

https://dspace.org/

An investigation into the merits of fuzzy logic control versus classical control.

Florence, Stuart Douglas.

1996
https://hdl.handle.net/10539/24641

Downloaded from a DSpace repository. https://dspace.org/

An investigation into the merits of fuzzy logic control versus classical control

S.D.Florence

A project report submitted to the Faculty of Engineering, University of the Witwatersrand, Johannesburg, in partial fulfilment of the requirements for the degree of Master of Science in Engineering.

University of the Witwatersrand, Johannesburg, July 1996.

Declaration

I declare that this project report is my own, unaided work. It is being submitted for the Degree of Master of Science in Engineering in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in any other University.

Signed this 26th day of JULY 19 96.



S.D.Florence

Abstract

Up to now the benefits and problems with fuzzy control have not been fully identified and its role in the control domain needs investigation. The past trend has been to show that a fuzzy controller can provide better control than classical control, without examining what is actually being achieved. The aim in this project report is to give a fair comparison between classical and fuzzy control. Robustness, disturbance rejection, noise suppression, nonminimum phase and dead time are examined for both controllers. The comparison is performed through computer simulation of classical and fuzzy controlled plant models. Fuzzy control has the advantage of non-linear performance and the ability to capture linguistic information. Translating quantitative information into the fuzzy domain is difficult; therefore when the system is easily mathematically modelled and linear, classical control is usually better. Which controller should be used depends on the application, control designer and information available.

Acknowledgements

I would like to thank :

- Dr. B. Wigdorowitz for his time, encouragement and supervision.
- Mandy and my family for their support and encouragement.

Table of Contents

Declaration ..... *i*

Abstract ..... *ii*

Acknowledgements ..... *iii*

Table of Contents ..... *iv*

List of Figures ..... *x*

List of Tables ..... *xiii*

List of Symbols ..... *xiv*

1. Introduction *1*

1.1 What is fuzzy control ? **1**

1.2 Fuzzy control **2**

1.3 Problems and myths surrounding fuzzy logic **3**

1.4 The aim of this investigation **6**

1.5 Approach taken **7**

2. Fuzzy logic concepts required in control *9*

2.1 Definition of a Fuzzy Logic System **9**

2.2 Rationale for Fuzzy Logic **10**

2.3 Fuzzy Set Theory **11**

2.3.1 Fuzzy Sets 11

2.3.2 Properties of Fuzzy Sets 12

2.3.3 Linguistic Variables 13
2.3.4 Fuzzy Numbers 14
2.3.5 Membership Functions 15
2.3.6 Operations on Fuzzy Sets 18
2.3.6.1 Union and Intersection 18
2.3.6.2 Complement 21
2.3.7 Fuzzy Relations and Compositions 22
2.3.7.1 Fuzzy Relations 22
2.3.7.2 Fuzzy Compositions 23

2.4 Fuzzy Logic Systems 26
2.4.1 Input Fuzzification 27
2.4.2 The Rules 27
2.4.2.1 Fuzzy Propositions 27
2.4.2.2 Fuzzy Rules 28
2.4.3 Fuzzy Inference Scheme 29
2.4.3.1 Inference of a single rule 29
2.4.3.2 Inference of a rule base 30
2.4.4 Defuzzification 35
2.4.4.1 Maximum Defuzzifier 35
2.4.4.2 Mean of Maxima Defuzzifier 35
2.4.4.3 Centroid or Centre-of-Gravity Defuzzifier 35
2.4.4.4 Singletons 36

2.5 Fuzzy Systems as Universal Approximators 39

2.6 Adaptive Fuzzy Control 40
2.6.1 Self-organising fuzzy control 40
2.6.2 Fuzzy relations as associative memories 41
2.6.3 Adaptation by fuzzy supervisors 41
2.6.3.1 Fuzzy PID control 41
2.6.3.2 Adaptive fuzzy expert controller[10] 42
2.6.4 Gradient-descent adaptation 42

2.7 Summary 42

3. Aspects of fuzzy and classical control 44

3.1 Definitions 44

3.1.1 Classical Control 44
3.1.2 Fuzzy Control 45

3.2 The classical control loop 46
3.2.1 Reason for feedback 46
3.2.2 Structure of a linear SISO feedback loop 47
3.2.3 Limitations of classical control 48
3.2.4 Background to QFT control 49

3.3 Control Aspects 49
3.3.1 Stability 49
3.3.1.1 QFT stability 50
3.3.1.2 Fuzzy stability 50
3.3.2 Disturbance rejection 52
3.3.2.1 QFT design 52
3.3.2.2 Fuzzy design 52
3.3.3 Sensor noise suppression 53
3.3.3.1 QFT control 53
3.3.3.2 Fuzzy control 54
3.3.4 Parameter variation 54
3.3.4.1 QFT control 55
3.3.4.2 Fuzzy control 55
3.3.5 Unmodelled dynamics 56
3.3.5.1 QFT control 56
3.3.5.2 Fuzzy control 56
3.3.6 Non-minimum phase plants 57
3.3.7 Dead time 58
3.3.8 Saturation 58
3.3.9 Temporal determinism 59
3.3.10 Reliability and safety 59

3.4 Applications of fuzzy control 60
3.4.1 History of fuzzy control 60
3.4.2 Summary of some applications of fuzzy control 61

3.5 Qualitative versus quantitative knowledge 64
3.5.1 Tools available for fuzzy control 66

3.6 Approach to comparing control systems 66

3.7 Summary **67**

4. Plant Modelling ***68***

4.1 Introduction **68**

4.2 Servo motor system **69**

4.2.1 Rationale 69

4.2.2 Assumptions 70

4.2.3 Limitations 70

4.2.4 Mathematical Model 71

4.2.5 Physical Parameters 72

4.2.6 Final Model 72

4.3 CSTR plant model **73**

4.3.1 Rationale 73

4.3.2 Assumptions 74

4.3.3 Mathematical Model 74

4.3.4 Parameter Values 78

4.3.5 Linear Model 78

4.4 Tank System **79**

4.4.1 Rationale 79

4.4.2 Assumptions 80

4.4.3 Limitations 80

4.4.4 Mathematical Model 80

4.4.5 Physical Parameters 81

4.4.6 Linearised plant 81

4.5 Miscellaneous plants **82**

4.6 Summary **82**

5. Controller design and comparison ***84***

5.1 Controller design approach **84**

5.1.1 QFT design 85

5.1.2 Fuzzy design 85

5.2 Controller specification and design **87**

5.2.1 Servo motor 88

5.2.1.1 Plant model 88
5.2.1.2 QFT specifications and design 88
5.2.1.3 Fuzzy specifications and design 89
5.2.2 CSTR plant 90
5.2.2.1 Plant model 91
5.2.2.2 QFT design 91
5.2.2.3 Fuzzy Design 93
5.2.3 Tank system 94
5.2.3.1 Plant model 94
5.2.3.2 QFT design 94
5.2.3.3 Fuzzy control 95

5.3 Controller design and comparison for specific aspects of plants 95
5.3.1 Non-minimum phase plant 96
5.3.2 Highly non-linear plant 99
5.3.3 Hybrid control of a first order plant with dead time 99

5.4 Simulation comparisons 103
5.4.1 Tracking 104
5.4.2 Disturbance rejection 107
5.4.3 Noise suppression 109
5.4.4 Unmodelled dynamics/model variation 111
5.4.5 Non-minimum phase 112
5.4.6 Dead time 112

5.5 Design Comparison 114

5.6 Advantages and disadvantages of fuzzy control 118

5.7 Summary 122

6. Conclusion 124

6.1 Future research 127

Appendix A : Model parameters 128

A.1 Servo motor 128

A.2 CSTR 128

A.3 Tank 131

Appendix B : Block Diagrams *132*

B.1 Servo motor **132**

B.2 CSTR **134**

C.3 Tank **139**

C.4 Nonminimum phase plant **141**

C.5 First order system with dead time **142**

Appendix C: Controllers *143*

C.1 QFT controllers **143**

C.1.1 Servo motor 143

C.1.2 CSTR 144

C.1.3 Tank 145

C.2 Fuzzy controllers **146**

C.2.1 Servo motor 147

C.2.2 CSTR 148

C.2.3 Tank 148

C.2.4 Non-minimum phase 149

C.3 Hybrid Controller **151**

C.3.1 Classical 151

C.3.2 Fuzzy 152

Appendix D : MATLAB® Code *153*

D.1 Servo motor **153**

D.1.1 servo_tf.m 153

D.1.2 pmap_s.m 154

D.1.3 qft_serv.m 159

D.2 CSTR **163**

D.2.1 c_curve.m 163

D.2.2 lin_cstr.m 166

D.2.3 i_cstr.m 173

D.2.4 qft_cstr.m 176

List of Figures

Figure 2-1 : Fuzzy set *tall* 12
Figure 2-2 : Linguistic variable *height* 14
Figure 2-3 : Fuzzy relation $R_{taller}(x, y)$ 23
Figure 2-4: Elements of a fuzzy logic system 26
Figure 2-5 : Fuzzy inference diagram for example 2-10 34
Figure 2-6 : Fuzzy inference diagram for example 2-11 38
Figure 3-1 : SISO feedback structure 47
Figure 3-2 : Distribution of fuzzy applications 65
Figure 4-1 : Circuit diagram of a DC motor 71
Figure 4-2 : Block diagram of DC Motor 72
Figure 4-3 : CSTR Plant 75
Figure 4-4 : CSTR Characteristic curve * - q removed by coil + - q formed by reaction 76
Figure 4-5 : Simplified block diagram of the CSTR 78
Figure 4-6 : Tank block diagram 82
Figure 5-1 : Nichols plot of non-minimum phase plant 97
Figure 5-2 : Step response of the QFT and fuzzy controllers for a non-minimum phase plant 98
Figure 5-3 : Adaptive Smith predictor control scheme 101
Figure 5-4 : Adaptive Smith predictor. (a) tuning steps (b) predictor dead time estimate td' 103
Figure 5-5 : Step response test 105
Figure 5-6 : Sine wave tracking test 106
Figure 5-7 : Step test on CSTR. (a) - QFT controller step response; (b) - fuzzy controller step response; (c) - control action for QFT controller; (d) - control action for fuzzy controller. 107
Figure 5-8 : Step disturbance rejection test results 108
Figure 5-9 : Time varying disturbance test results 109

Figure 5-10 : Servo noise suppression test results (a) - QFT controller response; (b) - fuzzy controller response; (c) - control action for QFT controller; (d) - control action for fuzzy controller..... 110
Figure 5-11 : CSTR noise suppression test results (a) - QFT controller response; (b) - fuzzy controller response; (c) - control action for QFT controller; (d) - control action for fuzzy controller..... 111
Figure 5-12 : Dead time step response test..... 113

Figure B.1: Servo motor main block diagram..... 132
Figure B.2: "Servo motor" subsystem..... 133
Figure B.3 : "Controllers" subsystem..... 133
Figure B.4 : "Noise" subsystem..... 133
Figure B.5 : Main CSTR System..... 134
Figure B.6 : "Controller" subsystem..... 135
Figure B.7 : "Fuzzy Controller" subsystem..... 135
Figure B.8 : "Noise" subsystem..... 135
Figure B.9 : "Band Limited Noise" subsystem..... 136
Figure B.10 : "CSTR" subsystem..... 136
Figure B.11 : "Mass balance equations" subsystem..... 137
Figure B.12 : "Rate of reaction" subsystem..... 137
Figure B.13 : "Block III" subsystem..... 137
Figure B.14 : "Block IV & V" subsystem..... 138
Figure B.15 :"q" subsystem..... 138
Figure B.16 : "Block IV" subsystem..... 139
Figure B.17 : Tank main system..... 139
Figure B.18 : "Controller" subsystem..... 140
Figure B.19 : "Tank" subsystem..... 140
Figure B.20 : Nonminimum phase plant system..... 141
Figure B.21 : First order system with dead time..... 142
Figure C.1 : Nichols plot of servomotor controller..... 143
Figure C.2 : Nichols plot of CSTR controller..... 144
Figure C.3 : Nichols plot of tank controller..... 145

Figure C.4 : Control surface for servo motor fuzzy controller 147
Figure C.5 : Control surface for CSTR fuzzy controller 148
Figure C.6 : Control surface for tank fuzzy controller 149
Figure C.7 : Control surface for non-minimum phase plant fuzzy controller 150
Figure C.8 : Nichols plot for hybrid Smith predictor controller (QFT section) 151
Figure C.9 : Control surface for hybrid Smith predictor controller (fuzzy section) 152

List of Tables

Table 2-1 : Summary of common membership functions 17
Table 2-2 : Effect of choosing different connectives 20
Table 2-3 : : Frequently used operator for and & or connectives 28
Table 3-1 : Summary of some fuzzy application 62
Table 5-1 : Comparison between fuzzy and classical control design 114

Table A.1 : Model parameters for servo motor..... 128
Table A.2 : Model parameter values for CSTR..... 128
Table A.3 : Model parameter values for tank system 131
Table C.1 : Fuzzy set symbols 146

List of Symbols

$\cap$	set intersection/conjunction
$\cup$	set union/disjunction
$\circ$	sup-min composition
$\mu(\cdot)$	membership function
T(·)	T-norm
S(·)	T-conorm/S-norm
I(·)	Inference
core(·)	core of fuzzy set
hgt(·)	height of fuzzy set
supp(·)	support of fuzzy set
x_i	i^{th} numerical input value
x'_i	i^{th} numerical measured input value
X_i	i^{th} input universe
A_i	fuzzy set for i^{th} input universe
$A_{i,k}$	fuzzy set for i^{th} input universe in antecedent of kth rule
r_k	k^{th} fuzzy rule
R	fuzzy relation
y_j	j^{th} numerical output
Y_j	j^{th} output universe
B_j	fuzzy set for j^{th} output universe
B'_j	fuzzy set for j^{th} output universe resulting from inference
CSTR	continuous stirred-tank reactor
dB	decibels
FAM	fuzzy associative memory

FLS	fuzzy logic system
LHP	left hand plane
LTI	linear time-invariant
LTIE	linear time-invariant equivalent
MIMO	multi-input multi-output
MISO	multi-input single-output
NMP	nonminimum phase
PID	proportional-integral-derivative
QFT	quantitative feedback theory/technique
RHP	right half plane
SISO	single-input single-output

1. Introduction

Fuzzy logic has recently become a popular topic in various fields especially control. Many journals, Internet newsgroups[1] and books on the theory and applications of fuzzy logic have become available. Despite all this, there is much misunderstanding about the nature of fuzzy logic and its limitations. The aim of this project report is twofold, firstly to investigate what fuzzy control has to offer and secondly to compare it to classical control.

1.1 What is fuzzy control ?

When controlling a process, human operators encounter complex patterns of process dynamic behaviour which are difficult to interpret and model. Lack of a mathematical model leads to problems in classical control design. These complex process behaviour patterns can often be reduced to a set of linguistic operating rules which the operator uses to enforce control on the process. The rules themselves include imprecise terms (hot, cold, fast, slow) which are multivalued. Fuzzy set theory and the application, fuzzy control, offer a method of using this imprecise multivalued information to control a process. Fuzzy set theory has been used in the fields of linguistics, psychology, economics, information retrieval[1] and the soft sciences (e.g. branching questionnaires[2]). Most problems which are defined using vague terminology can use a form of fuzzy logic to model the problem.

[1] The main newsgroup is `comp.ai.fuzzy`.

Fuzzy set theory deals with sets where the membership is a matter of degree and not absolute, as is the case with classical set theory. Fuzzy set theory serves as the basis for fuzzy logic[1]. Fuzzy logic is a rule based means of inferring an action (consequent) from an input (antecedent) using fuzzy sets. Fuzzy control is the application of fuzzy logic to the control of a physical process.

The fuzzy controller, in particular, consists of a collection of user defined control laws. These laws are fired in parallel. Each rule which is fired, contains a certain truth value which determines the degree to which the rule conditions have been met. The recommended control action is derived by combining the results of each rule in a prescribed manner as described in Chapter 2. The fuzzy controller may be defined as a non-linear mapping from the input space to the output space. The term "fuzzy" comes from the vagueness in the linguistic terms which are used in the statement of the rules. However, the fuzzy controller gives a precise meaning to the vague linguistic terms.

1.2 Fuzzy control

The first paper on fuzzy control was written by Zadeh[3]. This paper and others[4,5 et al] laid out the basic structure of the fuzzy controller. In 1974, Mamdani[6] proposed using fuzzy sets for the control of a physical process. Mamdani and Assilian[7] later reported on the application of fuzzy control to the control of the steam pressure and throttle setting of a small laboratory engine. This was the first practical use of fuzzy control and proved that it offered a viable control methodology. Lemke and Kickert[8] then used fuzzy control for the control of a warm water plant. Since then, fuzzy control has been formalised in the literature[9,10,11,12] which introduce the theory with suitable applications[13]. The

[1] The theoretic operations on fuzzy sets form the base for the logical operators.

trend has been to apply fuzzy logic to as many applications as possible without examining what exactly is being achieved[14 et al].

Fuzzy control is enjoying a large amount of "applications pull". This means that the success of applications has led to the theory being formally developed and problems being addressed. Fuzzy control books and papers[15] are being written for people with little or no background in fuzzy logic or control. Thus the trend is to ignore what lies below the "surface" and use fuzzy control without having an understanding of the concepts. This results in a control design where the trade offs and benefits are not fully understood.

Fuzzy control is very widely used in commercial consumer products such as video cameras, motor cars, washing machines and vacuum cleaners[16]. The western countries have been slower to accept fuzzy control due to its lack of mathematical tools, but the success of these applications has ensured that it is now taken seriously. Another reason for the slow acceptance is psychological. It is felt that instead of accepting the "fuzziness" of a system, the trend should rather be to obtain a more accurate model.

1.3 Problems and myths surrounding fuzzy logic

The hype surrounding fuzzy control has resulted in numerous myths regarding its use and benefits. Some of the claims made by fuzzy logic proponents are given below :

1. Fuzzy controllers are easy to understand and design. The skill level required to design a fuzzy control system is less than that required for a classical controller[15,17,18]. Therefore personnel and training costs are reduced.

2. Fuzzy controllers are more robust than classical controllers[19].

3. Fuzzy control is more suited to non-linear processes[10 et al].

4. Development time of fuzzy controllers is less than that of classical controllers[15,17].

5. Fuzzy control improves performance[17 et al].

6. Fuzzy control reduces hardware costs[17].

There are also numerous qualitative benefits claimed about productivity which will not be addressed here.

Claim (1) is true to a certain degree. The high level programming of fuzzy control systems is easy to understand and requires less control knowledge than required by a classical control system designer. This comes at a cost however. After designing the fuzzy controller, the quality of control is based solely on the correctness of the rules and the quality of the underlying fuzzy controller. The assumptions made by the fuzzy control designer must be appropriate to the system being controlled, so that the operator rules used by the controller are appropriate. To design a controller and take into account all the factors required by a classical control designer, the required knowledge about the controlled system is the same for both approaches. If it is not, then assumptions are made about the system which may not be valid.

Claim (2) concerning robustness is more complex. Typical investigations[20] claim that their fuzzy logic controller is robust if it works when rules are either removed or tampered with. This is not a very practical definition of robustness as fuzzy controllers do not generally simply lose rules. In this investigation, the classical definition of robustness is used. This involves the controller's ability to deal with parameter variations, noise, unmodelled dynamic system characteristics and disturbances. Whether the fuzzy controller is more robust than a classical controller depends on the rules and sets defining the fuzzy system, as will be shown in Chapter 5. A fuzzy system can model any linear system to an arbitrary

degree, so Claim (2) should be altered to the fact that a fuzzy system can be as (or more) robust than a classical system. The level of robustness depends on the designer. The root of the problem is that at present it is difficult to quantify the robustness in the fuzzy control domain.

Claim (3) regards the suitability of fuzzy control to a non-linear process. The fuzzy controller is a non-linear control method, so it is obvious that it has the capability of controlling a non-linear process better than a linear controller can. The situation, as is discussed in Chapters 5 and 6, is not always about the level of non-linearity of the plant, but also about the level of information about the plant. Simply because a system is non-linear does not mean that a fuzzy controller will easily be able to control the system. Significant qualitative knowledge is required about the dynamics of the plant. The situation is further confused by claims[21] that fuzzy systems offer better response on a linear system as well. This claim is also investigated in Chapter 5.

Claim (4) : The development time of fuzzy controllers is dependant on the control problem and designer, thus Claim (4) is a gross generalisation and cannot be seen as a benefit. The development time is a function of the amount of information available, the type of information, the complexity of the problem and numerous other factors. This also defeats Claim (1).

Claim (5) : Fuzzy logic control, through its non-linear nature, has the ability to provide better performance than a linear controller. The problem is in knowing what the right choices[1] are for the fuzzy controller which will give the superior control, as the methodology is not transparent with regard to these choices.

Claim (6) : With the recent speed of technological development, fuzzy chips are becoming more freely available at low prices. Thus a dedicated fuzzy chip is much

[1] The rules, membership functions, conjunctions and implication method need to be defined in any fuzzy control system.

cheaper than a similar DSP board, which would be required if the application requires complex control actions. However, conventional PID controllers still offer the cheapest and easiest control.

The main problem regarding fuzzy control is the lack of transparent design and analysis tools. The current methods of controller design and verification are simulation and complex, but limited mathematics. The design procedure is typically iterative with stability, robustness and other aspects being investigated through simulation. This is in contrast to the well developed transparent design and analysis tools for classical control design[22].

1.4 The aim of this investigation

Section 1.3 highlighted the uncertainty and problems regarding fuzzy control design and analysis. The aim of this investigation is to :

- define the position which fuzzy control should hold given its current level of development. This includes where fuzzy control should be used and where it could be used if some problems are overcome.
- define the areas of fuzzy control which require more investigation and thus require more research.
- qualitatively and quantitatively compare fuzzy control to classical control.
- investigate how fuzzy control deals with pertinent issues present in control theory.

Prior research around this topic[23,24] has restricted itself to issues regarding the suitability of fuzzy control to certain areas in the control domain. These papers look at past applications and from these determine the areas best suited to fuzzy

control. The aim of these papers has been to define where and when to use fuzzy control on a broad scale. What is lacking is a controller comparison using specific control issues. Noise rejection and other issues are glossed over. This investigation will deal in particular with robust control and issues pertinent in classical control theory (usually dealt with superficially if at all), such as stability, disturbance rejection, noise suppression and parameter variation. Dead time and non-minimum phase plants are also investigated. At all times the investigation is intended to be fair and the information used in the control design is noted. It is essential that the most appropriate information be made available to both the fuzzy and classical controllers for the investigation to be fair. Where the information is different this is noted.

The investigation is performed through two means :

- A literature survey of past applications, with emphasis on the problems fuzzy control has been successfully applied to and the benefits which were achieved.
- Using numerous computer based models wherein particular control aspects about fuzzy control are investigated.

This investigation showed that :

- The area where fuzzy control should be used is not obvious or easily defined. The particular circumstances surrounding the control problem determine when fuzzy control is more appropriate.
- Various aspects regarding fuzzy control need to be addressed. These include mathematical design and analysis tools as well as developing a clear understanding of the implications of certain design choices.
- The non-linear nature of fuzzy control as well as the ability to include qualitative information are major advantages over classical control.

- Methods of including certain control aspects (e.g. parameter variation, noise suppression) into the fuzzy design process is not clearly defined.

1.5 Approach taken

The investigation, as outlined in section 1.4, is conducted in the following way :

Chapter 2 introduces the essential theory required for the design and analysis of fuzzy controllers. It introduces only the theory relevant for this study as well as an overview of the types of fuzzy controllers available. Chapter 3 contains a literature survey of the current applications in fuzzy control, a discussion of the relevant control aspects and the approach taken in the investigation. These two chapters are essential for the rest of the investigation to be put into perspective.

The modelling of the computer based plants is given in Chapter 4. Chapter 5 is a summary of the designed controllers and a comparison of results. These results include design and implementation issues as well as the final simulation results. Conclusions based on these results are given Chapter 6.

All simulations were performed on Simulink under MATLAB® Version 4.2. The fuzzy logic toolbox was used to implement the fuzzy controllers. The QFT designs were developed with the aid of the MATLAB® QFT toolbox. All significant code and the Simulink models are given in the appendices.

2. Fuzzy logic concepts required in control

This chapter contains an introduction to the fuzzy set and fuzzy logic theory which is required to generate a fuzzy logic controller. The content deals only with the concepts relevant to fuzzy logic as applied to the control field, and thus, is not a comprehensive guide to fuzzy systems. This section is derived from a number of sources (particularly Jager[10], Mendel[11], Jang[15] and Kosko[12]), and for a more detailed approach to fuzzy logic, the reader is referred to them. There is a large resource of literature which reviews the required theory in more detail.

Firstly, the fuzzy system is defined and explained. Fuzzy set theory is then introduced followed by the principles of fuzzy logic controller design. The structure and principle of the fuzzy controller is also dealt with. The fuzzy logic system as a universal approximator is also discussed. Finally the concepts behind adaptive fuzzy control are introduced.

2.1 Definition of a Fuzzy Logic System

The term fuzzy is derived from the fact that membership to a set (e.g. tall people) is not crisp, but graded (e.g. 20% membership to the set tall). Thus the set "tall" is a 'fuzzy' concept with people of various heights being members to various degrees. Classical set theory calls for a bivalent[1] approach where membership is either true or false. Fuzzy set theory allows multivalence and hence various degrees of membership are possible[25].

A fuzzy logic system is a non-linear mapping of an input vector into an output vector. Fuzzy controllers are represented by *if-then* rules and can thus provide a user-friendly method of interfacing with the controller. The rules are fired[2] in parallel (or in an arbitrary order when programmed[3]), each to a certain degree[4]. The result of all the rules are aggregated to a final crisp value which forms the output of the fuzzy system[5]. The rules deal with linguistic variables which are defined (through membership functions) in the universe of discourse (see section 2.3 for definitions).

Fuzzy sets have come under a lot of criticism for being "probability in disguise". It is the author's opinion that this is not true. There is a similarity in the way it is termed, but the meaning is different. Probability still remains based on a bivalent system, where membership is definite. Fuzzy membership is multivalent and therefore membership is definite, but to varying degrees. For example, a 1.5 metre man could be seen as belonging 60% to the set "tall" and 40% to the set "not tall". To say that he has 60 % chance of being "tall" means that he is either tall or he is not, but he has more chance of being tall. The man cannot be a member of "tall" and "not tall" as is the case with fuzzy logic.

2.2 Rationale for Fuzzy Logic

The rationale used to justify the study of fuzzy logic in engineering is given by Lofti Zadeh (Principle of Incompatibility)[6] : "As the complexity of a system

[1] By bivalence, it is meant that the set membership can only take on two values, namely true (1) or false (0).

[2] By fired, it is meant that the rule antecedent is met. In a fuzzy sense, the rule antecedent is always met, but often this is to a zero degree, as the required membership is zero.

[3] The actual order is not important as the results are aggregated afterwards.

[4] The degree to which the rule is fired depends on the applicability of the antecedent. If the membership of the fuzzy set called in the antecedent is low, the degree of firing is low.

[5] This need not be true, but is true for a fuzzy controller.

increases, our ability to make precise and yet significant statements about its behaviour diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics." The ability to allow for a less precise definition of certain properties which are being controlled or analysed, allows a method of incorporating real-world "fuzziness" or uncertainty into a designed controller.

2.3 Fuzzy Set Theory

2.3.1 Fuzzy Sets

Zadeh[3] (1965) introduced fuzzy set theory in his paper "Fuzzy Sets", but others such as Lukasiewicz (who introduced multivalued logic) and Max Black[26] (who called it vagueness) had introduced the ideas earlier.

A fuzzy set (A) is a set with graded membership defined over a universe of discourse and is characterised by a membership function $\mu_A(x) \in [0,1]$. The universe of discourse is the input space for a specific fuzzy system input. A fuzzy set A, in the universe of discourse X, is denoted by :

$$\begin{aligned} A &= \sum_{i=1}^{n} \mu_A(x_i) / x_i \\ &= \mu_A(x_1) / x_1 + \ldots + \mu_A(x_n) / x_n \end{aligned} \qquad (2\text{-}1)$$

for a discrete X. Thus for each value of x on the universe of discourse X, $\mu_A(x)$ defines the membership of x to the set A. The summation sign does not denote actual summation, but rather the collection of the discrete points. Similarly, the forward slash does not denote division, but rather the association of the fuzzy set membership values with the input space.

For a non-finite (continuous) X, the equation can be written as :

$$A = \int_X \mu_A(x) / x \qquad (2\text{-}2)$$

where the integral sign does not denote integration, but rather the continuous collection of defined points.

Example 2-1 : Consider the fuzzy set *tall* with a discrete universe, $x_i \in$[1.5m, 2.1m]:

$$\text{"}tall\text{"} = (0/1.5 + 0.2/1.6 + 0.4/1.7 + 0.6/1.8 + 0.8/1.9 + 1/2.0 + 1/2.1)$$

This can be represented graphically (continuous approximation) by figure 2.1.

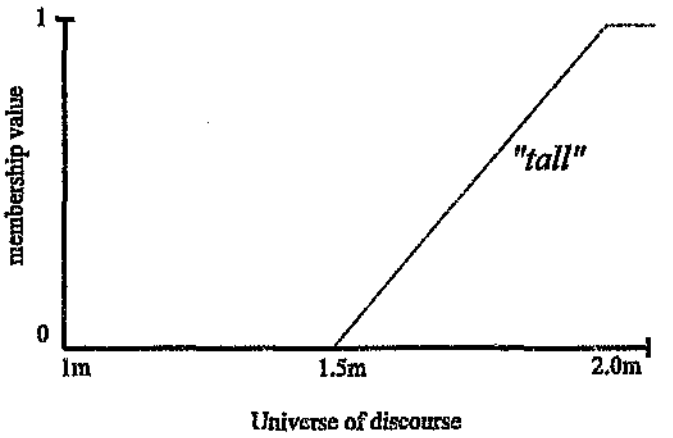


Figure 2-1 : Fuzzy set *tall*

2.3.2 Properties of Fuzzy Sets

Definition 1 : The height of a fuzzy set A is defined by :

$$\text{hgt}(A) = \text{height}(A) = \sup_{x \in X} \mu_A(x) \qquad (2\text{-}3)$$

A normal set is one which has a height of 1. Any fuzzy set can be converted into a normal set through division by the supremum of the set. The height corresponds to the maximum membership that the set allows.

Definition 2 : The core of the set is the part of the set for which the membership is unity :

$$\text{core}(A) = \{x \in X | \mu_A = 1\} \tag{2-4}$$

This is the range on the universe of discourse for which a fuzzy set gives complete membership.

Definition 3 : The support of the set, is the part of the set for which the membership is greater than 0 :

$$\text{supp}(A) = \{x \in X | \mu_A(x) > 0\} \tag{2-5}$$

If the support of a set is a single point, the set is known as a fuzzy singleton.

Definition 4 : The *α-cut* of a fuzzy set (also termed *level set*) is defined by :

$$\alpha - \text{cut}(A) = \left\{x \in X | \mu_A(x) \geq \alpha\right\} \tag{2-6}$$

The point where the membership is 0.5 is termed the *crossover point.*

There are more properties of fuzzy sets which are not discussed here as they are not necessary for the design of controllers.

2.3.3 Linguistic Variables

Zadeh[27] explains the linguistic variable in the following way : "In retreating from precision in the face of overpowering complexity, it is natural to explore the use of what might be called linguistic variables, that is, variables whose values are not numbers but words or sentences in a natural or artificial language". His rationale for using these variables is expressed by : "The motivation of the use of words or sentences rather than numbers is that linguistic characterisations are, in general, less specific than numerical ones."

Let u denote the name of a linguistic variable[1] (e.g. height) with numerical values denoted by x, where $x \in X$. The linguistic variable is usually decomposed into a set of terms, $T(u)$, which cover the universe of discourse. These terms can be characterised as fuzzy sets with membership functions.

Example 2-2 : If *height*(x) is a linguistic variable, then it can be decomposed into the following $T(height) = \{short, average, tall\}$ where the universe of discourse could span X=[1m, 2.1m]. Each term is characterised by a membership function indicating the degree of membership for any value of x within the universe of discourse X. Following on example 2-1, the following terms (which form the linguistic variable), can be defined :

[1] Sometimes x and u are used interchangeably, especially when the linguistic variable is a letter.

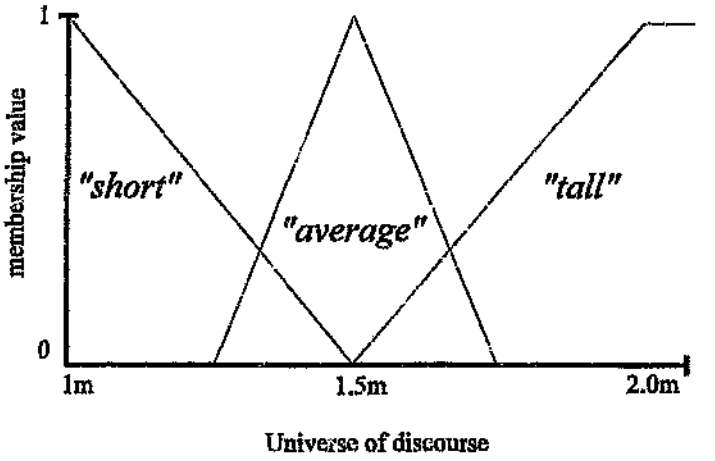


Figure 2-2 : Linguistic variable *height*

2.3.4 Fuzzy Numbers

A fuzzy number is a special form of fuzzy set. It is useful when uncertainty is present regarding the actual value of a number. A fuzzy number must meet the following criteria :

- the fuzzy set is convex
- the fuzzy set is normalised i.e. $height(F)=1$
- the membership function of the fuzzy set is piecewise continuous
- the core of the fuzzy set consists of one value only

Example 2-3 : A fuzzy number *"approx. 1.6m tall"* may be a triangle from 1.4m to 1.8m with vertex at 1.6m :

$$"approx.\ 1.6m\ tall" = (0/1.4 + 0.5/1.5 + 1/1.6 + 0.5/1.7 + 0/1.8)$$

A fuzzy interval must meet the same conditions for a fuzzy number but the core is not restricted to one number (e.g. "*from about 1.2m to about 1.4m tall*"). Operations can be performed on fuzzy numbers (e.g. addition, subtraction etc.) using the extension principle[6].

2.3.5 Membership Functions

A membership function is a curve defining the degree of membership for each point on the universe of discourse. The membership must be between 0 and 1, but does not necessarily have to include 0 and 1. The shape of the membership function can be arbitrarily set, based on the designer's discretion, or, as is currently the trend, the shape can be designed through optimisation procedures (like neural networks[10,15]).

Example 2-4 : The linguistic variable *height* contains the membership function for *tall*. The definition of tall people is not a universal one, and hence the shape will vary depending on the designer's perception of tall people. An infinite number of possibilities can be designed and it is objectively impossible to say which designer is most correct in his interpretation of tall.

The greater the number of membership functions (hence fuzzy sets) on the universe of discourse, the greater the resolution that can be achieved. The higher the density of the fuzzy sets on a certain part of the universe of discourse, the more complex the controller output as a function of the controller input can be defined[10].

Table 2-1 is a summary of the most common membership function shapes.

Table 2-1 : Summary of common membership functions

Shape	Equation
Triangular	$f(x,a,b,c)=\begin{cases}0, x\le a\\ \frac{x-a}{b-a}, a\le x\le b\\ \frac{c-x}{c-b}, b\le x\le c\\ 0, c\le x\end{cases}$
Gaussian	$f(x;\sigma,c)=e^{\frac{-(x-c)^2}{2\sigma^2}}$
Bell Curve	$f(x;a,b,c)=\frac{1}{1+\left\lvert\frac{x-c}{a}\right\rvert^{2b}}$
Trapezoidal	$f(x,a,b,c,d)=\begin{cases}0, x\le a\\ \frac{x-a}{b-a}, a\le x\le b\\ 1, b\le x\le c\\ \frac{d-x}{d-c}, c\le x\le d\\ 0, d\le x\end{cases}$

2.3.6 Operations on Fuzzy Sets

As with crisp sets, intersection, union and complement are defined in fuzzy set theory, but as the membership values are no longer restricted to {0,1}, but rather the interval [0,1], the operators are not uniquely defined.

