SYSTEMATIC ANALYSIS AND DESIGN OF FUZZY LOGIC CONTROLLERS FOR PROCESS CONTROL

GEORGE K.I. MANN

SYSTEMATIC ANALYSIS AND DESIGN OF FUZZY LOGIC CONTROLLERS FOR PROCESS CONTROL

by

© George K. I. Mann, B.Sc., M.Sc.

A Thesis Submitted to the School of Graduate Studies

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy in Engineering

Centre for Cold Ocean Resources Engineering and

Faculty of Engineering and Applied Sciences,

Memorial University of Newfoundland, Canada

September 1999

To My Parents.....

Abstract

Fuzzy Logic is relatively a recent development in the field of artificial intelligence. Since 1975. research and development in this field has had significant impact on industrial control including applications in many consumer products. A survey of the literature suggests that the majority of fuzzy control applications belong to the class of fuzzy PID-like or simply fuzzy PID controllers. Also the literature reveals that the existing design criteria require trial and error methods involving many computer simulations and need time to achieve satisfactory optimum control. This thesis presents a systematic study and analysis of fuzzy PID-type controllers with particular attention to process control. The work aims to remove the ad-hoc procedures and multi-dimensional complexity in the conventional fuzzy control designs and to present an analytical framework for the systematic design of fuzzy logic controllers.

The work investigates different fuzzy PID control structures including the conventional Mamdani-type controller. By expressing the fuzzy rules in different forms, each PID structure is distinctly identified. The rules are written in terms of the feed back error signals of a closed-loop control system. Therefore a general fuzzy PID controller output may be produced with three-, two- or one-input rule inference. A simple analytical procedure is developed to deduce the closed form expressions for generating outputs for general fuzzy PID controllers. The analysis starts with a linear-like fuzzy controller. Nonlinear fuzzy controllers are then systematically developed. The solution algorithm has the capability to generate the closed-form expressions to the general three-input fuzzy inference. The two- and one-input inferences are obtained as special cases of the general solution. The linear-like fuzzy output is used to identify the fuzzy PID actions in a dissociated form. The design of fuzzy controllers is then treated as a two-level tuning problem. The first level tunes the nonlinear PID gains and the second level tunes the linear PID gains. By assigning a minimum number of rules to each PID structure, the linear and nonlinear gains are explicitly presented. The tuning characteristics of each structure are evaluated with respect to their functional behaviours. The rule de-coupled and one-input rule structures proposed in this thesis provide greater flexibility and better functional properties than the conventional fuzzy controllers.

Non-linearity analysis is used to assess and rank the different fuzzy systems for fuzzy control. The normalized fuzzy output characteristics are identified for two-point control.

Thus, a performance criterion is developed to identify the non-linearity tuning properties of the fuzzy controllers. For each fuzzy PID type, a basis for non-linearity-tuning is developed. Using the new evaluation approach, different fuzzy systems are assessed. The min-max-gravity fuzzy reasoning has shown better nonlinear properties for fuzzy control applications. An alternative nonlinear control using spline-based functions is proposed. The geometrically based nonlinear controller has better nonlinear properties for PID control.

Linear PID controllers are analyzed in detail. The study is narrowed to process systems whose dynamics can be roughly approximated to first-order plus dead-time plant systems. The PID analysis covers the process systems having normalized time delay ranging from zero to any higher value. The time-domain-based analysis produces new PID tuning expressions for each case. The proposed tuning rules accommodate actuator saturation limits and avoid integral wind-up during the control. Numerical studies are made for higher order processes having monotonic open-loop characteristics. With the new tuning rules better performance is observed than with other commonly available tuning methods.

Fuzzy PID controllers are then evaluated for process control. A novel two-level tuning scheme is proposed for designing and tuning fuzzy controllers. For comparisons, three tuning methods are evaluated: (a) design based on a genetic algorithm (b) design based on a trial and error method of tuning and (c) design based on two-level tuning rules. The off-line design methods, such as genetic based tuning and trial and error based tuning, are unable to produce any improved control compared to linear PID controllers. The two-level tuning strategy uses the available linear control knowledge, and the resulting design always guarantees better performance than the linear controllers. The numerical simulations prove the new tuning method can be effectively used for any fuzzy PID controller type. Finally the two-level tuning is effectively implemented in a real time control problem. With the systematic two-level tuning, the fuzzy controllers are able to produce superior and improved performance to linear PID controllers. The design and tuning is simple and therefore the method can be extended to any process control problem.

Acknowledgement

This thesis work began in September 1995. Credit and my sincere thanks extend to many individuals and institutions for their immense support throughout this doctoral study.

Financial assistance has been provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), Canadian Space Agency. Petro-Canada Resources, Center for Cold Ocean Resources Engineering (C-CORE) and Faculty of Engineering and Applied Science in Memorial University of Newfoundland (MUN).

I have been fortunate and privileged indeed to work under Dr. Raymond G. Gosine, Associate Professor in Faculty of Engineering and Applied Science and Director of Intelligent Systems at C-CORE. Memorial University of Newfoundland. First, I would like to thank him for accepting my application for Ph.D. studies at MUN. I am grateful to him for providing excellent research environment, guidance, challenging academic discussions and friendly attitude during the course of the doctoral program. Also, I am grateful to supervisory committee members Dr. Michael Hinchey and Dr. Charles Randell for providing valuable comments and suggestions. I am thankful to Dr. James J. Sharp, former Associate Dean of Research, and Dr. Mahmoud Haddara. Associate Dean of Research and Ms. Moya Crocker, in the Faculty of Engineering and Applied Science at MUN, for their concerns about my graduate studies during the course of study. Appreciation is also due the C-CORE and all of its staff members for their kind and friendly support.

My special thanks extend to Dr. Bao-Gang Hu, former senior research engineer at C-CORE and Associate Professor in the National Laboratory of Pattern Recognition, Institute of Automation, P.R. China. I have been greatly inspired by him particularly for his original research and also for his constant encouragement, appreciation and valuable discussions during my past years of research in this chosen area.

I would like to thank University of Moratuwa, Sri Lanka for providing necessary leave to pursue doctoral studies. My sincere thanks to Professor P.A. de Silva, Dr. S. R. Tittagala, Dr. Lalith Gamage and Dr. Nalin Wickramarachchi at the University of Moratuwa for supporting my graduate application. Also the support provided by staff members at the Department of Mechanical Engineering in the University of Moratuwa are highly appreciated.

Finally, I am indebted to my wife Samanthika Shyamalee, for her patience and continuous support during my long hours of working in the university.

George K. I. Mann

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

ALG	:	Apparent Nonlinear Gain
ANG	:	Apparent Linear Gain
DA	:	Direct Action
ELC	:	Equivalent Linear Controller
FCS	:	Fuzzy Controller System
FLC	:	Fuzzy Logic Control
FLS	:	Fuzzy Logic Systems
GA	:	Genetic Algorithm
GPM	:	Gain and Phase Margins
GS	:	Gain Scheduling
IMC	:	Internal Model Control
KH	:	Khan-Lehman
ITSE	:	Integral Time Square Error
LLFLC	:	Linear-Like Fuzzy Logic Controller
MIMO	:	Multi-Input-Multi-Output
MMG	:	Min-Max-Gravity
NLFLC	:	Nonlinear-Like Fuzzy Logic Controller
PID	:	Proportional, Integral and Derivative
PSG	:	Product-Sum-Gravity
RZN	:	Refined Ziegler-Nichols
SISO	:	Single-Input-Single-Output
SOC	:	Self-Organizing Control
TP-PI	:	Two-Point PI
TSK	:	Takagi-Kang-Sugeno
ZA	:	Zhung-Atherton
ZN	:	Ziegler-Nichols

Symbols

$a_1,\ a_2,\ a_3$	:	distance between uniformly distributed input memberships
$a_{1,i}, a_{2,j}, a_{3,k}$	:	distance between non-uniformly distributed input memberships
B_r^n	:	Bernstein polynomial of degree n
$\boldsymbol{b}$	:	Bézier points
$d_1,\ d_2,\ d_3$	:	distance between uniformly distributed consequent memberships
d_m	:	distance between two consecutive output fuzzy singletons
$e(t), e(n)$	:	feedback error at time t and time instant n
$e,\ \Delta e,\ \Delta^2 e,\ \sum e$	:	error, error difference, change of error difference, sum-of-error
$\hat{\mathbf{e}}$	:	normalized input error vector
$\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3$	:	modal position vectors of antecedent membership functions
$\hat{e}_{1,ia},\ \hat{e}_{2,ja},\ \hat{e}_{3,ka}$	:	reference input error variables
$\hat{e},\ \Delta\hat{e},\ \Delta^2\hat{e},\ \sum\hat{e}$	:	normalized error variables
$E_i,\ \Delta E_j,\ \Delta^2 E_k$	:	linguistic variables corresponds to error inputs
$\mathbf{i}_a$	:	input index vector
i_a, j_a, k_a	:	nearest integers
i_n, j_n, k_n	:	transformed index vectors
k	:	process gain
K_1, K_2, K_2	:	normalized linear PID gains
K_b	:	normalized break-point gain
k_c	:	linear proportional controller gain
K_c	:	normalized linear proportional controller gain
$K_\mathrm{P},\ K_\mathrm{I},\ K_\mathrm{D}$	:	linear proportional, integral and derivative gains
$K_\mathrm{Pa},\ K_\mathrm{Ia},\ K_\mathrm{Da}$	:	aparent linear proportional, integral and derivative gains
$\hat{K}_\mathrm{Pa},\ \hat{K}_\mathrm{Ia},\ \hat{K}_\mathrm{Da}$	:	apparent nonlinear proportional, integral and derivative gains
K_u	:	normalized closed-loop ultimate gain
k_u	:	closed-loop ultimate gain
LAI	:	linearity approximation index
M	:	number of output fuzzy states
m_1, m_2, m_3	:	transformed incremental error terms
$N_1,\ N_2,\ N_3$	:	numbers of fuzzy states assigned for fuzzy input variables
NVI	:	non-linearity variation index
OS	:	peak overshoot
OS_d	:	desired peak overshoot
$r(t), r(n)$	:	reference input at time t and time instant n
s_1, s_2	:	non-linearity tuning parameters
$S_e,\ S_{ce},\ S_{rce},\ S_{se}$	:	normalizing scale factors for the error inputs
S_u	:	de-normalizing scale factor of the fuzzy controller output
T	:	time constant of first-order process model
T_d	:	derivative time constant

t_d	:	time delay of first-order process model
T_i	:	integral time constant
t_p	:	time corresponding to maximum controller signal
T_s	:	sampling period
T_r	:	rise time
T_{st}	:	settling time
t_u	:	ultimate time period
u	:	controller signal
$\hat{u}$	:	normalized fuzzy output signal
$\hat{u}_f$	:	projected normalized controller output
$\hat{u}_{\mathrm{L}}$	:	linear fuzzy output component
$\hat{u}_{\mathrm{NL}}$	:	nonlinear fuzzy output component
$\hat{u}_{Pn}$	:	normalized SISO fuzzy controller output
$\hat{\mathbf{u}}_{\mathbf{s}}$	:	vector defining fuzzy output singleton positions
U_m	:	linguistic variable of the FLC output
$\boldsymbol{X}$	:	Bézier polynomial
y	:	process response signal
y_a, y_m	:	two-points for TP-PI design
$y(t), y(n)$	:	feedback process signal at time t and time instant n
α	:	derivative weighting
α_1, α_2	:	nonlinear values for β_3
$\alpha_{n1}, \alpha_{n2}, \alpha_{n3}, \alpha_{n4}$	:	nonlinear values β_{n3}
$\delta\hat{\mathbf{x}}$	:	normalized incremental input vector
β	:	overshoot controlling parameter
$\beta_3, \beta_2, \beta_1$	:	nonlinear terms of three, two and one-input LLFLC outputs
$\beta_{n3}, \beta_{n2}, \beta_{n1}$	:	nonlinear terms of three, two and one-input NLFLC-II outputs
$\boldsymbol{\Delta}\hat{\mathbf{e}}$	:	differential input error vector
$\delta\hat{\mathbf{x}}_{\mathbf{a}}$	:	normalized incremental absolute input vector
$\delta x_{1,ia}, \delta x_{2,ja}, \delta x_{3,ka}$	:	incremental error input terms
γ	:	integral weighting
ϕ	:	nonlinear values for β_1
ϕ_{n1}, ϕ_{n2}	:	values for β_{n1}
ρ	:	proportional weighting
τ	:	scaled time
τ_d	:	scaled time constant
θ	:	nonlinear values for β_2
θ_0, θ_1	:	slope angels when $\hat{e} = 0$ and $\hat{e}_1 = \pm 1$
θ_{n1}, θ_{n2}	:	nonlinear values for β_{n2}
ζ	:	relative damping factor

Subscripts

e,ce,rce,se	:	error variables
i,j,k,m	:	fuzzy linguistic state
ia,ja,ka	:	reference modal positions
P,I,D,PI,PD,PID	:	different PID actions
w	:	error states

Superscripts

l	:	equivalent linear representation
P,I,D,PI,PD,PID	:	fuzzy PID controller element

Chapter 1

Introduction

1.1 General Background

Fuzzy Logic, which later became the Fuzzy Logic Systems (**FLS**) theory, is a mathematical concept that brings together the reasoning used by people with common sense and rules of thumb for the purpose of computer-controlled applications. The theory of vagueness (or fuzzy) extends from the early part of this century [1]. In 1965 Lotfi Zadeh of the University of California at Berkeley published the land mark paper "Fuzzy Sets" [2]. Although this work has been criticized for not properly citing and referring to the very early work of multi-valued logic by various logicians and scientists namely, Max Black, Charles Sanders Pierce, and Jan Lukasiewicz. Zadeh is presently considered to be the father of fuzzy logic [1]. The main reason for this recognition is that his work was able to cause a breakthrough from the conventional two-valued logic principle. More importantly his approximation theory contributed much to the development of knowledge based decision-making systems and made significant impact on many branches of engineering applications. The field amalgamates the cognitive information and set theory to encode linguistic fuzzy commands to represent the human decisions using numeric terms. The recent work of Zadeh on "computing with words" [3] attempts to fuse the fuzzy reasoning with probability theories for representing human decisions in a mathematical form. Control engineering is one of the major areas where fuzzy theory has been successfully applied. The approximate reasoning or compositional rule of inference [4] combined with fuzzy logic has become the rule of fuzzy inference [5] for decision making in Fuzzy

Logic Control (**FLC**). The fuzzy mathematics has been studied extensively since Zadeh's contributed work and the recent developments in universal approximation theory [1, 6, 7] are clear demonstrations of the progress. In 1974, Mamdani [8] pioneered the investigation of the feasibility of using compositional rule of inference that had been proposed by Zadeh [4], for controlling a dynamic plant. A year later, Mamdani and Assilian [9] developed the very first real-time implementation of FLC to control a laboratory steam engine plant. Inspired by this original work, FLC has been successfully implemented in many industrial applications including many commercial products [1, 10, 11]. Fuzzy control subway systems, combustion control, washing machines, cameras and camcorders, anti-skid breaking systems are few real time applications for which FLC has shown its potential in dealing with ill-defined and non-linear control problems. Also, the encoding of human experience into fuzzy paradigms avoids exhaustive search of exact system models and provides for an experience based problem solver [5, 12, 13].

In most FLC applications, the fuzzy controller replaces the existing linear Proportional (**P**), Integral (**I**) and Derivative (**D**) or **PID** controllers. This is not a surprise since linear PID controllers are used in the majority of industrial control loops [14]. The direct fuzzy representation of a PID controller is referred to as fuzzy PID-like [15] or simply fuzzy PID controllers. In a strict sense, the very first fuzzy controller applications [8, 9] were fuzzy PI controllers. This is the most common configuration adopted in many FLC applications. Therefore a significant number of in-depth theoretical and analytical investigations related to the Mamdani-type fuzzy PI structure were reported [16]–[31]. These researchers have contributed much to realize fuzzy control action in a more exact form and were able to provide greater transparency and also scope for analyzing fuzzy controller's controllability, stability and tuning for real-time applications.

The linear PID controller has a degree of adaptive capabilities to cope with parameter changes and model uncertainties [32]. Overall, the conventional PID controller has more design and tuning techniques available than intelligent controllers. Also the controller can be easily implemented (either in digital or analog form) compared to the complex fuzzy controllers or nonlinear control schemes [33]. Therefore the linear PID controller is still by far the most popular controller element today, even for nonlinear plant models [34]. Despite the complexity in designs, fuzzy controllers have generally shown improved performance com-

pared to linear PID controllers [5, 35]. However, in certain cases it has been reported that fuzzy controllers were unable to perform better than the linear PID controller [29]. Also, certain fuzzy controller designs have been highly criticized in the face of linear PID designs [36]. There are mainly two reasons for this. First, if the knowledge base system is not carefully selected for the given plant conditions, then any arbitrarily selected fuzzy system may produce poorer performance. Even in the presence of well-defined expert rules, the improper encoding or formulation of fuzzy variables can result in a bad controller. Second, when the linear PID controller is not properly defined for the comparison, then the claim of superiority of the fuzzy controller will become meaningless.

The fuzzy controller design is still a somewhat "*fuzzy*" process. This is due to the existence of different fuzzy mapping systems and also due to its multi-dimensional nature. Compared to the three-parameter linear PID controller design, the fuzzy PID variables can range from a minimum of three to any higher number, depending on the complexity of the knowledge base representation. Kosko [1] referred to this as "*the curse of dimensionality*". Rule exploration and reduction of rule base size [37] or compression of rules [38] have recently been given much attention to ease the curse of dimensionality. Although some still believe that there is no systematic design method for fuzzy controllers [15], several methods have been proposed during the recent years for systematic fuzzy controller designs [1].[39]–[43].

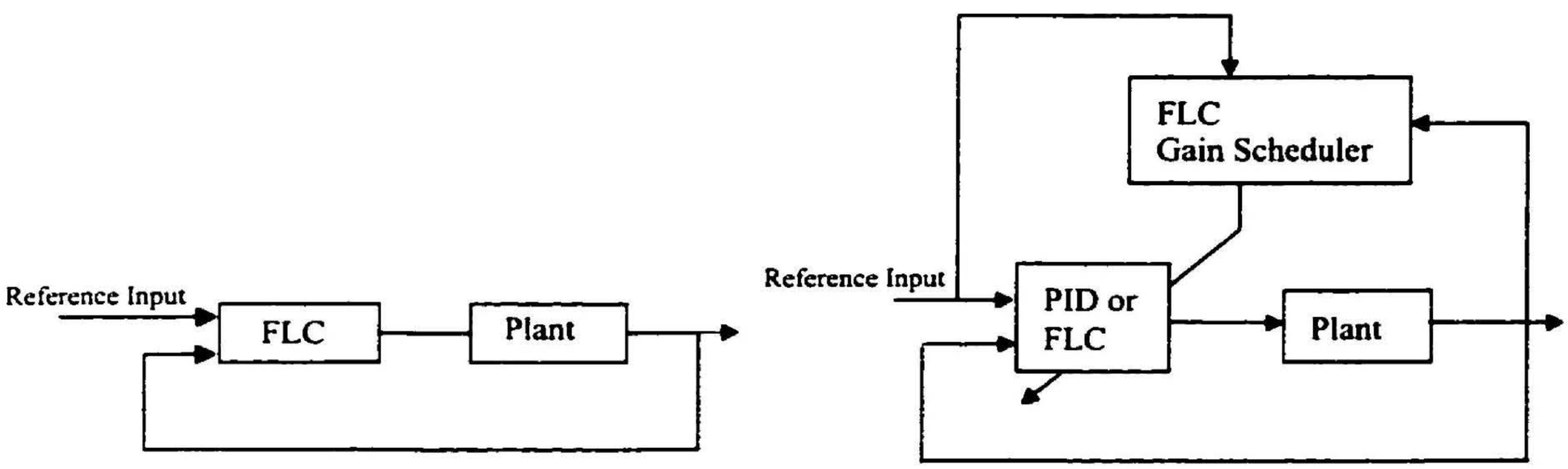
The fuzzy knowledge base system provides nonlinear transfer elements for nonlinear control [15]. The problem of designing a fuzzy system is considered as an approximation problem [44]. Generally the non-linearity that is required by a given process control system is unknown in the design process. The majority of FLC design techniques use off-line computer simulations (numerical optimizations [39]–[41] or training algorithms [1, 42]) to find the unknown nonlinear controller. While these methods have shown promising results for specific controller applications, there is a difficulty in generalizing such fuzzy controllers for a wider range of process specifications. Also, the solution is not guaranteed to offer a better performance controller upon implementation in real-time control. In contrast, the linear PID design techniques are quite satisfactory and well suited for wide range of process specifications. Fuzzy control is relatively new and unexplored. The challenge is to find out what it is good for, and more importantly, what it is not good for [45]. Therefore the main objective of this thesis is to perform a systematic analysis and design, particularly for devising simple

and practically possible fuzzy controllers for process control.

1.2 Overview of Fuzzy PID Control and Problem Identification

There are several types of fuzzy control systems that use FLC as an essential system component. As mentioned in Section 1.1, the majority of applications during the past two decades belong to fuzzy PID (or PID-like) controllers. Detailed lists of past applications are described in Chapter 2. These fuzzy controllers again can be classified into three types: Direct Action (**DA**) type, Gain Scheduling (**GS**) type and combination of DA and GS type. Figure 1.1 shows the arrangement of different fuzzy PID controllers. Again the majority of fuzzy PID types belong to the DA class of controllers: here the fuzzy PID controller is placed within the feedback control loop to compute the fuzzy PID actions through fuzzy reasoning. In GS type controllers, the fuzzy inference is used to compute the individual PID gains and the inference is either an error-driven self-tuning process [46] or a performance-based supervisory tuning [5]. The fuzzy reasoning in either DA or GS types attempts to provide a nonlinear hyper plane (or surface or curve) for the controller output. Therefore, in a strict sense, the fuzzy PID controller is a nonlinear PID controller with varying equivalent PID gains with respect to the error state variables, as opposed to linear PID control policy in the conventional schemes. However, any designer will eventually be faced with determining a considerable number of parameters before implementing any type of a fuzzy PID controller. The unknown design parameters that are related to fuzzy control can be first divided into two groups as shown in Table 1.1.

The design task has two main components. The first problem is to determine the knowledge base parameters or a fuzzy system to generate the necessary crisp control action. The second task is to tune the controller to obtain the desired performance of the process response. Fundamentally, the first task is derived from the knowledge acquisition process [1, 5, 42] or automatically synthesized from self-organizing control architecture [47]. In the conventional two-input type fuzzy controller (or Mamdani-type controller), the error and change of error in a given sampling instant are mapped to a two-dimensional crisp output surface in the error state space. When the tuning process (second task) is ignored then this output rep-



(a) Direct Action (DA) type

(b) Gain Scheduler (GS) type

Figure 1.1: Fuzzy PID controller systems in closed-loop control

Table 1.1: Fuzzy control design variables

Knowledge base variables	Tuning parameters
Input variables to the FLC	Normalizing scale factors
FLC configuration	De-normalizing scale factors
Output variables of the FLC	Linear PID gain parameters
Fuzzy variables and Membership functions	Support sets and their positions
"If-Then" rules	
Fuzzy Reasoning schemes	

resents the numerically coded, multi-level human experience or knowledge acquisition in the process of control. Such experience is usually not available for inner loop control problems. As an example, for high-speed control of robot manipulators the human experience is more in supervisory level control than in direct loop control [5]. Therefore in most direct fuzzy control applications, the task of defining a suitable knowledge base is a trial and error process or simply a preference that one makes from available fuzzy control systems. On the other hand in a supervisory tuning (GS type) the manual tuning heuristics that are available with a control expert are not precise enough for achieving a stable fuzzy controller. As a result such self tuning systems face the problem of *tuning-in-tuning*. By considering the existing design complexity and some misconceptions that exist in the current research, three major problem areas have been identified. In a systematic analysis process, these problems are interdependent. However, they are distinctly represented for clarity.

1.2.1 Problem I: The FLC structures and Input/Output Variables

The first problem type is related to the overall fuzzy controller configuration. Due to the wide range of applications of the most common Mamdani-type (fuzzy PI or PD type) rule based structure, this controller is now considered to be the conventional fuzzy PID controller [48]. However, in the past some difficulties associated with the tuning of this controller have been observed [48]. Therefore it can be argued that other types of fuzzy PID configurations are possible in the context of different knowledge based representations. The lack of understanding of the fuzzy PID configurations sometimes leads to misinterpretations as well. For example, the fuzzy PD controller in [24] has been used to control a non-self-integral type process. From control fundamentals it can be argued that such processes require some form of an integral action or an estimated finite controller signal to produce the steady state performance. Therefore the issue is how a PD type controller was able to produce the steady state performance with no integral or estimated control component. The second example the author wishes to cite is the fuzzy PI controller with anti-windup capabilities [49]. This controller has been used to control a self-integrating type process. Such processes can be simply controlled by a PD type controller with no integral action. The requirement of an integral control for a position control type problem that is shown in [49] is unnecessary since the steady off-set is always zero for such processes. If a PI type controller is employed, then

windup would definitely cause oscillation in the response. However the anti-windup PI controller in [49] has shown good response (with no overshoot) which is theoretically impossible with a PI type controller. Therefore the following issues are identified with respect to the problems involving different fuzzy PID structures.

1. How different input and output feedback error variables are determined to produce the individual or composed fuzzy PID actions.

2. How these input/output variables are correlated through fuzzy inferences to generate different fuzzy PID configurations.

3. How these PID configurations are differentiated with respect to their functional properties.

1.2.2 Problem II: Evaluation of Fuzzy Logic Systems

By observing the variables associated with fuzzy controllers in Table 1.1, there are enormous numbers of possibilities that lead to different fuzzy mapping systems. Mendel [50] states that this feature of having many possibilities does not require detailed understanding of a fuzzy system and he argues that each is analogous to the representation problem that we always face in engineering. This statement is partially true for two reasons. First, the choice of an FLS is not arbitrary. Some degree of optimization and suitability is necessary to enhance quality. As an example, a choice of PID-type versus sliding-mode type fuzzy controllers or DA type versus GS type PID controller, belongs to a representation problem. Thus, Mendel's statement is valid. Given a representation, there are many fuzzy systems that can provide satisfactory performance to a selected problem. Many fuzzy systems have redundant variables (not absolutely, but for practical situations) and as a result different fuzzy systems have similarities. Therefore even if two fuzzy systems are different in the representation, functionally both may be the same. However with respect to the required functional properties, one mapping system may be superior for control or implementation to the other. In that sense Mendel's argument is not valid. As an example, for producing linear output functions, one FLS may require a larger number of if-then rules [22] whereas a different FLS may only need a small number of rules [26, 39]. The argument can be extended to any nonlinear function in general.

In addition to the Zadeh-Mamdani fuzzy reasoning, many other types of fuzzy inference systems and de-fuzzification strategies [15, 51] have been used and developed in the past. Some attempts have been made at evaluating different fuzzy reasoning systems [20, 31, 52]. The present reasoning schemes or FLS theories can convert the words or linguistic labels to mathematics or numeric form but be unable to perform the reverse operation. This means the theory is insufficient to convert the numeric data (or inferred fuzzy set) to a linguistic label. If a fuzzy reasoning is able to produce similar linguistic commands as a human expert, then such a fuzzy mathematics could become absolutely standard as in crisp mathematics. FLS theories have largely bypassed this issue by the de-fuzzification process. As a result the present FLS theories accommodate any form of inference method as long as they serve the purpose. Although certain fuzzy representations have gone far away from the natural logic of vague theories, those systems have also been accepted as valid fuzzy reasoning schemes. The Takagi-Kang-Sugeno or **TSK** type fuzzy representation [39] is a clear example of that kind. However, if a fuzzy controller is considered as a nonlinear controller, then different fuzzy systems can be compared and evaluated in terms of non-linearity for more efficient and improved utilization. Therefore in this thesis the author wishes to explore other non-fuzzy types, that are functionally equivalent to fuzzy control systems. The particular issues related to this problem area are:

1. Identification of different fuzzy reasoning methods and their applicability to fuzzy control.

2. Determination and formulation of performance measures to identify the non-linearity characteristics and their effects on overall performance of FLC systems.

3. Evaluation of different fuzzy representations and reasoning schemes for better and efficient control.

1.2.3 Problem III: Fuzzy Controller Tuning

Most fuzzy products and systems that have been developed as commercial products are based on expert rules [1, 5, 11]. The numbers of rules in most applications are quite small for realizing an approximate fuzzy control system. However, fuzzy controller tuning is the hardest task. Given a FLS, the tuning procedure can extend to several hours or weeks of

software trial and error simulations [1]. As mentioned earlier, even the self-tuning controllers suffer the tuning-in-tuning. Also, most of the available tuning methods fall into the category of *systemized random-search-oriented systems* or *data-driven learning* systems [1].

In general the tuning of any fuzzy control system involves two levels of tuning [48]. The first level determines the non-linearity characteristics of the controller. This is achieved by the knowledge base parameters of the FLS. The second level determines the overall characteristics. The scale factors including normalized/de-normalized gains and other linear gains provide the overall magnification of the nonlinear controller output and hence the second level of tuning achieves the overall performance. In some fuzzy applications, the first level was determined a *priori* and then the second level parameters were either adjusted or tuned to achieve the desired response characteristics. The author believes this kind of a study loses the real meaning and usefulness of fuzzy control unless an *a priori* fuzzy system has been functionally proven elsewhere. This is the key reason why the majority of applications follow the conventional type fuzzy PID system. There are two reasons why two-level control identification is important for systematic designs. First, the high level tuning enables to reduce the multi-dimensional complexity. Second, the effect of knowledge-based control is isolated from the conventional (or linear) control characteristics and thereby the limitations of fuzzy systems are well recognized. The present computer-aided tuning methods have the disadvantage of separating these two levels of design. Those methods provide powerful tools for handling larger number of variables for the optimization. As a result the designers have over trusted those techniques and have set aside the traditional and well-developed linear (or nonlinear) control theories that have been developed through years of research. At the other extreme intelligent control systems are either criticized or ignored by the traditional control experts [36]. They often believe these systems are just unnecessary complications and that such systems are impractical for real-time control. Some researchers have attempted to fill this gap by bridging the conventional and fuzzy theories for tuning fuzzy controllers [43]. The importance of using conventional design techniques and the dangers of using ad hoc procedures for the design of fuzzy control is well explained in [45]. Therefore the issues related to this problem area are:

1. The identification of tuning levels where the knowledge based system is more transpar-

ent to control performance.

2. Tuning methods to cover a wider spectrum of process control problems.

3. Accommodation of linear control theories for fuzzy control design.

4. Computationally economic and less complex tuning procedures which control engineers can easily implement for real-time control problems.

5. Capabilities of the tuning process for implementation in adaptive and self-organizing control environments.

1.3 Research Objectives

The main objective of this research is to establish a systematic design procedure and an analytical framework for design of fuzzy logic controllers. The research attempts to address three major problem areas that have been identified in the above section. Since the PID controllers are more common particularly for process control problems. the research is focused on the development of fuzzy PID controllers.

1.3.1 Controlling Process

This study is focused on the typical feedback control of single-input single-output (**SISO**) processes. In the final designs. the work is further narrowed to the class of problems that have monotonic open-loop response except for the initial time period (or lag time) where such processes can be crudely approximated by first-order plus dead time process models. Such dynamics are common in many industrial process systems. However, the proposed FLC design in this thesis can be easily extended for other classes of process models.

1.3.2 Systematic Design Procedure

In order to achieve the main research objective of this thesis and to address the main issues related to the three specific problem areas. a systematic flow of analysis has been identified as shown in the Figure 1.2. The figure indicates that the study has two parts. The first

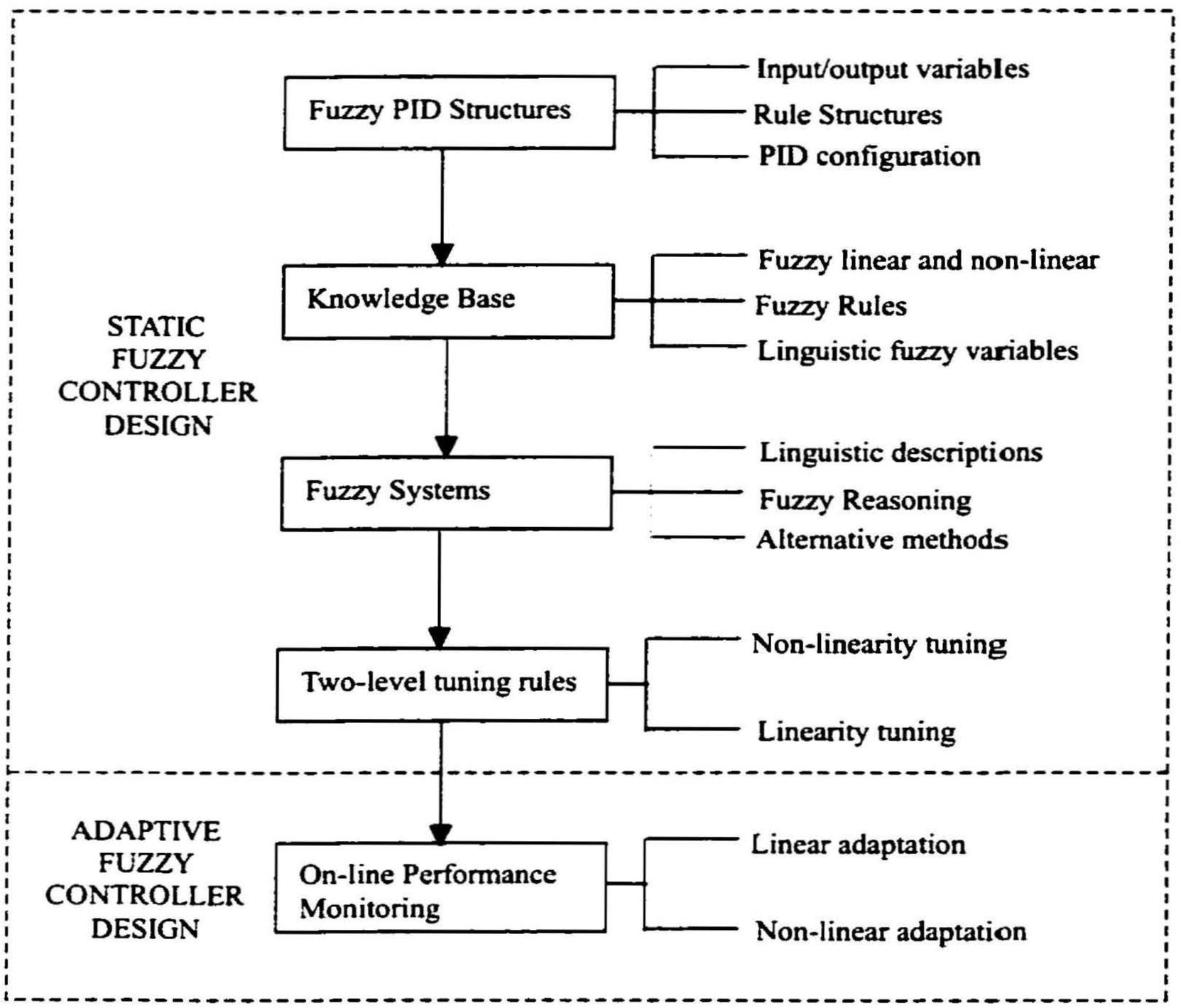


Figure 1.2: Flow of the systematic analysis and design procedure

part contributes to static fuzzy PID development and the second is for analyzing the self-organizing behaviour of fuzzy controllers. The steps are inter-related. Much of this thesis work is concerned with static FLC designs. The term "static" means the fuzzy system would not adapt or self-adjust to dynamic changes in the process or environment. On the other hand, the fuzzy system has nonlinear control characteristics that would eventually produce self-robustness in the controller [43]. This study will provide the basic features necessary for devising a self-organizing fuzzy controller with auto adaptive capabilities. Therefore the objectives of this research are to:

1. Propose a systematic procedure for establishing fuzzy PID control architectures using different knowledge based representations.

2. Investigate the functional behaviours of these controllers by establishing closed-form solutions to different fuzzy PID controllers.

3. Systematically reduce the dimensionality in the fuzzy systems for tuning.

4. Develop a new evaluating scheme for identifying and comparing different fuzzy systems.

5. Establish a simple and easy tuning scheme for fuzzy controllers in such a way that the performance is always guaranteed to be better than the linear PID controller.

6. Incorporate the existing PID tuning rules and theories for fuzzy controllers so that the proposed fuzzy controller designs can be readily used for other process specifications as well.

7. Identify the necessary functional properties to establish a systematic design procedure for self-organizing fuzzy controllers.

1.4 Organization of the Thesis

Chapter 1 addresses the importance of fuzzy system theory in fuzzy control. The main problem areas in the fuzzy controller design are explained. This chapter ends by providing a list of original contributions.

Chapter 2 provides a detailed literature review of fuzzy controllers including static and adaptive fuzzy controllers. The literature review explains how the past researchers have

attempted to solve the problems identified in the first chapter. The existing design techniques are briefly explained to identify the limitations in those approaches.

Chapter 3 performs an analysis of direct action fuzzy PID controllers. The general classes of fuzzy PID controllers are shown. In addition to the conventional fuzzy controller structure, other possible classes of fuzzy PID controllers are identified. For evaluating the functional properties of different controller structures, the closed-form solutions of different fuzzy system outputs are derived. The functional properties are identified for comparing different FLS configurations.

In Chapter 4 a nonlinear analysis is performed for ranking and identifying different fuzzy systems for control and tuning. Performance measures are defined for evaluating the non-linearity properties in the fuzzy systems. Also at the end, an alternative nonlinear control strategy, using B-spline curves is explained.

In Chapter 5 a time-domain analysis is made for linear PID controllers and a new tuning scheme for linear PID controllers is described. At the end numerical simulations are performed to confirm the performance.

In Chapter 6, a two-level tuning scheme for fuzzy PID controllers is developed. For comparison of the new scheme, two other tuning methods are examined. The new fuzzy tuning method uses the linear PID tuning rules developed in Chapter 5. Important tuning heuristics related to fuzzy control are established. At the end, the method is implemented in real time to control the temperature in a soil-cell. Due to the time limitations of this thesis work, further investigations on self-organizing ability of the fuzzy controllers have not been completely performed. However, important features of the two-level tuning described in this chapter can be effectively incorporated to establish an adaptive fuzzy controller with self-organizing ability. Therefore at the end of chapter 6, a systematic design procedure for self organizing fuzzy controllers is proposed.

Chapter 7 summarizes the research work and provides suggestions for future work within this area of research.

1.5 Original Contributions

- New fuzzy PID configurations have been identified. The proposed PID structures have

better functional properties than the conventional fuzzy PID controllers.

- A new analytical procedure is presented for the general three-input fuzzy inference based on min-max gravity reasoning. The two and one-input fuzzy outputs are obtained as special cases of the general three-input solution. The analysis is systematically extended to obtain solution algorithms for non-linear like fuzzy controllers. More significantly, the new solution algorithm avoids using excessive number of nonlinear expressions (e.g. 48 expressions for general three-input based solution). The simplicity is achieved by introducing a new transformation technique to the solution algorithm.

- Functional properties are identified to compare different fuzzy PID configurations. Fuzzy controller tuning levels are identified by defining apparent linear and apparent nonlinear PID gains. For each fuzzy configuration the tuning parameters are explicitly presented.

- A new evaluation scheme is presented to compare the applicability and efficiency of different fuzzy reasoning schemes for fuzzy PID control. A new graphical representation is developed to identify the non-linearity characteristics and thus the performance of fuzzy controllers.

- An alternative nonlinear PID controller using spline-based functions is developed.

- New tuning rules for linear PID controllers are proposed. The proposed tuning rules are applicable for the normalized dead time ranging from zero to any higher value. In addition the new tuning rules have the flexibility to design PID controllers to avoid integral wind-up. which is associated with the actuator saturation.

- A new two-level tuning scheme for fuzzy PID controllers is proposed. The proposed method uses the existing linear PID control theories and guarantees better overall performance than the linear PID controller. The tuning method is general and can be effectively used for any fuzzy PID type controller.

- A new self-organizing fuzzy PID controller is proposed.

Chapter 2

Review of Literature

2.1 Introduction

Over the past two decades there have been an increasing number of fuzzy controller applications in many industrial applications and a considerable amount of research has been performed for identifying the functionality of fuzzy controllers for better utilization and ease of design. The very first fuzzy controllers developed by Mamdani and his co-workers in [8, 9] were equivalent to fuzzy PI-like controllers. In [9] the pressure error and speed error signals were used to generate the incremental fuzzy PI controller signals to control the throttle opening change and heat input change for operating a laboratory steam engine plant. This application has demonstrated the robustness of fuzzy controllers, particularly for nonlinear process control. A similar fuzzy PI controller was later reported [53] for temperature control of a stirred tank. During the same period Kickert and Lemeke [54] implemented both PI type and P type fuzzy controllers for control of flow rates in a warm water plant. Mamdani's pioneering work also introduced the first and now the most common fuzzy reasoning method, called Zadeh-Mamdani *min-max-gravity* reasoning for fuzzy control. This chapter intends to evaluate the past developments since the Mamdani's work in the area of fuzzy control.

2.2 Theoretical Evaluations

One of the main advantages of fuzzy logic systems (FLS) is that it has better transparency amongst the available soft computing techniques. This means the input-output nonlinear

mapping of fuzzy systems may be expressed either as closed-form expressions or as graphical forms. As a result, considerable in-depth analytical investigations were undertaken to understand the functionality and thus enhance the design efficiency of fuzzy controllers.

2.2.1 Rule Base development

Braae and Rutherford [16] pioneered the theoretical investigation of fuzzy logic control (FLC) systems. The characteristics of the rule base on plant performance was systematically analyzed to show the selection and adjustment of fuzzy parameters. More importantly, linguistic stability and steady state analysis were done for a systematic development of a fuzzy rule base. Also the linguistic stability was analyzed by defining rules for obtaining linguistic explicit and implicit steady state convergence in a linguistic phase plane. Although this analysis is more qualitative compared to present status of fuzzy controller theories, the rules described in the paper are helpful to design fuzzy rule bases within safe stability margins. In 1990, Lee published a two-part significant paper [55] in FLC. The theories of FLC and design descriptions were well presented for many researchers to follow. Lee [55] identified the rule base by referring it to a closed-loop trajectory in a phase plane. Later Li and Gatland [56] used a similar state space trajectory approach to generate rule bases for fuzzy PID control. Linear control rules have been analytically investigated in [18]. Later linear and product rules have been investigated in a more general form [57]. Generation of rules by sub-division of the state space is shown in [15]. Some researchers [1, 15, 30, 55] suggest that having more rules near zero error region, assuming that fine partitioning near zero error, would improve the control resolution and therefore expect accurate control. However, more rules would reduce the *fuzziness* in the rule base [5] and make it more deterministic and rigid. If the control rules are designed with a monotonic style (usually a required property for fuzzy control), then fine partitioning usually creates a linear control policy near zero. This means the real benefit of generating the unknown nonlinear mapping in fuzzy systems would be lost while constraining the system to follow a linear control policy.

2.2.2 Inference Analysis

In early FLC applications, the controller outputs were computed off-line using fuzzy inferences and the final control actions were prepared in the form of a look-up table for implementations. The membership values of the fuzzy sets (or term sets) and the associated elements in the universe of discourse were defined in a discretized form. Therefore the accuracy of this method depends heavily on the level of discretization of controller variables. Also, any alteration of the fuzzy system would require repeated tedious calculations of new look-up tables. Therefore, closed-form analysis of fuzzy inference would result in both faster on-line implementation and also convenient accommodation of any parameter variations in adaptive control schemes. In addition, it produces a new environment for classical fuzzy controller analysis.

Ying and his co-researchers [18]–[23] have been able to deduce closed-form expressions for the output of a fuzzy controller. Their initial work had two parts. In the first part [18], they have analyzed the fuzzy controller under various fuzzy logic principles and deduced a perfect linear controller using the mixed logic. In the second part [19], the nonlinear FLC action was explicitly presented and closed-form expressions were deduced for equivalent nonlinear P and I gain terms. A linear de-fuzzification process has been proposed in the latter for emulating linear PI actions. Similar work has been later reported by Ying [20]–[22] regarding inference analysis and this has shown that a fuzzy controller action represented in a linear-like form is a global two-dimensional multilevel relay and a local nonlinear PD controller. Although Ying and other co-researchers contributed works have some constraints for a general fuzzy system, the analyzes have provided the scope for exploring the functionality of fuzzy controllers. Similar approaches are found in [24, 25, 29] for obtaining the closed-form solutions to the fuzzy controller output. A more general nature of analysis is shown in [57]: this uses both linear and additive (or product) rules for the analysis in contrast to linear rules in Ying's work. The analyzes shown in the above references were based on Zadeh-Mamdani min-max-gravity (**MMG**) fuzzy reasoning, except in [20] where different reasoning schemes have been analyzed for comparison. In the MMG reasoning the conjunction AND operator and the disjunction OR operator are interpreted with min and max operators respectively and the final defuzzification is performed by taking the center-of-area of the inferred fuzzy set. Due to the discontinuity of the min-max functions, the closed-form expressions of the FLC

output based on the MMG reasoning have a multi-phase solution structure. To avoid this discontinuity the AND and OR operators can be represented by product and sum functions respectively. Therefore product-sum-gravity (**PSG**) reasoning has become popular during recent years, for expressing the FLC action in a more compact form [26]. Such an inference would allow the knowledge base system analysis to be less complex, even with nonlinear knowledge base [58], than the linear representation of Ying's work. In 1985, Takagi and Sugeno [39] introduced a new representation to fuzzy rules. The method is popularly known as Takagi-Kang-Sugeno or TSK method and it has been widely employed in many fuzzy products, particularly in Japan. In the TSK representation the outputs of rules are expressed in a functional form. Therefore TSK method is easy to analyze compared to traditional fuzzy rules. Although this representation has greatly reduced the linguistic and natural representation of the conventional rule base systems, the method is numerically efficient for process modelling and identification when it is used with optimization methods. In [39], the TSK fuzzy system has been used to model and optimize industrial process control systems. Sugeno further extended the TSK based research for many engineering applications [59, 60, 12]. When both TSK and Zadeh's systems employ fuzzy singletons for the output fuzzy variables, both systems become identical. Using least squares error criteria, Fileve and Yager [61] derived an equivalent Zadeh's rule based PD controller from the TSK solution. The fuzzy reasoning based solutions with standard fuzzy inferences are usually difficult to express in algebraic form. In order to improve the computer aided designs for fuzzy systems, the FLS inference are now popularly expressed in algebraic forms [1, 104].

2.3 Classification of Fuzzy Control Systems

This literature search reveals that there are several types of control systems that use FLC as an essential component. In [62] the FLCs were classified into seven different types. The classification was based on the position and the level on which the FLC is placed in the overall control loop. In fact there are as many types of fuzzy controllers as conventional controller types. In most controller designs, linguistically based fuzzy controllers have replaced the conventional controller. When the classification is broadened, the fuzzy controllers can be grouped into few major categories as shown in Figure 2.1. This classification categorizes

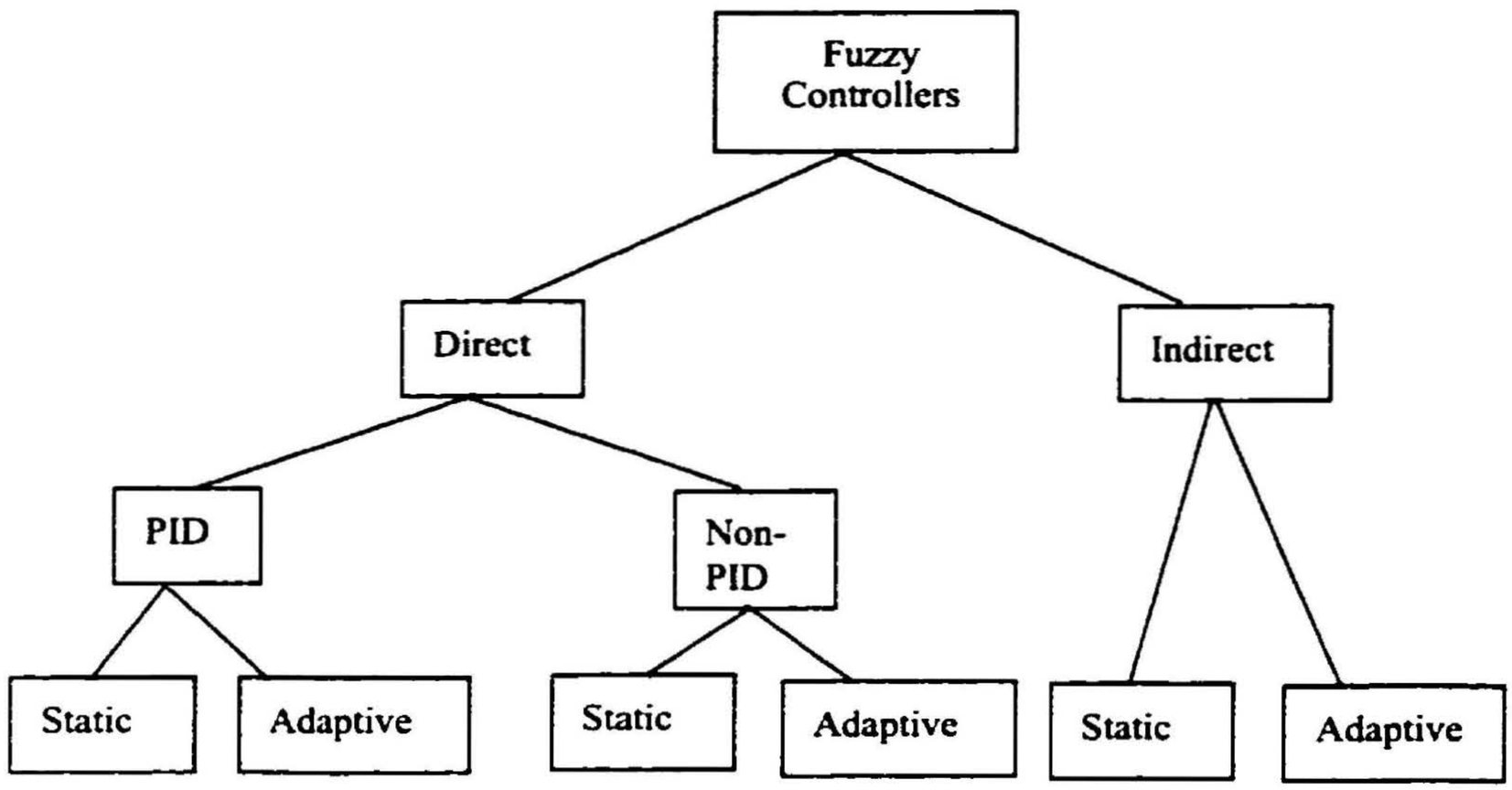


Figure 2.1: Classification of fuzzy controllers

two main classes: direct fuzzy control and indirect fuzzy control. The direct fuzzy control computes the controller action through the fuzzy inference and the fuzzy outputs are directly used to compute the controller signal. In the indirect type, the rule based fuzzy system identifies an approximated model of the process. This identification can be performed either by learning during the process of control or by training through observed input/output data. For each class, the controlling can be either static or adaptive. The direct fuzzy controllers are either fuzzy PID type or non-PID types. Much of the emphasis is placed on the PID type controllers.

2.3.1 Direct Fuzzy Controllers

This is the most common type of FLC used in past applications and research. As mentioned in the previous chapter, the first fuzzy controllers of Mamdani [8, 9] fall into this category. The FLC is directly placed within the closed-loop control system. Using the feedback signals, the fuzzy controller infers the controller action necessary to drive the process.

Static Fuzzy PID Controllers

The majority of direct fuzzy controllers are fuzzy PID types (Refer Figure 1.1 in Chapter 1). A few past applications are briefly shown in the Table 2.1. It is clear from the table that the majority of fuzzy PID applications belong to conventional two-input type fuzzy PI or PD type controllers. The hybrid actions of both incremental fuzzy PI and absolute fuzzy PD controllers are then used in making the fuzzy PID configurations. Aspects of different fuzzy PID configurations or structures are detailed in Chapter 3. In the majority of applications in the 80's and before, the fuzzy inferences or fuzzy calculations were done off-line. The control decision table or look-up tables were used for on-line implementations or simulations. Also in the majority of those cases, the design or tuning was ad hoc in nature. More systematic designs have been reported in the recent years. Fuzzy controllers can be trained when they are implemented through neural network [77]. Also, the genetic algorithm allows FLC systems to be designed through systematized exhaustive search [78]. Although there are some limitations and practical deficiencies in the latter methods, theoretically the design techniques do not require any knowledge of the process to be controlled. However, some approximated process model properties are always valuable for reducing the design time and computational overhead to achieve fast convergence of the solution. Use of other classical or linear control theories for FLC designs have recently become more popular. This is mainly due to the recent recognition of the intelligent control techniques by many classical control experts. The sliding-mode approaches [68, 73, 79], cell-to-cell mapping technique [41] and use of linear PID tuning heuristics [67, 71, 74] demonstrate the enhancement of the fuzzy control. In almost all applications shown in the table, FLCs are mostly compared to linear PID controllers. In many cases they have shown improved performance.

In certain cases multiple fuzzy PID controllers with switching have been used to get improved performance [29, 65, 72]. In [65], an additional PI rule base was used for fine control near set point. In [29], the two-input fuzzy PI controller showed poorer performance than the linear controller. Therefore an additional PD controller was implemented for coarse control, and near the command state the PI controller was used to get steady state performance. In [72], which dealt with a pH control system, the nonlinear gain variation with respect to the level of pH control was divided into three approximately linear regions. Depending on

Table 2.1: Past applications of static fuzzy PID controllers

[Ref.]'19'	Type	inputs	Rules	Application	Remarks
[9] '75	(a) PI (b) PI	$e,\Delta e$ $e,\Delta e$	15 09	Steam Engine	(a) Computes change in heat input (b) Computes change in throttle opening
[54]'76	(a) PI (b) I (c) P	$e,\Delta e$ e e	06 05 03	Flow control of a warm water plant	(a) Computes change in flow (b) Computes change in flow (c) Computes amount of flow (steady state values are known *a priori*)
[53]'77	PI	$e,\Delta e$	15	Temperature control of a stirred tank	Tuning is performed by changing the quantization levels
[16]'79	PI	$e,\Sigma e$	49	For many known and ill defined systems	Different tuning aspects of FLCs were demonstrated
[63]'88	(a) PI (b) PID	$e,\Delta e$ $e,\Delta e,\Delta^2 e$	49 343	Electric Furnace	Gain scheduling scheme is used for on-line adjustments of scale factors.
[64]'89	PI	$e,\Delta e$	64	Continuous biological process	Faster settling and less overshoot than the linear PID was observed.
[65]'89	Two-stage PI	$e,\Delta e$	4 & 6	Position control of a servo motor	Two rule bases for coarse and fine control.
[66]'89	PD	$e,\Delta e$	08	Control of welding torch position	One-dimensional two-variable FLC. Fuzzy filter is incorporated
[19]'90	PI	$e,\Delta e$	04	Several linear and non-linear plants	Employs simple four linear rules.
[41]'91	PD	$e,\Delta e$	05 & 9	Position control of a dc motor	Cell-to-cell mapping technique has been used to find the optimum FLCs.
[67]'92	(a) P (b) PI	Δe $e,\Delta e$	07 09	Pressure control of a Clinker cooler grill	The linear PID gains are related to determine the linear gains of FLCs.
[68]'92	PI	$e,\Delta e$	49	Position control of an inverted pendulum	The design is based on fuzzy sliding-mode control.
[69]'93	PI	$e,\Delta e$	40	Speed control of a hydraulic motor.	Rules are derived by trial and error
[70]'93	PI	$e,\Delta e$	09	Control of rail car air-conditioner.	Air conditioning capacity was obtained by the fuzzy inference
[46]'93	PID	$e,\Delta e$	49x3	Several linear and non-linear processes.	Gain scheduling scheme. Rules derived from the step-response behavior.
[71]'93	PID	$e,\Delta e$	49	Several linear and non-linear processes.	Gain scheduling scheme. The PID gains are parameterized by a single term.
[72]'94	PI	$e,\Delta e$	25x3	Stirred tank reactor for pH control.	Three rule bases for the three regions of the process gain.
[43]'94	PD	$e,\Delta e$	08	Second Order system	Sliding mode approach is used for obtaining an FLC for contour tracking.
[73]'94	PD	$e,\Delta e$	25 & 49	Position control of a dc servomotor.	The membership parameters are adjusted to obtain the desired switching surface for fuzzy sliding mode control.
[74]'94	PID	$e,\Delta e$	25	Position control of a dc servomotor.	The linear gains of the FLC are obtained by relating to linear PID gains.
[29]'95	PI & PD	$e,\Delta e$	04x2	Several linear and non-linear processes	PI & PD switches with respect to the feed back error for coarse and fine control.
[75]'95	PI	e	07	Stirred tank bio-reactor	Error driven gain scheduling for the self-tuning of linear PI controller.
[76]'95	PID	$e,\Delta e$	49x2	Temperature control of a separator and pH control.	Gain Scheduling scheme. Two sets of rules are derived from the behavior observed by moving the dominant poles.
[77]'95	PID	$e,\Delta e$	08	Speed control for a servomotor.	Rules are created for gain scheduling. The FLC is implemented via a radial basis function neural network for the tuning.
[48]'96	PID	$e,\Delta e$	49	Second and third order linear models	The rules are derived from different regions of the error state space trajectory.
[78]'96	PI	$e,\Delta e$	09-25	Inverted Pendulum	The genetic algorithm is implemented to find the optimum rules and memberships.
[79]'96	PD	$e,\Delta e$	09	Stabilization of an inverted pendulum by rotating the base.	Rules are derived from partitioning the state space. The tuning is performed from the sliding mode principles.
[80]'96	PI+D	$(e,\Delta e)$ + $(y,\Delta y)$	04+04	Several linear and nonlinear processes.	Two linear rule bases are used for fuzzy PI and D controllers each with two inputs.
[25]'97	PD+I	$e,\Delta e$	04+04	Laboratory flexible robot arm.	Two linear rule bases for PD and I controller each with two inputs.
[81]'97	PI	$e,\Delta e$	49x2	Temperature control in a solar plant.	Gain scheduling scheme. Fuzzy associated memory is used for inference.
[82]'98	PI	$e,\Delta e$	09	Level control in continuous casting.	The fuzzy controller is active when the system is perturbed and near set point a linear PID is used.

the level of pH control, three fuzzy PI controllers switched from one to another during the transient.

The error driven gain scheduling type fuzzy PID controllers (GS type) are less popular. This is due to the difficulty in generating error based self-tuning rules for the three gains. As a result the GS types normally utilize the linear PID tuning rules to formulate the knowledge base structure. As an example, in [71], the Ziegler-Nicholos tuning expressions have been simplified to a single unknown parameter to deduce the control rules. In [76] the movement of closed-loop poles in the control response is used for generating the control rules for the three gains. In the performance-based gain scheduling schemes, the rules have been derived from the basic behaviour of the PID gains on the plant response [5]. The performance-based GS type fuzzy controllers have higher degree of self-organizing properties and adaptive qualities than error driven GS type fuzzy controllers.

Static Fuzzy Non-PID Controllers

Other classes of direct fuzzy controllers can be treated as non-PID types. Although the TSK based fuzzy modelling is non-PID in nature, the method can be implemented for many types of control problems including PID types. The majority of other classes are more specific problem oriented than general or conventional. A good example is the automatic train operation system in [83]. The control system is guided by the optimization of many performance indices (e.g. human comfort, safety, energy consumption etc.). This fuzzy controller is implemented as a predictive controller for the train operation. The multi-input-multi-output (**MIMO**) fuzzy control of aircraft flight control [84] is direct non-PID type fuzzy control and uses rate of descent, glide slope and air speed for controlling the rpm and elevator adjustments simultaneously by a single rule base inference. Several non-PID type fuzzy controlling systems in Japanese industry and home appliances are detailed in [11]. In some cases a fuzzy sensor has been used to provide the feedback signal to the main fuzzy controller. The fuzzy sensor processes the human experience that identifies the quality of output data based on the observed inputs. In the combustion control problem [11], the evaporation rate, feeder on-off conditions and pressure loss of the incinerator are used to infer the refuse quality in a quantitative form to feedback for the fuzzy controllers inference. Based on the amount and type of dirtiness (muddy, oily etc), the fuzzy sensor assesses the

washing time in automatic washing machines. Similarly the fuzzy sensor in the vacuum cleaner [11] determines the suction power from the floor type and dust amount. The direct use of oral instructions (fuzzy commands) for control is described in [12]. The oral commands are first synthesized in a voice recognition system and the signals are then processed in a fuzzy controller to park a model car. The rules have been developed from the human control that is used in handling the steering wheel of a car.

Direct Adaptive Fuzzy Controllers

In an absolute sense, a simple PID regulator is an adaptive controller. But in control engineering the word adaptive is more explicitly defined for differentiating control algorithms. According to Åström and Wittenmark [32], an adaptive controller is "*a controller with adjustable parameters and a mechanism for adjusting parameters*". Other vocabulary based on this notion include "*self-organizing control*" (**SOC**) and "*learning control systems*". Although some fuzzy controllers have been called self-organizing or self-adjusting regulators, they are actually static controllers. Therefore direct adaptive controllers should be able to change the gains or tuning parameters of the controlling unit under the change of plant dynamics or/and parameters. The first fuzzy adaptive controller was introduced in 1979 [47], and again Mamdani has contributed to these developments. Adaptive controllers usually contain two extra components, a "process monitor" and an "adaptive mechanism" [15]. In order to classify the fuzzy adaptive controllers in a more specific sense, the direct fuzzy adaptive controllers are classified into two categories.

Type I The rule base parameters of the fuzzy (or fuzzy PID) controller are updated via fuzzy or non-fuzzy system.

Type-II The linear gains of a traditional PID controller are updated via a fuzzy system.

The direct fuzzy adaptive controllers in the past research are shown in Table 2.2.

The Type I adaptation includes reformulating or altering the rules and adjustment of membership parameters. The changing of knowledge base parameters (either rules or memberships) alters the nonlinear charcteristics of the fuzzy controller output. The changes in gains cause overall changes in performance. Direct update of rules is cumbersome and hard

Table 2.2: Past applications of direct fuzzy adaptive controllers

[Ref.] 19'	Type	Inputs	Adaptation by (Type)	Application	Remarks
[47]'79	PI	$e.\Delta e$	Rules (I)	Several linear models including time lags.	Performance measures are computed to update the fuzzy relational matrix.
[85]'88	PID	$e.\Delta e.\ \Delta^2 e$	Rules (I)	Several higher order models with added non-linearity.	Performance index table is designed to update the fuzzy rules.
[86]'88	PI	$e.\Delta e$	Rules (I)	Two-D robot manipulator.	Performance index table is designed to update the fuzzy rules.
[87]'88	PI	$e.\Delta e$	Rules (I)	Temperature control of a heater and speed control of a dc motor.	Performance index table is designed by using fuzzy rules to update the control rules.
[88]'91	PI	$e.\Delta e$	Rules (I)	Muscle relaxant process (second-order process with time delay)	Performance index table is designed by using fuzzy rules to update the control rules.
[89]'93	PI	$e.\Delta e$	Rules (I)	Anti-skid braking system	The inverse fuzzy model (performance table) updates the rules by following a reference model.
[90]'98	PI	$e.\Delta e$	MF(I)	Level control of a tank	The performance index table updates the fuzzy singleton values in the consequent during the learning.
[91]'82	PI	$e.\Delta e$	MF (I)	Control of the material level of a die casting plant	Six performance indices are used to modify the input MFs to provide adaptation.
[92]'91	PI	$e.\Delta e$	Consequent MFs (I)	First-order process and an inverted pendulum.	The adaptation is based on the predicted error (estimate) and observed maximum error.
[93]'92	PD	$e.\Delta e$	Fuzzy output (I)	Position control of a servomotor.	The fuzzy controller driven by integral error is used to alter the fuzzy PD output.
[94]'92	PID	$e.\Delta e$	Rules (I)	Linear second order damping and oscillating models.	Additional fuzzy rule base is used for modifying rules of the main fuzzy controller.
[95]'95	PI	$e.\Delta e$	Rules (I)	Temperature and Elevation controls.	The look-up table entries update by changing weights allocated for six performance zones.
[96]'90	PI & non-PID	$(e.\Delta e)$ & $(fq.co)$	Scale factors (II)	Octane control of a oil refinery unit (catalytic reformer)	The ratio of the two consecutive errors is taken as the performance measure to update the scale factors.
[5]'95	PID	e	Linear PID gains (II)	Non-minimum phase system.	Fuzzy supervisor for adjusting the linear PID gains (fuzzy GS type)
[58]'95	PID	$e.\Delta e$	Scale factors (II)	Linear second order system	According to the response overshoot the gains of the FLC are updated
[97]'89	PID	e	Linear PID gains (II)	Tracking of a planer robot manipulator	The rule de-coupled fuzzy inference makes the on-line gain adjustments. (fuzzy GS type)
[98]'94	PI	e	Linear PI gains (II)	Speed control of a dc motor.	Fuzzy supervisor for adjusting the linear PI gains. (fuzzy GS type)
[99]'95	PID	e	Linear PID gains (II)	Speed control of a dc motor.	Fuzzy supervisor for adjusting the linear PID gains. (fuzzy GS type)
[100]'92	PID	$e.\Delta e.\ \Delta^2 e$	Scale factors and rules (I & II)	Second order delay system.	Two additional fuzzy controllers driven by the performance measures are used to update scale factors and rules of the main fuzzy controller.

to represent in a mathematical sense. Therefore in most practical applications the rule alteration is accomplished by changing the fuzzy output look-up table. Therefore the real rule changes are implicit. In the first adaptive or self-organizing controller the fuzzy look-up table was modified via a learning algorithm. Although the controllers described in [85]- [89] have some differences with respect to the process of adaptations, the overall adaptive structure is equivalent to the first adaptive fuzzy controller developed by Procyk and Mamdani [47]. Some modifications or an improvement to this type was recently shown in [95]. The performance monitor decision table for correcting the controller uses heuristic knowledge. In essence. the performance monitor is an inverse fuzzy controller model [89]. The first issue in this controller is to determine which inputs in the past have contributed to present poor or good performance. This requires the estimation of plant time lag or the delay in reward because this determines when to apply the correction. In [47] an incremental model of the process relating the changes in the process input (control output) and changes in the process output is used to determine the amount of reinforcement required in the controller. In the SISO case it is a linear mapping. The adjustments are made to the control action that is responsible for the current poor performance. In other words the system looks at the entries in the control table which have resulted in poorer performance and then reinforcement is used to update the controller. Mathematically this is accomplished by updating the fuzzy relational matrix. If performance measure rules indicate finite corrections. then at each sampling state the fuzzy relational matrix is updated until the desired performance is established. The performance table acts like a model reference. In [89] a reference model is provided for quantifying the desired performance. The inverse fuzzy model is fired by the error signal of the reference model with respect to the actual plant output and it is then used to modify the rules (or look-up table) of the fuzzy controller. In [93] the alteration to the control action at any given time instant has been made proportional to the sum of error signal at one cycle time before the current time instant. The error integral then makes an additional fuzzy inference to find the future alteration required for the adaptation.

The membership functions or fuzzy set definitions are chosen to represent the linguistic terms or term values taken by the controller variable. Moreover, the membership function is the primary transfer element in the fuzzy inference for converting the linguistic knowledge (or fuzzy values) into mathematical or numeric terms. In most cases the sharpness of such

definitions cannot be precise and the optimum performance can be sought only through their variations. Many off-line techniques can be found for searching for the best fuzzy sets of a given set of linguistic variables [1]. Changing membership parameters and changing rules have a similar effect in terms of nonlinear control. The work in [91] evaluates six different performance measures to update the membership parameters of the consequent fuzzy variables. In [92] the control supervisor changes the consequent membership parameters. The supervisor uses the correlation function to evaluate the reliability of the predicted output and thus the quantified information is related to the membership function parameters.

The Type II adaptive controller is analogous to on-line tuning (auto tuning) of a linear PID controller. The performance monitor evaluates the performance attributes based on the response pattern and either a fuzzy or a non-fuzzy algorithm is implemented on-line to correct the linear gains. In [96] the performance is measured by the ratio of two consecutive error measurements and based on this scale factors (normalizing gains) of the two fuzzy controllers are changed to increase or decrease the speed of the response. For the applications in [5, 58, 97, 98, 99], linear PID controllers are on-line tuned by fuzzy controllers placed at an outer level of the closed-loop system (supervisory control). According to the classification mentioned in Chapter 1, these controllers are performance based GS type fuzzy PID controllers. They have the adaptive capabilities to change the PID parameters under varying process conditions. In [97], the manual PID controller tuning heuristics are coded to produce rule bases for on-line fuzzy tuning of a robot manipulator. Additional servo experts are placed at the control joints to monitor the response behaviour in order to feed the data (performance attributes) into fuzzy controllers. The fuzzy inference compute the necessary gain changes during the adaptation. Since the performance attributes are based on the response patterns (e.g. overshoot, speed of response, steady off-sets etc.) the adaptation operates with a lower bandwidth than that of the main control loop. In [5] the same concept was used to tune a linear PID controller for a servomotor. Similarly in [98, 99] linear PID controllers were on-line tuned using supervisory fuzzy control algorithms.

With two additional fuzzy controllers, the self-organizing structure in [100] simultaneously changes both the rules and the gains. The fuzzy rules for adjusting scale factors are similar to a performance based GS type PID controller. By comparing the response characteristics to a predetermined response pattern, another set of fuzzy rules were developed to change the

rules of the main fuzzy controller. Therefore this work is unique because it uses both linear and nonlinear tuning to obtain the self organizing structure of a nonlinear PID controller.

2.3.2 Indirect Fuzzy Controllers

Process identifications by input-output data has been traditionally done by using statistical methods, e.g. linear regression, stochastic approximation and correlation analysis. Combining the qualitative and quantitative data allows the system dynamics to be modeled with less effort than with statistical learning algorithm [1]. Fuzzy modelling is one of the main successes in FLS designs. Fuzzy relational descriptions have been studied for modelling control problems [101]–[105]. Those fuzzy relational concepts have been then used to approximate many industrial plant systems [102, 103]. Different fuzzy modelling and learning techniques are well described in [104]. When a model identification algorithm is implemented on-line for a control problem, the system can easily account for any changes happening to the process or to its surroundings and therefore adaptive control can be automatically achieved. For this reason, the modelling or identification through fuzzy-neural nets is termed as adaptive fuzzy systems [1, 104]. The computational and lengthy optimizing search techniques have practical limitations to use in true adaptive control applications. However, a practical model-based adaptive algorithm can be implemented through on-line learning scheme (i.e. by continual updating of the fuzzy model). Graham and Newell [105] implemented an adaptive control scheme based on the fuzzy modelling concepts to successfully control the level height of a mass storage tank. The on-line identification will modify the predefined model (theoretically an empty model) to match the process. Therefore performance is enhanced through adaptation. The model-based controller then computes the new control output using the prediction from the fuzzy model. The system predicts nine values for the performance and the one having the highest score of performance is selected to compute the control signal. Harris and other co-workers [35] have shown several applications of fuzzy model-based controllers and they have labeled this method as indirect self-organizing fuzzy controllers. In their method the controller output is predicted by the inverse causality mapping of the fuzzy relational map. The same composition operator is used for the inverse mapping. Wang [104] has analyzed several direct and indirect adaptive controllers in his book. The model based fuzzy controller predicts two nonlinear functions. In addition to the main controller that has been derived

from the identified model, a supervisory controller is identified to operate when the error is big. The supervisory controller produces no signal when the error is small. During the adaptation, the parameters of the fuzzy basis functions are updated. Recent work of using model based fuzzy control is described in [106] and they have successfully used it to control temperature and flow rates of a heat exchanger.

2.4 Design Methods

Fuzzy control theory is relatively new. Therefore systematic development and tuning of fuzzy controllers are not yet easily available in hard textbooks as compared to classical control theory. However, some researchers have published informative books to provide useful guidelines for design of fuzzy controllers [1, 5, 15, 35, 104]. To some extent each research paper can be considered to be a systematically analyzed design technique for a given application domain. However, they do not sometimes provide enough evidence or reasons for selecting many parameters related to the chosen FLS and also adopt ad hoc or trial and error methods for final tuning. Therefore a "*good fuzzy controller design*" should:

1. allow the method to be used by another user for the same or a similar class of a problem, and

2. remove the ad hoc design nature to tune with a systematically developed methodology or heuristically driven search.

2.4.1 Takagi-Sugeno Fuzzy modelling Approach

In 1985, Takagi and Sugeno [39] introduced a new fuzzy rule representation and a systematic optimization method to design fuzzy controllers. The method allows a systematic development of an optimized fuzzy model for a given set of input/output data. The consequence of each fuzzy rule is defined as a linear relation of the input variables. However, the TSK method can also be extended to represent the consequences as nonlinear functions of the input variables [50]. The fuzzy variables are initialized with the available knowledge about the process and its operation. The controller is systematically built in three steps, namely, consequence parameter identification, premise parameters identification and choice of premise

variables. Each step is interrelated. With the available input data, the method optimizes the performance index. In [39] the root mean square of the error is chosen as the performance measure. The TSK method is systematic. It can start with empty rules. The procedure then allows systematic search of fuzzy parameters. Later Sugeno and Kang [59] have shown the effectiveness of this method by applying this design for a multi-layer incinerator. In [60] Sugeno and Kang have further improved this fuzzy identification algorithm by incorporating a better search mechanism to identify the premise parameters.

2.4.2 Learning Schemes

The implementation of fuzzy controllers in a neural network or a similar learning scheme allows the fuzzy parameters to be tuned by a set of input-output data [1, 42, 104]. This model can be either the process model relating input output behaviour or inverse models of the process to determine the controller signal requirement in a control loop.

Neural nets with back-propagation are known to have universal approximating capability. The successes of neural networks, particularly in control [107], motivated researchers to implement fuzzy systems through neural nets so that the fuzzy system also becomes a universal approximator [1, 104, 108]. Training of fuzzy systems using many techniques to match input/output data are described in [104]. As an example the gradient decent training uses three layers to produce the unknown process dynamic function as a fuzzy identifier. The function which is generated by fuzzy inference is described in terms of three sets of unknown membership parameters (input modal positions, parameters related to the width of the Gaussian functions and output fuzzy singleton positions). Using back-propagation the unknown fuzzy parameters are trained. In essence the same nonlinear mapping of a given fuzzy system is equivalently represented by a neural network and the fuzzy system parameters are then conveniently trained using input/output data. The advantage of this representation is that fuzzy systems have clear physical meaning and therefore the initial parameters can be chosen from the available knowledge [104]. However, implementation of a given fuzzy system having approximation capabilities through a learning scheme is not always feasible [1]. Therefore some mathematical alterations are always necessary. Fuzzy systems or inferences have been transformed to standard additive model in [1] and fuzzy basis functions in [104] for convenient implementations in learning algorithms. The neuro-fuzzy systems have some limitations with

regard to implementation of the conventional min-max gravity reasoning. This is mainly due to the existence of a multi-phase solution structure and the non-existence of algebraic type solution in the min-max functions. Therefore in most neuro-fuzzy systems the reasoning uses product-sum inference and the weighted sum of the centers of the consequent fuzzy variables in the de-fuzzification. The membership functions (premise variables) are either Gaussian or B-splines types [42] so that each fuzzy variable can be expressed by a single expression. Many other similar training algorithms are well described in [1, 104], namely orthogonal least squares method, nearest neighborhood clustering etc. Kosko [1] has used standard additive fuzzy systems to control many real time applications. The fuzzy system allows the designer to incorporate the available initial knowledge into an approximated model and hence the learning algorithm can be made to converge within a short period of time [104]. This is the key advantage of using fuzzy systems in neural or other learning schemes.

