To the Graduate Council:

I am submitting herewith a thesis written by Brian Fortenbery entitled " Fuzzy Logic Control: a Tutorial View ." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Walter L. Green

W.L. Green, Major Professor

We have read this thesis
and recommend its acceptance:

J M Bailey

Marshall Pace

Accepted for the Council:

Carolyn Minkel

Associate Vice Chancellor and
Dean of the Graduate School

FUZZY LOGIC CONTROL: A TUTORIAL VIEW

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Brian Fortenbery
December 1995

DEDICATION

To my parents, who devoted their lives to their children, and my wife Judy who has given me constant support and encouragement.

ACKNOWLEDGEMENTS

I would like to thank my major advisor Dr. Walter Green for his endless patience and guidance. I would also like to thank my other committee members Dr. M.O. Pace and Dr. J.M Bailey for their help and suggestions.

In addition, I would like to thank my colleagues at Cleveland State Community College, especially Barry Martin, for their encouragement and support.

Last I would like to thank my wife and children, Judy, Whitney and William for their understanding and patience.

ABSTRACT

This thesis presents a tutorial on the application of fuzzy control. A literature search provided much of the background for the work. Proficiency in implementing and tuning fuzzy controllers was gained through extensive computer simulation. The salient points of controller implementation and tuning are presented. The fuzzy controller has characteristics comparable to classical PD and lead compensators. With modification fuzzy controllers can achieve characteristics comparable to classical PI and lag compensators. It is shown that the PD type fuzzy controller can exceed the performance of its classical counterparts. Steady state error prediction is demonstrated for a class of fuzzy controller. The PI type fuzzy controller possesses disturbance rejection characteristics superior to those of the PD type fuzzy controller.

TABLE OF CONTENTS

CHAPTER		PAGE
1.	**Introduction**	1
	1.0. Background	2
2.	**From Fuzzy Logic to Fuzzy Control**	7
	2.0. Overview	7
	2.1. Linguistic Control Strategies	7
	2.1.1. Fuzzy Subsets	8
	2.1.2. Fuzzy Theorems	10
	2.1.2.1. Fuzzification and Defuzzification	13
	2.2. The Fuzzy Rulebase	16
	2.2.1. The Standard Rulebase	18
	2.2.1.1. Implementation of the Rulebase	18
	2.3. Summary	22
3.	**Applying Fuzzy Control**	24
	3.0. Overview	24
	3.1. System Simulation with Fuzzy Control	24
	3.1.1. The Fuzzy Simulation Program	26
	3.2. The PD Type Fuzzy Controller	28
	3.2.1. The Performance Characteristics of PD Controllers	28
	3.2.2. The PD-Like Characteristics of Fuzzy Controllers	31
	3.2.3. Example 3-1: PD Controller Comparison	32
	3.3. The PI Type Fuzzy Controller	33
	3.3.1. The Performance Characteristics of PI Controllers	36

3.3.2. The PI-Like Characteristics of Fuzzy Controllers.....38
3.3.3. Example 3-2: PI Controller Comparison.....39
3.4. A Fuzzy "Lead " Controller.....42
3.4.1. The Performance Characteristics of Lead Compensators.....42
3.4.2. The Fuzzy Controller as a Lead Compensator.....44
3.4.3. Example 3-3: Lead Compensator Comparison.....44
3.4.4. Lag Compensator Comparison.....46
3.5. Tuning the Fuzzy Controller.....46
3.5.1. The Effects of Changing Scaling Factors.....51
3.5.2. The Effects of Changing Rulebase Entries.....53
3.5.3. Changing the Membership Functions.....54
3.5.4. Examples of Tuning Fuzzy Controllers.....55
3.5.5. Observations and Findings Regarding Tuning.....57
3.6. Error Coefficients for the Fuzzy Controller.....58
3.6.1. Predicting Steady State Error for Fuzzy Controllers.....58
3.6.2. Scaling for Steady State Error.....61
3.7. Summary.....62
4. **Linearity and Disturbance Rejection**.....63
4.0. Overview.....63
4.1. Nonlinear Characteristics of Fuzzy Controllers.....63
4.1.1. The Direct Fuzzy Algorithm.....63
4.1.2. Example 4-1: Fuzzy Method Comparison.....64
4.2. The Rulebase as Phase Plane.....65
4.3. Disturbance Rejection with the Fuzzy Controller.....71
4.3.1. Disturbance Rejection Characteristics of Fuzzy Controllers.....71
4.3.2. Example 4-2: Comparison to Conventional Controller.....72
4.4. Summary.....75

5. **Summary and Suggestions for Further Research** 79

5.0. Overview 79

5.1. Summary 80

5.2. Suggestions for Further Research 82

References 83

Appendix 87

Vita 101

LIST OF TABLES

TABLE	PAGE
2-1 Linguistic Variables	9
2-2 Fuzzy Rulebase	17
2-3 Rulebase Values	21
3-1 Summary of Results in Example 3-1	36
3-2 Summary of Results in Example 3-2	44
3-3 Summary of Results in Example 3-3	49
3-4 Summary of Results Using Lag Compensator	49
3-5 Rulebase with Numbered Cells	52
3-6 Steady State Errors for the PD Type Fuzzy Controller	60
3-7 Steady State Errors for the PI Type Fuzzy Controller	61
4-1 Active Cells in Fuzzy Rulebase	69
4-2 Active Cells in Linear Rulebase	69
4-3 Disturbance Response Data for Example 4-2	75

LIST OF FIGURES

FIGURE **PAGE**

2-1 Graphical Representation of Linguistic Variables.....11

2-2 Graphical Representation of Fuzzy Rules.....15

2-3 Plot of Fuzzy Controller Nonlinearity.....21

3-1 Block Diagram of Fuzzy Controller Applied to Plant.....25

3-2 Division of Transient Response into 4 Phases.....29

3-3 Unit Step Response for: System with Gain Compensation Only
System with Untuned Fuzzy Controller.....34

3-4 Unit Step Response for: System with PD Compensation
System with Tuned Fuzzy Controller.....35

3-5 Step Responses for: System with Gain Compensation Only
System with Untuned Fuzzy Controller.....41

3-6 Step Responses for: System with PI Compensation
System with Tuned Fuzzy Controller.....43

3-7 Step Responses for: Gain Compensation Only
Untuned Fuzzy Controller.....47

3-8 Step Responses for: Lead Compensation
Tuned Fuzzy Controller.....48

3-9 Step Responses for: Lag Compensation
Tuned (PI) Fuzzy Controller.....50

4-1 Step Response Using Direct Fuzzy Method and Different Tuning Points (Example 4-1).66

4-2 Calculated Difference between Direct Fuzzy Method
and Linear Interpolation (Example 4-1).....67

4-3 Step Response with Numbers Corresponding to Rulebase.....68

FIGURE **PAGE**

4-4 Step Response for: System with Tuned Rulebase

System with Additional Tuning from Table 4-1 70

4-5 Block Diagram of Armature Controlled DC Motor with Disturbance 73

4-6 Step Response of Lead-Lag Compensated System of Example 4-2... 74

4-7 Disturbance for: Classical Lead-Lag Controller

PD Type Fuzzy Controller

PI Type Fuzzy Controller 76

4-8 Disturbance Response for: PI Type Fuzzy Controller

Combination of Fuzzy Controllers 77

A-1 Simulation Diagram of Fuzzy Controller Applied to Plant 90

A-2 Code Implementation of Triangles 91

CHAPTER ONE

INTRODUCTION

The area of fuzzy logic, based on L.A. Zadeh's fuzzy set theory [1,2], with contributions from Mamdani, Baldwin, Pilsworth, Fukami, Mizumoto, and Tanaka, among others [3,4], has received significant attention in recent years and has prompted an increase in research. One subject of this research, which has been particularly successful, is that of fuzzy logic control. Fuzzy logic control is the application of fuzzy logic in the field of feedback control. The primary factor that makes application of fuzzy control attractive is that knowledge of the plant (a mathematical model) is not required. There are a number of commercial and industrial applications using this technique and they are documented in the literature [10,13]. Typical publications describe the system to be controlled and the results of the controller application. In most cases, the authors do not give details about the design process to the point of allowing the reader to implement a fuzzy controller. Furthermore, the specifics of how to tune the controller to obtain the best response and disturbance rejection characteristics are often vague. Without these, the controller cannot be expected to perform well enough to justify its use. Few papers cover tuning and disturbance rejection in detail, and because these are significant issues to the designer, this is a shortcoming of the current literature.

This thesis first presents an outline for the design of fuzzy controllers, and this is followed by comparisons to classical controllers with similar characteristics. Next, a discussion and explanation of detailed issues such as tuning and steady state error is provided. The reader should be able to design a fuzzy controller without any further research based on the procedure given. In the next section, some history and background of fuzzy control are presented.

1.0 BACKGROUND

Conventional control techniques usually require a model of the plant to be controlled in order to determine the parameters of the controller for a specified response. The model is acquired in the form of either a transfer function or a set of state equations representing the dynamics of the plant. However, it is often difficult to obtain system models that are accurate enough for good compensation. For this reason, in many industrial applications, the proportional plus integral plus derivative (PID) controller is used, (particularly the PI), allowing the user to shape the response by tuning the controller coefficients. This approach may not yield a robust control and retuning may be necessary in the case of plant parameter changes. This is one of the major reasons for the interest in fuzzy control; that is, an inherently robust control algorithm can be obtained without a model of the plant [5,6,7,8,9].

Fuzzy control is based on heuristic knowledge of the process to be controlled, taking linguistic (large,small,etc.) control strategies and converting them to automatic control strategies. In this way the fuzzy controller is able to mimic human reasoning within the constraints of the rules provided by the designer. The rules supplied to the controller are simple IF-THEN rules, based on the knowledge of the operator or designer. An example of such a rule is "If the motor speed is too slow, then the armature voltage to the motor should increase". The input variable to the controller is motor speed , and the output variable is the armature voltage to the motor. As far as the plant is concerned, these variables are obtained and processed in the same way as in a conventional controller. The main difference is in how the signals are handled inside the fuzzy controller where fuzzification and defuzzification take place. Later, these concepts will be further developed.

Traditional logic systems use exact reasoning in the sense that their logic values are fixed and single valued; there can be one and only one value associated with them. Fuzzy controllers, as mentioned previously, use linguistic control strategies based on their rules. These linguistic control strategies use approximate, rather than exact, reasoning, which allows them to

handle imprecise or uncertain data. This may be seen in the example rule above, where the input variable to the controller is expressed as "too slow ". In conventional controllers, there would be no place for such a variable; it is not well- defined. This is an example of a linguistic variable.

A linguistic variable can be defined as a fuzzy subset with the introduction of certain definitions, which will be developed later. The approximate reasoning accomplished by the linguistic control strategies (and fuzzy controllers) allows them to handle plants without transfer functions, which may be nonlinear and have parameter variations. A good example was set forth by Zadeh[10]: "In the car parking problem, the kinematics of the car and the geometry of the problem are well defined, but the final position and orientation of the car are not specified precisely. Thus, it is the imprecision of the goal that makes it possible for a human to park a car without making any measurements or calculations. Traditional control techniques fail to do this because they employ crisp (precise) rather than fuzzy algorithms to arrive at a solution". In most controller applications the final value is specified and uncertainties exist elsewhere in the system. Still, fuzzy control provides a flexible and simple approach to handling these uncertainties.

Zadeh developed fuzzy set theory in the 1960's thinking it would have application to "soft sciences "linguistics, economics, etc. It was soon realized, however, that the fields in which the dependencies between variables are well defined could also benefit from the use of fuzzy logic [10]. In the 1970's Zadeh published extensively on fuzzy control, beginning in 1972 with "A Rationale for Fuzzy Control ". Historically, the first application of fuzzy control was reported by Mamdani and Assilian in 1975 [11]. In their paper they describe the regulation of a steam engine carried out on a PDP-8 computer and applied to the plant. Their results showed (after tuning) a better quality of control with the fuzzy controller than the best control obtained with a fixed digital controller (with respect to rise time, settling time, and overshoot). Interestingly, their intent was to use the fuzzy control rules as a "primer "or starting point for a learning controller. Their results, however, were much better than expected with the fuzzy

controller alone and they concluded that it was a powerful control technique by itself. Since then, fuzzy control has received significant attention.

There have been many reported successes regarding fuzzy control [4,5,10,11,12,13]. The technique has found its way into many industrial and commercial applications. Today, applications such as shower heads, washing machines, vacuum cleaners, camcorders, cameras, and air conditioners have been reported [10]. However, many control engineers find fault with published works making lofty claims if they do not properly compare fuzzy controllers with conventional controllers. Such claims bear little weight if they are not presented in the proper perspective. Indeed, there are papers in existence which compare such controllers, but without specifications or proper frame of reference. Such treatments often draw conclusions which are unfounded, or worse, trivial ('it performed satisfactorily'). One may be inclined to ask 'satisfactorily with respect to what?"[14]. Unfounded claims can lead to a lack of acceptance in industry. Another problem control engineers have is that analysis techniques of fuzzy controllers are in their infancy. The fuzzy controller is a nonlinear process and does not lend itself to conventional analysis techniques, thus common milestones such as stability and steady state error are difficult to predict. There has been some progress made in the area of nonlinear analysis of these systems but it is not as well developed as classical control theory [15-24]. For instance, Kickert and Mamdani [15] analyzed the fuzzy process as a multilevel relay action, using describing functions to predict oscillations, but this only applied to a single input system. Ray and Majumder [16] extended this to a multiple-input, multiple-output, system using the concept of a sector -bound nonlinearity and the circle criterion, yielding a very interesting approach. Their methodology only applies to plants which can be decoupled and does not apply to plants for which there is no mathematical model (the main use of such controllers). Abdelnour et. al. [17,18], applied describing functions and the multilevel relay analogy to the analysis of the fuzzy controller. The results required a trial and error method and a model of the plant. In a later paper, Jenkins and Passino [19] developed some interesting results using Nyquist-type stability criteria (circle criterion) and steady state error prediction using a similar approach to that of Ray

and Majumder [16] . Their approach allowed them to modify their design to avoid instability and decrease steady state error. The method established limits for gain values in order to avoid instability but within these limits system performance was adjusted by a trial process. Steady state error was predicted quite well, but improvements in steady state error require trial and error tuning. These results applied to a controller with one input, the error signal, and were not extended to the two dimensional input case (error and error rate). An approach used by Tang and Mulholland [20], and later by Rogers[21], developed an equation relating the variables in the rulebase. A similar approach was used by Zheng [22]. This approach will be examined in this paper. Braae and Rutherford [23], and later Harris and Moore [24], reported on a promising method to synthesize the rulebase, which relies on a fuzzy model of the plant and the desired response. Such fuzzy models are difficult to obtain with accuracy.

Finally, there are few papers which go into detailed explanations of how such controllers are designed [25]. A brief overview is usually all most authors devote to the actual implementation of the theory. Consequently, there is a large number of people who, without time to research it, do not understand enough of the concept to attempt the implementation process.

One of the main goals of this paper is to provide a tutorial on fuzzy controllers that will focus on understanding basic concepts, implementation, and controller tuning. This assumes some background in control theory (the assertion [14] that fuzzy control can be implemented without any mathematical background is unfounded) and also assumes the ability to program computers to implement the control algorithm.

In this chapter, a brief overview of the history and some of the successes of fuzzy control have been briefly discussed. In Chapter Two, the notion of fuzziness, and its representation by linguistic variables is explained. Next, the processes of fuzzification and defuzzification of signals are explained in depth with an example. Chapter Three gives a detailed discussion of applying and tuning fuzzy controllers, since this is one major problem with the technique. Chapter Four addresses linearity and disturbance rejection. The nonlinear nature of the fuzzy controller is discussed and compared to a similar control strategy using linear

interpolation. The disturbance rejection characteristics of fuzzy controllers are compared to classical controllers. Finally, Chapter Five gives conclusions and suggestions for further research.

CHAPTER TWO

FROM FUZZY LOGIC TO FUZZY CONTROL

2.0 OVERVIEW

A discussion of how to go from fuzzy logic to fuzzy control should start with a description of the fundamental reasoning behind fuzzy control. The thought process which drives the approach is simple. The desired input/output relationships for a system are described in words. These words make up a *set of rules* that specify a controller output for each possible combination of inputs. Fuzzy set theory is applied to the rules in order to use them to control the plant.

In this chapter the use and the application of such linguistic control strategies are explained. Fuzzy set theory is developed only enough to understand its application to the control process. The concepts of fuzzy subsets and membership functions are presented and the fuzzy theorems most applicable to the task of automatic control are explained. Using those fuzzy theorems, the processes of fuzzification and defuzzification can be defined and demonstrated.

The *set of rules* referred to above can be concisely presented in *matrix form*, commonly called a *rulebase*. The structure of the *rulebase* is discussed, as well as its derivation. The *rulebase* is used in the control process practically always by inclusion in computer software. Two methods of doing this are discussed, the lookup table and the direct calculation method.

2.1 Linguistic Control Strategies

In order to understand how fuzzy logic contributes to control processes, some terminology should be introduced. The fuzzy controller in this thesis uses two inputs, the error , $e(t)$, and the error rate, $\dot{e}(t)$. From a comparison of plant output $c(t)$ to setpoint $r(t)$, the

error signal, $e(t) = r(t) - c(t)$, is acquired and differentiation is employed to obtain the rate of change of the error signal. Note: Both $e(t)$ and $\dot{e}(t)$ are functions of time. For convenience, the function of time notation will be omitted and thus $e(t) \rightarrow e$ and $\dot{e}(t) \rightarrow \dot{e}$. These two signals are inputs to the controller where the fuzzy decision process takes place based on the relationship between the error and error rate. A rulebase, previously generated, specifies an output for every possible combination of the input variables over the range of interest. Every output in the rulebase results from a single rule, for instance, " If the error is large and negative, and the error rate is large and negative, then the controller output shall be large and negative ". This statement may seem obvious, but it does not appear in a form readily applied to a plant. How can such linguistic rules be used to control a physical system? How does the term " large and negative" relate to a voltage to be applied to a servomotor? These are questions that are discussed in the following section.

2.1.1 Fuzzy Subsets

To answer these questions the fuzzy subset must be defined, along with the linguistic variable. The term " fuzzy " originates with the introduction of variables that are not precise. For example, small, big and very big are indecisive terms. These are *linguistic values* of a variable. For instance, the variable " temperature" may take on *linguistic values* such as *cool, warm, hot*, and *very hot*. Variables that are measured in a real system, however, are not fuzzy - they are termed " crisp". If a fuzzy subset is defined to denote a negative large error (NL), then there is a *range* of values of the error that belongs to this subset, and not just one value. This fuzzy subset, NL, is a linguistic variable that allows the error signal to be expressed in linguistic terms. Table 2-1 defines a range of error variables that are represented by linguistic names (variables). The actual error signal is scaled to match the range defined, also known as the Universe of Discourse. This scaling is an extremely important part of the tuning process and is usually set after experimentation with the controller.

TABLE 2-1: LINGUISTIC VARIABLES

POSITIVE LARGE	PL
POSITIVE MEDIUM	PM
POSITIVE SMALL	PS
ZERO	ZE
NEGATIVE SMALL	NS
NEGATIVE MEDIUM	NM
NEGATIVE LARGE	NL

To explain the use of these definitions, consider the error signal, e. A fuzzy subset NL, which is a function of the error variable, e, can be defined to consist of a specific range of values of e. Such a fuzzy subset has a membership function $\mu_{NL}(e)$ which takes on values between zero and one. The value of $\mu_{NL}(e)$ determines the degree of membership of the variable e in the subset NL. An error value having a membership of one is a full member of the subset, while a membership less than one indicates only part membership, and a membership of zero means the error value is not a member of NL [4,25]. The membership functions can take on any shape, defined by the user, centered at some nominal value where the membership value is one. The most common shape used is a triangle, due to its ease of computation and representation. Actually, other shapes can be used, such as gaussian profiles or trapezoids[1,25,26], but triangles are easy to work with mathematically and the resulting design gives reasonably good results. If the membership function for the fuzzy subset NL is a triangle centered at -6, with endpoints at -8 and -4, then an error value of -6 would have full membership in that fuzzy subset, and a membership value of 1.0. On the other hand, an error value of -5 would have a membership value of 0.5 in NL and 0.5 in the overlapping subset, NM. The linguistic variables and their membership functions can be graphically represented as shown in Figure 2-1 where an initial scale assignment has been made. This diagram applies to $\mu(e)$ vs. e. A similar diagram is used to define $\mu(\dot{e})$ vs. $\dot{e}$.