2.3.6.1 Union and Intersection

Let fuzzy sets A and B have membership functions μ_A and μ_B respectively. Zadeh[3] defined the following :

$$\begin{aligned} \mu_{A\cap B} &= \min(\mu_A(x), \mu_B(x)) \quad \text{(intersection)} \\ \mu_{A\cup B} &= \max(\mu_A(x), \mu_B(x)) \quad \text{(union)} \end{aligned} \tag{2-7}$$

If A and B are crisp sets, the definitions agree with the crisp set definitions. This is because the intersection and union criteria (discussed below) which must be met are the characteristic functions found in crisp set theory. The following definitions have also been proposed[1] :

$$\begin{aligned} \mu_{A\cap B} &= \mu_A(x)\mu_B(x) \quad \text{(intersection)} \\ \mu_{A\cup B} &= \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x) \quad \text{(union)} \end{aligned} \tag{2-8}$$

There are, however, an infinite number of algorithms which implement fuzzy intersection and union. Thus the general forms, T-norm (triangular norm) and T-conorm or S-norms (triangular conorms) are used respectively.

The T-norm (intersection) is a two place function $T : [0,1] \times [0,1] \rightarrow [0,1]$ which satisfies :

- boundary : $T(0,a) = T(a,0) = 0$, $T(a,1) = T(1,a) = a$

[1] Based on the definitions from probability.

- monotonicity : $T(a,b) \leq T(c,d)$ if $a \leq c$ and $b \leq d$
- commutativity : $T(a,b) = T(b,a)$
- associativity : $T(a, T(b,c)) = T(T(a,b), c)$

Many symbols are used for the T-norm, but the most popular is *. The convention used from this point on will be a capital T.

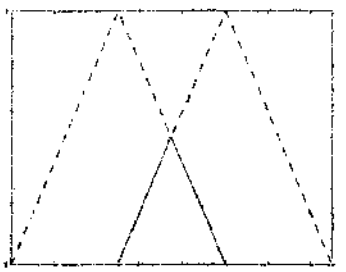
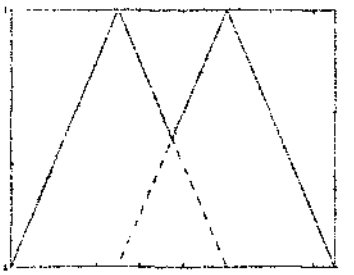
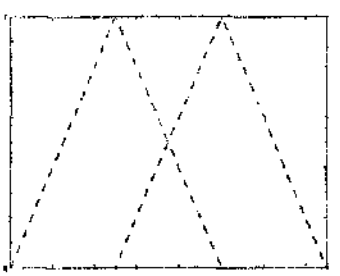
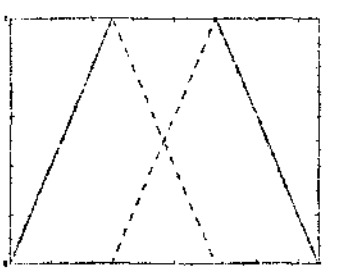
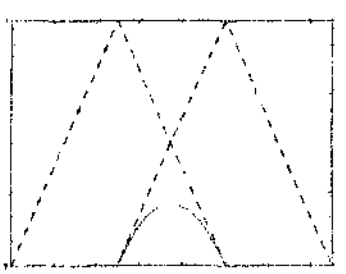
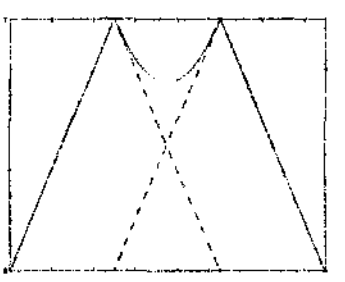
The rules for T-conorm are similar. The T-conorm or S-norm (union) is a two place function $S : [0,1] \times [0,1] \rightarrow [0,1]$ which satisfies :

- boundary : $S(0,a) = S(a,0) = a$, $S(a,1) = S(1,a) = 1$
- monotonicity : $S(a,b) \leq S(c,d)$ if $a \leq c$ and $b \leq d$
- commutativity : $S(a,b) = S(b,a)$
- associativity : $S(a, S(b,c)) = S(S(a,b), c)$

The fuzzy system designers can therefore design their own T-norms and S-norms should it be necessary. All T-norms and T-conorms will produce the same results when used with crisp set theory, but different results with fuzzy set theory.

Example 2-5 : Table 2-2 shows the results of different T-norms and T-conorms.

Table 2-2 : Effect of choosing different connectives

T-norm	T-conorm
Zadeh : $\min(\mu_A,\mu_B)$ 	Zadeh : $\max(\mu_A,\mu_B)$ 
Lukasiewicz : $\max(\mu_A+\mu_B-1,0)$ 	Lukasiewicz : $\min(\mu_A+\mu_B,1)$ 
probability : $\mu_A\mu_B$ 	probability : $\mu_A+\mu_B-\mu_A\mu_B$ 

2.3.6.2 Complement

Zadeh[3] proposed the following membership function for fuzzy complement :

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \qquad (2\text{-}9)$$

This definition is again, not unique. The requirements for the complement of a fuzzy set A are :

- $c(0) = 1$
- $c(a) < c(b)$, if $a > b$
- $c(c(a)) = a$

There are other definitions of fuzzy complement which meet the above requirements including the λ-complement according to Sugeno[28].

From the above definitions, it can be seen that for fuzzy sets, the crisp set Laws of Contradiction ($A \cup \bar{A} = U$) and Excluded Middle ($A \cap \bar{A} = \phi$) are broken. The implication of this is that it challenges the basis of Aristotlian binary logic. A cup can be both half full and half empty (or not half full) with fuzzy set theory, where crisp theory requires that it be either full or empty.

2.3.7 Fuzzy Relations and Compositions

2.3.7.1 Fuzzy Relations

The fuzzy sets discussed before have only had membership functic of one variable. They can be extended to more than one variable and are then termed fuzzy relations. The fuzzy relation represents a degree of presence or absence of association, interaction or interconnectedness between the elements of two or more fuzzy sets[11]. Given the fuzzy relation $R : X_1 \times ... \times X_n \rightarrow [0,1]$ then

$$R = \left\{ \mu_R(x_1,...,x_n) / (x_1,...,x_n) \middle| x_1 \in X_1,...,x_n \in X_n \right\} \qquad (2\text{-}10)$$

Example 2-6 : The fuzzy relation for approximately equal $R_{heights}(x, y)$, can be described by:

$$\begin{aligned} R_{heights}(x, y) = {} & 1/(1,1) + 0.5/(1,1.1) + 0.5/(1.1,1) + 1/(1.1,1.1) \\ & + 0.5/(1.1,1.2) + 0.5/(1.2,1.1) + 1/(1.2,1.2) \\ & + ... + 0.5/(2.4,2.5) + 0.5/(2.5,2.4) + 1/(2.5,2.5) \end{aligned}$$

for the space X=[1m,2.5m] & Y=[1m,2.5m]. This relation gives an indication of whether two people's heights, x and y, can be considered equal or not. For example if one person is 2m tall and the other is 2.1m, we can say that they are approximately equal to degree 0.5.

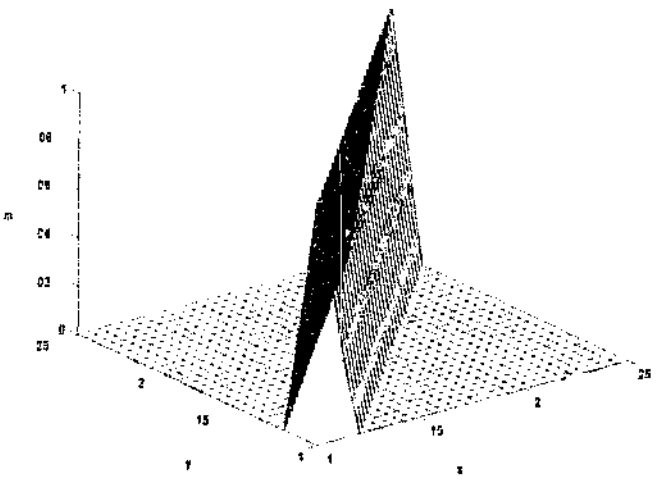


Figure 2-3 : Fuzzy relation $R_{height}(x, y)$

2.3.7.2 Fuzzy Compositions

Composition involves taking two relations from different product spaces and combining them to form another relation which relates the elements of the new product space. For the general case[1], the composition is denoted by :

$$R(U,W) = P(U,V) \circ Q(V,W) \tag{2-11}$$

where $\circ$ is the composition operator.

The fuzzy composition is defined by Zadeh[6], "suppose there exists a fuzzy relation R in $X \times Y$ and A is a fuzzy set in X, then a fuzzy subset B of Y can be induced by A, given the composition of R and A". This is denoted by

$$B = A \circ R \tag{2-12}$$

[1] For both crisp and fuzzy set theory.

Therefore if a relation R relating B and A is known, the given A (or B), B (or A) can be inferred. The sup-min composition, proposed by Zadeh, leads to the following implementation. If A is a fuzzy set with membership function $\mu_A(x)$ and R is a fuzzy relation with membership function $\mu_R(\ ;y)$:

$$\mu_B(y) = \sup_{x \in X} \min(\mu_A(x), \mu_R(x,y)) \qquad (2\text{-}13)$$

A generalisation of (2-12) using T-norms and T-conorms is :

$$\mu_B(y) = S[T(\mu_A(x), \mu_R(x,y))] \qquad (2\text{-}14)$$

Composition is the procedure that allows a fuzzy model to produce sensible outputs for previously unseen inputs, provided the fuzzy rule base and input sets are appropriate.

Example 2-7 : (adapted from Jager[10]) Following on example 2-6, where $R_{height}(x, y)$ is defined, and example 2-3, where $A_{\approx 1.6m}$ is defined, below is a discrete composition.

$$\mu_B(y) = \overbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.5 \\ 1 \\ 0.5 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}^{\mu_{\approx 1.80}(x)} \circ \overbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 1 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 1 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 1 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 1 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 1 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 1 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 1 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 1 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 1 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 1 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}^{\mu_{height \approx}(x,y)}$$

$$= \overbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 1 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}^{\min(\mu_{\approx}(x), \mu_{height \approx}(x,y))}$$

$$= \overbrace{\begin{pmatrix} 0 & 0 & 0 & 0.5 & 0.5 & 1 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \end{pmatrix}}^{\underset{x}{\text{Max}}\,\text{Min}(\mu_{\approx 1.80}(x), \mu_{\approx}(x,y))}$$

2.4 Fuzzy Logic Systems

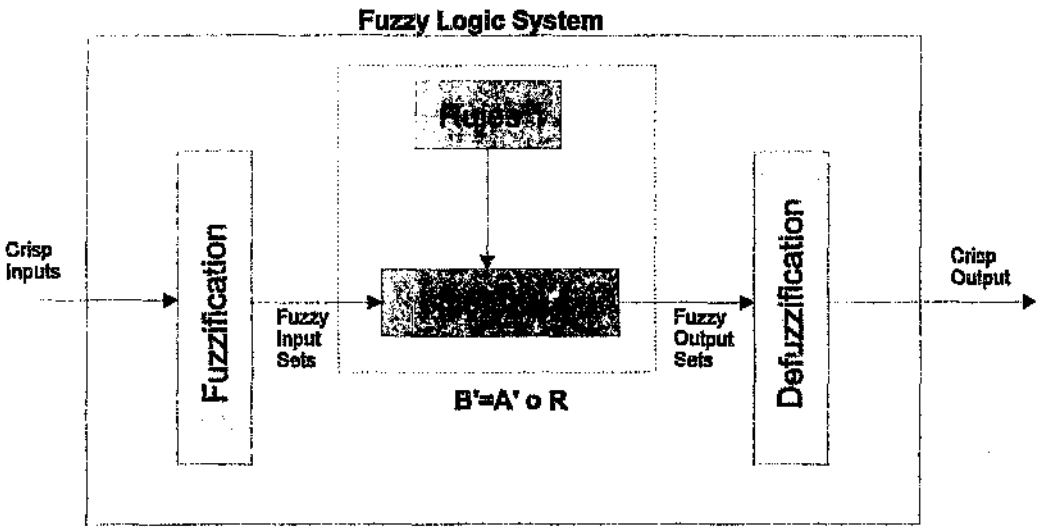


Figure 2-4: Elements of a fuzzy logic system

The structure of the most widely used fuzzy controllers is depicted in Figure 2-4. It consists of four major components, namely the rules, fuzzifier, inference engine and defuzzifier. Once the controller has been designed, the fuzzy system maps a crisp value on the input into a crisp value on the output i.e. $y=f(x)$, where f is some (usually non-linear) function.

Note : From this point forth, any symbol with a "dash" represents a measured variable (or derived from measurement). E.g. x' represents a measured value on the universe of discourse of X.

2.4.1 Input Fuzzification

The fuzzification stage of the fuzzy controller involves the construction of the fuzzy input relation. Fuzzification is not always necessary if the input is already a fuzzy relation. If the input to the controller is a numerical value (most common case), then the fuzzy input set A_i' is given by the singleton,

$$\mu_{A_i'}(x_i) = \begin{cases} 1, \text{ if } x_i = x_i' \\ 0, \text{ otherwise} \end{cases}$$

Some sources[15] include the matching of fuzzy propositions into the fuzzification stage.

2.4.2 The Rules

2.4.2.1 Fuzzy Propositions

A fuzzy proposition is a statement such as "x is large", where in this case "large" is a linguistic term referring to some fuzzy set on the universe of discourse of variable x. Logical connectives (*and* & *or*) are implemented using the T-norms and T-conorms.

Example 2-8 : x_1 is large and x_2 is medium $\equiv \mu_p(x_1,x_2) = T(\mu_{large}(x_1), \mu_{medium}(x_2))$.

The most common connectives which are used are summarised in Table 2-3 :

Table 2-3 : : Frequently used operator for and & or connectives

and	or	Source
$\min(\mu_A,\mu_B)$	$\max(\mu_A,\mu_B)$	Zadeh
$\max(\mu_A+\mu_B-1,0)$	$\min(\mu_A+\mu_B,1)$	Lukasiewicz
$\mu_A\mu_B$	$\mu_A+\mu_B-\mu_A\mu_B$	probability

Negation is related to the complement of the fuzzy set and may also be used in propositions e.g. x_1 is not large. As explained earlier, there are numerous possibilities for defining complement in fuzzy set theory, but the most commonly used one is (as defined by Zadeh[3]) :

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \qquad (2\text{-}15)$$

2.4.2.2 Fuzzy Rules

A fuzzy rule takes the form of an *if-then* statement, with the antecedents (*if* part) and consequents (*then* part) consisting of fuzzy propositions. Fuzzy connectives and negations are all possible in the rule.

Example : The fuzzy rule r (A_1, A_2 and B are vague linguistic terms) :

r : if x_1 is A_1 and x_2 is not A_2 then y is B

can be written as,

$$\mu_R(x_1, x_2, y) = \mathrm{I}(\mathrm{T}(\mu_{A_1}(x_1), 1 - \mu_{A_2}(x_2)), \mu_B(y))$$

where I is a fuzzy implication function (see section 2.4.3.1) and T is a conjunction based on the T-norm.

2.4.3 Fuzzy Inference Scheme

Note : The fuzzy system discussed here uses local inference[1] as it is more commonly used in fuzzy control applications[10].

2.4.3.1 Inference of a single rule

The inference of a single rule is the application of the composition of fuzzy relations. The compositional rule of inference was introduced by Zadeh[9]. It assumes that a fuzzy rule

if x is A then y is B

is represented by a fuzzy relation R. The result B' can then be inferred through the composition of A' and R :

$$B' = A' \circ R \qquad (2\text{-}16)$$

[1] Local inference (rule-based approach) means that each rule is inferred and the results of the inferences of the individual rules are aggregated afterwards. Global inference (relation based approach) means that the rules are aggregated and used for inference as a whole. This makes no difference if the inference with T-implications is used or the inputs are numerical.

Like the other operators in fuzzy set theory, the number of possible implication methods is infinite. The implication method chosen for control is typically the T-implication (based on the T-norm).

$$\mathrm{I}(a,b) = \mathrm{T}(a,b) \tag{2-17}$$

This complies with the classical implication as expected.

2.4.3.2 Inference of a rule base

The fuzzy rule base consists of rules of the form,

r_1 : if x_1 is $A_{1,1}$ and ... and x_{N_x} is $A_{N_x,1}$ then y is B_1
else
...
else
r_k : if x_1 is $A_{1,k}$ and ... and x_{N_x} is $A_{N_x,k}$ then y is B_k
else
...
else
r_{N_r} : if x_1 is A_{1,N_r} and ... and x_{N_x} is A_{N_x,N_r} then y is B_{N_r}

where N_r is the number of parallel rules which have an antecedent based on N_x variables.

The fuzzy inference procedure can be divided into the following sections,

1. The numerical values obtained from the sensors (input to the controller) need to be matched against the fuzzy rules. Thus, the numerical data x_i', is matched against the fuzzy propositions "x_i is $A_{i,k}$" in the antecedents of the fuzzy rules r_k:

$$\alpha_{i,k} = \mu_{A_{i,k}}(x_i') \tag{2-18}$$

where $\alpha_{i,k}$ is a numerical value indicating the extent of the matching.

If the controller input contains uncertainty or inaccuracy, the input can be modelled as a fuzzy number, and not a crisp number. In this case, the extent of matching is,

$$\alpha_{i,k} = \text{hgt}(A_i' \cap A_{i,k}) \tag{2-19}$$

where A_i' is the fuzzy number and *hgt* is the height of the fuzzy set.

2. Most rules will have more than one proposition, and therefore it is necessary to determine the overall degree of fulfilment β_k (DOF) for each rule in the rule base. A different method for each connective (*and* & *or*) is needed.

$$\text{AND} \quad : \beta_k = \prod_{i=1}^{N_X} \alpha_{i,k} \tag{2-20a}$$

$$\text{OR} \quad : \beta_k = \sum_{i=1}^{N_X} \alpha_{i,k} \tag{2-20b}$$

Often rules may have both connectives. If this is the case, a combination of the two methods is required.

3. The degree of fulfilment for each rule has now been determined. Thus, the implication can be made,

$$\mu_{B_k'}(y) = \text{I}(\beta_k, \mu_{B_k}(y)) \tag{2-21}$$

where I is the implication method used. The most commonly used method of inference in fuzzy control is the inference with T-implications (based on the T-norm) :

$$\mu_{B_k'}(y) = \mathrm{T}(\beta_k, \mu_{B_k}(y)) \qquad (2\text{-}22)$$

The result is that each consequent B_k is restricted by the β_k value by means of the T-norm, which represents the implication. Therefore, the rule is only fired to the degree that its antecedents are fulfilled.

4. After the third step, the result is N_r fuzzy output sets defined by the membership functions $\mu_{B_k'}$. The sets must be aggregated to achieve a final, single fuzzy set. There are different aggregation methods depending on the method of implication. The method used when using the T-implication is :

$$\mu_{B'}(y) = \bigcup_k \mu_{B_k'}(y) \qquad (2\text{-}23)$$

where $\cup$ is the union operator (or). If the *max* T-conorm is used, it results in a final fuzzy set which is the maximum of all the individual fuzzy resultant sets. Another popular method of aggregation is the summing method, where the individual sets are added together.

Weighting of rules is also possible. If every rule is given a weight $w_k \in [0,1]$, it becomes possible to ensure that some rules have more influence in the inference process. This is useful when a form of learning is being implemented.

If the *max-min* inference method is used[1] (most common method), then the process can be summarised by :

$$\mu_{B'}(y) = \overbrace{\max_k}^{\text{aggregation}} \underbrace{\min}_{\text{implication}} \left(\beta_k, \mu_{B_k}(y)\right) \qquad (2\text{-}24a)$$

where,

[1] *Max-prod* and *sum-prod* are also fairly common.

$$\alpha_{i,k} = \overbrace{\overbrace{\sup}^{\text{projection}} \underbrace{\min}_{\text{combination}}}^{\text{composition}} (\mu_{A_i'}(x_i), \mu_{A_i,k}(x_i)) \qquad (2\text{-}24b)$$

$$\beta_k = \overbrace{\min_i}^{\text{conjunction}} \alpha_{i,k} \qquad (2\text{-}24c)$$

Example 2-10 : Figure 2-5 illustrates a *fuzzy inference diagram* for a fuzzy system. Input A_1' is a fuzzy number (crisp number x_1 converted to fuzzy number) while A_2' is a crisp input (x_2 converted into a fuzzy singleton). For the example, the *min* operator (T-norm) is used for both the *and*-connective and implication. Sup-min[1] (sometimes called max-min) composition is used for the composition rule of inference. The rules are :

r_1 : if x_1 is $A_{1,1}$ and x_2 is $A_{2,1}$ then y is B_1

r_2 : if x_1 is $A_{1,2}$ and x_2 is $A_{2,2}$ then y is B_2

$$\alpha_{1,1} = \sup \ \min(\mu_{A_1'}(x'_1), \mu_{A_{1,1}}(x'_1))$$
$$\alpha_{1,2} = \sup \ \min(\mu_{A_1'}(x'_1), \mu_{A_{1,2}}(x'_1))$$
$$\alpha_{2,1} = \mu_{A_{2,1}}(x'_2)$$
$$\alpha_{2,2} = \mu_{A_{2,2}}(x'_2)$$
$$\beta_1 = \min(\alpha_{1,1}, \alpha_{2,1}) = \alpha_{2,1}$$
$$\beta_2 = \min(\alpha_{1,2}, \alpha_{2,2}) = \alpha_{1,2}$$
$$\mu_{B'(y)} = \max\left[\min(\beta_1, \mu_{B_1}(y)), \min(\beta_2, \mu_{B_2}(y))\right]$$

[1] This means that the aggregation is *max* and the implication method is *min*.

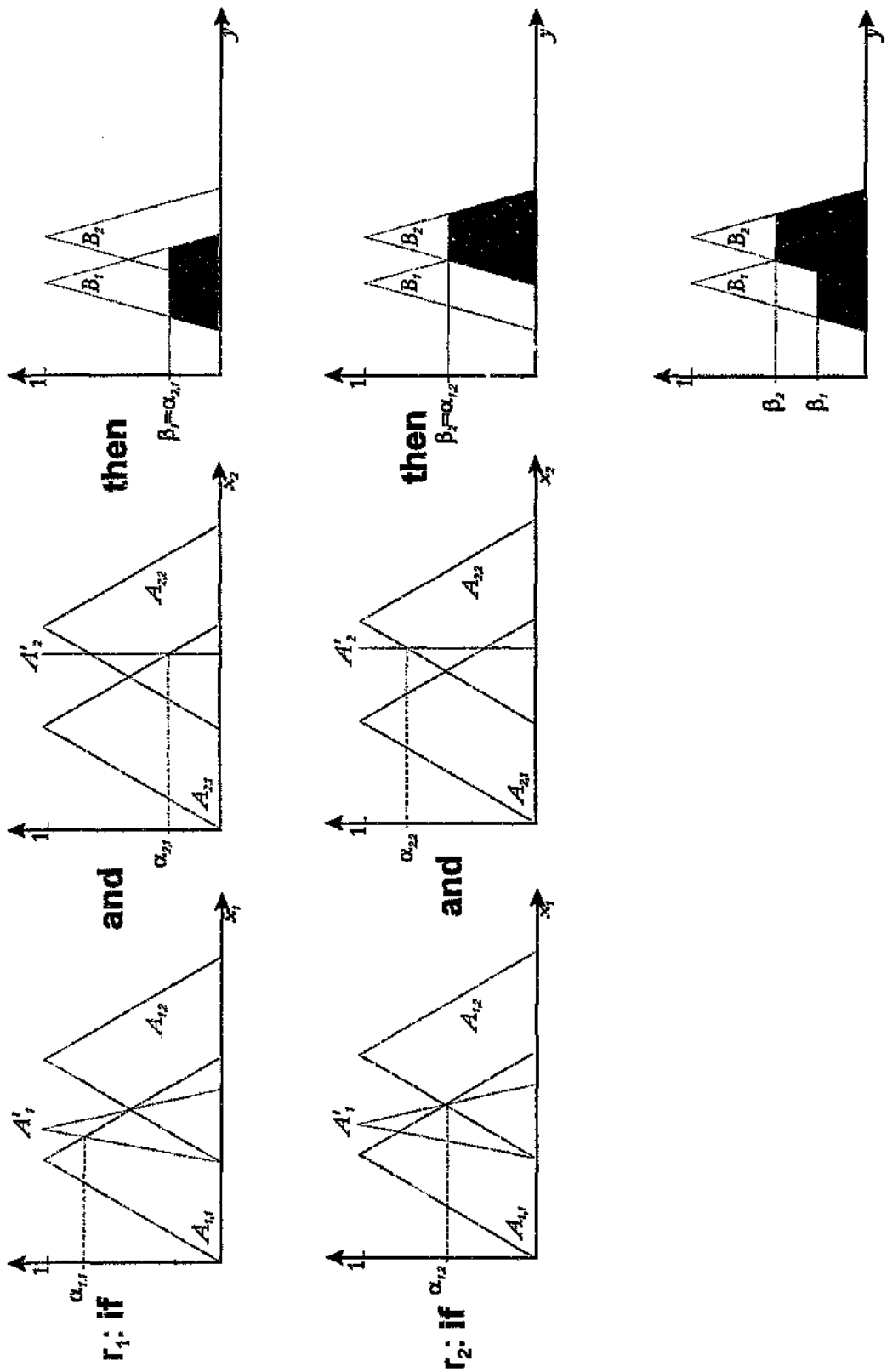


Figure 2-5 : Fuzzy inference diagram for example 2-10

2.4.4 Defuzzification

The defuzzification stage translates the final fuzzy s.. obtained after aggregation into a numerical crisp value which forms the output of the fuzzy controller. The method of defuzzification is not unique or bound by any specific criterion. In control applications, however, computational simplicity is desirable. Below is a summary of the most popular methods of defuzzification :

2.4.4.1 Maximum Defuzzifier

This defuzzifier takes the maximum of the output set B' as its final output value. This is not a good method as it ignores the set support distribution.

2.4.4.2 Mean of Maxima Defuzzifier

As the name implies, this method determines the values of y where a maximum in B' occurs and computes the mean of these y values.

2.4.4.3 Centroid or Centre-of-Gravity Defuzzifier

This method is based on the principle used to calculate the centre of gravity of a mass, except that the point masses are replaced by the membership function. In a

one dimensional fuzzy set, this is often called the centre-of-area defuzzification. The centre of gravity (COG) defuzzification method is defined by :

$$\bar{y} = \frac{\int_y \mu_{B'}(y).y\,dy}{\int_y \mu_{B'}(y)\,dy} \tag{2-25}$$

and the discrete form is

$$\bar{y} = \frac{\sum_{q=1}^{N_q} \mu_{B'}(y_q).y_q}{\sum_{q=1}^{N_q} \mu_{B'}(y_q)} \tag{2-26}$$

where N_q is the number of quantizations in the membership function $\mu_{B'}$. This is the most popular method, but is, however, not computationally efficient.

2.4.4.4 Singletons

While not a defuzzification method, having singletons on the output simplifies the defuzzification process significantly. These systems are often termed Sugeno-style fuzzy systems after their developer[13]. The consequent of the rules in this system are crisp values (or singletons). The implication method is multiplication and aggregation is the maximum of the output fuzzy sets. The deffuzification is then a weighted average of these singletons, (i.e. the height of each singleton determines the weight given to the corresponding y value).

The Sugeno-style fuzzy system can be generalised as,

$$\begin{aligned} r_k \quad : \quad & \text{if } x_1 \text{ is } A_{1,k} \text{ and } \ldots \text{ and } x_{N_x} \text{ is } A_{N_x,k} \\ & \text{then } y_1 = f_{1,k}(x_1, \ldots, x_{N_x}), \ldots, y_{N_y} = f_{N_y,k}(x_1, \ldots, x_{N_x}) \end{aligned} \tag{2-27}$$

The consequents of these rules are functions of the controller input x_i. Normally, the output function is linear and only one output is considered :

$$r_k \quad : \quad \text{if } x_1 \text{ is } A_{1,k} \text{ and } \ldots \text{ and } x_{N_X} \text{ is } A_{N_X,k} \quad \text{then } y = b_{0,k} + \sum_{i=1}^{N_X} b_{i,k} x_i \tag{2-28}$$

where $b_{i,k}$ and $b_{0,k}$ are constants. I.e. y is a linear function of the inputs x_i. A set of Sugeno rules can be seen as a set of local controllers. The Seguno style fuzzy system can also be viewed as a fuzzy supervisor which changes the parameters of a linear controller. This style of fuzzy system is ideal for use in adaptive systems where the constants can easily be tuned.

Example 2-11 · Figure 2-6 illustrates a *fuzzy inference diagram* for a Sugeno fuzzy system. The Sugeno fuzzy system illustrated is of *zero* order. Input A_1' is a fuzzy number (crisp number x_1 converted to fuzzy number) while A_2' is a crisp input (x_2 converted into a fuzzy singleton). For the example, the *min* operator (T-norm) is used for both the *and*-connective and implication. Sup-min (sometimes called max-min) composition is used for the composition rule of inference. The rules are :

r_1 : if x_1 is $A_{1,1}$ and x_2 is $A_{2,1}$ then y is $B_{0,1}$

r_2 : if x_1 is $A_{1,2}$ and x_2 is $A_{2,2}$ then y is $B_{0,2}$

w' ere $B_{0,1}$ and $B_{0,2}$ are constants. The output y is the weighted sum of the individual singletons.

$$\alpha_{1,1} = \sup \ \min(\mu_{A_1}(x'_1), \mu_{A_{1,1}}(x'_1))$$

$$\alpha_{1,2} = \sup \ \min(\mu_{A_1}(x'_1), \mu_{A_{1,2}}(x'_1))$$

$$\alpha_{2,1} = \mu_{A_{2,1}}(x'_2)$$

$$\alpha_{2,2} = \mu_{A_{2,2}}(x'_2)$$

$$\beta_1 = \min(\alpha_{1,1}, \alpha_{2,1}) = \alpha_{2,1}$$

$$\beta_2 = \min(\alpha_{1,2}, \alpha_{2,2}) = \alpha_{1,2}$$

$$y = \frac{\beta_1 . b_{0,1} + \beta_2 . b_{0,2}}{\beta_1 + \beta_2}$$

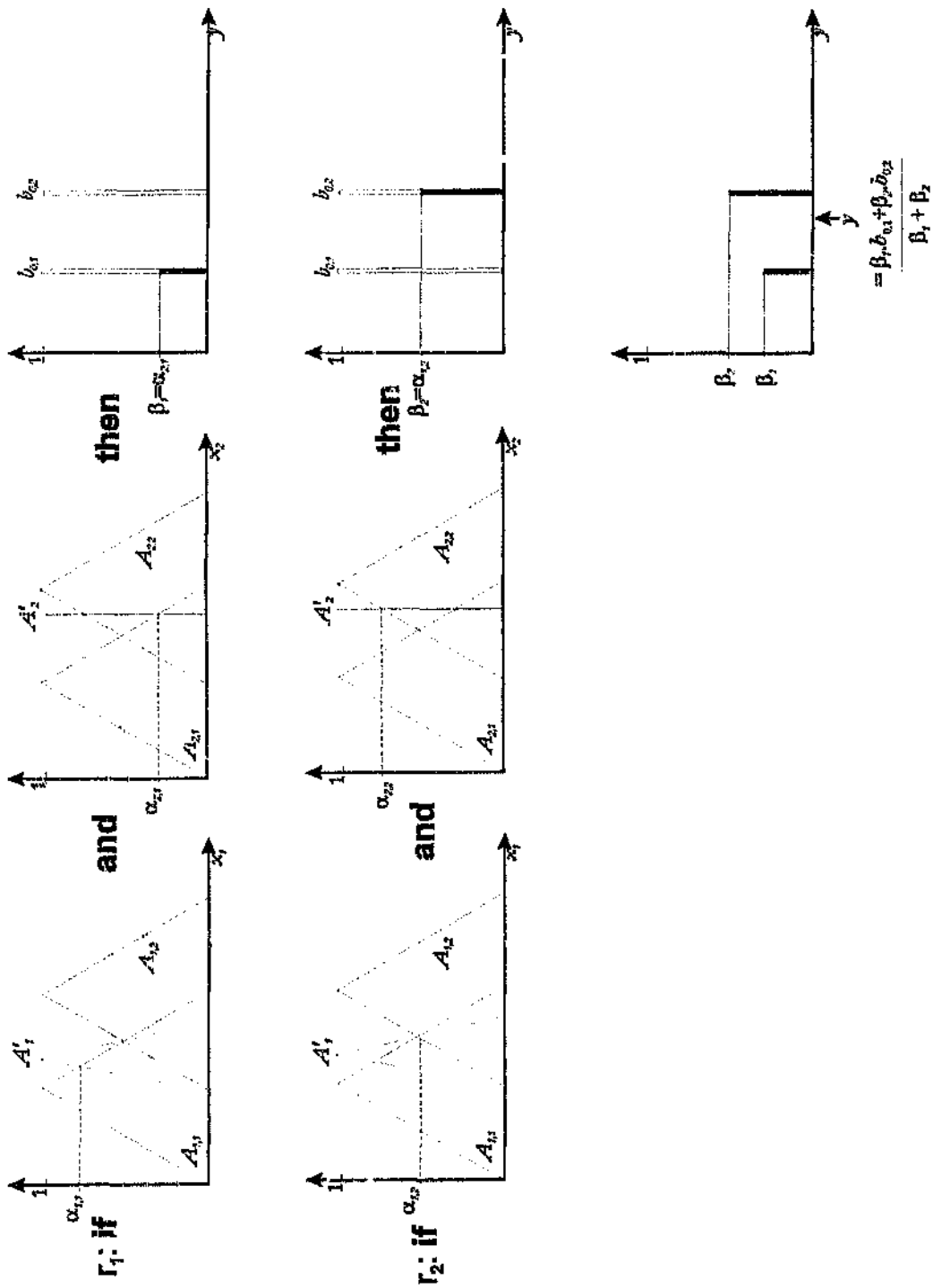


Figure 2-6 : Fuzzy inference diagram for example 2-11

2.5 Fuzzy Systems as Universal Approximators

A universal approximator theorem is an existence theorem. A universal approximator is a system which can uniformly approximate any real continuous non-linear function to an arbitrary degree of accuracy.

The fuzzy logic system has been proved to be a universal approximator, but only for specific systems. Wang and Mendel[29] have proved this true for a singleton fuzzy system which has product inference, product defuzzification, Gaussian membership functions and height defuzzification. Kosko[30] has proved this true (using fuzzy patches) for an additive fuzzy system which uses singleton fuzzification, centroid defuzzification, product inference and product implication. Thus a theorem is required which will prove that any arbitrary FLS is an universal approximator.

Although a fuzzy system may be able to approximate any real system, the theorem does not give any indication on how to specify the fuzzy system. All it indicates is that with enough fuzzy sets and rules, any system can be modelled[1]. The combinations possible in a fuzzy control system are infinite (due to the ability to define one's own T-norms, T-conorms, negation, inference, fuzzy set shapes and defuzzification methods). Therefore in any control application, there are an infinite number of possible controllers, each with their own characteristic shape and performance.

[1] This is true for any methodology based on logic.

2.6 Adaptive Fuzzy Control

The ability of a controller to adapt to various plant conditions provides a robust control system which is desirable when the plant is uncertain, time-varying or non-linear[1]. The adaptability can also provide a method of auto-tuning a controller which is not yet optimally set-up. Many adaptive fuzzy controllers have recently been put forward and offer a new approach to designing the fuzzy controller. Procyk and Mamdani[11] originally proposed the self-organising controller but this has since been revised numerous times. Recently, contributions in fuzzy neural networks have been put forward. These use a gradient-descent algorithm to adapt the parameters of the fuzzy system.

The detailed theory of the adaptive fuzzy controller will not be discussed here, but a general overview of the available techniques is given below.

2.6.1 Self-organising fuzzy control

The self-organising controller consists of two parts : a fuzzy controller and an adaptation mechanism. The adaptation mechanism consists of 3 sections : the performance measure, the minimal model and the rule modifier. The performance measure is a fuzzy system which takes the same inputs as the fuzzy controller, but instead of a control action, the output is a performance measure. The minimal process model is used to convert the performance measure into a control signal change. The rule modifier then changes the control rules as required based on a reinforcement method.

[1] Which is represented by parameter variation in linear systems.

2.6.2 Fuzzy relations as associative memories

In this approach a fuzzy relation is used to model the process, and the same fuzzy relation is used to derive the control actions (causality inversion). The advantage of this is that the control and a system model is obtained simultaneously[10]. Predictive strategies are required if a time delay is present.

2.6.3 Adaptation by fuzzy supervisors

This approach involves the adaptation of a controller through a supervisory controller. This can take on two forms, namely, a fuzzy controller tuning a classical controller or hierarchical fuzzy control.

