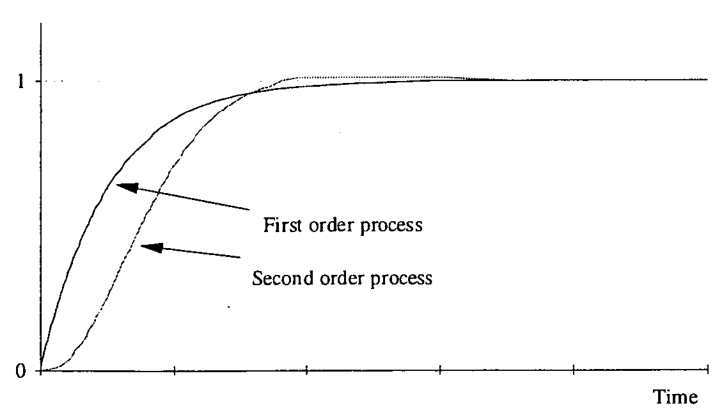


Fig.3-31 Step responses of First- and second-order processes with same settling time

Because of the switching line feature of the FLC systems, the delayed setup of response speed of a high order process will result in further increase (or decrease) of the control input from the FLC to force the phase plane trajectory of the process to follow the predetermined switching line. The further action in the control input will lead to oscillation of the phase plane trajectory around the switching line, because the FLC's decision for the further action is based on incorrect feedback information. Although the FLC can be tuned to obtain a satisfactory performance whose phase plane trajectory may perfectly follow the predetermined switching line, system performance will deteriorate quickly with variation of the process parameters, because parameter change will also affect the setup speed of the response.

Note that the effect of parameter change for a first-order process on the process response is different from that of the high order process. Parameter change only affects the response speed of a first-order process, but can alter the swiftness of setting up the response speed in the high order process.

In the following part of this section, methods will be developed to handle this high order problem with respect to two cases: high order process with and without integrating terms.

A. Non-integrating high order process

In the case of the non-integrating controlled process, an accumulator is used after the FLC to achieve zero steady state error, as shown in Fig.3-32(a). Functioning as an integrator, the accumulator makes the above problem with high order processes worse because of the impact of the integrator term on the stability of the closed loop system. That is, if the controlled process is non-integrating and high order, the control problem encountered has conflicting design objectives: zero steady state error, and non-oscillatory phase plane trajectory following the predetermined switching line.

From linear control theory, the above problem can be solved by introducing phase-lead compensation to the system. Phase-lead compensation can be implemented by introducing an extra *zero* to the system. This design idea leads the addition of a proportional path paralleled to the integrator after the FLC, thus creating a PI type compensator. It is known that this parallel proportional path can significantly reduce the impact of the integrator term on the stability of the closed loop system.

The structure of the modified fuzzy control system is illustrated in Fig.3-32(b). To change the effect of either the proportional term or the integral term on the system performance, a proportional gain K_p is used in the proportional path and the output scaling factor GU is put into the integral path. The control input u is calculated as

$$u = K_p \, \Delta u + GU \int \Delta u \, dt \tag{3-41}$$

or

$$u(k+1) = K_p \, \Delta u(k) + GU \sum_{i=0}^{i=k} \Delta u(i) \tag{3-42}$$

The compensation can be expressed in the transfer function form as follows

$$G(s) = \frac{GU(\alpha s + 1)}{s} \tag{3-43}$$

where $\alpha = K_p / GU$. The *zero* introduced into the system is at $(-1/\alpha)$.

With phase-lead compensation, the modified FLC can output a stronger control action during the beginning of the step response than the normal FLC, and the extra control action from the modified FLC will gradually disappear when the response speed is setting up. Hence the oscillation of phase plane trajectory around the predetermined switching line, caused by the over drive from the FLC, can be significantly reduced; and system robustness can be improved.

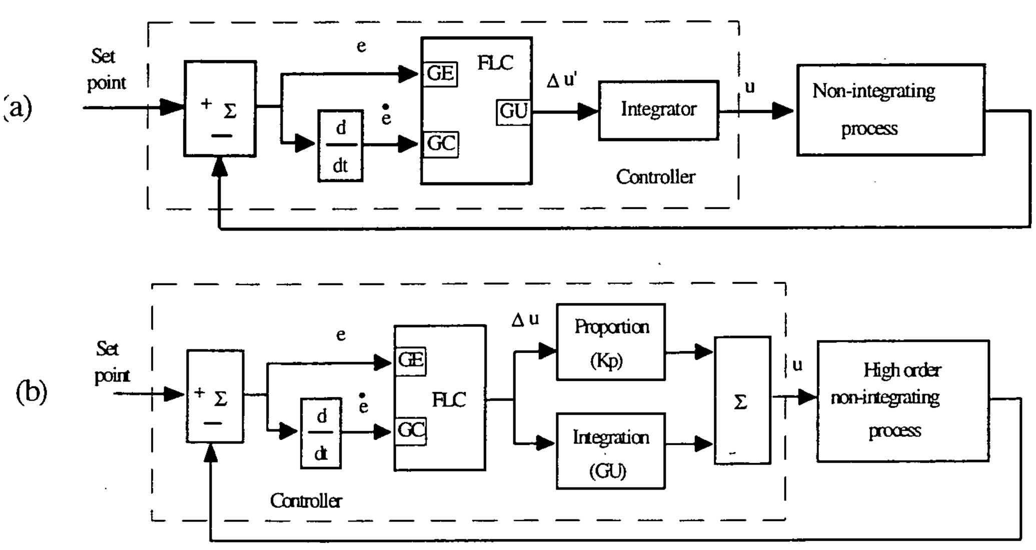


Fig.3-32 fuzzy control system for non-integrating processes (a) Basic FLC, (b) phase-lead FLC

B. Integrating high order processes

If there exists an integrating term in the control process, there is no need to use the accumulator after the FLC to achieve the zero steady state error. That is, the output of the FLC is used directly to drive the controlled process. In this case, the phase-lead compensation proposed above cannot be implemented.

Although there is no integrator after the FLC when the controlled process contains an integrating term, the whole system will present the same performance as the system containing a non-integrating process and a normal FLC. Therefore, the problem related the poor performance still exists. In addition, even if a high order process contains no integrating term, the delayed setting up of the response speed of the process can lead to the same problem as mentioned above.

From the fundamental principle of the FLC in controlling a process, *i.e.* the switching line theory, if the controller inputs can reflect the complete dynamics of the controlled process, it won't be a problem to obtain a required performance, because a FLC with an adequate rule base containing all possible system states can generate appropriate control action to handle the high order process. The reason for a superior robustness achieved in the first-order process is that the FLC inputs, e and $\dot{e}$, contain all system states. In case of the high order controlled processes, however, the designer has to pay the price to create and implement a complicated FLC rule base, and it is usually much more difficult to create a multi-variable rule base.

Often practical high order processes are approximated as second order processes, so attention will be restricted to the second order process. Generally, a second order process can be described by

$$\ddot{y} + a\dot{y} + by = f(u,t)\,, \tag{3-44}$$

where y is the output of the process, $f(u,t)$ is a function of control input from the controller. If a system state variable Y can be found with the following condition met

$$\dot{Y} = \ddot{y} + a\dot{y}\,, \tag{3-45}$$

a second order process becomes

$$\dot{Y} + by = f(u,t)\,. \tag{3-46}$$

Compared with the first order process

$$\dot{y} + by = f(u,t), \qquad (3\text{-}47)$$

it is clear that if both $\dot{Y}$ and y can be reflected in the inputs of the FLC, a robust performance similar to the first order process can be achieved.

Furthermore, it can be seen from (3-45) that $\dot{Y}$ will mainly reflect the $\dot{y}$ part because $\ddot{y}$ is often much smaller than $\dot{y}$ in practice. With $\dot{Y}$ as the *virtual derivative error*, therefore, the rule base for controlling the second order process will possess the same features as the rule base for controlling the first order process. This will greatly save time in designing the fuzzy controllers for second order process.

The structure of this modified FLC system can be illustrated as in Fig.3-33. Because the scaling factor GC is used for the mapping of the derivative input signal, an extra parameter β in the $\ddot{e}$ path will be sufficient to adjust the effect of $\ddot{e}$ on system performance. Due to the specific characteristic of the virtual derivative error $\dot{Y}$ in this FLC system, this method will be referred as the *virtual derivative error* method. In fact, this is a derivative feed-forward method which creates a phase lead effect to modify the system performance.

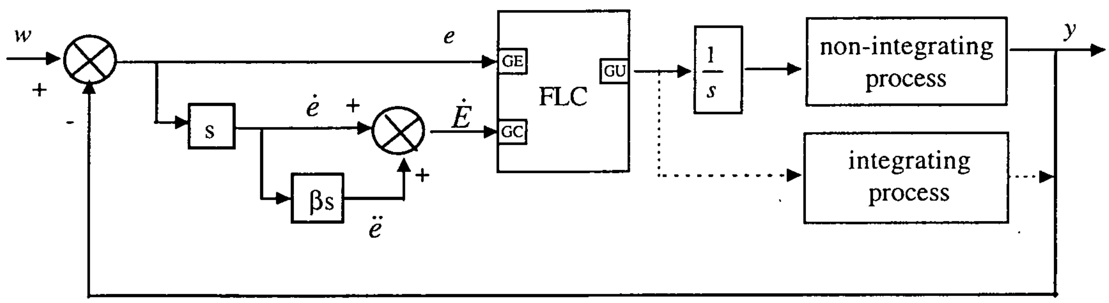


Fig.3-33 Fuzzy control system for high order processes

From Fig.3-33, it is found that in the virtual derivative error method, the practical $\dot{e}$ input is compensated by $\ddot{e}$. That is, not only $\dot{e}$, but also the change rate of $\dot{e}$ is considered by the FLC in its decision making. A BIG change rate of $\dot{e}$ means the process has got a sufficient amount of energy from the control input to produce a BIG change in $\dot{e}$. Though $\dot{e}$ may still be unsatisfied (too SMALL or too BIG) at present, there is no need to change the present control action if the combination of $\dot{e}$ and $\ddot{e}$ is satisfied.

This control algorithm can overcome the oscillation of the phase plane trajectory for a high order process resulting from overdriving, and provide a better performance and system robustness than the normal FLC algorithm.

3.6.3 Automatic design by genetic algorithm

Although works on designing a fuzzy control system and on successful applications to industrial control problems are widely reported, the available design methods so far have been exclusively by way of trial-and-error based techniques, using an initial model and then manual adjustments on-line. These do not necessarily yield the "best" design or control performance. It is almost impossible to optimize the design of such a multi-variable system as FLC systems to obtain satisfactory robustness in case of significant uncertainties in the controlled process.

Clearly, existing robust fuzzy logic controllers and their corresponding parameter ranges are known to the designer. The problem is how to automatically determine the controller parameters to achieve the “best” system robustness in a known parameter space of the controlled process. In fact, this design problem is a search problem which can be solved by search methods such as the genetic algorithm and simulated annealing.

Because of the non-linear and multi-variable characteristic of the FLC, the search method used for this purpose must be able to avoid local minima in the search space, *i.e.* the controller parameter space. For the above constraint, a genetic algorithm becomes the suitable choice.

Automatic design of FLC systems using genetic algorithms (GA) has been reported in several papers [49][52][108][109][111][112][113]. Most of these papers focus on the development of rule sets or membership functions or both for high performance of a specific controlled process such as a pH control process [49] and the cart-pole system [52] [112][109]. However, system robustness was not the objective of these research projects, though a brief examination of system robustness was presented [52]. Thus, the use of the

GA to determine all FLC parameters (membership functions, rule base, scaling factors, and sampling rate) simultaneously for optimal or near-optimal robust control is the main objective of this section.

In this section, a brief description of the GA will be presented (for a complete discussion, see [110]), and, as the essential issue in designing GA applications, the selection and encoding of the evaluated variables of FLC will be discussed. Finally, the evaluation function (or fitness function) will be designed with respect to the system robustness. Simulation results will be presented in *Chapter* 4.

A. Genetic algorithms

Genetic algorithms are probabilistic search techniques that emulate the mechanics of genetic evolution. Unlike many classical optimization techniques, genetic algorithms do no rely on computing local derivatives to guide the search process. Genetic algorithms also include random elements, which helps avoid getting trapped in any local minima.

Genetic algorithms explore a population of solutions in parallel. The size of the population is a free parameter, which trades off coverage of the search space against the time required to compute the next generation. Each solution in the population is coded as a binary string, called a *gene*, and a collection of genes forms a generation. A new generation evolves by performing genetic operations, such as selection, crossover, and mutation, on genes in the current population and then placing the products into the new generation.

In a simple genetic algorithm, operations are performed in the following order; selection, crossover, and mutation. Selection involves selecting two parent genes from the current generation. It is based probabilistically on a gene's fitness value; the higher the fitness of a gene, the more likely it can reproduce. After selecting two parents, crossover is performed according to a crossover probability. If crossover is to be performed, offsprings are constructed by copying portions of parent genes designated by random crossover points (two-point crossover shown in Fig.3-34). Otherwise, an offspring copies its entire gene from one of the parents. As each bit is copied from parent to offspring, the bit has the probability of flipping, or mutating, as shown in Fig.3-34. Mutation is believed to help inject any information that may have been lost in previous generations.

A gene's fitness is evaluated by decoding the gene's binary representation and then passing it through a fitness function. The fitness function is a means to rank solutions in the population and can include penalty terms in addition to raw performance measures. In this application of GA, a penalty strategy is used to favor populations of FLC parameters with stronger tolerance towards the variations of the process parameters.

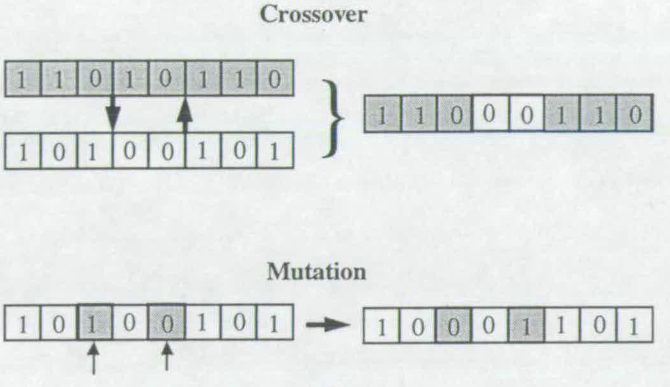


Fig.3-34 Genetic operations

A step-by-step description of a basic GA procedure is as follows:

1) encode the variables involved into suitable strings, often a binary string;
2) select, at random, an initial population of strings;
3) evaluate the fitness of each string;
4) choose the strings from the population probabilistically according to their fitness;
5) apply, with the respective probabilities, operations on chosen strings to obtain new strings, called offspring;
6) introduce the new strings into the original population
7) go to step 3 for a fixed number of iterations or until an acceptable performance is achieved.

B. System representation

It is a long recognized limitation of GA that, as a global search technique, the encoded bit string length should be kept as small as possible, since the size of the search space increases exponentially with the size of the string. Therefore, it is highly desirable to represent the system in a compact way to accomplish the evolution task. To meet this requirement, the following assumptions are made:

1) the FLC has two input variables (e, $\dot{e}$) and one output variable (Δu). Each variable has maximum seven fuzzy sets labeled as NB, NM, NS, ZE, PS, PM, PB;
2) each membership functions is a trapezoidal characterized by one reference point and three base lengths, except that the top length of the ZE membership function is zero with the peak located at the origin, as shown in Fig.3-35a;
3) NB is the consequent part of rules in which e is not larger than NM and $\dot{e}$ is not larger than NS; PB is the consequent part of rules in which e is not less than PM and $\dot{e}$ is not less than PS. This is reasonable when a complete rule base as shown in Fig.3-4c is used.

Note that trapezoidal membership functions instead of triangle ones are used because they can affect the insensitive areas on the switching line.

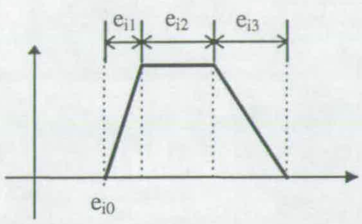
Under the above assumptions, membership functions of each variable can be represented in 28 parameters, namely, e_{i0}, e_{i1}, e_{i2}, e_{i3}. There will be 84 parameters in three FLC variables. The normalized universe of each variable is assumed to divided into 1000 units, and each base length has values from 0 to 255 units. All membership functions then can be represented as a string consisting of 84 one-byte unsigned characters.

Similarly, the fuzzy rule base contains 36 rules to be decided by the genetic algorithm. If these rules are numbered as shown in Fig.3-35b, then only the consequent part in each rule needs to be decided by the evolution. Thus, 36 rules can be represented as 36 3-bit binary strings because the consequent part of each rule take values from 8 possible values:

000 — NB;	001 — NM;	010 — NS;	011 — ZE;
100 — PS;	101 — PM;	110 — PB;	111 — DON'T CARE;

where DON'T CARE indicates that the rule has no effect on the system performance. By adding the DON'T CARE value for the consequent part, the rule's insertion and deletion are automatically done by the genetic algorithms.

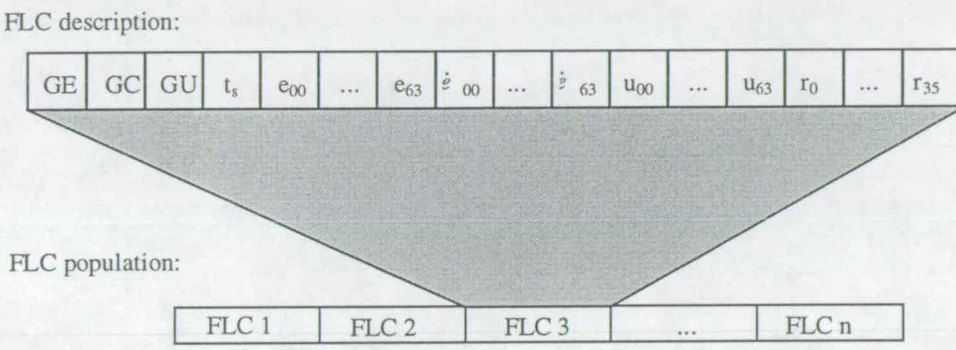
Together with 3 scaling factors and a sampling rate t_s, which are represented by 16 bits each, the entire binary string consists of 84 + 36 + 4 = 124 variables, or $84 \times 8 + 36 \times 3 + 4 \times 16 = 844$ bits, as shown in Fig.3-35c. If the membership functions are assumed symmetrical to the origin of this variable, the number of the parameters for representing the membership functions can be reduced to 42, and the string length becomes 508 bits. This is a complete but compact representation of a FLC, compared with Lee and Takagi's research [99] where 2880 bit per string was used.



(a)

e \ $\dot{e}$	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	r_0	r_1	r_2	r_3
NM	NB	NB	NB	r_4	r_5	r_6	r_7
NS	r_8	r_9	r_{10}	r_{11}	r_{12}	r_{13}	r_{14}
ZE	r_{15}	r_{16}	r_{17}	ZE	r_{18}	r_{19}	r_{20}
PS	r_{21}	r_{22}	r_{23}	r_{24}	r_{25}	r_{26}	r_{27}
PM	r_{28}	r_{29}	r_{30}	r_{31}	PB	PB	PB
PB	r_{32}	r_{33}	r_{34}	r_{35}	PB	PB	PB

(b)



(c)

Fig.3-35 Fuzzy system representation
(a) representation of membership functions (only variable *e* is illustrated);
(b) representation of rule base; (c) FLC representation (n population).

C. System evaluation

Another main issue in GA applications is the selection of the new generation of the evolved population. To do so, GA operations are applied to the current population. For a proper selection, each element in the population has to be evaluated and scored with a fitness value according to its performance.

Unfortunately, there is no mathematical model of fuzzy systems to provide the evaluation of a given FLC system. It has to be assumed that a sort of model (mathematical or fuzzy) of the controlled process is available for GA to be applied. This is the main drawback of the

GA applications in fuzzy control systems, and this is also recognized by other researchers [52].

With the above assumption of the availability of the controlled process model, simulation methods can be applied to evaluate a given FLC system, and the system robustness can be evaluated by the robustness index I_{rb} defined in *Section* 3.2.2. That is, the fitness function, *fit*, is defined as

$$\begin{aligned} fit = I_{rb} &= n - m + \frac{10^{-3}}{n} \sum_{n} P_{index}^{(i)} \\ &= \mathrm{n - m} + \frac{10^{-3}}{n} \sum_{n} (OS^{(i)} + US^{(i)} + CY^{(i)} + 10ISE^{(i)} + R_{st}^{(i)}) \end{aligned} \tag{3-48}$$

where $OS^{(i)}$, $US^{(i)}$, $CY^{(i)}$, $ISE^{(i)}$ and $R_{st}^{(i)}$ are the performance measures in the test point i (i = 1, ... , n). Because *fit* includes the five performance measures, the fitness function provides a continuous measure of the system robustness, which is important for obtaining a fine tuning of the evolved system performance. In addition, since the number of unsatisfied tests is addressed in *fit*, the penalty of these unsatisfied tests is applied in the evolution process.

It is obvious that the smaller the fitness value *fit* of a FLC system, the more test points there are in specification, and the more robust the FLC system. The objective of evolution is to minimize the fitness value of the FLC systems.

With this technique, all controller parameters bounded by the given ranges are mapped to a coded string. Variations in every integer of the string will then span the entire design space. Thus a systematic and automatic design approach can be established using evolutionary programming techniques. Further, the known good controller choices existing in manual designs can be put in the initial "generation" of strings so that the learning can start from this point rather than from scratch. Such knowledge would be very difficult to incorporate directly into an artificial neural network (ANN) where data either has to be logged a-priori from the system under study, or generated from mathematical expressions, before training the ANN.

D. Comparison with other research work

The design method presented above has the following features, compared with the early research work in this field. Firstly, system robustness is the main evolution objective. A penalty is adopted in the fitness function to promote the acceptable robust performance defined by the performance specifications. Trapezoidal membership functions are used, not only to allow the use of normal fuzzy set operation in fuzzy reasoning, but also to take into account the effect of the insensitive area on the switching line for system robustness. In the early work, however, the fitness function was designed with respect only to system performance, and only triangle membership functions were considered.

Secondly, the complete parameter range of the FLC is evaluated. Not only the membership functions and rule base are encoded in the search string as most of the early work did, but scaling factors and sampling rate are also considered. A fine tuning of the positions of membership functions can be done by changing the values of scaling factors, unlike Homaifar's work [52] where a refinement stage was used. Thus the computing time can be greatly reduced. In addition, the representation used in this research is compact.

Finally, by ignoring the unnecessary overlap of membership functions, the representation in this work allows GA to focus on the sensible membership functions instead of the orderless placement of membership functions.

3.7 Conclusion

In this *Chapter*, the measurements of the dynamic performance and system robustness of a control system are firstly defined from the engineering point of view. The structure of the FLC system studied in this work is then described. The robustness of the Mamdani-type FLC systems has been investigated. A switching line method is introduced to qualitatively analyze the dynamic performance of the SISO FLC systems. It is indicated that the position of the switching line on the phase plane determines the dynamic performance of the FLC, and the objective of the fuzzy controller algorithm is to drive the system phase plane trajectory to the steady state along the switching line. It is found that the robustness of FLC systems is closely related to the switching line performance and that it depends very little upon the model of the plant to be controlled. Based on the switching line method, the robustness of the FLC controlled first order process and second order process is studied with respect to the shape and position of the robust space.

The effects of FLC parameters (membership functions of the input-output variables, rule base and scaling factors) on system robustness have been investigated. The relation between the variation of the controller parameters and the position and shape of the switching line, thus, the robust performance, is emphasized. It is indicated that all of the parameters can affect the position, the shape or the angle of the switching line and the smoothness of the phase plane trajectory of the step input responses. The switching line angle is proportional to system overshoot and undershoot, but inversely proportional to the rise time in the step input response. Changes in the overlaps of membership functions of the input space alter the sensitivity of the FLC to the variation of input variables in that space. The effect of output membership functions and output scaling factor on system performance is similar to the process gain in the FLC systems.

Finally, methods are presented to design the FLC to achieve a robust performance. A design method for a robust FLC system is proposed based on the switching line theory. The position of the switching line is selected and the distributions of rules on both sides of the switching line are determined, based on the requirements to the system performance. The control problem with fuzzy control algorithms in case of high order processes is discussed and handled by modifying the structure of the FLC. An evolutionary method based on genetic algorithm is proposed to automatically design a robust fuzzy control system, under the assumption of the availability of the controlled process model.

All analysis presented in this *Chapter* is purely qualitative without mathematical confirmation due to the lack of the systematic analysis method which is applicable to explain the dynamic performance of fuzzy systems. The next chapter will present the results of an extensive series of simulation experiments designed to assess the robust performance of fuzzy logic control systems. It will be seen that the results obtained confirm the qualitative predictions described in this *Chapter*.

4

Experimental Investigation of the Robustness of Fuzzy Logic Control

4.1 Introduction

In the previous chapters the fundamental theory of fuzzy logic control was presented, a switching line method was developed to provide a qualitative analysis of the robustness of fuzzy control systems and new design methods were introduced to improve the robustness of FLC systems. In this chapter, experimental results will be presented to demonstrate the robustness of FLC systems, and the feasibility of the proposed methods for analyzing and designing the robust FLC systems will be demonstrated.

An experimental investigation of the robustness of the fuzzy control algorithm involves the collection of comprehensive information describing system performance with variations of system structure and parameters. Computer simulation will be used to implement these experiments since it is the most effective and economic way. Because designing a FLC does not require a mathematical model of the controlled processes, it will be assumed that only imperfect information about the controlled plants is available at the design stage, though mathematical models of plants are needed for the simulation. Since the widely used PID control method also requires little information about the controlled process, it will be used as a bench mark method for providing comparative performance information.

This chapter is organized as follows. In *Section* 4.2, the simulation system will be described, the specification for robust performance and the experimental procedure will be defined. In *Section* 4.3, a specially designed simulation package will be briefly described and its functions will be introduced. In *Section* 4.4, the robustness of the FLC system will be tested and a comparison study with the widely used PID control system will be presented in cases of the first order and second order controlled processes. The effects of FLC parameters on system robustness will be examined in *Section* 4.5. The feasibility of the robust design methods will be tested in *Section* 4.6, and the chapter conclusion will be given in *Section* 4.7.

4.2 Simulation design

All experimental work related to this research has been performed using simulation. Thus it is important to design a suitable simulation system, define the proper performance specification, and choose an effective experimental procedure for collecting the correct experimental results. These aspects will be discussed in this section.

4.2.1 Simulation system

The fuzzy control system designed in *Section* 3.2, *Chapter* 3 will be used as the main structure of the FLC system implemented in the simulation. All details related to the design of the fuzzy control system will not be presented again to save space. It should be noted that the membership functions and the rule base shown in Fig.3-4 in *Chapter* 3 are the basic settings. Effects of their variations on system performance will be examined. When the effect of some FLC parameters on system robustness is examined, the other parameters will be kept as the basic settings. As shown in *Chapter* 3, the FLC system will be implemented as a discrete system, since the discrete situation is more critical than the analogue situation because of the heavy computational task that has to be performed in the fuzzy controller.

To compare the robust performance of a fuzzy control system with a PID control system, a PID controller has been designed as an alternative method to regulate the output of a single-input, single-output process P around a set-point. The discrete-time equivalent expression for the PID controller used in this paper is given as:

$$u(k) = K_p e(k) + K_i \sum e(k) + K_d \dot{e}(k) \qquad (4\text{-}1)$$

where K_p, K_i, and K_d are the proportional gain, integral gain, and derivative gain of the controller, respectively.

To demonstrate the feasibility of the proposed methods, for analyzing and designing the robustness of FLC systems in *Chapter* 3, the typical first order and second order processes will be used to demonstrate the robustness of the FLC systems in comparison with the PID systems. High order processes will be used to demonstrate the feasibility of two phase-advanced methods and the switching line method. The 'cart-pole' system will be used to test the feasibility of the genetic algorithm design method.

4.2.2 Performance specifications

To investigate the robustness of a control system, it is important to define the acceptable performance, especially when performing a comparative study of different control algorithms. Obviously, a control system under strict performance requirements cannot tolerate big variations in process parameters, while a large robust parameter space requirement may lead to difficulties in designing the system performances. Suitable performance specifications should be defined in accordance with the practical requirement.

In this research, use has been made of the following performance specifications for acceptable performance, based on practical knowledge of the required performance of control systems:

$$\left.\begin{aligned} &\text{Overshoot } OS \le 5\% \\ &\text{Undershoot } US \le 5\% \\ &\text{Settle time ratio } R_{st} \le 3 \\ &\text{Ring cycle number } CY \le 3 \end{aligned}\right\} \qquad (4\text{-}2)$$

The steady state zone for measuring settling time is chosen for the purpose of the experimental investigation as 2% of the normalized input step.

It should be emphasized that the specific values of the above performance specifications are not important to the investigation of system robustness, because they are only used as a standard with which the system performance can be classified as either in or out of specifications. The aim of the investigation is to find the relationship between the controller parameters and system robustness.

4.2.3 Experiment procedure

The following experimental procedure was used to test the robustness of a control system under process parameter uncertainty:

Step 1: Select the tuning point, *i.e.* choose one point in each parameter variation range as the operation point.

Step 2: Tune controller parameters to obtain a satisfactory step response. If performing a comparison study of the robustness of the FLC system with other system, tune the parameters of both controllers to obtain a similar system response to a step input at the same tuning point. The requirement for the similar tuning point performance is important to make two systems comparable, and it will be denoted as STPP *condition.*

Step 3: Fix the controller parameters and test the system performance at different points of the process parameter space.

Step 4: Using the criteria defined in (4-2), determine the robust space and the robustness. Note that in all experimental test, system robustness will be measured by using the equation (3-6) in *Chapter* 3.

Experience of controlling the process is needed in tuning a control system. The main technique used to achieve the satisfied performance is to tune the sample rate and K_p, K_i, K_d in the PID system, or *GE*, *GC*, *GU* in the FLC system. The method for tuning the PID system can be found in [4], and that for the FLC system will be extensively discussed in *Chapter* 5.

4.3 Design of the experimental system

To investigate the robustness of fuzzy logic control systems and to compare the performance of FLC systems with that of the conventional PID control method, a simulation package called *FzySimu* (see Fi.g.4-1) has been developed in the C++ programming language. The advantages of this specially designed software over a commercial software are as follows.

1) It is possible to investigate the mechanism of the FLC in depth so as to have a

thorough understanding of the FLC systems;

2) It is convenient to build any software interfaces to other application programs, such as *RobustMap*, *GaOpt* and *RCPlot* in the *FzySimu* package, so that automatic data processing becomes possible; and

3) The cost advantage is obvious.

FzySimu consists of system simulation and analysis (*MultiSimu*), mapping the robust space (*RobuMap*), optimization by GA (*GaOpt*), auto-tuning (*AutoTune*), a real-time control interface (*RCPlot*), a practical FLC prototype and a third-order process emulator. The whole research environment can be illustrated as shown in Fig.4-1. *MultiSimu* is a multi-objective simulation tool which can be used to simulate different control algorithms and different controlled processes. *RobustMap* is a mapping tool which performs exhaustive searching in the user defined process parameter space. The size and position of the *Robust Space* are determined within which the system performance satisfied the specifications (3-3). *GaOpt* is an optimization tool using the genetic algorithm for selecting the controller parameter for improved robust performance. It is designed to automatically select the fuzzy membership functions, rule base, scaling parameters and sampling rate. *AutoTune* is a fuzzy tuning tool for tuning the fuzzy control system by altering the input-output scaling thus changing the membership grades of the input-output variables belong to the fuzzy subsets to which the fuzzy rules are closely related.

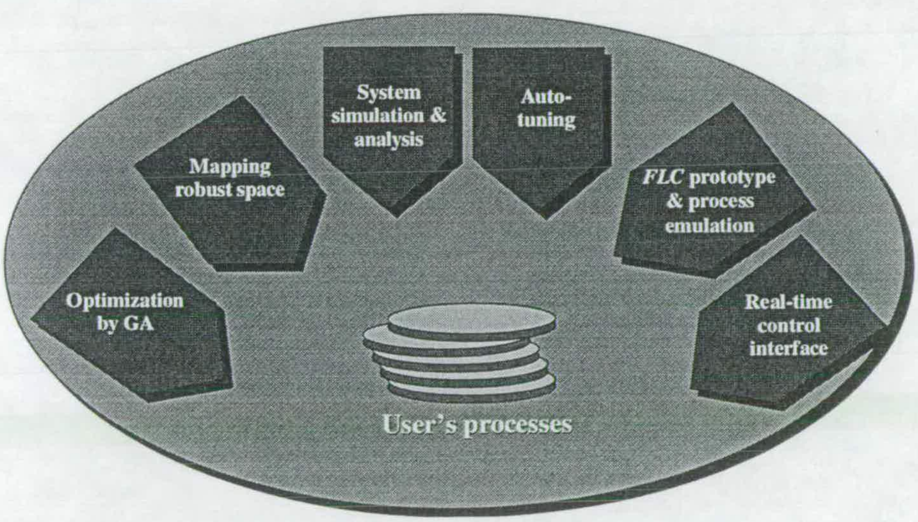


Fig.4-1 Simulation tools in *FuzySimu*

A FLC prototype was designed for the purpose of confirming the results produced by the simulator *MultiSimu*. A Motorola 68HC11 microprocessor based fuzzy controller and a third order linear SISO process emulator have been developed in a single printed circuit board. A software interface *RCPlot* based on the RS-232 communication method was implemented to (1) automatically transfer the controller parameter settings in the simulator into the practical prototype; (2) download the program from PC to the prototype, then run or debug it; (3) collect data from the prototype and show the system performance on the screen.

In the following, a brief description of each function of the simulation package will be presented, except for the auto-tuning function which will be presented in *Chapter* 5. More details about this software package can be found in the *Appendix A* and *Appendix B*.

4.3.1 System simulation and analysis

MultiSimu is a multi-objective simulation tool which can be used to simulate different control algorithms and different controlled processes. The block diagram of *MultiSimu* is shown in Fig.4-2. It consists of two levels: a supervisory level and a system level. The supervisory level is in charge of the pre-setting of the controller parameters and the simulated process parameters. It provides an on-line configuration, *i.e.* setting simulation and system parameters manually, and an off-line configuration, *i.e.* setting simulation and system parameters according to an external configuration file. The controller base currently includes the general FLC, phase advanced FLC, PID controller and open loop control. The process base currently includes processes described in *Section* 4.2. Any new controllers or processes can be easily added in the corresponding base.

At the system level, calculations are performed on the controller's action and the transient response of the process. The random noise, load disturbance, non-linear properties (saturation, dead zone, limited resolution in digital and analogue conversion, and different forward or backward gains) can be inserted in to the control system. Three types of reference signals: step, square wave and saw tooth, are provided. Measurements for the system performance are also computed at this level, and, if required, are output to the external routines which call *MultiSimu*. In addition, the controller action and the process output can be saved in a file or plotted on the screen, in form of a transient response, a phase plane trajectory, a control surface or a switching line.

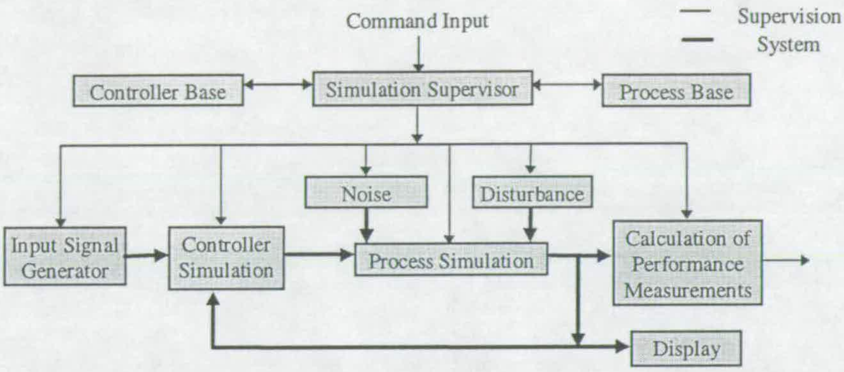


Fig.4-2 Block diagram of *MultiSimu*

4.3.2 Mapping of the robust space

RobustMap is a mapping tool which performs a searching task in the user defined process parameter space. Determination is made of the size and position of the so called *Robust Space* within which the system's performance satisfies the pre-defined criteria. To do this, *RobustMap* initializes and calls *MultiSimu* successively, and produces a data file which can be processed by Microsoft Excel or directly displays the robust space on the screen. The block diagram of the program is shown in Fig.4-3.

Like *MultiSimu*, *RobustMap* can be configured manually or based on a configuration file. The configurable variables include the variation range and search step of each process parameter, controller settings, performance specifications, the mapping method and the mapping task.

The mapping method consists of an exhaustive search (test all parameter points) and a boundary search (only test the points on the boundary of the robust space). The former method is designed for mapping the robust space of control systems containing processes with high non-linearity, where the robust space of the control system may contain several separated blocks, and the later is for mapping the robust space of control systems containing linear or slightly non-linear processes, where the robust space of the control system contains only one block. The mapping tasks include mapping the robust space and testing the

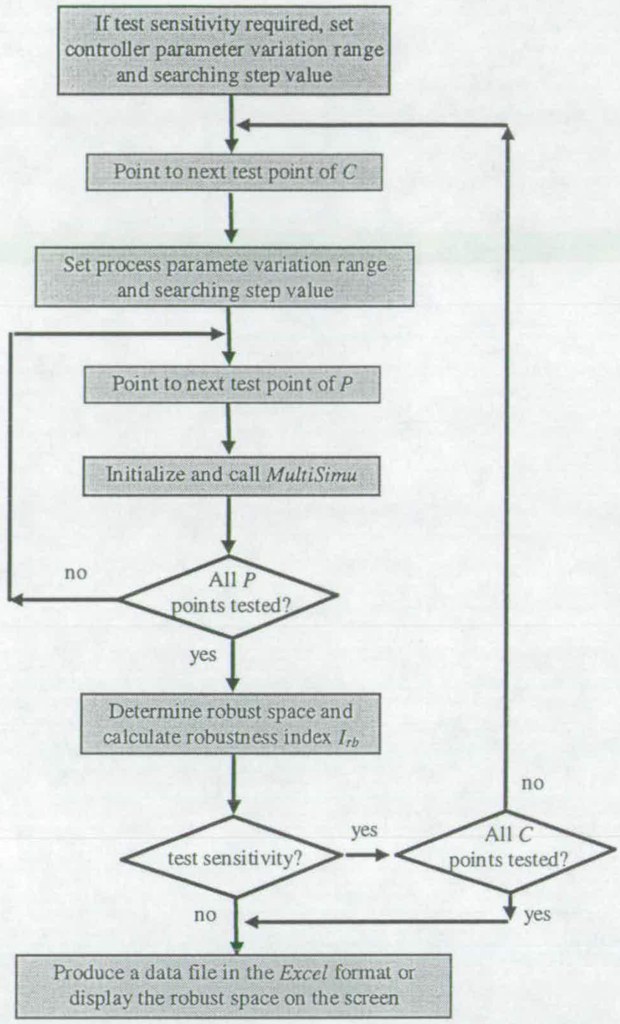


Fig.4-3 Block diagram of *RobustMap* (*C* for controller parameter, *P* for process parameter)

sensitivity of controller parameters.

4.3.3 Parameter optimization

GaOpt is a tool to optimize the robust performance by automatically selecting the controller's parameters including the fuzzy membership functions, rule base, scaling factors and sampling rate. It iteratively explores a population of solutions in parallel, and evolves a new generation by performing genetic operations on each solution. At the begin, the initial generation of FLC solutions are read from an initial file or are chosen randomly. After initialization, *GaOpt* performs an iterative loop: evaluation, convergence determination, and reproduction of new FLC generation. Fig.4-4 shows the block diagram of *GaOpt*.

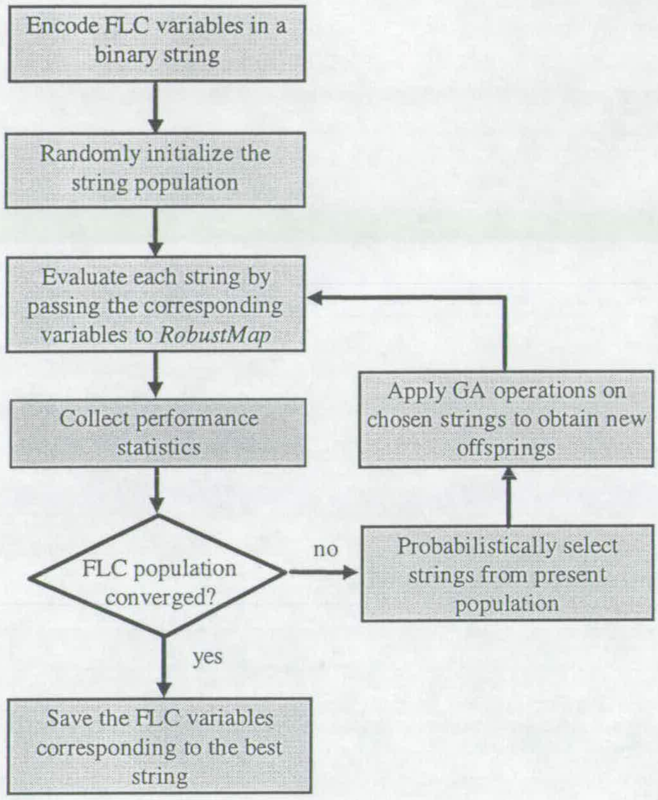


Fig.4-4 Block diagram of *GaOpt*

In the evaluation stage, *GaOpt* passes each solution (*i.e.* controller parameter group) to *RobustMap* to evaluate the robustness of the system with this solution and determine the fitness by means of the fitness function defined in *Chapter* 3. After evaluation, the statistics of the robust performance of each solution in the present FLC generation will be collected and the convergence of this searching process will be determined. If the FLC population has not converged, *GaOpt* will randomly select some FLC solutions from the present population. The selection algorithms implemented in *GaOpt* are proportional-selection based on the stochastic universal sampling algorithm and linear-ranking-selection both proposed by James E. Baker in [84] and [87] respectively. In the proportional selection scheme, each solution, c_i, is assigned the following fitness-proportionate selection probabilities:

$$p(c_i) = \begin{cases} \dfrac{fit(c_i)}{\sum_{j=1}^{k} fit(c_j)}, & 1 \le i \le k \\ 0, & k < i \le n \end{cases} \tag{4-3}$$

where k is the number of solutions to be selected, n is the total population. In linear ranking selection, the selection probabilities are generated based upon the relative ordering of the absolute fitness values of individual solutions, *i.e.* their indices i in the sorted population. The selection probabilities are calculated as

$$p(c_i) = \begin{cases} \frac{1}{k}(r_{\max} - (r_{\max} - r_{\min})\frac{i-1}{k-1}), & 1 \le i \le k \\ 0, & k < i \le n \end{cases} \quad (4\text{-}4)$$

where k is the number of solutions to be selected, n is the total population, r_{max} and r_{min} are maximum and minimum fitness values respectively. More detailed information is given in [68].

After selection, GA operations, mutation and crossover, will be applied to the selected solutions to form a new generation of FLC solutions. In the mutation operation, each position in the encoded string is given a chance r_m (mutation rate, default value is 0.001) of undergoing mutation. If mutation does occur, a random value is chosen from {0, 1} for the selected position. In the crossover operation, the adjacent pairs of the first $r_c n$ (r_c denotes the crossover rate, default value is 0.6; n denotes the population) FLC solutions in the selected population exchange information contained in a randomly selected segment, *i.e.* two-point crossover. Note that the subfunctions related to *Crossover*, *Mutate*, and *Select* in this program were adopted from *Genesis* 5.0 written by Jone J. Grefenstette.

Note that because of the simulation requirement and wide parameter variation range, evaluation is very time consuming so that the *GaOpt* function has been mainly performed on a Unix Sun work station.

4.3.4 FLC prototype

Because the experimental investigation of the robustness of the FLC systems has been entirely carried out in computer simulations, it is crucial that the simulation results are in conformance with the actual performance of a practical control system. Therefore, an FLC prototype, as shown in Fig.4-5, was designed for the purpose of confirming the results produced by the simulator *MultiSimu*. It contains a Motorola M68HC11 microprocessor, a

12-bit D/A converter, 32k RAM, 8k EPROM/RAM, a RS232 communication interface and a display/keypad port. The reason for using the Motorola 68HC11 microprocessor in this prototype is its simplicity and flexibility for development work. The 8-bit A/D converter on the microprocessor chip is used to sample the system output. The 12-bit, 4-quadrant D/A converter MAX501 is used to produce the control input. The RAM is used to store the dynamic performance of the system. The EPROM is used to store the control program. The RS-232 interface is built in to facilitate controller debug or any high level supervisory management. The display/keypad port can be used to display system status and input user commands when the controller works in stand-alone mode. The software for the prototype is designed using assembly language. Its block diagram is in Fig.4-6.

To confirm the simulation results from *FuzySimu*, the controlled processes are also needed in addition to the controller. Thus an operational amplifier base emulator of the process is designed. With this emulator, a first order process and a second order process are implemented to provide the system output information (see appendix B).

4.3.5 Real-time control interface

To develop a microprocessor embedded control system, a development tool is needed to perform the program downloading, debugging and evaluation. In addition, frequent changes in FLC parameters and comparison between the performance of simulated and practical systems requires automatic data processing. That is, this tool should have some CAD functions, not only a microprocessor development kit. Based on the above requirement, a control interface between the FLC prototype and the supervisory PC, called *RCPlot*, was developed. It has the following functions:

1) Automatically transfers the controller parameter settings in the simulator to the prototype program, and compiles the prototype program.
2) Downloads the prototype program to the prototype, runs or debugs it.
3) Transfer the dynamic performance data logged in the prototype to *FzySimu*, calculates all performance measures and shows the system's performance on the screen together with the simulation results.

For function 1, the program used in the prototype is organized as several function blocks:

monitoring and timing block (MT), PID block, FLC block, and configuration data (CD) segment, with the controller parameters in the configuration data block. All these blocks are written in assembly language. When the configuration of the prototype is required, a configuration data block will be created and a complete prototype program will be organized according to the simulation configurations. Then this prototype program will be compiled into machine code which is ready to run in the practical controller.

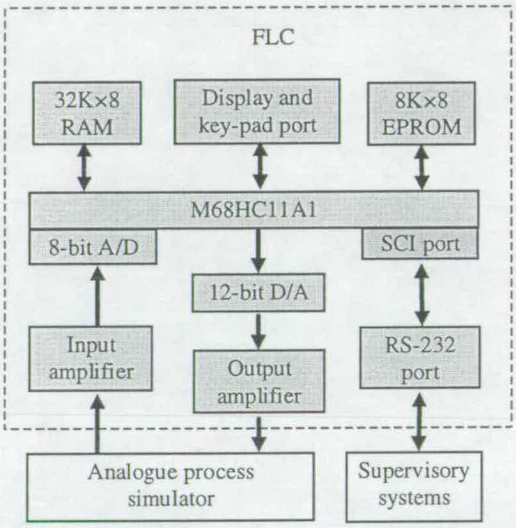


Fig.4-5 Structure of the FLC prototype

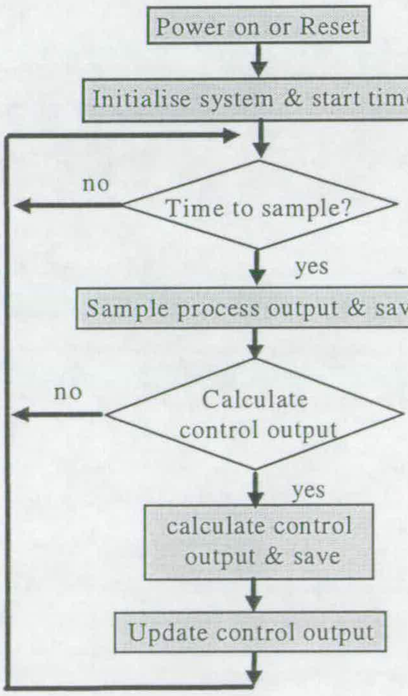


Fig.4-6 Block diagram of prototype software

For the communication task implied in functions 2 and 3, the M68HC11 is set to bootstrap mode, in which a communication program, called *Talker*, is automatically loaded from the host computer, after the microprocessor is powered up or reset. With *Talker*, any data transmission in serial format can be performed through the RS-232 standard port. By using 68HC11 development software, *PCBUG11*, the debug task in function 2 can be realized. Fig.4-7 shows a block diagram of *RCPlot*.

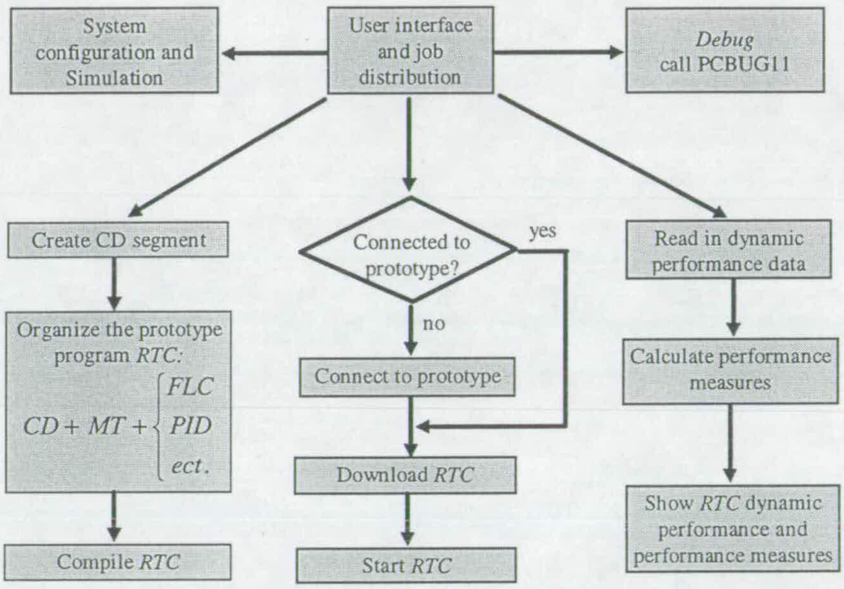


Fig.4-7 Block diagram of *RCPlot*

4.3.6 Designing a FLC system with *FuzySimu*

With help of the *FzySimu* package described above, the procedure to design a FLC system can be briefly stated as follows, assuming that a process model (either fuzzy or mathematical) and a set of performance specifications are given :

1) Set the parameters of the fuzzy logic controller, e.g. membership functions associated with each input and output variable, rule base, and the scaling factors. If any parameter is not available, the default one may be used.
2) Test the performance of the system using *MultiSimu*. Run *AutoTune* to tune the FLC's scaling factors to improve the performance if necessary.
3) Measure the robust space within pre-defined process' parameter space.

4) If the robust space is smaller than required, optimize the FLC's parameters using *GaOpt*.
5) Repeat step 3 and 4 until the robustness of the system is satisfied.
6) Test the above results by applying the FLC's parameters to the FLC prototype. If there is any obvious difference between the real performance and the simulation results, check if there is any mismatch between the real process and the model.

Note that a process model may not be available. There are many publications about how, given a process, to generate a suitable model. If the process model cannot be identified by any method, it is difficult to use any simulation based design tool to optimize the performance.

4.4 Experimental investigation of robustness of FLC systems

Investigating the robustness of the fuzzy control algorithm experimentally requires comprehensive information on system performance with variations of system structure and parameters. With the help of the simulation package *FuzySimu*, it is possible to explore the control performance of the entire system parameter space for which system robustness is concerned. In this section, the robustness of the FLC system will be tested for two general cases: first order process system and second order process system. A comparative study against the widely used PID control system will be presented in both cases. For each experiment, the parameter space will be given; the controller, the performance specifications and the experimental procedure described in *Section* 3.2 will be used.

4.4.1 First order process with time delay

The transfer function of the time delayed, first order process is given as;

$$G(s) = \frac{Ke^{-\tau s}}{1+Ts}. \tag{4-5}$$

Its parameter variation ranges are assumed to be K= (1.0~70.0), T = (50~8000), τ = (0 ~ 100). The unit used for the time constant, the time delay and the controller's sample rate was the process simulation sample period.

To explore the robustness of the FLC system with the above first order process as the controlled process (first order system for short), extensive simulation experiments were performed to test the system performances in the parameter space $K \times T \times \tau$. The step changes for K, T and τ were selected as 0.1, 50 and 100 respectively, thus the total number of test points in the parameter space was 224000. The input reference of the system was changed from $w = 0$ to $w = 2.5$ and all controller input/output signals were assumed to be constrained by a saturation with limits at ±10. The system was manually tuned at $K = 20$, $T = 4000$ and $\tau = 0$ to obtain a satisfied step response, giving the following scaling factors and the controller sample rate:

GE = 1.0, GC = 20.0, GU = 1.0, $t_s = 20$.

The robust space of the system without time delay was first tested. The results are shown as the shaded area in Fig.4-8a, where two scalings were used for the axis K to provide a clear view of the robust space in the low gain area. Step performances at 14 selected points as shown in Fig.4-8a were presented in Fig.4-8b, c and d.

From Fig.4-8, it is clear that the robust space of the first order system is a sector in the K-T plane, as predicted in the *Chapter* 3. In the high gain and small time constant situation (points d, g, h), the system generates a fast response so that oscillation of $\dot{e}$ around the switching line was exhibited before the output y reaches the setpoint (see Fig.4-8b). In low gain and large time constant situations (points b, c), the system gives a slow response so that large overshoot and undershoot were generated (see Fig.4-8c). When the time constant exceeds 6000 sample units (points f, i), the step response cannot meet the performance specifications, mainly due to the high overshoot generated by a relatively fast response of the controller (see Fig.4-8c). With a time constant less than 500 sample units (point 5), the step response is too slow, simply because of the relatively slow sample rate of the controller (Fig.4-8d).

The switching line feature of the FLC systems can also be found from the results. System performance at the tuning point (point e) is shown in Fig.4-8b. Similar performance was obtained at other parameter points inside the robust space. For example, in Fig.4-8d, the main part of the step responses at point 1 to 4 are nearly identical, presenting closely similar phase plane trajectories.