2.1.2 Fuzzy Theorems

The three theorems of fuzzy logic most applicable to fuzzy control are those of a) fuzzy intersection of two sets (AND, $\wedge$), b) fuzzy union of two sets (OR, $\vee$) and c) fuzzy complement of a set (NOT) [2,6,11]. The fuzzy representation of the AND function can be shown as

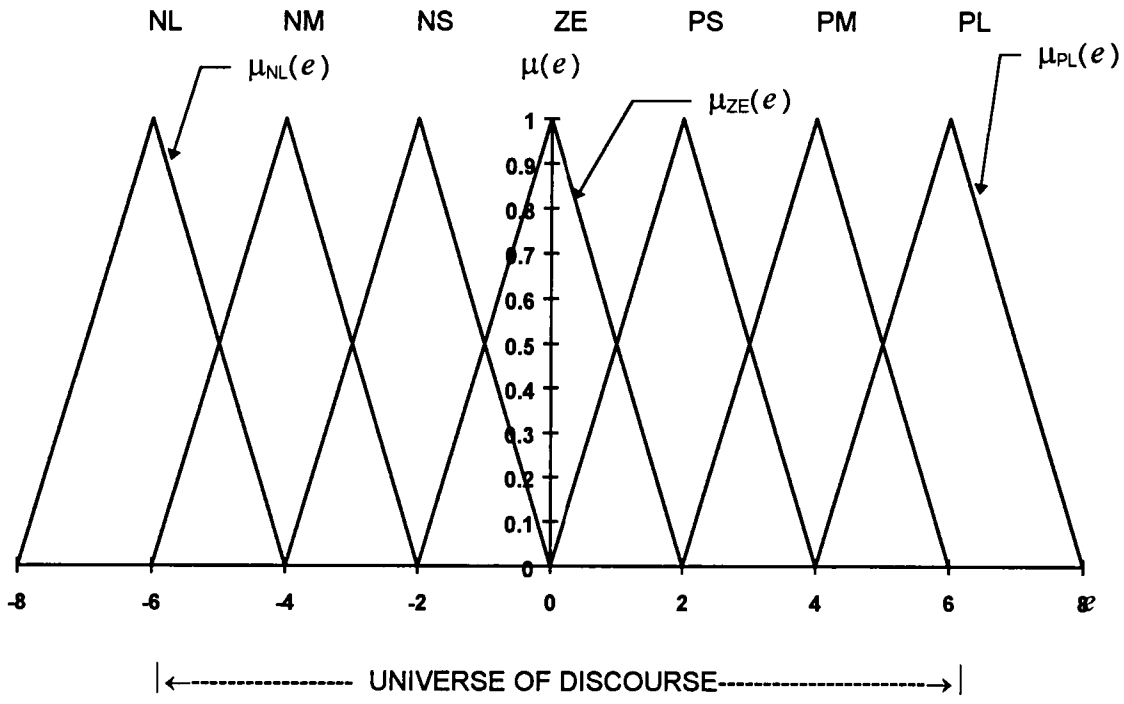


Figure 2-1: Graphical Representation of Linguistic Variables.

$$F(NL_e \wedge NS_{\dot{e}}) = MIN\left(\mu_{NL}(e), \mu_{NS}(\dot{e})\right) \quad (2\text{-}1)$$

where NL and NS are fuzzy subsets, $\mu_{NL}(e)$ and $\mu_{NS}(\dot{e})$ are membership functions, and e and $\dot{e}$ are the independent variables.

The fuzzy representation of the OR function can be shown as

$$F(NL_e \vee NS_{\dot{e}}) = MAX\left(\mu_{NL}(e), \mu_{NS}(\dot{e})\right) \quad (2\text{-}2)$$

and the fuzzy representation of the NOT function is

$$F(\overline{NL}_e) = \left(1 - \mu_{NL}(e)\right) \quad (2\text{-}3)$$

Example 2-1

Consider two sets of membership values as follows,

$$\mu(e) = (.3, .7, .7)$$

and

$$\mu(\dot{e}) = (.4, .5, 1.0),$$

then

$$F(\mu(e) \wedge \mu(\dot{e})) = [MIN(.3, .4), MIN(.7, .5), MIN(.7, 1.0)],$$

or

$$F(\mu(e) \wedge \mu(\dot{e})) = (.3, .5, .7).$$

The above values of $\mu(e)$ and $\mu(\dot{e})$ have been selected arbitrarily for this illustration. The procedure for determining $\mu(e)$ and $\mu(\dot{e})$ for a given e and $\dot{e}$ will be discussed in the following section.

2.1.2.1 Fuzzification and Defuzzification

The process of finding a membership function value corresponding to the value of the error signal at any sampling instant is called fuzzification. In order to implement the rule given above, the MIN function is used with the membership functions of the two signals, e and $\dot{e}$, since the antecedent consists of the intersection (ANDing) of the two input variables' linguistic representations. In an actual system, more than one rule " fires " at once, because the triangles overlap to some degree and an error value of -5 might be part member of the fuzzy subset NL as well as NM . This is illustrated in Example 2-2.

The following formula allows the fuzzy contributions of each of the applicable rules for a sampling period to be *defuzzified,* resulting in a " crisp " output signal. The crisp output signal is found using,

$$U = \frac{\sum f_i u_i}{\sum f_i}, \qquad (2\text{-}4)$$

where ,

f_i : is a weighting factor equal to the MIN of the two membership values associated with the fuzzy subsets in the antecedent of the *i*'th rule,

u_i : is the nominal (center) value of the fuzzy subset in the consequent of the *i*'th rule,

U: is the crisp output that is applied to the control system at that sampling instant [2,4,10,25,26,27,28,29].

A fact to be noted is that when the various fuzzy subsets (triangles) overlap as shown in Figure 2-1, for a given value of e only two rules contribute non-zero membership values. This is due to the overlapping nature of the triangles. The same can be said for $\dot{e}$. Thus for any given e and $\dot{e}$ there will be only 4 non-zero terms in each summation in Equation (2-4).

Example 2-2

Consider the following three rules:

Rule 1: If e is ZE and $\dot{e}$ is PS, then u_1 is PS

Rule 2: If e is PS and $\dot{e}$ is PM, then u_2 is PL

Rule 3: If e is PS and $\dot{e}$ is ZE, then u_3 is NS

The linguistic variables are now represented graphically by triangles. Thus, for the triangle shapes, the center of the respective triangle represents the location for maximum membership. The center values assigned in Figure 2-1 are 0 for ZE, 2 for PS, 4 for PM, etc. Figure 2-2 is a graphical representation of the membership functions representing these rules and how they are used in the *fuzzification* and *defuzzification* of the signals to obtain a crisp output for an error signal of 1 and an error rate of 1.5. The membership values for the error signal of 1 are $\mu(e)$=(0.5, 0.5, 0.5) and those for the error rate of 1.5 are $\mu(\dot{e})$=(0.75, 0, 0.25), as observed in Figure 2-2, so that

$$F(\mu(e) \wedge \mu(\dot{e})) = [\text{MIN}(0.5, 0.75), \text{MIN}(0.5, 0), \text{MIN}(0.5, 0.25)]$$

or $F(\mu(e) \wedge \mu(\dot{e}))$= (0.5, 0, 0.25)and the crisp output is, from Eq. (2-4),

$$U = (.5)(2)+(0)(6)+(0.25)(-2)/\ (0.5+0+0.25),$$

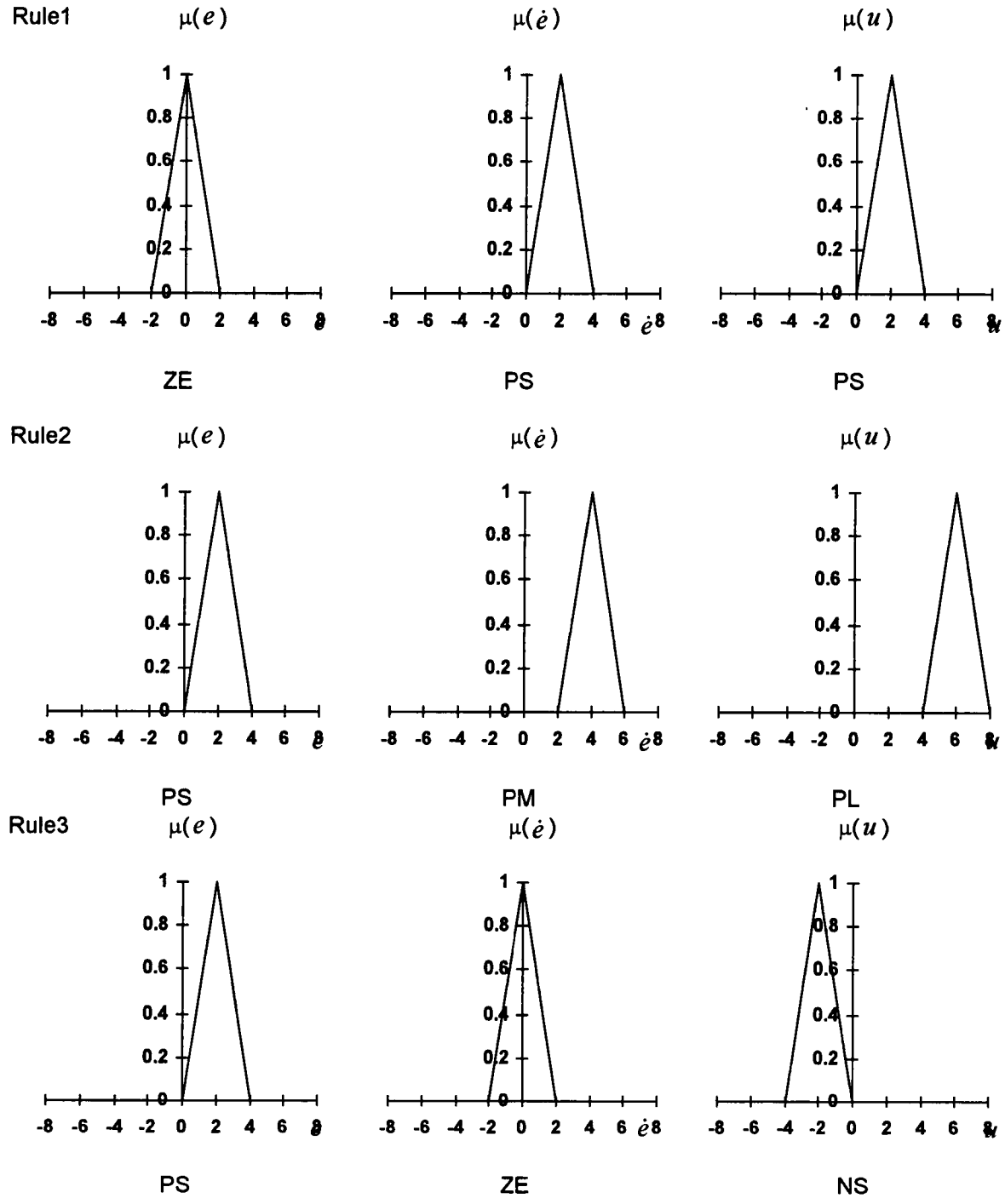


FIGURE 2-2: Graphical Representation of Fuzzy Rules.

U =0.667.

In practice more than three rules are used, and although the actual number is arbitrary, the literature indicates that more rules gives better results[30]. As indicated in Figure 2-1, there should be some overlap between the fuzzy subsets in order to ensure no signals " slip through the cracks ." The amount of overlap, as well as the shape and number of rules, is arbitrary. Occasionally, the literature indicates that different overlap gives slightly different results [22]. In Figure 2-1, there are seven fuzzy subsets, corresponding to seven linguistic variables. In order to obtain a distinct output for every possible combination of linguistic variables for e and $\dot{e}$, there must be 49 rules relating these subsets.

2.2 The Fuzzy Rulebase

There are many ways to record the rules generated for a fuzzy controller, probably the longest of which is the method in Figure 2-2. While this method allows one to calculate the crisp output values of Example 2-2 more easily, to tabulate 49 distinct control rules in this fashion would take several pages and extensive computer memory. However, since there are only 4 non-zero terms the calculations are reduced. The user must develop the software that will efficiently reduce the calculations to the particular 4 rules that apply.

In addition, once one has performed a few of these calculations it becomes much easier and the need for the graphical display is not so great. Furthermore, one usually programs a computer to do the calculations, as will be seen in this section. A much more concise and useful tabulation is known as a rulebase, and is shown in Table 2.

TABLE 2-2: FUZZY RULEBASE

e \ $\dot{e}$	NL	NM	NS	ZE	PS	PM	PL
NL	NL RULE 1	NL RULE 2	NL RULE 3	NL RULE 4	NM RULE 5	NS RULE 6	ZE RULE 7
NM	NL RULE 8	NL RULE 9	NL RULE 10	NM RULE 11	NS RULE 12	ZE RULE 13	PS RULE 14
NS	NL RULE 15	NL RULE 16	NM RULE 17	NS RULE 18	ZE RULE 19	PS RULE 20	PM RULE 21
ZE	NL RULE 22	NM RULE 23	NS RULE 24	ZE RULE 25	PS RULE 26	PM RULE 27	PL RULE 28
PS	NM RULE 29	NS RULE 30	ZE RULE 31	PS RULE 32	PM RULE 33	PL RULE 34	PL RULE 35
PM	NS RULE 36	ZE RULE 37	PS RULE 38	PM RULE 39	PL RULE 40	PL RULE 41	PL RULE 42
PL	ZE RULE 43	PS RULE 44	PM RULE 45	PL RULE 46	PL RULE 47	PL RULE 48	PL RULE 49

2.2.1 The Standard Rule Base

This rule base is not unique, but comparison of the input/output relationship to any step response will show that these linguistic variables are logical choices for the controller output, u_i. Some authors call this the Standard Rule Base (SRB) [22]. For instance, when a step input is initially applied to the system, in linguistic terms, e will be positive large (PL), and $\dot{e}$ will be negative. Assuming $\dot{e}$ is negative large, (NL) , the output of the plant is rapidly approaching the desired state, and thus it is desirable for the plant input (controller output, u_i) to be zero, since the conditions of the inputs are already as desired. These values can be changed, for instance, to increase the rate of change of the output, as will be seen later. If $\dot{e}$ is positive and large, the plant output is diverging from the desired state, and u_i should be positive large in order to cause the plant output to increase. If e and $\dot{e}$ were both negative large, then the plant output would be again diverging from its desired state, and u_i should be negative and large to reverse that trend. In creating this particular rulebase, it is assumed that the plant is one whose output increases with positive input and decreases with negative input.

2.2.1.1 Implementation of the Rulebase

In the implementation of the controller, Eq. (2-4) can either be used in a program to automatically determine the controller output U or U can be calculated for specific combinations of inputs and stored in a lookup table. In using the former approach, a program was written that used the actual triangles to determine membership values. Equation (2-4) was then used to calculate the crisp output of the controller. This program will be presented in Chapter 3. A second program was written which used if-then statements to determine which rules were applicable at a given sampling instant. The if-then statements allow the

circumvention of calculating the contribution of each rule, each time. As was seen earlier, more than one rule can fire at once. However, with the amount of overlap chosen in Figure 2-1, as noted earlier, no more than 4 rules can fire at one time since e can be a member of no more than 2 fuzzy subsets at once, and likewise for the error rate. Consequently, the program was designed to decide which 4 rules were applicable at each sampling instant, and use only those 4 in performing the calculation in Equation 2-4. This second program worked well, but the first one will be presented in Chapter 3 because of its simplicity. In practice the lookup table is used most often [25, 30]. The lookup table represents a shorter and much simpler approach. This approach allows the computer to simply extract U from a table based on the values of e and $\dot{e}$ at that sampling instant. Care must be taken in calculating and tabulating these values however. Using the defuzzification algorithm represented by Equation (2-4) results in a nonlinear relationship between U and e for a given value of $\dot{e}$. This nonlinearity is not apparent unless enough points in the table are calculated to preserve it. Thus the lookup table can deviate from a fuzzy controller if too little resolution is used in compiling the values. The tabulation of U for just the 49 input combinations in the SRB above (replacing each linguistic variable with a crisp number equal to its center value) shows that U will have crisp values equal to the center values of the output triangles. For instance, if e is exactly -6 (a center value) and $\dot{e}$ is exactly -6, then they both have membership function values of

$$\mu_{NL} = 1.0$$

and are members of no other subsets. Therefore, only one rule fires, and the minimum of the two memberships is 1.0. Therefore, using Equation 2-4, the controller output U for this combination of inputs is

$$U = \frac{1.0(-6)}{1.0} = -6$$

which is exactly the center value of the output fuzzy subset prescribed for this condition, (NL). Further analysis shows that for any input combination of center values for e and $\dot{e}$ the output will be the center value of the prescribed output fuzzy subset.

If e is 1 (not a center value) and $\dot{e}$ is -4 (a center value) then the membership value, $\mu_{ZE}(e)$ is 0.5 and $\mu_{PS}(e)$ is 0.5. The membership value for $\dot{e}$, $\mu_{NM}(\dot{e})$, is 1.0. In this case two rules fire, specifically

if e is ZE and $\dot{e}$ is NM then u_i is NM,

and

if e is PS and $\dot{e}$ is NM then u_i is NS.

In the use of Equation (2-4),

$$U = \frac{\sum_{i=1}^{2} f_i u_i}{\sum_{i=2}^{2} f_i},$$

one notes that $f_1 = f_2 = 0.5$ while $u_1 = -4$ and $u_2 = -2$.

The controller output is therefore ,

$$U = \frac{0.5(-4) + 0.5(-2)}{0.5 + 0.5} = -3,$$

which is the midpoint between NM and NS or -4 and -2. Further calculations show that for any midpoint values in the table, Equation (2-4) yields a result that implies linear interpolation between table values. It is only in between these values that the nonlinearity is evident. For instance, if e is allowed to vary from -1.0 to 1.0 and $\dot{e}$ is fixed at -1.0, the controller output U can be calculated and plotted as shown in Figure 2-3. Included in Figure 2-3 is the straight line resulting from a linear relationship for comparison. The fuzzy controller output is clearly nonlinear. The only exception to this is the case for which $\dot{e}$ is 0 (the center column of the rulebase). In the center column the relationship is linear. The controller output based on linear interpolation is very close to the actual calculated values, and the difference in control action is negligible in some cases. Table 2-3 gives the Standard Rule Base in numerical form.

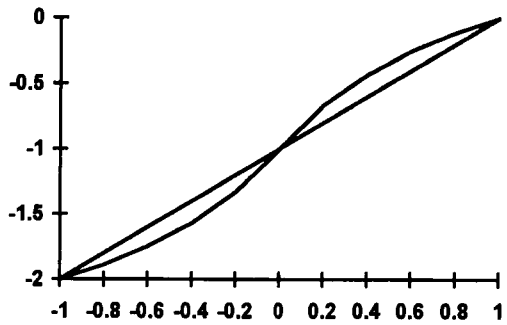


Figure 2-3

Plot of Fuzzy Controller Nonlinearity

TABLE 2-3: RULEBASE VALUES

e \ $\dot{e}$	-6	-4	-2	0	2	4	6
-6	-6	-6	-6	-6	-4	-2	0
-4	-6	-6	-6	-4	-2	0	2
-2	-6	-6	-4	-2	0	2	4
0	-6	-4	-2	0	2	4	6
2	-4	-2	0	2	4	6	6
4	-2	0	2	4	6	6	6
6	0	2	4	6	6	6	6

would be representative of a lookup table using linear interpolation between values. To ensure the nonlinearity is preserved, the lookup table must contain points between the center and midpoint values. Table 2-3 will be used in the simulation program in Chapter 3 because it contains the numerical values of the antecedents of the control rules. Equation (2-4) will be used to determine the actual output values, however. It should be emphasized that a scaling process will be performed on the values in Table 2-3. These particular values are used in this explanation and may or may not be appropriate values used for a given control system. Also, the symmetry and amount of overlap of the membership functions, even the shapes themselves, (see Figure 2-1) can be chosen to " customize " a particular design and the selection can well depend on the experience and astuteness of the control engineer.

2.3 Summary

Fuzzy logic, as applied to control systems, has been introduced in this chapter. The linguistic control strategies using linguistic variables and fuzzy subsets have been presented along with the membership functions and fuzzy theorems that allow the determination of the crisp control, U, to be applied to the plant. The concise representation of a set of control rules can be accomplished through the rulebase: a matrix of linguistic variables containing all possible input/output pairs. This rulebase can be simplified by using Equation (2-4) to calculate the crisp outputs for each input combination. The crisp outputs can be determined directly from Equation (2-4) , without approximation. Alternatively, a lookup table can be used. Using the lookup table saves code and computer time but quantization must be used between values not present in the table. The amount of nonlinearity preserved by this method depends on the resolution desired in the rulebase. The control action may deviate from that of the fuzzy controller when either quantization or linear interpolation is used. Therefore, it is recommended that the nonlinearity be represented in the lookup table to as high a degree as feasible. In the Examples in the following

Chapters, the actual fuzzy controller output values will be calculated in the simulation program but without including the calculation time as a delay.

In Chapter 3, the subject of applying a fuzzy controller is addressed in detail. Examples are given which show the additional steps necessary after the implementation process presented in this chapter. Two types of fuzzy controllers will be introduced and their similarities to conventional PD and PI controllers explained. Comparison to classical lead and lag controllers will also be presented. The difficulties commonly encountered in tuning the controllers and some of the research being done to overcome them are explored. The examples are simulated using the digital computer which is necessary to carry out the tuning process.

CHAPTER THREE

APPLYING FUZZY CONTROL

3.0 OVERVIEW

In this chapter the application of the fuzzy controller is demonstrated. First, an overview of the simulation program used to test the fuzzy designs is presented. The Advanced Continuous Simulation Language (ACSL) software is used for simulation support [34]. A block diagram is provided to facilitate the discussion. Next, the similarity of the fuzzy controller to a classical PD controller is described by giving a review and comparison of the PD controller's salient features to that of the fuzzy controller. The performance of each controller is compared in an example. Another popular fuzzy controller design mimics the classical PI controller. Following this presentation, a section is then devoted to exploring the similarities in design and performance of these two controllers.

The characteristics of phase lead compensation are reviewed. The PD type fuzzy controller is then compared to a lead compensator and it is shown that the two have similar characteristics. The use of phase lag compensation is reviewed and a performance comparison between the lag and the PI type fuzzy controller is given. Following the lag compensator comparison, fuzzy controller tuning is considered and conclusions are drawn from the literature as well as from preceding examples.