2.6.3.1 Fuzzy PID control

The concept behind this approach is to use a fuzzy system to tune the proportional, integral and derivative gains. Various approaches to tackling this idea have been proposed with some success, and the fuzzy tuning of PID controllers is being used in industry [32].

Fuzzy systems can also be used in conjunction with classical controllers to provide gain scheduling, time delay compensation and generally make the classical controllers more robust.

2.6.3.2 Adaptive fuzzy expert controller[10]

This is a hierarchical fuzzy control system with modules for direct control (continuous control of set-points), in-line adaptations (improve the steady state direct control) and on-line adaptations (supervisory module which makes changes based on the overall behaviour of the system). The inference session of a certain knowledge layer is started when the inference of the lower knowledge layer is finished. The final layer works on a time-scale which is several times slower than the slowest time-constant of the system.

2.6.4 Gradient-descent adaptation

These forms of adaptive fuzzy systems are also called fuzzy neural networks or neuro-fuzzy systems. The adaptation is based on a gradient-descent method which optimises the membership functions of the fuzzy system. The tuning is achieved through an objective function which is minimised (similar to learning in neural networks). This approach to adaptation is primarily used to obtain models, which are often used in control schemes.

2.7 Summary

Fuzzy set theory offers a new approach to tackling a problem. Unlike classical set theory, membership to a fuzzy set is graded. This approach is well suited to a human interface as the physical systems in the world are not always clearly defined. A fuzzy logic system consists of four major parts :

- Rules - These consist of an antecedent and consequent which are made up of fuzzy propositions. This is the most significant portion of the FLS as it defines the basic characteristic of the system.

- Fuzzification - In control applications, the fuzzification portion involves converting a crisp value (input to the controller) into a fuzzy singleton or fuzzy set. The fuzzy singleton is the most widely used approach as it simplifies the FLS significantly.

- Fuzzy inference - The inference system maps fuzzy sets into fuzzy sets using fuzzy implications. Each rule is interpreted as a fuzzy implication.

- Defuzzification - the conversion of the final fuzzy set into a crisp value which forms the output from the controller.

Although the fuzzy system appears to be a universal approximator, the vast array of design options makes the designer's choice a combination of experience, knowledge and preference.

Adaptive fuzzy systems offer a method of auto-tuning both fuzzy controllers and classical controllers. Various approaches to adaptive fuzzy control systems have been discussed.

3. Aspects of fuzzy and classical control

Chapter 2 introduced the theory and application of fuzzy logic to control design. The aim of this chapter is to provide a systems overview of fuzzy and classical control. Firstly, the relevant characteristics of the two control systems pertaining to the investigation are defined and discussed. The control problems experienced in feedback control loops are examined and the methods that each methodology uses to approach the problem is addressed. To illustrate where fuzzy control should be used, past applications and trends are examined. Finally, the issue of a fair comparison between controllers for an unbiased and meaningful conclusion is discussed.

3.1 Definitions

In order to make a worthwhile comparison, it is necessary to define what is being compared. This is essential if a clear understanding of the implications of the conclusions are to be achieved.

3.1.1 Classical Control

Classical control is a term which is applied to a body of techniques developed in the early years of control theory. It is characterised chiefly by the use of algebraic and graphical techniques applied to single input-single output (SISO) systems[33]. Classical control system design is usually performed using transfer function

descriptions. Closed loop specifications are usually made in terms of steady state error, rise time, settling time and other similar parameters. These are a few well known design techniques :

- Bode design
- root locus
- PID control

Robust control is an extension of the classical control techniques as it takes into account issues such as model uncertainty. Therefore plant non-linearities and other aspects which are difficult to control with conventional classical control can be compensated for. Quantitative feedback theory (QFT)[34] is a robust control design technique which will be used in the controller designs for this comparison. Another technique used for robust control is H_∞[33].

3.1.2 Fuzzy Control

A fuzzy logic controller is a non-linear mapping of an input vector (plant process measured variables (PV's)) into a scalar output. Fuzzy set theory and fuzzy logic establish the specific characteristics of the non-linear mapping. The fuzzy logic control methodology provides an interface between linguistic statements about control actions and the implementation of these statements in numerical form.

3.2 The classical control loop

In all the plant references, the SISO plant is assumed. This is due to the fact that this investigation only deals with SISO plants.

3.2.1 Reason for feedback

Feedback is typically used for one or more of the following reasons :

- To reduce the effect of *plant uncer[illegible]nty* or *unmodelled dynamics* of the system to be controlled. As there is uncertainty in the model, this cannot be compensated for in a prefilter. To reduce the effect of the unmodelled dynamics, feedback of the controlled variable is required, so that the state of the plant is known. If the plant model is known to an arbitrary accurateness, then feedback is not necessary.

- To reduce the effect of *external disturbances* on the plant. Unless all external disturbances can be measured and quantified, prefilter corrective action is impossible. Feedback is required to reduce the effect of the disturbances on the plant.

- To stabilise an unstable plant. If the plant is unstable in open loop, it is not realistically possible to stabilise the plant using a pre-filter. Pole-cancellation, while possible in theory, will not work in practice as the system poles are never known with sufficient accuracy, and hence the unstable pole will always be present. It should be noted that feedback can also destabilise an open loop stable plant.

3.2.2 Structure of a linear SISO feedback loop

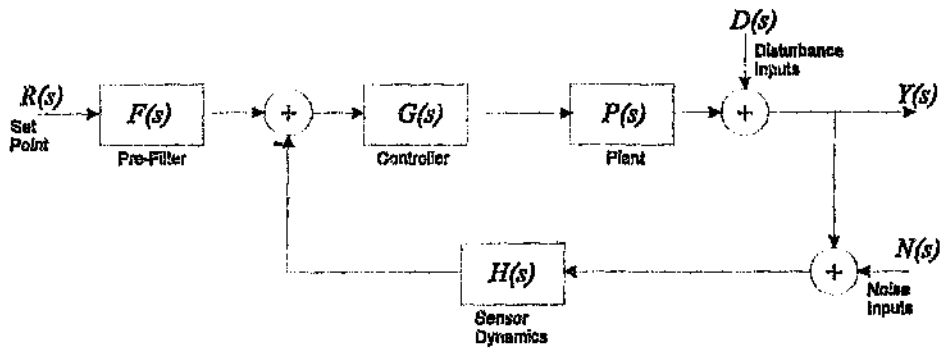


Figure 3-1 : SISO feedback structure

Consider a two-degree-of-freedom feedback structure as shown in Figure 3-1. The controller $G(s)$ performs the primary feedback tasks while the prefilter $F(s)$ shapes the overall system's response to achieve the specified performance.

The control system output $Y(s)$ is given by :

$$Y(s) = R(s)\frac{F(s)}{H(s)}T(s) + D(s)S(s) - N(s)T(s) \qquad (3.1)$$

where $L(s) = G(s)P(s)H(s)$ is the loop transmission

$S(s) = \dfrac{1}{1+L(s)}$ is the sensitivity function

$T(s) = \dfrac{L(s)}{1+L(s)}$ is the complementary sensitivity function

$D(s)$ is the disturbance input

$N(s)$ is the sensor or measurement noise input

$R(s)$ is the command input.

The block diagram can be reduced to unity gain feedback if the sensor feedback block $H(s)$ is removed and absorbed into the plant transfer function ($P'(s)=P(s)H(s)$) and prefilter ($F'(s)=F(s)/H(s)$). The equation then reduces to :

$$Y(s) = R(s)F'(s)T(s) + D(s)S(s) - N(s)T(s) \qquad (3.2)$$

3.2.3 Limitations of classical control

Consider the unity gain feedback SISO system introduced in Figure 3-1 with no prefilter. For all frequencies

$$S(s) + T(s) = 1,$$

which places some limitations on the plant's closed loop performance. Examining equation (3.2), the following can be observed :

- **Command tracking :** Assuming that $N(s)=D(s)=0$, then Y(s) is determined by $T(s)$, and hence $L(s)$.
- **Disturbance rejection :** $S(s)$ determines the extent to which a disturbance is attenuated. Thus $S(s)$ must be kept small, which is equivalent to a high open loop gain.
- **Noise Suppression :** To reduce the effect of noise, $T(s)$ must be kept small outside the control bandwidth, which is achieved through a low loop gain.

Looking at the above points, it is evident that at low frequencies, high gain is desirable for good command tracking and disturbance rejection. At high frequencies, a low open loop gain is required to suppress noise.

3.2.4 Background to QFT control

The QFT technique forms part of robust control design. It allows a single controller to be designed which will control a series of LTI plants to satisfy given bound specifications. Horowitz[34] introduced this theory and applied it to the SISO system. Since then, work on the QFT method has extended it to the MIMO case and it has been used in a variety of applications with success. The technique uses *structured uncertainty* at low frequencies and *unstructured uncertainty* (see section 3.3.4) at high frequencies[35]. Shaping of the open loop transfer function is performed on the Nichols Chart.

3.3 Control Aspects

The following is a summary of the aspects of a control system which are being investigated. Each one is introduced and then, where applicable, the classical and fuzzy methods of dealing with this aspect is discussed.

3.3.1 Stability

A stable system is one in which the output of the system does not grow without bound for any initial condition or for any bounded input. If a system is linearised or inherently linear, the stability (at the particular linearised point) is characterised by the pole positions. For a system to be stable, the poles must be in the left half of the s-plane (LHP). In a closed loop system, the zeros of $1+L(s)$ are the poles of both $T(s)$ and $S(s)$, which must all be in the LHP. Therefore in a linear controller, $G(s)$ must be chosen such that $1+L(s)$ has no zeros in the RHP.

3.3.1.1 QFT stability

The QFT design technique uses a Nichols chart for the controller design as it clearly illustrates the limitations of a feedback system. In order to determine whether a system is stable or not, the Nyquist Stability Criterion needs to be converted onto a Nichols chart. On a Nyquist Chart, the number of encirclements of the "-1 point" is critical. This maps to the (-180°, 0dB)[1] in the Nichols Chart. The Nyquist Stability Criterion and the actual mapping to the Nichols Chart is not discussed here, but is discussed in various texts[34,36].

To ensure a stability margin, the designer must keep L(s) a certain chosen distance away from this critical point. There are two common methods of specifying the contour around the -1 point :

$$|S(s)| = \frac{1}{|1+L(s)|} \leq \alpha \tag{3.3}$$

and

$$|T(s)| = \left|\frac{L(s)}{1+L(s)}\right| \leq \gamma \tag{3.4}$$

where α and γ are constants (usually > 1). This is an alternative and better specification to the conventional specification of gain and phase margin (only two points), as it defines a contour. Specification of $S(s)$ enforces a larger gain margin than $T(s)$ for the same phase margin. e.g. if α=2.3dB, this results in a phase margin of about 45° and gain margin of 15dB whereas if γ=2.3dB, the gain margin is 5dB.

[1] And multiples of -180°, depending on the number of unstable poles.

3.3.1.2 Fuzzy stability

Stability issues are one of the primary reasons for the slow acceptance of fuzzy control by the West. Fuzzy controllers are non-linear controllers and it is difficult to obtain general results on the analysis and design of fuzzy controllers[37]. Typically, the stability proofs found in the literature are restricted to cases where the fuzzy controllers are simple, like fuzzy-PID controllers[38].

Recently numerous papers have been released which tackle this problem[39,40,41,42]. These approaches use the definition of stability in the Lyapunov sense. The approach here is to model the plant numerically, convert the model into a fuzzy model and then prove stability. One advantage of this approach is that robust stability bounds can be enforced and limited parameter variation can be dealt with. These proofs are limited to Sugeno-style fuzzy systems and are usually complex and long. What is ideally needed, is a simple and universal test as is the case in classical control.

Stability defined by the bounded input - bounded output reasoning is unclear when used with fuzzy systems. Inputs always have to be bounded, as the input space is defined with bounds. The output of the controller, likewise, is bounded by definition. Thus a bounded input of the controller always results in a bounded output but this does not guarantee stability for unstable plants. However, the fuzzy system is often used with an integrator (i.e. the output of the controller is a change in control action) which complicates the stability issue. These forms of systems are sometimes called *dynamic* systems. To avoid confusion, this convention will be adopted, and fuzzy controllers which change their structure will be termed *self-organising* or *adaptive*.

An argument which has been put forward by Mamdani[43], is that considering the number of successful applications of fuzzy control in industrial processes and

consumer products, one could question the need for a mathematical stability analysis. Mamdani argues that stability is very important, but different approaches need to be found to study it. This could even mean prototype building for the purpose of approval certification. The problem with this approach is that the controller cannot be guaranteed to be stable during the initial design stage. Therefore simulations are required to verify its stability, but in practice a simulation model might not always be available.

3.3.2 Disturbance rejection

A disturbance is an undesired stimulus or input signal affecting the value of the system output. It may enter the plant before, in or after the plant. Sometimes the disturbances can be measured which then allows a suitable feedforward scheme to be used to reduce the effect of the disturbance. If the disturbance cannot be measured, then disturbance rejection in the feedback loop must be taken into account.

3.3.2.1 QFT design

The disturbance rejection bound is specified as a transfer function. The specification can be for an input or output disturbance. From equation 3.2, it can be seen that the amplification of the disturbance on the output is dependant on $S(s)$. Thus, ideally, $|S(s)| \ll 1$ which implies a high open loop gain. Usually the disturbances are of a low frequency relative to the plant's bandwidth, and the specification can take this into account. Only an upper bound is specified and usually consists of high attenuation at low frequencies approaching 1 at the mid to high frequencies.

3.3.2.2 Fuzzy design

Disturbance rejection in a fuzzy system is hard to incorporate meaningfully. Unlike the classical system, the bandwidth of the controller is theoretically unlimited since it is effectively a pure gain controller. The theory of high gain will achieve good disturbance rejection, but the noise aspect must be taken into account. Very little work has addressed this particular issue and more work needs to be done to give guidelines on designing a controller which will have good disturbance rejection. Questions like, "Does fuzzy control offer the ability to increase the disturbance rejection of a system?" and "How can one predict the extent of the disturbance rejection?" need to be addressed.

3.3.3 Sensor noise suppression

Sensor noise is introduced in the feedback path. While the magnitude of the noise is usually small (a few percent of the sensor reading), the effect can be significant. The solution to this problem is to roll-off $L(s)$ as soon as possible. Consider the transfer function relating the plant input $U(s)$ to the sensor noise output $(N_{out}(s)=N(s)H(s))$[22],

$$T_N(s)=\frac{U(s)}{N_{out}(s)}=\frac{-G(s)}{1+L(s)}=\frac{-P^{-1}(s)L(s)}{1+L(s)}=-P^{-1}(s)T(s)$$

If G(s) has a high gain at high frequencies, then $T_N(s)$ will be large and hence the noise will be amplified to significant levels resulting in controller inputs which are

meaningless (due to swamping by noise). $T(s)$ must roll off faster than $P^{-1}(s)$ rises[1]. If this is not the case, then noise becomes amplified at high frequencies.

3.3.3.1 QFT control

In order to maximise noise suppression, G(s) must be rolled off at as low a frequency as possible, and as fast as possible by maximising its phase lag. Thus G(s) must "hug" the robust stability bound, and once around the critical -1 point, significant lag must be introduced.

3.3.3.2 Fuzzy control

This is another section which has received very little formal investigation. Papers that do mention the noise aspect only show that the designed controller suppresses noise to some degree. What needs to be investigated are techniques in fuzzy control which ensure that noise will be suppressed and to what degree. Fuzzy control offers a very powerful tool for dealing with noise, and it is possible to reduce the effect of noise (but allow a small steady state error) by ignoring all controller inputs which are below a defined threshold.

[1] This means that the roll off rate of $T(s)$ must be faster than the roll off rate of $P(s)$.

3.3.4 Parameter variation

Parameter variation is introduced when there is modelling uncertainty. There are two categories for model uncertainties, namely *structured uncertainty* and *unstructured uncertainty*. Structured uncertainty assumes that the uncertainty is modelled, and hence can be quantified while unstructured only assumes that the frequency response lies between two bounds.

3.3.4.1 QFT control

The QFT design technique handles modelling uncertainty well. Parameter variation can be modelled as structured uncertainty. Using parameter variation, multiple plants can be developed and hence plant templates are formed which include all the possible modelled combinations of the physical plant. Using the QFT technique, the designer can then ensure that for all the plants in the template, the specification bounds will be met.

3.3.4.2 Fuzzy control

The fuzzy control system does have the ability to take into account measurement uncertainty through fuzzy numbers. This feature is rarely used in practice due to the increased complexity during the fuzzification stage. Instead, the measurement uncertainty can be included in the fuzzy input sets, as discussed in Chapter 2. The main problem is the lack of suitable design techniques which will take into account the extent of the model parameter variation. Tanaka[40] et al have developed methods (weak and strong robust conditions) of including model

3.3.4 Parameter variation

Parameter variation is introduced when there is modelling uncertainty. There are two categories for model uncertainties, namely *structured uncertainty* and *unstructured uncertainty*. Structured uncertainty assumes that the uncertainty is modelled, and hence can be quantified while unstructured only assumes that the frequency response lies between two bounds.

3.3.4.1 QFT control

The QFT design technique handles modelling uncertainty well. Parameter variation can be modelled as structured uncertainty. Using parameter variation, multiple plants can be developed and hence plant templates are formed which include all the possible modelled combinations of the physical plant. Using the QFT technique, the designer can then ensure that for all the plants in the template, the specification bounds will be met.

3.3.4.2 Fuzzy control

The fuzzy control system does have the ability to take into account measurement uncertainty through fuzzy numbers. This feature is rarely used in practice due to the increased complexity during the fuzzification stage. Instead, the measurement uncertainty can be included in the fuzzy input sets, as discussed in Chapter 2. The main problem is the lack of suitable design techniques which will take into account the extent of the model parameter variation. Tanaka[40] et al have developed methods (weak and strong robust conditions) of including model

uncertainty into the design process (t. trailer backing up from any initial position), but their method only ensures stability and not performance. Therefore, although techniques are being developed, they are complex and require a mathematical non-linear (or linear) model to be converted into a fuzzy model. Another problem can arise if the parameter variation changes the system dynamics significantly (e.g. from stable to unstable). The rules need to be chosen such that they are suitable for the plant's whole range of dynamics, and conflicting rules must be avoided.

One method used by some designers[20] to test 'robustness' of a fuzzy controller is to remove a certain percentage of the fuzzy rules, and then run tests to see if the controller still works. In terms of the definition of robustness, this method will say little about the true robustness of the controller. All this method is saying is that all the rules are not necessary. This approach is not used in this investigation.

3.3.5 Unmodelled dynamics

No model of a physical process includes every possible dynamic behaviour for $\omega \rightarrow \infty$. It is, therefore, necessary to ensure that the bandwidth of the controller does not exceed the modelled bandwidth.

3.3.5.1 QFT control

For a secure design, one should ensure that the plant model is known accurately up to the $L(s)$ phase crossover frequency. The roll-off must be fast enough such that any unmodelled resonant peaks will be below the stability contour.

3.3.5.2 Fuzzy control

The effect of unmodelled dynamics is unpredictable as the fuzzy controller does not have a definable frequency response. Another problem which arises is found in fuzzy modelling. The fuzzy model based on input-output plant data is only accurate up to the maximum frequency of the test data (at most). This fact is often overlooked in investigations. For such cases, simulations based on these models will give erroneous results for high frequency control signals.

3.3.6 Non-minimum phase plants

A non-minimum phase plant has at least one positive zero. This results in increased lag with magnitude increase. The effect of this is to provide an upper limit on the gain crossover frequency ω_c and therefore low frequency feedback benefits are minimised[22].

Fuzzy control offers a viable solution to this problem as the gain can easily be scheduled "independently" of frequency. No known research has as yet addressed this issue directly. This is investigated in Chapter 5.

3.3.7 Dead time

Dead time results in the system output being observable a certain time after the input is applied. This is a highly non-linear element which introduces a large amount of lag into the system and hence can cause instability. The effect of the time delay is that the maximum possible gain of the controller is reduced, and hence feedback benefits are minimal (settling time up to 9 t_d [44]) when the delay (t_d) is large compared to the dominant plant time constant.

Little investigation of fuzzy control with systems with significant dead time has been performed and is thus an area for future research. MacLeod[45] used a fuzzy system to implement a gain adaptive Smith predictor.

3.3.8 Saturation

One of the major problems in classical control is taking into account the amplitude and slew rate limitations of the control element. As a result, a well designed QFT controller (ignoring saturation) which meets all the specifications may not operate as expected when simulated or tested on the actual system. This problem can be avoided if the QFT bounds are carefully designed, such that the system will be able to achieve the required response without saturating[22]. If this property is totally ignored, it could lead to a degradation in the performance specification, an unstable system or one which damages the control element due to bang-bang behaviour.

A fuzzy system does not have this limitation. Because the input and output bounds are set at design time, the controller output can (usually) be chosen to match that of the control element. Problems can arise, however, in dynamic fuzzy controllers as secondary dynamic elements (e.g. an integrator) are used.

3.3.9 Temporal determinism

For a system to be reliable, determinism in time is essential[1]. Unless the information required is obtained within the required time period, the information is invalid[46]. The traditional approach is to make the system "run fast enough". This approach is totally unsuitable for use with fuzzy control and any AI approach. It is essential that the fuzzy controller is deterministic, regardless of the number of rules fired and the solution reached. Significant work has been done in ensuring that software is logically deterministic, but in a real-time process, the time factor cannot be ignored. This problem is compounded when adaptive or neuro-fuzzy systems are used. Here the system is constantly changing as the stimulus to the controller changes. Neural networks are inherently not temporally deterministic. Thus in order for these AI techniques to be accepted, determinism needs to be introduced[2].

3.3.10 Reliability and safety

The problem which faces control engineers is that there is increasing pressure to produce products which society can depend on[47]. The safety, reliability and predictability (logical determinism) of classical systems (especially PID systems) has been thoroughly investigated, but the new emerging AI technologies have not been. The reliability of fuzzy systems has not been investigated sufficiently which explains the reluctance of companies to adopt this new technology. e.g. Consider

[1] Often data consists of two parts : the value and the expiry time. Should the data be received after the expiry time has passed, then the data is not used and an error is signalled to the operator.

[2] Temporal determinism is essential especially in distributed systems. Therefore if AI techniques introduce more temporal uncertainty then they will not be used.

an adaptive fuzzy controller. If the controller is exposed to conditions which were not foreseen, the adaptive mechanism might cause the controller to behave worse than before and compound the problem. This exposes one of the problems with a complex fuzzy system. The equations describing the input-output relationship of the fuzzy controller are complex. Thus, although the system is predictable, the complexity may hide possible dangerous situations. Coding of the fuzzy controller into software can introduce bugs due, also, to the complexity of the controller. PID control, due to its simplicity, does not have this problem.

There are two approaches proposed by Rodd[47] :

- protect the plant from any adverse decisions made by the controlling system (currently used in industry for normal control applications), or
- make the controlling system deterministic (A better and more long term approach).

3.4 Applications of fuzzy control

This section serves as an introduction to the history and current applications of fuzzy control. This is given to indicate the trend in fuzzy control and where the future is moving. The list of fuzzy applications is by no means definitive, but does serve to illustrate where fuzzy control has been used.

3.4.1 History of fuzzy control

Fuzzy sets were introduced by Zadeh[3]. Fuzzy logic has since then been used in diverse fields such as sociology[2] (e.g. branching questionnaires) and engineering.

The first paper which applied fuzzy logic to the area of control was, again, Zadeh[3]. The first application reported was by Mamdani[6] and Mamdani and Assilian[7], who reported the use of fuzzy set theory to control the steam pressure and throttle of a small laboratory engine. Since this paper, fuzzy control has been used in a wide variety of processes including cement kilns[48] (revolutionary example), servo motors and various process plant applications.

3.4.2 Summary of some applications of fuzzy control

Firstly, each application is described, and then a table summarising the aspects of the fuzzy applications is given.

Tanaka[40] : A robust stability approach to solve the truck-trailer problem for any initial condition. Parameter uncertainty was taken into account and various robust stability conditions were investigated.

Bernard[49] : Implementation of a fuzzy controller for the control of power on a research reactor. No mathematical model was developed, instead all information was obtained from experience and "deep" knowledge of the dynamics.

Song[50] : An adaptive fuzzy controller which adjusts its rules based on the past control trends. Various systems were controlled including a second order plus dead time system and a CSTR.

Feng[51] : Development, simulation and implementation of a fuzzy logic controller for use in digital speed control of ac servo systems. The design also included an intelligent inverter which reduces the switching loss and the current harmonics in induction motors.

Li and Lau[21] : An investigation into using fuzzy logic in a system where fast and accurate response is required (servo motor). PID and MRAC control are used for comparison.

MacLeod[45] : A fuzzy logic system is used, in conjunction with a Smith predicator, to tune a classical controller. The fuzzy system changes the controller gains based on past plant inputs and outputs.

Assayama[52] et al : Kilns have proved to be the biggest promoter of fuzzy logic. This is because modelling of kilns is extremely difficult and hence fuzzy logic offers a way of automating the operation of the kilns.

Shaw[16] : An automatic transmission system was developed by Nissan to take into account the driving conditions when shifting gears. Data was obtained from an analysis of driving patterns for differing road conditions. Therefore the system would behave differently on a mountainous road than it would on a freeway.

Table 3-1 : Summary of some fuzzy application

Designer	Actual Process / Simulation	Math. Model available and stable ?	Linear / Non-linear Model	Adaptive / Static Controller	Fuzzy Type	Comparisons with classical controllers and were they fair?	Conclusion
Tanaka	S	Yes, (unstable)	Non	Static	Sugeno	None	Stability can be ensured for some systems
Bernard	P	No	Non	Static	Mamdani Singleton	Yes	Superior to classical approach.
Song (CSTR)	S	Yes (unstable)	Non	Adaptive	Mamdani	Comparison to FLC - (Badly tuned)	Provided better control than FSOC and FLC.
Feng at al	S & P	Yes (time-varying)	Non	Static	Mamdani	No direct comparison	Faster response with reduced switching loss.
Assayama et al	P	No	Non	Static	Mamdani	N/A	Automation of kilns is now possible.
Shaw	P	No	Non	Static (some dynamic properties)	Mamdani	Compared to existing system.	Drivability improved.
MacLeod	S	Yes	Non (Dead Time)	Static (fuzzy)	Mamdani	No	Robust controller for long dead times was achieved.
Li and Lau	P	Yes	Linear	Static	Mamdani	PID and MRAC	Fuzzy had less overshoot and a smoother response.

3.5 Qualitative versus quantitative knowledge

Fuzzy control is often claimed to be a model-free design approach. This is not true. While it may be seen as a mathematical-model-free approach, a "fuzzy" model or linguistic qualitative model is required. Knowledge is required about the dynamics and response of the system. The ranges of all the universe of discourses need to be known, and the chosen rules must implement some form of meaningful control. One aspect in which fuzzy control is growing, is the combination of both the mathematical and the linguistic information in the design of the fuzzy system. The trend is to convert the mathematical model into a fuzzy model, design the controller and after this, incorporate linguistic knowledge about the plant. It also offers possibilities of defining a system qualitatively, and then obtaining a qualitative model of the system.

Figure 3.2 indicates where, based on past applications, fuzzy logic can be used, and has been used, successfully.

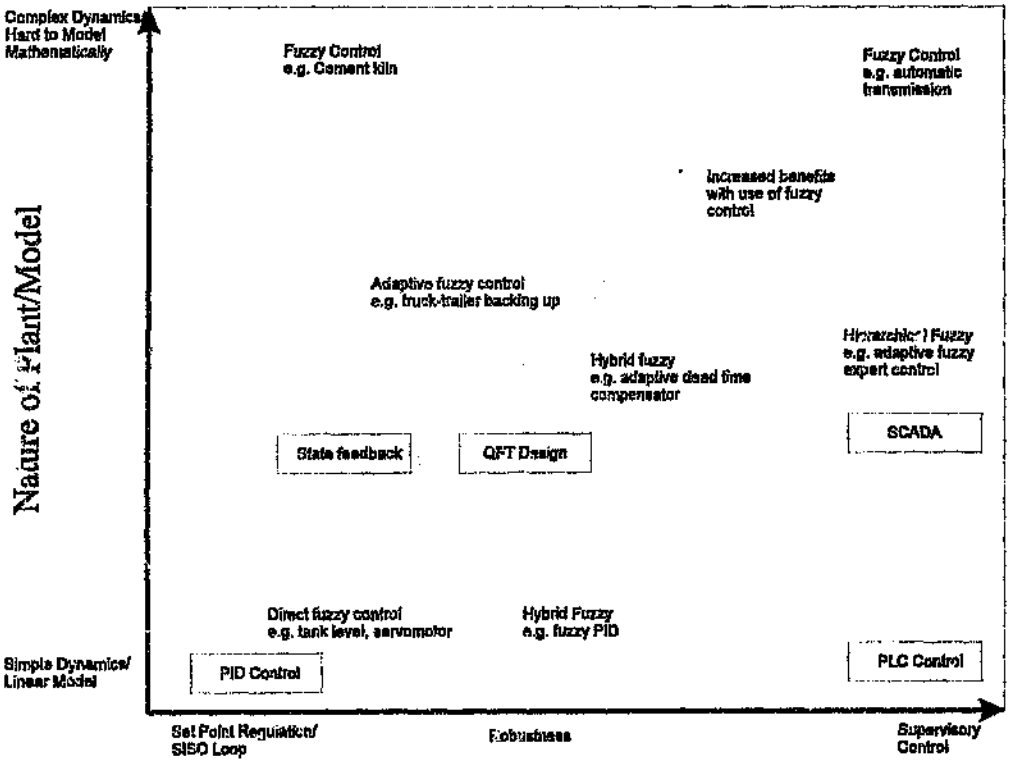


Figure 3-2 : Distribution of fuzzy applications

3.5.1 Tools available for fuzzy control

Recently, with the advances in desktop computing power, the number of fuzzy application tools has increased dramatically. The following is a list of some of the available fuzzy development and simulation tools :

- Fuzzy Logic Toolbox for MATLAB® (Mathworks)
- fuzzyTECH® (Inform Software Corp)
- UNAC's Fuzzy Tool (CICS Automation)
- FIDE (Fuzzy Inference Development Environment) (Aptronix)
- RICE (Routines for Implementing C Expert Systems) (Shareware)
- NEFCLASS (Neuro-Fuzzy System) (Shareware)
- FuzzyCLIPS (Fuzzy Expert System Shell) (NRC - Shareware)

Fuzzy systems can be coded onto standard microcontrollers or dedicated fuzzy logic coprocessors (Omron's FP3000). Fuzzy boards for PLC's, as well as fuzzy PLCs are becoming available.

3.6 Approach to comparing control systems

The approach taken in this investigation is to compare the control methodologies in as fair a manner as possible. Fairness was taken to mean that it is essential when doing the comparison, that the information supplied to each controller is the

most appropriate for the control design methodology. This has serious implications when comparing the systems. Most literature, when comparing to the classical system, uses the PID control standard. While this is an industry standard, it is not the best classical method available for control design. Thus the QFT technique is used as it offers an intuitive and clear method of designing robust controllers. The only variation on this approach is in the last investigation where the fuzzy-classical hybrid system is used.

The complexity, design time, robustness, ease of design and ease of implementation are all examined in each case, and hence, performance is not the only issue being investigated.

3.7 Summary

Definitions of both fuzzy and classical control have been proposed to try and indicate exactly what the investigation is attempting to compare. The control loop as well as the classical control problem has been summarised. This serves to illustrate the point that although there are limitations on classical linear control systems, the theory and its limitations are well understood and developed. This is not true for fuzzy logic control. The approach of the two control methodologies for certain aspects of the control loop have been introduced and discussed. A small selection of fuzzy applications used in industry and under investigation have been presented in an attempt to illustrate the diverse areas in which fuzzy control has been applied. The approach to the investigation is intended to be as fair and as impartial as possible.

4. Plant Modelling

4.1 Introduction

The plants chosen for this investigation each highlight a certain aspect i.. the control problem. All the models are SISO systems, as the MIMO case is not investigated here. Fuzzy controllers are essentially MISO systems and any fuzzy MIMO controllers can be decomposed into a MISO system.

The modelling serves two purposes :

- allows for the design of classical controllers (QFT),
- permits simulation on a computer.

When designing the fuzzy controllers, the mathematical model is ignored and the approach is such that only qualitative information about the external behaviour of the plant (black box) is used.

Simulations of the models are used to test the controllers and hence, no practical implementation is performed. Simulation offers the following advantages :

- Computer simulation is quicker and is not reliant on the actual operation time of the practical systems, but rather on the complexity of the system. Therefore a system with a time constant in the region of hours can be meaningfully simulated in a few minutes.
- Simulation is much more flexible and offers the attractive ability to perform a "what if" analysis easily.

- The cost of simulation is far less than building the proposed systems and alterations can easily be made.
- Ensuring a controlled environment where only the aspects which are to be investigated are present is difficult to achieve with a real system. In simulation, the designer has complete control over what external influences are present.

Despite these advantages, the practical implementation is still essential to ensure that the modelled results are accurate and reflect the actual practical results. Simulation has many limitations. If a controller works in simulation, the extent to which it will work on a practical plant is dependant on the correctness and completeness of the simulation model. If significant disturbances or dynamics have been ignored, then the controller might be totally unsuitable. When designing the controller in this investigation, the limitations of each model are noted and taken into account.

4.2 Servo motor system

The first model chosen is the armature controlled DC servomotor. The proposed system of control is positional control.

4.2.1 Rationale

The servomotor exhibits the following characteristics :

- Linear (or near linear) dynamics

- Marginally unstable when positional control is being implemented
- Fast ($\tau < 0.5$ s) response
- A well developed and accurate model is available.

The linear aspect of this model is seen as being important in this comparison. Jager[10] et al make the comment that fuzzy logic controllers are highly non-linear, and therefore only show significant performance improvements when used on equally non-linear plants. Other papers, such as Li and Lau[21], show fuzzy controllers outperforming PI and MRAC controllers on a servo system. With these conflicting ideas, the aim of this plant investigation is to try and reach a definitive conclusion. Aspects such as controller development time and the controller complexity levels for the designer have not been sufficiently analysed. The marginal instability of the system also offers the opportunity for the fuzzy controller to stabilise a system without the complexity of the system becoming an overriding factor (The CSTR model, in contrast, offers both instability, non-linearity and a high level of complexity).

4.2.2 Assumptions

The following assumptions are made when modelling the servo motor :

- Friction is linear i.e. the rotational damping coefficient is constant regardless of speed of rotation.
- The load is modelled as an inertial load with rotational friction.
- The electromagnetic, circuit and physical equations used are all valid for this application.

4.2.3 Limitations

The model chosen is limited in that all non-linear aspects of the motor have been ignored (e.g. stiction). The non-linearities are seen to be insignificant for this problem and therefore this limitation should not make the model invalid.

4.2.4 Mathematical Model

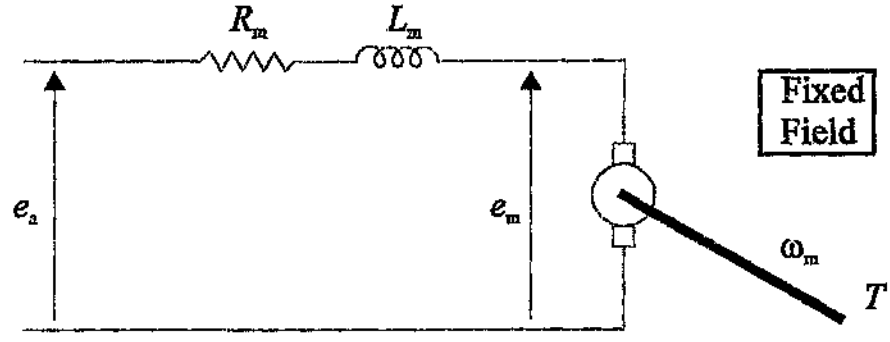


Figure 4-1 : Circuit diagram of a DC motor

For this control problem, the plant input is the voltage to the motor and the plant output is the motor position (θ).

The field circuit is excited from a constant dc source . The torque is thus only dependant on the armature current[53].

$$T(t) = K_T i_m \qquad (4\text{-}1)$$

The *back emf* produced, or induced voltage, is directly proportional to the speed of rotation :

$$e_m = K_b \dot{\theta} \tag{4-2}$$

Control of the motor is achieved through adjusting the armature voltage e_a. The polarity in turn decides the direction of rotation. Applying KVL around the armature circuit :

$$L_m \dot{i}_m + R_m i_m + e_m = e_a \tag{4-3}$$

The required torque depends on the load attached to the motor. As the load is modelled as a torque and frictional load, the torque is :

$$J\dot{\omega} + B\omega = T(t) \tag{4-4}$$

Combining (4-1)→(4-4) and choosing position (θ) as the state variable gives the final equation :

$$\frac{L_m J}{K_T}\dddot{\theta}_m + \frac{L_m B + R_m J}{K_T}\ddot{\theta}_m + \frac{R_m B + K_b K_T}{K_T}\dot{\theta} = e_a \tag{4-5}$$

As can be seen, the model is linear, and therefore linearisation is unnecessary. The block diagram of the motor is given in figure 4.2.