2.4.3 Use of Evolutionary Algorithms

Genetic algorithms (**GAs**) have recently become popular for overall tuning of multi-parameter control systems. By mimicking the principles of natural selection in an evolutionary process, GAs are able to evolve the solution to many optimization problems. Unlike gradient algorithms, GAs are not mathematically guided solvers. A GA performs the optimization process with a population of individuals, each of which represents a search point in the space of potential solutions to a given problem. Therefore for design one must know the correct range for each variable in the fuzzy system. For each variable a random population is defined. The method evolves a population of candidate solutions for the problem by applying a set of stochastic operators, such as *crossover*, *mutation* and *reproduction*. Each solution is evaluated using a fitness or objective function. A general GA defines equal search space to each parameter. In [109] the input membership parameters were obtained by using a simple genetic algorithm for centering a cart on a one-dimensional track. A hierarchical distributed genetic algorithm is described in [78]. The search in the latter is based on a multilevel resolution, where higher level clusters investigate wider search spaces with lower resolution than lower level clusters. A genetic algorithm takes considerable computing time before reaching a final solution. Also the accuracy depends on the length or the resolution of the binary vector assigned to each variable. Higher resolution often makes the search time longer and therefore

in the implementation there is a trade off between design time and accuracy. Therefore designs are generally performed off-line. However, a GA with a fast searching technique (called microgenetic) that suits adaptive controller has been reported in [110]. The robustness of a design heavily depends on the fitness function. This is in fact a problem for any numerical search technique. In [111] many GA based designs are compared with respect to different cost functions.

2.5 Summary

This literature survey chapter has shown the past development of fuzzy logic control applications. The past theoretical investigations of fuzzy systems made fuzzy control more transparent than other soft computing techniques. Over the years the original fuzzy inference process has been subjected to many alterations, particularly for better design and implementation purposes. Therefore fuzzy systems have an ongoing evolution of their theory. On the other hand, fuzzy systems have been used extensively in many industrial and engineering products, not necessarily as sales gimmicks, but as genuine improvements to the available controlling systems [1].

The past FLC applications have been classified into different categories. The direct fuzzy controller types have been used in many applications. The adaptive fuzzy controllers have been developed along with static fuzzy controller designs. These adaptive controllers can again be classified into different groups, depending on the adaptation procedure of the controller. The recent developments of learning techniques and fuzzy-neural systems allow the processes to be modeled by using fuzzy rules and as a result, the indirect type fuzzy controllers are now more popular in research. Numerically driven search techniques, such as neural network models, genetic algorithms etc., are quite powerful in handling multi-dimensional fuzzy controller models. Therefore automatic rule generators with proper input output behaviour and adaptive learning schemes are being extensively investigated in research.

In most designs (static or adaptive), the final tuning of controller requires considerable guessing and trial and error simulations for achieving a stable controller. As an example, a self-organizing controller may require tuning of its scaling factors and other gain variables in order for the self-tuning mechanism to be functional. Although fuzzy systems have been

labeled as model free, in real time controller applications the linguistic or expert knowledge is insufficient for achieving an absolutely model free and stable controller. Under these circumstances the only possibility of designing a system without the plant model would be by trial and error technique. The off-line numerical search techniques using computer simulations require the approximated plant model. If a numerically driven search technique can be implemented on-line, then such a design method would absolutely become a model free design. Again for plants whose unstable dynamics are hazardous to the surroundings or can causes damage to the system, those techniques would become impractical for on-line implementations.

Chapter 3

Systematic Development of Fuzzy PID Structures and Analysis of Fuzzy Inferences

3.1 Introduction

The last two chapters outlined the present status of fuzzy controllers and briefly described the current research in the area of fuzzy control. It has been observed that the majority of fuzzy controller applications fall into the category of fuzzy PID (or PID like [15]) controllers. The fuzzy PID controller by its definition verbalizes the linear difference equation of a PID controller. Although Mamdani and his co-workers [8, 9, 47] never attempted to define such a controller in the design principles, the fuzzy rules of those controllers are analogous to verbalized fuzzy PI rules. As a linear controller has many forms for its representation, the fuzzy PID controller also can be implemented in different forms. As an example, in [24], the linear PID action is transformed to an incremental form using a bi-linear transformation and it is then verbalized to produce an incremental fuzzy PD controller. In the conventional fuzzy controller (Mamdani-type) the nonlinear mapping of error inputs produces an absolute fuzzy PD signal. Tables 2.1 and 2.2 in Chapter 2 indicate that a fuzzy PID controller can differ structurally based on (a) numbers and types of inputs and (b) fuzzy rule base representation.

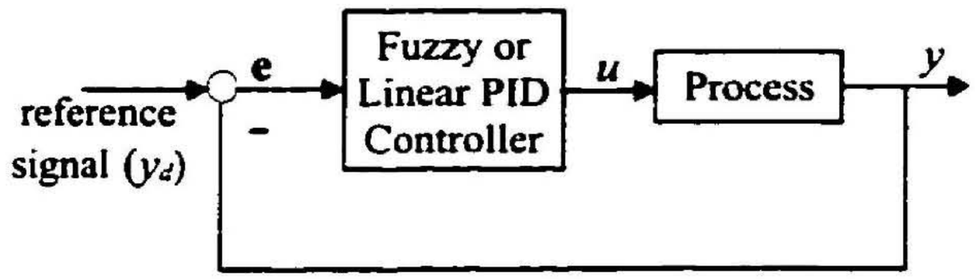


Figure 3.1: Cascade type feedback PID controlled system

In the majority of cases the Mamdani's configuration has been used to produce either fuzzy PI or fuzzy PD controllers with the error inputs namely error ($e(t)$) and error change ($\Delta e(t)$). The combination of the latter was used to produce fuzzy PID controllers [48. 74. 77]. In a few cases the third error variable, namely the change of error difference ($\Delta^2 e(t)$) has been used for deriving three dimensional fuzzy PID rules [63. 85. 100]. This is understandable since the three input rule structure requires a larger number of rules compared to the conventional configuration. In a few cases. some attempts have been made in the past to identify the fuzzy controllers in various forms with different knowledge base representations [112].

Therefore the first issue related to this chapter is the identification of different fuzzy PID controllers including commonly available types. The next step in this systematic investigation is the analysis of these controllers. Therefore in the second part the closed-form solutions to the different fuzzy systems are obtained in general forms. The closed-form solutions enable these controllers to implement in computer controlled systems and also to devise a tuning criterion. Therefore this chapter intends to address the following issues.

1. Identify the different fuzzy PID controllers including commonly available types.

2. Obtain closed-form solutions to the general fuzzy controller systems.

3. Identify the functional properties and differences of different controllers defined in (1).

Linear PID controllers can be classified into different categories with respect to the positioning of the three terms in the closed-loop control system. In computer controlled single-input-single-output (SISO) plant systems the PID controller in the cascade form is commonly used (Figure 3.1). Sometimes the derivative controller is separately generated using the feedback response signal to avoid the derivative kick associated with noisy signals. These other types [25] can be obtained by extending the fundamental principle outlined in this

chapter. Therefore, this study focuses on the SISO controller system shown in Figure 3.1.

The study is further narrowed to the development of direct action (DA) type controllers only. The rest of this chapter is organized into three main parts. Section 3.2 defines and identifies different fuzzy PID controllers. In Section 3.3, an analytical calculation is performed to derive closed-form solutions for different fuzzy controller systems. In Section 3.4, the nonlinear and linear PID gains are identified for each fuzzy PID controller structure. Finally the functional properties are compared.

3.2 Construction of Fuzzy PID Controllers

3.2.1 Fuzzy PID Elements

As shown in Figure 3.1, the error driven linear PID controller signal at any time instant (n) with a sampling period T_s can be expressed in two forms; (3.1) shows the output in the absolute form $(u_{\mathrm{PID}}(n))$ while (3.2) shows it in the incremental form $(\Delta u_{\mathrm{PID}}(n))$.

$$u_{\mathrm{PID}}(n) = K_{\mathrm{P}} e(n) + K_{\mathrm{I}} T_s \sum_{q=0}^{n} e(q) + (K_{\mathrm{D}}/T_s)\, \Delta e(n). \tag{3.1}$$

$$\begin{aligned} &\Delta u_{\mathrm{PID}}(n) = K_{\mathrm{P}} \Delta e(n) + K_{\mathrm{I}} T_s e(n) + (K_{\mathrm{D}}/T_s)\, \Delta^2 e(n), \\ &\text{and } u_{\mathrm{PID}}(n) = u_{\mathrm{PID}}(n-1) + \Delta u_{\mathrm{PID}}(n). \end{aligned} \tag{3.2}$$

The terms K_{P}, K_{I} and K_{D} stand for proportional, integral and derivative gains respectively. Using the feedback response signal $(y(n))$ and the reference or desired signal $(r(n))$ at the n-th sampling instant, the error variables are defined as,

$$\left.\begin{aligned} &\text{Error:} && e(n) = r(n) - y(n) \\ &\text{Error difference:} && \Delta e(n) = e(n) - e(n-1) \\ &\text{Change of error difference:} && \Delta^2 e(n) = \Delta e(n) - \Delta e(n-1) \\ &\text{Sum} - \text{of} - \text{error:} && \Sigma e(n) = \textstyle\sum_{q=0}^{n} e(q) \end{aligned}\right\}. \tag{3.3}$$

For developing fuzzy PID controllers the error terms (or PID actions) in equations (3.1) and (3.2) are considered as fuzzy variables. The above error variables in (3.3) are the four basic inputs to any fuzzy PID controller configuration. Since the linguistic expressions or terms are qualitative, the contexts of these variables are defined in a general universe of discourse.

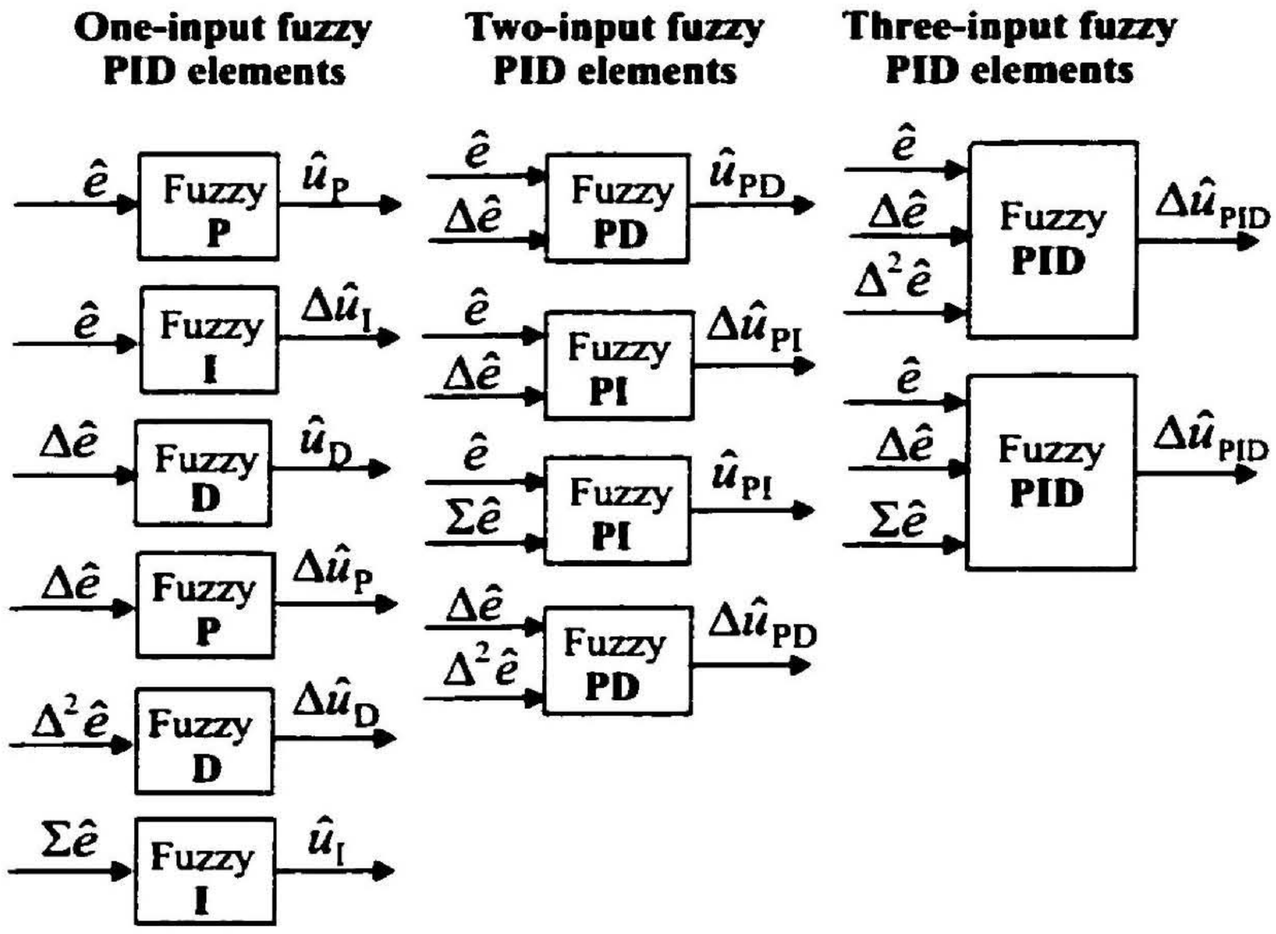


Figure 3.2: Fuzzy PID structural elements

Therefore using the scale factors (S) the normalized variables for (3.3) are defined. For convenience the time instant notation is sometimes dropped from some expressions.

$$\hat{e} = S_e e,\ \Delta\hat{e} = S_{ce}\Delta e,\ \Delta^2\hat{e} = S_{rce}\Delta^2 e,\ \Sigma\hat{e} = S_{se}\Sigma e. \tag{3.4}$$

Where $\hat{e}$, $\Delta\hat{e}$, $\Delta^2\hat{e}$ and $\Sigma\hat{e}$, are the normalized error variables corresponding to the error terms e, Δe, $\Delta^2 e$ and Σe respectively. The de-fuzzified output after the fuzzy reasoning is represented by $\hat{u}$.

With respect to each error element in (3.1) and (3.2), either absolute or incremental PID elements corresponding to the individual PID actions can be identified. These actions can be of either associated or dissociated forms. When the rule base is considered with one input fuzzy rules, the control rules can be identified for individual elements in each PID signal. Similarly when two or three terms (or actions) are taken together, then associated PID actions can be inferred while using either two or three input fuzzy rules. Hence, different fuzzy PID elements can be constructed with respect to total number of inputs as shown in Figure 3.2. The rule base corresponding to each PID element is identified by coupled or de-coupled fuzzy rules of the form "**If** (*input* 1 **and** *input* 2...) **then** (*output*)". In the case of two-input configurations, only PD and PI controller elements are considered. A subscript

with the normalized output variable $\hat{u}$ is used for identifying the corresponding action in a fuzzy PID controller. In deriving a practical fuzzy PID structure the following remarks are made:

***Remark* 3.1** It is difficult to formulate control rules with the input variable sum-of-error Σe, as its steady-state value is unknown for most control problems. As an example, the load disturbances at the plant input, dead weights and friction in drive systems are always unknown. Therefore it is difficult to identify membership values and their locations in the universe of discourse for defining control rules corresponding to steady-state conditions. It is possible to use this variable only if a *priori* knowledge about the steady state conditions is available [28].

***Remark* 3.2** For any fuzzy PID controller, the error (e) is considered the necessary input for deriving any PID structure. The error input provides the nonlinear proportional actions through the fuzzy inference. For any system to drive from a dead state, proportional control is the basic action required from the three-term PID controller. For example, in case of a steady off-set in the system response, or in case of a time-delay process, the magnitude of all error derivatives becomes negligible. In those circumstances the steady error is the only available information that can provide a finite control action to divert the output from a dead situation.

3.2.2 Fuzzy PID Controller Structures

By taking different combinations of the fuzzy PID structural elements defined in the previous section, different fuzzy PID configurations are now formed. Based on *Remarks* 3.1 and 3.2, some of the structural elements can be considered to be "*bad*" and can be eliminated in building a fuzzy PID structure. Therefore in this systematic investigation, six types of controllers are described for comparison. In 1975, Zadeh published a three-part paper [113] describing the fundamentals of fuzzy logic principles for using decision-making systems. Zadeh included many definitions and concepts to generalize the broader perspectives of *humanistic* systems. The FLC systems use some of those concepts for describing the knowledge base.

Define the linguistic variables that correspond to the input scaled variables $\hat{e}$, $\Delta\hat{e}$ and $\Delta^2\hat{e}$

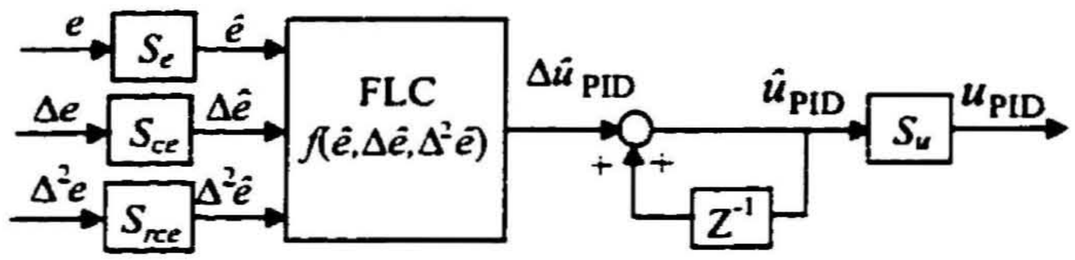


Figure 3.3: Three-input fuzzy PID (Type I)

as $\{E_i\}$, $\{\Delta E_j\}$ and $\{\Delta^2 E_k\}$ respectively. The indices i, j and k represent the linguistic values or fuzzy states of the input fuzzy variables and their ranges are: $i = 0, 1, 2, \ldots, N_1 - 1$, $j = 0, 1, 2, \ldots, N_2 - 1$ and $k = 0, 1, 2, \ldots, N_3 - 1$, where N_1, N_2 and N_3 denote the total numbers of fuzzy states assigned for each of the fuzzy variables. Let the de-normalizing scale factor S_u be given by the relation $u = S_u \hat{u}$ where u is the final controller output. Assign linguistic variables for the controller output as $\{U_m\}$ for absolute output signal $\hat{u}$, or $\{\Delta U_m\}$ for incremental signal $\Delta\hat{u}$. The index $m = 0, 1, 2, \ldots, M - 1$. The value M denotes the total number of fuzzy states defined for the output fuzzy variable. For each element used in the following structures, the nonlinear function $f(.)$ is used to denote the nonlinear mapping between inputs and output.

Type I: Three-input FLC structure with coupled rules

It is practically difficult to assign linguistic values or terms for the input $\Sigma\hat{e}$ as explained in *Remark* 1. Therefore with a three-input configuration the fuzzy PID controllers are unable to produce an absolute signal. Hence the possible inputs are $\hat{e}$, $\Delta\hat{e}$ and $\Delta^2\hat{e}$, corresponding to an incremental type fuzzy PID controller. Using the rule base notation of [5], Type-I fuzzy PID structure can be expressed by,

$$\underset{i,j,k}{\text{ELSE}} \left[\text{IF } \hat{e} \text{ IS } E_i \text{ AND } \Delta\hat{e} \text{ IS } \Delta E_j \text{ AND } \Delta^2\hat{e} \text{ IS } \Delta^2 E_k \text{ THEN } \Delta\hat{u}_{\text{PID}} \text{ IS } \Delta U_{m,\text{PID}}\right]. \tag{3.5}$$

The final PID control output is produced after taking the cumulative sum of the FLC output as shown in Figure 3.3. The total number of rules required for a complete description of the normalized space is $N_1 \times N_2 \times N_3$. The final controller output can be expressed by,

$$u_{\text{PID}}(n) = S_u \sum_{q=0}^{n} \Delta\hat{u}_{\text{PID}}(q). \tag{3.6}$$

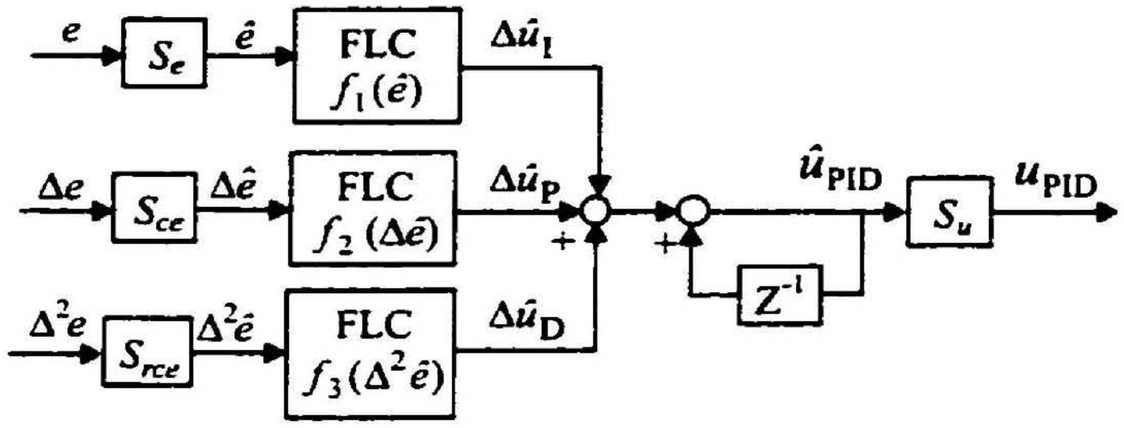


Figure 3.4: Three-input fuzzy PID (Type II)

Type II: Three-input FLC structure with de-coupled rules

The idea of knowledge based de-coupling has been used in [5, 97] to formulate a simple set of rules for GS type fuzzy controllers where the performance based inference is used for fuzzy tuning of conventional PID controllers. This idea is extended for DA type fuzzy PID applications to select three one-input structural elements corresponding to de-coupled rules of Type I for generating the fuzzy incremental control signals. Each incremental PID control action is now represented by a separate set of rules. The knowledge base is expressed by three rules sets.

$$\left.\begin{array}{l} \underset{i}{\text{ELSE}}\ [\text{IF } \hat{e} \text{ IS } E_i \text{ THEN } \Delta\hat{u}_{\text{I}} \text{ IS } \Delta U_{m_1,\text{I}}] \\ \underset{j}{\text{ELSE}}\ [\text{IF } \Delta\hat{e} \text{ IS } \Delta E_j \text{ THEN } \Delta\hat{u}_{\text{P}} \text{ IS } \Delta U_{m_2,\text{P}}] \\ \underset{k}{\text{ELSE}}\ [\text{IF } \Delta^2\hat{e} \text{ IS } \Delta^2 E_k \text{ THEN } \Delta\hat{u}_{\text{D}} \text{ IS } \Delta U_{m_3,\text{D}}] \end{array}\right\}. \tag{3.7}$$

The inference of each rule base is independent and the output constitutes three separate nonlinear functions. The total number of rules required is $N_1 + N_2 + N_3$. The fuzzy PID structure is shown in Figure 3.4. The final control action is given by,

$$u_{\text{PID}}(n) = S_u \sum_{q=0}^{n} \left(\Delta\hat{u}_{\text{P}}(q) + \Delta\hat{u}_{\text{I}}(q) + \Delta\hat{u}_{\text{D}}(q)\right). \tag{3.8}$$

Type III: Two-input FLC structure with coupled rules

By observing the two-input control elements shown in Figure 3.2, the elements having the inputs ($\hat{e}$, $\Delta\hat{e}$) are selected as the useful PID elements for fuzzy control. They correspond to the incremental PI or absolute PD signals. The other two-input control elements shown in

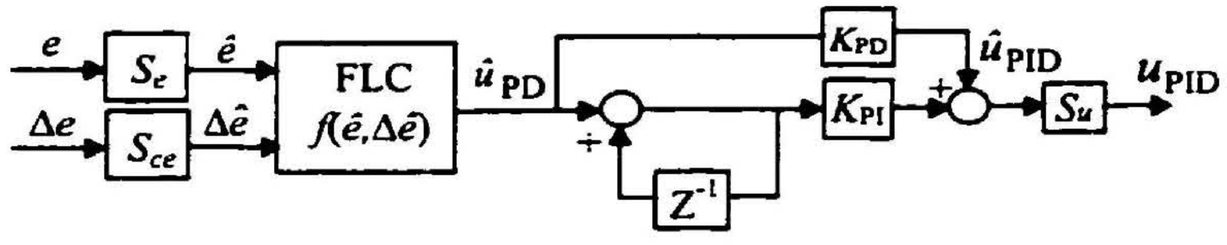


Figure 3.5: Two-input fuzzy PID(Type III)

the Figure 3.2 are eliminated according to the *Remarks* 1 and 2. By combining both PI and PD actions as shown in Figure 3.5, a two-input fuzzy PID controller can be formed. The rule base structure is identical to Mamdani-type fuzzy PI controller. The basic rule base of this conventional type is given by,

$$\underset{i,j}{\text{ELSE}} \ [\text{IF } \hat{e} \text{ IS } E_i \text{ AND } \Delta\hat{e} \text{ IS } \Delta E_j \text{ THEN } \hat{u}_{\text{PD}} \text{ IS } U_{m,\text{PD}}]\,. \tag{3.9}$$

The total number of rules required in this case is equal to $N_1 \times N_2$. With additional gains K_{PD} and K_{PI}, the final PID control signal shown in Figure 3.5 is given by,

$$\begin{aligned} u_{\text{PID}}(n) &= S_u[K_{\text{PI}} \sum_{q=0}^{n} \Delta\hat{u}_{\text{PI}}(q) + K_{\text{PD}}\hat{u}_{\text{PD}}(n)], \\ \text{where} \quad & \Delta\hat{u}_{\text{PI}}(n) = \hat{u}_{\text{PD}}(n). \end{aligned} \tag{3.10}$$

Type IV: Two-input FLC structure with de-coupled rules

The de-coupled structure corresponding to the two-input coupled structure is described next. When the rules are de-coupled from the two input fuzzy PD element, the individual P and D actions can be generated by two one-input elements described by the inputs $\hat{e}$ and $\Delta\hat{e}$ respectively. The two rule bases corresponding to the two one-input control elements are given by,

$$\left.\begin{array}{l} \underset{i}{\text{ELSE}} \ [\text{IF } \hat{e} \text{ IS } E_i \text{ THEN } \hat{u}_{\text{P}} \text{ IS } U_{m_1,\text{P}}] \\ \\ \underset{j}{\text{ELSE}} \ [\text{IF } \Delta\hat{e} \text{ IS } \Delta E_j \text{ THEN } \hat{u}_{\text{D}} \text{ IS } U_{m_2,\text{D}}] \end{array}\right\}. \tag{3.11}$$

From the one-input elements it can be inferred that $\hat{u}_{\text{P}} \equiv \Delta\hat{u}_{\text{I}}$ and by taking the cumulative sum of the fuzzy proportional action, the fuzzy PID structure is derived as shown in Figure 3.6. The total number of rules required in this case is equal to $N_1 + N_2$. With additional

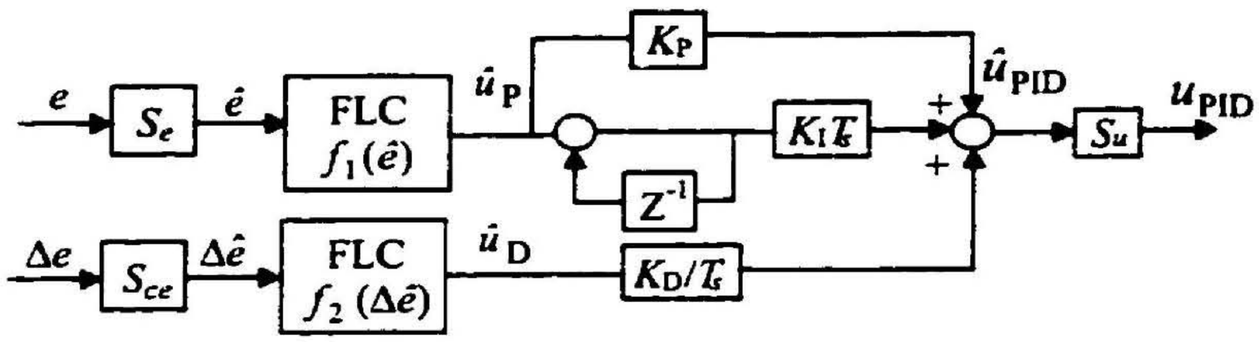


Figure 3.6: Two-input fuzzy PID(Type IV)

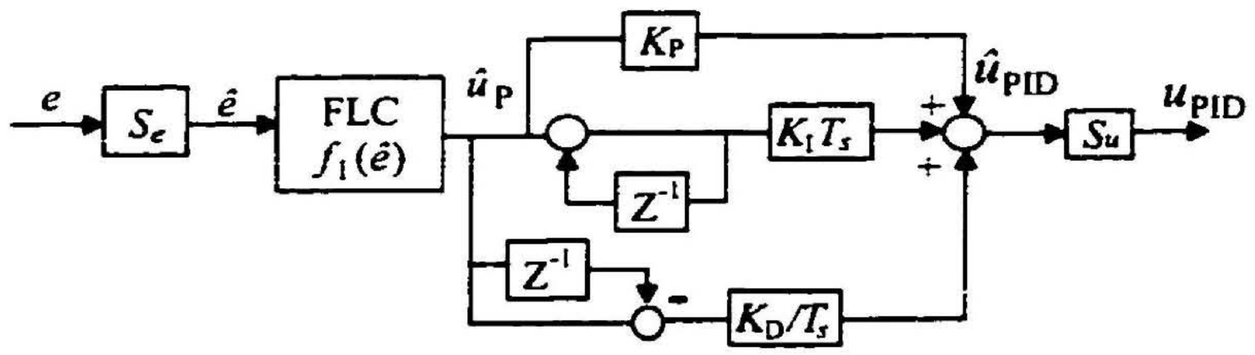


Figure 3.7: One-input fuzzy PID(Type V)

gains K_P, K_I and K_D, the final PID controller action is given by,

$$u_{PID}(n) = S_u[K_P \hat{u}_P(n) + K_I T_s \sum_{q=0}^{n} \hat{u}_P(q) + K_D/T_s \hat{u}_D(n)]. \tag{3.12}$$

Type V: One-input FLC structure with single rule-base

The error signal is the essential and fundamental control component in PID control (*Remark* 1). Therefore by using the input variable $\hat{e}$, a one-input fuzzy PID control system is formed. This is simply the nonlinear mapping of error into fuzzy proportional action. The rule base of the one-input fuzzy proportional control element is given by,

$$\underset{i}{\text{ELSE}} \ [\text{IF } \hat{e} \text{ IS } E_i \text{ THEN } \hat{u}_P \text{ IS } U_m]. \tag{3.13}$$

Similar to the previous case $\hat{u}_P \equiv \Delta\hat{u}_I$ and by assuming the analogy between the proportional and derivative actions as, $\hat{u}_D(n) \equiv \hat{u}_P(n) - \hat{u}_P(n-1)$, the fuzzy PID structure is derived as shown in Figure 3.7. This is the simplest fuzzy PID structure requiring only N_1 rules. With additional gains K_P, K_I and K_D, the final control action is given by,

$$u_{PID}(n) = S_u \left[K_P \hat{u}_P(n) + K_I T_s \sum_{q=0}^{n} \hat{u}_P(q) + K_D/T_s \left(\hat{u}_P(n) - \hat{u}_P(n-1)\right) \right]. \tag{3.14}$$

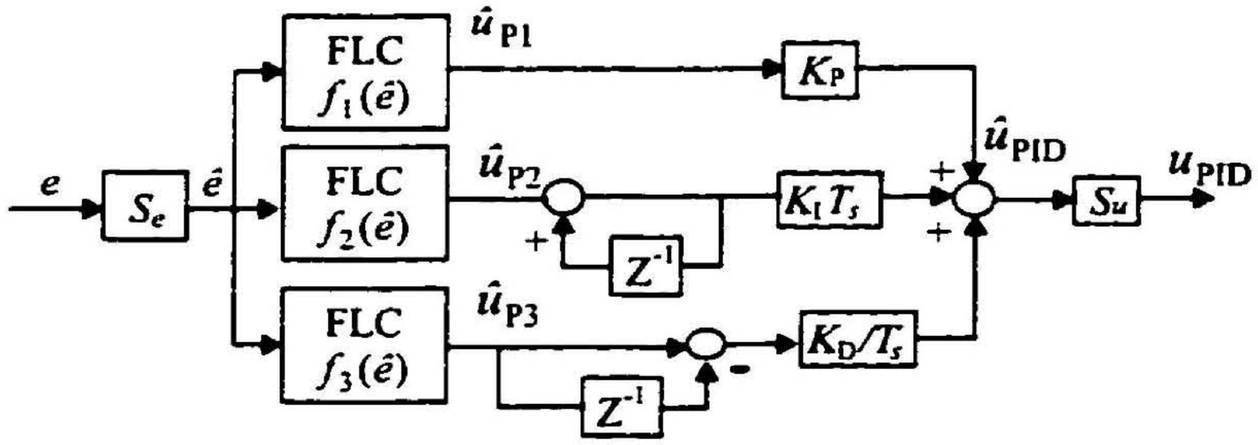


Figure 3.8: One-input fuzzy PID (Type VI)

Type VI: One-input FLC structure with three rule-bases

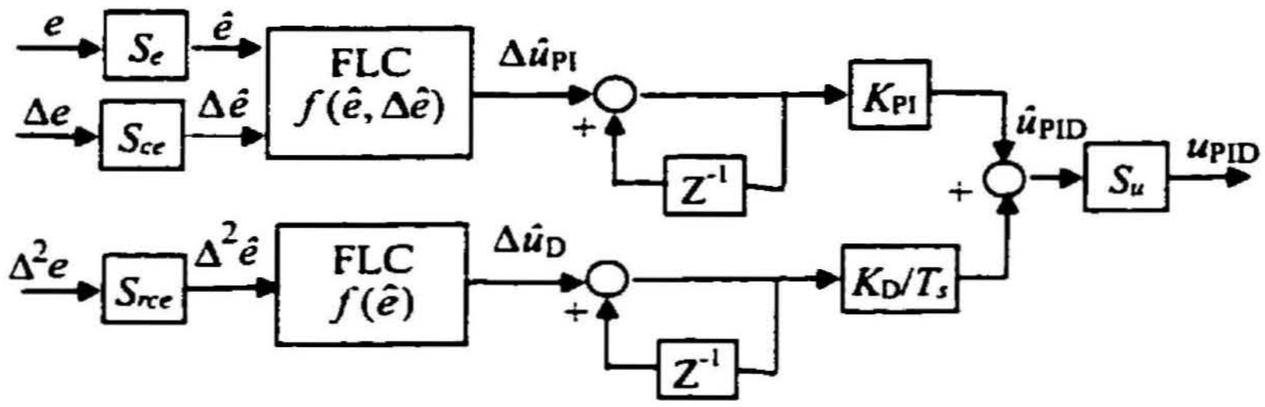
In this structure, three separate rule bases, using only error as the input variable, are used for generating three separate fuzzy proportional actions. The knowledge base parameters can be independently chosen or tuned to produce different non-linearity for the individual PID actions. With respect to the input error variable, three rule bases are defined as,

$$\left.\begin{array}{l} \underset{i}{\text{ELSE}}\ [\text{IF } \hat{e} \text{ IS } E_{1,i} \text{ THEN } \hat{u}_{\text{P}} \text{ IS } U_{m_1,\text{P1}}] \\ \underset{j}{\text{ELSE}}\ [\text{IF } \hat{e} \text{ IS } E_{1,j} \text{ THEN } \hat{u}_{\text{P}} \text{ IS } U_{m_2,\text{P2}}] \\ \underset{k}{\text{ELSE}}\ [\text{IF } \hat{e} \text{ IS } E_{1,k} \text{ THEN } \hat{u}_{\text{P}} \text{ IS } U_{m_3,\text{P3}}] \end{array}\right\}. \tag{3.15}$$

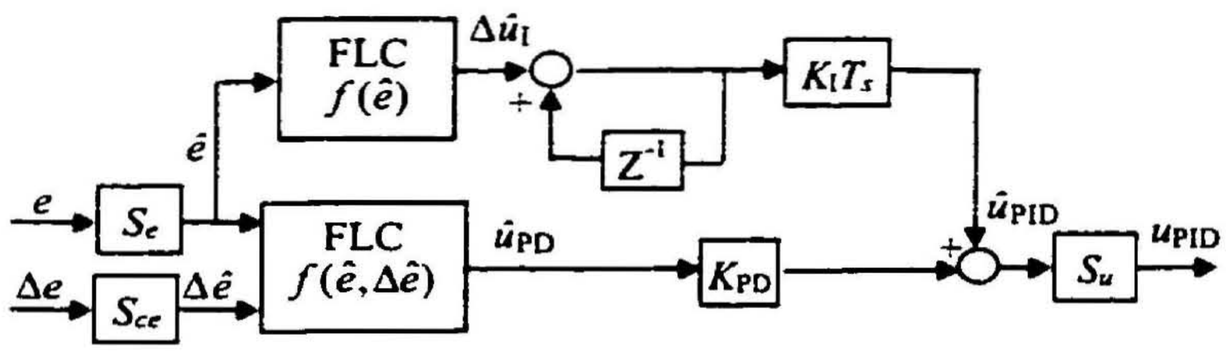
An additional integer suffix is used to separate the three proportional fuzzy rule bases. Using the same basic principle as used in the Type V controller, the integral and derivative actions are now generated using different nonlinear proportional sources as shown in Figure 3.8. The total number of rules required in this case is $N_1 + N_2 + N_3$. Using three additional gains K_{P}, K_{I} and K_{D}, the final control action is given by,

$$u_{\text{PID}}(n) = S_u \left[K_{\text{P}} \hat{u}_{\text{P1}}(n) + K_{\text{I}} T_s \sum_{q=0}^{n} \hat{u}_{\text{P2}}(q) + K_{\text{D}}/T_s \left(\hat{u}_{\text{P3}}(n) - \hat{u}_{\text{P3}}(n-1) \right) \right]. \tag{3.16}$$

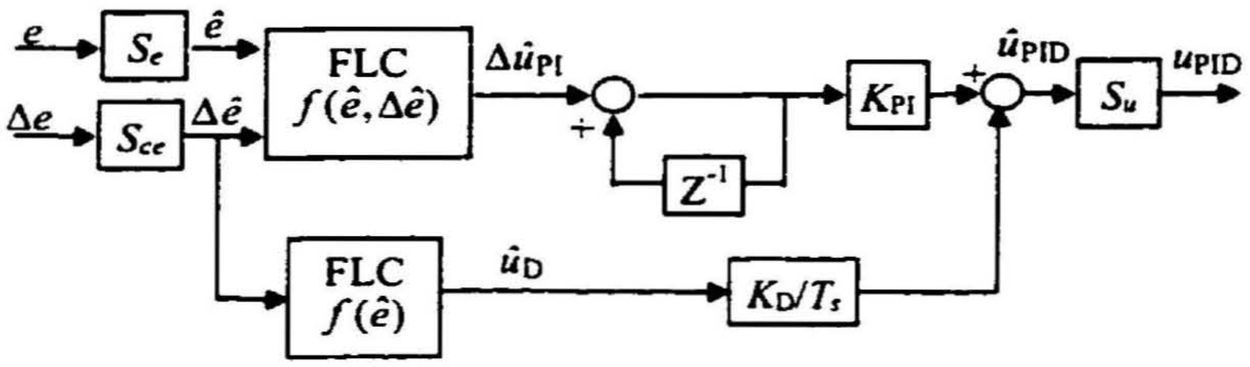
When the three rule bases are identical to each other (identical knowledge base parameters), the structure would be the same as the Type-V structure. Therefore, this is the most general form of the one-input fuzzy PID structure.



(a) Type VII



(b) Type VIII



(c) Type IX

Figure 3.9: Combine two and one-input type fuzzy PID controllers

Type VII-IX: Combined two and one-input fuzzy PID types

Three different structures are identified as shown in the Figure 3.9. The reader can now easily identify the fuzzy rule bases. The knowledge base structure and the final control outputs are not shown to reduce the text of this thesis.

3.2.3 Analogy between fuzzy PID and linear PID

The fuzzy PID controller is essentially a nonlinear PID controller. When the fuzzy PID elements are set to produce linear functions, then the above controllers would either become absolute or incremental type PID controllers. Due to the availability of tuning rules, the

linear PID controllers are easy to tune by changing the linear gains. In some cases manual tuning heuristics are used to adjust the linear gains [5]. As an example, the integral rate is increased for reaching a quick settling time or the derivative gain is increased for reducing the oscillations of the response. The changing of linear gains in a fuzzy controller also has similar effects. The increase of scale factors of an incremental fuzzy integral controller provide similar effect as a linear integral rate. At the same time the linear gains provides overall magnifications to the nonlinear control action. There are two main advantages of fuzzy PID controller that makes the FLC a better performance controller than its linear counterpart. First, the rule base allows the nonlinear control to be imposed on local control points. As an example, the membership parameter changes can bring local changes to the control signal without affecting the other regions of the control. The second benefit is that the normalization of control variables allow the controller to have hard control limits when the inputs fall outside the normalized regions. This property makes the fuzzy controller a sliding mode type controller for robust control. The scale factors adjust the width of the boundary layer in the error state space.

The functional advantages of a fuzzy controller are the non-linearity tuning as opposed to linear tuning in conventional controllers. Therefore the next sections are devoted to deduce the closed-form solutions to the fuzzy inferences for better identification of non-linearity in fuzzy systems.

3.3 Inference Analysis for Fuzzy PID elements

The purpose of this analysis is to provide an analytical base for evaluating and understanding the functional properties of different fuzzy PID controllers. Theoretically it is possible to express the output of any fuzzy system using closed-form expressions. When the rule bases become highly nonlinear and discontinuous then it requires identification of different regions in the error state space to isolate different linear regions to represent the multi-phase solution. In order to establish a systematic analytical procedure, the fuzzy controllers are first analyzed in a linear fashion. For that purpose a Linear-Like Fuzzy Logic Controller (**LLFLC**) is defined. The non-linearity is then systematically added to generate general expressions for Nonlinear Like Fuzzy Logic Controllers (**NLFLC**). For each case the standard

min-max-gravity (**MMG**) reasoning is used as the fuzzy inference. MMG reasoning has been widely used in the past for designing FLC systems and also it has been proven to be a valid inference scheme for fuzzy control [20]. In addition MMG has better nonlinear properties for control than other available reasoning methods. The next chapter evaluates and illustrates the nonlinear properties of MMG inference for fuzzy control.

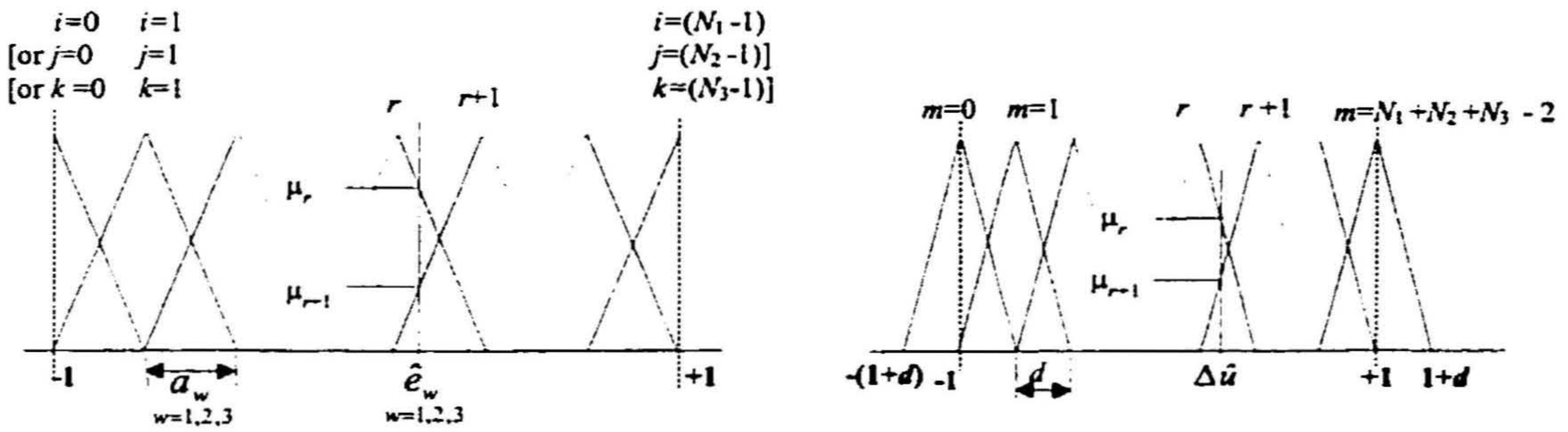
The main difficulty in fuzzy inference analysis is the visualization of the nonlinear output space with respect to the three error variables. Compared to a conventional two-input inference analysis shown in [19, 25], the three-input inference using MMG reasoning requires a minimum of 48 different equations to represent the controller output. To simplify this multi-phase complexity in the solution, a transformation technique is provided and the general solutions are expressed with the least number of nonlinear terms. The two and one-input solutions are obtained as special cases of the three-input solutions. Therefore this analytical frame has the ability to represent the output of any of the fuzzy PID configurations shown in the above section with higher computational efficiency.

3.3.1 Linear-Like Fuzzy Logic Controller (LLFLC) Analysis

Any variable can be linguistically partitioned into linear regions from "Negative Big" to "Positive Big" [22]. When the controller output is partitioned into linearly defined linguistic regions, then such a controller is termed a linear-like fuzzy logic controller. In order to achieve this, the input space is also uniformly partitioned from most negative to most positive. Then, by using linearly defined control rules a linear-like fuzzy control surface is generated. This kind of a linear surface is defined in [114]. The very first work on inference analysis [19] was based on an LLFLC with two inputs.

Let the three error inputs in any order be defined as $\mathbf{e} = \{e_1, e_2, e_3\}^T$. After scaling each of these inputs, let the normalized input error vector at any time instant be given by $\hat{\mathbf{e}} = \{\hat{e}_1, \hat{e}_2, \hat{e}_3\}^T$. Again the sampling instant has been removed for convenience. For each of these inputs use the same fuzzy sub-sets defined in the Section 3.2.2, and consider symmetrical membership functions for all the fuzzy variables. Then the universe of discourse of each input space is uniformly partitioned with 50% overlap of neighboring fuzzy terms.

The fuzzy membership functions are usually of uni-modal. This allows convenient placement of membership functions over the partitioned input/output space. However the mem-



(a) Input membership functions

(b) Output membership functions

Figure 3.10: Membership distributions of the LLFLC system

bership type has minimum effect on the overall non-linearity of the output. Most important parameters are the width (or α-cut width) of the support set and the modal position. The multi-modal membership functions, such as trapezoidal membership functions have equal truth value over a range of values and that usually results dead zones in the controller output. Therefore without loss of generality, triangular membership functions are used as specified below.

1. The universe of discourse of each input variable is defined to be within the continuous range [-1,1] as shown in Figure 3.10a. The total number of linguistic variables used for $\hat{e}_1, \hat{e}_2$ and $\hat{e}_3$ are N_1, N_2 and N_3 respectively and the corresponding distances between two adjacent memberships are given by,

$$a_1 = 2/(N_1 - 1),\ a_2 = 2/(N_2 - 1),\ a_3 = 2/(N_3 - 1). \tag{3.17}$$

The midpoints or the modal positions of membership functions provide the highest degree of membership value $\mu = 1$. Thus, the modal positions of the input membership functions or fuzzy sub-sets can be described by,

$$\hat{\mathbf{e}}_1 = \{\hat{e}_{1,i}\}^T,\ \hat{\mathbf{e}}_2 = \{\hat{e}_{2,j}\}^T,\ \hat{\mathbf{e}}_3 = \{\hat{e}_{1,k}\}^T. \tag{3.18}$$

For the LLFLC,

$$\hat{\mathbf{e}}_1(0) = \hat{\mathbf{e}}_2(0) = \hat{\mathbf{e}}_3(0) = -1 \text{ and } \hat{\mathbf{e}}_1(N_1 - 1) = \hat{\mathbf{e}}_2(N_2 - 1) = \hat{\mathbf{e}}_3(N_3 - 1) = 1.$$

2. The output linguistic variables are defined within the universe of discourse of $[-(1+d).(1+d)]$, where d is the distance between two adjacent output membership functions as shown in Figure 3.10b. The total number of membership functions defined for the output variable $\hat{u}$ is equal to $M = (N_1 + N_2 + N_3 - 2)$. Assign $d = d_3$ for three-input inference. Similarly assign $d = d_2$ and $d = d_1$ for two- and one-input inferences respectively. It can be easily verified that

$$d_3 = 2/(M-1) \text{ or } 1/d_3 = 1/a_1 + 1/a_2 + 1/a_3.$$

The modal positions of the output fuzzy memberships can be described by.

$$\hat{\mathbf{u}} = \{\hat{u}_m\}^T \text{ and } m = 0.1.\ldots.M. \tag{3.19}$$

For the LLFLC, $\hat{\mathbf{u}}(0) = -1$ and $\hat{\mathbf{u}}(M-1) = 1$.

3. Using $(N_1 \times N_2 \times N_3)$ rules, the rule base is defined as,

$$\underset{i.j.k}{\text{ELSE}} \ [\text{IF } \hat{e}_1 \text{ IS } E_{1.i} \text{ AND } \hat{e}_2 \text{ IS } E_{2.j} \text{ AND } \hat{e}_3 \text{ IS } E_{3.k} \text{ THEN } \hat{u} \text{ IS } U_{i+j+k}]. \tag{3.20}$$

Solution algorithm for the three-input LLFLC output

The general solution to the three-input LLFLC is provided with the following seven steps. The derivation of the nonlinear term is detailed in Appendix A.

Step 1 Define error saturation limits to satisfy $\hat{e}_w \in [-1, 1]$.

$$\hat{e}_w = \max(-1, \min(1, S_w e_w)) \tag{3.21}$$

where $w = 1, 2, 3$ and $\{S_w\}$ are the associated scale factors of the error inputs.

Step 2 Define an input index vector and reference error inputs. Let the index vector be,

$$\mathbf{i_a}(w) = \{i_a, j_a, k_a\}^T. \tag{3.22}$$

where i_a, j_a and k_a are the nearest integers given by,

$i_a = \text{round}((1+\hat{e}_1)/a_1)$, $j_a = \text{round}((1+\hat{e}_2)/a_2)$ and $k_a = \text{round}((1+\hat{e}_3)/a_3)$.

With this the reference error inputs are,

$$\left.\begin{aligned} \hat{e}_{1.ia} &= -1 + i_a a_1 \\ \hat{e}_{2.ja} &= -1 + j_a a_1 \\ \hat{e}_{3.ka} &= -1 + k_a a_1 \end{aligned}\right\}. \tag{3.23}$$

Step 3 Define normalized incremental input vectors.

Normalized incremental input vector and normalized absolute incremental input vector are respectively given by,

$$\delta\hat{\mathbf{x}} = \{\delta x_{1,ia}/a_1, \delta x_{2,ja}/a_2, \delta x_{3,ka}/a_3\}^T. \tag{3.24}$$

$$\delta\hat{\mathbf{x}}_a = \{|\delta x_{1,ia}| /a_1, |\delta x_{2,ja}| /a_2, |\delta x_{3,ka}| /a_3\}^T. \tag{3.25}$$

The incremental values are,

$$\delta x_{1,ia} = (\hat{e}_1 - \hat{e}_{1,ia}). \delta x_{2,ja} = (\hat{e}_2 - \hat{e}_{2,ja}). \delta x_{3,ka} = (\hat{e}_3 - \hat{e}_{3,ka}), \tag{3.26}$$

$$(\delta x_{1,ia}/a_1) \ .(\delta x_{2,ja}/a_2) \ .(\delta x_{3,ka}/a_3) \ \in [-0.5, 0.5].$$

Step 4 Perform input transformation.

1. Compute the transformed absolute incremental input vector $\{m_w\}^T$ and identify the corresponding incremental vector positions w_1, w_2 and w_3.

$$\left.\begin{aligned} |m_1| &= \max(\delta\hat{\mathbf{x}}_a) = \delta\hat{\mathbf{x}}_a(w_1) \\ |m_3| &= \min(\delta\hat{\mathbf{x}}_a) = \delta\hat{\mathbf{x}}_a(w_3) \\ |m_2| &= \delta\hat{\mathbf{x}}_a(6 - w_1 - w_3) = \delta\hat{\mathbf{x}}_a(w_2) \end{aligned}\right\}. \tag{3.27}$$

2. Compute the transformed true incremental inputs.

$$m_1 = \delta\hat{\mathbf{x}}(w_1). \ m_2 = \delta\hat{\mathbf{x}}(w_2), \ m_3 = \delta\hat{\mathbf{x}}(w_3). \tag{3.28}$$

3. Redefine the transformed index vector.

$$i_n = \mathbf{i}_a(w_1). \ j_n = \mathbf{i}_a(w_2). \ k_n = \mathbf{i}_a(w_3). \tag{3.29}$$

If $m_1 = m_2 = m_3$ then $w_1 = 1. w_2 = 2. w_3 = 3.$

Step 5 Obtain the nonlinear term.

The nonlinear term is obtained from the Table 3.1. Depending on the sign of the incremental values computed in **Step 4**. m_1 and i_n may be modified as shown in the table. The nonlinear term β_3 is equal to either α_1 or α_2, as given below.

$$\left.\begin{aligned} \alpha_1 &= \tfrac{1}{2}\left(\tfrac{7|m_3|+5|m_2|+3|m_1|-m_3^2-m_2^2-m_1^2}{1+|m_3|+|m_2|+|m_1|-m_1^2}\right) \\ \alpha_2 &= \tfrac{1}{2}\left(\tfrac{5|m_3|-3|m_2|+3|m_1|-m_3^2+m_2^2-m_1^2}{1+|m_3|+|m_2|+|m_1|-m_1^2}\right) \end{aligned}\right\}. \tag{3.30}$$

Table 3.1: Nonlinear term for the three-input LLFLC output

Sign			Modify		Nonlinear Term
m_1	m_2	m_3	m_1	i_n	β_3
+	+	+	m_1	i_n	α_1
-	+	+	$1-\lvert m_1\rvert$	i_n-1	α_1
+	-	+	m_1	i_n	α_2
-	-	+	$1-\lvert m_1\rvert$	i_n-1	α_2
+	+	-	$-1+\lvert m_1\rvert$	i_n+1	$-\alpha_2$
-	+	-	m_1	i_n	α_2
+	-	-	$-1+\lvert m_1\rvert$	i_n+1	$-\alpha_1$
-	-	-	m_1	i_n	$-\alpha_1$

Step 6 Reassign the modified index values to the input index vector.

$$i_a = \text{modified}(i_n).\ j_a = j_n,\ \ k_a = k_n. \tag{3.31}$$

Step 7 Compute the LLFLC output $\hat{u}$.

$$\hat{u} = \hat{u}_{ia+ja+ka} + d_3\beta_3. \tag{3.32}$$

where the reference modal position is

$$\hat{u}_{ia+ja+ka} = -1 + (i_a + j_a + k_a)d_3.$$

Using Table 3.1 and equations (3.23) and (3.26), the general LLFLC output $\hat{u}$ can be decomposed into two parts: a linear controller output ($\hat{u}_{\text{L3}}$) and a nonlinear controller output ($\hat{u}_{\text{NL3}}$).

$$\left.\begin{aligned} \hat{u} &= \hat{u}_{\text{L3}} + \hat{u}_{\text{NL3}} \\ \hat{u}_{\text{L3}} &= (\hat{e}_1/a_1 + \hat{e}_2/a_2 + \hat{e}_3/a_3)\ \ d_3 \\ \hat{u}_{\text{NL3}} &= (\beta_3 - \delta x_{1,ia}/a_1 - \delta x_{2,ja}/a_2 - \delta x_{3,ka}/a_3)\, d_3 \end{aligned}\right\}. \tag{3.33}$$

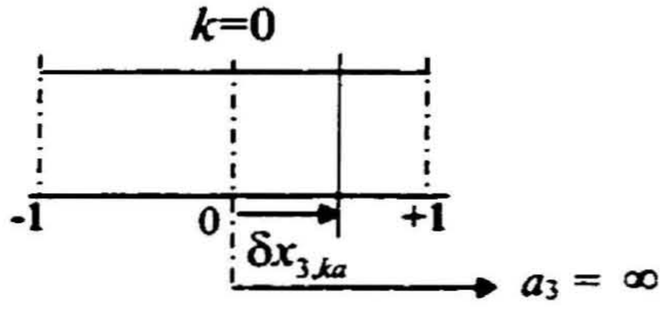


Figure 3.11: Input fuzzy variable with a single fuzzy set

Simplification for the two-input LLFLC output solution

When only two inputs are considered, the third variable can have only a single fuzzy set "*Any*" for any crisp input value. Therefore, the total number of fuzzy sets is equal to one and one can assign this for the redundant input variable. Assume this variable is $\hat{e}_3$ and $N_3 = 1$. From (3.17), $a_3 = 2/(N_3 - 1) = \infty$. The triangular membership function defined for the single linguistic variable will now have an infinitely long support set as shown in Figure 3.11. The fuzzy membership function will be a horizontal line with a unit grade of membership height. The modal position of the single fuzzy set becomes $\hat{e}_{3.ka} = 0$ with $k_a = 0$. Also, any normalized incremental input value measured from this modal position becomes $\lim_{a_3 \to 0}(\delta x_{3.ka}/a_3) = 0$. Thus for any input conditions the $\min(\delta \hat{\mathbf{x}}_\mathbf{a}) = 0$ which implies $m_3 = 0$.

The two-input rule base for generating the LLFLC surface can now be described by $(N_1 \times N_2)$ linear rules.

$$\underset{i,j}{\text{ELSE}} \ [\text{IF } \hat{e}_1 \text{ IS } E_{1,i} \text{ AND } \hat{e}_2 \text{ IS } E_{2,j} \text{THEN } \hat{u} \text{ IS } U_{i+j}] \,. \tag{3.34}$$

The modal spacing of output membership functions ($d = d_2$) is given by, $1/d_2 = 1/a_1 + 1/a_2$. Since now there are only two input variables, the eight cases in Table 3.1 reduce to four cases and α_2 is eliminated. For a two-input fuzzy controller, **Steps 1-7** are used while equating one of the input variables to zero. Taking the special case for α_1 when $m_3 = 0$ the corresponding nonlinear term (β_2) is shown in Table 3.2. With the modified terms the nonlinear term is equal to θ, where,

$$\theta = (\alpha_1)_{m_3=0} = \frac{1}{2}\left(\frac{5|m_2| + 3|m_1| - m_2^2 - m_1^2}{1 + |m_2| + |m_1| - m_1^2}\right). \tag{3.35}$$

Table 3.2: Nonlinear term for the two-input LLFLC output

Sign		Modify		Nonlinear Term
m_1	m_2	m_1	i_n	β_2
+	+	m_1	i_n	θ
-	+	$1-\lvert m_1\rvert$	i_n-1	θ
+	-	$-1+\lvert m_1\rvert$	i_n+1	$-\theta$
-	-	m_1	i_n	$-\theta$

The LLFLC output is given by,

$$\begin{aligned} \hat{u} &= \hat{u}_{ia+ja} + d_2\beta_2, \qquad (3.36) \\ \text{where } \hat{u}_{ia+ja} &= -1 + (i_a + j_a)d_2. \end{aligned}$$

Similar to the three-input case, the general output expression for the two-input LLFLC output can be obtained as the sum of linear ($\hat{u}_{\mathrm{L2}}$) and nonlinear ($\hat{u}_{\mathrm{NL2}}$) controller outputs.

$$\left.\begin{aligned} \hat{u} &= \hat{u}_{\mathrm{L2}} + \hat{u}_{\mathrm{NL2}} \\ \hat{u}_{\mathrm{L2}} &= (\hat{e}_1/a_1 + \hat{e}_2/a_2)\, d_2 \\ \hat{u}_{\mathrm{NL2}} &= (\beta_2 - \delta x_{1,i}/a_1 - \delta x_{2,j}/a_2)\, d_2 \end{aligned}\right\}. \qquad (3.37)$$

Simplification for the one-input LLFLC output solution

Similar to the two-input case, the second and third variables can now be assigned single fuzzy sets. Therefore both $a_3, a_2 \to \infty$ and the system simplifies to a one-dimensional problem. The corresponding LLFLC rule base structure can be represented by N_1 rules as,

$$\underset{i}{\mathrm{ELSE}}\ [\text{IF } \hat{e}_1 \text{ IS } E_{1,i} \text{ THEN } \hat{u} \text{ IS } U_i]\,. \qquad (3.38)$$

Allowing $k_a = j_a = 0$ and $(\delta x_{3,ka}/a_3) = (\delta x_{2,ja}/a_2) = 0$ for any (e_2, e_3) we can take the special case for α_1 when $m_3 = m_2 = 0$. The corresponding nonlinear term β_1 and its values are shown in Table 3.3. The term ϕ is given by,

$$\phi = (\alpha_1)_{m_3=m_2=0} = \frac{1}{2}\left(\frac{3|m_1| - m_1^2}{1 + |m_1| - m_1^2}\right). \qquad (3.39)$$

Table 3.3: Nonlinear term for the one-input LLFLC output

Sign	Nonlinear Term
m_1	β_1
+	φ
-	$-\varphi$

The one-input LLFLC output is given by,

$$\hat{u} = \hat{u}_{ia} + \beta_1 d_1, \tag{3.40}$$

where for a SISO LLFLC system $d = d_1 = a_1$ and $\hat{u}_{ia} = -1 + i_a d_1$. Similar to the two cases above, the general solution for a one-input LLFLC output can be expressed as the sum of linear ($\hat{u}_{\mathrm{L1}}$) and nonlinear ($\hat{u}_{\mathrm{NL1}}$) controller outputs given by,

$$\left.\begin{aligned} \hat{u} &= \hat{u}_{\mathrm{L1}} + \hat{u}_{\mathrm{NL1}} \\ \hat{u}_{\mathrm{L1}} &= \hat{e}_1 \\ \hat{u}_{\mathrm{NL1}} &= (\beta_1 - \delta x_{1.ia}/a_1)\, d_1 \end{aligned}\right\}. \tag{3.41}$$

Some properties of the LLFLC output

The LLFLC output is always within the range [-1,1]. When the normalized variables are quantized into half of the modal distance, the output has a linear function. This linear relation is given by the linear output $\hat{u}_{\mathrm{L}}$. This property has been described by the multi level relay [21]. The nonlinear output is bounded where,

$$|\hat{u}_{\mathrm{NL3}}| \leq 0.3233 d_3,\ |\hat{u}_{\mathrm{NL2}}| \leq 0.1588 d_2,\ |\hat{u}_{\mathrm{NL1}}| \leq 0.0414 d_1.$$

This bounded property can be obtained from the maximum/minimum nonlinear outputs and occurs when all the normalized incremental inputs are equal to each other with $m_1 = m_2 = m_3 = \pm 0.1972$. The positive and negative values correspond to maximum and minimum heights of $\hat{u}_{\mathrm{NL}}$. Also, the nonlinear term is only a function of the incremental input values. This indicates that the non-linearity is local and the LLFLC is globally linear. This satisfies the purpose of a linear-like or fuzzy linear definition. Since the maximum and minimum heights are directly proportional to the base width of the output membership functions,

the nonlinear term $\hat{u}_{\mathrm{NL}}$ has a diminishing effect with the number of rules. Therefore with sufficiently larger number of rules the LLFLC output will almost becomes a linear controller [21].

3.3.2 Nonlinear-Like Fuzzy Logic Controller (NLFLC)

When any of the knowledge base parameters in the LLFLC is changed or altered from its basic definition. then such a controller is called a nonlinear like fuzzy logic controller. An NLFLC can be generated with non-uniform partitioning of fuzzy variables or/and nonlinear rules or/and non-symmetrical membership functions. In this section three particular NLFLC systems are established for obtaining general solutions. The main purpose of the fuzzy control is to provide non-linearity to the control policy with respect to the error state variables. The alteration of knowledge base parameters (including rules) intends to provide different non-linearity to this control plane. The following remarks are made in devising such a FLC particularly for control objectives.

***Remark* 3.3** In general the alteration of rules by changing the whole context is hard and also difficult to visualize. In other words the changing of the absolute meaning of a rule is a difficult programming problem. In [47, 91], the so called rule alterations are performed by changing the entries in the output look-up table and the real meaning is implicit. Then the alteration of other parameters such as membership parameters is most convenient for obtaining different non-linearity than rules.

***Remark* 3.4** The output fuzzy variables should have term sets (or values) in such a way that the context of the fuzzy variables should be proportional to the projected distance measured from the switching line (or surface) in the error state space [114]. The line or surface is given by $\sum \hat{e}_w = 0$. This implies that the rules should have monotonic characteristics. The linear rules satisfy this condition. Secondly, any non-linearity of the controller output can be obtained by changing the partitioning point of the universe of discourse.

***Remark* 3.5** The final issue is the preserving of properties of fuzzy rule bases for control namely. *rule completeness*, *rule consistency*. *rule continuity* and *rule interaction* [15]. A clear description of these properties are described in [15, 35]. The full description

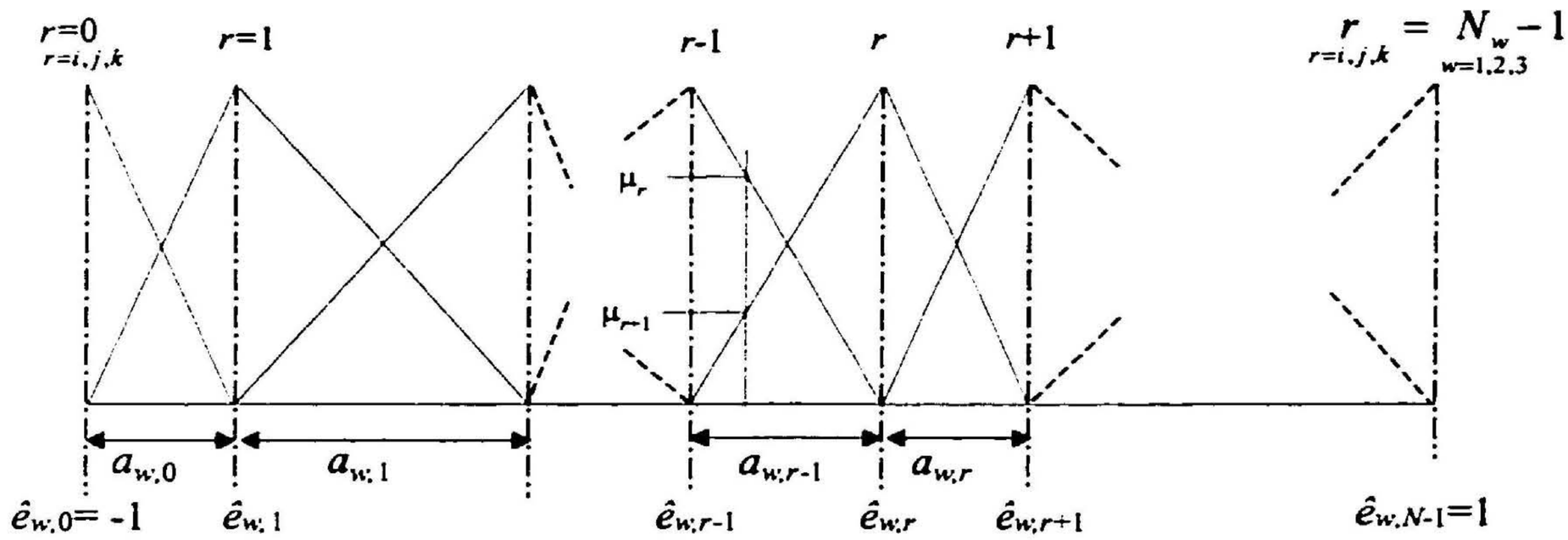


Figure 3.12: Input membership distributions of NLFLC-I and II systems

of rules with a rule matrix of size ($N_1 \times N_2 \times N_3$) and with overlapping neighboring input membership functions automatically preserves the rule completeness. The linear fuzzy rules remove the repetition of rules with the same antecedents and therefore rule consistency is achieved. The monotonic nature of consequent membership distribution generally provides the rule continuity (Remark 3.4). Finally the min-max-gravity reasoning avoids rule interactions [15].

NLFLC-I: Using non-uniform partitioning of input variables

The universes of discourse of all inputs are non-uniformly partitioned as shown in Figure 3.12. The algebraic sum of truth-values (membership values) for a given crisp input state is maintained at 1 as shown in Figure 3.12. Therefore the adjacent memberships are 50% overlapped to satisfy the latter condition. The output is uniformly distributed as in the LLFLC (Figure 3.10b). Consider the linear rule bases (equations (3.20). (3.34) and (3.38)). Now the modal positions of the input fuzzy variables given in (3.18) have non-uniform separations between the adjacent values. Therefore define the membership separations as,

$$\{a_{1,i}\} = \{\hat{e}_{1,i+1} - \hat{e}_{1,i}\}. \quad \{a_{2,j}\} = \{\hat{e}_{2,j+1} - \hat{e}_{2,j}\}. \quad \{a_{3,k}\} = \{\hat{e}_{3,k+1} - \hat{e}_{3,k}\}. \tag{3.42}$$

Solution algorithm for the NLFLC-I output

The algorithm follows that developed for LLFLC but with the modifications to Steps 2 and 3 given below. For obtaining the two or one input solutions to the general NLFLC-I, the LLFLC simplification is applied with modified steps, as given below.

Modified Step 2 Define input index vector and reference error inputs.

1. Compute the differential vectors.

$$\Delta\hat{\mathbf{e}}_{1a} = \{|\hat{e}_{1,i} - \hat{e}_1|\}^T,\ \Delta\hat{\mathbf{e}}_{2a} = \{|\hat{e}_{2,j} - \hat{e}_2|\}^T,\ \Delta\hat{\mathbf{e}}_{3a} = \{|\hat{e}_{3,k} - \hat{e}_3|\}^T. \quad (3.43)$$

2. Find the element having the minimum value and the corresponding vector positions.

$$\Delta\hat{\mathbf{e}}_{1a}(i_a) = \min(\Delta\hat{\mathbf{e}}_{1a}),\ \Delta\hat{\mathbf{e}}_{2a}(j_a) = \min(\Delta\hat{\mathbf{e}}_{2a}),\ \Delta\hat{\mathbf{e}}_{3a}(k_a) = \min(\Delta\hat{\mathbf{e}}_{3a}). \quad (3.44)$$

3. Obtain reference modal positions.

$$\hat{e}_{1,ia} = \hat{\mathbf{e}}_1(i_a),\ \hat{e}_{2,ja} = \hat{\mathbf{e}}_2(j_a),\ \hat{e}_{3,ka} = \hat{\mathbf{e}}_3(k_a). \quad (3.45)$$

Modified Step 3 Define the normalized incremental input vectors.

Normalized incremental input vector and normalized absolute incremental input vector are respectively given by.

$$\delta\hat{\mathbf{x}} = \{\delta x_{1,ia}/a_{1,ia}, \delta x_{2,ja}/a_{2,ja}, \delta x_{3,ka}/a_{3,ka}\}^T \quad (3.46)$$

$$\delta\hat{\mathbf{x}}_a = \{|\delta x_{1,ia}|\ /a_{1,ia}, |\delta x_{2,ja}|\ /a_{2,ja}, |\delta x_{3,ka}|\ /a_{3,ka}\}^T. \quad (3.47)$$

The incremental values are given by the equations (3.26).

While using the NLFLC-I steps the solution can be decomposed to the linear and nonlinear parts given below.

$$\left.\begin{aligned} \hat{u} &= \hat{u}_{\mathrm{L3}} + \hat{u}_{\mathrm{NL3}} \\ \hat{u}_{\mathrm{L3}} &= (\hat{e}_1/a_{1,ia} + \hat{e}_2/a_{2,ja} + \hat{e}_3/a_{3,ka})\ d_3 \\ \hat{u}_{\mathrm{NL3}} &= (\beta_3 - \delta x_{1,ia}/a_{1,ia} - \delta x_{2,ja}/a_{2,ja} - \delta x_{3,ka}/a_{3,ka})\, d_3 \end{aligned}\right\}. \quad (3.48)$$

Some Properties of the NLFLC-I

For two and one-input solutions, force the appropriate terms to zero. Since the input modal separations are non-uniform, the linear output in (3.48) has a variable linearity, where as in the LLFLC, the linear controller has global linear properties. Therefore the properties of the LLFLC are locally applicable for the NLFLC-I and the linear controller is piece wise linear about the error state space. The degree of global non-linearity is determined by the non-linearity of the input partitioning.