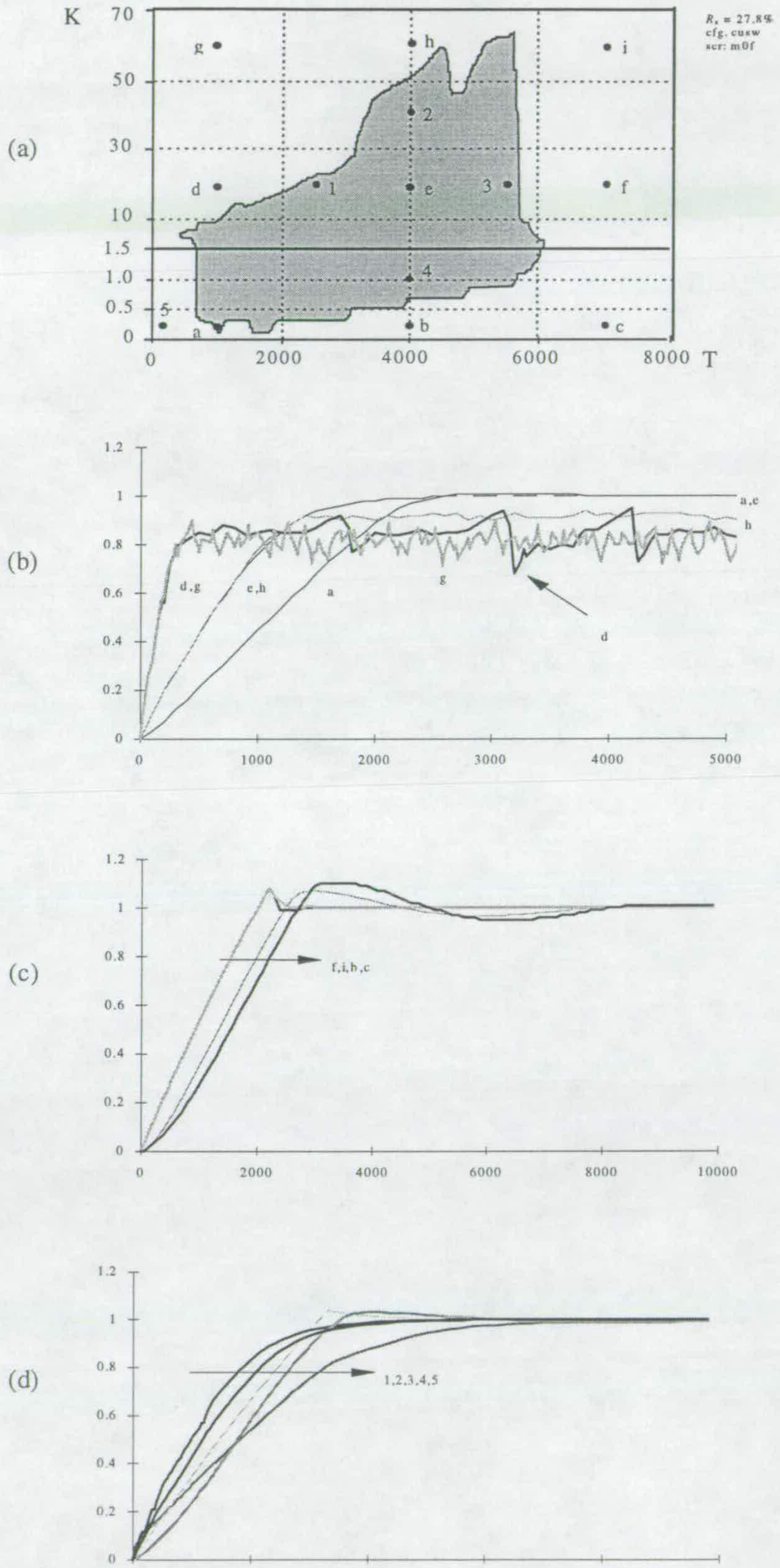


Fig.4-8 Robust performances of the first-order system
(a) robust space; (b)-(d) step performances at point a ~ i, 1 ~ 5 in (a)

Note that the step responses at points d, g and h have a positive steady state error. This results from the fast response speed (small time constant but large gain *K*). A small change in the control input would lead to big changes in the process output, so that the reach condition of the fuzzy controller cannot be satisfied. On the phase plane, the process output trajectory exhibits an oscillatory characteristic around the switching line. In addition, because the BIG control changes are used to restrain big overshoot when the output is becoming larger than the setpoint, according to the rule base in Fig.3-4c, it is much more difficult for the system output to reach the negative error zone than to stay in the positive error zone. Thus when an oscillation of phase plane trajectory around the switching line occurs, a positive system error is usually produced.

Similar tests were performed with a time delay, $\tau = 100$, added to the controlled process. The results are shown in Fig.4-9(a). From this figure, it can be seen that the robust space of the system is significantly reduced when the time delay is added, but it is still a sector shifted to the low gain area in K-T plane.

As expected from traditional control theory, system robustness is more sensitive to the process gain than to the time constant in a time delay system. Compared with the simulation results for $\tau = 0$, the robust space of the time-delay system is reduced more in the K direction (from 60.0 to 4.0) than in the T direction (from 6000 to 3700). The main reason for the reduced robust space is the oscillations caused by the relatively high gains.

Parameter variations around the robust space produce the same effect on the step response as in $\tau = 0$, except for the large output oscillations at high gain and small time constant. The effect of the time delay on system performance can be found from Fig.4-9(b - e). Clearly time delay impairs the switching line performance. From Fig.4-9e, although the responses at point 2 and 3 meet the performance specifications, the output trajectories are quite different. Moreover, time delay increases the overshoot or undershoot, which can be found in the step responses at points c, f and b. All these results demonstrate that the controller cannot keep the system state $(e, \dot{e})$ on the designed switching line, thus system performance is more sensitive to the parameter variations in time delay systems. This is an important observation which confirms the argument presented in *Section* 3.4.1.

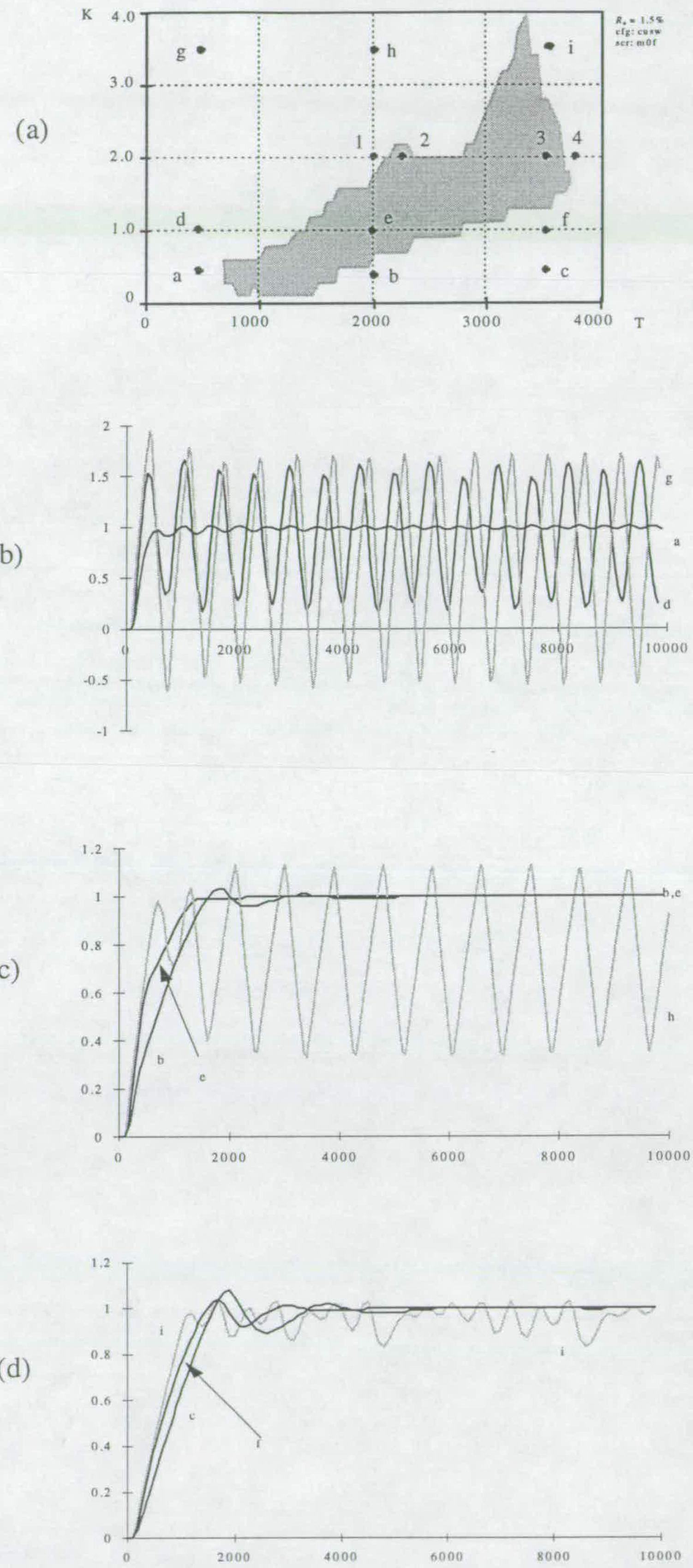


Fig.4-9a Robust space of the first-order system with time delay (τ=100)
(a) robust space; (b) to (e) step input performances at selected points in (a)

(e)

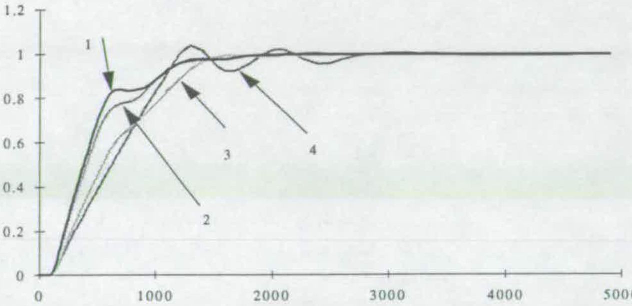


Fig.4-9 Simulation results of the first-order system with time delay (τ=100)
(a) robust space; (b) to (e) step input performances at selected points in (a)

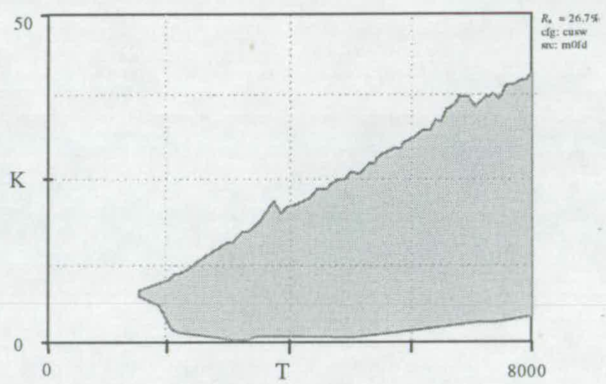


Fig.4-10 Robust space of the first-order system with time delay (τ=100) after re-turning.

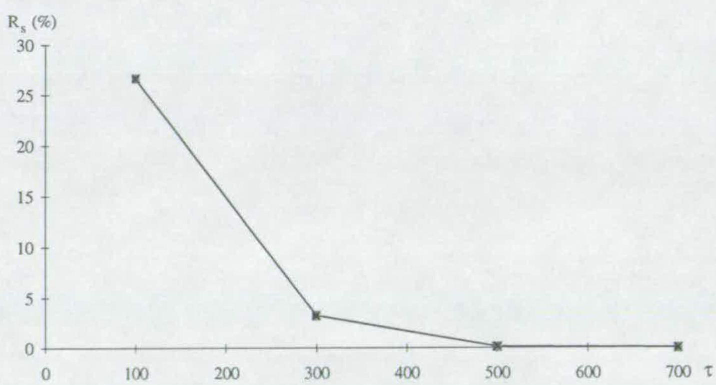


Fig.4-11 Illustration of the effect of time delay on system robustness
(tuning point: τ = 100)

Note that another reason for the decreased robustness in the time delay case, is that the system was not retuned when a time delay was introduced. If the system is tuned when the time delay exists, a bigger robust space than that in Fig.4-9(a) can be expected. Fig.4-10 shows the robust space of the first-order system when the controller was tuned at K = 5, T = 2000 and τ = 100, by setting the scaling factors and the controller sample rate to: GE = 1.0, GC = 10.0, GU = 0.2, t_s = 100.

Comparing with the robust space shown in Fig.4-9, the robustness of the same system was increased by 25%. This result clearly demonstrates the effect of the tuning point on system robustness. The reason for the increase in the robust space is that the re-tuned system was much slower than the first one (t_s changed from 20 to 100) and every change in the control input Δu was more carefully applied in the re-tuned system than in the first system (GU changed from 1.0 to 0.2). Thus the controller in the re-tuned system can tolerate more changes in the process gain than the previous one. Therefore, the tuning point should be carefully selected in practical systems if robustness is a particular requirement.

In addition, if the time delay varies, system robustness changes dramatically. Fig.4-11 shows the robustness in the $K\times T$ plane as a function of time delay τ. This demonstrates the predicted result in *Chapter* 3 that the basic FLC cannot properly handle the time delay, especially when the process gain is high.

In conclusion, extensive experimental tests show that FLC systems possess a strong ability to cope with the variations of parameter K and T in the first order process, but are less able to tolerate time delay variation. In addition, it is demonstrated that the robustness of the first order system stems from the switching line feature of the fuzzy logic controllers.

4.4.2 Second order process

The transfer function of the second order process is given as

$$G(s) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \qquad (4\text{-}6)$$

Its parameter variation ranges are assumed to be K = (1.0 ~ 70.0), ω_n = (1.0 ~ 20.0), and ζ = (0.5 ~ 1.5).

It is shown in *Chapter* 3 that the robustness characteristic of the FLC controlled second order process (second order system for short) can be expected to be different from that of the first order system due to the oscillatory characteristics of the process. The effect of ζ on the robust space of the system is equivalent to the effect of the time constant in the first-order system. The effect of ω_n on system robustness will be different because its variation only influences the response speed but not the overshoot. In this section, simulation results will be presented to demonstrate these results for $\zeta \leq 1.5$.

Extensive simulation experiments were performed to test second order system performances in the parameter space $K \times \omega_n \times \zeta$. The step changes for K, ω_n and ζ were selected as 0.5, 2.0, and 0.01 respectively. The input reference of the system was changed from $w = 0$ to $w = 2.5$ and the controller input/output signals were assumed to be constrained by a saturation with limits at ± 10. The system was tuned at $K = 10.0$, $\omega_n = 5.0$, $\zeta = 0.7$ and the following controller scaling factors were obtained:

$$GE = 1.2, GC = 10.0, GU = 0.01, t_s = 16.$$

The robust space on the K— ζ plane was tested first and the result is presented in Fig.4-12. Step performances at different points in the parameter space are presented to illustrate FLC performances for different operating situations.

From Fig.4-12, it can be found that the robust space of the second order system on the K — ζ plane exhibits a sector with shape similar to that predicted by the qualitative analysis presented in *Chapter* 3, except for a cut-off within a band of gain K. The reason for the cut-off phenomenon in the robust space can be found in the system step performances at points c, f and h. At point f, the process generated a response faster than the speed required by the switching line when the process response is near the set-point. Due to the deviation of the phase plane trajectory from the switching line, the FLC tries to bring the phase plane trajectory back to the switching line by reducing its control output. Because the controller acts relatively faster than the process response, the control u is reduced too much so that a big undershoot is generated. At point h, the controlled process gives the same response speed as at point f, and the run-away phenomenon happens again. However, the process response speed is increased (because K is increased at point h, compared with point f) and the control u is not reduced too much, thus no big undershoot can be produced. Also at point c, the response speed is much slower (because of the low gain) than the response

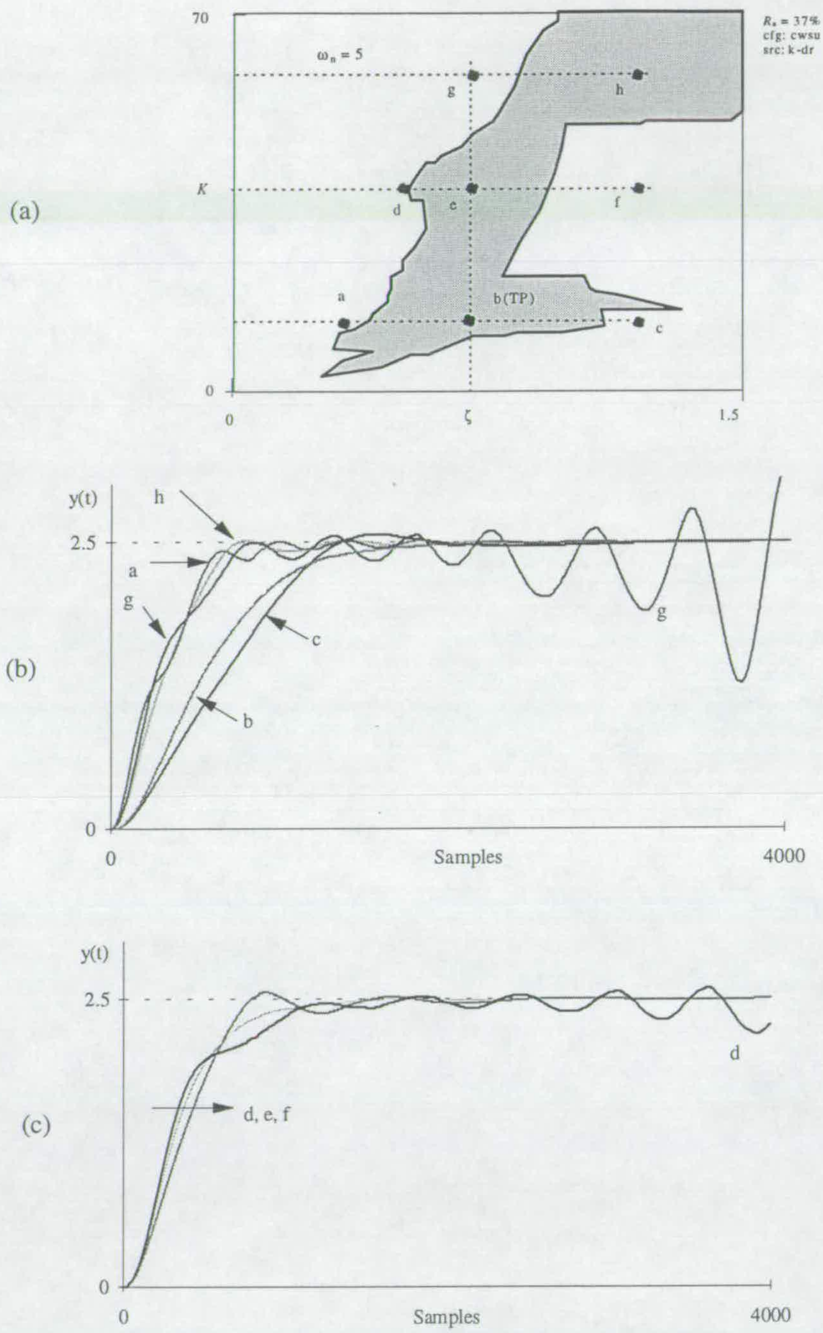


Fig.4-12 Robust space (on the K— ζ plane) and step responses of the second order system

speed at point f, so the FLC produces a different control action from that at point f, thus the run-away of the phase plane trajectory is not generated. Clearly, the cut-off phenomenon in the robust space results from the non-linear control property of the FLC system.

From the step responses shown in Fig.4-12, one can find that the system performances at points a, d, e, f, g and h exhibit a similar rising speed, which indeed confirms the switching line performance of a FLC system. In addition, step responses are oscillatory for small ζ (points a, d, g) and slow for large ζ (points c, b). These observations confirm the predictions made in *Chapter* 3. Note that the performance at point c is out of specification because at this point a big jump of settling time is caused by a small increase in the undershoot.

The robust space on the K— ω_n plane was also tested. The results and step responses at different points in the parameter space are presented in Fig.4-13. These results differ significantly from the robust space of the same system on the K — ζ plane. In Fig.4-13, it can be seen that the relationship between K and ω_n on the boundaries of the robust space can be approximated as

$$K = a\omega_n^2 + b$$

(a and b are constants), and the limitation on the maximum gain K can be clearly seen from the measured robust space, confirming the qualitative analysis in *Chapter* 3. This confirmation further demonstrates the validity of the use of the switching line method in analyzing fuzzy control systems.

It can be found from Fig.4-13c that, though the step responses at parameter points d and g are similar to that at point e, the system performances at points d and g are out of specification. Close examination reveals that the reason for this is that the settling time at point d or g is more than the specified settling time (defined as 3 times the equivalent time constant, *i.e.* $1/\zeta\omega_n$). The same reason can be found at point a. If a fixed settling time specification were used, *i.e.* 3 times of the equivalent time constant at the tuning point b, extension of the robust space would be expected in the direction of the large ω_n. Therefore, the settling time specification defined in *Section* 4.2.2 is more strict than the fixed one.

Similar to the robustness test in the first-order system, the effect of the position of the tuning point on the size of the robust space is found to be significant. For example, if the second order system is re-tuned at K = 10.0, ω_n = 2.0, ζ =0.8 to achieve a similar tuning point performance to that shown in Fig.4-13b (curve b), the robustness on the K— ω_n plane changes from the previous 15.7% to 5.4%.

In addition, the quality of the performance at the tuning point can affect the size of the robust space. In Fig.4-13, the performance measures of the step response of the second order system at the tuning point are

OS = US = 0%, ST =3.0, CY = 0, IES = 0.9.

If the system is re-tuned at the same tuning point to provide a faster and smooth response by changing the sampling period t_s =12 and GU = 0.007, the performance measures become

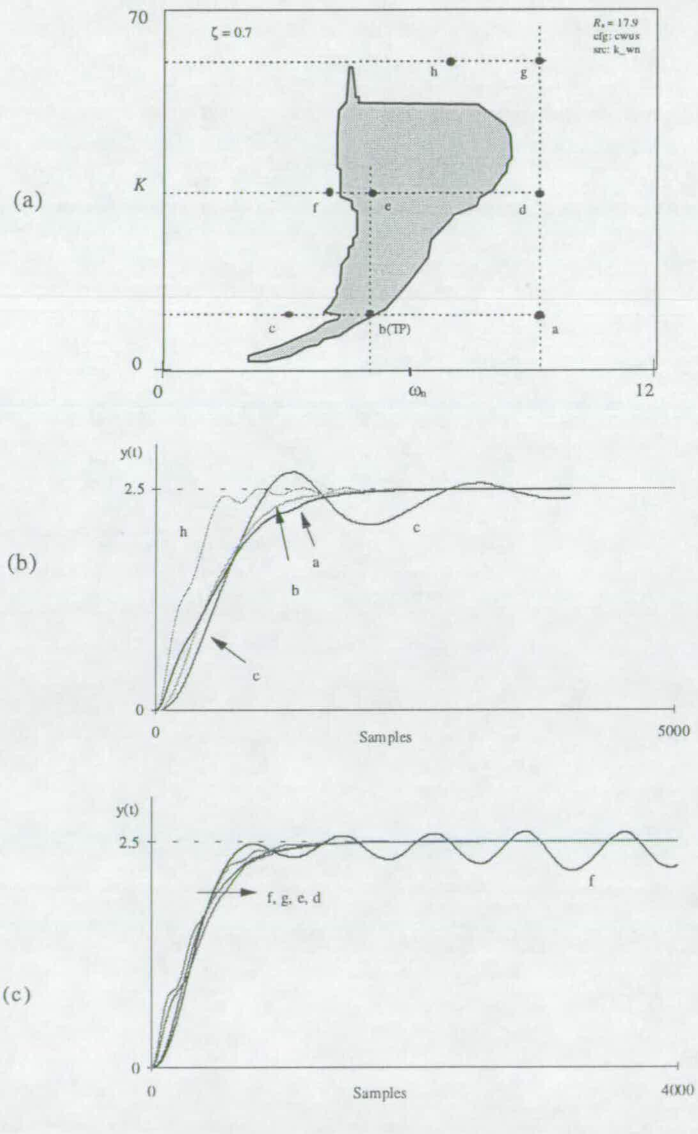


Fig.4-13 Simulation results of the second order system
(a) robust space on the K— ω_n plane;
(b) (c) step responses at point a to h

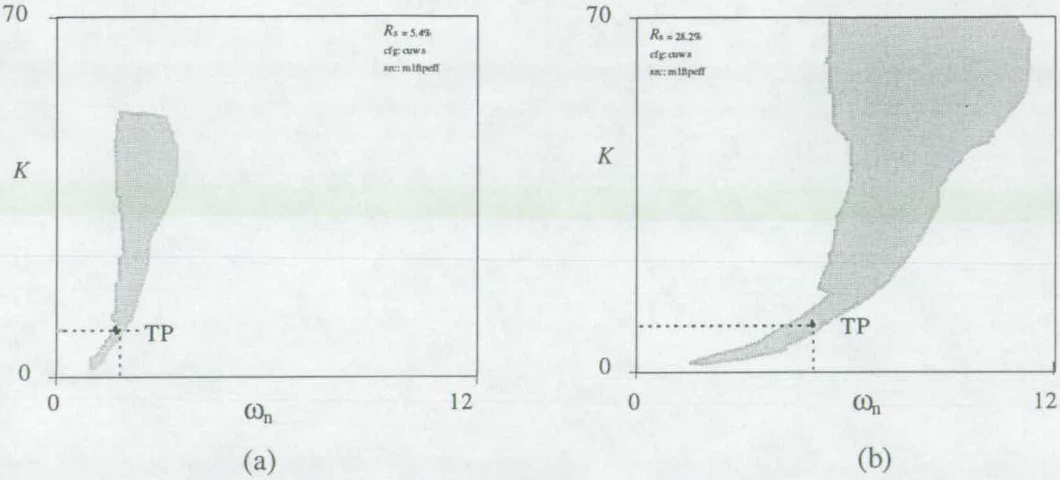


Fig.4-14 Effect of tuning point on system robustness
(a) effect of tuning point position;
(b) effect of performance quality at tuning point

OS = US = 0%, ST = 2.3, CY = 0; IES = 0.8,

and the robustness on the K— ω_n plane changes from 15.7% to 28.2%.

The robust spaces of the two re-tuned systems are shown in Fig.4-14. Comparing with the robust space shown in Fig.4-13, it can be found that the location of the tuning point and the performance quality at the tuning point not only affect the size of the robust space, but also determine the position of the robust area in the parameter space.

4.4.3 Comparative study with PID control systems

This section will be dedicated to performing an experimental investigation of the robust step performance of SISO systems controlled by FLC and PID algorithms. The objective is to quantify the claim that fuzzy control leads to an improved robust performance compared to the use of PID control. It should be noted that the reason for choosing to use PID systems is that the PID controller is well recognized as a robust controller with respect to input disturbance, system parameter variation and model uncertainties [1].

The robustness of two control systems will be tested based on the system designs presented in *Section* 4.2. The experimental procedures and the performance specifications presented in *Section* 4.2 will be used in both situations. Two types of controlled processes: first order process and second order process, will be investigated and their transfer functions and the parameter variation ranges are chosen to be the same as used in the previous section.

In addition, as only 4 parameters (K_i, K_p, K_d and t_s) in the PID system can be tuned to obtain the required performance, the tuned parameters of the FLC will be restricted to the scaling factors and the sampling rate, *i.e.* the membership functions and rule base in the FLC will not be changed.

To make the two types of systems comparable, identical process parameter variation ranges were chosen, and the two systems were tuned separately to meet the STPP condition.

A. First order process

In the simulation experiments, the parameter variation range of the first order process is chosen as K=(0.5~20.0), T=(50~2000), τ=(0~200). Tuning point was selected as K = 10.0, T = 1000, τ = 100. The following controller parameters were obtained:

FLC: GE = 0.5, GC = 12.0, GU = 0.075, t_s = 24;

PID: K_p = 0.22, K_i = 0.0065, K_d = 0.5, t_s = 24.

The tuning point performances of these systems are shown in Fig.4-15. The robust space of each system were tested and is shown in Fig.4-16. Note that the percentage numbers in Fig.4-16 are the system robustness in this test.

In Fig.4-16, some robust areas of the FLC system have a flat edge, say at points *a* and *b*, inside the tested parameter space. The shape of the tested robust area seems unlike the predicted shape in *Chapter* 3. The reason for this is that the whole robust area is larger than the robust area shown in the figure, therefore, the pictures in Fig.4-16 are the local and amplified view of the robust area. Fig.4-17 shows the step responses at two edge points: *a* (K=5, T=850, τ=0) and *b* (K=15, T=850, τ=0). It can be seen in Fig.4-17 that in both cases the control inputs are not saturated and the settling time is about 3 times the process time constant, *i.e.* 2550, which is the maximum settling time required by the settling time specification. As the time constant T decreases, the step response cannot reach the steady state within the specified settling time because of the switching line characteristic of the FLC.

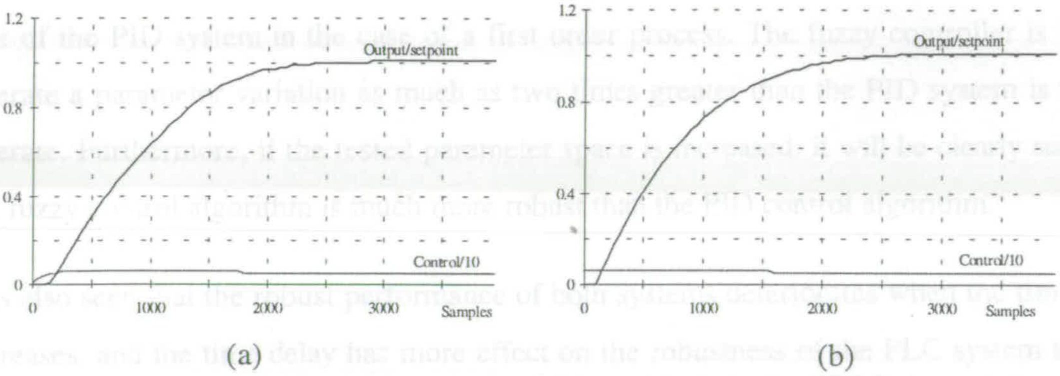


Fig.4-15 Step input response at tuning point. (a) FLC system; (b) PID system

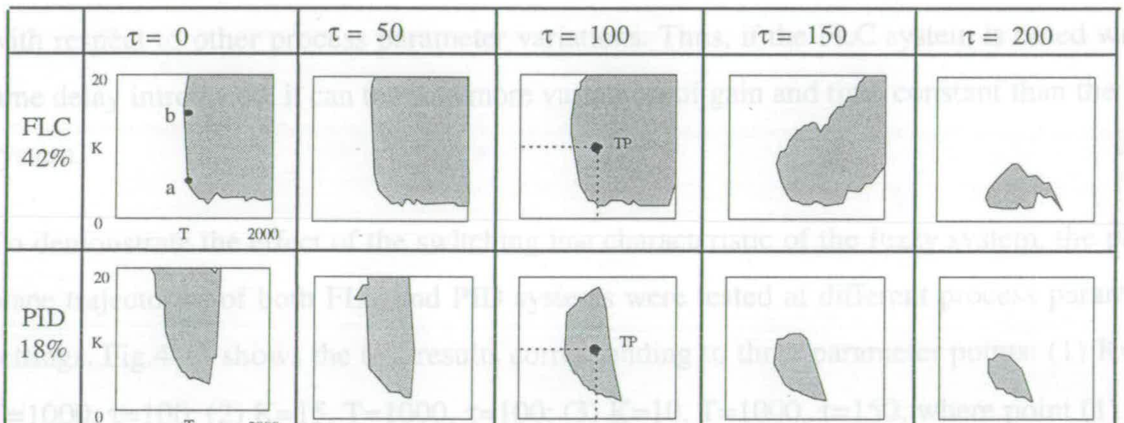


Fig.4-16 Robust spaces of FLC and PID controlled first order systems

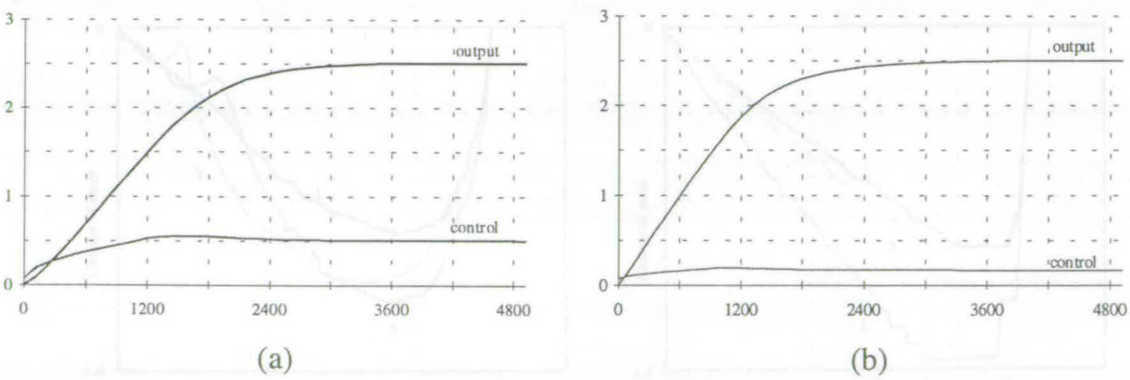


Fig.4-17 Step responses of the FLC system at the edge of the robust space

(a) K=5, T=850, τ=0; (b) K=15, T=850, τ=0.

gain change (trajectory 2) is smaller in the FLC system than in the PID system. Similar results can be obtained when the time constant T changes. However, the deviation that resulted from a time delay change is much bigger in the FLC system than in the PID system. These results demonstrated that, compared with the PID system, the FLC system is less capable of tolerating time delay changes, but is more robust under variations of the other process parameters.

B. Second order process

The same tests were performed for the second order linear process. In these tests, the parameter ranges were selected as K=(1.0 ~ 70); ζ = (0.5 ~ 1.3); ω_n = (1.0 ~ 80). Both FLC and PID systems were tuned at K = 10, ω_n = 40 and ξ = 0.9, and the following controller parameters were obtained:

FLC: GE = 1.0, GC = 6.0, GU = 0.02, t_s = 3;

PID: K_p = 0.045, K_i=0.005, K_d=0.2, t_s = 3.

Two step performances at the tuning point are shown in Fig.4-19. The robustness of each system was tested with the step changes as $\Delta K = \Delta\omega_n$ = 1.0, $\Delta\zeta$ = 0.2. Total number of tested parameter points was 28000. The results are shown in Fig.4-20. The robust space of the FLC system covers 4487 test points (16%), while that of the PID system covers 1529 points (5.5%).

From the location of the robust spaces of two systems, it can be found that the fuzzy logic controller can tolerate wider variations in gain K than the PID controller. Further investigation found that the overshoot of the PID system increases as K becomes larger, while the FLC system can maintain the overshoot almost constant by its switching line performance within a large range of K.

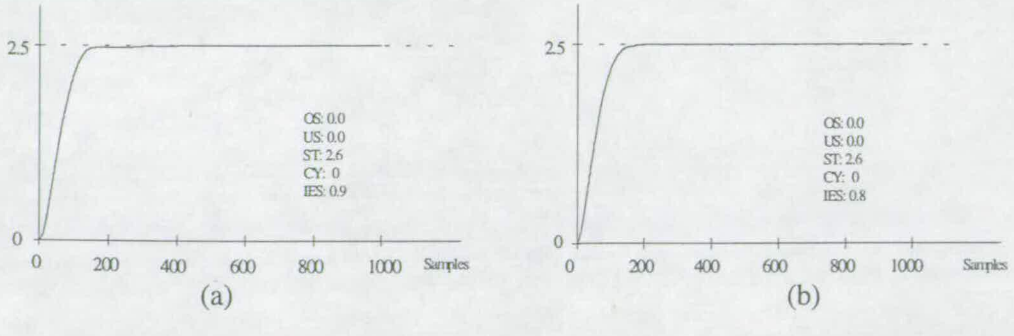
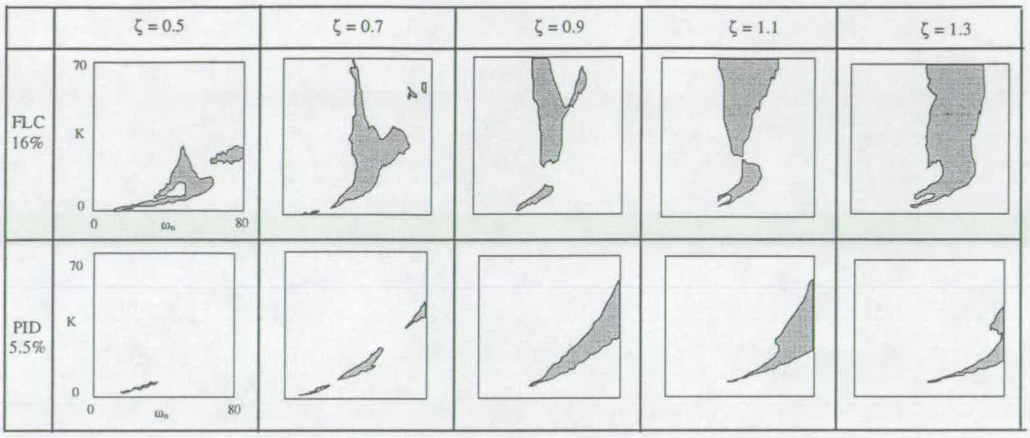


Fig.4-19 Step responses of (a) FLC and (b) PID systems at the tuning point



FLC system: cfg. cusw, src. m1f1610. PID system: src. m1p1610

Fig.4-20 Robust spaces of FLC controlled and PID controlled second order processes

From the variation of the robust areas in the $K - \omega_n$ plane, it is clear that the fuzzy logic controller is less sensitive to the changes of damping ratio than the PID controller. The reason can be explained by noting that the integral term in the PID controller cannot match the process response speed when ζ decreases from the tuning point, *i.e.* the dynamics of the process becomes faster, so that a large undershoot and/or a slow settling performance are produced by the controller.

From the width of the robust areas in Fig.4-20, it can be found that both systems are sensitive to changes of the frequency ω_n. As discussed in *Chapter* 3, ω_n in the second order process is associated with an oscillatory characteristic which is difficult to control without making use of the second order derivative of the system error. This problem will be further discussed in the next section.

It should be noted that the effect of the tuning point location and the tuning point performance on the system robustness cannot be neglected, especially the effect of the tuning point performance. For example, when both PID and FLC systems were tuned at

$$K = 10.0,\ \omega_n = 5.0,\ \zeta = 0.7\ ,$$

to achieve a step performance defined by:

OS = 0.0%, US = 0.0%, ST = 2.3, CY = 0, IES = 0.8,

the robustness of the PID and FLC systems became 4.5% and 2.7% respectively (the FLC parameters: GE = 1.2, GC = 10.0, GU = 0.002, t_s = 12; PID parameters: K_p = 0.024, K_i =

0.0023, K_d = 1.0, t_s = 12). In this case, the PID system is slightly superior to the FLC system.

Due to their non-linearity, FLC systems with different parameter settings can give similar performance, but their robust spaces will vary widely. From the author's experience, an improperly tuned FLC will not show any advantage over the PID system with respect to the robustness. The main reason for the low robustness of the FLC system in the above example is the slow sampling rate, which gives not enough time for the controller to adjust its control output before the system performance becomes unacceptable. Unfortunately, unlike the tuning of PID systems, there is no systematic method of tuning a FLC system for a required robust performance. This is one of the major drawbacks of the FLC systems.

Finally, the following conclusions can be drawn from the experimental investigations in this section.

(i) The robustness of the FLC systems stems from the constraints introduced by the switching line performance.

(ii) The switching line based qualitative analysis in predicting the shape of the robust space of the FLC systems is confirmed. The robust space of the first order FLC system is a sector in the *K-T* plane, and is significantly reduced when the time delay is added. The robust space of the second order FLC system is, approximately, a sector shape on the K — ζ plane and a curved sector shape on the K — ω_n plane.

(iii) Compared with a PID system, under conditions of similar tuning point performance, a properly designed and tuned FLC system is less capable of tolerating the time delay change, but is more robust under variations of process gain, time constant and damping ratio.

(iv) The performance of the fuzzy logic control system at the tuning point affects the position and the size of the system's robust space in the process parameter space. Careful tuning of a FLC is needed for obtaining the high robustness of FLC systems.

(v) The basic FLC designed in *Chapter* 3 exhibits less robustness in controlling high order processes than in controlling low order processes. Knowledge about high order dynamics of a controlled process should be included in the fuzzy rule base for controlling high order dynamic systems.

4.5 Effects of FLC parameters

It has been indicated in *Chapter* 3 that FLC parameters determine the switching line position, the reaching condition and the sliding condition of a FLC system and thus affect the system robustness. To design a robust fuzzy logic controller, it is essential to understand how and how much the controller parameters affect the system robustness. This section will be devoted to presenting an experimental investigation of the effects of the rule base, the membership function, and the scaling factors on robust system performance.

To investigate the effect of the rule base and the membership functions on system robustness, several fuzzy control systems with different rule bases and membership functions have been simulated, and the robustness of each system has been tested and compared with others cases. For a reasonable comparison of the system robustness, the scaling factors of each system were tuned to give similar performance at the same tuning point.

For testing the effect of each scaling factor on system robustness, the controller parameters were first tuned to obtain a specified performance at the tuning point, then the robust performance of the control system was tested at different values of this scaling factor, keeping other controller parameters unchanged.

The first order process with time delay was chosen as the controlled process, its parameter space was defined as

$$K = (1.0 \sim 70.0),\ T = (50 \sim 8000),\ \tau = (0 \sim 400),$$

and the tuning point was selected at $K = 35$, $T = 4000$ and $\tau = 100$.

4.5.1 Effect of rule base

Since the robustness of a FLC system is closely related to the switching line characteristic of the FLC, the position and the shape of the switching line will affect the robustness to a large extent. Therefore, emphasis has been made on the effect of the variations of the switching line on the system robustness. These variations of the switching line angle, and switching line shape were considered in this experiment. Five different rule bases have been selected as shown in Fig.4-21, and the robustness of the FLC system has been tested when each rule base was used. The results are shown in Table 4-1. Note that the default membership functions defined in *Section* 4.2 were used and the STPP condition was met in tuning these systems.

From Fig.4-21 it can be seen that the switching line angles of rule base (a), (b) and (d) at the steady state ($e = \dot{e}$ = ZE) are smaller than the switching line angles in rule base (c) and (e). Because the smaller switching line angle helps to limit the rising speed of the response, the fuzzy control system with rule base (a), (b) or (d) has less chance to produce large overshoot or undershoot, thus it is more capable of controlling large time constant processes. This is clearly shown in Table 4-1, where the fuzzy systems (a), (b) and (d) produced a similar robustness in the prescribed parameter space, while the robustness of the fuzzy systems (c) and (e) were only about 40% of that of the former three systems. This result demonstrates that the switching line angle can significantly affect the robustness of fuzzy control systems.

From the results in Table 4-1, it can also be found that the shape of the switching line does not make a significant difference to the system robustness. This results from the use of the same tuning point performance condition. The scaling factors of fuzzy systems with different rule bases have to be differently set so as to meet this condition. For example, rule base (d) allows a faster response than rule base (b) according to the switching line theory, but for the same tuning point performance, the scaling factor GE corresponding to the rule base (b) had to be set at twice the scaling factor GE corresponding to the rule base (d). That is, different scaling reduces the effect of the differently bent switching line on system robustness. If the same scaling factors are used in both cases, the different shapes of the switching line will lead to different responses. For example, when the scaling factors of two fuzzy systems (with rule base (d) and (e) respectively) were set as

GE = 1.0, GC = 12.0, GU = 0.05;

there is a big difference in two system performances when system error is small, as illustrated in Fig.4-22.

(a)

e \ $\dot{e}$	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NB	NM	NS	ZE	PS
NS	NB	NB	NM	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PM	PB	PB
PM	NS	ZE	PS	PM	PB	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB

(b)

e \ $\dot{e}$	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	PS	PM
NM	NB	NB	NB	NM	ZE	PM	PB
NS	NB	NB	NM	NS	PS	PM	PB
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NB	NM	NS	PS	PM	PB	PB
PM	NB	NM	ZE	PM	PB	PB	PB
PB	NM	NS	PM	PB	PB	PB	PB

(c)

e \ $\dot{e}$	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NB	NB	NM
NM	NB	NB	NB	NM	NM	NM	NS
NS	NB	NB	NM	NS	NS	ZE	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	ZE	PS	PS	PM	PB	PB
PM	PS	PM	PM	PM	PB	PB	PB
PB	PM	PB	PB	PB	PB	PB	PB

(d)

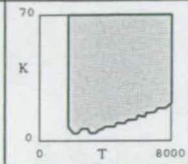
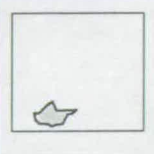
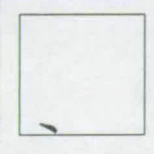
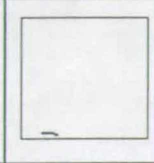
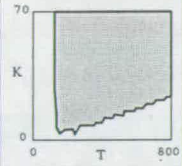
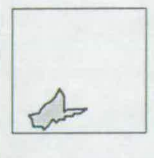
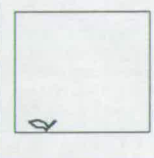
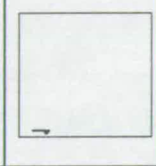
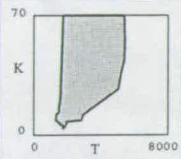
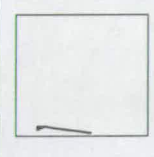
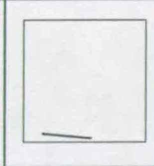
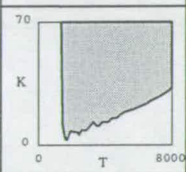
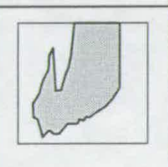
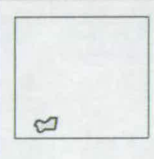
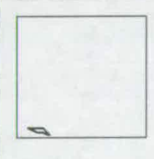
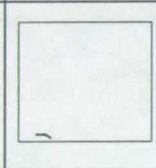
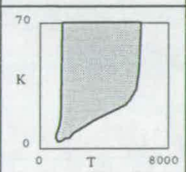
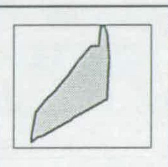
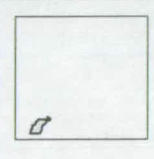
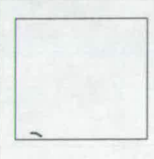
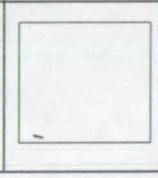
e \ $\dot{e}$	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NM	ZE
NM	NB	NB	NB	NM	ZE	PM	PM
NS	NB	NB	NM	NS	PS	PM	PB
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NB	NM	NS	PS	PM	PB	PB
PM	NM	NM	ZE	PM	PB	PB	PB
PB	ZE	PM	PM	PB	PB	PB	PB

(e)

e \ $\dot{e}$	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NB	NM	ZE
NM	NB	NB	NB	NM	NM	NM	PS
NS	NB	NB	NM	NS	NS	ZE	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	ZE	PS	PS	PM	PB	PB
PM	NS	PM	PM	PM	PB	PB	PB
PB	ZE	PM	PB	PB	PB	PB	PB

Fig.4-21 Rule bases used for testing the effect of rule base on system robustness

Table 4-1 Robust areas of the five fuzzy control systems using rule bases as shown in Fig.4-21

	$T_d = 0$	$T_d = 100$	$T_d = 200$	$T_d = 300$	$T_d = 400$	R_s
rule base (a)						12.9%
rule base (b)						13.9%
rule base (c)						7.0%
rule base (d)						10.9%
rule base (e)						7.5%

cfg: cusw. scr: m0f_rb.zip.

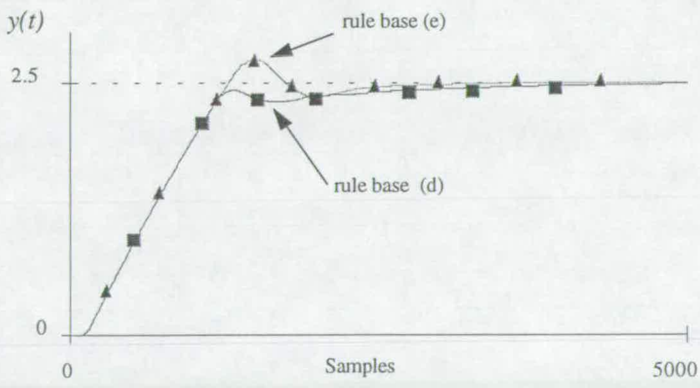


Fig.4-22 Illustration of difference in the step responses of fuzzy systems using rule base (d) and (e) in Fig.4-21, and with the same scaling factors

Although comparison among the above five systems indicates that the system with rule base (b) is the most robust, as far as process parameter variations are concerned, it should be noted that these comparison results can only be applied to low order linear process control. The objective of the comparison is to explore the function of the switching line for robust control, instead of finding the best rule base for robust control. Different control tasks may need different rule bases. Clearly the switching line will play an important role in robust control with fuzzy algorithms.

Note that although the STPP condition is important in the above comparison, it is difficult to tune the scaling factors of fuzzy control systems to produce the same performance when the switching lines are non-linear. For instance, a fuzzy control system with rule base (d) gives a fast start response but slow settling performance, while a fuzzy control system with rule base (e) produces a fast response in the whole transient period. Because the two switching line functions are different and non-linear, STPP condition can only be approximated by tuning the scaling factors. In the simulation tests, three performance measures, OS, US and IES, were used to determine the difference between the two step performances.

It is worth indicating that the scaling factors provide an effective method for tuning the fuzzy systems only when the switching line covers all the columns and rows of the rule base. If the switching line does not cover a row of the rules in the rule base, such as the rule base (c) in Fig.4-21, there will be an area on the phase plane $e - \dot{e}$ corresponding to this row of rules, within which any change of the scaling factor GC will have no effect on system performance. If the switching line does not cover a column of the rules in the rule base, like rule base (d) in Fig.4-21, there will be an area on the phase plane $e - \dot{e}$ corresponding to this column of rules, within which any change of the scaling factor GE will have no effect on system performance.

4.5.2 Effect of membership functions

For fuzzy control algorithms, membership functions of each input-output variable of the controller play an important role in determining the control system performance. Two essential aspects related to each membership function are the support and the position which, together with the fuzzy control rules, provide a complete control surface. It is known that the position of each membership function directly affects the position of the switching line of the FLC. So the effect of the position of each membership function will be similar to that of the rule base described in the last subsection. In addition, the location of membership functions can be determined from practical experience of controlling the specific process. But in practice the support of each membership function, thus the overlap between membership functions, is the main consideration in designing FLC systems. Therefore, investigation will only be made into the effect of the overlap of membership functions.

The overlap measurement of two membership functions is defined as follows. Suppose two adjacent membership functions μ_1 and μ_2 are as shown in Fig.4-23. Their overlap is measured by

$$\text{overlap}(\%) = 100a/b, \tag{4-7}$$

where a and b are as shown in Fig.4-23.

To explore the effect of the overlap of membership functions on system robustness, three type of membership functions, as shown in Fig.4-24, were selected for each of the fuzzy controller variables. Overlaps in each type of the membership functions were chosen as 50%, 100% and 150%.

The controller using membership functions with 100% overlap was defined as the reference controller, and other controllers were organized to allow only one of their variables to take the membership functions with 50% overlap or 150% overlap. Seven controllers in such combinations were tested to control the first order process with time delay τ = 100. The tuning point was selected at K = 35 and T = 4000. Under the STPP condition and with the

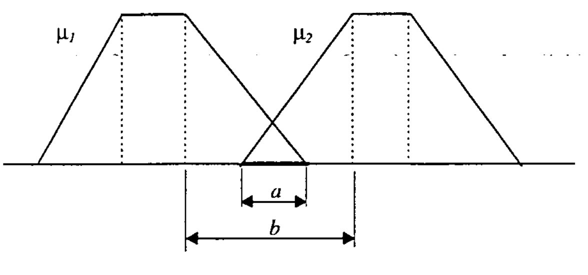


Fig.4-23 Definition of the overlap measurement of two adjacent membership functions

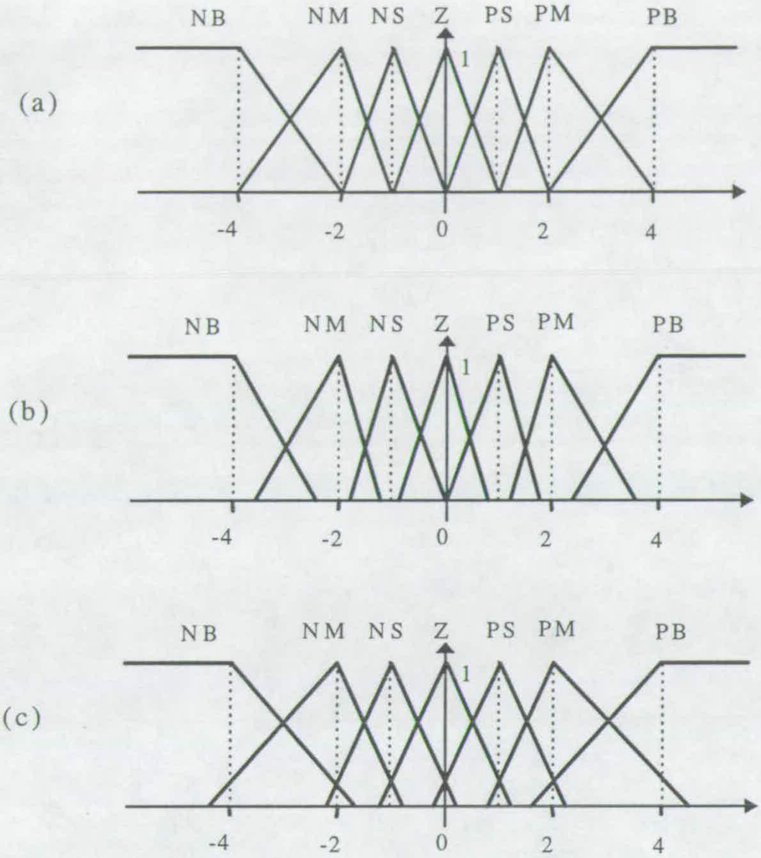
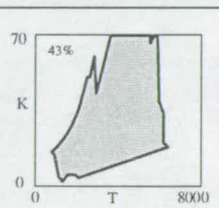
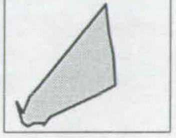
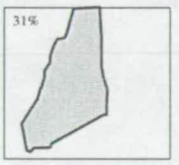
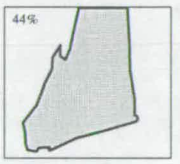
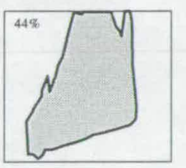
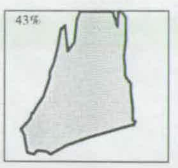
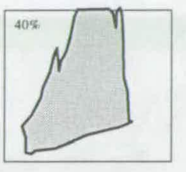


Fig.4-24 Membership functions to be used in the robustness tests
(a) 100% overlap; (b) 50% overlap; (c) 150% overlap.

Table 4-2 robust areas at different overlaps of membership functions

	Overlap 50%	Overlap 100%	Overlap 150%
μ(*e*)			
μ($\dot{e}$)			
μ(Δ*u*)			

cfg: cuws. src: m0f_mf

default rule base defined in *Section* 4.2, the robust areas of each system were mapped with the prescribed parameter space by using the simulation tool *FzySimu*. The results are shown in Table 4-2.