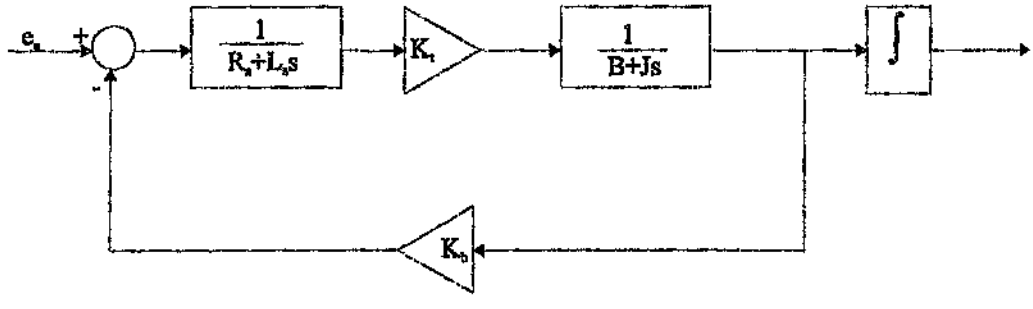


Figure 4-2 : Block diagram of DC Motor

4.2.5 Physical Parameters

The model used in this design is based on an Electrocraft Corporation servo motor (PN : 0400-04-041). The physical parameters are summarised in Appendix A. Modelling uncertainty is introduced through parameter variation.

4.2.6 Final Model

The final servo motor transfer function obtained is given below :

$$P(s) = \frac{1}{s} \frac{[7.6e5\ 1.2e6]}{s^2 + [613\ 673]s + [3.1e4\ 6.4e4]} \qquad (4.6)$$

The model is second order and linear. The parameter variation is large which is intentional. This will serve to indicate which controller can handle parameter variation better. As the plant is linea he effect of the parameter variation will be dominant and not be overshadowed by any non-linearities.

4.3 CSTR plant model

The CSTR (continuous stirred-tank reactor) is a system in which an exothermic, irreversible reaction occurs. The plant is unstable, highly non-linear, MIMO and sensitive to disturbances, noise and uncertainty[54]. This form of reactor is used extensively in industry for the treatment of effluent or neutralisation of substrates. Large plants have been used to produce polyethylene[55].

The plant used is a testbed designed by Gordon[54] to simulate the real process as closely as possible and provide a convenient testbed for controller verification.

4.3.1 Rationale

The CSTR is an ideal plant to investigate high non-linearity with large complexity. The plant offers the following characteristics :

- unstable around an operating point
- sensitive to noise and disturbances
- highly non-linear (see 4.3.2)
- large amount of parameter uncertainty present

4.3.2 Assumptions

In a system as complex as the CSTR, the assumptions are numerous and far-reaching. As the aim of this project is not to examine the operation of the CSTR, the assumptions are only summarised here. These are dealt with in more detail in Gordon[54].

- Reactant N is present in excess with M being the limiting reactant.
- The following are considered constant for the plant model :
  1. density
  2. specific heat capacities
  3. mole fraction of chemical feeds M and N
  4. temperature of input streams F_1 and F_2

- water is the inert substance
- M, N and P are all diluted to the extent that their densities are equal to water
- Ideal mixing is assumed
- Volume in tank is constant (inflow = outflow)
- The specific heat capacity relationships hold: $Cp_P = Cp_M + Cp_N$
- The cooling coil is always submerged.

4.3.3 Mathematical Model

The input to the CSTR is the coolant flowrate (F_c), while the output is the internal fluid temperature (T).

The derivation of the model is complex and is not dealt with in this report. A summary is given instead.

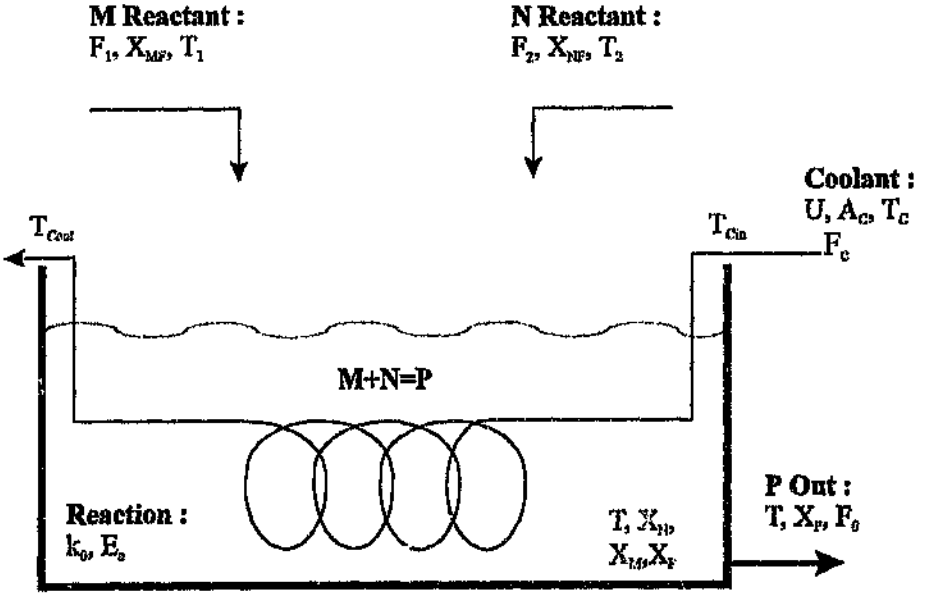


Figure 4-3 : CSTR Plant

Summary of Operation :

The CSTR consists of two input streams F_1 and F_2 which supply diluted reactants M and N. When reactants M and N come into contact (with a catalyst if necessary) an exothermic reaction occurs. The reaction is M + N ⇒ P. The inert substance (water) together with the product P (with remaining unreacted reactants left M and N) is removed (F_0). As the reaction is exothermic, the temperature must be controlled. This is achieved through the use of a cooling coil, through which a cooling substrate flows (F_c). Figure 4-4 shows the heat released by the reaction as well as the heat removed by the coolant (linear function of F_c). For a steady state condition, the heat removed must be equal to the heat added by the reaction. As can be seen, this results in three possible operating points. The top and bottom point (P_1 and P_3) are stable, while P_2 is unstable. Keeping the plant at this point is difficult as a slight deviation might cause an increase in temperature, which in turn results in a higher reaction rate and hence more heat released; leading to a rise in temperature.

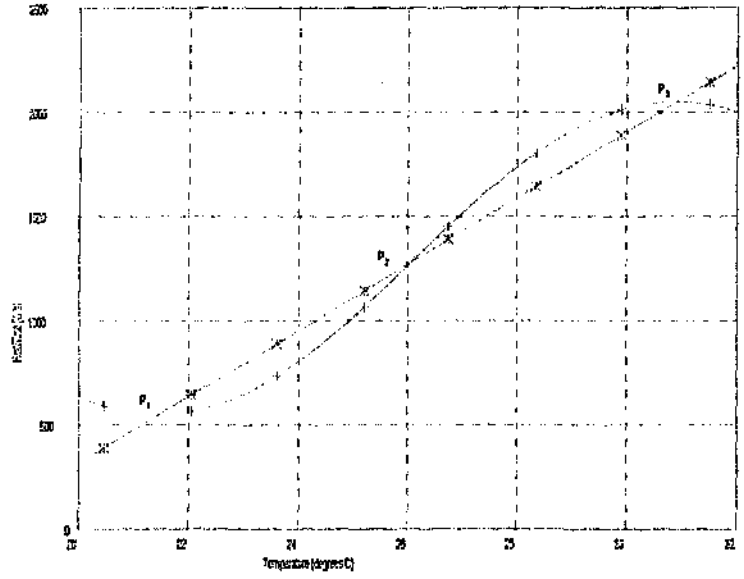


Figure 4-4 : CSTR Characteristic curve
* - q removed by coil
+ - q formed by reaction

The CSTR is to be operated at the unstable point for the following reason[56]:

- P_1 gives a low product yield due to the low temperature
- P_3 has a high relative temperature which could result in a destroyed catalyst and degradation of the product.

The equations for the system are summarised below :

$$\frac{dx_M}{dt} = \frac{1}{V\rho}\left[F_1\left(x_{M_F} - x_M\right) - k_0\rho x_M e^{-\frac{E}{RT}}V - F_2 x_M\right] \tag{4-7}$$

$$\frac{dx_N}{dt} = \frac{1}{V\rho}\left[F_1\left(x_{N_F} - x_N\right) - k_0\rho x_M e^{-\frac{E}{RT}}V - F_2 x_N\right] \tag{4-8}$$

$$\frac{dx_P}{dt} = \frac{1}{V\rho}\left[k_0\rho x_M e^{-\frac{E}{RT}}V - F_2 x_P - F_1 x_P\right] \tag{4-9}$$

$$\begin{aligned}\frac{dT}{dt} = \frac{1}{V\rho\beta}[&F_1\left(X_{M_F}\left(\Delta\hat{H}^o_{F_M} + \hat{C}_{P_M}\left(T_1 - T_0\right)\right) + \left(1 - X_{M_F}\right)\left(\Delta\hat{H}^o_{F_I} + \hat{C}_{P_I}\left(T_1 - T_0\right)\right)\right) + \\ &F_2\left(X_{N_F}\left(\Delta\hat{H}^o_{F_N} + \hat{C}_{P_N}\left(T_2 - T_0\right)\right) + \left(1 - X_{N_F}\right)\left(\Delta\hat{H}^o_{F_I} + \hat{C}_{P_I}\left(T_2 - T_0\right)\right)\right) - \\ &q \;-\; (F_1 + F_2)\gamma \;-\; V\rho\alpha\;]\end{aligned} \tag{4-10}$$

where

$$\begin{aligned}\alpha = &\frac{dx_M}{dt}\left(\Delta\hat{H}^\circ_{F_M} + \hat{C}_{P_M}(T - T_0)\right) + \\ &\frac{dx_N}{dt}\left(\Delta\hat{H}^\circ_{F_N} + \hat{C}_{P_N}(T - T_0)\right) + \\ &\frac{dx_P}{dt}\left(\Delta\hat{H}^\circ_{F_P} + \hat{C}_{P_P}(T - T_0)\right) - \\ &\left(\frac{dx_M}{dt} + \frac{dx_N}{dt} + \frac{dx_P}{dt}\right)\left(\Delta\hat{H}^\circ_{F_I} + \hat{C}_{P_I}(T - T_0)\right)\end{aligned} \tag{4-10a}$$

$$\beta = x_M \hat{C}_{P_M} + x_N \hat{C}_{P_N} + x_P \hat{C}_{P_P} + \left(1 - x_M + x_N + x_P\right)\hat{C}_{P_I} \qquad (4\text{-}10b)$$

$$\begin{aligned}\gamma = {} & x_M\left(\Delta\hat{H}^{\circ}_{F_M} + \hat{C}_{P_M}(T - T_0)\right) + \\ & x_N\left(\Delta\hat{H}^{\circ}_{F_N} + \hat{C}_{P_N}(T - T_0)\right) + \\ & x_P\left(\Delta\hat{H}^{\circ}_{F_P} + \hat{C}_{P_P}(T - T_0)\right) + \\ & \left(1 - x_M + x_N + x_P\right)\left(\Delta\hat{H}^{\circ}_{F_I} + \hat{C}_{P_I}(T - T_0)\right)\end{aligned} \qquad (4\text{-}10c)$$

A detailed block diagram of this system is given in Appendix B. Figure 4-5 contains a simplified block diagram. This block diagram gives a better understanding of the system and the interrelated properties.

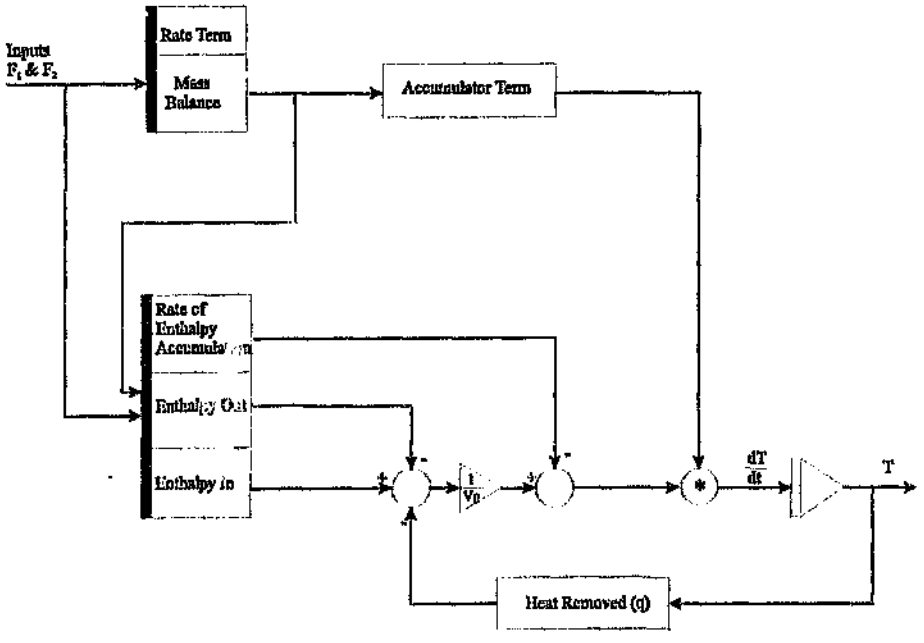


Figure 4-5 : Simplified block diagram of the CSTR

4.3.4 Parameter Values

The parameter values are chosen such that the steady state values are all achievable by the available physical equipment[64] These are given in Appendix A.

4.3.5 Linear Model

The CSTR is highly non-linear and needs to be linearised if any classical control techniques are to be used. The plant is linearised through partial differentiation with respect to the states and the input F_c. The resultant equations are long and complex, but result in a system with three zeros and four poles. At the operating points P_1 and P_3, all poles are negative as expected, but at P_2, one pole is positive. This is in agreement with the expected behaviour. The linear model was verified against the non-linear model and the behaviour is accurate for small perturbations around the respective operating point.

The linear plant is of fourth order. Analysis of this plant shows that pole-zero reduction is possible and the resultant system (including parameter variation) is :

$$P(s) = -\frac{[8.8431e1\ 1.1080e2]s + [4.8159e(-1)\ 5.6840e(-1)]}{s^2 + [8.5546e(-3)\ 9.5561e(-3)]s - [1.3253e(-6)\ 3.21114e(-6)]}$$

Note: Pole-zero reduction has been performed on the above plant to reduce the order of the plant. The plant is only linearised around the required operating point. Parameter variations were chosen by analysing the CSTR characteristic curve and ensuring that the variation is physically possible and meaningful. It should also be noted that the plant has a negative gain. This is expected as an increase in coolant flow causes a decrease in reactant temperature.

The actual CSTR has numerous disturbances. For the purpose of this investigation, only the temperature of the M reactant inflow will be examined. The uncertainty in the model is introduced through the kinetic rate constant and the required activation energy for the reaction.

4.4 Tank System

The tank system is chosen as a method of investigating dead time. Classical control struggles with dead time as the control gains are linear. Fuzzy control may offer some respite in this application as the control gains can be scheduled.

4.4.1 Rationale

The tank system is chosen as it illustrates a system which can have dead time. The delay is assumed to be between the request for a change in flow and the actual change in flow.

4.4.2 Assumptions

The following assumptions are made in the derivation of the model :

- Oscillations and ripples present on the surface are neglected.
- Laminar flow is assumed
- The valve characteristic is linear.
- The orifice equation is valid for this model.
- The density of the liquid is constant.

4.4.3 Limitations

This model is only used to highlight aspects of the control problem, and thus the model might not reflect the practical system as it would be implemented.

4.4.4 Mathematical Model

The input to the plant is the control valve and the output is the tank level.

The control valve of the tank system is on the outlet. A tank system is a form of capacity or integration. Thus the height of the tank can be given by :

$$h = \frac{1}{A.\rho}\int[q_{in} - q_{out}]dt \qquad (4\text{-}11)$$

where q_{in} and q_{out} is the mass flowrate into and out of the tank respectively. The tank is fed by an upstream supply with flow rate q_{in}. This passes into the top of the tank, giving the non-linear relationship :

$$q_{in} = K(u)\sqrt{\rho.g.h_1} \qquad (4\text{-}12)$$

where u is the normalised valve position (0→1).

The final equation for the tank (assuming a linear valve) is thus :

$$\dot{h} = \frac{1}{A.\rho}\left[q_{in} - K.u.\sqrt{\rho.g.h}\right] \qquad (4\text{-}13)$$

Dead time is included in the tank system. This dead time is implemented as a delay between a request for a change in input flow and the response. This system

offers an ideal system to investigate dead time as it is a simple first order system when linearised.

4.4.5 Physical Parameters

The physical parameters are chosen to represent a typical tank as would be found in industry. The parameters are given in Appendix A.

4.4.6 Linearised plant

The tank system is linearised over all possible set point levels.

$$P(s) = \frac{[0.0422\ 0.0895].e^{-t_d s}}{s + [0.0093\ 0.0426]}$$

The tank block diagram is given in Figure 4-6.

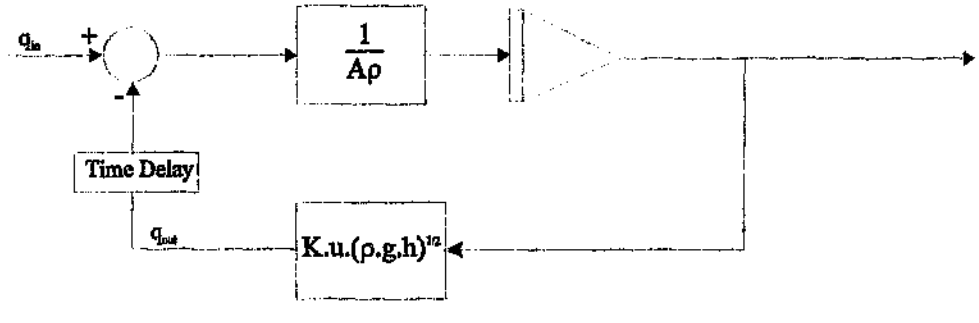


Figure 4-6 : Tank block diagram

4.5 Miscellaneous plants

The following plants are also used in the investigation :

- FOPDT (First order plus dead time) system,
- nonminimum phase system.

These plants are used to investigate specific aspects of controllers.

4.6 Summary

The servo motor is a linear system with fast dynamics. Conflicting reports about the role of fuzzy control with regard to linear systems makes this model applicable for the investigation. The CSTR is a highly non-linear and unstable plant. This plant is included in the investigation to highlight each controller's ability to handle non-linear dynamics and large instability.

The tank system with dead time and various other systems are included to illustrate specific control problems.

5. Controller design and comparison

Chapter 4 introduced the models which are being used in the investigation. This chapter summarises the design, implementation and results of the classical and fuzzy controllers. First, the design methodology used for each of the QFT and fuzzy controllers is presented. Then the controller design for each system is discussed. The results and conclusions are given with a summary at the end.

5.1 Controller design approach

The aim of this investigation is to ensure a fair comparison between the classical and fuzzy controllers. This requires that the information supplied to each controller is the most appropriate for the control design methodology. This also requires that the performance specifications chosen for each controller are equal and have the same meaning. This aspect is not easy to achieve, as each design technique requires the specification to be given in a different format. The classical (QFT) method requires the specification bounds to be in terms of a required frequency response[1] for whatever aspect of the plant the specification bound is being imposed. For the fuzzy controller (which is not easily mapped into the frequency domain), the specification bound is given as a qualitative requirement[2], which will be met through iteration.

Each design approach is discussed emphasising points which are relevant to this particular study.

---

[1] Or the bounds must be readily mapped into some form of frequency bound.

[2] This is not true for all forms of fuzzy controllers. Fuzzy controllers designed using a form of robust design technique can use some quantitative information in the design.

5.1.1 QFT design

For QFT[35], like all classical techniques (except PID), a model is required of the system to be controlled. The models need to be linearised around their respective operating points. If more than one operating point is present, then this can be included in the parameter variation (which leads to multiple plants). Some form of specification bounds are required for each system. These bounds vary depending on the system and type of requirements. In the case of the servo motor, tracking bounds as well as robust stability and disturbance bounds are required. In the CSTR, the tracking bounds are not necessary. In order to ensure that the most is obtained from the classical approach, the bounds are made as tight as possible, while still ensuring that the final control system is physically realisable. Thus if the bounds are easily met with little gain, the bounds are made more stringent. Simulation is used to verify that the controller meets the performance specifications.

5.1.2 Fuzzy design

There is no specific methodology for designing and tuning a fuzzy logic controller. These range from full a-priori knowledge with no tuning to on-line adaptive fuzzy and hybrid systems. This investigation deals only with a manually tuned fuzzy controller and fuzzy tuned classical controllers. Adaptive fuzzy systems often suffer from the problem inherent with neural networks : the final controller, while working, does not disclose meaningful information about why or what it is actually doing. In order for a fair comparison though, adaptive classical control would need to be compared with adaptive fuzzy control.

The fuzzy logic control design procedure followed is :

1. Define the system which must be controlled by specifying the inputs, outputs, disturbances, sensors and system operating information available. Therefore an overview of what must be controlled and the operating boundaries of the system is required.

2. Define the control goals qualitatively and quantitatively. As a fuzzy system cannot include the tracking or disturbance rejection bounds directly, qualitative bounds are required. Quantitative bounds are needed to define the required performance bounds. e.g. the step response must have a "time constant" (a fuzzy system is non-linear and hence the time constant as defined in the classical sense does not hold) of 10 seconds or the response must be within 5% of a set point after 10s. While these bounds cannot be directly included, the fuzzy system can be tuned until they are achieved.

3. Understand the external dynamics of the system. This amounts to obtaining the operator knowledge. For example, an operator who has used the system manually, knows how the system responds to certain inputs. It should be noted however, that operator knowledge is not always easily obtainable and often conflicting. This is, therefore, an "ideal" case when the operator information is meaningful and easily extracted.

4. Define the type of inputs and outputs of the fuzzy controller. This includes defining the required external dynamic components which determines whether the output is an absolute value or an incremental one.

5. Define the universe of discourse of the input and output. This includes the range of values which must be included. These typically can be calculated from external measurements or from design specifications.

6. Assign the fuzzy sets with their respective linguistic terms. The arrangement of these sets is significant as the closer together the sets are, the finer the

control which can be implemented resulting in a greater complexity. For large fuzzy control input signals, the sets can be spaced far apart, as the exact value of the current plant output is not significant, aside from the fact that the deviation from the setpoint is large. Typically the number of sets is three, five or seven. More sets and linguistic terms can be added later.

7. Assign the FAM bank or rule base. This is obtained from converting the knowledge gained from the operator (in step 3) into a set of rules.

8. The methods of fuzzification, connection, aggregation, implication and defuzzification need to be defined (see Chapter *4*).

9. Generate the control surface. Every possible output, for every possible input (over the universe of discourse) is shown in this surface, and hence the action is deterministic. It is therefore possible for the designer to have complete control over all areas of the control surface. By examining the control surface carefully, it is possible to identify any areas where the control is inappropriate or missing. The magnitude of the output is important and it should be noted whether the entire input and output space has been accounted for. It is typical of some defuzzification methods (e.g. centroid), not to use the entire range of the output space. This is due to the effect of averaging. This is compensated for by increasing the output space.

10. Simulate and tune the fuzzy controller until it is within the desired specification bounds. Tuning is usually difficult and requires numerous simulations. There are various methods which are employed in this investigation :

    - Examine the trajectory of the system plotted on the control surface. (i.e. a plot of *Error* vs. *dError*). Areas where the response is slow or not understood can be clearly seen and corrective action taken.

- Change the output sets (if the input set positions are understood to be correct and meaningful), such that more or less control action is applied at certain regions.
- Modify the rules when the action taken is clearly inappropriate for the situation.

5.2 Controller specification and design

A controller for each plant is defined which satisfies given performance bounds. The actual controllers and their respective plots (either Nichols or control surfaces) are given in Appendix C (unless directly relevant).

5.2.1 Servo motor

Aim : To determine the controller's ability to deal with a well defined linear plant, as well as clearly defined tracking bounds.

Plant Input : Signal voltage to amplifier

Output : Motor angular position (in degrees)

5.2.1.1 Plant model

The plant model for the QFT design case is described in Chapter 3. For the fuzzy design, however, the model needs to be converted into a qualitative one such that

control rules can be obtained. Through a series of step tests it is possible to qualitatively determine what voltage results in what motor speed. From this, the input and output ranges, as well as the rules were generated.

5.2.1.2 QFT specifications and design

The servomotor model is summarised in (5-1):

$$P(s) = \frac{[7.6e5\ 1.2e6]}{s^2+[613\ 673]s+[3.1e4\ 6.4e4]} \qquad (5\text{-}1)$$

As can be seen, the motor model is second order and linear. The specification bounds chosen for the step response are such that, ideally, the amplifier will not saturate when the desired tracking is obtained. This was achieved through analysing the maximum speed to which the system could respond, assuming limited voltage bounds. With the parameter variation, the fast poles varies from around -50 rad/s to -110 rad/s. This is a large model uncertainty to take into account and should test the controllers significantly.

The QFT method is ideal for this plant as it is already linear, and the specification bounds have been clearly defined. The bounds are summarised below :

- Tracking bounds : $TF_L(s) \le |T(s)F(s)| \le TF_U(s)$ (5-2)

$$\text{where } TF_L(s) = \left|\frac{100^2}{s^2+2.(0.85).(100)+100^2}\right| \qquad (5\text{-}2a)$$

$$\text{and } TF_U(s) = \left|\frac{150^2}{s^2+2.(0.65).(150)+150^2}\right| \qquad (5\text{-}2b)$$

- Robust stability : $|T(s)| \leq 1.2$ (5-3)

The final controller is :

$$G(s) = 0.295 \frac{\frac{s}{127.7} + 1}{\left(\frac{s}{5461} + 1\right)\left(\frac{s}{8357} + 1\right)} \quad (5\text{-}4)$$

5.2.1.3 Fuzzy specifications and design

The fuzzy system specification bounds are given below :

- Tracking bounds : Track a 10Hz sine wave (amplitude 180°) to within 5 %.
- Step Response : Reach 5% of set point value within 0.07s ($\tau \approx 0.026$s).

The tracking bounds are a test of the bandwidth of both systems. The QFT bound is such that the servo motor will track a 10Hz wave as well. The step response bound is a qualitative interpretation of the QFT bound.

The controller choices and control surface are given in Appendix C. Centroid defuzzification was used as the output is smoother than that obtained from other methods.

To determine the rules, knowledge of the servo motor's response to certain voltages is required. This qualitative knowledge was obtained from step tests on the simulation model. The operator knowledge together with logical reasoning produced the FAM bank.

Increasing the voltage in any direction (positive or negative) causes the motor's rotational speed to increase, therefore causing the angular position to change

faster[1]. The rules, based on this model can be generated. E.g. If the error *is* positive and large (PM) and the servo motor is moving to increase this error (PM or PS), then make the voltage such that it will change the direction and reduce the error (PL[2])

5.2.2 CSTR plant

Aim : To test the controllers on an unstable and highly non-linear plant.

Plant Input : Coolant flow

Output : Temperature of internal fluid

Disturbance : Temperature of incoming M reactant.

5.2.2.1 Plant model

The plant model for the QFT design case is described in Chapter 3. For the fuzzy design, however, the model needs to be converted into a qualitative one such that control rules can be obtained. The qualitative model of the CSTR *is* harder to obtain than for the servo motor model. This is due to the operating point being unstable. The operating point is obtained from examining the behaviour of the plant. At temperatures above P_2 (see Chapter 4), the temperature rises to a stable point, and below P_2, the temperature drops to another stable point. From this, an approximate operating point can be obtained. The maximum and minimum

[1] This is a qualitative interpretation of (5-1).

[2] Due to the convention used, a PL voltage results in a PM error eventually.

coolant flowrates[1], and the possible temperature values are used to determine the input and output bounds.

5.2.2.2 QFT design

The plant with parameter variation can be summarised in the form of a transfer function as :

$$P(s) = -\frac{[8.8431e1\ 1.1080e2]s + [4.8159e(-1)\ 5.6840e(-1)]}{s^2 + [8.5546e(-3)\ 9.5561e(-3)]s - [1.3253e(-6)\ 3.21114e(-6)]} \qquad (5\text{-}5)$$

Note : Pole-Zero reduction has been performed on the above plant to reduce the order of the plant. It should also be noted that the plant has a negative gain. This is expected as an increase in coolant flowrate cause a decrease in reactant temperature. In the design of the controllers, this negative sign is ignored and instead, the gain signs are interchanged at the feedback error calculation.

The plant system is slow with a step time constant in the region of 15 minutes. The QFT design procedure followed involved choosing the specification bounds and designing the controller. If the specification bounds were too tight and the designed controller saturated the valve too quickly, the bound specifications were reduced until the control action was physically realistic. This specification method is used such that the controller design could be considered the best

[1] The maximum and minimum flow rate are not the bounds of the fuzzy output range as the change in flow rate is the output. These values serve to indicate a rough value for the maximum rate of change which is used.

achievable with a linear controller under the given conditions. This is important as the fuzzy controller does not use bounds for its design as does the QFT, therefore an optimal linear controller design is required for a comparison. The specification for the controller design are :

Robust output disturbance bound : $|S(s)| \leq \left| \frac{s^2 + 0.101s + 1 \times 10^{-4}}{s^3 + 2.01s^2 + 1.02s + 1 \times 10^{-2}} \right|$ (5-6)

Robust stability bound : $|S(s)| \leq 1.2$ (5-7)

The robust stability bound corresponds to a phase margin of approximately 46° and a gain margin of 15dB. The gain margin is large as the CSTR is a highly non-linear system, and this specification will enforce a higher robustness to model parameter variations. A steady state error is allowed as this problem is a disturbance rejection one and not a tracking problem. Thus no integrator is required in the controller.

The plant is obviously unstable (due to the positive pole), and hence stabilising the CSTR is a primary objective. No input disturbance rejection is specified, as a specification on the output disturbance bound places limitations on the input bound. Any input disturbance can be converted into an output disturbance, so this is not seen as a limiting choice.

The controller designed is :

$$G(S) = 0.001789 \frac{1}{\left(\frac{s}{1.321} + 1\right)\left(\frac{s}{3.631} + 1\right)} \quad (5\text{-}8)$$

The Nichols plot of the plant, with the controller is given in Appendix C.

5.2.2.3 Fuzzy Design

The fuzzy system is well suited to this system and the generation of the required rules was easily achieved. The bounds for the CSTR are hard to quantify in terms of a fuzzy controller, but they are given below :

- Stable
- Disturbance rejection : minimise the effect of all disturbances to within 1 % of the set point within 100 seconds.

The disturbance rejection bound is a qualitative interpretation of the same bound (in the frequency domain) given in the QFT design. This is obtained from mapping the frequency domain bound into the time domain and examining exactly what is required.

The choice of fuzzy system and the control surface is given in Appendix C. This particular problem cannot be solved by a fuzzy system alone. If the output of the fuzzy system is defined as the actual flowrate, the problem cannot be robustly solved, for if the equilibrium point coolant flow rate is not constant and exact, the fuzzy output is useless, and oscillations will result. The fuzzy output must therefore be an incremental change in the flow rate. For this to be realisable, an integrator is required after the fuzzy system, and hence a dynamic system is required.

The rules chosen for this problem are logical and based on how the CSTR responds to changes in coolant flow (step tests).

An increase in coolant flow causes the temperature to drop quickly. Therefore the following rule would be logical : If the temperature is too high (error is PM) and there is no change in the temperature, then the valve should be opened fast (open fast), such that temperature drops (increase coolant flow).

5.2.3 Tank system

Aim : The aim of the tank system is to investigate the controller's ability to control a system with dead time.

Plant Input : Output flow rate.

Output : Level of water in the tank.

5.2.3.1 Plant model

The plant model for the QFT design case is described in Chapter 3. The fuzzy model is easy to determine since the system's response to change in input is obvious (see section 5.2.3.3).

5.2.3.2 QFT design

Numerous possible equilibrium points for different levels in the tank are used to obtain the parameter variation in the plant. The controller is designed for the tank model in (4-13) with dead time added (t_d=5s). The dead time introduces significant lag onto the system. The dead time has converted a first order system which is easy to control into a complex and difficult control problem. Like NMP systems, the dead time limits the maximum gain which can be added to the system. The specifications only call for a stable system with as much open loop

gain as is permissible. Meeting any additional specifications becomes secondary to stabilising the system. The final controller is :

$$G(s) = 0.03115\frac{\frac{s}{0.0229}+1}{s\left(\frac{s}{1000}+1\right)} \qquad (5\text{-}9)$$

5.2.3.3 Fuzzy control

Like the QFT system, the aim with the fuzzy controller is to stabilise the system and yet provide a fast response with as little oscillation as possible. Steady state error was allowed in the system to try and prevent oscillation when the system was near the equilibrium point. The rules were obtained though logic, but trying to include the time delay into a rule is very difficult. The rules are typically independent of time[1] and therefore taking time into account is difficult.

An increase in flow rate means that the tank will empty faster than before, and the level will drop. A suitable rule would therefore be : if the level is too high (NM) and there is no change in the level, then open the valve fast (open fast) to bring the level back down. As can be seen, the linguistic term "open fast" must take the time delay into account.

[1] Unless a clock becomes the input to the fuzzy controller.

5.3 Controller design and comparison for specific aspects of plants

The plants discussed in this section are theoretical plants which are included to illustrate certain aspects of the control problem. Although the plants are theoretical, they do represent physical plant characteristics.

5.3.1 Non-minimum phase plant

Designing a linear controller to control a non-minimum phase plant in a feedback loop is difficult. This is a system with a positive zero. High gain causes the stable poles to move towards the positive zero, which causes instability. To illustrate this concept, consider the following plant,

$$P(s) = \frac{(s-7)}{(s+3)(s+4)} \tag{5-10}$$

Figure 5-1, shows the Nichols plot of the system.

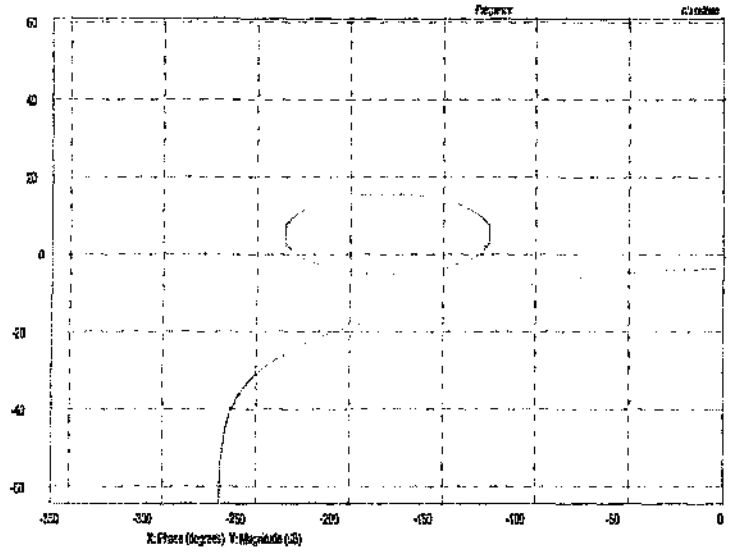


Figure 5-1 : Nichols plot of non-minimum phase plant

The RHP (right hand plane) zero has the property of providing lag with magnitude increase. Therefore if we try to reduce the lag through adding a LHP zero, the resultant is a system with finite gain at infinite frequency (unrealisable solution). Adding more LHP poles to prevent this results in too much lag. The result is that a RHP zero provides an upper limit on the gain crossover frequency ω_c and the low frequency feedback benefits are minimised.

Examining the plant given by (5-10), the maximum gain which can be applied is 7 before the system becomes unstable. A gain of 2.69 is required if the robust stability margin of $|T(s)| < 1.2$ is to be achieved. The steady state error for the system with a gain of 5 (small oscillations, but stable) is still around 25%, which is unacceptably high. Loop shaping is possible but as discussed before, it is extremely difficult with little benefit. Adding an integrator introduces another phase lag of 90°, therefore restricting the maximum gain to 1.7. This ensures no steady state error, but oscillations are present. The NMP has limited the ability to achieve the maximum benefits of the linear controller.