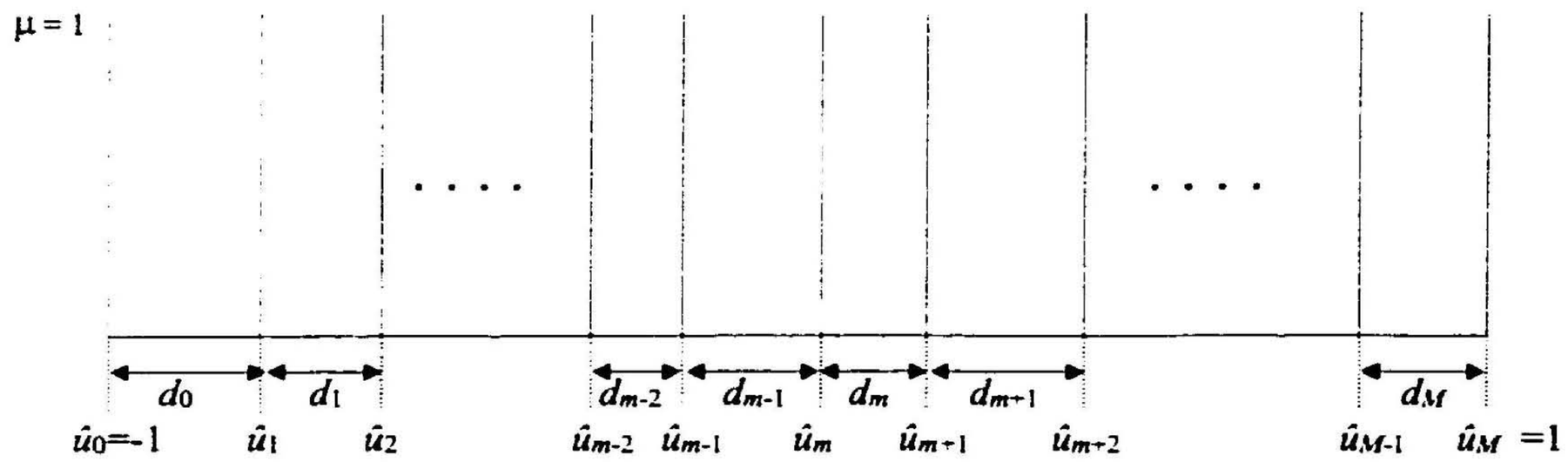


Figure 3.13: The distribution of output fuzzy singletons in the NLFLC-II system

NLFLC-II: Using non-uniform partitions to all fuzzy variables

In addition to the nonlinear partitioning of input space in NLFLC-I, the output space is also non-uniformly partitioned. Therefore the same input membership distributions shown in Figure 3.12 are assumed for this controller. The 50% overlap of membership partitions always guarantees the rule completeness. However, the overlapping condition is not a requirement for output membership functions for preserving any rule base properties. If the support sets are defined in such a way that the membership modal positions (or the crisp values that have the highest degree of confidence in the term sets) are monotonically placed along the output universe of discourse, then the rule base continuity can be preserved (*Remark* 3.4). Under these conditions, there is an enormous number of arrangements of support sets of the output fuzzy variables. Also the general solution requires a larger number of expressions to represent many ranges of input variables. The next controller (NLFLC-III) demonstrates this difficulty when using one-input inference with the least number of membership functions. To simplify this complexity fuzzy singletons are assigned for the output variable as shown in Figure 3.13. Therefore the vector defining the output fuzzy singleton positions are defined with,

$$\hat{\mathrm{u}}_s = \{\hat{u}_m\}^T \text{ where } m = i + j + k. \tag{3.49}$$

Define the non-uniform membership spacing as.

$$d_m = \hat{u}_{m+1} - \hat{u}_m. \tag{3.50}$$

Table 3.4: Nonlinear term for the three-input NLFLC-II output

Sign			Modify		Nonlinear Term
m_1	m_2	m_3	m_1	i_n	β_{n3}
+	+	+	m_1	i_n	α_{n1}
-	+	+	$1-\lvert m_1\rvert$	i_n-1	α_{n1}
+	-	+	m_1	i_n	α_{n2}
-	-	+	$1-\lvert m_1\rvert$	i_n-1	α_{n2}
+	+	-	$-1+\lvert m_1\rvert$	i_n+1	$-\alpha_{n3}$
-	+	-	m_1	i_n	$-\alpha_{n3}$
+	-	-	$-1+\lvert m_1\rvert$	i_n+1	$-\alpha_{n4}$
-	-	-	m_1	i_n	$-\alpha_{n4}$

Simplification for three-input conditions

Although the rules are similar to LLFLC systems, the consequent of the rules can now be expressed in a functional form as in TSK style, for representing the knowledge base. Hence three-input based rules are redefined as follows:

$$\underset{i,j,k}{\text{ELSE}} \ [\text{IF } \hat{e}_1 \text{ IS } E_{1,i} \text{ AND } \hat{e}_2 \text{ IS } E_{2,j} \text{ AND } \hat{e}_3 \text{ IS } E_{3,k} \text{ THEN } \hat{u} = \hat{u}_{i+j+k}] \,. \tag{3.51}$$

The same procedure for NLFLC-I can be used with modified Steps 5 and 7. Step 5 needs modifications for the nonlinear term to accommodate the new fuzzy output conditions and thus Step 7 is modified. Due to the non-symmetric positioning of the neighboring term sets, the symmetrical condition of Table 3.1 no longer exists.

Modified Step 5 Obtain the nonlinear term.

The nonlinear terms in Table 3.1 are modified and shown in Table 3.4. The modified expressions are given below.

Table 3.5: Nonlinear term for the two-input NLFLC-II output

Sign		Modify		Nonlinear Term
m_1	m_2	m_1	i_n	β_{n2}
+	+	m_1	i_n	θ_{n1}
-	+	$1-\lvert m_1\rvert$	i_n-1	θ_{n1}
+	-	$-1+\lvert m_1\rvert$	i_n+1	$-\theta_{n2}$
-	-	m_1	i_n	$-\theta_{n2}$

$$\left.\begin{aligned}
\alpha_{n1} &= \frac{d_{m_a}|m_1|+(d_{m_a}+d_{m_a-1})|m_2|+(d_{m_a}+d_{m_a-1}+d_{m_a-2})|m_3|}{1+|m_2|+|m_3|} \\
\alpha_{n2} &= \frac{d_{m_a}|m_1|-d_{m_a-1}|m_2|+(d_{m_a}+d_{m_a-1})|m_3|}{1+|m_2|+|m_3|} \\
\alpha_{n3} &= \frac{d_{m_a-1}|m_1|-d_{m_a-2}|m_2|+(d_{m_a-1}+d_{m_a-2})|m_3|}{1+|m_2|+|m_3|} \\
\alpha_{n4} &= \frac{d_{m_a-1}|m_1|+(d_{m_a-1}+d_{m_a-2})|m_2|+(d_{m_a-1}+d_{m_a-2}+d_{m_a-3})|m_3|}{1+|m_2|+|m_3|}
\end{aligned}\right\}. \tag{3.52}$$

where $m_a = i_a + j_a + k_a$.

Modified Step 7 Compute NLFLC-II output.

$$\hat{u} = \hat{u}_{m_a} + \beta_{n3}. \tag{3.53}$$

Simplification for two-input conditions

Similar to two-input LLFLC simplification, force $k_a = 0$. $m_3 = 0$ and the simplified rule base is given by,

$$\underset{i,j}{\text{ELSE}}\ \left[\text{IF } \hat{e}_1 \text{ IS } E_{1,i} \text{ AND } \hat{e}_2 \text{ IS } E_{2,j} \text{THEN } \hat{u} = \hat{u}_{i+j}\right]. \tag{3.54}$$

The modified nonlinear term (β_{n2}) is shown in Table 3.5 and the nonlinear terms are expressed as.

$$\left.\begin{aligned}
\theta_{n1} = (\alpha_{n1})_{m_3=0} &= \frac{d_{m_a}|m_1|+(d_{m_a}+d_{m_a+1})|m_2|}{1+|m_2|} \\
\theta_{n2} = (\alpha_{n4})_{m_3=0} &= \frac{d_{m_a-1}|m_1|+(d_{m_a-1}+d_{m_a-2})|m_2|}{1+|m_2|}
\end{aligned}\right\}. \tag{3.55}$$

where $m_a = i_a + j_a + k_a$. The simplified NLFLC-II output is given by,

$$\hat{u} = \hat{u}_{m_a} + \beta_{n2}. \tag{3.56}$$

Table 3.6: Nonlinear term for the one-input NLFLC-II output

Sign	Nonlinear Term
m_1	β_{n1}
+	ϕ_{n1}
-	$-\phi_{n2}$

Simplification for one-input conditions

The rule base is given by,

$$\underset{i}{\text{ELSE}} \ [\text{IF } \hat{e}_1 \text{ IS } E_{1,i} \text{ THEN } \hat{u} = \hat{u}_i] \,. \tag{3.57}$$

The modified nonlinear term (β_{n1}) is shown in Table 3.6 and the nonlinear terms are expressed as,

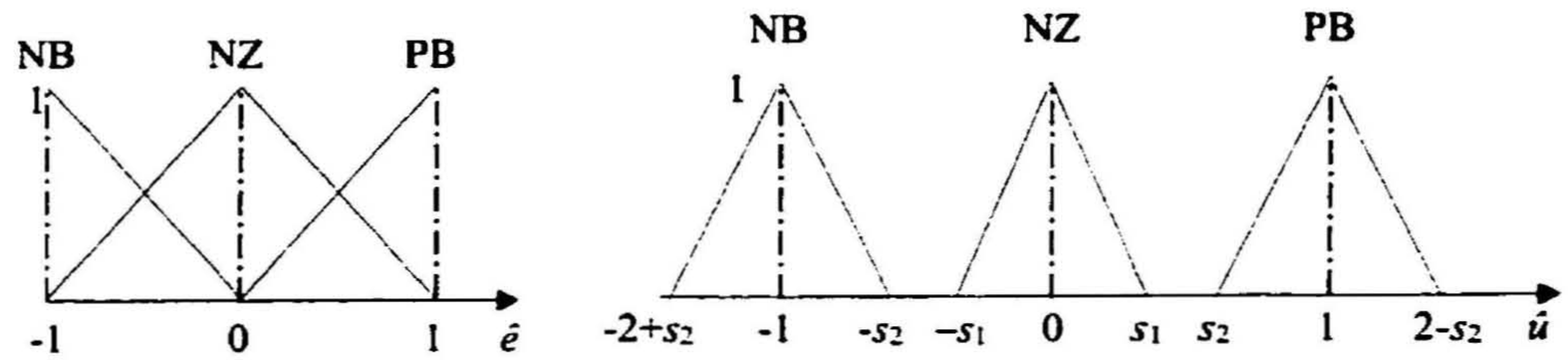
$$\left.\begin{aligned} \phi_{n1} &= (\alpha_{n1})_{m_3=m_2=0} = d_{m_a}|m_1| \\ \phi_{n2} &= (\alpha_{n4})_{m_3=m_2=0} = d_{m_a-1}|m_1| \end{aligned}\right\} . \tag{3.58}$$

Some properties of the NLFLC-II system

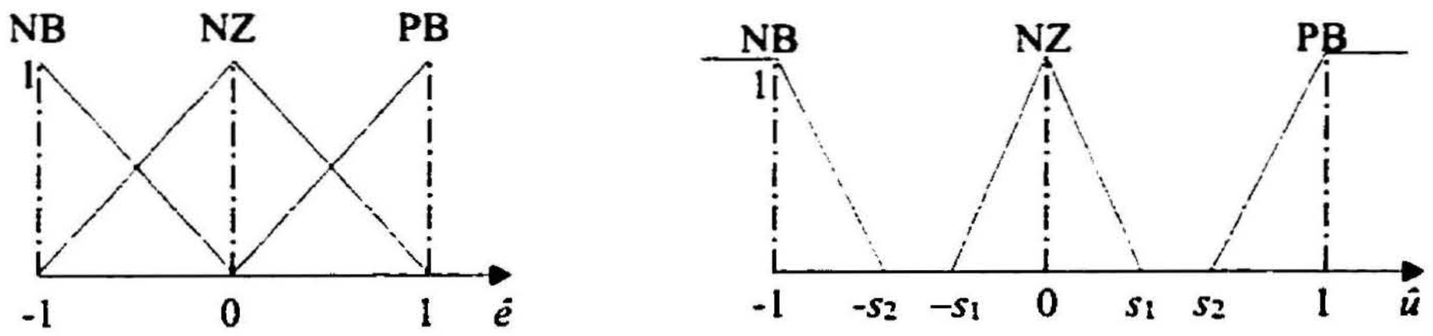
Overall the NLFLC-II controller has the capability to produce more non linearity in the general three-input controller output than the NLFLC-I controller. However, the fuzzy singleton simplification reduces the local non-linearity properties of the surface. It can be clearly seen that the nonlinear term corresponding to the one-input conditions (ϕ_{n1} and ϕ_{n2}) becomes linear and the local non-linearity diminishes. This makes the one-input NLFLC-II controller a piece-wise linear controller.

NLFLC-III: Three-ruled one-input NLFLC system

Under the one-input conditions, the NLFLC-II has limitations to produce better nonlinear control. This is mainly due to the limitations in defining output membership functions. This can be overcome by having more general types of membership functions for the consequent. In this case only three-rules are considered. More evaluations for this controller are explained in the next chapter. However if more rules are employed, the NLFLC-III type controller requires a very large number of expressions to represent the output behaviour. Assume the



a. Membership distribution for NLFLC-IIIA



b. Membership distribution for NLFLC-IIIB

Figure 3.14: Input/Output membership distributions for NLFLC-III Systems

input variable is $\hat{e}_1$ and consider three symmetrical rules as shown below.

$$\left.\begin{array}{l} \text{If } \hat{e}_1 \text{ is NB then } \hat{u} \text{ is NB} \\ \text{If } \hat{e}_1 \text{ is NZ then } \hat{u} \text{ is NZ} \\ \text{If } \hat{e}_1 \text{ is PB then } \hat{u} \text{ is PB} \end{array}\right\}. \tag{3.59}$$

The fuzzy variables are labelled with NB, NZ and PB to represent "*Negative Big*", "*Near Zero*" and "*Positive Big*" respectively. The modal positions of all fuzzy subsets are fixed as $\hat{\mathbf{e}}_1 = \hat{\mathbf{u}} = \{-1, 0, 1\}^T$ for satisfying the boundary conditions of the control system. As there are only three fuzzy variables, the input fuzzy sets are arranged with 50% overlap as shown in Figure 3.14. For obtaining different non-linearity the widths of output triangular membership functions are varied. Two different arrangements have been identified for output fuzzy variables as shown in Figure 3.14. In both cases the two membership parameters (s_1 and s_2) related to the width of output fuzzy variables have been chosen to get a different level of non-linearity in the output. The two arrangements are labelled as NLFLC-IIIA and NLFLC-IIIB as shown in the figure. In the first case (Figure 3.14a) the triangular memberships for the output are defined over the universe of discourse $[-(2+s_2), (2+s_2)]$. This arrangement allows the fuzzy output to be fully normalized within the range [-1,1]. In the second case (Figure

3.14b) the universe of discourse is strictly applied over [-1,1]. In both cases a symmetrical membership function for the "*Near Zero*" output fuzzy variable is assumed for achieving the zero conditions (or steady state conditions) of the controller. The membership parameters are constrained to vary within the following ranges:

$s_1 \in (0, 1]$ for keeping the NZ fuzzy sub-set triangular about the zero and

$s_2 \in [-s_1, 1)$ for obtaining unique expressions for the fuzzy output.

The derivation of the fuzzy outputs for NLFLC-III systems is given in the Appendix B. The solution has two main cases: non-overlapping or overlapping output memberships. Again due to the discontinuity of the min-max functions, the overlapping case has three different expressions for satisfying three different ranges for the input. The following intermediate variables are defined for easy representation of the solutions:

$$s_d = \frac{s_1 - s_2}{1 + s_1 - s_2}, \; z_2 = s_1 - s_2 \text{ and } z_2 = 1 - s_2. \tag{3.60}$$

Output solution for the NLFLC-IIIA

Case I (Non-overlapping): $s_1 \leq s_2$

$$\hat{u} = \frac{\hat{e}_1(2 - |\hat{e}_1|)z_2}{(1 - \hat{e}_1^2)s_1 + |\hat{e}_1|(1 - |\hat{e}_1|)z_2}. \tag{3.61-I}$$

Case II (Overlapping): $s_1 > s_2$ AND

IIa. $[(s_1 - s_2) \leq 1 \text{ AND } 0 \leq |\hat{e}_1| < s_d] \text{ OR } [(s_1 - s_2) \geq 1 \text{ AND } 0 \leq |\hat{e}_1| < 0.5]$.

$$\hat{u} = \frac{|\hat{e}_1|}{3}\left[\frac{(z_2^2 - s_1^2)(\hat{e}_1^2 - 3|\hat{e}_1| + 3) + 3(1 + 2z_2 - z_2|\hat{e}_1|)}{2s_1 + 2(1 - s_1 + z_2)|\hat{e}_1| - (s_2 + z_2)\hat{e}_1^2}\right]. \tag{3.61-IIa}$$

IIb. $[(s_1 - s_2) \leq 1 \text{ AND } s_d \leq |\hat{e}_1| < (1 - s_d)]$.

$$\hat{u} = \frac{\hat{e}_1}{3|\hat{e}_1|}\left[\frac{6z_2(2|\hat{e}_1| - \hat{e}_1^2) - z_1 s_d\,(3s_1 + z_1 - s_d(3s_1 + 2z_2))}{2s_1(1 - e_1^2) + 2z_2(2|\hat{e}_1| - \hat{e}_1^2) - z_1 s_d}\right]. \tag{3.61-IIb}$$

IIc. $[(s_1 - s_2) \leq 1 \text{ AND } (1 - s_d) \leq |\hat{e}_1| < 1] \text{ OR } [(s_1 - s_2) \geq 1 \text{ AND } 0.5 \leq |\hat{e}_1| < 1]$

$$\hat{u} = \frac{\hat{e}_1}{3|\hat{e}_1|}\left[\frac{(z_2^2 - s_1^2)(1 - |\hat{e}_1^3|) + 3(1 - |\hat{e}_1|) - 3z_2(\hat{e}_1^2 - 4|\hat{e}_1| + 1)}{s_1(1 - \hat{e}_1^2) - z_2(1 - 4|\hat{e}_1| + \hat{e}_1^2) + 2(1 - |\hat{e}_1|)}\right]. \tag{3.61-IIc}$$

Output solution for the NLFLC-IIIB

Case I (Non-overlapping): $s_1 \leq s_2$

$$\hat{u} = \frac{\hat{e}_1 z_2}{3}\left[\frac{3(1+s_2) - 3s_2|\hat{e}_1| - z_2\hat{e}_1^2}{2s_1 + 2z_2|\hat{e}_1| - (2s_1+z_2)\hat{e}_1^2}\right]. \tag{3.62-I}$$

Case II (Overlapping): $s_1 > s_2$ AND

IIa. $[(s_1 - s_2) \leq 1$ AND $0 \leq |\hat{e}_1| < s_d]$ OR $[(s_1 - s_2) \geq 1$ AND $0 \leq |\hat{e}_1| < 0.5]$

$$\hat{u} = \frac{\hat{e}_1}{3}\left[\frac{3(1-s_1^2) + s_1^2(3|\hat{e}_1| - \hat{e}_1^2)}{2s_1 + 2(1-s_1)|\hat{e}_1| - s_1\hat{e}_1^2}\right]. \tag{3.62-IIa}$$

IIb. $[(s_1 - s_2) \leq 1$ AND $s_d \leq |\hat{e}_1| < (1 - s_d)]$

$$\hat{u} = \frac{\hat{e}_1}{3|\hat{e}_1|}\left[\frac{(z_1 s_d(3s_d s_1 + 2s_d z_2 - 4s_1 + s_2) + |\hat{e}_1|z_2(3 + 3s_2 - 3|\hat{e}_1|s_2 - z_2\hat{e}_1^2)}{2s_1 - z_1 s_d + 2z_2|\hat{e}_1| - (2s_1+z_2)\hat{e}_1^2}\right].$$

(3.62-IIb)

IIc. $[(s_1 - s_2) \leq 1$ AND $(1 - s_d) \leq |\hat{e}_1| < 1]$ OR $[(s_1 - s_2) \geq 1$ AND $0.5 \leq |\hat{e}_1| < 1]$

$$\hat{u} = \frac{\hat{e}_1}{3|\hat{e}_1|}\left[\frac{(3s_2 + z_2^2 - s_1^2) - 3s_2^2|\hat{e}_1| + 3z_2^2\hat{e}_1^2 + (s_1^2 - 2z_2^2)|\hat{e}_1|^3}{(1 + s_1 + s_2) - 2s_2|\hat{e}_1| - s_1\hat{e}_1^2}\right]. \tag{3.62-IIc}$$

Equations (3.61) and (3.62) show the full description of the fuzzy output of the NLFLC-III system. It is now easy to visualize the complexity of the solution when this controller is added with more input variables. However the NLFLC-III system can be readily used for the rule de-coupled PID structures shown in Section 3.2.

Some properties of NLFLC-III

Due to the symmetric nature of the output partitioning about the zero, the NLFLC-III outputs are negatively symmetrical. The symmetrical conditions for the NZ fuzzy variable is a necessary condition for the steady state conditions. i.e. when $\hat{e}_1 = 0$ then $\hat{u} = 0$. The wider membership definition of the NLFLC-IIIA makes the output curve to be fully normalized within [-1,1]. The non-symmetry of the NB and PB fuzzy variables in the NLFLC-IIIB always makes the normalized output to be within $[-(2+s_2)/3, (2+s_2)/3]$.

3.4 Gain Analysis

This section identifies the PID gain terms related to fuzzy controller structures developed in Section 3.2. In general. for a given fuzzy PID structure, the controller design can be considered as *a two-level tuning* problem [48]. The first level of tuning deals with the knowledge base parameters that have direct influence on the nonlinear control action of the normalized crisp output. This is *non-linearity tuning* and is usually achieved by varying rules, membership functions. input and output partitions. The second level deals with the linear gains that provide overall magnifications to the control action in the error space. This is *linear tuning* and is usually achieved by changing linear gains in the PID structure including normalizing and de-normalizing scale factors. When both of these levels are combined, the fuzzy controller tuning becomes a higher dimensional problem and usually takes considerable time to search for the optimum set of tuning parameters. The two-level design identification allows the multi-parameter complexity to be split into two parts. Also, this identification enables one to identify the most effective tuning parameters for efficient design. This is one of the main objectives of this thesis work. The analysis identifies two types of gains. The Apparent Nonlinear Gains (**ANG**) are defined for the first level of tuning and the Apparent Linear Gains (**ALG**) are defined for the second level of tuning . The true PID gains are functions of both ANG and ALG gains. The explicit representations of the ANG terms are difficult for coupled rule bases. This requires the dissociation of the error terms from the general output. For simplicity first an LLFLC with the simplest form is considered for identifying the ANG terms. Therefore first the outputs of a simplest LLFLC are expressed in terms of error terms for identifying ANG terms for different structures.

3.4.1 LLFLC Based PID Outputs

First define the simplest type of a fuzzy controller based on LLFLC in order to obtain concise expressions relating the error variables. In this simplest LLFLC structure each input is assigned two uniformly distributed membership functions as shown in the Figure 3.15. Therefore $a_1 = a_2 = a_3 = 2$. Since the membership index values i, j and k have only two values. 0 and 1. first consider the positive incremental inputs measured from 0 index positions. For any given input error vector $\{\hat{e}, \Delta\hat{e}, \Delta^2\hat{e}\}$ the incremental values are:

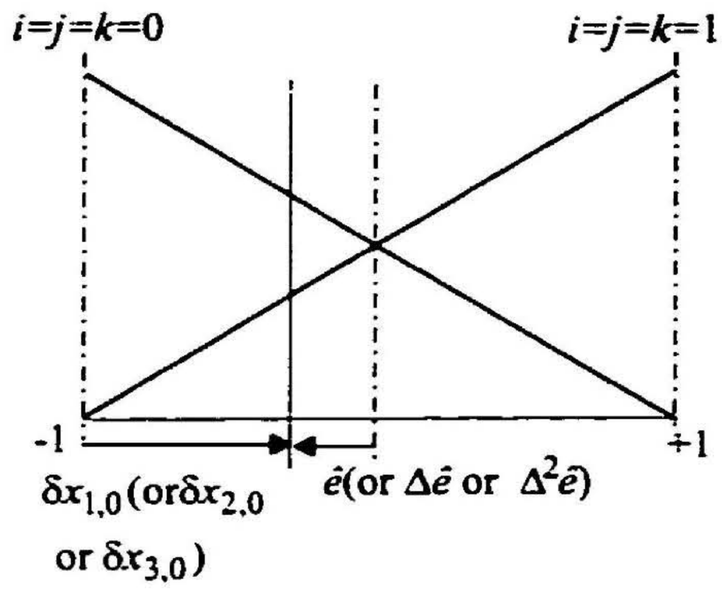


Figure 3.15: Two uniformly distributed memberships for the simplest LLFLC

$$\delta x_{1,0} = (1+\hat{e})/2,\ \delta x_{2,0} = (1+\Delta\hat{e})/2,\ \delta x_{3,0} = (1+\Delta^2\hat{e})/2.$$

Considering the PID structural elements in Figure 3.2, the LLFLC outputs are deduced.

For three-input elements

As it was shown in Section 3.3.2 and Table 3.1, the value of the non-linear term changes with respect to the relative difference between the normalized input variables. In order to express the outputs in terms of the actual input terms (without transformation) and also to aid the PID gain analysis, a single case is considered. Assume $(\delta x_{1,0}/a_1) > (\delta x_{2,0}/a_2) > (\delta x_{3,0}/a_3) \geq 0$. Using the general solution in (3.32) the corresponding output is given by,

$$\Delta\hat{u}_{\text{PID}} = \frac{1}{3}\left(\frac{4\hat{e} - 2\hat{e}^2 + 2\Delta\hat{e} - (\Delta\hat{e})^2 + 6\Delta^2\hat{e} - (\Delta^2\hat{e})^2}{9 - \hat{e}^2 - 2\Delta\hat{e} + 2\Delta^2\hat{e}}\right). \tag{3.63}$$

Equation (3.63) can be rewritten in the dissociated form as,

$$\Delta\hat{u}_{\text{PID}} = \frac{(2-\Delta\hat{e})\Delta\hat{e}}{3P} + \frac{(2-\hat{e})2\hat{e}}{3P} + \frac{(6-\Delta^2\hat{e})\Delta^2\hat{e}}{3P}.$$

$$\text{where } P = 9 - \hat{e}^2 - 2\Delta\hat{e} + 2\Delta^2\hat{e}.$$

Assuming the dissociated form $\Delta\hat{u}_{\text{PID}} = \Delta\hat{u}_{\text{P}}^{\text{PID}} + \Delta\hat{u}_{\text{I}}^{\text{PID}} + \Delta\hat{u}_{\text{D}}^{\text{PID}}$ define,

$$\begin{aligned} \Delta\hat{u}_{\text{P}}^{\text{PID}} &= \frac{(2-\Delta\hat{e})\Delta\hat{e}}{3P}, \\ \Delta\hat{u}_{\text{I}}^{\text{PID}} &= \frac{(2-\hat{e})2\hat{e}}{3P}, \\ \Delta\hat{u}_{\text{D}}^{\text{PID}} &= \frac{(6-\Delta^2\hat{e})\Delta^2\hat{e}}{3P}. \end{aligned} \tag{3.64}$$

The superscript PID is used to show the inference source.

For two-input elements

Considering the case, $(\delta x_{1,0}/a_1) > (\delta x_{2,0}/a_2) \geq 0$ and using the two-input solution in (3.36), it can be shown,

$$\hat{u}_{\mathrm{PD}} = \Delta\hat{u}_{\mathrm{PI}} = \frac{1}{2}\left(\frac{4\hat{e} + \hat{e}^2 + 4\Delta\hat{e} - (\Delta\hat{e})^2}{7 + 2\Delta\hat{e} - \hat{e}^2}\right). \tag{3.65}$$

Equation (3.65) can be rewritten in the dissociated form as.

$$\hat{u}_{\mathrm{PD}} = \Delta\hat{u}_{\mathrm{PI}} = \tfrac{\hat{e}(4+\hat{e})}{2Q} + \tfrac{\Delta\hat{e}(4-\Delta e)}{2Q},$$
$$\text{where } Q = 7 + 2\Delta\hat{e} - \hat{e}^2.$$

Assuming the dissociated forms $\Delta\hat{u}_{\mathrm{PD}} = \hat{u}_{\mathrm{P}}^{\mathrm{PD}} + \hat{u}_{\mathrm{D}}^{\mathrm{PD}}$ and $\hat{u}_{\mathrm{PI}} = \Delta\hat{u}_{\mathrm{P}}^{\mathrm{PI}} + \Delta\hat{u}_{\mathrm{I}}^{\mathrm{PI}}$ define,

$$\begin{aligned} \hat{u}_{\mathrm{P}}^{\mathrm{PD}} &= \Delta\hat{u}_{\mathrm{I}}^{\mathrm{PI}} = \hat{e}(4+\hat{e})/(2Q), \text{ and} \\ \hat{u}_{\mathrm{D}}^{\mathrm{PD}} &= \Delta\hat{u}_{\mathrm{P}}^{\mathrm{PI}} = \Delta\hat{e}(4-\Delta e)/(2Q). \end{aligned} \tag{3.66}$$

The superscript PI or PD is used to show the inference source.

For one-input elements

Considering the one-input LLFLC solution in (3.40), the fuzzy outputs for one-input control elements can be expressed by,

$$\left.\begin{aligned} \hat{u}_{\mathrm{P}} = \Delta\hat{u}_{\mathrm{I}} &= 4\hat{e}/(5-\hat{e}^2) \\ \hat{u}_{\mathrm{D}} = \Delta\hat{u}_{\mathrm{P}} &= 4\Delta\hat{e}/(5-(\Delta\hat{e})^2) \\ \Delta\hat{u}_{\mathrm{D}} &= 4\Delta^2\hat{e}/(5-(\Delta^2\hat{e})^2) \end{aligned}\right\}. \tag{3.67}$$

3.4.2 Apparent Nonlinear PID Gains

As explained in Section 3.4.1, the apparent non-linear gains are directly related to the normalized fuzzy output of a PID controller. Therefore the non-linear PID gains are deduced from the normalized output expressions of the fuzzy PID controllers described in Section 3.2. For the dissociated or de-coupled fuzzy PID actions, the ANG terms are defined as follows.

$$\begin{aligned} \hat{K}_{\mathrm{Pa}}(n) &= \hat{u}_{\mathrm{P}}(n)/\hat{e}(n), \\ \hat{K}_{\mathrm{Ia}}(n) &= \hat{u}_{\mathrm{I}}(n)/\sum_{q=0}^{n}\hat{e}(q) \text{ and} \\ \hat{K}_{\mathrm{Da}}(n) &= \hat{u}_{\mathrm{D}}(n)/\Delta\hat{e}(n). \end{aligned} \tag{3.68}$$

Table 3.7: ANG terms of different fuzzy PID structures

Type	ANG-Proportional $\hat{K}_{\mathrm{Pa}}(n)$	ANG-Integral $\hat{K}_{\mathrm{Ia}}(n)$	ANG - Derivative $\hat{K}_{\mathrm{Da}}(n)$
I	$\frac{1}{\hat{e}(n)}\sum_{q=0}^{n}\left(\frac{(2-\Delta\hat{e}(q))\Delta\hat{e}(q)}{3P(q)}\right)$	$\frac{1}{\sum_{q=0}^{n}\hat{e}(q)}\sum_{q=0}^{n}\left(\frac{2(2-\hat{e}(q))\hat{e}(q)}{3P(q)}\right)$	$\frac{1}{\Delta\hat{e}(n)}\sum_{q=0}^{n}\left(\frac{(6-\Delta^2\hat{e}(q))\Delta^2\hat{e}(q)}{3P(q)}\right)$
II	$\frac{1}{\hat{e}(n)}\sum_{q=0}^{n}\left(\frac{4\Delta\hat{e}(q)}{5-(\Delta\hat{e}(q))^2}\right)$	$\frac{1}{\sum_{q=0}^{n}\hat{e}(q)}\sum_{q=0}^{n}\left(\frac{4\hat{e}(q)}{5-(\hat{e}(q))^2}\right)$	$\frac{1}{\Delta\hat{e}(n)}\sum_{q=0}^{n}\left(\frac{4\Delta^2\hat{e}(q)}{5-(\Delta^2\hat{e}(q))^2}\right)$
III	$\hat{K}_{\mathrm{Pa}}^{\mathrm{PD}}=\frac{(4+\hat{e}(n))}{2Q(n)}$ $\hat{K}_{\mathrm{Pa}}^{\mathrm{PI}}=\frac{1}{\hat{e}(n)}\sum_{q=0}^{n}\left(\frac{(4-\Delta\hat{e}(q))\Delta\hat{e}(q)}{2Q(q)}\right)$	$\frac{1}{\sum_{q=0}^{n}\hat{e}(q)}\sum_{q=0}^{n}\left(\frac{(4+\hat{e}(q))\hat{e}(q)}{2Q(q)}\right)$	$\frac{(4-\Delta\hat{e}(n))}{2Q(n)}$
IV	$\frac{4}{5-\hat{e}(n)^2}$	$\frac{1}{\sum_{q=0}^{n}\hat{e}(q)}\sum_{q=0}^{n}\left(\frac{4\hat{e}(q)}{5-\hat{e}(q)^2}\right)$	$\frac{4}{5-(\Delta\hat{e}(n))^2}$
V	$\frac{4}{5-\hat{e}(n)^2}$	$\frac{1}{\sum_{q=0}^{n}\hat{e}(q)}\sum_{q=0}^{n}\left(\frac{4\hat{e}(q)}{5-\hat{e}(q)^2}\right)$	$\frac{4(5+\hat{e}(n)^2)}{5-\hat{e}(n)^2}$
VI	$\frac{\hat{u}_{\mathrm{Pn}}(\hat{e}(n),\mathbf{x}_1)}{\hat{e}(n)}$	$\frac{1}{\sum_{q=0}^{n}\hat{e}(q)}\sum_{q=0}^{n}\hat{u}_{\mathrm{Pn}}(\hat{e}(q),\mathbf{x}_2)$	$\frac{d\hat{u}_{\mathrm{Pn}}(\hat{e}(n),\mathbf{x}_3)}{d\hat{e}(n)}$

where. $\hat{K}_{\mathrm{Pa}}$. $\hat{K}_{\mathrm{Ia}}$ and $\hat{K}_{\mathrm{Da}}$ are the apparent nonlinear proportional. integral and derivative gains respectively. The ANG terms obtained for structure types I-VI are listed in Table 3.7 and the steps followed are described below.

ANG for Type I

Using the dissociated form given in (3.64). the normalized control action corresponding to (3.6) can be described by.

$$\hat{u}_{\mathrm{PID}}(n)=\sum_{q=0}^{n}\Delta\hat{u}_{\mathrm{P}}^{\mathrm{PID}}(q)+\sum_{q=0}^{n}\Delta\hat{u}_{\mathrm{I}}^{\mathrm{PID}}(q)+\sum_{q=0}^{n}\Delta\hat{u}_{\mathrm{D}}^{\mathrm{PID}}(q). \tag{3.69}$$

The equivalent form with ANG terms is:

$$\hat{u}_{\mathrm{PID}}(n)=\hat{K}_{\mathrm{Pa}}(n)\hat{e}(n)+\hat{K}_{\mathrm{Ia}}(n)\sum_{q=0}^{n}\hat{e}(q)+\hat{K}_{\mathrm{Da}}\Delta\hat{e}(n). \tag{3.70}$$

Substituting terms given in (3.64) into (3.69), the ANG terms that correspond to the arrangement in (3.70) are thus obtained.

ANG for Type II

The normalized control action corresponding to (3.8) can be described by,

$$\hat{u}_{\mathrm{PID}}(n) = \sum_{q=0}^{n} \hat{u}_{\mathrm{P}}(q) + \sum_{q=0}^{n} \hat{u}_{\mathrm{I}}(q) + \sum_{q=0}^{n} \hat{u}_{\mathrm{D}}(q). \tag{3.71}$$

The ANG expressions for (3.71) and (3.70) are identical. Substituting one-input element outputs in (3.67) to (3.71), the ANG terms that correspond to the arrangement in (3.70) are thus obtained.

ANG for Type III

Using the dissociated form given in (3.66), the normalized output corresponding to equation (3.10) in the dissociated form can be described by,

$$\begin{aligned} \hat{u}_{\mathrm{PID}}(n) &= K_{\mathrm{PD}} \left(\hat{u}_{\mathrm{P}}^{\mathrm{PD}}(n) + \hat{u}_{\mathrm{D}}^{\mathrm{PD}}(n) \right) \\ &\quad + K_{\mathrm{PI}} \left(\sum_{q=0}^{n} \Delta \hat{u}_{\mathrm{P}}^{\mathrm{PI}}(q) + \sum_{q=0}^{n} \Delta \hat{u}_{\mathrm{I}}^{\mathrm{PI}}(q) \right). \end{aligned} \tag{3.72}$$

The equivalent form with ANG terms is:

$$\begin{aligned} \hat{u}_{\mathrm{PID}}(n) &= K_{\mathrm{PD}} \left(\hat{K}_{\mathrm{Pa}}^{\mathrm{PD}}(n) \hat{e}(n) + \hat{K}_{\mathrm{Da}}(n) \Delta \hat{e}(n) \right) \\ &\quad + K_{\mathrm{PI}} \left(\hat{K}_{\mathrm{Pa}}^{\mathrm{PI}}(n) \hat{e}(n) + \hat{K}_{\mathrm{Ia}}(n) \sum_{q=0}^{n} \hat{e}(q) \right). \end{aligned} \tag{3.73}$$

Using the dissociated outputs for two-input element in equation (3.66) and substituting into equation (3.72), the ANG terms that correspond to the arrangement in (3.73) are thus obtained.

ANG for Type IV

In this de-coupled rule structure, the normalized output corresponding to (3.12) can be expressed by,

$$\hat{u}_{\mathrm{PID}}(n) = K_{\mathrm{P}} \hat{u}_{\mathrm{P}}(n) + K_{\mathrm{I}} T_s \sum_{q=0}^{n} \hat{u}_{\mathrm{P}}(q) + K_{\mathrm{D}}/T_s \hat{u}_{\mathrm{D}}(n). \tag{3.74}$$

The equivalent form with ANG terms is,

$$\begin{aligned}\hat{u}_{\mathrm{PID}}(n) &= K_{\mathrm{P}}\hat{K}_{\mathrm{Pa}}(n)\hat{e}(n) \\ &\quad + K_{\mathrm{I}}T_s\hat{K}_{\mathrm{Ia}}(n)\sum_{q=0}^{n}\hat{e}(q) + (K_{\mathrm{D}}/T_s)\hat{K}_{\mathrm{Da}}(n)\Delta\hat{e}(n).\end{aligned} \tag{3.75}$$

Substituting the one-input element outputs in (3.67) into (3.74), the ANG terms that correspond to the arrangement in (3.75) are thus obtained.

ANG for Type V

In this one-input structure, the normalized output corresponding to equation (3.14) can be expressed by,

$$\hat{u}_{\mathrm{PID}}(n) = K_{\mathrm{P}}\hat{u}_{\mathrm{P}}(n) + K_{\mathrm{I}}T_s\sum_{q=0}^{n}\hat{u}_{\mathrm{P}}(q) + K_{\mathrm{D}}/T_s\left(\hat{u}_{\mathrm{P}}(n) - \hat{u}_{\mathrm{P}}(n-1)\right). \tag{3.76}$$

The ANG expressions for (3.76) and (3.75) are identical. Substituting one-input element output for $\hat{u}_{\mathrm{P}}$ given in (3.67) into (3.76), the ANG terms that correspond to the arrangement in (3.75) are thus obtained. For small sampling time intervals the equivalent nonlinear derivative gain can be further simplified using the relation $\hat{K}_{\mathrm{Da}} = d\hat{u}_{\mathrm{P}}(n)/d\hat{e}(n)$.

ANG for Type VI

In this one-input structure, the normalized output corresponding to equation (3.16) can be expressed by,

$$\hat{u}_{\mathrm{PID}}(n) = K_{\mathrm{P}}\hat{u}_{\mathrm{P}}(n) + K_{\mathrm{I}}T_s\sum_{q=0}^{n}\hat{u}_{\mathrm{P}}(q) + K_{\mathrm{D}}/T_s\left(\hat{u}_{\mathrm{P}}(n) - \hat{u}_{\mathrm{P}}(n-1)\right). \tag{3.77}$$

Since the Type V structure is a special case of type VI, with the simplest LLFLC rule bases, both types are identical. A practical high performance fuzzy controller requires the knowledge base to have a nonlinear-like structure. However, for the normalized proportional controller output to be monotonic with respect to error, the rules must be arranged in the linear form. In order to illustrate this, the solution of the NLFLC-III system is assumed. Let the nonlinear proportional action obtained by the one-input fuzzy mapping be given by,

$$\hat{u}_{\mathrm{P}}(n) = \hat{u}_{\mathrm{Pn}}\left(\hat{e}(n), \mathbf{x}\right). \tag{3.78}$$

where $\{\mathbf{x}\}$ denotes the nonlinear tuning parameters of the one-input fuzzy knowledge base where $\mathbf{x} = \{s_1, s_2\}$. By using different non-linearity tuning vectors, three different proportional actions can be identified as.

$$\hat{u}_{\mathrm{P1}} = \hat{u}_{\mathrm{Pn}}(\hat{e}(n), \mathbf{x}_1)\,,\ \hat{u}_{\mathrm{P2}} = \hat{u}_{\mathrm{Pn}}(\hat{e}(n), \mathbf{x}_2)\,,\ \hat{u}_{\mathrm{P3}} = \hat{u}_{\mathrm{Pn}}(\hat{e}(n), \mathbf{x}_3)\,. \tag{3.79}$$

The ANG expressions for (3.77) and (3.75) are identical. Substituting (3.79) into (3.77), the ANG terms that correspond to the arrangement in (3.75) are thus obtained. Similar to Type V, a small sampling time can be assumed for obtaining the derivative ANG term.

3.4.3 Apparent Linear PID Gains

The second level tuning generally achieves the overall performance of a fuzzy controller. Given a fuzzy system this tuning behaviour is analogous to linear PID tuning. When the fuzzy inferences in the PID structures are allowed to produce linear functions, then the fuzzy PID controllers become perfect linear PID (either incremental or absolute) controllers. When the error terms of such a controller is arranged in the linear PID form given in (3.1) or (3.2), the gains become equivalent to linear PID gains. The linear controller part of the decomposed LLFLC controller ($\hat{u}_{\mathrm{L}}$) has a global linear structure where as in the NLFLC controllers it is local. The NLFLC controllers described in this chapter have been derived from the LLFLC basic structure. Therefore the decomposed linear controller of the LLFLC can be used as the basic equivalent linear form for all the NLFLC systems. Hence without loss of generality the following description is followed for obtaining an Equivalent Linear Controller (**ELC**) for the fuzzy PID systems.

1. Assume the scale factors for error variables given in (3.4).

2. For the general description given in Section 3.2.2, N_1, N_2 and N_3 are the numbers of the fuzzy membership functions assigned for input error variables.

3. Consider the linear rules used in Section 3.3 (*necessary condition*).

4. Assume the input membership functions are arranged according to the LLFLC structure (the actual arrangement may be non-uniform).

5. Without loss of generality consider the fully normalized conditions for the fuzzy output and include the transformation $\hat{u} \equiv \hat{u}/\hat{u}_{\max}$.

6. Consider the linear controller output ($\hat{u}_L$) described in LLFLC outputs.

7. The scale factor for error can be made fixed by using $S_e = 1/e_{\max}$ where the maximum error value ($e_{\max}$) changes as the set point varies.

The above description defines an Equivalent Linear Controller for the entire error space for defining the linear PID gains. The superscript l is used in each case to denote the equivalent linear conditions. Expressions (3.17) are assumed for determining the parameters a_1, a_2 and a_3.

ELC for the three-input element

Consider the linear controller output in (3.33). After substituting the scale factors the ELC output is obtained as

$$\Delta\hat{u}^l_{\text{PID}}(n) = \frac{d_3 S_e}{a_e}e(n) + \frac{d_3 S_{ce}}{a_{ce}}\Delta e(n) + \frac{d_3 S_{rce}}{a_{rce}}\Delta^2 e(n). \tag{3.80}$$

ELC for the two-input element

Considering the linear output in (3.37) the ELC output is given by

$$\hat{u}^l_{\text{PD}}(n) = \Delta\hat{u}^l_{\text{PI}}(n) = \frac{d_2 S_e}{a_e}e(n) + \frac{d_2 S_{ce}}{a_{ce}}\Delta e(n). \tag{3.81}$$

ELC for the one-input element

Considering the linear output in (3.41) the ELC output is given by

$$\left.\begin{aligned} \hat{u}^l_{\text{P}}(n) = \Delta\hat{u}^l_{\text{I}}(n) &= S_e e(n) \\ \hat{u}^l_{\text{D}}(n) = \Delta\hat{u}^l_{\text{P}}(n) &= S_{ce}\Delta e(n) \\ \Delta\hat{u}^l_{\text{D}}(n) &= S_{rce}\Delta^2 e(n) \end{aligned}\right\}. \tag{3.82}$$

The Apparent Linear Gains are the equivalent PID gains when the fuzzy controller is defined as an ELC. Therefore for each PID structure the ELC is identified in the form given by,

$$u^l_{\text{PID}}(n) = K_{\text{Pa}}e(n) + K_{\text{Ia}}\sum_{q=0}^{n} e(q)T_s + K_{\text{Da}}\Delta e(n)/T_s. \tag{3.83}$$

Table 3.8: ALG terms of different fuzzy PID structures

Type	K_{Pa}	K_{Ia}	K_{Da}	Tuning variables
I	$\frac{S_u S_{ce} d_3}{a_{ce}}$	$\frac{S_u S_e d_3}{a_e T_s}$	$\frac{S_u S_{rce} d_3 T_s}{a_{rce}}$	S_u, S_{ce}, S_{rce}
II	$S_u S_{ce}$	$\frac{S_u S_e}{T_s}$	$S_u S_{rce} T_s$	S_u, S_{ce}, S_{rce}
III	$d_2 S_u \left[\frac{S_{ce} K_{PI}}{a_{ce}} + \frac{S_e K_{PD}}{a_e} \right]$	$\frac{S_u S_e d_2 K_{PI}}{a_e T_s}$	$\frac{S_u S_{ce} d_2 K_{Pd} T_s}{a_{ce}}$	S_u, S_{ce}, K_{PD} $(K_{PI} = 1)$
IV	$S_u S_e K_P$	$S_u S_e K_I$	$S_u S_{ce} K_D$	S_u, S_{ce}, K_I $(K_P = K_D = 1)$
V	$S_u S_e K_P$	$S_u S_e K_I$	$S_u S_e K_D$	K_P, K_I, K_D $(S_u = 1)$
VI	$S_u S_e K_P$	$S_u S_e K_I$	$S_u S_e K_D$	K_P, K_I, K_D $(S_u = 1)$

where K_{Pa}, K_{Ia} and K_{Da} are the apparent linear PID gains. Substituting the ELCs in (3.80-3.82) into the output expressions given in (3.6), (3.8), (3.10), (3.12), (3.14) and (3.16) the ALG terms for the PID structure Types I-VI are obtained in the form given in (3.83). The derived ALG terms are tabulated in Table 3.8. Using the same approach the ALG terms of Types VII-IX can be obtained. Similar to the three PID gains the total unknown linear parameters of the fuzzy PID system also can be simplified to three unknown tuning parameters by forcing some redundant terms to unity. The simplified three unknown tuning parameters are shown in the last column of Table 3.8.

3.5 Functional Properties of Fuzzy PID Controllers

In the previous section, fuzzy PID controller design has been identified as a two-level tuning problem. The PID gains related to both of the tuning levels have been defined. The overall performance of a fuzzy PID controller depends on how accurately and effectively these gains have been chosen or adjusted during the design process. As there are many forms of fuzzy PID structures, the superiority of one structure lies in how easily the controller can be

implemented or adjusted for obtaining better performance over the linear PID controller. In most controller designs the exact plant model and changing process dynamics during the process of control are usually not unknown. In those circumstances the non-linearity that is required by the controller also becomes unknown before the implementation. However, past control experience usually helps to identify the unknown gain values. In performance based GS type fuzzy tuners the tuning rules are implemented in de-coupled form [5]. When the conventional PID tuning knowledge is extended for non-linearity tuning of fuzzy controllers, the identification of individual PID actions are possible in a de-coupled form. It is hard to find such control expert rules for coupled PID actions. In that respect the different fuzzy PID controller structures are compared for performance.

The superiority of fuzzy control is mainly due to the non-linearity tuning. A better fuzzy controller should allow maximum versatility and flexibility in tuning the nonlinear gains for achieving the best performance over linear control. Therefore the functional benefits or behaviours of the PID structures can be compared by the functional properties described below.

3.5.1 Action Association

The basic difficulty in coupled rule bases is the identification of those nonlinear tuning parameters relating the nonlinear PID gains. In all coupled rule structures the output actions are in the associated form. The action association refers to the availability of the three PID actions by a single output expression. The basic dissociation that has been performed for the simplest LLFLC structure is an attempt to identify the individual PID actions in a dissociated form. A similar approach has been employed in [20] to identify the ANG terms of the simplest PI controller using different inference methods. This process is artificial since the algebraic decomposition of nonlinear terms may not show the true representation of the individual PID outputs. Furthermore, if NLFLC systems are considered, the identification of the PID actions in a dissociated form will become a complex mathematical problem and as a result the nonlinear PID gains become non-transparent for independent tuning. The action association is one of major reasons why no satisfactory in-depth analysis has been done in identifying the nonlinear tuning parameters in an explicit form for the most common conventional controllers (Type-I and III).

3.5.2 Input Coupling

From the ANG terms it can be observed that the coupled rules produce coupled nonlinear gains. In the Type I controller the common denominator (P) in the three ANG terms shows the coupled nature of the inputs on individual PID actions. In other words the variation of one error variable affects all the ANG terms in the PID control. This behaviour may be an advantage for generalized damping [115], where the effect of error derivatives on the nonlinear gain terms can provide more damping. The disadvantage is that the proportional and integral actions are unnecessarily complicated by the effect of damping causing a more sluggish kind of response. For example, when the process is responding slowly, the coupled action of error rates tends to produce low equivalent gain for the proportional ANG value. This can be numerically verified by comparing the maximum proportional action when all the error derivatives are forced to zero. This is one of the reasons why in [29] the conventional fuzzy PI controller (Type III) was unable to perform better than an optimally designed linear PI controller. The input coupling sometimes has a negative effect on the overall (or linear) tuning. Although the ALG terms are similar to three PID gains, they are functions of the normalizing or scale factors. The change of an ALG value affects the overall magnification of each of the three ANG terms. For example. increasing the proportional ALG term in the Type I controller causes overall magnification or reduction of each of the three ANG terms.

3.5.3 Gain Dependency

Gain dependency occurs when one fuzzy action is generated by using another action as in Types III-V controllers. For example, in the Type III controller fuzzy PI is generated from fuzzy PD while in the Type V controller fuzzy (I) and fuzzy (D) actions are produced from fuzzy (P) action. This can be mathematically described as follows.

1. Dependency between the two-input coupled PI and PD controllers (Type III)

   The dependency between the PI and PD controllers is given by, $\hat{u}_{\mathrm{PI}}(n) = \sum_{q=0}^{n} \hat{u}_{\mathrm{PD}}(q)$. Replacing the normalized terms with ANG terms, the gain dependency can be expressed by,

$$\hat{K}_{\mathrm{Pa}}^{\mathrm{PI}}(n)\hat{e}(n) + \hat{K}_{\mathrm{Ia}}(n)\sum_{q=0}^{n}\hat{e}(q) = \sum_{q=0}^{n}\left(\hat{K}_{\mathrm{Pa}}^{\mathrm{PD}}(q)\hat{e}(q) + \hat{K}_{\mathrm{Da}}(q)\Delta\hat{e}(q)\right). \tag{3.84}$$

2. Dependency between P and I controllers (Type IV and V)

   The dependency that exists in the Types IV and V controller outputs is given by, $\hat{u}_{\mathrm{I}}(n) = \sum_{q=0}^{n} \hat{u}_{\mathrm{P}}(q)$. Replacing the normalized terms with ANG terms the gain dependency can be described by,

$$\hat{K}_{\mathrm{Ia}}(n) \sum_{q=0}^{n} \hat{e}(q) = \sum_{q=0}^{n} \hat{K}_{\mathrm{Pa}} \hat{e}(q). \tag{3.85}$$

   By assuming the continuous form for small sampling intervals, the above expression can be further simplified. The gain dependency can be described by the following nonlinear differential equation:

$$\hat{K}_{\mathrm{Ia}} + \frac{1}{2} \frac{d\hat{K}_{\mathrm{Ia}}}{d\hat{e}} \hat{e} - \hat{K}_{\mathrm{Pa}} = 0. \tag{3.86}$$

3. Dependency between P and D controllers (Type V)

   The dependency that exists in the Type V controller output is given by, $\hat{u}_{\mathrm{D}}(n) = \hat{u}_{\mathrm{P}}(n) - \hat{u}_{\mathrm{P}}(n-1)$. With ANG terms this gain dependency in the Type V controller can be expressed by,

$$\hat{K}_{\mathrm{De}}(n) \Delta \hat{e}(n) = \hat{K}_{\mathrm{Pa}}(n) \hat{e}(n) - \hat{K}_{\mathrm{Pa}}(n-1) \hat{e}(n-1). \tag{3.87}$$

   Considering small sampling intervals, the above can be described in a continuous form by the following nonlinear differential equation.

$$\hat{K}_{\mathrm{Pa}} + \frac{d\hat{K}_{\mathrm{Pa}}}{d\hat{e}} \hat{e} - \hat{K}_{\mathrm{Da}} = 0. \tag{3.88}$$

The gain dependency makes obtaining optimum nonlinear tuning of individual ANG terms impossible. For example, in the Type V controller, both integral and derivative gains follow the nonlinear proportional action in terms of nonlinear tuning. In case of optimum nonlinear tuning, this requires a compromise for achieving best performance. The conventional Type III controller shows a highly complex gain dependency. The independent nonlinear gain control in Type II and Type VI controllers allows the design to achieve the best independent nonlinear tuning in terms of ANG values.

3.6 Summary

The systematic analysis presented in this chapter facilitated the identification of different fuzzy PID structures, particularly de-coupled and one-input type controllers, which have not

been commonly used in previous applications. Closed-form expressions for fuzzy controllers have been systematically deduced to represent the output solutions to fuzzy controllers. The transformation procedure reduces the complexity of the min-max solutions and therefore the fuzzy controller analysis produces general solutions to PID structures. The limitations of the closed-form analysis to the fuzzy systems have been clearly identified. The gain analysis enables one to simplify the fuzzy PID design as a two-level tuning task. The apparent gains have been deduced for both the tuning levels. Important functional properties have been identified for comparing and evaluating different fuzzy PID controllers. The comparison generally proves the de-coupled rule structures have the advantages of obtaining dissociated PID terms with de-coupled nonlinear gains. Also independent fuzzy rule bases avoids the gain dependency for better designs. Rule de-coupling results in the isolation of nonlinear PID actions from the coupled actions. This may sometimes causes loss of certain non-linearity features that are available in coupled rules. As an example the isolation of error change from the proportional action causes the system to be less robust. However the unknown non-linearity in the coupled actions usually causes designers to deal with complex fuzzy controller tuning procedures. The available tuning PID knowledge can be well exploited only if the rules are de-coupled and dissociated from the coupled actions. Chapter 4 explains how de-coupled rules are effectively employed to obtain non-linearity tuning for ANG terms.

Chapter 4

Analysis and Evaluation of Non-linearity in Fuzzy Systems

4.1 Introduction

The main objective of this chapter is to analyze the non-linearity aspects of the fuzzy control actions in order to realize a suitable non-linearity tuning scheme. The analysis of the non-linearity enables development and selection of efficient fuzzy systems, reasoning schemes for better control. The fuzzy control design is primarily a formulation of an unknown nonlinear mapping system for a partially known process system[108]. Although the fuzzy controller's output has been analyzed for its non-linearity [20, 116], the real effect of this non-linearity on plant performance has not been adequately addressed. The non-linearity should be related to the non-linearity tuning parameters or knowledge base parameters. If this has a higher dimension, it is difficult to understand the variation of such non-linearity within a higher parameter space. This chapter presents a suitable evaluation scheme for the non-linearity of fuzzy systems developed in Chapter 3.

As explained previously, overall performance is based on the two levels of tuning or equivalently, the tuning of apparent linear and nonlinear PID gains. Given a fuzzy system with a fixed non-linearity (or with fixed knowledge base structure) it is theoretically possible to obtain overall performance (or convergence) by adjusting the apparent linear gains. In such circumstances it is difficult to isolate the exact effect of non-linearity from the overall

plant response. Therefore the first issue in this chapter is to frame an evaluation method for identifying different non-linearity in the fuzzy output. This identification would lead to an estimate of the necessary number of fuzzy variables required for the general fuzzy systems described in the previous chapter.

The second most important issue in this work is to search for a suitable fuzzy reasoning scheme for fuzzy PID control. As mentioned in Chapter 1, there are no rigid mathematical set operations that are unique for fuzzy reasoning. This is not the case with binary logic. However, MMG is the oldest and most popular reasoning scheme for fuzzy control. The fuzzy calculation is based on basic set operations. The triangular *t*-norms and *s*-norms include several types of set operations [35]. Therefore one can find several combinations of $t-$ and $s-$ norms to generate different reasoning schemes for fuzzy control. In addition to the MMG inference, the *product-sum-gravity* (PSG) reasoning scheme has become quite popular in generating fuzzy control actions. Unlike min-max functions, the product-sum functions are continuous. Also the *sum* allows the general output of the FLC systems to be expressed in an algebraic (or additive) form [1, 104, 116]. Hence the use of PSG is relatively convenient, particularly for neuro-fuzzy implementations. Mizumoto [52] has pioneered the evaluation and comparison of different fuzzy reasoning schemes for feedback control. He has used computer simulations to validate and prove the superiority of the PSG reasoning. For the work he has used a specific knowledge base configuration and a first-order plus dead time process model. This research was important because it has provided the initial comparison and also indicated the unsuitability of some reasoning systems for fuzzy control. The fuzzy reasoning method has direct effect on the ANG terms or first-level of tuning. In addition to the nonlinear properties of the fuzzy system, the overall performance is greatly affected by the scale factors. Therefore Mizumoto's approach is impractical to use for assessing and ranking different fuzzy systems. Ying [20] later performed an analytical calculation to produce the ANG terms corresponding to different reasoning methods. The comparison was based on the conventional two-input PI configuration with a simple fuzzy linear controller. Unlike the Mizumoto's approach, this work has isolated the effects of fuzzy reasoning from the linear parameters for establishing nonlinear gains. He concluded that the bounded product inference is invalid and inappropriate for fuzzy control. The paper further admits that the analysis cannot conclude the efficiency of the fuzzy inferences for control performance. It is important

that strong non-linearity is an asset for control, but how much of non-linearity or what kind of non-linearity the fuzzy system can generate is vital for control. A similar approach has been recently adopted in [31] to compare different t-norm sum-gravity inferences. In [31] the FLC was considered with shrinking span membership functions (or simply nonlinear like controller) as opposed to linear like controller in [20]. The analysis was able to quantify the maximum non-linearity of the fuzzy system under given reasoning conditions. However the approach is unsuitable for evaluating the non-linearity variation that an inference method could produce under given tuning conditions. In all the past evaluations [20, 31, 52], the MMG and PSG were identified as valid reasoning methods. The other most popular fuzzy system in the area of fuzzy control is the Takagi-Sugeno's [39] (TSK) functional fuzzy representation. The TSK method is popularly used for fuzzy modelling applications. However, only in very fewer applications (e.g. [117]) that the TSK system has been used for fuzzy PID control. This research has provided enough motivations to compare the general TSK representation as a different fuzzy reasoning scheme for fuzzy PID control. In order to address this issue sufficiently different fuzzy systems are analyzed in terms of their nonlinear properties.

The basic features and properties of non-linearity in an LLFLC is fixed. It is possible to change the magnification of the non-linearity of the output of an LLFLC by increasing or decreasing the number of fuzzy variables in a given direction of control. It has been found that the available non-linearity variations in an LLFLC is not significant enough to provide considerable improvements to fuzzy PID control. Therefore the nonlinear properties of the NLFLC systems are first evaluated. A new performance evaluation scheme is generated for quantifying the non-linearity and its degree of freedom for ANG terms. Using this new evaluation scheme, the non-linearity of different fuzzy systems, including different reasoning methods, is compared for ranking. At the end, an alternative nonlinear control using parametric based Bézier curves are described.

4.2 Formulation of Non-linearity for Fuzzy Control

The linear control is perhaps the most conservative control policy for feedback control. That is one of the reasons why linear PID controller is satisfactorily functioning in variety of industrial control loops. The fuzzy PID controllers with the nonlinear control techniques

provide better performance than the linear counterpart. In that case it is important to evaluate the drawbacks of linear control and the potential benefits of nonlinear control before designing a fuzzy controller. Consider a set point controlled process system. The objectives are to attain,

(a) fast rise time.

(b) low settling time.

(c) least peak overshoot.

(d) steady state conditions within a narrow error margin, and

(e) quick recovery and adaptability under load or external disturbances.

The error integral criterion [118] is a mathematical representation of the above requirements. There is no perfect controller, that can individually optimize all of the above objectives. As an example the linear PID designs based on error integral optimizations in [118] provide two separate expressions for the PID gains, one is for achieving set point control properties and the other is for achieving load disturbance properties. This is mainly due to conflict in the controller policies at two different points in the control. As an example, the increase in integral gain for obtaining load disturbance properties causes the set point overshoot to be excessive and also can cause integral-windup. The linear control policy is uniformly applied over the entire control plane and hence cannot allow exceptions. Fuzzy control has the advantage of providing local control. In other words the apparent nonlinear PID gains allow equivalent or the overall PID gains to be different in different control regions.

4.2.1 Non-linearity for Local Control

Consider two control points in the control plane which are expected to provide improved control for set point control problems. The first point corresponds to the normalized error variables at the extremes or when $|\hat{e}_w| = 1$. The local control at this point can provide a higher driving force for faster rise or milder control for smooth transitions etc. The second point corresponds to the target point or when $|\hat{e}_w| = 0$. The local control at this point can influence for example, the load disturbance properties, steady state properties etc. The nonlinear tuning of the fuzzy controllers provides independent adjustment to the ANG terms

at the chosen two points. By considering more than two points it is possible to improve the control performance, at the cost of higher parameter complexity. The author believes this is an important area of further research.

4.2.2 Preferred Properties for Fuzzy Outputs

The next step in this analysis is to find the fuzzy parameters and a suitable rule base system that can provide the above local control properties. Without loss of generality the following constraints are imposed for reaching the objectives of this design. Although some of the properties have been acquired during the construction of the LLFLC and NLFLC systems, they are repeated for clarity. Again consider any three error inputs and for convenience assign $\hat{e} = \hat{e}_1$, $\Delta\hat{e} = \hat{e}_2$, and $\Delta^2\hat{e} = \hat{e}_3$. The preferred properties **P1** through **P5** are outlined below.

P1 All the controller variables are normalized to the compact region [-1,1]. The normalizing and de-normalizing factors will preserve the generalities.

P2 For achieving a unique tuning criterion set the fully normalized conditions by imposing the boundary condition: when $\hat{e}_w = \pm 1$, then $\hat{u} = \pm 1$.

P3 For steady state properties: when $\hat{e}_w = 0$, then $\hat{u} = 0$.

P4 For one-one correspondence and rule continuity the output is required to be monotonic and continuous and therefore set $\partial\hat{u}/\partial\hat{e}_w > 0$.

P5 For symmetrical control at the set point consider the anti-symmetric property,

$$\hat{u}(+\hat{e}_1, +\hat{e}_2, +\hat{e}_3) = -\hat{u}(-\hat{e}_1, -\hat{e}_2, -\hat{e}_3)$$

4.2.3 Non-linearity Tuning Variables

Once the fuzzy system is designed to provide the above properties, the next step is to determine the tuning parameters and the two-levels of tuning. Since ANG terms are related to the slopes of the control plane in the respective error directions, the angles or slopes of the tangents drawn at the chosen control points are selected as the main tuning variables for obtaining non-linearity tuning for local control. In order to isolate the tuning parameters from the associated ANG terms in coupled outputs of coupled rule bases, the slopes are measured in the planes of the individual error axes. Figure 4.1a shows a control curve that has been

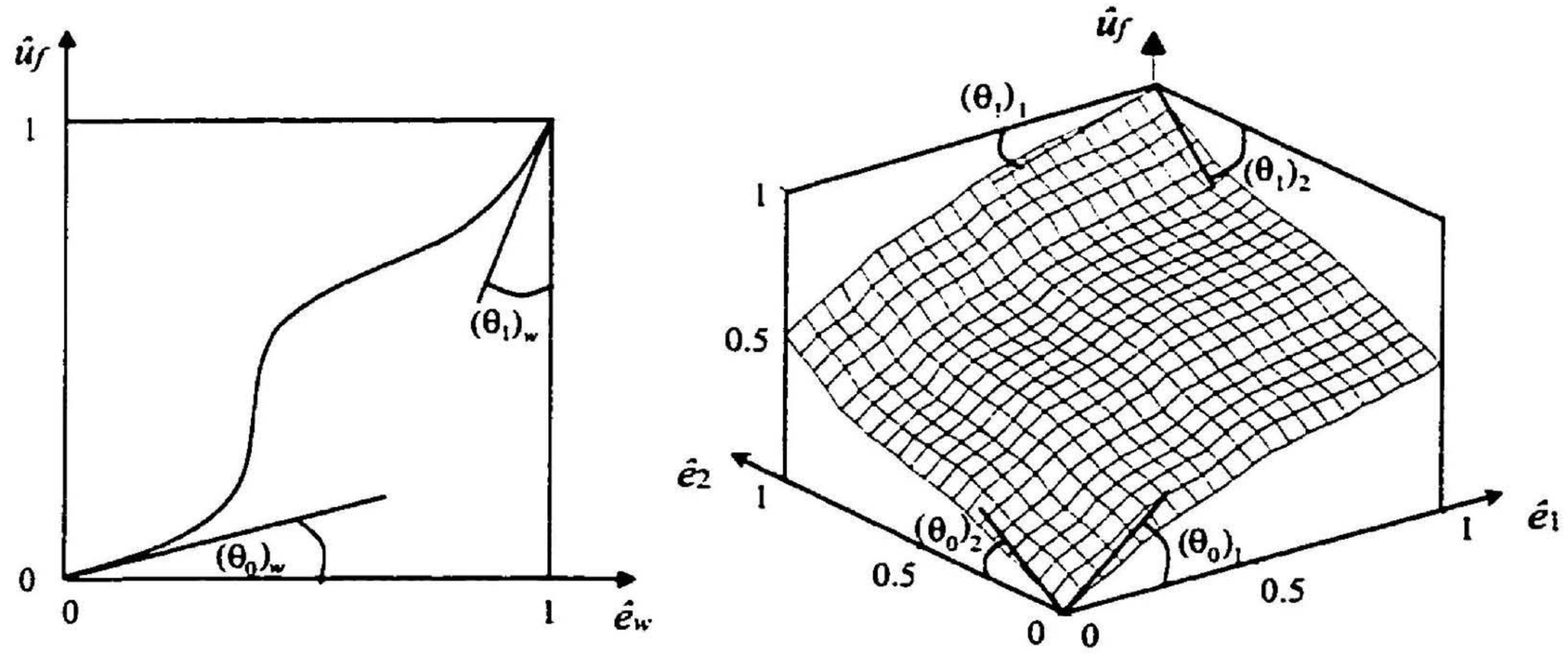


Figure 4.1: Non-linearity tuning parameters for local control

projected into a chosen error variable. The gradients of the tangents drawn at the zero and maximum points are selected as the tuning variables for achieving the non-linearity tuning. The measurement of these gradient angles (θ_0 and θ_1) with respect to a two-dimensional NLFLC-I control surface is shown in Figure 4.1b. Therefore for a given fuzzy system the state of its non-linearity is described by these angles. In general, for the three-input coupled rule base, these angles can be described by,

$$(\theta_0)_w = \left(\frac{\partial \hat{u}_f}{\partial \hat{e}_w}\right)_{\hat{e}_w=0}, \ (\theta_1)_w = \left(\frac{\partial \hat{u}_f}{\partial \hat{e}_w}\right)_{\hat{e}_w=1}, \tag{4.1}$$

where the fuzzy output $\hat{u}_f$ in (4.1) is the projected control curve onto $\hat{e}_w - \hat{u}$ plane and is

$$\hat{u}_f = \hat{u}(\hat{e}_p = 0), \ p = 1, 2, 3 \text{ and } p \neq w.$$

This can be an incremental or absolute signal depending on the type of the PID structure. The fuzzy system designed for the PID control should allow independent variations of θ_0 and θ_1 within the range $[0, 90^o]$ for obtaining local control at the chosen control points. Some researchers believe that more rules or fuzzy variables near the set point or zero error can bring accurate control [1, 30, 65]. Therefore the membership functions are sometimes placed in such a way that the support sets become narrower (called shrinking span memberships) near zero [30, 31]. It has been seen from the LLFLC and NLFLC analysis that this arrangement would produce a more rigid linear surface near zero and the slope of the normalized surface would

converge to 45^o as a linear PID controller. Therefore the design will lose the real benefits of fuzzy control. In a properly designed fuzzy controller the non-linearity tuning should produce this linear function as a special case of the fuzzy control.

4.3 Development of Fuzzy Systems for Two-point Control

In this part the FLC systems generalized in Chapter 3 are considered for obtaining the specific two-point control characteristics. With respect to non-linearity tuning, the LLFLC systems have limitations in providing sufficient and significant nonlinear control as compared to NLFLC systems. In fact the LLFLC is a special case of the NLFLC system. Therefore the non-linearity analysis is not performed for LLFLC systems. The nonlinear tuning of the NLFLC systems are obtained by altering the fuzzy partitioning points in the input and output spaces. As there are two independent non- linearity indicators corresponding to the two control points, the NLFLC systems require at least two independent parameters for representing the tuning of each ANG term in the fuzzy PID system.

4.3.1 NLFLC-I System

By observation it can be concluded that the number of membership functions needed in NLFLC-I system for obtaining this local control can be accomplished by assigning two independent partitioning points for each input variable. Under these conditions the input membership parameters can be fixed for a general three-input NLFLC-I rule base system as;

$$N_1 = N_2 = N_3 = 7.$$

The number of output fuzzy variables is $M = N_1 + N_2 + N_3 - 2 = 19$. For two- and one-input fuzzy rule bases $M = 13$ and $M = 7$ respectively. Therefore for any given input variable the modal positions can be defined by,

$$\hat{\mathbf{e}}_w = \{-1, -(s_2)_w, -(s_1)_w, 0, (s_1)_w, (s_2)_w, 1\}^T. \tag{4.2}$$

The partitioning is arranged in such a way that the membership positions are symmetrically placed about the zero modal position to facilitate the property P5. This special arrangement of the membership functions for all the antecedent variables is shown in Figure 4.2. The

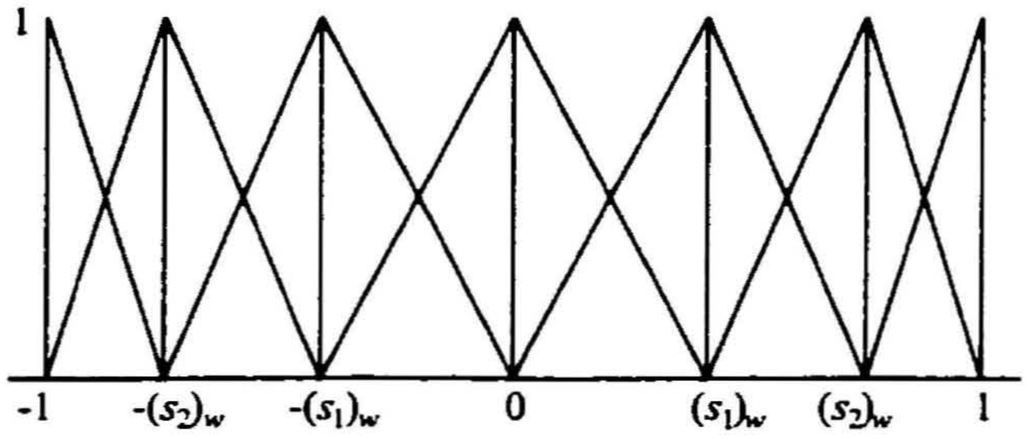


Figure 4.2: The simplified membership distributions of the antecedents of NLFLC-I and II systems

non-linearity tuning is determined by the input partitioning points that are specified by the distances $(s_1)_w$ and $(s_2)_w$. By varying these positions it is possible to change the angles or the slopes of the projected curves at the two specified control points. The ranges for the tuning parameters can be specified by:

$$(s_1)_w \in (0, (s_2)_w]. \quad (s_2)_w \in [(s_2)_w, 1). \tag{4.3}$$

When the control surface is projected to one of the axis then the shape of the curve in Figure 4.1 would be equivalent to a one-input inference with reduced magnifications to the height. Therefore the angles obtained for each error axis of the NLFLC-I output are described below.

For three-input rule base

$$\left.\begin{aligned} (\theta_0)_w &= \arctan\left(\tfrac{1}{6(s_1)_w}\right) \\ (\theta_1)_w &= \arctan\left(\tfrac{1}{6(1-(s_2)_w)}\right) \end{aligned}\right\}. \tag{4.4}$$

with $w = 1, 2, 3$.

For two-input rule base

$$\left.\begin{aligned} (\theta_0)_w &= \arctan\left(\tfrac{1}{4(s_1)_w}\right) \\ (\theta_1)_w &= \arctan\left(\tfrac{1}{4(1-(s_2)_w)}\right) \end{aligned}\right\}. \tag{4.5}$$

with $w = 1, 2$.

For one-input rule base

$$\left.\begin{aligned} (\theta_0)_w &= \arctan\left(\tfrac{1}{2(s_1)_w}\right) \\ (\theta_1)_w &= \arctan\left(\tfrac{1}{2(1-(s_2)_w)}\right) \end{aligned}\right\}. \tag{4.6}$$

with $w = 1$.

4.3.2 NLFLC-II System

In addition to the above membership distribution given in (4.2), the NLFLC-II system has variable fuzzy output singletons. While having two more additional output positions corresponding to a single variable, more depth or more variations to the two slopes can be obtained. However with coupled rules the number of fuzzy singleton positions available within the normalized space is high. To reduce the parameter complexity, again the total number of unknown parameters related to output fuzzy singleton positions defined in the equation (3.49) are reduced to two. The projected slopes are obtained by using the NLFLC output solutions derived in Section 3.3. As in the NLFLC-I system, the two input membership parameters allowed to vary within the same limits as described in (4.3). The membership distribution of the antecedents are similar to NLFLC-I in Figure 4.2.

For three-input controllers

The output singleton positions are simplified to,

$$\{\hat{u}_m\} = \left\{ \begin{array}{l} -1, -(2/3+\hat{u}_2), -(2/3+\hat{u}_1), -1/3, -(1/3+\hat{u}_2), -(1/3+\hat{u}_1), 1/3, -\hat{u}_2, \\ -\hat{u}_1, 0, \hat{u}_1, \hat{u}_2, 1/3, (1/3+\hat{u}-1), (1/3+\hat{u}_2), 2/3, (2/3+\hat{u}_1), (2/3+\hat{u}_2), 1 \end{array} \right\}. \tag{4.7}$$

The allowable ranges for the fuzzy singleton parameters are,

$$\hat{u}_1 \in [0, \hat{u}_2] \text{ and } \hat{u}_2 \in [\hat{u}_2, 1/3],$$

and the slope angles are obtained as

$$\left. \begin{array}{l} (\theta_0)_w = \arctan\left(\frac{\hat{u}_1}{(s_1)_w}\right) \\ (\theta_1)_w = \arctan\left(\frac{(1/3-\hat{u}_2)}{(1-(s_2)_w)}\right) \end{array} \right\}, \tag{4.8}$$

with $w = 1, 2, 3$.