A section on steady state error follows, providing further guidance in the tuning of the fuzzy controller.

3.1 System Simulation with Fuzzy Control

Figure 3-1 shows a block diagram configuration for the fuzzy controller applied in a unity negative feedback loop. The inputs, e and $\dot{e}$, and output U must be scaled to properly couple

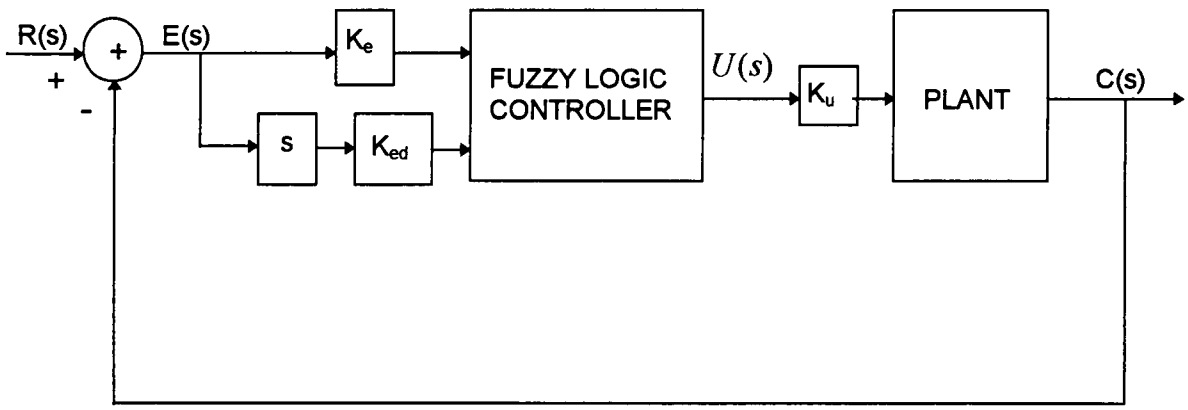


Figure 3-1 Block Diagram of Fuzzy Controller Applied to Plant

the system signals to the Universe of Discourse for each variable. For the system shown, this can be accomplished by adjusting K_e, K_{ed}, and K_u for e, $\dot{e}$ and U, respectively. Initial values for these scaling factors can be determined from expected values of the individual variables. This plays on the experience of the control engineer. As an illustration , if e is expected to stay within the range from -1 to 1 and the Universe of Discourse for error is -6 to 6, then a logical starting point for K_e would be 6. This is, in fact, the value used in some of the simulations to be shown later. The process of choosing scaling factors will be discussed further in Section 3.5, *Tuning the Fuzzy Controller.*

3.1.1 The Fuzzy Simulation Program

There are several simulation packages that can be used to study Fuzzy Control. These include, but are not limited to, MATRIX$_X$ [32], MATLAB [33] ACSL [34] and SIMNON [35]. For this thesis, ACSL was selected because it is a familiar package and because of its block diagram nature that fits very nicely to PI, PD, lead and lag controllers. Also, disturbances are easy to include with the system. A copy of the ACSL program used to simulate the controller with a first order lag plus delay is provided in the Appendix along with a programming diagram. An explanation of how the fuzzification/defuzzification is programmed is given. This plant will be used in the first example. Other plants are easily modelled by adding the necessary blocks.

The constants in the Initial Section of the simulation program can be changed in the *interactive mode*, allowing different settings to be applied in a trial and error format. The variables' functions can be summarized as follows:

R:	The reference input
C:	The plant output
FIN:	Time (in seconds) of simulation duration
K:	Overall system gain

K_e:	Error scaling factor
K_{ed}:	Error rate scaling factor
K_u:	Output scaling factor
KR:	Magnitude of step input
P:	Plant pole in S-plane
Z:	A value close to zero (avoids division by zero)
ME:	Slope for error triangles
MD:	Slope for error rate triangles
IC:	Initial condition for delay
TD:	Delay time
CMN:	Data point interval
U:	Controller output

The error signal E and its derivative ED are scaled to obtain EF and EDF. These values are used in later equations to determine membership values. The membership values are used with Equation (2-4) to obtain the controller output, U. U is then scaled and EK is input to the plant. The program simulates the plant by starting with EK and successively performing integrations and algebraic operations to obtain the output C.

The implementation of the fuzzy logic comes last in the program, like a subroutine. Each rule is separated for convenient reading. A brief outline of the implementation is given, and a detailed explanation included in the appendix.

For each rule, a straight line equation is specified for each side of the triangles representing the membership functions. A point on each line is determined based on the value of EF (which is the scaled value of e) or EDF (which is the scaled value of $\dot{e}$). The negative portions are removed by a MAX operation, and the values are restricted to the individual triangle by a MIN operation. With the two membership values $\mu(e)$ and $\mu(\dot{e})$ thus calculated, the

minimum is taken for each pair and used in Equation (2-4). Thus the defuzzified control U is obtained and used in the simulation in the beginning of the program.

In the next two sections, the two types of fuzzy controller (PD and PI), are formalized.

3.2 The PD Type Fuzzy Controller

The configuration of the fuzzy controller described thus far is known as the PD type fuzzy controller. In order to illustrate the reasoning behind this title, a review of the characteristics of PD control will be presented and compared to those of the fuzzy controller.

3.2.1 The Performance Characteristics of PD Controllers

The classical proportional plus derivative (PD) control scheme is characterized by the following function relating controller output to the controller input, error.

$$U(t) = K_P e(t) + K_D \frac{de(t)}{dt}. \tag{3-1}$$

As such, it is a strategy that follows an intuitive approach to shaping the step response of a feedback control system. For an illustration consider Figure 3-2. Figure 3-2 divides a portion of the transient response into 4 phases which correspond to the 4 quadrants of the phase plane (e vs. $\dot{e}$). As the output approaches the setpoint from zero initially (phase 1), the error signal is decreasing and the derivative term is negative. This combination of inputs in Equation (3-1) imparts a damping effect on the system, reducing overshoot. If the system output still overshoots the setpoint, then the error is negative as well as the derivative term (phase 2). These two negative terms act together to force the system output back toward the desired value. As the output begins decreasing towards the setpoint (phase 3), the error is still negative but the derivative is now positive. This will act to reduce undershooting the setpoint. When the

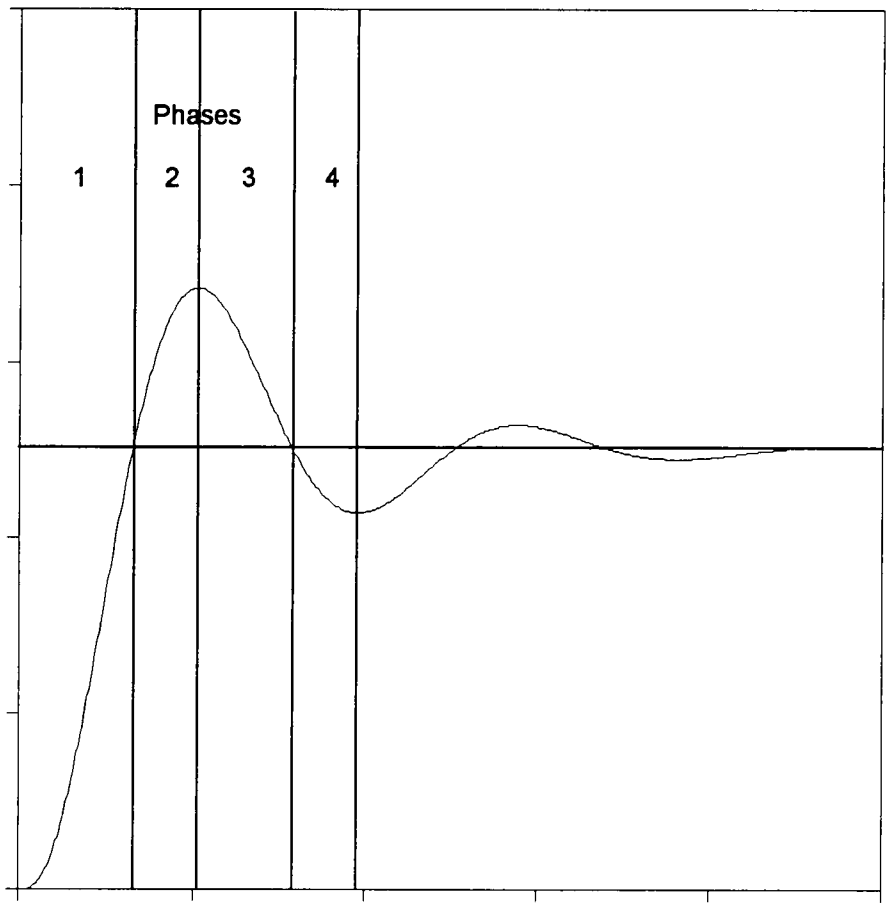


Figure 3-2

Division of Transient Response into 4 Phases

error becomes positive again as the output falls below the setpoint (phase 4), the derivative is positive and the two positive terms reinforce each other to force the output back toward the desired value. This cooperative action between the proportional and derivative terms is well known, and can also be represented in the phase plane. In the time domain then, the proportional gain, K_P, has immediate impact on the rise time of the system, while the derivative gain, K_D, affects the damping (hence overshoot) of the response. These relationships will be useful in tuning the fuzzy controller, as will be shown in section 3.5.

The transfer function of the PD controller in the s (Laplace) domain is

$$G_c(s) = K_P + K_D s \tag{3-2}$$

or

$$G_c(s) = K_D\left(s + \frac{K_P}{K_D}\right). \tag{3-3}$$

This transfer function indicates that a zero is placed at $s = -\frac{K_P}{K_D}$ in the s-plane. The effect on the frequency response is the same as a high-pass filter: there is an increase in bandwidth and introduction of positive phase (lead). These effects can cause a faster response and increased phase margin. The increase in phase margin corresponds to an improvement in relative stability and will result in less overshoot. Of course, as is also well known, a high pass filter will allow noise to be amplified and passed to the system, which is an undesirable effect.

In the process control industry, the proportional plus integral plus derivative (PID) controller is extensively used in feedback loops because of its versatility. There are widely accepted guides for tuning such controllers [36] which use criteria such as the open-loop frequency of oscillation and the three-quarter settling time, as with the Cohen-Coon and Ziegler-

Nichols methods [36], and optimization in terms of the IAE, ISE and ITAE as with the Rovira-Murrill-Smith method [36]. While these methods are well known and widely used for PID and PI controllers, there are no such popular criteria for the PD controller. This is likely because many processes are first order (or modelled as such) and the PD approach does not eliminate steady state error. Additionally, the high pass characteristics of the controller cause undesirable noise to be passed on to the process. Therefore, if one intended to use the PD controller, the attributes mentioned above, bandwidth (affecting speed of response), and phase margin (affecting overshoot) would have to provide the necessary guidance in tuning. In the next section, the similarities of the fuzzy controller to the classical PD controller will be presented and discussed.

3.2.2 The PD-Like Characteristics of Fuzzy Controllers

To comprehend how the fuzzy controller is similar to the PD controller, consider the rulebase in Table 2-3. The entries in the rulebase are chosen in the same intuitive manner as described in Section 3.2.1, where the cooperative action of the proportional and derivative terms of the PD controller was analyzed. A brief look at the rulebase will indicate that, in terms of crisp numbers, the controller output is approximately the sum of the two input values, e and $\dot{e}$. With the exception of the upper left and lower right corners, it is exactly the sum of e and $\dot{e}$. For instance, if e is 2 and $\dot{e}$ is -2 , then U is zero. This summation holds for most of the input conditions that would commonly occur, since the upper left and lower right corners of the rulebase do not represent common input conditions unless the system is unstable or has excessive overshoot, both of which would be avoided.

This is similar to the classical proportional plus derivative (PD) controller, and the output of the PD type fuzzy controller can be described as (compare to Equation (3-1))

$$U(t) = K_u\left(K_e e(t) + K_{ed}\frac{de(t)}{dt} \right). \qquad (3\text{-}4)$$

The behavior is quite similar to the PD controller, as long as the fuzzy controller is governed by the Standard Rule Base. The constants K_e, K_{ed}, and K_u are the same ones as described in Section 3.1, in the discussion of the simulation program. They also correspond to the coefficients in Equation 3-1, K_P and K_D, and this relationship provides some guidance in tuning the fuzzy controller as previously stated in Section 3.2.1. The next section compares the classical PD controller to the PD type fuzzy controller.

3.2.3 Example 3-1: PD Controller Comparison

A common model used to represent the plant in the process control industry is the first order lag plus delay. The transfer function of such a plant can be represented by

$$G_P(s) = \frac{Ke^{-\theta s}}{\tau s + 1}, \qquad (3\text{-}5)$$

where θ and τ are the delay time and time constant, respectively. Consider a process in which the delay, θ, is 3 seconds and the time constant, τ, is 8 seconds, with a gain, K, such that the steady state error from a unit step input is 0.25. It is desired to maintain the overshoot at less than 10%. A gain of K =3.0 achieves the error of 0.25, but causes the overshoot for a step input to be approximately 54.3%.The transfer function is

$$G_P(s) = \frac{3e^{-3s}}{8s + 1}. \qquad (3\text{-}6)$$

The zero of the PD controller can be positioned such that the overshoot is reduced. The amount of reduction is limited since only 90° of phase can be added by a single zero. Therefore , the gain may require reduction also. Referring to the Bode diagram, the zero must be larger than the pole to guarantee the gain magnitude curve will cross 0 dB. If the zero is chosen much larger than the pole, it will not affect the overshoot. If the zero is chosen too close to the pole, there will be reduction of overshoot, but the settling time is compromised. The zero for this problem was set at 0.714 radians/second. This achieved overshoot reduction while maintaining low settling time. The overshoot required additional reduction which was achieved through setting K = 2.3.

The PD type fuzzy controller was used to compensate the same system with the same original gain. The step responses of the system with gain compensation only and the untuned fuzzy controller are shown in Figure 3-3. The final step responses of both controllers are shown in Figure 3-4. In comparing the controllers' performance, the specifications were met by the classical PD controller, since a 10% overshoot was achieved but the reduction in gain caused an increase in steady state error. In the case of the fuzzy controller the specification for overshoot was achieved with an improvement in steady state error. The peak time and settling time were both longer than the classical PD controller. These results are tabulated in Table 3-1. The steps in tuning the fuzzy controller will be covered in more detail in section 3.5.

3.3 The PI Type Fuzzy Controller

Another popular configuration for the fuzzy controller is commonly referred to as the PI type fuzzy controller. Before an explanation is given of how this fuzzy controller is similar to a classical PI controller, a review of the classical PI controller is given.

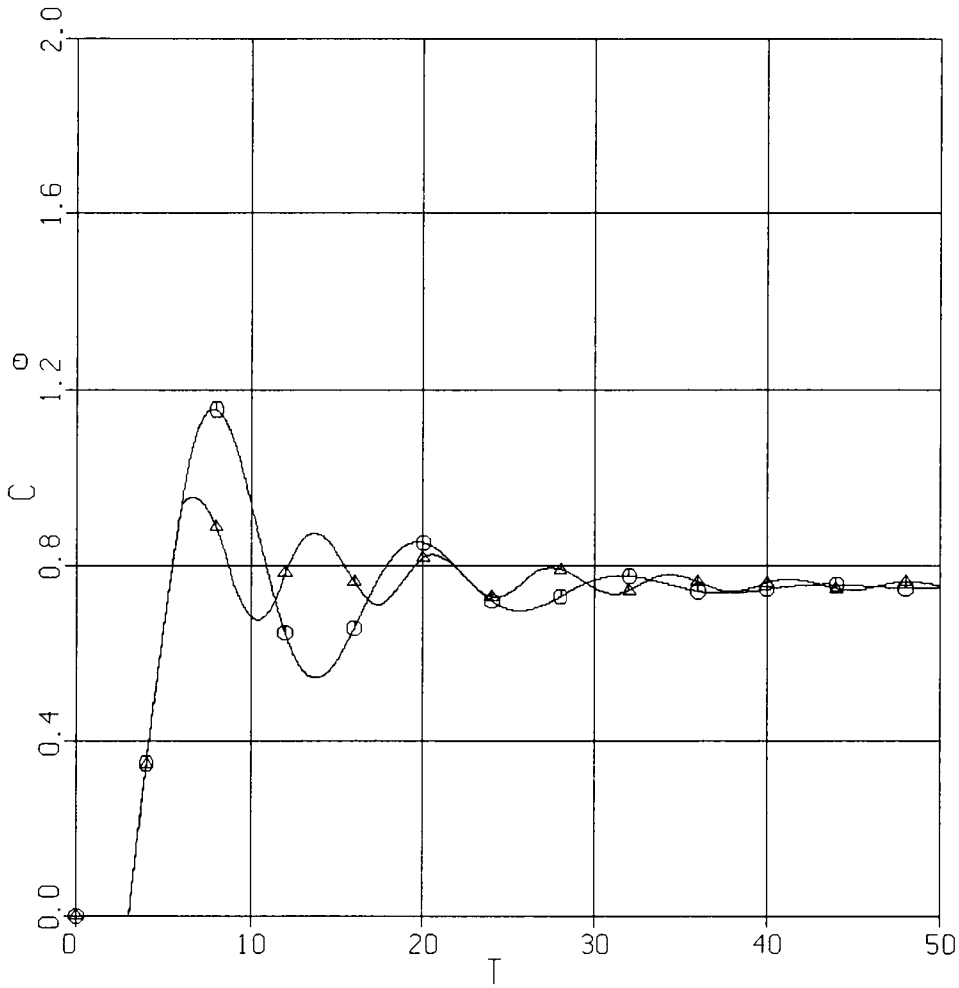


Figure 3-3

Unit Step Response for:

○ System with Gain Compensation Only

Δ System with Untuned Fuzzy Controller

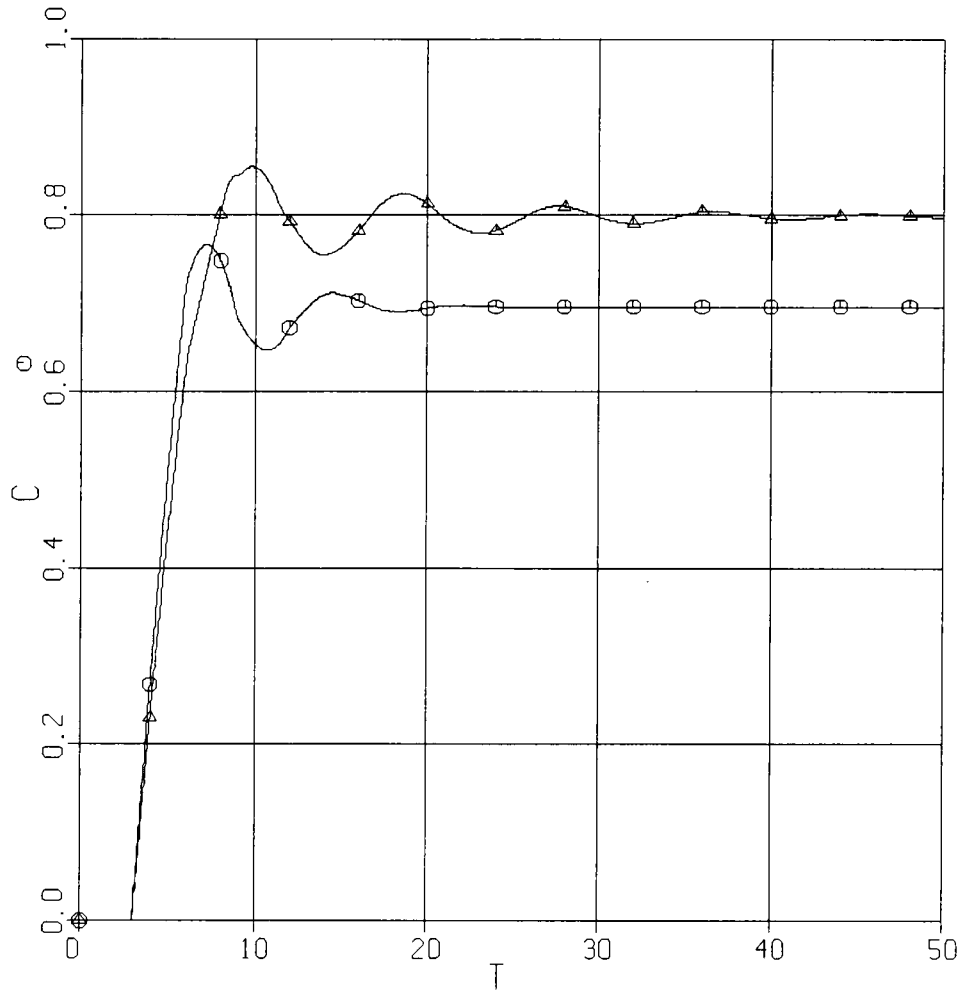


Figure 3-4

Unit Step Response for:

○ System with PD Compensation

Δ System with Tuned Fuzzy Controller

Table 3-1: Summary of Results in Example 3-1

Compen-sation ↓ / Attribute →	Overshoot	Peak Time	Settling Time	Steady State Error
Gain Compensation	54.3%	7.85s	33.5s	0.25
PD Compensation	9.9%	7.2s	15s	0.303
Untuned Fuzzy	28.4%	6.7s	42.7s	0.25
Tuned Fuzzy	6.9%	9.85s	24.4s	0.20

3.3.1 The Performance Characteristics of PI Controllers

The classical proportional plus integral (PI) control algorithm can be described by the following input/output relationship:

$$U(t) = K_P e(t) + K_I \int e(t)dt \,. \tag{3-7}$$

The strategy of this approach is that as the response settles out, if there is any steady state error, the integral term continues to ramp up, increasing in magnitude until it effectively removes the error altogether. If the error is positive, then the integral term is positive and increasing, which causes the output to increase and reduce the error. If the error is negative, the integral term is also negative and increasing in magnitude, which causes the output to decrease and reduce the error. Ultimately, this strategy will eliminate the steady state error completely. From a time domain standpoint, the proportional gain, K_P, has immediate impact on the speed of response

of the system, while the integral gain, K_I, affects the steady state error. This is an abbreviated version of the PID controller mentioned earlier, with the drawback that there is no immediate control over damping.