From the results shown in Table 4-2, it can be found that the variation in the overlap of membership functions does not create obvious effects on system robustness except for two cases: $\mu(e)$ in 150% overlap and $\mu(\dot{e})$ in 50% overlap.

Further investigation found that when the overlaps of membership functions $\mu(e)$ are increased, the controller looses the ability in handle large gain processes; when the overlaps of membership functions $\mu(\dot{e})$ are decreased, the controller looses the ability in handle large time constant processes. Analytically, if the overlaps of membership functions $\mu(e)$ are increased, the supports of the BIG and MEDIUM membership functions become wider. The controller becomes more likely to give an over-drive control output, so that the controlled process, with large gain, generates large overshoot or undershoot more easily. Similarly, if the overlaps of membership functions $\mu(\dot{e})$ are decreased, the supports of the BIG and MEDIUM membership functions become narrower. In this case, the derivative error $\dot{e}$ of a sluggish system finds it more difficult to reach the BIG or MEDIUM level, so that the controller is more likely to generate excess control output. Thus overshoot or undershoot will increase.

4.5.3 Effects of scaling factors

In the investigation of the effect of scaling factors on the robust performance of a fuzzy control system, the membership functions and rule base described in *Section* 4.2 were used to control a first order process with time delay. The scaling factors were manually tuned at the tuning point K = 35, T = 4000 and τ = 100, and a satisfactory step response was obtained with GE = 0.5, GC = 9.0, GU = 0.05, and t_s = 50. Then one of these scaling factors was varied, with the others unchanged, to test the system robustness within the prescribed process parameter space. The test results are shown in Fig.4-25

From the results, it can be found that the manually tuned scaling factors did not produce the best robustness. System robustness can be improved by selecting suitable values for the scaling factors. For instance, if GE is changed from 0.5 to 0.4 and the other controller parameters are kept unchanged, system robustness will increase from 50% to 68%. Note that these improvements on system robustness are only local optima. The global optimum can be obtained by searching all possible controller parameter settings and this will be discussed in the next section.

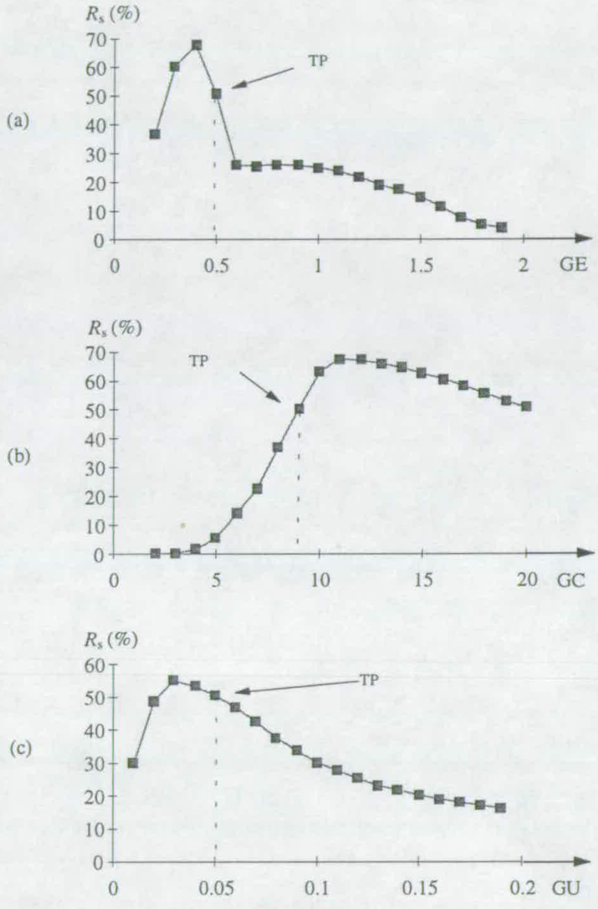


Fig.4-25 Effects of scaling factors on system robustness.

The results also indicate that the robustness of a fuzzy control system is sensitive to the scaling factors. To maintain the robustness obtained with the original settings, the variation of each scaling factor cannot be large. A careful examination found that the robust range of parameter T moves to the higher-value region when GC increases; the robust range of parameter K moves to the lower-value region when GU increases. This demonstrates that GC has a close relation with the process time constant and GU with the process gain. These relations can be used to improve system robustness by adapting GC and GU according to the changes in time constant and process gain respectively.

To explain the narrow range of scaling factor values maintaining the required robust performance, the effect of each scaling factor on system performance at the tuning point was tested. The results are shown in Fig.4-26. From the results, the following facts can be discovered which are in line with the analysis based on the switching line theory in *Chapter* 3.

1) A small GE will lead to a slow response and a big overshot because small GE leads to a wide tolerance band of error around the set-point. This problem becomes more serious if *K* becomes small or *T* becomes large. A large GE will cause limit cycling at the steady state because the performance measure becomes more sensitive around the set-point and less sensitive during rise-time, especially when *K* is large or *T* is small.

2) Decreasing GC will lead to a quick response but this increases overshoot because the controller becomes less sensitive to the error changes. Increasing GC will give a slow rising speed of the step response, but limit cycling will occur if gain *K* is large and the performance can not meet the settling time requirement if *K* is small.

3) GU functions purely as an overall gain factor of the controller which varies the magnitude of the process input. A small GU leads to slow responses so that the overshoot could be large if *T* is big, while a big GU will give a fast response but could cause limit cycling when *K* is increased.

Finally, this section can be concluded as follows. In a fuzzy control system, system robustness can be influenced by the rule base, the membership functions, and the scaling factors, because the variation in any of these controller parameters can change the position and/or the shape of the switching line which in turn determines the whole system performance. Experimental results shows that the robustness of a fuzzy control system are sensitive to the position of the switching line but less sensitive to the shape of the switching line and the overlaps of membership functions. In addition, it is indicated that the robustness of a fuzzy control system can be optimized by properly selecting the controller parameters.

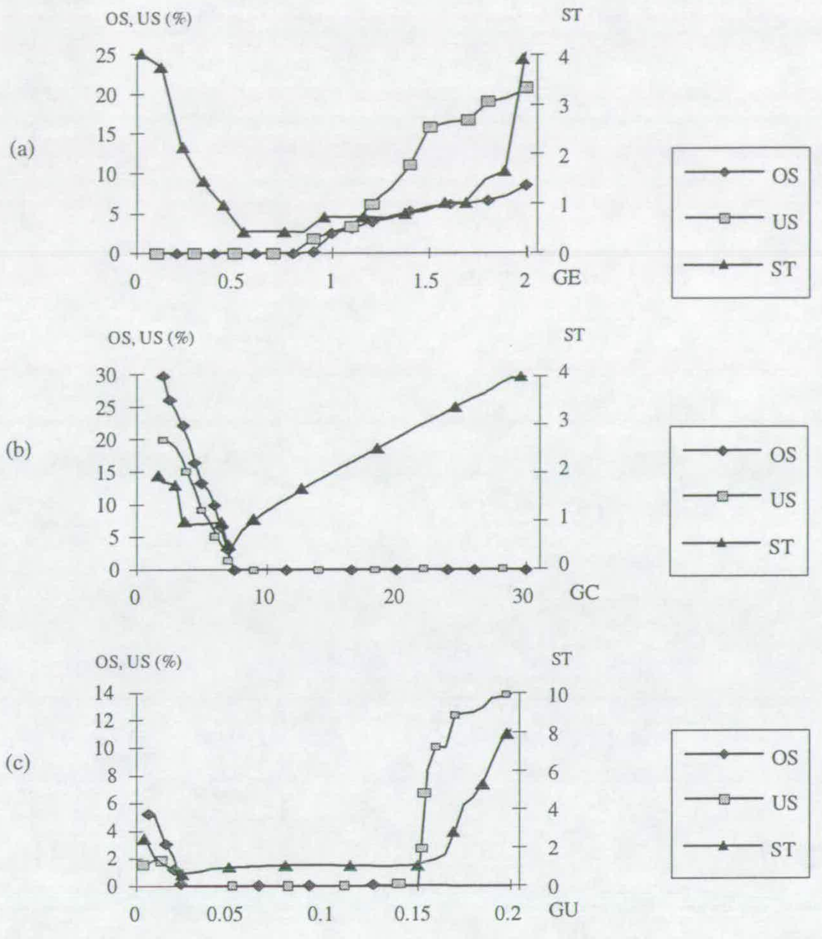


Fig.4-26 Effects of scaling factors on system performance at the tuning point

4.6 Experimental valuation of the proposed design methods

In *Chapter* 3, the switching line method was proposed for designing a robust FLC system. Based on this method, the position of the switching line can be selected and the distribution of rules on both sides of the switching line can be determined. For the higher order process control problem, two different phase advanced methods have been introduced for improving the robustness of the fuzzy control systems. For automatic design of a robust fuzzy control system, an evolutionary method based on the genetic algorithm was proposed.

In this section, experimental evaluation of the proposed methods will be performed, *i.e.* the controller will be designed by using the proposed methods, and the robustness of this controller will be investigated by allowing the process parameters to vary in certain ranges. To demonstrate the value of the proposed methods in improving system robustness, comparison with the standard FLC will be carried out with respect to system robustness.

4.6.1 Switching line method

The following process was chosen to be controlled by the fuzzy logic controller designed by using the switching line method:

$$G(s) = \frac{K\omega_n^2(s - 1/T)}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)}. \qquad (4\text{-}9)$$

This is a third order non-minimum phase process which is often used to approximate the dynamics of flexible-joint robot arms and vehicles driven by motors [1]. Parameter K, ω_n and ζ are assumed to vary in the following ranges:

K = (1.0 ~ 7.0), ω_n = (1.0 ~ 10.0) and ζ = (0.5 ~ 0.9),

and the zero ($1/T$) is fixed at 0.025 ($T = 40$). The operation point is selected as K=2.5, ω_n = 3.0, $\zeta = 0.7$.

According to the switching line design method described in *Chapter* 3, the phase plane trajectory of the required system performance and the dynamic region of the controlled process are required to select the FLC parameters. Because the process (4-9) is linear, its dynamic region on the phase plane can be measured by detecting the two extreme phase plane trajectories, *i.e.* the slowest response and the fastest response. From the linear system theory, two extreme responses correspond to the following two parameter settings:

(i) $K = 1.0$, $\omega_n = 1.0$, $\zeta = 0.9$; (ii) $K = 7.0$, $\omega_n = 10.0$, $\zeta = 0.5$.

Fig.4-27 shows two measured phase plane trajectories at these two parameter points. Because of the integrating characteristic of the controlled process, the unit impulse was used to test the phase plane trajectories of the process. In addition, the error in the steady state was normalized to 1.0.

Based on the switching line method, the membership function of the input variables, e and $\dot{e}$, can be determined as illustrated in Fig.4-28. First, the required phase plane trajectory was designed as the line c shown in Fig.4-28(a), which is based on the consideration that system responses should be fast enough when system error is BIG so as to meet the settling time requirement and slows enough when system error is SMALL so as to avoid high overshoot or undershoot. Secondly, a simple symmetrical rule base was chosen because the controlled process is linear. Thirdly, based on the rule base and the required phase plane trajectory, the membership functions of the input variables were determined as shown in Fig.4-28(a), where the membership functions of each variable were assumed symmetrical to the origin of this variable.

The output membership functions were determined as follows. From (3-37) to (3-40) in *Chapter* 3, the following two equations can be obtained

$$cog(S_{\Delta u}) \leq \frac{|\Delta \dot{e}|_{max} \cdot T_{min}}{K_{max}} \tag{4-9}$$

$$cog(B_{\Delta u}) \geq \frac{\Delta e \cdot T_{max}}{t_{reach} \cdot K_{min}} \tag{4-10}$$

where $|\Delta \dot{e}|_{max}$ is the maximum error change deviation allowed for the phase plane trajectory to slide on the switching line in the sliding period. Δe is error change covered by the reaching period determined by the required switching line, t_{reach} is the time spent in the reaching period, $S_{\Delta u}$ and $B_{\Delta u}$ are fuzzy sets SMALL and BIG of the controller output change, Δu.. Note that a symmetrical output membership function is assumed.

Equation (4-9) is related to the sliding condition. For the control of linear processes, $|\Delta \dot{e}|_{max}$ corresponds to the high gain and small time constant situation, like the phase plane trajectory *b* in Fig. 4-28(a), where small change in the control input can lead to a fast and

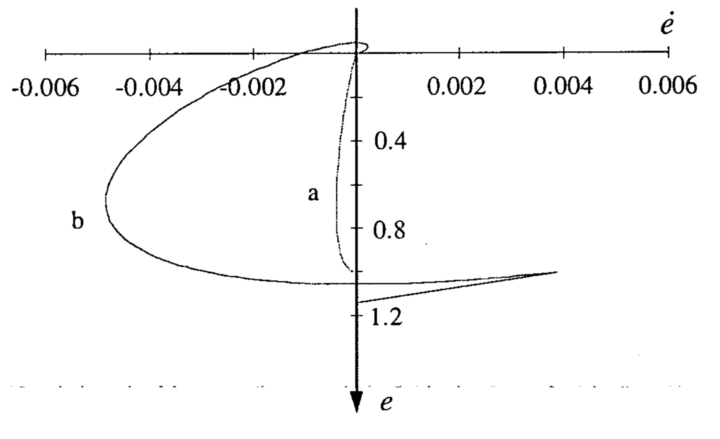
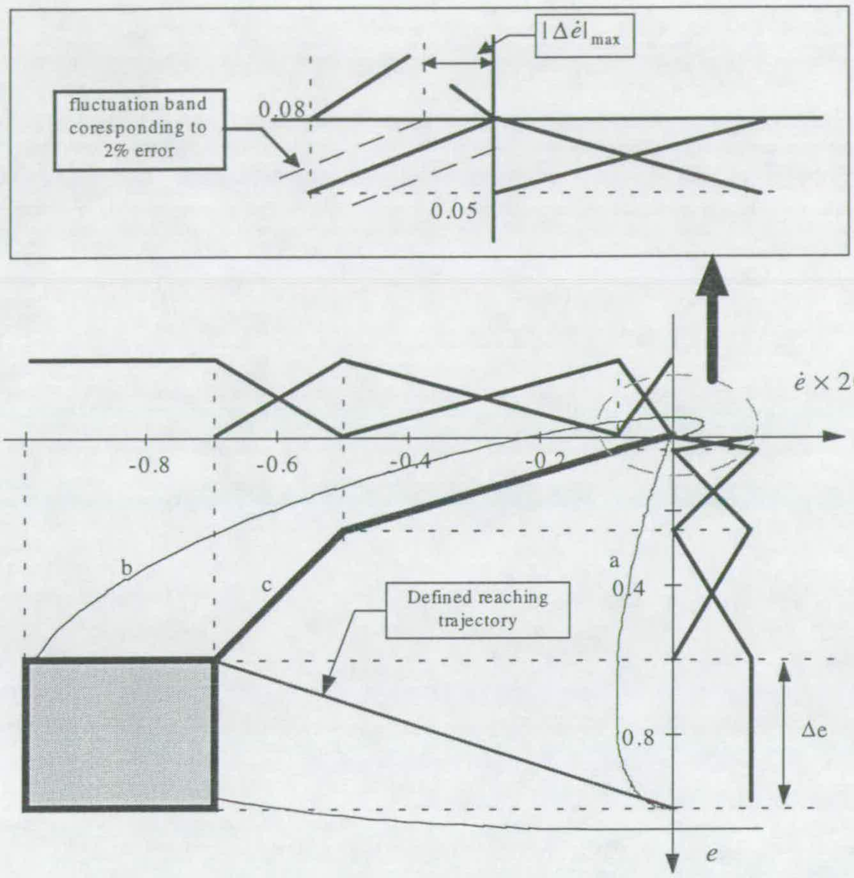
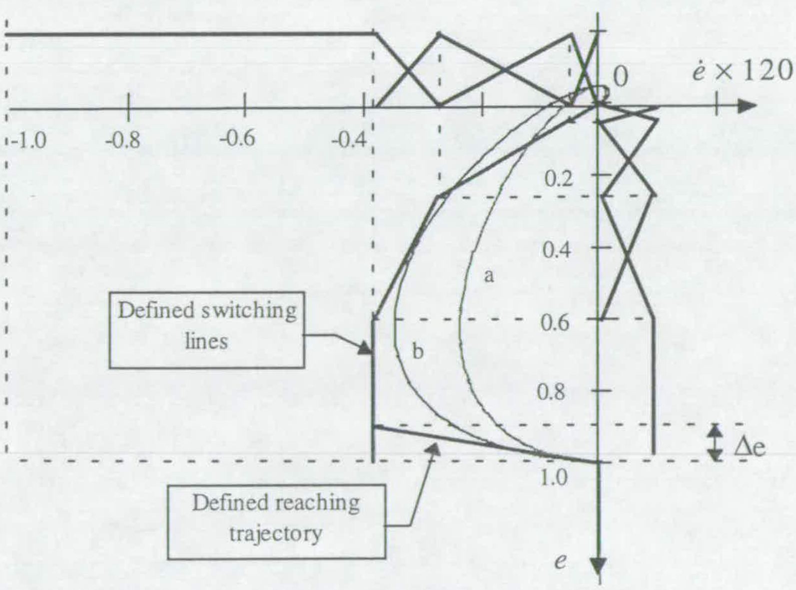


Fig.4-27 Open-loop impulse response of process (4-9) at
(a) $K= 1.0$, $\omega_n = 1.0$, $\zeta = 0.9$ and
(b) $K=7.0$, $\omega_n =10.0$, $\zeta = 0.7$.



(a)



(b)

Fig.4-28 Design of membership functions of the input variables from the open-loop performance of the controlled process: (a) based on entire parameter space; (b) based on the adjusted parameter space.

significant change in the process output; thus it has a great tendency to deviate from the designed switching line trajectory. From Fig. 4-28(a), it can be seen that if a 2% fluctuation of error is allowed, then $|\Delta \dot{e}_{\max}| = 1.5 \times 10^{-4}$.

Equation (4-10) is related to the reaching condition. For linear control, main consideration should be made on the slow response situation, like the phase plane trajectory *a* in Fig. 4-28(a), where large control action is needed to bring the system state to the required switching line trajectory. Parameters Δe and t_{reach} correspond to the required switching line performance. From Fig. 4-28(a), it can be seen that $(\frac{\Delta e}{t_{reach}})$ is the average error change in the reaching period and can be approximated by the error change at the middle point of the reaching trajectory, *i.e.* $\frac{\Delta e}{t_{reach}} \approx 3.5/200 = 0.0175$.

Considering $(\zeta\omega_n)^{-1}$ as the equivalent time constant, then from the parameter variation ranges, $T_{min} = 0.11$, $T_{max} = 2$, $K_{min} = 1.0$, and $K_{max} = 7.0$. So, from (4-10) and (4-11), following relations can be obtained:

$$\text{cog}(B_{\Delta u}) \geq 0.0175,$$
$$\text{cog}(S_{\Delta u}) \leq 2.36 \times 10^{-6}.$$

Obviously, $\text{cog}(S_{\Delta u})$ is so small that it will take long time for the FLC to adjust its output to adapt the error change, and consequently a big overshoot or undershoot will take place. The reason is that the required parameter space is too big for the FLC to maintain an acceptable performance. An adjustment of either the performance requirements or the size of the parameter space is needed. Here the size of the required robust space is adjusted to

$$K = (2.5 \sim 4.0),\ \omega_n = (4.0 \sim 6.0) \text{ and } \zeta = 0.7.$$

Within this reduced parameter space, the variation region of the phase plane trajectories of the controlled process (4-9) were found to be in the area between trajectory *a* and trajectory *b* as shown in Fig. 4-28(b). Similarly, the following parameter values can be obtained:

$$|\Delta \dot{e}_{\max}| = 1.5 \times 10^{-4},\ \frac{\Delta e}{t_{reach}} = 0.18/120 = 0.0015,$$

$$T_{min} = 0.238,\ T_{max} = 0.357,\ K_{min} = 2.5, \text{ and } K_{max} = 4.0.$$

Now the reaching condition and sliding condition become;

$\cos(B_{\Delta u}) \geq 2.1 \times 10^{-4}$;

$\cos(S_{\Delta u}) \leq 8.9 \times 10^{-6}$.

Intuitively, the above conditions are applicable. So the control change membership functions can be selected as follows:

$\text{cog}(B_{\Delta u}) = 2.5 \times 10^{-4}$;

$\text{cog}(S_{\Delta u}) = 5.0 \times 10^{-6}$,

and $\text{cog}(M_{\Delta u})$ is chosen as 5.0×10^{-5}.

The robustness of the FLC system designed above was tested in the previously defined parameter space; and the experimental results is shown in Fig.4-29. Note that the system was tuned at the operation point, and the scaling factors were set as

GE=1.0, GC=120, GU=0.025, t_s = 10.

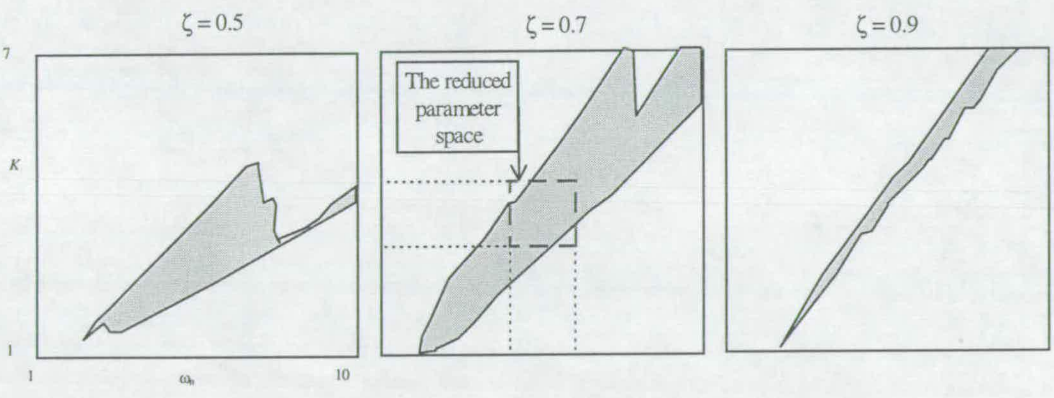


Fig. 4-29 The tested robust space of the FLC system designed by the switching line method (Ref. sl_desig.248)

In Fig.4-29, it can be seen that system performance meets the performance criteria within the main part of the reduced parameter space, and the robustness R_s in the whole parameter space reaches 14.8%. If the damping ratio ζ = 0.7 is only considered, the robustness index becomes 24.9%. It should be noted that, if the FLC described in *Section* 3.2 in *Chapter* 3 is used to control this process, the system robustness will not exceed 8% under the STPP condition.

From this experimental test of the switching line design method, it is demonstrated that the this method is effective for designing a robust fuzzy control system, and the design procedure is quite simple and easy to use. However, this method is only suitable for the SISO systems, and the required information about the open-loop responses of the controlled process may not be available. Also, like most design methods for control systems, tuning is very important to achieve the designed performance. Further work is needed to extend this method to other types of control processes and reduce the dependence on knowledge about the controlled plants.

4.6.2 Phase advanced methods (PAM)

In *Section* 3.6.2 of *Chapter* 3, two phase-advanced fuzzy control methods were developed to handle the problems experienced with the standard fuzzy logic control of high-order processes. This section represents the use of these two methods to control the high order processes which are found to be more difficult for the standard fuzzy logic control method to achieve high robustness compared with the case of first order processes. Comparison with the standard fuzzy logic control method will be made to demonstrate the advantage of the proposed phase-advanced methods. The experimental procedure defined in *Section* 4.2.3, the membership functions and the rule base defined in *Section* 3.2 were used.

A. Non-integrating high order process — PAM-I

The controlled process was chosen to be the time delayed non-integrating second order process:

$$G(s) = \frac{K\, e^{-\tau s}}{(1+T_1 s)(1+T_2 s)}. \tag{4-11}$$

The robustness of the closed-loop system was investigated by allowing the parameters of this process to vary in the following ranges;

$K = (0.5 \sim 9.5)$, $T_1 = (50 \sim 500)$, $T_2 = (0 \sim 225)$, $\tau = 20, 50$.

The robustness of both the FLC system and the PAM-I system have been investigated within the process parameter space $K \times T_1 \times T_2$. Scaling factors of both controllers were separately tuned at $K = 4.5$, $T_1 = 250$, $T_2 = 100$ and $\tau = 20$, and the STPP requirement was obtained with the following parameter settings:

FLC: GE = 1.0; GC = 18.0; GU = 0.07, $t_s = 7$.

PAM-I: GE = 0.5; GC = 12.0; GU = 0.1; $K_p = 2.0$, $t_s = 7$.

Each system's step response at the tuning point is closely similar to that shown in Fig. 4-30. Note that if the response of the FLC system is made faster than shown, oscillations or a large overshoot will result.

The robust space of each system is shown in Fig.4-31 and Fig.4-32, where the upper and lower surfaces of the robust space are presented. The robust space of the PAM-I system and the FLC system take 680 and 465 test points in 1000 tests respectively. A similar result can be found if $\tau = 50$ where in this case the robust space of the PAM-I system and the FLC system take 643 and 295 test points respectively in 1000 tests.

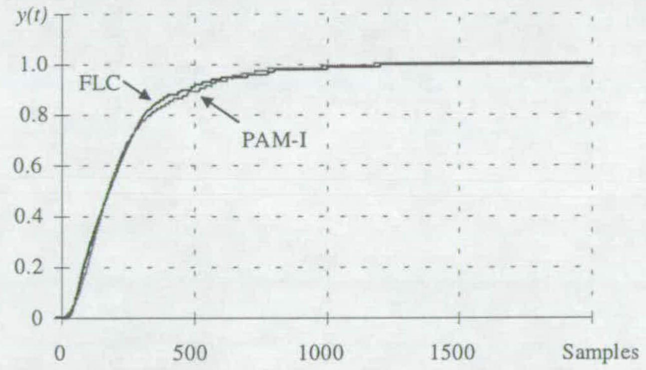


Fig. 4-30 Step responses of FLC and PAM-I systems at the tuning point

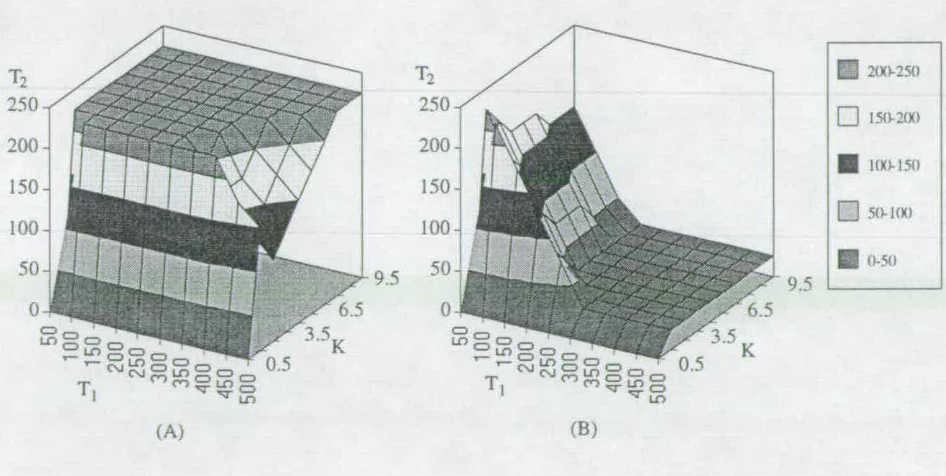


Fig. 4-31 Robust space of the PAM-I system (A) upper surface; (B) lower surface.

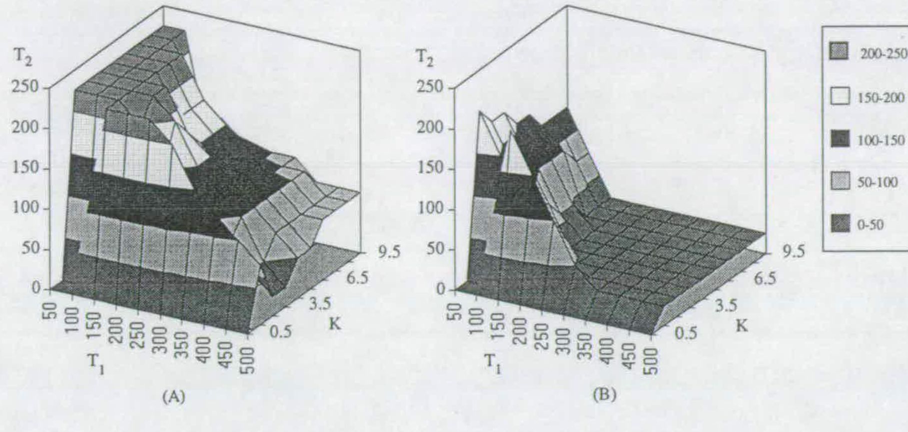


Fig. 4-32 Robust space of the FLC system (A) upper surface; (B) lower surface.

B. Integrating high order processes — PAM-II

The controlled process is chosen to be the integrating second order process with time delay:

$$G(s) = \frac{K\omega_n^2 e^{-\tau s}}{s(s^2 + 2\zeta\omega_n + \omega_n^2)}. \tag{4-12}$$

This process cannot be control by the PAM-I method because an accumulator after the fuzzy inference engine is not required. Also due to the time delay term in the process, it will be very difficult to control the process by the standard FLC. This high order process, however, is suitable for using the second phase-advanced control algorithm PAM-II.

The robustness of the closed-loop system was investigated by allowing the parameters of this process to vary in the ranges shown below;

$K = (1.0 \sim 7.0)$, $\omega_n = (1.0 \sim 10.0)$, $\zeta = (0.5 \sim 0.9)$, $\tau = 100, 200$.

The robustness of both the FLC system and the PAM-II system have been investigated within the process parameter space $K \times \omega_n \times \zeta$. Scaling factors of both controllers were separately tuned at $K = 3.0$, $\omega_n = 5.0$, $\zeta = 0.7$ and $\tau = 200$, and the STPP requirement was obtained with the following parameter settings:

FLC: GE = 1.0; GC = 50.0; GU = 0.009.

PAM-II: GE = 0.9; GC = 70.0; GU = 0.011; $\beta = 10.0$.

The sample rate is $t_s = 20$. The step response of each system at the tuning point is shown in Fig. 4-33 from which it can be seen that they are closely similar.

The robust space of each system is shown in Fig.4-34 and Fig.4-35, where the upper and lower surfaces of the robust space are presented. The robustness of the PAM-II system and the FLC system are 20.6% and 13.1% in 2700 tests respectively. When the time delay was set as $\tau = 100$ without re-tuning the system, the robust index of the PAM-II system and the FLC system become 9.8% and 4.0%.

Comparing Fig. 4-31 with Fig.4-32 and Fig.4-34 with Fig.4-35, it can be clearly seen that both phase-advanced methods can achieve higher robustness in controlling the high order processes than the standard fuzzy logic control algorithm. This improvement becomes more significant when the dynamic impact of the controlled process is increased. This is reflected in the system performance when T_2 in process (4-11) is increased or ζ in the process (4-12) is decreased, where the robust space of the phase-advanced FLC system is much larger than that of the standard FLC system. When the dynamic impact of the controlled process

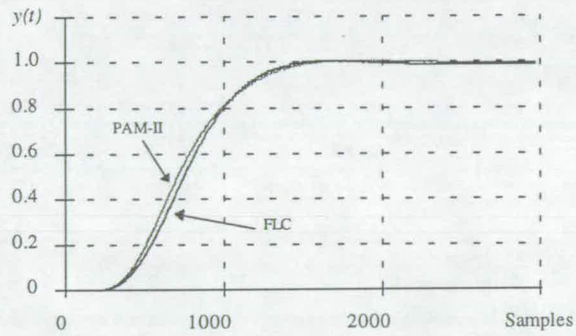


Fig. 4-33 Step responses of FLC and PAM-II systems at the tuning point

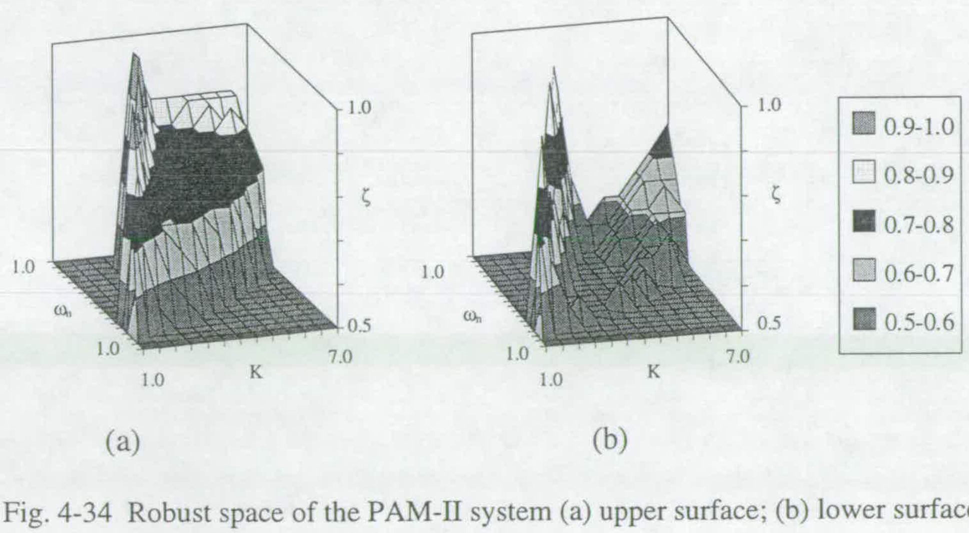


Fig. 4-34 Robust space of the PAM-II system (a) upper surface; (b) lower surface.

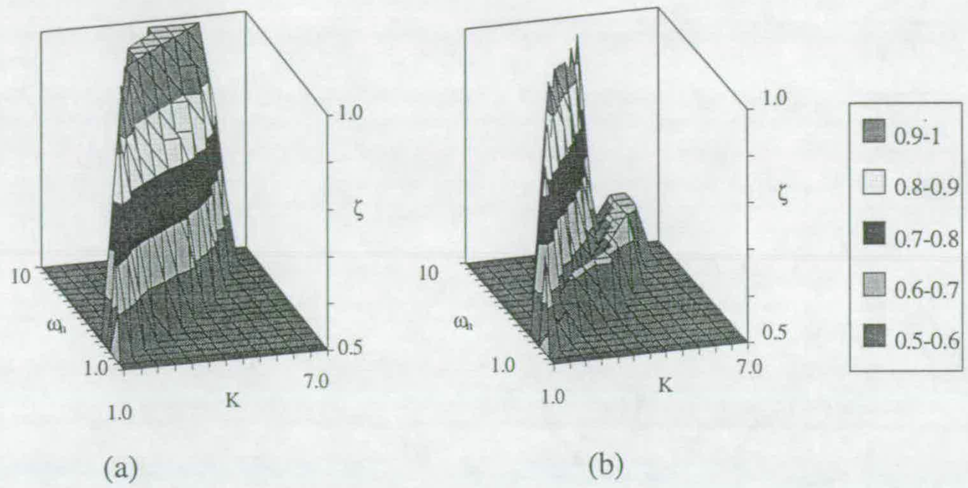


Fig. 4-35 Robust space of the FLC system (a) upper surface; (b) lower surface.

is decreased, the advantage of the proposed phase-advanced methods becomes negligible, which can be reflected when T_2 in process (4-11) is small or when ζ in process (4-12) is large.

However, tuning the phase-advanced fuzzy control system becomes more complicated than the standard fuzzy control system, because the additional parameter produces significant effect on system performance, especially in the high order case. From practical experience in tuning these systems, it is found that they can be tuned at first by setting the proportional gain K_p in PAM-I or the derivative gain β in PAM-II to zero, and then by tuning K_p or β to improve the system performance. Generally, when phase-advanced control is applied, the scaling factors should be adjusted slightly to increase the system response speed.

4.6.3 Genetic algorithms

Genetic algorithms have been successfully used as a global search method for optimal conditions in many field. There are also applications in designing fuzzy logic control systems for obtaining optimized controller parameters. Since the design of a robust fuzzy control system in this research aims to obtain a maximum parameter space where the system performance meets the prescribed performance specification, the design task can be transferred to a searching problem in the concerned process parameter space which can be performed by the genetic algorithm.

However, successful application of the genetic algorithm needs a simulation model of the controlled process which, in fact, is not required by the fuzzy control method. Nevertheless, in some cases where the process model can be approximated, then the genetic algorithm can be used as a systematic design method for this fuzzy control system. An example of this type of process is the carpole system [121] which is often used as the benchmark of control methods.

This section presents the results of an investigation of the robust performance of the cart-pole system controlled by a FLC optimised by applying the genetic algorithm. The design method described in *Chapter* 3 was used and the simulation experiments were carried out to allow the parameters of the fuzzy controller to be optimized. The robust performance of resulting system was compared with that of the manually tuned fuzzy control system.

The robustness of the cart-pole system defines the ability of the controller to keep the pole upright and position the cart in the required zone for a required length of time T_{up}, when the cart-pole parameters, like the mass of the cart and the length of the pole, vary in certain ranges.

The cart-pole process can be described by the following equations:

$$\left.\begin{aligned} m_c\ddot{p} + m_p\ddot{p} + m_pl\ddot{\theta}\cos\theta - m_pl\dot{\theta}^2\sin\theta &= Ku \\ J\ddot{\theta} + m_pl(\ddot{p}\cos\theta + l\ddot{\theta}) - m_pgl\sin\theta &= 0 \end{aligned}\right\}, \qquad (4\text{-}13)$$

where m_c is the mass of the cart, p is the horizontal displacement of the cart, m_p is the mass of the pole, l is the length of the pole, θ is the pole's angle to the vertical, u is the control force exerted on the cart, $J = \frac{1}{3}m_pl$ is the moment inertia of the pole, and g = 9.8 m/s^2 is the acceleration due to gravity, K is the gain of the control input. The parameters of the cart-pole process at the tuning point are set as follows:

$m_c = 1.0\ kg$; $l = 1.0\ m$; $m_p = 0.1\ kg$; $K = 4.0$,

and the initial value of θ and p are assumed as $\theta(0)$ = 0.35 *rad*, $p(0)$ = 0 (origin of the horizontal coordinate). The parameter m_c and l are assumed to vary in the following ranges:

m_c = 0.5 to 5.0 kg; l = 0.3 to 2.5 m.

From the mathematical model of the cart-pole system, it can be seen that, in addition to the nonlinearity of the process, the control task with the cart-pole system is to control two output variables, *i.e.* θ and y, by operating only one control input u, the horizontal force applied to the cart. Obviously, there is correlation between the two outputs, which, in principle, requires de-coupling techniques and specific fuzzy control rules, if the FLC is used. Moreover, if the number of the controller inputs is constrained to two, so as to avoid the increase of rules and computing complexity, a method has to be developed to include all useful system dynamic information in the input signals.

The method used in this research can be stated as follows. Intuitively, the pole balance is more important than centering the cart. The pole leaning to the left can be balanced by a control force to the left, and *vice versa*. After the pole is balanced, the cart can be centered by a "kicking" in the opposite direction of centering, for instance, if the cart is on the left of the required position, a control force to the right has to be applied. A kicking control in one direction will make the pole lean in another direction, and the next balance control will bring the cart to the required position.

Based on the above intuitive control action, the system error signal e can be calculated as follows:

$$e = \begin{cases} \theta & \textit{when} \text{ balancing the pole} \\ \theta + w_p y & \textit{when} \text{ centering the cart} \end{cases} \tag{4-14}$$

where w_p is the weight of position y in the error signal, and the centering task should be performed only when the pole's leaning angle θ is relatively small and the cart is moving away from the required position.

With the system error defined as above and the change rate of this error as the controller input signals, the standard FLC can be applied directly to control the cart-pole system. Therefore, the fuzzy controller designed in *Chapter* 3 will be used as the initial prototype with the membership functions and the rule base as shown in Fig.3-4. When the controller is optimized by the genetic algorithm, the objective function for evaluating the performance of the control system will be defined as the time period T_{up} during which the pole is kept upright and the cart's position is within $\pm 2m$ from the center $y = 0$.

Based on the algorithms developed in *Chapter* 3, the *GaOpt* function in the *FzySimu* simulation package was used to carry out the optimization. The simulation was set as follows:

Gene number: 35;

Population Size: 100;

Crossover Rate: 0.6;

Mutation Rate: 0.001;

Total generation number: 20;

$T_{up} > 10{,}000$ sample periods.

The 35 genes include GE, GC, GU, t_s, w_p, and 30 variables of the whole membership functions which are constrained as symmetrical to their origins because the dynamic performance of the cart-pole system will not change with the pole angle or the cart positions. The initial population was randomly selected by the program.

After 20 generations of evolution in these 100 populations, the best robust performance was found in the system with the membership functions as shown in Fig.4-36 and the following parameters:

GE = 8.0, GC = 547.0, GU = 36.3, $t_s = 2$, $w_p = 26.2$.

System performance at the operating point is shown in Fig.4-37 and the robust performance in the cart-pole parameter space $m_c \times l$ is shown in Fig. 4-38.

As a comparison, the initial prototype of the FLC was manually tuned by adjusting the scaling factors, the sample rate and the position weight for the best robust performance with the concerned cart-pole parameter space. The best result is obtained when the tuned controller parameters were set as follows:

GE = 4.0, GC = 200.0, GU = 50.0, t = 5, $w_p = 2$.

Fig. 4-39 and Fig. 4-40 present the operating point performance and the robust space of this manually tuned system.

From the operating point performance of both systems, it is found that the manually tuned system produced a smaller IES value than the GA tuned system; but the robustness space of

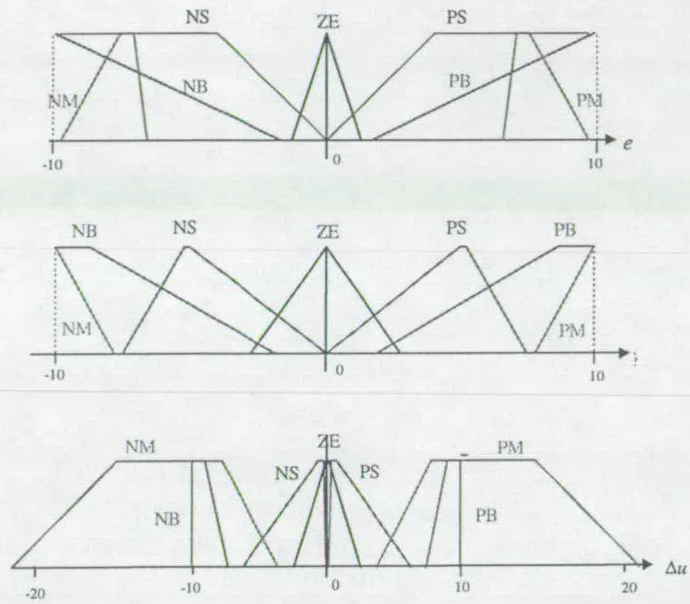


Fig. 4-36 Membership functions generated by the GA optimisation

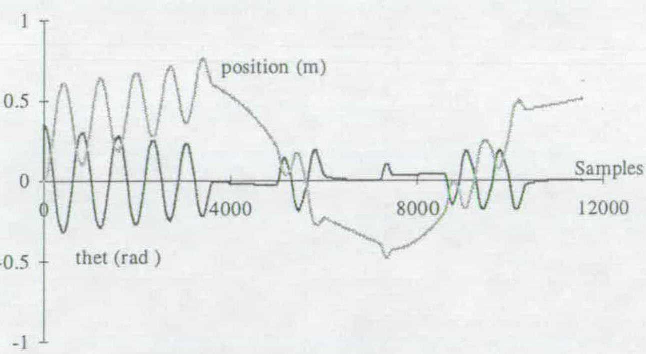


Fig.4-37 Operating point performance of the GA optimised cart-pole fuzzy control systems

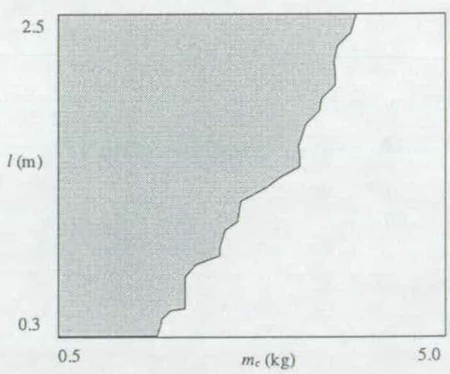


Fig.4-38 Robust space of the GA optimised cart-pole fuzzy control systems

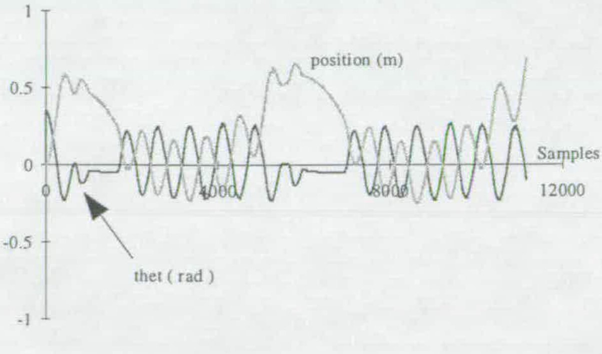


Fig.4-39 Operating point performance of the manually tuned cart-pole fuzzy control system

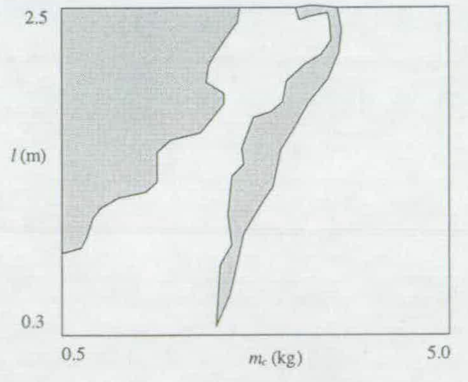


Fig.4-40 Robust space of the manually tuned cart-pole fuzzy control system

the former system only took 30% of the tested parameter space while the later system can meet the performance specification in 55.8% of the parameter space. These results show (i) that the ability of manual tuning is limited to the adjustment of few parameters, while the genetic algorithm can perform multiple parameter tuning for a global optimization, and (ii) that manual tuning usually pays more attention to the performance of the operating point than other working points and this often leads to a local optimum (GA optimization can overcome this drawback). Also it is very time consuming to tune nonlinear systems (*e.g.* the fuzzy logic controlled cart-pole system); it requires system knowledge and practical experience.

From Fig.4-36 it can be found that the membership functions generated by the GA optimization are quite strange. For instance, the membership function of the MEDIUM control change takes a much shorter support than expected. But by careful examination of the system responses, it can be found that the control output is often strong and changes sharply to keep the pole balanced, thus the SMALL control output near the switch line should be relatively large and the MEDIUM control rules may not often be used.

It should be noted that a suitable representation of the optimized parameters is very important for the fast search of the global optimum in the controller parameters. For instance, the symmetrical constraint on the fuzzy membership functions leads to a sensible solution of the optimization problem and saves computing time.

4.7 Conclusion

In this chapter, an experimental investigation of the robustness of FLC systems has been presented, and the feasibility of the proposed methods for analyzing and designing the robustness of FLC systems has been demonstrated.

At first, the simulation system was designed, performance specifications were defined, and the experimental procedure was chosen for collecting the experimental results. Then, a simulation tool was developed to perform the system simulation, robust space mapping, auto-tuning, optimization and the real time control tasks.

With this software tool, the robustness of the fuzzy logic control system was investigated as a controller for the typical first order process and second order process, and the qualitative analysis in *Chapter* 3 of the robustness of fuzzy control systems has been confirmed. It is demonstrated that (1) the robustness of the FLC systems stems from the constraints introduced by the switching line; (2) the robust space of the first order FLC system is a sector in the *K-T* plane, and its area is significantly reduced when time delay is introduced; (3) the robust space of the second order FLC system is an approximate sector shape in K — ζ plane and a curved sector shape in K — ω_n plane; (4) a properly designed and tuned fuzzy logic control algorithm is more robust than the PID control algorithm in controlling the first order and second order processes; (5) the performance of the fuzzy logic control system at the tuning point affects the position and the size of the system's robust space in the process parameter space; (6) the standard FLC designed in *Chapter* 3 exhibits less robustness in controlling high order processes than in controlling low order processes; knowledge about high order dynamics of a controlled process should be included in the fuzzy rule base for controlling high order dynamic systems.

It has been shown that in a fuzzy control system, system robustness can be influenced by the rule base, the membership functions, and the scaling factors, because the variation in any of these controller parameters can change the position and/or the shape of the switching line which in turn determines the whole system performance. Experimental results showed that the robustness of a fuzzy control system is sensitive to the position of the switching line but less sensitive to the shape of the switching line and the overlaps of membership functions.

Experimental evaluation of the proposed design methods for the FLC system has been performed in the last part of this Chapter. The FLC was successfully designed by using the switching line method and the robustness of this controller in controlling a complicated high order process has been found to provide a significant improvement in system robustness. It is also demonstrated that both phase-advanced methods can achieve higher robustness in controlling the high order processes than the standard fuzzy logic control algorithm, and this improvement becomes more significant when the high order characteristic of the controlled process is increased. As a systematic design method, a genetic algorithm has been experimentally investigated and is found to be a very powerful method for coping with the multi-variable optimization problems associated with the design of fuzzy logic controlled systems.

Among all the experimental investigations, however, it is found that (i) knowledge about the controlled process is very desirable for the design of a robust fuzzy control system, especially information of the dynamic changes in the process; (ii) as indicated before, tuning the fuzzy control system, although very difficult, is important for the best system performance; (iii) the selection of suitable fuzzy control rules is essential for robust control purpose, especially in the case of non-linear process control. Further work is needed to investigate systematic design methods for fuzzy controllers in the case of non-linear processes and high process uncertainties. The tuning problem will be investigated in the next chapter.

5

Auto-Tuning of Fuzzy Controller

5.1 Introduction

System tuning is an unavoidable step in any implementation of the automatic control of a process. Controller parameters will be tuned to achieve the designed system performance at the operating point, whatever control technique is used. A fuzzy logic controller is a non-linear and multi-variable controller, and each of its parameters can affect the performance of the fuzzy control system. Thus manual tuning of these parameters usually becomes a critical operation which can provide unsatisfactory system performance. Because of the nonlinear and the multi-variable features of the FLC, a systematic methodology for parameter tuning of FLCs has not been developed.

Many researchers have addressed their efforts to finding effective tuning methods during the last few years. Several analytical methods, such as those reported by [115-118] are based on of the relationship between the simplest FLC and the conventional PI controller. They provide some theoretical guidelines for tuning fuzzy control systems. Also, many tuning algorithms have been studied, such as the adaptive self-tuning method [114], the evolution method [60, 62], the gradient descent method [58], and the correlation function method [48]. In these studies, a mathematical model of the controlled process is assumed to be available, and parameter optimization is automatically performed by the simulation method. With these methods, the performance of the resulting fuzzy control system can be significantly improved. However, since a considerable computing effort is required to implement these methods and the searching algorithms are random to some extent, they are not suitable for *on-line* tuning of a fuzzy control system.

In the case of on-line tuning, the controller is tuned while the controller is performing the control task. The available information is the measurable system states, such as the system error, and error change. The system performance should be kept in the safe region, *e.g.* less than 20% overshoot. The process under control is often not allowed to turn on and off frequently. From these requirements, the tuned controller parameters should be kept as few as possible and the tuning algorithm should be effective so as to minimise the tuning time.

Among the various parameters which affect the performance of fuzzy systems, tuning the fuzzy rules and the membership functions of the input-output fuzzy variables of the controller will be found to be most effective to improve the performance of a fuzzy control system. But it is very complicated, thus very time consuming, to perform on-line tuning because of the multi-variable nature of the rule base and the membership functions. However, the results of fuzzification and defuzzification in the FLC can be changed by applying a different scaling of the controller variables, *i.e.* the system performance can be changed by altering the input-output scaling. Although the input-output scaling in the FLC cannot affect the rule's distribution and the overlap of membership functions, it provides a general modification of the control surface of the FLC. Compared with the tuning of membership functions or the tuning of a fuzzy rule base, it is clear that tuning the input-output scaling, *i.e.* three scaling factors, is the simplest and most effective way to tune a fuzzy control system. In addition, by tuning a fuzzy control system in this way, it is possible to use knowledge about the relationship between the scaling factors and the dynamic performances of a family of fuzzy control systems. By organizing this tuning knowledge into a fuzzy rule base, it is possible to perform the auto-tuning of that family of fuzzy control systems by using a fuzzy logic algorithm.

The results of a study of an auto-tuning method based on a fuzzy logic algorithm will be presented in this *Chapter*. The aims of the study were to automatically tune the scaling factors of a family of fuzzy control systems based on practical tuning experiences, and to obtain a relatively satisfactory system performance when the fuzzy system is initially started. It will be assumed that the membership functions and the rule base are properly designed before auto-tuning commences. This knowledge-based tuning strategy should be independent of the process under control, provided that the process belongs to the prescribed family. In addition, the on-line feature of the tuning method will be identical to other tuning methods.

It should be noted that the performance of the fuzzy control system at the tuning point will be addressed in this chapter. System robustness will not be the direct objective of the auto-tuning technique, though examinations have been made of the robustness of the auto-tuned systems. This is based on arguments established from previous chapters: the robustness of a fuzzy logic control system stems from the switching line characteristic of the controller; the position and the shape of the switching line is mainly determined by the rule base and the input-output membership functions; the effect of the tuning activity on the robustness of a fuzzy control system results from the selection of a tuning point and the performance quality at that tuning point. Normally, for on-line tuning, the tuning point is naturally fixed and is difficult to alter. As for the quality of the tuning point performance, it is the objective of the auto-tuning task to obtain the switching line performance. Therefore, this *Chapter* will focus on the tuning methodology for a required system performance.

This *Chapter* is organized as follows. The fuzzy tuning algorithm and the procedure are presented in *Section* 5.2. In *Section* 5.3, the effectiveness of this tuning algorithm is assessed by applying it to four fuzzy control systems where the controlled processes are a first order process, a time delay process. a second order process and an artificial nonlinear process respectively. Note that attention will be restricted to the SISO fuzzy control system with step input testing. The *Chapter* is concluded in section 4.

5.2 Fuzzy tuning algorithm

In most technical applications, fuzzy controllers receive crisp inputs and have to give a crisp control output. Fig.5-1 shows the general structure of fuzzy control systems. In this typical case, the operation of a fuzzy controller requires fuzzification of the inputs and defuzzification of the control outputs. To do this, each crisp variable is attached to all fuzzy sets defined in the universe of discourse by means of membership functions. For simplicity, the membership functions are defined within a normalized interval in most cases. Therefore, the crisp variables have to be scaled (normalized) by scaling factors, so that they fit into the normalized universe of discourse.