For two-input controllers

The output singleton positions are simplified to,

$$\{\hat{u}_m\} = \left\{ \begin{array}{c} -1, -(1/2+\hat{u}_2), -(1/2+\hat{u}_1), -1/2, -\hat{u}_2, -\hat{u}_1, \\ 0, \hat{u}_1, \hat{u}_2, 1/2, (1/2+\hat{u}-1), (1/2+\hat{u}_2), 1 \end{array} \right\}. \tag{4.9}$$

The allowable ranges for the fuzzy singleton parameters are

$$\hat{u}_1 \in [0, \hat{u}_2] \text{ and } \hat{u}_2 \in [\hat{u}_2, 1/2],$$

and the slope angles are obtained as

$$\left.\begin{aligned}(\theta_0)_w &= \arctan\left(\frac{\hat{u}_1}{(s_1)_w}\right)\\ (\theta_1)_w &= \arctan\left(\frac{(1/2-\hat{u}_2)}{(1-(s_2)_w)}\right),\end{aligned}\right\} \tag{4.10}$$

with $w = 1, 2$.

For one-input controllers

The output singleton positions are simplified to,

$$\{\hat{u}_m\} = \{-1, -\hat{u}_2, -\hat{u}_1, 0, \hat{u}_1, \hat{u}_2, 1\}. \tag{4.11}$$

The allowable ranges for the fuzzy singleton parameters are.

$$\hat{u}_1 \in [0, \hat{u}_2] \text{ and } \hat{u}_2 \in [\hat{u}_2, 1],$$

and the slope angles are obtained as

$$\left.\begin{aligned}(\theta_0)_w &= \arctan\left(\frac{\hat{u}_1}{(s_1)_w}\right)\\ (\theta_1)_w &= \arctan\left(\frac{(1-\hat{u}_2)}{(1-(s_2)_w)}\right)\end{aligned}\right\}, \tag{4.12}$$

with $w = 1$.

Since the output membership positions are related to all input variables in a coupled fuzzy rule base. the variation of the output singleton positions affects the non-linearity and the projected slopes in all directions. Therefore an independent non-linearity tuning with the output membership positions are impracticable with coupled rules. Thus the non-linearity tuning parameters can be grouped into two parts. The terms $(s_1)_w$ and $(s_2)_w$ can be grouped as *primary non-linearity tuning parameters* where as $\hat{u}_1$ and $\hat{u}_2$ can be grouped as *secondary non-linearity tuning parameters.*

4.3.3 NLFLC-III System Using MMG Reasoning

This controller was specially designed for the de-coupled rule bases and the fuzzy outputs have been already identified with respect to the non-linearity tuning parameters. Therefore the slope angles of the two controllers defined for this category are shown for defining the tuning. The MMG based solutions to this system are shown in Section 3.3.2. Therefore by directly substituting the output solutions given in (3.61) and (3.62) into (4.1) the slope

angles related to a SISO fuzzy system are shown below. For convenience assign $s_1 = (s_1)_1$, $s_2 = (s_2)_1$: and $\theta_0 = (\theta_0)_1$, $\theta_1 = (\theta_1)_1$.

For NLFLC-IIIA system

$$\theta_0 = \begin{cases} \arctan\left[2(1-s_1)/s_1\right] & \text{for } s_1 \leq s_2. \\ \arctan\left[(2-s_1-s_2)(2+s_1-s_2)/2s_1\right] & \text{for } s_1 > s_2. \end{cases}$$
$$\theta_1 = \begin{cases} \arctan\left[2s_1/(1-s_2)\right] & \text{for } s_1 \leq s_2. \\ \arctan\left[(s_1+s_2)(2+s_1-s_2)/(2-2s_2)\right] & \text{for } s_1 > s_2. \end{cases} \tag{4.13}$$

For NLFLC-IIIB system

$$\theta_0 = \begin{cases} \arctan\left[3(1-s_2^2)/(2s_1(2+s_2))\right] & \text{for } s_1 \leq s_2. \\ \arctan\left[3(1-s_1^2)/(2s_1(2+s_2))\right] & \text{for } s_1 > s_2. \end{cases}$$
$$\theta_1 = \begin{cases} \arctan\left[4s_1/(1-s_2)\right] & \text{for } s_1 \leq s_2. \\ \arctan\left[(s_1+s_2)(4+3s_1-s_2)/\left((1-s_2)(2+s_2)\right)\right] & \text{for } s_1 > s_2. \end{cases} \tag{4.14}$$

4.3.4 NLFLC-III Using PSG Reasoning

Both the NLFLC-IIIA and B systems can be again analyzed using the PSG reasoning. The detailed derivation is shown in Appendix B. From the derivation it can be seen that the relationships between the angles are the same for both the A and B systems. The *product-sum* functions are continuous and as a result the output solution of the NLFLC-III system can be expressed with a single equation. Also the two membership parameters can be allowed to vary within,

$$s_1 \in (0, 1] \text{ and } s_2 \in [-1, 1).$$

For comparison, the angles for the PSG reasoning based NLFLC-IIIB system are shown below.

$$\theta_0 = \arctan\left[(1-s_2)/2s_1\right].$$
$$\theta_1 = \arctan\left[2s_1/(1-s_2)\right]. \tag{4.15}$$

4.3.5 NLFLC-III Using TSK Representation

In this case for the same fuzzy input variables the fuzzy output variables are functionally expressed in terms of the input crisp variables. In the traditional TSK representation, the

outputs of the fuzzy rules are expressed in a linear form. Those functions are defined with the intention of representing any unknown control surface by linearly defined fuzzy regions or simply as a linearized function for a given range of inputs [39]. However when those linear models are used for a SISO fuzzy rule base, the non-linearity variation will become somewhat limited. In order to achieve better local control properties, the output functions are expressed as a polynomial function of the input variable. In fact by dropping one parameter, the polynomial can be made to the traditional linear form. Therefore this is somewhat a modified TSK representation to suit the fuzzy PID control. Hence, the three rules corresponding to the NLFLC-III input fuzzy variables in Figure 3.14 are defined as

$$\left.\begin{array}{l} \text{If } \hat{e}_1 \text{ is NB then } \hat{u} = -s_0 + s_1\hat{e}_1 - s_2\hat{e}_1^2 \\ \text{If } \hat{e}_1 \text{ is NZ then } \hat{u} = 0 \\ \text{If } \hat{e}_1 \text{ is PB then } \hat{u} = s_0 + s_1\hat{e}_1 + s_2\hat{e}_1^2 \end{array}\right\}. \tag{4.16}$$

The detail derivation for the fuzzy output is shown in Appendix B. The slope angles corresponding to the TSK based NLFLC-III output are given by:

$$\left.\begin{array}{l} \theta_0 = \arctan(1 - s_1 - s_2) \\ \theta_1 = \arctan(1 + s_1 + 2s_2) \end{array}\right\}. \tag{4.17}$$

In order for the TSK output to be monotonic for any crisp error input, the two parameters s_1 and s_2 should be constrained to vary within the ranges **R1** or **R2**.

R1 : $s_1 \in [-6.464, 0.464]$ and $s_2 \in [s_{2a}, s_{2b}]$ where

$$\left.\begin{array}{l} s_{2a} = \frac{(1-s_1)}{2} - \frac{1}{6}\sqrt{9 - 18s_1 - 3s_1^2} \\ s_{2b} = \frac{(1-s_1)}{2} + \frac{1}{6}\sqrt{9 - 18s_1 - 3s_1^2} \end{array}\right\}. \tag{4.18}$$

R2 : $s_2 \in [0.4]$ and $s_1 \in [s_{1a}, s_{1b}]$ where

$$\left.\begin{array}{rl} s_{1a} = & -0.5\left(\sqrt{12s_2 - 3s_2^2} + 3s_2\right) \\ s_{1b} = & 0.5\left(\sqrt{12s_2 - 3s_2^2} - 3s_2\right) \end{array}\right\}. \tag{4.19}$$

4.4 Non-linearity Variations and Evaluation

The two-point control analysis has simplified the general fuzzy systems into several unknown membership parameters (or partitioning points). Except for the NLFLC-II, all other systems

have been reduced to two unknown parameters per variable for non-linearity tuning. As the slopes at the control points are the key parameter for obtaining different non linearity, the strength or the capacity of a fuzzy system can be measured by observing the variation of these slope angles with respect to the non-linearity tuning parameters. As the angles are defined only in respective error directions (projected angles) of the error variables, only the SISO fuzzy systems are considered for this evaluation. Therefore five fuzzy systems are compared as described below.

C1 : NLFLC-I controller having one-input inference (MMG reasoning)

C2 : NLFLC-IIIA controller with MMG reasoning

C3 : NLFLC-IIIB controller with MMG reasoning

C4 : NLFLC-III controller with PSG reasoning

C5 : NLFLC-III controller with TSK reasoning

4.4.1 Performance Measures and Comparison of FLC Systems

Non-linearity Variation Index (NVI)

The non-linearity variation is defined to assess the magnitude of non-linearity and the degree of variation it can produce by changing the knowledge base parameters of a given fuzzy system. From a mathematical point of view this is a complex task. However, a general expression is defined to assess the non-linearity and it is then simplified for the one-dimensional case to evaluate and compare different FLC systems.

$$NVI(n_v, n_t, n_e) = \frac{\text{Admissible space in } n_e \text{ space}}{\text{Complete space in } n_e \text{ space}}. \tag{4.20}$$

where,

n_v = Total number of input variables in the rule base

n_t = Total number of non-linearity tuning parameters

n_e = Total number of non-linearity examination parameters

The definition reflects that NVI is dimensionless and its maximum value is always 1. Therefore a fuzzy system giving a high NVI can be treated as a better nonlinear and more versatile

controller with greater flexibility to change its non-linearity for wider range of control requirements. The output of each fuzzy system is evaluated on the same basis so that the NVI representation is unique. The NLFLC-II controller is eliminated as its output for one-input variable is a piece-wise linear curve. Also this curve does not have the C^1 continuity properties. For set point control consider the two-point control strategy. One-input fuzzy knowledge base implies $n_v = 1$. Each controller is tuned by changing the two nonlinear tuning parameters s_1 and s_2 assigned for a single variable and therefore $n_t = 2$. The two slope angles imply $n_e = 2$. It is clear that by increasing n_t and n_e it is possible to acquire tight control at many locally chosen points in the closed-loop control. Hence the evaluation is done on the basis of $NVI(1, 2, 2)$. For each SISO type controller described above, the variation of the slope angles with the two non-linearity-tuning parameters are depicted in Figure 4.3–4.7. As an example, the Figure 4.3 is constructed by using the two expressions given in equation (4.6). The variation of two angles in each diagram with respect to the two tuning parameters are shown by contour lines. As an example the parameters s_1 and s_2 are varied within the limits given in (4.3) to observe the variation of two angles in Figure 4.3. Similarly Figures 4.4, 4.5, 4.6 and 4.7 are obtained by using expressions respectively given by (4.13), (4.14), (4.15) and (4.17). The tuning parameters are constrained to vary within the limits specified for each controller type.

The percentage area of the admissible space determines the non-linearity freedom of the fuzzy system for producing a variety of non linear curves. The contour lines for tuning parameters have been drawn for each case to obtain the non-linearity tuning. Therefore the non-linearity variation diagram can be referred to as the non-linearity tuning diagram for the first-level of tuning. The tuning while using this diagram is explained in Chapter 6.

Linearity Approximation Index (LAI)

If the fuzzy outputs are perfectly linear then the fuzzy PID controllers would become identical to a digital linear PID controller in either absolute or incremental form. When a fuzzy system together with a suitable fuzzy reasoning mechanism has the capability to produce this function, the fuzzy controller can perform no worse than a linear controller. The ideal linear representation would be when the slope of the curve is uniform and equal to 45^o. Therefore the *most linear* point related to each tuning diagram is when $\theta_0 = \theta_1 = 45^o$ throughout.

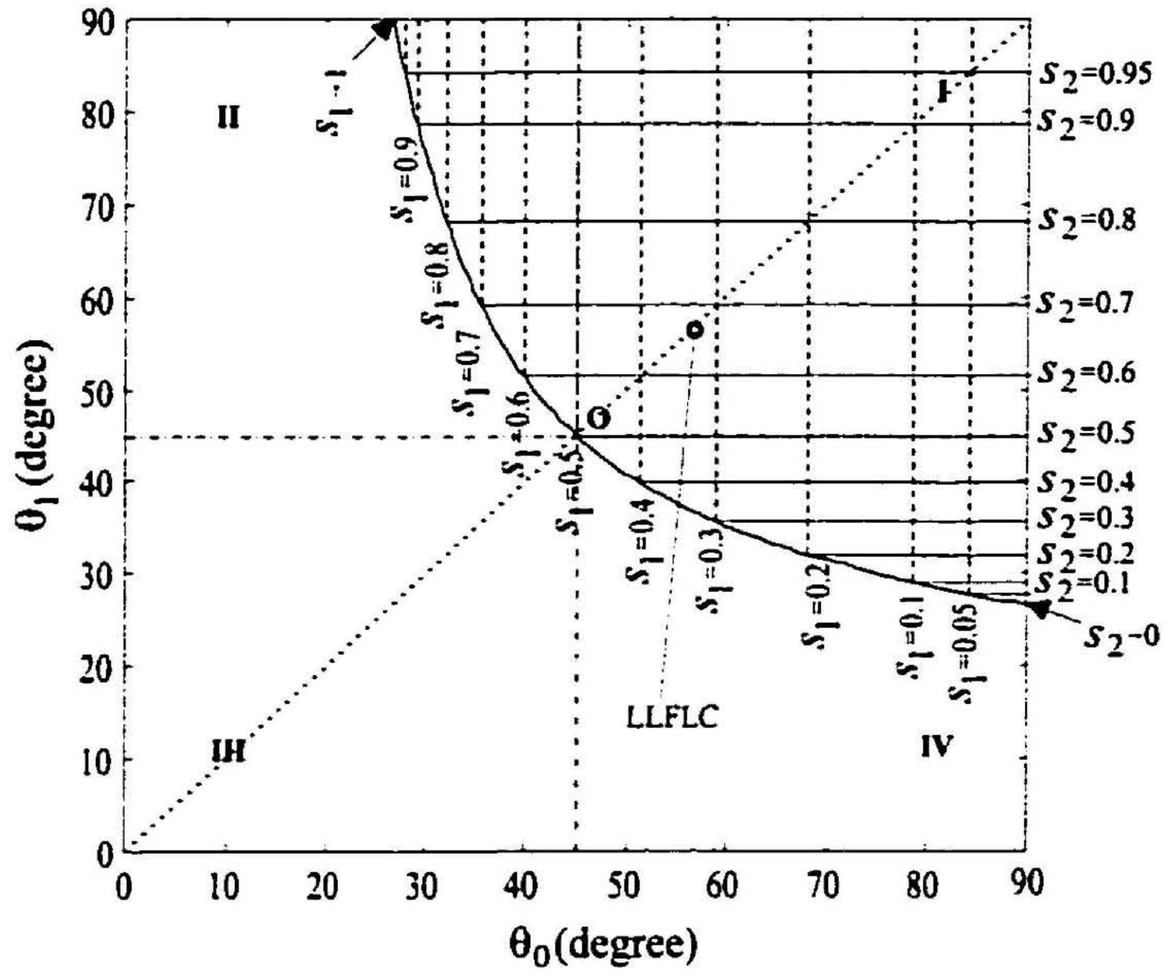


Figure 4.3: Non-linearity variation diagram for controller C1

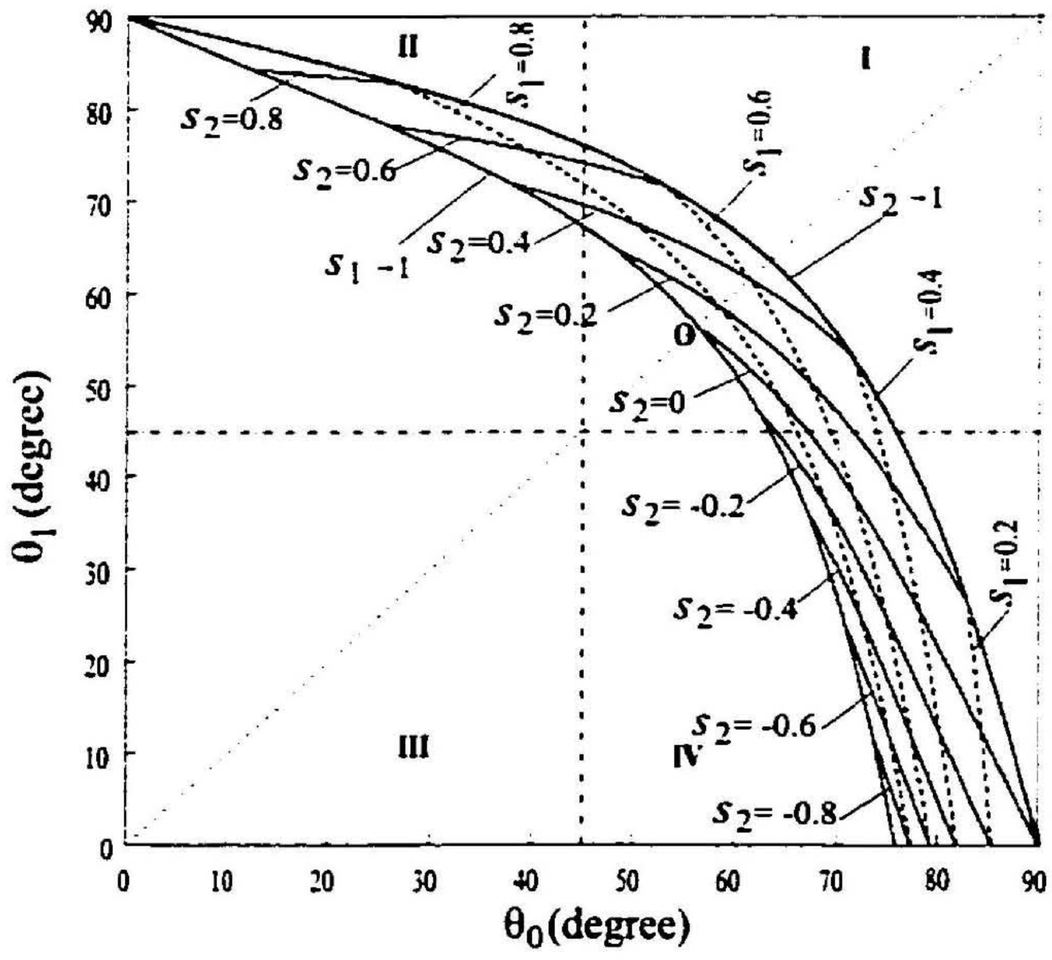


Figure 4.4: Non-linearity variation diagram for controller C2

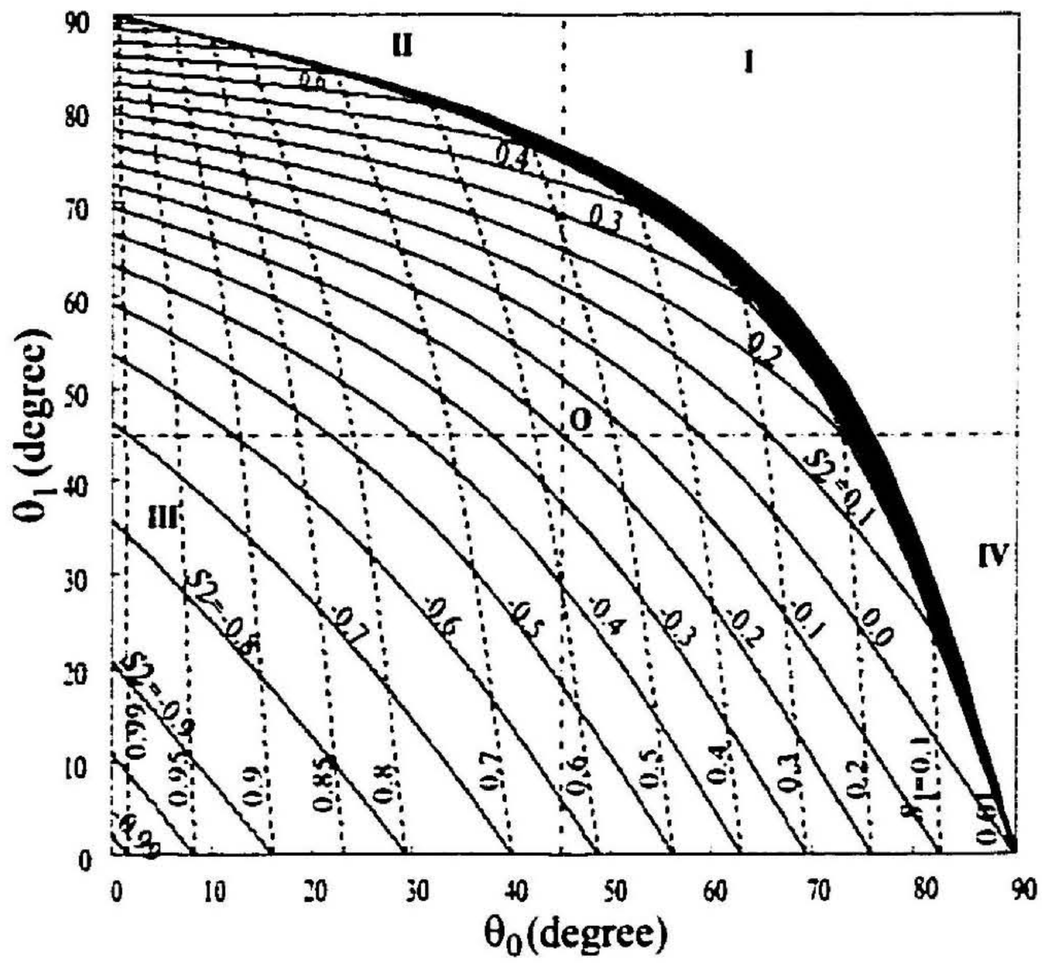


Figure 4.5: Non-linearity variation diagram for controller C3

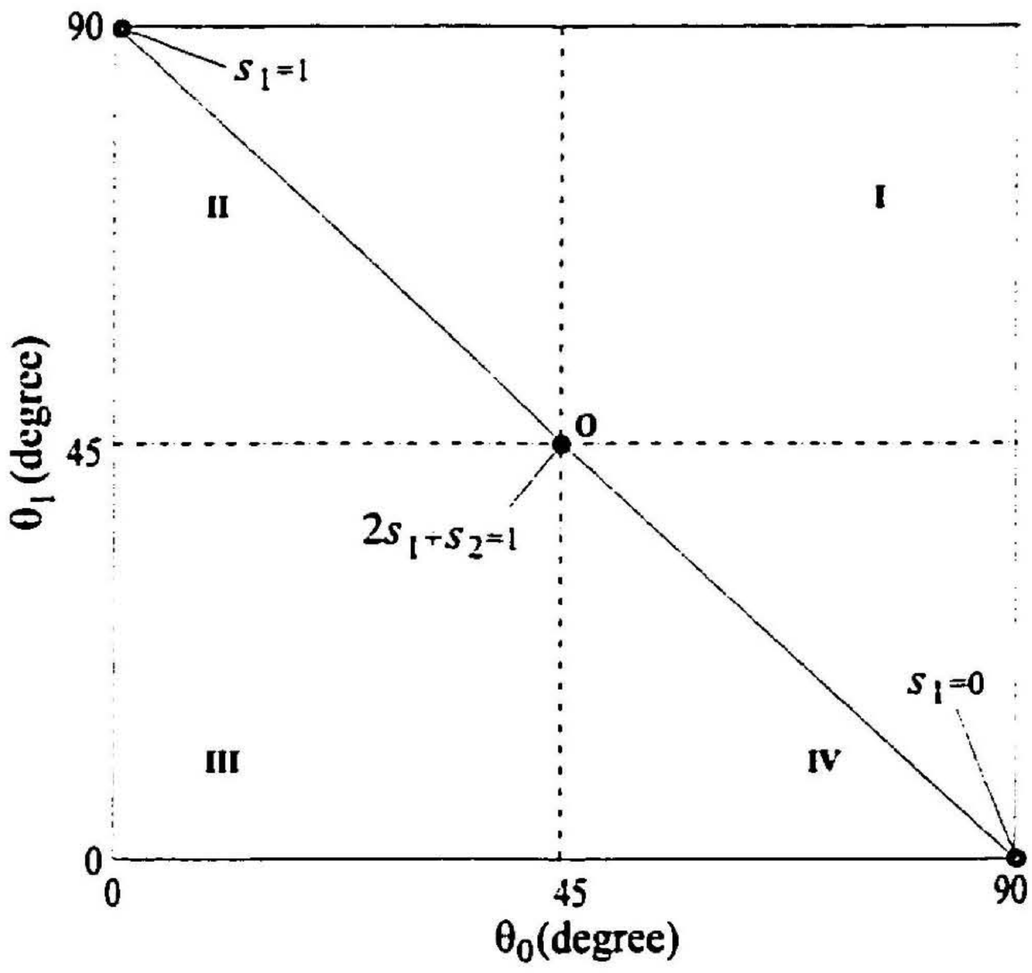


Figure 4.6: Non-linearity variation diagram for controller C4

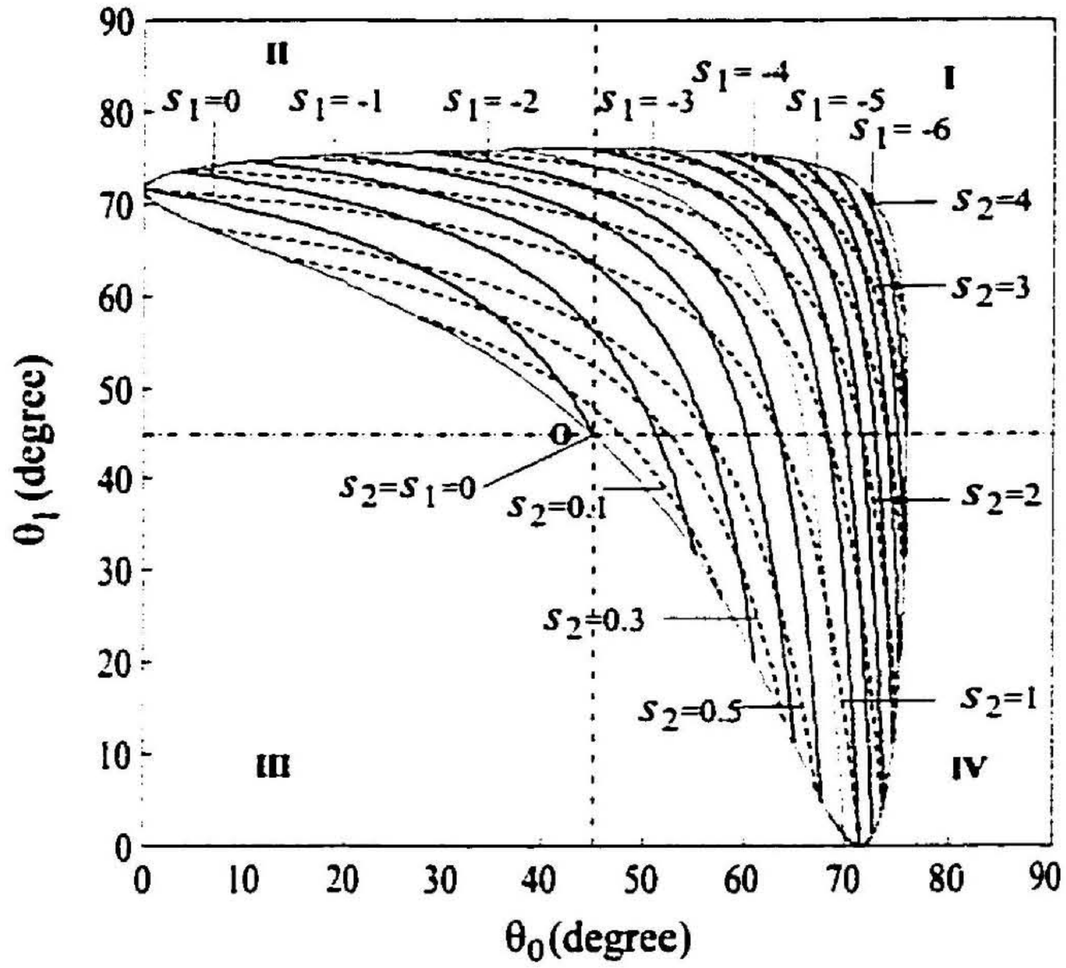


Figure 4.7: Non-linearity variation diagram for controller C5

Except for the NLFLC system denoted by C2, all other systems have the capability to obtain this closed position. The point **O** in all tuning diagrams indicates the corresponding closest positions. The point closest to the ideal point is chosen for the C2 controller. Therefore the capability of generating the ideal linear controller is considered as a preferred feature for a fuzzy system and therefore a performance measure is identified by the linear approximation index (LAI):

$$LAI = 1 - \frac{\max|\hat{u}(\hat{e}_1, s_{1O}, s_{2O})| - \bar{u}(\hat{e})}{\max|\hat{u}(\hat{e}_1, s_{1O}, s_{2O})|} \tag{4.21}$$

where, s_{1O}, s_{2O} are the tuning parameters that correspond to the most linear point **O** shown in all tuning diagrams. The linear function $\bar{u}$ is constrained to pass through the (0,0) and (1,1). Since the fuzzy output is fully normalized within [-1,1], the linear function in this case is when $\bar{u} = \hat{e}$.

Availability

The greater availability of non-linearity types is important for a general controller. With respect to the four different quadrants in the tuning diagrams, four basic nonlinear curve types have been identified and are shown in Figure 4.8. When a given fuzzy system has

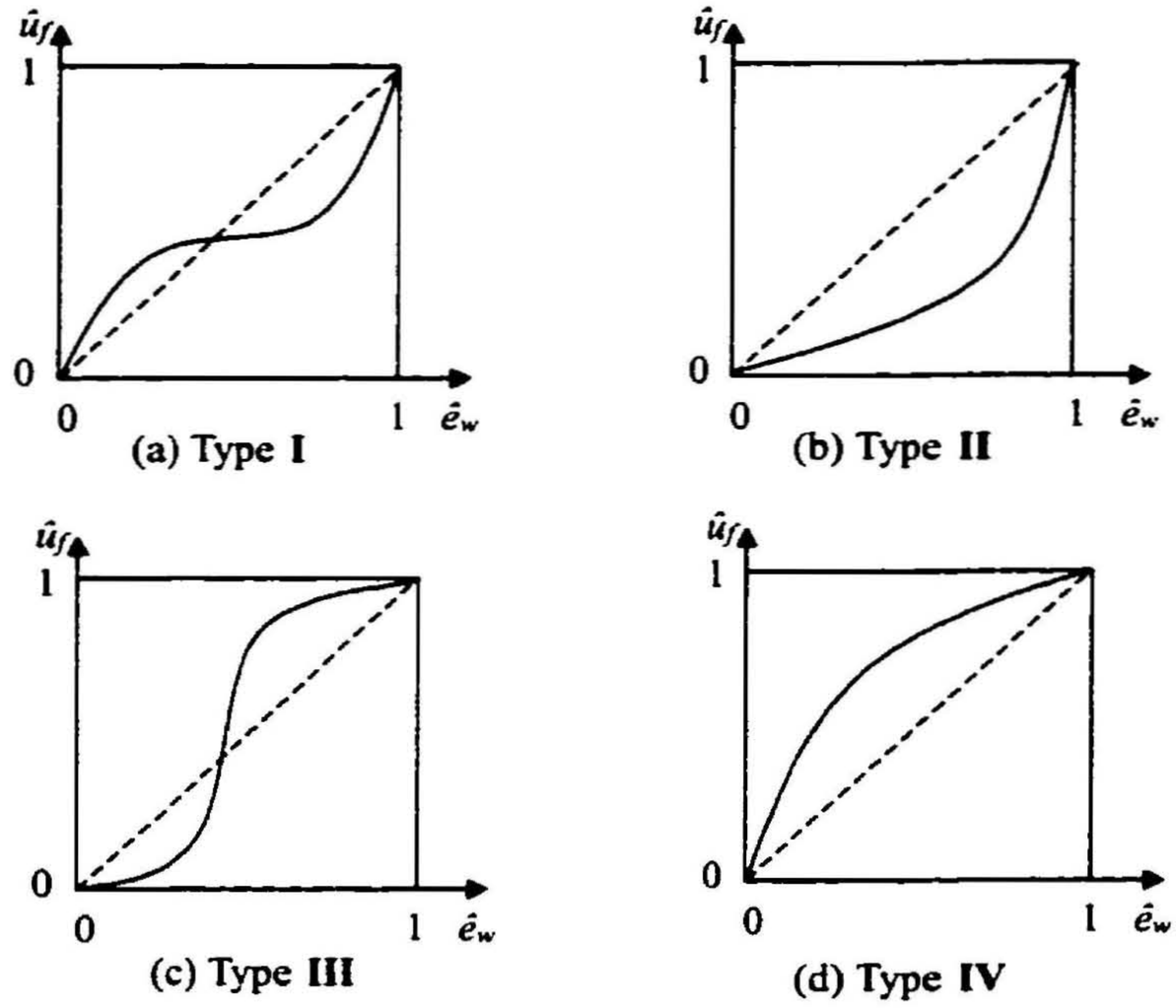


Figure 4.8: Different types of nonlinear curves corresponding to four quadrants in the tuning diagrams

the capability to produce all four types then such a controller will have greater availability to suite wider range of controller dynamics. This feature is important for adaptive or self-organizing controllers. The change of plant dynamics may require varying the non-linearity of the control. Thus the maximum score for *Availability* is 4.

Comparison

The three quantitative indexes computed for the fuzzy systems (C1-C5) are shown in Table 4.1. In controller C1 the approximated linear controller corresponds to $s_1 = s_2 = 0.5$. The modal positions of the 2^{nd} and 3^{rd} input membership variables and the modal positions of the 5^{th} and 6^{th} input modal positions of the NLFLC-I coincide. According to the range of definitions for the tuning variables, this equal condition does not exist. Therefore the corresponding LLFLC system with $s_1 = 1/3$ and $s_2 = 2/3$ is considered as the best linear representation for deriving the LAI value. The best linear condition or the position **O** for the controller C2 is when $s_1 = 1$ and $s_2 = 0$. This corresponds to the three-rule LLFLC system.

Table 4.1: Performance of different FLC systems

Controller	NVI	LAI	*Availability*
C1	0.3281	0.9862	3
C2	0.144	0.959	3
C3	0.7545	0.9739	4
C4	0	1	2
C5	0.286	1	3

The approximate linear conditions of the controller C3 is when $s_1 = 0.566$ and $s_2 = -0.2$. The TSK and PSG methods have the ability to generate perfect linear conditions. However, the linear approximations of the other systems are practically linear enough to implement as a linear PID controller. The evaluation scheme suggests that the controller C3 possess the best merits to implement for fuzzy PID applications. The (C3–C5) controllers are identical except the reasoning mechanism. This proves that MMG reasoning has the most capabilities to provide better nonlinear control than other reasoning schemes. The PSG reasoning has the poorest ranking in this evaluation. Also the tuning diagram implies the two tuning variables are insufficient for providing a greater non-linearity variation. This means that the PSG requires more tuning variables or higher dimensionality for providing the equivalent non-linearity properties. As an example, the non-linearity effect of the PSG system can be equivalently obtained by the MMG system with a single tuning parameter. The TSK representation (controller C5) is the simplest and has shown sufficient merits to use for fuzzy control. This is mainly due to the polynomial representation of the output function. With the linear representation the performance of the TSK will become similar to PSG. However, alteration of the output functions in the TSK system allows better manipulation of the non-linearity effect in the resulting output function for fuzzy control. Although the output expression of the TSK system is simpler, further analysis is required to define the valid ranges for the tuning variables for obtaining the desired properties of the controller output. On the other hand the natural representation of membership functions in MMG and PSG systems preserves the most generalities for PID control. The low NVI of the controller C1 and

highest score for controller C3 proves that the variation of input membership functions is less effective than varying the output membership parameters. However the tuning diagram for the C1 system can be used for non-linearity tuning of rule coupled fuzzy PID systems.

4.5 Alternative Nonlinear Controller Using Bézier Functions

The above non-linearity analysis motivated a search for alternative input-output mapping systems to generate the desired nonlinear control surfaces or curves for fuzzy control. A simple polynomial is a quick guess for such alternatives. By observing the properties of the nonlinear proportional action. a nonlinear curve using spline-based functions is explained in [33]. This section will explain how such a system is effectively used to produce nonlinear functions. which can provide greater non-linearity with respect to the three performance measures defined in the previous section.

4.5.1 Bézier Curves: Definition

The computer aided geometric designs includes many ways of generating curves or surfaces to approximate input/output data points. These functions. if carefully selected. can be implemented as an alternative to fuzzy controller output by suitably selecting the necessary tuning parameters. The *spline curves* are piece-wise functions consisting polynomial pieces of n degree. The construction of such curves can be made by the parameterization. The *Bézier curves* are special piece-wise polynomials designed to produce smoother curves and surfaces for geometric modelling. Many of the text books written in the area of computer aided geometric designs provide this standard mathematical procedure of obtaining such curves and surfaces. For this work the two textbooks [119, 120] have been selected as the references to explain the generation of Bézier curves. Most of the notation in [119] are used to present the variables in this section.

The *Bernstein polynomials* are the basic functions used for Bézier curves. The Bernstein polynomials are derived from the binomial formula. The polynomials of degree n is given by,

$$B_r^n(0) = C_r^n(1-s)^{n-r}s^r, \quad r = 0, 1, \ldots, n. \tag{4.22}$$

$$\text{where } C_r^n = \frac{n!}{r!(n-r)!}.$$

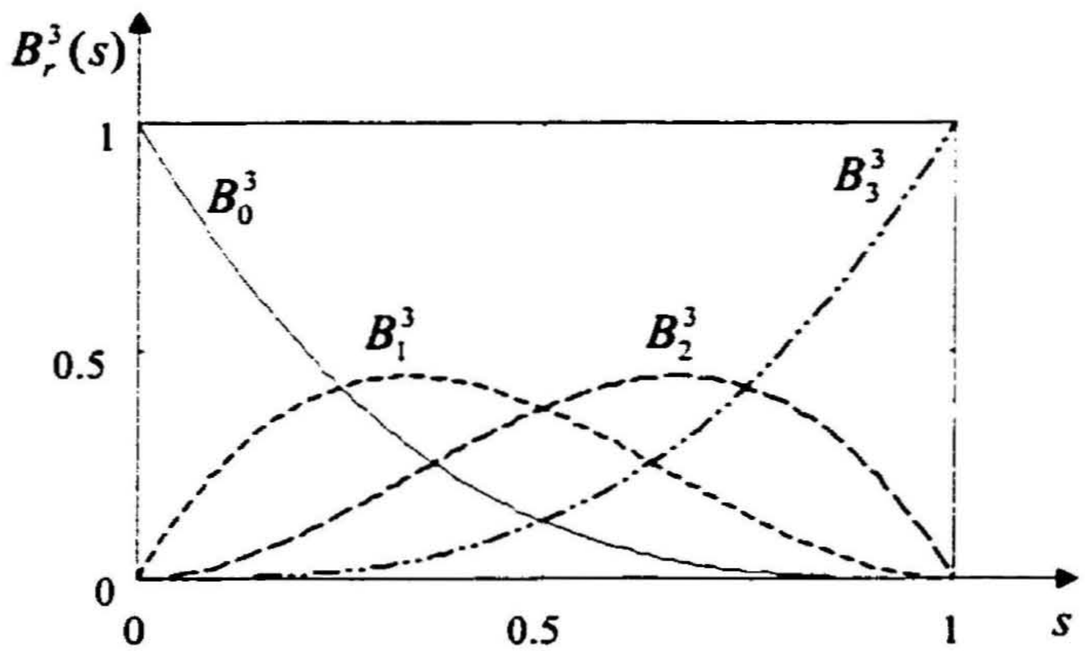


Figure 4.9: The Bernstein polynomials of degree 3

Consider the parameter interval. $s \in [0, 1]$. Then the above function carries the following properties.

$$\begin{aligned} &B_r^n(0) = B_r^n(1) = 0, && r \neq 0,\ r \neq n \\ &B_0^n(0) = B_n^n(0) = 1, && B_0^n(1) = B_n^n(0) = 0 \\ &B_r^n(s) \geq 0.\, s \in [0, 1], && \max B_r^n(s) = B_r^n(r/n) \\ &B_r^n(s) = B_{n-r}^n(1-s), && \textstyle\sum_{r=0}^n B_r^n(s) = 1. \end{aligned} \tag{4.23}$$

Figure 4.9 shows the graphs of the Bernstein polynomials $B_r^3(s)$ for $s \in [0, 1]$.

Using the Bernstein polynomials as basis functions, a *Bézier curve* or *Bézier polynomial* of degree n can be represented in the parametric form.

$$\boldsymbol{X}(s) = \sum_{r=0}^{n} \boldsymbol{b}_r B_r^n(s), \tag{4.24}$$

where $\boldsymbol{b}_r$ points in two ($\Re^2$) or three ($\Re^3$) dimensional space are referred to as *Bézier points*. The polygon formed by connecting the Bézier points is called the *Bézier polygon*. It has been proved that Bézier curves are always tangent to the Bézier polygons at the end points (i.e. $r = 0$ and $r = 1$). This implies that a Bézier curve starts at $\boldsymbol{b_0}$ and ends at $\boldsymbol{b_n}$ and the lines $\overline{\boldsymbol{b_0 b_1}}$ and $\overline{\boldsymbol{b_{n-1} b_n}}$ of the Bézier polygon are tangent to the Bézier curve. This particular feature can be exploited to develop non-linearity tuning for the two control points in the PID control. To evaluate the polynomial in (4.24) at a given point $s = s^*$, the recursive method called *de Casteljau algorithm* can be conveniently used.

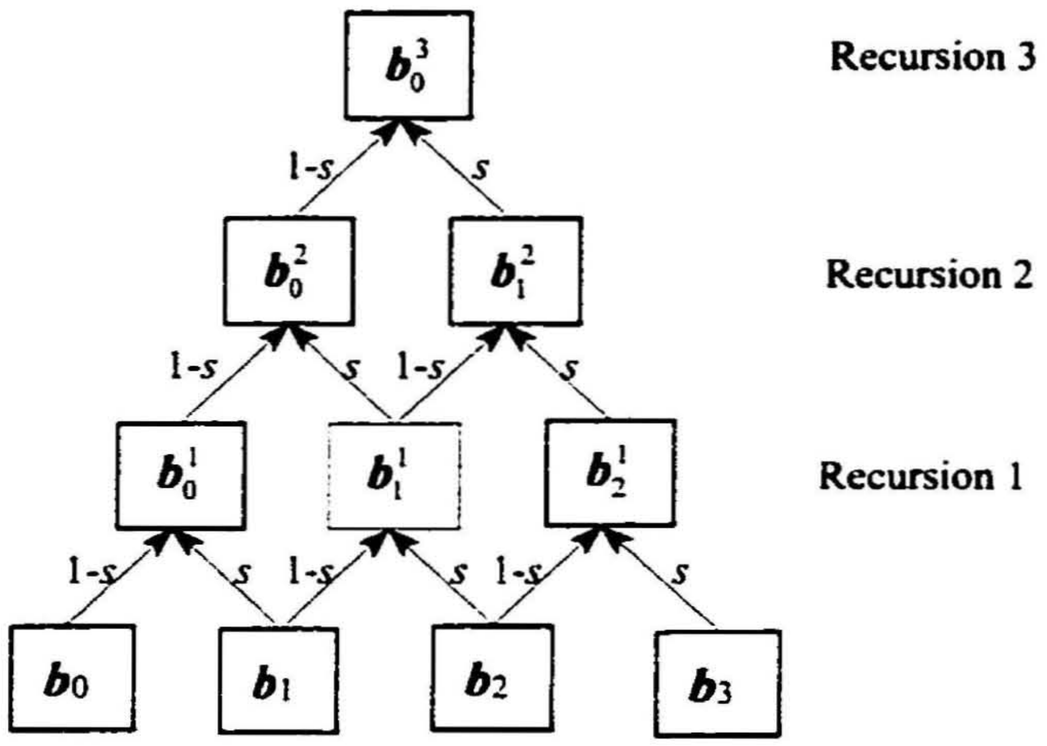


Figure 4.10: The recursion pattern of the *de Casteljau algorithm*

4.5.2 Nonlinear Curve Design as an Alternative to One-input Fuzzy PID Elements

Using the above Bézier definition, it is possible now to generate a SISO control output as an alternative to the solutions described for NLFLC-III systems. The control objectives again are to realize the local non-linearity tuning at the two control points defined in this chapter. First consider four Bézier points in the $\Re^2$ space for defining the Bézier curve as:

$$\{\boldsymbol{b_r}\} = \{\boldsymbol{x_r}, \boldsymbol{y_r}\}$$

where $r = 0, 1, 2, 3$. The vectors x and y are the horizontal and vertical co-ordinates of the Bézier points. Using the *de Casteljau algorithm* the Bézier curve can be represented in the following recursive form. The recursion pattern is shown in Figure 4.10.

Recursion 1 :

$$\boldsymbol{b_0^1}(s) = (1-s)\boldsymbol{b_0} + s\boldsymbol{b_1}$$

$$\boldsymbol{b_1^1}(s) = (1-s)\boldsymbol{b_1} + s\boldsymbol{b_2}$$

$$\boldsymbol{b_2^1}(s) = (1-s)\boldsymbol{b_2} + s\boldsymbol{b_3}$$

Recursion 2 :

$$\boldsymbol{b_0^2}(s) = (1-s)\boldsymbol{b_0^1} + s\boldsymbol{b_1^1}$$

$$\boldsymbol{b_1^2}(s) = (1-s)\boldsymbol{b_1^1} + s\boldsymbol{b_2^1}$$

Recursion 3 :

$$\boldsymbol{b_0^3}(s) = (1-s)\boldsymbol{b_0^2} + s\boldsymbol{b_1^2}$$

The degree-3 Bézier curve is therefore;

$$\boldsymbol{X}(s) = \boldsymbol{b_0^3}(s) \tag{4.25}$$

The objective is to simplify the above Bézier curve in (4.25) to generate the control curve for the range $\hat{e}_1 \in [0.1]$ while satisfying the necessary properties shown above. The range $\hat{e}_1 \in [-1.0]$ can be easily obtained by imposing the negative symmetrical property.

The end points are fixed to satisfy the properties P2 and P3 in section 4.2.2 and therefore assign.

$$\boldsymbol{b_0} = \{0,0\}^T \text{ and } \boldsymbol{b_3} = \{1,1\}^T. \tag{4.26}$$

Let,

$$\boldsymbol{b_1} = \{s_{1.x}, s_{1.y}\}^T \text{ and } \boldsymbol{b_2} = \{s_{2.x}, s_{2.y}\}^T. \tag{4.27}$$

The co-ordinate values $\{s_{1.x}, s_{1.y}, s_{2.x}, s_{2.y}\}$ are the non-linearity tuning parameters for the control. Assign,

$$\boldsymbol{X}(s) = \{\hat{e}_1, \hat{u}\} \tag{4.28}$$

Substituting the co-ordinates in (4.26) and (4.27) to the three recursions in (4.25), the error mapping can be obtained as:

$$\hat{e}_1 = 3s(1-s)^2 s_{1.x} + 3s^2(1-s)s_{2.x} + s^3 \tag{4.29-a}$$

$$\hat{u} = 3s(1-s)^2 s_{1.y} + 3s^2(1-s)s_{2.y} + s^3 \tag{4.29-b}$$

where $s \in [0,1]$ for the output to be within the range $\hat{u} \in [0,1]$. Figure 4.11 shows the functional procedure of obtaining the error mapping "$\hat{e}_1$ to $\hat{u}$". For a given set of tuning parameters, the controller output is calculated from the mapping "$\hat{e}_1$ to s" followed by "s to $\hat{u}$". During the control, the input error value is known for each sampling instance. This requires the solving of the cubical polynomial of $\hat{e}_1 = f(s)$ in (4.29-a). From the Bézier curve properties it can be easily inferred that for any normalized error input value, there is one unique solution for s that is always guaranteed to be within the range [0,1]. Also

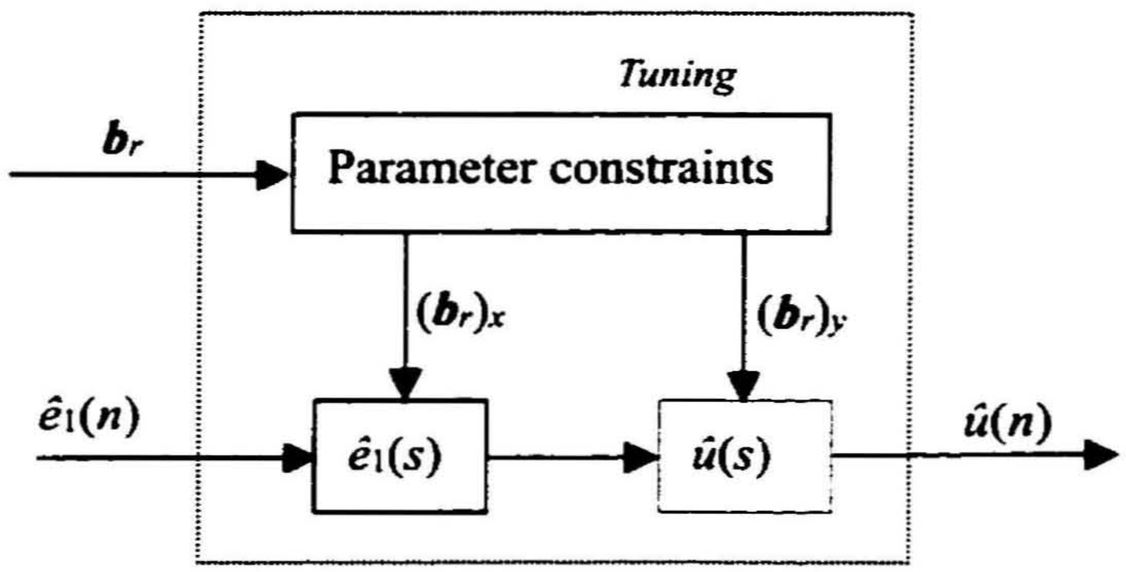


Figure 4.11: Parameterized calculation of the normalized control action

the monotonic property of this control function can be found in [33]. Therefore it can be concluded that the control action generation using Bézier polynomials can preserve all the desired properties required for nonlinear PID control. Alternatively it can be said that the Bézier curve can be designed to obtain the same properties of a fuzzy control output. The interpolating characteristic is common for both fuzzy [104] and spline based functions and therefore both the systems can share the same universal approximation property.

4.5.3 Performance Evaluation

The performance measures defined above in section 4.3 can be used to evaluate the nonlinearity of the alternative control action. Although the mapping is a little complex due to parameterization, the control curve is highly continuous (up to C^2). The two slope angles are given by:

$$\theta_0 = \left(\frac{s_{1,y}}{s_{1,x}}\right) \quad \theta_1 = \left(\frac{s_{2,y}}{s_{2,x}}\right) \tag{4.30}$$

It is very clear now the slopes can be individually varied by moving the control points $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$ and therefore the whole admissible space can be realized. From (4.30) it can be seen that for a given θ_0 and θ_1 either x or y value is redundant, and therefore effectively the tuning can be performed only by two tuning parameters. The tuning diagram drawn for these conditions are shown in Figure 4.12. The point **O** provides the perfect linear conditions. Therefore under these circumstances the alternative nonlinear control shows the 100% performance having $NVI = 1$, $LAI = 1$ and $Availability = 4$.

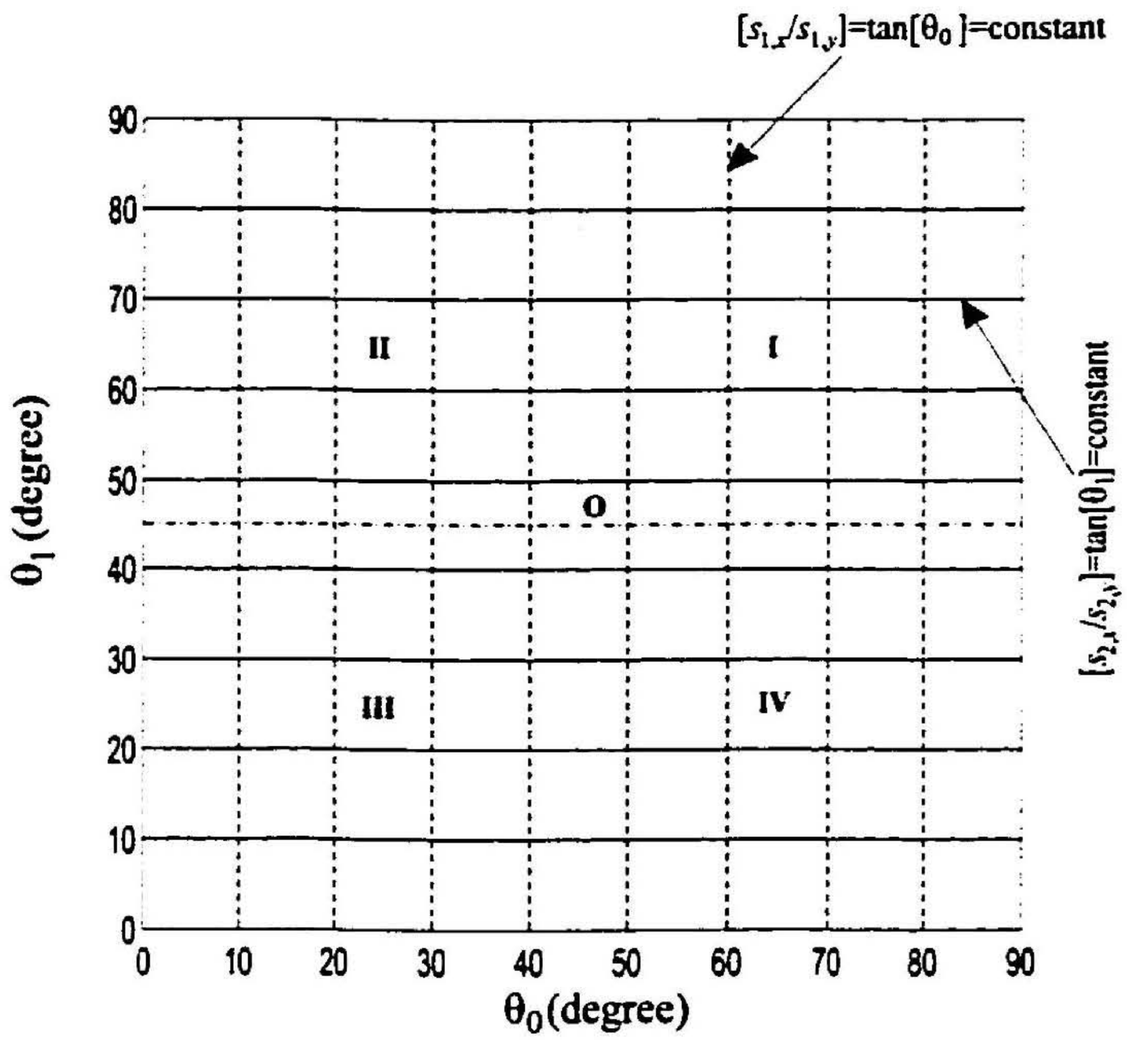


Figure 4.12: Non-linearity variation diagram of the control curve based on Bézier function

4.5.4 Realization of More Local Control Points

So far including the fuzzy analysis, the non-linearity has been examined only for the two points (near zero and near maximum error conditions). The non-linearity tuning was confined to only two slope values. If a third point is selected, the local control properties can be again obtained by changing the slope values as shown in Figure 4.13. In the figure the third point is arbitrarily denoted by the Bézier point $\boldsymbol{b_3}$. Only the positive region is shown for convenience. The control curve is now defined by six Bézier points and denoted by $\boldsymbol{b_r}$ with $r = 0, 1, \ldots, 6$. Consider two subdivisions shown by the regions denoted by the envelops of $\boldsymbol{b_0 b_y b_3 b_x}$ and $\boldsymbol{b_3 b_{y1} b_6 b_{x1}}$ as shown in Figure 4.13. The boundary conditions to the control curve will set $\boldsymbol{b_0}$= $\{0, 0\}$ and $\boldsymbol{b_6}$= $\{1, 1\}$. The third Bézier point is predetermined and therefore assign $\boldsymbol{b_3}$= $\{(\hat{e}_1)_I . (\hat{u})_I\}$. For realizing the C^1 continuity at the intermediate Bézier point, set

$$\overline{\boldsymbol{b_3 b_2}} = -\overline{\boldsymbol{b_3 b_4}}. \tag{4.31}$$

In other words the two Bézier points, $\boldsymbol{b_2}$ and $\boldsymbol{b_4}$, are symmetrically aligned with $\boldsymbol{b_3}$. The tuning of each region is similar to two-point control. By varying the positions of the points $\boldsymbol{b_1}$, $\boldsymbol{b_2}$, $\boldsymbol{b_4}$ and $\boldsymbol{b_5}$ the local control tuning at the three points are obtained. Due the continuity

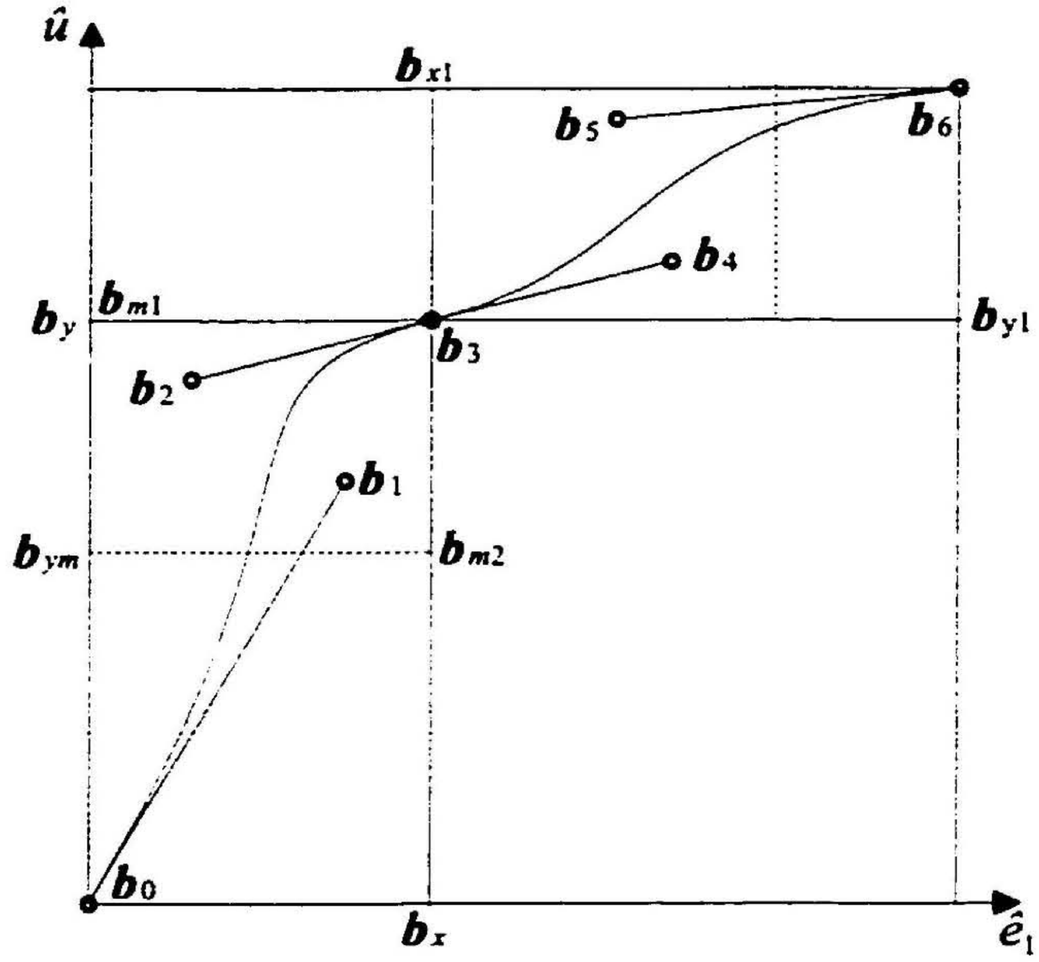


Figure 4.13: Realization of local control at three control points

constraint given in (4.31), the number of angles of interest is three. The permissible regions of the Bézier points can be described as:

$$\boldsymbol{b_1} \in \text{area enclosed by the points } (\boldsymbol{b_0 b_y b_3 b_x})$$

$$\boldsymbol{b_5} \in \text{area enclosed by the points } (\boldsymbol{b_3 b_{y1} b_6 b_{x1}})$$

$$\boldsymbol{b_2} \in \text{area enclosed by the points } (\boldsymbol{b_3 b_{m1} b_{m2} b_{m3}})$$

where.

$$|\overline{\boldsymbol{b_3 b_{m1}}}| = \min\left(|\overline{\boldsymbol{b_3 b_y}}|, |\overline{\boldsymbol{b_3 b_{y1}}}|\right)$$

$$|\overline{\boldsymbol{b_3 b_{m3}}}| = \min\left(|\overline{\boldsymbol{b_3 b_x}}|, |\overline{\boldsymbol{b_3 b_{x1}}}|\right)$$

$$\overline{\boldsymbol{b_y b_{m2}}} = \overline{\boldsymbol{b_3 b_{m3}}}.$$

The permissible region for the point $\boldsymbol{b_4}$ can be similarly obtain by imposing the continuity constrain in (4.31). These regions are indicated in Figure 4.13. The control curve can be generated recursively as shown in Figure 4.14. However the computation can be simplified by generating the two curves separately by using the two intermediate functions $\boldsymbol{b_0^3}(s)$ and $\boldsymbol{b_3^3}(s)$ (Figure 4.14). The subdivision will change the range for the parameter s. Therefore consider a *local parameter* q [120] for the interval $c_1 \leq s \leq c_2$:

$$q = (s - c_1)/(c_2 - c_1) \tag{4.32}$$

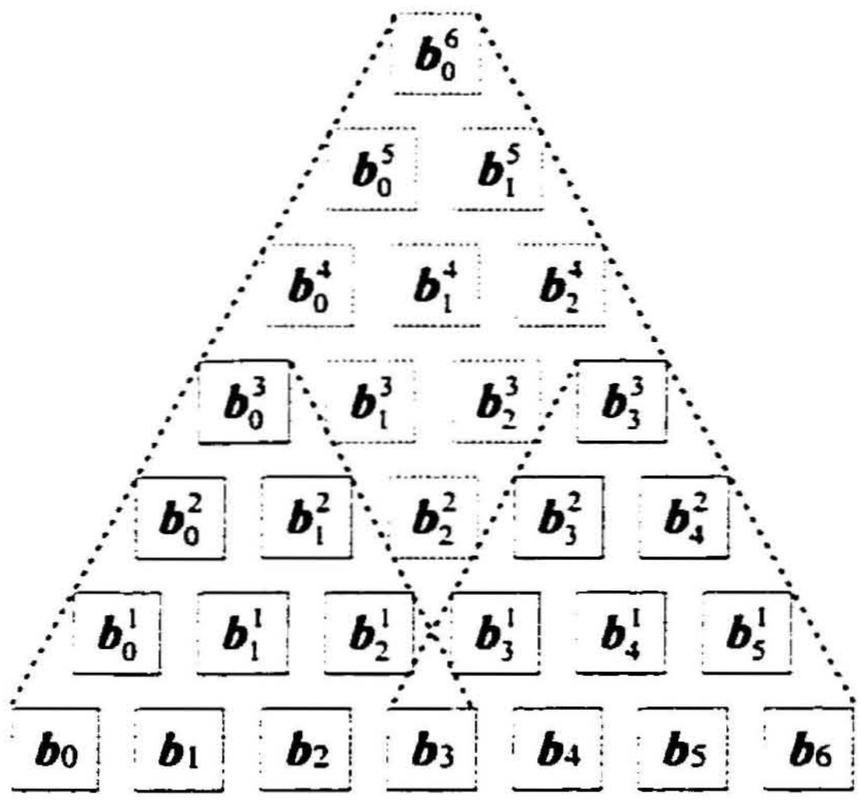


Figure 4.14: Recursion pattern for two subdivisions

The interval values are calculated from (4.29-a) using the third control point. Then the Bézier curves corresponding to respective subdivisions are calculated using the nonlinear mappings. "$\hat{e}_1$ to q" followed by "q to $\hat{u}$".

4.6 Summary

In this chapter the FLC systems developed in Chapter 3 are analyzed further for evaluating the non-linearity features for fuzzy control. The non-linearity has been identified at two locally selected control points in the control action. This identification helps to reduce the parameters required for the fuzzy system to generate the desired types of nonlinear control actions. Also, it avoids the unnecessary guessing of membership functions for fuzzy control variables. The slopes of the control surface at the chosen control points are selected to vary the ANG terms. Thus the non-linearity features required by a given fuzzy system have been developed. Mainly three performance measures have been identified for assessing and ranking different fuzzy systems and reasoning schemes. These measures were identified with respect to the two-point control. The new ranking system can compare not only the reasoning schemes, but also the different knowledge base arrangements. As an example the controllers labelled by C1–C3 use different rule base and membership arrangements within the same class of fuzzy reasoning. The overlapping membership functions for the input variable is a necessary

condition for preserving the rule base properties. Therefore the input membership variables are more constrained than the output membership variables and as a result the non-linearity tuning by using output membership functions parameters are more effective than using input membership parameters. However in coupled rule bases, the independent tuning can be accomplished only by varying the parameters of input fuzzy variables. Therefore NLFLC-I system has limitations to obtain higher variation to the non-linearity as compared to NLFLC-IIIA system. The NLFLC-IIIB system with fully normalized conditions has not been able to produce better non-linearity than the NLFLC-IIIA system. This evidence demonstrates that different membership arrangements can produce a significant difference to the performance in fuzzy control.

The reasoning schemes have a major impact in the fuzzy control. Most of the inference types can be eliminated at the first instance by observing the desired properties of the control action. As an example, some reasoning schemes are unable to produce continuous and monotonic control actions with respect to the error state variables. Those unsuitable reasoning systems have been eliminated in the previous studies [20, 52]. In this evaluation, the PSG reasoning was identified as the most in-efficient fuzzy inference for fuzzy control. It is possible to improve the performance of PSG based fuzzy systems by including more tuning variables and parameters. Although TSK based system has lower NVI value, the TSK representation allows greater flexibility in choosing the desired non-linearity. Also it is possible to change the output functions in the TSK based systems for reaching greater performance.

The research has enabled the identification of an alternative nonlinear control system using Bézier polynomials. Those functions are designed primarily to interpolate data points for generating smooth geometric surfaces and curves. The same interpolating characteristics in the fuzzy systems enable to use of fewer rules to obtain higher degree of non linearity for control than the traditional expert systems. However the Bézier polynomials have better geometrical features than fuzzy systems and as a result the geometrically based performance evaluation scheme showed the highest ranking in this evaluation.

Chapter 5

Development of New Linear PID Tuning Rules

5.1 Introduction

The previous two chapters identified different fuzzy systems and the tuning basis for fuzzy controllers. The tuning basis has enabled simplification of the multi-dimensional fuzzy controller design to two-level tuning problem. The two tuning levels are inter-related to each other. For example, it is possible to fix the knowledge base parameters of the first level tuning and to perform the second level of tuning to identify the linear apparent gains. This is the most common criteria used in many applications [9, 97, 80] where the fuzzy controller parameters are arbitrarily fixed using predetermined fuzzy sub-sets and then using either an exhaustive numerical search or trial and error computer simulations the linear gains are chosen. However, the tuning of apparent linear gain (ALG) terms are similar to linear PID gains of a conventional controller. In this chapter the author performes time-domain based analytical procedure to develop new PID tuning rules for conventional controllers. There are three primary motivations for this research. First, the study of linear PID tuning provides a better foundation and sufficient knowledge to extend the analysis for fuzzy PID controller tuning. Secondly, the properly defined linear PID controller can be used as a benchmark for obtaining improved performance of the fuzzy controllers. Thirdly, the available linear PID tuning rules have some drawbacks as explained in the following section.

In this chapter a time-domain analysis and thus the design of PID controllers for first-order process models are described. The PID analysis includes three types of first-order plant models:

(a) zero or negligible normalized time delay

(b) low to medium normalized time delay and

(c) large to very large normalized time delay.

The mathematical analysis proves that the optimum PID controller for plants having zero or negligible time delay is a PI controller with zero derivative action. For zero time delay plants the PI terms based on the actuator's capacity and set-point overshoot are explicitly derived. The analysis is then extended to processes having low to medium time delay. In this case a new PID tuning scheme is proposed. It will be proven analytically that when the time delay becomes large the derivative action of the PID controller has negligible effect on the transient response. Using a separate time response analysis. a new PI tuning scheme for large normalized time delay is then derived. This PI design is based on user defined two points on the response curve and allows higher flexibility of the design. The proposed tuning rules are capable of accommodating the actuator saturation limits. This distinguishing feature provides a PID controller design that prevents the integral wind-up of the process associated with the actuator saturation. Numerical studies for higher order processes having monotonic open-loop characteristics are shown. The performance is compared with other commonly available tuning rules. With the new tuning rules improved performance is observed and the rules have the capability to cover time delays ranging from zero to any higher value.

5.2 Overview

Development of the PID controllers is based on many years of engineering innovation [121]. As a result of extensive investigations to devise ways of choosing optimum controller settings for the PID controllers. Ziegler and Nichols showed that optimum controller settings could be estimated using open and closed-loop tests on the plant [122, 123]. The method is referred to as **ZN** rules. The ZN settings usually experience excessive overshoot of the plant response and also difficult to tune plants that have a relatively longer time delay. With the ease of

computations, the numerical optimization and curve fitting techniques have later become significant in devising formulae for PI and PID parameters. The error integral criteria are the most common for such optimizations [118, 124, 125, 126]. Most of these tuning schemes are valid only for a limited range of normalized time delay problems. More recently Khan and Lehman [127] have used extensive simulations and data fittings to obtain PI tuning formulae. The significant attraction of the latter method is that a single set of expressions for PI terms has been derived to satisfy the normalized time delay ranging from 0.2 to 20. In all these optimization approaches, the PID parameters are arbitrarily expressed as functions of process terms, typically in terms of normalized time delay (ratio of process time delay to time constant) and first-order time constant. A significant development of ZN based tuning was shown by Hang *et al.* [128]. They have critically examined the ZN settings and by introducing an additional variable (set-point weighting) the excessive overshoot in the original ZN settings have been reduced while preserving the same load disturbance characteristics. In development of the former refined ZN (**RZN**) rules the ZN PI settings have been completely revised to cope with relatively long time delay problems. Introducing adaptive or variable set-point weighting, RZN based response has been recently further enhanced [132]. The RZN method shows excellent load disturbance characteristics, but it cannot be used for very long time delay processes.

Frequency domain analysis and development of PID tuning are reported in [129, 130, 131, 132]. These methods have the advantage of obtaining different PID parameters by selecting desired points in the Nyquist curve [129, 132] or by using user specified phase and gain margins [130]. The latter is called as Gain-Phase-Margin (**GPM**) tuning. Therefore these methods possess some form of flexibility to obtain different PID settings for a given process specification. Pemberton [133] used time-domain analysis to yield PID tuning for first order and over-damped second order plants with time delay. Due to the two term approximations of the time series expansion, the method was unsuccessful for plants having a normalized time delay greater than 0.5. The internal model control (**IMC**) approach to design PID parameters is shown in [134]. The controller having the pole zero cancellation was derived by taking a low-pass filter together with the inverse function of the estimated model excluding the process-lag components and right-hand process zeros. The IMC design simplifies the PID settings to a single parameter, which is directly related to the proportional gain and

therefore the response speed. Therefore IMC design has the advantage of obtaining PID parameters to accommodate the actuator saturation [135]. Also the PID formula for zero time delay simplifies to the pole zero cancellation PI controller with zero derivative gain. Due to the first-order Padè approximation of the exponent term, the applicability of the PID formula is limited to relatively short time delay problems. Hang *et al.*, [136] have presented a comparative study of the IMC and GPM approaches. The study concluded the IMC design has a lower flexibility in terms of robustness.

Based on this literature review, this chapter attempts to address three main issues related to PID tuning. The first is related to PID control of processes modeled with zero or negligible time delays. The numerical optimization in [137] and the IMC-PID tuning parameters [134] suggest that for zero time delay processes the best selection is a PI controller. While the majority of tuning rules such as ZN, RZN, GPM and error integral optimized rules are not applicable, the IMC design simplifies the PID settings to a pole zero cancellation PI controller for zero time delay plants. However, a time domain based closed form mathematical analysis of PID controlled response for zero time delay has not so far been reported in the literature. In the first part of this chapter the first-order plants with zero time delay is analyzed and the necessary PID controller settings are analytically obtained. The second issue is related to PID parameter selection to accommodate the maximum capacity or gain of the actuator while avoiding the hazard of integral wind-up. The applicability of tight PID control based on normalized time delay and the normalized process gain was discussed in [138]. The importance of the PID design to limit the overshoot of controller signal has been argued at length [135, 139]. The analysis is then extended to deduce new PID tuning scheme applicable for short and medium-long time delay. The PID controller gains are selected based on the actuator's saturation. The practical limits for PI and PID control are also deduced. The third and final issue is related to plants having large normalized time delay. To accommodate them, a new PI tuning scheme is analytically derived and the tuning is based on user defined two points in the transient response curve. Therefore this chapter intends to provide a complete analysis of PID tuning for first-order plant models covering the complete range of normalized time delay.

5.2.1 Controller/Process Specifications

Let the first order plant model with time delay t_d, time constant T and steady state gain k be given by the transfer function,

$$G(s) = \frac{k \exp(-t_d s)}{Ts + 1}. \tag{5.1}$$

The experimental identification of the three terms using many techniques is well described in [140]. In this analysis the linear controller is defined in the continuous form for time-domain analysis and therefore some of the PID controller specifications defined in Chapter 3 are redefined for convenience. Using the conventional notations, the PID controller signal (refer to Figure 3.1) at the time t can be described in the following form:

$$u(t) = k_c \left(e(t) + T_d \frac{de(t)}{dt} + \frac{1}{T_i} \int_0^t e(t)dt \right). \tag{5.2}$$

Where the controller gain is k_c, derivative time constant is T_d and the integral rate is T_i. The feed back error signal $e(t)$ = reference signal $r(t)-$ response signal $y(t)$. In the practical PID controller with a derivative filter, the controller output is given by,

$$u(t) = k_c \left(e(t) - T_d \frac{dy_f(t)}{dt} + \frac{1}{T_i} \int_0^t e(t)dt \right), \tag{5.3}$$

$$\frac{T_d}{N} \frac{dy_f(t)}{dt} = y(t) - y_f(t).$$

The N is an arbitrary number associated with the derivative filter. For $N > 10$ the same PID parameter values obtained from (5.2) can be implemented with the derivative filter without any significant difference [118]. The other practical forms generally implemented in commercial controllers can be obtained from (5.3) as described in [140]. Using the gain notations used in Chapter 3 (5.2) is rewritten in the form,

$$u(t) = K_{\mathrm{P}} e(t) + K_{\mathrm{D}} \frac{de(t)}{dt} + K_{\mathrm{I}} \int_0^t e(t)dt. \tag{5.4}$$

where the linear proportional, integral and derivative gains are respectively given by $K_{\mathrm{P}} = k_c$, $K_{\mathrm{I}} = k_c/T_i$ and $K_{\mathrm{D}} = k_c T_d$.

