Neuro-Fuzzy Predictive Control of an Information-Poor System

Richard Thompson

Trinity College, Oxford

Submitted in partial fulfilment of the requirements for the Degree of Doctor of Philosophy at the University of Oxford.

Hilary Term, 2002

Neuro-Fuzzy Predictive Control of an Information-Poor System

Richard Thompson
Trinity College, Oxford

Submitted in partial fulfilment of the requirements for the Degree of
Doctor of Philosophy at the University of Oxford.

Hilary Term, 2002

Abstract

While modern engineering systems have become increasingly integrated and complex over the years, interest in the application of control techniques which specifically attempt to formulate and solve the control problem in its inherently uncertain environment has been moderate, at best. More specifically, although many control schemes targeted at Heating, Ventilating and Air-Conditioning (HVAC) systems have been reported in the literature, most seem to rely on conventional techniques which assume that a detailed, precise model of the HVAC plant exists, and that the control objectives of the controller are clearly defined. Experience with HVAC systems shows that these assumptions are not always justifiable, and that, in practice, these systems are usually characterized by a lack of detailed design data and a lack of a robust understanding of the processes involved.

Motivated by the need to more efficiently control complex, uncertain systems, this thesis focuses on the development and evaluation of a new neuro-fuzzy model-based predictive control scheme, where certain variables used in the optimization remain in the fuzzy domain. The method requires no training data from the actual plant under consideration, since detailed knowledge of the plant is unavailable.

Results of the application of the control scheme to the control of thermal comfort in a simulated zone and to the control of the supply air temperature of an air-handling unit in the laboratory are presented. It is concluded that precious resources (as measured by actuator activity, for example) need not be wasted when controlling these systems. In addition, it is also shown that a very precise (and sometimes not necessarily accurate) control value computed at each sample is unnecessary. Rather, by defining the system and its environment in the fuzzy domain, the fuzzy decision algorithms developed here may be employed to get an "acceptable" control performance.

Acknowledgements

This thesis would not have been possible without the help and advice from many people whom I have come to know during my career at Oxford. I would therefore like to take this opportunity to express sincere gratitude to a few persons who, in one way or another, assisted in making my years at Oxford fruitful and memorable.

First and foremost, I must thank my supervisor, Dr. Arthur L. Dexter, for his continuous support. In particular, his advice about an approach to postgraduate research in general and a possible approach to neuro-fuzzy control in particular are very much appreciated. He is, and always has been, a constant and unwavering source of motivation.

In addition, my stay at Oxford has been greatly enriched by the many people I have been privileged enough to meet. I am grateful to all the past and present members of the Neuro-Fuzzy and Self-Tuning Control groups; in particular, Woei-Wan Tan, Karl Tredwell, Newton Maruyama, Xiongfu Liu, Darius Ngo, Amanda Lee and Junio Brandizzi for many wonderful discussions of various topics regarding neural networks and fuzzy theory, Rob Bowyer and Renping Liu for their help with the computer system, and Mamdouth Atia for his advice on LATEX. Also, the efforts of many others who have help me in the laboratory in one way or another, including the generous scholarship from the Rhodes Trust, are hereby acknowledged.

Deep appreciation is also felt for the "Caribbean Crew" here at Oxford and in London, Birmingham, Loughborough and Cambridge. They have really assisted in creating that homely environment which was sometimes necessary to help overcome homesickness.

I would also like to acknowledge the encouragement given to me by my mother, Ms. Enid Doyle. Her encouraging words, prayers and additional financial support will never be forgotten. Likewise, I must also acknowledge the support given to me by my family and friends back home in Jamaica for their love, understanding and support. None of this would have been possible without them. I do hope that my small accomplishment so far has made them proud.

Finally, to my close friends here in the UK and in the USA, I say a big thank you. In particular, to Shalani Alisharan, Rafael Cox, Dale Davis, Danielle Jackson, Mariame McIntosh, Zana McIntosh, Erica Mobley, Lindley Nevers, Alicia Reid, Julian Rodney, Tasha Scott, Vasan Seshradi, Arnee Simmons and Cory Williams, words cannot express

my thanks to you for the joy and strength you have given me during my stay at Oxford.

To my mother, Enid.

To laugh often and love much;

To win the respect of intelligent persons and the affection of children;

To earn the approbation of honest critics and to endure the betrayal of false friends;

To appreciate beauty;

To find the best in others;

To give of one's self;

To leave the world a bit better, whether by a healthy child, a garden patch,

or a redeemed social condition;

To have played and laughed with enthusiasm and sung with exultation;

To know that even one life has breathed easier because you have lived

- this is to have succeeded.

Ralph Waldo Emerson (1803-1882)

Contents

List of figures vii

List of tables viii

1 Introduction 1

1.1 Information-Poor Systems 1

1.1.1 Types of System Uncertainties 3

1.1.2 Control of Information-Poor Systems 5

1.1.3 Current Control Practices 5

1.2 Introduction of the Work in this Thesis 10

1.3 Motivation for Work Described in this Thesis 12

1.3.1 Thermal Comfort Control in Buildings 12

1.3.2 Supply Air Temperature Control in HVAC Systems 13

1.4 Objectives of the Work Described in this Thesis 14

1.5 Fundamental Concepts of Fuzzy Sets and Systems 15

1.5.1 Fuzzy Sets 15

1.5.2 Fuzzy Set Representation 16

1.6 Structure and Generation of the Neuro-Fuzzy Model 18

1.6.1 Neuro-fuzzy Model Representation 19

1.6.2 Structural Identification, Training and Fuzzy Identification of Neuro-fuzzy Models 20

1.7 Outline of the Thesis 21

2 Neuro-Fuzzy Predictive Control with Fuzzy Criteria 23

2.1 Introduction 23

2.2 Proposed Control Scheme 25

2.3 The Neuro-Fuzzy Process Model 26

2.4 The Fuzzy Cost Function: Two Basic Approaches 27

2.4.1 Single Fuzzy Cost Function, Single Fuzzy Goal 28

2.4.2 Multiple Fuzzy Cost Functions, Multiple Fuzzy Goals 29

2.5 The Width of the Fuzzy Goal: Defining σ_{ζ_1} 32

2.5.1 Achievable Fuzzy Goals with Preservation of Relative Goal Importance . . . 34
2.6 The Fuzzy Decision-Maker . . . 34
2.7 Concluding Remarks . . . 38

3 Conditional Defuzzification and Fuzzy Proximity **40**
3.1 Introduction . . . 40
3.2 Examining Different Defuzzification Schemes . . . 40
3.2.1 The Traditional Height Defuzzification Scheme . . . 41
3.2.2 The Conditional Defuzzification Scheme . . . 41
3.2.3 The Actuator Activity Parameter, α . . . 43
3.2.4 The Conditional Maximum Possibility Defuzzification Scheme . . . 44
3.2.5 The Conditional Nearest Neighbour Defuzzification Scheme . . . 46
3.2.6 The Conditional Height Defuzzification Scheme . . . 47
3.2.7 The Conditional Linear-Interpolation Defuzzification Scheme . . . 49
3.3 The Fuzzy Proximity Function . . . 51
3.3.1 Fuzzy Proximity Algorithm . . . 53
3.4 Concluding Remarks . . . 59

4 Neuro-Fuzzy Control of Thermal Comfort **60**
4.1 Introduction . . . 60
4.2 Zone Temperature Control Scheme . . . 61
4.3 Simulink Zone Model . . . 62
4.3.1 Simplified Physical Model of the Zone Temperature . . . 63
4.3.2 The Fuzzy Relational Model of the Zone Temperature . . . 70
4.3.3 Training and Identification of the Fuzzy Model . . . 73
4.3.4 The Fuzzy Relational Array, R . . . 75
4.4 Multiple Fuzzy Cost Functions Used in the Zone Temperature Control Scheme . . . 78
4.5 Fuzzy Prediction with a Fuzzy Input . . . 79
4.5.1 Developing the Fuzzy-Input Neuro-fuzzy Model . . . 79
4.6 The Fuzzy Goals . . . 81
4.7 Zone Control Results . . . 84
4.7.1 PI Control of Zone . . . 85

4.7.2 NFMBP Control of Zone Temperature: Applications in Two Different Zones Control Modes 87
4.8 Concluding Remarks 90

5 The Control of Supply Air Temperature using Generic Fuzzy Models 92
5.1 Introduction 92
5.2 Laboratory Test Rig Used in the Application of Supply Air Temperature Control Scheme 93
5.2.1 The Cooling Coil Subsystem 94
5.3 Implementation of Control Scheme 95
5.4 Description of Experiments Carried out on Test Rig 96
5.5 PI Control of Supply Air Temperature 97
5.6 Modelling the Cooling Coil Subsystem 97
5.7 Developing the Generic Neuro-Fuzzy Model of the Cooling Coil 99
5.7.1 Structure of the Neuro-Fuzzy Model 101
5.7.2 Training and Identification of the Neuro-Fuzzy Model using Generic HVACSim+ Data 103
5.7.3 The Fuzzy Goal Function 108
5.7.4 Fuzzy Predicted Temperatures from the Generic Model 110
5.8 Results of Experiments on the Test Rig 111
5.8.1 Intermediate Experimental Results: Justification for the Inclusion of Normalized Air Mass Flow Rate in Model 111
5.8.2 Final Experimental Results 112
5.9 Concluding Remarks 118

6 Selection of Application Dependent Parameters 121
6.1 Introduction 121
6.2 The Effect of Different Defuzzification Schemes on the Value of the Control Signal 121
6.2.1 Simple Matlab Zone Simulation for Illustration of the Effect of Different Defuzzification Schemes 122
6.2.2 Control Plots under Different Defuzzification Schemes 123
6.3 Sensitivity of Control Activity to the choice of the Actuator Activity Parameter, α 124
6.4 Choosing σ_c and σ_e 126

6.5 Concluding Remarks 128

7 Conclusions and Future Work **134**
7.1 Conclusions 134
7.2 Suggestions for Future Work 136

A Fuzzy Arithmetic **138**
A.1 Introduction 138
A.2 Fuzzy Arithmetic 138
A.2.1 Fuzzy Numbers and Fuzzy Intervals 138
A.2.2 Arithmetic Operations on Fuzzy Numbers and Fuzzy Intervals . 139

B Fuzzy Relational Arrays and The RSK Training and Identification Scheme **143**
B.1 Introduction 143
B.2 The RSK Identification Scheme 143
B.3 The Modified RSK Identification Scheme 144
B.4 A Section of the Fuzzy Relational Array used in Chapter 4 145

C Matlab Codes used in the Implementation of the Control Scheme **147**
C.1 Introduction 147
C.1.1 Matlab Implementation of the Control Scheme 147
C.1.2 Fuzzy Proximity Matlab Code 152
C.1.3 Matlab Codes used in the Implementation of the different Defuzzification Schemes 153
C.1.4 Matlab Code used in the Implementation of the Post-Processing Algorithm 157
C.1.5 Random Training Matlab Code 157
C.1.6 Simple Matlab Zone Simulation/Controller used in Chapter 6 . . 162

D Simulink Zone Model of Thermal Behaviour of Zone **170**
D.1 Four High-Level Schematics of Simulink Zone Model 170

References **170**

List of Figures

1.1 Different Types of Uncertainties 4
1.2 Model-Based Predictive Control 10
1.3 Crisp and Fuzzy Sets 16
1.4 Continuous and Discrete Fuzzy Sets 17
1.5 An Example of a Convex and a Non-Convex Fuzzy Set 18

2.1 The Two Basic Approaches to the Fuzzy Cost Function 24
2.2 Neuro-Fuzzy Model-Based Predictive Control 25
2.3 The Relative Importance of Two Fuzzy Goals 31
2.4 Defining the Width of a Triangular Fuzzy Goal 32
2.5 Defining the Width of a Trapezoidal Fuzzy Goal 33
2.6 Fuzzy Matching used in the Fuzzy Decision-Making Scheme 35

3.1 Some Problems with Height Defuzzification 42
3.2 Illustrating Different Defuzzification Schemes 45
3.3 Illustrating the CHD Scheme 48
3.4 Illustrating the CLID Scheme 49
3.5 Fuzzy Subtraction of Two Fuzzy Numbers: One Contained within the Next 52
3.6 Fuzzy Proximity of Two Fuzzy Numbers: One Contained within the Other 53
3.7 Fuzzy Proximity of Two Fuzzy Numbers 54
3.8 Fuzzy Proximity of Two Fuzzy Numbers 55
3.9 The Z space of Fuzzy Proximity 56

4.1 Fuzzy Model-Based Comfort Control 61
4.2 Simulink Model: Lowest Level (level 5) of Zone Model 63
4.3 Simple Linear Thermal Model of Zone 65
4.4 Neuro-Fuzzy Model of Zone Temperature 71
4.5 Fuzzy Reference Sets Defined on the Input and Output Spaces 72
4.6 Prediction Plots of the Neuro-Fuzzy Model 76
4.7 Prediction Plots of the Neuro-Fuzzy Model 77
4.8 Fuzzy-Input Neuro-Fuzzy Prediction 80
4.9 Fuzzy Goals Defined on Comfort and Energy 82
4.10 Achievable Fuzzy Goal 83

4.11 Ambient Temperature Variation during Simulation 84
4.12 Direct Normal External Solar Heat Radiation during Simulation 85
4.13 PI Control with $k_p = 0.05\,\mathrm{kgs}^{-1}\ ^{\circ}\mathrm{C}^{-1}$, I=10 s 86
4.14 NFMBPC Scheme with $\sigma_c = 1$, $\sigma_e = 10$ 88
4.15 NFMBPC Scheme with $\sigma_c = 2$, $\sigma_e = 4$ 89

5.1 Schematic Diagram of the Laboratory Air-Handling Unit 93
5.2 Schematic Diagram of the Cooling Coil Subsystem of the Test Rig 95
5.3 PI Controller for Supply Air Temperature Control 97
5.4 Generating Generic Fuzzy Model of the Cooling Coil 100
5.5 Training on D1, D2 and D3 102
5.6 Neuro-Fuzzy Model of Supply Air Temperature 103
5.7 Fuzzy Reference Sets defined on the Normalized Input and Output Spaces 104
5.8 Post-processing of Fuzzy Predicted Outlet Air Temperature 106
5.9 Pseudo-Random Control Sequence used and Corresponding Outlet Air Temperature observed during Training 107
5.10 An Example of the Fuzzy Goal Function Defined on Supply Air Temperature 109
5.11 Fuzzy Predicted Temperature when $\dot{m}_a(n) = 0.4$ kg/s and $T_{ao}(n) = 14\ ^{\circ}\mathrm{C}$. 111
5.12 Fuzzy Predicted Temperature when $\dot{m}_a(n) = 1.0$ kg/s and $T_{ao}(n) = 14\ ^{\circ}\mathrm{C}$. 112
5.13 Intermediate Results: Supply Air Temperature and Control Action Plots at **High Air Flow Rate**. 113
5.14 Intermediate Results: Supply Air Temperature and Control Action Plots at **Low Air Flow Rate**. 114
5.15 Supply Air Temperature Plots: PI Control vs. NFMBP Control of Supply Air Temperature at **High Air Flow Rate**. 115
5.16 Control Activity Plots: PI Control vs. NFMBP Control of Supply Air Temperature at **High Air Flow Rate**. 116
5.17 Supply Air Temperature Plots: PI Control vs. NFMBP Control of Supply Air Temperature at **Low Air Flow Rate**. 118
5.18 Control Activity Plots: PI Control vs. NFMBP Control of Supply Air Temperature at **Low Air Flow Rate**. 119

6.1 Control Plots using the CLID Scheme, $\alpha = 0.8$ 123
6.2 Control Plots using the CHD Scheme, $\alpha = 0.8$ 124
6.3 Control Plots using the THD Scheme, $\alpha = 0.8$ 125
6.4 Control Plots using the CMPD Scheme, $\alpha = 0.8$ 126

6.5 Control Plots using the CNND Scheme, $\alpha = 0.8$ 127
6.6 $\sigma_c = 1$, $\sigma_e = 4$, $\alpha = 0.8$ 129
6.7 $\sigma_c = 1$, $\sigma_e = 4$, $\alpha = 0.6$ 130
6.8 $\sigma_c = 1$, $\sigma_e = 4$, $\alpha = 0.4$ 131
6.9 $\sigma_c = 1$, $\sigma_e =3$, $\alpha = 0.8$ 131
6.10 $\sigma_c = 1$, $\sigma_e = 4$, $\alpha = 0.8$ 132
6.11 $\sigma_c = 1$, $\sigma_e = 10$, $\alpha = 0.8$ 132
6.12 $\sigma_c = 2$, $\sigma_e = 10$, $\alpha = 0.8$ 133

A.1 A Comparison of (a) a Real Number and (b) a Crisp Interval with (c) a Fuzzy Number and (d) a Fuzzy Interval 140

D.1 Simulink Model: Upper Level (level 1) of Zone Model 171
D.2 Simulink Model: Middle Level (level 2) of Zone Model. Zone 3 is the one of interest in this work. 171
D.3 Simulink Model: Lower Level (level 3) of Zone Model 172
D.4 Simulink Model: Penultimate (Sub) Level (level 4) of Zone Model 172

List of Tables

4.1 Design Information of Simplified Physical Zone Model 70
4.2 Section of Training Data for Zone Model 73
4.3 Average Energy Consumption and Control Activity Associated with the Three Different Control Schemes 90
4.4 Relative Energy Consumption and Control Activity Associated with the Three Different Control Schemes (PI Control = 100%) 90

5.1 Variables Associated with the Cooling Coil Subsystem 96
5.2 Design Parameters for each Cooling Coil Design 99
5.3 Detailed Coil Data for the Cooling Coil Subsystems of D1, D2 and D3 . . 101
5.4 Maximum Percentage Activation of Fuzzy Input Reference Sets by Training Data: Training at 0.4 kg/s on Coil Design D1 108
5.5 Relative Control Activity Associated with Both Control Schemes 117

6.1 Comfort and Energy Tolerance Profile of Matlab Simulated Zone 122
6.2 Control Activity and Energy Consumption typical of Different Defuzzification Schemes 128
6.3 Varying α: **Average** Energy Consumption and Control Activity 128
6.4 Varying α: **Relative** Energy Consumption and Control Activity 129
6.5 Varying σ_c and σ_e: **Relative** Energy Consumption and Control Activity . 133

Chapter 1

Introduction

In control theory, there are a number of approaches that present themselves as candidates to be used in solving the control problem. The choice of controller design is very much dependent on, among other things, the nature of the control problem at hand as well as the amount of information we have available about the process under control; including process dead-times, process order, process gain and dynamics and system disturbances. Certainly, some control design techniques are more robust to changes and uncertainty in these variables than others, while other choices in controller design may place more emphasis on optimizing transient response and minimizing steady-state offset errors. Choosing a controller design is therefore usually a trade-off between robustness and controller performance.