When the controlled process parameters vary with time or have high uncertainty, or robust control is required, it is important that the input-output signals are properly scaled so as to obtain the best system performance. The importance of a suitable choice of scaling factors is clearly shown by the fact that poorly chosen scaling factors will result in the shifting of the operating area in parameter space to the boundaries of the normalized universe of discourse. Procyk [31], Palm [48] and Wang [65] have shown the effects of scaling factors on the dynamic performance of fuzzy control systems. Note that optimal scaling not only depends on the variable's properties but also on the shape and position of the membership functions and the dynamics of the plant to be controlled

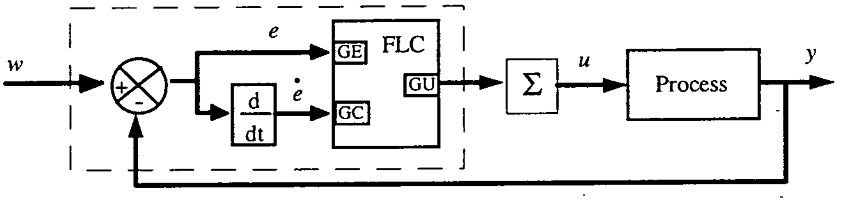


Fig.5-1 General structure of fuzzy control systems.

As discussed in *Chapter* 3, increasing the scaling factor is equivalent to compressing the corresponding membership functions towards the origin of the related axis. Proportionally increasing the support of each fuzzy subset related to a fuzzy variable at both sides of the origin is equivalent to decreasing the scaling factor of that fuzzy variable. If the membership functions are fixed, increasing the scaling factor of a variable is equivalent to amplifying this variable. Thus, it is obvious that the bigger the scaling factor of a variable, the more sensitive will the controller be to the small values of this variable, and the less sensitive to big values of this variable. Based on the above basic concepts and switching line theory, the input scaling factors can be used to modify the sensitivity of the fuzzy controller to the input signals, and the output scaling factor can be used to modify the loop gain.

For any process dynamics, there must be a suitable setting of scaling factors with which the fuzzy control system will give an optimum performance. Different process dynamics may require different scaling factors for this optimum performance. For a family of process dynamics, *e.g.* the first order dynamics, there must be some rules relating the different process dynamics to the suitable scaling factor settings. For example, the output scaling factor GU can be tuned to adapt the variations in the process gain K to obtain the same system performance, provided that the product $GU \cdot K$ is kept constant.

Because of the existence of these relational rules between the process dynamics and the suitable scaling factors, although some of them may not be known, it is possible to realise on-line auto-tuning of the fuzzy control systems by utilising these relational rules. The essential part of the auto-tuning method presented in this *Chapter* will be a knowledge base, called the fuzzy tuning rule base, which contains some of these relational rules obtained from practical tuning experience. The main task of the auto-tuning process is to detect the present system performance and to work out suitable scaling factors by fuzzy inference from the knowledge base. The entire tuning process can be illustrated as shown in Fig. 5-2. It is, in fact, a kind of fuzzy control system with the system dynamic performance as its input and the scaling factors as its controlled object.

In the following, attention will be restricted to the standard fuzzy control system controlling the first and second order processes. The design of the fuzzy auto-tuning engine concentrates on input-output variables, membership functions and the fuzzy tuning rules. Finally the tuning procedure will be presented.

5.2.1 Variables of the fuzzy auto-tuning engine (FATE)

From the previous discussion, it is known that the input information of the FATE is the system performance, and the output is the action required to tune the scaling factors. For

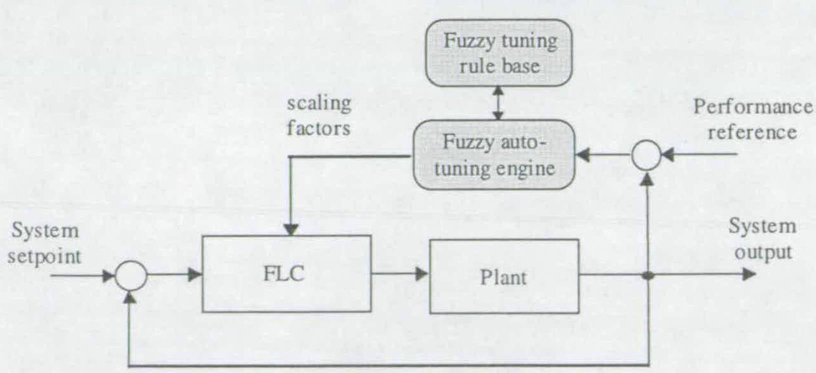


Fig. 5-2 Illustration of the auto-tuning fuzzy control system

the implementation of the fuzzy tuning algorithm, performance measures have to be selected and the variables representing the tuning actions have to be defined.

Two types of performance measures were used, corresponding to the initial phase tuning (including initialization, see section 5.2.3) and the final phase tuning of the scaling factors and sample rate. Initial phase tuning was based on the dynamic performance, measured by the error change rate $\dot{e}$, and it was designed to handle the special problems, such as oscillation or large damping. The final phase tuning was based on the performance measures: overshoot (OS), undershoot (US), number of ring cycles (CY) and integrated error square (IES). Note that these performance measures are detected during the transient period, thus they indicate the system dynamics and whether the steady state can be reached or not.

The membership functions for these performance measurements are defined in Fig.5-3(a) based on practical experience in designing control systems. Note that IES is normalized as follows:

$$IES = \frac{1}{Nw^2}\sum_{k=1}^{N} e^2(k)\,, \tag{5-1}$$

where w is the setpoint change, $e(k)$ is error at the instance k, N is total number of samples.

The output variables of the FATE are the scale factors GE, GC and GU. Membership functions for these scale factors, as shown in Fig.5-3(b), were selected with the help of practical tuning experience of fuzzy control systems. Fig.5-3(c) shows the membership functions (singleton type) of changes in the scale factors with the horizontal scale giving the required multiplier.

5.2.2 Classification of systems

To generate the tuning rules for the fuzzy control system, a classification of control systems is required, based on significant performance features. From the results of simulation experiments and practical knowledge of process structure, control systems can be divided into three groups: (1) simple systems, *i.e. first* order linear system without time delay; (2) complicated systems, *i.e.* time delayed systems and high order systems; (3) unstable

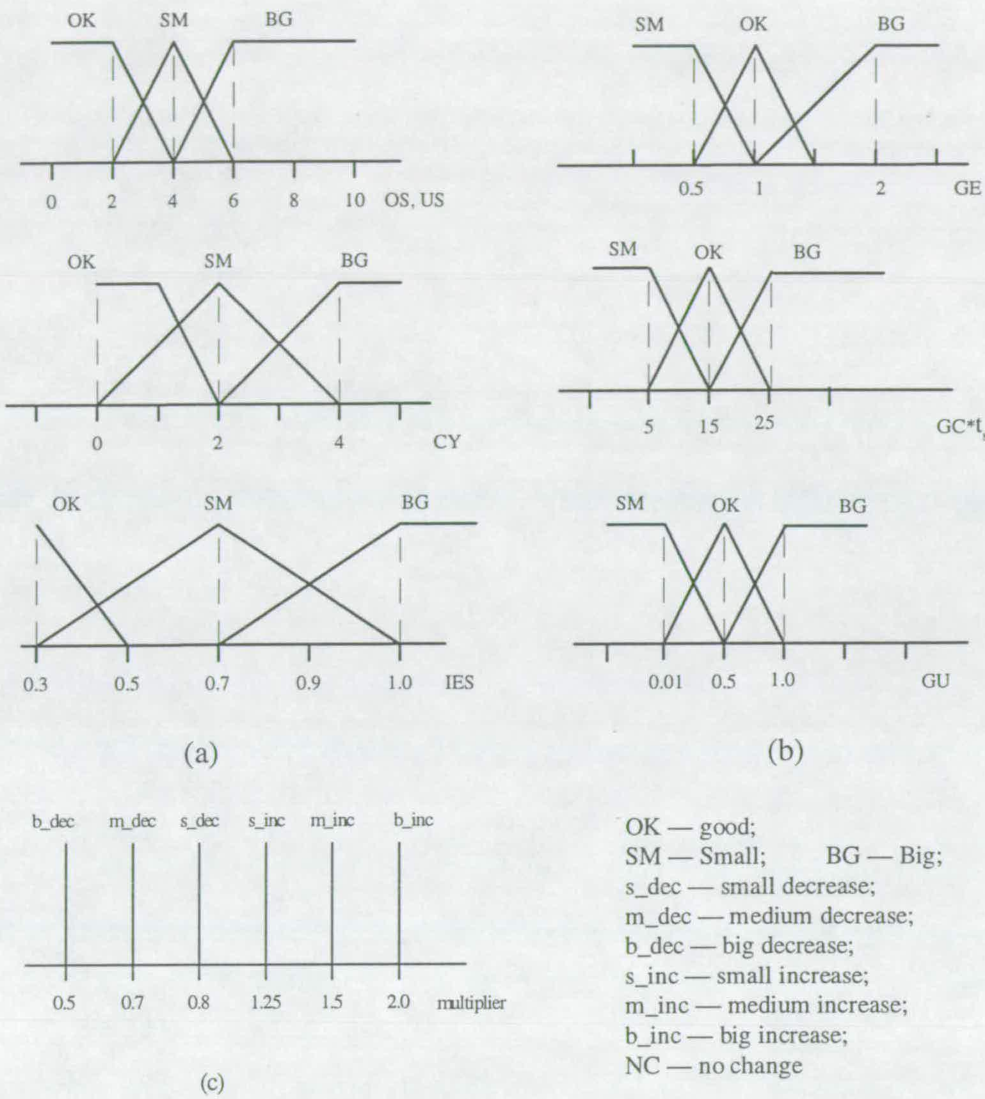


Fig. 5-3 Membership functions (MFs) of variables in the tuning rules
(a) MFs for system performance measurements;
(b) MFs for scale factors; (c) MFs for the changes in scale factors.

systems, such as the inverted pendulum system. The first two situations will be considered in this chapter.

A system is classified as either a simple or a complicated controllable system by noting features of the output change rate, $\dot{y}$, when a step input with fixed amplitude is applied to the open-loop system. Fig.5-4 illustrates the variations of $\dot{y}$ in the first order system and the second order system. It can be seen from Fig.5-4 that If $\dot{y}$ decreases monotonically within several sample periods, the system is a simple controllable system. Other systems with bounded OS, US and CY will be sorted as a complicated controllable system, such as any high order systems. A time delay system is put into the complicated category. Fig.5-5 illustrates the classification method.

In Fig.5-5, controlled processes were sorted according to the initial response of the system to a step input. Due to insufficient information about the controlled process, at this stage, the system was only sorted as either a simple or a complicated system. As more information becomes available from the system output and by evaluating the status of undershoot, overshoot and number of ring cycles generated by the process when controlled by a default fuzzy controller, the system will be further identified as oscillating, non-oscillating, or uncontrollable. In both oscillating and non-oscillating situations, performance will be further classified with respect to the status of overshoot, undershoot and integrated error square. Each status of the response is named as "Onn", where nn is a order number of these statutes, as shown in Fig.5-5. This classification has actually defined each of the fuzzy subsets on which the tuning rules are based.

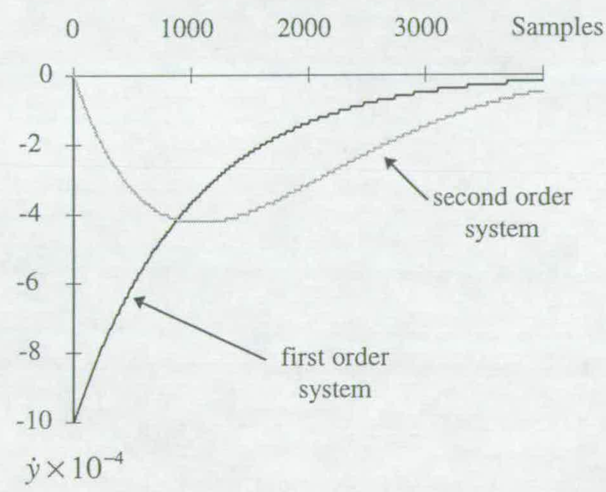


Fig.5-4 Illustration of the difference of output change rate between the first order system and the second order system

5.2.3 Tuning algorithm

The tuning algorithm is the key element in this auto-tuning technique. It consists of two parts: the initial phase tuning and the final phase tuning. The initial phase tuning is performed from the initialization through the whole transient period. Its objectives are to provide a suitable initial setting of the scaling factors and the sample rate, based on the system dynamic performance, and to adjust these tuned parameters during the transient period to prevent the system state from being out of the dynamic scope covered by the final phase tuning strategies. The final phase tuning is performed at the end of the transient period or when the system states are obviously out of the required performance specification, such as big overshoot or oscillation. It provides a full adjustment of the tuning parameters, based on fuzzy rules derived from practical experience of manual tuning.

Scaling factors and the sample rate are initialized at the beginning of the tuning procedure when only a few samples of error change $\dot{e}$ are available. Suppose two $\dot{e}$ samples are $\dot{e}(t_0)$ and $\dot{e}(t_1)$ ($t_1 > t_0$). The initial values of these parameters are empirically determined as follows. If $\dot{e}(t_0) \geq \dot{e}(t_1)$ and time delay $T_d = 0$, the controlled process is a first order process, and the scaling factors and the sampling rate are initialized as

$$GE = 1.0; GC = 7.0; GU = 0.5; t_s = 10. \tag{5-2}$$

Otherwise, the controlled process is a time delay process or a second order process, and the scaling factors and the sampling rate are initialized as

$$GE = 0.5; GC = 6.0; GE = 0.01, t_s = 10. \tag{5-3}$$

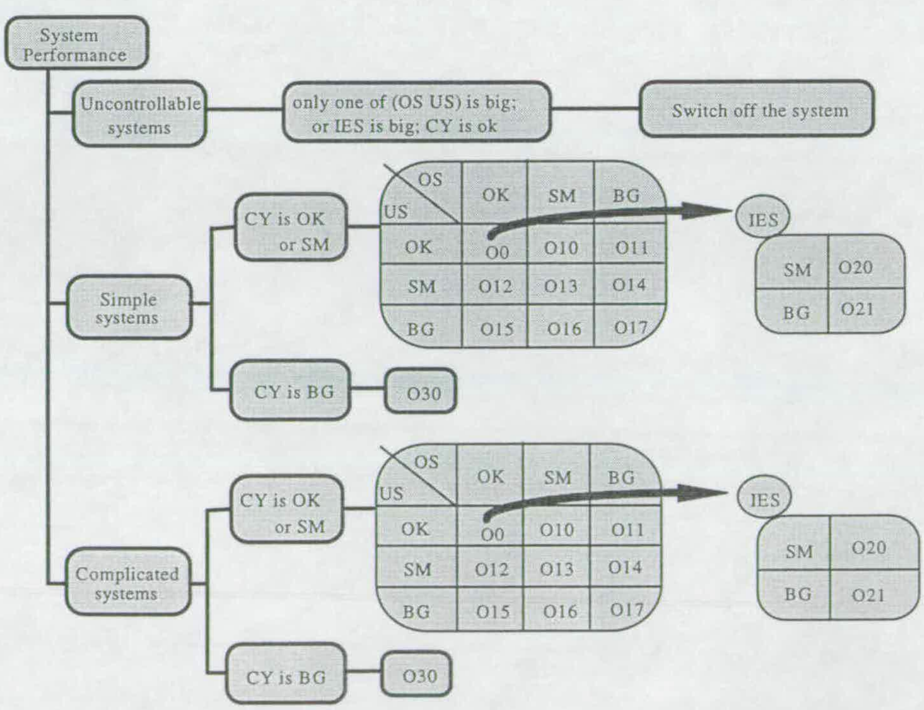


Fig.5-5 Classification of control systems and system's performance

The sample rate is set to $t_s = 1$ at the beginning of the tuning procedure to provide the fastest sampling. After a few samples of the error change $\dot{e}(t)$ was available, t_s is empirically adjusted to

$$t_s = 0.012/\dot{e}^*, \tag{5-4}$$

where $\dot{e}^*$ is the average value of the available samples of $\dot{e}(t)$.

In practice, the above initialization is very simple. It only provides a different initial response for different types of controlled process, *i.e.* a fast response in the first order process and a slow response in the case of the other processes. However, if the controlled process is dynamically fast, the above parameter setttings may lead to oscillation, and if the controlled process is dynamically slow, the control system may take long time to reach the final phase tuning stage.

To overcome these problems, the initial tuning phase has been applied to adjust the scaling factors and the sampling rate during the transient period so that the final tuning phase strategy can handle the system dynamic performance. The special situations considered in the initial tuning phase and the actions taken are listed in Table 5-1. Note that the tuning actions in Table 5-1 are also based on the practical experience of manual tuning.

By using *trial and error* and simulation methods, tuning rules were generated for different system performance status. Exhaustive simulation tests have been carried out to find the proper action taken to modify the scaling factors and sample rate t_s in different situations. Table 5-2 and Table 5-3 give these heuristic tuning rules for both simple and complicated controllable systems respectively. Symbols in the tables come from the definitions in Fig.5-5. From these tables and Fig.5-5, a rule, say rule No.10 for simple systems, can be read as:

IF number of ring cycle **IS** good,

AND overshoot **IS** good,

AND undershoot **IS** good,

AND integrated error square **IS** big

THEN small increase GE, big decrease GC, small increase GU.

It must be stressed that the tuning rules in Table 5-2 and Table 5-3 are based on the fuzzy logic controller defined in *Chapter* 3 and the first and second order processes.

In addition, in contrast with tuning a complicated system, the sample rate in a simple system is not changed while tuning the scaling factors. Sample rate is calculated at the begin of the tuning procedure when the first feedback information is available. This method leads to a reasonable performance in simple systems and may lead to oscillation in complicated systems. Therefore tuning the sample rate is necessary in the complicated system situation.

5.2.4 Tuning procedure

Before applying the above tuning rules, information about the performance of the control system is needed. To excite the controlled process to show up its properties in response to a step input without the system generating unacceptable signal changes, the selection of the control signal in the tuning stage becomes crucial. In this research, the following methods have been used for safe starting:

1. Closed-loop fuzzy logic control was used. Because fuzzy logic control can be set to give a very gentle action, though the response may be too slow, big overshoot and undershoot in the system response can be restricted. After more information about the system has been collected, the fuzzy controller parameters can be tuned to improve the performance.

Table 5-1 Initial phase tuning strategies

Situations	Dynamic behavior	Action taken
Big overshoot or undershoot	either OS or US is BIG but CY is OK;	start final phase tuning
Oscillation	CY is BIG;	start final phase tuning
Large steady state error	$\dot{e}$ is ZERO but e is NOT ZERO;	increase GE
Fast oscillation of $\dot{e}$	sign of $\dot{e}$ changes fast and the oscillation amplitude of $\dot{e}$ is big.	decrease GC
Fast oscillation of $\dot{e}$ around the switching line	sign of Δu changes fast.	decrease GU
Large damping	system state (e, $\dot{e}$) has been far from the switching line for a certain time.	increase GC and GU

Table 5-2. Rules for tuning simple controllable processes

Rule No.	System status	Other Conditions	GE action	GC action	GU action
1	O10			s_inc	
2	O11			b_inc	
3	O12			s_dec	b_dec
4	O13			s_inc	s_inc
5	O14			b_inc	
6	O15		s_dec		b_dec
7	O16		s_dec	b_dec	
8	O17			s_inc	b_inc
9	O20			s_dec	s_inc
10	O21		s_inc	b_dec	s_inc
11	O30	GC is BG	m_dec	b_dec	
12	O30	GU is BG			b_dec
13	O30	GC is OK GU is OK	m_dec		

Table 5-3. Rules for complicated controllable processes

Rule No.	System status	Other Conditions	t_s Action	GE action	GC action	GU action
14	O10					s_dec
15	O11			s_dec	m_inc	m_dec
16	O12				s_inc	s_dec
17	O13				s_inc	
18	O14			s_dec	m_dec	m_dec
19	O15			s_dec	s_inc	m_dec
20	O16			s_dec	m_dec	m_dec
21	O17			s_dec		m_dec
22	O20		s_dec	s_inc		s_inc
23	O21		s_dec	m_inc	b_dec	m_inc
24	O30	IES is SM	m_inc	s_inc		m_dec
25	O30	IES is BG		m_inc		b_dec

2. A fraction of the required setpoint change of the tuned system was chosen to be the practical setpoint change at the tuning stage. This is used to prevent the process from overdriving in most circumstances. Also this method will lead to more than one tuning procedure in the tuning stage, and therefore gives more chance for the tuning algorithm to improve its tuning performance.

Fig.5-6 illustrates the tuning procedure. The process control input is kept at zero for an initial period of 100 sample intervals which are initialized to give a relatively fast sampling action. During this period (stage 1) the system output $y(t)$ is monitored to check for plant noise or disturbances which will be used for the calculation of the system status. At stage 2, the system is configured as an open-loop to measure the time delay and set the sample rate. The control input of the process is switched on with a fixed amplitude of 10% of the difference between the set point and the present process output. The system dead time T_d is considered as the time period from the beginning of stage 2 to the time when the output of the process begins to change (see Fig.5-6). As the system output begins to increase, 10 samples of process output are taken and used together with the dead time T_d to set the sample rate t_s and initial settings of the parameters of the controller. Next, in stage 3, the system is configured as a closed loop system and the control task is handed on to the fuzzy controller with the modified setpoint change described above. As the system output progresses to the steady state, the system performance will be calculated, heuristic tuning rules will be applied and FLC scale factors will be tuned. After that, if the tuning setpoint is not equal to the required setpoint, the tuning setpoint will be renewed and a new tuning process begins (stage 4). After the all tuning stages, the FLC with fixed scale factors will take the control task (stage 5).

If the system loses control during the tuning stage, the tuning algorithm will switch off the control input and bring the system to a safe stop condition. Under some circumstance, for example the cart-pole system, this method of dealing with unstable processes may fail, but it will lead to a safe stop in most practical situations.

5.3 Simulations

To verify the adequacy of the fuzzy tuning method described above, an auto-tuning function was implemented in the simulation tool *FzySimu* and applied to perform the tuning tasks of four fuzzy logic control systems. The transfer functions of these processes are as follows:

$$\text{I:} \quad G(s)=\frac{K}{1+Ts} \tag{5-5}$$

$$\text{II:} \quad G(s)=\frac{Ke^{-T_d s}}{1+Ts} \tag{5-6}$$

$$\text{III:} \quad G(s)=\frac{K\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2} \tag{5-7}$$

$$\text{IV:} \quad \dot{y}(t)=\frac{1}{T}\left(\frac{1-e^{-y(t)}}{1+e^{-y(t)}}+Ku(t)\right) \tag{5-8}$$

where K is the static gain, T is time constant, T_d is time delay, ω_n is the natural frequency, ζ is the damping ratio, $u(t)$ and $y(t)$ are input and output of the process respectively. The first three processes are linear processes and the non-linear process IV [32] was used to demonstrate the ability of the tuning algorithm to cope with other fuzzy control systems which the tuning algorithm was not designed for.

The process parameters at the tuning point were set as follows. Note that the units for the time variables, such as T and T_d, is the simulation time scale in the computer.

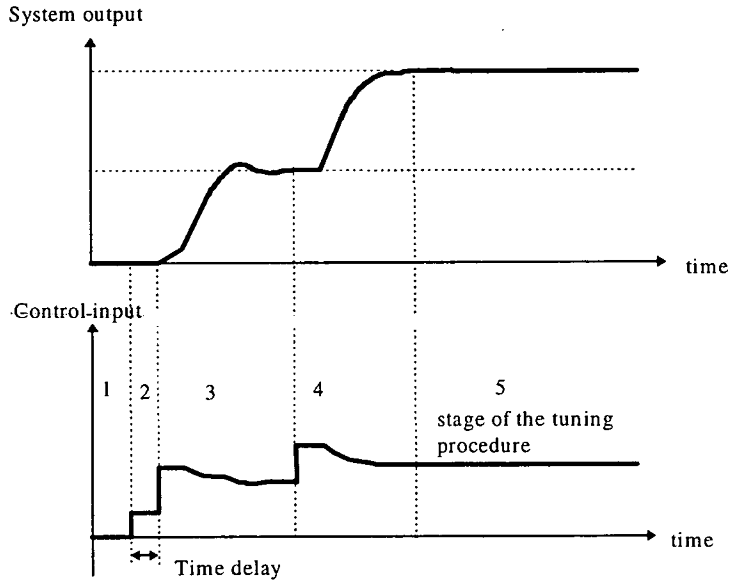


Fig.5-6 The auto-tuning procedure

Process I: $K = 5,\ T = 1000$;

Process II: $K = 5,\ T = 1000,\ T_d = 100$;

Process III: $K = 2.5,\ \omega_n = 5.0;\ \xi = 0.8$;.

Process IV: $K = 10,\ T = 1000$.

The standard fuzzy logic controller was used to control these processes. The rule base and the membership functions for the input and output variables are as shown in Fig.3-4 in *Chapter* 3.

The FLC parameters obtained after auto-tuning are as shown in Table 5-4. Fig.5-7, Fig.5-8 Fig.5-9 and Fig.5-10 illustrate the performances of the four systems respectively. Each figure shows the step response of the system before, during and after the tuning procedure. Note that the scaling factors and the sample rate used in the pre-tuning tests are set as shown in (5-2) and (5-3).

From Fig.5-7 to Fig.5-9, it can be clearly seen that without tuning, a generally designed fuzzy logic controller with the initial settings of the tuned parameters gave an unacceptable performance even when it was used to control a simple process like process I. This poor performance can be explained as the poor mapping of controller input and output variables from the practical variable spaces to the designed ones. This demonstrates that it is necessary to tune the scaling factors and the sampling rate in a fuzzy logic control system to meet the design specifications.

Table 5-4. FLC parameters after tuning

System	GE	GC	GU	t_s
I	1.0 (1.0)	7.0 (7.0)	0.5 (0.5)	21 (10)
II	0.8 (0.5)	5.68 (6.0)	0.09 (0.01)	46 (10)
III	1.0 (0.5)	5.73 (0.6)	0.129 (0.01)	24 (10)
IV	1.0 (0.5)	14 (0.6)	0.105 (0.01)	13 (10)

Note: initial values are shown in brackets.

The performances of the tuned systems shown in the figures appear to be unsatisfactory during the tuning stage, because the output of the system stopped rising in the middle of the tuning procedure. This is because two tuning procedures are performed for each system and each procedure needs a complete system response before carrying out the tuning task. It is found that, for better performance, two or more tuning procedures are generally needed if the controlled process is a complicated one.

It can be found from the results that an oscillatory system, before tuning, performed much better after applying the proposed fuzzy tuning algorithm. And from Table 5-4, it can be found that the scaling factors and sample rate were changed significantly by the tuning operation. Though the parameters may not be optimized, for example, the step response in Fig.5-7(c) could be faster, the system's responses are acceptable from an engineering point of view.

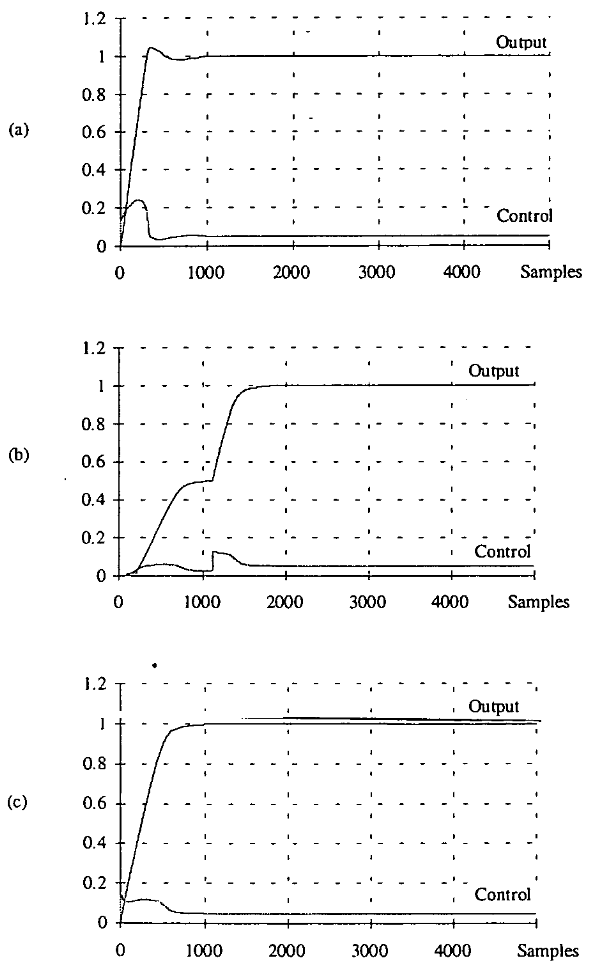


Fig.5-7 Step response of system I before (a), during (b), and after (c) tuning.

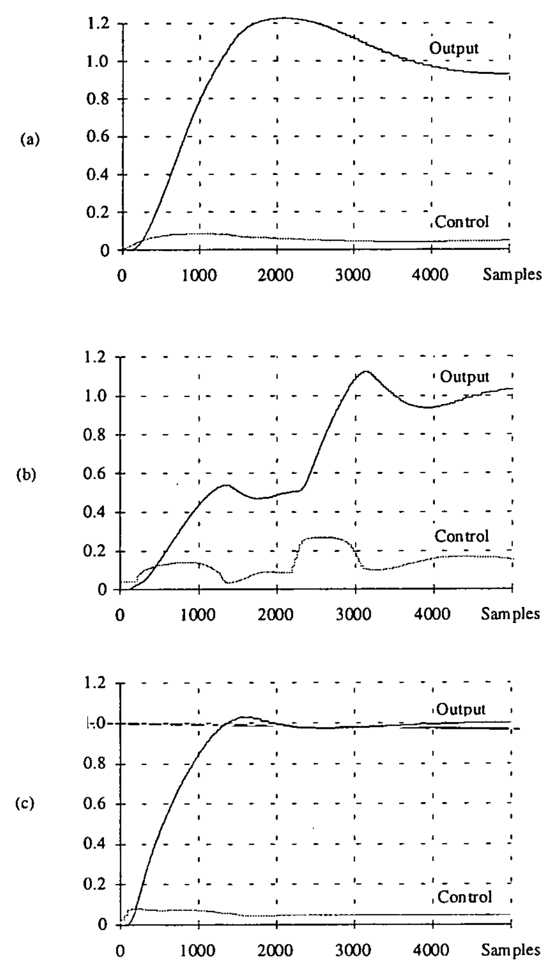


Fig.5-8 Step response of system II before (a), during (b), and after (c) tuning.

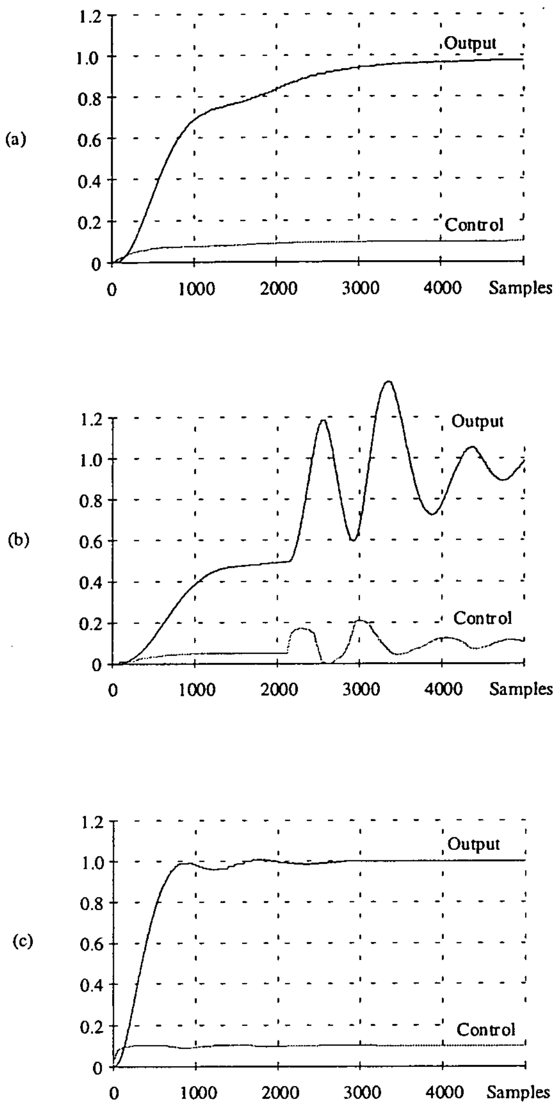


Fig.5-9 Step response of system III before (a), during (b), and after (c) tuning.

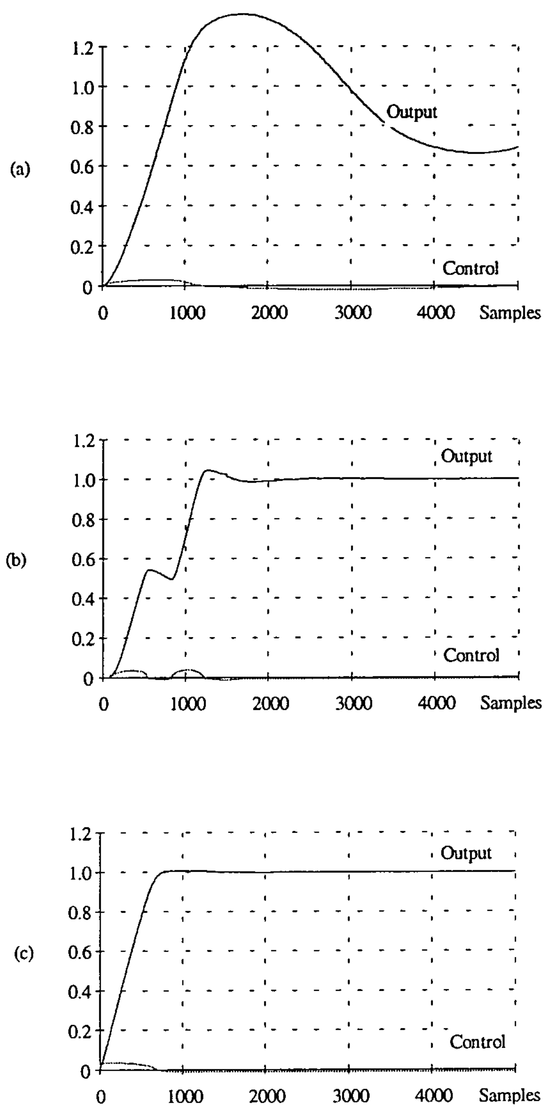


Fig.5-10 Step response of system IV before (a), during (b), and after (c) tuning.

The robustness of the three linear auto-tuned FLC systems described above was tested and compared with that of their manually tuned counterparts. The tuning point and the tested parameter space of each system are as follows;

System I: tuning point $K = 5$, $T = 1000$;

tested parameter space: $K = (1 \sim 20)$, $T = (50 \sim 2000)$;

System II: tuning point $K = 5$, $T = 1000$, $T_d = 100$;

tested parameter space: $K = (1 \sim 20)$, $T = (50 \sim 2000)$;

System III: tuning point $K = 2.5$, $\omega_n = 5.0$, $\zeta = 0.8$;

tested parameter space: $K = (1 \sim 20)$, $\omega_n = (2 \sim 12)$.

Each system was manually tuned to give a satisfactory performance at the tuning point; and the parameters obtained were used in the robustness test of this system. Then, the auto-tuning algorithm was applied to each system, and, similarly, the robustness of the resulted system was tested. The tuned parameters resulting from the two different tuning methods and the robustness of the tuned systems are given in Table 5-5.

From the results it can be seen that the tuned parameters obtained from the two tuning methods are quite different. In the first order system, the auto-tuning method generated a slower sample rate and a smaller GC than the manual tuning did. If the system is complicated, the auto-tuning method produced a slower sample rate and a larger GU. This is because different methods were used to handle the unsatisfactory response in the two types of tuning. For example, to decrease the overshoot in the simple system, the auto-tuning method decreased the sample rate, while the manual tuning method increased GC.

With respect to system robustness, the two tuning methods gave similar results except in the time delayed system where the manually tuned system generated much better robustness. The robustness of the second order system should be much better according to the test

Table 5-5 Comparison of auto-tuned FLC systems with manually tuned FLC systems with respect to tuned parameters and system robustness

System	Manual tuned FLC		Auto-tuned FLC	
(Tuning point, TP)	Parameters	I_{rb} (%)	Parameters	I_{rb} (%)
I: First order process (TP: K=5, T=1000)	GE = 1.0 GC = 18 GU = 0.5 t_s = 5	54.8 (K-T)	GE = 1.0 GC = 7.0 GU = 0.5 t_s = 21	63.8 (K-T)
II: First order process with time delay (TP: K=5, T=1000, T_d = 100)	GE = 0.8 GC = 16 GU = 0.05 t_s = 20	38.6 (K-T)	GE = 0.8 GC = 5.68 GU = 0.09 t_s = 46	13.5 (K-T)
III: Second order process (TP: K=2.5, ω_n =5.0, ζ=0.8)	GE = 0.7 GC = 4.0 GU = 0.04 t_s = 18	3.2 (K-ω_n)	GE = 1.0 GC = 5.73 GU = 0.129 t_s = 24	4.4 (K-ω_n)

results in *Chapter* 4; and the robustness of the first order system is quite outstanding. Note that system robustness was not addressed in both tuning methods, and similar system performance at the tuning point was not obtained, which can influence system robustness.

It is worth noting that the empirical nature of this fuzzy tuning algorithm makes it quite difficult to obtain the general tuning rules and to define the membership functions of each variable. Some tuning rules suitable for one type of fuzzy systems may not benefit other fuzzy systems. This is the main reason for performing the process classification before carrying out system tuning, and it does effectively reduce the difficulty in gathering the tuning rules. Due to time limitations, only two types of controlled processes were studied and the process parameter settings at the tuning point were restricted to the following conditions:

$$\frac{T'}{K} < 50, \quad \frac{T'}{T_d} > 5, \tag{5-9}$$

where K is the process gain, T' is the time constant or the equivalent time constant $1/(\zeta\omega_n)$ of the process, T_d is the time delay in the process.

5.4 Summary

A great number of parameters define the performance of a fuzzy control system. Hence its tuning at most times becomes a critical and very time consuming operation which can provide unsatisfactory system results. In this *Chapter*, a fuzzy tuning algorithm is developed for the purpose of auto-tuning the scaling factors and the sampling rate in the standard fuzzy controller. Based on extensive simulation experiments, fuzzy tuning rules for scaling factors and sample rate have been generated. A complete tuning procedure was designed. The adequacy of this tuning algorithm has been tested by simulations, carried out on processes which are commonly used as the approximation of the practical processes and also an artificially designed non-linear processes used in [32]. Results show that the fuzzy tuning algorithm is effective, and the time and effort used in tuning the fuzzy controllers was significantly reduced.

Because of the time limitation and the difficulty in the rule collection, the tuning rule base discussed in this chapter does not cover all tuning situations. In addition, system robustness was not addressed by the tuning algorithm. Further research work is needed to confirm its effectiveness in other tuning situations, and to investigate the tuning strategy for improving system robustness.

6

Summary and Conclusions

6.1 Summary

This thesis has presented the results of an investigation of the robustness of the widely used Mandani-type fuzzy logic control systems under wide variation of the parameters of the controlled SISO processes.

The basic concepts of fuzzy set and fuzzy logic were reviewed to provide a theoretical background for this research work. The FLC is represented by five functional blocks: scaling, fuzzification, reasoning, rule base, and defuzzification. Studies of FLC systems have been briefly reviewed from the point of view of parameter selection and rule derivation. A literature survey was carried out in the area of robust control with fuzzy algorithms.

To investigate the robust performance of the FLC systems, the measurements of the dynamic performance and system robustness of a control system were firstly defined from an engineering point of view, and the concepts of robust space and robustness index were introduced. The robustness of the FLC systems was investigated by analyzing the structure of the fuzzy rule base and membership functions of the input-output variables. Based on the close relation of the fuzzy rule base and the system dynamic trajectory on the phase plane, a switching line method was introduced to qualitatively analyze the dynamic performance of the SISO FLC systems. Based on the switching line method, the robustness of the FLC controlled first and second order processes was predicted with respect to the shape and position of the robust space.

The effects of FLC parameters (membership functions of the input-output variables, rule base and scaling factors) on system robustness were investigated. The movements of the position and the shape of the switching line with the variation of the controller parameters were analyzed, and its relation with the system performance was reported.

To improve the robustness of the FLC system, three design methods were proposed. The first one was based on the switching line characteristic of the FLC system. The second method, called phase advanced FLC, was introduced to handle the control of high order processes with fuzzy algorithms. The third method was an evolutionary method based on a genetic algorithm, which was used to automatically design a robust fuzzy control system, assuming the availability of the controlled process model.

To demonstrate the validity of the analytical results and design methods, the robustness of FLC systems was investigated by further experiments. At first, performance specifications and the experimental procedure were defined. A simulation tool was developed to perform the system simulation, robust space mapping and optimization by GA. The simulation results were also confirmed by practical results derived from practical control systems.

With the simulation method, the robustness of the fuzzy logic control system was investigated by controlling first order and second order processes. The FLC was tuned at a selected tuning point to achieve an acceptable system response; then system performances at all selected points within a prescribed process parameter space were tested and the system robust space was measured. The results from the qualitative analysis of the robustness of fuzzy control systems was confirmed. The factors affecting the position of the robust space was investigated by observing the system dynamic performance on the boundaries of the robust space. A comparative study was also performed with the widely used PID control systems.

Effects of the rule base, the membership functions, and the scaling factors on system robustness were examined. In the rule base tests, emphasis was placed on the effect of the variations of the switching line on the system robustness. In the membership function tests, the effect of the overlap ratio of the membership functions was investigated and seven FLCs with different overlapped membership functions were examined. In the scaling factor tests, the sensitivity of system robustness to each scaling factor was assessed.

Experimental evaluation of the proposed design methods for the FLC system was performed by applying them to control different processes. An FLC was designed by the switching line method to control a third order non-minimum phase process. Two phase advanced FLCs were respectively applied to controlling a non-integrating second order process and an integrating second order process. The genetic algorithm was used to obtain an optimized design of a FLC controlled cart-pole system with respect to system robustness.

Finally, to overcome the tuning difficulty of FLC systems, effort was made to develop a fuzzy tuning algorithm to perform the on-line auto-tuning of the FLC controlled first order and second order processes. Based on extensive simulation experiments, the fuzzy rules for tuning scaling factors and the sample rate were collected. A complete tuning procedure was designed. The adequacy of this tuning algorithm was assessed.

6.2 Conclusions

The superior robustness of the widely used fuzzy logic control technique is claimed in many of the publications reviewed in *Chapter* 2. However the evidence is mainly anecdotal and only a few research papers on the robustness of fuzzy control systems have been found. Investigation of the sensitivity of system performance due to parameter variations was performed experimentally by both Mamdani and Kasko. Studies on the robust stability of fuzzy control systems were found in Tanaka's work. The robust performance of fuzzy control systems was investigated primarily by Palm and a few others by using the principle of the sliding mode control. A review of this research showed that a complete investigation of the robust performance of FLC systems was not yet available. In addition, there was no systematic method to analyze the stability of FLC systems, thus it is very difficult to investigate the robust performance of fuzzy control systems.

As a result of the investigation into the close relation of the fuzzy rule base and the system dynamic trajectory on the phase plane presented in *Chapter* 3, a switching line method was introduced to qualitatively analyze the dynamic performance of the SISO FLC systems. It was shown that the position of the switching line on the phase plane determines the dynamic performance of the FLC, and the objective of the fuzzy control algorithm is to drive the system phase trajectory to the steady state along the switching line.

The following conclusions are drawn from the qualitative analysis presented in *Chapter* 3 and the experimentally results presented in *Chapter* 4: (1) The robustness of the FLC systems arises from the constraints introduced by the rule base determined switching line and it depends very little upon the model of the plant to be controlled; (2) The robust space of the first order FLC system is a sector in the *K-T* plane, and its area is significantly reduced when a time delay is added to the process; (3) The robust space of the second order FLC system is an approximate sector shape on the $K — \zeta$ plane and a curved sector shape on the $K — \omega_n$ plane; (4) A properly designed and tuned fuzzy logic control algorithm is more robust than the PID control algorithm in controlling the first order and second order processes; (5) The position of the tuning point and the performance of the fuzzy logic control system at the tuning point affect the position and the size of the system's robust space in the process parameter space; (6) The standard FLC exhibits less robustness in controlling high order processes than in controlling low order processes, and knowledge about high order dynamics of a controlled process should be included in the fuzzy rule base for controlling high order dynamic systems.

The robust space of a fuzzy control system can be optimized by changing the FLC parameters. The rule base, the input membership functions, and the input scaling factors can affect the position and/or the shape of the switching line. The robustness of a fuzzy control system is sensitive to the position of the switching line, but less sensitive to the shape of the switching line and the overlap of membership functions. The switching line angle proportionally controls system overshoot and undershoot, but it exerts inverse proportional control of the rise time in the step input response. Changes to the overlap of membership functions in the input space alter the sensitivity of FLC on variation of input variables in that space. The effect of output membership functions and output scaling factor on system performance is equivalent to a variation of process gain.

If the open-loop dynamics of the controlled SISO process is available, the switching line method can be used to design a FLC to achieve robust performance within an appropriate parameter space of the process. The membership functions of the input output variables of the controller can be determined by meeting the reaching condition and the sliding condition of the system at the operating point. Adjustment of the required robust space and the required system performance may be needed. Tuning of the scaling factors and the sampling rate is also important for obtaining the required performance. The main problem of this method is the requirement for knowledge of the process dynamic performance which may not be available in practice.

If the controlled process is high order, the phase advanced methods can be used to improve system robustness. This is a simple method but it has been experimentally demonstrated as an effective method. If the dynamic model of the controlled process is available, the robustness of the fuzzy control system can be optimized by using the genetic algorithm. It has also been demonstrated to be an effective method for obtaining the global optimization.

Among all the experimental investigations, however, it is found that (i) knowledge about the controlled process is very desirable for the design of a robust fuzzy control system, especially information about the dynamic changes in the process; (ii) as indicated before, tuning the fuzzy control system is very difficult but important for obtaining the satisfactory system performance.

The tuning difficulty encountered in all fuzzy control systems is addressed in *Chapter* 5. The effectiveness of the auto-tuning algorithm, employing fuzzy logic theory and the author's experience, is experimentally demonstrated to be effective in the case of first order and second order processes. Results also show that the fuzzy tuning algorithm can be used to tune some non-linear fuzzy control systems. It has been found that the system performance was significantly improved by the tuning algorithm with respect to the performance specifications. The tuning method can be used to develop more general and more powerful tuning tools for control systems.

Finally, robust control with the fuzzy logic algorithm is still a developing research area. Many difficulties in theoretical analysis, practical experiments and literature references were encountered in this work. Great effort was directed to establishing an experimental basis for an analytical approach to fuzzy control systems.

6.3 Recommendation for future work

The following areas of investigation constitute the recommendations for future work.

- Quantitative analysis of the robustness of the fuzzy control systems, including the effects of controller parameters on system robustness.
- Extending the switching line theory to MISO fuzzy control systems.
- Investigation of auto-tuning strategies for improving system robustness and extension of the fuzzy tuning algorithm for more general fuzzy control systems.
- A full investigation of the systematic design method for a required robust performance.
- Investigation into the effects of noise on system performance.
- Development of software tools for analyzing system robustness.
- Investigation of the systematic design methods of fuzzy controllers in the case of non-linear processes.

The results of the research presented in this thesis indicate that the switching line method, the phase advanced FLC, the GA optimization technique, the auto-tuning algorithm and the simulation tool *FzySimu* provide solutions to some of the present day problems with fuzzy control systems. It is hoped that the work presented in this thesis will have made a clear contribution to this challenging field.

Bibliography

[1] R. C. Dorf, *Modern control systems,* Addison-Wesley Publishing Company, Inc. 1992.

[2] M. Morari, and E. Zafiriou. *Robust process control,* Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[3] L. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man. and Cybernetics,* SMC-3, 28-44, 1973.

[4] D. W. St. Clair, *Controller tuning and control loop performance*, Published by *Straigh-line Control Company, Inc.* 1995.

[5] E. H. Mamdani, "Twenty years of fuzzy control: experiences gained and lessons learnt," *IEEE Proceedings,* vol.1, pp.339-344, 1993.

[6] E. H. Mamdani, "Fuzzy control in Japan – learning lessons," *IEE Proceedings,* vol.(?), 1993.

[7] R. Jager, *Fuzzy logic control,* PhD Thesis, Technische Universiteit Delft, 1995.

[8] G. F. Mauer, "A fuzzy logic controller for an ABS breaking system," *IEEE Transactions on Fuzzy Systems,* vol.3, No.4, pp.381-388, November 1995.

[9] K. Tanaka and M. Sano, "Concepts of stability margin for fuzzy systems and design of robust fuzzy controllers," *Proceeding of Second IEEE International Conference on Fuzzy Systems,* vol.1, pp.29-34, 1993.

[10] R. Palm, "Sliding mode fuzzy control," *IEEE Int. Conf. on Fuzzy Systems*, San Diego, pp.519-526, 1992.

[11] P. Wang and D. P. Kwok, "Analysis and synthesis of an intelligent control system based on fuzzy logic and the PID principle," *Intelligent System Engineering*, pp.157-171, winter 1992.

[12] H. T. Nguyen, C.W. Tao and W.E. Thompson, "An empirical study of robustness of fuzzy systems," *Proceedings. of 2nd IEEE International Conference on Fuzzy Systems*, vol.2, pp.1340-1345, 1993.

[13] M. Raae and D. A. Rutherford, "Theoretical and linguistic aspects of the fuzzy logic controller," *Automatica,* vol.15, pp.553-577, 1979.

[14] J. Q. Chen, J. H. Lu, and L. J. Chen, "Analysis and synthesis of fuzzy Closed-loop control systems," *IEEE Transaction On Systems, man and Cybernetics*, vol.25, No.5, pp.881-888, 1995.

[15] R. M. Tong, "Some properties of fuzzy feedback systems," *IEEE Transaction On Systems, Man and Cybernetics*, vol.10, No.6, pp.327-330, 1980.

[16] W. Pedrycz, *Fuzzy control and fuzzy systems*, Research Studies Press Ltd., John Wiley & Sons Inc., 1993.

[17] J. Favilla, A. Machion and F. Gomide, "Fuzzy traffic control - adaptive strategies," *Second IEEE Int. Conf. on Fuzzy Systems,* vol.2, pp.506-511, 1993.

[18] R. N. Lea, J. Hoblit and Y. Jani, "Fuzzy-logic based robotic arm control," *Second IEEE Int. Conf. on Fuzzy Systems,* vol.2, pp.140-147, 1993.

[19] S. W. Kim and J. J. Lee, "Resolved motion rate control of redundant robot using fuzzy-logic," *Second IEEE Int. Conf. on Fuzzy Systems,* vol.2, pp.333-338, 1993.

[20] H. Ying, M. Mceachern, D. W. Eddleman and L. C. Shephard, "Real-time fuzzy control of mean arterial-pressure in postsurgical patients in an intensive-care unit," *Second IEEE Int. Conf. on Fuzzy Systems,* vol.2, pp.255-260, 1993.

[21] G. Klett, "Application of fuzzy control in chemical distillation processes," *Second IEEE Int. Conf. on Fuzzy Systems,* vol.2, pp.375-377, 1993.

[22] Whitbread, "Development tools for neuralogix' fuzzy microcontroller," *Embedded System Engineering*, vol. 1, no. 4, pp. 18-19, 1993.

[23] T. Yamakawa and K. Miki, "A simple fuzzy computer hardware system employing min and max operations — A challenge to 6th generation computer," *Proc. 2nd IFSA congress,* Tokyo, Japan, pp.551-555, July 1987.

[24] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller—part I and II," *IEEE Tran. on Syst., Man, and Cybern.,* vol.20, no.2, pp.404-435, 1990.

[25] R. R. Yager and D. P. Filev, *Essentials of Fuzzy Modeling and Control,* John Wiley & Sons, Inc. 1994.

[26] L. Zadeh, "Fuzzy sets," *Informat. Control*, vol. 8, pp. 338-353, 1965.

[27] R. Sutton and D.R. Towill, "An introduction to the use of fuzzy sets in the implementation of control algorithms", *J. Inst. of Electronic and Radio Engineers*, vol.55, no.10, pp.357-367, 1985.

[28] L. A. Zadeh, "Fuzzy sets and systems," *Proc. Symp. Syst. Theory*, Polytech. Inst. Brooklyn, pp.29-37, 1965.

[29] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies 7*, pp.1-13, 1975.

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

[31] T. J. Procyk and E. H. Mamdani, "A linguistic self-oganizing process controller," *Automatica*, vo.15, no.1, pp.15-30, 1979.

[32] L. X Wang, Stable Adaptive Fuzzy Control of Nonlinear Systems. *IEEE Transactions of Fuzzy Systems*, vol. 1, No. 2, pp.146-155. May 1993.

[33] M. Braae, D.A. Rutherford, "Selection of parameters for a fuzzy logic controller," *Fuzzy Sets and Systems*, No.2, pp.185-199, 1979.

[34] W. J. M. Kickert and E. H. Mamdani, "Analysis of a fuzzy logic controller," *Fuzzy Sets and Systems*, No.1, pp.29-44, 1978

[35] B. Pease, "What's all this fuzzy logic stuff, anyhow? (Part I)," *Electronic Design,* pp.77-79, May 13, 1993.

[36] B. Pease, "What's all this fuzzy logic stuff, anyhow? (Part II)," *Electronic Design,* pp.95-98, Novenber 1, 1993.

[37] B. Pease, "What's all this fuzzy logic stuff, anyhow? (Part III)," *Electronic Design,* pp.105-108, Novenber 11, 1993.

[38] J. M. Mendel and S. Chand, *Fuzzy logic systems for signal processing and control,*

A three day short course presented by United Signals & Systems, Inc., 1993.

[39] M. Braae and D.A. Rutherford, "Fuzzy relations in control setting," *Kybernetes 7*, pp.185-188, 1978.

[40] L. I. Larkin, "A fuzzy logic controller for aircraft flight control," *Industrial Applications of Fuzzy Control,* M. Sugeno (ed.), Amsterdam, North Holand, pp.87-104, 1985.

[41] E. M. Schaf and N. J. Mandic, "The application of a fuzzy controller to the control of a multi-degree-of-freedom robot arm," *Industrial Applications of Fuzzy Control,* M. Sugeno (ed.), Amsterdam, North Holand, pp.87-104, 1985.

[42] L. X. Wang and J.M. Mendel, "Generating fuzzy rules by learning from examples", *IEEE Trans. Syst., Man, Cybern.*, vol.22, no.6, pp.1414-1427, 1992.

[43] I. G. Umbers and P.J. King, "An analysis of human-decision making in cement kiln control and the implication of automation," *Int. J. Man Mach. Studies,* vol.12, no.1, pp.11-23. 1980.

[44] T. Takagi and M. Sugeno, "Derivation of fuzzy control rules from human operator's control action," *Proc. of the IFAC Symp. on Fuzzy Information, Knowledge Representation and Decision Analysis,* Marseilles, France, July 1983, pp.55-60.