In the frequency domain, this controller can be represented by the following transfer function:

$$G_c(s) = K_P + \frac{K_I}{s}, \tag{3-8}$$

or

$$G_c(s) = \frac{K_P(s + \frac{K_I}{K_P})}{s}. \tag{3-9}$$

The transfer function shows that a zero is placed at $s = -\frac{K_I}{K_P}$ and a pole is added at the origin in the s-plane. This controller has the same effect on the frequency response as that of a low pass filter. There is an attendant decrease in bandwidth and addition of phase lag. These effects can cause a more sluggish response and more overshoot, unless care is taken in the choice of the coefficients. The major advantages of this approach over the PD controller is that a low pass filter does not pass noise into the system and the addition of the pole increases the system type and causes the steady state error to be eliminated. The disadvantages are a slower response, phase lag, and no derivative term to control overshoot.

The PI controller is widely used in the process control industry because of the steady state error control and its ease of tuning compared to the PD or PID controllers, which are generally more difficult to tune due to the introduction of undesired noise signals into the system. This is a controller for which the methods mentioned in section 3.2.1 are directly applicable, as the tables for tuning coefficients to minimize ISE, ITAE, and IAE are designed for PID and PI

configurations explicitly. These tables will be used later in an example. In the next section the PI type fuzzy controller is presented.

3.3.2 The PI-Like Characteristics of Fuzzy Controllers

The PI type fuzzy controller is a popular design characterized by integrating the output of the PD type fuzzy controller. To see why this is a logical choice, refer to Table 2-3. As an example, if e is 2 and $\dot{e}$ is -2 then the *change* in U, as indicated in the table, should be 0. A brief review of the other rules in the table confirms that this change in strategy is sensible. The output changes from

$$U_{PD}(t) = K_u\left[K_e e(t) + K_{ed}\frac{de(t)}{dt}\right] \tag{3-10}$$

to

$$U_{PI}(t) = \int K_u\left[K_e e(t) + K_{ed}\frac{de(t)}{dt}\right]dt \tag{3-11}$$

and

$$U_{PI}(t) = K_u\left[K_{ed} e(t) + \int K_e e(t)dt\right] \tag{3-12}$$

so the controller equation is now similar to the classic proportional plus integral (PI) controller given in Equation 3-7. The coefficients are the same as for the PD type fuzzy controller, but now the roles are changed. The error scaling factor, K_e, now multiplies the integral term and the derivative scaling factor, K_{ed}, now multiplies the proportional term. This relationship cannot be taken as an absolute, however, because of the nature of the rule-based control. The error signal

still applies to the rulebase in the same manner as before, as does the error rate signal. Only after the decision making process is complete is the integration performed. This unusual relationship will be explained further in Section 3.5.4. The PI type fuzzy controller does have certain advantages over the PD type:

1. It improves steady state error in the same fashion as the classical PI controller,
2. It has better disturbance rejection characteristics than the PD type fuzzy controller.

The steady state error improvement will be demonstrated in the next example, while disturbance rejection will be discussed in Chapter 4. The next section gives an example comparing a classical PI design with the PI type fuzzy controller.

3.3.3 Example 3-2: PI Controller Comparison

This example gives the design for a classical PI controller using ITAE minimization coefficients from Rovira [36] and compares the results with the best response obtainable from a PI type fuzzy controller. Consider the same plant process from Example 3-1:

$$G_P(s) = \frac{Ke^{-3s}}{8s+1}. \tag{3-13}$$

with $K = 3.0$, as before. The coefficients for optimization with respect to ITAE from Rovira (for setpoint changes) are determined from the following equations relating the system parameters to the tabulated values. The compensator transfer function is given by

$$G_c(s) = K_c\left(1 + \frac{1}{T_i s}\right), \tag{3-14}$$

where

$$KK_c = A\left(\frac{\theta}{\tau}\right)^B, \tag{3-15}$$

and

$$\frac{\tau}{T_i} = A + B\left(\frac{\theta}{\tau}\right). \tag{3-16}$$

In these equations , θ is the plant delay, τ is the plant time constant of the system and K is the plant gain. A and B are the parameters tabulated by Rovira. From Table 6-4 in [36], for the proportional gain, A=0.586 and B=-0.916. These values in Equation (3-15) lead to a value of $K_c = 0.48$. Next, for the integral gain, A=1.03, and B=-0.165. These values in Equation (3-16) lead to a value of $T_i = 8.263$. The PI controller becomes

$$G_c(s) = 0.48\left(\frac{s + 0.121}{s}\right). \tag{3-17}$$

Using this design the system step response has an overshoot of approximately 5.5% and a peak time of 13 seconds. As expected, this response is somewhat slower than the PD compensated system, but in this case, the steady state error is zero.

The PI type fuzzy controller is now used to compensate the same system with the same gain. Again, the untuned fuzzy controller improved the overshoot, but in this case, the response time was degraded. After tuning, the fuzzy controller was able to speed up the response and still maintain a similar overshoot, but the response was not as good as the classical PI controller. While the steady state error for a step input was zero for both controllers, the steady state error for a ramp input was superior for the fuzzy controller. The subject of steady state error prediction will be addressed in more detail in Section 3.6. The step responses are shown in Figure 3-5 for the original system and untuned fuzzy controller. The step responses for the PI controller and the

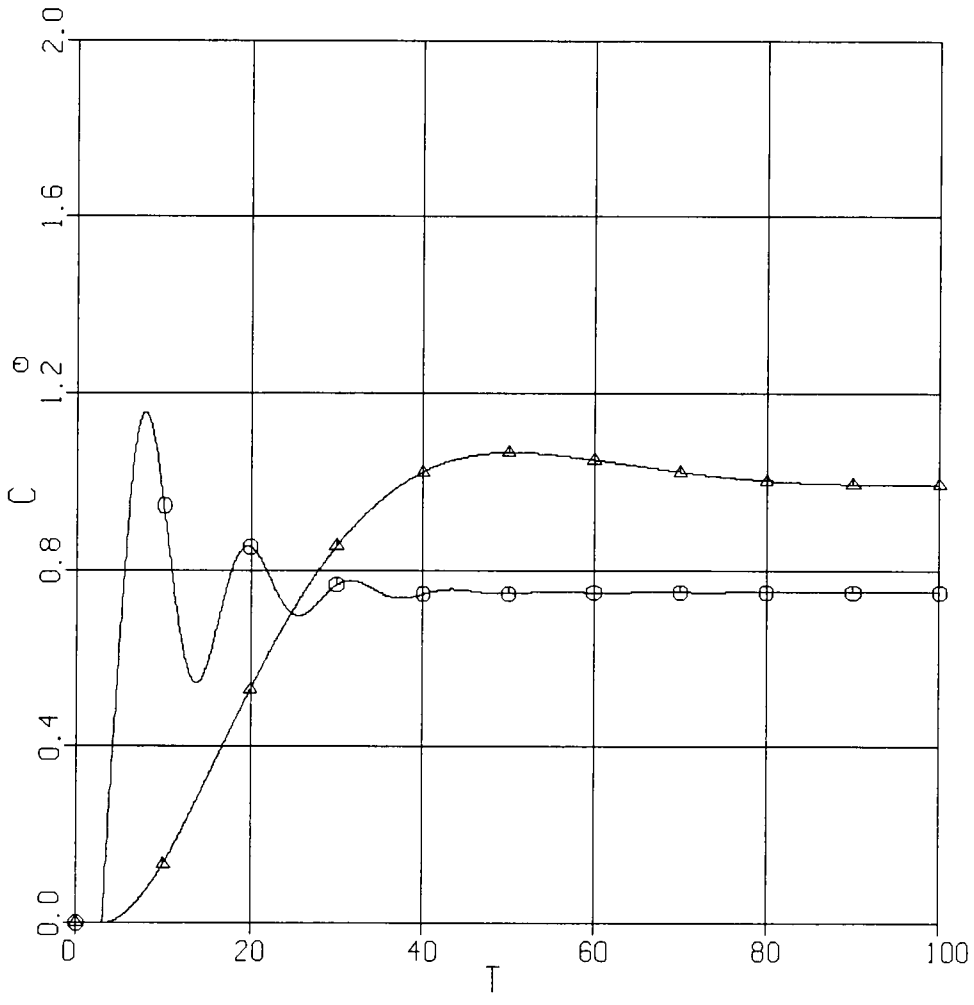


Figure 3-5

Step Responses for:

○ System with Gain Compensation Only

Δ System with Untuned Fuzzy Controller

tuned fuzzy PI type controller are shown in Figure 3-6. The results are shown in Table 3-2. In the next section, the fuzzy controller is compared to a lead compensator.

3.4 A Fuzzy " Lead" Controller

The control action of the PD type fuzzy controller can be compared to lead compensation, based on the discussion in Section 3.2 since phase lead is common to both control schemes. The following sections give a review of lead compensation and how the fuzzy controller is similar.

3.4.1 The Performance Characteristics of Lead Compensators

The phase lead compensator is well known in classical control theory. As the name implies, the main intent of this control strategy is to introduce phase lead into the system, improve the bandwidth and hence improve the transient response. Based on a desired amount of phase margin and the zero dB crossover of the open-loop system, the compensator can be determined. In the frequency domain, the transfer function of the phase lead controller is

$$G_c(s) = K\frac{(s+z)}{(s+p)}, \tag{3-18}$$

where $p = az$. The factor a determines the frequency separation (ratio) of the pole and zero and therefore the maximum phase angle associated with the controller. The phase angle will be between 0 and 90°. In addition, since this is a lead compensator, $0\langle a\langle 1$. This places a limitation

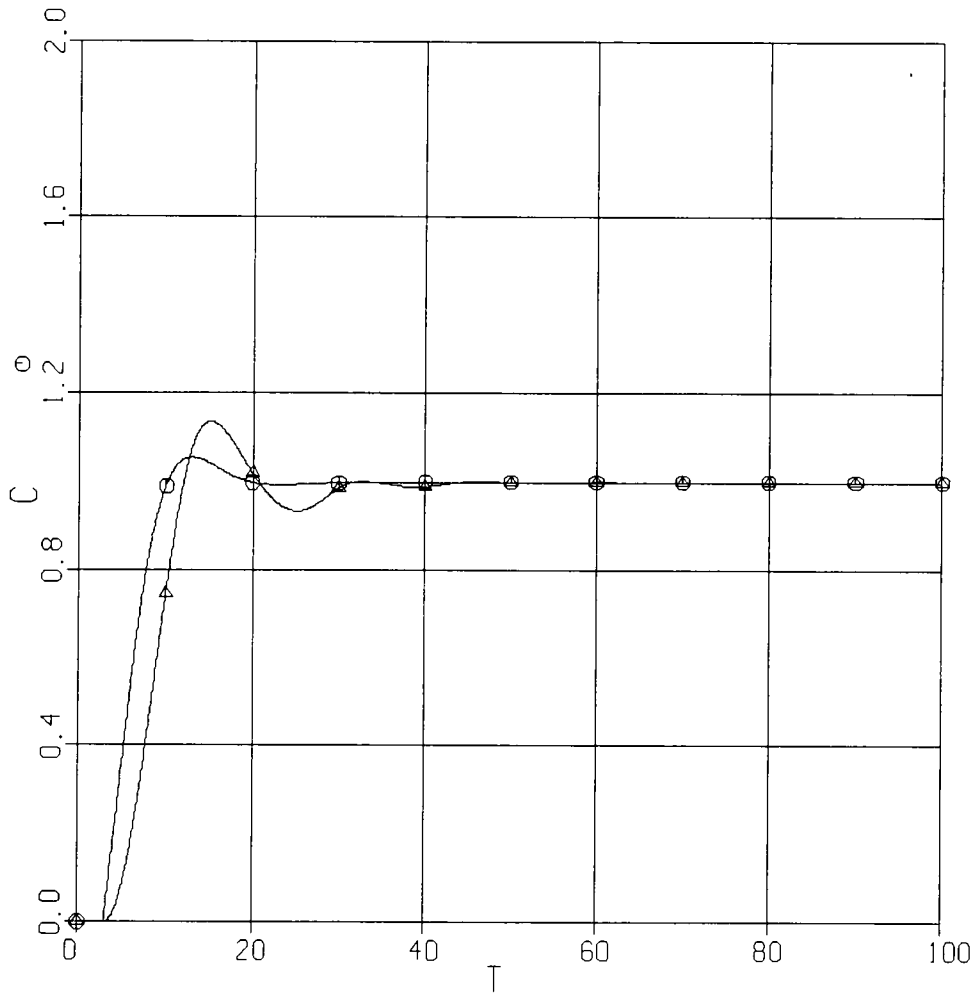


Figure 3-6

Step Responses for:

○ System with PI Compensation

Δ System with Tuned Fuzzy Controller

Table 3-2: Summary of Results of Example 3-2

Compen-sation ↓ / Attribute →	Overshoot	Peak Time	Settling Time	Steady State Error
GainCompensation	54.3%	7.85s	33.5s	0
PI Compensation	5.5%	13s	17s	0
Untuned Fuzzy	6.7%	51s	72s	0
Tuned Fuzzy	13.5%	15s	28s	0

on how much the controller can improve the phase margin of the system, but there will be an improvement in speed of response and overshoot if the pole and zero are properly placed. Taken in proper perspective, the main use of the lead compensator is to improve the speed of response. The next section discusses the fuzzy controller in relation to the phase lead controller.

3.4.2 The Fuzzy Controller as a Lead Compensator

The PD type fuzzy controller is capable of achieving a similar type of improvement in the system response as the lead controller described in section 3.4.1. Specifically, as mentioned in section 3.2.2, the PD type fuzzy controller improves the speed of response and reduces overshoot, due to its similarity to the classical PD controller. The nonlinear nature of the controller precludes frequency domain comparisons such as phase angle and bandwidth considerations. To illustrate these similarities, an example is given in the next section. The design (tuning) of the fuzzy lead controller is addressed in section 3.5.

3.4.3 Example 3-3: Lead Compensator Comparison

This example compares the overshoot, peak time and settling time of the step response

of a phase lead compensated system as well as the ramp error coefficient to those of a PD type fuzzy controller compensated system. Consider a continuous unity negative feedback system in which the plant has a transfer function

$$G_P(s) = \frac{100K}{s(s+36)(s+100)}. \tag{3-19}$$

The compensated system should meet the following specifications:

T_P (time to first peak) of 0.1 seconds

Less than 20% overshoot for a step input

K_v (velocity error coefficient) of 40 sec^{-1}.

The system is first compensated classically using a lead compensator of the form given in Equation (3-18). A design that meets these specifications is [36]

$$G_c(s) = 2.82\frac{(s+25.27)}{(s+71.43)}. \tag{3-20}$$

The final compensated system open-loop transfer function is

$$G_P(s)G_c(s) = 406{,}080\left[\frac{s+25.27}{s(s+36)(s+100)(s+71.43)}\right], \tag{3-21}$$

where K is 1440 for K_v =40.

The PD type fuzzy controller was used to compensate the same system. The untuned fuzzy controller gives a sluggish response compared to gain compensation only. The step response of the untuned fuzzy controller is shown in Figure 3-7 along with the response of the

system with gain compensation only. Figure 3-8 shows the step responses with lead compensation and with the tuned fuzzy controller. The tuned fuzzy controller gives a response comparable to that of the lead compensator, and the results are tabulated in Table 3-3.

The fuzzy controller required a gain increase for the untuned result in Figure 3-7. Before this gain increase the response could not be observed on the scale shown. As seen in Figure 3-8 and Table 3-3, the fuzzy controller performance exceeded the specifications. The ramp error coefficient was obtained by setting the input to a ramp and recording the steady state error.

3.4.4 Lag Compensator Comparison

For comparison purposes, a tabulation of results using a lag compensator and a PI type fuzzy controller to compensate the system from Example 3-3 is given in Table 3-4. Recall that the plant transfer function is given by:

$$G(s) = \frac{100K}{s(s+36)(s+100)}. \tag{3-19}$$

Plots of the step responses are given in Figure 3-9. As expected, the lag and PI type fuzzy compensators give a slower response than the lead or the PD type fuzzy controller. The fuzzy controller did not respond as fast as the classical controller, but had less overshoot. In the next section, a tuning procedure for the fuzzy controller is presented.

3.5 Tuning the Fuzzy Controller

Tuning the fuzzy controller is a challenging task because of the lack of well developed tools for use by the designer. In the classical case, tools such as Root Locus, Bode and Nyquist plots and the Routh-Hurwitz criterion are all well known and readily adaptable to graphical as well

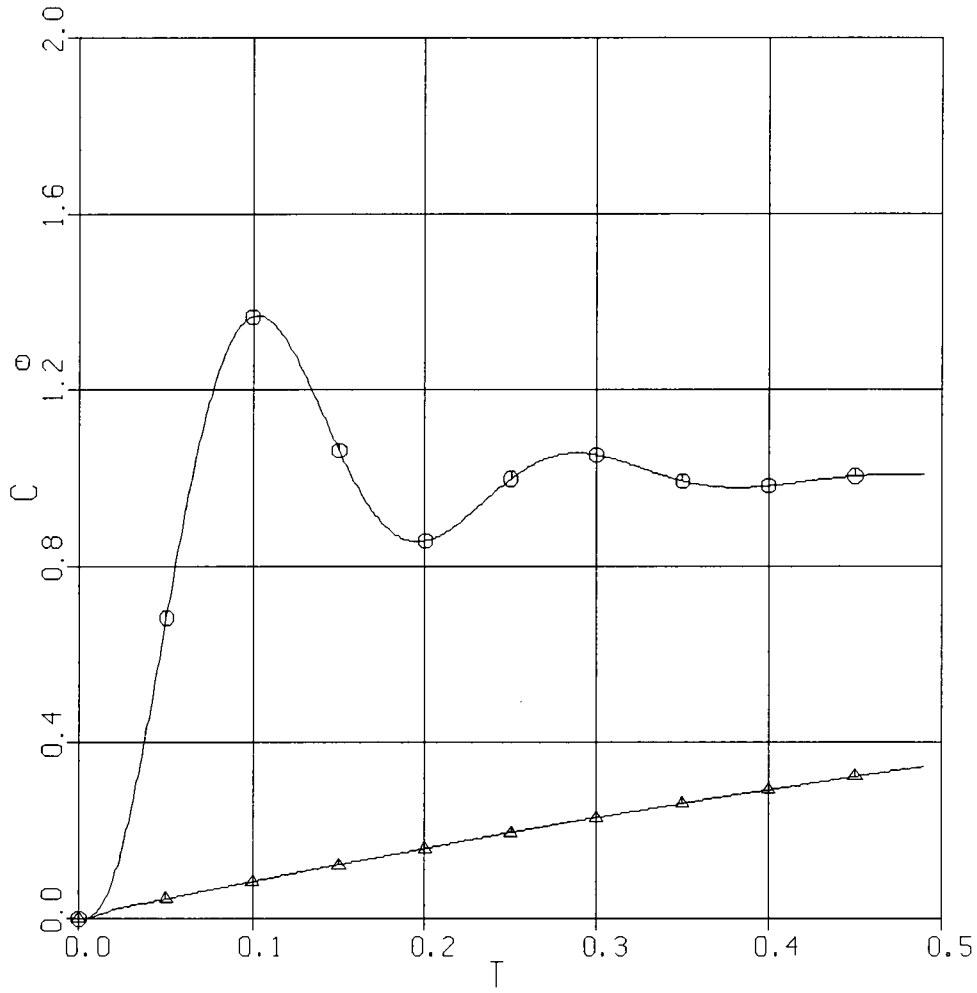


Figure 3-7

Step Responses for:

○ Gain Compensation Only

Δ Untuned Fuzzy Controller

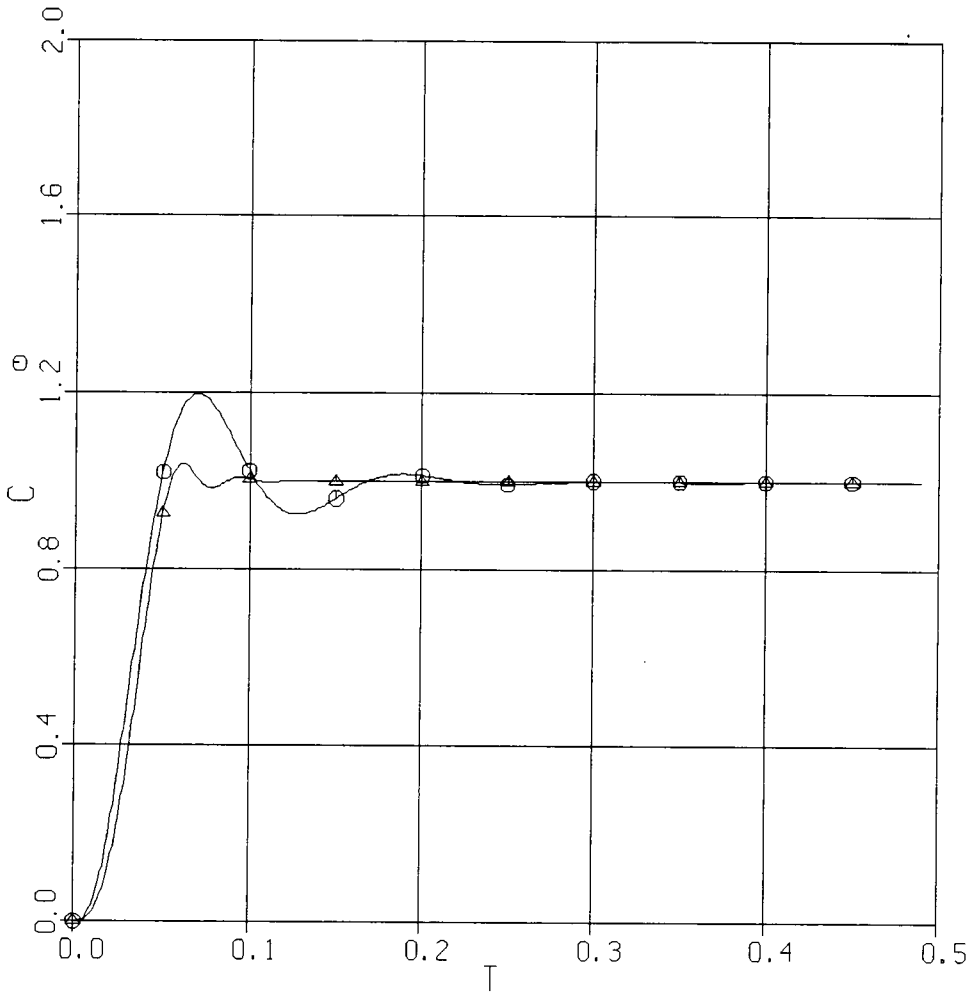


Figure 3-8

Step Responses for:

○ Lead Compensation

Δ Tuned Fuzzy Controller

Table 3-3: Summary of Results from Example 3-3

Compen-sation ↓ / Attribute →	Overshoot	Peak Time	Settling Time	K_v
GainCompensation	37%	0.103s	0.39s	40 sec^{-1}
Lead Compensation	20%	0.07s	0.158s	40 sec^{-1}
Untuned Fuzzy	NA	NA	NA	NA
Tuned Fuzzy	3.9%	0.061s	0.068s	60 sec^{-1}

Table 3-4: Summary of Results Using Lag Compensator

Compen-sation ↓ / Attribute →	Overshoot	Peak Time	Settling Time	K_v
Lag Compensation	17%	0.164s	0.586s	40 sec^{-1}
Tuned (PI) Fuzzy	6.5%	0.25s	0.383s	∞ sec^{-1}

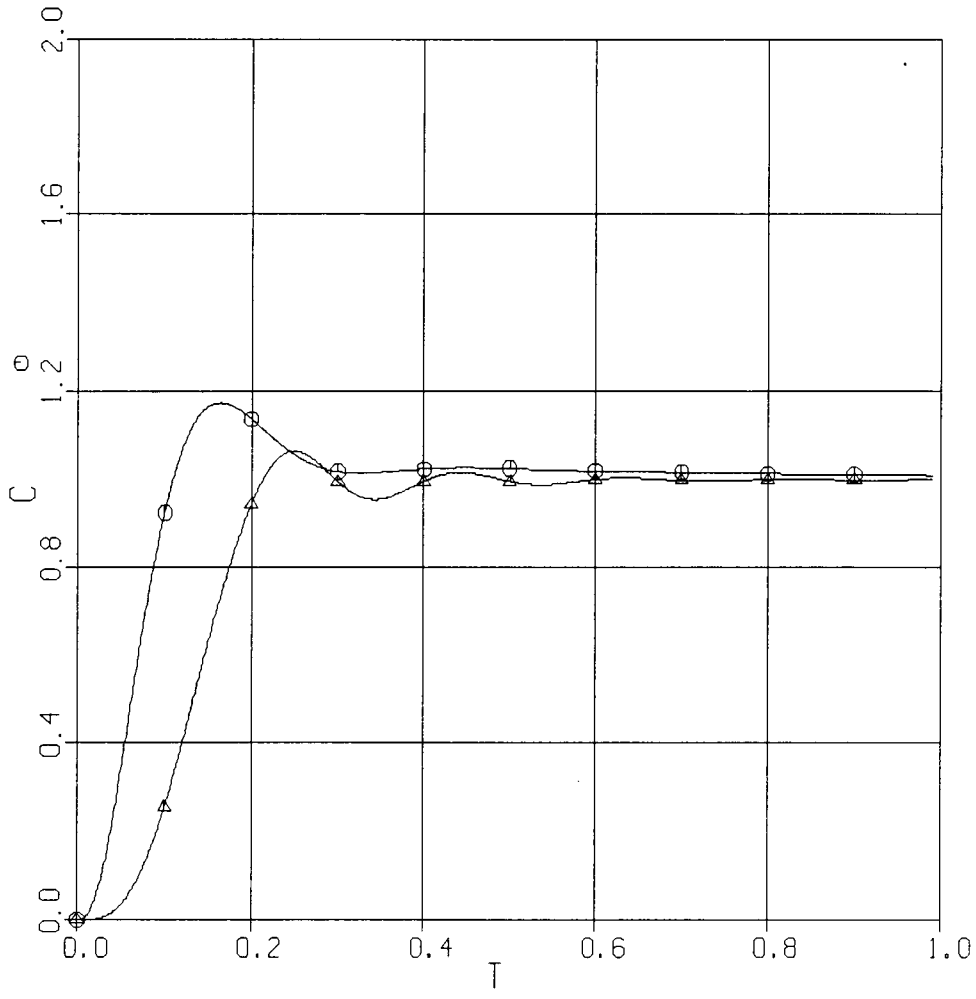


Figure 3-9

Step Responses for:

○ Lag Compensation

Δ Tuned (PI) Fuzzy Controller

as computer aided design. For the fuzzy controller, trial and error is the dominant tuning technique. As such, it has not been a simple matter to discuss stability or steady state error of the compensated system until it is tested (either physically or by simulation). Since in some cases it could be disastrous to physically test a system without having a reasonably good idea of what to expect, computer simulation is the principal testing procedure in use. In Chapter 1 an overview of some of the research performed with respect to fuzzy control system analysis was given and such research continues today. However, until the theory is further developed, the trial and error (simulation) method will persist.

Tuning the fuzzy controller does have a somewhat structured approach [22], based on experience with changing the parameters. There are three main parameters that can be adjusted to improve the response of the fuzzy-compensated system. These are : 1) Scaling Factors, 2) Individual Rules (Rulebase Entries) and 3) Membership Functions. Each of these parameters and its effect on the system response is discussed in the following sections. The details of tuning as applied to previous examples will be given.

3.5.1 The Effects of Changing Scaling Factors

Referring to Figure 3-1, the input variables e and $\dot{e}$ are multiplied by scaling factors K_e and K_{ed} before being used to determine the crisp output of the controller. As mentioned in Section 3.2.2, from a transfer function standpoint, these scaling factors play a role similar to the coefficients of the classical PD controller. From an application standpoint, they play a different role, that is, they actually condition the signals to be compatible with the fuzzy controller. In the sense that analog signals are scaled to match the ranges of computer interface circuits and transducers, the error and error rate signals are scaled to match the Universe of Discourse of the fuzzy rulebase. This scaling is not imperative to the extent that the system will fail without it, but scaling the inputs allows the entire rulebase to be effective rather than only a few rows. To

illustrate this point, consider the matrix in Table 3-5, in which the individual cells have been assigned identification numbers from 1 through 49.

If the input signal to the system is a unit step function and the system is stable, then the error can be expected to remain within the range $-1 \leq e \leq 1$. If the center value for NL is -1 and for PL is +1, then the Universe of Discourse matches the range of the error variable. If however, the Universe of Discourse ranges from -6 to 6, as described in Chapter 2, then in order to match the

Table 3-5: Rulebase with Numbered Cells

$e \backslash \dot{e}$	NL	NM	NS	ZE	PS	PM	PL
NL	1	2	3	4	5	6	7
NM	8	9	10	11	12	13	14
NS	15	16	17	18	19	20	21
ZE	22	23	24	25	26	27	28
PS	29	30	31	32	33	34	35
PM	36	37	38	39	40	41	42
PL	43	44	45	46	47	48	49

two ranges, the Universe of Discourse would require reduction to the range -1 to 1. It is easier, however, to retain the -6 to 6 Universe of Discourse scaling and simply scale the error signal. For instance, if the error will range from -1 to 1 and the Universe of Discourse ranges from -6 to 6, then multiplying the error signal by 6 would match the ranges. This has the same effect as changing the Universe of Discourse but is simpler to achieve.

To see why scaling is important, consider what happens if the Universe of Discourse ranges from -6 to 6 and the error signal is not scaled. If the scaling factor K_e is left at 1, then the only rules having any possible effect on the system would be rules 15-35, assuming that the error rate matches its Universe of Discourse. While this situation would not cause any catastrophic failure, if there are 49 rules available, it makes sense to try to use them all. In addition, if the scaling is such that all 49 rules are effective, the controller is more versatile, since the rulebase

entries can also be adjusted. Furthermore, scaling the error rate signal must be considered, for the same reasons, although it is more difficult to predict the range of the error rate.

The output scaling factor, K_u , is analogous to open-loop system gain and can be used to achieve gain compensation for the system. Depending on how accurate the designer's model of the system is (if there is one), this scaling factor can be adjusted with some confidence, based on knowledge of classical gain compensation. In summary, the scaling factors play an important role in tuning the fuzzy controller because they can maximize the number of effective rules, which in turn allows adjustment of additional rules. The scaling factors are also similar in nature to the coefficients of the classical PD controller which gives some guidance in further tuning their values. Changing K_e affects the speed of response of the system and changing K_{ed} affects overshoot.

The next section presents the effects of adjusting the rulebase entries.

3.5.2 The Effects of Changing Rulebase Entries

In Chapter 2, the Standard Rulebase was presented in Table 2-2. As described in Section 3.2.2, the entries in this table are based on an intuitive approach to shape the step response through the judicious application of the proportional and derivative terms. When tuning the fuzzy controller, these entries can be changed to improve the response. As an illustration, consider Table 3-5. Increasing the entries in cells 29,30,31,36,37,38,43,44 and 45 increases the speed of response. The reason these entry changes effect this type of change is evident when the rulebase is compared to the phase plane. In the phase plane, the region these cells corresponds to is one in which the error is positive and the error rate is negative, or the system output is increasing toward the setpoint (see Figure 3-2). If these entries are increased, then the controller input to the plant is increased *but only when the error and error rate are in this region.* This has an advantage over simply increasing the overall gain of the system, because overall gain affects *all* the regions of the phase plane. The global effect of changing overall gain leads to

the familiar trade off between rise time and overshoot. By changing the entries in individual cells, the effect of the gain increase is selective to a specific region of the phase plane and the trade off is lessened. Using similar reasoning for rulebase entries, the entries in the left hand side of the rulebase can be made more negative to decrease overshoot. This process makes the controller output more prohibitive of increasing plant output, but can be selected to affect only the region of the phase plane in which increasing plant output is undesirable. By contrast, increasing K_{ed} increases damping in a global sense, affecting the entire phase plane. These are some of the changes to the rulebase that have been noted to be effective in the tuning procedure. Specific examples will be given in Section 3.5.4.

It should be noted that in normal (stable) operation, some of the 49 rules are seldom used. For instance, rule 1 applies if the error and error rate are both large and negative, a condition which corresponds to the system output rising above the setpoint and increasing rapidly. This situation implies an unstable system, which would be avoided if possible. Similarly, the rules in the extreme lower right-hand corner of the rulebase should seldom be needed, if the system is stable. The next section discusses another tuning possibility, that of changing membership functions.

3.5.3 Changing the Membership Functions

The response of the system is sensitive to changes in the width and position of the membership functions. It is not intuitively obvious what effects such changes will have on the system response. Most of the literature advocates changing the shape of the membership functions with the assistance of neural network algorithms. Since this entails another topic of research, this method of tuning is not considered here. The next section explains the thought processes involved in tuning the controllers presented previously.

3.5.4 Examples of Tuning Fuzzy Controllers

This section begins with the plant from Example 3-1. In Figure 3-2,. the untuned fuzzy controller response is seen to have 28.4% overshoot, and a settling time of 42.7 seconds. To achieve the final response, which had a settling time of 24.4 seconds, and 6.9% overshoot, the first step was to choose an initial value for the scaling factors. If the error scaling factor K_e is increased, the overshoot will also increase. Because the system gain, K, was set to 3.0, K_{ud}, the output scaling factor for the PD option in the program, was set to 0.333. This effectively cancelled K, allowing K_e to be increased to 3.0 without appreciably affecting the response. Next, K_{ed} was adjusted to match the range of $\dot{e}$. The final value was K_{ed} =0.8, but this was not reached until after several iterations. The next step was to adjust rulebase entries. The step response was not necessarily slow (the rise time was comparable to the classical controller) , but it exhibited excessive overshoot. Therefore, rulebase entries in cells 29,30 and 31 (see Table 3.5) were changed to -6 (NL), and cell 39 to 2 (PS). These changes had the effect of increasing the damping of the system and thus reduced the overshoot. After each change, the ranges of e and $\dot{e}$ were checked to decide whether further adjustments could be made.

Example 3.2 compared the PI controller to a PI type fuzzy controller for the same system. The classical design required a significant reduction in gain to reduce the overshoot. The fuzzy controller required a gain reduction for stability. Through trial and error, K_{ui}, the output scaling factor for the PI option in the program, was given a value of 0.02. This reduced the output to a workable range. The next step was to choose K_e and K_{ed}. Through trial and error, settings of K_e =6 and K_{ed} =25 were chosen. The high value of K_{ed} was necessary to reduce overshoot. The rulebase entries were adjusted next. The values in cells 29,30 and 31 were set to -4. During each iteration, the same ideas were used:

1. K_e matches the inputs to the rulebase and actually speeds up response.

2. A larger K_{ed} reduces overshoot.

3. More negative values in the left hand side of the table increase damping and reduce overshoot.

This approach is still based on the relationships of the scaling factors to rise time and overshoot taken from the classical PD controller even though it was mentioned in section 3.2.2 that the roles of the scaling factors were reversed in the case of the PI type fuzzy controller. This approach is still valid because the scaling takes place before the integration. Therefore, the relationship between the scaling factors and the rulebase output in Equation (3-4) still holds. It is only after the output is chosen from the rulebase that the integration takes place. Next, the lead compensation example demonstrates how cells in the same region of the rulebase can speed up the response if they are made more positive.

In the lead compensation example the PD fuzzy controller was used to achieve similar performance to the lead controller. The untuned fuzzy controller achieved a sluggish step response and additional gain was required. This initial gain was added by setting K_{ud}=36. The response at that point was shown in Figure 3-7 as " untuned". Since in this case the speed of response required improvement and overshoot was not yet a problem, K_e and K_{ud} were increased but K_{ed} was not changed from its initial value of 1. In addition, several rulebase entries were increased in order to speed up the response. In the final design the scaling factors were set to K_e=6, K_{ed}=1 and K_{ud}=360. As for the rulebase entries, the values in cells 43,44 and 45 were set to 6, and those in cells 36,37 and 38 were set to 5. The final step response was shown in Figure 3-8, demonstrating that the system met specifications. This example shows that:

1. Increasing K_e and K_{ud} speeds up the response.

2. Increasing entries in the lower left hand corner of the rulebase speeds up the response.

It should be emphasized that the final combination of tuning factors is not unique; there are probably many other combinations that would work as well and possibly better. Such is the nature of a trial and error tuning process. The next section summarizes the results of these tuning examples.

3.5.5 Observations and Findings Regarding Tuning

The first step in tuning fuzzy controllers is to choose the initial scaling factors. An appropriate choice is one which matches the ranges of the input variables to the rulebase and the ranges of the controller output to the plant. These need not be the final choices, as further improvement in the response may be possible by readjusting the scaling factors. The next step is to adjust the rulebase entries. These adjustments are dictated by the type of improvement needed in the plant response. For instance, increasing entries in the lower left hand corner of the rulebase can decrease response time while decreasing those entries can increase damping. Entries in the upper left corner affect overshoot, while those in the upper right hand corner affect undershoot. This is evident when the rulebase is compared to the phase plane. Each quadrant of the phase plane corresponds to a different phase of the transient response (see Figure 3-2). Entries in each quadrant impact the transient response in the corresponding phase. Once it is determined that the best response has been obtained through changing the rulebase entries, the scaling factors can be adjusted again. This has the potential for further improvement in the response, and once improvement is achieved, the rulebase entries can be readjusted. The tuning process can proceed in this manner until the designer is satisfied that the specifications are met or that no further improvement can be made. This process can be time consuming and tedious and still may not produce an optimum design. For this reason, research in the area of autotuners for these controllers has recieved significant attention [27,28].

This section has presented the details of tuning the fuzzy controllers from the previous examples and summarized the findings and observations made during the process. The next

section introduces a way to predict another measurable specification that can assist in the task of tuning, the steady state error.

3.6 Error Coefficients for the Fuzzy Controller

In the design of classical control systems, the specifications often include a constraint on the steady state error. The steady state error is related to the open-loop system gain which in turn has an impact on system response time and stability. It would be advantageous to have such a constraint available to the fuzzy controller designer since there is little guidance available with respect to scaling factor settings. This section presents a method of predicting steady state error for fuzzy controllers that provides the desired constraint and specification.

3.6.1 Predicting Steady State Error for Fuzzy Controllers

It is generally held that the nonlinearity of the control algorithm inherent in fuzzy controllers precludes prediction of steady state error using classical linear methods. As a result, research has been performed that develops methods of steady state error analysis of the fuzzy control system using nonlinear methods [17,18,19]. Nevertheless, a study of the Standard Rulebase reveals an interesting relationship for cases in which the error rate is zero (steady state). Recall that the Standard Rulebase is based on symmetrical membership functions (triangles). This symmetry is necessary for the following analysis. Referring to Table 2-3, if the system is in the steady state, the error rate is zero and the output of the rulebase (and therefore the controller) comes solely from the center column. The values for e are listed in the leftmost column. Comparison of these two columns shows that they are identical. Therefore, if the error signal is *not* scaled, the output of the rulebase is equal to the error, and if the error *is* scaled, the rulebase output is equal to the scaling factor multiplied by the error. This relationship is shown as

$$U_{ss}(t) = K_e e(t). \tag{3-22}$$

This equation is identical to that of a feedback control system with gain compensation only. It must be emphasized that Equation (3-22) holds only for symmetrical triangles. To see why this is so, consider the case for which the designer has increased the slope of the left side of the PS triangle for error to 1.0. The original slope was 0.5 (see Figure 2-1). If e=0.2 and $\dot{e}$=0 the membership values will be skewed toward PS, making the controller output larger than it would have been for symmetrical membership functions. It is therefore seen that in the steady state the fuzzy controller is transparent to the control system, and the system possesses the characteristics of a *linear* system. Since this is the case, the same error coefficients that apply to linear control systems must also apply to fuzzy control systems. These error coefficients usually imply a knowledge of the transfer function of the plant. For the fuzzy controller the error coefficients are proportional to the original error coefficients associated with the plant.

In the case of a unit step input, a type 0 system with transfer function $G_P(s)$ has a steady state error of

$$e_{ss} = \frac{1}{1 + K_P} \tag{3-23}$$

where

$$K_P = G_P(0). \tag{3-24}$$

Since in steady state the fuzzy controller exhibits nearly linear behavior, the scaling factors K_e and K_{ud} convert to gain factors that become multipliers of the plant transfer function. Therefore, the error coefficient for a type 0 system with a unit step input is

$$K_{pf} = K_e K_{ud} K_p \,, \tag{3-25}$$

where K_{pf} signifies the position error coefficient of the fuzzy control system as opposed to K_p for the conventional system without the fuzzy controller. This is, as stated earlier, proportional to the original error coefficient. This equation holds for the PD type fuzzy controller but in the case of the PI type fuzzy controller an integrator is added, effectively increasing the system type by 1. The steady state error for a PI type fuzzy controller with a step input is therefore 0 and the error coefficient is infinite. The ramp error coefficient however is finite for this case and is found in the same fashion as the step error coefficient,

$$K_{vf} = K_e K_{ud} K_v \,, \tag{3-26}$$

where K_{vf} is the fuzzy ramp error coefficient and K_v is the original ramp error coefficient found in the normal way. If one considers a type 1 system, the steady state error is such that the PD type fuzzy controller has an infinite coefficient for a step input and a finite coefficient for a ramp input, while the PI type fuzzy controller gives an infinite coefficient for both inputs. These results are tabulated in Tables 3-6 and 3-7.

Table 3-6 Steady State Errors for the PD Type Fuzzy Controller

System Type	Step Input	Ramp Input
0	$e_{ss} = \dfrac{1}{1 + K_{pf}}$	$e_{ss} = \infty$
1	$e_{ss} = 0$	$e_{ss} = 1/K_{vf}$

Table 3-7 Steady State Errors for the PI Type Fuzzy Controller

System Type	Step Input	Ramp Input
0	$e_{ss} = 0$	$e_{ss} = 1/K_{vf}$
1	$e_{ss} = 0$	$e_{ss} = 0$

The next section uses the equations in Tables 3-6 and 3-7 to predict steady state error and assist in the choice of scaling factors.