1.1 Information-Poor Systems

Non-linearities, uncertainties, and complexity are playing an increasing role in a wide spectrum of control problems in new technologies (e.g., manufacturing, robotics, medical instrumentation, space technology) as well as in older technologies (e.g., aircraft control, process control). In this work, a control scheme is formulated which attempts to address some of the problems which might be encountered in the control of systems where the values of certain variables may have some degree of uncertainty associated with them. These systems are usually referred to as uncertain systems, but, because the control scheme developed here is also applicable to systems about which there is a general lack of design information, the systems considered here will be referred to by the more generic name of *Information-Poor Systems.*

To various degrees, there is a lack of information about all real, complex systems. None of these systems can be totally defined, and one can never truly claim to know

everything about any given one of these systems. However, while it is difficult to define succinctly what one means by a system being information-poor, certain systems, more so than others, are so poorly defined and understood that they may arguably be classified as Information-Poor Systems.

One definition of an Information-Poor System is one in which certain variables and parameters which characterize the system are ill-defined and/or have a significant amount of vagueness associated with them. In addition, a system which may be too complex to have its behaviour modelled to a fair degree of certainty may also qualify as an information-poor system (Howell, 1994). Information-poor systems are also characterized by a lack of design data. For example, in a Heating, Ventilating and Air Conditioning (HVAC) system, certain physical quantities of the cooling coil subsystem, such as the number of rows of tubes and the number of tubes per row, may be difficult to determine several years after the coil has been installed. Howell (Howell, 1994) describes information-poor plants as those which may have a bare minimum of sensors available with which to operate the process, or as those plants for which the sensors are sampled at frequencies which are low relative to the dynamics of the plant. It might be difficult to increase this frequency because the data might include off-line chemical analyses, for example, or necessitate the purchase of considerably more expensive on-line instrumentation. Even if enough sensors are available, the sensors measuring particular variables of the system may not be accurate enough to produce data of sufficiently high quality or the variables may not be able to be measured directly by the sensors available. For example, the accurate measurement of air and water flows in ducts or pipes may be extremely difficult and expensive due to turbulence caused by fans and pumps and due to physical conditions such as non-uniformity and bends in the pipes.

As implied above, a good example of an information-poor system is an HVAC system. HVAC systems are highly non-linear and usually less precisely defined than other electrical or mechanical systems (Isermann, 1995). Additionally, they often lack sufficient design data and there are usually considerable constraints imposed by the non-ideal behaviour of sensors and of actuators such as valves and dampers (So, Chan and Tse, 1997). For example, valves and dampers in a typical HVAC system may suffer from hysterisis and some amount of leakage and there often are constraints imposed by rate limits. These factors in addition to the fact that the control objective is seldom precisely defined, make HVAC systems good candidates to be considered as information-poor systems. The work developed in this thesis is applied to HVAC systems and subsys-

tems.

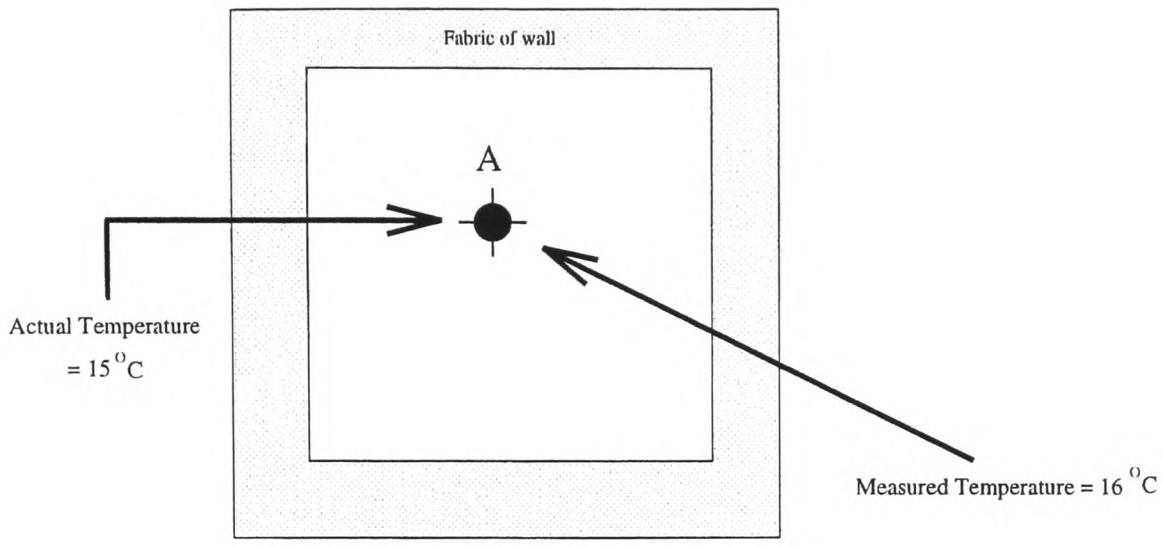
1.1.1 Types of System Uncertainties

In general, uncertainties common in information-poor systems may be categorized into three distinct groups.

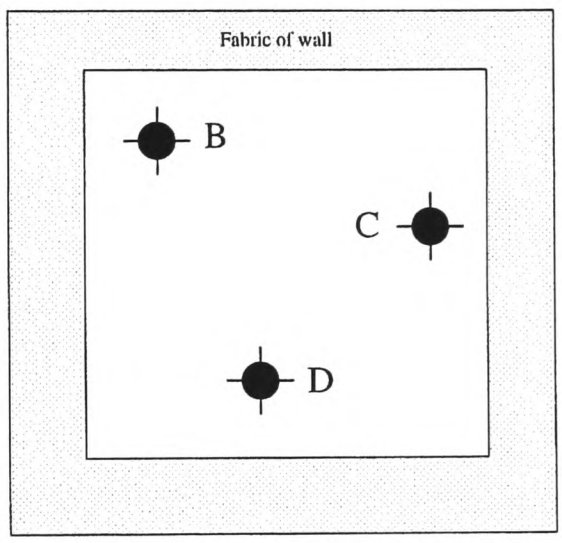
The first group involves measurement uncertainties due to sensor imperfections. For this class of uncertainties, even if it is known exactly what is being measured, inherent biases and imperfections in the device and/or the method used for acquiring the measurement represent the main source of uncertainty. For example, in Figure 1.1(a), which shows a room in a house, even though the temperature at point A is 15°C, the thermometer may measure it as 16°C due to inaccuracies inherent in the measuring device.

The second group involves uncertainty surrounding the quantity actually being measured. For this class of uncertainties, even if our measuring instrument and method are "perfect", uncertainties arise because it is not known which quantity is actually being measured (as opposed to which quantity is desired to be measured). For example, consider Figure 1.1(b). Three "perfect" thermometers are positioned and spatially separated such that the temperature at different points in the room volume may be obtained. If the "room temperature" is to be measured and controlled in order that its occupants may experience the least discomfort, there may be uncertainty as to which thermometer best quantifies this desired measurement. Even if a weighted average of all three thermometers is taken, the fact that there may be temperature stratification of the air in the room and, therefore, that there may be uncertainty surrounding the extent to which the temperature reading of each thermometer should contribute to this average temperature (taking into consideration the purpose for which this temperature reading will be employed), makes the quantity "room temperature" a difficult one to measure.

The final category of system uncertainty involves uncertainty in the relationship between variables of the system. In systems of this type, even if our sensor and method used for acquiring the measurement are "perfect" and it is known with absolute cer-



(a) Discrepancies between Actual and Measured Temperatures



(b) Which Sensor measures "Room Temperature"?

Figure 1.1: Different Types of Uncertainties

tainty which quantities are being measured, the way in which these quantities act together to determine some output which is itself a function of them, is uncertain.

Usually, any given system under control will possess, to some degree, one or more of these types of system uncertainties. Information-poor systems, by definition, will contain some or all of these system uncertainties acting together to an appreciable extent.

1.1.2 Control of Information-Poor Systems

Because information-poor systems are so ill-defined, controlling them tends to present a problem. Control techniques which require a mathematical model of the process (Peitsman and Soethout, 1997) usually prove difficult to formulate, since accurate modelling of these information-poor systems is impossible.

Generally, adaptive or robust control would probably be the appropriate technique for problems where a mathematical model of the system is available. These control techniques usually assume a linear plant, and derive linear controllers accordingly (Leigh, 1992). However, complex information-poor plants tend to be so non-linear that these linear techniques cannot effectively be applied since the inherent non-linearities of the plant degrade the theoretical performance of these linear controllers. In addition, the uncertainty in the information-poor plants being considered makes it difficult to define the bounds on the transfer function with the fair degree of certainty required of robust control. Non-linear techniques have therefore emerged as one possible solution to controlling these systems. According to Bremner *et al.* (Bremner and Postlethwaite, 1997), the only way to completely justify the use of a non-linear technique is to have an expert in linear control techniques spend the same resources in time and computing power, and then compare the performance in the resulting controllers, but this is completely impractical in industrial studies. They continue, "A more pragmatic approach when deciding whether to use non-linear techniques on processes which are suspected to possess significant non-linearities is to ask why not?"

1.1.3 Current Control Practices

A number of approaches to the control of information-poor systems have been attempted in practice. For example, in the HVAC industry, it is common practice to use a single conservatively tuned Proportional-plus-Integral (PI) or PI-plus-Derivative (PID) controller to operate the air-handling unit (Wang et. al., 2001). Predictive, self-tuning and adaptive control techniques have also been applied to HVAC processes (De Keyser, 1985), (Nesler, 1986), (Radke and Isermann, 1987), (Dexter and Haves, 1989), (Wallenborg, 1991). Predictive control techniques have increased in popularity in recent years (Virk, Cheung and Loveday, 1991), however, the problems associated

with obtaining a suitable process model for prediction remain. To overcome this problem, the approach has therefore been to detune the controller making it more robust to uncertainties in the model. This is done at the cost of reduced performance. In addition, gain scheduling schemes have been proposed for HVAC applications (Geng and Dexter, 1990), but the expected improvement in controller performance has been disappointing. Ling and Dexter (Ling and Dexter, 1994) have proposed expert control of an air-conditioning plant using *a priori* knowledge and based on the generalized predictive control approach (Clarke, Mohtadi and Tuffs, 1987). Although minimal effort was required to design the controller (an important consideration in the HVAC industry where detailed system modelling is usually not economical), the resulting regulation performance of the controller was no better than that of a PI controller.

Because of the practical problems associated with these methods, research interests have turned to other promising approaches; more specifically, approaches involving artificial intelligence techniques.

Artificial Neural Network Control

Recent developments in artificial intelligence techniques have found applications in the control field (Curtiss et. al., 1994). One such technique involves the use of an Artificial Neural Network (ANN) to model the behaviour of a highly non-linear plant, with the objective of adequately controlling this plant by embedding the ANN predictor in a predictive control structure (Curtiss et. al., 1993).

Hao *et al.* (Hao et. al., 1994) obtained promising results by applying an ANN to some rather academic and analytical control problems. However, the results were less encouraging when So *et al.* (So, Chan and Tse, 1997) used an ANN in a self-learning fuzzy control scheme for the control of a standard air-handling unit in a more realistic HVAC system.

Hunt *et al.* (Hunt and Sbarbaro, 1991) investigated the use of ANN for non-linear internal model control of dynamic systems. Their analysis, though convincing for the simulated plant they considered, assumes that the plant inverse exists; an assumption not necessarily true for many real systems. Additionally, Willis *et al.* (Willis et. al., 1991) examined the uses of ANN in process engineering. They concluded that ANNs

had the capability of providing 'fast' inferences of important, but 'difficult-to-measure', process outputs from other easily measured variables. However, they cited the lack of inherent dynamic modelling in the neural network structures they had considered as a disadvantage since, in many practical situations, dynamic relationships exist between inputs and outputs. As such, the ANN considered failed to capture the essential characteristics of the system.

Artificial neural network control of information-poor non-linear dynamic systems also suffers from the problem of not being transparent enough to facilitate easy translation of the inner workings of the network into a form that can be understood by the operators of the plant. In addition, the problem of finding the optimal network topology for modelling a given system exists.

Fuzzy Logic Control

Fuzzy Logic Control is yet another product of the developments made in artificial intelligence techniques. When mathematical models are not available, or when the process behaviour is only partially understood, fuzzy logic control can be used to control the systems (Cao et. al., 1997). Fuzzy logic was first applied to control systems by Ebrahim H. Mamdani and his colleagues (Mamdani, 1974), (Mamdani and Assilian, 1975). The proposed controller used direct control to achieve the control objective, and was similar in structure to the PI control system used in industry.

The fuzzy controller has successfully controlled complex industrial processes such as cement kilns (Norman and Naveed, 1985). Funakoshi and Matsuo (Funakoshi and Matsuo, 1995) developed a fuzzy controller which used a thermal comfort index known as the Predicted Mean Vote (PMV) to control the temperature in a passenger train. The structure was again similar to that of the PI controller. Simulation results showed that acceptable control was achieved by applying this method. However, the method used to generate the rules for this controller was unclear. In addition, the use of fuzzy logic to control thermal comfort has been proposed in (Gouda, Danaher and Underwood, 2001), where PMV was controlled. The results presented were convincing for the simulated system considered, however, no attempt was made at applying the scheme to an actual zone.

Fuzzy control has also been proposed for use in building applications, as in (Sousa, Babuska and Verbruggen, 1997). In (Bruant et. al., 1994), Bruant *et al.* used fuzzy control to control the indoor air quality in a naturally ventilated building where the window opening area was selected as the control variable. Additionally, Huang and Nelson (Huang and Nelson, 1994) applied fuzzy control to an HVAC system. Although the controller outperformed the PID controller used as comparison, the method suffered from a lack of a systematic way of generating the rules for the controller. In fact, rule generation appears to be one of the more serious problems involved with fuzzy controllers. This is so because it is often difficult to express human experience exactly using linguistic rules in a simple form. Moreover, human experience and expert knowledge may differ from one expert to another, so that there may be a lack of consistency in rule generation. Because of this, fuzzy logic controllers sometimes fail to obtain satisfactory results (Huang and Nelson, 1999) and may be quite time consuming to design.