[45] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man and Cybern.,* vol.17, no.1, pp.116-132, 1985.

[46] S. Isaka and A.V. Sedald, "An optimization approach for fuzzy controller design," *IEEE Trans. Syst., Man and Cybern.*, vol.22, no.5, pp.1469-1473, 1992.

[47] G. M. Abdelnour, C.H.Chang, F.H. Huang and J.Y. Cheung, "Design of a fuzzy controller using input and output mapping factors," *IEEE Trans. Syst., Man and Cybern.*, vol.21, no.5, pp.952-960, 1991.

[48] R. Palm, "Scaling of Fuzzy Controllers Using the Cross-Correlation", *IEEE Transactions on Fuzzy Systems*, vol..3, No.1, pp.116-123, 1995.

[49] C. K. Karr and E. J. Gentry, "Fuzzy control of pH using Genetic algorithms," *IEEE Trans. on Fuzzy Systems*, vol.1, no.1, pp.46-53, 1993.

[50] H. X. Li and H. B. Gatland, "Enhanced methods of fuzzy logic control," *Proceedings of Fuzz-IEEE/IFES'95*, vol.1, pp.331-336, Yokohama, Japan, Mar. 1995.

[51] M. Togai and P. Wang, "Analysis of a fuzzy dynamic system and synthesis of its controller," *Int. J. Man Mach. Studies,* vol. 22, pp.355-363, 1985.

[52] A. Homaifar and E. McCormick, "Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms," *IEEE Tran. Fuzzy Syst.*, vol. 3, no. 2, pp.129-138, May 1995.

[53] E. H. Mamdani, "Applications of fuzzy algorithms for simple dynamic plant," *Proc. IEEE,* vol.121, no.12, pp.1585-1588, 1974.

[54] S. G. Kong and B. Kosko, *Neural networks and fuzzy systems, Chap. 9 Comparison of fuzzy and neural truck backup control system*, Prentic-Hall international, Inc., pp.339-361, 1992

[55] C. M. Liaw and J. B. Wang, "Design and implementation of a fuzzy controller for

a high performance induction motor drive," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 4, pp.921-929, 1991.

[56] S. Chiu and S. Chand, "Fuzzy logic for control of roll and moment for a flexible wing aircraft," *IEEE Control Systems*, vol. 11, no. 4, pp. 42-48, 1991.

[57] F. R. Rubio, M. Berenguel and E. F. Camacho, "Fuzzy logic control of a solar power plant," *IEEE Trans. Syst., Man, Cybern.*, vol. 3, no. 4, pp. 459-468, 1995.

[58] B.Demaya, R. Palm, S. Boverie and A. Titli, "Multilevel Qualitative and Numerical Optimisation of Fuzzy Controller". *Proceedings of FUZZ-IEEE/IFES'95*, vol. 3, pp.1149-1154. Yokohama, Japan. March 1995.

[59] A. Pagni, R. Poluzzi, and G. Rizzotto, "Automatic synthesis, analysis and implementation of a fuzzy controller," *Proc. of 2nd IEEE int. conf. on fuzzy syst.*, vo.1, pp.105-110, 1993.

[60] C. Cavalcante, J. Cardoso, J.J.G. Ramos and O.R. Neves, "Design and Tuning of a Helicopter Fuzzy Controller", *Proceedings of FUZZ-IEEE/IFES'95*, vol.3, pp.1549-1554. Yokohama, Japan. March 1995.

[61] R. Palm, "Robust control by fuzzy sliding mode," *Automatica*, vol. 30, no. 9, pp. 1429-1437, 1994.

[62] F. Bolata and A. Nowè, "From fuzzy linguistic specifications to fuzzy controller using evolution strategies", *Proceedings of FUZZ-IEEE/IFES'95*, vol. 3, pp.1089-1094. Yokohama, Japan. March 1995.

[63] H. T. Nguyen, V. Kreinovich, and D. Tolbert, "On robustness of fuzzy logic," *Second IEEE Int. Conf. on Fuzzy Systems*, vol.2, pp.543-547, 1993.

[64] S. Thompson, "Robust control — an introduction," *Measurement and Control*, vol. 26, pp.235-241, Oct. 1993.

[65] J. Wang and J. R. Jordan, "An experimental investigation of the dynamic performance of fuzzy logic control," *Transactions of the Institute of Measurement and Control*, vol.16, no. 3, pp.142-148, 1994.

[66] J. R. Jordan and J. Wang, "The Design of Fuzzy Logic Controlled SISO Processes with step Response Constraints," *Proceedings of 3rd IFAC/IFIP AARCT'95*, pp. 481-485, Ostend, Belgium, May 1995.

[67] J. Wang and J. R. Jordan, "The robust step performance of PID and fuzzy logic controlled SISO systems," *Proceedings of FUZZ-IEEE / IFES'95*, vol.1, pp.325-330, Yokohama, Japan, Mar. 1995.

[68] F. Hoffmeister and T. Bäck, "Genetic algorithms and evolution strategies: Similarities and differences," *Proc. the First Int. Conf. on Parallel Problem Solving from Nature*, Dortmund, Germany, pp.447-461, 1990.

[69] K. Self, "Self-designing with fuzzy logic,", *IEEE Spectrum*, pp.42-105, November 1990.

[70] Y. F. Li and C.C. Lau, "Development of fuzzy algorithms for servo systems," *IEEE Control Systems Magazine*, pp.65-71, Apr. 1989.

[71] D. K. Kahaner, "Report on 2nd fuzzy logic systems institute: Hardware practice seminar 15 Dec. 1990," Source: kahaner@xroad.cc.u-tokyo.ac.jp, 21 Jan. 1991.

[72] D. K. Kahaner, "Report on fuzzy theory," Source: kahaner@xroad.cc.u-tokyo.ac.jp, 17 May, 1991.

[73] D. K. Kahaner, "Fuzzy logic," Source: kahaner@xroad.cc.u-tokyo.ac.jp, 24 August, 1991.

[74] D. K. Kahaner, "Fuzzy system research in Japan (Schwartz)," Source: kahaner@xroad.cc.u-tokyo.ac.jp, 21 Nov., 1991.

[75] R. M. Tong, "A control engineering review of fuzzy systems," *Automatica*, vol.13, pp.559-569, 1977.

[76] R. M. Tong, "On fuzzy control and fuzzy reviewing," *IEEE Control Systems*, pp. 5-7, June 1993.

[77] K. M. Passino, "Bridging the gap between conventional and intelligent control," *IEEE Control Systems*, pp.12-18, June 1993.

[78] E. H. Mamdani, "Advances in the linguistic synthesis of fuzzy controllers," *Int. J. Man Mach. Studies*, vol. 8, no. 6, pp. 669-678, 1976.

[79] E. H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis," *IEEE Trans. Computer*, vol. C-26, no. 12, pp. 1182-1191, 1977.

[80] J. Maiers and Y. S. Sherif, "Applications of fuzzy set theory," *IEEE Trans. Syst. Man Cybern.*, vol. 15, no. 1, pp. 175-189, 1985.

[81] P. M. Larsen, "Industrial applications of fuzzy logic control," *Int. J. Man Mach. Studies*, vol. 12, no. 1, pp. 3-10, 1980.

[82] D. Ruan, L. V. D. Durpel and P. D'hondt, "Fuzzy engineering in nuclear research application," *Proceedings of FUZZ-IEEE / IFES'95*, vol.1, pp.211-218, Yokohama, Japan, Mar. 1995.

[83] J. A. Bernard, "Use of rule based systems for process control," *IEEE Contr. Syst. Mag.* , vol. 8, no. 5, pp. 3-13, 1988.

[84] J. E. Baher, "Reducing bias and inefficiency in the selection algorithm," *Proc. the Second Int. Conf. on Genetic Algorithm and Their Applications*, Hillsdale, New Jersey, pp.14-21, 1987.

[85] D. Filev and R. R. Yager, "Simplified methods of reasoning in fuzzy models," *Proceedings of FUZZ-IEEE / IFES'95*, vol.1, pp.211-218, Yokohama, Japan, Mar. 1995.

[86] J. Wang, *Adaptive Control of Time-Varying Processes*, MSc. thesis (Chinese), Beijing Agricultural Engineering University, China, 1988.

[87] J. E. Baher, "Adaptive selection methods for genetic algorithms," *Proc. the First Int. Conf. on Genetic Algorithm and Their Applications*, Hillsdale, New Jersey, pp.101-111, 1985.

[88] V. I. Utikin, "Variable structure systems with sliding mode: a survey," *IEEE Transaction of Automatic Control*, vol. 22, pp. 212-222, 1977.

[89] H. Berenji, Y. Y. Chen, C. C. Lee, J. S. Jang, and S. Murugesan, "A hierarchical approach to designing approximate reasoning-based controllers for dynamic physical systems," *Uncertainty in Artificial Intelligence*, vol.6, eds. P. P. Bonissone, *et al.* North-Holland, Amsterdam, 1991.

[90] O. Itoh, H. Migita, Y. Irie and J. Itoh, "Application on fuzzy control to automatic crane operation," *Japanese Journal of Fuzzy Theory and System*, vol.6, no.2, pp.283-296, 1994.

[91] T. Heckenthaler and S. Engell, "Approximately time-optimal fuzzy control of a two-tank system," *IEEE Control Systems*, pp.24-30, June 1994.

[92] O. Moreira-Tamayo, J. Pineda de Gyvez and E. Sanchez-Sinencio, "Filter tuning system using fuzzy logic," *Electronic Letters,* vol.30, no.11, pp.846-847, 26th May 1994.

[93] R.D.Marchbank and M. Jamshidi, "A fuzzy control approach to a two-axis mirror laser beam alignment system," *Eng. Applic. Artif. Intell.*, vol.6, no.2, pp.117-122, 1993.

[94] J.R. layne, K.M. Passino and S. Yurkovich, "Fuzzy learning control for antiskid braking systems," *IEEE Trans. Contr. Syst. Techno.*, vol.1, no.2, pp.122-129, 1993.

[95] R. Meier, J. Nieuwland, A.M. Zbinden and S.S. Hacisalihzade, "Fuzzy logic control of blood pressure during anesthesia," *IEEE control systems*, pp.12-17, Dec. 1992.

[96] D.A. Linkens, "Adaptive and intelligent control in anesthesia," *IEEE Control Systems*, pp.6-11, Dec.1992.

[97] D.A. Linkens and S.B. Hasnain, "Self-organising fuzzy logic control and application to muscle relaxant anesthesia," *IEE Proceedings-D,* vol.138, no.3, pp.274-284, 1991.

[98] J. Wang and Y. Sun, "Microcomputer embedded adaptive controller for incubators", *Proc. International Symposium on Agricultural Engineering*, pp821-824, Beijing, Sep. 1989.

[99] M. A. Lee and H. Takagi, "Integrating design stages of fuzzy systems using genetic algorithms," *Proc. IEEE Int. Conf. Fuzzy Syst.* (FUZZ-IEEE'93), pp.236-245, 1993.

[100] L.J. Huang and M. Tomizuka, "A self-paced fuzzy tracking controller for two-dimensional motion control," *IEEE Trans. Syst., Man, Cybern.*, vol.20, no.5, pp.1115-1124, 1990.

[101] G. C. Hwang and S. C. Li, "A stability approach to fuzzy control design for non-linear systems", *Fuzzy Sets and Systems*, vol. 48, pp.279-287, North-Holland, Amsterdam, 1992.

[102] S. Kawaji and N. Matsunaga, "Fuzzy control of VSS type and its robustness", *Preprints of IFSA'91*, Brussels, Engineering volume, pp.81-88, July 1991.

[103] J. C. Wu and T. S. Liu, "A sliding-mode approach to fuzzy control design", *IEEE Trans. Control Sys. Tech.*, vol. 4, no. 2, pp. 141-150, 1996.

[104] H. Ying, "Practical design of nonlinear fuzzy controllers with stability analysis for regulation process with unknown mathematical models," *Automatica*, vol. 30, no. 7, pp. 1185-1195, 1994.

[105] H. Kiendl and J. J. Rueger, "Stability analysis of fuzzy control systems using facet functions," *Fuzzy Sets and Systems*, no. 70, pp. 275 - 285, 1995.

[106] K. Tanaka and M. Sugeno, "Stability analysis and design of fuzzy control systems," *Fuzzy Sets and Systems*, no. 45, pp. 135 - 156, 1992.

[107] G. Bretthauer and H. P. Opitz, "Stability of fuzzy systems - A survey," *Proc. EUFIT'94*, Aachen, pp. 283 - 290; 1994

[108] C. Karr, "Applying genetics to fuzzy controller," *AI Expert*, vol.6, no.2, pp.26-33, 1991.

[109] C. Karr, "Design of an adaptive fuzzy logic controller using a genetic algorithm," *Proc. 4th Int. Conf. of Genetic Algorithms*, San Diego, USA, pp.450-457, July, 1991.

[110] J. H. Holland, *Adaptation in Natural and artificial systems*. Ann Arbor, MI. University of Michigan, 1975.

[111] P. Thrift, "Fuzzy logic synthesis with genetic algorithms," *Proc. 4th Int. Conf. of Genetic Algorithms*, San Diego, USA, pp.509-513, July, 1991.

[112] H. Nomura, I. Hayashi and N. Wakami, "A self-tuning method of fuzzy reasoning by genetic algorithm," *Proc. Int. Conf. Fuzzy Systems and Intelligent Control*, Louisville, KY, pp.236-245, 1992.

[113] Y. Qian, P. Tessier and G. Dumont, "Fuzzy logic based modeling and optimization," *Proc. 2nd Int. Conf. Fuzzy Logic and Neural Networks*, (IIZUKA'92), pp.349-352, 1992.

[114] K. Kiriakidis *and* A. Tzes, "Application of implicit self tuning fuzzy control to non-linear systems," *Proceedings of FUZZ-IEEE/IFES'95*, vol. 3, pp.1419-1426. Yokohama, Japan. March 1995.

[115] J. Xu, C. Lui and C. Hang, "Modified tuning of a fuzzy logic controller," *Engng Applic. Artif. Intell.* Vol. 9, No.1, pp.65-74, 1996.

[116] G. Chen and H. Ying, "Stability analysis of nonlinear fuzzy PI control systems," *Proc. 3rd Int. Conf. on Fuzzy Logic Application,* pp.128-133, 1993.

[117] S.J. Qin and G. Borders, "A multi-region fuzzy logic controller for nonlinear process control," *IEEE Tran. Fuzzy Syst.* February, pp.74-81, 1994,

[118] J. X. Xu, W.K. Ho, C. Lui and C.C. Hang, "Tuning of fuzzy PI controllers based on gain and phase margin specifications and ITAE index," *Proc. IEEE SICICI'95*, pp.198-203, 1995.

[119] Motorola, M68HC11 *Reference Manual*, Rev 3, 1991.

[120] S. Boverie, B. Demaya, J. Lequellec and A. Titli, "Fuzzy logic control for high order systems", *Proc. 2nd IEEE International Conference on Fuzzy Systems*, vol. 1, pp.117-121, 1993.

[121] S. Geva and J. Sitte, "A cartpole experiment benchmark for trainable controllers", *IEEE Control Systems*, pp.40-51, Oct. 1993.

APPENDIX A

Design of the simulation system

The research work reported in this thesis is mainly based on computer simulation techniques. A simulation package, called *FzySimu*, has been developed in the C++ language. It contains the following functions: performance simulation (*MultiSimu*), performance analysis (*Analysis*), robustness test (*RobuMap*), robustness optimisation (*GaOpt*), auto-tuning (*AutoTune*) and real-time control interface (*RCPlot*). *Chapter* 5 has already presented a very detailed description of the *AutoTune* function and it will not be discussed again. In this appendix, the methods used for the implementation of the other five functions will be presented, and the features and the usage of these functions will be described.

A.1 Performance simulation

The performance simulation function is the key function of the whole package. It performs the following operations: process simulation, controller simulation, system configuration, performance evaluation and result output. System configuration operation and result output operation provide a user interface for the interaction between the program and the user. The other three operations perform calculations of the entire system states and the measurements of the system performance. They form a stand-alone function that can be called by other functions, such as the robustness test function and optimisation function, to provide an evaluation of the control system under investigation.

It should be noted that since the fuzzy logic controller is generally implemented by software techniques to carry out the complicated computations, the system is simulated in the discrete format. A digital sampler with period of t_s is added before the controller to obtain the system error information.

A.1.1 Process simulation

There are three types of processes currently implemented in the simulation package. They are as shown in the following.

Model I: Linear process with time delay and simple non-linear terms

$$G_1(s) = \frac{K\, G_n\, \omega_n^2\, (1+T_3\ s)(1+T_4\ s)\, e^{-\tau s}}{s^n (1+T_1\ s)(1+T_2\ s)(s^2+2\zeta\omega_n s+\omega_n^2)}; \qquad \text{(A-1)}$$

where K is the constant gain, n = 0 or 1 is the number of integrators in the process, T_1 and T_2 are the time constant related to process poles, T_3 and T_4 are the time constant related to process zeros, ω and ζ are the natural frequency and the damping ratio, respectively, of the second order term in the process, τ is the time delay in the process, G_n is a non-linear gain with dead-zone and it is defined as follows;

$$G_n(x) = \begin{cases} 1, & \text{if } x > \text{D}; \\ 0, & \text{if } |x| \le \text{D}; \\ \lambda x & \text{if } x < \text{-D}; \end{cases} \qquad \text{(A-2)}$$

λ is the process gain when the input is negative, D > 0 is the dead-zone.

- Model II: Cart-pole process

$$\left.\begin{aligned} m_c\ddot{p} + m_p\ddot{p} + m_p l\ddot{\theta}\cos\theta - m_p l\dot{\theta}^2\sin\theta &= Ku \\ J\ddot{\theta} + m_p l(\ddot{p}\cos\theta + l\ddot{\theta}) - m_p gl\sin\theta &= 0 \end{aligned}\right\}, \qquad \text{(A-3)}$$

where m_c is the mass of the cart, p is the horizontal displacement of the cart, m_p is the mass of the pole, l is the length of the pole, θ is the pole's angle to the vertical, u is the control force exerted on the cart, $J = \frac{1}{3}m_p l$ is the moment inertia of the pole, and g = 9.8 m/s^2 is the acceleration due to gravity, K is the gain of the control input.

- Model III: Non-linear process

$$T\dot{y} = \frac{1-e^{-y}}{1+e^{-y}} + Kx; \qquad \text{(A-4)}$$

where x is the input, y is the output, T is the time constant, and K is the process gain.

To simulate the Model I process, the following basic functions were separately implemented.

- Zero $(1+T_z s)$: its output at instance $(k + 1)$ is calculated as

$$y_z(k+1) = (1+T_z)x_z(k) + T_z x_z(k-1), \tag{A-5}$$

  where $x_z(k)$ and $y_z(k)$ are the input and output of the zero unit respectively. Note that the simulation period is assumes as 1 in all process simulations.

- Pole $\frac{1}{(1+T_p s)}$: its output at instance $(k + 1)$ is calculated as

$$y_p(1+k) = y_p(k)\cdot e^{-1/T_p} + x_p(k)\cdot(1-e^{-1/T_p}), \tag{A-6}$$

  where $x_p(k)$ and $y_p(k)$ are the input and output of the pole unit respectively.

- Complex poles $\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$: its output at instance $(k + 1)$ is simulated as:

$$\dot{y}_c(k) = \dot{y}_c(k-1)\cdot e^{-1/T_{c1}} + \{x_p(k) - y_c(k)\}\cdot(1-e^{-1/T_{c1}}) \tag{A-7}$$

$$y_c(k+1) = \sum_1^k \frac{\dot{y}_c(k)}{T_{c2}} \tag{A-8}$$

  where $T_{c1} = \frac{1}{2\zeta\omega_n}$, $T_{c2} = \frac{2\zeta}{\omega_n}$, $x_c(k)$ and $y_c(k)$ are the input and output of the complex pole term, respectively.

- Time delay $e^{-\tau s}$: this term is realised by using the memory loop with the loop length equal to the time delay τ. Every input values to this term is saved into this memory loop and popped out after τ simulation periods.

- Integrator $\frac{1}{s}$: it is implemented as an accumulator and is assumed that the integrating time constant is 1.

By calling these basic functions, the dynamic performance of the Model I process can be obtained. Also, Model I process can be set to be one time lag process, two time lag process, a standard second order process, or any combination of the basic units.

The sampling period for the process simulation is defined as the basic unit of time constants in the process and the smallest time constant in the process should be at least 10 times larger than the basic time unit so as to obtain the approximate dynamic performance of the simulated process. In addition, the saturation constraints can be selected to applied to each function to approximate the practical plant.

A.1.2 Controller simulation

There are eight types of controllers implemented in the simulation package. They are PID, FLC, first phase advanced FLC (PAM-I), second phase advanced FLC (PAM-II), self-organised FLC (SOFLC), predictive FLC, multiple input FLC, PID paralleled with FLC. The last four controllers are not related to this research work, thus they will not be described here.

A. FLC

A two-input-one-output FLC was implemented in the simulation. Two inputs are error, $e(k)$, and change of error, $\dot{e}(k)$. The output variable is the control output, $u(k)$. The input output variables were non-fuzzy (crisp) variable. The FLC converted the crisp input variable into fuzzy variables, performed a fuzzy reasoning based on the fuzzy rules and provided a suitable control output.

There were four fundamental operations in the FLC: scale mapping, fuzzification, fuzzy reasoning and defuzzification. In the scale mapping, the proportional mapping was used in the simulation. Fuzzification transfers a crisp variable to a fuzzy variable based on the defined membership functions of this fuzzy variable, and the defuzzification performs the opposite conversion by using the COG method. A membership function $\mu(x)$, as shown in Fig.A-1, is represented by four parameters (x_0, x_1, x_2, x_3) in the simulation. Each fuzzy variable of the FLC was defined by seven fuzzy sets in the universe of [-10, 10].

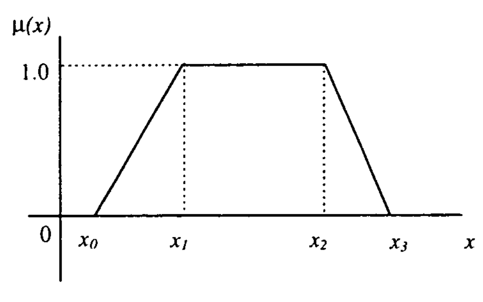


Fig.A-1 Definition of membership function

In the fuzzy reasoning, seven fuzzy labels, NB, NM, NS, ZE, PS, PM and PB, were represented as 0, 1, ..., 6, respectively. Seven membership grades in each fuzzy input variable formed a data group, denoted as $\mu_i(e)$ and $\mu_i(\dot{e})$ (i = 0, 1,..., 6). With these notations, the fuzzy rule base can be represented by a data matrix, with the fuzzy labels of e as the column index and fuzzy labels of $\dot{e}$ as the raw index. Two most commonly used fuzzy reasoning methods, the Max-Min and Max-product, were implemented in the simulation.

B. PID

The practical output of the PID controller was calculated as

$$u(k+1) = K_p e(k) + K_i \sum_{i=1}^{k} e(i) + K_d \dot{e}(k) \tag{A-9}$$

where K_p, K_i and K_d are the proportional gain, the integral gain, the derivative gain of the controller, respectively.

C. PAM-I

The practical output of the PAM-I controller was calculated as

$$u(k+1) = GU \sum_{i=1}^{k} \Delta u^*(i) + \alpha \cdot \Delta u^*(k) \tag{A-10}$$

where $\Delta u^*(i)$ is the normalised output change from the FLC at instance i, α is the weighting factor in the proportion path of the PAM-I.

D. PAM-II

The practical output of the PAM-II controller was calculated as

$$u(k+1) = f_{flc}(\ e(k),\ \dot{e}(k) + \beta\ \ddot{e}(k)\ ) \tag{A-11}$$

where $f_{flc}(\bullet)$ denotes the FLC function, β is the weighing factor of $\ddot{e}(k)$ in the PAM-II.

A.1.3 Performance measurements

The main objective of the simulation function is to measure the dynamic performance of the simulated system. As presented in *Chapter* 3, five parameters, OS, US, ST, CY and IES, were used as the measurements of a system performance. Based on the definitions of these measurements, a step response of a control system was simulated and the parameters measured.

To obtain the performance measurements, process output $y(k)$ and its change rate $\dot{y}(k)$ are checked at each simulation step. If $y(k)$ is increasing compared with $y(k\text{-}1)$, the maximum output, y_{max}, is tested. If $y(k)$ is decreasing compared with $y(k\text{-}1)$, then the minimum process output, y_{min}, is tested. If the sign of $\dot{y}(k)$ changes compared with $\dot{y}(k\text{-}1)$, then the ring number m increases 1. If $y(k)$ is not in the steady state zone (defined as from 98% to 102% of the steady state output $y(\infty)$), then the settle time t_{st} is set as the sample step k. For IES, the $e^2(k)$ is simply accumulated to E_{sum} during the simulation period. At the end of the simulation period, five performance measurements are calculated as follows;

$$OS = \frac{y_{\max} - w}{w} \times 100\%; \qquad US = \frac{w - y_{\min}}{w} \times 100\%;$$

$$CY = \frac{m}{2}; \qquad ST = \frac{t_{st}}{T_{sum}};$$

$$IES = \frac{E_{sum}}{w^2 T_l};$$

where w is the setpoint change, T_{sum} is the sum of all time constants, time delay and/or equivalent time constants in the controlled process, and T_l is the total number of simulation steps in the simulation period. Note that OS and US will be set to 0 if they are negative.

A.1.4 System configuration

To perform a simulation of a control system, the simulation programs have to be configured first, which includes selecting the controlled process, the controller, and other system parameters such as the length of simulation period, the type and amplitude of the reference signal. The simulation software provides a user interface for the system configuration, as shown in Fig.A-2.

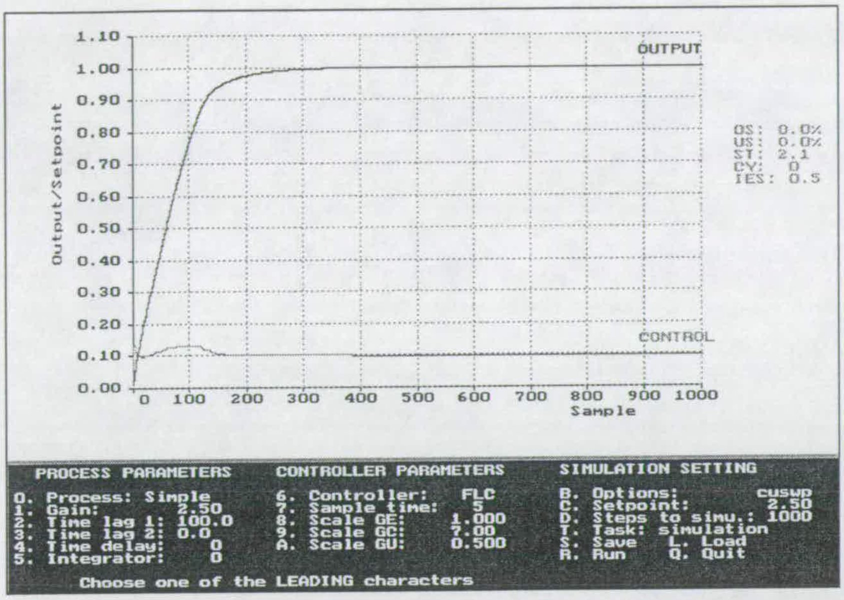


Fig.A-2 A screen capture of the user interface provided by the simulation program.

The screen is divided into two parts (two windows in the Xwin under UNIX). The upper part is the output area that is used to plot the process control input *u* and output *y* and to show the system performance measurements. The lower part is for system configuration. There are three columns in the configuration area, indicating the process parameters, the controller parameters and simulation parameters, respectively. Each column contains several items. And each item indicates one or more parameter values, or an operation. The first letter, called a command code, in each item is used to modify the parameter value(s) or staring an operation. When one of the command codes related to parameters is entered, an explanation of the command code will be given in the bottom line and new parameter values can be entered. When one of the command codes related to an operation is entered, the operation will be performed.

The "TASK" item is used to assign a job to the simulation. The jobs are system simulation, control surface, switching line, phase trajectory, robustness test (interactive mode), optimisation by genetic algorithm (interactive mode) and real-time control interface. System simulation is the topic of this section, and other jobs will be explained in other sections.

The "Option" item in the simulation column provides the following options for performing system simulation.

(1) Reference options:

u -- use step input as system reference;

r -- use single step pulse as the system reference;

q -- use square wave as the system reference;

t -- use ramp signal as the system reference.

(2) Controller options:

s -- apply saturation constraint to controller signals;

8 -- apply 8-bits resolution to controller input/output signals;

c -- normalise controller input/output signals to system setpoint;

w -- use DOF of each fuzzy rule as the weight in the COG operation (FLC);

m -- use Max_Product reasoning method if set, otherwise use Max-Min method.

(3) System options:

p -- plot system response in the normalised format, i.e. [-1, 1];

d -- apply load disturbance in the simulation;

n -- add random noise to system output signal;

U -- add ring number of the process control to the CY measurement.

The "Setpoint" item indicates the maximum change in the system reference. The "SAVE" command saves all system parameters into a file in a defined order and format. The "LOAD" command is used to initial the simulated system by loading the system parameters from a parameter file in the required format. The "RUN" command starts the simulation to perform the required task.

The required format of the parameter file is illustrated by an example as shown in Fig.A-3. The first line of the parameter file contains the controller type, process type, system options and the number of controller parameters. The second line and the third line list the process parameter settings at the tuning point in the following order;

Model I: *K, Td, T1, T2, T3, T4*, ω_n , ζ, *n*, dead-zone *D*, λ;

Model II: *K*, m_c, m_p, *l*, the maximum force F_{max} applied to the cart, and the initial pole angle $\theta(0)$;

Model III: *K, T*.

The controller parameters are listed in the last part of the file. Note that the data format of the membership functions in the file is different from the definition of the membership functions in Fig.A-1. The format used to represent a membership function in the initial parameter file (INI-file) is (x_0, D_1, D_2, D_3) where $D_1 = x_1 - x > 0$, $D_2 = x_2 - x_1 > 0$, $D_3 = x_3 - x_2 > 0$ (see Fig.4-1). Using this format is only for the convenience of calculation in the simulation. These controller parameters are saved in the following order;

- if the controller is FLC:

  GE, GC, GU, t_s

  membership functions of error (28 parameters),

  membership functions of $\dot{e}$ (28 parameters),

  membership functions of Δu (28 parameters), rule matrix (49 parameters);

```
0 cuswp f 137
TP: K=2.0 Td=0 T1=0.0 T2=0.0 T3=-40.0 T4=0.0
  Wn=3.0 Dr=0.7 n=1 Dz=0.0 Gr=1.0
1.0 12.0 0.025  12

-10.0 0.0 4.0 3.4
-6.0 3.4 0.0 2.1
-2.6 2.1 0.0 0.5
-0.5 0.5 0.0 0.5
0.0 0.5 0.0 1.4
0.5 1.4 0.0 3.4
2.6 3.4 4.0 0.0

-10.0 0.0 3.0 2.0
-7.0 2.0 0.0 4.2
-5.0 4.2 0.0 0.8
-0.8 0.8 0.0 0.8
0.0 0.8 0.0 4.2
0.8 4.2 0.0 2.0
5.0 2.0 3.0 0.0

-10.0 5.0 0.0 5.0
-1.9 0.9 0.0 0.9
-0.2 0.1 0.0 0.1
-0.1 0.1 0.0 0.1
0.0 0.1 0.0 0.1
0.1 0.9 0.0 0.9
0.0 5.0 0.0 5.0

0 0 0 0 1 2 4
0 0 0 1 2 3 4
0 0 1 2 3 4 5
0 1 2 3 4 5 6
1 2 3 4 5 6 6
2 3 4 5 6 6 6
2 4 5 6 6 6 6
```

Fig.A-3 An example of the parameter file

- if the controller is PAM-I:
  GE, GC, GU, t_s, α
  membership functions of error (28 parameters),
  membership functions of $\dot{e}$ (28 parameters),
  membership functions of Δu (28 parameters),
  rule matrix (49 parameters);
- if the controller is PAM-II:
  GE, GC, GU, t_s, β
  membership functions of error (28 parameters),
  membership functions of $\dot{e}$ (28 parameters),
  membership functions of Δu (28 parameters),
  rule matrix (49 parameters);
- if the controller is PID: K_p, K_i, K_d, t_s.

Clearly the simulation program provides convenient methods for the system configuration. The control system can be configured manually or by loading in the parameter file. But the membership functions and the rule base can only be initialised by the parameter file.

A.1.5 Simulation and output

Fig.A-4 shows the block diagram of the simulation program. It consists of two parts: the initialisation and the simulation. In the first part, the controller and the process are selected, and their parameters are initialised. The second part calculates system states at every sampled point based on the controller's algorithm and the process dynamics. The simulation is in a loop structure with one loop equals to a simulation period. The performance measurements are also determined at each sampling point. Note that the blocks in the dotted square are an independent function which can be called from other package functions, such as the function for the robustness test.

Simulation results were both displayed on the monitor and saved to a data file. In the DOS version, graphic user interface (GUI) was designed and system dynamic performance was plotted on the screen, see Fig.A-3. In UNIX version, system dynamic performance was displayed in a *gnuplot* window which was connected to the simulation program by the 'pipe' function, see Fig.A-5. The information displayed on the screen and saved in the data file depends on the 'TASK' selection. The data file is mainly used to create figures in the Microsoft word document by Microsoft Excel.

A.2 System analysis

In addition to the above simulation function, *FzySimu* provides three system analysis functions for fuzzy logic control systems. These functions are the control surface function, the switching line function and the phase trajectory function. These functions can be selected from the "TASK" item in the simulation parameter column as shown in Fig.A-2.

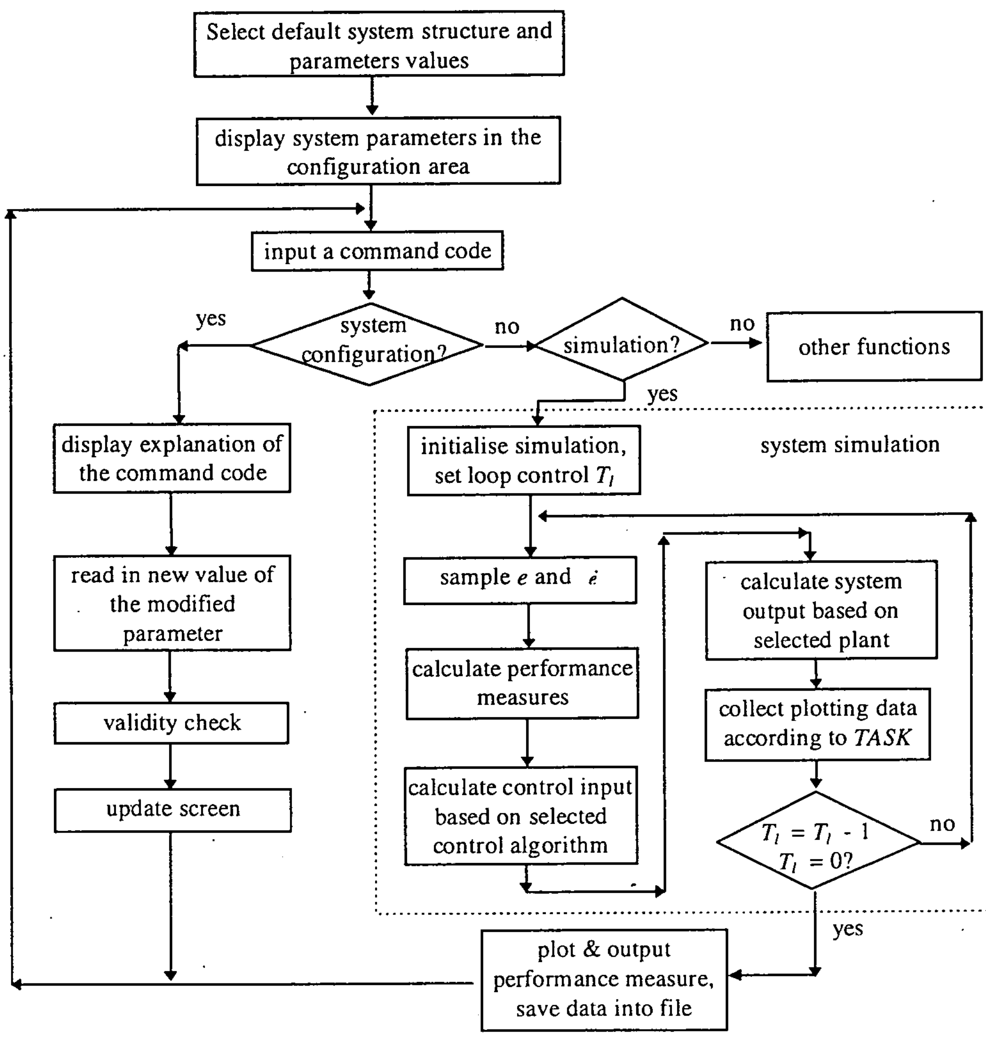


Fig.A-4 Block diagram of the simulation program

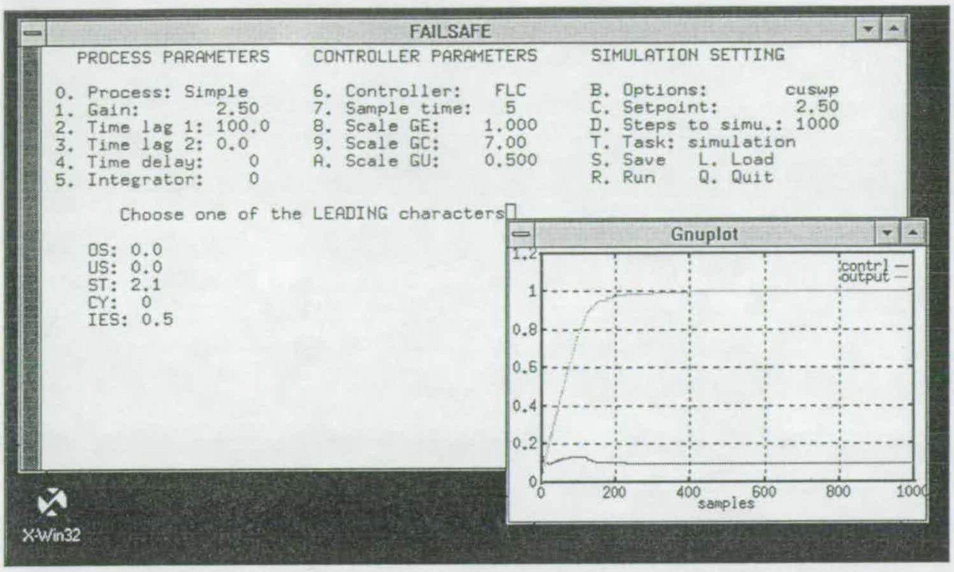


Fig.A-5 Screen capture of the performance simulation function in UNIX (Xwin).

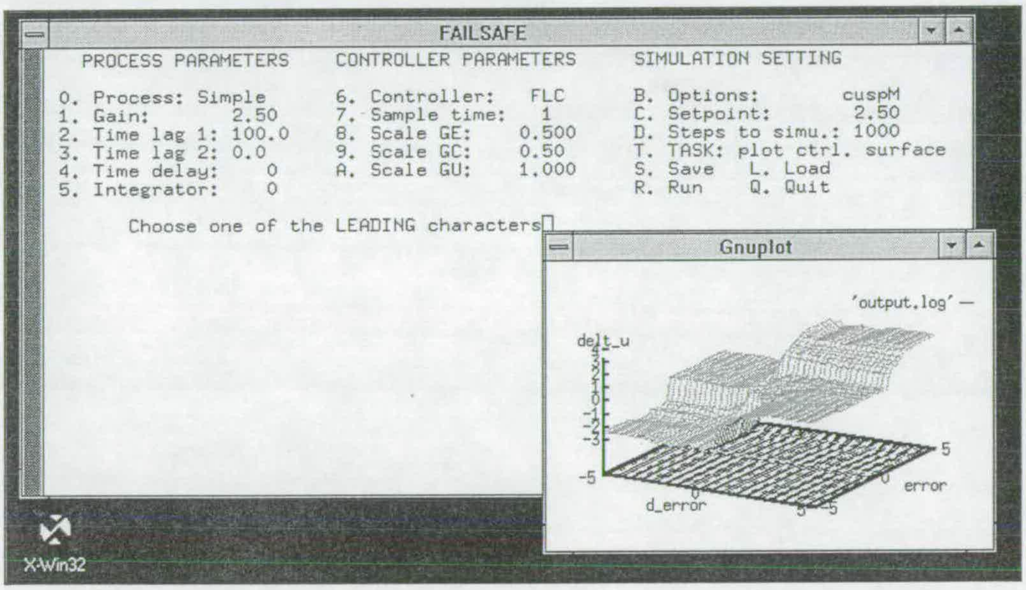


Fig.A-6 A screen capture of the control surface function (UNIX)

In the control surface function, the output changes, Δu, at all points $(e, \dot{e})$ in the normalised input space of the FLC are calculated and a three dimensional control surface $e \times \dot{e} \times \Delta u$ is plotted. A screen capture of the control surface function is shown in Fig.A-6. The control surface provides a graphical description of the transfer function of the simulated FLC, and it can be used to analyse the controller performance at different system states.

In the switch line function, the system states $(e(k), \dot{e}(k))$ where the control change Δu is zero are searched in the normalised input space of the FLC. The switching line formed by all these points are plotted in the $e \times \dot{e}$ plane. A screen capture of the switch line function is shown in Fig.A-7. The switching line function can be used to study the dynamic performance of an FLC system since the switching line of the FLC determines the dynamic performance of the control system.

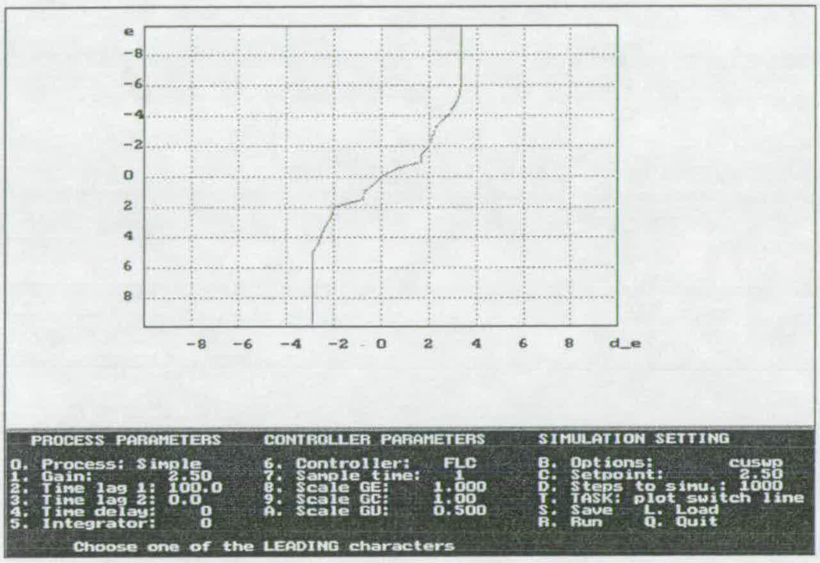


Fig.A-7 A screen capture of the switch line function

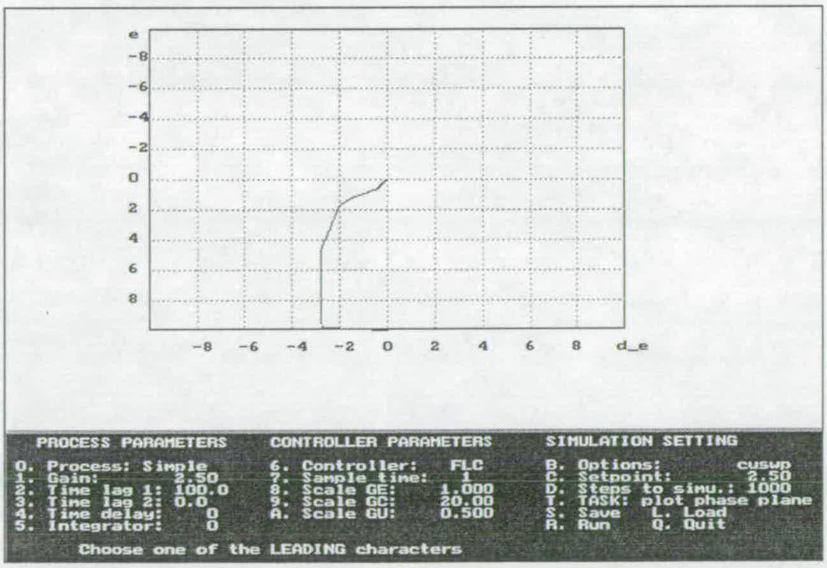


Fig.A-8 A screen capture of the phase trajectory function

As indicated by its name, the phase trajectory function provides a tool to plot the phase trajectory of a control system on the phase plane. Fig.A-8 shows a sampled screen of this function. The phase trajectory can be used to study the switching line characteristic of the fuzzy control system and design the FLC parameters to obtain the required system performance.

These analysis functions can be directly accessed from the simulation user interface as shown in Fig.A-2. The block diagram of these functions is shown in Fig.A-9. The phase trajectory function is similar to the simulation function, except for the different output information. In the control surface and switching line functions, the control output change Δu is calculated at 1600 points evenly distributed in the input space, $e \times \dot{e}$. The 1600 data groups, ($e(k)$, $\dot{e}(k)$, Δu(k)), are used to produce the control surface and the points where Δu(k) = 0 are used to plot the switching line.

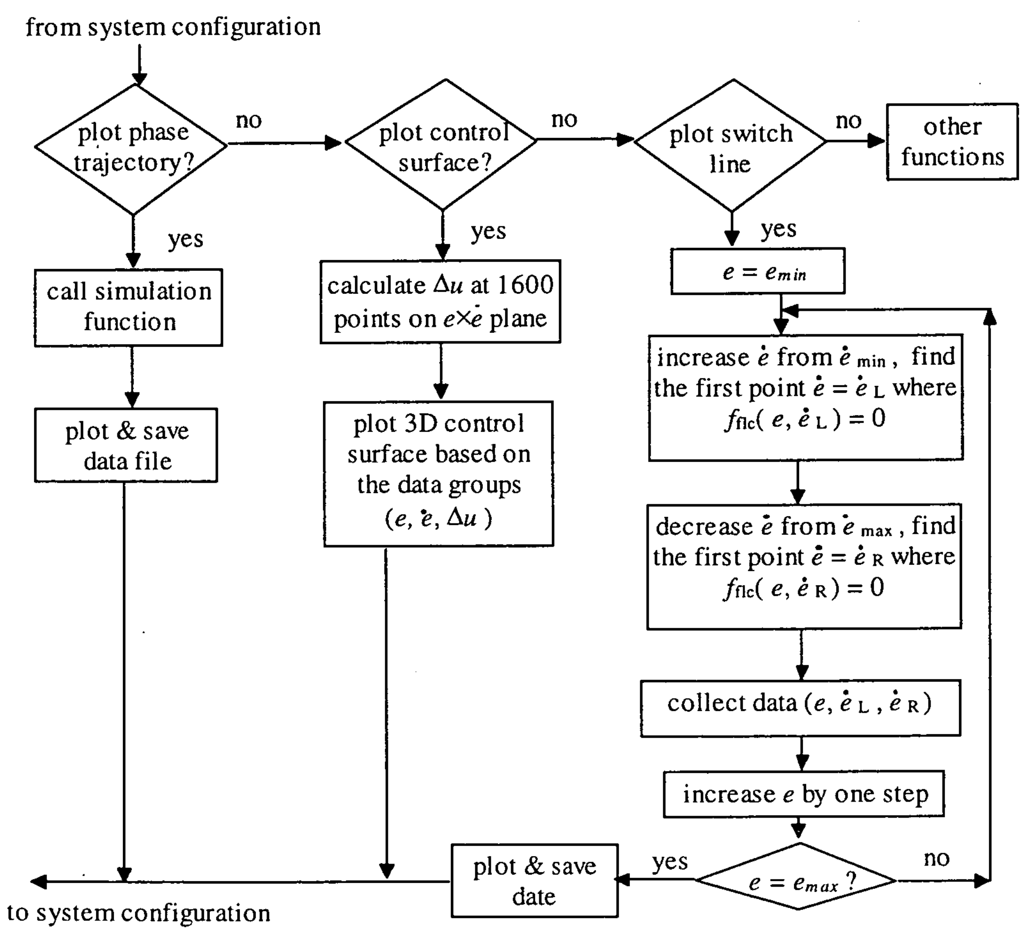


Fig.A-9 Block diagram of the analysis functions

A.3 Robustness test (RBT)

Investigation of the robustness of the fuzzy control system is the main task of this research. The simulation package *FzySimu* provides an experimental tool for pursuing this study. Two investigation areas have been emphasised in the *FzySimu*: robust space detection and sensitivity test. In addition, *FzySimu* provides two working modes: the interactive mode and the quiet mode. All these are discussed bellow.

A.3.1 Robust space detecting (RSD)

Given a control system, the process parameter variation range Φ and the required performance specifications, the RSD function searches for the parameter space ϕ ($\Phi \in \phi$) within which the system performance meets the performance specification. To obtain the information about the system performance in different process parameter settings, test points are selected in the parameter space and simulation of the system step response at each test point is performed.

The simulation package *FzySimu* provides a function to detect the robust space of the system in the parameter space determined by three parameter variation ranges $[x_1, x_2]\times[y_1, y_2]\times[z_1, z_2]$, where x, y and z are process parameters. Step length, Δx, Δy, and Δz are used to select test points in each parameter range. The detected robust space is plotted on the screen as x-y plane plots at each test point of parameter z, called *slices*. The results are also saved into a file suitable for MS-*Exel* to re-create these slices or plot a three dimensional figure.

Two searching methods have been used in the RSD function: full searching and edge searching. The full searching method measures the performance of a control system at all test points in the process parameter space, and determines the robust space by checking if the system performance meets the required specifications. This method is suitable for systems with multiple separated patches in robust spaces in the parameter space. In the edge searching method, the function starts with the full searching method; after one point on the edge of the robust space has been detected, the function will test the system performance only at those test points around the robust space edge, and thus the position of the edge of the robust space can be discovered. It should be noted that the edge searching method is faster than the full searching method, but it is not suitable for systems with multiple separated robust spaces in the parameter space. Fig.A-10 shows the block diagram of the RSD function.

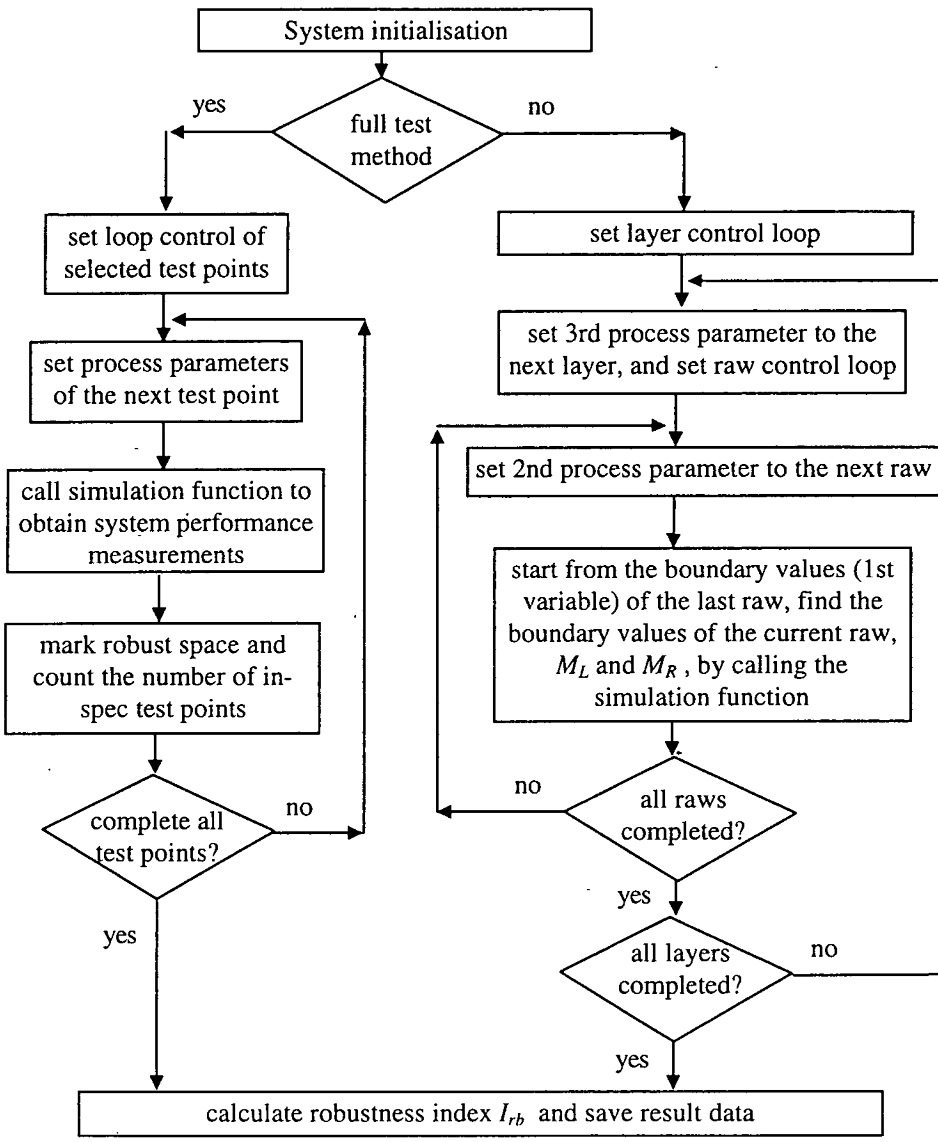


Fig.A-10 Block diagram of RSD function

A.3.2 Sensitivity test (SST)

To design or tune a robust controller, it is necessary to know the effect of controller parameters to system robustness. Controller parameters may play different roles for obtaining a good robust performance. The sensitivity test function in the simulation package

FzySimu provides a tool to investigate the effect of four tuning parameters, GE, GC, GU and t_s, to the robustness of fuzzy control systems. Moreover, the effect of these parameters to system performance can also be tested by the SST function.

To test the effect of a parameter to system robustness, the variation range and the variation step of this parameter are to be selected. For each value of this parameter, the RSD function is called to obtain the size of the robust space within the interested process parameter space. The relation between the parameter and the system robustness is then represented by the data pairs of the tested parameter value and the size of the robust space. Similarly, to test the effect of a parameter on system performance, the variation range and the variation step of this parameter was selected. For each value of this parameter, the system simulation function is called to obtain five performance measurements defined in *Chapter 3.* Then the effect of the tested parameter on system performance can be indicated by the data pairs of the tested parameter value and the performance measurements. Fig.A-11 shows the block diagram of the SST function.