5.3 Analysis I: For Zero Time Delay

The Laplace form of (5.4) can be expressed with the initial error signal $e(0)$ by,

$$U(s) = K_{\mathrm{P}} E(s) + K_{\mathrm{D}} \left(sE(s) - e(0) \right) + K_{\mathrm{I}} E(s)/s. \tag{5.5}$$

The Laplacian form of the plant response of a unity feed back cascade PID controller system with no external disturbance is given by,

$$Y(s) = \frac{1}{s + (K_{\mathrm{D}}s^2 + K_{\mathrm{P}}s + K_{\mathrm{I}})\,G(s)} \left[\left(K_{\mathrm{D}}s^2 + K_{\mathrm{P}}s + K_{\mathrm{I}} \right) G(s)R(s) - K_{\mathrm{D}}e(0)sG(s) \right]. \tag{5.6}$$

For unit step response $R(s) = 1/s$. Substituting (5.1) with $t_d = 0$ to (5.6), the output can be simplified to,

$$Y(s) = \frac{1}{(K_3 + T)s^2 + (K_1 + 1)s + K_2} \left[K_1 + \frac{K_2}{s} \right], \tag{5.7}$$

where the normalized PID gain terms are expressed as, $K_1 = kK_{\mathrm{P}}$, $K_2 = kK_{\mathrm{I}}$ and $K_3 = kK_{\mathrm{D}}$. The main objective in this exercise is to relate the PID parameters to the closed-loop response behaviour. Therefore expressions are deduced for the rise time and overshoot (or undershoot) of (5.7). The derivation is based on the nature and positions of the closed-loop poles in the *s-plane*. Details of this derivation are given in Appendix C.

5.3.1 Rise Time and Peak Overshoot in the Transient Response

Case I: The closed-loop poles are real and distinct

By examining the real closed-loop poles, a general relationship between the normalized gains was established and is given by,

$$K_2 = \frac{K_1}{(K_3 + T)} \left(1 + \beta \frac{(K_1 - 1)^2}{4K_1} \right). \tag{5.8}$$

where, β is a positive real number and its range has been constrained to be within $0 \leq \beta < 1$ for the closed-loop poles to be real and distinct. Also within this range the peak overshoot, when $K_1 > 1$, is always positive. Case II covers $\beta = 1$, which corresponds to equal roots. Also, it can be easily proved that when $\beta = 0$ the peak overshoot of the response is zero.

Case I-a : $K_1 > 0$ and $0 < \beta < 1$

The rise time (T_r) based on 0-100% response and peak overshoot (OS) are given by,

$$T_r = \frac{(K_3 + T)}{(K_1 - 1)\sqrt{1 - \beta}} \ln \left[\frac{1 + \sqrt{1 - \beta}}{1 - \sqrt{1 - \beta}} \right] \tag{5.9}$$

$$OS = \frac{(1 + \sqrt{1 - \beta})}{(\gamma - \sqrt{1 - \beta})} \left[\frac{(\gamma - \sqrt{1 - \beta})(1 - \sqrt{1 - \beta})}{(\gamma + \sqrt{1 - \beta})(1 + \sqrt{1 - \beta})} \right]^{\frac{\gamma + \sqrt{1 - \beta}}{2\sqrt{1 - \beta}}}, \tag{5.10}$$

respectively where $\gamma = (K_1 + 1)/(K_1 - 1)$.

Case I-b : $\beta = 0$

In this case the overshoot $OS = 0$ and T_r based on 10-90% response is given by,

$$T_r = \left[\frac{(K_3 + T)}{(K_1 + 1)}\right] \ln 9. \tag{5.11}$$

Case II: Closed-loop poles are real and equal

This case refers to the gain relationship (5.8) with $\beta = 1$.

Case II-a : $K_1 > 1$

The T_r based on 0-100% response and OS are given by.

$$T_r = \left[\frac{2(K_3 + T)}{(K_1 - 1)}\right], \tag{5.12}$$

$$OS = \frac{(K_1 - 1)}{(K_1 + 1)} \exp\left(\frac{-2K_1}{K_1 - 1}\right). \tag{5.13}$$

From (5.13) it is clear that when $K_1 < 1$ the system shows stable over-damped response.

Case II-b : $K_1 = 1$ The critically damped response has zero overshoot and the rise time based on 10-90% is given by.

$$T_r = (K_3 + T) \ln(9). \tag{5.14}$$

Case III: Closed-loop poles are complex with negative real parts

This case is realized when the relative damping factor ζ of the closed-loop system is chosen within $0 < \zeta < 1$, while satisfying the normalized gain relationship given by,

$$K_2 = \frac{(K_1 + 1)^2}{4\zeta^2(K_3 + T)}. \tag{5.15}$$

The peak overshoot of the under damped response can be shown as,

$$OS = \sqrt{(1 + L^2)(1 - \zeta^2)} \exp\left(\frac{-\zeta(\psi + \vartheta)}{\sqrt{1 - \zeta^2}}\right), \tag{5.16}$$

where $L = \frac{\zeta}{\sqrt{1-\zeta^2}} \frac{(K_1-1)}{(K_1+1)}$, $\psi = \arctan \frac{\sqrt{1-\zeta^2}}{\zeta}$, and $\vartheta = \arctan(1/L)$ for $K_1 \geq 1$ or $\vartheta = \pi - \arctan(1/|L|)$ for $K_1 < 1$. It can also be shown that OS is always positive when ζ is within 0 and 1. Therefore the rise time based on 0-100% can be shown as,

$$T_r = \frac{\zeta(K_3 + T)\omega}{(K_1 + 1)\sqrt{1 - \zeta^2}}. \tag{5.17}$$

5.3.2 Optimal Tuning Law for Processes Having Zero Time Delay

By observing the rise time given for all the cases above, we can clearly see that the addition of the derivative term, which corresponds to K_3 in (5.9), (5.11), (5.12), (5.14), and (5.17) slows down the transient response. Also the overshoot in all cases can be controlled by the normalized proportional gain while choosing the integral gain satisfying the gain relations ((4.8) or (4.15)) correspond to each case. Therefore it can be concluded that for the optimum design of PID controller for any first-order process model with zero lag-time, the derivative gain should be zero. The theoretical model with the PI controller has an infinite gain margin and the system can be operated with any value of a controller gain. The upper saturation level of the actuator gain can determine the maximum gain of the controller. Therefore the maximum controller signal $u_{\max}$ per unit step response has been derived and is given as follows.

Case A : Closed-loop poles are real or are complex with $K_1(2\zeta - 1) \geq 1$

$$u_{\max} = u(0) = K_P.$$

Case B : Closed-loop poles are complex with $K_1(2\zeta - 1) < 1$

The peak controller signal corresponding to the time,

$$t_p = \frac{1}{(K_1 + 1)} \frac{2\zeta T}{\sqrt{1-\zeta^2}} \arctan \left[\left(\frac{(K_1 + 1)^2 - (2K_1\zeta)^2}{4K_1^2(1-\zeta^2) - (K_1 - 1)^2} \right) \frac{\sqrt{1-\zeta^2}}{\zeta} \right] \tag{5.18}$$

and $u_{\max} = K_P e(t_p) + K_I \int_0^{t_p} e(t)dt$. This can be simplified as,

$$u_{\max} = \frac{1}{k} \left(\frac{K_1 + 1}{2\zeta} - \frac{2K_1\zeta}{K_1 + 1} \right) \exp \left(-\frac{(K_1 + 1)}{2\zeta T} t_p \right) - \frac{1}{k}. \tag{5.19}$$

The above cases hold if $K_1 > 1$. With 5% or less overshoot, Case B occurs when the normalized proportional gain is closer to 1. Also the value given in (5.19) is not significantly greater than K_P. Therefore in most cases the proportional gain K_P can have any value as high as the actuator's upper limit of saturation. This condition is implicitly stated in the IMC design [134]. The remaining integral gain can be selected by choosing a desired level of overshoot of response. Let OS_d denote the desired peak overshoot level. The optimal tuning law can be stated as follows.

1. *With zero time delay, PI controller is optimum where* $K_3 = 0$ *and therefore* $K_D = 0$.

2. *Select the proportional gain based on the actuator saturation. If the actuator's upper limit of saturation is* U_u, *then select* $(K_P)_{max}$ *by assigning* $u_{max} = U_u$. *This allows the fastest rise time.*

3. *If the* $K_1 \leq 1$, *select the necessary relative damping for the given* OS_d *from (5.16) and thus compute* K_2 *using (5.15).*

4. *If* $K_1 > 1$ *first assume the closed-loop poles are real and equal and compute the peak overshoot using (5.13). If the computed value is greater than* OS_d *then select* β *for the given* OS_d *from (5.10) and then compute* K_2 *using (5.8). Otherwise choose relative damping and compute as in (3).*

Example E1

Assume the plant parameters of (5.1) as $k = 2$, $T = 1$ and the time delay $t_d = 0$. Consider two cases with actuator limits $U_u = 1.5$ and $U_u = 10$ units. Assume the design of a PID controller to satisfy 5% overshoot of unit set-point response ($OS_d = 0.05$). From the above results the PI is optimal and therefore $K_3 = 0$. The proportional gain is limited by the actuator limits and therefore consider two cases corresponding to $u_{max} \leq 1.5$ and $u_{max} \leq 10$. Assume equal conditions to illustrate the tuning.

Case I : $u_{max} = 1.5$

First assume $K_P = (K_P)_{max} = 1.5$ and therefore $K_1 = 3$. The overshoot OS is first evaluated when the closed-loop poles are equal. Using (5.13), OS is 0.025. Since $OS < OS_d$ the specified OS_d can be achieved only when the closed-loop poles are complex. Using (5.16) with $OS = 0.05$, the corresponding relative damping (ζ) is 0.9014. $K_1(2\zeta - 1) = 2.41$ and therefore the maximum controller signal falls to the Case A and the initial assumption is correct. From (5.15), $K_2 = 4.923$. The PI gains are therefore, $K_P = 1.5$ and $K_I = 2.461$.

Case II : $u_{max} = 10$

Again assume $K_P = 10$ and $K_1 = 20$. Similarly by using (5.13) OS is 0.112. Since $OS > OS_d$ the specified overshoot occurs only when the closed-loop poles are real and

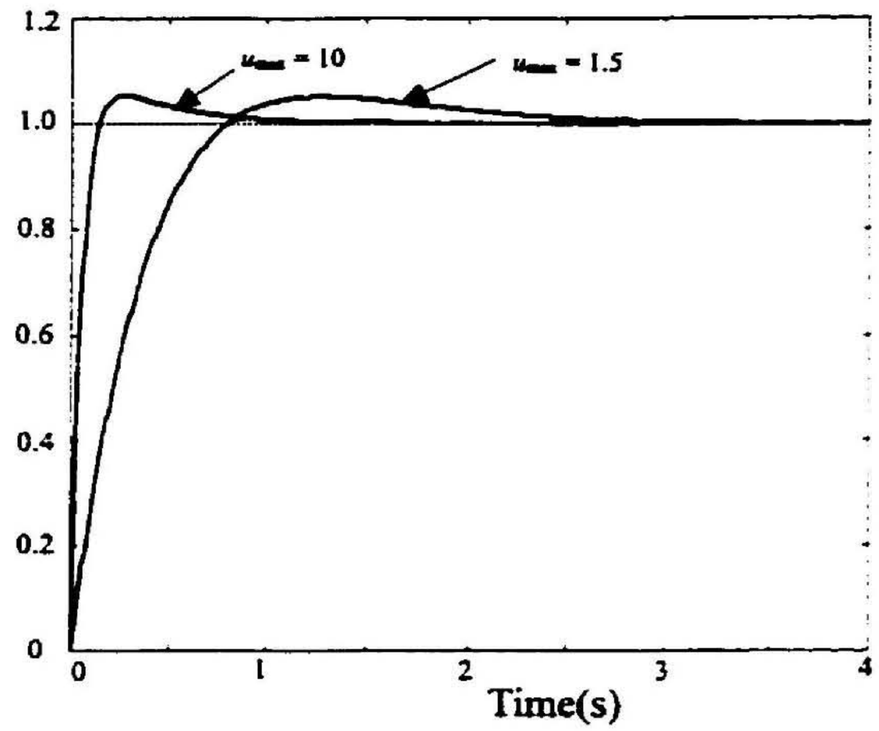


Figure 5.1: Unit step response of Example E1.

distinct. Therefore $u_{max} = K_P$. Using (5.10) with $OS = 0.05$, the value for $\beta = 0.371$ is obtained. From (5.3) $K_2 = 53.52$. The PI gains are therefore, $K_P = 10$ and $K_I = 26.76$. Figure 5.1 shows the response curves of this example.

5.4 Analysis II: For Processes with Measurable Time Delay

The exact time-domain analysis of $G(s)$ with the time exponent is a complex mathematical task, owing to the non-linear exponential term in the transfer function. The Padè approximation [134] or truncated time series approximation [133] of the exponent term results in losing significant poles that exist at distances from the imaginary axis of the s-plane. This section presents a systematic approach to obtain PID tuning rules using time-domain analysis.

5.4.1 Ultimate Gain and Ultimate Frequency

The definition of ultimate gain and ultimate period refers to the continuous oscillation of the closed-loop response with constant amplitude when the process is controlled only through proportional control. In this section we simplify the expressions relating ultimate gain and ultimate period to obtain concise expressions that would be helpful in determining the process terms. With zero derivative and integral actions, the closed-loop characteristic equation with

the plant model of (5.1) is given by,

$$Ts + 1 + K_c \exp(-st_d) = 0. \tag{5.20}$$

The normalized gain, $K_c = k_c k$ where k_c is the proportional controller gain when both integral and derivative terms are zero. The root locus of (5.20) has infinite number of root-locus branches; the primary branch which lies between $-j\pi$ and $j\pi$ is the most important [141] for the response behaviour. By using the fundamentals of root-locus construction, it can be shown that the break point gain (K_b) of the inner locus is given by,

$$K_b = (T/t_d) \exp(-1 - t_d/T). \tag{5.21}$$

Also, the ultimate normalized gain (K_u) at which the inner locus cuts the imaginary axis and the corresponding frequency (ω_u) are given by,

$$K_u = \sqrt{1 + T^2\omega_u^2}, \tag{5.22}$$

$$\omega_u - t_d - \arcsin(1/K_u) = \pi/2. \tag{5.23}$$

By simplifying (5.22) and (5.23) the expressions for the two terms can be described by:

$$T = \frac{t_u}{2\pi}\sqrt{k^2k_u^2 - 1} \text{ and } t_d = \frac{t_u}{2\pi}\left(\frac{\pi}{2} + \arcsin\left(\frac{1}{kk_u}\right)\right). \tag{5.24}$$

where k_u and t_u are the ultimate gain and ultimate period of the process system respectively. Using a feedback relay experiment, an estimate of k_u and t_u can be obtained as explained in [140]. By estimating the process gain (k) using an open-loop test and by using (5.24) estimates for T and t_d can be made.

The ultimate gain provides the margin of safety, where as the actual operating gain (proportional) of a PID controller is very much smaller than k_u. On the other hand the break point gain in (5.21) is too small and the response would take a longer time to reach the set-point. Therefore the operating proportional gain should be in between these two limits.

5.4.2 PI and PID Tuning Analysis

The excessive overshoot of ZN tuning and its inapplicability to long normalized time delay are mainly due to the overestimate of gains and the wind-up associated with the integral action. This in fact is not a serious problem when the time delay is negligible or zero. The limiting

values of the gain terms very much depend on the actuator's upper level of saturation. Also, the allowable overshoot of the controller signal is limited in most of industrial problems [139]. Therefore the main criterion of the performance is considered to be the overshoot.

The evaluation of roots of the general characteristic equation with the three-term PID controller is a difficult mathematical task. In order to simplify the analysis some of the results given in section 5.2 are assumed. It was shown in the Section 5.2 that when the closed-loop poles are complex the set-point response always experiences overshoot. Since this task is based on minimum overshoot, assume the gain relationship given in (5.8) which corresponds to the case where the closed-loop poles are real and distinct. Further it has been observed that when the overshoot control parameter β is set to zero, the response of the plant with no time delay has zero overshoot. Therefore in this exercise β is set to zero. The gain relationship of (5.8) now simplifies to,

$$K_2 = \frac{K_1}{(K_3 + T)}. \tag{5.25}$$

PI tuning

For the PI controller set $K_3 = 0$. Equation (5.25) simplifies to $K_2 = K_1/T$. The closed-loop characteristic equation with the PI controller then simplifies to,

$$(Ts + 1)\left(s + \frac{K_1}{T}\exp(-st_d)\right) = 0. \tag{5.26}$$

The roots of the inner root locus are at $-1/T$ and $-1/t_d$. The first root corresponds to cancellation of the dominant process pole by the PI controller. With these PI settings it can be seen that the theoretical response during the second delay period would be a straight line. The second root corresponds to the break point gain (K_{1b}), and is given by.

$$K_{1b} = (T/t_d)\exp(-1) = 0.368(T/t_d).$$

The break point gain exhibits the critical damping condition of the closed-loop system. The analysis so far shows that the normalized proportional gain is always a function of the scaled time constant (T/t_d). Therefore it is now reasonable to write a general expression for K_1 as,

$$K_1 = \rho\tau_d. \tag{5.27}$$

where the scaled time constant is defined as the reciprocal of the normalized time delay,

$$\tau_d = \frac{T}{t_d}.$$

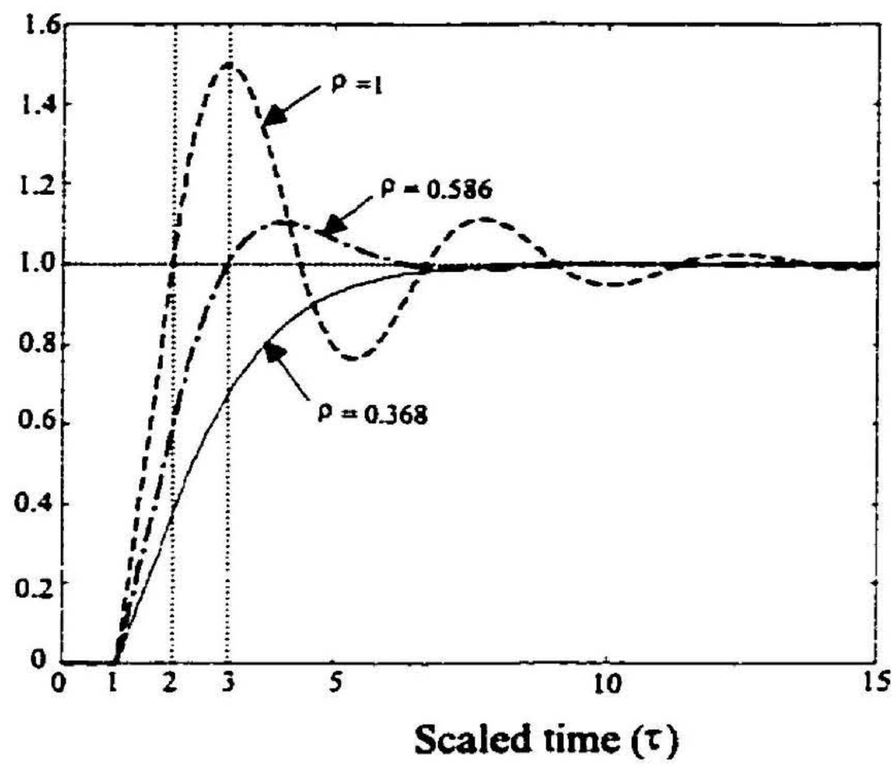


Figure 5.2: Effect of proportional weighting (ρ) on step response

The coefficient ρ is termed as *proportional weighting*. A value for ρ is usually selected to generate the required performance characteristics. In ZN step-response method, this term is fixed and assigned 0.9 for PI controller and 1.2 for the PID controller. With the above PI settings the plant response for unit step input has been further analyzed. For simplifying the response expressions define the scaled time as,

$$\tau = \frac{t}{t_d}.$$

Assume the total time delay period occurs between the controller output and the plant input. For three initial scaled time periods the time response expressions are shown below. In order to avoid the overshoot during each period, the limiting values for ρ are also shown.

$$\text{(a)}\ 0 \leq \tau \leq 1 \quad y(\tau) = 0. \tag{5.28a}$$

$$\text{(b)}\ 1 \leq \tau \leq 2 \quad y(\tau) = \rho(\tau - 1) \qquad \text{For } y(2) \leq 1,\ \rho \leq 1. \tag{5.28b}$$

$$\text{(c)}\ 2 \leq \tau \leq 3 \quad y(\tau) = \rho(\tau - 1) - \tfrac{1}{2}\rho^2(\tau - 2)^2 \quad \text{For } y(3) \leq 1,\ \rho \leq 0.586. \tag{5.28c}$$

Figure 5.2 shows the response variations for three PI settings that correspond to the three proportional weighting limits and it can be observed how response is varied for each limit of proportional weighting. From this analysis it is clear now that if the closed-loop gain is less than the break point gain or $\rho \leq 0.368$, the PI controller would be sufficient. The addition

of derivative controller within this range makes the response more sluggish and will take a longer duration to reach the set-point value. Practical control problem can allow a small overshoot to ensure faster rise time and also a faster settling time within an allowable error tolerance. The simulation results showed that when $\rho \leq 0.51$, the overshoot $OS < 5\%$. This corresponds to about 50% of the set-point response within the second period of time delay. In summary, the pole zero cancellation PI controller is,

$$\left.\begin{aligned} K_{\mathrm{P}} &= \rho\frac{T}{kt_d} \\ K_{\mathrm{I}} &= \rho\frac{1}{kt_d} \end{aligned}\right\}. \tag{5.29}$$

PID tuning

The tight PID controller can be allowed only when the actuator gain allows the proportional weighting to exceed 0.51 (refer next section). It can be seen in the PI design that having $\rho > 0.51$ causes an excessive overshoot. Therefore the derivative action is imposed to minimize the undesirable overshoot while achieving a faster rise, which is limited in the above PI design. During the first unit of scaled time period, the error is constant and therefore the derivatives of error are also zero. As a result, when the time is within this period, $0 \leq \tau \leq 1$, only proportional and integral control actions are active and the derivative action is idle. With the PI settings in (5.29) and when $\rho = 1$, the response reaches its target value during this period and overshoot is inevitable (Figure 5.2). Hence the maximum limit for the proportional weighting can be set as $\rho_{\mathrm{max}} = 1$. The next exercise is to find the necessary amount of derivative action required to bring this overshoot to a minimum. Assume the normalized derivative gain can be represented by, $K_3 = \alpha T$. The α is termed as *derivative weighting*. Using the same gain relationship as in (5.25), the first order differential equation showing the time response within the scaled time $n \leq \tau \leq n+1$ can be expressed as,

$$\tau_d\frac{dy(\tau)}{d\tau}+y(\tau) = \rho\tau_d e(\tau-n)+\alpha\tau_d\frac{de(\tau-n)}{d\tau}+\frac{\rho}{(\alpha+1)}\left[\int_n^{\tau} e(\tau-n)d\tau+\sum_{j=0}^{n-1}\int_j^{j+1} e(\tau-j)d\tau\right] \tag{5.30}$$

and therefore the time response $y(\tau) = f(\tau_d, \rho, \alpha)$.

When the proportional weighting ρ is fixed, the overshoot of the response can now be controlled by α alone. The over weighting of damping through α lowers the gain margin and at the maximum limit of α the system may become unstable. As an example, when $\alpha = 1$, the

gain margin drops below 1dB. Therefore a safety range for α is set as $0 \leq \alpha < 1$. From (5.30) it can be inferred that for the optimum control based on a given cost function, α is always a function of ρ only. The response behaviour of different values of α have been investigated and it has been observed that when α is increased from zero the response overshoot also increases. This is due to increase of ρ and if α is closer to one, the system becomes too oscillatory. This latter behaviour is mainly due to the association of the term α in the integral term. Also it can be argued that the required derivative weighting is low when the allowable proportional weighting is low. At an extreme the control can be accomplished only by PI control. By numerical simulations the acceptable values for the derivative weighting have been observed. When $\rho = 1$ then α is set at 0.4. Similarly for the other limit, when $\rho = 0.5$, α is set at 0.1. The PID settings derived in [125], which corresponds to optimum integral of the absolute value of the error (IAE) have been carefully analyzed. The equivalent α and ρ computed for the PID values in [125] have shown an approximately linear relation to each other. Therefore within the range of $0.5 \leq \rho \leq 1$ the relationship between two weightings is assumed to be linear and fixed as,

$$\alpha = 0.6\rho - 0.2 \ . \tag{5.31}$$

The optimum PID setting is now performed using the overshoot as the performance criterion. The analysis up to now has simplified the PID settings to a single unknown variable (ρ). This term can be adjusted until a desirable overshoot is achieved. For this exercise, the overshoot specification is chosen as 5%. Since with the simplified PID settings the response is only a function of τ_d and ρ, a single relationship for proportional weighting can be obtained in terms of the scaled time constant τ_d. A simulation experiment was performed for different τ_d values and the proportional weighting was adjusted in each simulation to retain 5% or less overshoot. The plot of ρ versus τ_d is shown in Figure 5.3. Since the derivative weighting has been now set proportional to the proportional weighting (equation (5.31)), a low value of ρ means that the amount of damping needed by PID is also low. From this variation we can conclude that when the normalized time delay is very small (or $\tau_d \gg 1$), PI control is sufficient. This is in agreement with the previous results on zero dead time tuning. Also, when the normalized dead time is sufficiently large (or $\tau_d \ll 1$), the amount of damping requirement reduces. This is mainly due to existence of many closed-loop poles near the

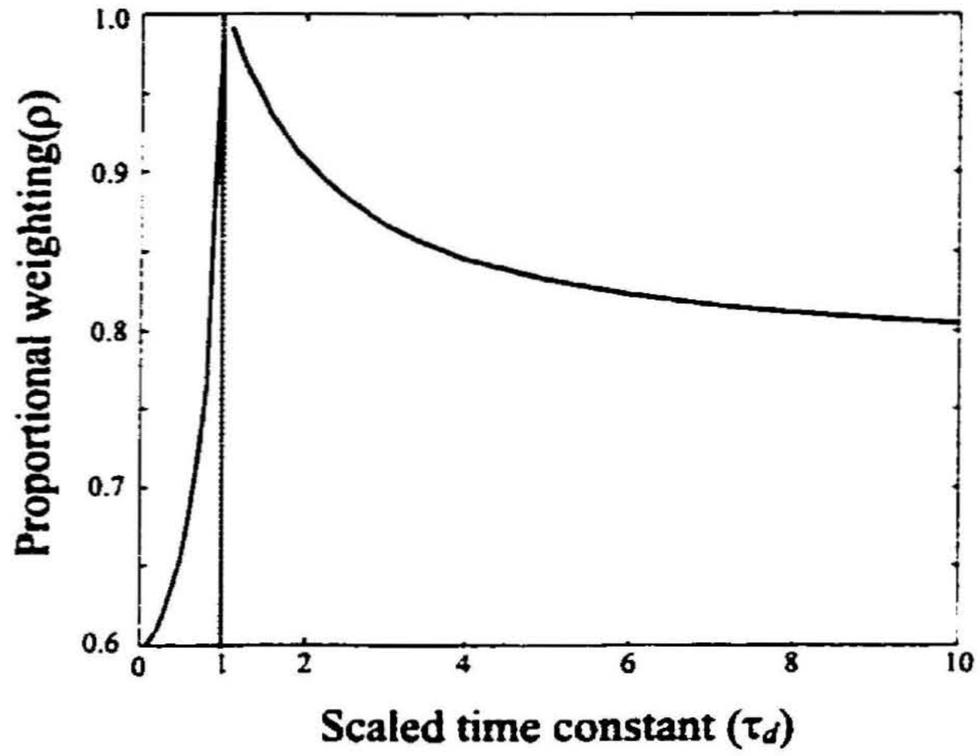


Figure 5.3: Proportional weighting in PID control for 5% or less overshoot

imaginary axis, where the effect of zero addition by the derivative term is not significant enough to change the response characteristics. In order to derive a PID tuning formula, a least square curves have been fitted to the curve shown in Figure 5.3 and the expressions for the proportional weighting to provide 5% or less overshoot are obtained.

For relatively short time delay problems $\tau_d \geq 1$

$$\rho_{\text{PID}} = 0.770 + 0.245\,(\tau_d)^{-0.854}\,. \tag{5.32a}$$

For relatively long time delay problems $\tau_d < 1$

$$\rho_{\text{PID}} = 0.603 + 0.275\,(\tau_d)^{2.4}\,. \tag{5.32b}$$

Therefore the new PID tuning laws can be summarized and for convenience they are rewritten in terms of absolute process terms. The recommended range for the proportional weighting is $0.51 \leq \rho \leq 1$.

$$\left.\begin{aligned} K_{\text{P}} &= \frac{\rho}{k}\left(\frac{T}{t_d}\right) \\ K_{\text{D}} &= \frac{\alpha T}{k} = \frac{(0.6\rho - 0.2)}{k} T \\ K_{\text{I}} &= \frac{K_{\text{P}}}{k K_{\text{D}} + T} = \frac{\rho}{k(0.6\rho + 0.8)}\left(\frac{1}{t_d}\right) \end{aligned}\right\}. \tag{5.33}$$

A suitable value for ρ is obtained from (5.32). For fine-tuning, manipulate ρ to adjust the overshoot. The phase and gain margins drawn for this PID tuning law are shown in Figure

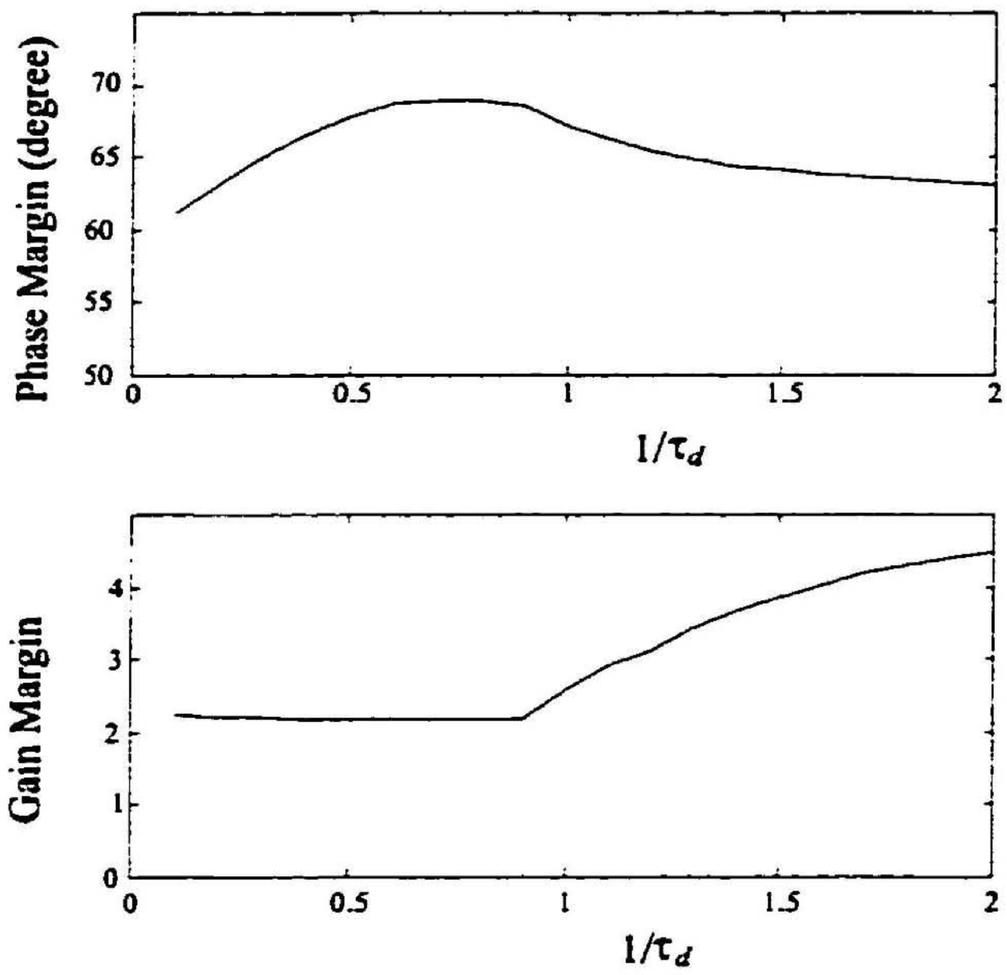


Figure 5.4: Gain and phase margins of the new PID settings vs. normalized time delay

5.4 and observed phase and gain margins are respectively above 60^o and 2 respectively. This is in agreement with most of the existing tuning techniques [131].

Optimum controller selection for relatively short time delay problems

In set-point control of any first order process, the integral action causes the control signal to rise monotonically during the initial time delay period. The wind-up problem can be avoided only if the actuator has a higher capacity than the maximum PID controller signal required for the desired settings. With the above PI or PID settings, the maximum controller signal during the transient response of this particular process shows less than 10% overshoot and has an extremely low bandwidth. In such circumstances an actuator with little extra capacity can drive a system with no danger of integral wind-up. Therefore processes with short time delay are first considered for the optimum controller selection. The above analysis has shown that the tight PID control can be allowed only when the proportional weighting is chosen at a higher level than the value permissible for PI control. Therefore by knowing the maximum permissible value of the proportional weighting it is possible to determine whether tight PID control can be allowed or not.

With the PI settings shown in (5.29), the maximum controller signal can be determined using the expressions given in (5.28). The maximum PI controller signal per unit step response is therefore given by,

$$u_{\max} = u(\tau_p) = K_{\mathrm{P}}\left(1 - y(\tau_p)\right) + K_{\mathrm{I}} t_d \int_0^{\tau_p} \left(1 - y(\tau)\right) d\tau, \tag{5.34}$$

where τ_p is the time at which the PI controller signal reaches its maximum. There are two cases to be considered.

Case A When $\rho\tau_d \geq 1$ $\qquad \tau_p = 1.$

Case B When $\rho\tau_d < 1$ $\qquad \tau_p = 1 + (1 - \rho\tau_d)/\rho.$

The maximum allowable limit for the proportional weighting (ρ_a) is therefore determined by the allowable value of the maximum controller signal. To find the ρ_a value, set the upper margin of the actuators gain to the maximum controller signal in (5.34) (i.e. $u_{\max} = U_u$). Substituting the response expressions of (5.28) into (5.34) the limiting values are obtained as:

$$\left.\begin{aligned} \rho_a &= kU_u/(1+\tau_d) && \text{when } \rho\tau_d \geq 1 \\ \rho_a &= \left(\sqrt{1+\tau_d^2(2U_u - 1)} - 1\right)/\tau_d^2 && \text{when } \rho\tau_d < 1 \end{aligned}\right\}. \tag{5.35}$$

Set $\rho \leq \rho_a$ for the PI controller settings in (5.29) to work with no integral wind-up. If the optimum proportional weighting based on a given performance level is ρ_{PI}, then select $\rho = \min(\rho_a, \rho_{\mathrm{PI}})$. Based on 5% or less overshoot criterion, $\rho_{\mathrm{PI}} = 0.51$.

With the new PID settings it can be shown that the maximum controller signal always occurs at the end of the first dead time period i.e. $\tau = 1$. As the error derivative (rate of change of response) during this period is zero or negligible, the derivative control action has no influence. If there is no saturation limit, the controller signal reaches the maximum value when $t = t_d$ or $\tau = 1$ and then falls due to the addition of derivative action. Using the new PID tuning rules, a condition for the maximum PID signal per unit step response can be expressed by:

$$u_{\max} = K_{\mathrm{P}} + K_{\mathrm{I}} t_d \leq U_u.$$

Using the new PID rules in (5.32),

$$\rho\tau_d + \frac{\rho}{k(0.6\rho + 0.8)} \leq U_u. \tag{5.36}$$

In order to find the allowable limit for the proportional weighting of PID control (ρ_b), consider the equal condition of the above equation and solve the quadratic expression given by,

$$0.6\tau_d\rho_b^2 + (0.8\tau_d - 0.6kU_u + 1)\rho_b - 0.8kU - u = 0. \tag{5.37}$$

To avoid the integral wind-up during PID control, select $\rho = \min(\rho_b, \rho_{\text{PID}})$

The analysis above shows that the requirement for a PID controller arises only when the system can allow a higher gain which is governed by the saturation limits. Based on the above analysis the following rule is stated for optimum controller selection.

If $\rho_a < \rho_{\text{PI}}$ then the optimum selection is a PI *controller. Otherwise the tight control can be allowed by a* PID *controller.*

For very small normalized time delay problems, the limiting proportional weighting for PI control (ρ_a) becomes too small (5.35) and PI controller would be the optimum. This agrees with the results of section 5.2. The above selection rule can be safely used until the time delay is less than 1.5 times the time constant (i.e. $\tau_d > 1/1.5$).

5.5 Analysis III: Two-Point Design of a PI Controller for Long Time Delay Processes

In the previous analysis it was observed that when the normalized time delay becomes larger, the required proportional weighting to retain 5% or less overshoot is reduced to the minimum and the derivative action becomes ineffective. The process response with PI and PID controller becomes almost identical. This phenomenon has been also observed in the IMC PID design [134]. The process dynamic differs mainly due to the existence of closed-loop poles closer to the imaginary axis of the *s-plane*. Therefore it can be concluded that for larger normalized time delay processes the best PID controlled response can be obtained with a zero derivative and the PI controller is optimum. The pole zero cancellation PI controller shows an extremely low bandwidth and the response is more sluggish. Therefore in this section a separate PI tuning formula is developed for such processes where $T \ll t_d$ or $\tau_d \ll 1$.

The main problem associated with larger time delay is the integral wind-up over a longer period. Secondly the proportional weighting can allowed to be have a higher value for quick

response. As a result the equal weighting in the PI rules described in section 5.2 overestimates the integral action and gives a lower estimate of proportional weighting. Larger dead time processes require a slower rate of integration to avoid any integral-windup and also excessive overshoot. Therefore the normalized integral gain can be assumed to be inversely proportional to the dead time. The normalized PI gains are now redefined as,

$$\left.\begin{aligned} K_1 &= \rho_l \tau_d \\ K_2 &= \gamma_l / t_d \end{aligned}\right\}. \tag{5.38}$$

where ρ_l is the redefined proportional weighting and γ_l is the *integral weighting* for the PI control. In the pole zero cancellation PI controller defined in (5.29) the two weightings were equal to each other and the maximum value permitted for the proportional weighting was one. Since the time delay is sufficiently greater than the process time constant the two weightings can be set at $\gamma_l < \rho_l$ and ρ_l can be allowed to take a value greater than one.

The first-order differential equation showing the time response within the scaled time $n \leq \tau < n+1$ can now be simplified as,

$$\tau_d \frac{dy(\tau)}{d\tau} + y(\tau) = \rho_l \tau_d e(\tau - n) + \gamma_l \left[\int_n^{\tau} e(\tau - n) d\tau + \sum_{j=0}^{n-1} \int_j^{j+1} e(\tau - j) d\tau \right]. \tag{5.39}$$

The step response solution of (5.39) for the first three delay periods has been obtained and the final expressions are given below. The equal conditions of the response equations correspond to boundary conditions of the past and future responses.

(a) $0 \leq \tau \leq 1$

$$y(\tau) = 1. \tag{5.40a}$$

(b) $1 \leq \tau \leq 2$

$$y(\tau) = \rho_l \tau_d + \gamma_l \left((\tau - 1) - \tau_d \right) - \tau_d (\rho_l - \gamma_l) \exp\left(-\frac{\tau - 1}{\tau_d} \right). \tag{5.40b}$$

(c) $2 \leq \tau \leq 3$

$$y(\tau) = y_1(\tau) + y_2(\tau). \tag{5.40c}$$

where

$$y_1(\tau) = \left[(\rho_l - \gamma_l)(\tau - 2) + (\rho_l - 2\gamma_l)\tau_d - \exp\left(-\frac{1}{\tau_d} \right) \right] (\rho_l - \gamma_l)\tau_d \exp\left(-\frac{\tau - 2}{\tau_d} \right),$$

$$y_2(\tau) = \tau_d(\rho_l - \gamma_l)(1 - \rho_l\tau_d + 2\gamma_l\tau_d) + \gamma_l + \gamma_l\left(1 - 2(\rho_l - \gamma_l)\tau_d - \gamma_l\frac{(\tau-2)}{2}\right)(\tau - 2).$$

In this case the scaled time constant is considered to be very much less than one and the exponent terms in (5.40) can be assumed to be negligible compared to the terms with no exponent. Hence the two expressions (5.40a) and (5.40b) can be approximately expressed as,

For $1 \leq \tau \leq 2$ $$y(\tau) \approx \rho_l\tau_d + \gamma_l\left((\tau - 1) - \tau_d\right). \tag{5.41a}$$

For $2 \leq \tau \leq 3$ $$y(\tau) \approx y_2(\tau). \tag{5.41b}$$

It can be observed from above expressions that the response during the second period of scaled time given by (5.41a) is a monotonically increasing function during the valid period of time. This implies that it is impossible for the PI controlled response to reach its steady state before the time $\tau = 2$. This is the clear limitation of PI controller performance when it is used for long dead time process models [142]. With this limitation the system can be allowed to accelerate as much as possible during the second delay period and control the overshoot in the third period. Therefore two target points are first defined corresponding to the two response periods. Assume the response level to be reached at the end of second time duration $y(2) = y_a$. Then set $y_a < 1$ to avoid any excessive overshoot of response. By substituting this condition into (41a) we obtain,

$$\rho_l = \frac{y_a - \gamma_l(1 - \tau_d)}{\tau_d}. \tag{5.42}$$

In order to achieve fast settling, the response during the third delay period can be allowed to reach the peak of the overall response. Let the expected maximum response height of the unit step response be $y_m(\geq 1)$. By substituting the time given by $dy_2(\tau)/dt = 0$ into (5.41b) y_m is obtained as;

$$y_m = 1/2 + \gamma_l + \tau_d(\rho_l - \gamma_l)(\rho_l\tau_d - 1). \tag{5.43}$$

Solving (5.42) and (5.43) the integral weighting is obtained as,

$$\gamma_l = \frac{1}{2(1 - \tau_d)}\left[2(y_m - 1) - y_m\tau_d + \sqrt{2 - 4(y_m - y_a) + \tau_d(y_m^2\tau_d - 4y_a + 2)}\right]. \tag{5.44}$$

The two equations given in (5.42) and (5.44) provide the necessary PI weightings to estimate the normalized gains in terms of the process parameters and the expected two target points. For very large normalized time delay problems, the approximation error is negligible and the

user can decide two suitable target points and can easily determine the PI gains. The error of the approximation becomes dominant when the normalized lag time is medium long in which case y_a will be an overestimate and y_m will be an underestimate. For larger time delay problems, the above estimations can very accurately be used. The numerical simulations provide the following safe limits for the target points assuming the allowable overshoot is about 5% .

Valid range for the time delay: $t_d > T$.

$1 < \frac{t_d}{T} < 2$	$y_a = 0.6,\ y_m = 1.02;$
$2 \leq \frac{t_d}{T} < 4$	$y_a = 0.7,\ y_m = 1.02;$
$4 \leq \frac{t_d}{T} < 6$	$y_a = 0.8,\ y_m = 1.02;$
$6 \leq \frac{t_d}{T}$	Accuracy is sufficient to predict the two points

The proposed PI setting is easy to understand. Irrespective of the magnitude of time delay, the user can select two points from the desired set-point response. Since the response is slow for long time delays, the integral wind-up with the above settings would not be a serious problem. With about 10% extra capacity of an actuator, the response with 5% or less overshoot can be easily accommodated with no integral wind-up. The gain and phase margins computed for these settings while seeking $y_a = 0.8$ and $y_m = 1.02$ is shown in Figure 5.5. If a higher gain or phase margin is sought, the two target points can be changed appropriately.

5.6 Simulation Examples

The following computer simulations show the effectiveness of the tuning procedure developed so far. For all simulations the PID controller with the derivative filter shown in (5.3) is employed, and without loss of generality N=10 is used throughout. The approximated process parameters are evaluated either by using a relay experiment or from a plant open-loop step test. With proportional control, the theoretical closed-loop system of a first order plant with zero dead time has infinite gain margin. Also, when the time delay is very long the system has a very low critical gain. The critical gain evaluation by the relay experiment for negligible time delay or very long time delay gives rather an erroneous estimate. Therefore plants having a negligible time delay or a very long time delay are estimated by using the open-loop step response method that is described in [143]. For all cases, the process gains are

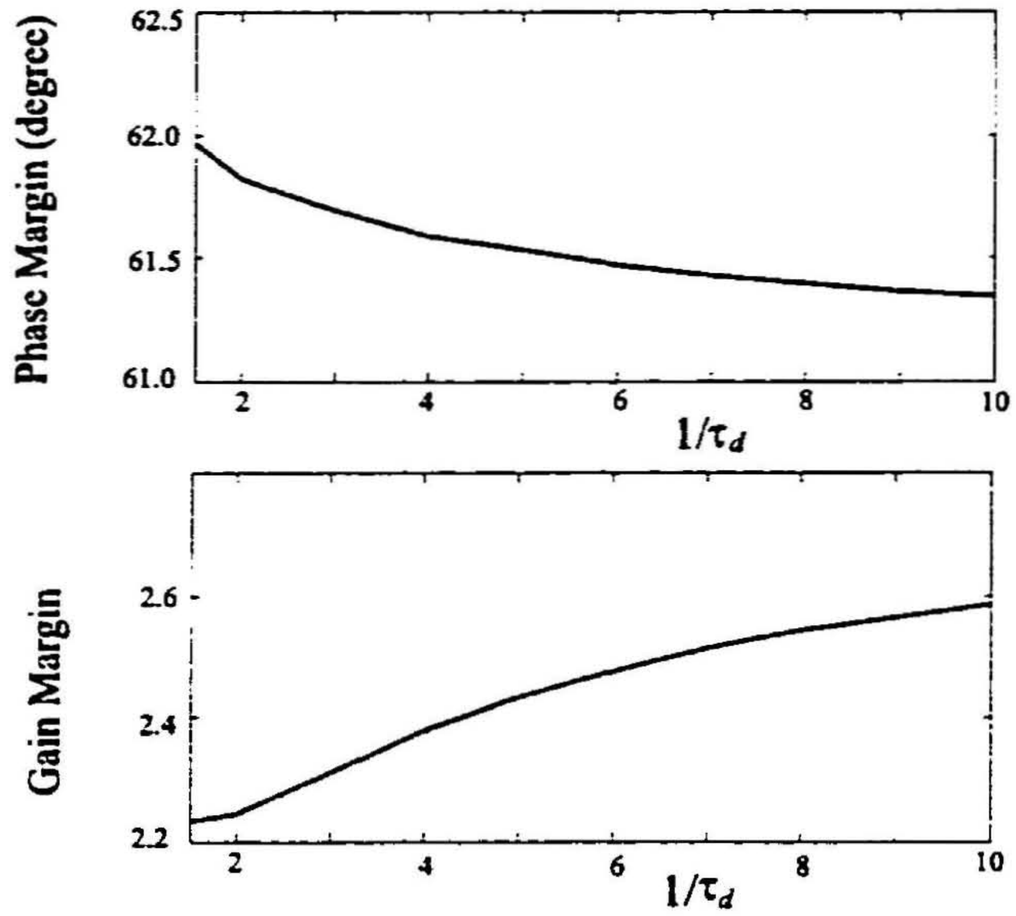


Figure 5.5: Gain and phase margins of the two-point PID settings corresponding to $y_a = 0.8$ and $y_m = 1.2$.

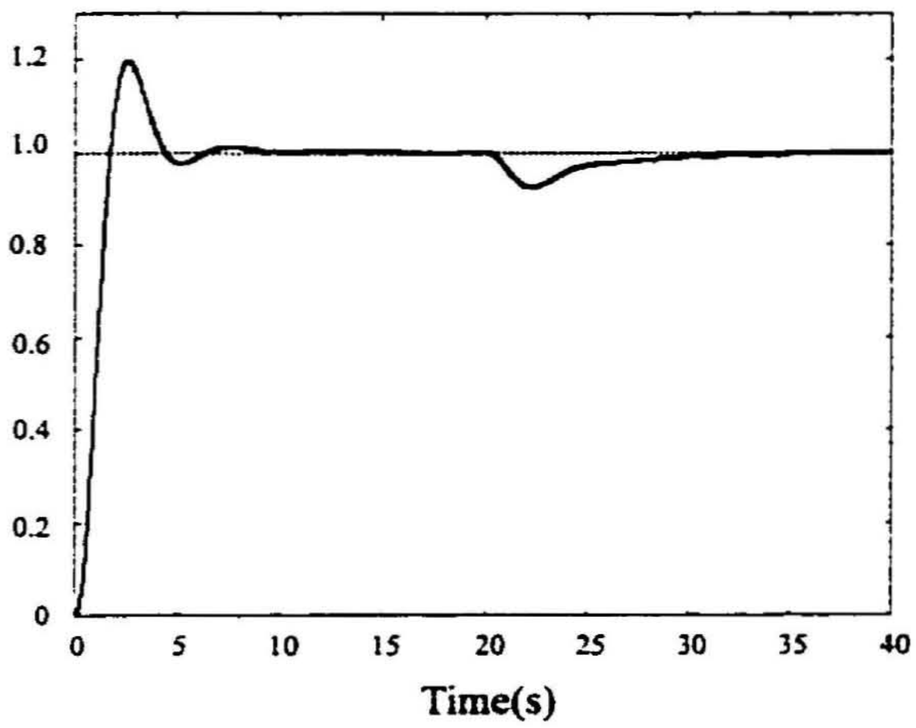
estimated by the open-loop step response test. During each simulation a constant 50% load disturbance is added. Except example 2, all the others have been chosen from the literature.

Example E2 : A third-order system with negligible time delay.

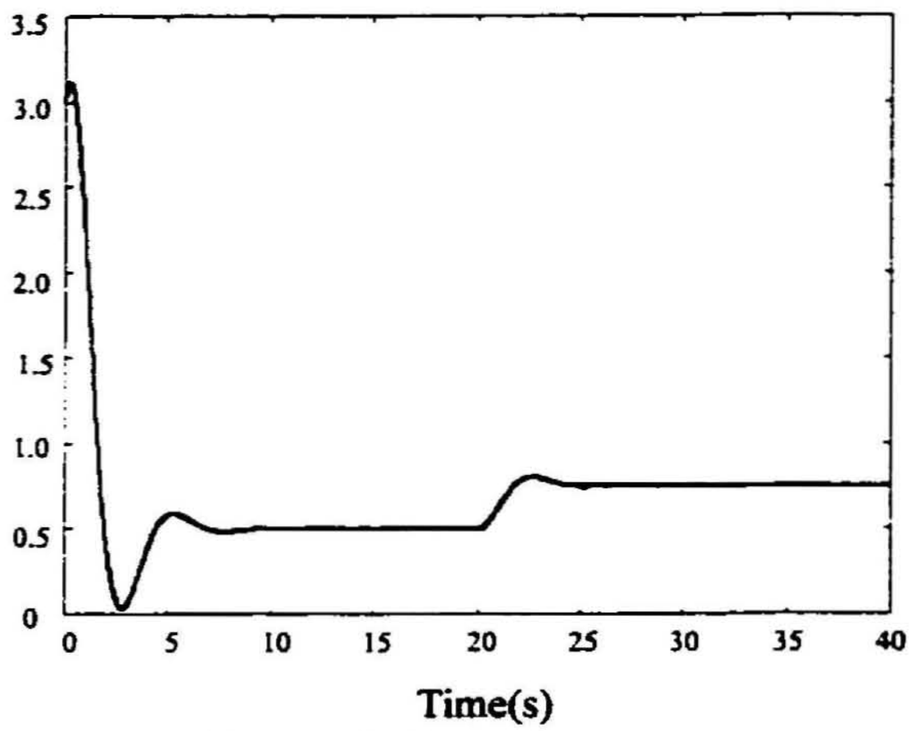
It is hard to find an example in literature with an estimate of zero dead time. Therefore we have chosen a third order transfer function model given by,

$$G(s) = \frac{2}{(5s+1)(0.5s+1)(0.2s+1)}.$$

The open-loop step response test yields $k = 2$ and $T = 5.72$s and the time delay component is negligible. For unit step response, the steady state controller gain required is 0.5. Let the actuator saturation limits be given by [0 4]. Using the results of section 5.3, the optimum PID controller is PI. Using the tuning law in section 5.3.2, $u_{\max} < 4$, select $K_{\mathrm{P}} = 3$ or $K_{\mathrm{I}} = 6$. By assuming the maximum expected overshoot to be 2% ($OS_d = 0.02$) obtain $K_{\mathrm{I}} = 0.6086$. The response curve for this PI setting is shown in Figure 5.6. The response achieves its fastest rise without exceeding the saturation limits, but shows somewhat poor load disturbance characteristics. By increasing the integral gain, it is possible to obtain better load disturbance, but at the expense of poor



(a) Unit step response



(b) Manipulator signal

Figure 5.6: Closed-loop response of Example E2

transient response. Since this design is based on controller signal overshoot, minimum set-point overshoot is the prime concern in the PI settings.

Example E3 : A second-order system with relatively a short time delay.

This example is taken from reference [118]. The transfer function is given by,

$$G(s) = \frac{\exp(-0.5s)}{(s+1)^2}.$$

The closed-loop relay experiment yields the ultimate gain $k_u = 4.476$ and the ultimate period $t_u = 2.3$s. Using (5.24) the process estimates are $T = 1.5971$s and $t_d = 0.6585$s. The process gain $k = 1$. The design is related to a short normalized time delay problem. According to the design criteria, the optimum selection is either a PI or PID controller, which depends on the actuator saturation. Using (5.32a), the proportional weighting for 5% or less overshoot is 0.88. The estimated PID parameters using (5.33) are $K_{\mathrm{P}} = 2.13$, $K_{\mathrm{I}} = 1.01$, and $K_{\mathrm{D}} = 0.524$. Next, actuator saturation limits are imposed as [0 2] and $U_u = 2$. This corresponds to twice the controller gain at steady state. With this limit, the above proportional weighting value violates (5.36) where there is a possibility of integral wind-up. Using (5.35) the limiting proportional weighting for PI design is 0.729. Since this value is higher than the limiting value corresponding to a PI controller, it is possible to allow tight control through a PID controller. The limiting proportional

weighting calculated from (5.36) implies $\rho \leq 0.608$. Using the PID formula given in (5.33) and assigning $\rho = 0.6$ the PID terms are given by, $K_P = 1.455$, $K_I = 0.7855$, and $K_D = 0.255$. For comparison, the RZN PID [128] settings and Zhung and Atherton's ITSE optimized PID (**ZA-PID**) settings [118] are also tested. With the estimated process terms above the RZN-PID terms: $K_P = 2.686$, $K_I = 2.335$, and $K_D = 0.772$ and the set-point weighting is 0.54 and ZA-PID terms; $K_P = 2.307$, $K_I = 1.28$, and $K_D = 0.636$ have been obtained.

The simulation results are shown in Figure 5.7. The proposed method shows better step response performance. Also with no actuator saturation limits, the response corresponding to the proposed method shows acceptable load disturbance properties. However when the actuator saturation is imposed, the proposed method requires lowering of the gains (weightings) to accommodate limiting requirements and therefore shows poor load disturbance characteristics compared to other two methods. It can be seen from Figure 5.7-c, that the proposed method has satisfied the limiting conditions of the actuator gain with no integral wind-up and therefore the response has not been affected by the integral wind-up. The controller signals based on the other two designs have reached the upper saturation limit and the response has been affected by the wind-up. This example illustrates the flexibility of the PID design for accommodating the actuator saturation. The IMC-PID design also has the flexibility to choose PID parameters based on actuator saturation. However the performance of the latter method on this example has shown similar response compared to the proposed design and therefore it has been excluded from the diagram.

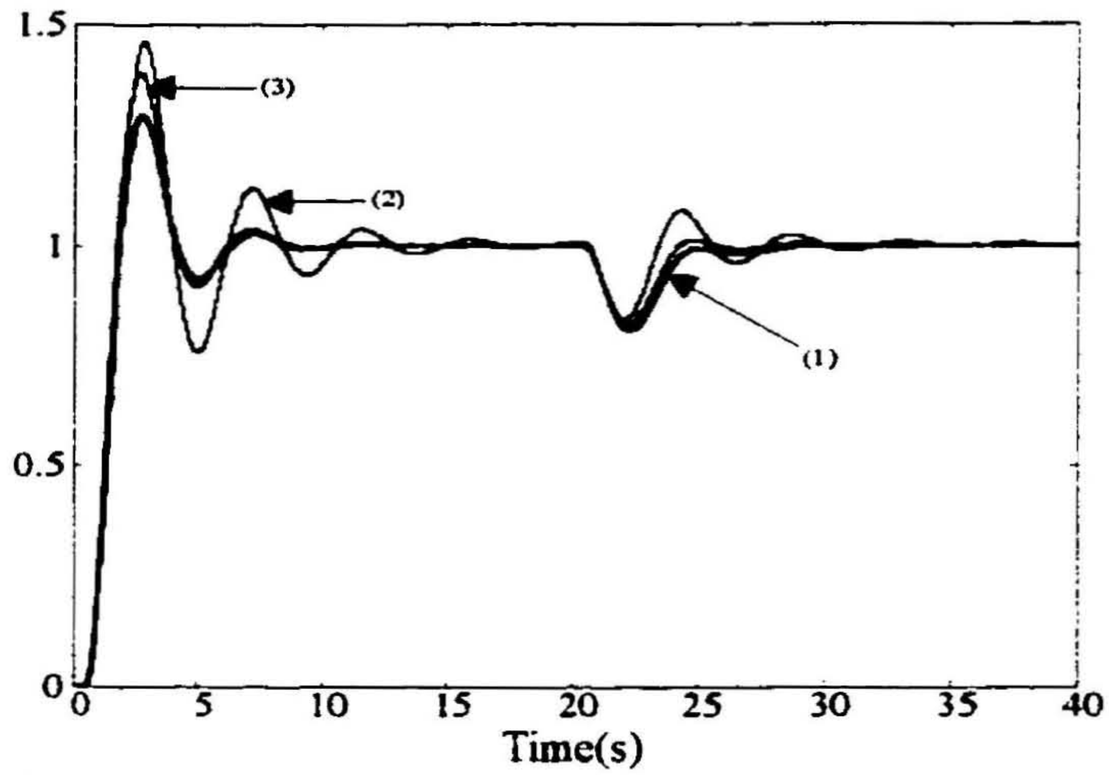
Example E4 : A second-order plant having a normalized time delay closer to one.

This example was chosen from reference [130] and the transfer function is given by,

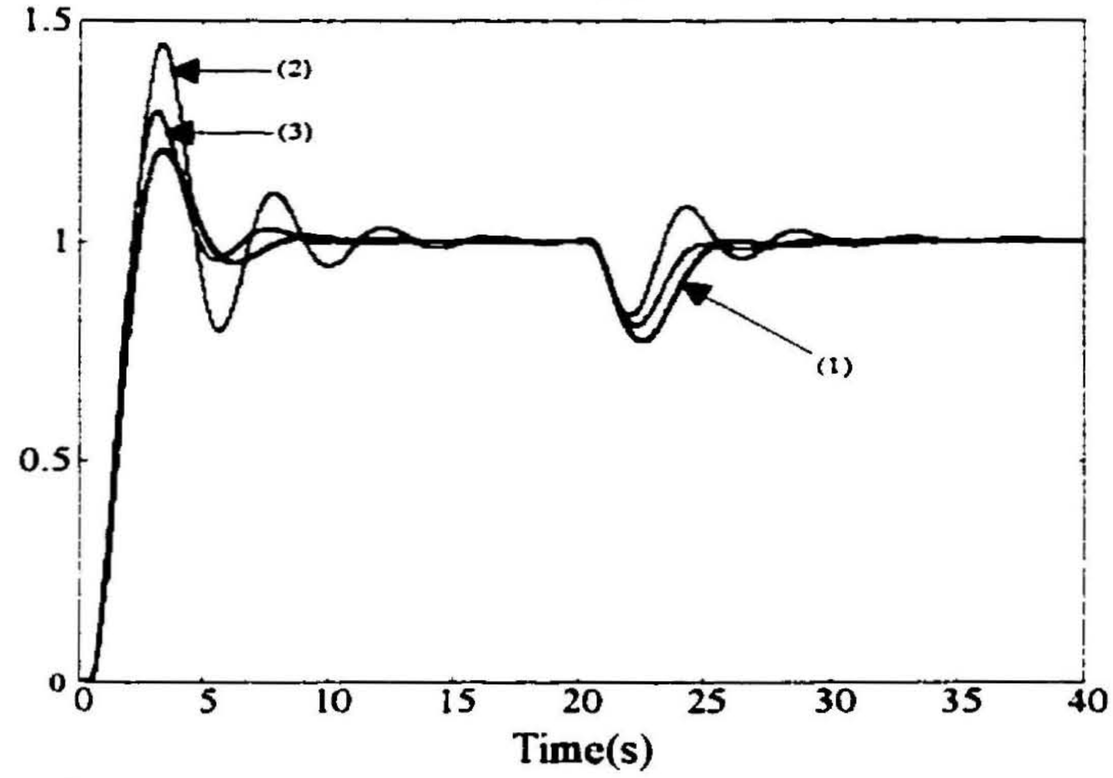
$$\frac{\exp(-s)}{(s+1)(0.5s+1)}.$$

The closed-loop relay experiment yields $k_u = 2.137$ and $t_u = 4.1$s. The estimates are $k = 1$, $T = 1.232$s and $t_d = 1.343$s. The normalized time delay is therefore 1.09. Using the 5% or less overshoot criteria for the PID control (equation (5.32a)), $\rho = 0.827$. Using (5.33) the PID gain terms are; $K_P = 0.759$, $K_I = 0.475$ and $K_D = 0.365$. For comparison, the Zhung and Atherton's (ZA)ITSE optimized PID settings [118]

(a) Response with no actuator saturation



(b) Response with saturation limits



(c) Manipulator signal with saturation limits

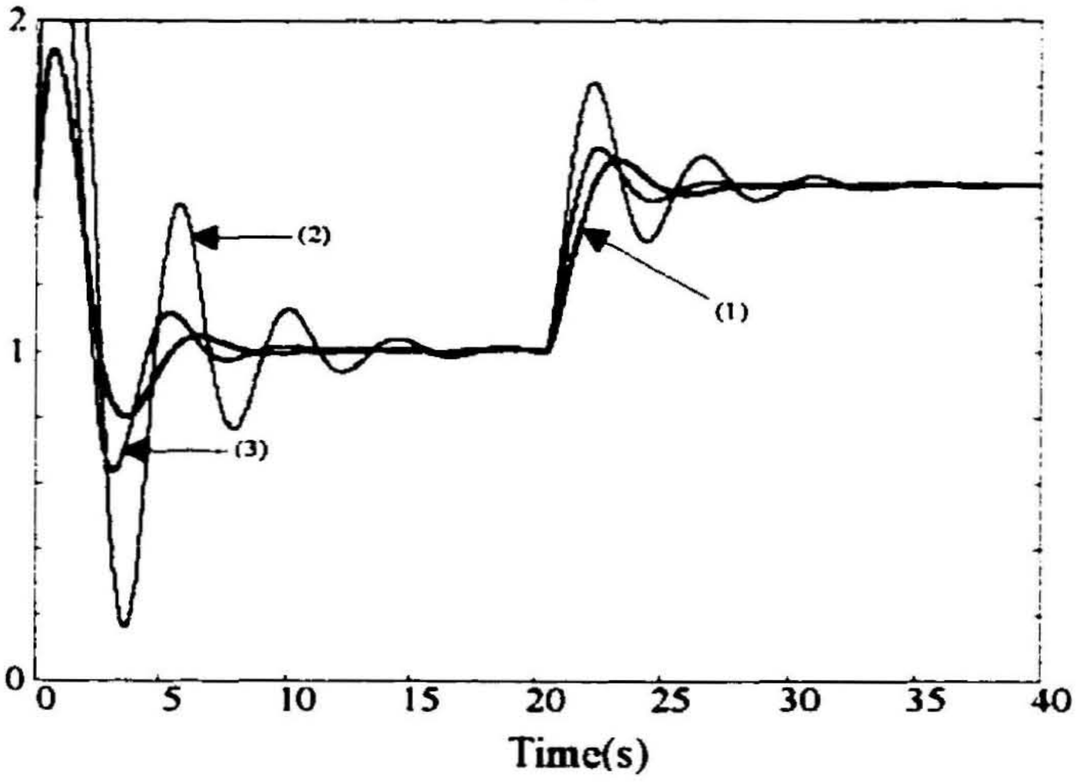


Figure 5.7: Closed-loop response of Example E3. (1) Proposed, (2) RZN, (3) ZA (ITSE optimum)

valid for the longer normalized delay and the PID settings based on gain and phase margin specifications (GPM-PID settings) in [130] are also used. The ZA-PID [118] settings are; $K_P = 1.087$, $K_I = 0.6371$ and $K_D = 0.5526$. The GPM-PID [130] settings corresponding to a gain margin of 3 and a phase margin of 60^o are $K_P = 0.780$, $K_I = 0.520$ and $K_D = 0.260$. The latter values are obtained after transforming the PID terms given in [130] to the present form by using the expressions given in [140]. The response curves are shown in Figure 5.8. The proposed PID settings show minimum set-point overshoot and a satisfactory load disturbance characteristics compared to both ZA-PID and GPM-PID methods. The RZN PID [128] setting is inadequate to cover the particular example and with the closed-loop estimated values, the response has shown quite an oscillatory response.

Example E5 : Higher-order plant with long normalized time delay.

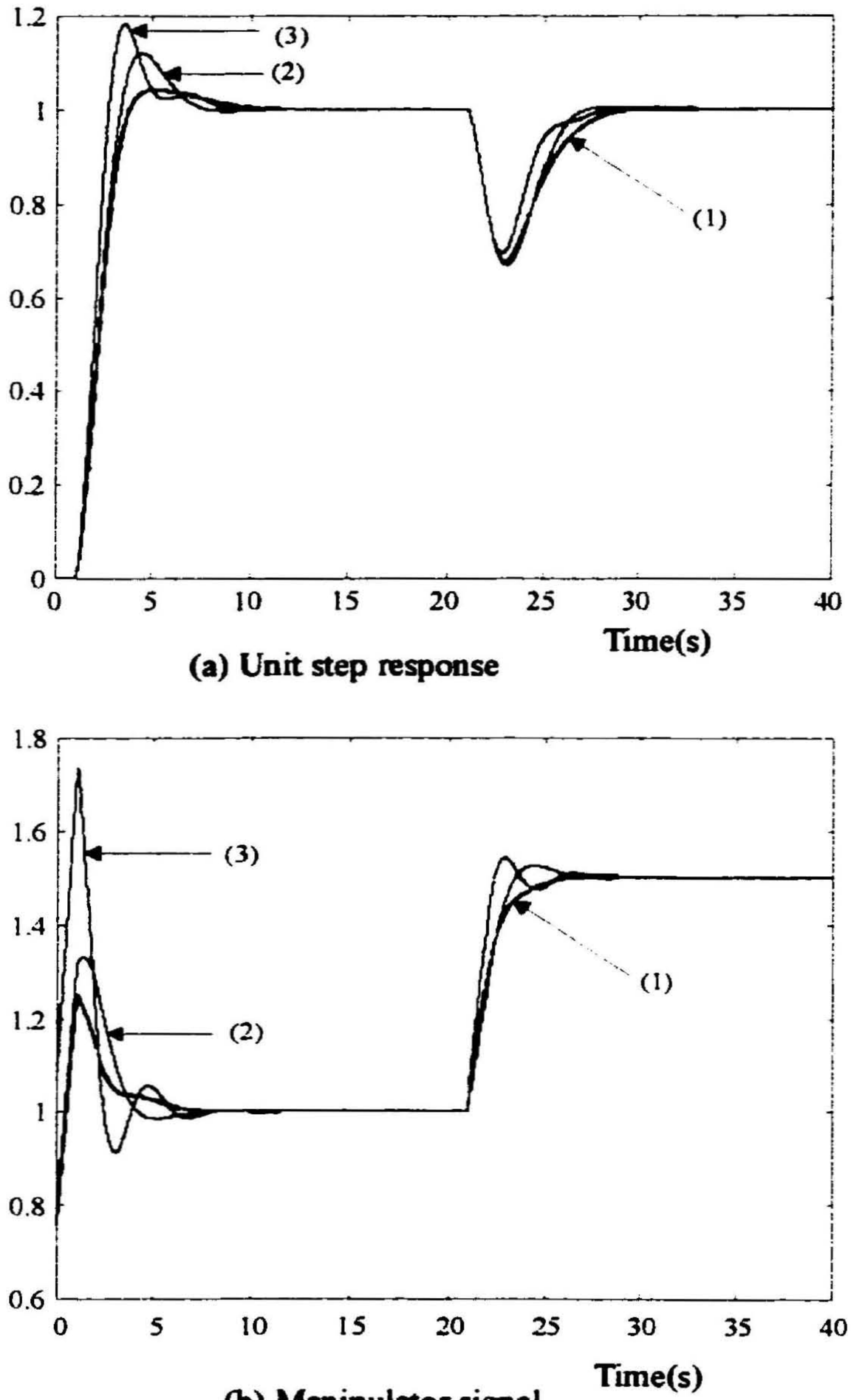
This example was chosen from reference [142]. The transfer function of the model is given by,

$$G(s) = \frac{\exp(-t_d s)}{(s+1)(0.5s+1)(0.5s+1)(0.125s+1)}.$$

For simulations two values for the time delay t_d are considered.

Case-I: $t_d = 4$s.

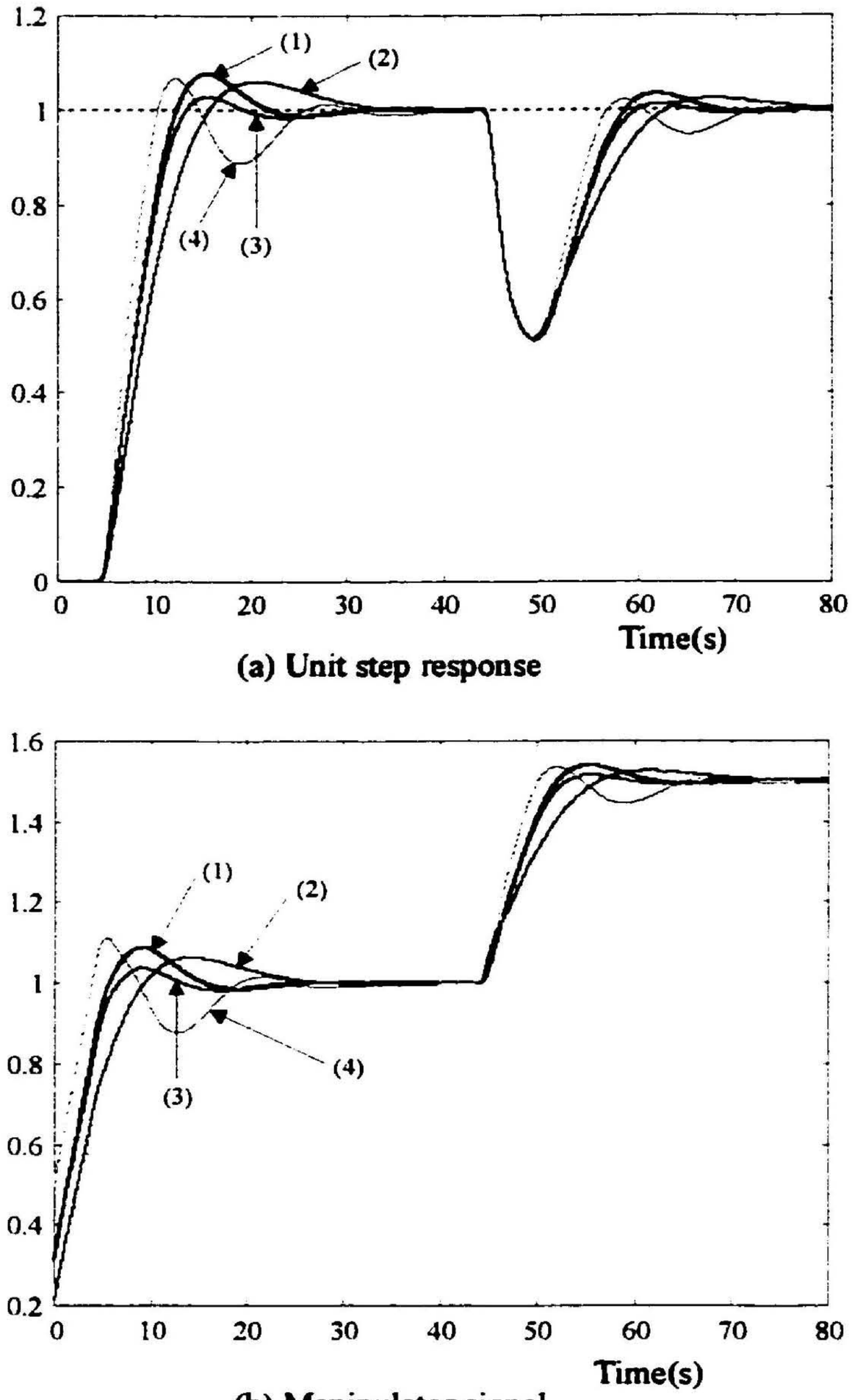
The closed-loop relay experiment yields, $k_u = 1.302$ and $t_u = 11.46$s. The estimates are $k = 1$, $T = 1.521$s and $t_d = 4.462$s. The normalized time delay is therefore 2.934. Using the proposed PID rules in (5.32), $\rho = 0.6238$ and the corresponding gain terms using (5.33) are: $K_P = 0.2126$, $K_I = 0.1191$ and $K_D = 0.2651$. The two-point PI design (**TP-PI**) is next considered for the comparison. Using the recommended values given in section 5.5, the two points are decided as $y_a = 0.7$, $y_m = 1.02$. Using the expressions (5.44) and (5.42) the PI weightings are $\rho_l = 0.9058$ and $\gamma_l = 0.133$. Using (5.38) the PI terms are $K_P = 0.3088$ and $K_I = 0.133$. For comparison, the IMC-PI [134] and PI settings developed for large normalized time delay processes by Khan and Lehman (**KH-PI**) [127] are also simulated. The IMC-PI terms corresponding to $\epsilon/t_d = 1.7$ are $K_P = 0.4946$ and $K_I = 0.1318$. The KH-PI parameters are $K_P = 0.3193$ and $K_I = 0.1249$. The unit step response curves and the variation of the manipulator signal are shown in Figure 5.9.