Neuro-Fuzzy Control

The strength of ANNs lies in the fact that they have strong modelling capabilities, especially for complex systems not amenable to mathematical analysis. They do so by observing model output data from given input data, and adjusting certain internal parameters (weights) in order to allow this model output data to match the desired output data from the actual system (training). The main appeal of fuzzy logic, as it pertains to control, is its ability to deal with situations where the available sources of information are inaccurate, subjectively interpreted or uncertain. Neuro-fuzzy control has therefore emerged as a control technique which combines the ability of fuzzy logic to handle unquantifiable uncertainties in the control scheme, (Sugeno, 1985), (Terano, Asai and Sugeno, 1994) with the strong modelling and knowledge extraction capabilities of ANNs. For this reason, the technique seems to be an effective control scheme for handling information-poor systems; especially if they are non-linear (Sing and Postlethwaite, 1997). Also, the fuzzy relational model employed in some neuro-fuzzy based control schemes can easily be identified from process data so that no detailed knowledge of the process dynamics is required (Sing and Postlethwaite, 1997), and the structure of this fuzzy relational model is more transparent compared to modelling based on the use of neural networks alone, for example in (Sousa, Babuska and Verbruggen, 1997),

where neuro-fuzzy modelling is applied to the control of HVAC systems.

Postlethwaite (Postlethwaite, 1994) proposed a model-based neuro-fuzzy controller in which he developed a neuro-fuzzy model of the process under control, and embedded it in a relatively conventional model-based controller. Neuro-fuzzy training and identification of the predictive fuzzy relational process model involved making open loop variations in the controlled variable. The technique was applied to the simulated liquid level control used by Graham and Newell (Graham and Newell, 1989) and it was shown that the Integral Absolute Error (IAE) was better than the best results obtained in (Graham and Newell, 1989). Although the particular simulated system might not be considered information-poor, the results were encouraging, and later, Bremner and Postlethwaite (Bremner and Postlethwaite, 1997) successfully applied the technique to the control of a grain dryer; a highly non-linear information-poor system found in the chemical industry.

Karr *et al.* (Karr and Gentry, 1993) have investigated neuro-fuzzy control of a pH system using genetic algorithms. They applied a technique for producing adaptive fuzzy logic controllers in which genetic algorithms were used to alter fuzzy membership functions on-line. However, because genetic algorithms optimize by selectively searching the solution space based on the mechanics of natural genetics, this technique may be unsuitable for on-line applications since intermediate control values generated while searching could possibly cause the process to become unstable.

Sing *et al.* (Sing and Postlethwaite, 1997) also used neuro-fuzzy techniques in their investigation of pH control. They showed that neuro-fuzzy control could adequately cope with the significant non-linearity and deadtime exhibited by the neutralization process and suggested that the ease of implementation of the control scheme allows it to be extended to other processes with unknown dynamics. However, they also stated that for processes with highly changing dynamics such as the pH control of a wastewater treatment plant, the fixed model controller suggested might not offer sufficiently acceptable control and an adaptive version would need to be developed.

Neuro-fuzzy control has also been applied to a building energy management system by Glorennec (Glorennec, 1991). He developed a controller that learned to identify the rules it must use (rule extraction) by employing a neural network. He applied this controller to a simulated building thermal regulation problem. Although the technique

provided acceptable control, the building simulation model used was quite crude, and therefore the results implied very little about the application to a real system.

1.2 Introduction of the Work in this Thesis

The work in this thesis attempts to contribute to some of the current techniques of neuro-fuzzy modelling and predictive control, and, in so doing, develop a control scheme more appropriate for application to the task of controlling an actual information-poor system. The general approach employed is known as Neuro-Fuzzy Model-Based Predictive Control (NFMBPC).

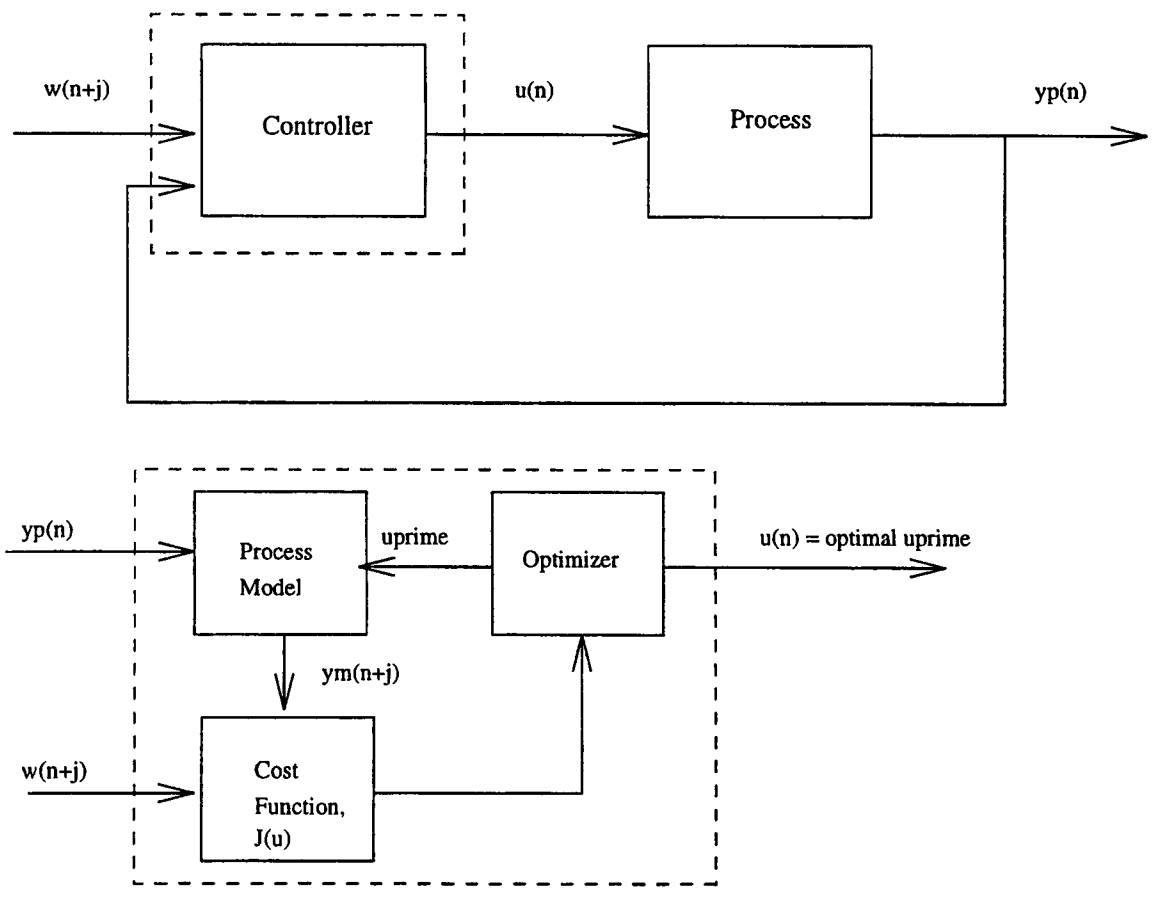


Figure 1.2: Model-Based Predictive Control

Model-Based Predictive Control (Figure 1.2) seeks to provide acceptable controller performance by predicting the output of the process under control at some time $n+j$ in the future, where n represents the current sampling instant, and $j = 1, 2, 3 \cdots$. This prediction is usually based on an assumed model of the process and on an assumed

scenario for the future control signals (Astrom and Wittenmark, 1989). Having a value for the predicted output of the plant at time $n+j$, the controller then chooses a control signal to drive this future plant output to the setpoint at time $n+j$ sampling intervals in the future, $w(n+j)$, taking into account the control activity required to do so.

A fundamental concept of predictive control theory is that a control horizon is assumed beyond which all control *increments* become zero (Clarke, Mohtadi and Tuffs, 1987). Also, the receding-horizon approach is normally used in current predictive control applications whereby, although a new sequence of control actions is calculated at each sampling instant when a new measurement of process output is obtained, only the first one is applied to the process, and the optimization is repeated at the next sampling instant (Astrom and Wittenmark, 1989).

In such a scheme, the control cost to be optimized is given by:

$$J(\mathbf{u}) = \sum_{j=N_1}^{N_2} (\mathbf{w}(n+j) - \mathbf{y_p}(n+j))^2 + \lambda \sum_{j=1}^{N_u} (\mathbf{u}(n+j-1) - \mathbf{u}(n+j-2))^2 \qquad (1.1)$$

where $\mathbf{w}$ and $\mathbf{y_p}$ are the shaped setpoint trajectory and actual process output, respectively, $\mathbf{u}$ is the vector of control signals, λ is a weighting factor on changes in the control action and N_1 and N_2 are the minimum and maximum costing horizon, respectively (Clarke, Mohtadi and Tuffs, 1987). N_u is known as the control horizon. Since $\mathbf{y_p}(n+j)$ is not known at the current sample, its predicted value at this current sample time, $\hat{\mathbf{y}}(n+j \mid n) = \hat{\mathbf{y}}(n+j)$ is used. If the model used to generate this prediction is a neuro-fuzzy model, the approach is known as NFMBPC.

For a single-input-single-output (SISO) system employing such a scheme, consider a two-step-ahead prediction model ($N_1 = 1, N_2 = 2$). The controller of such a scheme consists of three parts as shown in Figure 1.2(b): i) an assumed process neuro-fuzzy model, ii) a crisp cost function to be optimized, $J(u)$, and iii) an optimizer (an algorithm used to find the control signal, $u_{optimal}$, corresponding to the optimal value of $J(u)$). The controller therefore accepts the current value of process output along with the next value of setpoint to produce the optimal current control action that would, theoretically, drive the process to the setpoint in two steps. Often, the optimizer uses direct search optimization (e.g. Sequential Quadratic Programming outlined in (Grace,

1992)), and so, assuming a control horizon, N_u, of 1, the cost function,

$$J(u) = (w(n+1) - \hat{y}(n+1))^2 + (w(n+2) - \hat{y}(n+2))^2 + \lambda(\Delta u(n))^2 \qquad (1.2)$$

must explicitly be defined. In addition, variables such as the setpoint, the control action and both the one-step-ahead and two-step-ahead predicted outputs are all in the crisp domain.

The control scheme being proposed investigates the use of fuzzy definitions of these variables (with the exception of the final control action applied to the process, which, by nature of current actuator technology, must be in the crisp domain [1]). The motivation for this approach is outlined below.

1.3 Motivation for Work Described in this Thesis

The work outlined and developed in this thesis is motivated by both the "Thermal Comfort Control (TCC) problem" in buildings, and the control of supply air temperature in HVAC systems. Recognizing that such problems involve the control of a complex system in which both the system variables and control objective may be poorly defined, the scheme attempts to provide acceptable control without the need for detailed system modelling, and with an emphasis on reducing system resources suspected of being wasted by the application of conventional techniques.

1.3.1 Thermal Comfort Control in Buildings

The thermal comfort of the occupants of a zone depends on many factors including metabolic rates, clothing, air temperature, mean radiant temperature, air velocity and humidity (Fanger, 1982). Controlling thermal comfort, therefore, usually involves the

[1] An interesting discussion of an alternative to conventional sensor/actuator technology, called a *fuzzy sensor*, is given in (Mauris, Benoit and Foulloy, 1997).

dauntingly difficult task of controlling one or more of these variables. In most buildings, however, only temperature and humidity can be controlled. Indeed, in many European buildings, over a wide range of humidities, only zone temperature is controlled (Thompson and Dexter, 2001).

In addition to the uncertain non-linear system behaviour usually associated with comfort control in buildings, the TCC problem is also characterized by inherent uncertainties present in the control environment and by a lack of design data as well, as discussed in Section 1.1. In addition, the control objectives of the TCC problem are usually poorly defined - the relative importance of maintaining thermal comfort of the occupants and of minimizing energy consumption may not be known precisely or may even be time-varying in some unknown manner. This trade-off between comfort and energy consumption is often realized by choosing appropriate setpoints for the local controllers, and is sometimes left entirely to the operators (Kummert et. al., 1997). The conventional approach to controlling these so-called information-poor systems (Howell, 1994) may not necessarily be the most efficient since valuable computer resources are usually expended in calculating precise values in an inherently imprecise environment.

1.3.2 Supply Air Temperature Control in HVAC Systems

The control of supply air temperature in HVAC systems can also be quite difficult to achieve, again because of the uncertainties and non-linearities that usually characterize these systems (Sousa, 1998). With regards to the work presented here, the control of supply air temperature on an experimental air-handling unit is particularly attractive because it provides a means of assessing the feasibility of the proposed control scheme on a real system. Motivated by the unique challenges posed by such a practical problem (as opposed to merely a theoretical or simulated one), the work developed here seeks to contribute to the pool of knowledge already established regarding the treatment of various practical issues encountered when attempting to control these systems.

1.4 Objectives of the Work Described in this Thesis

As an alternative to conventional approaches, the work outlined in this thesis proposes the formulation of information-poor problems in the natural, imprecise and fuzzy states that characterize these problems. Then, capitalizing on the advantages of neuro-fuzzy modelling and fuzzy decision-making techniques, it seeks to solve the control problem of these systems.

More specifically, the main objectives of this work are to examine the major problems associated with conventional control techniques as they pertain to the control of non-linear, information-poor dynamic systems, and to investigate the use of fuzzy decision-making techniques in the solution of this type of problem. Having gained an appreciation for both the limitations and advantages of model-based predictive control, a new form of this technique capable of handling fuzzy definitions of the input/output variables and of the control objective, will be developed. The control scheme will be applied to a zone comfort control problem, where energy considerations will also be taken into account. To test the controller's performance in a real-world, information-poor environment, and to get an indication of its robustness in the presence of disturbances, the scheme will also be applied to the control of an experimental air-handling unit. These problems represent environments in which the control objective is not precisely defined and for which detailed system modelling would not be financially feasible because of the uncertainties involved in the systems.

In addition, a form of defuzzification more appropriate for fuzzy decision-making under uncertainty will be developed. It will be shown that this defuzzification scheme has distinct advantages over conventional schemes, when used in the proposed approach to controller design.

Finally, the feasibility of using generic neuro-fuzzy models to control uncertain systems will be demonstrated.

As a preamble to the work presented in the following chapters, Section 1.5 below introduces some basic concepts of fuzzy sets and systems.

1.5 Fundamental Concepts of Fuzzy Sets and Systems

1.5.1 Fuzzy Sets

Fuzzy sets are a generalization of conventional set theory that were introduced by L.A. Zadeh in 1965 as a mathematical way to represent vagueness in everyday life (Bezdek, 1993).

Conventional set theory divides the elements of some given universe of discourse into members and non-members; two distinct groups. Thus, if a particular element is not a member, it is necessarily a non-member (Figure 1.3). However, human beings tend not to be as distinct in the classification of elements. Rather, we use terms such as *tall* people and *hot* day, where membership in the classification of *tall* and *hot* may vary from person to person; concepts which are quite vague and ambiguous. In an attempt to capture this spirit of human classification and reasoning, fuzzy set theory removes the sharp line of demarcation between members and non-members of a given set and instead, replaces it with a more gradual transition between the two as in Figure 1.3.

A fuzzy set is characterized by a membership function which is a representation of the degree of membership of the elements of the universe of discourse in that fuzzy set. This degree of membership is equivalent to the extent to which it is believed that the elements are similar or comparable with the concept represented by the fuzzy set. Therefore, elements having a higher degree of membership are closer to the concept represented by the fuzzy set than elements having a lower degree of membership. The degree of membership is usually some real number on the closed interval [0,1] as shown in Figure 1.3. Further, in fuzzy systems (systems using fuzzy set theory and fuzzy logic to reason), the fuzzy sets are usually assigned some linguistic term such as "negative small" (or NS for short) and the degree of membership of the independent variable x in the fuzzy set NS is written $\mu_{NS}(x)$.