The fuzzy controller approach is well suited to this problem. Due to the fuzzy system's non-linear characteristics the problem is avoided. A fuzzy controller for

the above system is easy and quick to design. The knowledge which was used in the design was the black box step test. From this, the only indication that the system is more complex than a first order, or damped second order system, is the initial negative response of the system. Using the time constant of the system, ranges for the change in error are quantified, as well as the error range and the control action range. The rules are straightforward and logical. It is assumed for this problem, that saturation is not present in the control action. The output of the fuzzy controller is actually a change in the required control action, hence, the output is passed through an integrator before reaching the plant (see section 5.2.2.3 on the CSTR). The initial fuzzy system was conservative and a slow step response resulted. After a few iterations, the step response had an acceptable rise time with zero steady state error. The fact that the system is NMP had little effect on the controller design.

No literature has been found on applying the fuzzy control technique to NMP plants. As has been shown, fuzzy control may provide a quick and accurate tool for control of such plants and should be investigated with further research. The controller information is given in Appendix C.

Note : the fuzzy controller is not the optimal, nor does it ensure the fastest response. It merely illustrates the point that this is a situation ideally suited for a fuzzy controller. The response can be made faster (essentially, as the gain is non-linear, the step response time constant could be made infinitely small, assuming no saturation). The problem, also found with the servo motor controller, is that the fuzzy controller becomes difficult to tune when very fast responses are required, as the overshoot must be prevented. The aim of this problem is to show that the problem of NMP which complicates linear controller design can be easily overcome due to the non-linear nature of the fuzzy controller.

The step responses for the two controllers are given in Figure 5-2 :

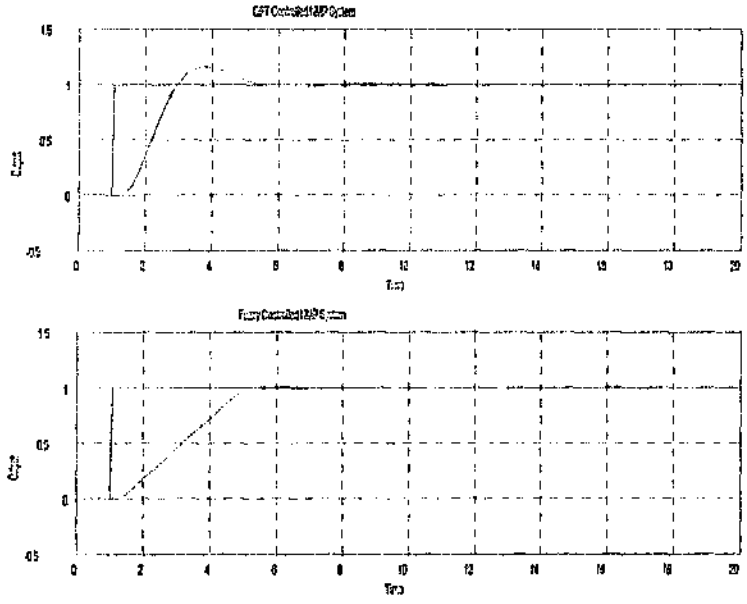


Figure 5-2 : Step response of the QFT and fuzzy controllers for a non-minimum phase plant.

5.3.2 Highly non-linear plant

A highly non-linear plant (asymptotically unstable) is :

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u + P.x_2^3 \end{aligned} \qquad (5\text{-}11)$$

where $P > 0$.

The system is highly unstable and approaches infinity in under 2 seconds (depending on the value of P). The plant is easily linearised, but analysis of the linear plant around the operating point (x_1=0), shows that the variation in the model behaviour for different initial conditions is large and hence the QFT technique requires use of an LTIE set[67]. The plant is, however, easily stabilised using state feedback.

Designing a fuzzy controller for this plant is not easy. A fuzzy controller for the situation when P=1, was obtained and this stabilises the plant effectively. It is not possible however, to stabilise the plant when P=10 as the plant instability is too fast, and it is difficult to track, "catch" and counter the response fast enough. Theoretically, the fuzzy system should be able to stabilise the system, but the design is difficult.

The conclusion reached from this controller design is that if the plant can be accurately modelled, linear feedback control, possibly with state feedback, offers a better solution. The system could arguably be controlled through fuzzy control, if access could be gained to the inner parts of the system that is state feedback with fuzzy control.

5.3.3 Hybrid control of a first order plant with dead time

Dead time is a major problem in physical plants, and is present in numerous industrial process plants. The presence of dead time necessitates the lowering of the controller gain to maintain stability. Many methods have been proposed to counter dead time, including :

- Smith predictor[58]
- analytical predictor[59]
- general purpose algorithm (Gautam and Mutharasan)[59]

The design proposed in this section is a hybrid controller comprising a Smith predictor which is tuned by a fuzzy controller.

The Smith predictor is the best known dead time compensation technique and is extensively used in industry[58,59]. It is model based and should ideally be

implemented digitally. The development of the theory of the Smith predictor is not given here, but Deshpande and Ash[52] et al give a good summary.

Consider a first order plant with dead time :

$$P(s) = \frac{K_p . e^{-t_d s}}{\tau . s + 1} \qquad (5\text{-}12)$$

where K_p is the process gain

t_d is the dead time and

τ is the process time constant.

This system could represent numerous physical systems including a tank level, conveyor system or a stirred tank (temperature).

MacLeod[45] developed a hybrid fuzzy-PID Smith predictor system which used a similar first order system (conveyor belt). His fuzzy system tuned the gain of the PI controller depending on the actual plant gain. The method of control posed in this investigation is similar. Although the Smith predictor is a powerful control technique, it does have problems. If the dead time is not known accurately and hence differs from the actual dead time of the plant, then the system performance degrades, and becomes worse the more inaccurate the estimate is. In order to improve the robustness of the Smith predictor, it is proposed that a fuzzy system be used to modify the dead time of the predictor such that it minimises the delay error. A block diagram of a Smith predictor, including the adaptive fuzzy proposal, is given in Figure 5-3.

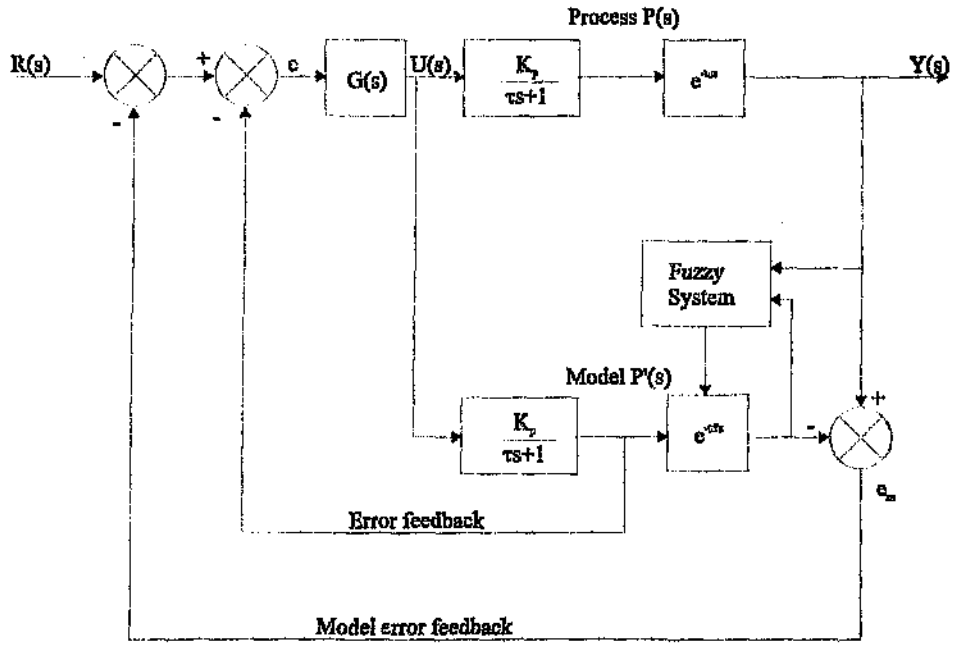


Figure 5-3 : Adaptive Smith predictor control scheme

The problem with tuning the dead time is that if the system is in steady state, the effect of the discrepancy will not be noticed until there is a step change or a disturbance. Therefore, in order to tune the system, a series of step changes is required. It is from these step changes that the tuning occurs. The fuzzy system has two inputs, namely the plant output and the predictor output (both after the delays). The output of the fuzzy system is the incremental change in the predictor time delay estimation. The principle on which the system works is :

- The system must be near steady state and be stable. This means that the predictor must have a reasonable guess of the actual delay, such that the system can move to a steady state condition.

- When a tuning step is applied, the plant output and the predictor output are examined by the fuzzy controller.

- The actual magnitude of the outputs is not important provided the step is above the specified noise threshold value in the fuzzy system. This means that

the system need not be in perfect equilibrium and hence disturbances and noise will not affect the tuning of the system provided they remain below this noise threshold value. What is important is the timing of the two inputs. If one of the step responses occurs before the other, then the predictor value of the time delay is incremented accordingly.

- The system will also be independent of changes in the plant parameters. As discussed above, it is only the timing and the threshold values which are important.

- This system will only change the time delay on the positive part of the step. This can be changed if a faster tuned system is required, but the system complexity will increase.

The method of tuning the system is not seen as impractical. It is obviously suited to a system which is continually changing for example a tracking system. The tuning mechanism can be made to work with other forms of signals including a sine wave. In a disturbance rejection problem, the method proposed is aimed at a plant where the operator has noticed that the system performance is not optimal, and by running the step tuning, they can readjust the dead time compensator. The changes in the actual dead time value will only occur when the tuning is started and not during normal operation. Unless the disturbance can be measured, there is no inherent method of tuning the dead time. If the disturbance can be measured, then a system similar to the one above may be possible, where the disturbance forms an input to the fuzzy system.

The results of the simulations were excellent and showed the fuzzy system tuning the dead time accurately. It must be remembered, however, that this is a simulation and the plant is only a first order system.

Figure 5-4, shows the step tuning for the system. The actual dead time is 6 seconds, but the initial predictor estimate is 4 seconds. The severe oscillation for $100 < t < 200$ indicates the problem inherent with Smith predictors when the

dead time estimate is not accurate. The fuzzy system manages to tune the system despite the severe oscillation.

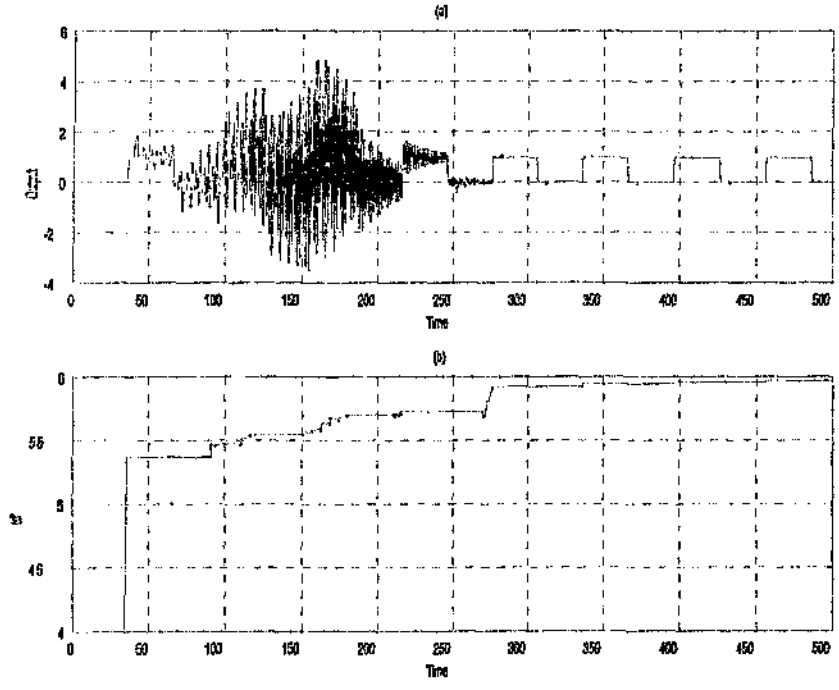


Figure 5-4 : Adaptive Smith predictor. (a) tuning steps (b) predictor dead time estimate td'

5.4 Simulation comparisons

Section 5.4 summarises the results of the comparisons between the fuzzy and classical controllers. The comparisons are limited to specific control design aspects, and a more general comparison about design issues is given in section 5.5.

5.4.1 Tracking

Servomotor : This forms the basic test for the comparison. The QFT controller performs deterministically and is within the bounds as is expected for all of the plants. In contrast to this is the fuzzy controlled servomotor. While the nominal plant meets the specifications well (an even a better response than the linear controller), the performance variation of the other plants is significant, with some having large overshoot and others critically damped. This is obviously a great limitation of the fuzzy approach, as model uncertainty is very difficult to control within the specification bounds (and its effect is unpredictable). Although it is possible to ensure that it is within the bounds, it takes a large amount of time to tune the controller to that degree without using some form of training mechanism (e.g. neural networks).

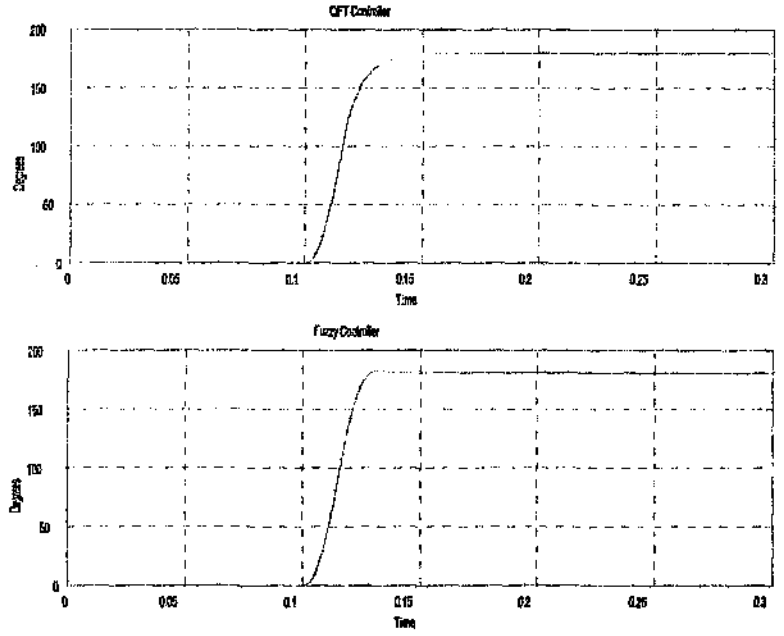


Figure 5-5 : Step response test

From the previous test it is possible to see that in the step response, it is possible for the fuzzy controller to outperform the QFT on certain plants. This is due to the non-linear gain. An anomaly was noticed in the sine wave tracking test (180 degree 10 Hz sine wave). The QFT controlled servomotor was within (1.5% or 2.5°), while the fuzzy controller could only manage (22% or 40°). Up to now most papers[21,60] use the step response as the definitive test in showing fuzzy to be superior to linear controllers. From the plots shown, it can be seen that in the case of a fuzzy system, just because the step response is within the required bounds, the sine wave response is not predictable. The linear controller offers predictable behaviour at various frequencies which is lacking in the fuzzy approach. No literature found seems to have made a point of investigating this issue. This problem is relevant especially in a tracking problem, such as this one.

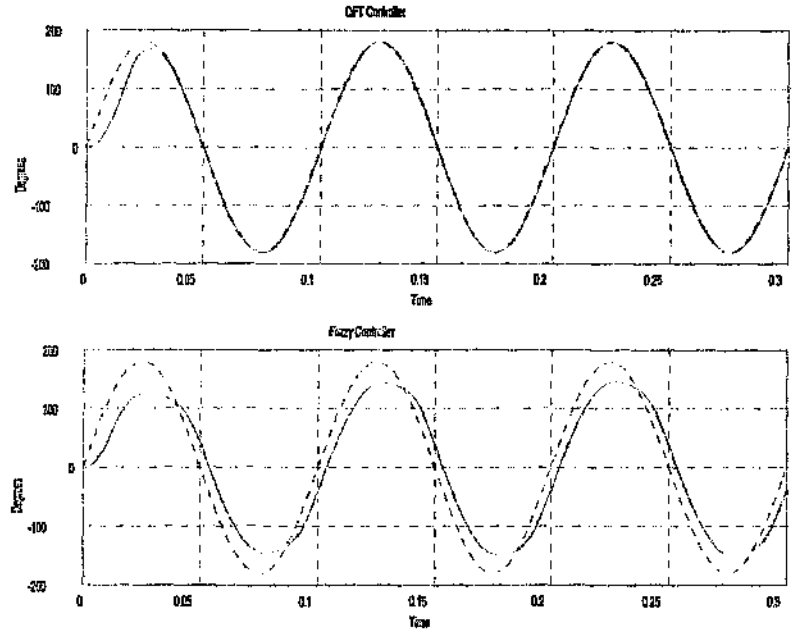


Figure 5-6 : Sine wave tracking test

CSTR : While the problem is inherently one of disturbance rejection, the step response forms a common and simple method for comparison between the two systems. The fuzzy controller showed an excellent step response with a fast rise time and no oscillation, even outside the operating point. What should be evident in the response, however, is a "bend" in the step response when in the vicinity of the set point. This is due to the "braking" rules being fired, which slow down the response when approaching the set point. The QFT, as expected, works well within the operating point bounds, but oscillates outside this point. This problem highlights the advantage of this fuzzy approach, namely that the rules can take in a wider operating region. With QFT control, all the operating regions must be included in the design stage. The precise bounds of the operating region are not clearly defined in the fuzzy design approach[1] whereas the operating region bounds in the QFT design are clearly defined before simulation.

[1] Unless a "robust" fuzzy design technique is used.

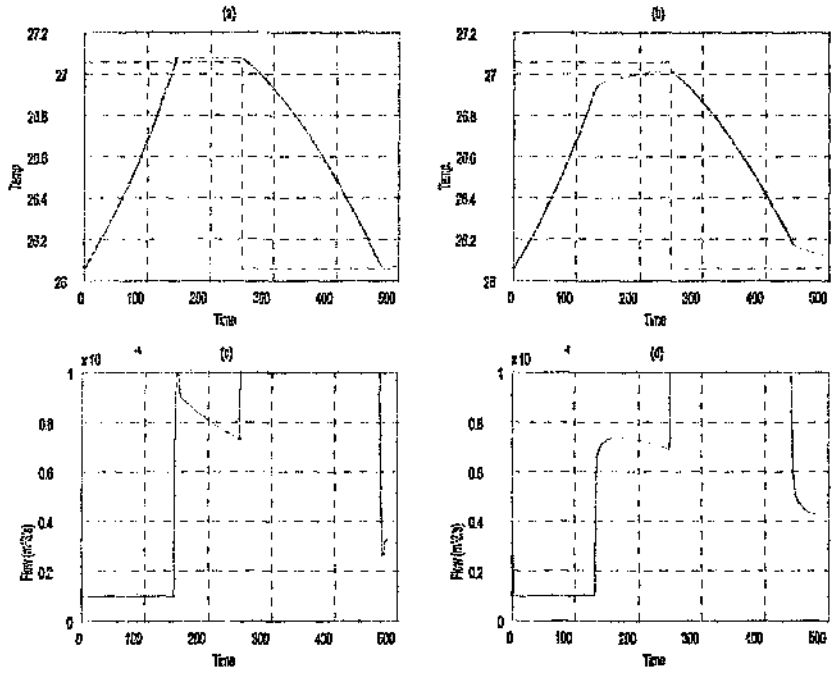


Figure 5-7 : Step test on CSTR.
(a) - QFT controller step response;
(b) - fuzzy controller step response;
(c) - control action for QFT controller;
(d) - control action for fuzzy controller.

5.4.2 Disturbance rejection

CSTR : The disturbance applied is a step in temperature of the M reactant inflow. Both controllers rejected the disturbance well. The QFT controller settled to a new operating point within a few minutes, but the new point was 0.02 degrees (0.001%) off the set point. The fuzzy controller limited the set point change to within 0.003 of a degree before settling down slowly. Thus it can be concluded that both controllers rejected the disturbance (which was a significant one) effectively.

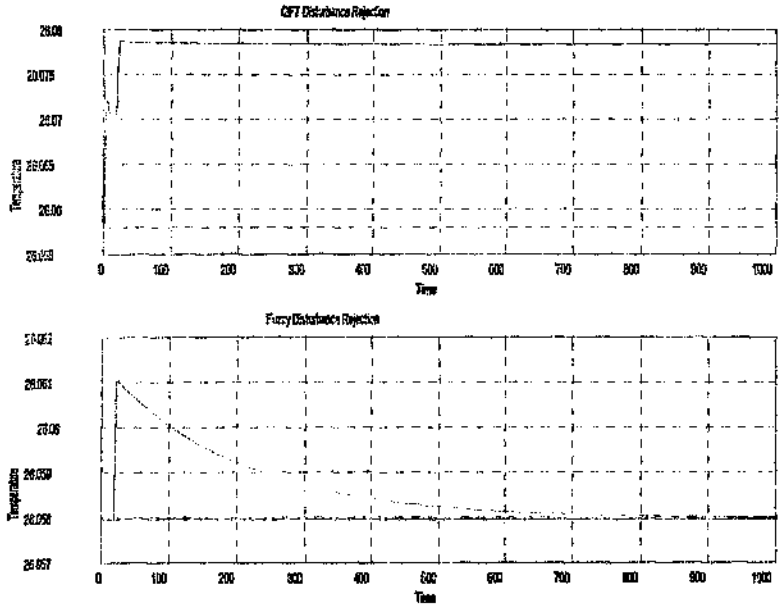


Figure 5-8 : Step disturbance rejection test results

A time varying disturbance was also tested. Again the temperature of the M reactant inflow was changed through a 0.01Hz sin wave of magnitude 2°. Again both controllers maintained the CSTR within acceptable bounds, with the fuzzy controller deviating slightly less.

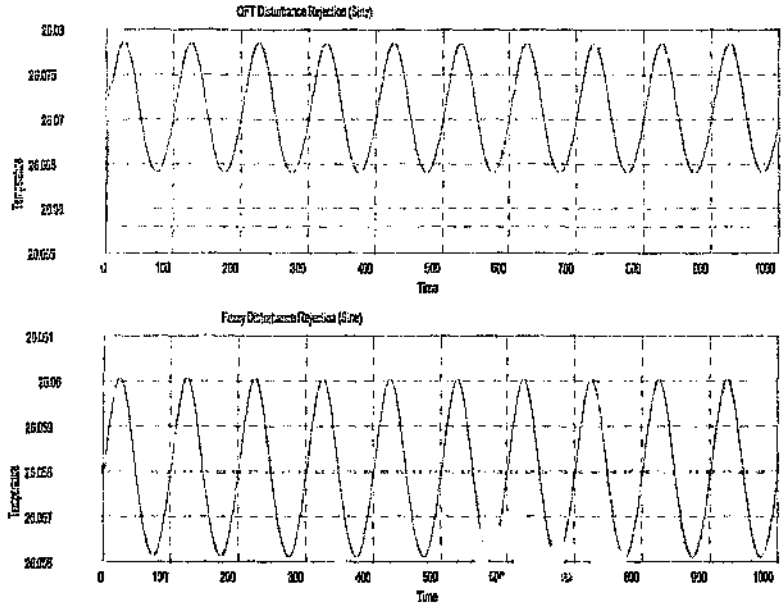


Figure 5-9 : Time varying disturbance test results

5.4.3 Noise suppression

Servo : Normally distributed band limited noise was used in the tests. Various frequency bands were used from 10Hz to 200Hz. In all cases, the fuzzy system exhibits slightly better noise rejection than the linear system does. The improvement varies from negligible (25Hz-150Hz) to a 50% reduction in response (100Hz-200Hz). It is possible that each controller could exhibit better behavi · · if designed differently, but it is believed that fuzzy control will allow better noise rejection due to its non-linear nature which allows more flexibility with scheduling the gains.

Figure 5-10 is a plot of both controller's noise response (50Hz→150Hz) of mean zero and standard deviation 0.4023. The QFT controlled system has a high bandwidth which results in the noise having a significant effect on the system output. The noise causes the QFT controller to continually saturate the amplifier.

The fuzzy controller reduces the noise a little better, and the amplifier saturation is better controlled.

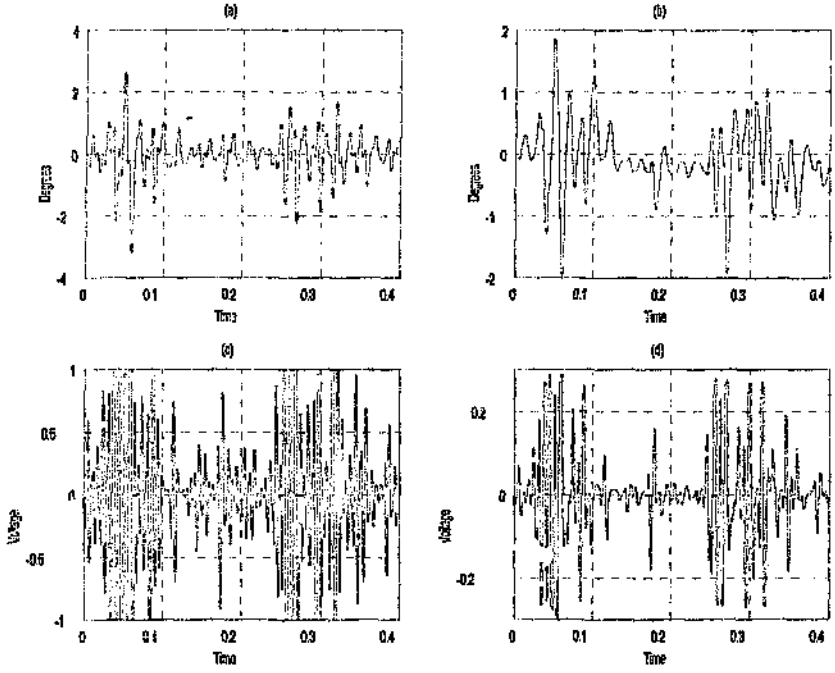


Figure 5-10 : Servo noise suppression test results
(a) - QFT controller response;
(b) - fuzzy controller response;
(c) - control action for QFT controller;
(d) - control action for fuzzy controller.

CSTR : To test the controller's response to noise, band-limited (0.1Hz-1Hz) noise with a mean of zero and a standard deviation of 0.0418 is applied to both systems. The QFT system deviates slightly (0.0273°) from its steady state position, but shows a very irregular response with a large amount of oscillation. The fuzzy controller deviates slightly more (0.0977°), but shows a much smoother response. This is due to the ability to schedule the gain around the operating point. Thus in this case, the gain is kept low, so although the deviation might be greater, the overall response is smooth and does not continually saturate the control device as shown in Figure 5-11. This helps to reduce the stress and load on the final control device. The QFT controller continually saturates the control device, while the fuzzy controller has a much smoother control action in the presence of noise. The

windowing[1] technique offers a possible method of reducing the systems noise sensitivity, as the gains around the operating point can be accurately set up to "filter" out most of th : noise. This offers a possible future research topic.

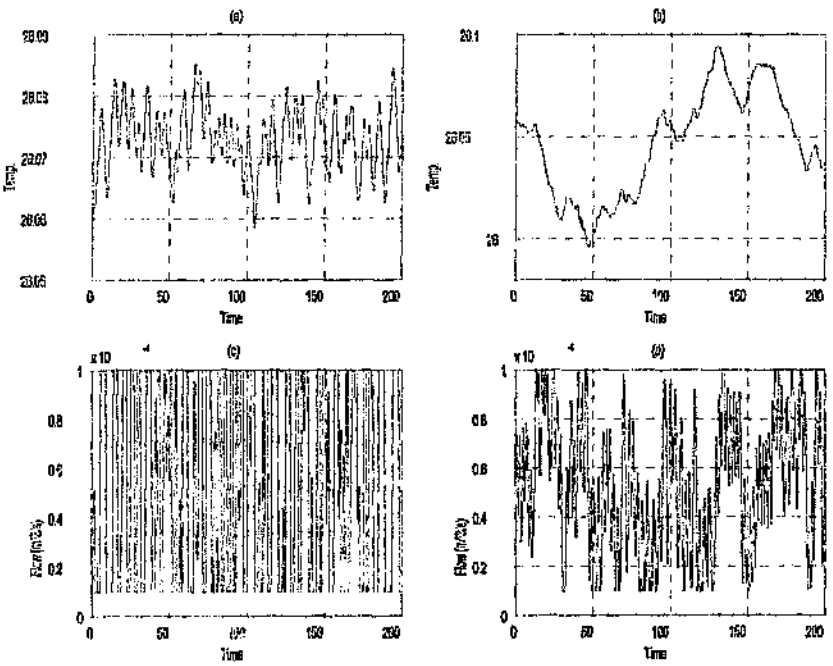


Figure 5-11 : CSTR noise suppression test results
(a) - QFT controller response;
(b) - fuzzy controller response;
(c) - control action for QFT controller;
(d) - control action for fuzzy controller.

5.4.4 Unmodelled dynamics/model variation

CSTR : The unmodelled parameter variation test involves setting a new operating point for the system. Both controllers manage to achieve this for the nominal plant, but the fuzzy system does not have any oscillations. Thus this advantage of

[1] The windowing technique uses (at least) two fuzzy controllers. The first controller is used when the error is large and therefore gives coarse control. When the error is below a certain threshold, control is passed over to the other fuzzy controller which administers fine control over the plant. This allows individual controllers to be optimised for different areas of the operating region.

the fuzzy system is illustrated, namely that it can be more robust over a great variety of conditions as it is not designed for a specific case as is the QFT controller[1]. This has a great implication when the uncertainty is large or the ideal operating point of a system is not known. Both systems had trouble controlling the plant when large unmodelled parameter variation is used or there are large set point changes (QFT can handle this, but must be explicitly designed for it). Analysis of the characteristic curve indicates that the system is very non-linear at these points (small changes in coolant flow cause large variations in temperature) and the controllers tend to overshoot due to this problem. The fuzzy system cannot deal with this situation well as the required actions are different when approaching the extremities of the curve. Start up and shut down of a system like this would normally be done by an operator, or it could be performed by another fuzzy system. This demonstrates the area when fuzzy system design becomes problematic. If the rules governing the control of a system are not constant, but change depending on the system state, then multiple fuzzy systems must be used.

5.4.5 Non-minimum phase

The non-minimum phase plant was investigated in section 5.[illegible]. The outcome of this investigation is that the fuzzy controller offered an excellent method of overcoming one of the limitations of classical control. The rules which are used for the non-minimum phase plant are similar to those for a minimum phase plant, and hence the design was relatively simple.

[1] The problem with the fuzzy approach, however, is that the response is not as predictable as it is with the classical approach.

5.4.6 Dead time

Tank : The tank system developed is used to investigate the control of dead time. The classical system is straightforward to design, but as expected, the feedback benefits are severely limited by the nature of dead time (phase lag). The fuzzy system, however, is not as easy to design. Defining the rules for a system with dead time is harder than for a system without dead time. This is logical since it is easier for an operator to control a simple process with no dead time, and hence reduce the action to rules, than it is to control a simple action with dead time. The actual rules become difficult to define and are plant specific. If the delay is a certain length, then the braking rules need to be defined such that they are fired when the plant reaches a certain part in its *Error/dError* trajectory. If the plant changes, the required position of these braking rules changes and hence oscillations occur. In the classical case, plant variations can be easily taken into account, although the variation cannot always be reduced to the desired levels. This can be seen with the fuzzy controller. The level for which the fuzzy controller is designed works well (but allowed a steady state error), but when a different level set point is specified, then oscillations are present. Neither controller is robust or offers any meaningful control. The classical approach has the advantage of a clear representation of what the controller will achieve, while the fuzzy approach allows non-linear gains to achieve better performance. The conclusion reached from this is that neither system is suited and a time delay compensation technique is required.

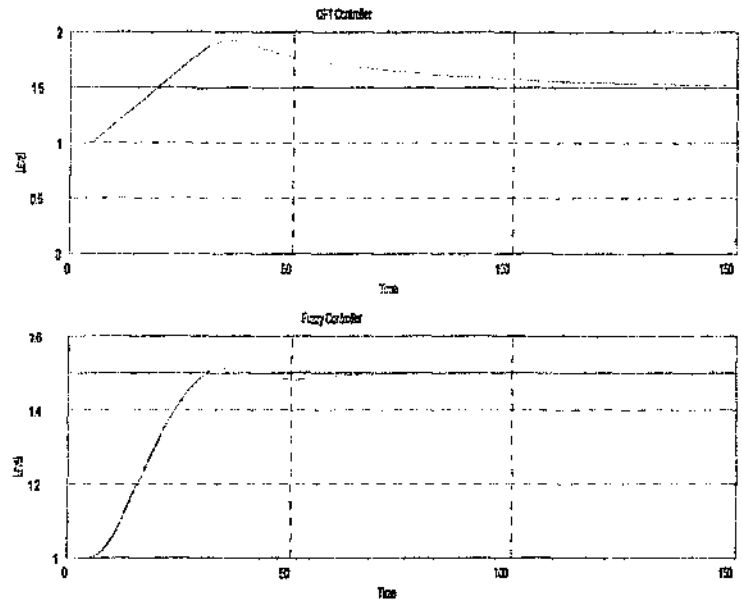


Figure 5-12 : Dead time step response test

Hybrid : As discussed earlier, the hybrid system offers exceptional robustness in a dead time situation, and therefore shows a good combination of qualitative and quantitative knowledge in the controller design. The control offered is far superior to either of the systems on their own, with a little added complexity.

5.5 Design Comparison

The following is a summary of the similarities and differences between the classical and fuzzy control design techniques.

Table 5-1 : Comparison between fuzzy and classical control design

Aspect	Classical Design	Fuzzy Design
Frequency information	The classical design technique uses frequency information directly in the design through transfer functions[1].	The fuzzy system offers no method of incorporating frequency information directly into the design. Should there be some specific information which is important and must be included, external dynamic elements are required (frequency analyser, differentiator etc.).
Parameter variation and uncertainty	A classical design technique such as QFT allows structured and unstructured uncertainty to be easily taken into account.	Fuzzy control is now starting to formally be able to handle parameter variation through robust design techniques (using Sugeno style fuzzy systems) and using fuzzy numbers. Specification of the fuzzy sets and numbers is not definite and hence the experience of the designer plays a large role in the quality of the fuzzy controller. The fuzzy number method of handling uncertainty is not well used, and instead the uncertainty is incorporated into the fuzzy sets on the universe of discourse.

[1] Or a variation of them e.g. state space.

Table 5-1 (cont.) : Comparison between fuzzy and classical control design

Design Choices	Clear and definite design option.. with well documented implications characterise the classical design technique.	The design options in a fuzzy controller are numerous and their effect ill-understood. This could be interpreted as a fuzzy controller having greater flexibility, and this is true, but their effects are not easily predictable.
Optimal controllers	In the QFT approach the optimal controller is one which meets all the specifications exactly and "hugs" the required stability contour. Thus the optimal controller is clearly defined, although not always obtainable.	There is no optimal controller in a fuzzy system (if complexity is ignored). In theory, the fuzzy controller can provide control which is optimal, in the sense that no other form of controller could be better (but could be equal). The problem is that the number of fuzzy sets and hence rules will tend to infinity.
Input and output ranges	This information is not always known, but should be, when designing a classical controller. This information is essential to be able to take into account saturation.	The input and output ranges of the plant, and hence the controller must be known apriori. While these can be tuned later, a good estimate is required. Cut off filters are required to ensure that the signals are not outside the fuzzy controller's defined range.

Table 5-1 (cont.): Comparison between fuzzy and classical control design

Operating region behaviour	A mathematical model is required, and therefore, if well modelled, all dynamic will be taken into account.	Selection of the operating regions must be carefully made. All dynamic elements must be included or incorrect rules may be used at inappropriate times. All encompassing rules are only then possible for when the error is large. It should be noted that these rules and information are not always easily obtainable from an operator for numerous reasons[61].
Saturation	In the classical approach, the level of saturation must be known and the bounds designed in accordance with this.	Saturation of the control element is not a problem when designing a static fuzzy controller. When designing a dynamic controller (with dynamic external elements) and the output is a change rather than an absolute control action, then saturation can be a problem. The rate of change can easily be changed and therefore the designer does have complete control over the speed at which the control device will saturate.

Table 5-1 (cont.) : Comparison between fuzzy and classical control design

Stability	Stability in linear classical control is well documented and the theory is well understood.	Achieving a stable controller is generally easy, but obtaining the desired levels of performance is often difficult. The servo motor illustrates this problem, where the tracking bounds were very hard to meet, but stability was easily achieved. Designing to ensure stability, without simulation, is difficult and several methods have been proposed[39,40 et al].
Gain Specification	Using a linear controller, the gains in certain regions of the operating space cannot easily be achieved and gain scheduling is typically used.	The fuzzy system offers the ability to specify gains at specific points in the transient trajectory of the system. Therefore, if the *Error/dError* trajectory map is analysed, and portions of slow movement are evident, the gain in that region can be increased. Although this works well in theory, achieving this level of proficiency requires careful planning and experience.