3.6.2 Scaling for Steady State Error

The scaling factors in the previous examples were chosen based on matching the inputs to the rulebase and trial and error improvement of the step response. In addition to these criteria, steady state error can be included in the design and can assist in setting scaling factors. In Example 3-3, the Lead Compensator Comparison required a ramp error coefficient K_v =40. The specification was exceeded by the fuzzy controller and matches that given by equation (3-26), since K_e=6, K_{ud}=360, and K_p=0.02778. The result was K_v =60. The specification of an error coefficient actually was used in setting the scaling factors in this example and they were adjusted to meet the requirement. In classical design one of the first things done is to set the gain to meet the steady state error specification. This can also be done with the fuzzy controller, and would allow selection of scaling factors early in the design (tuning) phase.

3.7 Summary

In this chapter the two types of fuzzy controller, the PD type and the PI type, have been defined and described within the framework of classical control theory. The performance of the fuzzy controllers was compared to that of the corresponding classical controllers and it was shown that the fuzzy controllers give a comparable quality of control. This facilitated the readers' understanding of the fuzzy control process, since it was presented in terms familiar to the engineer.

In the examples, the PD type fuzzy controller exhibited better relative stability due to its selective damping than the classical PD controller, but a slightly slower response. When compared to the classical lead controller, the PD type fuzzy controller gave a faster response with less overshoot and a better settling time. In the PI comparison example the PI fuzzy controller gave a response with longer rise time and settling time and slightly higher overshoot than the classical design.

The tuning process was explained for the examples and some overall conclusions on tuning were presented. Steady state error was discussed and an interesting observation allowing the prediction of error was presented.

The next chapter examines the nonlinear nature of the fuzzy controller and compares it to a linear lookup table. The Standard Rulebase is shown to be a useful tool in tuning when the cells that are activated during the transient period are determined. The disturbance rejection characteristics of the two types of fuzzy controller are compared with that of a classical lead-lag controller.

CHAPTER FOUR

LINEARITY AND DISTURBANCE REJECTION

4.0 OVERVIEW

The implementation of the fuzzy controller was presented in Chapter 3. In the discussion of the simulation program it was mentioned that the fuzzy controller exhibits a nonlinear relationship between e and U with $\dot{e}$ fixed. This nonlinearity only occurs between the cell to cell values. The nonlinear transition will be further explored in this chapter and compared to a cell to cell rulebase which exhibits a linear relationship (linear interpolation). The system performance for the two cases will be compared.

The rulebase for a fuzzy compensated system after tuning is presented with an added feature. The active cells will be indicated on the rulebase as the system output goes through the transient. This provides an interesting viewpoint of the controller action. It may also provide some guidance in tuning the controller.

The disturbance rejection characteristics of the fuzzy controller are investigated. The PD type and PI type fuzzy controller will be compared to a conventional controller.

4.1 Nonlinear Characteristics of Fuzzy Controllers

4.1.1 The Direct Fuzzy Algorithm

In the Examples in Chapter 3 the simulation program used was a direct implementation of Equation (2-4). Other implementation methods are possible. In Section 2.2.1.1 it was shown that this Equation results in a nonlinear input/output relationship. An example of this was shown in Figure 2-3 for $\dot{e}$ =-1. Other values of $\dot{e}$ yield similar curves, with the exception of $\dot{e}$ =0. When

the error rate is not changing, the mapping is linear. However, if the triangles are non-symmetric, the relationship between e and U is nonlinear even with $\dot{e}$ =0.

Another way to implement the fuzzy controller is to use a lookup table. This involves calculating the controller output for various input values and tabulating the results. The amount of resolution determines whether the nonlinearity is preserved. Many applications use quantized membership functions, and these designs achieve good results [25, 30]. One reason the lookup table is popular is that computation time is reduced. This is important when implementing the fuzzy controller in real time. However, there is an inherent loss of resolution in this method. The question arises as to whether the quantized lookup table gives comparable results to the direct fuzzy method. This question would require additional research, especially for a broad spectrum of plant configurations including linear and nonlinear plants.

Another possibility is to use a lookup table that uses linear interpolation between entries. This approach does not preserve the nonlinearity of the fuzzy controller. The work in the following section shows that for some cases the difference between the two methods is negligible. An easy way to implement such a table is through the use of the TABLE command in ACSL [34]. The computer has only to retrieve the controller output from the table based on the values of the inputs.

The next section compares the linear lookup table with the direct fuzzy method.

4.1.2 Example 4-1: Fuzzy Method Comparison

Consider the plant from Example 3-3. The open-loop transfer function is

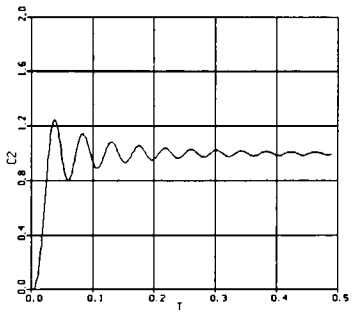
$$G(s) = \frac{100K}{s(s+36)(s+100)}. \tag{4-1}$$

The objective in this example is to compensate the plant with the direct fuzzy method used in Example 3-3 and compare to compensation using linear interpolation. This was done for five different tuning points such that the difference between the two compensation schemes could be compared under varying conditions. The difference between the system outputs was then calculated and plotted. Figure 4-1 shows the step response of the direct fuzzy controller, Equation (2-4) for the 5 tuning conditions. Figure 4-1 (a) represents higher values of scaling factors and 4-1 (e) represents lower values (less oscillation). The step responses for the linear interpolation method were similar. Figure 4-2 shows plots of the difference between the two methods. The difference is seen to be smaller for the tuning conditions corresponding to the better step response. Experiments have been performed [37] that allow the user to vary the amount of nonlinearity exhibited by the fuzzy controller. The results of these experiments imply that more nonlinear plants require more nonlinear behavior in the controller. This is not conclusive and warrants further research work.

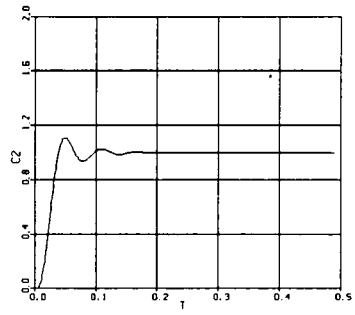
The next section presents the rulebase with an indication of which cells are active during the transient response.

4.2 The Rulebase as Phase Plane

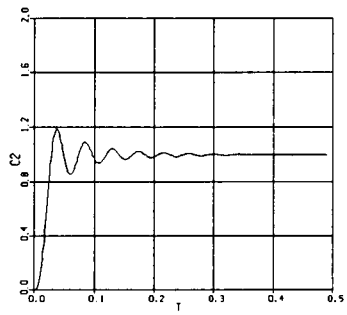
An interesting look at the rulebase is obtained when the transient is studied to determine which cells are active at different points on the step response. Consider Figure 4-3, where the fuzzy controlled step response of the system in Section 4-1 is divided into 10 ms portions (on a shorter time scale). For each of these points on the response, the scaled values of e and $\dot{e}$ were determined by printing them out in ACSL. Table 4-1 shows the fuzzy rule *most* active for each of these points. The active cells are shaded and numbered in accordance with the points shown in Figure 4-3 (t_0 -t_8). Table 4-2 is a similar tabulation using linear interpolation. The cells are referred to as *most* active because



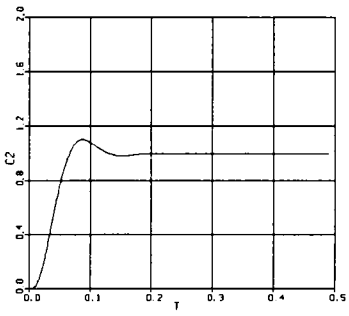
(a)



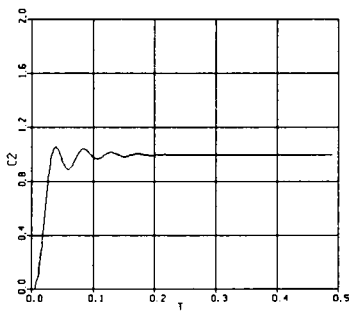
(d)



(b)



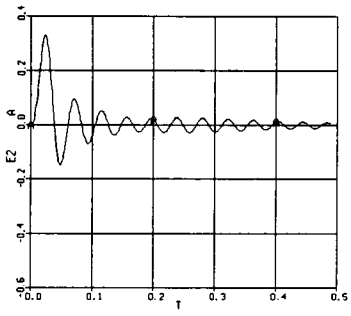
(e)



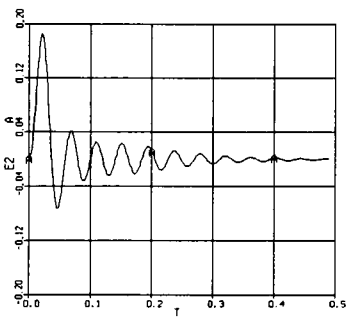
(c)

Figure 4-1

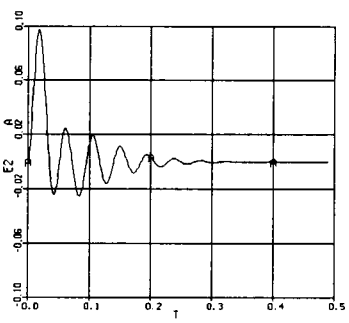
Step Responses Using Direct Fuzzy Method and Different Tuning Points (Example 4-1)



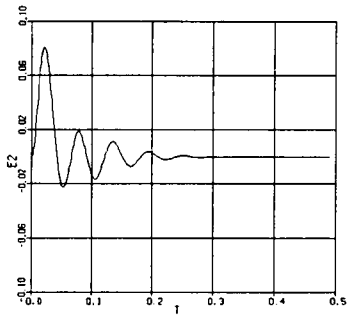
(a)



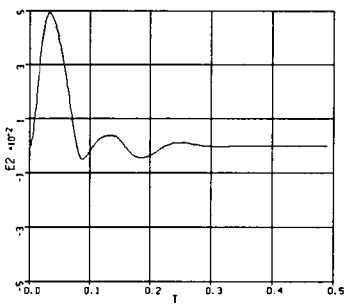
(b)



(c)



(d)



(e)

Figure 4-2

Calculated Difference between Direct Fuzzy

Method and Linear Interpolation (Example 4-1)

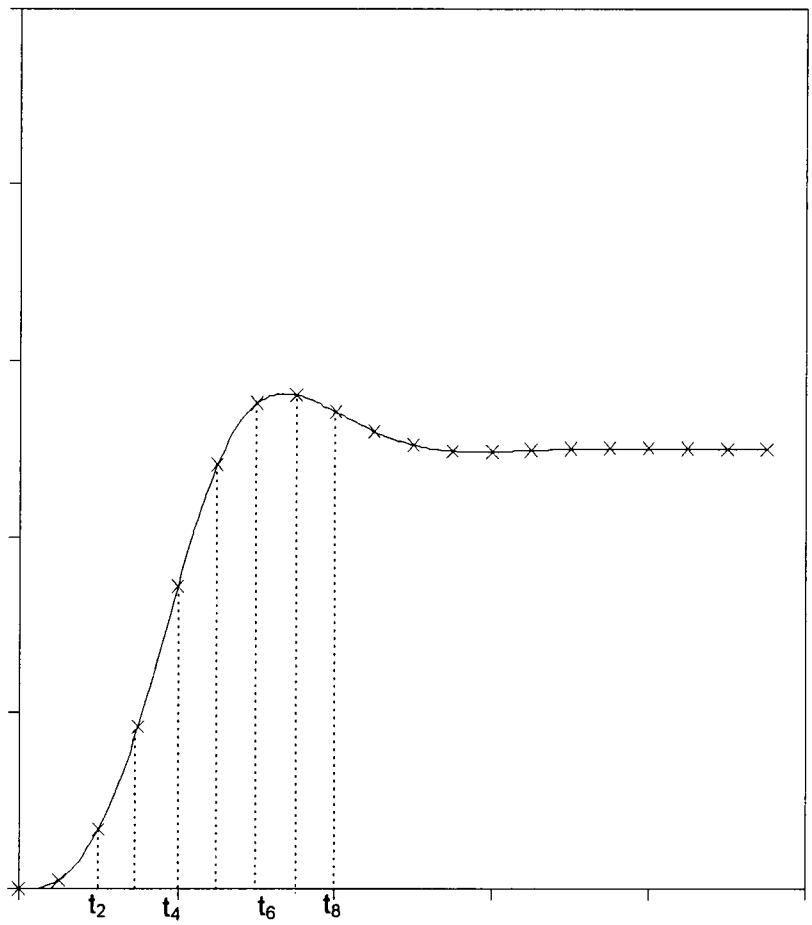


Figure 4-3
Step Response with Numbers Corresponding to Rulebase

TABLE 4-1: ACTIVE CELLS IN FUZZY RULEBASE

e \ $\dot{e}$	NL	NM	NS	ZE	PS	PM	PL
NL	NL	NL	NL	NL	NM	NS	ZE
NM	NL	NL	NL	NM	NS	ZE	PS
NS	NL	NL	$NM_{(t6)}$	$NS_{(t7)}$	ZE	PS	PM
ZE	NL	$NM_{(t5)}$	NS	$ZE_{(t8)}$	PS	PM	PL
PS	$NM_{(t4)}$	NS	ZE	PS	PM	PL	PL
PM	$PL_{(t3)}$	PL	PL	PM	PL	PL	PL
PL	PL	$PL_{(t2)}$	$PL_{(t1)}$	PL	PL	PL	PL

TABLE 4-2: ACTIVE CELLS IN LINEAR RULEBASE

e \ $\dot{e}$	NL	NM	NS	ZE	PS	PM	PL
NL	NL	NL	NL	NL	NM	NS	ZE
NM	NL	NL	NL	NM	NS	ZE	PS
NS	NL	NL	$NM_{(t6)}$	NS	ZE	PS	PM
ZE	$NL_{(t5)}$	NM	NS	$ZE_{(t7)}$	PS	PM	PL
PS	$NM_{(t4)}$	NS	ZE	PS	PM	PL	PL
PM	$PL_{(t3)}$	$PL_{(t2)}$	PL	PM	PL	PL	PL
PL	PL	PL	$PL_{(t1)}$	PL	PL	PL	PL

they are fuzzy subsets and neighboring cells may be partially active. The main observation to be made here is that this mapping gives specific information about which cells can affect the response. Therefore, Tables 4-1 and 4-2 can be used for additional tuning. Recall that in the original design, table entries for entire regions of the rulebase were adjusted. This was based on intuition. With the help of Table 4-1, the table entries can be adjusted more effectively, further improving the step response. In fact, adjusting the cells corresponding to times 1 and 2 to 10, and the cell corresponding to time 4 to -10 reduces the overshoot and speeds up the response. This is shown in Figure 4-4, where the overshoot decreased from 12.6% to 5% and the peak time decreased from 67ms to 59ms. This tabulation can therefore be a useful tool for tuning.

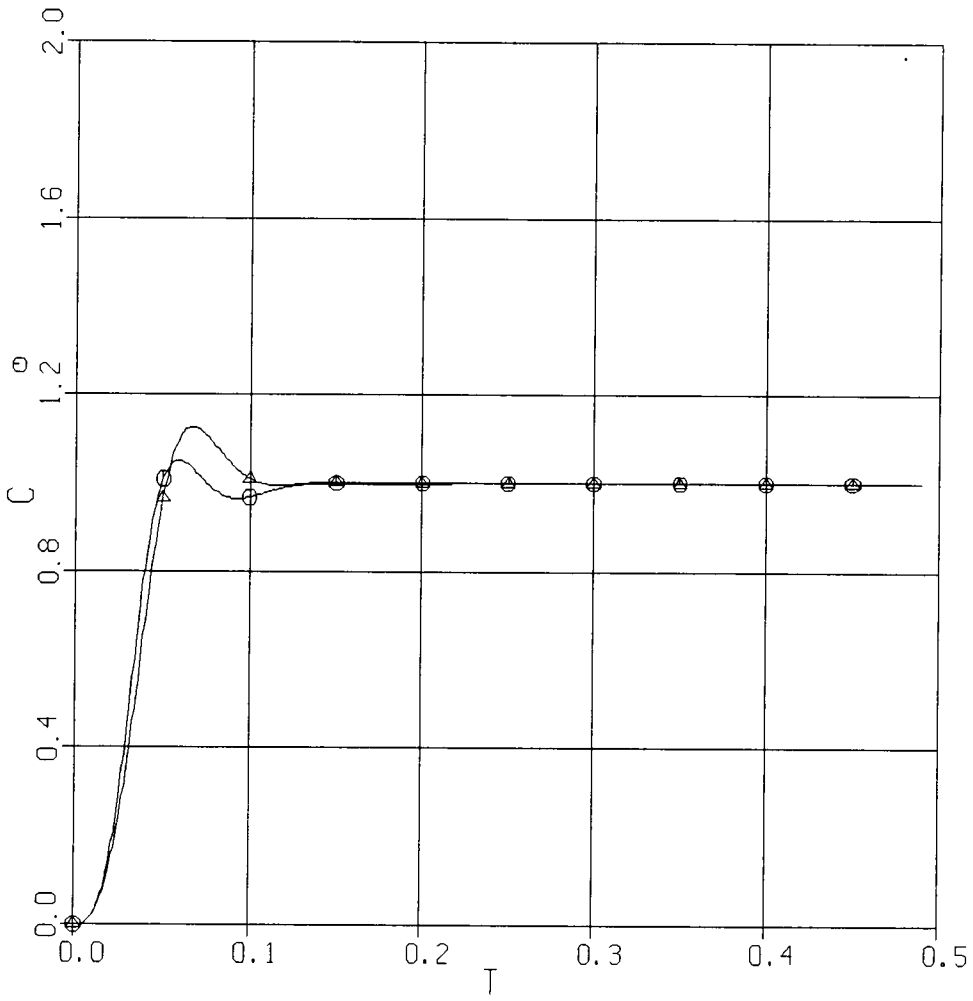


Figure 4-4

Step Response for:

○ System with Tuned Rulebase

Δ System with Additional Tuning from Table 4-1

4.3 Disturbance Rejection with the Fuzzy Controller

One of the most important characteristics for a control system to possess is that of disturbance rejection. In nearly any application, the process or device being controlled will experience changes in loading (torque, heat, pressure, etc.). Consequently, it is imperative that the controller be capable of recovering from disturbances in order to maintain control. In this section, the disturbance rejection characteristics of the fuzzy controller will be explored, for both the PD and PI types. These will be compared to a classical controller.

4.3.1 Disturbance Rejection Characteristics of the Fuzzy Controllers

To begin a discussion of disturbance rejection, it is necessary to re-examine the rulebase of Table 2-3. If a disturbance (assume a unit step disturbance) is introduced to the system just after the controller (and before the plant), the PD type fuzzy controller will not be able to completely reject the effects of the disturbance. To see why this is so, consider the output of the controller, U. What happens when the disturbance is introduced is that the input to the plant is no longer U, but $U+1$. In essence, this is the same as adding 1 to all the values in Table 2-3. All the 0s in the diagonal become 1s, all the 2s become 3s, etc. The result of this effect is described as follows: the fuzzy controller is based on If-Then statements, and those statements are designed to drive the plant to a state with a desired error value (in this case 0), which in turn causes the controller output U to be 0. The rules are designed that way. Thus when $e=0$ and $\dot{e}=0$, $U=0$. When a 1 is added to every cell in the table, the rules have been changed. Now the controller output is to be 0 when $e=-1$. The controller still performs well, but to a different set of rules. Scaling factors reduce the final value of e because the controller output will be 0 when $K_e K_u e=-1$. To eliminate the error, an integrator after the controller is necessary. A fuzzy configuration that already has an integrator before the plant is the PI type fuzzy controller, and indeed, this configuration has the capacity to reject the disturbance postulated above. Example

4-1 will demonstrate these findings with a model of an armature-controlled DC motor. The disturbance will be applied to the plant as a torque disturbance.