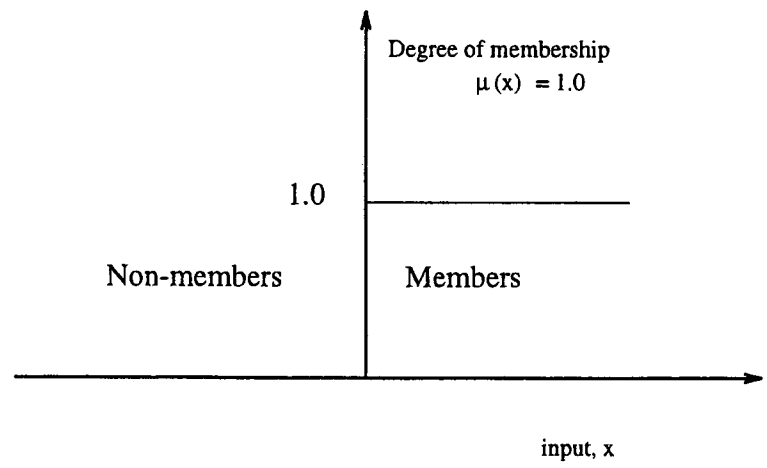


Figure 1.3: Crisp and Fuzzy Sets

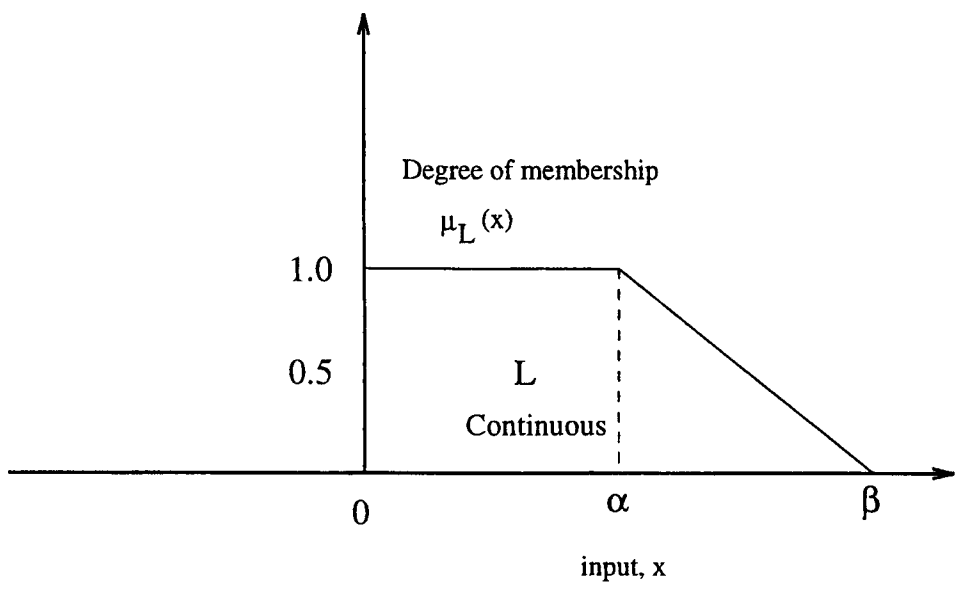
1.5.2 Fuzzy Set Representation

Fuzzy sets are usually represented as a continuous function of the independent variable. For example, the continuous fuzzy set, L, is shown in Figure 1.4 and is defined as a function with two parameters as follows:

$$L(x;\alpha,\beta) = \begin{cases} 1 & 0 \leq x < \alpha, \\ (\beta - x)/(\beta - \alpha) & \alpha \leq x \leq \beta, \\ 0 & x > \beta. \end{cases} \tag{1.3}$$

However, in the literature (Driankov et. al., 1996), it is not uncommon to use a discrete representation, particularly where computations with fuzzy sets are to be carried out on digital computers. The discrete representation, also shown in Figure 1.4, is used in this thesis. Here fuzzy sets are represented as a set of ordered pairs:

$$L(x;i) = \{(x_i, \mu_L(x_i))\} = \{(0,1), (\alpha/2, 1), (\alpha, 1), ((\beta-\alpha)/2, 0.5), (\beta, 0))\},\ x_i \in X \tag{1.4}$$



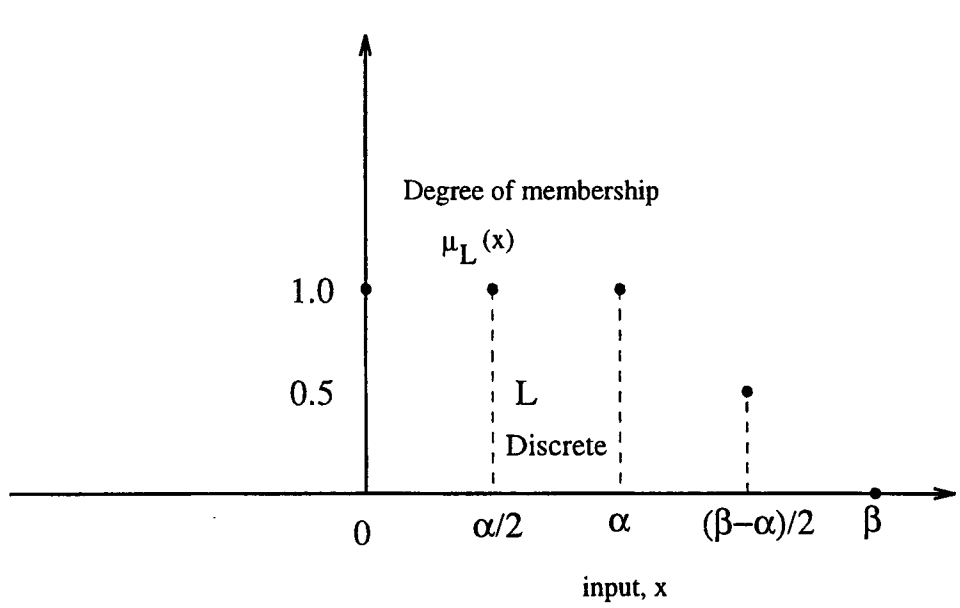


Figure 1.4: Continuous and Discrete Fuzzy Sets

In addition, the fuzzy sets used in fuzzy control theory are usually *convex*, as is the case in this work. A fuzzy set A is convex if and only if:

$$\forall x, y \in X\ \forall \lambda \in [0,1] : \mu_A\left(\lambda \cdot x + (1-\lambda) \cdot y\right) \geq min\left(\mu_A(x), \mu_A(y)\right). \tag{1.5}$$

Figure 1.5 shows a convex and a non-convex fuzzy set.

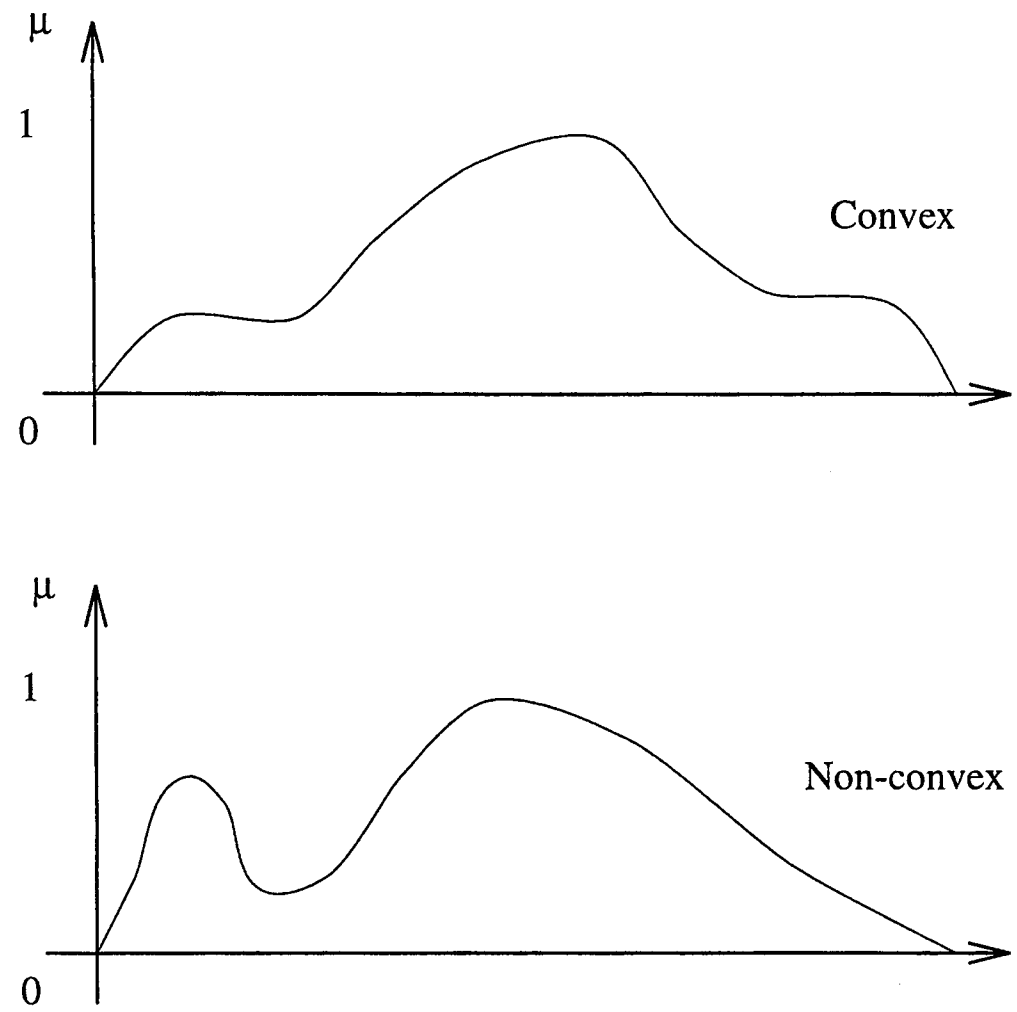


Figure 1.5: An Example of a Convex and a Non-Convex Fuzzy Set

1.6 Structure and Generation of the Neuro-Fuzzy Model

Neuro-fuzzy models are used in this work. These models differ from fuzzy models in that, whereas the rules that characterize a particular fuzzy model are obtained heuristically or from experts, the rule base of a neuro-fuzzy model consists of rules generated from training data that reflect the behaviour of the system being modelled. This training data is usually obtained either directly from the system, or from a simple model thereof, and the rules thus obtained are generally thought to be more consistent than those that might have been obtained from various experts.

In designing a neuro-fuzzy model for a dynamic system, three basic steps are usually followed (Batur et. al., 1995):

1. Structural identification (i.e. determination of the model structure)

2. Choice of reference membership functions

3. Construction of models (training the model and identifying the fuzzy rules)

1.6.1 Neuro-fuzzy Model Representation

Three popular methods of representing neuro-fuzzy models used in control systems are fuzzy rules, Takagi-Sugeno (T-S) fuzzy models and fuzzy relational equations (arrays).

The heart of any fuzzy system is the set of *if* ... *then* expert rules (usually called the fuzzy rule base of the system) which relate the input fuzzy sets to the output fuzzy sets. In neuro-fuzzy systems, these rules are generated from training data obtained from the system (or a simple model thereof) and, in essence, the rules form a model of the fuzzy system under consideration. Since the expert rules contain knowledge about the system, this system is sometimes referred to as a Fuzzy Knowledge-Based System. Approaches using fuzzy rules have been proposed in (Wang and Mendel, 1992).

T-S fuzzy models (Takagi and Sugeno, 1985) are another way of representing the fuzzy rules of a neuro-fuzzy system. For such a model, the rule consequents are not fuzzy sets but crisp functions of the model input. For example, a rule in a T-S fuzzy model might be:

$$\textbf{If } x \text{ is } A \textbf{ then } y \text{ is } f(x) \tag{1.6}$$

where x is the input variable, y is the consequent output variable, and A is a fuzzy set defined on the input universe of discourse. This representation of neuro-fuzzy models is common in the literature (Sousa, 1998).

A fuzzy relational array (model) is yet a third way of representing the fuzzy rule base of a fuzzy model. The entries of this array are a measure of the extent to which the fuzzy inputs *relate* to the fuzzy output (Postlethwaite, 1994), (Graham and Newell, 1989). Work involving fuzzy relational arrays may be found in (Pedrycz, 1984) and (Xu and Lu, 1987).

In this thesis, a fuzzy relational array is used as the core of the neuro-fuzzy predictive model. Work by Tan and Dexter (Tan and Dexter, 1996) shows that a single-input-single-output (SISO) fuzzy relational array with all the universes of discourse characterized by triangular fuzzy sets with 50% overlap is able to exactly model a piece-wise linear system if *sum-of-product* fuzzy inferencing and *height defuzzification* are used.

1.6.2 Structural Identification, Training and Fuzzy Identification of Neuro-fuzzy Models

A fundamental task in control theory in general, and fuzzy control in particular, is solving the *identification problem* - the problem of determining or *identifying* the characteristics of a black box from the input/output relationship of this black box, usually by experimental means.

In this work, the black box represents the particular process under control, and the identification problem encountered here is structural identification and estimation of the parameters of the fuzzy rules. For a given problem, a fixed number of inputs is used to characterize the dynamics of the black box (which is usually the case in structural identification).

After structural identification takes place, the neuro-fuzzy model must then be *trained* in order for it to replicate the behaviour of the process being modelled. Training is an attempt to capture the dynamics of the behaviour of the modelled process by exciting different frequencies and operating points across the controllable range of the process. This may be done in different ways, the most convenient of which usually involves interacting with a simulation of the process being controlled.

The data captured during training are then used in a *fuzzy identification scheme* (Sing and Postlethwaite, 1996) in order to generate the fuzzy relational array, R, which relates the inputs to the output of the model. The identification of this relational array is tantamount to the generation of the model, as will be shown in Chapters 4 and 5.

1.7 Outline of the Thesis

This thesis outlines some new developments in neuro-fuzzy predictive control, applicable to the control of uncertain systems. It also describes modifications to conventional defuzzification approaches, in an attempt to derive a more efficient defuzzification scheme.

Chapter 2 outlines a systematic approach to the development of the proposed control scheme. It deals with the groundwork and theoretical framework that forms the basis of this research. Various new concepts are introduced, and issues central to the application of the theory discussed.

Chapter 3 discusses two aspects of the thesis specific to the development and implementation of the proposed control scheme. In particular, it highlights certain problems associated with traditional defuzzification as it pertains to the control of information-poor systems. A proposed defuzzification scheme, called conditional defuzzification, is detailed, and the effect on the final control action of applying different forms of the conditional defuzzification scheme is shown. The chapter also investigates the development of the fuzzy proximity function.

Chapter 4 presents a first attempt at applying the theory proposed in Chapters 2 and 3 to the control of a system. Issues surrounding the development of the scheme as it pertains to the control of thermal comfort in a zone are discussed, and results are presented which compare the performance of the NFMBPC scheme to that of a PI controller.

In Chapter 5, the control scheme is applied to an actual air-handling unit in the control laboratory of Oxford University. Control of the supply air temperature of this rig using neuro-fuzzy generic models is attempted, and the work done in this chapter represents an application of the scheme to a real problem. The results from this chapter show that it is possible to apply this approach to systems other than theoretical or simulated ones, and demonstrates the robustness of the generic model approach, as compared to PI control, usually necessary in real-world uncertain systems.

In Chapter 6, the effect of choosing various application dependent parameters is examined. In particular, the sensitivity of the control scheme to the choice of a defuzzifi-

cation scheme, the choice of the actuator activity parameter, α, of Chapter 3 and the criteria tolerances discussed in Chapters 2 and 4, σ_c and σ_e, are discussed. Results are presented which demonstrate the effect of these choices on system resources such as energy consumption and the control activity of the controller.

Finally, Chapter 7 draws conclusion on the feasibility of the proposed scheme from the work done and results presented in this thesis. In addition, possible directions of future work are presented.

Supplementary material relevant to the thesis are contained in the 4 appendices.

Appendix A discusses fuzzy arithmetic; a concept used to compute with fuzzy numbers in this thesis. Appendix B looks at the RSK training and identification scheme used in Chapters 4 and 5. In addition, Section B.4 of Appendix B shows part of the actual fuzzy relational array, R, derived from the training data and used in the system studied in Chapter 4. Appendix C outlines various Matlab codes employed in the implementation of the proposed control scheme. Finally, in Appendix D, various levels of the Simulink simulation of thermal behaviour of the zone of Chapter 4 are given, for completeness.