Table 5-1 (cont.) : Comparison betwee fuzzy and classical control design

Plant non-linearities	The classical controller is typically linear and therefore cannot cater for non-linearities in the same manner[1].	The fuzzy system offers the ability to cater for non-linearities in the system. e.g. the tank system will empty and fill up at different rates. This effect can be accounted for by changing the output sets for opening and closing.
Noise sensitivity	As discussed in section 3.2.3, *T(s)* must be kept small outside the control bandwidth to reduce the effect of the noise.	If the gain around the operating point is reduced, the fuzzy system can reduce the system's sensitivity to noise and disturbance (e.g. use a windowing technique).
Steady state errors	The classical approach will have a steady state error., unless an integrator is used.	A typically designed fuzzy system will not allow for a steady state error unless the designer wishes one.
Design model information	A mathematical model is required for classical design. Qualitative knowledge is not easily incorporated.	Little mathematical model information is required about some plants (but not all) for a controller to be designed. Qualitative operator knowledge is required. This is illustrated by the CSTR which was stabilised very easily. The rules were obvious and simple. Yet the control achieved was superior to the classical with a fraction of the design time.

[1] QFT can handle non-linearities through multiple linear plants.

5.6 Advantages and disadvantages of fuzzy control

From the above comparison the benefits of fuzzy control can be summarised as follows :

- Fuzzy controllers can be highly non-linear. This means that the closed loop system can achieve performance unmatched by a linear classical controller. The interface to this non-linear control is easy to understand (i.e. rules) and offers an intuitive method of implementing powerful control over a system.

- The non-linear nature means that the fuzzy controller is not bound by the same restrictions that a linear controller is. This involves the inclusion of plant non-linearities (e.g. saturation, slew rate) into the design as well as linear performance restrictions (there is no response time/overshoot trade off).

- The ability to implement local control. Each area of the fuzzy control surface can be optimised to suit the specific state that the plant is in. This differs from the classical approach where one controller must suit all states (unless gain scheduling is used).

- Ability to capture operator knowledge. This is possibly the greatest asset of fuzzy control. Using classical control, only numerical knowledge can be used in the design. With the fuzzy approach, qualitative knowledge can now be included in the controller. This has implications when the qualitative knowledge about a plant is comprehensive, but little is known quantitatively. This qualitative knowledge has significant value in monetary terms, and a method of using this could save significant design costs. There is no need to pay large sums of money to develop a plant mathematical model, when a qualitative one already exists.

- Noise suppression and disturbance rejection can be better handled. Fuzzy controllers do not inherently reject noise better than the classical controller,

but they do have the ability to. This is derived from their inherent non-linear nature.

- Normally, when a fuzzy controller is designed, it ensures that the system will not have steady state error. This is in contrast to the classical problem where high gain, or an integrator, is required.

- There are methods of using fuzzy logic systems to model a plant. Using the inverse of this fuzzy model presents a method to apply control to the plant.

- Fuzzy controllers can implement complex objectives well. If a cost function needs to be minimised or another supervisory approach is required, the fuzzy controller is well suited. This is because supervisory objectives can easily be reduced to a rule form. The higher the level of controller required, the more complex the objectives become which increases the benefits which fuzzy control has to offer.

- High level adaptation of other controllers is possible. When control engineers are designing a classical controller, they make choices based on their knowledge of the plant. These choices can often be reduced to rule form which can then be used to implement on-line changes to another controller. The increased control performance comes from the ability to add more information to the existing form of control.

- Adaptive fuzzy systems offer high robustness as they can change their structure as the plant parameters change. This form of dynamic control requires a very basic model of the plant, while fine tuning can then be performed while the plant is operating.

- The "fuzzy" nature of fuzzy control allows imprecise concepts and ideas to be transformed into meaningful control actions which can be effectively used on a plant. This is particularly important when the operator's knowledge is vague and imprecise.

- The control designer does not require detailed control theory to design a workable fuzzy controller. This offers opportunities for less qualified personnel to design controllers. The easy interface to the control concepts provides a better situation for the operator to become involved in the design.

- The effect of external influences on a plant are not always easily mathematically definable. The fuzzy controller, using more sensors, can use information about external influences to provide better control through the use of more information which was ignored before.

The following disadvantages have been identified :

- There are very few analysis tools available. The biggest issue related to this is the stability problem. Although methods have been proposed to address this issue[40,39 et al], no overall solution has been found. This means that when a controller is designed, the stability of the controller is always in question, as it has not been proved stable for all states and inputs.

- No direct method of incorporating quantitative information is possible. As illustrated by the servo motor where clearly defined tracking bounds are present, the fuzzy controller cannot include these bounds in the design. Repeated simulation is required to analyse the performance and ensure the closed loop system is within the specified bounds.

- Classical control is well defined and has a large body of theory and design tools supporting it. Fuzzy control has no transparent design tools supporting it and more work needs to be done to develop more design and analysis techniques.

- The design process relies heavily on the designer to make choices regarding certain aspects of the fuzzy controller (e.g. defuzzification method). The number of choices are vast, with little understanding of the implications of these choices.

- The design method is iterative and often time-consuming depending on the plant being controlled. Classical control design methodologies offer predictable controllers provided the modelling, non-linearity compensation and design philosophy have been followed.

- Including frequency concepts in the "fuzzy" domain is difficult. The fuzzy controller must be given the frequency information through a differentiator or similar device. This is in contrast to the linear classical controller, where the frequency response of the system can be well defined and information regarding this domain easily included.

- Adaptive systems require a disturbance to be placed on a plant before the adaptive system can make meaningful changes to the controller.

- If the fuzzy system is working well with one set of conditions, one cannot infer that it will work with another set (e.g. although the step response may be acceptable, the sine wave tracking response may not be).

- The system is far more complex than a simple transfer function controller. This means that implementation costs are usually higher, especially when external dynamic elements are required.

- The quality of the final controller is largely dependant on the quality of the designer. Two fuzzy control designers may come up with two controllers which may both work within the specification bounds, but be very different. The implication of this is that one controller may be more robust than the other without quantifiable reasons. The classical control situation is different. If the desired specification bounds are used in the QFT design process, the controller obtained will resemble another controller designed by the same process. The reason this is true is that the classical design technique (QFT) quantifies the bounds such that the required controller is one which meets these bounds. The fuzzy control design techniques does not offer an easy

method of including these bounds and hence the controller design level (under or over designed) is not easily determined.

- There is no best controller in fuzzy control. As the design process is iterative, the controller can always be improved upon, but the complexity increases as does the design time.

- Fuzzy controllers are typically static systems. This means that external dynamic elements may have to be added before effective control can be obtained (e.g. an integrator or some form of storage element).

5.7 Summary

The design of the QFT classical controllers and the fuzzy controllers is very different. Each uses the knowledge available in different ways and hence each has benefits, based on the dominance of the nature of the technique and the knowledge.

The fuzzy controller shows superior noise suppression and noise rejection in most instances, which is due to the ability of the designer to schedule non-linear gains. The classical technique gives better control when the plant is linear (servo motor) or when fuzzy rules and sets are not easily obtainable. The CSTR investigation demonstrates the power and possible future of fuzzy logic control.

The combination of the two systems results in a powerful and robust controller. This is achieved as it is now possible to combine both knowledge forms available into a better controller.

6. Conclusion

Fuzzy control is a method of developing controllers for processes by means of multivalent linguistic information. The information which is used in the controller design is thus not the same as that used in the classical control design. A totally fair comparison is difficult for this very reason. That fuzzy control works, is not disputed, but where it should be used needs to be qualified. The investigation brought to light some issues which are already known and documented[10,23,62 et al], but new issues about the application of fuzzy control have been discovered. The scope of this investigation was limited to pure fuzzy systems, classical QFT systems and hybrid fuzzy/classical systems. Adaptive fuzzy systems were not investigated[1] and this should be borne in mind when reading the conclusions. The benefits of fuzzy control can be derived from two characteristics of fuzzy systems, namely their non-linear nature and their ability to capture multivalent linguistic information[2].

It should be noted that a fuzzy controller can always provide better control than a linear classical controller, due to its non-linear nature. The development time, cost and complexity all need to be included when comparing the systems.

The myth that fuzzy systems, "fuzzify", cloud and hide information is not true. What it does is represent this knowledge in a different manner, which is often more meaningful than bivalent mathematical knowledge.

Another myth is that fuzzy systems are more robust. While they have the ability to be more robust through localised non-linear control, they are not inherently

[1] This would require a comparison between adaptive fuzzy control and adaptive classical control.

[2] This has many consequences including the fact that the controller can use human rationale for its design making.

more robust. This advantage should be further investigated such that it can be effectively used and clearly understood.

Fuzzy control caters for vague linguistic information. It results in a static non-linear gain controller. This is an advantage at present due to the inadequate non-linear classical control design tools. If such a tool were developed, the main advantage of fuzzy control will be its ability to capture linguistic information.

Both classical control techniques and fuzzy control design techniques use knowledge of the plant in the design, but in different forms. Classical control uses a mathematical model, while fuzzy control uses a linguistic model.

From the literature reviewed in Chapter 3 and the comparison detailed in Chapter 5, it is suggested that fuzzy controllers offer the greatest benefits in the following areas:

- If the process cannot be mathematically modelled either due to complexity, time or cost, then fuzzy control should obviously be used.
- When there is a large amount of qualitative knowledge about the plant through vast operator experience, it is much cheaper to convert this into a fuzzy controller than to develop mathematical models for use in classical control.
- When the objectives are complex and require some form of human "intelligence", fuzzy control can include this "intelligence" through rules.
- In a hybrid system where both operator knowledge and mathematical knowledge can be combined into a system which will utilise them effectively.
- If the system dynamics are well defined, but highly non-linear, the fuzzy controller can be easier and quicker to design than multiple linear controllers to deal with the non-linearity.

- If the performance bounds cannot be met with a linear controller, due to the properties of linear controllers, then a non-linear fuzzy controller can be used to achieve these bounds, but at the cost of an increased design time and high complexity.

- When the system characteristics (e.g. non-minimum phase) make linear control difficult.

- If noise or disturbances are problematic, fuzzy control can be used to reduce the effect through careful design using its ability to implement localised control.

- Automation of high-level objectives or when the objectives can be better explained through rules rather than mathematical formulas.

If a classical controller is working well, there is no reason why it should be changed. However, if the classical controller is not achieving satisfactory results, then the fuzzy controller can offer methods of improving control through the inclusion of more knowledge into the system. Many issues need to be addressed before fuzzy control will be accepted as an addition to or replacement of classical control. The question of stability, inclusion of numerical knowledge and various aspects which have been discussed in Chapter 5 need to be addressed. At present fuzzy and classical control should be seen to be complementary. Each shows significant power when used exclusively in their own domains[1], but when used in regions where their domains overlap, the better controller depends on the situation. The type of specification bounds must be taken into account when determining the type of control required. (e.g. the fuzzy controlled servo motor could not track the sine wave but showed excellent results on the step test).

[1] The classical domain is totally quantitative with clearly defined models and bounds. The fuzzy domain can be seen as a combination of both qualitative and quantitative information, but its ability to include the quantitative knowledge is limited.

When a full mathematical model (relatively linear) of the system is already available then classical control will be sufficient (provided the requirements can be met with a linear controller). If a mathematical model is incomplete or lacking, or the dynamics are highly non-linear then the system will be better controlled with a fuzzy controller. The complexity of the system and the objectives must be borne in mind when choosing the form of control to be used.

6.1 Future research

There is a vast amount of research which needs to be done in the area of fuzzy control. Two main areas can be identified :

- Design and analysis tools : One of the major problems with fuzzy control is the lack of tools for incorporating knowledge into the design and analysis of the final design. The major problem is the issue of stability. How should stability in terms of a fuzzy control system be defined ? How can one define robust stability (stability margins) in the area of fuzzy control ? A method of determining stability regardless of the type of system is required. This could possibly be obtained from a combination of the plant and controller map surface. The inclusion of tracking bounds and other bounds (noise, input disturbance, output disturbance) need to be addressed.

- Quantification of the effects of design options : The design of a fuzzy controller involves numerous choices regarding the number and shape of membership functions, fuzzification, defuzzification and various other aspects. The effect of choosing one method over another is poorly understood. What needs to be investigated and quantified is the effect each choice has on the overall system, and therefore a method of quantifying robustness, noise

suppression, disturbance rejection and uncertainty must be incorporated into the design.

Another area which needs investigation is the conversion from fuzzy model to numerical model and vice versa. If a good method can be obtained to achieve this, then both qualitative and quantitative knowledge can be used in the model and controller design.

Appendix A : Model parameters

A.1 Servo motor

Table A.1 : Model parameters for servo motor

Parameter	Value	Unit
K_T	4.382×10^{-2}	*N.m/A*
K_b	$39.4 \times 10^{-3} \rightarrow 48.2 \times 10^{-3}$	*V/(rad/s)*
J_{motor}	$29.57 \times 10^{-6} \rightarrow 47.88 \times 10^{-6}$	$N.m.s^2$
B_{motor}	$92.41 \times 10^{-6} \rightarrow 184.15 \times 10^{-6}$	*N.m.s/rad*
R_m	0.8	Ω
L_m	1.2	*mH*

A.2 CSTR

Table A.2 : Model parameter values for CSTR

Parameter	Meaning	Value	Units
M-Reactant			
F_1	flow rate of M into tank	3.60789	$moles.s^{-1}$
X_{MF}	mole fraction of M in feed stream	.002	

ΔH_{FM}	heat of formation of M	-640	kJ.mole^{-1}
C_{PM}	specific heat capacity of M	32.00	J.mole^{-1}.°C^{-1}
ρ_M	density of M	55506	moles.m^{-3}
T_1	temperature of feed stream F_1	18	°C
N-Reactant			
F_2	flow rate of N into tank	3.60789	moles.s^{-1}
X_{NF}	mole fraction of N in feed stream	0.08	
ΔH_{FN}	heat of formation of N	-740	kJ.mole^{-1}
C_{PN}	specific heat capacity of N	29	J.mole^{-1}.°C^{-1}
ρ_N	density of N	55506	moles.m^{-3}
T_2	temperature of feed stream F_2	18	°C
Inert			
ΔH_{FI}	heat of formation of inert, I	-571.68	kJ.mole^{-1}
C_{PI}	specific heat capacity of I	75.4	J.mole^{-1}.°C^{-1}
ρ_I	density of I	55506	moles.m^{-3}
Product			
ΔH_{FP}	heat of formation of	-2400	J.mole^{-1}

	product P		
C_{PP}	specific heat capacity of P	61	$J.mole^{-1}.°C^{-1}$
ρ_P	density of P	55506	$moles.m^{-3}$
Constants			
k_0	pre-exponential kinetic rate constant or frequency factor	6.5e32→7.8e32 Nominal : 7.1e32	s^{-1}
E_a	activation energy	202.9e3→203e3 Nominal : 6203e3	$J.mole^{-1}$
Operating point			
h	height of liquid in the tank	.28	m
F_c	flow rate of coolant	$50e^{-6}$ (nominal plant)	$m^3.s^{-1}$
T_{cin}	temperature of flow entering cooling coil	18	°C
$F_0=F_1+F_2$	flow rate exiting the tank	7.21578	$moles.s^{-1}$
U	overall heat transfer coefficient	$\frac{1}{\frac{1}{U_0}+\alpha.F_c^{-0.8}}$ where α = 3.4e-7 $W^{-1}.m^{4.4}.°C^{-1}.s^{-0.8}$	$W.m^{-2}.°C^{-1}$

		U_0=3000 $W.m^{-2}.°C^{-1}$	

A.3 Tank

Table A.3 : Model parameter values for tank system

Parameter	Meaning	Value	Units
r	radius of tank	1	m
h	height of tank	2	m
h_0	initial height of tank	1.5 (nominal)	m
ρ	density of fluid	1	$kg.m^3$
K	valve coefficient	0.0669	

Appendix B : Block Diagrams

These diagrams are extracted from Simulink under MATLAB®.

B.1 Servo motor

Summary : Selection of the method of control and the existence of noise must be made before simulation. These button control flags which are used by the system. There is a dedicated plotting program which is opened by clicking on the plotter button. The method of integration used for all simulations is *Linsim* with a maximum step size of 0.1s.

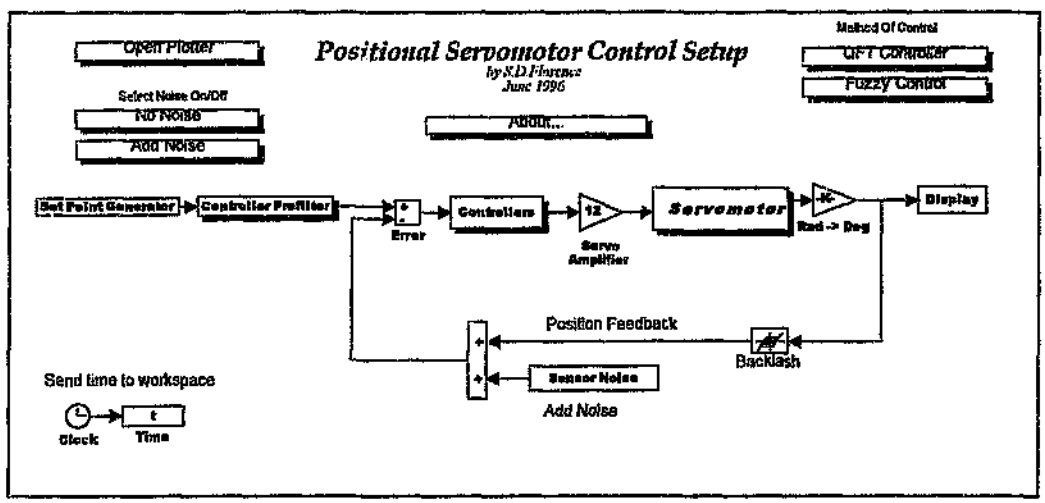


Figure B.1: Servo motor main block diagram

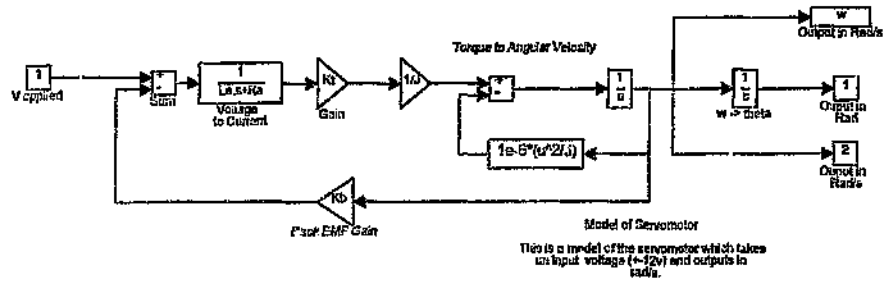


Figure B.2: "Servo motor" subsystem

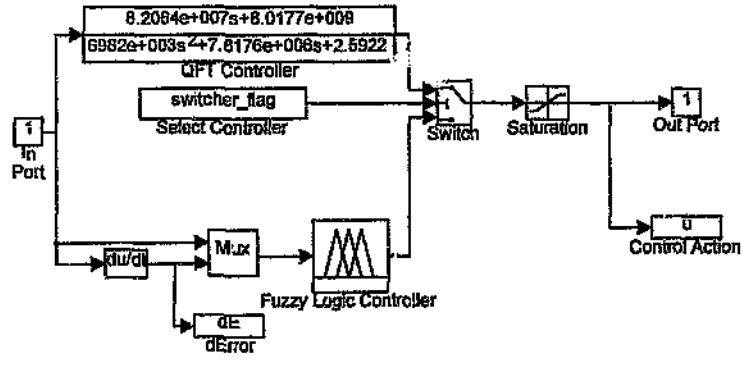


Figure B.3 : "Controllers" subsystem

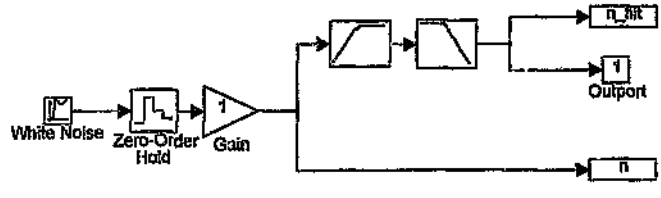


Figure B.4 : "Noise" subsystem

B.2 CSTR

Summary : The CSTR must be initialised (constants and state variables loaded) and the means of control and presence of noise and disturbances need to be chosen. The relevant code is given in Appendix D. The integration method used is *Runge-Kutta 5* with a maximum step size of 1s.

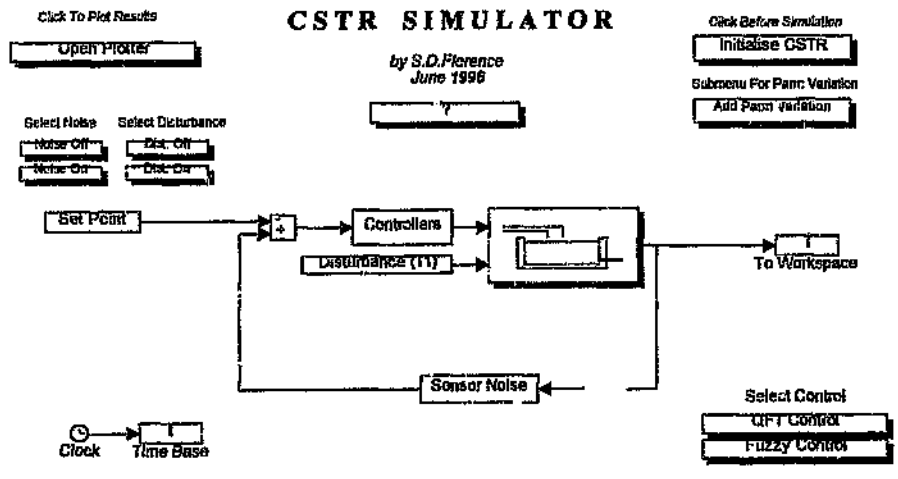


Figure B.5 : Main CSTR System

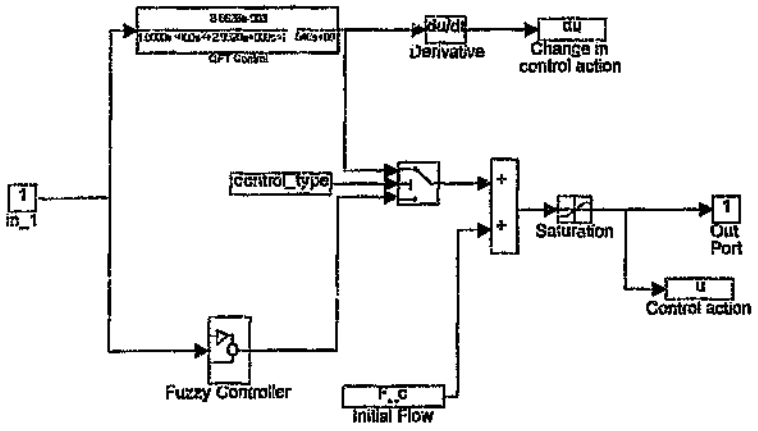
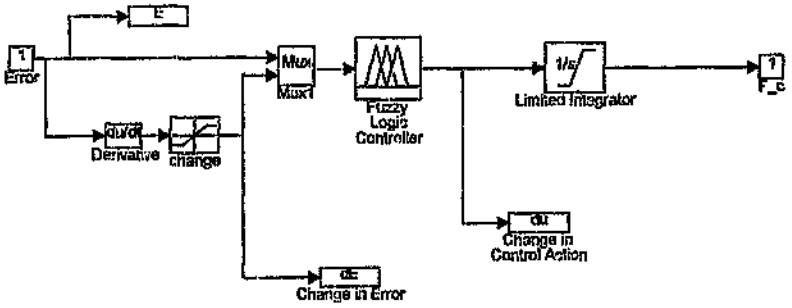


Figure B.6 : "Controller" subsystem



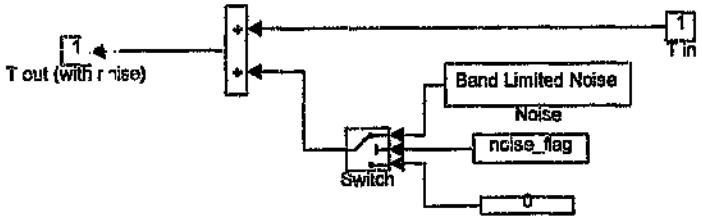


Figure B.8 : "Noise" subsystem

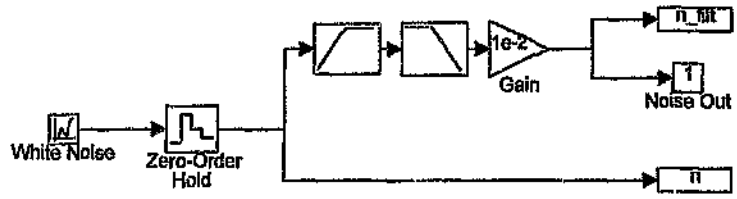


Figure B.9 : "Band Limited Noise" subsystem

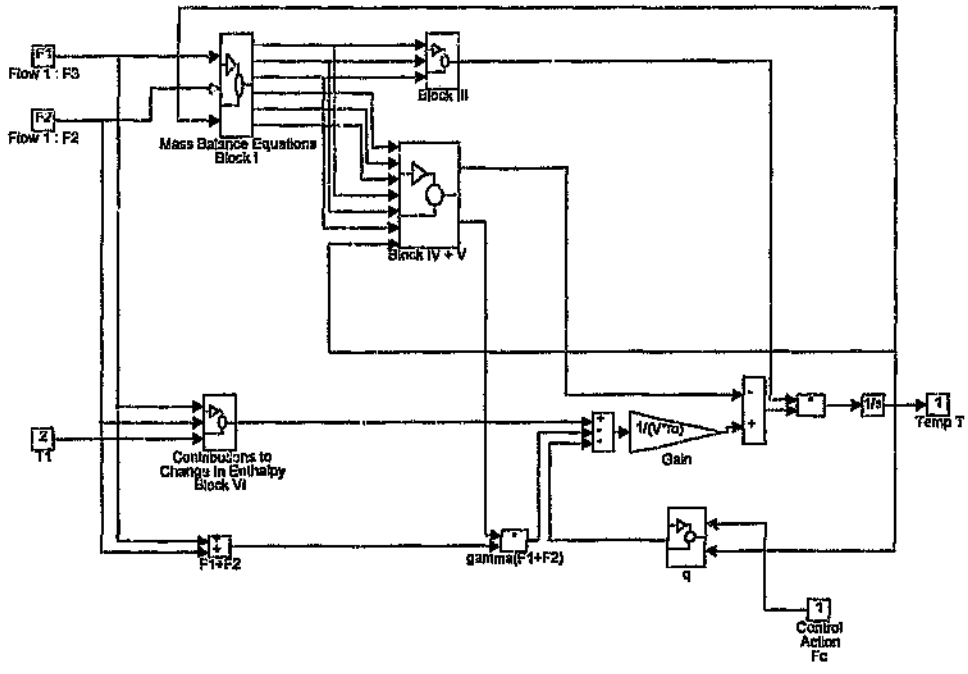


Figure B.10 : "CSTR" subsystem

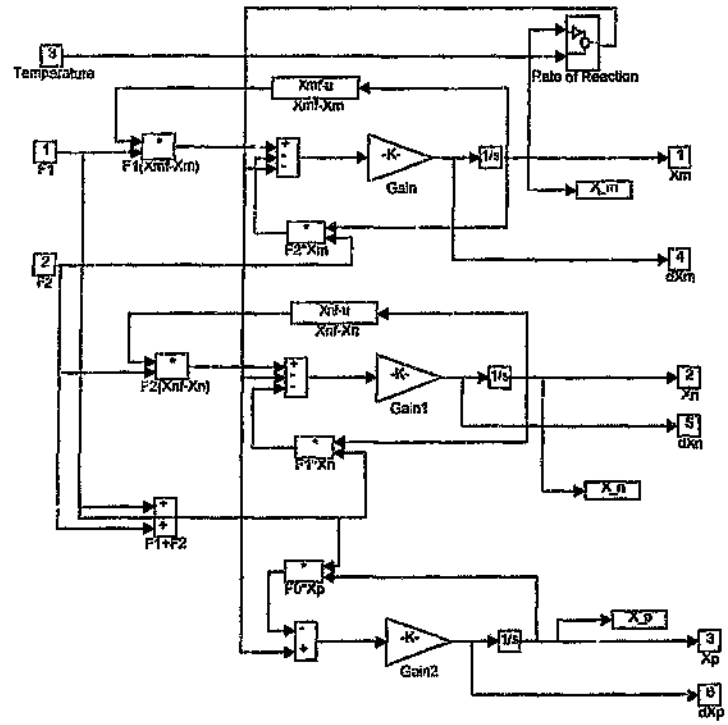


Figure B.11 : "Mass balance equations" subsystem

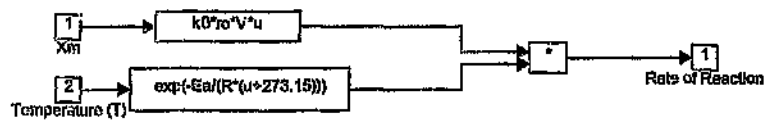


Figure B.12 : "Rate of reaction" subsystem

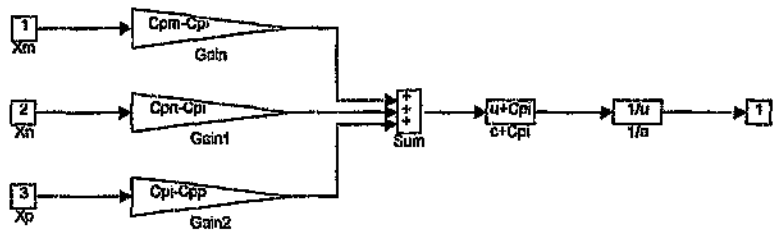


Figure B.13 : "Block III" subsystem

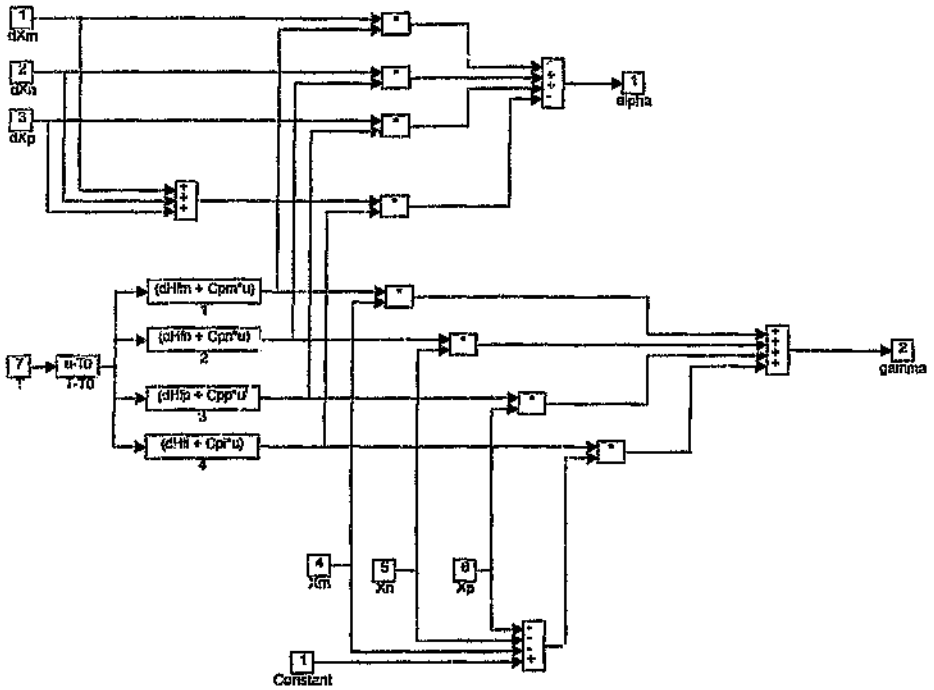


Figure B.14 : "Block IV & V" subsystem

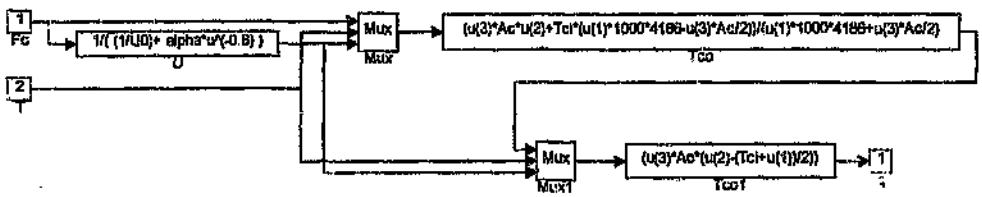


Figure B.15 :"q" subsystem

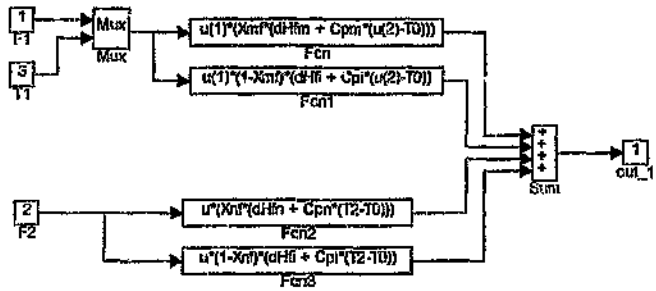


Figure B.16 : "Block IV" subsystem

B.3 Tank

Summary : Select the means of control and then run the simulation. The integration method used is *Linsim* with a maximum step size of 1.

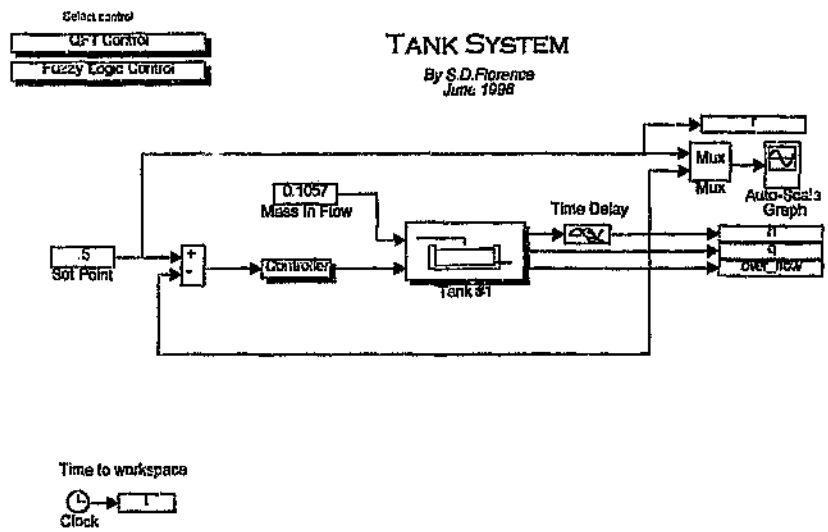


Figure B.17 : Tank main system

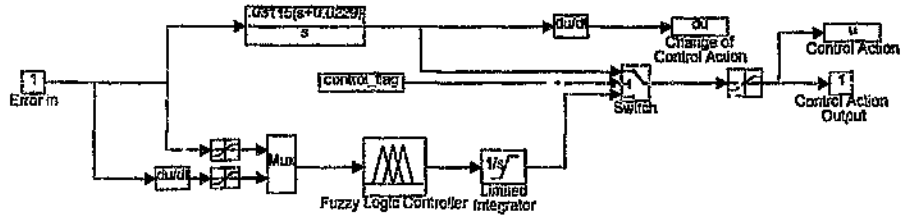


Figure B.18 : "Controller" subsystem

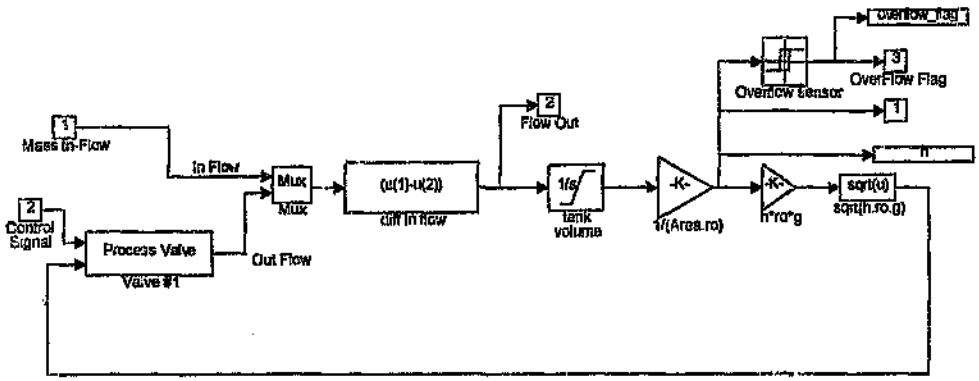


Figure B.19 : "Tank" subsystem

B.4 Nonminimum phase plant

Summary : The integration method used is *Runge-Kutta 5* with a step size of 10.