(a) Unit step response

(b) Manipulator signal

Figure 5.8: Closed-loop response of Example E4. (1) Proposed, (2) GPM, (3) ZA (ISTE optimum)



(a) Unit step response

(b) Manipulator signal

Figure 5.9: Closed-loop response of Example E5-I.
(1) Proposed TP-PI (y_a= 0.7, y_m= 1.02), (2) Proposed PID, (3) KH-PI, (4) IMC-PI

Case II: $t_d = 10$s.

For better approximation, the parameters are estimated by the open-loop step response. The estimates are $k = 1$, $T = 1.5$s and $t_d = 10.5$s. Based on the new two-point PI design, two designs are evaluated. By assigning the two points as $y_a = 0.75$, $y_m = 1$ the PI gains are $K_P = 0.2281$ and $K_I = 0.0580$. Also the second design is considered by using $y_a = 0.8$, $y_m = 1.02$ and the computed PI gains are $K_P = 0.2516$ and $K_I = 0.0609$. Again the KH-PI settings [127] are calculated for the comparison and the corresponding PI gains are $K_P = 0.2655$ and $K_I = 0.0579$. The corresponding unit step response curves are shown in Figure 5.10. It can be seen from the example E5 that the proportional weighting allowed for PID settings is low. As a result the performance based on the proposed two-point PI setting is superior to the proposed PID controller. This proves that the PI controller is sufficient for controlling larger normalized time delay processes. Figure 5.9-b shows that when the normalized time delay is large the signal overshoot is minimal and therefore a proper PID design giving no excessive overshoot of response automatically satisfies the actuator limitations. The IMC-PI design employed the Padè approximation, which resulted in poor PI settings and therefore exhibited the poorest response characteristics. Although KH-PI settings have shown satisfactory performance with zero overshoot, the proposed method has more flexibility to select PI parameters for user specified response behaviour. The two designs in example E5-II show how one can adjust the overshoot by simply lowering the anticipated two points in the response curve. The results show better performance of TP-PI settings in both step response and disturbance rejection. The two point PI design analysis in section 3.3 clearly provides the best that a PID controller can achieve for long time delay control. Further improvement to the transient response requires employment of different control algorithms [142]. Any increase in the rise time by raising the first expected point y_a would produce rather an excessive overshoot of the response. The other tuning methods, such as RZN-PI, ZA-PI and GPM-PI are unable to provide solutions to very long time delay problems.

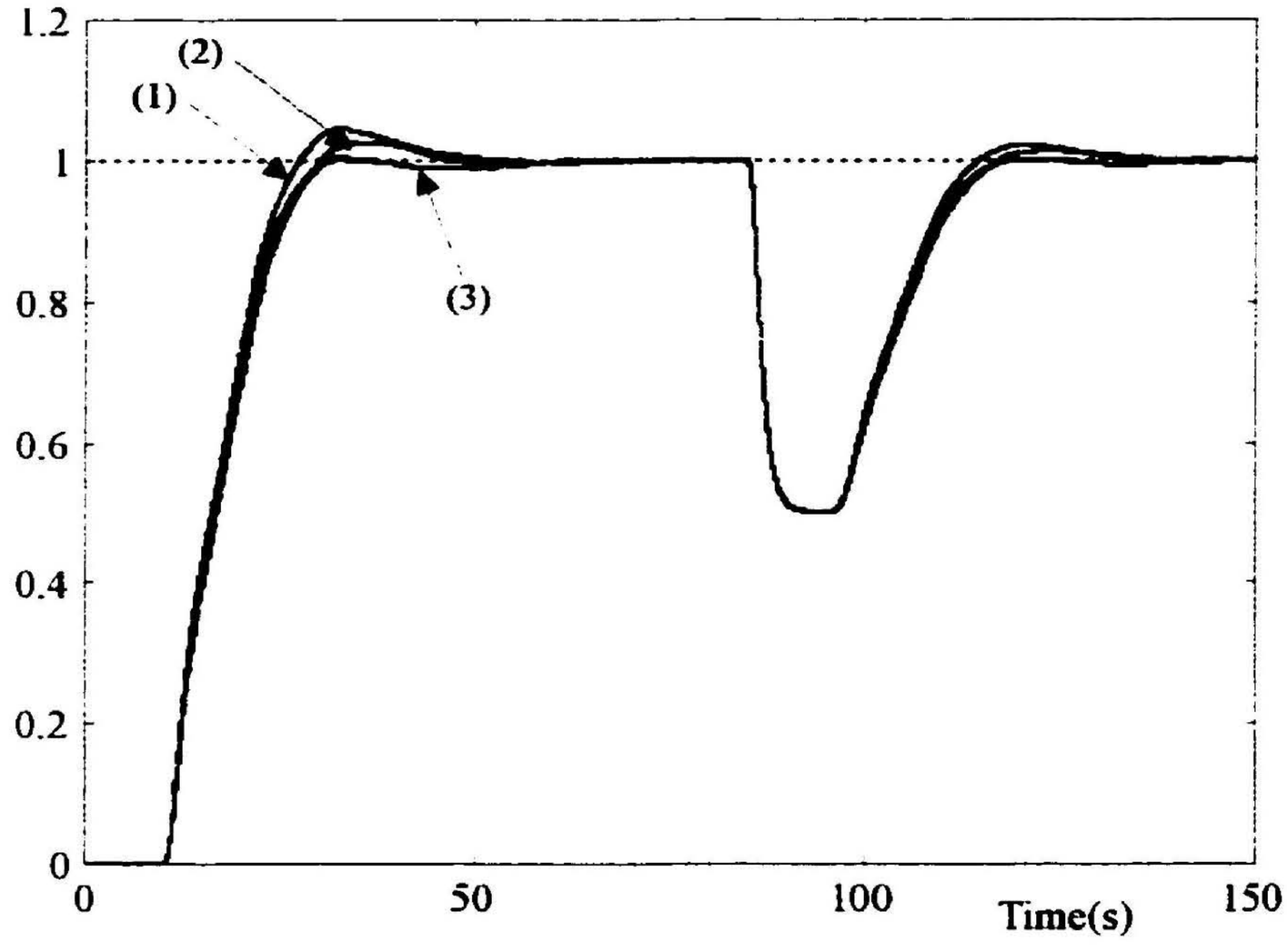


Figure 5.10: Closed-loop response of Example E5-II. (1) Proposed TP-PI ($y_a = 0.75, y_m = 1$),(2) Proposed TP-PI ($y_a = 0.8, y_m = 1.02$), (3) KH-PI.

5.7 Summary

This chapter presented a new time response based design methodology for PID controllers. The present PID tuning is applicable only for the classes of problems that have monotonic open-loop response except for the initial time period where the system can be roughly approximated to a first-order process with a time delay. Based on the magnitude of normalized time delay, three types of tuning rules have been developed to cover the time delay ranging from zero to any higher value. The PID tuning rules for zero dead time processes have been analytically obtained. It has been shown that the derivative action is detrimental to those plants having negligible or large normalized time delay. Tight PID control can be applied for plants having low to medium normalized time delay. The new design technique has the flexibility to accommodate actuator saturation and avoid integral wind-up in the transient response. Based on the actuator's upper limit of saturation, a selection of PI or PID controllers for such plants have been described. For large normalized time delay plants, a new PI tuning scheme based on user defined two-points of the time response curve has been derived. The analysis has also shown the existing limitations of PI control with respect to transient response performance.

Overall the simulation results from the proposed method have shown better performance compared with other available tuning rules. The approximation of higher order systems to first order plus dead time model always presents an error in the estimates. As an example, the estimated lag time always presents a significant error. In example E4 the estimated lag time is 35% inaccurate compared to the true value. However the design is able to provide satisfactory performance despite the imprecise estimate of process parameters. In order to understand the applicability of this tuning strategy, the proposed tuning rules has been used to design a PID controller for controlling the chemical concentration of a nonlinear reactor model [144].

Chapter 6

Two-Level Tuning of Fuzzy PID Controllers

6.1 Introduction

The work so far has enabled the identification of suitable fuzzy systems and provided insight knowledge of tuning linear PID parameters. In Chapter 4 the two level tuning parameters related to different fuzzy systems described in Chapter 3 were identified. The identification was based on two-point non-linearity tuning of fuzzy controllers. The non-linearity variation diagrams were developed for obtaining the first level tuning. In Chapter 5 linear tuning was separately analyzed and new linear PID tuning rules were developed. This chapter develops a design scheme for fuzzy controllers by incorporating both the linear and nonlinear tuning levels.

The tuning of fuzzy controllers is one of the main areas of fuzzy control research. The *curse of dimensionality* during the rule explosion [1] is the main draw-back and fuzzy control designers are still unable to find an effective tuning algorithm. However, the recent increase in computing power enabled most designers to adopt numerical optimization techniques for generating optimum or near optimum solutions to fuzzy systems. The numerical optimization methods have been successfully used for designing linear PID controllers [118] and those designs have been used for wider range of process applications. Recently genetic algorithms and neural network implementations have become more common for solving a larger number

of unknown parameters in fuzzy systems. However, those applications are somewhat specific and unable to generalize for wider process specifications. On the other hand, control experts, who have considerable experience in traditional control, were quite skeptical of using fuzzy logic and as a result the accumulated control experience has not been fully explored for fuzzy controller designs [45]. However the challenge of using fuzzy control has been accepted by many leading researchers and today many scientists and engineers are attempting to link the traditional control theories to the design of fuzzy controllers [93, 43, 116, 145, 146].

Researchers have attempted to generalize design techniques for conventional fuzzy PID controller [16, 43, 117, 56, 58]. In a few cases the tuning aspects have been discussed [56, 58]. A PI tuning scheme for the apparent linear gain (ALG) terms of a conventional fuzzy PI controller is described in [147]. The tuning of ALG terms of the simplest conventional fuzzy PI controller are described in [145, 146]. The latter work uses the conventional gain-phase margin (GPM) approach [145] and the extended circle criterion [146] to derive the tuning formulas for the ALG terms. The tuning schemes in [147, 145, 146] have shown better results than the linear PI controllers. In [148] the non-linearity tuning is identified by classifying the rules into two groups; responsiveness and stability aspects of the control. Also the effect of scaling and membership adjustments have been described for the conventional type fuzzy controllers. In [56] the design of conventional fuzzy PID is identified as a two-level tuning problem and described a way of obtaining ALG terms for conventional fuzzy PID type controllers. However, the non-linearity tuning was not sufficiently or explicitly described for implementing a two-level tuning task. The past tuning schemes are applicable only for conventional type fuzzy PI controllers and have not been extended to PID type fuzzy controllers. This is due to the poor functional properties that exist with the Mamdani type fuzzy PID controller as described in Chapter 3. Generally it is a hard analytical task to identify the non-linearity tuning for general fuzzy PID structure. Therefore in most cases the fuzzy systems are simplified to a greater extent in order to obtain simplified fuzzy output expressions for the analysis [145, 146].

In this chapter the two-level tuning methodology is employed to devise a convenient procedure for tuning fuzzy controllers and to aid in implementing for real-time process control. In this chapter, three different tuning and designing methods are described. The first approach uses the genetic search technique, the second uses a trial and error solution, while the

third uses the two-level tuning principle. Finally, both the linear and fuzzy controllers are tuned for controlling the temperature of a soil cell.

6.1.1 Problem Definition

For fuzzy controller design consider the class of problems that have a monotonic open-loop response except for the initial time period where such processes can be crudely approximated by first-order plus dead time process model as described in Chapter 5.

6.1.2 Fuzzy Controller Systems

For demonstration of the tuning strategies, three fuzzy controller systems that have been described in Chapters 3 and 4 are redefined. The design parameters related to the two tuning levels are also identified.

FCS-I

The system is defined as follows:

Fuzzy PID system: Type VI structure.

Fuzzy logic system: NLFLC-IIIB system.

This structure uses the nonlinear error mapping to produce three fuzzy proportional actions (Chapter 3). The NLFLC-IIIB system uses two non-linearity tuning parameters to obtain the two-point control as described in the Section 4.3.3. Therefore for the three PID actions the first-level tuning parameters are defined as follows.

PID action	Error variable	First-level tuning parameters
absolute P	e	$s_{1,\mathrm{P1}}, s_{2,\mathrm{P1}}$
absolute I	e	$s_{1,\mathrm{P2}}, s_{2,\mathrm{P2}}$
absolute D	e	$s_{1,\mathrm{P3}}, s_{2,\mathrm{P3}}$

The ALG terms for this fuzzy PID controller as described in Table 3.8 are rewritten here for clarity.

$$K_{\mathrm{Pa}} = S_u S_e K_{\mathrm{P}}$$

$$K_{\mathrm{Pa}} = S_u S_e K_{\mathrm{I}}$$

$$K_{\mathrm{Pa}} = S_u S_e K_{\mathrm{D}}$$

FCS-II

The system is defined as follows:

Fuzzy PID system: Type I structure.

Fuzzy logic system: NLFLC-I system.

In section 4.3.1 the NLFLC-I has been simplified to obtain two-point control by changing the input membership parameters. The non-linearity tuning parameters that related to each axis in the error space determines the corresponding nonlinear PID actions in the dissociated form. Therefore the non-linearity tuning parameters that related to simplified NLFLC-I system for two point control can be redefined as follows.

PID action	Error variable	First-level tuning parameters
incremental P	Δe	$(s_1)_2, (s_2)_2$
incremental I	e	$(s_1)_1, (s_2)_1$
incremental D	$\Delta^2 e$	$(s_1)_3, (s_2)_3$

The ALG terms for this fuzzy PID is described in Table 3.8. The three terms are rewritten here for clarity.

$$K_{Pa} = (S_u S_{ce} d_3)/a_2$$

$$K_{Ia} = (S_u S_e d_3)/(a_1 T_s)$$

$$K_{Da} = (S_u S_{rce} d_3 T_s)/a_3$$

From the two-point control design shown (section 4.3.1), $a_1 = a_2 = a_3 = 1/3$ and $d_3 = 1/9$. For unit-step control set $S_e = 1$.

FCS-III

The system is defined as follows:

Fuzzy PID system: Type III structure.

Fuzzy logic system: NLFLC-I system.

The FCS-III uses the conventional Mamdani-type fuzzy PID structure. Due to the two error term definitions, the PID actions are defined either in absolute PD or PI form. The two-input NLFLC-I that has been simplified for the two-point control in Chapter 4 is again used for defining the first-level tuning parameters. Therefore the first-level tuning parameters are redefined as follows.

PID action	Error variable	First-level tuning parameters
absolute P or incremental I	e	$(s_1)_1, (s_2)_1$
incremental P or absolute D	Δe	$(s_1)_2, (s_2)_2$

The apparent linear gains for this fuzzy PID as described in Table 3.8 are rewritten here for clarity.

$$K_{\mathrm{Pa}} = d_2 S_u \left(S_{ce} K_{\mathrm{PI}} / a_2 + S_e K_{\mathrm{PD}} / a_1 \right)$$

$$K_{\mathrm{Ia}} = \left(S_u S_e d_2 K_{\mathrm{PI}} \right) / (a_1 T_s)$$

$$K_{\mathrm{Da}} = \left(S_u S_{ce} d_2 K_{\mathrm{PD}} T_s \right) / a_2$$

From the two-point control design shown (section 4.3.1) , $a_1 = a_2 = 1/3$ and $d_3 = 1/6$. For unit-step control set $S_e = 1$. For realizing a three linear term tuning, force $S_u = 1$.

6.2 GA Based Designs

The genetic algorithm (GA) is a numerically driven search algorithm and the searching mechanism uses mechanics of natural selection and natural genetics to evolve a better (or optimum) solution [149]. The method is easy to implement in computers and requires no local derivatives to guide the search process. The method evolves the candidate solutions for the problem by iteratively applying a set of stochastic operators, namely reproduction, crossover and mutation. In this work the Matlab program developed in [150] is used for optimizing the tuning parameters of fuzzy controllers. The program uses the simple genetic algorithm described in [149]. The implementation of this numerical search uses the following seven steps.

1. *Parameter coding:* The input variables to the search engine are first translated into binary bit strings. String lengths can be determined by the maximum and minimum allowable values of each input variable. Longer string length increases the resolution.

2. *Generation:* The process begins by randomly generating an initial population of strings, each having the bit length specified in 1. The population size is a compromising factor. Larger value increases the possibility of including the optimum solution in the first few generations, but with the expense of computation time. This possibility can be increased by having shorter bit-string length, but with reduced accuracy. This particu-

lar contradiction (resolution versus running time) does not allow the GAs to reach the global optima.

3. *Fitness evaluation:* The input parameters that have been generated in the generation process is then sent to the performance test. In this case, the parameters are used to control the given process model and the quality of the response performance is evaluated against a given fitness or objective function.

4. *Reproduction:* Reproduction is a process by which the strings (or inputs) with larger fitness values are selected with higher probabilities. The most common method is the weighted roulette selection [149]. First the previous generation is ranked according to the percentage of total fitness. The roulette wheel is weighted according to the percentage of fitness. Spinning the wheel reproduces a new population. The strings having higher weight in the roulette wheel are reproduced with a higher probability. This procedure automatically purges out the bad solution candidates and the new population eventually produces higher total fitness in the next trial than the previous population.

5. *Crossover:* After reproduction, crossover is performed in two steps. First the members of the newly reproduced strings in the mating pool are mated at random. Each pair of strings that has been selected for mating undergoes crossover by swapping the bit characters. In order to do that an integer position along the string is selected uniformly at random.

6. *Mutation:* Mutation is the occasional random alteration of the value of a string position. This process will reinforce the chance of reaching the optimal point. The mutation plays a secondary role in the GA and therefore the frequency of mutation is usually kept with a small probability.

7. *Iteration:* The GA iterates by repeating the processes 3-7 until it arrives at predetermined ending conditions. The iteration is usually set to stop after reaching minimum error conditions in the successive iterations or after a predetermined number of cycles.

6.2.1 Formulation of the FLC for GA

The first part of this exercise is to formulate the fuzzy or linear PID control problem for running the GA. The GA based design is tested only for the FCS-I. It was shown in Chapter 3 that the type VI structure has better functional properties than the conventional and other coupled fuzzy PID structures. For convenience the equation (3.16) is repeated while including the sampling time T_s.

$$u(n) = S_u \left(K_{\mathrm{P}} \hat{u}_{\mathrm{P1}}(n) + K_{\mathrm{I}} T_s \sum_{q=0}^{n} \hat{u}_{\mathrm{P2}}(q) + \frac{K_{\mathrm{D}}}{T_s} \left(\hat{u}_{\mathrm{P3}}(n) - \hat{u}_{\mathrm{P3}}(n-1) \right) \right) \tag{6.1}$$

When the fuzzy proportional outputs in (6.1) are set to exact linear form and set equal to the normalized error, $\hat{e}$, then the implementation is a linear PID controller. In Chapter 4 it was shown that this particular fuzzy system has better nonlinear properties for the nonlinear tuning. In order to accommodate the output to be fully normalized within [-1,1] (refer preferred features in the Section 4.2.2), the output expressions that are shown in the equations (3.62) have been further divided by the maximum output value. The three fuzzy proportional actions are expressed as functions of the nonlinear tuning parameters, where

$$\hat{u}_{\mathrm{P1}}(n) = f\left(\hat{e}(n), s_{1,\mathrm{P1}}, s_{2,\mathrm{P1}}\right), \ \hat{u}_{\mathrm{P1}}(n) \in [-1, 1]$$

$$\hat{u}_{\mathrm{P2}}(n) = f\left(\hat{e}(n), s_{1,\mathrm{P2}}, s_{2,\mathrm{P2}}\right), \ \hat{u}_{\mathrm{P2}}(n) \in [-1, 1]$$

$$\hat{u}_{\mathrm{P3}}(n) = f\left(\hat{e}(n), s_{1,\mathrm{P3}}, s_{2,\mathrm{P3}}\right), \ \hat{u}_{\mathrm{P3}}(n) \in [-1, 1]$$

The general upper and lower bounds of the non-linearity tuning parameters required for the GA are defined as,

$$0 < s_1 \leq 1, \ -1 < s_2 < 1.$$

For convenience the valid ranges for s_1 and s_2 that define the NLFLC-III controller are included in the fitness function of the GA system.

The four linear gains in (6.1) can be simplified to three terms by forcing one of the variables to unity. However, the GA implementation requires these values to be specified within a predetermined range. Therefore without loss of generality the following ranges are specified for the variables. Assume unit-step response of a closed-loop PID control system.

- From the apparent gain analysis (refer section 3.4.4) $S_e = 1/e_{\max} = 1$.

- Considering the actuator saturation the range of the de-normalizing factor is set to: $0 < S_u \leq U_u$.

- The three apparent linear gains are normalized to: $0 \leq K_{\mathrm{P}} \leq 1,\ 0 \leq K_{\mathrm{I}} \leq 1,\ 0 \leq K_{\mathrm{D}} \leq 1$.

Overall the fuzzy system is reduced to a total of ten tuning parameters to solve in the GA; six non-linearity tuning parameters and four linear tuning parameters. Suppose J_T is the overall performance index of the system, then for the fuzzy PID controller above,

$$J_T = f\left(K_{\mathrm{P}}, K_{\mathrm{I}}, K_{\mathrm{D}}, S_u, s_{1.\mathrm{P}1}, s_{2.\mathrm{P}}, s_{1.\mathrm{P}2}, s_{2.\mathrm{P}2}, s_{1.\mathrm{P}3}, s_{2.\mathrm{P}3}\right). \tag{6.2}$$

In optimum PID designs the error integral criteria is most common [118, 124, 125, 126]. In this exercise the overall performance is defined using three indexes, namely integral of the square of the error (ISE), peak overshoot (OS) and settling time(T_{st}). With suitable weightings the overall performance can be written as,

$$J_T = w_1 \frac{\mathrm{ISE}}{\max e(n)} + w_2 OS + w_3 \frac{T_{st}}{T_f}. \tag{6.3}$$

Where T_f is the total simulation time. The weighting values can be adjusted according to the performance specifications. As an example, if overshoot is more critical than other two performance indices, a greater value for the weighting w_2 can be allocated. The GA developed in [150] uses the maximization of the fitness function. Therefore the fitness function is defined for the valid ranges of the non-linearity tuning parameters as:

$$Fitness = \begin{cases} 0 & s_2 < 0 \text{ and } |s_2| > s_1 \\ \frac{1}{1+J_T} & \text{otherwise} \end{cases}. \tag{6.4}$$

6.2.2 Simulations Using GA

In this section numerical studies are performed to examine the applicability of GA for process control. For simulations the approximated first-order process parameters of the examples in Chapter 5 (examples E2, E3, E4 and E5) are tested. In each trial of GA the fitness function in (6.4) is evaluated against the unit-step response of the approximated model and equal weighting is imposed for each performance index ($w_1 = w_2 = w_3 = 1$). The following parameters are kept constant for each test.

Table 6.1: The GA based optimum fuzzy PID parameters

Example	Process Parameters	Optimized Parameters	
	k, T, t_d, U_u	$s_{1,P1}, s_{2,P1}, s_{1,P2}, s_{2,P2}, s_{1,P3}, s_{2,P3}$	K_P, K_I, K_D, S_u
E2	2, 5.72, 0. 4	0.0412, 0.0812, 0.5089, -0.2635, 0.5089, -0.1069	0.9685, 0.3937, 0.0472, 3.97
E3	1, 1.5971, 0.6589, 2	0.9688, -0.7336, 0.7350, -0.1695, 0.9922, 0.4886	0.9528, 0.5276, 0.0157, 1.7166
E4	1, 1.232, 1.343, 2	0.4231, 0.6766, 0.8597, -0.0598, 0.9065, 0.4102	0.4961, 0.2913, 0.1417, 1.8268
E5	1, 1.521, 4.462, 1.5	0.7116, -0.3576, 0.3140, 0.5669, 0.5869, -0.2949	0.9370, 0.2283, 0.7480, 0.6732

Population Size = 100

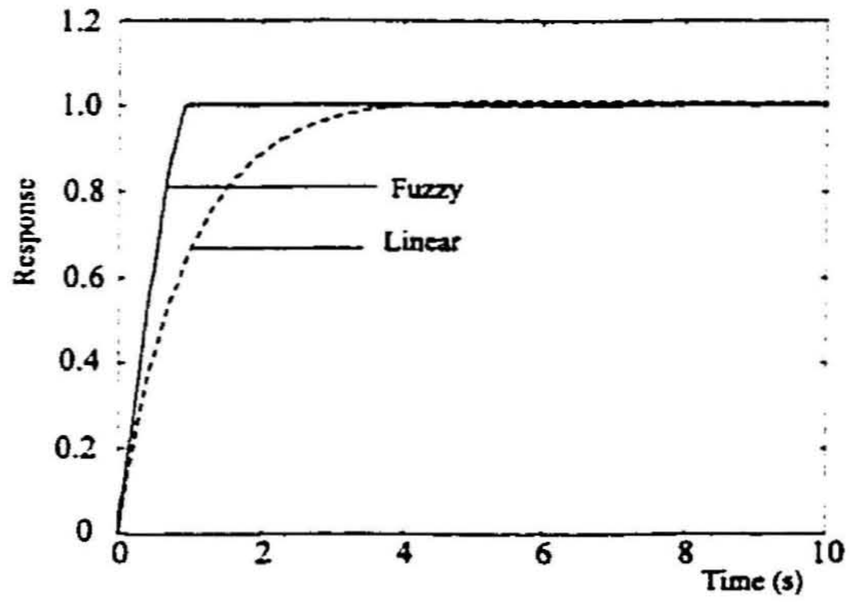
Number of generations = 100

Cross-over factor = 0.9

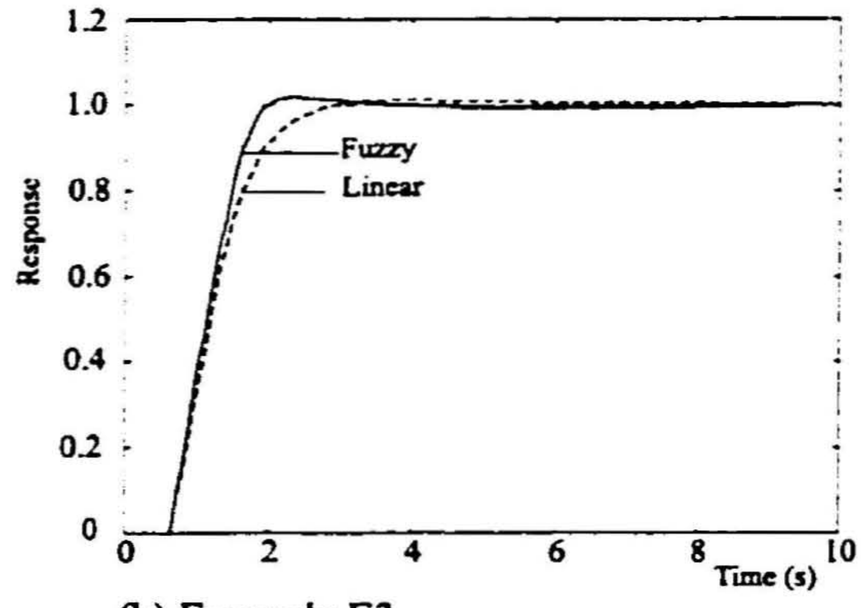
Mutation factor =0.05

bit-length = 7

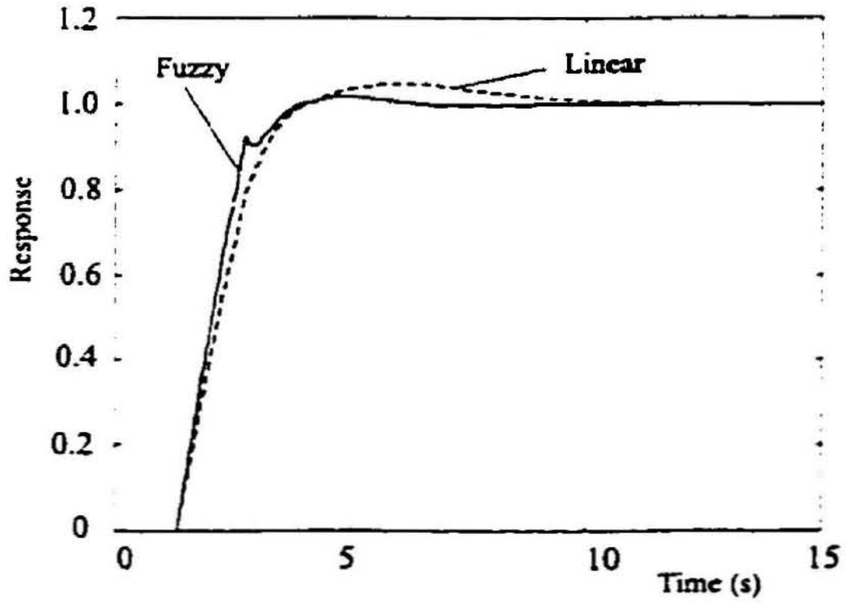
The step response curves corresponding to the theoretical process parameters are shown in Figure 6.1. For comparison the response curves of the linear PID control are also shown. Table 6.2 shows the performance indexes defined in the fitness function. For comparison the performance indexes corresponding to the linear PID controller are also shown. The linear PID parameters are based on the proposed tuning scheme. Overall the fuzzy controllers perform better (greater fitness) than the linear PID controller. However the GA does not converge to the global optimum. This is due to the small population size and the low resolution. Also the definition of a proper fitness function plays a significant role in the final solution. As an example, although the overall fitness of the fuzzy response shown for the example E5 has a superior value, the overshoot is considerably higher than the linear PID controller. This can be avoided by allocating greater weight to the overshoot.



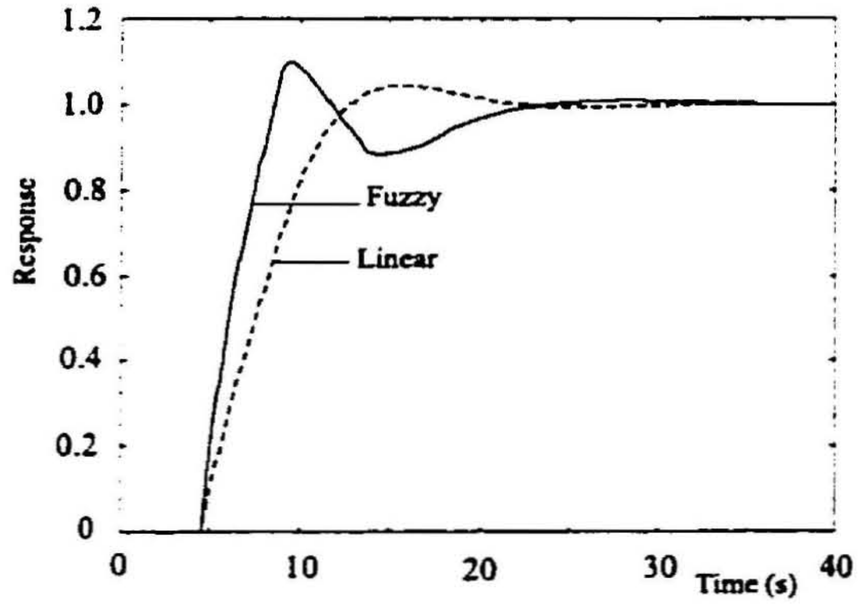
(a) Example E2



(b) Example E3



(c) Example E4



(d) Example E5

Figure 6.1: Unit step response curves of the approximated process models. The optimum fuzzy controller PID parameters are based on genetic designs

Table 6.2: Performance comparison of GA based fuzzy PID and linear PID controllers

Example	Fuzzy PID			Linear PID		
	ISE	T_{st}	$OS\%$	ISE	T_{st}	$OS\%$
E2	0.2931	0.95	0.36	0.49	3.2	0.97
E3	1.011	1.90	1.56	1.058	2.6	0.87
E4	1.859	3.6	1.62	1.957	8.4	4.58
E5	5.673	21.1	9.85	6.567	19.2	4.22

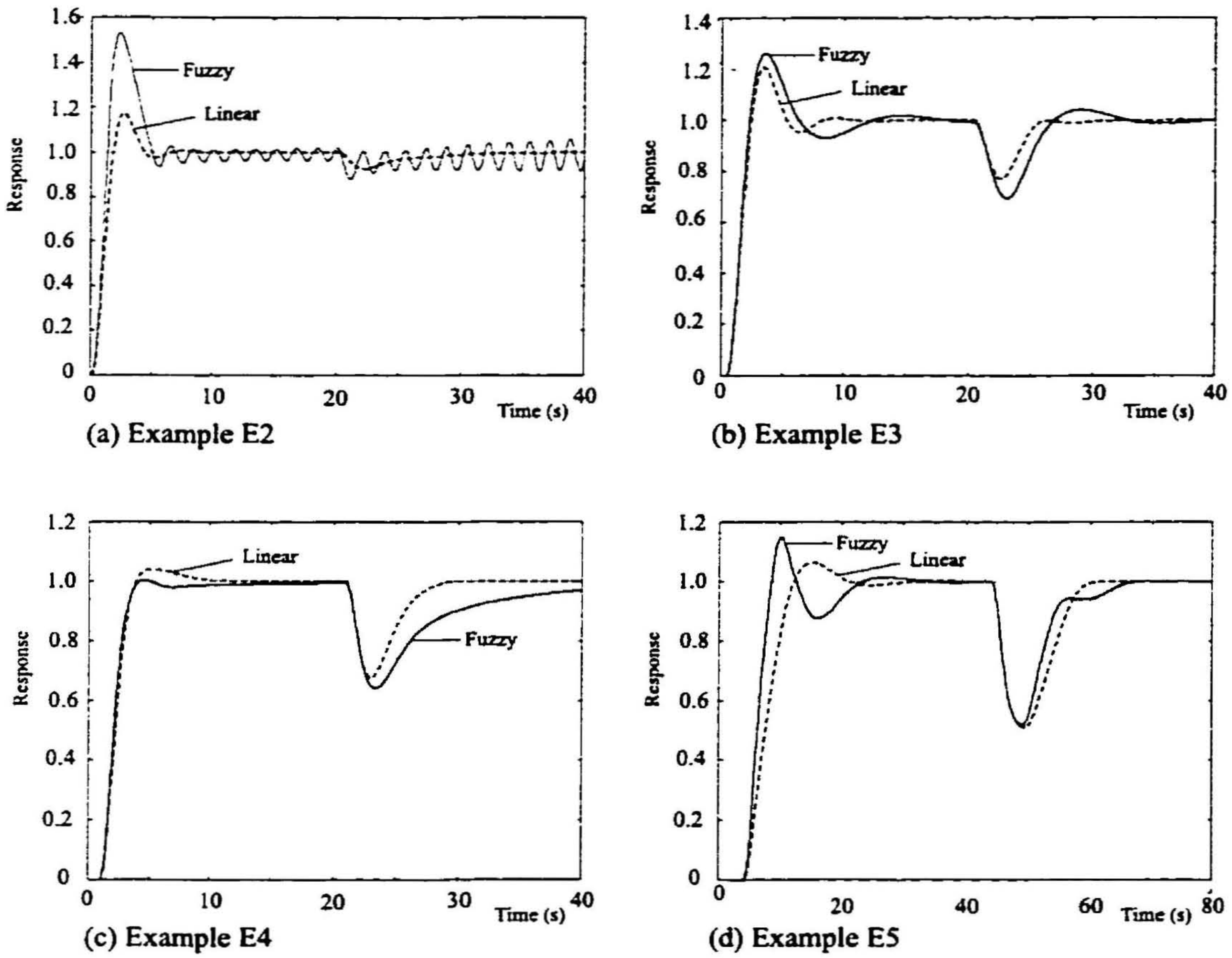


Figure 6.2: Unit step response curves of the actual process models. The optimum fuzzy controller PID parameters are based on genetic designs

The GA based fuzzy controllers are then simulated to observe the actual process responses. For each case a constant 50% load disturbance also has been added. The response curves for the four examples are shown in Figure 6.2. From the curves it can be easily seen that the GA based fuzzy controllers were unable to perform any better than the linear controllers when they were implemented for the actual process system. The reasons and drawbacks are summarized as follows.

- The fuzzy PID parameters have been optimized only for the approximated process model and therefore the fitness function will become no longer valid for the unknown plant dynamics that exist in the true system. The numerically optimized fuzzy controllers can only be used when the actual process model is completely known *a priori*.

The alternative is to use the GA on-line. The GA works by randomly generating numbers. Then each combination is tested against the fitness. Such a random trial can be allowed only if the system can be allowed to run with no hazard either to the system itself or to its surroundings.

- The optimized linear PID parameters (using either GA or any other numerically driven search [118]) usually work well with the true plant system. This is due to the conservative linear control policy. With two-point local control in the above fuzzy controllers, the PID actions can take severe control actions at the chosen control points. As an example, the optimized non-linearity for u_{P1} in example E2 has a very high gradient near the zero error or equivalently the apparent linear proportional gain near zero has become extremely large. This leads to a low relative stability of the system (lower gain margin) and as a result the system response is more oscillatory under the unknown perturbations. The same argument can be extended for other local control points.

- The optimization through GA and the expected performance is greatly affected by the definition of the fitness function. As an example the performance of example E4 shows that the fuzzy controller has very poor load disturbance properties compared to the linear PID system. This is due to lower apparent nonlinear integral action near zero error. This can be overcome by including the load disturbance rejection properties into the fitness function. However, the PID control has contradictory policy between the set-point control and load disturbance characteristics. In most cases two controlling phenomena have been separately taken for designing PID parameters [118].

6.3 Two-level Design Based on Trial and Error Tuning

This section attempts to tune the fuzzy controllers using trial and error technique. Again the same fuzzy PID system described by the FCS-I is tested with simulations. In the GA based designs, both the tuning levels are taken simultaneously for tuning. When the two tunings are taken separately, there are two possibilities. In the first alternative the non-linearity tuning parameters (or first-level tuning) can be decided and then the ALG terms can be adjusted until satisfactory performance has been achieved. The second alternative is to find the best

first level tuning under given second level tuning (or apparent linear gains). In this section the first alternative is performed. The second alternative is shown in the next section.

The non-linearity is selected using the non-linearity tuning diagrams that were described in Chapter 4. For the NLFLC-IIIB system consider the tuning diagram that is shown in Figure 4.4. The trial and error tuning is performed in two levels.

- Using the non-linearity tuning diagram (Figure 4.4), select the non-linearity tuning parameters (s_1 and s_2) for each PID action.
- Then adjust the apparent linear PID gains until a desired (or satisfactory) performance level is reached.
- Repeat the above two steps until the controller gives superior results compared with those of the linear counterpart.

For this demonstration fuzzy PI controllers are tuned for the process system that was described in example E3. For the fuzzy controller described in the equation (6.1), the apparent linear PID gains are adjusted by modifying the PID weightings that were described in Chapter 5. For clarity the apparent linear PI gains in (5.38) are redefined in terms of the approximated process parameters as,

$$K_{\mathrm{P}} = \rho_f \frac{T}{t_d}, \; K_{\mathrm{I}} = \frac{\gamma_f}{t_d} \tag{6.5}$$

Where ρ_f and γ_f are new proportional and integral weightings respectively chosen for the fuzzy PI controller. Force the derivative term $K_{\mathrm{D}} = 0$ for realizing a PI controller. For comparison the pole zero cancellation linear PI controller (refer section 5.4.2) is chosen.

Using a trial and error procedure for the two levels of tuning, three acceptable fuzzy PI controllers are chosen. The first two controllers (design 1 and design 2) are designed for obtaining improved response for the set-point control while the third design (design 3) is chosen to obtain both improved set-point control and improved load disturbance properties. The tuned parameters of the three designs and the linear PI design are shown in Table 6.3.

When performing the trial and error fuzzy PI design the following should be noted.

- Theoretically it is possible to select any arbitrary non-linearity tuning parameters (first level) for each PID action and then adjust the PID weighting until desired performance is reached. However such a design cannot always be better than the linear controller.

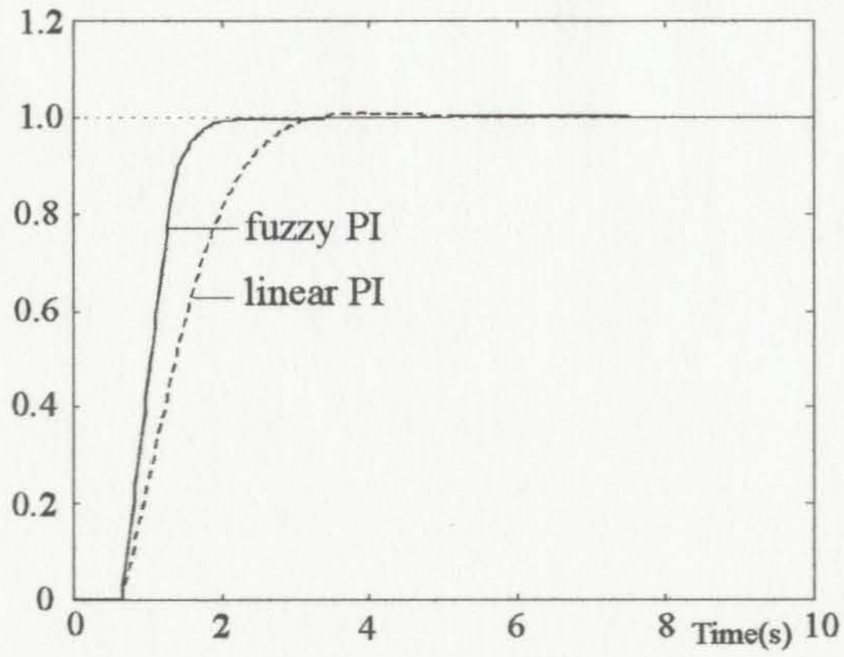
Table 6.3: The trial and error based fuzzy PI parameters for the example E3

Design	First-level tuning		Linear PID
	$s_{1,P1}, s_{2,P1}$	$s_{1,P2}, s_{2,P2}$	ρ_f, γ_f
design 1	0.9, 0.6	0.5669, -0.2	0.86, 0.61
design 2	0.95,-0.9	0.5669, -0.2	0.58, 0.53
design 3	0.5, 0.5	0.45, -0.3	0.64, 0.46
linear	–	–	0.45, 0.45

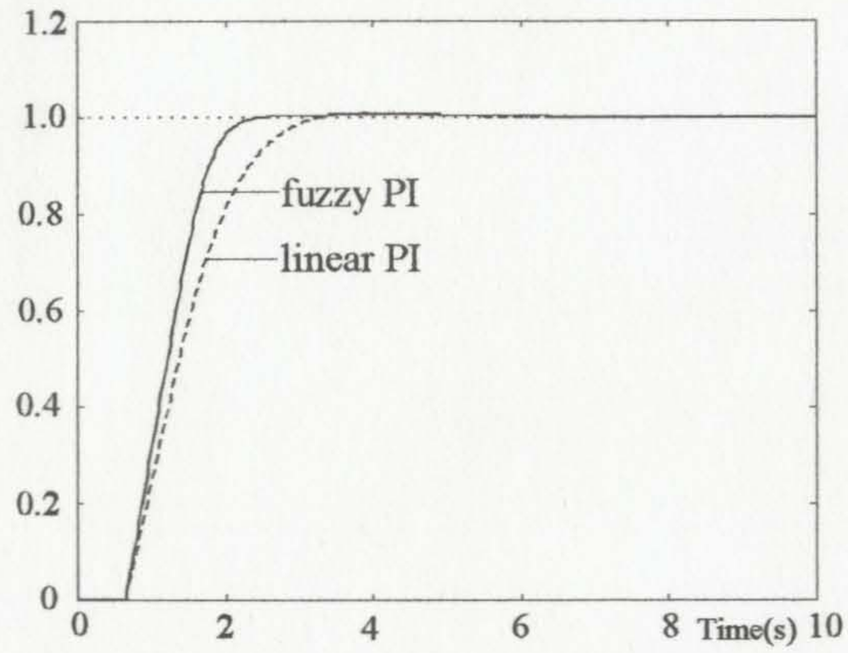
- In both the set-point control designs (design 1 and 2), the non-linearity for the proportional action is set with $\theta_0 < 45^o$, so that the control action which causes excessive overshoot near zero becomes milder to keep the off-set minimum. Also for the set-point control, the integral actions are kept approximately linear (point **O** in Figure 4.4).

- At steady state the integral action is more dominant. The milder proportional action near zero causes the load disturbance response to have a higher overshoot. Therefore for better load disturbance properties the integral action near zero is increased ($\theta_0 > 45^o$), while the proportional action near zero is kept approximately linear ($\theta_0 \approx 45^o$).

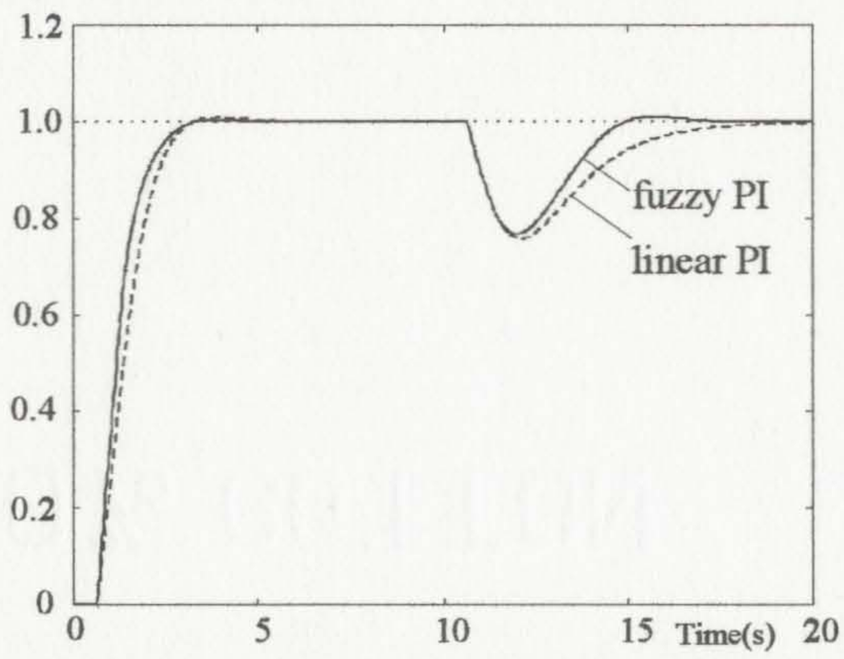
The response curves corresponding to the above designs are shown in Figure 6.3. For each case the linear PI design is shown for comparison. Overall the three designs perform better than the linear PI controller when they are simulated for the approximated first-order process model. Design 3 shows better load disturbance rejection properties while maintaining better step response characteristics. However when the three designs are tested on the actual process system in example E3, none of the fuzzy controllers was able to produce any better performance than the linear PI controller. As shown in Figure 6.3(d), the three responses corresponding to the three designs have shown poorer step response and load disturbance properties. Similar to the GA based designs, the trial and error optimization only accounts for the approximated model. When the system is subjected to the unknown plant dynamics, the fuzzy controllers perform poorer than the linear controller.



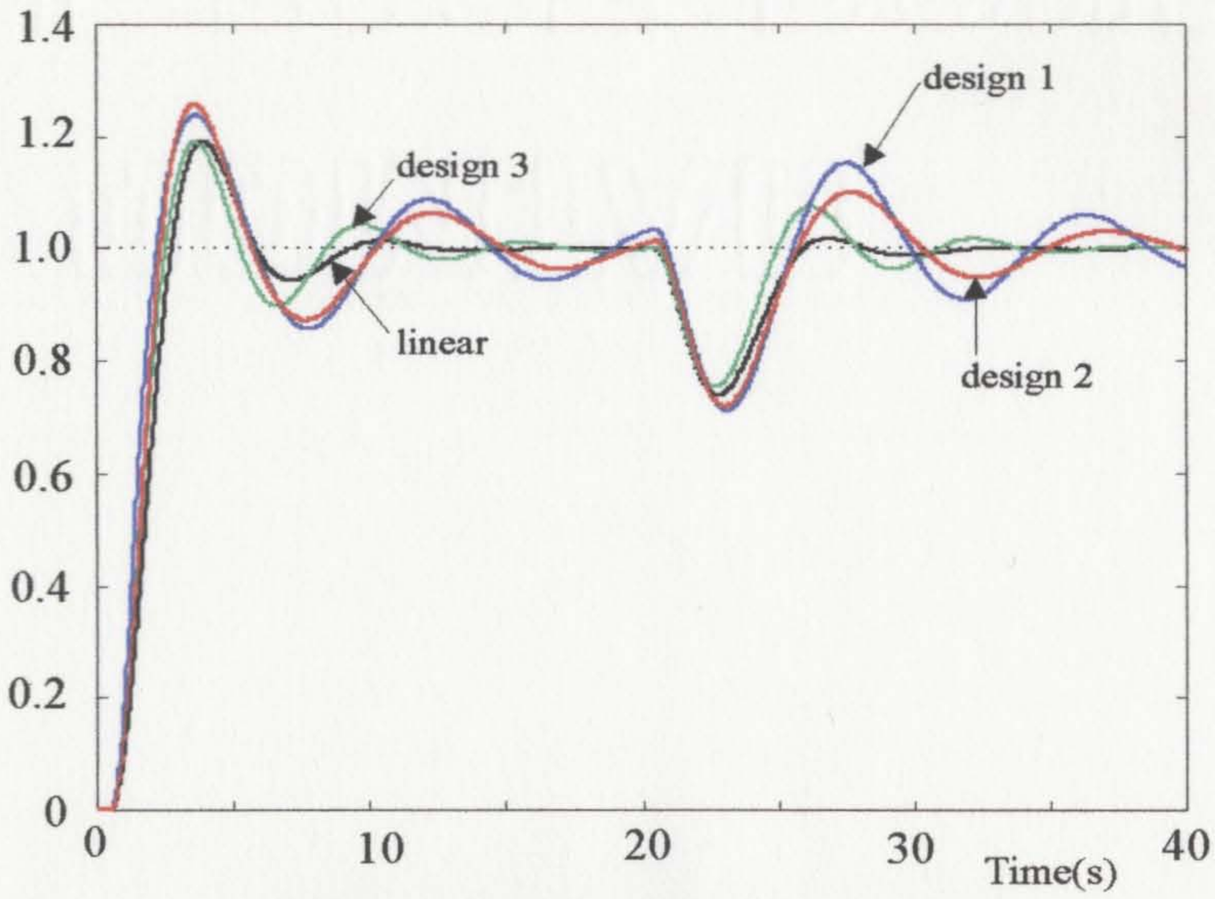
(a) Responses of the approximated model (**design 1**)



(b) Responses of the approximated model (**design 2**)



(a) Responses of the approximated model (**design 3**)



(d) Responses of the actual process model

Figure 6.3: Unit step response curves of Example E3 based on trial and error designs.

6.4 Two-level Design Based on Tuning Heuristics

After identifying the draw-backs in the previous fuzzy controller tuning methods, this section aims at developing a two-level tuning criteria for obtaining improved performance when they are implemented on actual process systems. From the above two methods it is very clear that the off-line fuzzy controller designs can be effectively used only when the process parameters are completely known. The non-linearity tuning has a significant effect on the true dynamic system. On the other hand the off-line linear PID designs are acceptable for real-time implementations. The designs using Ziegler-Nicholos (ZN) based tuning [122] have been successfully implemented in many industrial control applications [140, 121]. The numerically optimized designs [118] and trial and error optimized designs [127] have also recently shown improved linear PID control. This is mainly due to the conservative linear control policy applied in the first-level tuning. The tuning rules that are developed in [145, 146] never attempt to define non-linearity tuning. Instead an insignificant non-linearity (or slightly off from the linear) for the PI control has been used through a simplified fuzzy PI algorithm.

While observing the simplified nature of the linear PID control, the two-level tuning is developed to address three main objectives. Firstly, the design should be simple and easy to understand for direct implementation. Secondly the overall performance should be always better than that with the linear PID controller. Thirdly, the method should be applicable for wider range of process specifications.

In this proposed tuning strategy the tuning is performed in the reverse order. The second-level linear tuning is first performed followed by the first-level non-linearity tuning. The tuning heuristics related to two tuning levels are described next.

6.4.1 First-level Tuning Heuristics

As the linearity tuning is initially fixed, the effects of non-linearity tuning under given linear gains are first studied. Set-point response and response due to a disturbance are carefully analyzed for first-order processes. Keeping the linear gains at a constant level the two slope angles θ_0 and θ_1 are varied to obtain different non-linearity. Using a known linear tuning formula the linear PID controllers can be initially tuned. Therefore the non-linearity is adjusted after the given fuzzy system has been tuned to produce the linear or approximately

Table 6.4: First-level tuning heuristics under constant linear gains. SR: speed of response, OS: overshoot

PID action	Effect of θ_0		Effect of θ_1	
	Set-point response	Response under load disturbance	Set-point response	Response under load disturbance
Proportional	Increase OS	Slower SR	Slower SR	Minimum effect
Integral	Increase OS	Increase SR	Slower SR	Minimum effect
Derivative	Add damping	Add damping	Increase OS	Minimum effect

linear control functions. The effects of varying the two angles from the linear approximation point **O** (refer Figure 4.3-4.7) on the set-point control and load disturbance characteristics are then observed. These observations are tabulated and shown in Table 6.4 as *first-level tuning heuristics.*

6.4.2 Second-level Tuning Heuristics

From the previous two methods it has been observed that the linearity tuning for the desired performance depends on the first-level tuning. As an example the two trial and error designs (design 1 and 2) in the previous section have shown that changing the non-linearity in nonlinear proportional control gives different sets of linear gains for the set-point control. Since the linear design is given the priority in this tuning sequence, the apparent linear gains are chosen while using the conventional linear control knowledge. Therefore the apparent linear PID parameters for this particular class of problems are selected using the tuning rules developed in Chapter 5.

6.4.3 Two-level Tuning Strategy

The two levels are performed in the following sequence.

- First the fuzzy controller is tuned to produce the approximated linear PID function and the apparent linear gains are selected using available linear PID tuning rules. The linear PID or the approximated linear fuzzy PID controller is implemented for the

actual process and its true response is observed.

- Using the first-level tuning heuristics the non-linearity tuning is performed to obtain an improved response without changing the linear gains. This is accomplished by moving the point corresponding to the approximated linear control in the non-linearity tuning diagram while following the tuning heuristics that are given in Table 6.4. If the original response from the linear PID control is satisfactory, then no adjustments are necessary. A linear controller may be used without resorting to any fuzzy system.

The above method has the following three main features. First, the design utilizes the well-developed linear PID technology and therefore its applicability can be widened to many industrial control problems where the linear PID controllers are currently used. Secondly, the performance of the fuzzy control can be guaranteed to be better or equal to the linear PID controller, as the linear controller is a special case of the nonlinear fuzzy controller. Finally, the fuzzy controller application would be stable as long as the linear controller is stable for the particular application. Generally the stability of fuzzy controllers are determined by the linear gains of the closed-loop control system, as the non-linear gains have bounded behaviour in the normalized control output space. The tuning criteria suggests a closed-loop stable linear controller design. The non-linearity tuning is performed mainly to improve the relative stability of the corresponding linear system. As the relative stability has been improved by the higher level tuning the fuzzy controller would eventually becomes a more robust controller than its linear counterpart. However, a precise affirmation is possible only if a quantitative analysis is performed on the stability aspects of fuzzy controllers.

6.4.4 Simulation Examples

This section demonstrates the above two-level tuning through numerical simulations. The FCS-I system has been already used in the above two sections of this chapter for demonstrating the weaknesses in genetic based and trial and error based designs. The FCS-II and FCS-III systems are further added to demonstrate the generalization of the methodology for other fuzzy systems and also to show some drawbacks that exist in the rule-coupled fuzzy systems. Further, each fuzzy system is again considered for the two-point control as described in Chapter 4.

Example E2

This example from Chapter 5 has a negligible time delay and according to the linear PID analysis PI is optimum. The PI version of the Type I structure is obtained by forcing the second error derivative gain to zero and is equivalent to two-input Mamdani-type controller. Therefore only FCS-I and FCS-III systems are tested for this example.

Tuning FCS-I

According to the new tuning procedure, first the FCS-I is set to work at its most linear position (point O in the non-linearity tuning diagram in Figure 4.4) and the apparent linear gains are set using the linear gain results of the same example shown in Chapter 5. This design had an ideal set-point overshoot of 2%. The maximum controller signal had to be kept below the upper gain limit of 4. The actual response showed poor load disturbance characteristics and its step-response overshoot was 17.5%. This behavior is common when the normalized dead time (t_d/T) is relatively small or negligible. It is possible to improve the load disturbance properties by increasing the linear PI gains, but at the expense of higher overshoot in the transient response. Also the actuator limitations would not allow these gains to be increased as desired. The first-level tuning heuristics in Table 6.4 are now used to improve the performance.

1. For improving the slow response behavior in the load disturbance, the integral action near zero or equivalently $(\theta_0)_{\mathrm{I}}$ is increased. The $(\theta_0)_{\mathrm{P}}$ is unchanged.
2. To reduce the overshoot, the response speed is reduced by increasing both $(\theta_1)_{\mathrm{P}}$ and $(\theta_1)_{\mathrm{I}}$.

The first-level tuning parameters and the corresponding points in the non-linearity tuning diagram obtained after two iterations are.

$$[s_{1,\mathrm{P}1}, s_{2,\mathrm{P}1}] = [0.45, 0.5]\,, \quad [(\theta_0)_{\mathrm{P}}\,, (\theta_1)_{\mathrm{P}}] = [\deg 45, \deg 74.5]\,.$$
$$[s_{1,\mathrm{P}2}, s_{2,\mathrm{P}2}] = [0.25, 0.3]\,, \quad [(\theta_0)_{\mathrm{I}}\,, (\theta_1)_{\mathrm{I}}] = [\deg 67.2, \deg 55]\,.$$

The response curves are shown in Figure 6.4. The non-linearity tuning of the fuzzy controller was able to produce excellent load disturbance characteristics without sacrificing the transient response characteristics. The set-point overshoot is also reduced

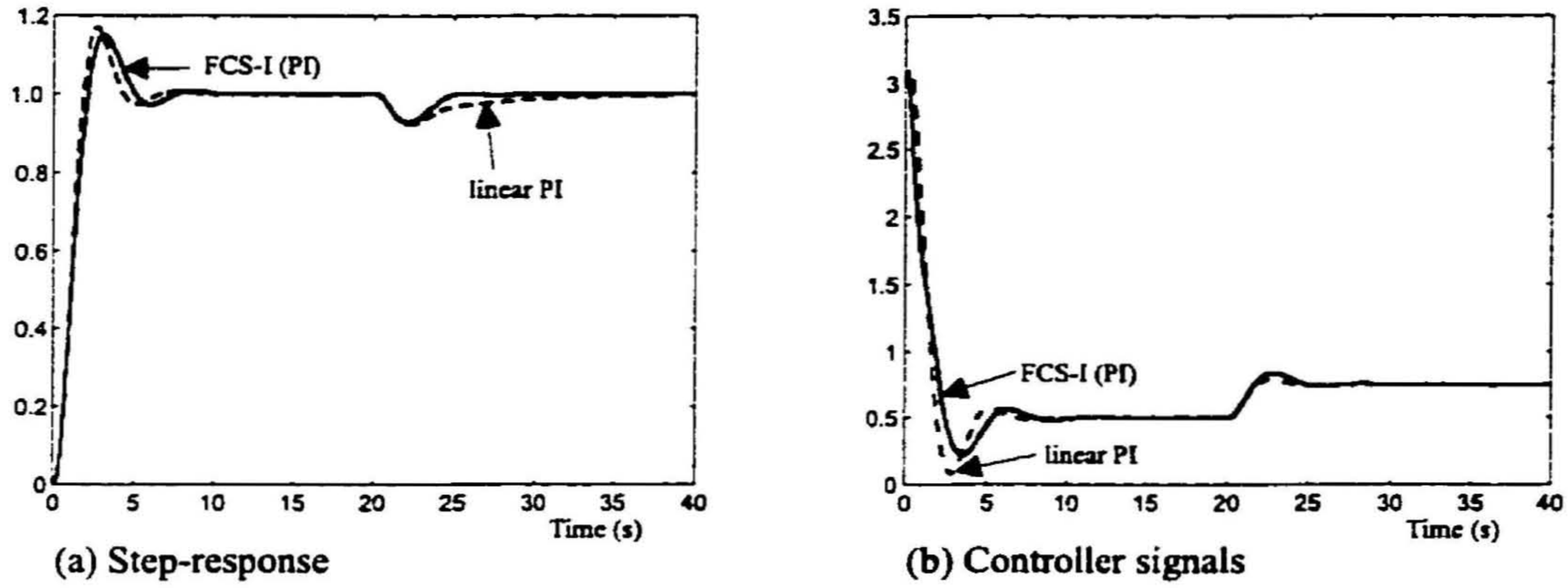


Figure 6.4: Unit step response curves of the process in example E2. The FCS-I parameters are based on two-level tuning

to 15% and the controller is able to keep within the same upper and lower saturation limits of the actuator as shown in Figure 6.4(b).

Tuning FCS-III

The non-linearity diagram of the NLFLC-I system for this controller system is shown in Figure 4.3. According to the definition of the non-linearity tuning ranges, the most linear point O in the diagram is not feasible and the corresponding LLFLC is considered for realizing the most linear approximation. Corresponding to the PI type controller of the PID structure Type-III, force $K_{\mathrm{PD}} = 0$. Using the linear PI tuning data the linear tuning parameters are: $K_{\mathrm{PI}} = 1.2172T_s$, $S_{ce} = 4.92/T_s$. For the LLFLC, set all the input membership functions uniformly distributed over the normalized input space. With this setting the FCS-III system is simulated for the actual system and the corresponding output is shown in Figure 6.5. The response has shown very poor performance compared to the linear PI system of the FCS-I. This is due to the functional drawbacks of rule coupled fuzzy controllers as described in the Chapter 3. Also, the most linear approximation is functionally not equivalent to a conventional linear PID controller. However, using the first-level tuning heuristics the non-linearity tuning is then performed following the tuning heuristics for the projected slope angles. The non-linearity tuning is performed as follows.

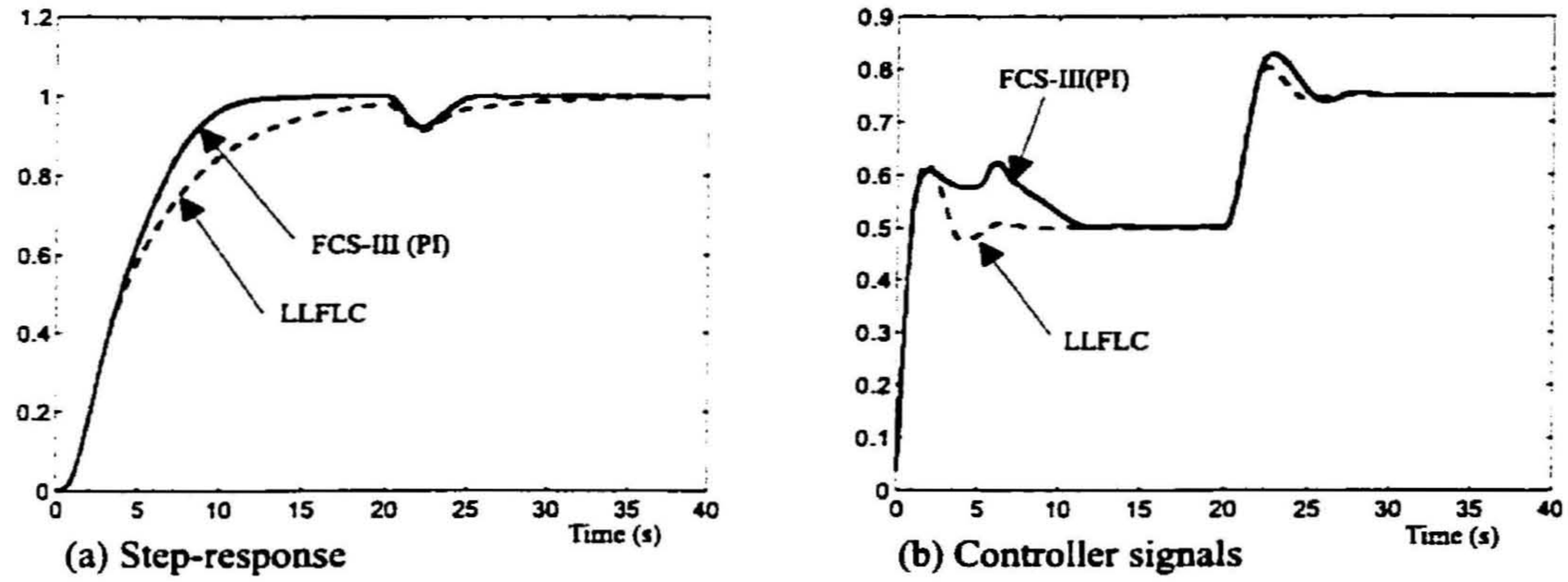


Figure 6.5: Unit step response curves of the process in example E2. The FCS-III parameters are based on two-level tuning

1. To get faster rising, both angles $(\theta_1)_P$ and $(\theta_1)_I$ are decreased.
2. To improve the load disturbance properties, angle $(\theta_0)_I$ is increased and $(\theta_0)_P$ is decreased.

It can be seen that due to the lower representation of the admissible space in the non-linearity tuning diagram that is shown in Figure 4.3, the flexibility of changing the slope angles is somewhat restricted. However, using three iterations the tuning parameters obtained for this FCS-III system are:

$$[(s_1)_1, (s_2)_1] = [0.2, 0.3], \quad [(\theta_0)_I, (\theta_1)_I] = [\deg 68.1, \deg 35.5]$$
$$[(s_1)_2, (s_2)_2] = [0.4, 0.5], \quad [(\theta_0)_P, (\theta_1)_P] = [\deg 51.3, \deg 45]$$

The corresponding response curves are shown in Figure 6.5. With the two-level tuning, reasonable performance is obtained and the non-linearity tuning performs well for load disturbances. Although the overshoot of the tuned controller is minimum, the transient step response is more sluggish and slower compared with the FCS-I system. This is mainly due to the incompatibility of the linear PI tuning terms for the rule based coupled system. Also, the fuzzy controller output is an incremental type as compared to the absolute signal in the FCS-I. To better illustrate this, the linear PI terms of the corresponding LLFLC system are altered and the linear tuning is separately performed for the approximated model. With altered linear terms given by $K_{PI} = 1.2172T_s$,

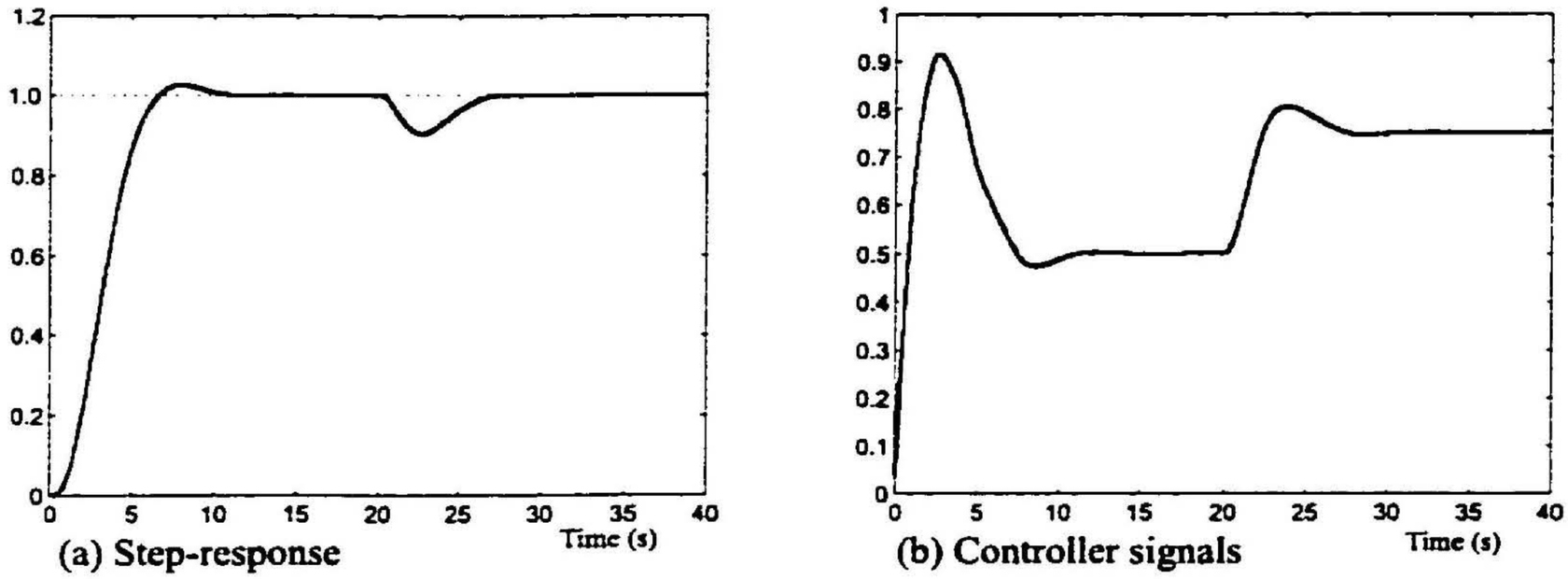


Figure 6.6: Unit step response curves of the process in example E2. The FCS-III is based on the LLFLC structure with altered linear PID gains

$S_{ce} = 2.46/T_s$, the LLFLC is then tested for the actual process system. The response curves are shown in Figure 6.6. The change of linear gains enabled the FCS-III to reach improved performance while maintaining the overshoot of the response within 4% (Figure 6.6). However the speed of response is still low compared to the FCS-I system. As the performance with LLFLC is satisfactory, no more further adjustments to the non-linearity tuning parameters are required.

Example E3

For this example the three fuzzy controllers (FCS) are tuned for obtaining the step response of the second order process of this example. Also actuator saturation limits of [0 2] are imposed for each case.

Tuning FCS-I

As in the previous example, the non-linearity of FCS-I is first set to operate at its most linear position. The apparent gain values are obtained from the linear design shown in Chapter 5. The corresponding linear design is shown in Figure 6.7. In this case the transient response has shown poor performance and the peak overshoot is about 20%. However the load disturbance characteristics have shown satisfactory results. The aim of non-linearity tuning is to improve the transient response characteristics while preserving similar load disturbance properties. The tuning is performed in three steps

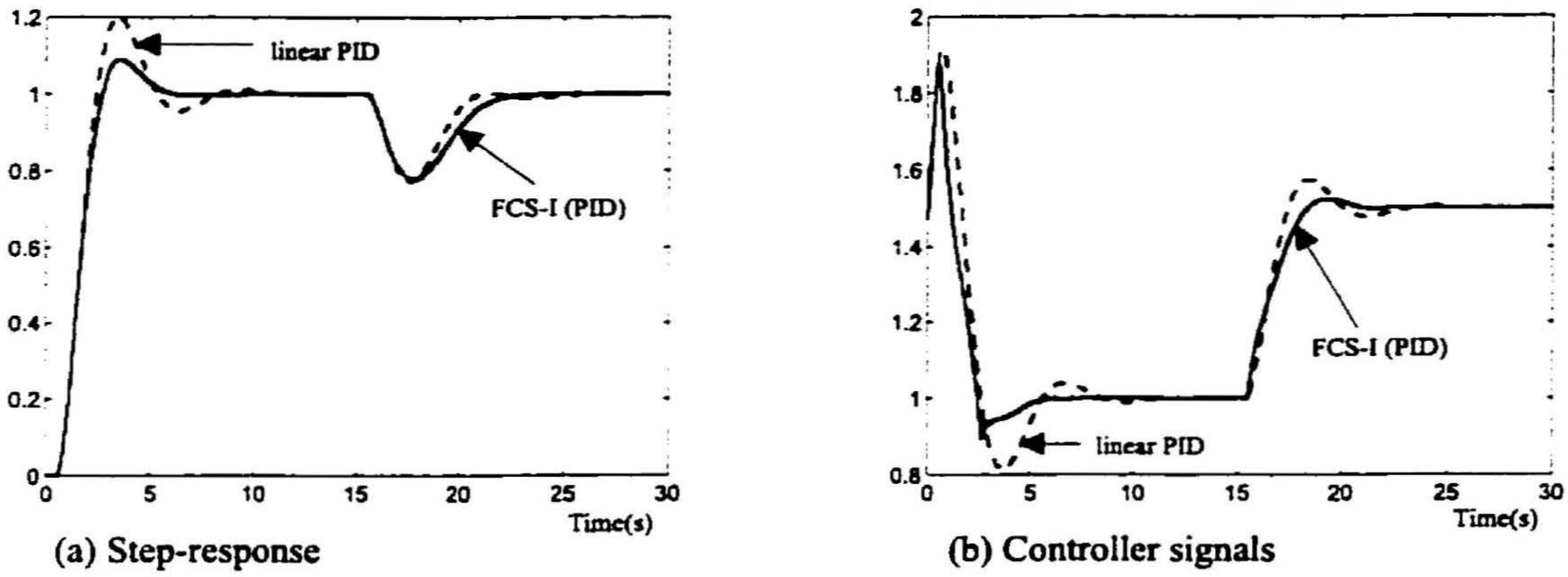


Figure 6.7: Unit step response curves of the process in example E3. The FCS-I parameters are based on two-level tuning

as follows.

1. The local control of proportional and integral actions at zero error are unchanged to preserve the same load disturbance properties. Therefore $(\theta_0)_P$ and $(\theta_0)_I$ are unchanged.
2. Both the angles $(\theta_1)_P$ and $(\theta_1)_I$ are increased to reduce the response speed during the transient.
3. Near zero error, the local control of the derivative action is increased to add more damping to reduce oscillations. Therefore $(\theta_0)_D$ is increased.

The final non-linearity tuning parameters and the corresponding points in the non-linearity tuning diagram obtained after three iterations are shown below.

$$[s_{1,P1}, s_{2,P1}] = [0.45, 0.5], \quad [(\theta_0)_P, (\theta_1)_P] = [\deg 45, \deg 74.5]$$
$$[s_{1,P2}, s_{2,P2}] = [0.45, 0.5], \quad [(\theta_0)_I, (\theta_1)_I] = [\deg 45, \deg 74.5]$$
$$[s_{1,P3}, s_{2,P3}] = [0.2, 0.2], \quad [(\theta_0)_P, (\theta_1)_P] = [\deg 73, \deg 45]$$

Both the response curves that correspond to linear and fuzzy controllers are shown in Figure 6.7. It can be seen from the diagram that the transient response has been considerably improved. The overshoot is reduced to 8%, faster settling time is achieved with least oscillations and almost the same load disturbance properties are maintained.