Chapter 2

Neuro-Fuzzy Predictive Control with Fuzzy Criteria

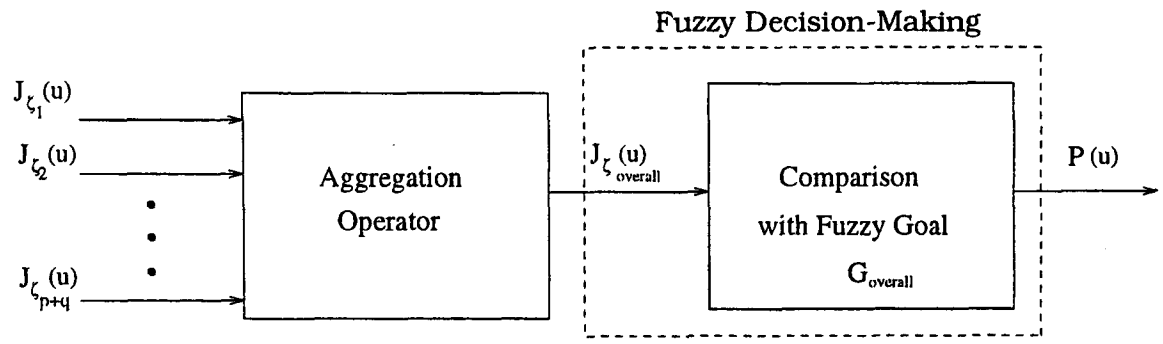
2.1 Introduction

Consider a control scenario where there are p control objectives and q constraints. Now suppose the control objectives and constraints are not clearly defined. For example, an objective of the controller in, say, a zone comfort control problem may be to keep the occupants of the zone "reasonably comfortable" or to keep the temperature in the zone at "about 21 °C", while a constraint might be that the energy consumption of the system be "less than design value". Since the objectives and constraints of this system are ill-defined, it may reasonably be classified as an information-poor system. The $p+q$ ill-defined objectives and constraints taken together may be considered fuzzy decision criteria (Sousa, 1998), since they define the conditions under which the control decision will be made, and are imprecise.

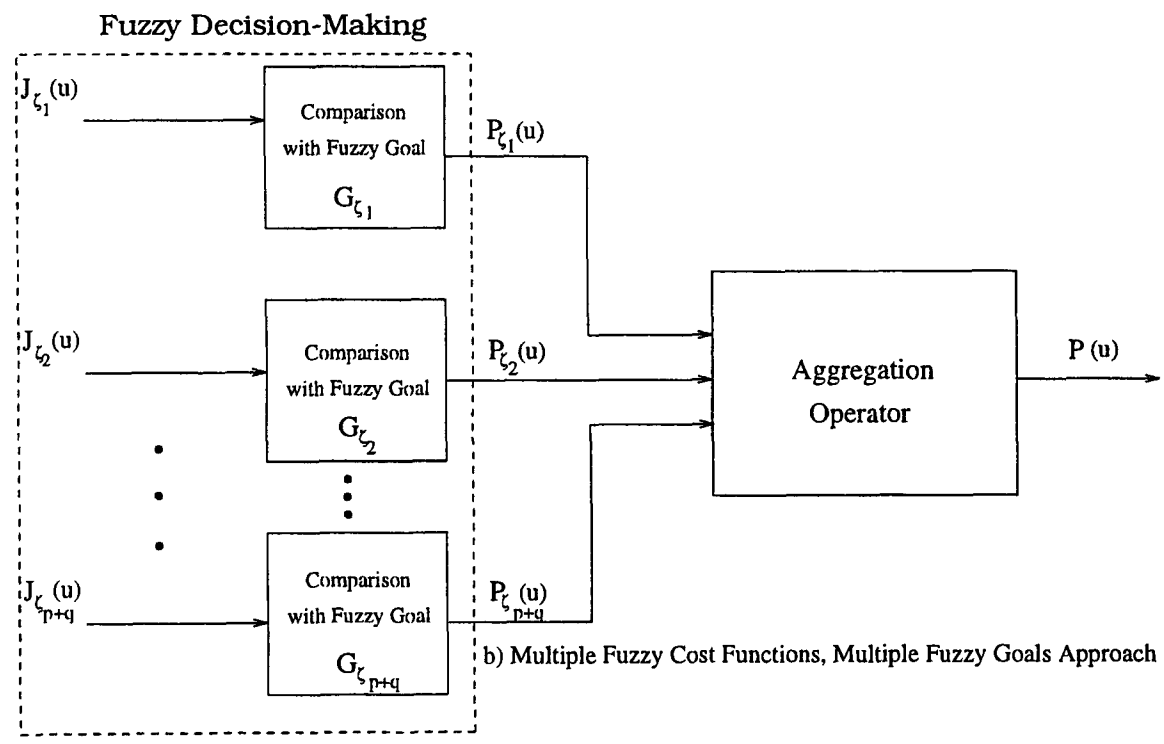
In this chapter, a fuzzy approach to solving the problem of controlling information-poor, non-linear dynamic systems of this type is proposed. A direct (and, in this case, exhaustive) search method of optimizing the control decision is used, and two different approaches of employing the $p+q$ fuzzy criteria in choosing the control action, presented.

In the first approach, fuzzy measures of the cost of control with respect to the $p+q$ fuzzy criteria are formed for each discrete member on the universe of discourse of the control action and are then weighted and combined by some aggregation operator to form one fuzzy representation of the overall cost of each particular control action. This overall measure of cost is then evaluated against some pre-defined *fuzzy goal*, in a

process known as *fuzzy decision-making* (Section 2.6), in order to determine the extent to which the goal is satisfied.



a) Single Fuzzy Cost Function, Single Fuzzy Goal Approach



b) Multiple Fuzzy Cost Functions, Multiple Fuzzy Goals Approach

Figure 2.1: The Two Basic Approaches to the Fuzzy Cost Function

In the second approach, again fuzzy measures of the cost of control with respect to the $p+q$ fuzzy criteria are formed for each discrete member on the universe of discourse of the control action. However, each criterion is now compared to its respective fuzzy goal defined on the space of that criterion (there are therefore now $p+q$ fuzzy goals) in order to evaluate the extent to which this particular control action satisfies this goal. The overall extent to which each control action satisfies *all* $p+q$ goals taken together is then evaluated using an aggregation operator.

Figure 2.1 shows both approaches.

2.2 Proposed Control Scheme

The proposed control scheme is one based on a combination of neuro-fuzzy modelling and model-based predictive control techniques. A block diagram of the scheme is shown in Figure 2.2. This diagram is the fuzzy equivalent of Figure 1.2 in Chapter 1, with the neuro-fuzzy controller here consisting of: i) a Neuro-Fuzzy Model (NFM) of the plant under control, ii) a Fuzzy Cost Function block, and iii) a Fuzzy Decision-Maker (the algorithm that finally chooses the optimal control signal, $u_{optimal}$).

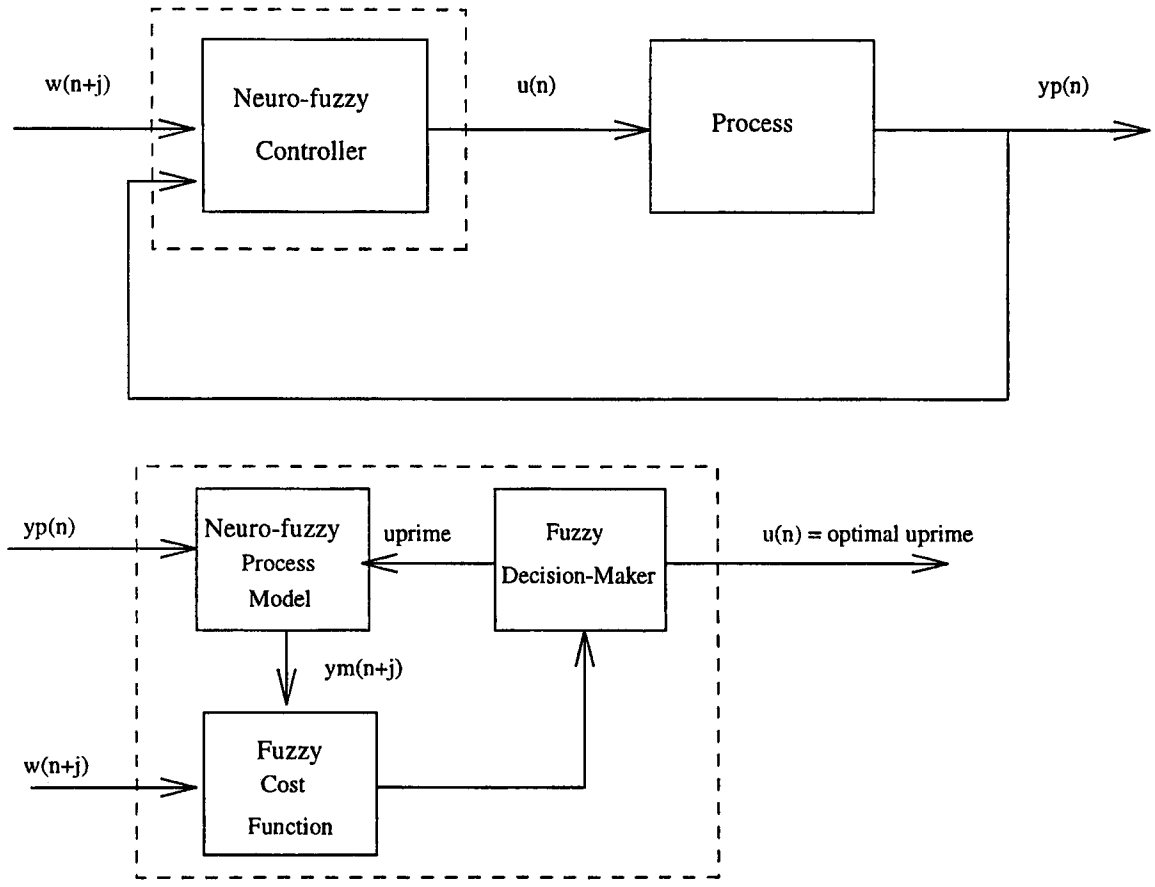


Figure 2.2: Neuro-Fuzzy Model-Based Predictive Control

Using candidate values of optimal control action from the Fuzzy Decision-Maker along with the current process output, the Neuro-Fuzzy Process Model block generates a fuzzy description of the j-step-ahead predicted process output values, and passes these values on to the Fuzzy Cost Function block (see Section 2.4). The cost functions generated in this block are computed by considering both the setpoint trajectory (which may or may not be fuzzy) and the fuzzy predicted values received from the Neuro-Fuzzy Process Model. These fuzzy cost functions are passed to the Fuzzy Decision-Maker to determine the degree to which they satisfy pre-defined fuzzy goals. The candidate

control action which best satisfies the fuzzy goal or combination of fuzzy goals is chosen as the optimal control action to be applied to the plant.

The entire decision-making process therefore takes place in the fuzzy domain as mentioned in Section 1.2. To this end, it has become necessary to examine more closely some functions normally used when computing with crisp numbers, since it is now necessary to establish whether these functions would remain valid for computations with fuzzy numbers.

One such function is similarity, of which arithmetic subtraction is a specific example. For example, in a conventional crisp objective function, subtraction is used to determine the extent to which two crisp quantities are similar; the larger the difference between the two quantities, the more dissimilar they are. In Section 3.3 of Chapter 3, it is shown that, for the purposes of this thesis, subtraction results in unacceptable results when applied to fuzzy numbers. A modification to the fuzzy subtraction function, fuzzy proximity, is therefore explored and developed in Chapter 3.

2.3 The Neuro-Fuzzy Process Model

With reference to Figure 2.2, the Neuro-Fuzzy Process Model block contains a neuro-fuzzy model of the process under control. This NFM uses a predetermined number of inputs to generate a *fuzzy* description of the process output. Unlike conventional NFMs, therefore, the output from this model is *not* defuzzified, but remains in the fuzzy domain to be passed on to the Fuzzy Cost Function block.

This novel approach to neuro-fuzzy modelling of not defuzzifying the model output is an attempt to retain the inherent fuzziness of the system under control. Since the output is not defuzzified, emphasis is not placed on generating a model which would result in a precise output when defuzzified.

In the literature, there are various methods which may be used to generate NFMs. The choice of NFM design is usually dependent upon the specific application of the model. As discussed in Section 1.6.1, the applications examined in this thesis use a fuzzy relational array as the NFM, the generation of which is outlined in Section 1.6.

2.4 The Fuzzy Cost Function: Two Basic Approaches

It is usually the case in model-based predictive control that the control cost to be optimized takes on a form similar to that of Equation 1.1 in Chapter 1:

$$J(\mathbf{u}) = \sum_{j=N_1}^{N_2} (\mathbf{w}(n+j) - \mathbf{y_p}(n+j))^2 + \lambda \sum_{j=1}^{N_u} (\mathbf{u}(n+j-1) - \mathbf{u}(n+j-2))^2 \qquad (2.1)$$

This sum-quadratic form allows gradient descent approaches to be conveniently applied for optimization purposes, since such an objective function has an analytical solution for unconstrained linear systems. However, nonconvex optimization problems can often occur in systems where non-linearities are present. While it is generally agreed that the cost function should include a term reflecting the control performance (usually taken as the square of the error between predicted plant output and setpoint), in cases where non-linearities are present, the sum-quadratic cost function does not have any advantage over other types of cost functions and there is then no substantial practical justification for choosing it over one which could possibly give a better description of the (imprecise) objective (Sousa, 1998). The choice of this sum-quadratic cost function is then, therefore, somewhat arbitrary, and there is no guarantee that the optimization process will produce truly optimum values of control action. In other words, while these chosen values of control action may optimize the cost function, there is uncertainty as to the extent to which the cost function represents the cost to be optimized. This argument provides a basis for exploring the use of a fuzzy cost function.

The Fuzzy Cost Function of Figure 2.2 generates a fuzzy description of the cost of applying a particular control action to the process under control. It can be viewed as the cost of obtaining a particular fuzzy objective given some fuzzy constraint, as a function of the control action. It is this fuzzy description of cost that the Fuzzy Decision Maker attempts to optimize in deciding on the final control signal to drive the plant.

There are two basic approaches utilized in computing the cost of a particular control action. The difference between the two approaches lies in how the costs of control with respect to individual criteria are aggregated to establish an overall cost representing

the scenario when all criteria are considered together.

2.4.1 Single Fuzzy Cost Function, Single Fuzzy Goal

In this approach to evaluating the fuzzy cost function, the individual costs of control with respect to different criteria are first combined to form one cost which is then evaluated against a pre-defined fuzzy goal (Figure 2.1(a)). So, for the $p+q$ fuzzy criteria $\zeta_1, \zeta_2, \ldots \zeta_{p+q}$, one cost function,

$$J_{\zeta_{overall}}(u) = J_{\zeta_1}(u) \oplus J_{\zeta_2}(u) \oplus \ldots \oplus J_{\zeta_{p+q}}(u) \tag{2.2}$$

is formed. Since $J_{\zeta_1}(u), J_{\zeta_2}(u) \ldots J_{\zeta_{p+q}}(u)$ are all fuzzy, the $\oplus$ symbol is the aggregation operator used to represent addition involving these fuzzy numbers. Arithmetic operations involving fuzzy numbers and intervals are explored in Appendix A.

Equation 2.2 ascribes equal importance to all fuzzy criteria, $\zeta_1, \zeta_2, \ldots \zeta_{p+q}$, in determining the cost of control. However, this need not necessarily be the case in a given scenario under consideration. For example, under certain conditions, it may be desirable that ζ_1 carry more weight in determining the cost of control, than does ζ_2. The relative importance of a particular criterion with respect to the remaining criteria is expressed by multiplying the fuzzy cost of control with respect to each criterion by a weight. Therefore, Equation 2.2 becomes

$$J_{\zeta_{overall}}(u) = \lambda_1 \cdot J_{\zeta_1}(u) \oplus \lambda_2 \cdot J_{\zeta_2}(u) \oplus \ldots \oplus \lambda_{p+q} \cdot J_{\zeta_{p+q}}(u) \tag{2.3}$$

and, for the case where just two fuzzy criteria are considered, Equation 2.3 reduces to

$$J_{\zeta_{overall}}(u) = \lambda_1 \cdot J_{\zeta_1}(u) \oplus \lambda_2 \cdot J_{\zeta_2}(u) \tag{2.4}$$

where the relative importance of $J_{\zeta_2}(u)$ to $J_{\zeta_1}(u)$is given by the ratio $\lambda = \lambda_2/\lambda_1$, and where $\lambda_1, \lambda_2 \in [0, 1]$. Further, λ_1 and λ_2 serve to bring both criteria to the same

arbitrary universe so that they may be added. For example, if ζ_1 is a comfort criterion and ζ_2 is an energy criterion, then λ_1 could have $\$/°C$ as units and λ_2 could have $\$/kWh$ as units. This brings both $\lambda_1 \cdot J_{\zeta_1}(u)$ and $\lambda_2 \cdot J_{\zeta_2}(u)$ to the same arbitrary universe of dollars, \$. The relative importance of ζ_1 and ζ_2 may be preserved by preserving the ratio $\lambda = \lambda_2/\lambda_1$. So Equation 2.3 changes to one containing one weight, λ,

$$J_{\zeta_{overall}}(u) = J_{\zeta_1}(u) \oplus \lambda \cdot J_{\zeta_2}(u) \tag{2.5}$$

This is a more convenient form of the fuzzy cost function since it gives the importance of all fuzzy criteria relative to the unit weight.