4.3.2 EXAMPLE 4-2: Comparison to Conventional Controller

Consider a position control system for a motor with the following transfer function:

$$G(s) = \frac{1}{s[(s+5)(s+10)+Kb]} \qquad (4\text{-}2)$$

where Kb is a back emf coefficient included to simulate an armature-controlled dc motor, as shown in Figure 4-5. For this example, Kb will be set to 1. To properly model a load torque disturbance, the disturbance, $T_d(s)$, must be introduced into the model after the armature time constant and before the load time constant so that

$$T_L(s) = T_m(s) - T_d(s) \qquad (4\text{-}3)$$

as shown in the block diagram of Figure 4-5. Although this is not the same configuration described in the discussion of the fuzzy rulebase above, it is more realistic, and the rejection characteristics are the same. For comparison, a lead-lag compensator will be simulated with this system. The compensator will have the transfer function [31]

$$G_c(s) = \frac{(s+4)(s+.1)}{(s+7.3)(s+.01)} . \qquad (4\text{-}4)$$

Figure 4-6 shows the step response of the compensated system. The overshoot=4.5% and the peak time=1.41 seconds. To demonstrate the disturbance rejection characteristics of the compensated system, the system in question was simulated with zero input and a disturbance

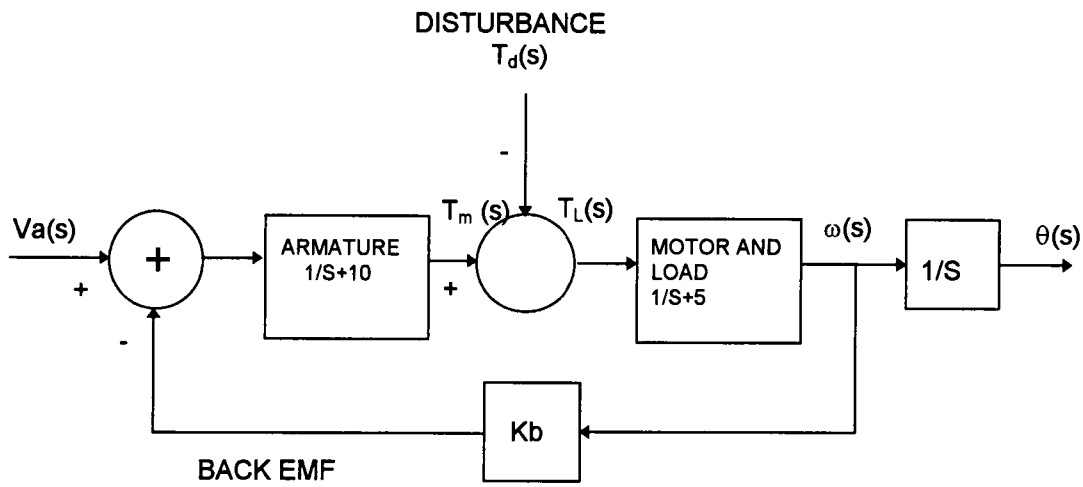


Figure 4-5

Block Diagram of Armature Controlled DC Motor with Disturbance

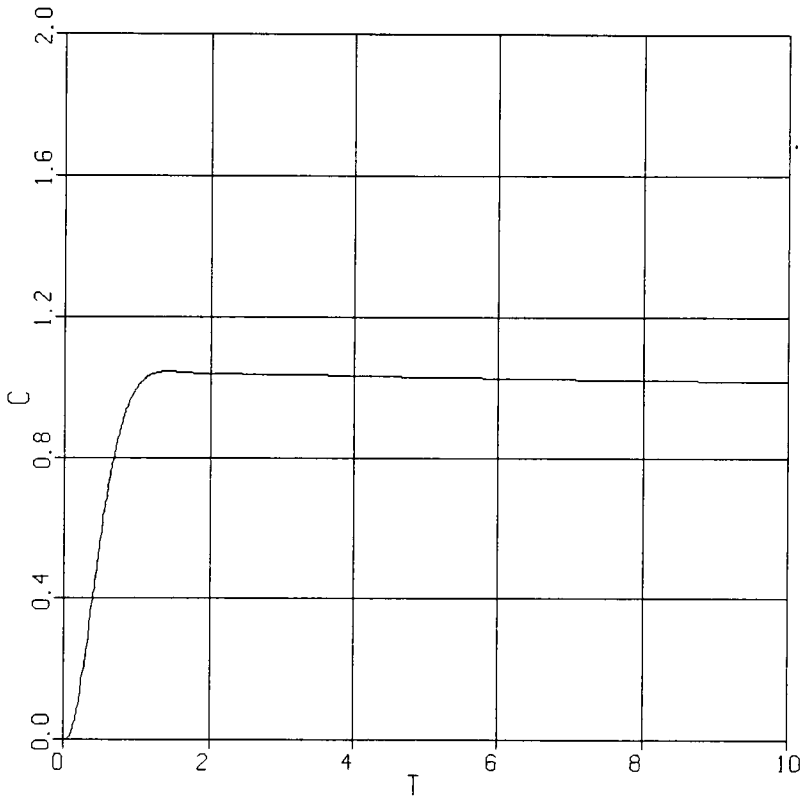


Figure 4-6

Step Response of Lead-Lag Compensated System of Example 4-2

$T_d(s) = \frac{-e^{-4s}}{s}$ applied to the plant. The results are tabulated in Table 4-3. Figure 4-7 shows the response of the system to the disturbance for the classical controller and both types of fuzzy controller.

Table 4-3 Disturbance Response Data for Example 4-2

	Peak Value	Final Value	Settling Time
Lead-Lag Controller	.46	.048	63s
PD Type Fuzzy	.353	.333	5.4s
PI Type Fuzzy	.43	0	10s
Combination Fuzzy	.13	0	11s

This example demonstrates that the fuzzy controllers can limit the disturbance response to maximum values comparable to the lead-lag controller, and actually settle faster than the classical compensator. Another interesting configuration is the combination of the fuzzy PD and PI controllers. Such a configuration would combine the zero error of the PI type with the smoother response of the PD type fuzzy controller with results as shown in Table 4-3. Figure 4-8 shows the results of this combination compared to the PI type fuzzy controller used in Example 4-2. This Figure shows the combination is capable of lower overshoot, zero error and a similar settling time.

4.4 Summary

In this chapter the difference between the direct fuzzy method, the quantized lookup table and the linear lookup table were presented. The direct fuzzy algorithm was compared to the linear lookup table in an example. The two methods gave similar results.

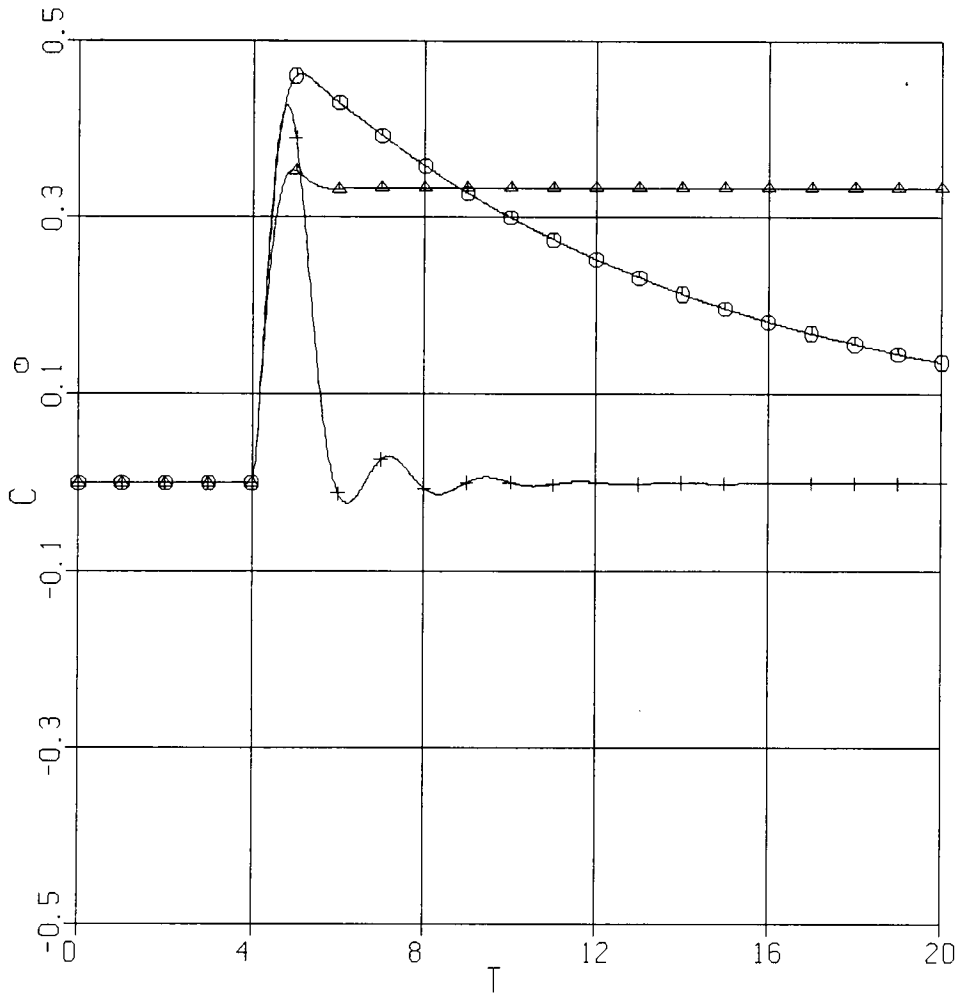


Figure 4-7

Disturbance Response for:

0 Classical Lead-Lag Controller

Δ PD Type Fuzzy Controller

\+ PI Type Fuzzy Controller

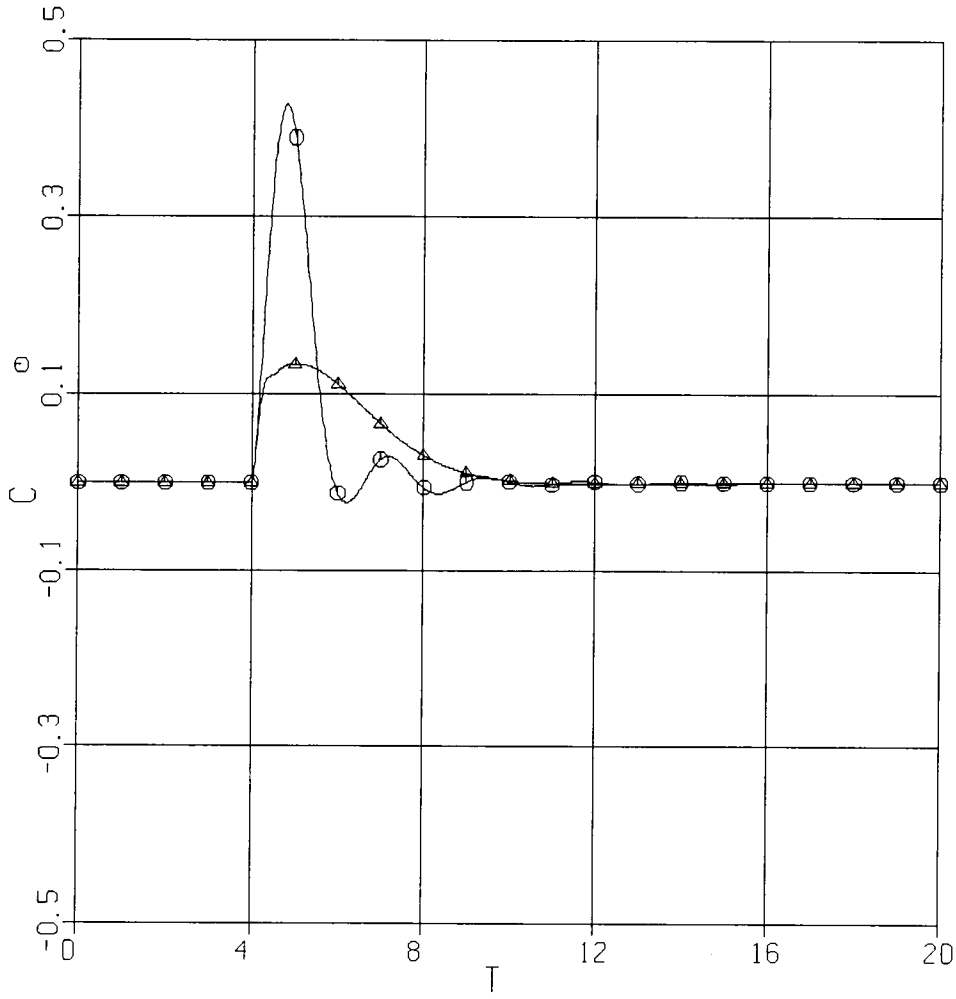


Figure 4-8
Disturbance Response for:

o PI Type Fuzzy Controller
Δ Combination of Fuzzy Controllers

The active cells in the rulebase during the transient were shown in Tables 4-1 and 4-2 for the two methods in Example 4-1. This is an interesting viewpoint, similar to a phase plane plot with fuzzy boundaries. The value of such a tabulation is that it lends guidance to the designer as to which cells should be further tuned. Three cells were readjusted for the system in Example 4-1 based on this table and the results were plotted. The plot shows an improvement in overshoot and peak time.

The disturbance rejection characteristics of the two types of fuzzy controller were compared. The results show that the PD type fuzzy controller does not eliminate the error from the disturbance while the PI type fuzzy controller does. The PD type fuzzy controller has a smoother response than the PI type. Both types have a faster settling time than the classical lead-lag controller in the example. Finally, the combination of the PD and PI type fuzzy controllers was shown to give a response with the desirable characteristics of both types.

The next chapter presents a summary and conclusions from the thesis.

CHAPTER FIVE

SUMMARY AND SUGGESTIONS FOR FURTHER RESEARCH

5.0 OVERVIEW

This thesis has presented a tutorial viewpoint of fuzzy control. Examples have been given that illustrate the fuzzy control strategy. The importance of this work is that it represents a sizable literature search and provides a good reference for persons starting a study of fuzzy control. One notable conclusion is that the fuzzy controller using the Standard Rulebase is similar to the classical proportional plus derivative controller. The reasons for this were presented in Section 3-2, and illustrated with an example. With a minor modification a proportional plus integral type fuzzy controller can be implemented. Section 3-3 examined this configuration and provided an example comparing the fuzzy PI controller to the classical PI controller. The PD and PI type fuzzy configurations can be compared to the classical lead and lag controllers. This was done in Sections 3-4 and 3-5. Examples were again provided for these comparisons. A section on tuning was provided based on what was learned through research. Tuning the fuzzy controller is a necessary but challenging task. Section 3-5 showed several steps that would be helpful to the reader attempting to implement a fuzzy controller. Steady state error prediction is usually reserved for linear systems. This is because the prediction relies on the final value theorem which uses the Laplace transform. The fuzzy controller is nonlinear and thus raises concerns about predicting steady state error. It is shown in Section 3-6 that for the special case of symmetrical membership functions equivalent fuzzy error coefficients can be defined. These fuzzy error coefficients function in the same way as those of a linear system. Additional interesting issues such as linearity, phase plane plots and disturbance rejection were presented in Chapter 4.

5.1 Summary

The history and successes of fuzzy control were highlighted in Chapter 1. The fact that fuzzy control has yet to receive wide acceptance was revealed and explained. The nonlinear analysis tools for fuzzy control are not well developed and this causes concern among practicing engineers.

In Chapter 2 the groundwork was laid for the application of fuzzy control. The concepts of fuzzy subsets and linguistic variables were introduced. Fuzzy theorems were presented which allow the manipulation of those linguistic variables. These concepts clarified the notion of developing linguistic control strategies and applying them to a physical system through " crisp " variables. This procedure centered around the fuzzification/defuzzification process. A concise representation of the control rules formed for a system was presented and used throughout the paper. This representation is known as a rulebase. An explanation of the implementation of the fuzzy controller was given using the rulebase.

The simulation program was explained in detail in Chapter 3. This explanation was based on the most straightforward program possible, which was included in the Appendix.
Comparisons of fuzzy controllers to classical controllers showed that comparable results are possible without any need for a mathematical model. Comparisons of the PD type fuzzy controller to classical controllers showed that the PD type fuzzy controller is capable of improving rise time and overshoot for a system. These comparisons were presented in Examples 3-1 and 3-2. Comparisons of the PI type fuzzy controller to the classical controllers were given in Examples 3-3 and 3-4. These comparisons showed that the PI type fuzzy controller can achieve zero steady state error. The response times were slower for the PI type fuzzy controller, however.

The steps for tuning fuzzy controllers were presented in Section 3-5. The first step was to adjust the scaling factors. Then the rulebase entries were tuned. These steps were performed in an iterative approach, using trial and error and some intuition. Several observations and recommendations were made based on experience. It was shown in Section 3-6 that for a fuzzy controller with the triangular membership functions used in this paper, the steady state error is predictable. This is true for symmetrical membership functions only (the sides of the triangles must have the same slope). Since many designers change membership functions as a part of the tuning process, the symmetrical membership functions must be considered a special case. For the case of symmetrical membership functions, it was shown that as long as $\dot{e}$ =0, the output of the controller is equal to the input multiplied by a scaling factor. In steady state, the scaling factor is thus playing the role of open-loop system gain. The error coefficients, K_p and K_v, can therefore be applied. Equivalent fuzzy error coefficients K_{pf} and K_{vf} were shown to be the same as in the linear system. This was demonstrated for Example 3-3.

The defuzzification algorithm interpolates nonlinearly between cells in the rulebase. This was explained in Section 2.2.1.1 and further examined in Section 4-1. The quality of control using the direct fuzzy algorithm was compared to that of linear interpolation. The results were nearly identical after tuning. More research is in order to fully understand the relationships between the performance of the controller and the interpolation of the algorithm.

An interesting approach to analyzing the fuzzy system response was presented in Section 4-2. The cells in the rulebase which are activated the most in the transient period were shaded in, showing a so-called fuzzy phase plane plot. This was shown to be useful in further tuning and improving the response, since the designer can see which rules are most effective for the system.

The disturbance rejection characteristics of both types of fuzzy controller were compared. It was shown that the PI type fuzzy controller is capable of eliminating the disturbance

while the PD type has a steady state error. A combination of the two was shown to be effective, joining the characteristics of both types into a single controller.

5.2 Suggestions for Further Research

Future research is needed to determine the impact of using the quantized lookup table as opposed to the direct fuzzy method. In addition, the differences, if any, in using the linear interpolation lookup table should be examined. The quantized lookup table is often used in practice, and computation time is the reason for its popularity. It is unclear whether the savings in computation time outweigh the higher degree of accuracy in the direct fuzzy method. This is a typical engineering trade-off which deserves further investigation. The linear lookup table may not control nonlinear processes as well as the nonlinear fuzzy rulebase. Such a table uses an interpolation algorithm which may make it too slow for practical use. These are issues which need additional attention.

Another area that would provide an interesting avenue of research is the effect of membership function shape on system response. It is evident from the literature [27,28] that the fuzzy control system response is sensitive to changes in the slope and even position of the triangles. Just how these affect the system response is unclear. It is therefore suggested that a study of these relationships be made.

REFERENCES

REFERENCES

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

2. Lofti A. Zadeh, "Fuzzy Logic", IEEE Computer, pp. 83-93, April 1988.

3. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part 1", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 20, No. 2, pp. 404-418, March/April 1990.

4. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part 2", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 20, No. 2, pp. 419-435, March/April 1990.

5. King and E.H. Mamdani, " The Application of Fuzzy Control Systems to Industrial Processes", Automatica, Vol. 13, No. 3, pp. 235-242, 1977.

6. Ranier Palm, "Sliding Mode Fuzzy Control" IEEE International Conference on Fuzzy Systems, pp. 519-526, 1992

7. Tsu-Tian Lee and Kuo-Yang Tu, "Fuzzy Logic Controller Design Based on Variable Structure Control" Proceedings of the 1993 IEEE /RSJ International Conference on Intelligent Robots and Systems, pp.958-964, July 1993

8. Takeshi Yamakawa, " A Fuzzy Inference Engine in Nonlinear Analog Mode and its Application to a Fuzzy Logic Control" IEEE Transactions on Neural Networks, Vol.4, No. 3,, May 1993

9. G.V.S. Raju and Jun Zhou, " Fuzzy Rule Based Approach for Robot Motion Control" ,IEEE International Conference on Fuzzy Systems, pp. 1349-1356, 1992

10. Kandel and G. Langholz, eds. "Fuzzy Control Systems" CRC Press, 1994.

11. Mamdani and S. Assilian, " An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller ", International Journal of Man-Machine Studies, pp. 1-13, 1975.

12. Mamdani ," Advances in the Linguistic Synthesis of Fuzzy Controllers", International Journal of Man-Machine Studies, pp. 669-678, 1976.

13. Sugeno, ed. "Industrial Applications of Fuzzy Control", Elsevier Science Publishers BV, The Netherlands, 1985.

14. Atherton, "A Describing Function Approach for the Evaluation of Fuzzy Logic Control", Proceedings of the American Control Conference, pp. 765-766, 1993.

15. Kickert and E.H. Mamdani, "Analysis of a Fuzzy Logic Controller", Fuzzy Sets and Systems, pp. 29-44, 1978

16. Kumar S. Ray and D. Dutta Majumder, "Application of Circle Criteria for Stability Analysis of Linear SISO and MIMO Systems Associated with Fuzzy Logic Controller", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-14, No. 2, March/April,1984, pp. 345-349.

17. Ghassan M. Abdelnour, John Y. Cheung, Chir-Ho Chang, and Guillermo Tinetti, "Application of Describing Functions in the Transient Response Analysis of a Three-Term Fuzzy Controller", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, No. 2, March/April 1993, pp. 603-606.

18. Ghassan M. Abdelnour, Chir-Ho Chang, Feng-Hsin Huang, and John Y. Cheung, "Design of a Fuzzy Controller Using Input and Output Mapping Factors", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, No. 5, September/October 1991, pp. 952-960.

19. David F. Jenkins and Kevin M. Passino, " An Introduction to Nonlinear Analysis of Fuzzy Control Systems" Unpublished Ohio State University Manuscript.

20. Kwok L. Tang and Robert J. Mulholland, "Comparing Fuzzy Logic with Classical Controller Designs" IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-17, No. 6, pp. 1085-1087, November/December 1987

21. Steve Rogers " Infinite Band Controller with a Fuzzy Tuner" ISA Transactions Vol.31, No, 4, pp.19-24, 1992.

22. Li Zheng, "A Practical Guide to Tuning of Proportional and Integral (PI) like Fuzzy Controllers", Proceedings of the First IEEE Conference on Fuzzy Systems, March,1992, pp. 633-640.

23. Braae and D.A. Rutherford, "Theoretical and Linguistic Aspects of the Fuzzy Logic Controller", Automatica, Vol. 15, pp. 553-577, 1979.

24. Harris and C.G. Moore, " Phase Plane Analysis Tools for a Class of Fuzzy Control Systems", IEEE International Conference on Fuzzy Systems, pp. 511-518, 1992.