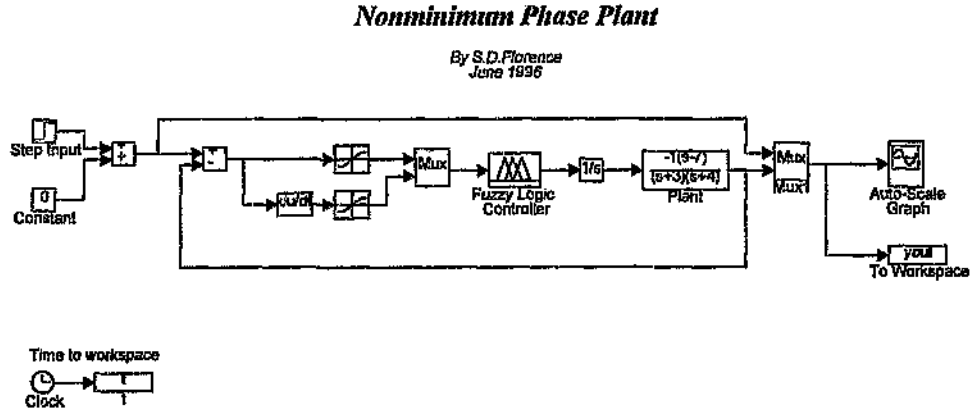


Figure B.20 : Nonminimum phase plant system

B.5 First order system with dead time

Summary : The set point is a series of steps. The nominal set point is the value of the normal operating set point before a step is imposed on it. Tuning is only performed on the "*up*" portion of the step.

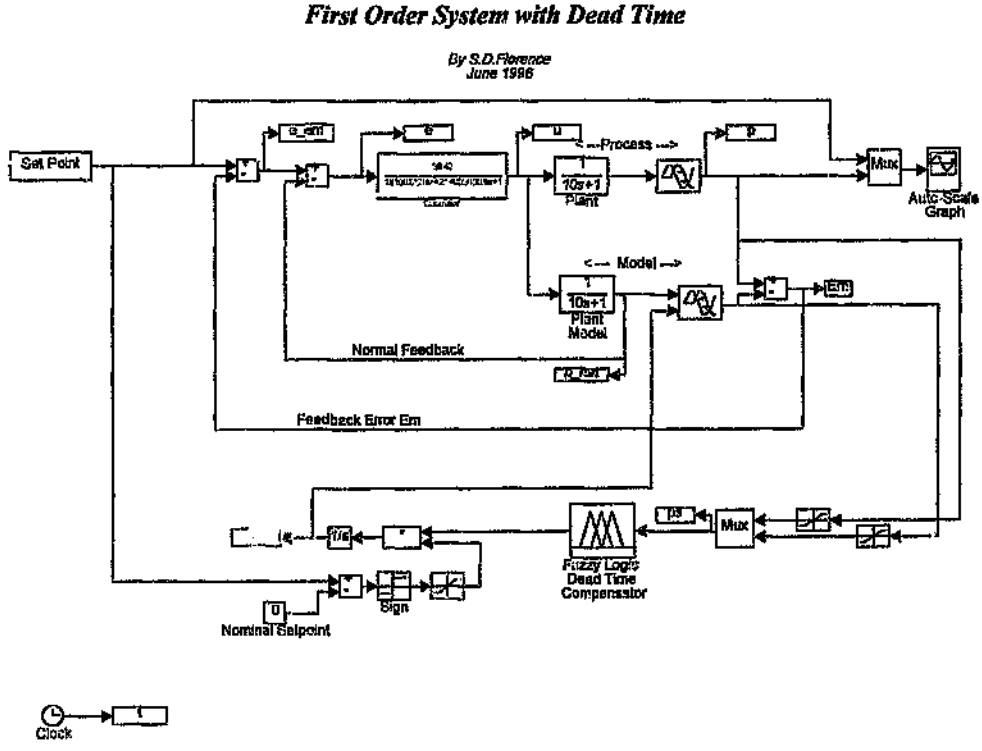


Figure B.21 : First order system with dead time

Appendix C: Controllers

This section summarises all the controllers used in the investigation.

C.1 QFT controllers

C.1.1 Servo motor

Controller :

$$G(s) = 0.295 \frac{\frac{s}{127.7} + 1}{\left(\frac{s}{5461} + 1\right)\left(\frac{s}{8357} + 1\right)}$$

The Nichols plot is given below.

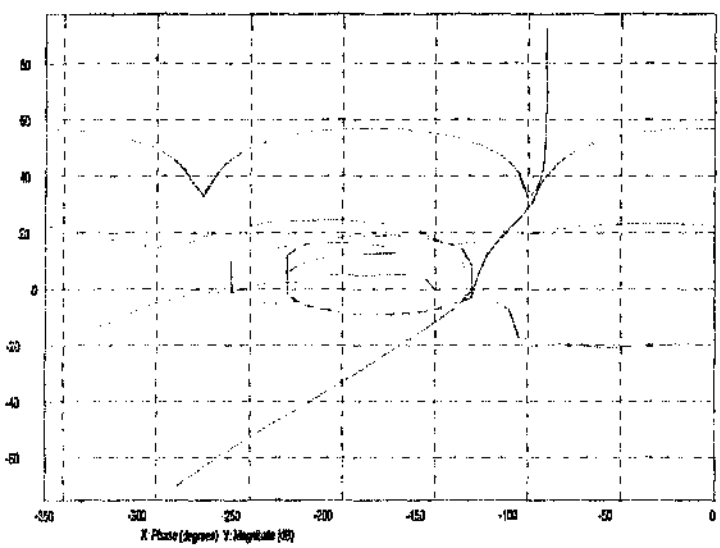


Figure C.1 : Nichols plot of servomotor controller

C.1.2 CSTR

Controller :

$$G(S) = 0.001789 \frac{1}{\left(\frac{s}{1.321}+1\right)\left(\frac{s}{3.631}+1\right)}$$

The Nichols plot of the plant, with the controller is:

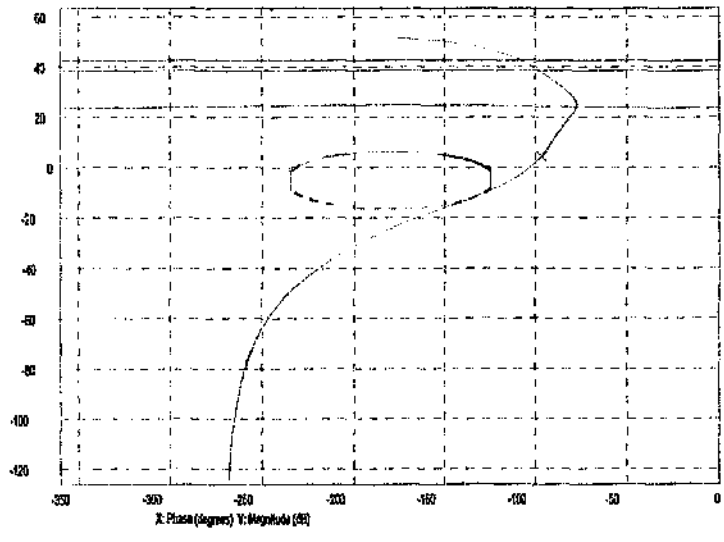


Figure C.2 : Nichols plot of CSTR controller

C.1.3 Tank

Controller :

$$G(s) = 0.03115\frac{\frac{s}{0.0229}+1}{s\left(\frac{s}{1000}+1\right)}$$

The Nichols plot for the system is :

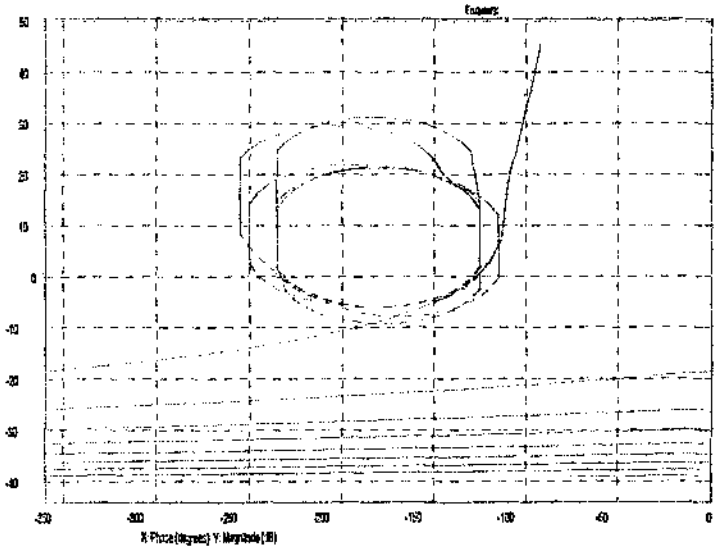


Figure C.3 : Nichols plot of tank controller

C.2 Fuzzy controllers

The designed fuzzy controllers are given in this section. For each controller, the choices made, rules and control surface are given. The exact membership functions for each controller have not been included.

Table C.1 summarises the symbols used in the rule base.

Table C.1 : Fuzzy set symbols

Symbol	Meaning
NL	Negative large
NM	Negative medium
NS	Negative small
N	Negative
Z	Zero
P	Positive
PS	Positive small
PM	Positive medium
PL	Positive large

C.2.1 Servo motor

Style	Mamdani
Fuzzification	Singleton
Input [Error]	[-360 360]°
[dError]	[-5000 5000] °/sec
Output [Voltage]	[-1.4 1.4]V (×12 amplifier)
Membership function	Error, Voltage : Triangular & trapezoidal dError : Gaussian & trapezoidal
Inference	Min
Aggregation	Max
Defuzzification	Centroid
And	Min
Or	Max

		dError				
		NM	*NS*	*Zero*	*PS*	*PM*
Error	*NM*	NL	NL	NL	NL	NL
	NS	NL	NL	NS	NS	NS
	Zero		NS	Zero	PS	
	PS	PS	PS	PS	PL	PL
	PM	PL	PL	PL	PL	PL

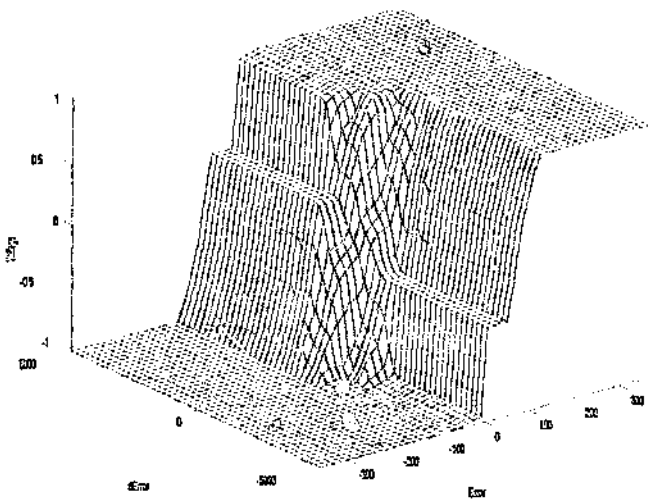


Figure C.4 : Control surface for servo motor fuzzy controller

C.2.2 CSTR

Style	Mamdani
Fuzzification	Singleton
Input Range : Error dError	[-10 10]° [-0.03 0.03] °/sec
Output Range : ΔFlow	[-0.00015 0.00015]$m^3.s^{-2}$
Membership function	Error : Triangular & Gaussian dError : Gaussian Flow : Triangular
Implication	Min
Aggregation	Max
Defuzzification	Centroid
And	Prod
Or	Prob. or

		dError		
		N	*Zero*	*P*
Error	*NM*	close fast	close fast	close fast
	NS	close fast	close slow	close slow
	Zero	close slow	no change	open slow
	PS	open slow	open slow	open fast
	PM	open fast	open fast	open fast

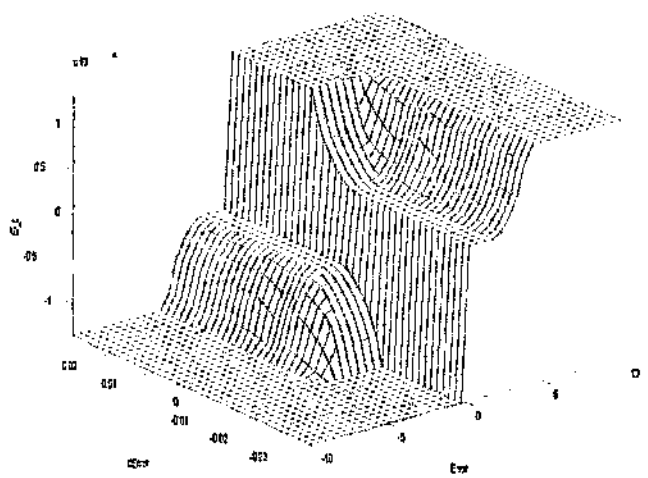


Figure C.5 : Control surface for CSTR fuzzy controller

C.2.3 Tank

Style	Mamdani
Defuzzification	Singleton
Input Range: Error dError ΔFlow	[-2 2]m [-0.05 0.05] m/sec [-0.065 0.065] $kg.s^{-2}$
Membership Functions	Error : Triangular and trapezoidal dError & ΔFlow : Triangular
Implication	Min
Aggregation	Max
Defuzzification	Centroid
And	Min
Or	Max

Error \ dError	*NM*	*NS*	*Zero*	*PS*	*PM*
NS	open fast	open fast	open slow	open slow	no change
Zero	open fast	no change	no change	no change	close fast
PS	no change	close slow	close slow	close fast	close fast

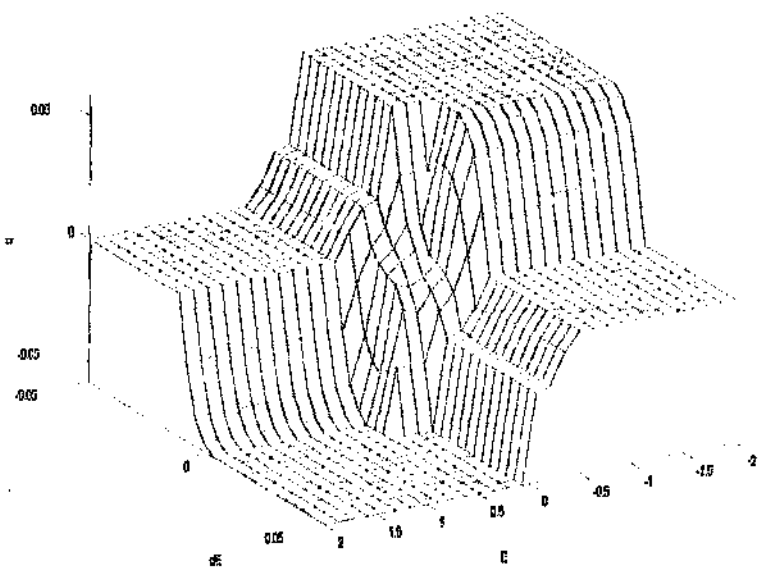


Figure C.6 : Control surface for tank fuzzy controller

C.2.4 Non-minimum phase

Style	Mamdani
Fuzzification	Singleton
Input Range : Error dError	[-5 5] [-5 5]
Output Range : Output	[-3.5 3.5]
Membership Function	Error : Triangular and trapezoidal dError and Output : Triangular
Implication	Min
Aggregation	Max
Defuzzification	Centroid
And	Min
Or	Max

Error \ dError	*N*	*Zero*	*P*
NS	N	NS	NS
Zero	NS	Z	PS
PS	PS	PS	P

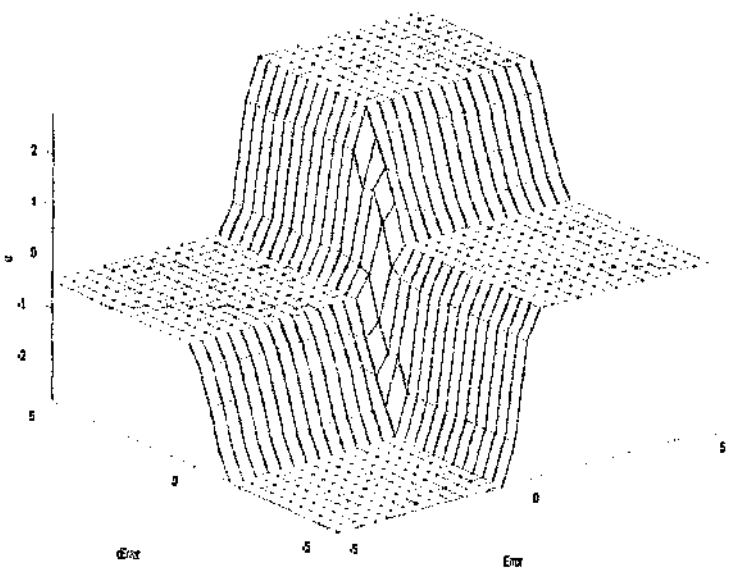


Figure C.7 : Control surface for non-minimum phase plant fuzzy controller

C.3 Hybrid Controller

C.3.1 Classical

Controller :

$$G(s) = \frac{38.42}{\frac{1}{100.6^2}s^2 + 2\frac{0.4053}{100.6}s + 1}$$

Nichols chart (no dead time):

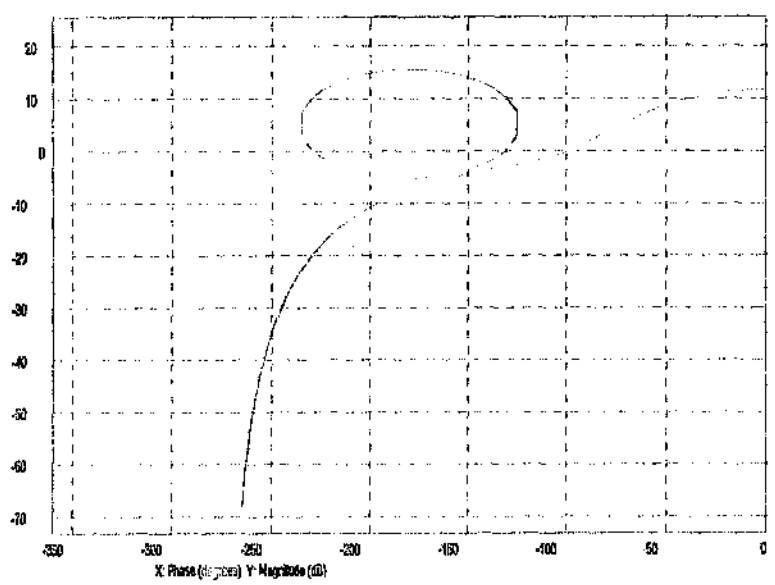


Figure C.8 : Nichols plot for hybrid Smith predictor controller (QFT section)

C.3.2 Fuzzy

	Mamdani
	Singleton
Input	[-2 2] [-2 2]
	[-0.75 0.75]
	P & P_hat :Triangular and trapezoidal Δt_d : Triangular
	Min
	Max
	Centroid
	Min
	Max

		P_hat				
		NM	*NS*	*Zero*	*PS*	*PM*
P	*NM*	Z	Z	Z	Z	Z
	NS	Z	Z	Z	Z	Z
	Zero	Z	Z	Z	PS	PM
	PS	Z	Z	NS	Z	Z
	PM	Z	Z	NM	Z	Z

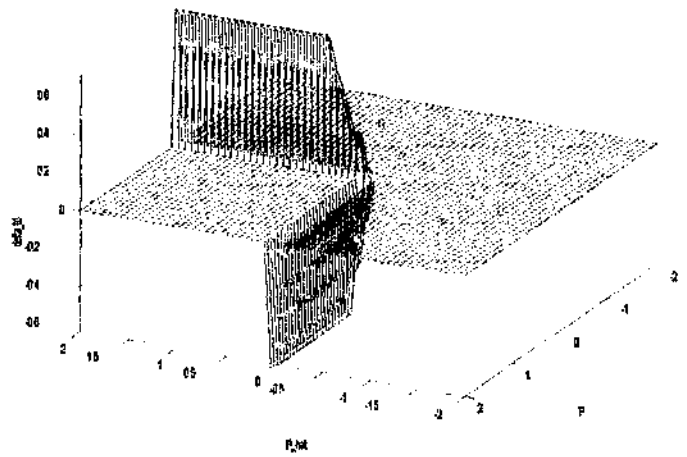


Figure C.9 : Control surface for hybrid Smith predictor controller (fuzzy section)

Appendix D : MATLAB® Code

This section includes the significant MATLAB® code which was written for this investigation.

Note : Only code for the servo motor and CSTR is given. All other code generated is not included as it is derived closely from the given code. Some of the code given works in conjunction with the Simulink models given in Appendix B.

D.1 Servo motor

D.1.1 servo_tf.m

```
% ***************************************
% ***  Servomotor : Positional Control  ***
% ***************************************

% Linear Model Generator
% Prepared by S.D.Florence
% First Edit : October 1995
% Last Edit : June 1996

% Calculates the transfer function of the servo motor

% Inputs : None
% Outputs : transfer function : n,d
%           zeros, poles and gain : z,p,k

format compact;

% Kt : Torque Constant [4.382e-2]

Kt = 4.382e-2;

% Kb : Back EMF Constant [39.4e-3 40.2e-3]
```

```
Kb = 39.4e-3;
%Kb = 48.2e-3;

% Ra : Armature Resistance [0.8 1.2]

Ra = 0.8;

% J : Moment of Inertia [29.57e-6 47.88e-6]

J = 29.57e-6;
%J = 47.88e-6;

% La : Armature Inductance [1.2e-3]

La = 1.2e-3;

% B : Rotational Friction [92.41e-6 184.15e-6]

B = 92.41e-6;
%B = 184.15e-6

% Servo Amplifier

Ka = 1;

% Transfer Function

n = [(Kt/(J*La))];
d = [1 (B/J+Ra/La) ((Ra*B)/(La*J)+(Kt*Kb)/(La*J)) 0];

[z,p,k]=tf2zp(n,d);
```

D.1.2 pmap_s.m

```
% ********************
% *** Plot UI Menu ***
% ********************
%
% Prepared by S.D.Florence
% First Edit November 1995
% Last Edit June 1996

% Prepares and plots the servo motor graphs
```

```
% For use with the Simulink model

% Inputs : op_no (from simulink)
% Ouputs : Graphics

grid_on = 1;
hold_on = 0;

if op_no == 1
            figNumber = 1;
            figNumber=figure(...
            'Name','Data Plotting',...
            'NumberTitle','off',...
            'Units','normalized',...
            'Position',[0.10 0.45 0.65 0.45]);
            op_no = 2;
end;

if op_no == 2

            h = uimenu(figNumber,'Label','General');
            uimenu(h,'Label','Toggle Zoom',...
            'Callback','op_no=100;pmap_s');
            h = uimenu(figNumber,'Label','Plant');
            uimenu(h,'Label','Plot Tracking Position',...
            'Callback','op_no=3; pmap_s;');
            uimenu(h,'Label','Plot Velocity (Rad/s)',...
            'Separator','on',...
            'Callback','op_no=4; pmap_s;');
            uimenu(h,'Label','Plot Velocity (RPM)',...
            'Callback','op_no=5; pmap_s;');


            h = uimenu(figNumber,'Label','Controller');
            uimenu(h,'Label','Plot Control Action',...
            'Callback','op_no=10; pmap_s;');
            uimenu(h,'Label','Plot Error',...
            'Separator','on',...
            'Callback','op_no=11; pmap_s;',.
            uimenu(h,'Label','Effect of Pre-Filter',...
            'Callback','op_no=12; pmap_s;');

            h = uimenu(figNumber,'Label','Fuzzy Aspects');
            uimenu(h,'Label','Plot Map Trajectory',...
            'Callback','op_no=15; pmap_s;');
```

```
            h = uimenu(figNumber,'Label','Noise');
            uimenu(h,'Label','Plot Noise',...
            Callback','op_no=16; pmap_s;');
            uimenu(h,'Label','Plot Distribution',...
            'Callback','op_no=17; pmap_s;');


end;

if op_no == 100

            zoom;

end;

if op_no == 101
            hold_on = ~(hold_on);
end;

if op_no == 3

            if hold_on == 0, cla;
            else hold on, end;
            plot(t,r,'w',t,theta,'r');
            title('Tracking Of Servomotor')
            xlabel('Time (s)');
            ylabel('Degrees');
            if grid_on == 1, grid, end;


end;

if op_no == 4

            if hold_on == 0, cla;
            else hold on, end;
            plot(t,w);
            title('Velocity')
            xlabel('Time (s)');
            ylabel('Rad/s');
            if grid_on == 1, grid, end;

end;

if op_no == 5
```

```
            if hold_on == 0, cla;
            else hold on, end;
            plot(t,60*w/(2*pi));
            title('Velocity')
            xlabel('Time (s)');
            ylabel('RPM');
            if grid_on == 1, grid, end;

end;

if op_no == 10

            if hold_on == 0, cla;
            else hold on, end;
            plot(t,u);
            title('Control Action')
            xlabel('Time (s)');
            ylabel('Normalised Control');
            if grid_on == 1, grid, end;

end;

if op_no == 11

            if hold_on == 0, cla;
            else hold on, end;
            plot(t,r-theta);
            title('Error Signal')
            xlabel('Time (s)');
            ylabel('Degrees');
            if grid_on == 1, grid, end;

end;

if op_no == 12

            if hold_on == 0, cla;
            else hold on, end;
            plot(t,r,'w',t,r_dash,'r');
            title('Effect of Pre-Filter - Before and After')
            xlabel('Time (s)');
            ylabel('Degrees');
            if grid_on == 1, grid, end;

end;
```

```
if op_no == 15

            if hold_on == 0, cla;
            else hold on, end;
            plot(w*60/(2*pi),[r-theta],'r');
            hold on;
            number = round(linspace(1,length(t),10));
            w2 = w(number);
            theta2 = theta(number);
            r2 = r(number);
            plot(w2*60/(2*pi),[r2-theta2],'wo');
            title('Movement in Map')
            xlabel('RPM');
            ylabel('Error');
            if grid_on == 1, grid, end;
            hold off;
            axis([-2500 2500 -360 360]);
end;

if op_no == 16

            if hold_on == 0, cla;
            else hold on, end;
            subplot(211);
            plot(t,n);
            grid
            subplot(212);
            title('Noise - Unfiltered')
            xlabel('Time (s)');
            ylabel('Degrees');

            plot(t,n_filt);
            grid;
            title('Noise - Filtered')
            xlabel('Time (s))');
            ylabel('Degrees');
end;

if op_no == 17

            if hold_on == 0, cla;
            else hold on, end;
            subplot(211);
            hist(n);
            title('Noise - Unfiltered')
            subplot(212);
```

```
            hist(n_filt);
            title('Noise - Filtered')

end;
```

D.1.3 qft_serv.m

```
% ******************************
% *** Servomotor QFT Design ***
% ******************************

% Prepared By S.D.Florence
% First Edit : July 1995
% Last Edit : June 1996

% QFT design for servo motor

% Inputs : None
% Outputs : Graphics, and variables as selected

clc;
clear;
format compact;

% Assumptions made :

%           model valid to 50Hz i.e. second order sufficient

% ********************************************************
% Part I :
% PROBLEM DATA
% ============>
% computing the boundary of plant templates
disp('Computing plant templates (20 plants at 8 frequencies)....')
drawnow

c = 1;      a = 613; b = 3.1e4;

for k = [7.6e5,1.2e6]*180/pi,
            nump(c,:) = k; denp(c,:) = [1,a,b,0]; c = c + 1;
end
```

```
a = 673;
for k = [7.6e5,1.2e6]*180/pi,
            nump(c,:) = k; denp(c,:) = [1,a,b,0];c = c + 1;
end

a = 613; b = 6.4e4;
for k = [7.6e5,1.2e6]*180/pi,
            nump(c,:) = k; denp(c,:) = [1,a,b,0]; c = c + 1;
end

a = 373; b = 6.4e4;
for k = [7.6e5,1.2e6]*180/pi,
            nump(c,:) = k; denp(c,:) = [1,a,b,0];c = c + 1;
end

a = 613; b = 5e4;
for k = [7.6e5,1.2e6]*180/pi,
            nump(c,:) = k; denp(c,:) = [1,a,b,0];c = c + 1;
end


% Compute responses
w = [.01,.1,1,6,10,20,100,1e3,1e5]*2*pi;          % w in Hertz

wb=[.1,6,10,1e3]*2*pi;
P=freqcp(nump,denp,w);

disp(' ')
disp('plottmpl(w,wb,P); %show templates')
drawnow
plottmpl(w,wb,P), title('Plant Templates');
qpause;close(gcf);


% ************************************** *******
% Part II:
% BOUNDS
% ======

disp(' ')
disp('Computing bounds...')
disp(' ')

disp('bdb1=sisobnds(1,w,w,W1,P,R); %margins');
drawnow
```

```
% *** Robust Margin Weight ***

W1=1.3; % robust margins weight
R = 0; % define radius
bdb1=sisobnds(1,w,wb,W1,P,R);
disp('plotbnds(bdb1), %show bounds');
plotbnds(bdb1),title('Robust Margins Bounds');
qpause;close(gcf);

% *** Tracking Bounds ***

disp(' ')
disp('bdb7=sisobnds(7,w,wbd7,W7,P); %tracking bounds');
drawnow
wbd7=[1,6,20]*2*pi;

% *** Upper Bound ***

wn_u = 150;
zeta_u = .65;
nu = wn_u^2;
du = [1 2*zeta_u*wn_u wn_u^2];


mu=freqcp(nu,du,w);

% *** Lower Bound ***

wn_l = 100;
zeta_l = 0.85;
nl = wn_l^2;
dl = [1 2*zeta_l*wn_l wn_l^2];

ml=freqcp(nl,dl,w);


W7=[abs(mu);abs(ml)]; % tracking weight
bdb7=sisobnds(7,w,wbd7,W7,P);
disp('plotbnds(bdb7); %show bounds');
plotbnds(bdb7),title('Robust Tracking Bounds');
qpause;close(gcf);

% *** Combine Bounds ***

disp(' ')
```

```
disp('bdb=grpbnds(bdb1,bdb7); %grouping bounds')
drawnow
bdb=grpbnds(bdb1,bdb7);
disp('plotbnds(bdb); %show all bounds')
plotbnds(bdb),title('Margins and Tracking Bounds');
qpause;close(gcf);

% Intersection Of Bounds

disp(' ')
disp('ubdb=sectbnds(bdb); %intersect bounds')
drawnow
ubdb=sectbnds(bdb);
disp('plotbnds(ubdb); %show bounds')
drawnow
plotbnds(ubdb),title('Intersection of Bounds');
qpause;close(gcf);

% ******************************************************
% Part III:
% DESIGN
% ======

disp(' ')
disp('Design')
disp('lpshape(w1,ubdb,nL0,dL0,del0,nc0,dc0); %loop shaping')
drawnow
nompt=1; nL0=nump(nompt,:); dL0=denp(nompt,:); % nominal plant
del0=0;
w1=logspace(-2.4,200);
nc0 = [  0  2.3101e-003  2.9800e-001];
dc0 = [2.1912e-008  3.0278e-004  1.0000e+000];
lpshape(w1,ubdb,nL0,dL0,del0,nc0,dc0);  %qpause;
numC=nc0; denC=dc0;
G=freqcp(numC,denC,w);


disp(' ')
disp('pfshape(7,w,wbd7,W7,P,[],G,1,nf0,df0); %pre-filter shaping')
drawnow

nf0=[];df0=[];
pfshape(7,w,wbd7,W7,P,[],G,1,nf0,df0); qpause;
numF=nf0; denF=df0;

% ******************************************
```

```
% Part IV:

% ANALYSIS
% ========

disp(' ')
disp('Analysis....')

P=freqcp(nump,denp,wl);
G=freqcp(numC,denC,wl);
F=freqcp(numF,denF,wl);

disp(' ')
disp('chksiso(1,wl,W1,P,R,G); %margins spec')
drawnow
chksiso(1,wl,W1,P,R,G);
qpause;close(gcf);
%
disp(' ');
disp('chksiso(7,wl,W7,P,R,G,[],F); %tracking spec')
drawnow
ind=find(wl<=15);
mu=[]; ml=[]; W7=[];
mu=freqcp(800,[1 40 800],wl);
ml=freqcp(2,[1 2],wl);
W7=[abs(mu);abs(ml)]; % tracking weight
chksiso(7,wl(ind),W7(:,ind),P(:,ind),R,G(ind),[],F(ind));
qpause;close(gcf);
```

D.2 CSTR

D.2.1 c_curve.m

```
% ************************************************
%          Characteristic Curve of CSTR Model
% ************************************************


% Prepared by S.D. Florence
% First Edit : October 1995
% Last Edit : February 1996

% Generates the characteristic curve for a CSTR
```

```
% Inputs : None
% Outputs : Graphics

clear;

% *** M Flow ***
dHfm = -640e3; % J.mole(-1)                    Heat of Formation of M
Cpm = 32.00; % J.mole(-1).deg K(-1)            Specific Heat of M
ro_m = 55506; % mole.m(-3)                     Density of M
T1 = 18; % deg(C)                              Temperature in
F1 = 3.60789;
Xmf = 0.002;


% *** N Flow ***
dHfn = -740e3; % J.mole(-1)                    Heat of Formation of N
Cpn = 29.00; % J.mole(-1).deg K(-1)            Specific Heat of N
ro_n = 55506; % mole.m(-3)                     Density of N
T2 = 18; % deg(C)                              Temperature in
F2 = 3.60789;
Xnf = 0.08;

% *** Inerts ***
dHfi = -571.68e3; % J.mole(-1)                 Heat of formation of inerts
Cpi = 75.415; % J.mole(-1).deg K(-1)           Specific Heat of inerts
ro_i = 55506; % mole.m(-3)                     Density of inerts

% *** P Flow ***
ro_p = 55506; % mole.m(-3)                     Density of P
dHfp = -2400e3; % J.mole(-1)                   Heat of formation of P
Cpp = 61.00; % J.mole(-1).C(-1)                Specific Heat of P
ro_p = 55506; % mole.m(-3)                     Density of P

% *** Constants ***
k0 = 7.1e32; % s(-1)                           Pre-exp. kinetic rate constant
Ea = 203e3; % J.mole(-1)
R = 8.3144; % J.mole.C(-1)                     Ideal gas constant

% *** Tank ***
h = 0.28; % m                                  Initial Height
s = 0.41; % m                                  Side
V = (s*s)*h; % m(3)                            Volume

% *** Cooling Coil ***
```

```
Tci = 18; %                                          Initial Temperature of inert
coil_diameter = 0.015;
coil_diameter_in = 0.0130;
coil_length = 6.9008;
V_coil = coil_length*pi*(coil_diameter_in/2)^2;
Ac = coil_length*2*pi*coil_diameter/2;



% *** Assumptions ***
rc = ro_p;
T0 = 25; % deg(C)                                   Reference Temperature
U0 = 3000;
alpha = 3.4e-7;

T = 20:0.05:32; %                                    Temp range

F_c = 50e-6; %                                       Coolant flow
Fc = F_c;

% Heat Added

for i=1:length(T)
        Xm = F1*Xmf/(F1+F2+k0*ro*V*exp(-Ea/(R*(T(i)+273.15))));
        rate_term_init = k0*ro*V*Xm*exp(-Ea/(R*(T(i)+273.15)));
        Xn = (F1*Xnf - rate_term_init)/(F1+F2);
        Xp = rate_term_init/(F1+F2);



h_in(i) =  F1*(Xmf*(dHfm+Cpm*(T1-T0))+(1-Xmf)*(dHfi+Cpi*(T1-T0)))+...
              F2*(Xnf*(dHfn+Cpn*(T2-T0))+(1-Xnf)*(dHfi+Cpi*(T2-T0)))-...
           (F1+F2)*(Xm*(dHfm + Cpm*(T(i)-T0)) + ...
           Xn*(dHfn + Cpn*(T(i)-T0)) + ...
           Xp*(dHfp + Cpp*(T(i)-T0)) + ...
           (1-Xm-Xn-Xp)*(dHfi+Cpi*(T(i)-T0)));
end;

% Heat Removed

for i=1:length(T)

           U = 1/((1/U0) + alpha*Fc^(-0.8));
           tn = Fc*4186e3 - U*Ac/2;
           tn = U*Ac*T(i) + Tci*tn;
           td = Fc*4186e3 + U*Ac/2;
```

```
            Tco= tn/td;
            Tcave = (Tci+Tco)/2;
            q(i) = U*Ac*(T(i)-Tcave);

end;

plot(T,q,'w',T,h_in,'w');hold on;
xlabel('Temperature (degrees C)');
ylabel('Heat/Time');
%title('q removed by coil (Linear) and q formed by reaction');
grid;

suptitle('System Characteristic Equation');

spots=linspace(10,length(T)-10,8);
plot(T(spots),q(spots),'wo',T(spots),h_in(spots),'w+');
```