2.4.2 Multiple Fuzzy Cost Functions, Multiple Fuzzy Goals

The alternative approach to evaluating the fuzzy cost of control is one involving the use of multiple fuzzy cost functions evaluated against respective pre-defined fuzzy goals. Each cost function in this approach is first established with respect to each individual criterion. So, with this approach, $p + q$ fuzzy cost functions $(J_1(u) \ldots J_{p+q}(u))$ are formed:

$$\begin{aligned} J_1(u) &= J_{\zeta_1}(u) \\ J_2(u) &= J_{\zeta_2}(u) \\ &\vdots \\ J_{p+q}(u) &= J_{\zeta_{p+q}}(u) \end{aligned} \tag{2.6}$$

The cost functions are then evaluated against respective fuzzy goals to establish the extent to which each satisfies its goal. Therefore, if possibility is used as a measure of the extent to which two fuzzy numbers are similar, then the possibility, P, is given by:

$$P_{\zeta_1}(u) = Poss(J_1(u) \mid G_{\zeta_1})$$

$$\begin{aligned} P_{\zeta_2}(u) &= Poss(J_2(u) \mid G_{\zeta_2}) \\ &\vdots \\ P_{\zeta_{p+q}}(u) &= Poss(J_{p+q}(u) \mid G_{\zeta_{p+q}}) \end{aligned} \tag{2.7}$$

where G_{ζ_1} is the fuzzy goal defined with respect to criterion ζ_1, G_{ζ_2} is the fuzzy goal defined with respect to criterion ζ_2, and so on. The structure of fuzzy goals used in this thesis is presented in Section 2.5. From Equation 2.7, it is seen that a possibility vector (fuzzy number) is produced, P, as a function of u, for each criterion considered (Babuska, Sousa and Verbruggen, 1995). These $p+q$ possibility vectors can be combined, via an appropriate aggregation operator, to evaluate the overall extent to which each control action satisfies *all* fuzzy goals defined with respect to each criterion, taken together (Kaymak, Sousa and Verbruggen, 1996).

In this approach, the relative importance of the fuzzy criteria is established implicitly, via the construction of the fuzzy goals. Since the fuzzy goals are pre-defined based on the specific control scenario under consideration, it is possible to select the relative sizes of the *supports* of these goals defined on each criterion. The support of a fuzzy set A, denoted $supp(A)$, is a crisp set defined as:

$$supp(A) = \{x \in X \mid \mu_A(x) > 0\} \tag{2.8}$$

Since the fuzzy sets used in this work to describe fuzzy goals are convex (Section 1.5.2), the support of a fuzzy goal is an interval, and its size is called the *width* (See Figures 2.4 and 2.5) of the set (Driankov et. al., 1996). If the support of this convex fuzzy set is bounded, as is usually the case in fuzzy control, then $width(A)$ is given by:

$$width(A) = max\,(supp(A)) - min\,(supp(A)) \tag{2.9}$$

The relative widths of $G_{\zeta_1}, G_{\zeta_2}, \ldots G_{\zeta_{p+q}}$ are directly correlated with the relative importance of $\zeta_1, \zeta_2 \ldots \zeta_{p+q}$. Since a fuzzy goal with a smaller width has a smaller subset of values with a non-zero degree of membership, then satisfying it is more difficult than satisfying a goal with a larger width. If all goals $G_{\zeta_1}, G_{\zeta_2}, \ldots G_{\zeta_{p+q}}$ must be satisfied to some extent, as is the case using this approach, then more weight is therefore automat-

ically ascribed to goals with smaller widths. The corresponding criteria upon which these goals are defined are therefore more important than the criteria associated with goals possessing larger widths.

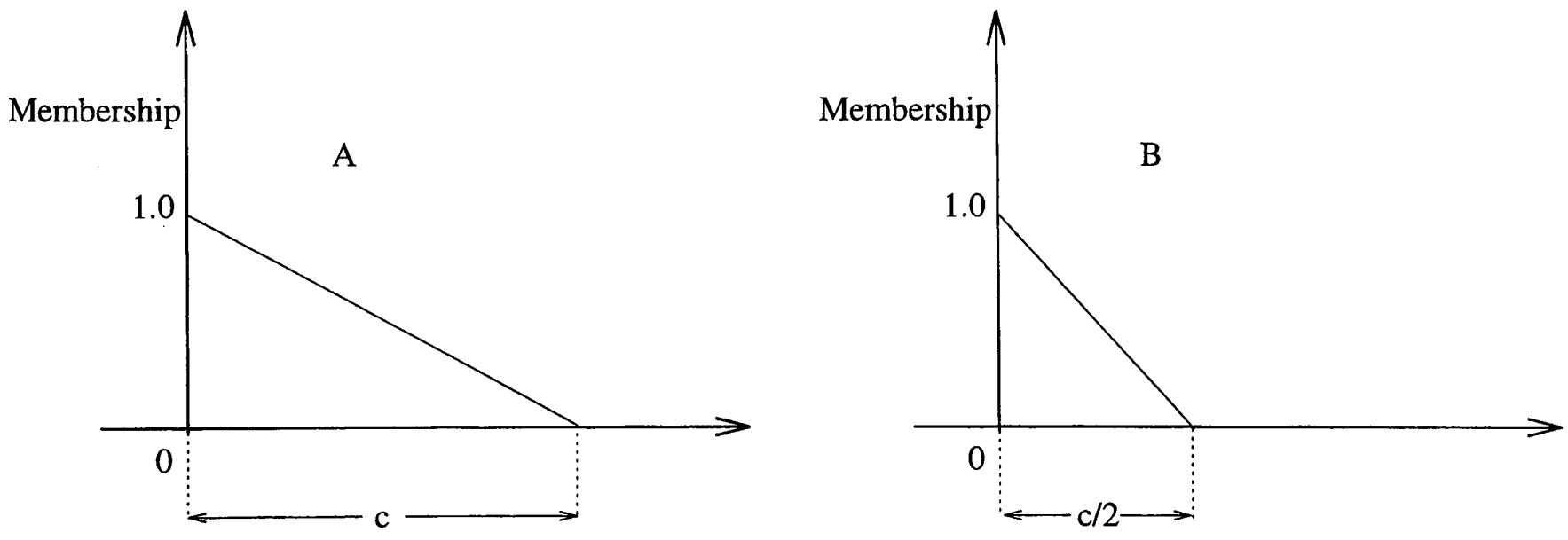
So, the importance of a fuzzy criterion is inversely proportional to the width of the fuzzy goal defined on that criterion. Or,

$$Importance(\zeta_1) \; \alpha \; \frac{1}{width(G_{\zeta_1})}, \tag{2.10}$$

and therefore,

$$\frac{Importance(\zeta_1)}{Importance(\zeta_2)} = \frac{width(G_{\zeta_2})}{width(G_{\zeta_1})} \tag{2.11}$$

Figure 2.3 illustrates this concept of relative goal importance.



Fuzzy goal B is twice as important as fuzzy goal A, since the width of B is half that of A

Figure 2.3: The Relative Importance of Two Fuzzy Goals

2.5 The Width of the Fuzzy Goal: Defining σ_{ζ_1}

Since, for the Multiple Fuzzy Cost Functions, Multiple Fuzzy Goals approach, the width of the fuzzy goal directly establishes the importance of the criterion upon which the goal is defined, there needs to be an efficient mechanism by which this width can be easily and accurately established. Remembering that, in practice, a fuzzy goal is characterized by a number of points on its universe in the discrete domain (Section 1.5.2), a *criterion tolerance*, σ_{ζ_1}, may be defined such that:

$$\sigma_{\zeta_1} = \frac{width(G_{\zeta_1})}{\delta} \tag{2.12}$$

where δ is the constant distance between any two successive points on the universe of discourse of the goal function defined on ζ_1 (Figure 2.4).

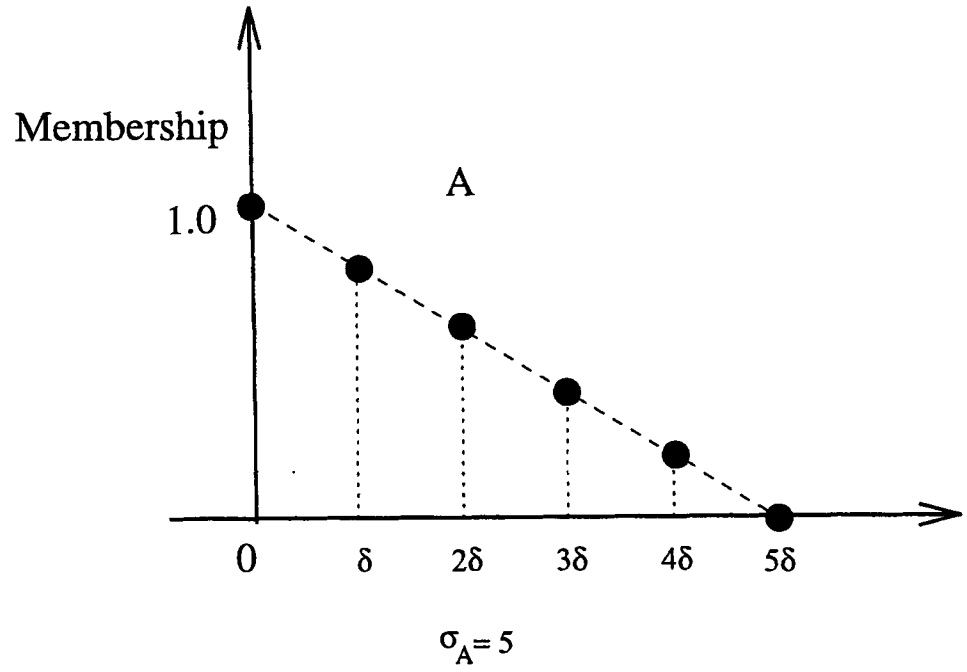


Figure 2.4: Defining the Width of a Triangular Fuzzy Goal

Given δ, $width(G_{\zeta_1})$ may therefore be defined by choosing σ_{ζ_1}:

$$width(G_{\zeta_1}) = \sigma_{\zeta_1} \cdot \delta \tag{2.13}$$

Since δ is fixed, the value of σ_{ζ_1} represents the number of discrete points along the

universe of discourse of the goal contributing to the width of the goal. A larger value of σ_{ζ_1} indicates a wider goal than does a smaller value of σ_{ζ_1}, and therefore implies that criterion ζ_1 is less important (dissatisfaction of the goal defined on ζ_1 is more tolerable) than it otherwise would have been, had σ_{ζ_1} been smaller. It is therefore seen that this parameter, σ_{ζ_1}, allows the control engineer to directly stipulate the importance of a particular criterion. Furthermore, for $p + q$ fuzzy criteria, choosing $\sigma_{\zeta_1}, \sigma_{\zeta_1}, \ldots \sigma_{\zeta_{p+q}}$ establishes the relative importance of the criteria.

It should be noted that, strictly speaking, this definition of σ_{ζ_1} holds only for triangular fuzzy sets. For a trapezoidal fuzzy set, W, define the *core* of the fuzzy set, $core(W)$, as a crisp set such that:

$$core(W) = \{x \in X \mid \mu_W(x) = 1\} \tag{2.14}$$

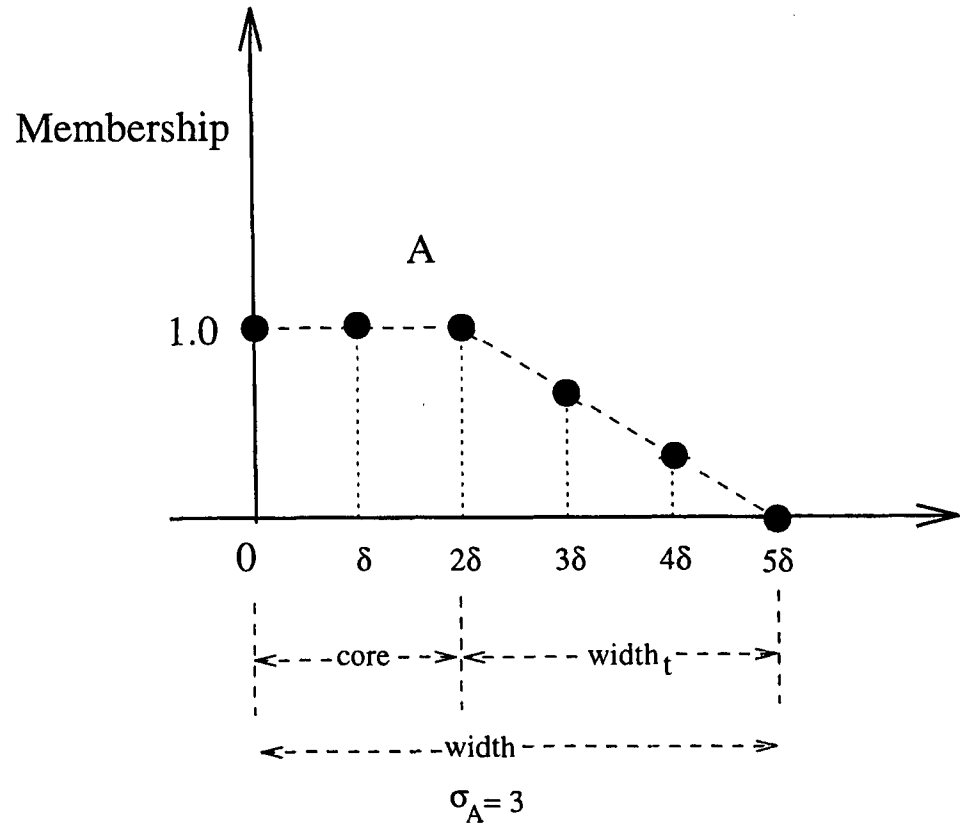


Figure 2.5: Defining the Width of a Trapezoidal Fuzzy Goal

This is illustrated in Figure 2.5. The width of W is therefore the sum of the core and a portion of the support which, for the purpose of convenience, will be designated the *triangular width*, $width_t$. It is the triangular width which is modulated by changes in

σ_{ζ_1}, so that, for trapezoidal fuzzy sets:

$$width(G_{\zeta_1}) = core(G_{\zeta_1}) + \sigma_{\zeta_1} \cdot \delta \tag{2.15}$$

2.5.1 Achievable Fuzzy Goals with Preservation of Relative Goal Importance

In this work, the fuzzy goals used are constructed to be what may be described as "achievable" because they have the capability of adapting their geometry so that, for the particular control problem, they remain realistic and practical. Therefore, if after exhaustively searching through the fuzzy sets defined on the control action, there exists no value of this action which, to some extent, satisfies *all* goals simultaneously, the goals are assumed to be unrealistic for the particular control problem, and are relaxed so that they become achievable. For the case where there are two fuzzy goals, this adjustment is automatically done by increasing the values of σ_{ζ_1} and σ_{ζ_2} so that the widths of the goals are increased, until the goals become achievable. To preserve the relative importance of the two goals, the ratio $\sigma_{\zeta_1}/\sigma_{\zeta_2}$ is kept constant during goal adjustment.

2.6 The Fuzzy Decision-Maker

The Fuzzy Decision Maker has one basic function: to choose, at each sample, the optimal control action that will drive the plant. In order to do this, it must first have the cost of using this particular control action evaluated with respect to the fuzzy objective and fuzzy constraint imposed upon it. This is done in the Fuzzy Cost Function block as explained in Section 2.4. Once this fuzzy cost of control is established, there needs to be a method of evaluating how close the cost has come to satisfying the goal of "about zero" cost (i.e. minimum cost). Clearly, conventional methods of optimization break down here since both the cost function and the goal are fuzzy. To this end, the use of a decision-making technique applicable to fuzzy costs and goals has been explored.