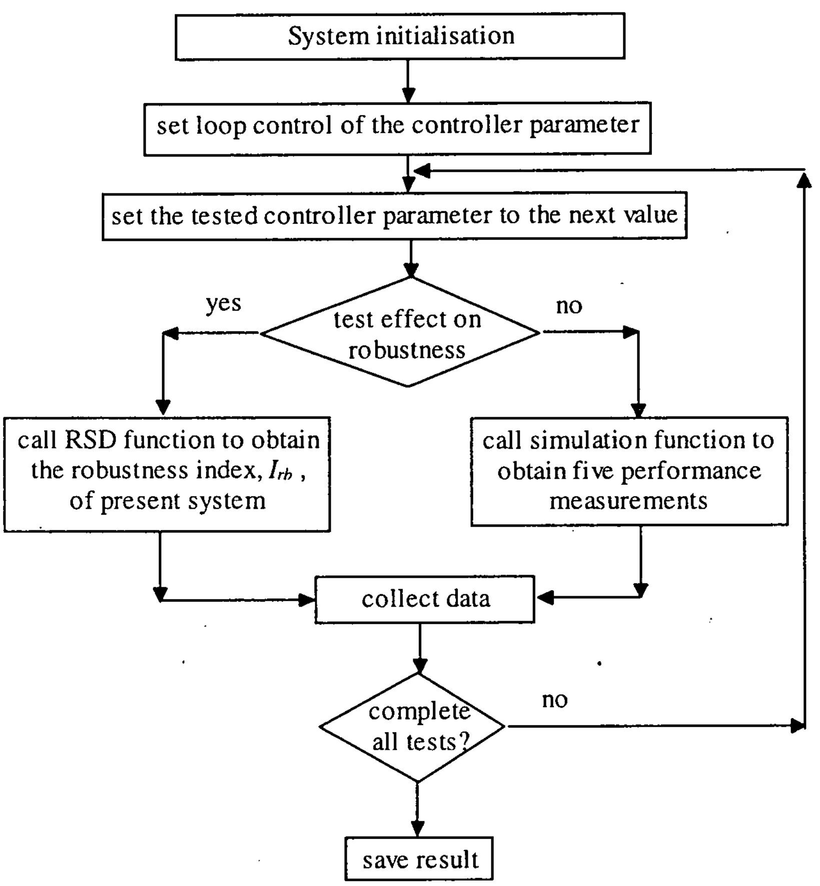


Fig.A-11 Block diagram of SST function

A.3.3 Working modes

FzySimu provides two working modes, interactive mode and quiet mode, for the robustness test function.

A. Interactive mode

When the robustness test function is working in the interactive mode, all system parameters and simulation settings are displayed on the screen, system structure and parameters can be manually initialised, and detected robust space is plotted on the screen. The screen is divided into three parts: system initialisation area in the bottom, RBT initialisation area in the top and the plotting area in the middle. All initialisation items are similarly initialised as in the simulation function. The plotting area shows all slices of robust space at different test points of the third parameter. A screen capture of the interactive mode of the robust space detecting function is shown in Fig.A-12.

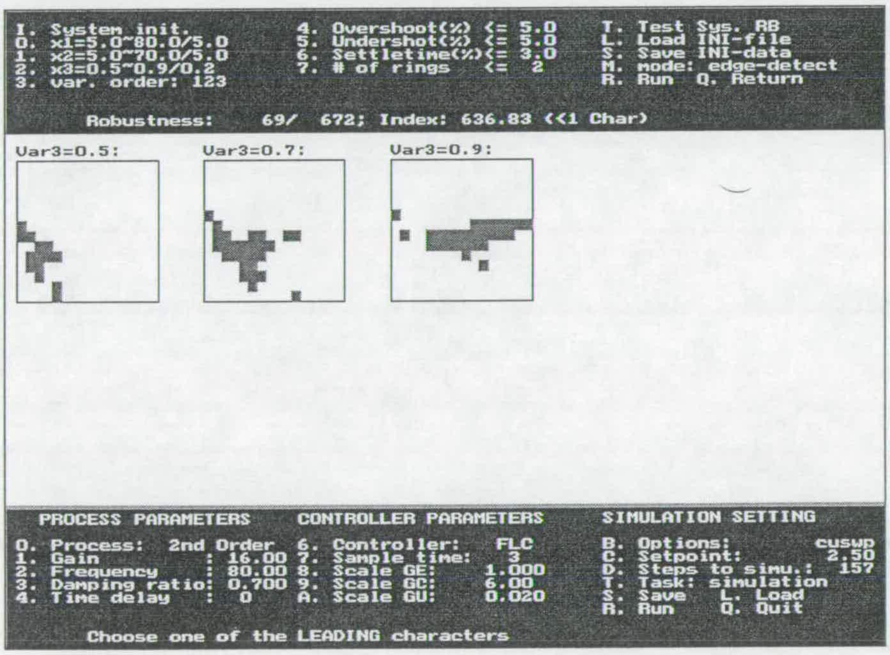


Fig.A-12 A screen capture of the interactive working mode of the RBT functions

Items in the system initialisation area are the same as those in the simulation function. Items in the RBT initialisation area include the variable parameters which the system robustness is considered with, their variation ranges, their step changes in the detecting tests, the required performance specifications, the function task, and the file name for saving the results. More explanations on the items are given bellow.

The variable parameters of the process are indicated by their order numbers in the list shown in the system initialisation area. Take Fig.A-12 as an example. Item 0 in the RBT initialisation area indicates 'variable order: 123', which means that the variation of the first three parameters in the parameter list will be considered in the system robustness test. Since the second order process with time delay is selected as the controlled process as shown in the system initialisation area and its parameters are listed in the order of K, ω_n, ζ and T_d, so the variable parameters are K, ω_n and ζ. If the variable order is 124, then the variable parameters become K, ω_n and T_d. Note that the 'variable order' also indicates the format for plotting the robust space. Because the variable parameters are assumed in the order of *x-y-z*, the slices of the robust space will be plotted in the plane determined by the first two parameters.

The variation ranges and varying step lengths of three variable parameters are indicated in item 1 to 3 as "From/To/Step".

The TASK item lists the following 9 sub-functions and anyone can be selected as the present task;

1) RSD;
2) GE effect to system robustness;
3) GC effect to system robustness;
4) GU effect to system robustness;
5) t_s effect to system robustness;
6) GE effect to performance measurements;
7) GC effect to performance measurements;
8) GU effect to performance measurements;
9) t_s effect to performance measurements.

The interactive mode provides a tool for a manual tuning of process parameters and direct view of the detected robust space. It is suitable for a preliminary test in which the number of test points is small, thus the test is fast.

B. Quiet working mode

From the above description of the RBT function, it can be seen that if the number of test points is big, the RBT function will be very time consuming because it needs to call the simulation function to evaluate the system performance at each test point. Sometime it takes almost an hour to view the robust space of a slow dynamic system if the program runs in MS-DOS. One method used in solving this problem is to develop the fast detecting method — the edge detecting method. The another method is to prepare the operation results of the program in the off-work hours so that the time for waiting for the results can be greatly reduced. The main objective of the quiet working mode is to allow the RBT function to be automatically run in the off-working hour on the UNIX system by using the *crontab* function.

In the quiet working mode, no system parameter is displayed on the screen, except for the progress information. The progress information in the robustness test function is a percentage of the explored parameter space in the required parameter space. System parameters and all other configurations are done by using the initialisation file (INI-file) and the input file (IN-file). The INI-file uses the same format of the initialisation file used in the simulation function as described in *Section* A.1.4. The IN-file has the format as illustrated in Fig.A-13, and it can be created by another program, called *STUP*, which runs separately to let the user to configure the RBT function.

When the RBT function is executed in the quiet mode, it will firstly read these initial files to configure the simulation system and the program, secondly it determines the task to perform according to the "Mapping Task" in the IN-file, then it calls the appropriate sub-function to create the result data file.

A.4 Robustness optimisation

Because of the multiple parameter and non-linear characteristics of the fuzzy logic controller, it is very difficult to design an FLC to provide an optimised robust performance. In this research, the genetic algorithm (GA) is used to optimise the robustness of the fuzzy control system by selecting the controller parameter settings. The function, called *GaOpt*, in *FzySimu* package provides this optimisation.

```
               TASK = m
         Controller = f
         Plant Type = 3
    Plant variables = 178
 1st variable from = 1.000
                to  = 7.000
               step = 0.100
 2nd variable from = 1.000
                to  = 20.000
               step = 0.100
 3rd variable from = 0.700
                to  = 0.701
               step = 0.200
           Setpoint = 2.5
     OvershootSpec = 5
    UndershootSpec = 5
    SettletimeSpec = 3
  Ring NumberSpec = 2
       Mapping Task = 5
```

Fig.A-13 An example of IN-file for RBT functions

Since the objective of the optimisation is to obtain the maximum robust space in the interested process parameter space, system robustness is tested in each group of controller parameter settings (called a *gene*) by calling the RSD function. In addition, the genetic algorithm is based on an iterative procedure of selection-evaluation, the entire optimisation process will be much more time-consuming than the RSD function. Therefore, the edge detection method in the RSD function is mainly used, the interactive working mode and the quiet working mode are also designed for the *GaOpt* function, and this function is mainly run on UNIX.

The initialisation and the working modes of the *GaOpt* function are the same as those in the RBT function. Fig.A-14 shows a screen capture of the interactive mode of the this function in MS-DOS. Fig.A-15 gives an example of the IN-file used for the function initialisation when the program runs in the quiet mode..

The genetic algorithm used in this research is a standard one, and the main operations, such as the *Crossover*, *Mutate* and *Select,* are adopted from the *Genesis* 5.0 written by Jone J. Grefenstette. In the *Crossover* operation, two-point crossover on the entire population is performed. In the *Mutate* operation, each position in the population string is given a chance (mutation rate) of undergoing the operation. The selection algorithms implemented in *GaOpt* are proportional selection based on the stochastic universal sampling algorithm and linear ranking selection both proposed by James E. Baker in [84] and [87] respectively.

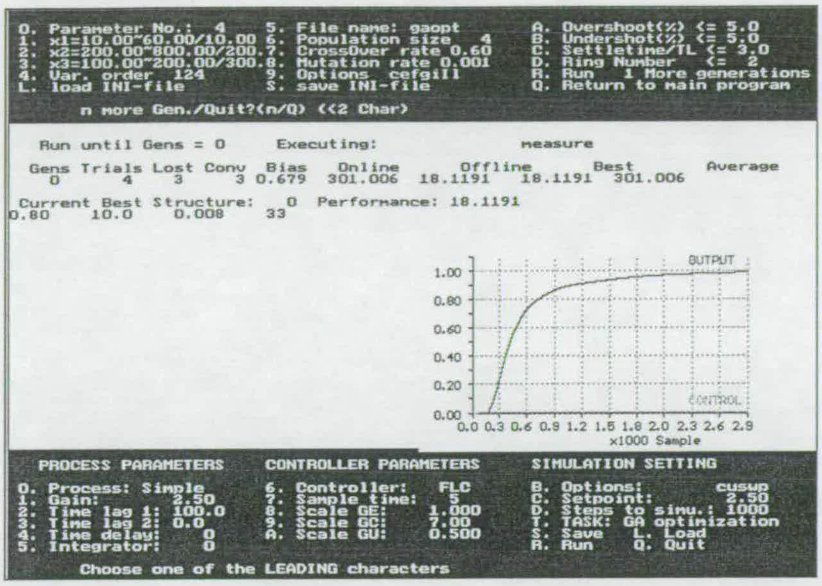


Fig.A-14 A screen capture of the interactive mode of *GaOpt* in MS-DOS.

```
                TASK = g
          Controller = f
          Plant Type = 3
     Plant variables = 178
  1st variable from = 1.000
                 to = 50.000
               step = 1.000
  2nd variable from = 2.000
                 to = 12.000
               step = 1.000
  3rd variable from = 0.500
                 to = 0.900
               step = 0.200
            Setpoint = 2.5
       OvershootSpec = 5
      UndershootSpec = 5
      SettletimeSpec = 3
     Ring NumberSpec = 2
        Mapping Task = 5
         Experiments = 1
        Total Trials = 1000
     Population Size = 50
      Crossover Rate = 0.6
       Mutation Rate = 0.001
      Generation Gap = 1.0
      Scaling Window = 5
     Report Interval = 100
    Structures Saved = 10
   Max Gens w/o Eval = 2
       Dump Interval = 0
         Dumps Saved = 0
          GA Options = cefgiIl
         Random Seed = 123456789
            Rank Min = 0.75
```

Fig.A-15 An example of the IN-file used in *GaOpt*.

In the evaluation stage, *GaOpt* passes each solution (*i.e.* controller parameter group) to *RobustMap* to evaluate the robustness of the system with this solution and determine the fitness by means of the fitness function defined in *Chapter* 3. After evaluation, the statistics of the robust performance of each solution in the present FLC generation will be collected and the convergence of this searching process will be determined. If the FLC population is not converged, *GaOpt* will randomly select some FLC solutions from present population. The block diagram of the *GaOpt* function has already presented in *Chapter* 4.

A.5 Real-time control interface

This function of the *FzySimu* package was only designed for the development of the microprocessor based real time controller (μPRC) used to confirm the simulation results of this package. Normally, the relevant development tools (both hardware and software) for the related microprocessor are necessary for this kind of development. But, on one hand, the microprocessor development tools are normally costly, on the other hand, the communication interface between this development tool and the simulation tool *FzySimu* is still needed for the automatic transmission of the controller's configuration data required by the extensive experiments. Therefore, the Motorola M68HC11 microprocessor was chosen for implementing the real time controller due to its simplicity in development [119]. The real time control interface, *RCPlot*, was designed in the *FzySimu* package for this real time controller (MS-DOS platform only).

RCPlot function performs the following operations: code generation, communication between a PC and the real time control system (RTCS), and displaying the dynamic response of the RTCS.

In the code generation operation, all controller's configuration data are firstly collected and saved into a data file as "DB" macros of the M68HC11 assembly. Note that each floating point variable in the simulated controller is represented in the μPRC as a fixed point variable in the format of 16 or 32 bits with 8 bits after the decimal point. Then this data file (DT) is merged with the library files to form the assembly program of the real time control system. The library files are specially designed for the RTCS in M68HC11 assembly, and they are the head file, the system management file, and four controller files. The head file defines the globe variables in the control software. The system management file provides the data sampling, data collection and communication in the RTCS. The controller file calculates the control input based on the control algorithm, such as FLC and PID. Finally, the assembly program is compiled by the compiler, ASMHC11, to generated the machine code corresponding to the present simulated controller.

The communication between the RTCS and the host PC is carried out via the RS-232 interface. Because no software permanently resides in the RTCS, the configuration of the RTCS is specially designed for this research. The M68HC11 in the RTCS was set to the boot-trap mode [119] which allows 256 bytes program to be downloaded into the on-chip RAM. A special program, called *Talker* designed by Motorola, is downloaded from the host PC to the RTCS via the serial port. With the talker running in the RTCS, the asynchronous serial communication becomes active and the read-write operations in the RTSC can be controlled by the host PC. With these basic functions provided by the *Talker*, the fundamental operations, such as downloading, executing and debugging, required by the development of microprocessor systems can be performed.

Note that the boot-trap working mode of M68HC11 does not satisfy the requirement for the dynamic memory and I/O ports in the real time systems. Therefore, after loading the *Talker*, the working mode of M68HC11 is changed to the expanded mode via re-configuration of the control register HPRIO and re-locating the interrupt vectors, see the block diagram shown in Fig.A-16. After this re-configuration, the RTCS based on M68HC11 has 32K bytes of RAM as well as the basic development functions.

For downloading and executing the control program (the machine codes), the communication protocol defined by the *Talker* (Table A-1) is implemented in the *RCPlot*. Note that to execute the control program, the "Memory Write" command is issued to modify the return address in the stack so that when the *Talker* finishes the interrupt services for the communication and returns to the main program, the control program is started. A similar technique is used to stop the execution of the control program.

Another operation performed by the *RCPlot* is to display the response of the RTCS and compare it with the simulation result. In this operation, the system performance data saved by the control program are read from the RTCS to the host PC and plotted together with the simulation results. Fig.A-17 shows a screen capture of the *RCPlot* function.

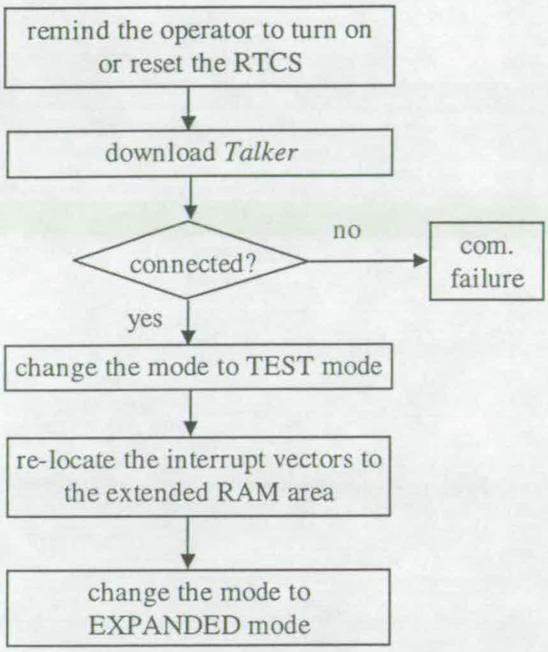


Fig.A-16 Block diagram of re-configuration of the RTSC

Table A-1 Communication protocol

Code (Hexadecimal)	Operation
01H	Memory read
41H	Memory write
81H	Register read
0C1H	Register write
0B5H	Break point information

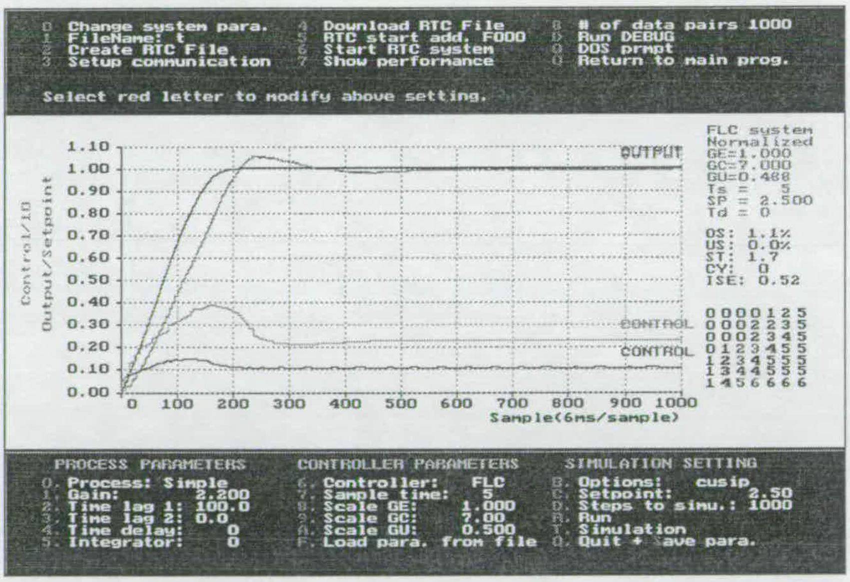


Fig.A-17 A screen capture of the *RCPlot* function

APPENDIX B

Confirmation of Simulation Results

B.1 Introduction

The research work presented in this thesis is mainly based on computer simulations and extensive experimental results. The simulation tool played a very important role in this research work. Therefore, the creditability of the simulation results is critical to this research. After the design of the simulation program, the simulation results of different control systems have been confirmed by comparing with the results from a practical control system to which the control algorithms were applied.

A practical control system was designed for the confirmation. It consists of a microprocessor embedded universal controller (MEUC) and a third order analogue emulator (AE3). Control algorithms were implemented in the controller by software techniques. The controlled process (not more than third order) was realized in the analogue emulator by hardware configuration. The communication between the control system and a personal computer was supported so that automatic data processing and system configuration can be performed by using the *RTPlot* function in the simulation package *FzySimu* (see *Appendix* A).

In this appendix, the implementation of the MEUC is presented in *Section* B.2 with respect to both hardware and software. *Section* B.3 describes the structure of AE3 and the method for process emulation. In *Section* B.4, dynamic performance of the practical control system is compared with the simulation results. It is concluded in *Section* B.5 that the simulation program is reliable and the results from the simulation compare well with results obtained from the practical systems.

B.2 MEUC

The following facts have been considered in designing the MEUC:

- It is supposed that the controller process is controlled by the analogue voltage signal and its output can be transferred (by certain sensors) into a voltage signal. Both the control input and process output signals are d.c. in the range of [-12v, 12v].
- The communication between the controller and a PC is required for the purpose of development and high level supervisory management.
- The controller can record system dynamic performance for the confirmation of the simulation results.

All these facts will be reflected in the hardware and software described as follows.

B.2.1 Hardware design

The FLC circuit is shown in Fig.B-1. Due to its simplicity and the flexibility of developing feature, the Motorola M68HC11 microprocessor is used in the MEUC. As introduced in the *Appendix A*, the M68HC11 microprocessor system can be economically developed by using the *boot-trap* function. This function can be selected by setting the selection lines (MODA and MODB) to the ground. However, M68HC11 can not access the external memory and I/O ports in the *boot-trap* mode. It is necessary to re-configure the M68HC11 into the *expanded* mode, which can be carried out by the software (see *Appendix A*). For the data recording, two RAM chips, 62256 and 6264, have been used to provide 40k byte memory space. Two jumpers were used for re-configuring the 32k RAM (62256) into the 32k EPROM (27C256) for the stand-alone working mode of the MEUC. In order to communicate with the PC machine, the Max232 chip was used to provide an RS-232 serial port which is commonly used in PCs.

Since the control signal and the process feedback signal are analogue, the converters from analogue to digital (A/D) and digital to analogue (D/A) are important elements in the controller. In practical control systems, the number of the input or output signals could be more than one. So the multi-channel A/D and D/A interfaces were designed in the MEUC.

M68HC11 has 8 × 8 bits A/D converter which can be used as the A/D interfaces. Two channels of D/A (12 bits) were designed with MAX501 which can provide analogue output in the range from -5v to 5v. Two jumpers were used for selecting the reference signals of the MAX501 in order to change the sign of the controller output, see Fig.B-2.

Since the inputs of the A/D converter require a signal variation from 0v to 5v, there should be a level shifter to convert the controller input signals from the range (-12v ~ 12v) to (0v ~ 5v). Similarly, there should be a level shifter to convert the controller output signals from the range (-5v ~ 5v) to (-12v ~ 12v). The level-shift circuit, as shown in Fig. B-3, consists of four CA3140 operational amplifiers with flexible adjustability.

In addition, considering the possibility to turn it into a commercial product, the LCD display interface (LM032L) and the keypad interfaces (74C922N) were implemented in the controller.

The access addresses of all components in the MEUC are listed in Table B-1.

Table B-1 Component address list

Component	Address (Hex.)
RAM 62256 (EPROM 27256)	8000 ~ 0FFFF
RAM 6264	4000 ~ 7FFF
D/A Max501 (1)	4010 (LSB), 4018 (MSB)
D/A Max501 (2)	4014 (LSB), 401C (MSB)
A/D (0-7) (on chip)	1030 (control), 1031 ~ 1034 (results)
RS-232	102B ~ 102E (control), 102F (data)
LM032L	4008
74C922N	4007

B.2.2 Software design

To make the real time control system comparable with the simulation system, the real time control software were designed to implement the control algorithm in the same manner as the simulation, and the controller parameters, such as the membership functions, rule base

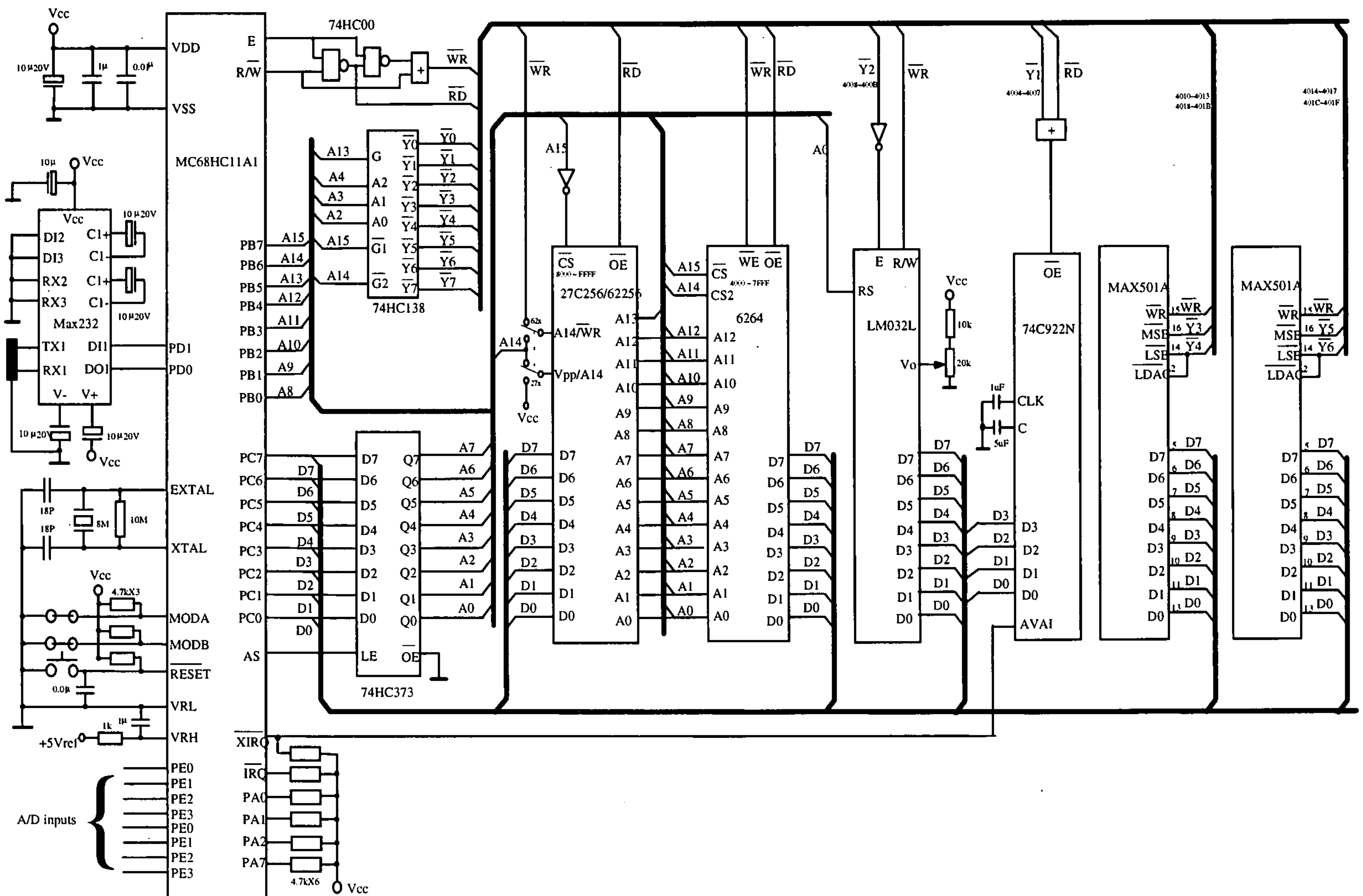


Fig.B-1 MEUC circuit

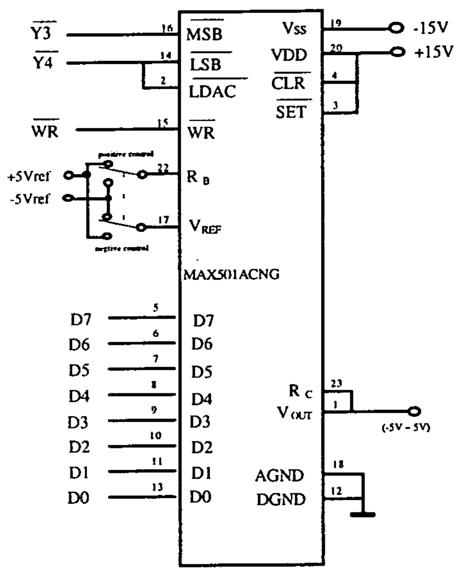


Fig.B-2 D/A converter

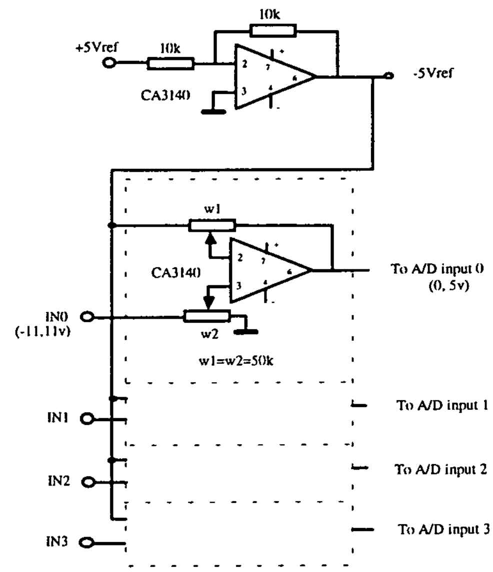


Fig.B-3 Level-shift circuit

and sample rate, are directly transferred from the simulation configuration to the data segment when compiling the machine codes in the real time controller (see *Appendix A*). In addition, the controller parameters in the simulation are also constrained to the 16 bit (binary) resolution.

Since the FLC control algorithm takes about 5ms to work out the control input in the MEUC, the timing unit in the MEUC is set to 6ms. That is, one sampling period in the simulation is equivalent to 6ms in the real time control system.

To increase the calculation speed, the following methods are used:

- The FLC's software was designed in assembly language, not converted from any high-lever language..
- Triangle-shape membership functions were used for all fuzzy variables in the MEUC.

The block diagram of the program used in MEUC is shown in Fig.4-6 in *Chapter 4*. Note that the program used in the MEUC is only for the confirmation experiments. Some functions, such as the LCD display and keypad management, are not implemented in the program.

B.3 Process emulation

In the robustness study of the FLC systems, the following two processes are mainly used.

- First order with time delay: $G_1(s) = \dfrac{Ke^{-\tau s}}{1+Ts}$.
- Second order process: $G_2(s) = \dfrac{K\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$.

To confirm the simulation results, these processes have been emulated by using the analogue hardware. The implementation of these processes is shown in Fig.B-4, where the process parameters can be calculated as follows.

- For $G_1(s)$: $K = \dfrac{R_{12}R_{22}}{R_{11}R_{21}}$, $T = R_{22}C_2$; time delay τ is implemented by software.
- For $G_2(s)$: $K = \dfrac{R_{12}}{R_{11}}$, $\zeta = \dfrac{1}{2}\sqrt{\dfrac{T_3}{T_2}}$, $\omega_n^2 = \dfrac{1}{T_2 T_3}$, $T_2 = R_{22}C_2$, $T_3 = R_3C_3$, $R_{12}=R_{14}$, $R_{21}=R_{22}$.

Because variations of the parameters are required by the study, each operational amplifier unit was implemented to enable multiple choices of capacitor or resistor values. With the jump connectors, it is very easy to obtain any combination of these components.

Note that the time delay τ is implemented by software. A 2k byte RAM block is reserved to record up to 1000 samples of the control u. Each sample of the control u is retrieved after the time delay. Therefore, the time delay can be set from 0 to 6 second (6ms per sample).

B.4. Simulation confirmation

To perform the confirmation, 10 experiments were carried out with different system configurations. These experiments are listed below.

1) First order process (low K, high T) controlled by FLC. Process: $G_1(s)$ with K=1.07, T=2.1s, τ=0. Controller: FLC with GE=1, GC=10, GU=0.2, t_s=20.

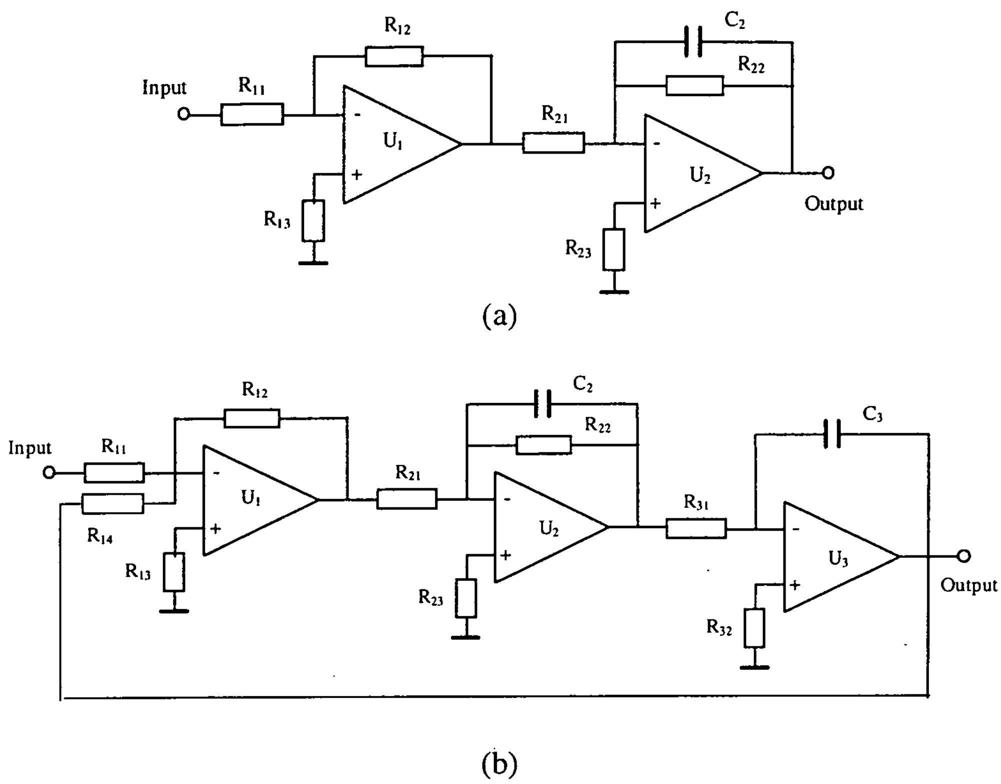


Fig.B-4 Hardware implementation of the controller processes.
(a) First order; (b) second order.

2) First order process (low *K*, high *T*, with τ) controlled by FLC. Process: $G_1(s)$ with K=1.07, T=2.1s, τ=50. Controller: FLC with GE=1, GC=10, GU=0.2, t_s=20.
3) First order process (high *K*, high *T*, with τ) controlled by FLC. Process: $G_1(s)$ with K=9.7, T=2.1s, τ=50. Controller: FLC with GE=1, GC=10, GU=0.08, t_s=20.
4) First order process (low *K*, low *T*, with τ) controlled by FLC. Process: $G_1(s)$ with K=1.07, T=1.05s, τ=50. Controller: FLC with GE=1, GC=10, GU=0.2, t_s=20.
5) First order process (high *K*, high *T*, with τ) controlled by PID. Process: $G_1(s)$ with K=9.7, T=2.1s, τ=50. Controller: PID with K_p=0.5, K_i=0.0117, K_d=5, t_s=20.
6) First order process (low *K*, high *T*, with τ) controlled by PID. Process: $G_1(s)$ with K=1.07, T=2.1s, τ=50. Controller: PID with K_p=0.6, K_i=0.043, K_d=5, t_s=20.
7) First order process (low *K*, low *T*, with τ) controlled by PID. Process: $G_1(s)$ with K=1.07, T=1.05s, τ=50. Controller: PID with K_p=0.6, K_i=0.043, K_d=5, t_s=20.
8) First order process (low *K*, low *T*) controlled by PID. Process: $G_1(s)$ with K=1.07, T=1.05s, τ=0. Controller: PID with K_p=0.6, K_i=0.043, K_d=5, t_s=20.

9) Second order process (low K, low ζ, high ω_n) controlled by FLC. Process: $G_2(s)$ with K=0.76, ω_n=27, ζ=0.5. Controller: FLC with GE=1, GC=2, GU=0.5, t_s=12.
10) Second order process (high K, high ζ, low ω_n) controlled by FLC. Process: $G_2(s)$ with K=1.66, ω_n=8, ζ=1.45. Controller: FLC with GE=0.8, GC=10, GU=0.2, t_s=12.

The confirmation results are presented from Fig.B-5 to Fig.B-14. Each figure shows the step response of the real time control system and the simulation result. The important parameters are indicated with their values in each figure.

Note that legend in each figure has following meanings;

u_rt: real time control; y_rt: real time output;

u_sm: simulation control; y_sm: simulation output.

B.5 Conclusion

From the confirmation results, it can be found that there is a good match between the real system performances and the simulation results if all system parameters are set the same. The most obvious difference between these two system's performances is that there are small ripples in the control signal of the real system which are not observed in the simulations. This situation has been examined carefully by debugging the real system and it was found that the conversion error in A/D chips and the noise from the power supply is the main reason for this problem. The effect of noise in practical systems cannot be neglected. A small GC will help to reduce this effect. When the performance at small GC is not satisfied, selecting a bigger sample rate may help to improve the performance. In simulating a practical process, the internal saturation of the process should be considered. Even if the input/output of the process are not saturated, the state variable inside, for example, the output of the first operational amplifier may saturate. In addition, the variation ranges of system parameters should be scaled uniformly. The most important unification is the membership functions of the FLC.

It is concluded that the simulation software used in this research correctly predicts the main features of the performance of practical control systems.

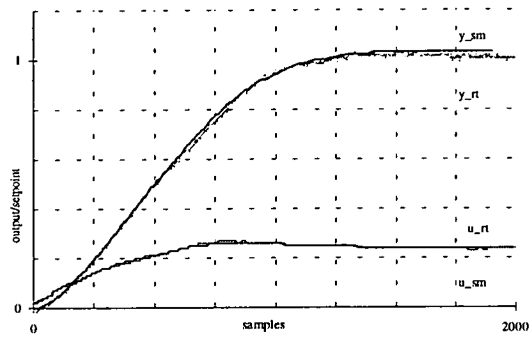


Fig.B-5 Experiment 1.

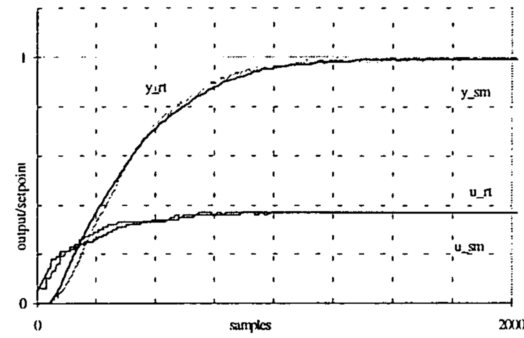


Fig.B-6 Experiment 2.

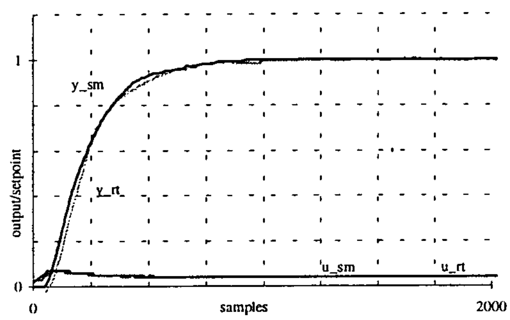


Fig.B-7 Experiment 3.

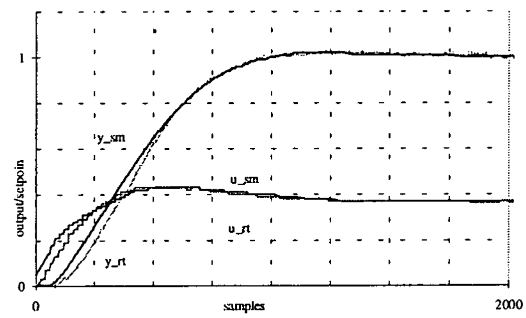


Fig.B-8 Experiment 4.

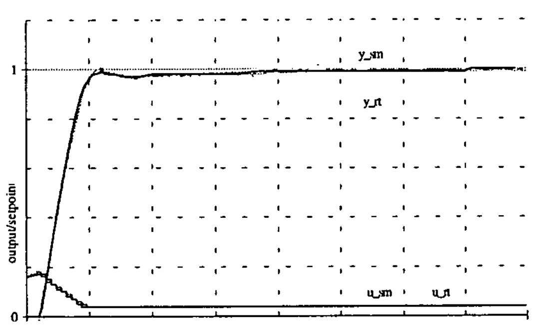


Fig.B-9 Experiment 5.

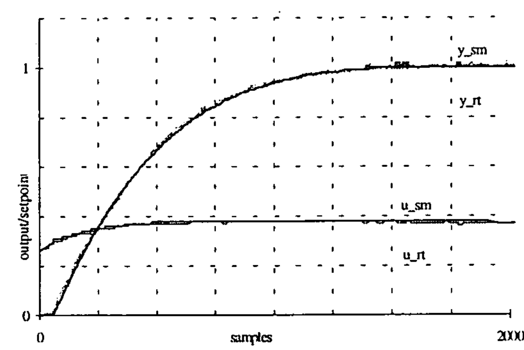


Fig.B-10 Experiment 6.

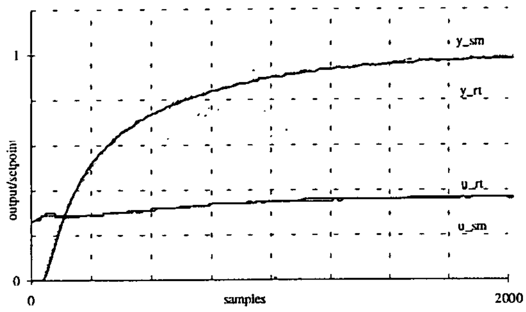


Fig.B-11 Experiment 7.

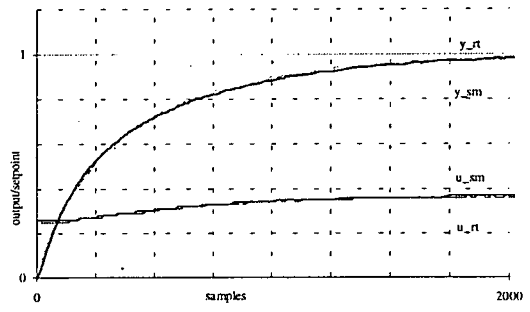


Fig.B-12 Experiment 8.

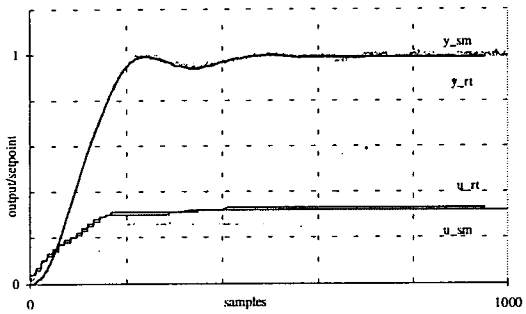


Fig.B-13 Experiment 9.

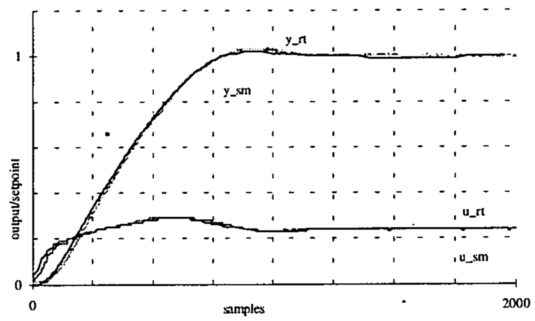


Fig.B-14 Experiment 10.

AN EXPERIMENTAL INVESTIGATION OF THE DYNAMIC PERFORMANCE OF FUZZY LOGIC CONTROL

J. Wang and J.R. Jordan

Abstract

This paper presents the results of an experimental (simulation based) investigation of the effect of scaling factor choice on the step response of a fuzzy logic controlled system. It is shown that if scaling factors are adjusted to give the best step response then small changes of process parameters (eg. time constant and gain) quickly degrades this good performance. Alternatively, if the performance required is degraded from the best step response obtainable by tuning the scaling factors, then scaling factors can be selected to give an acceptable performance over a wide variation of process parameters.

Key Word -- fuzzy logic control; scaling factor; robust performance.

1. Introduction

Fuzzy logic controllers (FLCs) are being used successfully in an increasing number of application areas, such as pilot-scale process control (King and Mamdani, 1976), pH control (Karr and Gentry, 1993), control of muscle relaxant anaesthesia (Linkens and Hasnain, 1991), and laser beam alignment (Marchbanks and Jamshidi, 1993). In many applications (Chiu \f2et al.\f1, 1991; Liaw and Wang, 1991; Pedrycz, 1993; Ray and Majumder, 1985; Tong, 1977), it is found that the performance of fuzzy logic control is robust under a wide range of variation in process parameters. This is a desirable feature since, in most control applications, the accuracy of the plant model is not guaranteed, and plant characteristics may change considerably during normal operation.

In many of these applications that the robust feature was observed, the effect of the parameters of the FLC on this performance feature was not a primary concern. For example, Procyk and Mamdani (1979) gives a description of the effect of scaling factors on system performance, but the rule-learning ability and the convergence of a self-organised fuzzy logic controller was the major concern of that paper.

In practical design, the rule base and the membership functions are usually first decided, and then the scaling factors (and also the sample rate) are adjusted to obtain the required dynamic performance. It is a very demanding procedure to choose these parameters if the range of variation of process parameters is relatively wide and a robust dynamic performance is required. General guidelines for tuning the performance of a fuzzy logic controller are not available.

This paper presents results demonstrating the effect of FLC's parameters on the range of variation of process parameters, over which the dynamic response (eg. a step response) remains within the designed specification. An exhaustive search of the process parameter space, K×T, and the scaling factor space, GE×GC×GU, has been carried out by simulation to establish the range of acceptable system performance. From these experiments guidelines are being developed for the design of fuzzy logic controllers.

For the purpose of this paper, robust performance is used to indicate a dynamic response that remains within the designed specification.

2. Fuzzy Logic Controller

To demonstrate the performance of a fuzzy control system, a simple fuzzy logic controller was simulated and used to control a simulated single-input-single-output process. The controller, originally conceived by Mamdani and Assilian (1975) and based on Zadeh's fuzzy set-theory (1965), was designed to regulate the output of the process around a set-point. A block diagram of the simulated controller is shown in Fig.1.

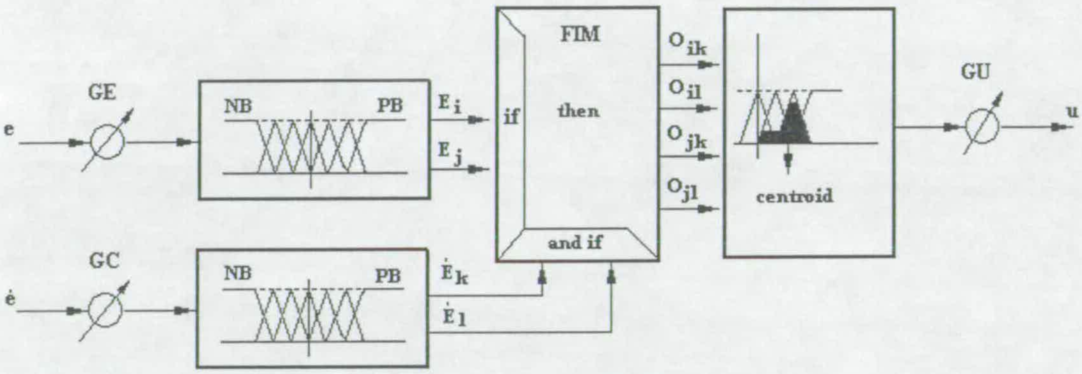


Fig.1 The fuzzy logic controller.

The controller first calculates the error and the change of error from the samples of set-point and feedback of the process output. Then these numerical values are fuzzified into linguistic values with degrees of membership to predefined fuzzy sets. After evaluating the fuzzy rules, several linguistic values of control output are generated and defuzzified into a numerical value which is used to control the process.

In the simulation, two inputs to the FLC, error (e) and change of error ($\dot{e}$) are calculated at step k as:

$$e(k) = S(k) - O(k) \quad (1)$$

$$\dot{e}(k) = \{e(k) - e(k-1)\} / t_s \quad (2)$$

where S is the set point, O is the output of the process, and t_s is the sampling interval of the controller. The inputs and output of the FLC is limited within a defined range [-10, 10], and the sampling rate for the process simulation is designed to be more than 5 times higher than that of the controller to ensure that a continuous performance is approximated.

The input and output variables of the FLC are associated respectively with three fuzzy sets: ERROR, CHANGE IN ERROR and CHANGE IN PROCESS INPUT (denoted by E, C and U respectively). Seven fuzzy subsets are chosen for each fuzzy set and named as NB, NM, NS, ZE, PS, PM, PB. Fuzzy subsets contain elements with degrees of membership. A fuzzy membership function $m_A : Z \rightarrow [0, 1]$ assigns a real number between 0 and 1 to every element z in the universe of discourse Z. This number $m_A(z)$ indicates the degree to which the object or data z belongs to the fuzzy set Z. Equivalently, $m_A(z)$ defines the *fit* value of element z in Z.

The corresponding fuzzy subsets, E_k , C_k and U_k can be expressed by a set of ordered pairs, i.e.

$$E_k = [e, \mu_{E_k}(e)] \subset E \quad (3)$$

$$C_k = [\dot{e}, \mu_{C_k}(\dot{e})] \subset C \quad (4)$$

$$U_k = [u, \mu_{u_k}(u)] \subset U \quad (5)$$

where e, $\dot{e}$ and u are elements of non-fuzzy variables, and $m_{E_k}(e)$, $m_{C_k}(\dot{e})$ and $m_{U_k}(u)$ are the corresponding membership values which give the degree to which the element is a member of the subset.

Fuzzy membership functions can have different shapes depending on the designer's preference or experience. In practice triangular and trapezoidal shapes are widely used because they help capture the modeller's sense of fuzzy numbers and simplify computation. Fig.2 shows membership function graphs of the fuzzy subsets, where NB for Negative Big; NM for Negative Medium; NS for Negative Small; PB for Positive Big; PM for Positive Medium; PS for Positive Small and ZE for Zero. In Fig.2 the fuzzy set ZE is narrower than the other fuzzy sets. The narrow set permits fine control near the set point while the wider fuzzy sets permit coarse control when the output is far from the set point.

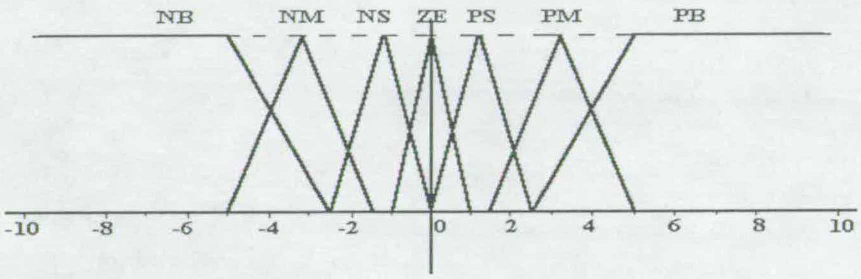


Fig.2 The fuzzy membership function.

A scaling factor for each variable is used to adjust the sensitivities of the controller to this variable. These scaling factors are gain of error (GE), gain of change of error (GC) and gain of control output (GU). The fuzzyfication procedure consists of first scaling the actual value by multiplying them by a suitable scaling factor and then quantizing the result according to the defined membership function. The effect of fuzzyfication can be altered by changing the value of the scaling factor or changing the definition of the membership function.

The fuzzy rule base provides an inference mapping from the input fuzzy universe to the output fuzzy universe and takes the form of a set of linguistic conditional statements:

$$\text{IF } E \text{ is } E_k \text{ and } C \text{ is } C_k \text{ THEN } U \text{ is } U_k \text{ .}$$

Every rule expresses a single control action at the stated system situation. The entire fuzzy rules base is usually represented by a single matrix called a fuzzy inference matrix (FIM). Fig.3 shows the FIM used in the simulation study.

		$\dot{e}$						
		NB	NM	NS	ZE	PS	PM	PB
e	NB	NB	NB	NB	NB	NB	NB	NS
	NM	NB	NB	NB	NM	NM	NM	PS
	NS	NB	NB	NM	NS	NS	PS	PM
	ZE	NB	NM	NS	ZE	PS	PM	PM
	PS	NM	NS	PS	PS	PM	PM	PM
	PM	NS	PM	PM	PM	PM	PM	PM
	PB	PS	PB	PB	PB	PB	PB	PB

Fig.3 Fuzzy rule base

The design of the FIM used is based on experience obtained in controlling the simulated process manually. The control heuristic is illustrated in Fig.4. At phase 1 and 3 where the process output is moving away from the set-point at more than a relatively small speed, the control input to the process should be greatly reduced (or increased) to drive the output back to the set-point. At phase 2 and 4 where the output is moving toward the set-point, the control action should be carefully decided according to the change of the output so as to keep the output going at a satisfactory speed without a big overshoot. All these imprecise decisions can be executed by the fuzzy logic controller.

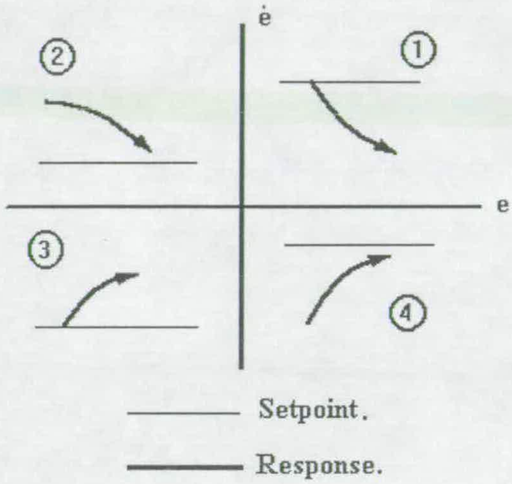


Fig.4 General control strategy

The centre-of-area method is used to defuzzify the output fuzzy variable. Since the output of the FLC is the change in control, an accumulator after the FLC is inserted to calculate the control input to the process if there is no integrator in the process to be controlled.

To simplify the evaluation of the compositional rule of inference and avoid heavy computational task, the fuzzy rule base in Fig.3 is represented in the form of a two-dimension data group. That is, let seven numbers (0-6) denote the seven fuzzy subsets NB, NM, NS, ZE, PS, PM, PB respectively, then Fig.3 can be rewritten into a 7×7 integer data group as follows:

```
FIM[7][7] = {( 0, 0, 0, 0, 0, 0, 2 );
             ( 0, 0, 0, 1, 1, 1, 4 );
             ( 0, 0, 1, 2, 2, 4, 5 );
             ( 0, 1, 2, 3, 4, 5, 5 );
             ( 1, 2, 4, 4, 5, 5, 5 );
             ( 2, 5, 5, 5, 5, 5, 5 );
             ( 4, 6, 6, 6, 6, 6, 6 )}
```

This data group is easy to store in the computer memory and it is easy to use it to implement the linguistic inference operations.