Tuning FCS-II

The FCS-II uses the three-input rule base structure (Type I). Therefore the system is first set to operate with the LLFLC fuzzy structure for realizing the most linear controller signal. Using the apparent linear gains the normalizing scale factors for the Type I PID controller are: $S_u = 2.3565T_s$. $S_{ce} = 1.8523/T_s$ and $S_{rce} = 0.3246/T_s^2$. The performance of this most linear setting is shown in Figure 6.8. It can be seen from the diagram that the process response is fairly acceptable. However using the non-linearity tuning heuristics the performance is further improved. The first-level tuning is obtained by using the non-linearity tuning diagram that corresponds to the NLFLC-I (Figure 4.3).

1. In order to reduce the oscillatory behavior in the transient, the local control of the derivative action near zero is increased by adding more damping. Therefore $(\theta_0)_D$ is increased.
2. To improve the load disturbance properties, the local control of integral action near zero i.e. $(\theta_0)_I$ is increased. Also for reducing the overshoot, the angle $(\theta_1)_I$ is also increased.
3. The proportional control action is unchanged.

The final non-linearity tuning parameters and the corresponding points in the non-linearity tuning diagram after three iterations are shown below.

$$[(s_1)_1, (s_2)_1] = [0.3, 0.8], \quad [(\theta_0)_I, (\theta_1)_I] = [\deg 59, \deg 68.2]$$

$$[(s_1)_2, (s_2)_2] = [1/3, 2/3], \quad [(\theta_0)_P, (\theta_1)_P] = [\deg 56.3, \deg 56.3]$$

$$[(s_1)_3, (s_2)_3] = [0.15, 0.65], \quad [(\theta_0)_D, (\theta_1)_D] = [\deg 73.3, \deg 55]$$

The corresponding response curves are shown in Figure 6.8. The non-linearity tuning was able to obtain better transient and load disturbance properties than the linearly represented fuzzy PID controller. Also, the controller signal was able to keep well within the saturation limits of the actuator. The non-linearity tuning performed for the above system is relatively small and the fuzzy system was able to produce better results with least effort. However, some difficulties associated with the coupled behavior

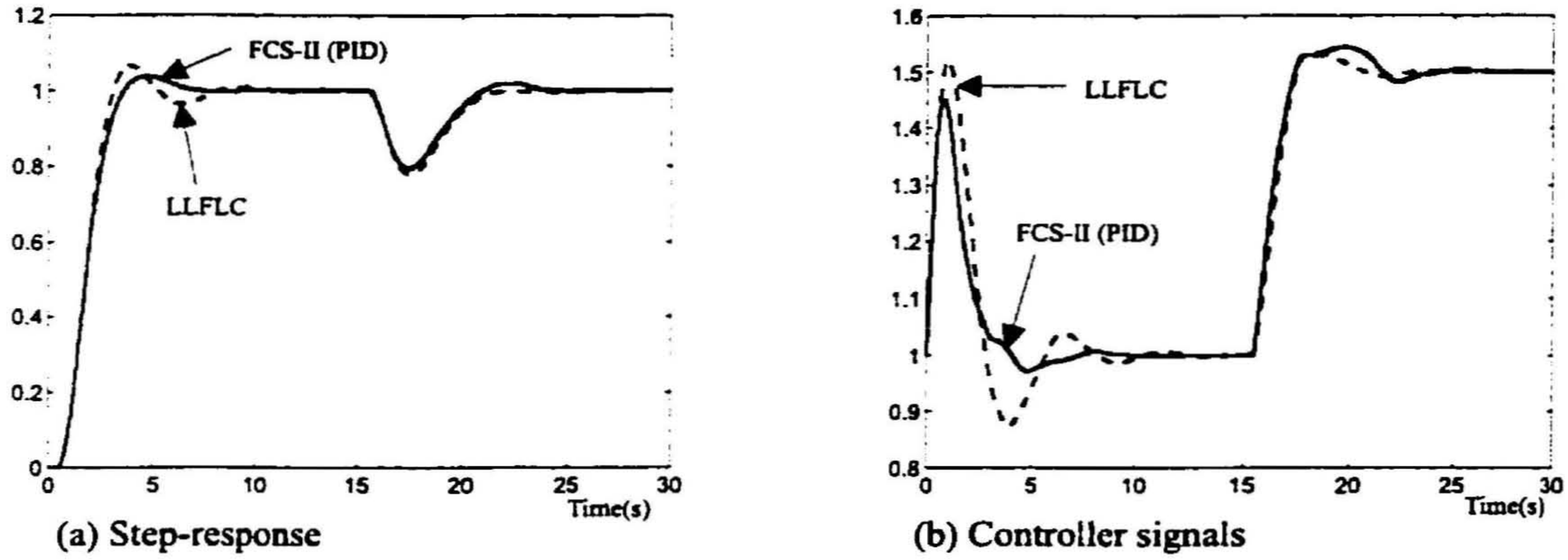


Figure 6.8: Unit step response curves of the process in example E3. The FCS-II parameters are based on two-level tuning

was observed during the simulations. As an example. the changing of local control of proportional action gave bad performance.

Tuning FCS-III

Similar to the previous case. the FCS-III is set to work with the LLFLC structure corresponding to the two input case. Again the apparent linear gains are used to compute the linear gains of the Type III PID structure. Due to the coupled nature of the proportional action in both fuzzy PI and PD terms. the gain terms have two solution sets. The proportional effect in one of the solutions is higher. The second solution has shown a more sluggish kind of response behavior. This is one of the draw backs of the conventional Mamdani-type fuzzy structure where the designer faces a problem of selecting proper ALG terms before implementing the linear controller. However for this illustration the better performance gain terms in the solutions are selected as: $K_{\mathrm{PI}} = 2.3565T_s$. $K_{\mathrm{PID}} = 3.9031$ and $S_{ce} = 0.196/T_s$. Without loss of generality set $S_u = 1$. The response curve corresponding to this linear representation is shown in Figure 6.9. Overall this shows higher oscillatory behavior. Using the non-linearity tuning diagram as in the previous examples. the first-level tuning is performed as follows.

1. To reduce the oscillations, the effect of both proportional and integral actions near zero is reduced. Therefore both $(\theta_0)_P$ and $(\theta_0)_I$ are reduced.
2. In order to reduce the speed of the transient response, both $(\theta_1)_P$ and $(\theta_1)_I$ are increased.
3. To provide more damping, $(\theta_0)_D$ is increased.

As discussed in Chapter 3. the Type III fuzzy PID structure has inferior functional properties for the non-linearity tuning. In addition to the coupled nature of the tuning associated with general coupled fuzzy rule base systems, the gain dependency that exists in the overall system further adds complexity to achieve the independent non-linearity tuning. In this configuration the same input variable ($\hat{e}$) provides both nonlinear proportional and integral actions respectively in the fuzzy PD and PI controller elements. Therefore the desired non-linearity tuning described in items 1 and 2 can be simultaneously obtained by tuning the projected angles of the error variables. However, if the desired tuning for proportional and integral actions are contradictory, this tuning would become a complex problem. Similarly the desired tuning shown in item 3 is obtained by changing the tuning parameters corresponding to the error variable ($\Delta\hat{e}$). This in turn causes the proportional action in the fuzzy PI element to change. The tuning is accomplished in three iterations. The non-linearity tuning parameters and the corresponding angles in the non-linearity diagram are:

$$[(s_1)_1, (s_2)_1] = [0.6, 0.9], \quad [(\theta_0)_P, (\theta_1)_P] = [(\theta_0)_I, (\theta_1)_I] = [\deg 39.8, \deg 78.7]$$

$$[(s_1)_2, (s_2)_2] = [0.25, 0.9], \quad [(\theta_0)_D, (\theta_1)_D] = [(\theta_0)_P, (\theta_1)_P] = [\deg 63.4, \deg 78.7]$$

Both the linear and fuzzy controller based response curves are shown in Figure 6.9. The non-linearity tuning was able to improve the response considerably. The overshoot of the transient response is reduced from 32% to 12.2%. However the action association in the coupled output causes deterioration of the load disturbance properties. From the controller signals it can be observed that in both cases the controller experienced integral wind-up. This is due to the choice of the faster controller from the two solution sets of the linear tuning. This example clearly demonstrates the weaknesses that are inherent in the Mamdani-type fuzzy PID systems.

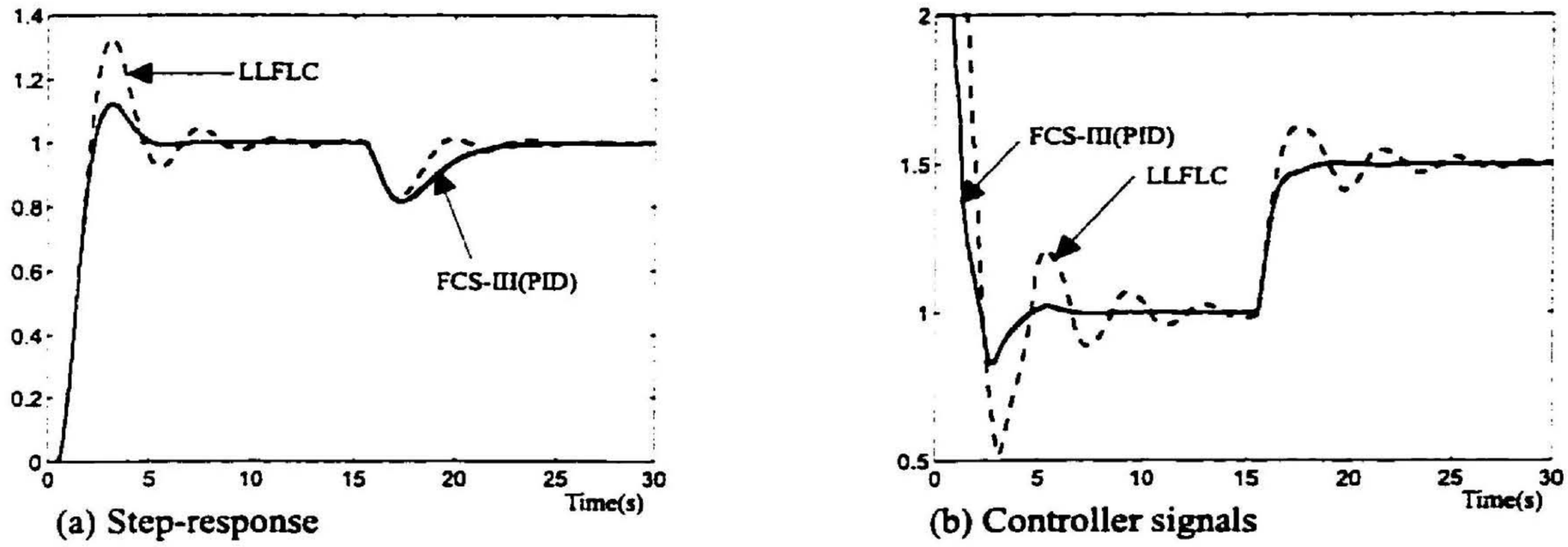


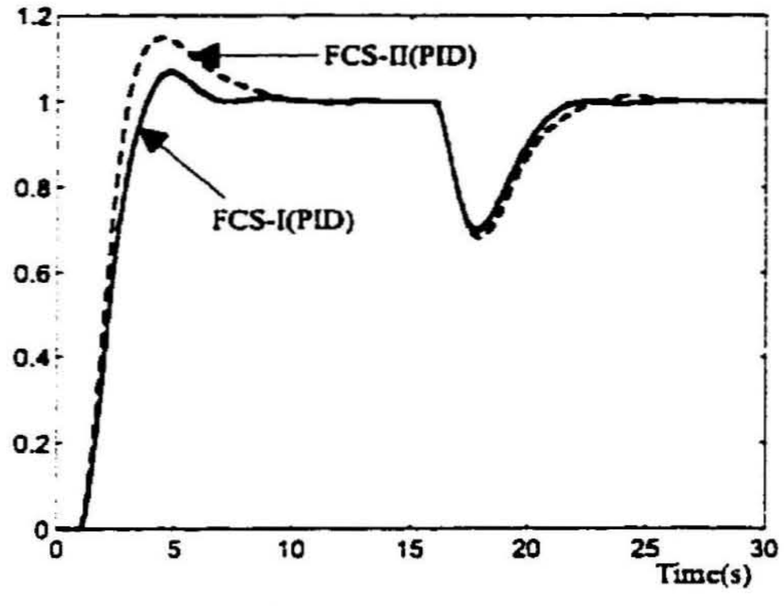
Figure 6.9: Unit step response curves of the process in example E3. The FCS-III parameters are based on two-level tuning

Example E4

This example from chapter 5 illustrates a second order process with approximately unity normalized time delay. For each FCS the linear gains are obtained using the PID solution in Chapter 5. However, the FLS system was unable to get real values from the quadratic expressions for the linear gains. This proves that Mamdani-type fuzzy PID systems have some limitations to use the conventional linear PID design knowledge for solving the ALG terms. Therefore the FCS-I and FCS-II are tuned for this example and the response curves are shown in Figure 6.10. Due to poor functional properties the FCS-II was unable to produce better performance. However the FCS-I has produced better performance during both transient and load disturbances. The action association in the coupled fuzzy controller systems was unable to perform the independent tuning.

6.5 Closed-loop Control of a Heating System

This section describes real time temperature control of a soil-cell. The hardware configuration is shown in Figure 6.11. The control circuit is designed to control temperature at three different locations using three different heaters. The three separate channels for the heaters allow independent supply of heat to the soil cell. For this experiment the program voltage supplied to each heater is kept uniform and therefore the three heaters are set to supply equal



Step-response

Figure 6.10: Unit step response curves of the process in example E4. The FCS-I and FCS-II parameters are based on two-level tuning

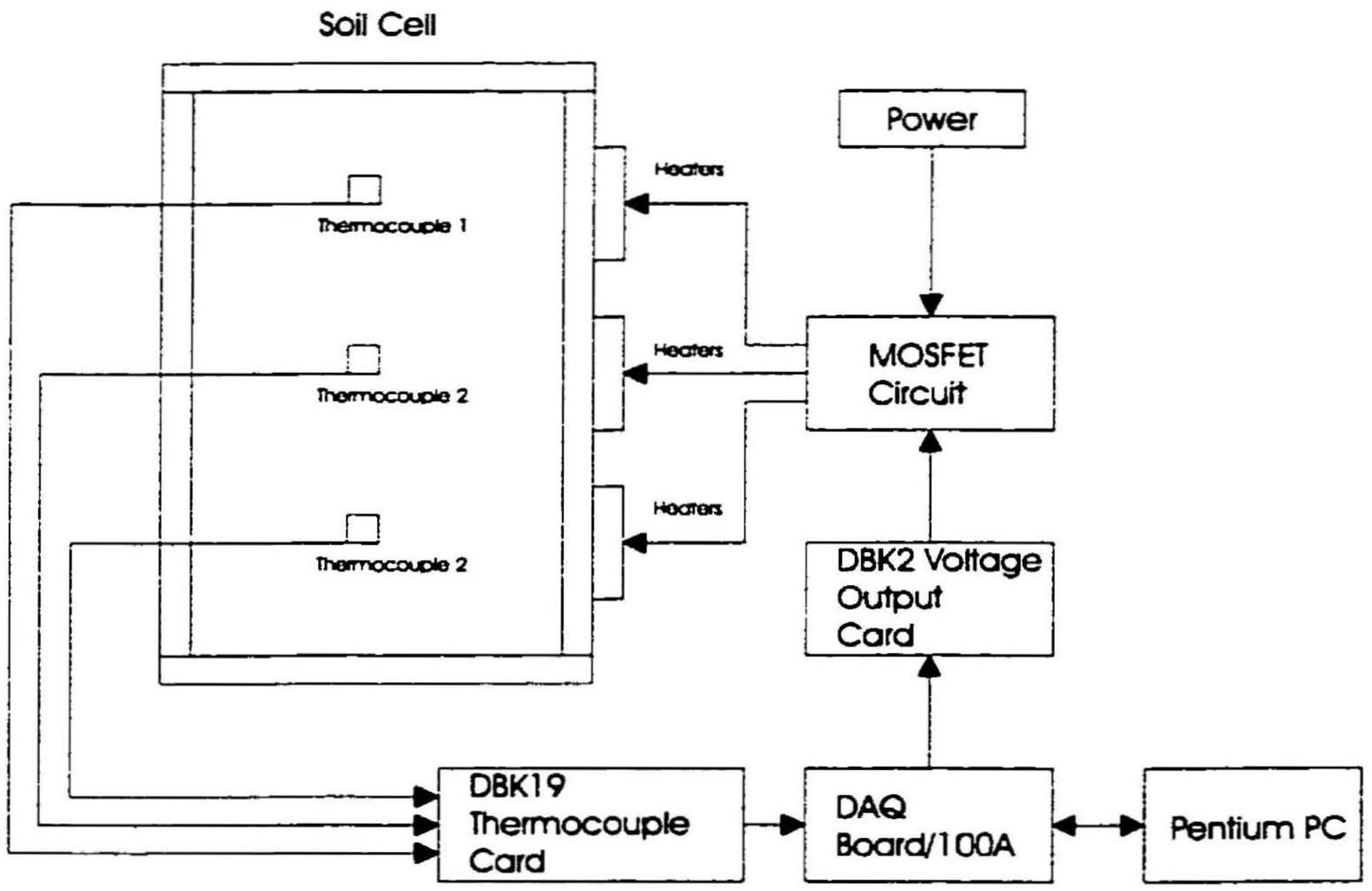


Figure 6.11: Hardware configuration of the feedback temperature control system

quantities of heat simultaneously to the soil cell. The control algorithm is set to control the set point temperature at location 1.

The computer control system is set up using the DAQ/100A (Omega Engineering, Inc.) data acquisitions system. The DBK2 voltage output card is programmed to produce the manipulator signals to the soil-cell assembly. The voltage output card has 14-bit D/A resolution and is set for ± 5V operation. The Thermofoil type heaters from Minco Procuts, Inc. are placed at the circumference of the soil cell. The available resistance of each heater coil is 7.1Ω with 10VDC power supply. At each level, three foil heaters are connected in series and the maximum power availability is therefore three times of a single heater coil. Three different DC power sources are connected to each heater assembly. The voltage from DBK2 card ranges from 0 to 10V. This program voltage is used to change the duty cycle of the MOSFET circuit. The switching frequency is 795Hz. The program voltage from 0.05 to 9.99VDC has the ability to change the percentage ON time from 17.4% to 95.5% of the cycle period. A separate relay is mounted to switch off the heaters completely when the program voltage reads zero. Thus the power at each heater assembly can be varied from 0 to 39.34W. The calibrated heater power versus program voltage is approximated to two linear curves for computing the heater power requirement in the control algorithm and the approximation error is less than 1%.

THe DBK19 thermocouple card is configured directly to read temperatures. The card has 12-bit A/D resolution. The temperature sensors (thremocouples) are from Omega Engineering. Inc. (Type T) and has the temperature range from -200^{o}C to 400^{o}C. At each point three sensors are placed and the average temperature is taken as the soil temperature reading. The time constant at 22.48^{o}C air is 0.177s. The standard deviation of measurements taken in air at the latter average temperature is 0.0943^{o}C.

In the first part of this experiment a number of open-loop tests were carried out to determine the average process parameters related to first-order approximation. The corresponding values are set per heater basis, but in the application all the heaters are simultaneously supplied with an equal amount of power. The three parameters were identified using the open-loop test described in [143] and the computed average parameters are given by;

$$k = 8.6\mathrm{W}/\deg C/\text{heater},\ T = 150\,\min.,\ t_d = 27\,\min.$$

The normalized time delay t_d/T was 0.18: so the process belongs to the category of short time delay problems. As the heater capacities are limited by their saturation values, the practical limits for the PI or PID controllers can be set for this process by using the rules that are described in Chapter 5 (section 5.4.2). In order to implement both PI and PID type controllers the set-point height is chosen as 25°C from the starting temperature of the experiment. A fixed sampling time of 1 min. is used for all experiments.

6.5.1 Linear PI and PID Control

For PI control the proportional weighting (ρ) is chosen as 0.45. Using the linear PI rules (equation (5.29)) the two gain parameters are: $K_P = 0.2907$ and $K_I = 0.0019$.

For PID control the proportional weighting (ρ) is chosen as 0.7. This value is selected in order to satisfy the maximum power limits of the heaters. Using the PID tuning law (equation (5.33)) the PID parameters are; $K_P = 0.4522$, $K_I = 0.0025$ and $K_D = 3.8372$. The set-point response curves and the controller signals are shown in Figure 6.12. In both the cases an overshoot of about 14% is observed. The PID controller has better rise time and settling time than the linear PI controller. However, due to variations in the environment temperature during the control time period, the two controllers show different steady state controller signals.

6.5.2 Fuzzy PI and PID Control with FCS-I

Fuzzy PI

The purpose of fuzzy control now is to provide improved control to the PI controlled temperature. In general the linear PI response has shown quite satisfactory results. However it can be seen from Figure 6.12 that the steady state temperature tracking is less accurate. Due to the slow nature of this temperature control process, the slower variation of the surrounding temperature also affects the performance. As the normalized time delay is relatively low in this case, the load disturbance characteristics of the conventional PI controller are generally poor. Therefore the purpose of the fuzzy PI controller is to provide better load disturbance properties for accurate tracking of the set point and also to preserve the same transient characteristics with lower overshoot of the response.

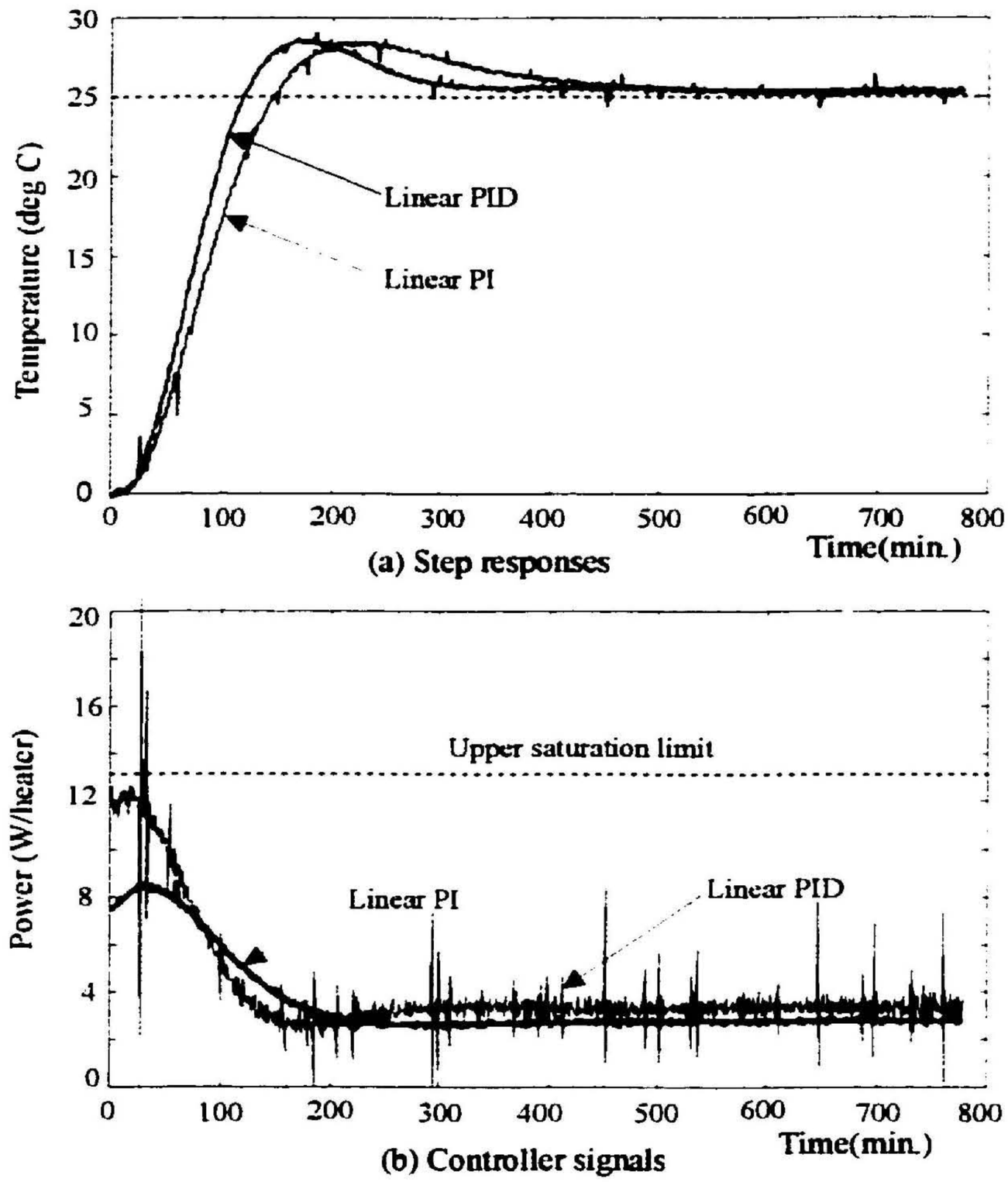


Figure 6.12: Response curves correspond to closed-loop linear PI and PID control of the temperature.

The controller FCS-I provides the fuzzy control. The previous work of this thesis has provided enough evidence to show the better functional properties of this system compared to other conventional fuzzy controller types. The non-linearity tuning is performed using the non-linearity variation diagram given in Figure 4.4. The tuning follows two steps.

1. The local control of integral action at zero error is increased for improving the load disturbance properties. The proportional action near zero is unchanged. Therefore $(\theta_0)_I$ is increased and $(\theta_0)_P$ is unchanged.

2. To keep low overshoot of the transient response the speed of response is reduced by increasing both local proportional and integral actions near the maximum error values. Therefore both $(\theta_1)_P$ and $(\theta_1)_I$ are increased.

Using the non-linearity tuning diagram in Figure 4.4, the first level tuning parameters for the FCS-I are chosen to be,

$$[s_{1,P1}, s_{2,P1}] = [0.45, 0.5]$$

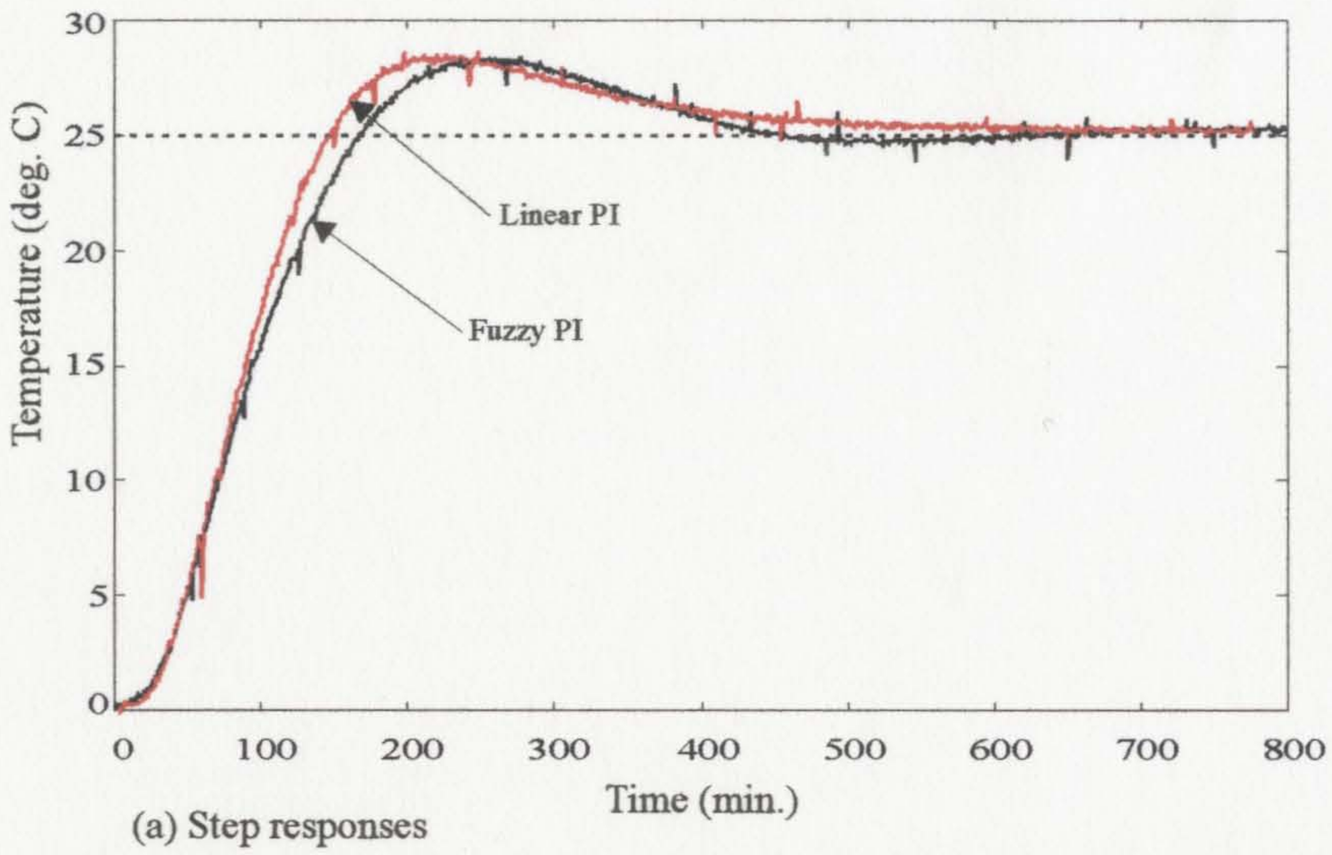
$$[s_{1,P2}, s_{2,P2}] = [0.3, 0.35]$$

The response curves are shown in Figure 6.13. For comparison and clarity the linear PI results are also redrawn.

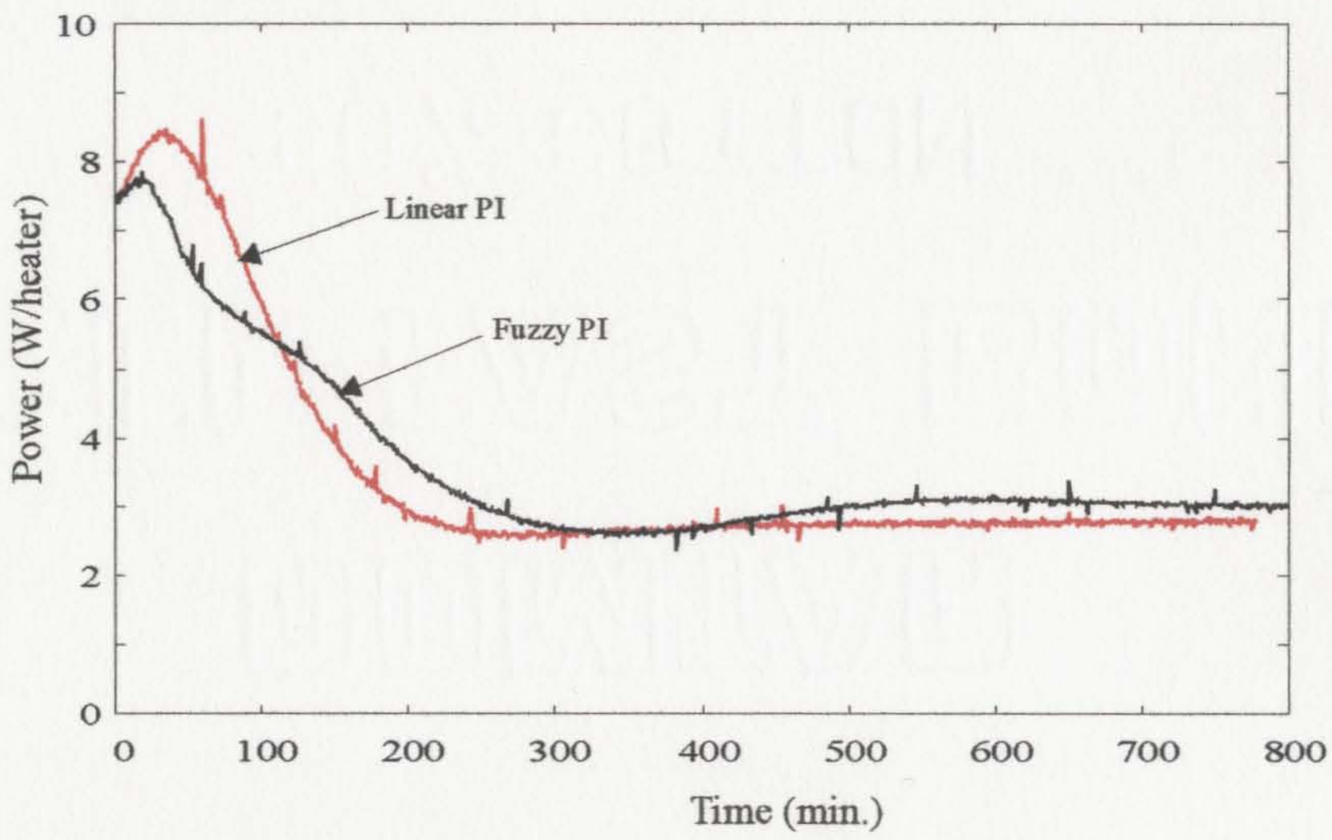
It can be seen from the figure that the fuzzy PI controller was able to produce more accurate tracking of the temperature than the linear PI controller. The controller signal also shows a lower overshoot and milder control than the linear controller. However the overshoot of the transient response remained almost the same compared to the linear case. This is due to limitations that exist in the non-linearity tuning of fuzzy control. As an example, with the increase of local integral control near zero error, the increase of the local control near maximum error is limited due to the limited availability of the admissible area in the quadrant II of the non-linearity diagram.

Fuzzy PID

The linear PID controller behavior is similar to the linear PI response except that it provides faster rise and settling as shown in Figure 6.12. However its set-point tracking is poorer. Therefore the same non-linearity tuning for proportional and integral actions can be performed. In addition the linear PID response shows no oscillations during the transient. This



(a) Step responses



(a) Controller signals

Figure 6.13: Response curves correspond to closed-loop fuzzy PI control of temperature

implies that the linear PID controller has enough damping. Therefore in the non-linearity tuning the fuzzy derivative action is kept at the most linear control point in Figure 4.4. Hence the non-linearity tuning parameters for fuzzy PID control can be summarized as follows.

$$[s_{1,P1}, s_{2,P1}] = [0.45, 0.5]$$

$$[s_{1,P2}, s_{2,P2}] = [0.3, 0.35]$$

$$[s_{1,P3}, s_{2,P3}] = [0.5669, -0.20]$$

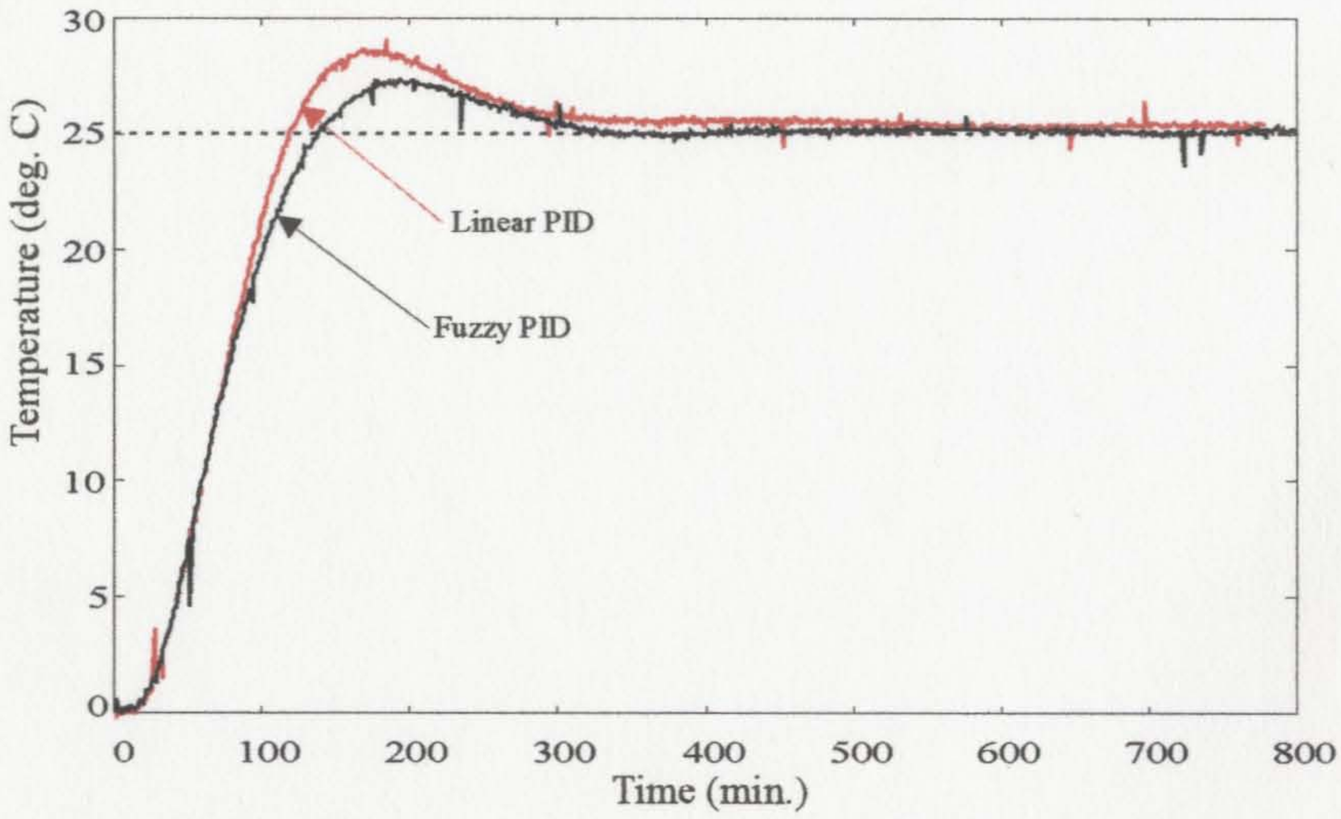
The response curves are shown in Figure 6.14. For comparison and clarity the linear PID results are also redrawn.

It can be seen from the figure that the fuzzy PID controller is able to produce superior performance than its linear counterpart. In addition to the accurate tracking of the temperature, the transient response characteristics have been considerably improved with lower overshoot. In order to verify the repeatability, the linear and fuzzy PID controller experiments are repeated. An additional sensor was also placed outside the soil cell and the environment temperature also recorded during the experiment. The results are shown in Figure 6.15. With fluctuation of the surrounding temperature, the improved load disturbance characteristics of the fuzzy PID controller allowed it to track with a higher precision than the linear controller. The fuzzy controller clearly able to improve the overall performance and also the robustness of the corresponding linear controller.

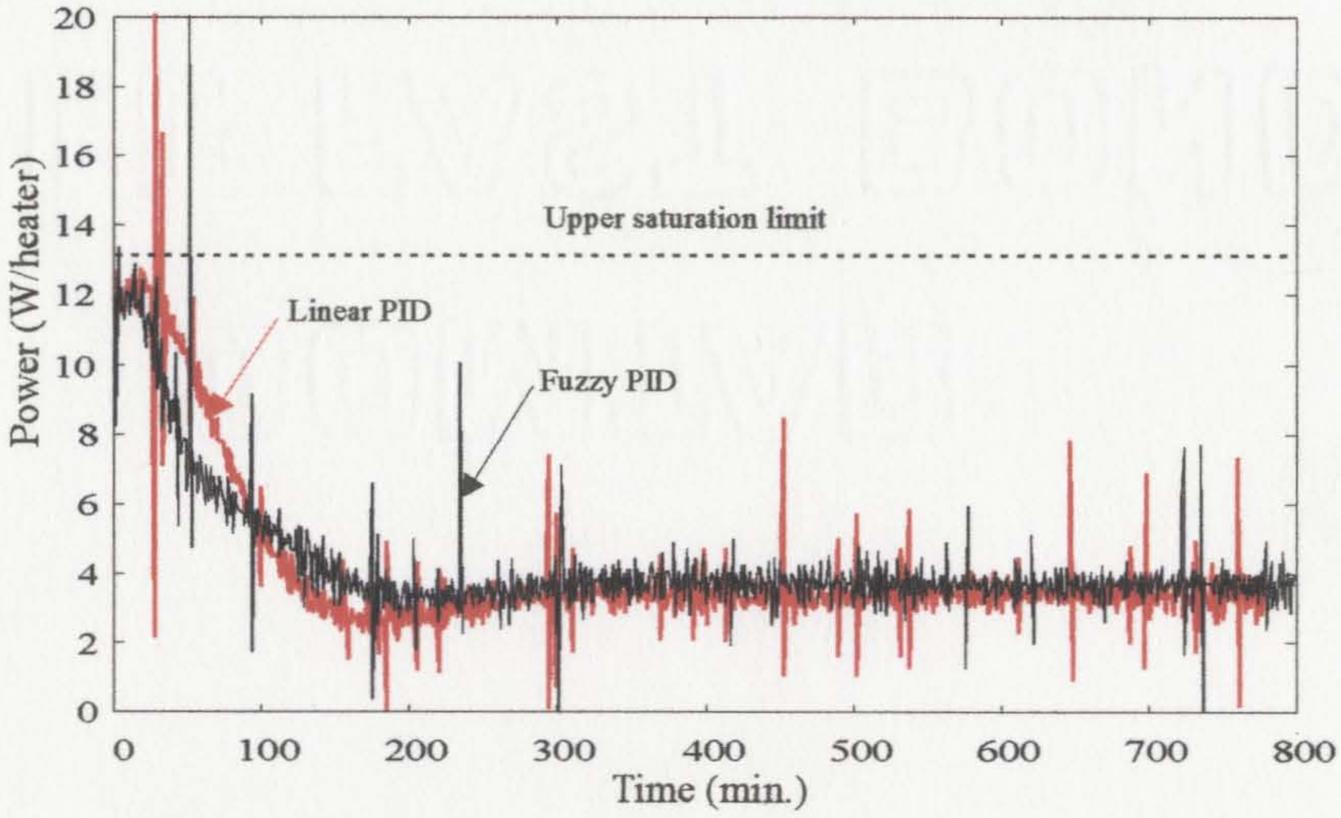
It is important to note that the results reported are from the first test trials. The example well illustrates the validity and application capabilities of the proposed design. Also, in all the cases the controller signal was able to keep within its upper and lower limits while avoiding integral wind-up.

6.6 Adaptive Fuzzy Controller Designs

This section proposes a novel self-organizing fuzzy controller for adaptive control environment. As described in Chapter 1 adaptive controllers have a performance monitor and an adaptive mechanism. In traditional controller designs adaptive control can be either direct or indirect adaptive [32]. As mentioned in Chapter 2, the self-organizing fuzzy controller can also be either direct or indirect.

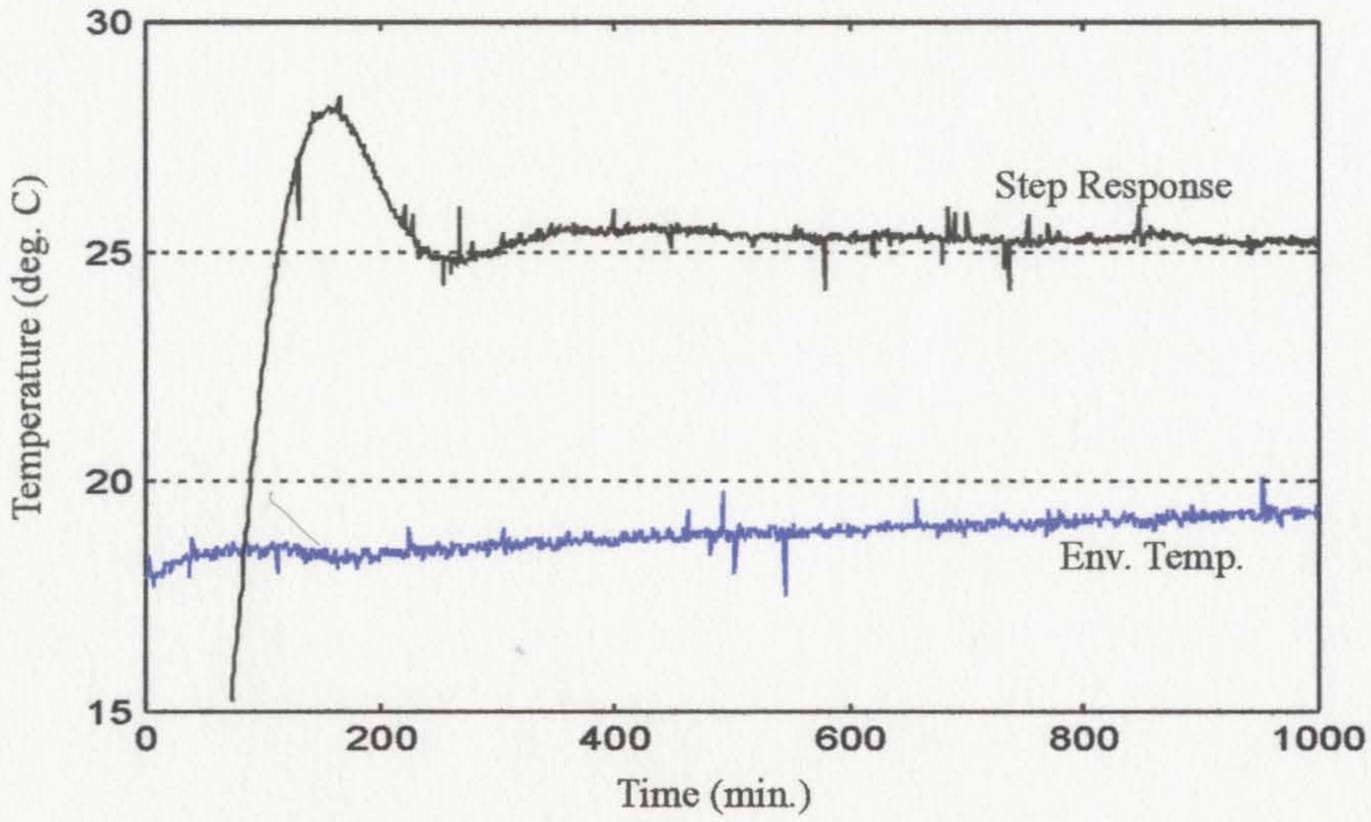


(a) Step responses

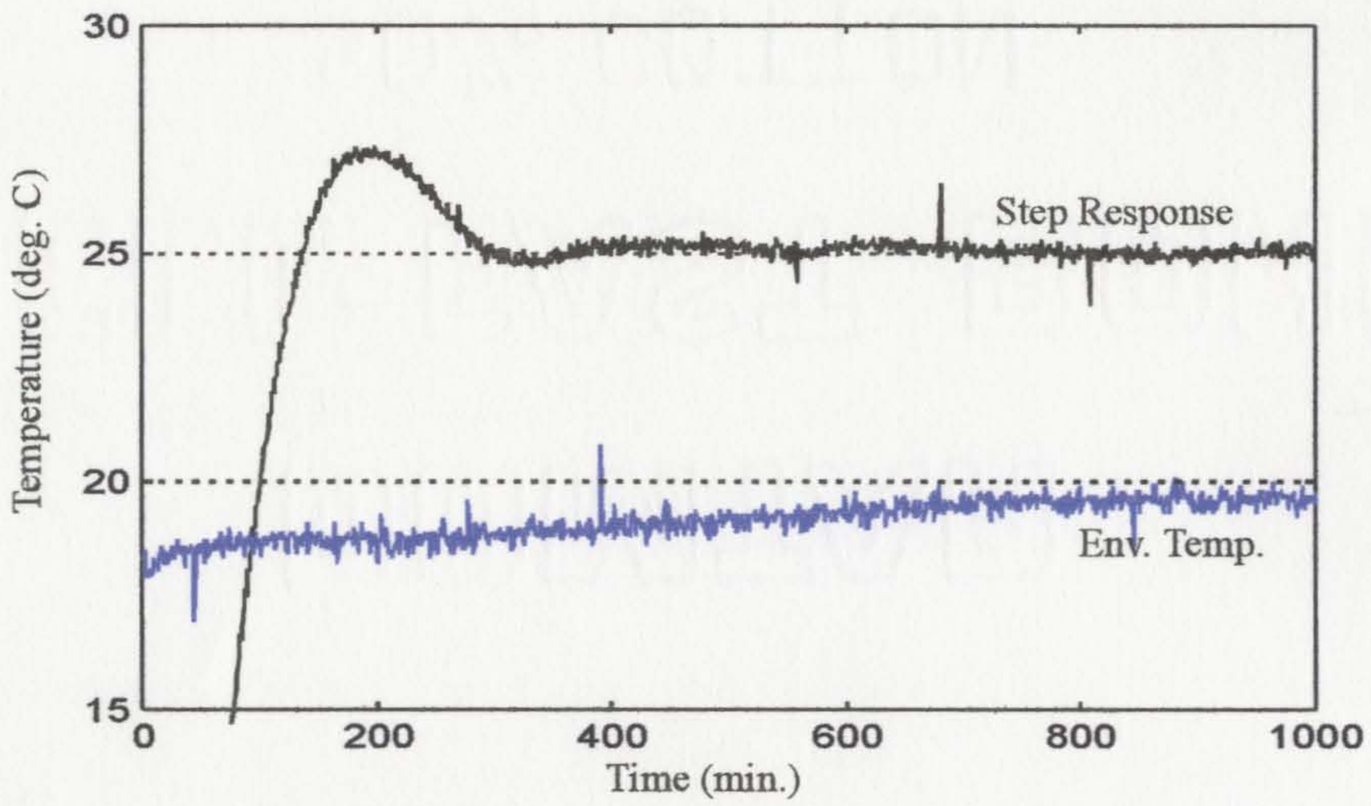


(b) Controller signals

Figure 6.14: Response curves correspond to closed-loop fuzzy PID control of the temperture



(a) Linear PID results



(b) Fuzzy PID results

Figure 6.15: Temperature tracking with PID controllers under varying environment temperature

In the previous sections of this chapter it has been observed that off-line design of fuzzy controllers can be made successful only if the predicted model is sufficiently accurate compared to the real dynamic behavior. However. the two-level design confirms that fuzzy control is valuable and more useful for controlling unknown plant dynamics than known dynamic characteristics. This is the typical black box implementation of the FLC. As an example, if a process can be modeled on-line either by using a fuzzy or neural network, then the system can be made to follow the model as long as the dynamic structure is uniform throughout. When the dynamics of the process changes due to external disturbances or to change in process parameters, the controller or the model should have enough robustness properties to converge the response to the desired state. In gain scheduled type self-organizing fuzzy PID controller [97. 5], the linear PID controller parameters are updated by the fuzzy tuner. The decision making process is at a higher level where the performance characteristics drives the fuzzy controller to update the linear PID gains during the adaptation process. The main problem in those GS type controllers is the irreversibility of the tuning. As an example, once the linear PID parameters are updated due to the presence of a sudden load disturbance, the rule base system is incapable of returning to the original state when the disturbance disappears or is removed. This is due mainly to incorporation of two levels of tuning into single decision making process. The problem of *tuning-in-tuning* that is inherent in adaptive fuzzy controllers can be avoided only if the two levels are isolated from the adaptation process.

The proposed hierarchical self-organizing fuzzy controller architecture is shown in Figure 6.16. The two levels are placed separately for performing linear and nonlinear tuning. When the first level is removed, the adaptive controller is identical to a conventional direct adaptive controller. Also, if the on-line estimated process parameters are accurate the conventional control theories can provide accurate control to the system by the adaptation of the linear gains. In other words, if the inner control system were accurate the upper level would be idle. The performance monitor is placed to observe the response characteristics. The observer can compute the necessary performance attribute before executing the first-level tuning. As an example, these performance attributes are typically observed over several sampling periods such as *overshoot, oscillations, speed of response, steady offset etc.* Therefore the first level tuning occurs at a lower frequency than the second level of tuning.

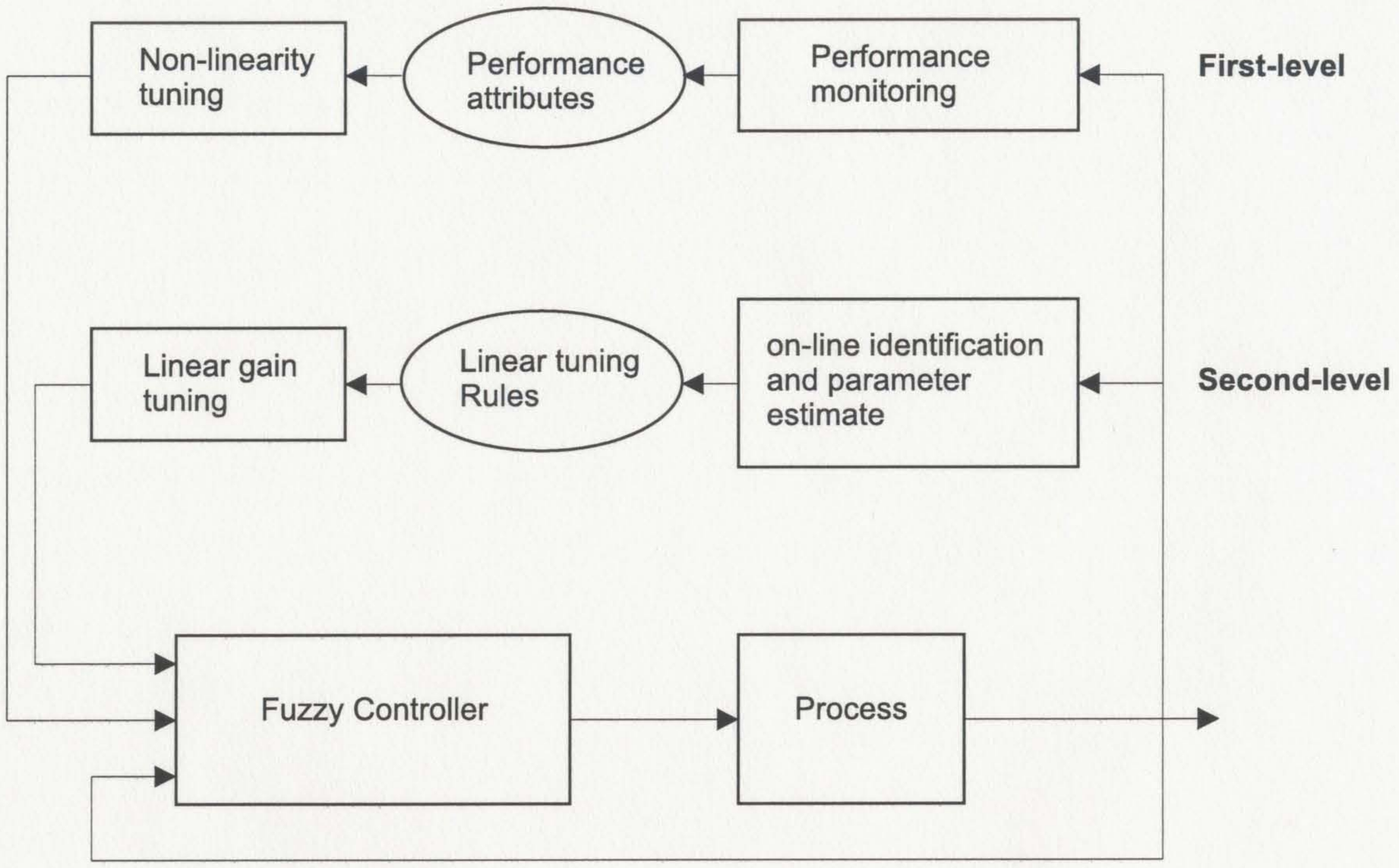


Figure 6.16: Hierarchical self-organizing fuzzy controller

6.7 Summary

This chapter has developed a new two-level tuning scheme for fuzzy controllers. It has also shown that off-line fuzzy controller designs are applicable only if the predicted model of the process accurately reflects the actual dynamic process. Due to the conservative linear control policy, the conventional PID controllers can be satisfactorily designed off-line for real time implementations. Therefore the fuzzy controller properties are embedded in linear control to compensate the unknown or unmodeled plant dynamics. The applicability of the two-level tuning has been proved by the simulation examples and also by a real time control application. The results prove the applicability and validity of the two-level tuning of fuzzy controllers. Further, the incorporation of linear control theories enhances the performance and removes unnecessary complex analysis of fuzzy controller tuning for linear gains.

The fuzzy controller enhances the robustness properties of the closed-loop PID control system. As an example, the local change of PID gains at zero error improves the load disturbance characteristics. This can be clearly seen from the temperature control system. The linear PID controller unable to track the set-point temperature against the changing

surrounding temperature. The first order plus dead time approximation always leads to imprecise estimates of process parameters. This usually results poor transient or steady state performance. However the fuzzy controller design able to simultaneously improve both transient and steady state performance without causing any instability to the overall control system. As an example. in the examples shown in this chapter, the controller signals were able to maintain within their capacity limits.

Although in certain cases the conventional type fuzzy controllers with coupled rules have shown better performance than rule de-coupled fuzzy PID systems. overall those controllers have limitations to perform the non-linearity tuning. The functional deficiencies of coupled systems make them more special kind than general purpose. The proposed one-input fuzzy PID controller with the Type-VI structure has better flexibility and adaptability for wider range of process models than conventional fuzzy controllers.

A novel self-organizing fuzzy controller is proposed for adaptive control. The isolation of two levels from fuzzy controller tuning enabled the non-linearity tuning to compensate the unknown or unmodelled plant dynamics. Hence the fuzzy adaptive controller expected to be more robust than the conventional adaptive control. The disadvantage of the method is that it requires the linear PID controller to be implemented before implementing the fuzzy control. Therefore the method looses some of the advantages in off-line designs.

The proposed two-level tuning method has the following benefits.

1. The fuzzy PID tuning uses the existing linear PID controller theories to obtain apparent linear PID gains. Therefore the method exploits the existing linear control theory and the applicability can be extended for processes where linear PID technology is currently being used.

2. The fuzzy controller can be safely implemented as long as the linear PID controller is stable for the system of interest.

3. The process uncertainties due to modeling error can be compensated by the fuzzy control. Therefore the two levels of tuning can be easily implemented for an adaptive controller.

4. Due to de-coupled identification and adjustment of fuzzy PID actions, simpler tun-

ing heuristics related to the fuzzy control can be established and the method can be extended to all fuzzy PID structure types.

5. The tuning method is simple and more generalized for industrial control applications.

6. The tuning heuristics reduce the search space to a minimum and therefore the design time is considerably reduced.

Chapter 7

Conclusions and Future Research

The thesis attempts to present a systematic design methodology for fuzzy controllers in the context of feedback PID controllers. The study allowed us to identify the functionality of the fuzzy controller actions in exact forms and also to generalize the design for a wider spectrum of control applications. This chapter highlights the new findings and gives recommendations for future research.

7.1 Conclusions

7.1.1 Fuzzy PID Structures

In addition to the commonly available fuzzy PID controller structures, new fuzzy PID controller configurations are identified. The design of a fuzzy controller is treated as a two-level tuning problem. The basic concept of this thesis work is to remove the existing curse of dimensionality in fuzzy control design by a properly defined tuning criteria. Therefore the tuning for each fuzzy controller is then identified by the apparent nonlinear and linear gain terms (ANG and ALG). When the FLC system is completely known, the variations of ANG terms with respect to the error state variables are also known. In optimal designs, this is usually achieved by varying the fuzzy knowledge base parameters; this has a direct influence on the nonlinear characteristics of the control surface or curve. Recently the nonlinear function approximation properties of fuzzy systems have been exploited to train or approximate fuzzy systems for highly nonlinear dynamic systems. However, in most cases the nonlinear function

that is required to control the unknown (or partially known) process dynamics is unknown. The same is true for fuzzy PID control action. The tuning heuristics and rules available from the conventional linear PID control designs are usually available in de-coupled form. Therefore these tuning rules can be used to approximate the unknown nonlinear functions in a single dimension to produce de-coupled and independent tuning for the ANG terms of the fuzzy controllers. In that perspective different fuzzy PID controller structures are assessed in terms of their functional behaviors. The *action association*, *input coupling* and *gain dependency* are the main functional deficiencies that exist in the conventional fuzzy controllers. As a result, the design of those controllers always requires heavy computations and numerical simulations even for simple process systems. The proposed rule de-coupled and gain-independent fuzzy control systems offer better functional properties for efficient design and implementations.

7.1.2 Non-linearity Analysis

The study of non-linearity and its effect on plant performance in a general form is a hard mathematical task. The basic reason is the unknown non-linearity requirement of fuzzy controller output. However, some basic non-linearity tuning properties are identified for de-coupled nonlinear control. Better performance of the fuzzy controller is sought by the local nonlinear control features in the fuzzy outputs. Thus the nonlinearity tuning parameters are identified for fuzzy control actions for tuning the nonlinear PID gains in local control regions. This particular feature of local control has been used for identifying the non-linearity tuning of fuzzy controllers. For rule coupled systems this local control behavior is identified in dissociated form. Therefore a two or three dimensional control action is projected to a single dimension to perform the non-linearity tuning. Based on the non-linearity tuning, a new non-linearity evaluation method is proposed. The analysis is one-dimensional and therefore convenient to evaluate any given fuzzy system with least effort. The new evaluation method has identified and ranked different fuzzy controller systems for better non-linearity designs. More importantly the performance measures have identified the limitations and drawbacks in different fuzzy controller systems and fuzzy reasoning methods. This enables one to identify immediately the primary parameters necessary for non-linearity tuning. Motivated by the non-linearity designs, an alternative nonlinear control system is proposed. The geometrical design features in the Bézier Functions have been exploited to generate parametric based

nonlinear proportional actions. As the control curve is based on geometrical features, the geometrically based non-linearity evaluation method shows 100% performance, in terms of nonlinear control. However, this system is not used or investigated further in order to limit the content of the thesis solely to fuzzy control based systems.

7.1.3 Linear PID Tuning

The study of the linear PID controller has provided insight into the behavior of PID control actions for a better understanding and design of fuzzy controllers. For particular classes of problems, new tuning rules are formulated. The significance of this work is that the tuning rules (a) are applicable for processes whose normalized time delay ranges are from zero to any higher value and (b) have the capability to select the optimum PID controller based on the actuator gain limits. Further, the new tuning rules have shown improved performance compared to other commonly available tuning rules including error integral optimized tuning rules. This linear PID tuning is used as the bench mark for obtaining improved control through fuzzy control.

7.1.4 Two-level Fuzzy PID Tuning

Finally, a simple and novel fuzzy controller tuning criterion is developed. The off-line designs based on the genetic algorithm cannot perform any better than linear PID controllers when they are implemented for true process sytems. This is due to poor modeling of the process. However, the genetic based tuning is quite powerful, particularly for handling a larger number of tuning parameters. It can be concluded that the GA based fuzzy controllers can be effectively used only when the process model is available in an accurate form. The proposed two-level tuning principle allows the use of the available linear PID tuning heuristics. According to the proposed method, first the linear PID controller is designed for its best performance. If the linear system is satisfactory then there is no need to implement a fuzzy controller. The fuzzy controller is therefore tuned to compensate for modelling errors and unknown plant dynamics. In other words, fuzzy control is effective for unknown process dynamics rather than for known process models.

Overall the conventional two-input fuzzy PID controller has shown poor performance

compared to other fuzzy controller systems. One reason is the coupled nature in both of the tuning levels. Secondly, the coupled rules do not allow independent tuning for individual PID actions and also the tuning variables have lower degree of freedom within the tuning space.

7.1.5 Real-Time Implementation

The validity of the linear and fuzzy controller designs was tested in a real-time temperature control problem. All the experiments yielded better results and those results have been obtained in the first trial itself. The proportional weighting for both PI and PID controllers was chosen taking into consideration of heater saturation. The non-linearity tuning has shown improved temperature tracking when using both fuzzy PI and fuzzy PID controllers. The applicability and robustness of the proposed fuzzy controller tuning are clearly demonstrated in the experiment.

7.2 Summary of New Findings

1. New fuzzy PID controller configurations have been developed. The simplest one-input type PID configuration has shown sufficient merits to use for wider range of process control applications.

2. New algorithms have been developed to generate solutions to different fuzzy control systems. The algorithm simplifies min-max-gravity reasoning based fuzzy outputs to represent using least number of nonlinear expressions.

3. Although some previous researchers have proposed a two-level tuning for fuzzy controllers they were unable to express the two level tuning parameters in an explicit form. New definitions were developed to describe fuzzy PID gains. The apparent linear and nonlinear gains (ALG and ANG) were explicitly defined for two-level tuning.

4. A new graphical method was developed for representing the nonlinearity in fuzzy control. The effects of ANG terms were identified by slopes drawn at local control points in the fuzzy output (or projected output). The tuning of the ANG terms were thus quantitatively obtained.

5. A novel performance criteria was developed for assessing the performance and validity of different fuzzy controller systems.

6. An alternative non-linear control. using spline-based functions was developed.

7. Using time-domain analysis. a novel linear PID tuning scheme was developed. Overall the new tuning scheme has shown better performance compared to other commonly available tuning methods.

8. A new two-level fuzzy controller tuning scheme was developed. The method includes both linear and nonlinear tuning of fuzzy controllers. The tuning method is more general and can be used for tuning of any type of a fuzzy PID controller.

9. A new self-organizing fuzzy PID controller was proposed.

7.3 Future Research

The benefits and conclusions derived from this systematic study indicate that further research would be beneficial and rewarding. The two-level tuning method has shown enough evidence to suggest that a systematic study could be immensely helpful in designing high performance fuzzy controllers for a wider range of process control problems. The identification of fuzzy controller action in a more exact form and the decomposition of its terms for obtaining desired control performance are useful for further understanding the functionality of fuzzy controllers.

7.3.1 Adaptive Fuzzy Controllers

The two-level tuning principle can be effectively used for adaptive fuzzy controllers (Chapter 6). Further analysis is required to quantify the non-linearity tuning against the process response patterns. However, this quantification requires normalization of non-linearity tuning for a wider range of process dynamics problems. However, a better understanding of conventional adaptive control techniques would definitely be an asset for establishing a fuzzy adaptive controller.

7.3.2 Stability

Although there have been many studies related to fuzzy controller stability, the relative and absolute stability aspects of fuzzy controllers are still under investigation. The fuzzy sliding-mode design approach provides Lyapunov stability of the system. However, there is no systematic study yet available for obtaining hard practical limits for the fuzzy controller parameters. The separation of linear and nonlinear aspects of the fuzzy controllers allows one to use conventional linear or nonlinear stability theories to design fuzzy controllers with better stability properties.

7.3.3 Hybrid-type Direct Fuzzy Control

The work of this thesis mainly focused on "one degree-of-freedom" PID controllers. However it is possible to extend the application of two level design methodology for other hybrid controller systems [24]. As an example a coupled PID controller can be used in conjunction with one-input type PID controllers. The coupled system can provide more robustness in terms of external disturbances while the de-coupled system may produce better transient characteristics. Also the other higher order linear PID systems [151] can be represented by fuzzy controllers by suitably selecting the correct fuzzy structural elements as proposed in Chapter 3. The addition of local control to multi degree-of-freedom PID systems can be performed either in coupled or de-coupled form.

7.3.4 Indirect Fuzzy Control

The other aspect in this area of research is indirect fuzzy controller designs. In the indirect control the process is identified by a fuzzy or neural network model. If such a process model is available (either on-line or off-line), the fuzzy controller can be effectively optimized to produce superior performance against the conventional methods. Also it has been observed in the GA based designs that the fuzzy controller can produce excellent performance when the process model is accurate compared to the real process system. Therefore incorporation of other intelligent schemes has better scope for controlling applications, particularly for adaptive controllers.

Bibliography

[1] B. Kosko, *Fuzzy Engineering.* Prentice Hall, New Jersey, 1997.

[2] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[3] L. A. Zadeh, "Fuzzy logic = Computing with words," *IEEE Trans. on Fuzzy Systems*, vol. 4, pp. 103–111, 1996.

[4] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 3, pp. 28–44, 1973.

[5] C. W. de Silva, *Intelligent control: Fuzzy logic applications.* CRC Press, New York, 1995.

[6] L.-X. Wang and J. M. Mendel, "Fuzzy basic functions, Universal approximation, and orthogonal least-squares learning," *IEEE Trans. on Neural Networks*, vol. 3, pp. 807–814, 1992.

[7] J. J. Buckley, "Universal fuzzy controllers," *Automatica*, vol. 28, pp. 1245–1248, 1992.

[8] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Proceedings of the IEE in Control and Science*, vol. 121, pp. 1585–1588, 1974.

[9] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man-Machine Studies*, vol. 7, pp. 1–13, 1975.

[10] M. Sugeno, *Industrial Applications of Fuzzy Control.* Noth-Holland, 1985.

[11] H. Hirota, *Industrial Applications of Fuzzy Technology.* Translated by H. Solomon. Springer-Verlag, Tokyo, 1993.

[12] M. Sugeno, T. Murofushi, T. Mori, and T. Tatematsu, "Fuzzy algorithmic control of a model car by oral instructions ," *Fuzzy Sets and Systems*, vol. 32, pp. 207–219, 1989.

[13] H. Ono, T. Ohnishi, and Y. Terada, "Combustion control of refuse incineration plant by fuzzy logic," *Fuzzy Sets and Systems*, vol. 32, pp. 193–206, 1989.

[14] G. Chen, "Conventional and fuzzy PID controllers: An overview," *Int. J. of Intelligent Control and Systems*, vol. 1, pp. 235–246, 1996.

[15] D. Driankov, H. Hellendoorn, and M. Reinfrank, *An Introduction to Fuzzy Control.* Springer-Verlag, New York, 1996.

[16] M. Braae and D. A. Rutherford, "Selection of parameters for a fuzzy logic controller ," *Fuzzy Sets and Systems*, vol. 2, pp. 185–199, 1979.

[17] M. Braae and D. A. Rutherford, "Theoretical and linguistic aspects of the fuzzy logic controller," *Automatica*, vol. 15, pp. 553–577, 1979.

[18] W. Siler and H. Ying, "Fuzzy control theory: The linear case," *Fuzzy Sets and Systems*, vol. 33, pp. 275–290, 1989.

[19] H. Ying, W. Siler, and J. J. Buckley, "Fuzzy control theory: A nonlinear case," *Automatica*, vol. 26, pp. 513–520, 1990.

[20] H. Ying, "The simplest fuzzy controllers using different inference methods are different nonlinear proportional-integral controllers with variable gains," *Automatica*, vol. 29, pp. 1579–1589, 1993.

[21] H. Ying, "A nonlinear fuzzy controller with linear control rules is the sum of a global two-dimensional multilevel relay and a local nonlinear proportional-integral controller," *Automatica*, vol. 29, pp. 499–505, 1993.

[22] H. Ying, "Analytical structure of a fuzzy controller with linear control rules," *Information Sciences*, vol. 81, pp. 213–227, 1994.

[23] H. Ying, "Sufficient conditions on general fuzzy systems as function approximators," *Automatica*, vol. 30, pp. 521–525, 1994.

[24] H. A. Malki, H. Li, and G. Chen, "New design and stability analysis of fuzzy proportional-derivative control systems," *IEEE Trans. on Fuzzy Systems*, vol. 2, pp. 245–254, 1994.

[25] H. A. Malki, D. Misir, D. Feigenspan, and G. Chen, "Fuzzy PID control of a flexible-joint robot arm with uncertainties from time-varying loads," *IEEE Trans. on Control System Technology*, vol. 5, pp. 371–378, 1997.

[26] M. Mizumoto, "Realization of PID controls by fuzzy control methods," *Fuzzy Sets and Systems*, vol. 70, pp. 171–182, 1995.

[27] A. E. Hajjaji and A. Rachid, "Analytic formulation of linguistic rules for fuzzy controller," *Fuzzy Sets and Systems*, vol. 73, pp. 219–225, 1995.

[28] C.-L. Chen, P.-C. Chen, and C. K. Chen, "Analysis and design of fuzzy control systems," *Fuzzy Sets and Systems*, vol. 57, pp. 125–140, 1993.

[29] C.-L. Chen and F.-C. Kuo, "Design and analysis of a fuzzy logic controller," *Int. J. Systems Sci.*, vol. 29, pp. 1579–1589, 1995.

[30] C.-L. Chen and C.-T. Hsieh, "User-friendly design method for fuzzy logic controllers," *IEE Proc.-Control Theory Appl.*, vol. 143, pp. 358–366, 1996.

[31] C.-L. Chen, S.-N. Wang, C.-T. Hsieh, and F.-Y. Chang, "Theoretical analysis of crisp-type fuzzy logic controllers using various *t*-norm sum-gravity methods," *IEEE Trans. on Fuzzy Systems*, vol. 6, pp. 122–136, 1998.

[32] K. J. Åström and B. Wittenmark, *Adaptive Control* . Addison-Wesley, USA, 2 ed., 1995.

[33] B.-G. Hu, G. K. I. Mann, and R. G. Gosine, "Control curve design for nonlinear (or fuzzy) proportional control action using spline-based functions," *Automatica*, vol. 34, pp. 1125-1133, 1998.

[34] F. L. Lewis, C. T. Abdellah, and D. M. Dawson, *Control of Robot Manipulators*. Macmillan Publishing Company, 1993.

[35] C. J. Harris, C. G. Moore, and M. Brown, *Intelligent Control: Aspects of Fuzzy Logic and Neural Nets*. World Scientific, London, 1993.

[36] A. L. Schwartz, "Comments on "Fuzzy logic for control of roll and moment for a flexible wing aircraft"," *IEEE Control Systems Magazine*, pp. 61–62, Feb. 1992.

[37] J. Yen and L. Wang, "Simplification of fuzzy rule based systems using orthogonal transformation," in *Proceedings of the Sixth IEEE Intn'l Conf. on Fuzzy Systems*, vol. 1, pp. 253–258, Barcelona, Spain, July 1-5 1997.

[38] L. T. Kòczy, "Fuzzy If then rule models and their transformation into one another," *IEEE Trans. on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 26, pp. 621–637, 1996.

[39] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 15, pp. 116–132, 1985.

[40] J. Kim, Y. Moon, and B. P. Zeigler, "Designing fuzzy net controllers using genetic algorithms," *IEEE Control Systems Magazine*, pp. 66–72, June 1995.

[41] S. Smith and D. J. Comer, "Automated calibration of a fuzzy Logic controller using a cell state space algorithm," *IEEE Control Systems Magazine*, pp. 18–28, August 1991.

[42] M. Brown and C. J. Harris, *Neurofuzzy Adaptive Modelling And Control* . Prentice Hall, London, 1994.

[43] R. Palm, "Robust control by fuzzy sliding mode," *Automatica*, vol. 30, pp. 1429–1437, 1994.

[44] X.-J. Zeng and M. G. Singh, "Approximation theory of fuzzy systems-SISO case," *IEEE Trans. on Fuzzy Systems*, vol. 2, pp. 162–176, 1994.

[45] K. M. Passino, "Bridging the gap between conventional and intelligent control," *IEEE Control Systems Magazine*, pp. 12–18, June 1993.

[46] Z.-Y. Zhao, M. Tomizuka, and S. Isaka, "Fuzzy gain scheduling of PID controllers," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 23, pp. 1392–1398, 1993.