Consider the trapezoidal fuzzy goal function, G_{ζ_1}, centred at 0 as in Figure 2.6.

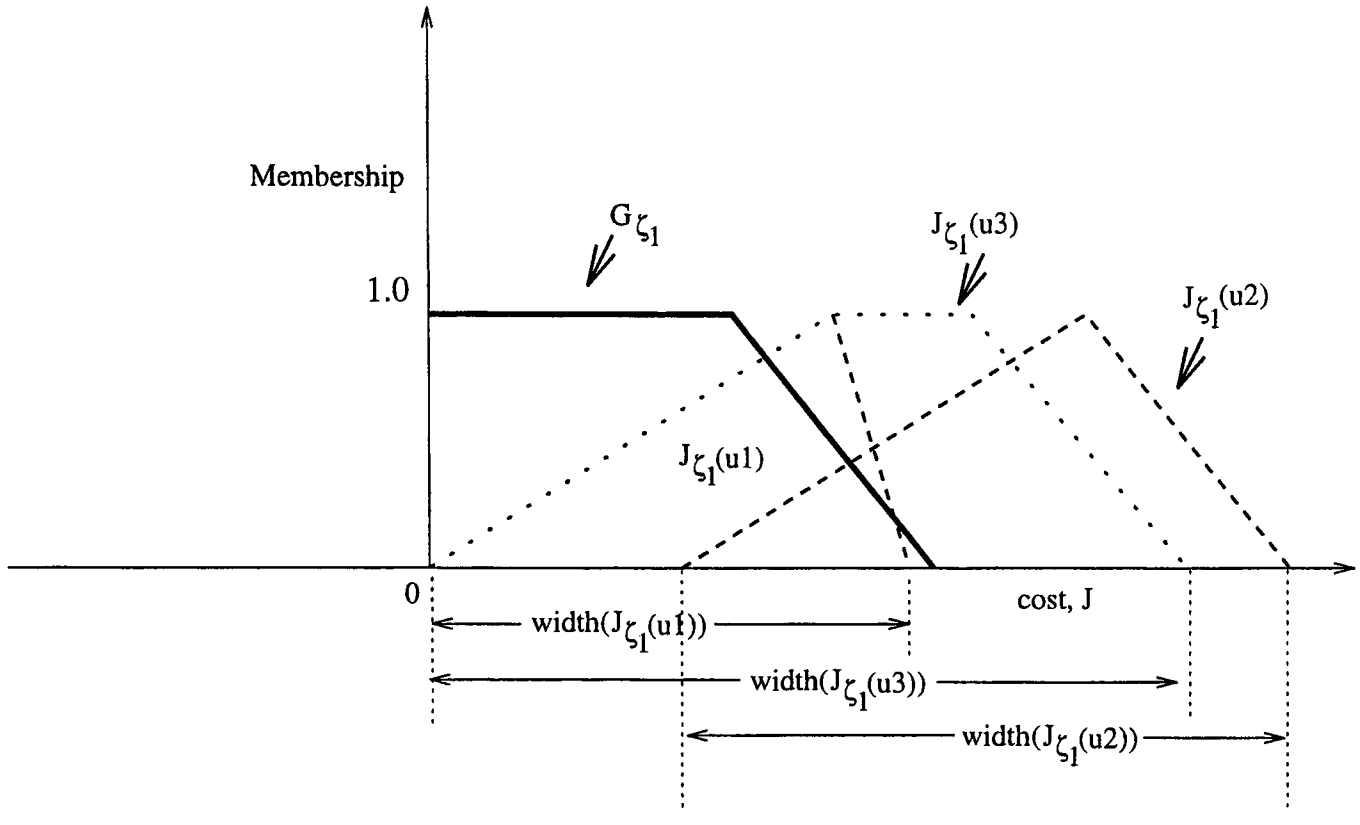


Figure 2.6: Fuzzy Matching used in the Fuzzy Decision-Making Scheme

To decide on the optimal control action to be applied to the plant, each $J_{\zeta_1}(u)$ generated by exhaustively stepping through each of the fuzzy sets defined on the control action may be compared or *matched* with the fuzzy goal function to see to what degree the two are similar. A fuzzy matching scheme must therefore be employed (Dexter and Benouarets, 1997).

The degree of similarity (fuzzy matching) between each $J_{\zeta_1}(u)$ and the G_{ζ_1} fuzzy set is taken as the ratio of *the common area between them* to *the total area of the* $J_{\zeta_1}(u)$ *fuzzy set*:

$$P_{\zeta_1}(u) = \frac{CA(J_{\zeta_1}(u) \mid G_{\zeta_1})}{TA(J_{\zeta_1}(u))} \tag{2.16}$$

where

$$CA(J_{\zeta_1}(u) \mid G_{\zeta_1}) = J_{\zeta_1}(u) \cap G_{\zeta_1} = \int_{-\infty}^{\infty} \min(J_{\zeta_1}(u), G_{\zeta_1})\, dJ \tag{2.17}$$

and

$$TA(J_{\zeta_1}(u)) = \int_{-\infty}^{\infty} (J_{\zeta_1}(u))\ dJ \tag{2.18}$$

However, since both the fuzzy cost function and the fuzzy goal function are discrete possibility distributions with a finite number of elements (Section 1.5.2), Equations 2.17 and 2.18 above become:

$$CA(J_{\zeta_1}(u) \mid G_{\zeta_1}) = J^*_{\zeta_1}(u) \cap G_{\zeta_1} = \frac{j_r - j_1}{r-1} \cdot \left(\frac{b_1 + b_r}{2} + \sum_{i=2}^{r-1} \min\left(J^*_{\zeta_1}(u,i), G_{\zeta_1}(i)\right) \right) \tag{2.19}$$

and

$$TA(J_{\zeta_1}(u)) = \frac{j_r - j_1}{r-1} \cdot \left(\frac{c_1 + c_r}{2} + \sum_{i=2}^{r-1} (J^*_{\zeta_1}(u,i)) \right) \tag{2.20}$$

by standard trapezium rule integration, where there are r discrete elements $(j_1, \ldots, j_r)$ along the J-axis of the fuzzy set $J^*_{\zeta_1}(u)$ (which is also equal to the number of discrete elements in G_{ζ_1}), and where:

$$\begin{aligned} b_1 &= \min(J^*_{\zeta_1}(u,1), G_{\zeta_1}(1)) \\ b_r &= \min(J^*_{\zeta_1}(u,r), G_{\zeta_1}(r)) \\ c_1 &= J^*_{\zeta_1}(u,1) \\ c_r &= J^*_{\zeta_1}(u,r) \end{aligned}$$

Since the fuzzy sets $J^*_{\zeta_1}(u)$ and $\min\left(J^*_{\zeta_1}(u,i), G_{\zeta_1}(i)\right)$ are represented such that the first and last elements of these sets are 0, then $b_1 = b_r = c_1 = c_r = 0$, and Equations 2.19 and 2.20 reduce to:

$$CA(J_{\zeta_1}(u) \mid G_{\zeta_1}) = J^*_{\zeta_1}(u) \cap G_{\zeta_1} = \frac{j_r - j_1}{r-1} \cdot \left(\sum_{i=2}^{r-1} \min\left(J^*_{\zeta_1}(u,i), G_{\zeta_1}(i)\right) \right) \tag{2.21}$$

and

$$TA(J_{\zeta_1}(u)) = \frac{j_r - j_1}{r-1} \cdot \left(\sum_{i=2}^{r-1} (J^*_{\zeta_1}(u,i)) \right) \tag{2.22}$$

resulting in Equation 2.16 reducing to:

$$P_{\zeta_1}(u) = \frac{CA(J_{\zeta_1}(u) \mid G_{\zeta_1})}{TA(J_{\zeta_1}(u))} = \frac{\sum_{i=2}^{r-1} \min\left(J^*_{\zeta_1}(u,i), G_{\zeta_1}(i)\right)}{\sum_{i=2}^{r-1} (J^*_{\zeta_1}(u,i))} \tag{2.23}$$

This method is preferred to taking the possibility of one fuzzy set given the other (Lee et. al., 1992), as given in Equation 2.7, since the width of each $J_{\zeta_1}(u)$ fuzzy set is now taken into consideration. If the possibility scheme were used, fuzzy cost functions having equal points of intersection with the fuzzy goal function, G_{ζ_1}, but having different widths, would be deemed equally similar to this goal function. This is clearly not the case. The ratio of areas scheme attempts to address this problem.

For example, in Figure 2.6, $J_{\zeta_1}(u_1)$ clearly satisfies the goal the best. However, possibility measures would give equal goal satisfaction to $J_{\zeta_1}(u_1)$ and $J_{\zeta_1}(u_3)$, since the possibilities of these functions given the fuzzy goal are equal:

$$Poss(J_{\zeta_1}(u_1) \mid G_{\zeta_1}) \;=\; Poss(J_{\zeta_1}(u_3) \mid G_{\zeta_1}) \tag{2.24}$$

With the ratio of areas scheme, the ratio of the common area between $J_{\zeta_1}(u_1)$ and G_{ζ_1}, $CA(J_{\zeta_1}(u_1) \mid G_{\zeta_1})$, to the total area of $J_{\zeta_1}(u_1)$, $TA(J_{\zeta_1}(u_1))$, is greater than the ratio of $CA(J_{\zeta_1}(u_3) \mid G_{\zeta_1})$ to $TA(J_{\zeta_1}(u_3))$:

$$\frac{CA(J_{\zeta_1}(u_1) \mid G_{\zeta_1})}{TA(J_{\zeta_1}(u_1))} > \frac{CA(J_{\zeta_1}(u_3) \mid G_{\zeta_1})}{TA(J_{\zeta_1}(u_3))} \tag{2.25}$$

$J_{\zeta_1}(u_1)$ is therefore chosen as the function that best satisfies G_{ζ_1}. Note that either scheme would reject $J_{\zeta_1}(u_2)$ as the optimal solution.

Once the optimal control action is chosen by the Fuzzy Decision-Maker, in order to drive the actuator, it must be defuzzified. In the implementation of the control scheme, suboptimal values of the control action are not actually discarded, but are retained (with their corresponding membership grades) for use in an appropriate defuzzification scheme of choice. The resulting candidate control values, along with their respective membership grades equal to the value of the ratio of areas calculated, therefore form a fuzzy set of optimal control actions to be applied to the plant. A number of appropriate defuzzification schemes are explored in this work, including the introduction and development of *Conditional Defuzzification* outlined in Chapter 3.

Although both the Single Fuzzy Cost Function, Single Fuzzy Goal approach of Section 2.4.1, and the Multiple Fuzzy Cost Functions, Multiple Fuzzy Goals approach of Section 2.4.2 have been developed, the application of the theory uses the latter approach as the basis upon which to formulate the control scheme. This is due to the high computational demand that would be associated with the practical implementation of the scheme under the Single Fuzzy Cost Function, Single Fuzzy Goal approach.

2.7 Concluding Remarks

An approach to formulating and solving the control problem has been presented. More specifically, for systems which are information-poor, and which contain some inherent uncertainty either in how different system variables interact, or in describing the objective of the controller, the approach presented has applied neuro-fuzzy predictive control techniques *without* defuzzification of the model output. Fuzzy criteria are described for each control problem in an attempt to retain the uncertainties associated with the control problem.

Two forms of the control scheme have been outlined. In the first approach, the individual costs of control with respect to different criteria are first combined to form one cost which is then evaluated against a pre-defined fuzzy goal. In the second approach, each cost function is first established with respect to each individual criterion. The cost functions are then evaluated against respective fuzzy goals to establish the extent to which each satisfies its goal, after which aggregation takes place.

The structure of the fuzzy goals used in this work (including the importance of the widths of the goals), as well as the relationship between the criteria tolerances and the establishment of the relative goal importance in the Multiple Fuzzy Cost Functions, Multiple Fuzzy Goals approach have also been outlined. It has been established that the goal width is inversely proportional to the importance of the criterion upon which the goal is defined. Also, a discussion of the fact that the fuzzy goals are designed to be "achievable" has ensued. This feature allows optimistically unrealistic criteria tolerances to be dynamically adjusted to more realistic (achievable) values during optimization.

Finally, it was shown that the *ratio of areas* scheme for fuzzy matching would produce superior results over the possibility function, when used in matching the fuzzy cost function to the fuzzy definition of the goal function in this work.

The following chapter discusses further developments explored in this work.

Chapter 3

Conditional Defuzzification and Fuzzy Proximity

3.1 Introduction

This chapter highlights two ideas introduced in the development of the proposed neuro-fuzzy predictive control scheme: conditional defuzzification and the fuzzy proximity function.

Two broad categories of defuzzification schemes are explored with respect to their effect on the final control action applied to the process under control. Firstly, the Traditional Height Defuzzification (THD) Scheme is presented. Next, a modification to the THD scheme, called Conditional Defuzzification (CD), is investigated. Conditional Defuzzification is further subdivided into (i) Conditional Maximum Possibility Defuzzification (CMPD), (ii) Conditional Nearest Neighbour Defuzzification (CNND) (iii) Conditional Height Defuzzification (CHD) and (iv) Conditional Linear-Interpolation Defuzzification (CLID). All four subdivisions are examined.

In addition, in Section 3.3, the fuzzy proximity function is developed, along with a presentation of the algorithm that performs the function.

3.2 Examining Different Defuzzification Schemes

Defuzzification is the process whereby a variable in the fuzzy domain is transformed into an appropriate (equivalent) variable in the crisp domain. This is necessary since

actuators are designed to interact with crisp and not fuzzy signals.

3.2.1 The Traditional Height Defuzzification Scheme

The traditional height defuzzification scheme is a common method of defuzzification found in the literature(Driankov et. al., 1996). It is a possibility-weighted average of values at the apex of the fuzzy sets defined on the output universe of discourse.

Consider a set of r output fuzzy reference sets B_1, B_2,$\cdots$, B_{r-1}, B_r, defined on the space of an output variable, y . If the possibility that the output is represented by the j^{th} output set, B_j $(j \leq r)$, is μ_{B_j}, then height defuzzification produces as its crisp output:

$$\bar{y} = \frac{\sum_{j=1}^{r} \mu_{B_j} \cdot \Psi_j}{\sum_{j=1}^{r} \mu_{B_j}} \tag{3.1}$$

where Ψ_j is the value at the apex of the j^{th} output set on the universe of discourse.

3.2.2 The Conditional Defuzzification Scheme

Though defuzzification in the traditional sense of Section 3.2.1 usually provides acceptable control values, in a few cases where there are equal high possibilities of conflicting values of the variable under consideration, defuzzification can produce a meaningless result.

For example, consider a system where the output is characterized by two fuzzy sets defined on its universe of discourse, L and R, as shown in Figure 3.1. The possibility values of the output suggest that there is a 50% belief that this output is in fuzzy set L and a 50% belief that this output is in fuzzy set R. Traditional height defuzzification here produces an output which falls midway between the L and R sets, given from Equation 3.1 by:

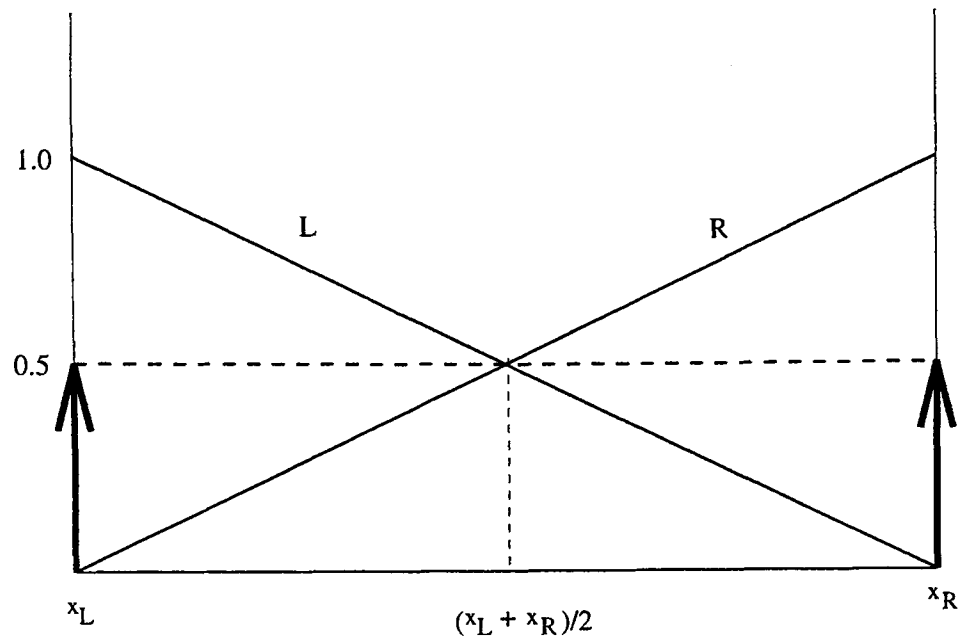


Figure 3.1: Some Problems with Height Defuzzification

$$\begin{aligned} \bar{x} &= \frac{\mu_L(x_L) \cdot x_L + \mu_R(x_R) \cdot x_R}{\mu_L(x_L) + \mu_R(x_R)} \\ &= \frac{0.5x_L + 0.5x_R}{0.5 + 0.5} \\ \Rightarrow \bar{x} &= \frac{x_L + x_R}{2}, \end{aligned} \tag{3.2}$$

If this system is such that the arithmetic mean of the values at the apexes of the two fuzzy output sets produces an output which is worse (i.e. more meaningless) than either output, then traditional height defuzzification fails here. In other words, if our fuzzy inferencing scheme suggests a "dumb-bell" possibility distribution (where there are large degrees of belief on either side of the universe of discourse), then height defuzzification may not necessarily be an optimal defuzzification strategy.