25. Li and C.C. Lau, Development of Fuzzy Algorithms for Servo Systems", IEEE Control Systems Magazine, pp. 65-71, April 1989

26. Kosko, "Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence", Prentice Hall, Englewood Cliffs, 1992.

27. Yongmei Wang, "Self-Tuning Fuzzy-PID Controller Design", M.S. Thesis, The University of Tennessee, Knoxville, August, 1993.

28. Gaiye Zhou, "Automatic Design of Autotuners for PID Controllers", PhD Dissertation, The University of Tennessee, Knoxville, August 1993.

29. Zhibin Duan, "Wind Turbine Control System Design and Simulation", M.S.Thesis, The University of Tennessee, Knoxville, August 1993.

30. Chung-Yuen Won, Sei-Chan Kim and B.K. Bose, " Robust Position Control of Induction Motor Using Fuzzy Logic Control" IEEE -IAS Annual Meeting Conference Record, 1992, pp. 472-481.

31. Dorf and Bishop, " Modern Control Systems " 7th edition, Addison Wesley, 1995.

32. Bahram Shahian and Michael Hassul, " Control System Design Using Matrix$_x$ " , Prentice-Hall, 1992.

33. Matlab " High-Performance Numeric Computation Software" John Little, Mathworks, Inc. 1991.

34. "Advanced Continuous Simulation Language" Reference Manual, Edition 10, Mitchell & Gauthier Associates, 1991.

35. " Simnon User's Guide ",Hilding Elmquist and Karl Johan Asrom , Department of Automatic Control, Lund Institute of Technology

36. C.L. Smith, "Digital Computer Process Control" Intext Educational Publishers, ISBN 0-7002-2401-7, 1972

37. A. Bastian, " A Simple Approach to Handle the Nonlinearity of a Fuzzy Controller Due to the Max-Min Method " Proceedings of the 1993 International Joint Conference on Neural Networks 1993, pp. 794-797.

APPENDIX

A programming block diagram suitable for simulation with ACSL is shown in Figure A-1. The various variables shown as inputs and outputs to the blocks correspond to actual program variable names. Figure A-2 follows, to facilitate the following discussion.

The simulation program is shown next. The part that represents the first order lag plus deadtime is straghtforward. There are, however, 49 rules and each rule has its own associated code. It is sufficient to describe a single rule since they have similar program elements. For this purpose Rule 1 is considered. The diagram related to Rule 1 is shown in Figure A-2, along with the code segment for Rule 1. The equations FA1=ME*EF +4 and FB1=-ME*EF-2 describe the waveforms of Figure A-2 (a). Referring to Table 2-2, Rule 1 is " If e is NL and $\dot{e}$ is NL then U is NL ." For error, FA1 represents the left side of the triangle for NL, and FB1 represents the right side of that triangle. For a given value of e a point on each line is specified. P1=MAX(FA1,Z) and Q1=MAX(FB1,Z) are max operations ensuring that only non-negative values of $\mu(e)$ are used. X1=MIN(P1,Q1) is the minimum of P1 and Q1, ensuring that if the value of e is outside the range of the triangle, the membership value is 0. The same reasoning is used for $\dot{e}$, with R1, S1, and Y1. Z1=MIN(X1,Y1) is the minimum of the two membership functions. $Z_i = f_i$ in Equation (2-4). The contribution of the rule to the numerator of Equation (2-4) is found by multiplying Z1 by the appropriate value from Table 2-3, which is implemented by dimensioning a 7x7 array, U(7,7). This forms one element of the summation in Equation (2-4). This process is repeated for each of the 49 rules. At the end of the program the calculation of ZT, UT and U completes the fuzzification/defuzzification process. ZT is the sum of all the minimums of the membership values, ZT= $\sum_i f_i$. It will be used in the denominator of Equation (2-4). UT is the sum of all the numerator terms, UT= $\sum_i f_i u_i$. The quotient U=UT/ZT is the crisp output of the fuzzy controller. It should be emphasized that only 4 of the rules contribute to the output at any one time, and more efficient programs are possible. This program was used here for ease of comprehension.

In the first part of the program, the controller output U is scaled and applied to the plant as EK. EK acts as the input to the delay block, whose output is X. X acts as the input to the first order lag block, simulated by the REALPL command. The output of that block is the system output, C.

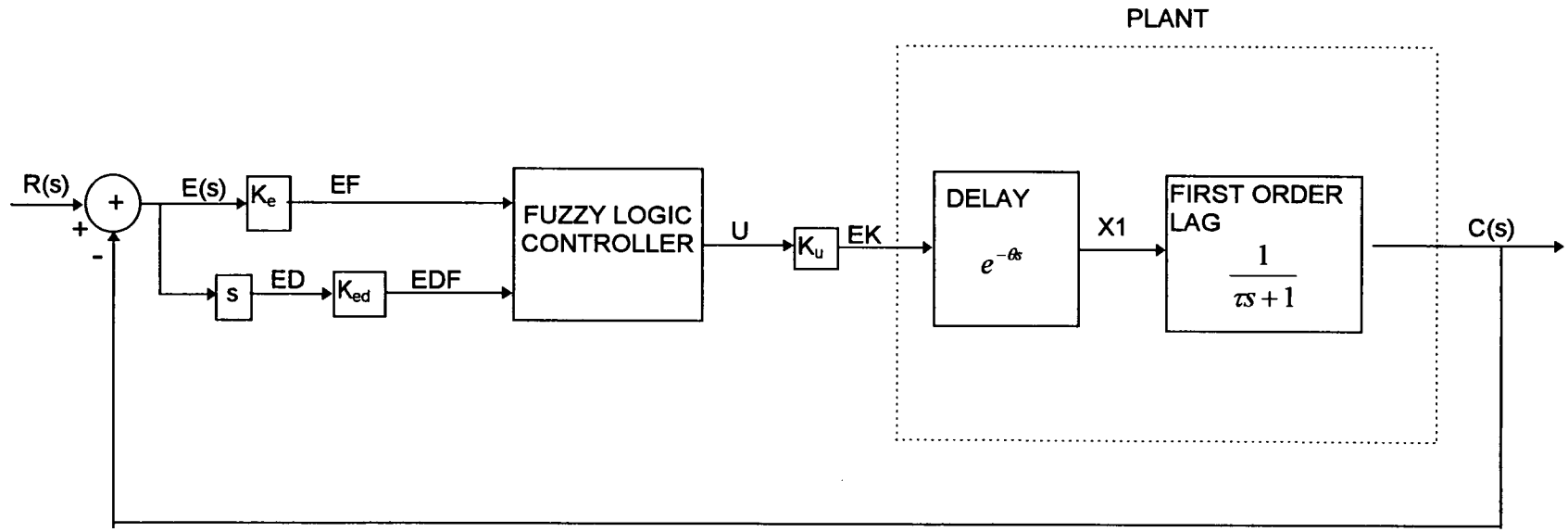
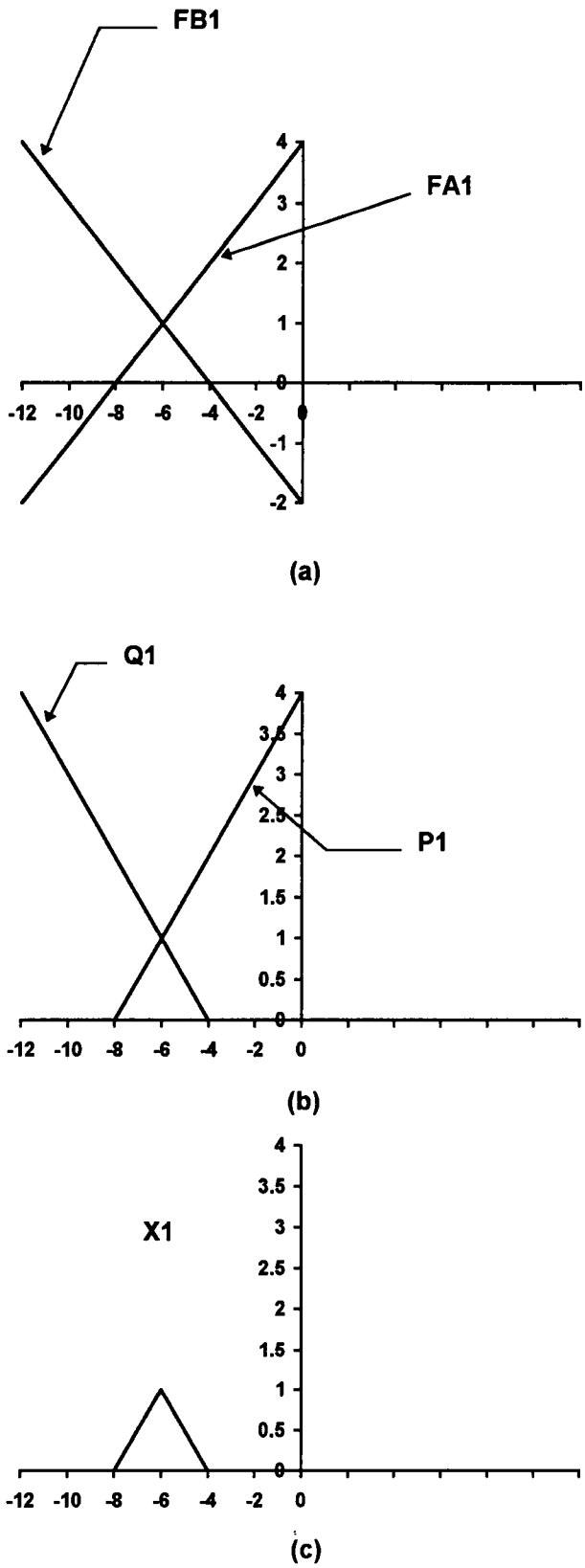


Figure A-1
Simulation Diagram of Fuzzy Controller Applied to Plant




```
!RULE 1
FA1=ME*EF+4
FB1=-ME*EF-2
FC1=MD*EDF+4
FD1=-MD*EDF-2
P1=MAX(FA1,Z)
Q1=MAX(FB1,Z)
R1=MAX(FC1,Z)
S1=MAX(FD1,Z)
X1=MIN(P1,Q1)
Y1=MIN(R1,S1)
Z1=MIN(X1,Y1)
U1=Z1*U(1,1)
```

Figure A-2 Code Implementation of Triangles

PROGRAM LISTING FOR FUZZY CONTROLLER

```
PROGRAM FUZZY CONTROLLER
INITIAL
CONSTANT P1=0.125, Z=1.0E-20,FIN=9.99,K=1.0, KU=1.0, ME=0.5, MD=0.5
CONSTANT KE=1.0, KED=1.0,KR=1.0,IC=0.0,TD=3.0,CMN=0.01
CINTERVAL CINT=0.02
END
DYNAMIC
DERIVATIVE
R=KR*1000*(RAMP(0.0)-RAMP(0.001))
E=(R-C)
EF=KE*E
ED=DERIVT(0.0,E)
EDF=KED*ED
EK=U*KU
X1=DELAY(EK,IC,TD,1000,CMN)
C=K*REALPL(1/P1,X1,0.0)
TERMT(T.GT.FIN)
END

DIMENSION U(7,7)
CONSTANT U=-6,-6,-6,-6,-4,-2,1E-20 &
,-6,-6,-6,-4,-2,1E-20,2 &
,-6,-6,-4,-2,1E-20,2,4 &
,-6,-4,-2,1E-20,2,4,6 &
,-4,-2,1E-20,2,4,6,6 &
,-2,1E-20,2,4,6,6,6 &
,1E-20,2,4,6,6,6,6

!RULE 1
FA1=ME*EF+4
FB1=-ME*EF-2
FC1=MD*EDF+4
FD1=-MD*EDF-2
P1=MAX(FA1,Z)
Q1=MAX(FB1,Z)
R1=MAX(FC1,Z)
S1=MAX(FD1,Z)
X1=MIN(P1,Q1)
Y1=MIN(R1,S1)
Z1=MIN(X1,Y1)
U1=Z1*U(1,1)


!RULE 2
FA2=FA1
FB2=FB1
FC2=MD*EDF+3
FD2=-MD*EDF-1
P2=MAX(FA2,Z)
Q2=MAX(FB2,Z)
R2=MAX(FC2,Z)
```

```
S2=MAX(FD2,Z)
X2=MIN(P2,Q2)
Y2=MIN(R2,S2)
Z2=MIN(X2,Y2)
U2=Z2*U(2,1)


!RULE 3
FA3=FA1
FB3=FB1
FC3=MD*EDF+2
FD3=-MD*EDF
P3=MAX(FA3,Z)
Q3=MAX(FB3,Z)
R3=MAX(FC3,Z)
S3=MAX(FD3,Z)
X3=MIN(P3,Q3)
Y3=MIN(R3,S3)
Z3=MIN(X3,Y3)
U3=Z3*U(3,1)


!RULE 4
FC4=MD*EDF+1
FD4=-MD*EDF+1
R4=MAX(FC4,Z)
S4=MAX(FD4,Z)
X4=X1
Y4=MIN(R4,S4)
Z4=MIN(X4,Y4)
U4=Z4*U(4,1)


!RULE 5
FC5=MD*EDF
FD5=-MD*EDF+2
R5=MAX(FC5,Z)
S5=MAX(FD5,Z)
X5=X1
Y5=MIN(R5,S5)
Z5=MIN(X5,Y5)
U5=Z5*U(5,1)


!RULE 6
FC6=MD*EDF-1
FD6=-MD*EDF+3
R6=MAX(FC6,Z)
S6=MAX(FD6,Z)
X6=X1
Y6=MIN(R6,S6)
Z6=MIN(X6,Y6)
U6=Z6*U(6,1)
```

```
!RULE 7
FC7=MD*EDF-2
FD7=-MD*EDF+4
R7=MAX(FC7,Z)
S7=MAX(FD7,Z)
X7=X1
Y7=MIN(R7,S7)
Z7=MIN(X7,Y7)
U7=Z7*U(7,1)


!RULE 8
FA8=ME*EF+3
FB8=-ME*EF-1
P8=MAX(FA8,Z)
Q8=MAX(FB8,Z)
X8=MIN(P8,Q8)
Y8=Y1
Z8=MIN(X8,Y8)
U8=Z8*U(1,2)


!RULE 9
X9=X8
Y9=Y2
Z9=MIN(X9,Y9)
U9=Z9*U(2,2)


!RULE10
X10=X8
Y10=Y3
Z10=MIN(X10,Y10)
U10=Z10*U(3,2)


!RULE 11
X11=X8
Y11=Y4
Z11=MIN(X11,Y11)
U11=Z11*U(4,2)


!RULE 12
X12=X8
Y12=Y5
Z12=MIN(X12,Y12)
U12=Z12*U(5,2)


!RULE 13
X13=X8
Y13=Y6
Z13=MIN(X13,Y13)
```

```
U13=Z13*U(6,2)

!RULE 14
X14=X8
Y14=Y7
Z14=MIN(X14,Y14)
U14=Z14*U(7,2)


!RULE 15
FA15=ME*EF+2
FB15=-ME*EF
P15=MAX(FA15,Z)
Q15=MAX(FB15,Z)
X15=MIN(P15,Q15)
Y15=Y1
Z15=MIN(X15,Y15)
U15=Z15*U(1,3)


!RULE 16
X16=X15
Y16=Y2
Z16=MIN(X16,Y16)
U16=Z16*U(2,3)


!RULE 17
X17=X15
Y17=Y3
Z17=MIN(X17,Y17)
U17=Z17*U(3,3)


!RULE 18
X18=X15
Y18=Y4
Z18=MIN(X18,Y18)
U18=Z18*U(4,3)


!RULE 19
X19=X15
Y19=Y5
Z19=MIN(X19,Y19)
U19=Z19*U(5,3)


!RULE 20
X20=X15
Y20=Y6
Z20=MIN(X20,Y20)
U20=Z20*U(6,3)
```

```
!RULE 21
X21=X15
Y21=Y7
Z21=MIN(X21,Y21)
U21=Z21*U(7,3)


!RULE 22
FA22=ME*EF+1
FB22=-ME*EF+1
P22=MAX(FA22,Z)
Q22=MAX(FB22,Z)
X22=MIN(P22,Q22)
Y22=Y1
Z22=MIN(X22,Y22)
U22=Z22*U(1,4)


!RULE 23
X23=X22
Y23=Y2
Z23=MIN(X23,Y23)
U23=Z23*U(2,4)


!RULE 24
X24=X22
Y24=Y3
Z24=MIN(X24,Y24)
U24=Z24*U(3,4)


!RULE 25
X25=X22
Y25=Y4
Z25=MIN(X25,Y25)
U25=Z25*U(4,4)


!RULE 26
X26=X22
Y26=Y5
Z26=MIN(X26,Y26)
U26=Z26*U(5,4)


!RULE 27
X27=X22
Y27=Y6
Z27=MIN(X27,Y27)
U27=Z27*U(6,4)


!RULE 28
X28=X22
```

```
Y28=Y7
Z28=MIN(X28,Y28)
U28=Z28*U(7,4)


!RULE 29
FA29=ME*EF
FB29=-ME*EF+2
P29=MAX(FA29,Z)
Q29=MAX(FB29,Z)
X29=MIN(P29,Q29)
Y29=Y1
Z29=MIN(X29,Y29)
U29=Z29*U(1,5)


!RULE 30
X30=X29
Y30=Y2
Z30=MIN(X30,Y30)
U30=Z30*U(2,5)


!RULE 31
X31=X29
Y31=Y3
Z31=MIN(X31,Y31)
U31=Z31*U(3,5)


!RULE 32
X32=X29
Y32=Y4
Z32=MIN(X32,Y32)
U32=Z32*U(4,5)


!RULE 33
X33=X29
Y33=Y5
Z33=MIN(X33,Y33)
U33=Z33*U(5,5)


!RULE 34
X34=X29
Y34=Y6
Z34=MIN(X34,Y34)
U34=Z34*U(6,5)


!RULE 35
X35=X29
Y35=Y7
Z35=MIN(X35,Y35)
```

```
U35=Z35*U(7,5)


!RULE 36
FA36=ME*EF-1
FB36=-ME*EF+3
P36=MAX(FA36,Z)
Q36=MAX(FB36,Z)
X36=MIN(P36,Q36)
Y36=Y1
Z36=MIN(X36,Y36)
U36=Z36*U(1,6)


!RULE 37
X37=X36
Y37=Y2
Z37=MIN(X37,Y37)
U37=Z37*U(2,6)


!RULE 38
X38=X36
Y38=Y3
Z38=MIN(X38,Y38)
U38=Z38*U(3,6)


!RULE 39
X39=X36
Y39=Y4
Z39=MIN(X39,Y39)
U39=Z39*U(4,6)


!RULE 40
X40=X36
Y40=Y5
Z40=MIN(X40,Y40)
U40=Z40*U(5,6)


!RULE 41
X41=X36
Y41=Y6
Z41=MIN(X41,Y41)
U41=Z41*U(6,6)


!RULE 42
X42=X36
Y42=Y7
Z42=MIN(X42,Y42)
U42=Z42*U(7,6)
```

```
!RULE 43
FA43=ME*EF-2
FB43=-ME*EF+4
P43=MAX(FA43,Z)
Q43=MAX(FB43,Z)
X43=MIN(P43,Q43)
Y43=Y1
Z43=MIN(X43,Y43)
U43=Z43*U(1,7)


!RULE 44
X44=X43
Y44=Y2
Z44=MIN(X44,Y44)
U44=Z44*U(2,7)


!RULE 45
X45=X43
Y45=Y3
Z45=MIN(X45,Y45)
U45=Z45*U(3,7)


!RULE 46
X46=X43
Y46=Y4
Z46=MIN(X46,Y46)
U46=Z46*U(4,7)


!RULE 47
X47=X43
Y47=Y5
Z47=MIN(X47,Y47)
U47=Z47*U(5,7)


!RULE 48
X48=X43
Y48=Y6
Z48=MIN(X48,Y48)
U48=Z48*U(6,7)


!RULE 49
X49=X43
Y49=Y7
Z49=MIN(X49,Y49)
U49=Z49*U(7,7)
```

```
ZT1=Z1+Z2+Z3+Z4+Z5+Z6+Z7+Z8+Z9+Z10+Z11+Z12+Z13+Z14+Z15+Z16+Z17+Z18+Z19+Z2
0
ZT2=Z21+Z22+Z23+Z24+Z25+Z26+Z27+Z28+Z29+Z30+Z31+Z32+Z33+Z34+Z35+Z36+Z37
ZT3=Z38+Z39+Z40+Z41+Z42+Z43+Z44+Z45+Z46+Z47+Z48+Z49
ZT=ZT1+ZT2+ZT3


UT1=U1+U2+U3+U4+U5+U6+U7+U8+U9+U10+U11+U12+U13+U14+U15+U16+U17+U18+U19
UT2=U21+U22+U23+U24+U25+U26+U27+U28+U29+U30+U31+U32+U33+U34+U35+U36+U37
UT3=U38+U39+U40+U41+U42+U43+U44+U45+U46+U47+U48+U49+U20
UT=UT1+UT2+UT3

EP=UT/ZT
END
END
```

VITA

Brian Fortenbery was born in Knoxville, Tennessee on January 16, 1958. He attended schools in Oak Ridge, Tennessee. He entered the University of Tennessee, Knoxville in 1984 and earned the Bachelor of Science degree in Electrical Engineering in December, 1987. While working in the nuclear power field, he entered the graduate school at the University of Tennessee in 1989. In 1990, he accepted a position at Cleveland State Community College, in the electricity/electronics department. While teaching there, he received the Master of Science degree in Electrical Engineering in December, 1995.