Another approach to simplify the evaluation of the fuzzy rule is that there should be no more than two fuzzy subsets whose membership functions are overlapped. Because of this restriction, four rules at most can be invoked at any system state, which leads to a great reduction in computing load and meets most control requirement in practice. For example, if $e(k) = 2.0$, and $\dot{e}$ (k) = -1.0 after multiplying the scaling factors at step k, it can be found from Fig.2 that only PS and PM in the E universe, and NS and ZE in the C universe have a non-zero fuzzy degree of membership. Therefore only four rules (FIM[4, 2], FIM[4, 3], FIM[5, 2], FIM[5, 3]) are needed.

3. Experiment results

To demonstrate the performance of the fuzzy controller, a simulation study has been carried out on a single-input-single-output system. The process to be controlled has the following transfer function:

$$G(s) = \frac{K}{Ts+1} \tag{6}$$

where K is the process gain within a range of (50 ~ 400), T is the process time constant within a range of (50 ~ 400). Note that the time constant is expressed using the fast simulation sampling interval of the process (not t_s) as the unit interval.

A step response test was used to determine the effect of the FLC's parameters (GE, GC, GU) on the robustness of the fuzzy logic control system. The criteria under which the performance of the fuzzy system is considered acceptable were specified as:

(1) Rise time <= 4T;
(2) Overshoot <= 5%;
(3) Settling time <= 5T.

Note that the rise time is the time for the system output to rise from 10% to 90% of the steady state value. Also the settling time denotes the time spent for the system output to get into the output range specified by 99% to 101% of the steady state value.

In this section, two investigations will be reported. The first one investigated the robust performance of the system with scaling factors tuned to give the best step response at the tuning point, and the second one investigated the effect of scaling factors on the robust performance of the fuzzy logic controlled system.

3.1 Robust performance test

To test the robust performance of the fuzzy logic controlled system, the scaling factors of the FLC are tuned to give a best response (fast without overshot) at K=100 and T=100 (tuning point). Then, with these scaling factors unchanged, the system performance was evaluated for parameter ranges of K=40 to 400 and T=40 to 400. An exhaustive search of the process parameter space, K×T, has been carried out to establish the range of acceptable system performance. Fig.5 shows the system performance at the tuning point with GE=1.5, GC=8.0 and GU=0.5. The variation range of gain K at different T, within which the acceptable performance is obtained, is shown in Table 1.

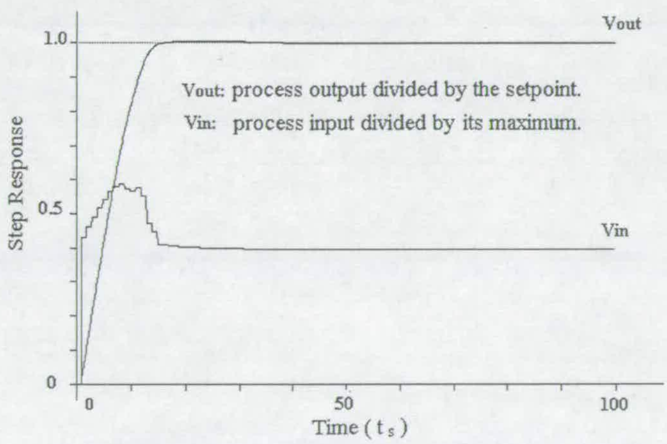


Fig.5 System performance at tuning point
(GE=1.5; GC= 8.0; GU=0.5; K=100; T=100; t_s=10)

T	50	100	150	200	250
K	50 ~ 200	50 ~ 250	50 ~ 150	50	50

Table 1: Robust range of K at different T

(GE=1.5; GC= 8.0; GU=0.5; t_s =10)

From Fig.5 it can be found that the step response of the system is very fast with 1.5T rise time without overshot. From Table 1 we can also see that the robust performance range of the system is within K=50 to 150 and T=50 to 150 which is a significant increase in robust range compared with a PI controller whose robust range was found to be K=85 to 120 and T=85 to 120 under the same test conditions.

The robust range can be significantly increased by tuning the values of these scaling factors. Many simulations have been carried out and it has been found that, with the same membership functions and rule base, a robust range for K of 50 to 250 and T of 50 to 200 can be achieved when scaling factor are chosen as GE=1.8, GC=12.0 and GU=0.25. Responses at the tuning point and the terminal points of the specified ranges (K=T=100; K=250, T=50; K=50 T=200; K=T=50; K=250, T=200) are shown in Fig.6.

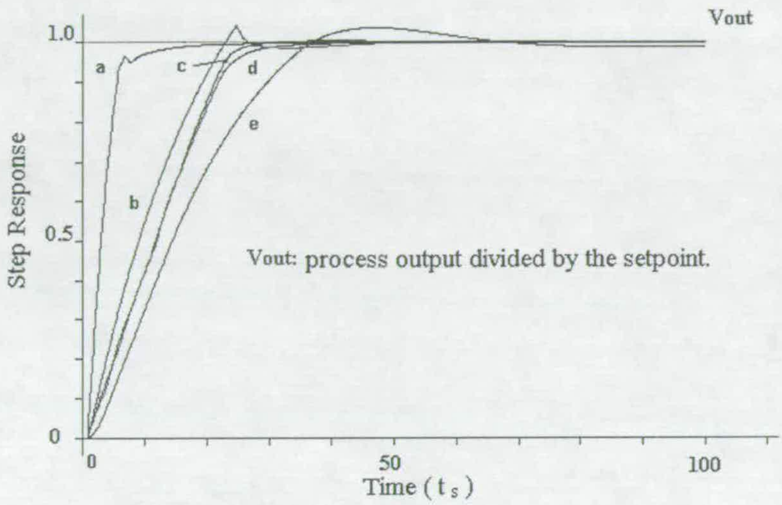


Fig.6 System performance with robust objective.
a: K=250, T=50; b: K=250, T=200; c: K=100, T=100;
d: K= 50, T= 50; e: K= 50, T=200.

From the results stated above, it is obvious that a very good performance can be obtained if the variation range of the plant parameters is not wide, while by appropriately choosing proper scaling factors, a wide robust range can be achieved.

3.2 Effect of scaling factors on the robust performance

To investigate the effect of scaling factors on the robust performance of the fuzzy logic controlled system, first the parameters of the FLC was chosen to give a satisfactory performance over the process parameter ranges K=50 to 250, T=50 to 250. The parameters which gave this performance are listed below:

GE=1.8; GC=12.0; GU=0.25.

Then one of these three parameters was varied about the value specified above, with the others unchanged to investigate the parameter range which still gave the desired performance. Table 2 to Table 4 show the results of these experiments.

These tables show that the specified robust performance over the parameter range K=50 to 250, T=50 to 200 can be achieved when only GE varies from 1.5 to 1.9, or only GC from 11 to 15, or only GU from 0.20 to 0.30. These scaling factor ranges were narrower than expected. However, a careful examination of these tables shows that the acceptable performance range of T moves to the higher-value region when GC increases; the acceptable performance range of K moves to the lower-value region when GU increases. This result shows that GC has a close relation with the process time constant and GU with the process gain, and this suggests that GC can be changed if T is changed and/or GU can be changed if K is changed to meet the performance requirement.

K range		*T* 50	100	150	200	250
	0.7	-	-	50 ~ 250	100 ~ 250	100 ~ 250
	0.9	-	50 ~ 250	50 ~ 250	50 ~ 250	150 ~ 250
	1.1	150 ~ 250	50 ~ 250	50 ~ 250	50 ~ 250	50
	1.3	100 ~ 250	50 ~ 250	50 ~ 250	50 ~ 250	50
	1.5	50 ~ 250	50 ~ 250	50 ~ 250	50 ~ 250	50
GE	1.7	50 ~ 250	50 ~ 250	50 ~ 250	50 ~ 250	50
	1.9	50 ~ 250	50 ~ 250	50 ~ 250	50 ~ 250	50
	2.1	50 ~ 200	50 ~ 250	50 ~ 250	50 ~ 250	50
	2.3	50 ~ 200	50 ~ 250	50 ~ 250	50 ~ 250	50
	2.5	50 ~ 200	50 ~ 200	50 ~ 250	50 ~ 250	50
	2.7	50 ~ 150	50 ~ 200	50 ~ 250	50 ~ 250	50
	2.9	50 ~ 150	50 ~ 150	50 ~ 250	50 ~ 250	50

Table 2: Variation range of gain *K* at different GE and *T*
(GC=12.0; GU=0.25; t_s =10)

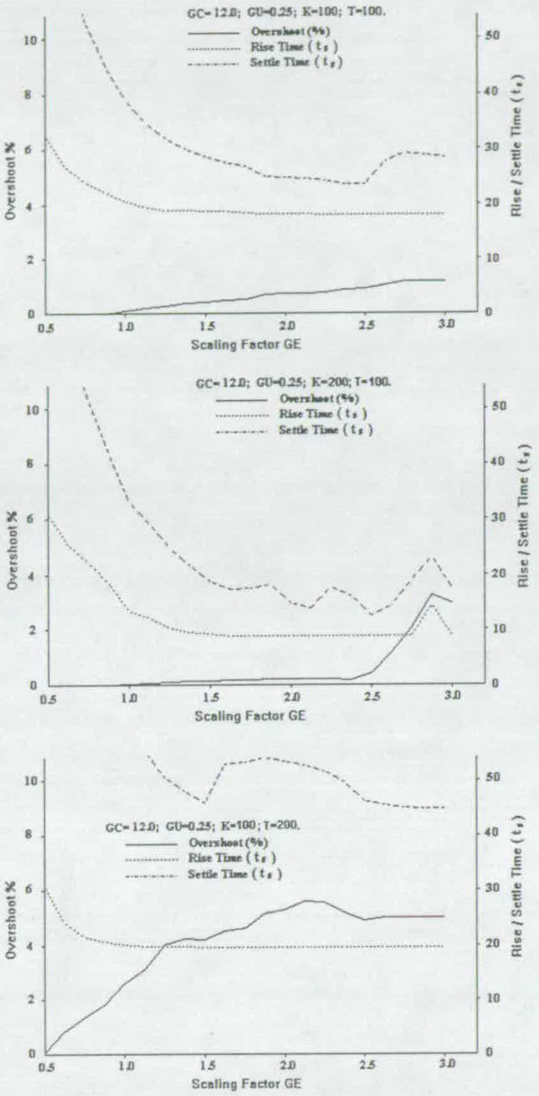
K range		*T* 50	100	150	200	250
	5	50 ~ 250	50 ~ 200	-	-	-
	7	50 ~ 250	50 ~ 250	50 ~ 250	50	50
	9	50 ~ 250	50 ~ 250	50 ~ 250	50	50
	11	50 ~ 250	50 ~ 250	50 ~ 250	50 ~ 250	50
GC	13	50 ~ 250	50 ~ 250	50 ~ 250	50 ~ 250	50 ~ 250
	15	50 ~ 200	50 ~ 250	50 ~ 250	50 ~ 250	50 ~ 250
	17	200	50 ~ 250	50 ~ 250	50 ~ 250	50 ~ 250
	19	150	50 ~ 250	50 ~ 250	50 ~ 250	50 ~ 250
	21	-	100 ~ 250	50 ~ 250	50 ~ 250	50 ~ 250
	23	-	100 ~ 200	50 ~ 250	50 ~ 250	50 ~ 250

Table 3: Variation range of gain *K* at different GC and *T*
(GE=1.8; GU=0.25; t_s =10)

T range		*K* 50	100	150	200	250
	0.05	-	100	50 ~ 100	50 ~ 100	50 ~ 100
	0.10	100	50 ~ 100	50 ~ 150	50 ~ 150	50 ~ 200
	0.15	50 ~ 150	50 ~ 150	50 ~ 200	50 ~ 200	50 ~ 200
	0.20	50 ~ 200	50 ~ 200	50 ~ 200	50 ~ 200	50 ~ 200
	0.25	50 ~ 250	50 ~ 200	50 ~ 200	50 ~ 200	50 ~ 200
GU	0.30	50 ~ 250	50 ~ 200	50 ~ 200	50 ~ 200	50 ~ 200
	0.35	50 ~ 250	50 ~ 200	50 ~ 200	50 ~ 200	100 ~ 200
	0.40	50 ~ 250	50 ~ 250	50 ~ 200	100 ~ 200	100 ~ 200
	0.45	50 ~ 250	50 ~ 250	50 ~ 200	100 ~ 200	100 ~ 200
	0.50	50 ~ 250	50 ~ 250	100 ~ 200	100 ~ 200	150 ~ 200
	0.55	50 ~ 250	50 ~ 250	150 ~ 250	100 ~ 250	150 ~ 200

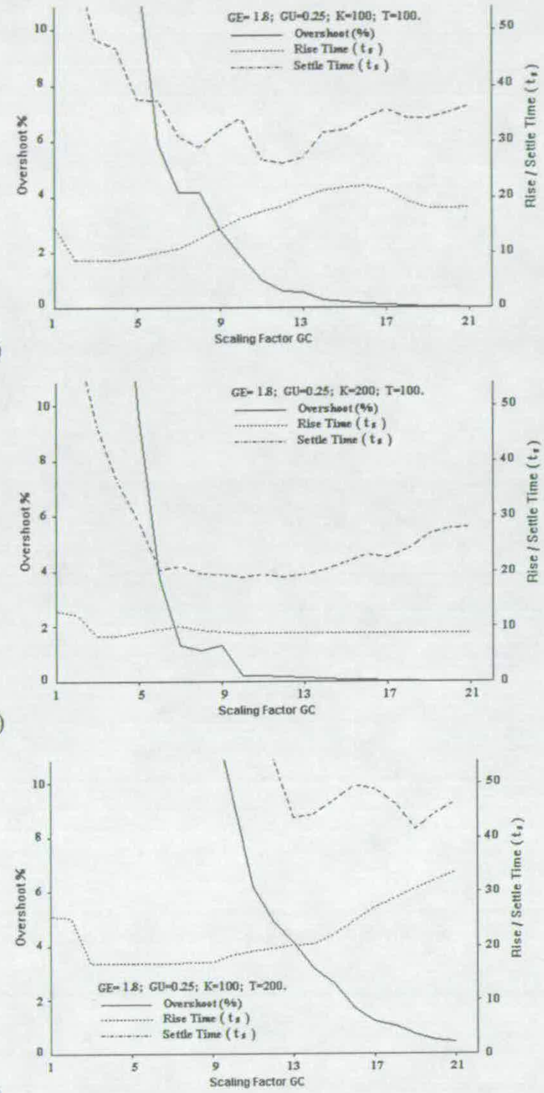
Table 4: Variation range of gain *T* at different GU and *K*
.(GE=1.8; GC=12.0; t_s =10)

An explanation for the narrow range of scaling factor values giving the required robust performance can be obtained by considering Fig.7, Fig.8 and Fig.9. Fig.7 shows that if K is small and T is large, a small GE will lead to slow response and a big overshot. This is because the tolerance band around the set-point is wide and considerable steady-state error and overshoot are acceptable. On the other hand, if K is large and T is small, a large GE will cause limit cycling at the steady state because the performance measure becomes more sensitive around the set-point and less sensitive during rise-time.



(a)
(b)
(c)

Fig.7 Effect of GE a. K=100, T=100. b. K=200, T=100. c. K=100, T=200.



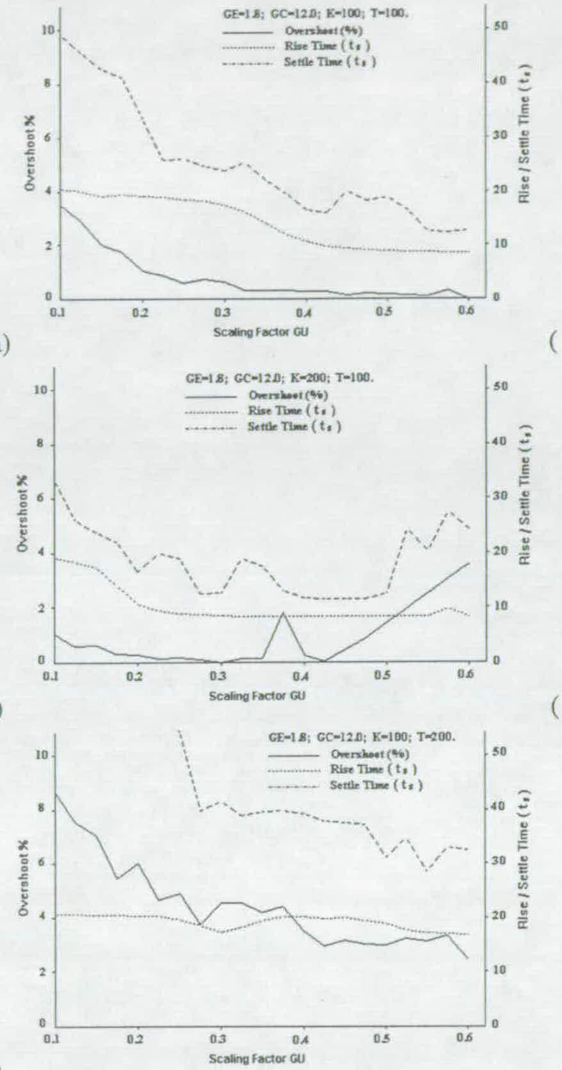
(a)
(b)
(c)

Fig.8 Effect of GC a. K=100; T=100 b. K=200, T=100. c. K=100, T=200

Change in GC affects the sensitivity of the controller to the change of error. Fig.8 shows that decreasing GC will lead to a quick response but this increases overshoot, while increasing GC will give a gentle rising speed of the step response but limit cycling will occur when the process gain is large. When GC is large, the performance can not meet the first requirement of the performance specification defined above if K is small. When GC is small, the controller can not discriminate between small rates of change so that it can not take action sufficiently early to overcome time lags if the process has a large time constant.

GU functions purely as an overall gain factor of the controller which varies the magnitude of the process input. Fig.9 give a demonstration of the effect of GU. A small GU leads to slow responses so that the overshoot could be large if T is big, while a big GU will give a fast response but could cause limited cycling when K is increased.

When time delay is added to the controlled process, the parameters of the controller have to be adjusted to meet the specified requirements. But it is much more difficult to obtain a wide robust range. Also, the variation range of the parameters of the controller giving acceptable performances become even narrower if the process has a time delay than that without one.



(a)

b)

c)

Fig.9 Effect of GU a. K=100; T=100. b. K=200, T=100. c. K=100, T=200.

3.3 Some tuning experiences

From practical observation of the performance of the simulated system, it has been found that there are upper and lower limits on the values of GE, GC and GU to give an acceptable performance. For example, if the gain of error GE is too small the fuzzy algorithm suffers from indecision whereas if it is too large the fuzzy algorithm performs erratically.

It is also found that, generally GE determines the amount of overshoot, GC determines the dynamic speed of the performance and GU influences the close-loop gain of the system. To achieve a good performance, GC should be changed proportionally to the time constant T and GU should be changed inversely proportional to the gain K.

Some general guidelines have been found in tuning these scaling factors. If error in steady state is big, then increase GE; if overshot is big, then decrease GE or increase GC. If system is not stable and output change is sharp and non-periodic, try decreasing GU or increasing GC. If system is not stable and output change is periodic but not sharp, try decreasing GU or GE. If time delay exists, GE or GU should be reduced or GC be increased.

4. Conclusion

The studies presented in this paper involve many simulation experiments on the robust performance of fuzzy logic control. Effort was mainly made to investigate the effect of the parameters of fuzzy logic controller on the robust performance of the system. It can be concluded that:

(1) The simulation result shows that, by choosing proper scaling factors, a very good performance can be obtained if the variation range of the plant parameters is not wide. Alternatively a wide robust range can be achieved if the performance required is degraded from the best step response obtainable by tuning the scaling factors

(2) The scaling factor ranges which give required robust performance are quite narrow. The robust performance range of T moves to the higher-value region when GC increases and the robust performance range of K moves to the lower-value region when GU increases. This result shows that GC has a close relation with the process time constant and GU with the process gain.

(3) The scaling factor GE affects the performance measure of error. The performance measure becomes more sensitive around the set-point and less sensitive during rise-time if GE is big. GC affects the dynamic speed of the whole system so that a big GC will lead to a quick response if there is any

interference in the system output. GU functions purely as an overall gain factor of the controller which varies the magnitude of the process input.

It should be realised that there are many factors in addition to the scaling factors, such as definition of membership functions, elements in rule base and sample interval, which can affect the performance of the system. Different combinations of values of these factors may result in the same response. The problem of choosing the values of the scaling factors to achieve a best robust performance is difficult to solve even when the controlled plant is a simple first order process.

5. References

Chiu S., Chand S., Moore D. and Chaudhary A. 1991. "Fuzzy Logic for Control of Roll and Moment for a Flexible Wing Aircraft", *IEEE Control Systems Magazine*, vol. 11, no. 4, pp.42-48.

Karr C.L. and Gentry E.J. 1993. "A Fuzzy Control of pH Using Genetic Algorithms", *IEEE Trans., On Fuzzy System*, vol. 1, no. 1, pp. 46-53.

King P.J. and Mamdani E.H. 1976. "The Application of Fuzzy Control System to Industrial Processes", *Automatica*, vol. 13, pp. 235-242.

Liaw C.M. and Wang J.B. 1991. "Design and Implementation of a Fuzzy Controller for a High Performance Induction Motor Drive", *IEEE Trans. on Syst., man, and Cybern.*, vol. 21, no. 4, pp. 921-929.

Linkens D.A. and Hasnain S.B. 1991. "Self-organising Fuzzy Logic Control and Application to Muscle Relaxant Anaesthesia", *IEE Proceedings_D*, vol. 138, no. 3, pp. 274-284.

Mamdani E.H. and Assilian S. 1975. "A Fuzzy Logic Controller for a DynamicPlant", *Int. J. Man-mach. Studies*, vol. 7, no. 1, pp. 1-13.

Marchbanks R.D. and Jamshidi M. 1993. "A Fuzzy Control Approach to A Two Axis Mirror Laser Beam Alignment System", *Engng. Applic. Artif. Intell.*, vol. 6, no. 2, pp. 117-122.

Procyk T.J. and Mamdani E.H. 1979. "A Linguistic Self-organising Process Controller", *Automatica*, vol. 15, no. 1, pp. 15-30.

Ray K.S. and Majumder D.D. 1985. "Fuzzy Logic Control of a Nonlinear Multivariable Steam Generating Unit Using Decoupling Theory", *IEEE Trans. on Syst., man, and Cybern.*, vol. 15, no. 4, pp. 539-558.

Tong R.M. 1977. "A Control Engineering Review of Fuzzy Systems", *Automatica*, vol. 13, pp. 559-569.

Zadeh L.A. 1965. "Fuzzy sets", *Informat. Control*, vol. 8, pp. 338-353.

The Robust Step Performance of PID and Fuzzy Logic Controlled SISO Systems

J. Wang and J. R. Jordan
Dept. Electrical Engineering
University of Edinburgh,
Edinburgh, UK

Abstract

This paper introduces an experimental (simulation based) investigation of the robust step performance of SISO systems controlled by FLC and PID algorithms. The results of a large number of simulation experiments are presented quantifying the claim that fuzzy control leads to an improved robust performance with respect to the use of PID control. The effect of tuning-point selection and tuning-point performance on the robust properties of the FLC and PID systems is also presented.

1. INTRODUCTION

Fuzzy logic control (FLC) has been claimed to exhibit a very good robust performance in many research papers [1 - 4]. The robust performance of an FLC has been experimentally tested in this published research work by changing the process parameters or by choosing different type of processes. It is commonly found that in a certain range of process dynamics the fuzzy control is insensitive to process parameter changes and acceptable step responses can be obtained in a relatively wide parameter space. However, none of these research papers gave a comparison with other control methods. In this paper, the results of a simulation study of the control of SISO systems are presented in an attempt to quantify the claim that fuzzy control leads to an improved robust performance with respect to the use of PID control.

In the next section, the simulated system is briefly described and the FLC and PID algorithms are presented. Section 3 presents the results derived from the simulated systems and Section 4 concludes the paper.

2. THE SIMULATED SYSTEMS

2.1. The Controllers

To compare the performance of a fuzzy control system with a PID control system, a fuzzy logic controller and a PID controller were designed to regulate the output of a single-input, single-output process around a set-point. A block diagram of the simulated control system is shown in Fig.1. Error e and derivative error $\dot{e}$ are calculated at step k as:

$$e(k) = x(k) - y(k) \tag{1}$$

$$\dot{e}(k) = \dot{y}(k) = [y(k) - y(k-1)] / t_s \tag{2}$$

where x is the set point, y is the output of the process, and
t_s is the sampling interval of the controller. To minimise overload of the controller input, the derivative signal was obtained from the process output y. The inputs and the output of the controller are limited to lie within the range (-10, 10), and

the sampling rate for the process simulation was designed to be more
than 5 times higher than that of the controller to ensure that a continuous performance is approximated.

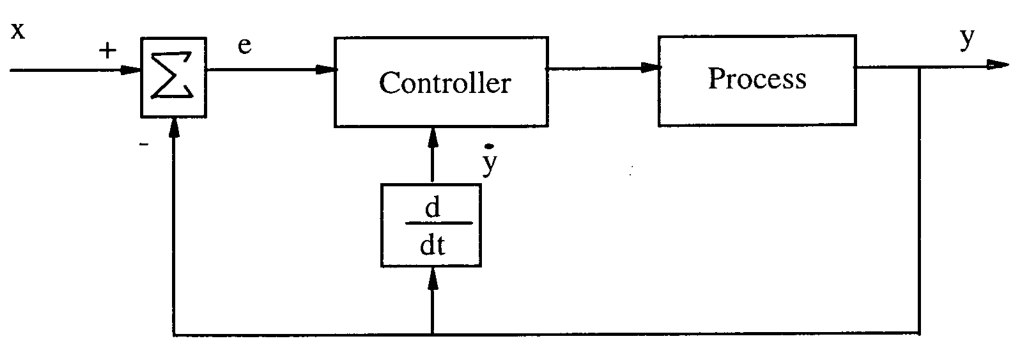


Fig.1 Structure of the simulated system

The input and output variables of the fuzzy logic controller are associated respectively with three fuzzy sets: ERROR, CHANGE IN ERROR and CHANGE IN PROCESS INPUT (denoted by ***E***, ***C*** and ***U*** respectively). Seven fuzzy subsets are chosen for each fuzzy set and named as NB, NM, NS, ZE, PS, PM, PB. Fuzzy subsets contain elements defined by a degree of membership parameter.

Fuzzy rule base and membership functions are designed heuristically or by the trial-and-error method with the objective of achieving a specified dynamic performance. To map the practical signal variation spaces to the designed fuzzy spaces, scale factors, GE (gain of error), GC (gain of change of error) and GU (gain of control output), are used.

The basic principle of the operation of the controller is described as follows. Suppose there are N rules in fuzzy rule base. The change of control input Δu is calculated as:

R_i : if e is A_i and $\dot{e}$ is B_i, then Δu_i is C_i ($i = 1, \cdots\cdots N$)

$$\alpha_i = \mu_{A_i}(e) \wedge \mu_{B_i}(\dot{e}) \tag{3}$$

$$\Delta\mu_i = cog(\alpha_i \wedge \mu_{C_i}) \tag{4}$$

$$\Delta u = \frac{\sum_{i=1}^{N} \alpha_i \Delta u_i}{\sum_{i=1}^{N} \alpha_i} \tag{5}$$

Note that cog(·) is the value obtained by defuzzifying with the centre-of-gravity method. An accumulator is used after the FLC to generatethe control signal from Δu.

The discrete-time equivalent expression for the PID controller used in this paper is given as:

$$u(k) = K_p e(k) + T_s \sum e(k) / T_i + K_d \Delta e(k) / T_s \tag{6}$$

where $u(k)$ is the control signal, $e(k)$ is the error and Δ $e(k)$ the change of error. K_p, T_i, K_d, and T_s are the proportional gain, integral time constant, derivative gain, and sampling period for the controller, respectively.

2.2. Controlled processes

To investigate the performance of FLC and PID algorithms in controlling SISO processes, the following processes were controlled by these algorithms. Note that the unit for the time constants is the sampling interval in the process simulation, and this unit will be used for all time constants in this paper.

Model 1: $G(s) = \dfrac{K}{1+Ts}$, (K = 1.2 to 10, T = 50 to 400);

Model 2: $G(s) = \dfrac{Ke^{-T_d s}}{1+Ts}$, (K = 1.2 to 10, T = 50 to 400, T_d = 0 to 50);

Model 3: $G(s) = \dfrac{K}{(1+T_1 s)(1+T_2 s)}$, (K = 1.2 to 10, T_1 = 50 to 400, T_2 = 0 to 225).

The reason for choosing such processes is that they are often met in practical control engineering. Experiments were carried out to determine the range of process parameter values that gave a controlled step response satisfying peak overshoot and settling time specifications. This parameter range is called the robust range in this paper. Use is made of the number of acceptable performance tests, together with the total number of the tests as the measure of the robust range of a system.

3. EXPERIMENTAL INVESTIGATION

In the robust parameter range the system's performance is considered acceptable (for the purposes of this paper) providing that:

(1) Maximum overshoot $Y_o \leq 5\%$;

(2) 2% settling time $T_s \leq 5T_\Sigma$;

Note that the 2% settling time T_s denotes the time spent for the system output to get into the output range specified by 98% to 102% of the steady state value, and
T_Σ in criteria (2) denotes the sum of all time constants in the controlled process.

The following experimental procedure was used:

Step 1: select one tuning point, i.e. choose one middle point in each parameter variation range for that parameter.
Step 2: tune controller parameters of both FLC system and PID system to obtain a similar system response (*i.e.* Y_o and T_s) at the tuning point to the step input.
Step 3: Fix the parameters of both FLC and PID. Test the system performance at different points of the process parameter space.
Step 4: Using the criteria defined above, determine the robust range for each system.

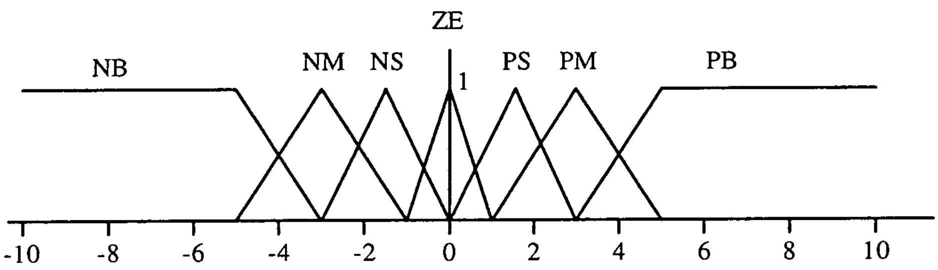
In the following, the experiment results derived from each process are presented together with a brief discussion.

Model 1

This is a simple process but it is often used as an approximation to many industrial processes. Parameters of both controllers are separately tuned at K=4.5, T=250 (the tuning point). The membership functions and the rule base in the FLC are shown in Fig.2. Other parameters of the controllers are:

FLC: GE = 1.0; GC = 12.0; GU = 1.0.
PID: $K_p = 1.0$; $T_i = 45.0$; $K_d = 1.0$.

After carrying out step 3 and step 4 in the procedure, percent overshoot and 2% settling time are automatically collected and presented in the form shown in Table 1 and Table 2 by the simulation program. The robust range is shown as the non-shaded area in each table. Robust ranges of the FLC system and the PID system take 88 and 56 test points respectively. From these results it would appear that the first order process can be controlled by the FLC with an improved robust range.



(a)

e \ $\dot{e}$	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NB	NS	NS	ZE	PS
NS	NB	NB	NB	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PM
PS	NM	NS	ZE	PS	PM	PM	PM
PM	NS	ZE	PS	PS	PM	PM	PM
PB	ZE	PS	PM	PB	PB	PB	PB

(b)

Fig. 2 Membership functions (a) and rule base (b) of the FLC

It is worth noting that tuning point selection is important if wide robust range is needed over the variation range of the plant's parameters, especially for the PID system. For example, if these two systems are tuned at K=1.5 and T=100 to give a similar tuning-point performance as above (GE=1.8, GC=8.0, GU=2.5; K_p=2.3; T_i=8.0; K_d=1.0), robust ranges of FLC system and PID system take 71 and 27 test points respectively. Considering this result, it can also be conclude that in the first order system case, the FLC is not only robust to the plant parameter change, but also to the selection of the tuning point.

Model 2

This process model is also often used as an approximate model for many industrial processes. Due to the pure time delay in its response to the control signal, this plant is much more difficult to control than the first model process. In practical control engineering, a controller tuned to handle process dead time tends to respond rather slowly while attempting to correct a deviation in the process output [5]. Although the process dead time is simple to estimate from a reaction curve, it is much harder to identify automatically during normal loop operations when the process input and the process output vary simultaneously, or noise and load disturbances can not be neglected.

TABLE I

Performance of FLC system at different points of the parameter space

Y_o , T_s		K									
		0.5	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5
T	50	0.00, 299	0.00, 299	0.00, 282	0.00, 234	0.00, 203	0.00, 184	0.00, 170	0.00, 298	2.46, 297	4.37, 299
	100	0.75, 426	0.26, 275	0.14, 242	0.09, 235	0.07, 217	0.06, 195	0.05, 181	0.04, 172	0.04, 163	0.03, 156
	150	1.97, 401	0.73, 272	0.35, 206	0.21, 208	0.16, 210	0.13, 207	0.10, 194	0.09, 182	0.08, 174	0.07, 169
	200	3.16, 620	1.34, 272	0.71, 202	0.40, 182	0.26, 189	0.21, 196	0.17, 199	0.14, 191	0.12, 184	0.11, 176
	250	4.32, 647	1.96, 281	1.16, 214	0.75, 173	**0.42, 172**	0.29, 178	0.23, 184	0.20, 190	0.16, 190	0.15, 183
	300	5.48, 665	2.63, 428	1.70, 220	1.20, 183	0.82, 161	0.47, 160	0.31, 168	0.26, 178	0.22, 183	0.20, 187
	350	6.61, 1070	3.31, 457	2.43, 330	1.58, 196	1.25, 164	0.85, 155	0.52, 154	0.36, 159	0.28, 172	0.24, 175
	400	7.67, 1120	4.09, 478	3.17, 363	2.37, 285	1.92, 177	1.34, 156	0.79, 152	0.54, 151	0.39, 152	0.30, 167
	450	8.60, 1153	4.74, 494	3.73, 383	2.94, 316	2.27, 261	2.01, 206	1.29, 152	0.77, 148	0.50, 150	0.44, 148
	500	8.61, 1209	5.31, 509	4.16, 399	3.45, 333	2.78, 285	2.38, 240	2.01, 186	1.44, 148	0.84, 146	0.54, 147

Note: the first number is Y_o and the second one is T_s .

TABLE II

Performance of PID system at different points of the parameter space

Y_o , T_s		K									
		0.5	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5
T	50	0.00, 299	0.00, 299	0.00, 299	0.00, 299	0.00, 299	0.00, 299	0.00, 299	0.00, 299	0.00, 299	0.00, 299
	100	0.00, 549	0.00, 549	0.00, 549	0.00, 549	0.00, 465	0.00, 401	0.00, 350	0.00, 308	0.00, 272	0.00, 241
	150	0.00, 799	0.00, 799	0.00, 575	0.00, 441	0.00, 353	0.00, 288	0.00, 236	0.00, 193	0.00, 156	0.00, 124
	200	0.00,1049	0.00, 667	0.00, 401	0.00, 275	0.00, 204	0.00, 160	0.00, 130	0.00, 109	0.00, 93	0.00, 82
	250	0.00,1299	0.35, 522	0.64, 315	0.75, 227	**0.80, 178**	0.81, 146	0.81, 124	0.80, 107	0.79, 95	0.77, 84
	300	0.00,1457	2.05, 807	2.52, 642	2.57, 513	2.50, 425	2.40, 359	2.29, 306	2.19, 263	2.09, 224	2.00, 91
	350	0.76,1321	4.00,1098	4.44, 819	4.35, 667	4.14, 568	3.92, 496	3.70, 441	3.50, 396	3.32, 358	3.16, 326
	400	2.08,1965	5.91,1234	6.25, 924	6.02, 762	5.68, 657	5.33, 582	5.02, 525	4.73, 478	4.47, 439	4.25, 406
	450	3.48,2299	7.73,1332	7.95,1002	7.57, 832	7.11, 724	6.66, 647	6.25, 588	5.88, 540	5.56, 500	5.27, 467
	500	4.94,2549	9.43,1412	9.53,1066	9.02, 889	8.45, 778	7.90, 699	7.41, 638	6.97, 590	6.58, 549	6.24, 515

From the practical engineering point of view, the slow-control method is adopted and the dead time is assumed unknown in the simulation. Parameters of both controllers are separately tuned at K=4.5, T=250, T_d=10 (tuning point). The membership functions and the rule base in the FLC are the same as shown in Fig.2. Other parameters of the controllers are:

FLC: GE = 0.8; GC = 25.0; GU = 0.1.

PID: K_p = 0.28; T_i = 200.0; K_d = 5.0.

After carrying out step 3 and step 4 in the procedure, information on the system performances is automatically collected by the simulation program, but can not be shown here due to the page limitation. Two 3-D surface charts (Fig.3 and Fig.4) have been included to demonstrate these data. Points in the parameter space above the surface shown give an unacceptable step response.

Simulation data shows that the robust ranges of the FLC system and the PID system cover 623 and 378 test points in 1000 test points respectively. From these figures, we can also see that for a model 2 process with a fixed time delay, the FLC system has a wider K-T area than that of the PID system, and for high gain, high time constant processes, the FLC system can cope with higher time delay than the PID system.

Note that the performance of the FLC system at the tuning point gives a stronger effect on the robust range than that of the PID system. For example, if we set the controller parameters to decrease the settle time from the above 830 to 580 at the same tuning point (GE=0.8, GC=20.0, GU=0.15; K_p=0.3, T_i=160.0, K_d=5.0), then the robust range becomes only 491 test points for the FLC system but 338 test points for the PID system in 1000 test points.

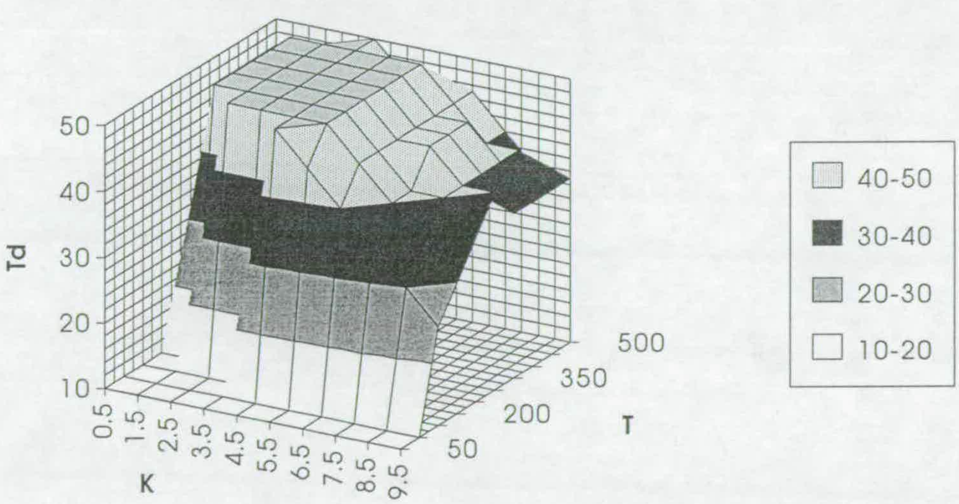


Fig.3 Robust space of FLC in controlling Model 2 process

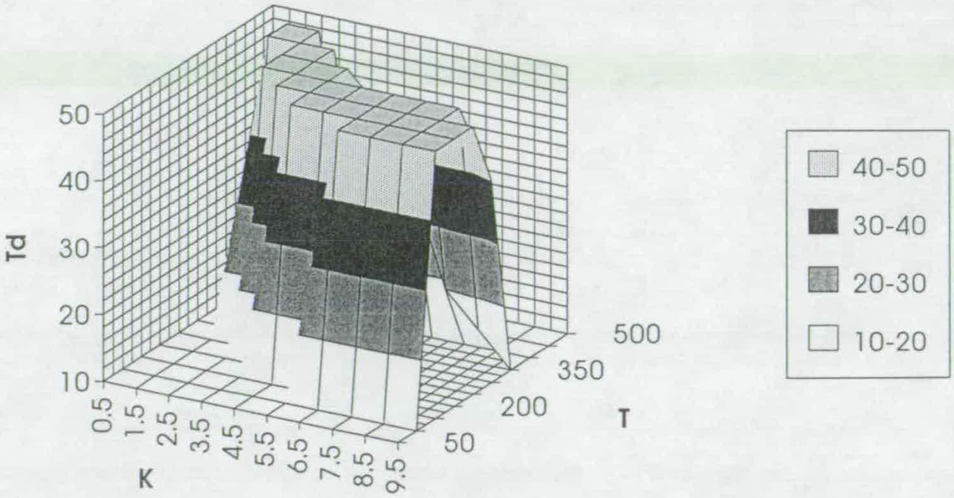


Fig.4 Robust space of PID in controlling Model 2 process

Model 3

Fuzzy logic control of high order processes requires the use of higher order derivative error information [6]. However, the high order derivative error information is difficult to obtain in practical engineering situations due to the typically noisy environment. So an FLC system performance when controlling a high order system is usually inferior compared with the first order case. On the other hand, there is no special problems for a PID controller.

Many experiments show that the poor performance of fuzzy logic control of high order systems is due to the delayed information input to the FLC. Therefore the slow response control method used with time-delay processes should be effective for controlling high order processes.

Experimental methods used for the model 2 process was applied to the model 3 process, and the parameters of the controllers tuned at K=4.5, T_1=250, T_2=100 (tuning point) are:

FLC: GE = 0.8; GC = 33; GU = 0.06.
PID: $K_p = 0.18$; $T_i = 330$; $K_d = 10$.

Table III and Table VI indicate the ranges of T_2 at each point of the $K \times T_1$ space. The T_2 ranges are the ranges within which the step response of the corresponding system is acceptable. The robust ranges of the FLC system covers 553 test points, while that of the PID system covers 315 points in 1000 test points. From the tables it can be found that for the same tuning-point performance condition, the FLC algorithm is more robust than the PID algorithm.

To test the effect of tuning-point selection on robust performance, two systems were tuned at K=1.5, T_1=100, T_2=100 to give the same response as above, and the controller parameters were GE=1.1, GC=22, GU=0.2, K_p=0.4, T_i=65, K_d=5. Acceptable performance was achieved at 305 and 74 test points by the FLC and PID algorithms respectively. So it is confirmed again that the selection of the tuning point affects the system's robust range and this effect is stronger in the PID system than in the FLC system.

The effect of performance at the tuning-point on the system's robust range has been tested. The two systems were re-tuned at K=4.5, T_1=250, T_2=100 to give a fast response (*i.e.* decrease the settling time from 1000 to 700), and the controller parameters were GE=0.78, GC=28, GU=0.08, K_p=0.21, T_i=260, K_d=5. Acceptable performance was achieved at 511 and 304 test points by the FLC and PID algorithms respectively. The difference is not large, but further investigation has shown that the changes of the controller's parameters lead to different positioning of the robust range in the plant parameter space. In the case of the model 2 process, the robust range moved out of the plant's parameter space when the controller parameters were changed, so the robust range within the plant' parameter space became smaller. While in the case of the model 3 process, because the shifting of the robust range was still in the plant's parameter space, there was no obvious change in the system's robust range.

TABLE III

Robust range of FLC system – Model 3 case

T_2 range		*K*									
		0.5	*1.5*	*2.5*	*3.5*	*4.5*	*5.5*	*6.5*	*7.5*	*8.5*	*9.5*
T_1	*50*				225	200 ~ 225	200 ~ 225	175 ~ 225	175 ~ 225	150 ~ 225	150 ~ 225
	100			20 ~ 225	175 ~ 225	150 ~ 225	125 ~ 225	125 ~ 225	125 ~ 225	100 ~ 225	100 ~ 225
	150		20 ~ 225	150 ~ 225	125 ~ 225	100 ~ 225	75 ~ 225	75 ~ 225	75 ~ 225	50 ~ 225	50 ~ 225
	200		150 ~ 225	10 ~ 225	75 ~ 225	50 ~ 225	50 ~ 225	25 ~ 225	25 ~ 225	25 ~ 200	25 ~ 175
	250		10 ~ 225	50 ~ 225	25 ~ 225	**25 ~ 225**	25 ~ 225	25 ~ 200	25 ~ 175	25 ~ 175	25 ~ 150
	300		50 ~ 225	25 ~ 225	25 ~ 225	25 ~ 200	25 ~ 175	25 ~ 175	25 ~ 150	25 ~ 150	25 ~ 125
	350		25 ~ 200	25 ~ 225	25 ~ 200	25 ~ 175	25 ~ 150	25 ~ 150	25 ~ 150	25 ~ 125	25 ~ 125
	400		25 ~ 175	25 ~ 200	25 ~ 175	25 ~ 150	25 ~ 150	25 ~ 125	25 ~ 125	25 ~ 125	25 ~ 100
	450		50 ~ 150	25 ~ 175	25 ~ 150	25 ~ 125	25 ~ 125	25 ~ 125	25 ~ 100	25 ~ 100	25 ~ 100
	500		50 ~ 125	25 ~ 125	25 ~ 125	25 ~ 125	25 ~ 100	25 ~ 100	25 ~ 100	25 ~ 100	25 ~ 100

TABLE VI

Robust range of PID system – Model 3 case

T_2 range		K									
		0.5	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5
T_1	50						200 ~ 225	175 ~ 225	150 ~ 225	150 ~ 225	125 ~ 225
	100					200 ~ 225	150 ~ 225	125 ~ 225	125 ~ 225	100 ~ 225	75 ~ 225
	150				200 ~ 225	150 ~ 225	100 ~ 225	75 ~ 225	200 ~ 225	200 ~ 225	25 ~ 175
	200			225 ~ 225	150 ~ 225	100 ~ 225	50 ~ 200	25 ~ 175	25 ~ 150	25 ~ 125	25 ~ 125
	250			175 ~ 225	100 ~ 225	**25 ~ 225**	25 ~ 150	25 ~ 125	25 ~ 125	25 ~ 100	25 ~ 75
	300			125 ~ 225	25 ~ 225	25 ~ 225	25 ~ 125	25 ~ 100	25 ~ 75	25 ~ 75	25 ~ 50
	350			75 ~ 225	25 ~ 175	25 ~ 225	25 ~ 100	25 ~ 75	25 ~ 50	25	25
	400		200 ~ 225	25 ~ 225	25 ~ 125	25 ~ 225	25 ~ 50	25			
	450		200 ~ 225	25 ~ 175	25 ~ 100	25 ~ 225					
	500		125 ~ 225	25	25 ~ 50						

4. CONCLUSION

This paper presents the results of a large number of simulation experiments quantifying the claim that fuzzy control leads to an improved robust performance with respect to the use of PID control in the case of SISO systems. The results of the simulation studies reveal several aspects of robustness in fuzzy control:

(i) The fuzzy control system is more robust than the PID control system in the SISO case.

(ii) The performance of the fuzzy logic control system is much less sensitive to the selection of the tuning-point than that of the PID control system.

(iii) The performance of the SISO fuzzy logic control system at the tuning point affects the position of the system's robust range in the process parameter space, but does not have an obvious effect on the size of the robust range.

Further work is required to investigate the effect of system non-linearity and the robust performance of MIMO systems,.

Acknowledgement

The authors would thank the British ORS organisation and the faculty of Science and Engineering, University of Edinburgh, UK, for their financial supports.

References

[1] G.M. Abdelnour, C.H.Chang etc., "Design of a fuzzy controller using input and output mapping factors", *IEEE trans. syst., man, cybern.*, vo.21, no.5, pp.952-960, 1991.

[2] C.W. Tao, R.Mamlook etc., "Reduction of complexity for a robust fuzzy controller", *Proc. of 2nd IEEE int. conf. on fuzzy syst,* vo.2, pp.1346-1349, 1993.
[3] M. Braae and D.A. Rutherford, "Theoretical and linguistic aspects of the fuzzy logic controller", *Automatica*, vo.15, pp.553-577, 1979.
[4] R. Sutton and D.R. Towill, "An introduction to the use of fuzzy sets in the implementation of control algorithms", *Journal of the institution of electronic and radio engineers*, vo.55, no.10, pp.357-367, 1985.
[5] V. Vance J Vandoren, The Challenges of Self-Tuning Control, *Control Engineering*, February 1994, pp.77-79.
[6] S. Boverie etc., Fuzzy Logic Control for High Order Systems, *Proc. of 2nd IEEE International Conference on Fuzzy Systems*, vo.1, pp.117-121, 1993.

The Design of Fuzzy Logic Controlled SISO Processes with Step response Constraints

J R Jordan and J Wang
Dept. of Electrical Engineering
University of Edinburgh
Edinburgh, EH9 3JL U.K.

Abstract

The results of a simulation based investigation of the step performance of fuzzy logic controlled proportional processes are presented. Conventional fuzzy logic control and a modified form, introducing the use of a proportional path in parallel with the fuzzy control output accumulator, are investigated. The controlled process was second order, with and without a time delay term. The phase advance provided by the modified controller clearly leads to a better step performance, especially when high order processes are controlled. Experimental results are presented showing that the modified controller provides a more robust performance with respect to variation of the parameters of the controlled process.

1 Introduction

Closed loop operation of non-integrating process requires the use of a controller with an integral term if a zero error condition is to be achieved. A PI controller uses a series integral term with a parallel proportional term. This parallel proportional path significantly reduces the impact of the integrator term on the stability of the closed loop system. A conventional fuzzy logic controller uses a series connected accumulator (integrator) to convert the control change information output from the defuzzification process into a signal that is used to control the process. This paper presents the results of an investigation into the use of a parallel term across the accumulator to improve the stability of fuzzy logic control when the controlled process is a high order system. The step response performance of the modified fuzzy logic controller is investigated and the results of a comparative study of the robustness of the responses of fuzzy and modified fuzzy control, with respect to variation of the parameters of the controlled process, are presented.

Section 2 of this paper introduces the modified fuzzy logic controller and section 3 describes the experimental system. The results presented in section 4 shows that when the controlled process has high order the system stability has been improved and the ability of the system to tolerate variations of process parameters is enhanced when the modified fuzzy controller is used.

2 The fuzzy logic controller

The basic configuration of a fuzzy control system is shown in Fig.1, where GE, GC and GU are scaling factors for input and output variables of the FLC. Generally, because the output of the FLC is CHANGE IN CONTROL ($\Delta u'$), an integrator (accumulator) is used after the FLC to convert the control change information output into a signal that is used to control the non-integrating process, *e.g.*

$$u = \int \Delta u' \, dt = GU \int \Delta u \, dt \tag{1}$$

or

$$u(k+1) = GU \sum_{i=0}^{i=k} \Delta u(i) \tag{2}$$

where Δu or $\Delta u(i)$ (i = 0,1, ... k) are the crisp output defined in the output space of the FLC. However, when the controlled process is a high order process, this integrating term introduce stability problems [1] [2].

By analogy with the PI controller the modified fuzzy controller uses a parallel proportional path to reduce the impact of the integrator term on the stability of the closed loop system. The structure of the modified fuzzy control system is illustrated in Fig. 2. To change the effect of either the proportional term or the integral term on the system performance, a proportional gain K_p is used in the proportional path and the output scaling factor GU is put into the integral path. Therefore the control input u is calculated as

$$u = K_p \, \Delta u + GU \int \Delta u \, dt \tag{3}$$

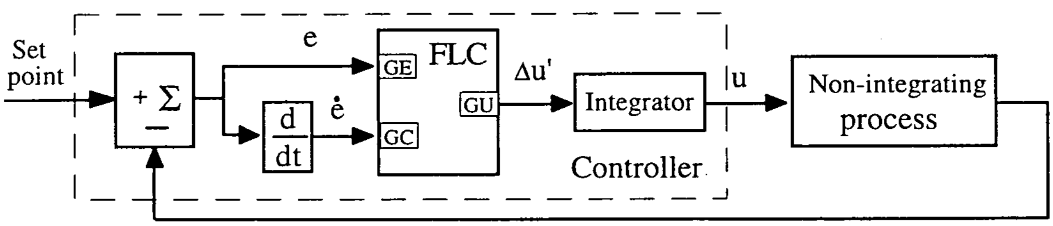


Fig.1 Basic structure of fuzzy logic control systems

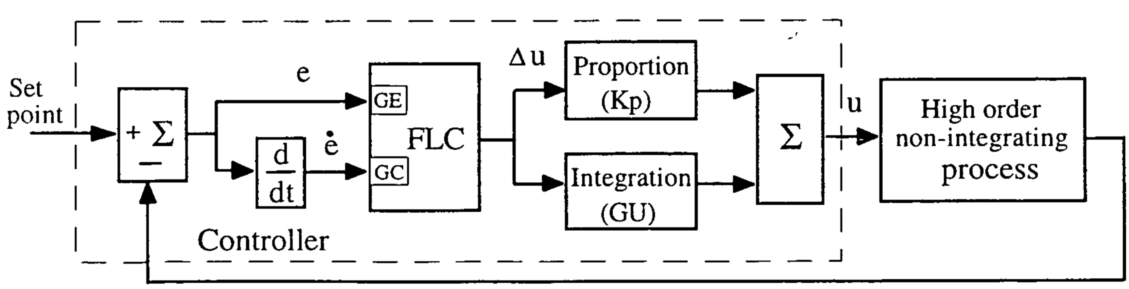


Fig. 2 The structure of the modified fuzzy control system.

or

$$u(k+1) = K_p \Delta u(k) + GU\sum_{i=0}^{i=k}\Delta u(i) \tag{4}$$

3 Experimental system

The systems shown in Fig. 1 and 2 were simulated using specially designed software package. Performance was evaluated by applying a step change to the set point input and observing the resulting output response of the control process.

The basic principle of the operation in the FLC controller is described as follows. Suppose there are N rules in fuzzy rule base. The change of control input Δu is calculated as:

R_i : if e is A_i and $\dot{e}$ is B_i, then Δu_i is C_i (i = 1, ······ N)

$$\alpha_i = \mu_{A_i}(e) \wedge \mu_{B_i}(\dot{e}) \tag{5}$$

$$\Delta\mu_i = cog(\alpha_i \wedge \mu_{C_i}) \tag{6}$$

$$\Delta u = \frac{\sum_{i=1}^{N}\alpha_i \Delta u_i}{\sum_{i=1}^{N}\alpha_i} \tag{7}$$

where cog(·) is the value obtained by defuzzifying with the centre-of-gravity method, A_i , B_i and C_i are fuzzy subsets of e, $\dot{e}$ and Δu respectively, and μ_{A_i}, μ_{B_i} and $\Delta\mu_i$ are the membership functions of A_i , B_i and C_i respectively.

Fig.3 and Fig.4 shows the rule base and membership functions used in the simulation experiment with the MFLC and FLC. Input and output variables use the same group of membership functions, and the range of each variable is normalised to [-10, 10] by applying the corresponding scaling factor. The rule base and membership functions are designed by the trial-and-error method. In the experiments, the rule base and membership functions are kept unchanged but scaling factors GE, GC, GU and the proportional gain K_p are tuned to obtain a more robust system performance because the scaling factors give strong effect on the performance of the FLC system [3].