D.2.2 lin_cstr.m

```
function [num,den]=l_cstr_c(n)

% *************************
% *** Linear CSTR Model ***
% *************************

% Created by S.D.Florence
% First Edit : 23 September 1995
% Last Edit : 3 February 1996

% Generates the linear model of the CSTR at one of the three
% equilibrium points.

% Inputs : n  - point number (1,2,3)
% Ouputs : num,den - transfer function


format compact;

% ************************************************


% Constants
% *********
```

```
% *** M Flow ***
dHfm = -640e3; % J.mole(-1)                    Heat of Formation of M
Cpm = 32.00; % J.mole(-1).deg K(-1)            Specific Heat of M
ro_m = 55506; % mole.m(-3)                     Density of M
T1 = 18; % deg(C)                              Temperature in
F1 = 3.60789;
Xmf = 0.002;

% *** N Flow ***
dHfn = -740e3; % J.mole(-1)                    Heat of Formation of N
Cpn = 29.00; % J.mole(-1).deg K(-1)            Specific Heat of N
ro_n = 55506; % mole.m(-3)                     Density of N
T2 = 18; % deg(C)                              Temperature in
F2 = 3.60789;
Xnf = 0.08;

% *** Inerts ***
dHfi = -571.68e3; % J.mole(-1)                 Heat of formation of inerts
Cpi = 75.415; % J.mole(-1).deg K(-1)           Specifife Heat of inerts
ro_i = 55506; % mole.m(-3)                     Density of inerts

% *** P Flow ***
ro_p = 55506; % mole.m(-3)                     Density of P
dHfp = -2400e3; % J.mole(-1)                   Heat of formation of P
Cpp = 61.00; % J.mole(-1).C(-1)                Specific Heat of P
ro_p = 55506; % mole.m(-3)                     Density of P

% *** Constants ***
k0 = 7.1e32; % s(-1)                           Pre-exp. kinetic rate constant
Ea = 203e3; % J.mole(-1)
R = 8.3144; % J.mole.C(-1)                     Ideal gas constant

% *** Tank ***
h = 0.29; % m                                  Initial Height
s = 0.41; % m                                  Side
V = (s*s)*h; % m(3)                            Volume

% *** Cooling Coil ***

Tci = 18; %                                    Initial Temperature of inert
coil_diameter = 0.015;
coil_diameter_in = 0.0130;
coil_length = 6.9000;
V_coil = coil_length*pi*(coil_diameter_in/2)^2;
Ac = coil_length*2*pi*coil_diameter/2;
```

```
% *** Assumptions ***
ro = ro_p;
T0 = 25; % deg(C)                                   Reference Temperature
U0 = 3000;
alpha = 3.4e-7;


% Paremeter Variation;

k0 = 7.1e32;
Ea = 203e3;

% ************************************************

% Steady State Conditions
% =======================


% Mid Point
if n==1,
            Fc_ss = 50e-6;
            T_ss = 26.057964;
end

% Top Point
if n==2,
            Fc_ss = 50e-6;
            T_ss = 30.9101;
end

% Low Point
if n==3,
            Fc_ss = 50e-6;
            T_ss = 21.4649;
end

%Fc_ss = 42.9e-6;

T_ss_k = T_ss + 273.15;
Xm_ss = F1*Xmf/(F1+F2+k0*ro*V*exp(-Ea/(R*(T_ss+273.15))));
rate_term_init = k0*ro*V*Xm_ss*exp(-Ea/(R*(T_ss+273.15)));
Xm_ss = (F1*Xmf - rate_term_init)/(F1+F2);
Xp_ss = rate_term_init/(F1+F2);
```

```
% Derived Steady State Conditions

Beta_ss = Xm_ss*Cpm + Xn_ss*Cpn + Xp_ss*Cpp + ...
              (1 - Xm_ss - Xn_ss - Xp_ss)*Cpi;

U = 1/((1/U0) + alpha * Fc_ss^(-0.8));
Tco = (U*Ac*T_ss + Tci* (Fc_ss*4186e3 - U*Ac/2))/...
           (Fc_ss*4186e3 + U*Ac/2);
Tave_ss = (Tci + Tco)/2;
q_ss = U*Ac*(T_ss -  Tave_ss);

Gamma_ss = Xm_ss*(dHfm + Cpm*(T_ss-T0)) + ...
               Xn_ss*(dHfn + Cpn*(T_ss-T0)) + ...
               Xp_ss*(dHfp + Cpp*(T_ss-T0)) + ...
               (1 - Xm_ss - Xn_ss  -Xp_ss)*(dHfi + Cpi*(T_ss-T0));

alpha_ss = 0;

C = (F1*(Xmf*(dHfm + Cpm*(T1-T0)) + (1   Xmf)*(dHfi + Cpi*(T1-T0)))+...
     F2*(Xnf*(dHfn + Cpn*(T2-T0)) + (1 - Xnf)*(dHfi + Cpi*(T2-T0))));

CC=C;

TB_ss = (C - q_ss - (F1+F2)*Gamma_ss - V*ro*alpha_ss);

k_ss = k0*V*ro*exp(-Ea/(R*T_ss_k));


% ************************************************

% Partial Differentials
% =====================

% U

dU_dFc = 0.8*alpha*(((1/U0)+alpha*Fc_ss^(-0.8))^(-2))*Fc_ss^(-1.8);


% Output Temperature (Tco)

dTco_dT  = U*Ac/(Fc_ss*1000*4186+(U*Ac/2));
dTco_dFc = (dU_dFc*Ac*T_ss + Tci*(4186e3 - dU_dFc*Ac/2))*...
           (Fc_ss*4186e3 + U*Ac/2)^(-1) + ...
           (U*Ac*T_ss + Tci*(Fc_ss*4186e3 - U*Ac/2))*...
```

```
(-1)*(4186e3 + dU_dFc*Ac/2)*(Fc_ss*4186e3 + U*Ac/2)^(-2);


% Average Temperature

dTav_dT  = 0.5*dTco_dT;
dTav_dFc = 0.5*dTco_dFc;


% q - Energy from system

dq_dXm = 0;
dq_dXn = 0;
dq_dXp = 0;
dq_dT  = U*Ac - U*Ac*dTav_dT;
dq_dFc = dU_dFc * Ac*T_ss - dU_dFc*Tave_ss*Ac - dTav_dFc*U*Ac;

% dalpha - differantial term;

dalpha_dXm = 0;
dalpha_dXn = 0;
dalpha_dXp = 0;
dalpha_dT  = 0;
dalpha_dFc = 0;

% Beta

dBeta_dXm = Cpm - Cpi;
dBeta_dXn = Cpn - Cpi;
dBeta_dXp = Cpp - Cpi;
dBeta_dT  = 0;
dBeta_dFc = 0;

% Gamma -

dGamma_dXm = (dHfm + Cpm * (T_ss - T0)) - dHfi - Cpi*(T_ss - T0);
dGamma_dXn = (dHfn + Cpn * (T_ss - T0)) - dHfi - Cpi*(T_ss - T0);
dGamma_dXp = (dHfp + Cpp * (T_ss - T0)) - dHfi - Cpi*(T_ss - T0);
dGamma_dT  = Xm_ss*Cpm + Xn_ss*Cpn + Xp_ss*Cpp + ...
                (1 - Xm_ss - Xn_ss - Xp_ss)*Cpi;
dGamma_dFc = 0;


% Temperature

dBeta2_dXm = (-1/(Beta_ss^2))*dBeta_dXm;
```

```
dBeta2_dXm = (-1/(Beta_ss^2))*dBeta_dXm;
dBeta2_dXn = (-1/(Beta_ss^2))*dBeta_dXn;
dBeta2_dXp = (-1/(Beta_ss^2))*dBeta_dXp;
dBeta2_dT  = (-1/(Beta_ss^2))*dBeta_dT;
dBeta2_dFc = (-1/(Beta_ss^2))*dBeta_dFc;

dTB_dXm = (-dq_dXm - (F1+F2)*dGamma_dXm - V*ro*dalpha_dXm);
dTB_dXn = (-dq_dXn - (F1+F2)*dGamma_dXn - V*ro*dalpha_dXn);
dTB_dXp = (-dq_dXp - (F1+F2)*dGamma_dXp - V*ro*dalpha_dXp);
dTB_dT  = (-dq_dT  - (F1+F2)*dGamma_dT  - V*ro*dalpha_dT);
dTB_dFc = (-dq_dFc - (F1+F2)*dGamma_dFc - V*ro*dalpha_dFc);

dT_dXm = (1/(V*ro))*(dBeta2_dXm * TB_ss + (1/Beta_ss)*dTB_dXm);
dT_dXn = (1/(V*ro))*(dBeta2_dXn * TB_ss + (1/Beta_ss)*dTB_dXn);
dT_dXp = (1/(V*ro))*(dBeta2_dXp * TB_ss + (1/Beta_ss)*dTB_dXp);
dT_dT  = (1/(V*ro))*(dBeta2_dT  * TB_ss + (1/Beta_ss)*dTB_dT);
dT_dFc = (1/(V*ro))*(dBeta2_dFc * TB_ss + (1/Beta_ss)*dTB_dFc);


% Xm concentration

%NB T_ss in kelvin;

dk_dXm = 0;
dk_dXn = 0;
dk_dXp = 0;
dk_dT  = k0^ro*V*(Ea/R)*(1/(T_ss_k^2))*exp(-Ea/(R*T_ss_k));
dk_dFc = 0;

dXm_dXm = (1/(V*ro))*(-F1-F2-k_ss);
dXm_dXn = 0;
dXm_dXp = 0;
dXm_dT  = -(1/(V*ro))*Xm_ss*dk_dT;
dXm_dFc = 0;


% Xn Concentration

dXn_dXm = -k_ss/(ro*V);
dXn_dXn = (1/(ro*V))*(-F1-F2);
dXn_dXp = 0;
dXn_dT  = -(1/(V*ro))*Xm_ss*dk_dT;
dXn_dFc = 0;


% Xp Concentration
```

```
dXp_dXm = k_ss/(ro*V);
dXp_dXn = 0;
dXp_dXp = (1/(V*ro))*(-F1-F2);
dXp_dT  = (1/(V*ro))*Xm_ss*dk_dT;
dXp_dFc = 0;

% ****************************************

% State Space
% ===========

A  =   [dXm_dXm dXm_dXn dXm_dXp dXm_dT;...
        dXn_dXm dXn_dXn dXn_dXp dXn_dT;...
        dXp_dXm dXp_dXn dXp_dXp dXp_dT;...
        dT_dXm  dT_dXn  dT_dXp  dT_dT];

B   =  [dXm_dFc;dXn_dFc;dXp_dFc;dT_dFc];

C   =  [0 0 0 1];

%C   =  [1 0 0 0];

D   =  [0];

[num,den] = ss2tf(A  ,C,D);
[z,p,k] = tf2zp(num,den);

disp('Results :');
disp(['Gain  :']);
disp(k);
disp(['Zeros :']);
disp(z);
disp(['Poles :']);
disp(p);

disp('.............');
disp('Transfer Function');
disp(num);
disp('--------------------------------');
disp(den);

% Remove poles/zeros that cancel
disp(' ');

[num_min,den_min]=minreal(num,den);
disp('Minmised Transfer Function is :');
```

```
disp(num_min);
disp('-----------------------------------------');
disp(den_min);
disp(' ');

[z,p,k] = tf2zp(num_min,den_min);

disp(['Gain  :']);
disp(k);
disp(['Zeros :']);
disp(z);
disp(['Poles :']);
disp(p);
```

D.2.3 i_cstr.m

```
% ****************************************
% ***   Initialise Simulink CSTR Model ***
% ****************************************

% By S.D.Florence
% First Edit : October 1995
% Last Edit : June 1996

% Initialise the CSTR simulation
% For use with Simulink model

% Inputs : None
% Outputs : Variables to workspace

clc;
disp('Clearing Variables....');
clear;
disp('Initialising Variables...');

% *** M Flow ***
dHfm = -640e3; % J.mole(-1)                    Heat of Formation of M
Cpm = 32.00; % J.mole(-1).der K(-1)            Specific Heat of M
ro_m = 55506; % mole.m(-3)                     Density of M
T1 = 18; % deg(C)                              Temperature in
F1 = 3.60789;
Xmf = 0.002;
```

```
% *** N Flow ***
dHfn = -740e3; % J.mole(-1)                    Heat of Formation of N
Cpn = 29.00; % J.mole(-1).deg K(-1)            Specific Heat of N
ro_n = 55506; % mole.m(-3)                     Density of N
T2 = 18; % deg(C)                              Tempezature in
F2 = 3.60789;
Xni = 0.08;

% *** Inerts ***
dHfi = -571.68e3; % J.mole(-1)                 Heat of formation of inerts
Cpi = 75.415; % J.mole(-1).deg K(-1)           Specifife Heat of inerts
ro_i = 55506; % mole.m(-3)                     Density of inerts

% *** P Flow ***
ro_p = 55506; % mole.m(-3)                     Density of P
dHfp = -2400e3; % J.mole(-1)                   Heat of formation of P
Cpp = 61.00; % J.mole(-1).C(-1)                Specific Heat of P
ro_p = 55506; % mole.m(-3)                     Density of P

% *** Constants ***
k0 = 7.1e32; % s(-1)                           Pre-exp. kinetic rate constant
Ea = 203e3; % J.mole(-1)                       R = 8.3144; % J.mole.C(-1)Ideal gas constant

% *** Tank ***
h = 0.28; % m                                  Initial Height
s = 0.41; % m                                  Side
V = (s*s)*h; % m(3)                            Volume

% *** Cooling Coil ***

Tci = 18; %                                    Initial Temperature of inert
coil_diameter = 0.015;
coil_diameter_in = 0.0138;
coil_length = 6.2008;
V_coil = coil_length*pi*(coil_diameter_in/2)^2;
Ac = coil_length*2*pi*coil_diameter/2;



% *** Assumptions ***
ro = ro_p;
T0 = 25; % deg(C)                              Reference Temperature
U0 = 3000;
alpha = 3.4e-7;
```

```
% Initial Conditions

T_init = 26.057965;
%T_init = 26.057965;  % Up
%T_init = 26.057964;  % Down
xm_init = F1*Xmf/(F1+F2+k0*ro*V*exp(-Ea/(R*(T_init+273.15))));
rate_term_init = k0*ro*V*xm_init*exp(-Ea/(R*(T_init+273.15)));
xm_init = (F1*Xmf - rate_term_init)/(F1+F2);
xp_init = rate_term_init/(F1+F2);

F_c = 50e-6;

% Disturbance

%F1=20;
%T2=20;

% Parameter Variation

%k0 = 6.5e32;
%Ea = 204e3;

disp('Variables Initialised.');
disp(' ');

% *** Load Controllers ***

% Fuzzy Controller

disp('Loading Controller.....');
CSTR_F3 = readfis('cstr_f3');
disp('Loaded.');


% *** Setting Default Flags ***

disp ('Setting Default Flags...');

% Set Default Control
control_type = 1;

% Set Noise Flag
noise_flag = -1;

% Set Disturbance Flag
dist_flag  = -1;
```

```
disp('Flags Set.');
disp('  ');
disp('CSTR Initialised.');
```

D.2.4 qft_cstr.m

```
% ***************************
% *** QFT Design for CSTR ***
% ***************************

% Author: S. Florence
% First Edit : February 1996
% Last Edit : June 1996

% Interactive design for CSTR
% Requires the QFT toolbox

% Inputs : None
% Outputs : Graphics, and variables as selected
% Specs as given in main report

clear;

pse = 1; %                                    Pause flag
clc


echo off
clc

% PROBLEM DATA

% compute the boundary of the plant templates
disp('Computing plant templates.... ')
drawnow

c = 1; k = 8.8431e1; b = 4.8159e-1; d = 8.5546e-3;
for e = -[1.3253e-6 1.9e-6 2.5e-6 3.2114e-6],
 nump(c,:) = [k,b];  denp(c,:) = [1,d,e];  c = c + 1;
end
k = 8.8431e1; b = 4.8159e-1; d = 9.5561e-3;
for e = -[1.3253e-6 1.9e-6 2.5e-6 3.2114e-6],
```

```
 nump(c,:) = [k,b];  denp(c,:) = [1,d,e];  c = c + 1;
end
k = 8.8431e1; b = 5.6840e-1; d = 9.5561e-3;
for e = -[1.3253e-6 1.9e-6 2.5e-6 3.2114e-6],
 nump(c,:) = [k,b];  denp(c,:) = [1,d,e];  c = c + 1;
end
k = 8.8431e1; b = 5.6840e-1; d = 8.5546e-3;
for e = -[1.3253e-6 1.9e-6 2.5e-6 3.2114e-6],
 nump(c,:) = [k,b];  denp(c,:) = [1,d,e];  c = c + 1;
end

k = 1.1080e2; b = 4.8159e-1; d = 8.5546e-3;
for e = -[1.3253e-6 1.9e-6 2.5e-6 3.2114e-6],
 nump(c,:) = [k,b];  denp(c,:) = [1,d,e];  c = c + 1;
end
k = 1.1080e2; b = 4.8159e-1; d = 9.5561e-3;
for e = -[1.3253e-6 1.9e-6 2.5e-6 3.2114e-6],
 nump(c,:) = [k,b];  denp(c,:) = [1,d,e];  c = c + 1;
end
k = 1.1080e2; b = 5.6840e-1; d = 9.5561e-3;
for e = -[1.3253e-6 1.9e-6 2.5e-6 3.2114e-6],
 nump(c,:) = [k,b];  denp(c,:) = [1,d,e];  c = c + 1;
end
k = 1.1080e2; b = 5.6840e-1; d = 8.5546e-3;
for e = -[1.3253e-6 1.9e-6 2.5e-6 3.2114e-6],
 nump(c,:) = [k,b];  denp(c,:) = [1,d,e];  c = c + 1;
end


% *** Nominal Plant ***

nump(c,:) =[9.9281e+001, 5.3098e-001];
denp(c,:) =[1.0000e+000,  8.9364e-003, -2.5800e-006]; c= c+1;


disp('No of Plants :');
disp(c-1);

disp(' ');


nompt = c-1;  % define nominal plant case

% Compute responses
w = [0.0001,.001,.01,.1,1];
p=freqcp(nump,denp,w);
```

```
 nump(c,:) = [k,b];  denp(c,:) = [1,d,e];  c = c + 1;
end
k = 8.8431e1; b = 5.6840e-1; d = 9.5561e-3;
for e = -[1.3253e-6 1.9e-6 2.5e-6 3.2114e-6],
 nump(c,:) = [k,b];  denp(c,:) = [1,d,e];  c = c + 1;
end
k = 8.8431e1; b = 5.6840e-1; d = 8.5546e-3;
for e = -[1.3253e-6 1.9e-6 2.5e-6 3.2114e-6],
 nump(c,:) = [k,b];  denp(c,:) = [1,d,e];  c = c + 1;
end

k = 1.1080e2; b = 4.8159e-1; d = 8.5546e-3;
for e = -[1.3253e-6 1.9e-6 2.5e-6 3.2114e-6],
 nump(c,:) = [k,b];  denp(c,:) = [1,d,e];  c = c + 1;
end
k = 1.1080e2; b = 4.8159e-1; d = 9.5561e-3;
for e = -[1.3253e-6 1.9e-6 2.5e-6 3.2114e-6],
 nump(c,:) = [k,b];  denp(c,:) = [1,d,e];  c = c + 1;
end
k = 1.1080e2; b = 5.6840e-1; d = 9.5561e-3;
for e = -[1.3253e-6 1.9e-6 2.5e-6 3.2114e-6],
 nump(c,:) = [k,b];  denp(c,:) = [1,d,e];  c = c + 1;
end
k = 1.1080e2; b = 5.6840e-1; d = 8.5546e-3;
for e = -[1.3253e-6 1.9e-6 2.5e-6 3.2114e-6],
 nump(c,:) = [k,b];  denp(c,:) = [1,d,e];  c = c + 1;
end


% *** Nominal Plant ***

nump(c,:) =[9.9281e+001, 5.3098e-001];
denp(c,:) =[1.0000e+000,  8.9364e-003, -2.5800e-006]; c= c+1;


disp('No of Plants :');
disp(c-1);

disp(' ');


nompt = c-1;  % define nominal plant case

% Compute responses
w = [0.0001,.001,.01,.1,1];
P=freqcp(nump,denp,w);
```

```
disp('No of freq points :');
disp(length(w));
disp(' ');

disp(' ')
disp('plottmpl(w,W,P,nompt); %show templates')
drawnow
plottmpl(w,W,P,nompt), title('Plant Templates')
if pse qpause; end
close(gcf);


% ************************ BOUNDS ************************


% *** Robust Margin ***

disp(' ')
disp('Computing bounds...')
disp(' ')
disp('bdb1=sisobnds(1,w,wbd1,W1,P,R,nompt); %margins')
drawnow

R=0;
wbd1 = [.0001,.001,.01,.1,1]; % compute bounds at all frequencies in w
W1 = 1.2;  % define weight
bdb1 = sisobnds(2,w,wbd1,W1,P,R,nompt);
disp('plotbnds(bdb1); %show bounds')
drawnow
plotbnds(bdb1),title('Robust Margins Bounds');
if pse qpause; end
close(gcf);

% *** Output Disturbance ***

disp(' ')
disp('bdb2=sisobnds(2,w,wbd2,W2,P,R,nompt); %output disturbance rejection')
drawnow

wbd2=[.0001,.001,.01]; % the frequency array
W2=abs(freqcp([1,1.01e-1,1e-4],[1,2.01,1.02,1e-2],w));
bdb2 = sisobnds(2,w,wbd2,W2,P,R,nompt);
disp('plotbnds(bdb2); %show bounds')
drawnow
plotbnds(bdb2),title('Robust Output Disturbance Rejection Bounds');
```

```
if pse qpause; end
close(gcf);


% *** Input Disturbance Bounds ***

disp(' ')
disp('bdb3=sisobnds(3,w,wbd3,W3,P,R,nompt); %input disturbance rejection')
drawnow
wbd3=[.0001,.001,.01]; % the frequency array
W3 = 1000;% define weight
%W3=abs(freqcp(100*[1,1.01e-1,1e-4],[1,2.01,1.02,1e-2],w));
bdb3 = sisobnds(3,w,wbd3,W3,P,R,nompt);
disp('plotbnds(bdb3); %show bounds')
plotbnds(bdb3),title('Robust Input Disturbance Rejection Bounds');
if pse qpause; end
close(gcf);

disp(' ')
disp('bdb=grpbnds(bdb1,bdb2); %grouping bounds')
drawnow
bdb=grpbnds(bdb1,bdb2);
disp('plotbnds(bdb); %show all bounds')
plotbnds(bdb),title('All Bounds');
if pse qpause; end
close(gcf);

disp(' ')
disp('ubdb=sectbnds(bdb); %intersect bounds')
drawnow
ubdb=sectbnds(bdb);
disp('plotbnds(ubdb); show bounds')
drawnow
plotbnds(ubdb),title('Intersection of Bounds');
if pse qpause; end
close(gcf);


% ********************** DESIGN ****************************

disp(' ')
disp('Design')
disp('lpshape(w1,ubdb,nL0,dL0,del0,nc0,dc0); %loop shaping')
disp(' ')

nc0=[];dc0=[]; %                              Initial controller
```

```
% Controller 1 - causes oscillations

%k0 = (.2479*0.00876*3.11/.1961)*(4.789*4.789);
%[nc0,dc0]=zp2tf([-.1961],[-0.008876,-3.11,-0.8438+4.714i,-0.8438-4.714i]',k0);

% Controller 2 - Also Oscillations

%k0 = (.2447*0.00876*8.397*6*6/.1961);
%[nc0,dc0]=zp2tf([-.1961],[-0.008876,-8.397,-4.8+3.6i,-4.8-3.6i]',k0);

%k0 = (.005771*0.0009226*0.4868/.0138);
%[nc0,dc0]=zp2tf([-0.0138],[-0.0009226,-0.4868]',k0);

% Final controller

k0 = (0.001806*1.321*3.631);
[nc0,dc0]=zp2tf([],[-1.321, -1.631]',k0);


w1 = logspace(-5,2,100);  % define a frequency array for loop shaping
nL0=nump(nompt,:); dL0=denp(nompt,:);
del0=0; % no delay
lpshape(w1,ubdb,nL0,dL0,del0,nc0,dc0);
qpause;
numC=nc0; denC=dc0;


% ANALYSIS
disp(' ')
disp('Analysis....')
disp('Re-define a more dense plant template (100 points)....')
drawnow

P=freqcp(nump,denp,w1);
G=freqcp(numC,denC,w1);

disp(' ')
disp('chksiso(1,w1,W1,P,R,G); %margins spec')
drawnow
chksiso(1,w1,W1,P,R,G);
qpause;close(gcf);

disp(' ')
disp('chksiso(2,w1,W2,P,R,G); %output disturbance rejection spec')
```

```
drawnow
ind=find(wl<=10);
W2=abs(freqcp(0.02*[1,64,748,2400],[1,14.4,169],wl));
chksiso(2,wl(ind),W2(ind),P(:,ind),R,G(ind));
qpause;close(gcf);

disp(' ')
disp('chksiso(3,wl,W3,P,R,G); %input disturbance rejection spec')
dr wnow
chksiso(3,wl(ind),W3,P(:,ind),R,G(ind));
qpause;close(gcf);
```

References

[1] Radecki T. An evaluation of the fuzzy set theory approach to information retrieval, in Trappl R., Findler N.V. and Horn W., *Progress in Cybernetics and Systems Research*, Vol. 11 : Proceedings of a Symposium organised by the Austrian Society for Cybernetic Studies, Hemisphere Publishing Company, 1982.

[2] Zadeh, L.A. A fuzzy-algorithmic approach to the definition of complex or imprecise concepts, *Systems Theory in the Social Sciences*, 1976, pp 201.

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

[4] Zadeh, L.A. Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-3, No. 1, 1973, pp 28-44.

[5] Zadeh, L.A. Calculus of fuzzy restrictions. In L.A. Zadeh, K.S.Fu, K. Tanaka and M. Shimura (editors), *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*, Academic Press, 1975, pp 1-39.

[6] Mamdani, E.H. Application of fuzzy algorithms for control of a simple dynamic plant, Proceedings of the IEE, Vol. 112, 1974, pp 743-753.

[7] Mamdani, E.H. and Assilian, S. A fuzzy logic controller for a dynamic plant, *International Journal of Man-Machine Studies*, Vol. 7, 1975, pp 1-13.

[8] Kickert W.J.M. and van Naute Lemke H.R. Application of a fuzzy controller in a warm water plant, *Automatica*, Vol. 12, 1976, pp 301-308.

[9] Pedrycz W. Fuzzy Control and Fuzzy Systems, 2nd (extended) edition, Research Studies Press Ltd. England 1993.

[10] Jager, R. *Fuzzy Logic in Control*, PhD Thesis, Technical University Delft, 1995.

[11] Mendel, J.M. Fuzzy Logic Systems for Engineering : A Tutorial, *Proceedings of the IEEE*, Vol. 83, No. 3, March 1995, pp 345-377.

[12] Kosko, B. *Neural Networks and Fuzzy Systems : A Dynamical Systems Approach to Machine Intelligence*, Prentice Hall Inc, 1992.

[13] Sugeno, M. *Industrial Applications of Fuzzy Control*, Elsevier Science Pub. Co., 1985.

[14] Heckenthaler T. and Engell S. Approximately time-optimal fuzzy control of a two-tank system, *IEEE Control Systems*, June 1994, pp 24-30.

[15] Jang, J.S.R. and Gulley, N. *Fuzzy Logic Toolbox for Use with MATLAB*, The Mathworks Inc., Natick, Mass, 1995.

[16] Shaw, I. Introducing Fuzzy Logic, *Computech*, Vol. 8, No. 3, March-June 1992.

[17] Apptronix Inc, Fuzzy Logic - From Concept to Implementation, Doc. No. EDU01V10-0193, 1993.

[18] von Altrock C. NeuroFuzzy Technologies, Sensor Review, Vol. 12, No. 3., 1994, pp 15-17.

[19] Bannatyne R. Development of fuzzy logic in embedded control, Sensor Review, Vol. 14, No. 3, 1994, pp 11-14.

[20] Kong S.G. and Kosko, B. Adaptive fuzzy systems for backing up a truck-and-trailer, *IEEE Transactions on Neural Networks*, Vol. 3, No. 2, March 1992, pp 211-223.

[21] Li, Y.F. and Lau C.C. Development of fuzzy algorithms for servo systems, *IEEE Control Systems Magazine*, April 1989, pp 65-71.

[22] Pritchard, C.J. Determining signal levels in robustly controlled plants using frequency domain closed loop system specifications, PhD Thesis, University of the Witwatersrand, 1995.

[23] Thomas, D.E. and Armstrong-Hélouvry B. Fuzzy logic control - A taxonomy of demonstrated benefits, *Proceedings of the IEEE*, Vol. 83, No. 3, March 1995, pp 407-421.

[24] Pfeiffer B. and Isermann R. Criteria for successful applications of fuzzy control, *Engineering Applications of Artificial Intelligence*, Vol. 7, No. 3, 1994, pp 245-253.

[25] Kosko, B. *Fuzzy Thinking : The New Science of Fuzzy Logic*, Flamingo, 1994.

[26] Black, M. Vagueness : An Exercise in Logical Analysis, *Philosophy of Science*, Vol. 4, 1937, pp 427-55.

[27] Zadeh, L.A. The concept of a linguistic variable and its application to approximate reasoning, *Information Sciences*, Vol. 8, pp 199-249, and Vol. 9, 1975, pp 43-80.

[28] Sugeno, M. Fuzzy measures and fuzzy integrals : a survey, In M.M. Gupta, G.N. Sardis and B.R. Gaines (editors), *Fuzzy Automata and Decision Processes*, North-Holland Publishers, 1977, pp 89-102.

[29] Wang, L.X. and Mendel, J.M. Fuzzy basis functions, universal approximation and orthogonal least square learning, *IEEE Transactions on Neural Networks*, Vol. 3, 1992, pp 807-813.

[30] Kosko, B. Fuzzy systems as universal approximators, *IEEE International Conference on Fuzzy Systems*, 1992, pp 1153-1162.

[31] Procyk, T.J. and Mamdani, E.H. A linguistic self-organising process controller, *Automatica 15*, 1979, pp 15-30.

[32] Bartos, F.J. Fuzzy logic continues to grow, well guarded by user's wraps, *Control Engineering*, September, 1994, pp 47-51.

[33] Shahian, B. and Hassul, M. *Control System Design using MATLAB*, Englewood Cliffs Prentice-Hall, 1993.

[34] Horowitz, I.M. *Quantitative Feedback Theory (QFT)*, Volume 1, QFT Publications, Boulder, Colarado 80303, 1992.

[35] Azvine, B. and Wynne, R.J. A review of quantitative feedback theory as a robust control system design technique, *Transactions of The Institute of Measurement and Control*, Vol. 14, No. 5, 1992, pp 265-279.

[36] Borghesani, C Chait, Y and Yaniv O. *Quantitative Feedback Theory Toolbox for use with MATLAB*, The MathWorks Inc, 1994

[37] Driankov, D. Hellendoorn, H. and Reinfrank, M. *An Introduction to fuzzy control*, New York, USA, Springer-Verlag, 1993.

[38] Malki, H.A., Li, H. and Chen, G. New design and stability analysis of fuzzy proportional-derivative control systems, *IEEE Transactions on Fuzzy Systems*, Vol. 2, No. 4, pp 245-254.

[39] Johansen, T. A. Fuzzy Model Based Control : Stability, Robustness and Performance Issues, *IEEE Transactions on Fuzzy Systems*, Vol. 2, No. 3, 1994, pp 221-233.

[40] Tanaka, K. and Sano, M. A robust stabilisation problem of fuzzy control systems and its application to backing up control of a truck-trailer, *IEEE Transactions on Fuzzy Systems*, Vol. 2, No. 2. 1994, pp 119-133.

[41] Tanaka, K. and Sugeno, M. Stability analysis and design of fuzzy control systems, *Fuzzy Sets and Systems 45*, 1992, pp 135-156.

[42] Chen, C., Chen P, and Chen, C. Analysis and design of fuzzy control system, *Fuzzy Sets and Systems 57*, 1993, pp 125-140.

[43] Mamdani, E.H. Twenty years of fuzzy control : experiences gained and lessons learnt, *IEEE*, 1993, pp 339-344.

[44] Deshpande, P.B. and Ash, R.H. *Elements of Computer Process Control with Advanced Control Applications*, Instrument Society of America, 1981.

[45] MacLeod, I.M A controller incorporating a fuzzy logic adaptive mechanism for processes with long dead times, Fuzzy Logic in Control Systems, Rand Afrikaans University, Auckland Park, October 1992, South African Council for Automation and Computation.

[46] Rodd, M.G., Verbruggen H.B. and Krijgsman A.J. Artificial intelligence in real-time control, *Engineering Applications of Artificial Intelligence*, Vol. 5, No. 5, 1992, pp 385-399.

[47] Rodd, M.G. Safe AI - is this possible?, *Engineering Applications of Artificial Intelligence*, Vol. 8, No. 3, 1995, pp 243-250.

[48] Umbers I.G. and King P.J. An analysis of human decision-making in cement kiln control and the implications for automation, *International Journal of Man-Machine Studies*, Vol. 12, 1980, pp 11-23.

[49] Bernard, J.A. Use of a rule based system for process control, International Conference on Industrial Electronics, *IEEE Control Systems Magazine*, October 1988, pp 3-12.

[50] Song, J.J and Park, S. A fuzzy dynamic learning controller for chemical process control, *Fuzzy Sets and Systems 54*, 1993, pp 121-133.

[51] Feng, F.C. and Sheng, N.Y. Application of fuzzy logic in the speed control of ac servo systems and an intelligent inverter, *IEEE Transactions on Energy Conversion*, Vol. 8, No. 2, June 1993, pp 312-318.

[52] Assayama, H., Burton, P., Gerstacker, J., Kohono, T. Matsui, S. and O'Lionaird, E. Fuzzy logic control of perlite plant, *Computing and Control Engineering Journal*, IEEE, December 1994, pp 293-298.

[53] D'Azzo, J.J and Constantine, H.H. *Linear Control System Analysis and Design : Conventional and Modern*, McGraw-Hill, 2nd Edition, 1981.

[54] Gordon, A. *The Design and Construction of a Prototype Testbed for Laboratory Experimentation and Controller Testing*, MSc Project Report, University of the Witwatersrand, 1996.

[55] Balchen, J.G. and Mumme', K.I. *Process Control : Structures and Applications*, Van Nostrand Reinhold Company, New York, 1988.

[56] Stephanopoulos, G. *Chemical Process Control : An Introduction to Theory and Practice*, Prentice Hall, New Jersey, 1984.

[57] Horowitz I. Quantitative feedback theory, *IEE Proceedings Pt. D : Control Theory and Applications*, Vol. 129, No. 6, 1982, pp 215-226.

[58] Roffel B. and Chin P. *Computer Control in the Process Industries*, Lewis Publishers, USA, 1987.

[59] Deshpande, P.B. and Ash, R.H. *Elements of Computer Process Control with Advanced Control Applications*, Instrument Society of America, 1981.

[60] Lovrich, A.I. and Hellenthal, B. Servo Control of a DC-brush Motor using Fuzzy Logic, *Quantum*, 1995, pp 21-23.

[61] Efstsathiou J. Expert systems, fuzzy logic and rule-based control explained at last, *Trans. Inst. M.C.*, Vol. 10, No. 4, July-September 1888, pp 198-206.

[62] Redelinghuys D. and Stothert A. The myth-application of fuzzy logic, *Quantum*, June 1995, pp 16-20.

Author: Florence, Stuart Douglas.
Name of thesis: An investigation into the merits of fuzzy logic control versus classical control.

PUBLISHER:
University of the Witwatersrand, Johannesburg
©2015

LEGALNOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page)for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.