[47] T. J. Procyk and E. H. Mamdani, "A linguistic self-organizing process controller," *Automatica*, vol. 15, pp. 15–30, 1979.

[48] H.-X. Li and H. B. Gatland, "Conventional fuzzy control and its enhancement," *IEEE Trans. on Systems, Man, and Cybernetics-Part B:Cybernetics*, vol. 26, pp. 791–797, 1996.

[49] J. Lee, "On methods for improving performances of PI-type fuzzy logic controllers," *IEEE Trans. on Fuzzy Systems*, vol. 1, pp. 298–301, 1993.

[50] J. M. Mendel, "Fuzzy logic systems for engineering: A tutorial ," *Proceedings of the IEEE*, vol. 83, pp. 245–377, 1293, 1995.

[51] T. Jiang and Y. Li, "Generalized defuzzification strategies and their parameter learning procedures," *IEEE Trans. on Fuzzy Systems*, vol. 4, pp. 64–71, 1996.

[52] M. Mizumoto, "Fuzzy controllers under various fuzzy reasoning methods," *Information Sci.*, vol. 45, pp. 129–151, 1988.

[53] P. J. King and E. H. Mamdani, "The application of fuzzy control systems to industrial processes," *Automatica*, vol. 13, pp. 235–242, 1977.

[54] W. J. M. Kickert and H. R. V. N. Lemke, "Application of a fuzzy controller in a warm water plant," *Automatica*, vol. 12, pp. 301–308, 1976.

[55] C. C. Lee, "Fuzzy logic control systems: Fuzzy logic controller Part I and II," *IEEE Trans. on System, Man, and Cybernetics*, vol. 20, pp. 404–435, 1990.

[56] H.-X. Li and H. B. Gatland, "A new methodology for designing a fuzzy logic controller"," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 25, pp. 505–512, 1995.

[57] F. Bouslama and A. Ichikawa, "Fuzzy control rules and their natural control rules," *Fuzzy Sets and Systems*, vol. 48, pp. 65–86, 1992.

[58] W. Z. Qiao and M. Mizumoto, "PID type fuzzy controller and parameter adaptive method," *Fuzzy Sets and Systems*, vol. 78, pp. 23–35, 1996.

[59] M. Sugeno and G. T. Kang, "Fuzzy modeling and control of multilayer incinerator," *Fuzzy Sets and Systems*, vol. 18, pp. 329–346, 1986.

[60] M. Sugeno and G. T. Kang, "Structure identification of fuzzy model," *Fuzzy Sets and Systems*, vol. 28, pp. 15–23, 1988.

[61] D. P. Fileve and R. R. Yager, "On the analysis of fuzzy logic controllers," *Fuzzy Sets and Systems*, vol. 68, pp. 39–66, 1994.

[62] P. P. Wang and C.-Y. Tyan, "Fuzzy dynamic system and fuzzy linguistic controller classification," *Automatica*, vol. 30, pp. 1769–1774, 1994.

[63] X.-T. Peng, S.-M. Liu, T. Yamakawa, P. Wang, and X. Liu, "Self-regulating PID controllers and their applications to a temperature controlling Process," in *Fuzzy Computing; Theory, Hardware, and Applications* (M. M. Gupta and T. Yamakawa, ed.), pp. 355–364, Elsevier Science, North Holland, 1988.

[64] E. Czogala and T. Rawlik, "Modelling of a fuzzy controller with application to the control of a biological processes," *Fuzzy Sets and Systems*, vol. 31, pp. 13–22, 1989.

[65] Y. F. Li and C. C. Lau, "Development of fuzzy algorithms for servo Systems," *IEEE Control System Magazine*, pp. 65–71, August 1989.

[66] S. Murakami, F. Takemoto, H. F. Fujimura, and E. Ide, "Weld-line tracking control of arc welding robot using fuzzy logic controller," *Fuzzy Sets and Systems*, vol. 32, pp. 221–237, 1989.

[67] F. Matía, A. Jiménez, R. Galàn, and R. Sanz, "Fuzzy controllers: lifting the linear-nonlinear frontier," *Fuzzy Sets and Systems*, vol. 52, pp. 113–128, 1992.

[68] G.-C. Hwang and S.-C. Lin, "A stability approach to fuzzy control design for nonlinear systems," *Fuzzy Sets and Systems*, vol. 48, pp. 279–287, 1992.

[69] J. Ambuel, L. Steenhoek, R. Smith, and T. Colvin, "Control of hydrostatic transmission output speed: Development and Comparison of PI and Hybrid Fuzzy-PI Controllers," *American Society of Agricultural Engineers*, vol. 36, pp. 1057–1064, 1993.

[70] S. Sugawara and T. Suzuki, "Application of fuzzy control to air conditioning environment," *J. Therm. Biol.*, vol. 18, pp. 465–472, 1993.

[71] S.-Z. He, S. Tan, F.-L. Xu, and P.-Z. Wang, "Fuzzy self-tuning of PID controllers," *Fuzzy Sets and Systems*, vol. 56, pp. 37–46, 1993.

[72] S. J. Qin and G. Borders, "A Multiregion Fuzzy Logic Controller for Nonlinear Process Control," *IEEE Trans. on Fuzzy Systems*, vol. 2, pp. 74–81, 1994.

[73] S. Kawaji and N. Matsunaga, "Fuzzy control of VSS type and its robustness," in *Fuzzy Control and Systems* (A Kandes and G. Langholz, ed.), pp. 225–242, CRC Press, USA, 1994.

[74] S. K. Nam and W. S. Yoo, "Fuzzy PID control with accelerated reasoning for d.c. servo motors," *Engineering Application of Artificial Intelligence*, vol. 7, pp. 559–569, 1994.

[75] K. B. Ramkumar and M. Chidambaram, "Fuzzy self-tuning PI controller for bioreactors," *Bioprocess Engineering*, vol. 12, pp. 263–267, 1995.

[76] J. S. Baras and N. S. Patel, "Derivation of fuzzy rules for model-free tuning of PID controllers," *J. of Intelligent and Fuzzy Systems*, vol. 3, pp. 59–69, 1995.

[77] M. Strefezza and Y. Dote, "Neuro-fuzzy proportional-integral-differential controller for alternate current servomotor," *Integrated Computer-Aided Engineering*, vol. 2, pp. 87–95, 1995.

[78] J. Kim and B. P. Zeigler, "Designing fuzzy logic controllers using a multiresolutional search paradigm," *IEEE Trans. on Fuzzy Systems*, vol. 4, pp. 213–226, 1996.

[79] J. C. Wu and T. S. Liu, "A sliding-mode approach to fuzzy control design," *IEEE Trans. on Control Systems Technology*, vol. 4, pp. 141–150, 1996.

[80] D. Misir, H. A. Malki, and G. Chen, "Design and analysis of a fuzzy proportional-integral-derivative controller," *Fuzzy Sets and Systems*, vol. 79, pp. 297–314, 1996.

[81] M. Berenguel, E. F. Camacho, F. R. Rubio, and P. C. K. Luk, "Incremental fuzzy PI control of a solar power plant," *IEE Proc. Control Theory Appl.*, vol. 144, pp. 596–604, 1997.

[82] M. Dussud, S. Galichet, and L. P. Foulloy, "Application of fuzzy logic control for continuous casting mold level control," *IEEE Trans. on Control Systems Technology*, vol. 6, pp. 246–256, 1998.

[83] S. Yasunobu and S. Miyamoto, "Automatic train operation by predictive fuzzy control," in *Industrial Applications of Fuzzy Control* (M. Sugeno, ed.), pp. 1–18, Elsevire Science, North Holland, 1985.

[84] L. I. Larkin, "A fuzzy logic controller for aircraft flight control," in *Industrial Applications of Fuzzy Control* (M. Sugeno, ed.), pp. 87–103, Elsevire Science, North Holland, 1985.

[85] K. Sugiyama, "Rule-based self-organizing controller," in *Fuzzy Computing: Theory, Hardware, and Applications* (M. M. Gupta and T. Yamakawa, ed.), pp. 341–353, Elsevier Science, North Holland, 1988.

[86] R. Tanscheit and E. M. Scharf, "Experiments with the use of a rule-based self-organising controller for robotics applications," *Fuzzy Sets and Systems*, vol. 26, pp. 195–214, 1988.

[87] S. Shao, "Fuzzy self-organizing controller and its application for dynamic processes," *Fuzzy Sets and Systems*, vol. 26, pp. 151–164, 1988.

[88] D. A. Linkens and S. B. Hasnain, "Self-organising fuzzy logic control and application to muscle relaxant anesthesia," *IEE Proceedings-D Control Theory Appl.*, vol. 138, pp. 274–284, 1991.

[89] J. R. Layne, K. M. Passino, and S. Yurkovich, "Fuzzy learning control for antiskid braking systems," *IEEE Trans. on Control Systems Technology*, vol. 1, pp. 122–129, 1993.

[90] Y. P. Singh, "A modified self-organizing controller for real-time process control applications," *Fuzzy Sets and Systems*, vol. 96, pp. 147–159, 1998.

[91] G. Bartolini, G. Casalino, F. Davoli, M. Mastretta, R. Minciardi, and E. Morten, "Development of performance adaptive fuzzy controllers with application to continuous casting plants," in *Cybernetics and System Research* (R. Trappl, ed.), pp. 721–728, Amsterdam, North Holland, 1982.

[92] C. Batur and V. Kasparian, "Adaptive expert control," *Int. J. Control*, vol. 54, pp. 867–881, 1991.

[93] C. J. Li and J. C. Tzou, "A new learning fuzzy controller based on the P-integrator concept," *Fuzzy Sets and Systems*, vol. 48, pp. 297–303, 1992.

[94] W. Z. Qio, W. P. Zhuang, T. H. Heng, and S. S. Shan, "A rule self-regulating fuzzy controller," *Fuzzy Sets and Systems*, vol. 47, pp. 13–21, 1992.

[95] S. Sheoni, K. Ashenayi, and M. Timmerman, "Implementation of a learning controller," *IEEE Control Systems Magazine*, pp. 73–80, June 1995.

[96] W. H. Bare, R. J. Mulholland, and S. S. Sofer, "Design of a self-tuning rule based controller for a gasoline refinery catalytic reformer," *IEEE Trans. on Automatic Control*, vol. 35, pp. 156–164, 1990.

[97] C. W. de Silva and A. G. J. MacFarlane, *Knowledge-based control with application to robots*, vol. 123, Springer-Verlag, 1989.

[98] B.-M. Pfeiffer and R. Isermann, "Self-tuning of classical controllers with fuzzy-logic," *Mathematics and Computers in Simulation*, vol. 37, pp. 101–110, 1994.

[99] R. Ketata, D. D. Geest, and A. Titli, "Fuzzy controller: design, evaluation, parallel and hierarchical combination with a PID controller," *Fuzzy Sets and Systems*, vol. 71, pp. 113–129, 1995.

[100] M. Maeda and S. Murakami, "Fuzzy gain scheduling of PID controllers," *Fuzzy Sets and Systems*, vol. 51, pp. 29–40, 1992.

[101] R. M. Tong, "Synthesis of fuzzy models for some industrial processes," *Internat. J. General Systems*, vol. 4, pp. 143–162, 1978.

[102] R. M. Tong, "The evaluation of fuzzy models derived from experimental data," *Fuzzy Sets and Systems*, vol. 4, pp. 1–12, 1980.

[103] E. Czogala and W. Pedrycz, "On identification in fuzzy systems and its applications in control problems," *Fuzzy Sets and Systems*, vol. 6, pp. 73–83, 1981.

[104] L.-X. Wang, *Adaptive Fuzzy Systems and Control:Design and Stability Analysis*, Prentice-Hall, New Jersey, USA, 1994.

[105] B. P. Graham and R. B. Newell, "Fuzzy adaptive control of a first-order process," *Fuzzy Sets and Systems*, vol. 31, pp. 47–65, 1989.

[106] O. Nelles, M. Fischer, and R. Isermann, "Identification of fuzzy models for predictive control of nonlinear time-variant processes," in *Proceedings of the IFAC Symposium on Artificial Intelligence in Real-Time Control*, vol. 1, pp. 457–462, Kuala Lumpur, Malaysia, Sept. 22-25 1997.

[107] W. T. Miller, "Real time application of neural networks for sensor-based control of robots with vision," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 19, pp. 825–831, 1989.

[108] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 23, pp. 665–685, 1993.

[109] A. Homaifar and E. McCormick, "Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 3, pp. 129–138, 1995.

[110] K. Krishnakumar, "Microgenetic algorithms for stationary and nonstationary function optimization," *in SPIE Proc. Intell. Control Adaptive Syst.*, vol. 1196, pp. 289–296, 1989.

[111] H.-J. Cho, K.-B. Cho, and B.-H. Wang, "Fuzzy hybrid control: Automatic rule generation using genetic algorithms," *Fuzzy Sets and Systems*, vol. 92, pp. 305–316, 1997.

[112] R. R. Yager, "Alternative structures for knowledge representation in fuzzy logic controllers," in *Fuzzy Control Systems* (A. Kandel and G. Langholz, ed.), pp. 99–137, CRC Press, Inc. USA, 1993.

[113] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning I, II, III," *Information Sciences*, vol. 8 and 9, 1975.

[114] K. L. Tang and R. J. Mulholland, "Comparing fuzzy logic with classical controller designs," *IEEE Trans. on System, Man, and Cybernetics*, vol. 17, pp. 1085–1087, 1987.

[115] D. E. Thomas and B. A.-Hélouvry, "Fuzzy logic control - A taxonomy of demonstrated benefits," *Proceedings of the IEEE*, vol. 83, pp. 407–421, 1995.

[116] F. L. Lewis and K. Liu, "Towards a paradigm for fuzzy logic control," *Automatica*, vol. 32, pp. 167–181, 1996.

[117] B. Armstrong, "FLC design for bounded separable functions with linear input-output relations as a special case," *IEEE Trans. on Fuzzy Systems*, vol. 4, pp. 72–79, 1996.

[118] M. Zhung and D. P. Atherton, "Automatic tuning of optimum PID controllers," *IEE Proceedings-D Control Theory Applic.*, vol. 140, pp. 216–224, 1993.

[119] J. Hoschek and D. Lasser, *Fundamentals of Computer Aided Geometric Design*, A K Peters, Massachusetts, 1993.

[120] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design, A practical guide*, Academic Press, USA, 3 ed., 1993.

[121] S. Bennett, "Development of the PID controller," *IEEE Control System Magezine*, pp. 58–63, Dec. 1993.

[122] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *ASME Trans.*, vol. 64, pp. 759–768, 1942.

[123] J. G. Ziegler and N. B. Nichols, "Process lags in automatic-control circuits," *ASME Trans.*, pp. 433–444, July 1943.

[124] A. M. Lopez, J. A. Miller, C. L. Smith, and P. W. Murrill, "Tuning Controllers with error-integral criteria," *Instrumentation Technology*, vol. 3, pp. 57–62, 1967.

[125] A. A. Rovira, P. W. Murrill, and C. L. Smith, "Tuning controllers for set-point changes," *Instrument and control systems*, vol. 42, pp. 67–69, 1969.

[126] A. Kaya, O. Akron, and T. J. Scheib, "Tuning of PID controllers of different structures," *Control Engineering*, pp. 62–65, July 1988.

[127] B. Z. Khan and B. Lehman. "Setpoint PI controllers for systems with large normalized dead time," *IEEE Trans. on Control Systems Tech.*, vol. 4, pp. 459–466. 1996.

[128] C. C. Hang, K. J. Åström, and W. K. Ho, "Refinements of the Ziegler-Nichols tuning formula," *IEE Proceedings-D Control Theory Applic.*, vol. 138, pp. 111–118. 1991.

[129] K. J. Åström and T. Hägglund. "Automatic tuning of simple regulators with specifications on phase and gain margins," *Automatica*, vol. 20, pp. 645–651, 1984.

[130] W. K. Ho, C. C. Hang, and L. S. Cao, "Tuning of PID controllers based on gain and phase margin specifications," *Automatica*, pp. 497–502, March 1995.

[131] W. K. Ho, P. Gan, E. B. Tay, and E. L. Ang, "Performance and gain and phase margins of well-known PID tuning formulas," *IEEE Trans. on Control Systems Technology*, vol. 4, pp. 473–477. 1996.

[132] L. Wang, T. J. D. Barnes, and W. R. Cluett, "New frequency-domain design method for PID controllers," *IEE Proc.-Control Theory Appl.*, vol. 142, pp. 265–271. 1995.

[133] T. J. Pemberton, "PID: The logical control algorithm," *Control Engineering*, vol. 19, pp. 66–67. 1972.

[134] D. E. Rivera, M. Morari, and S. Skogestad, "Internal model control. 4. PID controller design," *Automatica*, vol. 25, pp. 252–265. 1986.

[135] M. Morari and P. Campo. "Response to comments on internal model control. 4. PID controller design," *Ind. Eng. Chem. Res.*, vol. 26, pp. 3163–2163. 1987.

[136] C. C. Hang, W. K. Ho, and L. S. Cao, "A comparison of two design methods for PID controllers," *ISA Trans.*, vol. 33, pp. 147–151. 1994.

[137] A. M. Lopez, P. W. Murrill, and C. L. Smith. "Tuning PI and PID digital controllers," *Instrument and Control Systems*, vol. 42, pp. 89–95. Feb. 1969.

[138] K. J. Åström, P. P. C. C. Hang, and W. K. Ho, "Towards intelligent PID control," *Automatica*, vol. 28, pp. 1–9. 1992.

[139] T. J. Harris and B. D. Tyreus, "Comments on internal model control. 4. PID controller design," *Ind. Emg. Chem. Res.*, vol. 26, pp. 2161–2162. 1987.

[140] K. J. Åström and T. Hägglund. *Automatic Tuning of PID Controllers*. Instrument Society of America, Research Triangle Park, USA, 1988.

[141] K. Ogata, *Modern Control Engineering*. Prentice Hall, NJ. USA, 3 ed., 1997.

[142] T. Hägglund, "A predictive PI controller for processes with long dead time," *IEEE Control Systems Magazine*, pp. 57–60, Feb. 1992.

[143] J. A. Miller, A. M. Lopez, C. L. Smith, and P. W. Murill, "A comparison of controller," *Control Engineering*, vol. 14, pp. 72–75, Dec. 1967.

[144] R. M. A. D. Rajapakse, "Application of genetic algorithms and fuzzy logic techniques for adaptive control of nonlinear systems," *Ph.D. dissertation*, Department of Quantum Engineering and Systems Science, The University of Tokyo, Japan, Sept. 1998.

[145] J. -X. Xu, C. Liu and C. C. Hang, "Tuning of fuzzy PI controllers based on gain/phase margin specifications and ITAE index," *ISA Transactions*, vol. 35, pp. 79–91, 1996.

[146] J. -X. Xu, C. Liu and C. C. Hang, "Designing a stable fuzzy PI control system using extended circle criterion," *Int. J. of Intelligent Control and Systems*, vol. 1, pp. 355–366, 1996.

[147] S. Hayashi, "Auto-tuning fuzzy PI controller," in *Proceedings of the Intn'l Fuzzy Systems Association Conference*, pp. 41–44, Brussels, 1991.

[148] L. Zheng, "A practical guide to tune of fuzzy proportional and integral (PI) like fuzzy controllers," in *Proceedings of the IEEE Intn'l Conf. on Fuzzy Systems*, pp. 633–640, San Diego,CA, March 1992.

[149] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-wesley Pub. Co..Inc..USA, 1989.

[150] C. R. Houck, J. A. Joines and M. G. Kay, "A genetic algorithm for function optimization: A Matlab implementation," in *www.mathworks.com* ,1994.

[151] R. M. DeSantis, "A novel PID configuration for speed and position control," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 116, pp. 542-550, 1994.

Publications during the course of the Ph.D. program

Journal Papers

1. G. K. I. Mann. B.-G. Hu and R. G. Gosine. "Analysis of direct action fuzzy PID controller structures." *IEEE Trans. on System. Man. and Cybernetics Part B: Cybernetics.* Accepted as a regular paper for publication in June 1998.

2. B.-G. Hu. G. K. I. Mann and R. G. Gosine. "Control curve design for nonlinear (or fuzzy) proportional control action using spline-based functions." *Automatica.* vol. 34, pp. 1125-1133, 1998.

3. B.-G. Hu. G. K. I. Mann and R. G. Gosine. "Study of dimensionality of fuzzy inference for fuzzy PID controllers". *Acta Automatica Sinica* vol. 24. No. 5. (in Chineese), 1998.

4. G. K. I. Mann, B.-G. Hu and R. G. Gosine, "Two-level tuning of fuzzy PID controllers" *IEEE Trans. on System. Man. and Cybernetics Part B: Cybernetics* (submitted for review).

5. G. K. I. Mann, B.-G. Hu and R. G. Gosine. "Time-domain based design and analysis of new PID tuning rules." *IEE Proceedings-Control Theory and Applications* (submitted for review).

6. B.-G. Hu, G. K. I. Mann and R. G. Gosine. "New methodology for analytical and optimal design of fuzzy PID controllers" *IEEE Trans. on Fuzzy Systems* (accepted as regular paper).

7. B.-G. Hu. G. K. I. Mann and R. G. Gosine. "Functional evaluation approach of fuzzy PID controllers" *IEEE Trans. on Fuzzy Systems* (submitted for review).

Conference Papers

1. G. K. I. Mann. B.-G. Hu and R. G. Gosine. "Derivation and analysis of three-input inference for fuzzy PID controllers," *Proceedings of the IEEE International Conference in Systems. Man. and Cybernetics.* California, pp. 1910-1915. Oct. 1998.

2. G. K. I. Mann, B.-G. Hu and R. G. Gosine. "Evaluation of fuzzy reasoning schemes for fuzzy controllers," *Proceedings of the Sixth International Conference on Fuzzy Theory and Technology.* North Carolina. vol. I. pp. 175-178. Oct. 1998.

3. G. K. I. Mann. B.-G. Hu and R. G. Gosine. "How fuzzy PID controllers are manually tuned for better performance?," *Proceedings of the Sixth International Conference on Fuzzy Theory and Technology.* North Carolina, vol. I, pp. 171-174,Oct. 1998.

4. B.-G. Hu. G. K. I. Mann and R. G. Gosine. "Nonlinearity variation analysis of one-input fuzzy PID controllers," *Proceedings of the IEEE International Conference in Systems. Man. and Cybernetics.* California, pp. 1916-1921. Oct. 1998.

5. G. K. I. Mann, B.-G. Hu and R. G. Gosine, "Analysis and performance evaluation of linear-like fuzzy PI and PID controllers," *Proceedings of the Sixth International Conference in Fuzzy Systems*, Barcelona, Spain, pp. 383-390, July 1997.

6. G. K. I. Mann, B.-G. Hu and R. G. Gosine, "Fuzzy PID controller structures," *Proceedings, IEEE 1997 Canadian Conference on Electrical and Computer Engineering*, St. John's, Canada, pp. 788-791, May 1997.

7. G. K. I. Mann, B.-G. Hu and R. G. Gosine, "Analysis and development of new PID controller tuning rules for first-order processes," *Preprints, IFAC Symposium on Artificial Intelligence in Real-Time Control*, Kula Lumpur, Malayasia, pp. 203-209, Sept. 1997.

8. B.-G. Hu, G. K. I. Mann and R. G. Gosine, "Theoretic and genetic designs of a three-rule fuzzy PI controller," *Proceedings of the Sixth International Conference in Fuzzy Systems*, Barcelona, Spain, pp. 489-496, July 1997.

9. B.-G. Hu, G. K. I. Mann and R. G. Gosine, "Theoretic derivation of control actions for three rule fuzzy PID controllers," *Proceedings, IEEE 1997 Canadian Conference on Electrical and Computer Engineering*, St. John's, Canada, pp. 792-795, May 1997.

10. B.-G. Hu, G. K. I. Mann and R. G. Gosine, "Fuzzy logic control of a first-order process," *Proceedings, The Seventh Annual Newfoundland Electrical and Computer Engineering Conference*, St. John's, Canada, May 1996.

11. B.-G. Hu, G. K. I. Mann and R. G. Gosine, " How to evaluate fuzzy PID controllers without using process information?," *14 th World Congress of International Federation of Automatic Control*, Beijing, China, 1999.

Appendix A

Derivation of Nonlinear Terms in the Three-Input Inferences

A.1 Nonlinear Term of the LLFLC Output

For this derivation, assume the incremental input values from the reference modal positions given in **step 3** of the solution procedure shown in section 3.3.2 satisfy the following condition.

$$\frac{\delta x_{3.ka}}{a_3} \leq \frac{\delta x_{2.ja}}{a_2} \leq \frac{\delta x_{1.ia}}{a_1} \leq 0.5 \text{ or} \tag{A.1}$$
$$\frac{\delta x_{3.ka}}{a_3} \leq \frac{\delta x_{2.ja}}{a_2} \leq 1 - \frac{\delta x_{1.ia}}{a_1} \leq 0.5.$$

The shaded areas in Figure A.1 show these relative input conditions. This particular region is selected to give a simple and concise expression for the nonlinear term $\mathcal{B}_3$. Any other region in the incremental input space is then transformed to this space by the input transformation shown in **step 4** of the solution procedure in section 3.3.2. Consider the linear rule base given for the general three-input case by (3.20) in Chapter 3. For given crisp inputs $\{\hat{e}_1^*, \hat{e}_2^*, \hat{e}_3^*\}^T$ the final control decision U' is determined by applying the Zadeh-Mamdani's min-max reasoning (compositional rule of inference) as described in [5]. It is given by.

$$\mu_{U'}(\hat{u}) = \max_{i,j,k} \min \left[\mu_{E_{1.i}}(\hat{e}_1^*), \mu_{E_{2.j}}(\hat{e}_2^*), \mu_{E_{3.k}}(\hat{e}_3^*), \mu_U(u)\right]. \tag{A.2}$$

The fuzzy inference will fire a maximum of eight fuzzy rules to produce eight non-zero fuzzy outputs (clipped outputs) against any arbitrary three fuzzy singleton input values. The

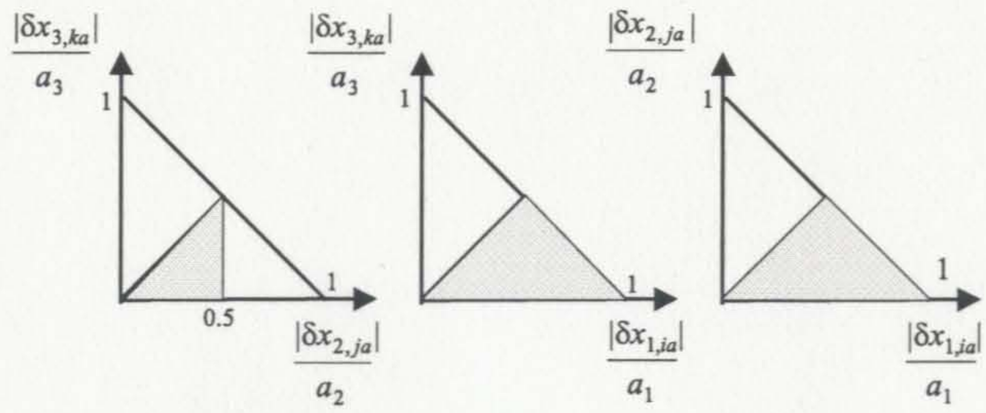


Figure A.1: Relative positions of the incremental inputs

clipped fuzzy output produced by a single rule inference is a trapezoid. After the union of all clipped outputs, the final fuzzy set, U' can have four different shapes with respect to the reference crisp output position $\hat{u}_{ia+ja+ka}$. The reference output is when all crisp inputs are at membership modal positions. The input conditions and the resultant fuzzy outputs corresponding to each case is shown in Figure A.2. For convenience the subscript $ia+ja+ka$ is represented by m_a in the figures and tables of this appendix. The membership functions used for each rule fired and the heights (h) of the trapezoids produced for each rule are shown in Table A.1 through A.4. As an example, the rule R1 shown in Table A.2 reads as "**If** ($\hat{e}_1$ is $E_{1,i}$ and $\hat{e}_2$ is $E_{2,j}$ and $\hat{e}_3$ is $E_{3,k}$) **then** $\hat{u}$ is U_{i+j+k}". The rules having the same output fuzzy labels are combined by the "max" operation. Thus the maximum height of the trapezoids having the same support sets is described by ($h_{\max}$).

Defuzzification: The COA based defuzzified value can be expressed as [5],

$$\hat{u} = \frac{\int_{u \in U} u \mu_u(u) du}{\int_{u \in U} \mu_u(u) du} \tag{A.3}$$

where the membership function U' with its support set is given by $U = \{u | \mu_u(u) > 0\}$. This refers to the center of the shaded areas shown in Figure A.2. From these diagrams the membership heights shown in the $h_{\max}$ columns of tables can be expressed as follows.

$$\mu_{r+1} = \mu_{r-1} = \frac{|\delta x_{w,r}|}{a_w}, \quad \mu_r = 1 - \frac{|\delta x_{w,r}|}{a_w} \tag{A.4}$$

$$\text{where } r = ia, ja, ka \text{ and } w = 1, 2, 3.$$

Using the membership equations above and applying the COA defuzzification method for each case, the nonlinear term β_3 is obtained as shown in the Table A.5. The values α_1 and α_2 are given by equation (3.30) with,

$$|m_1| = \frac{|\delta x_{1,ia}|}{a_1}, \; |m_2| = \frac{|\delta x_{2,ja}|}{a_2}, \; |m_3| = \frac{|\delta x_{3,ka}|}{a_3}. \tag{A.5}$$

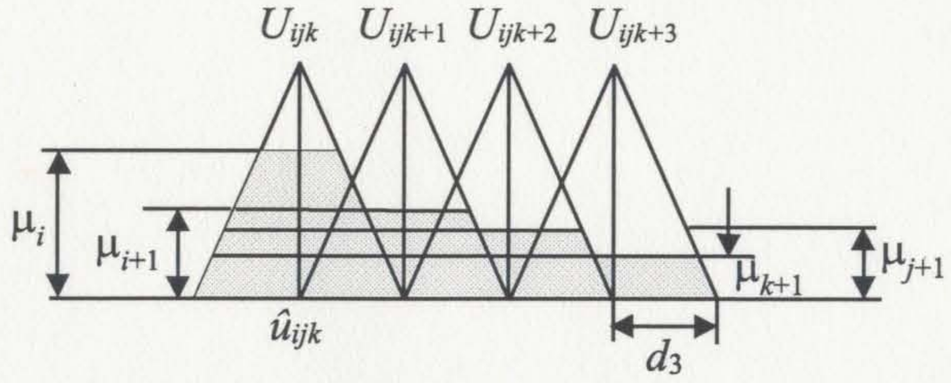
Table A.1: Rule implication and fuzzy out puts for Case I

Rule	$\hat{e}_1$	$\hat{e}_2$	$\hat{e}_3$	$\hat{u}$	h	$h_{\max}$
R1	$E_{1,i}$	$E_{2,j}$	$E_{3,k}$	U_{m_a}	μ_i	μ_i
R2	$E_{1,i+1}$	$E_{2,j}$	$E_{3,k}$	U_{m_a+1}	μ_{i+1}	
R3	$E_{1,i}$	$E_{2,j+1}$	$E_{3,k}$	U_{m_a+1}	μ_{j+1}	μ_{i+1}
R4	$E_{1,i}$	$E_{2,j}$	$E_{3,k+1}$	U_{m_a+1}	μ_{k+1}	
R5	$E_{1,i+1}$	$E_{2,j+1}$	$E_{3,k}$	U_{m_a+2}	μ_{j+1}	
R6	$E_{1,i+1}$	$E_{2,j}$	$E_{3,k+1}$	U_{m_a+2}	μ_{k+1}	μ_{j+1}
R7	$E_{1,i}$	$E_{2,j+1}$	$E_{3,k+1}$	U_{m_a+2}	μ_{k+1}	
R8	$E_{1,i+1}$	$E_{2,j+1}$	$E_{3,k+1}$	U_{m_a+3}	μ_{k+1}	μ_{k+1}

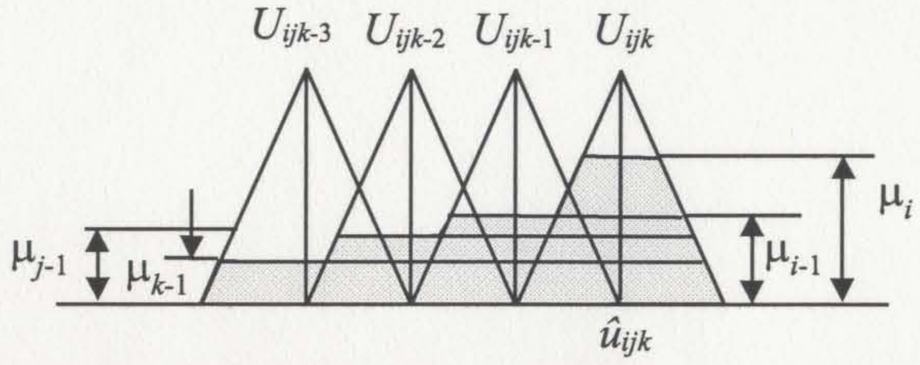
Table 3.1 given in section 3.3.2 shows 8 cases with respect to the sign of the incremental inputs. The extra four cases shown have been transformed to the region shown in Figure A.1 by modifying the modal position and thus changing the direction of the maximum incremental input. This procedure would eliminate the use of an excessive number of formulae for representing different input conditions.

A.2 Nonlinear Term of the NLFLC-I Output

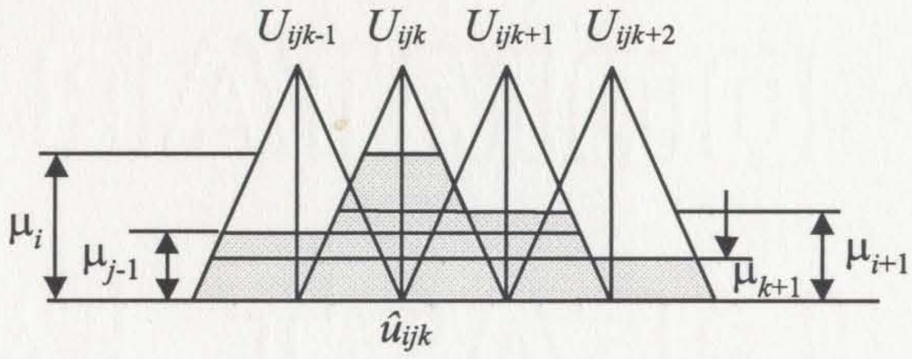
From the results shown in Chapter 3, the nonlinear term of the NLFLC-I is given by the same β_3 derived in the LLFLC output except the incremental input values are modified according to the modified steps 2 and 3, as it was shown in section 3.3.3. The modified **step 2** identifies the corresponding antecedent membership functions of the input variables in order to fire the rules during the inference. The inputs are then transformed to the same region as shown in Figure A.1. As the output membership functions are uniformly distributed the nonlinear term is unique with respect to the normalized incremental input terms.



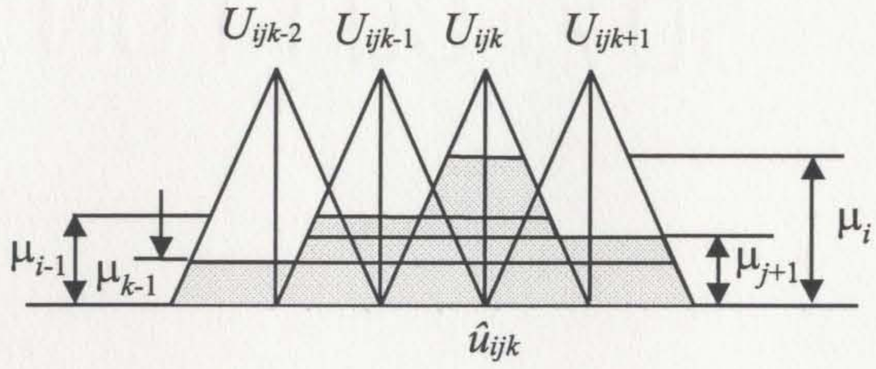
Case I: $\delta x_{1,ia} \geq 0, \delta x_{2,ja} \geq 0$ and $\delta x_{3,ka} \geq 0$.



Case II: $\delta x_{1,ia} \leq 0, \delta x_{2,ja} \leq 0$ and $\delta x_{3,ka} \leq 0$.



Case III: $\delta x_{1,ia} \geq 0, \delta x_{2,ja} \leq 0$ and $\delta x_{3,ka} \geq 0$.



Case IV: $\delta x_{1,ia} \leq 0, \delta x_{2,ja} \geq 0$ and $\delta x_{3,ka} \leq 0$.

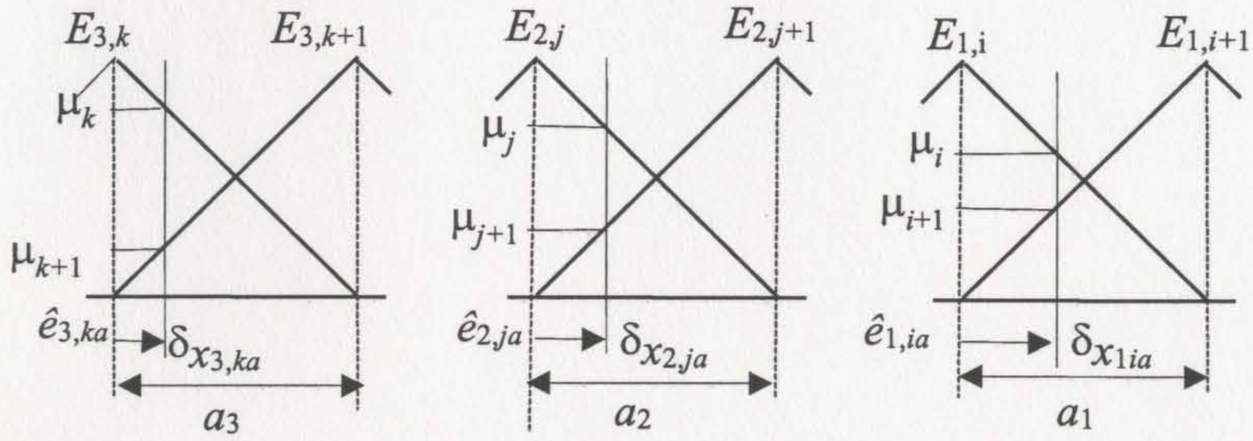


Figure A.2: The LLFLC fuzzy output shapes corresponding to different input conditions. The incremental inputs are measured from the modal positions. The subscript $m_a \equiv ia + ja + ka$.

Table A.2: Rule implication and fuzzy outptus for Case II

Rule	$\hat{e}_1$	$\hat{e}_2$	$\hat{e}_3$	$\hat{u}$	h	$h_{\max}$
R1	$E_{1,i}$	$E_{2,j}$	$E_{3,k}$	U_{m_a}	μ_i	μ_i
R2	$E_{1,i-1}$	$E_{2,j}$	$E_{3,k}$	U_{m_a-1}	μ_{i-1}	
R3	$E_{1,i}$	$E_{2,j-1}$	$E_{3,k}$	U_{m_a-1}	μ_{j-1}	μ_{i-1}
R4	$E_{1,i}$	$E_{2,j}$	$E_{3,k-1}$	U_{m_a-1}	μ_{k-1}	
R5	$E_{1,i-1}$	$E_{2,j-1}$	$E_{3,k}$	U_{m_a-2}	μ_{j-1}	
R6	$E_{1,i-1}$	$E_{2,j}$	$E_{3,k-1}$	U_{m_a-2}	μ_{k-1}	μ_{j-1}
R7	$E_{1,i}$	$E_{2,j-1}$	$E_{3,k-1}$	U_{m_a-2}	μ_{k-1}	
R8	$E_{1,i-1}$	$E_{2,j-1}$	$E_{3,k-1}$	U_{m_a-3}	μ_{k-1}	μ_{k-1}

Table A.3: Rule implication and fuzzy outptus for Case III

Rule	$\hat{e}_1$	$\hat{e}_2$	$\hat{e}_3$	$\hat{u}$	h	$h_{\max}$
R1	$E_{1,i}$	$E_{2,j-1}$	$E_{3,k}$	U_{m_a-1}	μ_{j-1}	μ_{j-1}
R2	$E_{1,i}$	$E_{2,j}$	$E_{3,k}$	U_{m_a}	μ_i	
R3	$E_{1,i+1}$	$E_{2,j-1}$	$E_{3,k}$	U_{m_a}	μ_{j-1}	μ_i
R4	$E_{1,i}$	$E_{2,j-1}$	$E_{3,k+1}$	U_{m_a}	μ_{k+1}	
R5	$E_{1,i+1}$	$E_{2,j-1}$	$E_{3,k+1}$	U_{m_a+1}	μ_{k+1}	
R6	$E_{1,i}$	$E_{2,j}$	$E_{3,k+1}$	U_{m_a+1}	μ_{k+1}	μ_{i+1}
R7	$E_{1,i+1}$	$E_{2,j}$	$E_{3,k}$	U_{m_a+1}	μ_{i+1}	
R8	$E_{1,i+1}$	$E_{2,j}$	$E_{3,k+1}$	U_{m_a+2}	μ_{k+1}	μ_{k+1}

Table A.4: Rule implication and fuzzy outptus for Case IV

Rule	$\hat{e}_1$	$\hat{e}_2$	$\hat{e}_3$	$\hat{u}$	h	$h_{\max}$
R1	$E_{1,i-1}$	$E_{2,j}$	$E_{3,k-1}$	U_{m_a-2}	μ_{k-1}	μ_{k-1}
R2	$E_{1,i-1}$	$E_{2,j}$	$E_{3,k}$	U_{m_a-1}	μ_{i-1}	
R3	$E_{1,i-1}$	$E_{2,j+1}$	$E_{3,k-1}$	U_{m_a-1}	μ_{k-1}	μ_{i-1}
R4	$E_{1,i}$	$E_{2,j}$	$E_{3,k-1}$	U_{m_a-1}	μ_{k-1}	
R5	$E_{1,i}$	$E_{2,j}$	$E_{3,k}$	U_{m_a}	μ_i	
R6	$E_{1,i-1}$	$E_{2,j+1}$	$E_{3,k}$	U_{m_a}	μ_{j+1}	μ_i
R7	$E_{1,i}$	$E_{2,j+1}$	$E_{3,k-1}$	U_{m_a}	μ_{k-1}	
R8	$E_{1,i}$	$E_{2,j+1}$	$E_{3,k}$	U_{m_a+1}	μ_{j+1}	μ_{j+1}

Table A.5: The nonlinear output term

Case	I	II	III	IV
β_3	α_1	$-\alpha_1$	α_2	$-\alpha_2$

A.3 Nonlinear Term of the NLFLC-II Output

With the modified steps 2 and 3 assume the incremental inputs satisfy the modified condition given in (A.1).

$$\frac{\delta x_{3.ka}}{a_{3.ka}} \leq \frac{\delta x_{2.ja}}{a_{2.ja}} \leq \frac{\delta x_{1.ia}}{a_{1.ia}} \leq 0.5 \text{ or} \tag{A.6}$$
$$\frac{\delta x_{3.ka}}{a_{3.ka}} \leq \frac{\delta x_{2.ja}}{a_{2.ja}} \leq 1 - \frac{\delta x_{1.ia}}{a_{1.ia}} \leq 0.5.$$

Use the same input transformation given in the **step 4** of the LLFLC solution procedure to transform any other input conditions to the transformed region shown in Figure A.1. First, for any given crisp input values the control decision is determined by the Z-M min-max reasoning as shown in equation (A.2). With fuzzy singletons in the outputs the truth values will become zero. i.e. $\mu_{U'}(\hat{u}) = 0 \ \vee \hat{u} \neq \hat{u}_m$. Therefore the COA defuzzification only yields the weighted sum of the singleton values given by,

$$\hat{u} = \frac{\sum_{u \in U_m} \hat{u}_m \mu_u(u)}{\sum_{u \in U_m} \mu_u(u)} \tag{A.7}$$

where the membership function U' with its support set is given by the singleton values $U_m = \{u | \mu_u(u) > 0\}$. The maximum truth values produced by the inference are similar to those shown in the Tables A.1-A.4. The resultant singleton outputs correspond to four different cases are shown in Figure A.3.

For the four cases shown in the Figure A.3 the fuzzy outputs are given by the following expressions.

For Case I:

$$\hat{u} = \frac{\mu_i \hat{u}_{m_a} + \mu_{i+1} \hat{u}_{m_a+1} + \mu_{k+1} \hat{u}_{m_a+2} + \mu_{j+1} \hat{u}_{m_a+3}}{\mu_i + \mu_{i+1} + \mu_{k+1} + \mu_{j+1}} \tag{A.8}$$

For Case II:

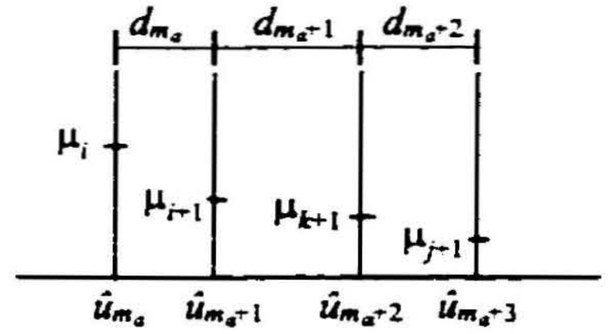
$$\hat{u} = \frac{\mu_i \hat{u}_{m_a} + \mu_{i-1} \hat{u}_{m_a-1} + \mu_{j-1} \hat{u}_{m_a-2} + \mu_{k-1} \hat{u}_{m_a-3}}{\mu_i + \mu_{i-1} + \mu_{j-1} + \mu_{k-1}} \tag{A.9}$$

For Case III:

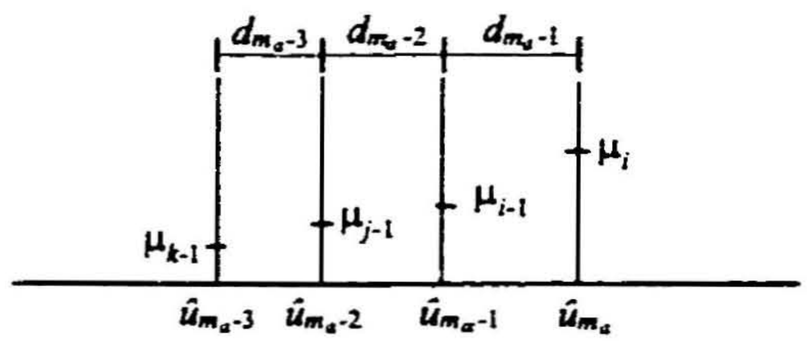
$$\hat{u} = \frac{\mu_{j-1} \hat{u}_{m_a-1} + \mu_i \hat{u}_{m_a} + \mu_{i+1} \hat{u}_{m_a+1} + \mu_{k+1} \hat{u}_{m_a+2}}{\mu_{j-1} + \mu_i + \mu_{i+1} + \mu_{k+1}} \tag{A.10}$$

For Case IV:

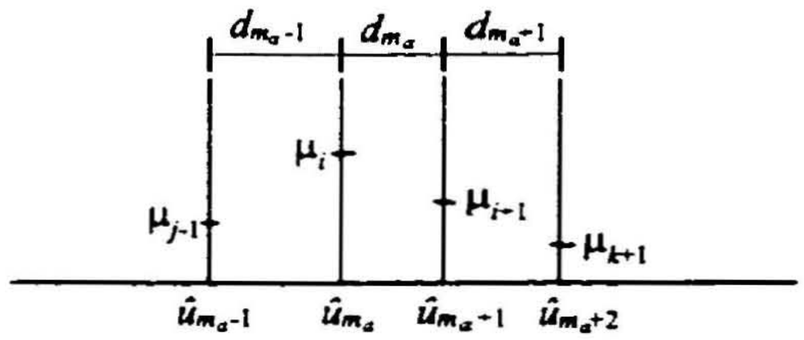
$$\hat{u} = \frac{\mu_{k-1} \hat{u}_{m_a-2} + \mu_{i-1} \hat{u}_{m_a-1} + \mu_i \hat{u}_{m_a} + \mu_{j+1} \hat{u}_{m_a+1}}{\mu_{k-1} + \mu_{i-1} + \mu_i + \mu_{j+1}} \tag{A.11}$$



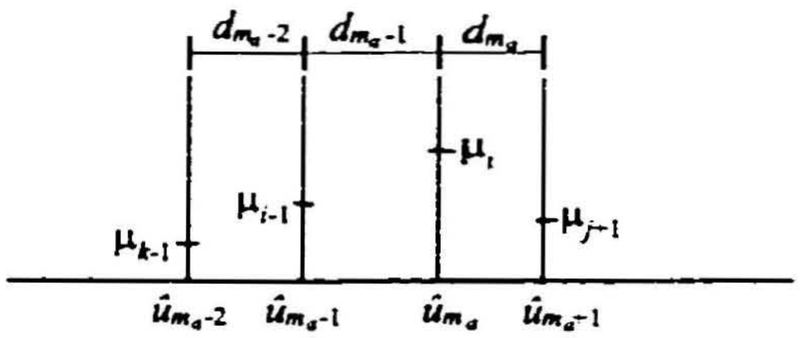
Case I: $\delta x_{1,ia} \geq 0, \delta x_{2,ja} \geq 0$ and $\delta x_{3,ka} \geq 0$.



Case II: $\delta x_{1,ia} \leq 0, \delta x_{2,ja} \leq 0$ and $\delta x_{3,ka} \leq 0$.



Case III: $\delta x_{1,ia} \geq 0, \delta x_{2,ja} \leq 0$ and $\delta x_{3,ka} \geq 0$.



Case IV: $\delta x_{1,ia} \leq 0, \delta x_{2,ja} \geq 0$ and $\delta x_{3,ka} \leq 0$.

Figure A.3: The NLFLC-II fuzzy output shapes corresponding to different input conditions. The subscript $m_a \equiv ia + ja + ka$.

Table A.6: The nonlinear output term

Case	I	II	III	IV
β_{n3}	α_{n1}	$-\alpha_{n4}$	α_{n2}	$-\alpha_{n3}$

Substituting $a_1 = a_{1,ia}, a_2 = a_{2,ja}, a_3 = a_{3,ka}$ into equation (A.4) the above expressions are transformed to the form given in (3.53). The nonlinear term resulting after the simplification are given in Table A.6.

Due to the non-symmetrical nature of the membership shapes the nonlinear term β_{n3} is given by four nonlinear terms as given in the table. The other four cases shown in Table 3.4 are obtained by the input transformation.

Appendix B

Derivation of the NLFLC-III output terms

The NLFLC-III is a three-rule SISO fuzzy system. The output expressions obtained for different fuzzy systems are detailed here.

B.1 MMG Reasoning Based Fuzzy Outputs

For any crisp input value the inference always fires two rules simultaneously except when $\hat{e}_1 = 0$ or $\hat{e}_1 \pm 1$. The same error saturation limits given in **step 1** of the LLFLC solution procedure (equation(3.21)) is imposed. For a given crisp input value $\hat{e}_1^*$, the control decision U' is determined by the compositional rule of inference applied for the three rules in the NLFLC-III system.

$$\mu_{U'}(\tilde{u}) = \max_{i=1,2,3} \min\left[\mu_{E_1^*}, \mu_{U_i}(u)\right]. \tag{B.1}$$

In order to obtain the fuzzy output expressions the COA de-fuzzification in (A.3) is applied for U'.

B.1.1 NLFLC-IIIA

According to the ranges defined for the membership parameters s_1 and s_2, the output has two main cases: overlapping and non-overlapping conditions. Again, due to the discontinuity of

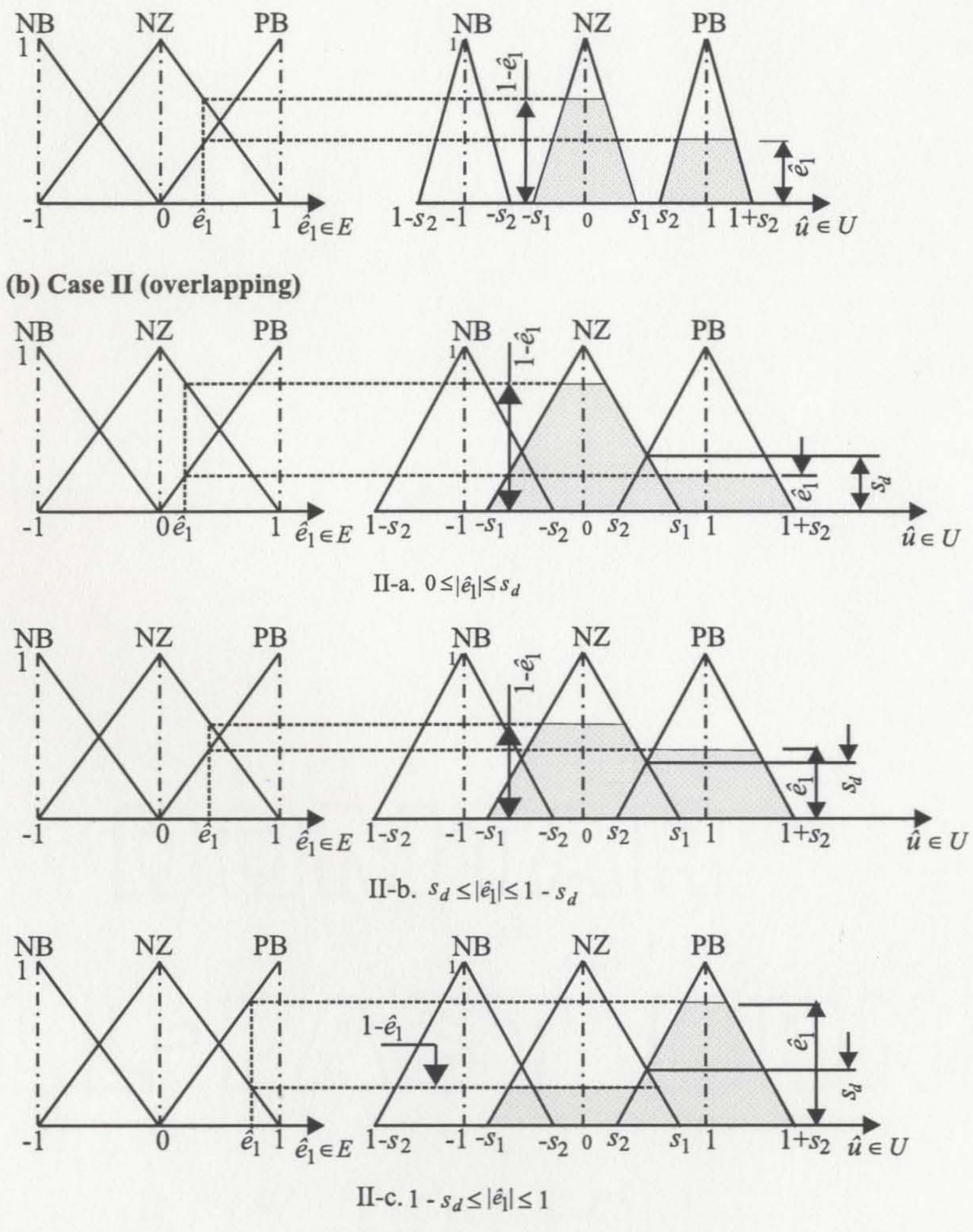


Figure B.1: The MMG reasoning based fuzzy outputs (shaded areas) of the NLFLC-IIIA system.

the min-max functions, the overlapping conditions can have three fundamental output shapes for U', depending on the input error value. The four different shapes correspond to the two cases are shown in Figure B.1. By taking the center of shaded areas for each case the output expressions in (3.61) have been obtained. When $s_2 < 0$ and $|s_2| < s_1$ there are three different fundamental shapes for the fuzzy output. However the final expressions for those conditions are identical to the three sub-cases in the overlapping conditions. The combined conditions are shown in the solution described in equation (3.61).

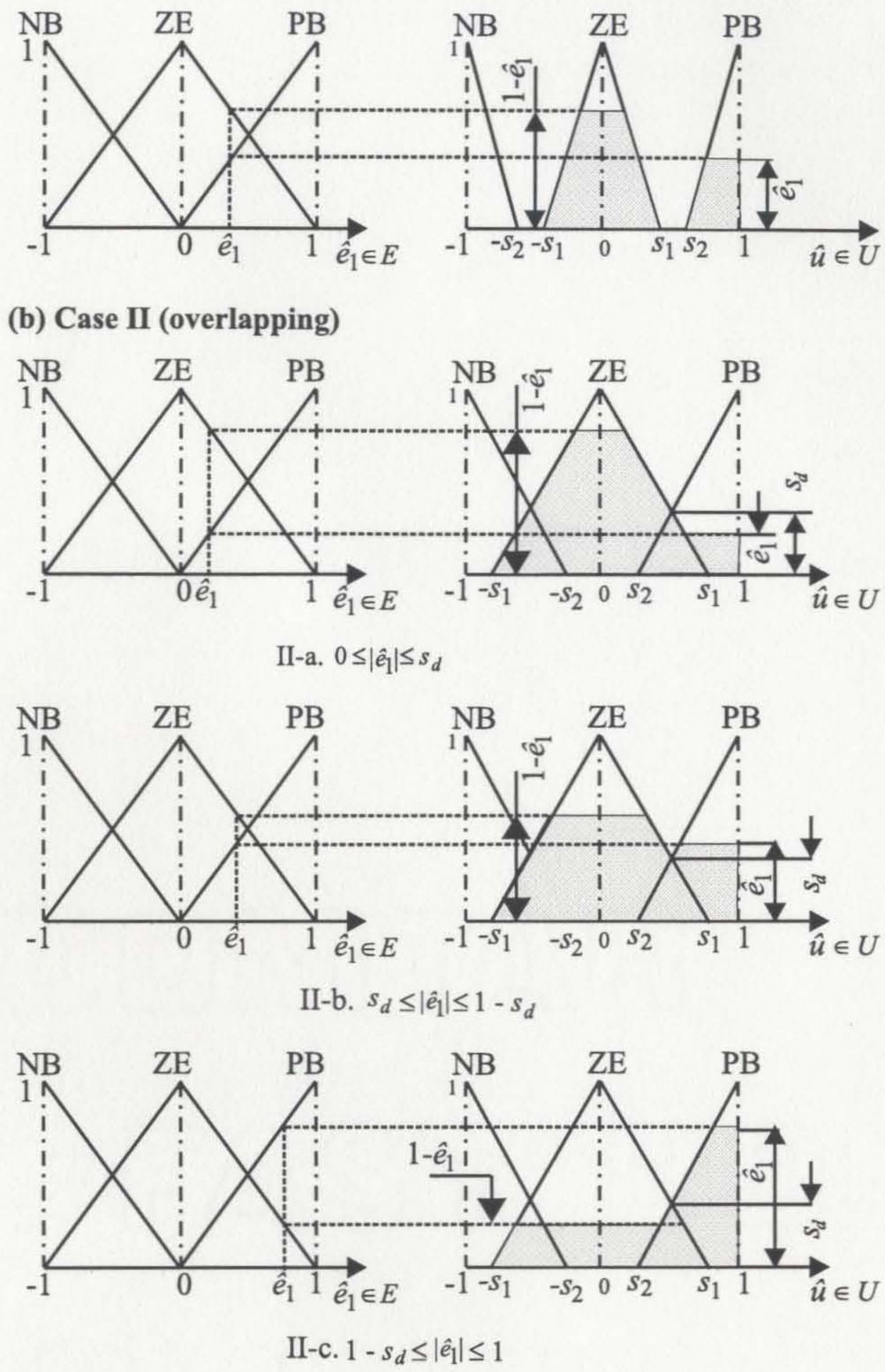


Figure B.2: The MMG reasoning based fuzzy outputs (shaded areas) of the NLFLC-IIIB system

B.1.2 NLFLC-IIIB

This system has partially defined triangular membership functions for the fuzzy subsets NB and PB. The derivation principle similar that shown in the previous section. The four different shapes correspond to non-overlapping and overlapping output membership functions are shown in Figure B.2. By taking the center of shaded areas the expressions in (3.62) have been obtained.

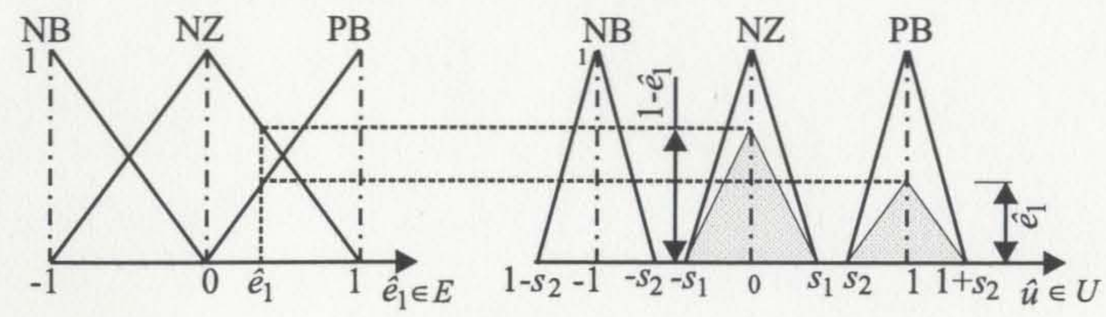


Figure B.3: The PSG reasoning based fuzzy outputs (shaded areas) of the NLFLC-IIIA system

B.2 PSG Reasoning Based Fuzzy Outputs

With the PSG reasoning the product and sum operations replaces the min and max operations in the MMG described in the previous inferences. Therefore for a given crisp input value $\hat{e}_1^*$, the control decision U' for the three rules are now given by;

$$\mu_{U'}(\hat{u}) = \sum_{i=1,2,3} \left[\mu_{E_i} . \mu_{U_i}(u)\right] . \tag{B.2}$$

Using the COA de-fuzzification shown in (A.3) the final output is obtained. The PSG simplification is less complex and the output value does not depend on the overlapping conditions. The resulting fuzzy outputs for both NLFLC-IIIA and NLFLC-IIIB are shown in the Figures B.3 and B.4 respectively.

B.2.1 NLFLC-IIIA

For both the overlapping and non-overlapping cases, the fuzzy output simplifies to a single expression and is given by;

$$\hat{u} = \frac{(1 - s_2)\hat{e}_1}{s_1(1 - \hat{e}_1) + (1 - s_2)\hat{e}_1} \tag{B.3}$$

The above expression is fully normalized within [-1,1]. By taking the first derivative of

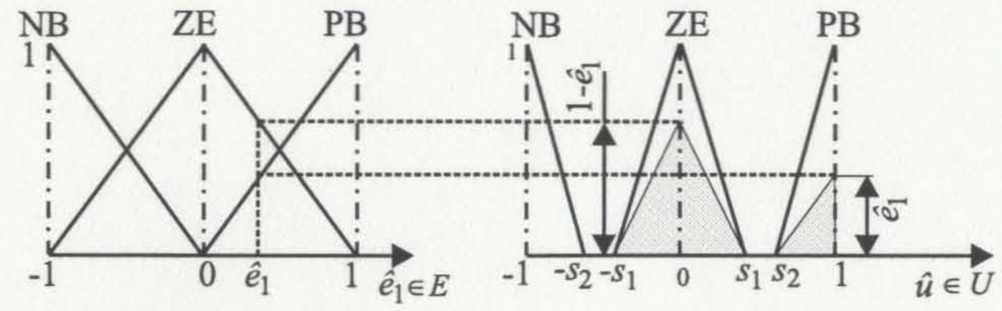


Figure B.4: The PSG reasoning based fuzzy outputs (shaded areas) of the NLFLC-IIIB system

(B.3), the two slopes angles θ_0 and θ_1 can be obtained as;

$$\left.\begin{aligned} \tan\theta_0 &= \frac{d\hat{u}}{d\hat{e}_1} = \frac{1-s_2}{s_1} \\ \tan\theta_1 &= \frac{d\hat{u}}{d\hat{e}_1} = \frac{s_1}{1-s_2} \end{aligned}\right\} \tag{B.4}$$

B.2.2 NLFLC-IIIB

For both the overlapping and non-overlapping cases, the fuzzy output simplifies to a single expression and is given by;

$$\hat{u} = \frac{1}{3}\left[\frac{(1-s_2)(2+s_2)\hat{e}_1}{2s_1(1-\hat{e}_1)+(1-s_2)\hat{e}_1}\right] \tag{B.5}$$

The expression (B.5) is partially normalized and its maximum and minimum values lie within a lower limits. In order to obtain the fuzzy normalized conditions the (B.5) is further divided by $\hat{u}_{\max} = (2+s_2)/3$. For both the overlapping and non-overlapping cases, the fuzzy output simplifies to a single expression and is given by;

$$\hat{u} = \frac{(1-s_2)\hat{e}_1}{2s_1(1-\hat{e}_1)+(1-s_2)\hat{e}_1} \tag{B.6}$$

By taking the first derivative of (B.6), the two slope angles given in (4.15) have been obtained.

B.3 TSK Reasoning Based Fuzzy Outputs

The TSK based fuzzy system uses the same input membership functions that are shown in Figure 3.14. The output is expressed in terms of the input crisp variables by a linear (or nonlinear) form. Considering second order output expressions the corresponding three rules can be expressed in a general form as.

$$\left.\begin{aligned} &\text{If } \hat{e}_1 \text{ is NB then } \hat{u}_1 = a_0 + a_1\hat{e}_1 + a_2\hat{e}_1^2 \\ &\text{If } \hat{e}_1 \text{ is NZ then } \hat{u}_2 = b_0 + b_1\hat{e}_1 + b_2\hat{e}_1^2 \\ &\text{If } \hat{e}_1 \text{ is PB then } \hat{u}_3 = c_0 + c_1\hat{e}_1 + c_2\hat{e}_1^2 \end{aligned}\right\} \tag{B.7}$$

The output of the TSK system is also obtained by using min-max-gravity principle. However TSK system can be solved using the PSG reasoning as well. With the SISO system the MMG and PSG based systems are identical. The final output for the SISO system is given by:

$$\hat{u} = \frac{\sum_{i=1,2,3} \mu_{E_i}(\hat{e}_1)\hat{u}_i}{\sum_{i=1,2,3} \mu_{E_i}} \tag{B.8}$$

Using the three rules in (B.7), the output can be expressed as:

$$\left.\begin{aligned} \hat{u} &= (a_0 + a_1\hat{e}_1 + a_2\hat{e}_1^2)\hat{e}_1 + (b_0 + b_1\hat{e}_1 + b_2\hat{e}_1^2)(1 - \hat{e}_1) \quad \text{for } \hat{e}_1 \leq 0 \\ \hat{u} &= (c_0 + c_1\hat{e}_1 + c_2\hat{e}_1^2)\hat{e}_1 + (b_0 + b_1\hat{e}_1 + b_2\hat{e}_1^2)(1 - \hat{e}_1) \quad \text{for } \hat{e}_1 \geq 0 \end{aligned}\right\} \tag{B.9}$$

The output is then constrained to follow the preferred properties of the fuzzy outputs described in section 4.2.2. First consider the positive range of the error. For obtaining the steady state properties, i.e. for satisfying when $\hat{e}_1 = 0$, $\hat{u} = 0$, set $b_0 = 0$. Similarly for obtaining the normalized conditions, i.e. when $\hat{e}_1 = 1$, $\hat{u} = 1$, set $a_0 + c_1 + c_2 = 1$. For further simplification consider two slope angles related to $\hat{e}_1 = 0$ and $\hat{e}_1 = 1$. In order to reduce the non-linearity tuning to two parameters, and without loss of generality force $b_1 = b_2 = 0$. Assign $s_1 = c_1$ and $s_2 = c_2$. For realizing the anti-symmetric property, set $a_0 = -c_0$, $a_1 = c_1$ and $a_2 = -c_2$. Substituting the above conditions into (B.9), the TSK based output can be expressed in the following simplified form.

$$\hat{u} = \hat{e}_1\left(1 - s_1 - s_2 + s_1|\hat{e}_1| + s_2|\hat{e}_1|^2\right) \tag{B.10}$$

The first derivative of (B.10) for the positive range is given by.

$$\frac{d\hat{u}}{d\hat{e}_1} = (1 - s_1 - s_2) + 2s_1\hat{e}_1 + 3s_2\hat{e}_1^2 \tag{B.11}$$

Substituting $\hat{e}_1 = 0$ and $\hat{e}_1 = 1$ the slope angles in (4.17) hav been obtained. In order to find the valid ranges the monotonic property of the output is imposed. i.e. for any given $\hat{e}_1$ assume $d\hat{u}/d\hat{e}_1 > 0$. Thus the valid ranges given in (4.18) and (4.19) have been obtained.

Appendix C

Time response analysis with zero delay time

The derivation of the peak overshoot and the rise time related to the three cases described in section 5.2.1 (Chapter 5) is detailed as follows.

Case I : The closed-loop poles are real and distinct.

The response equation given in (5.7) can be represented in the partial fraction form as,

$$Y(s) = \frac{K_1}{(K_3 + T)(s + a)(s + b)} + \frac{K_2}{(K_3 + T)(s + a)(s + b)} \tag{C.1}$$

Using the gain relationship in (5.8) the negative poles are given by.

$$a = \frac{1}{2(K_3 + T)} \left((K_1 + 1) - (K_1 - 1)\sqrt{1 - \beta} \right) \tag{C.2}$$

$$b = \frac{1}{2(K_3 + T)} \left((K_1 + 1) + (K_1 - 1)\sqrt{1 - \beta} \right). \tag{C.3}$$

The parameter β adjusts the distance between the two poles.

When $K_1 > 0$ then $0 < a < b$.

When $\beta = 1$ then $a = b$.

Taking the inverse Laplace of (C.1) and then defining the feed back error for unit step response as $y(t) = 1 - e(t)$. the error response is given by.

$$e(t) = \frac{1}{(b - a)} \left[\left(\frac{K_1}{K_3 + T} - a \right) \exp(-bt) - \left(\frac{K_1}{K_3 + T} - b \right) \exp(-at) \right]. \tag{C.4}$$

The rise time corresponding to 0-100 % is given by. $e(T_r) = 0$. Using (A2) and (A3) the rise time given by (5.9) has been obtained. At the peak overshoot. $\frac{de(t)}{dt}\,|_{t=t_m} = 0$. The first derivative of (C.4) simplifies to.

$$\frac{de(t)}{dt} = \frac{\exp(-at)}{(b-a)}\left[-b\left(\frac{K_1}{K_3+T} - a\right)\exp\left(-(b-a)t\right) + a\left(\frac{K_1}{K_3+T} - b\right)\right]. \quad \text{(C.5)}$$

The peak overshoot corresponds to two solutions of $t = t_m$.

$$\exp(-at_m) = 0 \quad \text{or} \quad \text{(C.6)}$$

$$\exp\left(-(b-a)t_m\right) = \frac{a\left(K_1 - b(K_3+T)\right)}{b\left(K_1 - a(K_3+T)\right)} \quad \text{for } K_1 > 1 \quad \text{(C.7)}$$

$$\exp\left(-(a-b)t_m\right) = \frac{a\left(K_1 - a(K_3+T)\right)}{b\left(K_1 - b(K_3+T)\right)} \quad \text{for } K_1 < 1 \quad \text{(C.8)}$$

The first solution in (C.6) corresponds to the steady state conditions of the response. The second solution in (C.7) and (C.8) represents the transient peak. By substituting into the error response in (C.4), we can show,

$$e(t_m) = -\frac{1}{b}\left(\frac{K_1}{K_3+T} - b\right)\exp(-at_m) \quad \text{for } K_1 > 1 \quad \text{(C.9)}$$

$$e(t_m) = -\frac{1}{a}\left(\frac{K_1}{K_3+T} - a\right)\exp(-bt_m) \quad \text{for } K_1 < 1 \quad \text{(C.10)}$$

The overshoot $OS = -e(t_m)$. Using the gain relations in (5.8) and (A2) the peak overshoot in (5.10) has been obtained. For zero overshoot conditions $e(t_m) = 0$. This corresponds to:

$$\left.\begin{aligned} b &= \tfrac{K_1}{K_3+T} \quad \text{for } K_1 > 1 \\ a &= \tfrac{K_1}{K_3+T} \quad \text{for } K_1 < 1 \end{aligned}\right\} \quad \text{(C.11)}$$

Using (C.2) and (C.3) we can show $\beta = 0$. By substituting these conditions into error response we obtain.

$$e(t) = \exp(-bt) \quad \text{for } K_1 > 1$$

$$e(t) = \exp(-at) \quad \text{for } K_1 < 1$$

From (C.11). the rise time corresponding to 10-90% in (5.11) has been obtained.

Case II : Closed-loop poles are real and equal.

With $\beta = 1$.

$$a = b = \frac{K_1 + 1}{2(K_3+T)}. \quad \text{(C.12)}$$

From (5.8),

$$K_2 = \frac{(K_1+1)^2}{4(K_3+T)}. \tag{C.13}$$

The response equation now simplifies to:

$$Y(s) = \frac{K_1}{(K_3+T)(s+a)^2} + \frac{K_2}{(K_3+T)(s+a)^2 s}. \tag{C.14}$$

By taking the inverse Laplace the error response can be shown as.

$$e(t) = \left(at - \frac{K_1 t}{(K_3+T)} + 1\right)\exp(-at). \tag{C.15}$$

From (C.15) the rise time corresponds to 0-100% in (5.12) has been obtained. By equating the first derivative of (C.15) to zero we obtain the peak overshoot corresponds to time.

$$t_m = \frac{4K_1(K_3+T)}{(K_1+1)(K_1-1)}.\ (K_1 \neq 1). \tag{C.16}$$

Substituting (C.16) to (C.15) the peak overshoot in (5.13) has been obtained. When $K_1 = 1$ the response has the critical damping conditions and $t_m = \infty$. From (C.15) the rise time corresponds to 0-100% in (5.14) has been obtained.

Case III : Closed-loop poles are complex with negative real parts.

The response equation in (5.7) can represent by.

$$Y(s) = \frac{K_1}{K_2}\frac{\omega_n^2}{(s^2+2\zeta\omega_n s+\omega_n^2)} + \frac{\omega_n^2}{s(s^2+2\zeta\omega_n s+\omega_n^2)}. \tag{C.17}$$

where $\omega_n = \sqrt{K_2/(K_3+T)}$ and ζ is as defined in (5.15). Using the inverse Laplace of (C.17) the error response is given by.

$$e(t) = -\sqrt{1+L^2}\left[\sin\left(\sqrt{1-\zeta^2}\omega_n t - \vartheta\right)\right]\exp(-\zeta\omega_n t). \tag{C.18}$$

Where L and ϑ are defined in (5.16). By equating the non-exponent term in (C.18) to zero. the rise time corresponds to 0-100% in (5.17) has been obtained. By taking the first derivative of (C.18) the time at the peak overshoot can be obtained as.

$$t_m = \frac{(\psi+\vartheta)}{\omega_n\sqrt{1-\zeta^2}} \tag{C.19}$$

The angle ψ is defined in (5.16). Substituting (C.19) to (C.18) the peak overshoot in (5.16) has been obtained.