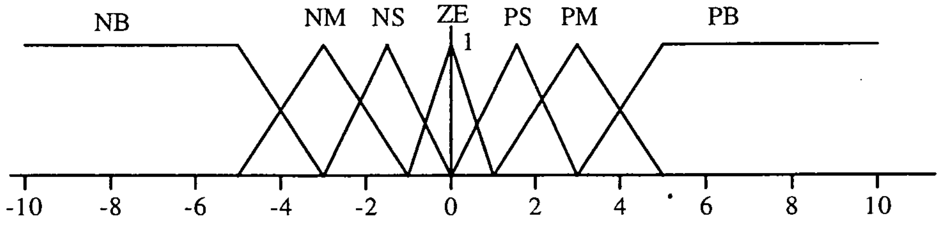


Fig. 3 Membership functions used in the FLC

		$\dot{e}$						
		NB	NM	NS	ZE	PS	PM	PB
	NB	NB	NB	NB	NB	NM	NS	ZE
	NM	NB	NB	NB	NS	NS	ZE	PS
	NS	NB	NB	NB	NS	ZE	PS	PM
e	ZE	NB	NM	NS	ZE	PS	PM	PM
	PS	NM	NS	ZE	PS	PM	PM	PM
	PM	NS	ZE	PS	PS	PM	PM	PM
	PB	ZE	PS	PM	PB	PB	PB	PB

Fig. 4 Rule base used in the FLC

Two kinds of high order processes were chosen to be controlled by the FLC and MFLC. Their transfer functions are :

$$G_1(s) = \frac{K}{(1+T_1 s)(1+T_2 s)}; \tag{8}$$

$$G_2(s) = \frac{K\,e^{-\tau s}}{(1+T_1 s)(1+T_2 s)}. \tag{9}$$

The process, $G_2(s)$, is a good approximation to many of the commonly found industrial processes.

The robustness of the performance of the closed loop systems was investigated by allowing the parameters of the controlled process to vary in the ranges shown below :

$K = 0.5$ to 9.5; $T_1 = 50$ to 500;
$T_2 = 0$ to 225; $\tau = 20, 50$.

Note that the unit for the time constants is the sampling interval in the process simulation, and this unit will be used for all time parameters in this paper.

Two parameters, overshoot and settling time, are selected as the important features of the step responses of the systems investigated. The system's performance is considered acceptable (for the purposes of this paper) providing that:

(1) Maximum overshoot $Y_o \le 5\%$;

(2) 2% settling time $T_s \le 5T_\Sigma$.

Note that the 2% settling time T_s denotes the time spent for the system output to get into the output range specified by 98% to 102% of the steady state value, and
T_Σ in criteria (2) denotes the sum of all time constants in the controlled process.

The process parameter space ($K \times T_1 \times T_2$ for process $G_1(s)$ or $K \times T_1 \times T_2 \times \tau$ for process $G_2(s)$) is investigated with given controller parameters GE, GC, GU, and K_p to determine the sub-space within which the system's performance is acceptable. This sub-space of the parameter space is called the **robust space** of the system.

The following experimental procedure was used:

Step 1: Select a common tuning point for the two systems, i.e. choose a middle value in each parameter variation range as the value of that parameter at the tuning point.

Step 2: Tune controller parameters of both FLC system and MFLC system to obtain a best step response (*i.e.* Y_o and T_s) at the tuning point for each system.

Step 3: Fix the parameters of both FLC and MFLC. Test the system performance at different points in the process parameter space. The test points are evenly located on the parameter space.

Step 4: Using the criteria defined above, determine the robust space for each system.

4 Results

In the following, the experiment results derived from each process are presented together with a brief discussion.

***Model 1*:** Scaling factors of both controllers are separately tuned at K = 4.5, T_1 = 250, T_2= 100 (the tuning point), and they are:

FLC:	GE = 0.75; GC = 26.0; GU = 0.09.
MFLC:	GE = 1.0; GC = 20.0; GU = 0.2; K_p = 5.0.

Each system's step response at the tuning point is shown in Fig. 5 where an obvious difference between two settling times can be found. It worth noting that if the response of the FLC system is made faster than that shown in the Fig.5, oscillations or a large overshoot will result.

After carrying out step 3 and step 4 in the procedure, percent overshoot and 2% settling time are collected and presented in the form of 3-D surface charts shown in Fig.6 and Fig.7. The robust space of each system lies between the upper and lower surfaces (see Fig.6 and Fig.7) defined in the $K \times T_1 \times T_2$ space. The robust space of the MFLC system and the FLC system take 811 and 505 test points in 1000 tests respectively. From these results it would appear that the second order process can be controlled by the MFLC with an improved robust space.

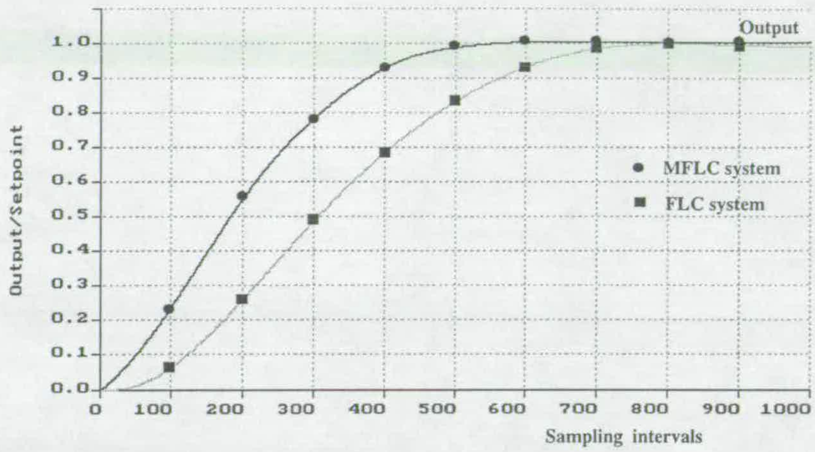


Fig. 5 Step responses of FLC and MFLC systems at the tuning point

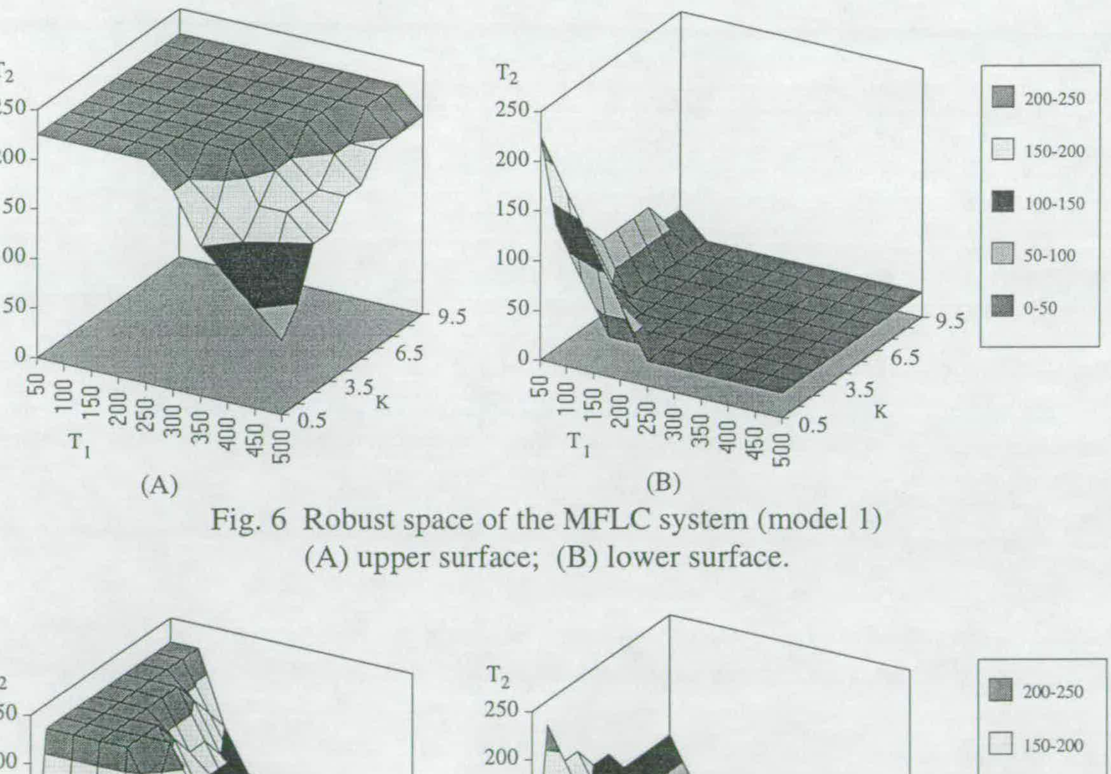


Fig. 6 Robust space of the MFLC system (model 1)
(A) upper surface; (B) lower surface.

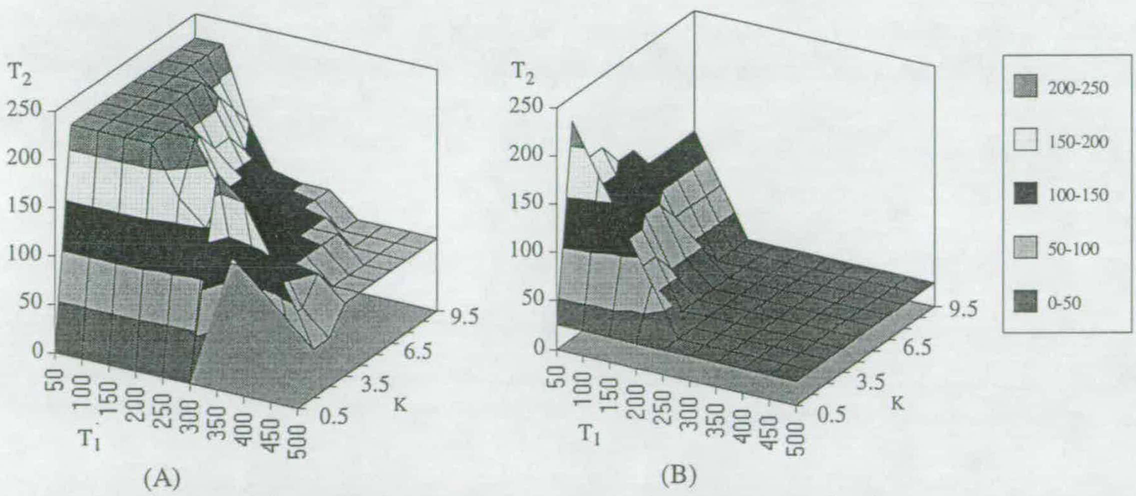


Fig. 7 Robust space of the FLC system (model 1)
(A) upper surface; (B) lower surface.

Model 2: Due to the pure time delay in its response to the control signal, this process is much more difficult to control than the first model process. In practical control engineering, a controller tuned to handle process dead time tends to respond rather slowly while attempting to correct a deviation in the process output [4]. Although the process dead time is simple to estimate from a reaction curve, it is much harder to identify automatically during normal loop operations when the process input and the process output vary simultaneously, or noise and load disturbances can not be neglected. Hence more complicated control algorithms requiring this information are difficult to use in practice.

From the practical engineering point of view, the slow-control method is adopted and the dead time is assumed unknown in the simulation. Scaling factors of both controllers are separately tuned at $K = 4.5$, $T_1 = 250$, $T_2 = 100$ and $\tau = 20$ (tuning point), and they are:

FLC: GE = 0.6; GC = 23.0; GU = 0.07.
MFLC: GE = 0.5; GC = 20.0; GU = 0.1; $K_p = 2.0$.

Each system's step response at the tuning point is shown in Fig. 8 from which no obvious difference can be found. but if the response of FLC system is made faster than that shown in the Fig.8, oscillations or a large overshoot will result.

The robustness of each system is tested at $\tau = 20$ and the robust space of each system lies between the upper and lower surfaces (see Fig.9 and Fig.10) defined in the $K \times T_1 \times T_2$ space. The robust space of the MFLC system and the FLC system take 680 and 465 test points in 1000 tests respectively. A similar result can be found if $\tau = 50$ where in this case the robust space of the MFLC system and the FLC system take 643 and 295 test points respectively in 1000 tests. From these results it can be concluded that the second order process with pure time delay can be controlled by the MFLC with an improved robust space.

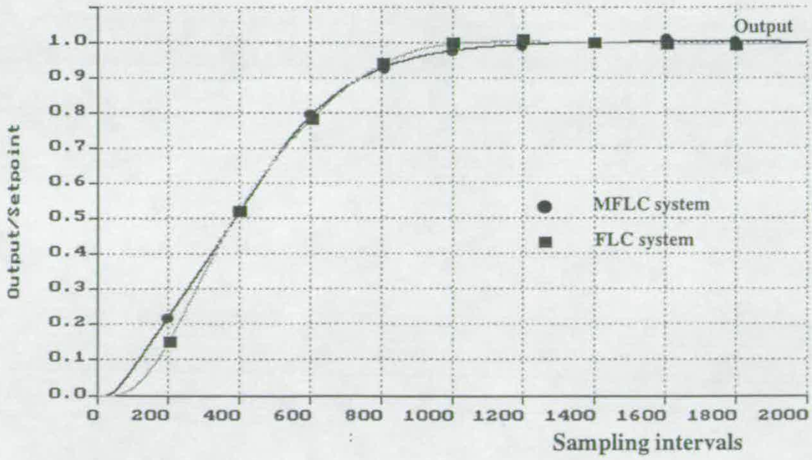
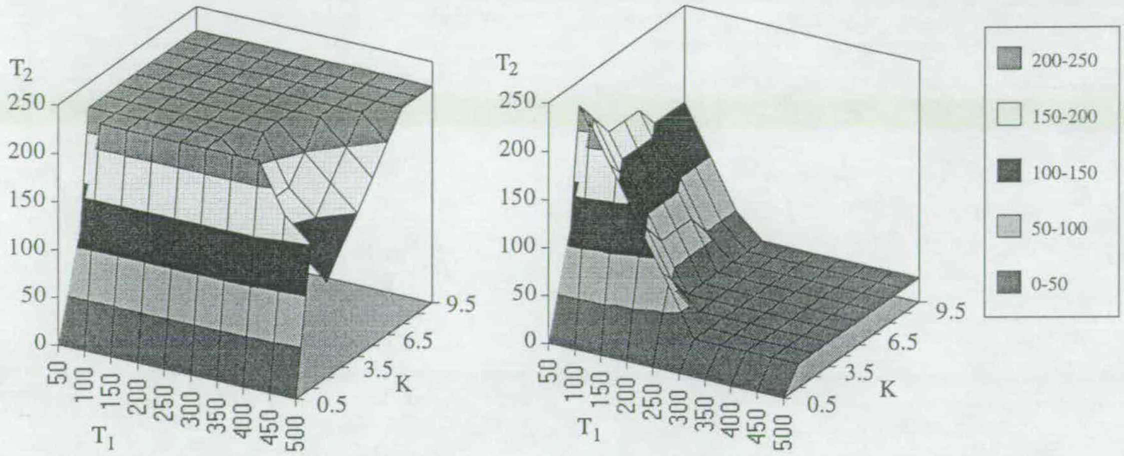
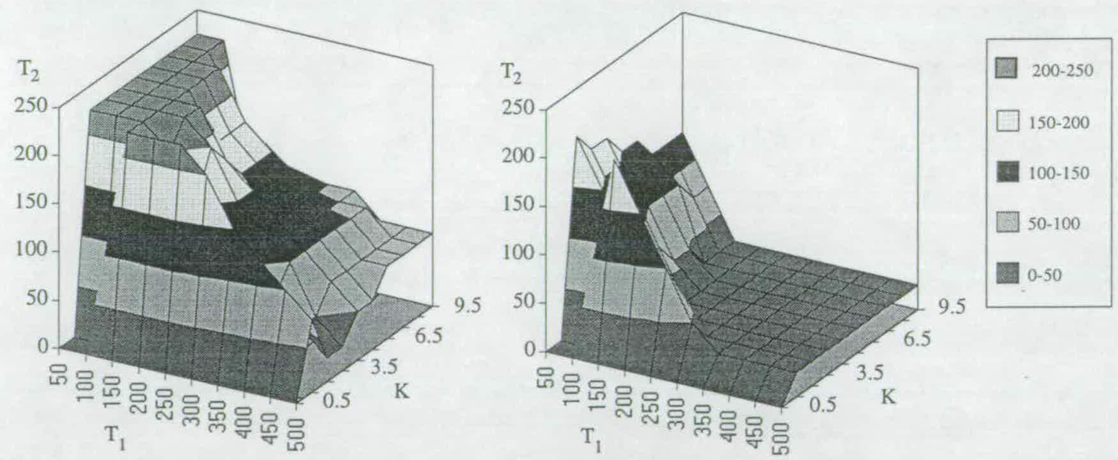


Fig. 8 Step responses of FLC and MFLC systems at the tuning point



(A) (B)

Fig. 9 Robust space of the MFLC system (model 2)
(A) upper surface; (B) lower surface.



(A) (B)

Fig. 10 Robust space of the FLC system (model 2)
(A) upper surface; (B) lower surface.

It is worth noting that all parameters of both the FLC and the MFLC, such as scaling factors, rule base and the membership functions, affect the robustness of the system [3]. This is the main reason for the confusion that often arises when the performance of a FLC system is compared with its traditional counterparts. Also the multi-variable nature of the tuning requirement of FLC systems makes their optimisation difficult and time consuming. Fortunately, the requirements of practical engineering can be generally met by a non-optimised, or sub-optimised FLC system.

5. Conclusion

This paper has described the results of a simulation based investigation of the step performance of fuzzy logic controlled processes. Two processes were investigated: namely processes having proportional second order transfer function with and without a time delay term. Closed loop control was implemented using a conventional fuzzy control algorithm and a modified algorithm that introduced a proportional term in parallel with the accumulator used to implement a series integral term. The acceptable step response was defined by a maximum overshoot less than or equal to 5% with a 2% settling time less than or equal to 5 times the sum of all time constants in the controlled process. The closed loop system was tuned to give an acceptable step response and the process parameters varied until an unacceptable step response was obtained.

It was found that the modified fuzzy control algorithm could be applied to give a faster monotonic step response than can be achieved with the conventional algorithm. In addition for the process examples investigated a significantly improved range of acceptable performance was obtained as the parameters of the controlled process were varied over a wide range of values. Previous papers [2,3] have demonstrated that conventional fuzzy control has an improved robust performance compared with the standard PID control algorithm. The improved performance of the modified fuzzy control algorithm is particularly apparent when it is used to control a high order process.

6 References

[1] S. Boverie, B. Demaya, J. Lequellec and A. Titli, Fuzzy Logic Control for High Order Systems, *Proc. of 2nd IEEE International Conference on Fuzzy Systems*, vo.1, pp.117-121, 1993.

[2] J. Wang and J.R. Jordan, The Robust Step Performance of PID and Fuzzy Logic Controlled SISO Systems, submitted for presentation to The International Joint Conference of the Fourth IEEE International Conference of Fuzzy Systems and the Second International Fuzzy Engineering Symposium, Japen, March 20-24, 1995.

[3] J. Wang and J.R. Jordan, "An Experimental investigation of the dynamic performance of fuzzy logic control," *Trans. of the Institute of Measurement and Control*, vo.16, No.3, pp.142-148, 1994.

[4] V. Vance and J. Vandoren, The Challenges of Self-Tuning Control, *Control Engineering*, February 1994, pp.77-79.

AUTO-TUNING ALGORITHM FOR FUZZY CONTROL SYSTEMS

Jianzhou WANG, Jim R. JORDAN

University of Edinburgh, Edinburgh, EH9 3JL UK

Abstract. The lack of a general methodology for parameter tuning of fuzzy controllers is a major drawback to a more widespread use of the fuzzy control technique. In this paper, a tuning algorithm for the fuzzy controllers is proposed which features on-line performance, uses fuzzy logic algorithms and does not require detailed information about the process to be controlled. Based on exhaustive simulation experiments, the fuzzy tuning rules for scaling factors and sample rate are presented. A complete tuning procedure has been designed. The adequacy of this tuning algorithm has been tested by simulations carried out on a second order process with time delay and other SISO processes. Results show that the fuzzy tuning algorithm is effective and can save time and effort in tuning fuzzy controllers.

Keyword. auto-tuning, fuzzy logic control, scaling factor.

1. Introduction

A great number of parameters define the performance of a fuzzy controller. Hence its tuning at most times becomes a critical operation which can provide unsatisfactory system results. The lack of a general methodology for parameter tuning of fuzzy controllers is a major drawback to a more widespread use of this control technique. For this reason, many researchers have addressed their efforts to finding effective tuning methods during the last few years. Adaptive self-tuning[1], the evolution method[2,3], the Rosenbrok identification method, the gradient descent method[4], and the correlation function method[5], have been studied to cope with the tuning difficulty of the fuzzy controller. Most of these methods are successful and valuable for the practical design of fuzzy systems. However they all required models of the process to be controlled. In this paper, a tuning method is proposed, which features on-line performance, uses fuzzy logic algorithms and does not require detailed information about the process to be controlled.

Among the parameters which affect the performance of fuzzy systems, intuitively it will be most effective to change the performance of the fuzzy control system by tuning the fuzzy rules and/or the membership functions of the fuzzy subsets related to the input output fuzzy variables of the controller. But it is very complicated to perform an on-line tuning because of the multi-variable feature of the controller and the typically high non-linearity of the process. Nevertheless the system performance can be changed by simply altering the input-output scaling, thus changing the membership grades of the input-output variables belonging to the fuzzy subsets which the fuzzy rules are closely related to. Experimentally tuning the fuzzy control system in this way, it is possible to obtain the expert knowledge about the relationship between the scaling factors and the system's performance, and therefore to perform the auto-tuning of a fuzzy system by using a fuzzy algorithm.

The aim of the research presented in this paper is to automatically tune the scaling factors with the help of fuzzy algorithms to obtain a relatively satisfactory system performance when the fuzzy system is initially started. The fuzzy tuning algorithm and the procedure is presented in section 2. In section 3, the effectiveness of the tuning algorithm is assessed by application to a simulated fuzzy control system. The paper is concluded in section 4. In this paper, attention will be restricted to single-input single-output processes.

2. Method

2.1 Tuning Algorithm

In most technical applications, fuzzy controllers receive crisp inputs and have to give a crisp control output. Fig.1 shows the general structure of fuzzy control systems. In this typical case, the operation of a fuzzy controller requires fuzzification of the inputs and defuzzification of the control outputs. To do this, each crisp variable is attached to a fuzzy set defined in the universe of discourse by means of membership functions. For simplicity, in most cases the membership functions are defined within a normalised interval. Therefore, the crisp variables have to be scaled (normalised) by multiplication with scaling factors, so that they fit into the normalised universe of discourse.

Note that the optimal scaling not only depends on the variable's properties but also on the shape and position of the membership functions used and the dynamics of the plant to be controlled. The importance of a suitable choice of scaling factors is clearly shown by the fact that poorly chosen scaling factors will result in the

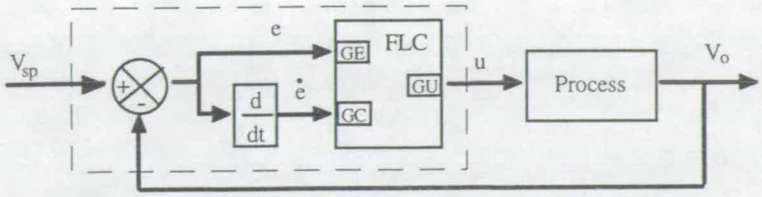


Fig.1 General structure of fuzzy control systems.
GE, GC and GU are scaling factors of the FLC.

shifting of the operating area in parameter space to the boundaries of the normalised universe of discourse. Paper [5,6,7] have shown the effects of scaling factors on the dynamic performance of fuzzy control systems.

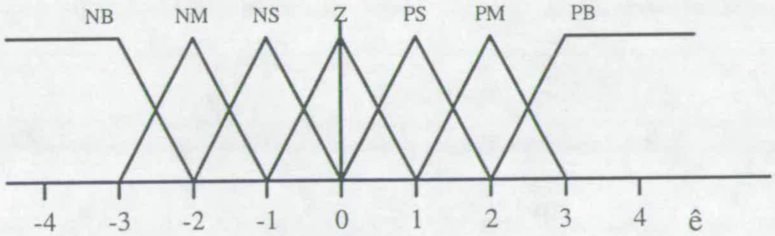
Generally, for a given fuzzy controller, increasing the scaling factor is equivalent to compressing the corresponding membership functions towards the origin of the related axis. On other hand, proportionally increasing the support of each fuzzy subset related to a fuzzy variable in both side of the origin is equivalent to decreasing the scaling factor of that fuzzy variable. For example, membership functions for error $\mu(e)$ in Fig.2(a) with scaling factor GE=2, are equivalent to those shown in Fig.2(b) if GE=1; *i.e.* if *error* = 1.0, after scaling, it will be PM with membership grade 1.0 in both situations shown in Fig.2. Therefore if the membership functions are fixed, the bigger the scaling factor of one variable, the more sensitive will the controller be to the small values of this variable, and less sensitive to big values of this variable.

To perform the auto-tuning of scale factors, the relationship, *i.e.* fuzzy rules for tuning, between the scale factors and the system's performance is essential and may vary in different system configurations. Also, before tuning is performed, the current system performance is required. To obtain information about the performance of the system to be tuned, an effective driving signal has to be carefully selected. In the following, the variables in the tuning rules will first be defined. Then rules for auto-tuning the scale factors in different situations will be discussed. Finally the tuning procedure related to the driving signal will be designed.

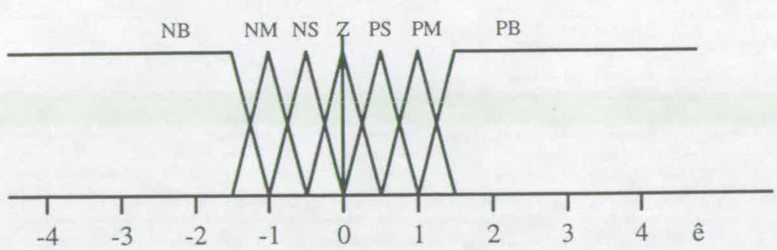
2.2 Tuning variables

Due to their simplicity in practical control engineering, the overshoot (OS), undershoot (US), number of ring cycles (CY), and integrated error square (IES) of the controlled process response to a step input have been selected as the performance indicators and thus the main input variables. According to the common practical specifications of a control system, the membership functions for these performance measurements are defined in Fig.3 (a). Note that IES is normalised as follows:

$$IES = \frac{1}{M^2}\sum e^2(k) \qquad (1)$$



(a) GE = 2



(b) GE = 1

Fig.2 Equivalent membership functions (ê is the scaled error) where M is the magnitude of the step input, *i.e.* the set point change, $e(k)$ is the system error at instance k.

The output variables of the tuning rules are the scale factors GE, GC and GU. Membership functions for these scale factors, as shown in Fig.3(b), were designed with the help of design experience in fuzzy control systems. Fig.3(c) shows the membership functions (singleton type) of changes in the scale factors with the horizontal scale giving the required multiplier.

2.3 Classification of systems

To generate the tuning rules for the fuzzy control system, a classification of control systems is required based on significant performance features. From the results of simulation experiments and practical knowledge of process structure, control systems can be divided into three groups: (1) simple controllable systems, such as a low order linear system without time delay; (2) complicated controllable systems; (3) uncontrollable systems. The first two situations will be considered in this paper. A system is classified as either a simple or a complicated controllable system by noting features of the error change when a step input with fixed amplitude is applied to the open-loop system. If the error change decreases monotonically within several sample periods, the system is a simple controllable system. For example, any first order linear process demonstrates this feature. Other systems will be sorted as a complicated controllable system, such as any high order systems. A time delay system is put into the complicated category. Fig.4 illustrates the classification method.

In Fig.4, controlled processes are sorted according to the initial response of the system to a step input. Due to insufficient information about the controlled process, at this stage, the system is only sorted as either a simple

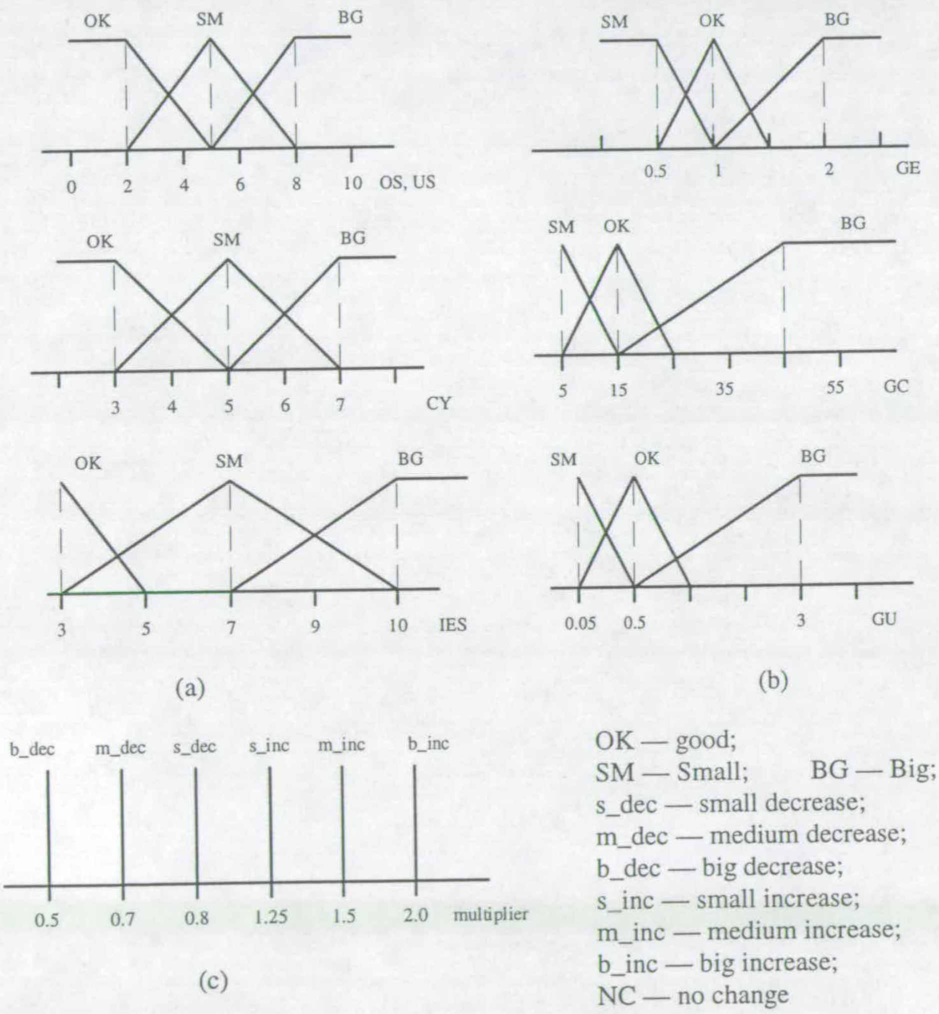


Fig. 3 Membership functions (MFs) of variables in the tuning rules
(a) MFs for system performance measurements;
(b) MFs for scale factors;
(c) MFs for the changes in scale factors.

or a complicated system. As more feedback information becomes available from the system's output and by evaluating the status of undershoot, overshoot and number of ring cycles generated by the system when controlled by a default fuzzy controller, the system will be further identified as oscillating, non-oscillating, or uncontrollable. In both oscillating and non-oscillating situations, performance will be further classified with respect to the status of overshoot, undershoot and integrated error square. Each status of the response is named as "Onn" (nn is a number here) in Fig.4. This classification has actually defined each of the fuzzy subsets on which the tuning rules are based and will be discussed next.

2.4 Tuning rules

By using *trial and error* and simulation methods, tuning rules are generated at each different status of the system performance. Exhaustive simulation tests have been carried out to find the proper action taken to modify the scaling factors and sample rate T_s at different situations. Table 1 and Table 2 give these heuristic tuning rules for both simple and complicated controllable systems respectively. Symbols in the tables come from the definitions in Fig.4. From these tables and Fig.4 , a rule, say rule No.10 for simple systems, can be read as:

IF *number of ring cycle* **IS** *good,*
AND *overshoot* **IS** *good,*
AND *undershoot* **IS** good,
AND *integrated error square* **IS** big
THEN *small increase GE,*
big decrease GC,
small increase GU.

Note that in contrast with tuning a complicated system, sample rate in a simple system is not changed while tuning the scaling factors. In fact, sample rate is calculated at the begin of the tuning procedure when the first feedback information is available. The method to determine the sample rate is empirically designed as:

$$T_s = \begin{cases} 0.024 / \dot{E}, & \text{for simple system;} \\ 0.012 / \dot{E}, & \text{for complicated systems.} \end{cases} \quad (2)$$

where $\dot{E}$ is the first non-zero sample of error change of the system. This method leads to a reasonable performance in simple systems and may lead to oscillation in complicated systems. Therefore tuning the sample rate is necessary in the complicated system situation.

2.4 Tuning procedure

Before applying the above tuning rules, information about the performance of the control system is needed. To excite the controlled process to show up its properties in the response to a step input without the system generating unacceptable signal changes, the

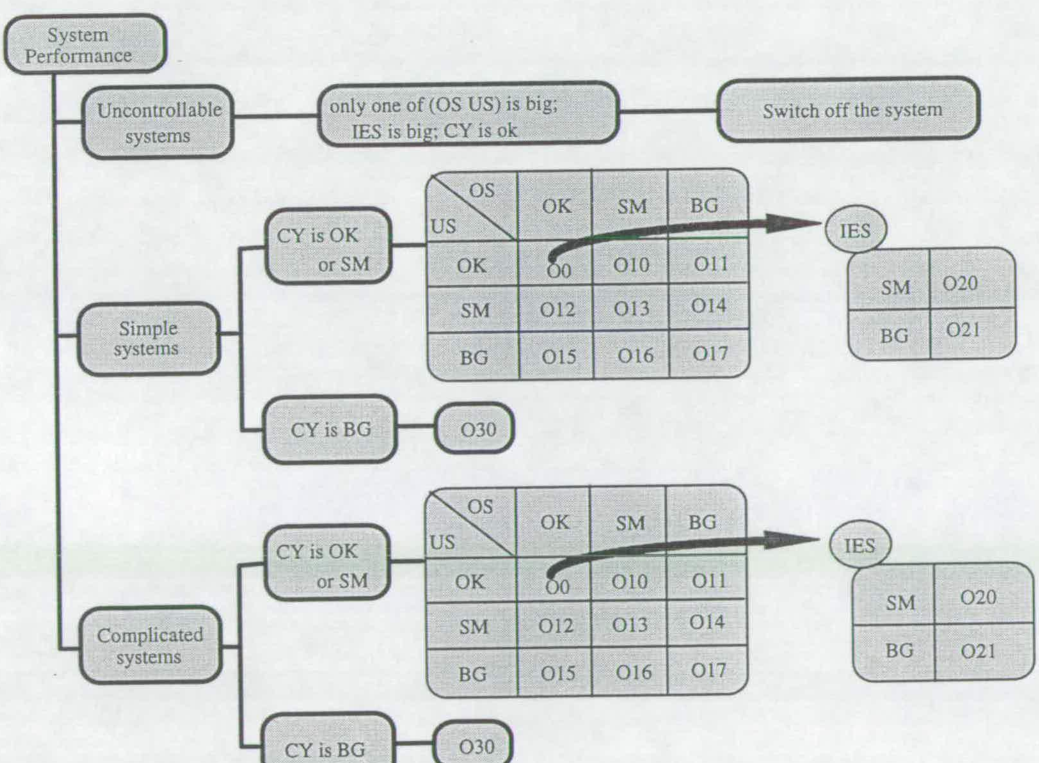


Fig.4 Classification of control systems and system's performance

selection of the control signal in the tuning stage becomes crucial. In this paper, the following methods have been used for safe starting:

1. Closed-loop fuzzy logic control is used. Because fuzzy logic control can be set-up to give a very gentle action, though the response may be too slow, the safety of the control system can be guaranteed. After more information about the system has been collected, the fuzzy controller parameters can be tuned to improve the performance.

2. A fraction of the required setpoint change of the tuned system was chosen to be the practical setpoint change at the tuning stage. This is used to prevent at most circumstances the process from overdriving. Also this method will lead to more than one tuning procedure in the tuning stage, and therefore gives more chance for the tuning algorithm to improve its tuning performance.

Fig.5 illustrates the tuning procedure. From Fig.5, the process control input change is kept at zero for an initial period of 100 sample intervals which are initialised to give a relatively fast sampling rate. During this period (phase 1) the system output $y(t)$ is monitored to check for plant noise or disturbances which will be used for the calculation of the system status. In phase 2, the system is configured as an open-loop to measure the process time delay and set the sample rate. The control input of the process is step increased with a fixed amplitude of 10% of the total setpoint change introduced by the tuning procedure. The system dead time T_d is considered as the time period from the beginning of phase 2 to the time when the output of the process changed by 2%. As the system output begins to increase, 10 samples of process output are taken and used with the dead time T_d to set the sample rate T_s and the initial setting of the parameters

Table 1
Rules for tuning simple controllable fuzzy systems

Rule No.	System status	Other Conditions	GE action	GC action	GU action
1	O10			s_inc	
2	O11			b_inc	
3	O12			s_dec	b_dec
4	O13			s_inc	s_inc
5	O14			b_inc	
6	O15		s_dec		b_dec
7	O16		s_dec	b_dec	
8	O17			s_inc	b_inc
9	O20			s_dec	s_inc
10	O21		s_inc	b_dec	s_inc
11	O30	GC is BG	m_dec	b_dec	
12	O30	GU is BG			b_dec
13	O30	GC is OK GU is OK	m_dec		

Table 2
Rules for complicated controllable fuzzy systems

Rule No.	System status	Other Conditions	T_s Action	GE action	GC action	GU action
14	O10					
15	O11			s_dec	m_inc	m_dec
16	O12				s_inc	s_dec
17	O13				s_inc	
18	O14			s_dec	m_dec	m_dec
19	O15			s_dec	s_inc	m_dec
20	O16			s_dec	m_dec	m_dec
21	O17			s_dec		m_dec
22	O20		s_dec	s_inc		s_inc
23	O21		s_dec	m_inc	b_dec	m_inc
24	O30	IES is SM	m_inc	s_inc		m_dec
25	O30	IES is BG		m_inc		b_dec

of the controller. Next in phase 3, the system is configured as a closed loop system and the control task is handed on to the fuzzy controller with a setpoint step equal to 50% of the total setpoint change. As the system output progresses to the steady state, the system performance will be calculated, heuristic tuning rules will be applied and FLC scale factors will be tuned. After that, the setpoint will be further increased and a new tuning process begins (phase 4). After the tuning stage, the FLC with fixed scale factors will take the control task (phase 5).

If the system loses control during the tuning stage, the tuning algorithm will switch off the control input and bring the system to a safe stop condition. Under some circumstance, for example the cart-pole system, this method of dealing with unstable processes may fail, but it will lead to a safe stop in most practical situations.

3. Simulation and Results

To determine its adequacy, the fuzzy tuning algorithm is applied to the FLC control systems. Two common processes and a non-linear process are tested by simulations. The transfer functions of these three process are as follows:

$$\text{I:} \quad G(s) = \frac{Ke^{-T_d s}}{1+Ts} \tag{3}$$

$$\text{II:} \quad G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} Ke^{-T_d s} \tag{4}$$

$$\text{III:} \quad \dot{y}(t) = \frac{1}{T}\left(\frac{1-e^{-y(t)}}{1+e^{-y(t)}} + Ku(t)\right) \tag{5}$$

where K is the static gain, T is time constant, T_d is time delay, ω_n is the natural frequency, ζ is the damping ratio, $u(t)$ and $y(t)$ are input and output of the process respectively.

The process parameters are:

Process I: $K = 2.5,\ T = 400,\ T_d = 200$;

Process II: $K = 2.5,\ T_d = 200$,

$\omega_n = 6.0;\ \xi = 0.8$;.

Process III: $K = 2.5,\ T = 400$.

Note that the unit for the time variables, such as T and T_d, is the simulation time scale in the computer.

The fuzzy logic controller uses error $e(t)$ and change of error $\dot{e}(t)$ as its inputs and change of control as its output. $e(t)$ and $\dot{e}(t)$ are calculated as follows:

$$e(k) = V_{sp}(k) - V_o(k) \tag{6}$$

$$\dot{e}(k) = e(k) - e(k-1) \tag{7}$$

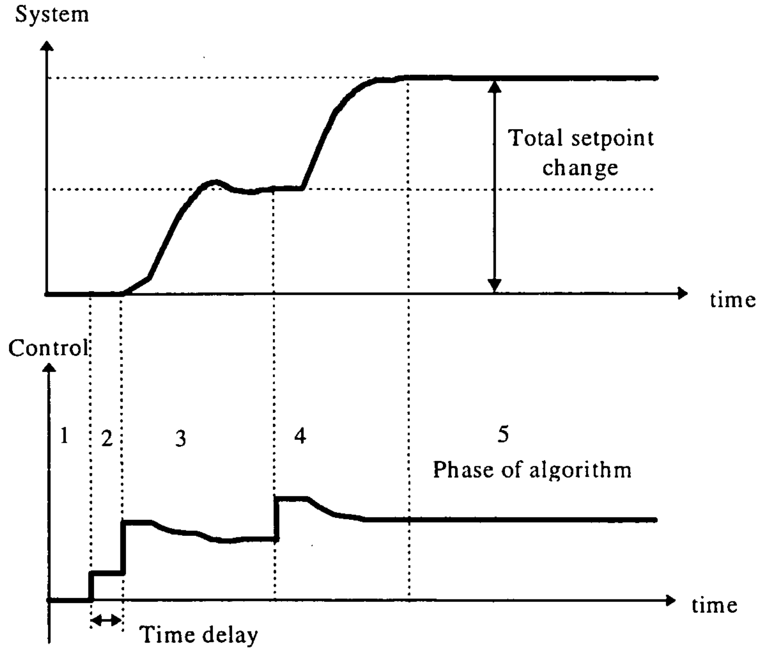


Fig.5 The auto-tuning procedure

The membership functions for the input and output variables, and the rule base are shown in Fig.6. The scaling factors and the sample rate of the FLC are initialised as:

$$GE = 1.0; GC = 10.0; GU = 0.05; T_s = 10.$$

Note that to make the controller give a gentle control, small GU is chosen. GE, GC and T_s are chosen empirically by using the approximate knowledge of the controlled processes.

After auto-tuning, the FLC parameters are shown in Table 3. Fig.7, Fig.8 and Fig.9 illustrate the system performances of process I, II and III respectively. Each figure shows the step response of the system before, during and after the tuning procedure.

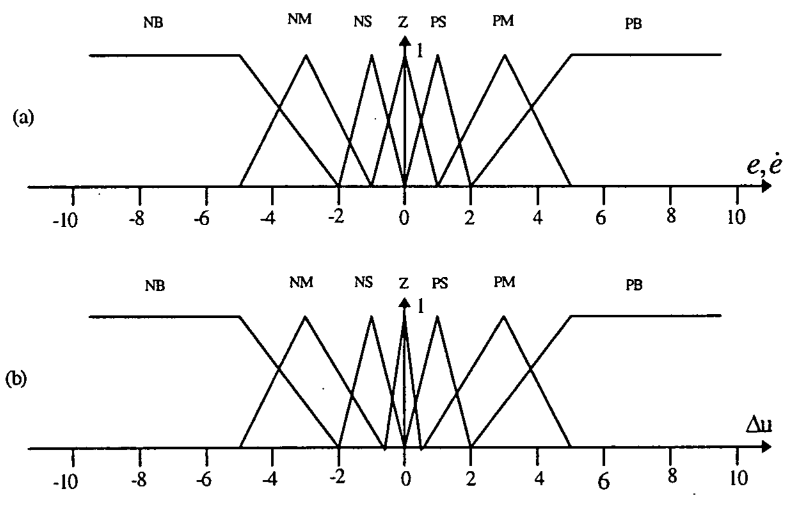
Table 3
FLC parameters after tuning

Process	GE	GC	GU	T_s
I	0.5	12.0	0.075	16
II	0.625	2.94	0.031	19
III	0.8	17.56	0.15	16

From Fig.7 to Fig.9, it is clear that without tuning, a generally designed fuzzy logic controller often gives an unacceptable performance even if it is used to control a simple process like process I. This is easy to understand because it is common practice to tune a control system to meet the designed specifications. This poor performance can be explained as the poor mapping of a controller input and output variables from the practical variable spaces to the designed ones.

The performances of the tuned systems shown in the figures appear to be unsatisfactory during the tuning stage, because the output of the system stopped rising in the middle of the tuning procedure. This is because two tuning procedures are performed for each system and each procedure needs a complete system response before carrying out the tuning task. It is found that for a better performance, two or more tuning procedures are generally needed if the controlled process is a complicated one.

It also clear that an oscillatory system performed much better after applying the proposed fuzzy tuning algorithm. And from Table 3, it can be seen that the scaling factors and sample rate were changed significantly by the tuning procedure. Though the parameters may not be optimised, for example, the step response in Fig.7(C) could be faster, the system's responses are acceptable from the engineering point of



(c)

e \ ė	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NB	NS	NS	ZE	PS
NS	NB	NB	NB	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PM
PS	NM	NS	ZE	PS	PM	PM	PM
PM	NS	ZE	PS	PS	PM	PM	PM
PB	ZE	PS	PM	PB	PB	PB	PB

Fig.6 FLC parameters. (a) input membership functions; (b) output membership functions; (c) Rule base.

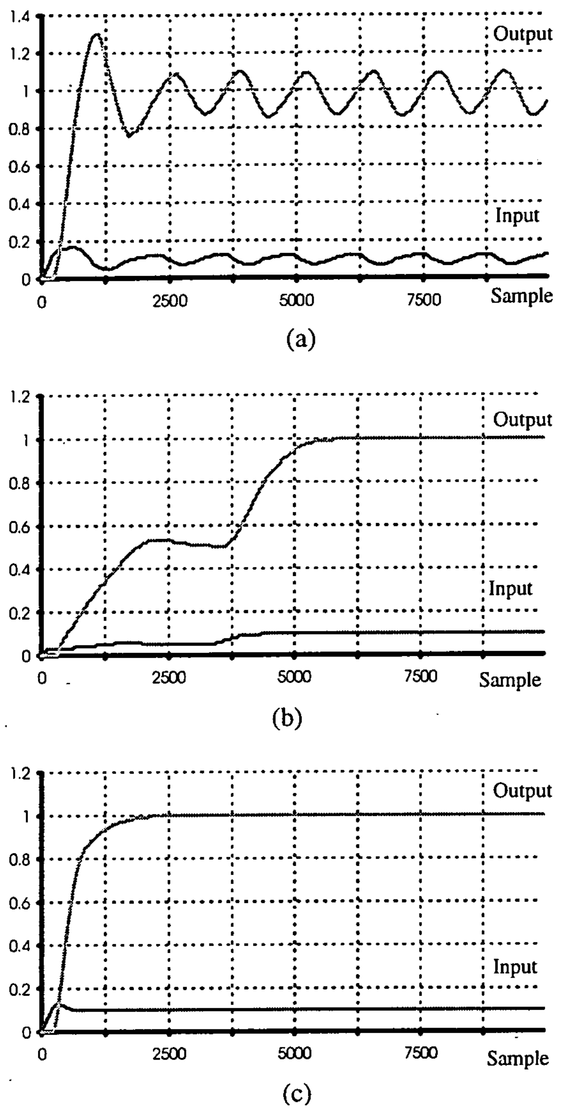


Fig.7 Step response of system I before (a), during (b), and after (c) tuning.

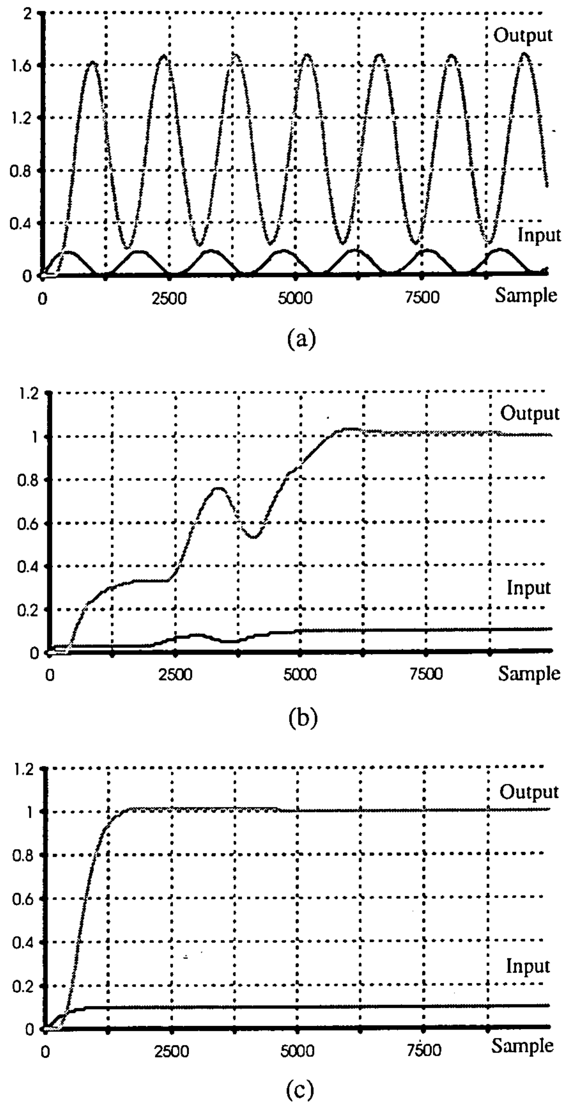


Fig.8 Step response of system II before (a), during (b), and after (c) tuning.

view.

It is worth noting that the empirical nature of this fuzzy tuning algorithm make it quite difficult to generate the tuning rules and define the membership functions of each variables. Hence the tuning rule base discussed above may not cover some tuning situations of the fuzzy control systems. Further work is needed to test and improve this algorithm.

4. Conclusion

A great number of parameters define the performance of a fuzzy control system. Hence its tuning at most times becomes a critical and very time consuming operation which can provide unsatisfactory system results. In this paper, a fuzzy tuning algorithm for the fuzzy controllers is proposed. Based on extensive simulation results, the fuzzy tuning rules for scaling factors and the sample rate are presented. A complete tuning procedure is designed. The adequacy of this tuning algorithm has been tested by simulations carried out on processes which are commonly used as the approximation of the practical processes and an artificially designed non-linear processes used in [8]. Results show that the fuzzy tuning algorithm is effective and quite robust with respect to changes of the controlled processes. The system's performance after the fuzzy tuning was quite acceptable. The system's safety was also guaranteed during the tuning stage by the specially designed default fuzzy control method. Experiments show that the time and effort expended in tuning the fuzzy controllers were significantly saved. Further work is needed to confirm its effectiveness in other tuning situations of fuzzy control systems.

References

1. Kiriakidis *and* A. Tzes, "Application of Implicit Self Tuning Fuzzy Control to Non-linear Systems", *Proceedings of FUZZ-IEEE/IFES'95*, Vol. 3, pp.1419-1426. Yokohama, Japan, March 1995.

2. F. Bolata *and* A. Nowè, "From Fuzzy Linguistic Specifications to fuzzy Controller Using Evolution Strategies", *Proceedings of FUZZ-IEEE/IFES'95*, Vol. 3, pp.1089-1094. Yokohama, Japan. March 1995.

3. C. Cavalcante, J. Cardoso, J.J.G. Ramos and O.R. Neves, "Design and Tuning of a Helicopter Fuzzy Controller", *Proceedings of FUZZ-IEEE/IFES'95*, vol.3, pp.1549-1554. Yokohama, Japan, March 1995.

4. B. Demaya, R. Palm, S. Boverie and A. Titli,. "Multilevel Qualitative and Numerical Optimisation of Fuzzy Controller", *Proceedings of FUZZ-IEEE/IFES'95*, Vol. 3, pp.1149-1154. Yokohama, Japan, March 1995.

5. R. Palm, "Scaling of Fuzzy Controllers Using the Cross-Correlation", *IEEE Transactions on Fuzzy Systems*, Vol..3, No.1, pp.116-123, 1995.

6. J. Wang and J.R. Jordan, "An Experimental Investigation of the Dynamic Performance of Fuzzy Logic Control," *Transactions of the Institute of Measurement and Control*, Vol.16, No. 3, pp.142-148, 1994.

7. T. J. Procyk and E. H. Mandani, "A Linguistic Self-Orgnizing Process Controller," *Automatica*, Vol. 15, pp.15-30, 1979.

8. L.X Wang, "Stable Adaptive Fuzzy Control of Nonlinear Systems," *IEEE Transactions of Fuzzy Systems*, Vol. 1, No. 2, pp.146-155. May 1993.

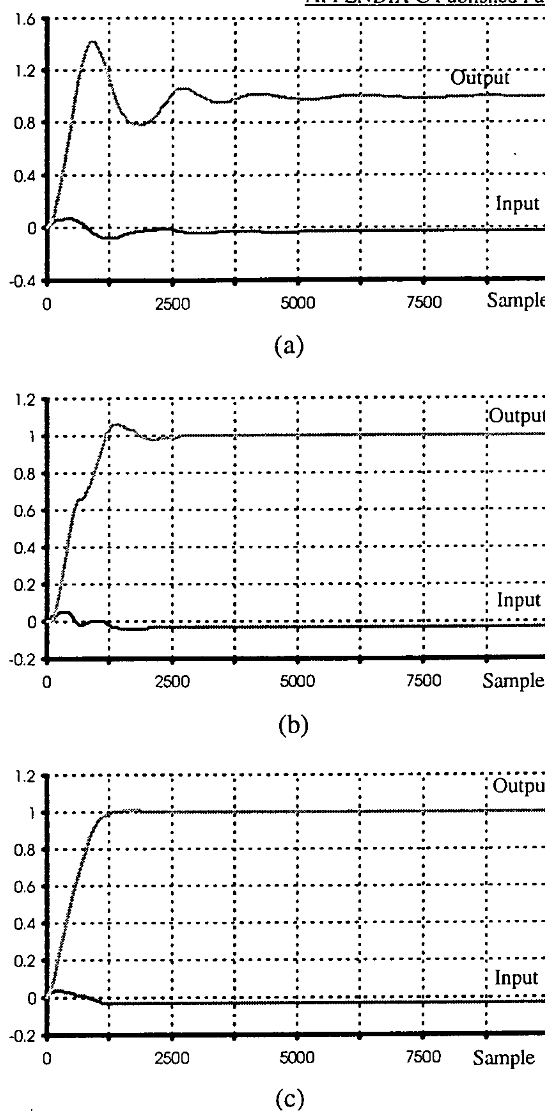


Fig.9 Step response of system III before (a), during (b), and after (c) tuning.