In addition, in systems which have a high degree of uncertainty, traditional height defuzzification on a fuzzy actuator output signal in effect takes an overall average of all actuator signals on the universe of discourse, and combines them to form one crisp signal. However, since the system is so uncertain, this crisp signal need not be the most optimal action to take and, more importantly, may even cause the actuator to chatter, since these defuzzified values of the actuator signal may change slightly with each sample. Also, defuzzification in this traditional sense effectively discards any information which might be contained within the inherent fuzziness of the control

action prior to defuzzification. Moreover, defuzzifying the output at each sample step introduces a considerable computational demand on the control algorithm which may not be necessary if the defuzzified value does not differ significantly from the previously applied control signal.

In an attempt to address some of the issues surrounding traditional height defuzzification, an alternative defuzzification scheme called *conditional defuzzification* has been developed.

Conditional defuzzification addresses two issues central to the concept of defuzzification as it pertains to actuator signals of digital control systems:

- *i) at the end of any given sample period, is the new value of actuator signal produced sufficiently different from the previously applied signal to justify a movement in the actuator, and*
- *ii) if the answer to part i) is 'yes', what value should the new actuator signal assume?*

The defuzzification scheme is therefore so named because it first tests the condition in *i)* above, and defuzzifies only if this condition is met. If the condition in *i)* is not met, the previous value of the applied actuator signal is used as the current value, and no movement in the actuator occurs.

Depending on the particular control application, this defuzzification scheme may therefore have a greater potential of reducing actuator wear and control activity compared to traditional height defuzzification. In so doing, its effect may therefore be compared with the effect of the '$\lambda \cdot \Delta u$' term usually found in conventional GPC cost functions.

3.2.3 The Actuator Activity Parameter, α

A defining feature of the conditional defuzzification schemes outlined in this work is the actuator activity parameter, α. The actuator activity parameter is a tuning parameter which regulates the tolerance for movement of the control actuator. This is

achieved indirectly by incorporation into the chosen conditional defuzzification scheme algorithm, as explained below.

This parameter, which is a real number defined such that $0.0 \leq \alpha \leq 1.0$, has the effect of increasing the activity of the actuator for values closer to 1.0, while decreasing the actuator activity for values closer to 0.0. It may be thought of as the minimum acceptable possibility of the optimal control action, given the previously applied control: a value of $\alpha = 0.0$ means the condition is never satisfied and therefore no defuzzification of the fuzzy output takes place at that sample step, while a value of $\alpha = 1.0$ means that the condition is always satisfied, and defuzzification takes place at the sample step in question. For values of α between 0.0 and 1.0, at any given sample step defuzzification takes place only on some condition evaluated as a function of α. This addresses the 'should the actuator move?' question. If defuzzification does indeed take place at a particular sample step, it is either maximum-possibility-, nearest-neighbour-, height- or linear-interpolation-defuzzification. This addresses the 'where should the actuator go?' question.

The effect of different values of α on the performance of the controller in systems incorporating conditional defuzzification is illustrated in Section 6.3.

3.2.4 The Conditional Maximum Possibility Defuzzification Scheme

Consider the scenario where, at the end of the n^{th} sample period, the membership function of the optimal control action for a particular system is as shown in Figure 3.2.

Here, $u_j(n)$ is the j^{th} apex of the fuzzy reference sets defined on the universe of discourse of the output. There are six such sets in the scenario under consideration. The fuzzy set, B, represents the set of optimal control signals to be applied to the plant at the current sampling instant, n. In addition, $\mu_B(u_j(n))$ is the membership function of $u_j(n)$ in B.

Under the CMPD scheme, the condition tested is as follows:

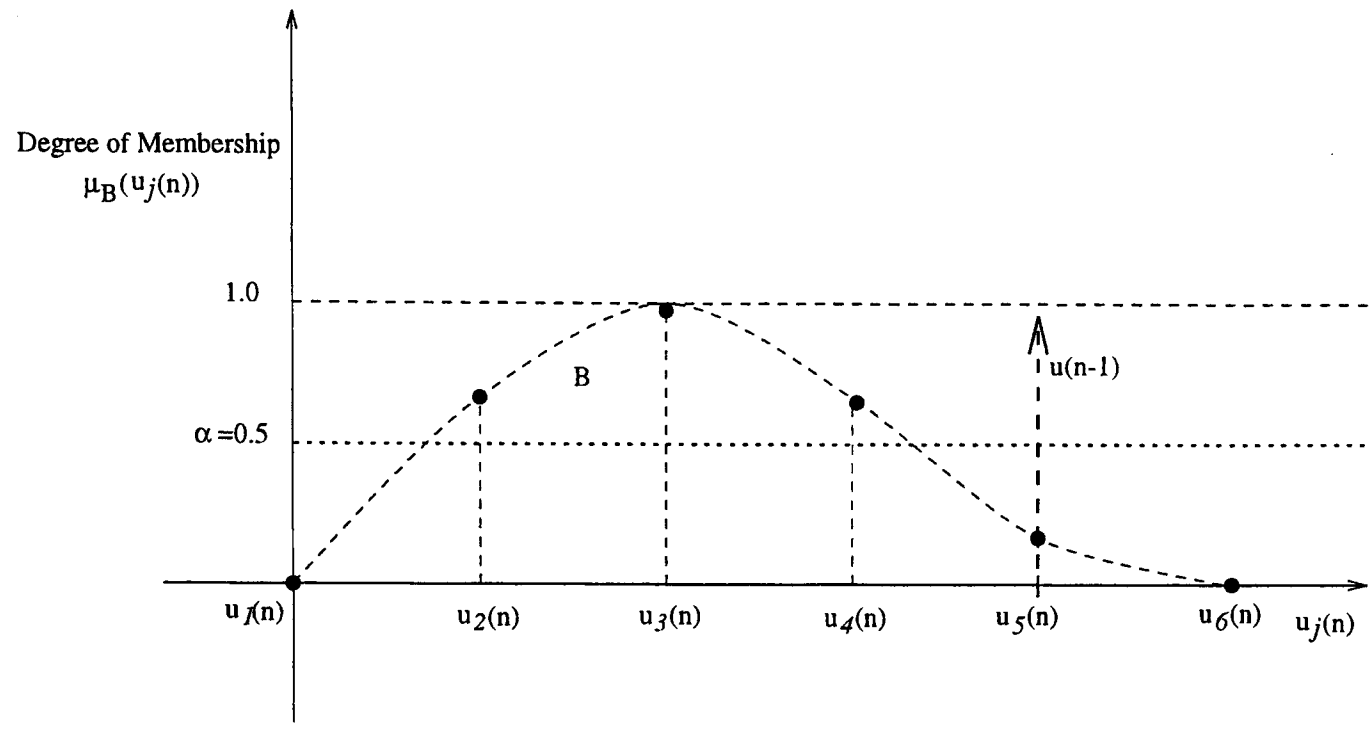


Figure 3.2: Illustrating Different Defuzzification Schemes

$$\mu_B(u(n-1)) < \alpha \cdot height(B) \tag{3.3}$$

where $height(B)$ is the height of the fuzzy set B defined as the maximum grade of membership in the set.

In other words, a test is done to see if the degree of membership of the previously applied control action in the membership function of the optimal control action is less than α times the maximum membership grade in this set of optimal actions. If this condition fails, then the previously applied control action is deemed to be close enough to any new control action suggested by the set of optimal control actions, and *no defuzzification takes place.* However, if this condition is satisfied, then defuzzification takes place by simply setting the new control action to the value of the control action with the *maximum* degree of membership in the optimal set of actions.

So, in Figure 3.2 where $u(n-1) = u_5(n)$, $\alpha = 0.5$ and $height(B) = 1$, the algorithm would be,

If:

$$\mu_B(u_5(n)) < 0.5 \cdot 1 \tag{3.4}$$

Then:

$$u(n) = u_3(n) \tag{3.5}$$

Else:

$$u(n) = u(n-1) = u_5(n) \tag{3.6}$$

The optimal control action applied to the plant at the n^{th} sample is therefore given by:

$$u(n) = \begin{cases} Def_{max}(B) & \text{if} \quad \mu_B(u(n-1)) < \alpha \cdot height(B), \\ u(n-1) & \text{otherwise,} \end{cases} \tag{3.7}$$

where $Def_{max}(B)$ is the maximum possibility defuzzification scheme defined such that the defuzzified output is simply that value on the universe of discourse which has the maximum membership in the set of possibilities.

3.2.5 The Conditional Nearest Neighbour Defuzzification Scheme

Again, consider the scenario depicted in Figure 3.2. Under the CNND scheme, the condition tested is the same as that of Equation 3.3. If the condition is false no defuzzification takes place, as usual, because the previously applied control action is deemed to be close enough to any new control action suggested by the set of optimal control actions. However, if the condition is true, the defuzzification which takes place is such that the new optimal control action applied to the actuator is the *nearest neighbour* to the last applied control action, *that fails to satisfy the condition of Equation 3.3.*

Continuing to use Figure 3.2 as an example, where $u(n-1) = u_5(n)$, $\alpha = 0.5$ and

$height(B) = 1$, the control action applied to the actuator would be $u_4(n)$ if $\mu_B(u_5(n)) < 0.5$, but would remain at $u_5(n)$ otherwise.

In general, therefore, under the CNND scheme, the optimal control action applied to the plant at the n^{th} sample is given by:

$$u(n) = \begin{cases} Def_{NN}(B) & \text{if} \quad \mu_B(u(n-1)) < \alpha \cdot height(B), \\ u(n-1) & \text{otherwise,} \end{cases} \tag{3.8}$$

where $Def_{NN}(B)$ is the nearest neighbour defuzzification scheme defined such that the defuzzified output is the nearest neighbour of the previously applied control action on the universe of discourse of the output which has a possibility value greater than or equal to α times the maximum possibility value in the set of optimal control actions.

Note that for both the CMPD and CNND defuzzification schemes, the control signal applied to the actuator at any given sample must necessarily be on one of the apexes of the fuzzy reference sets defined on the universe of discourse of the output (i.e., $u_1(n)$ or $\cdots$ or $u_j(n)$, if there are j fuzzy reference sets defined on the output).

3.2.6 The Conditional Height Defuzzification Scheme

In the CHD scheme, the condition tested is:

$$\mu_B(u'(n)) < \alpha \cdot height(B) \tag{3.9}$$

where $u'(n)$ is the closest apex value to the previously applied control signal, $u(n-1)$.

In Figure 3.3, $u'(n) = u_5(n)$.

If the condition of Equation 3.9 is false, then no defuzzification takes place. However, if the condition is true, traditional height defuzzification takes place. Note that in this scheme, since the previously applied control action will not necessarily have been on one of the apexes of the fuzzy reference sets defined on the output universe of discourse

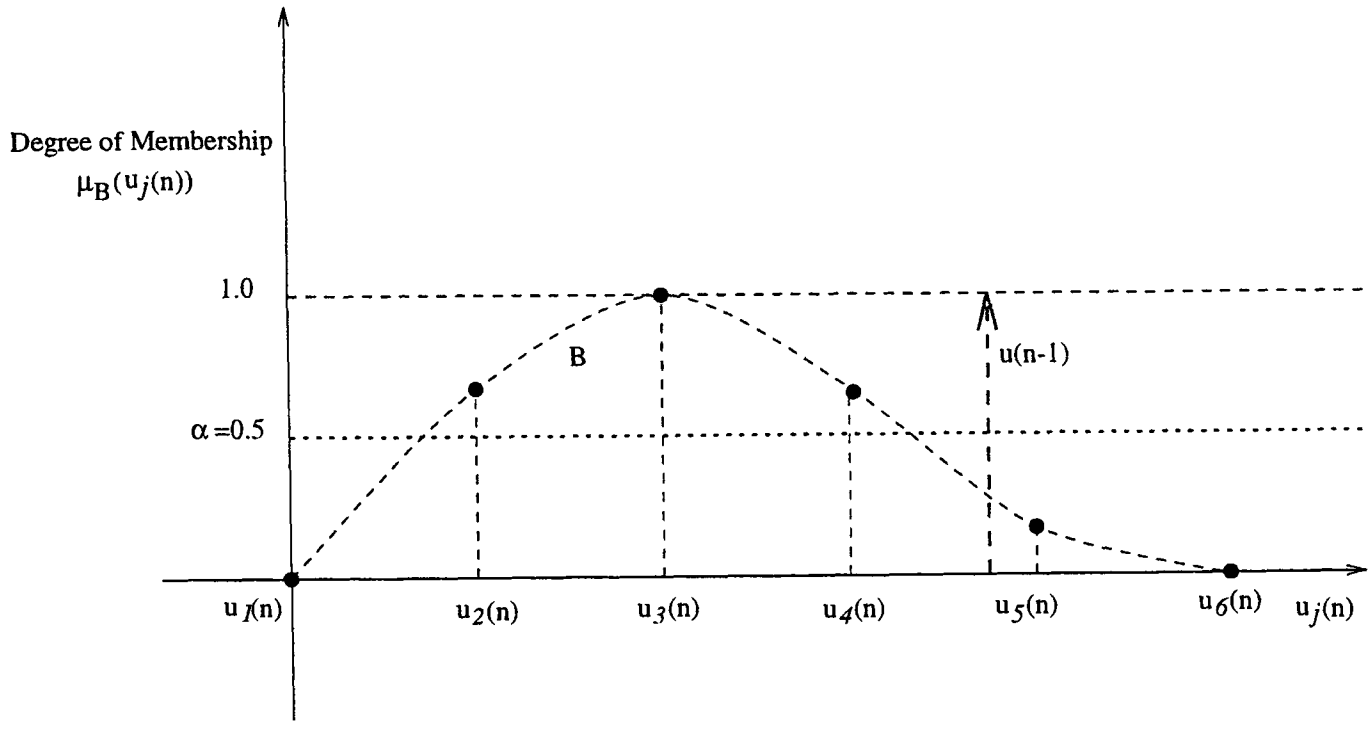


Figure 3.3: Illustrating the CHD Scheme

(due to the fact that traditional height defuzzification was used to compute it), then the test on the condition of Equation 3.9 above is done on *the closest apex value to the previously applied control signal*, denoted $u'(n)$.

Consider Figure 3.3 as an example, where $u'(n) = u_5(n)$, $\alpha = 0.5$ and $height(B) = 1$. The algorithm under CHD would be,

If:

$$\mu_B(u_5(n)) < 0.5 \cdot 1 \tag{3.10}$$

Then:

$$u(n) = \bar{u}(n) \tag{3.11}$$

Else:

$$u(n) = u(n-1) \tag{3.12}$$

where

$$\bar{u}(n) = \frac{\sum_{j=1}^{6} \mu_B(u_j(n)) \cdot u_j(n)}{\sum_{j=1}^{6} \mu_B(u_j(n))} \tag{3.13}$$

The optimal control action applied to the plant at the n^{th} sample is therefore given by:

$$u(n) = \begin{cases} \bar{u}(n) & \text{if} \quad \mu_B(u') < \alpha \cdot height(B), \\ u(n-1) & \text{otherwise,} \end{cases} \tag{3.14}$$

3.2.7 The Conditional Linear-Interpolation Defuzzification Scheme

The CLID defuzzification scheme is illustrated in Figure 3.4.

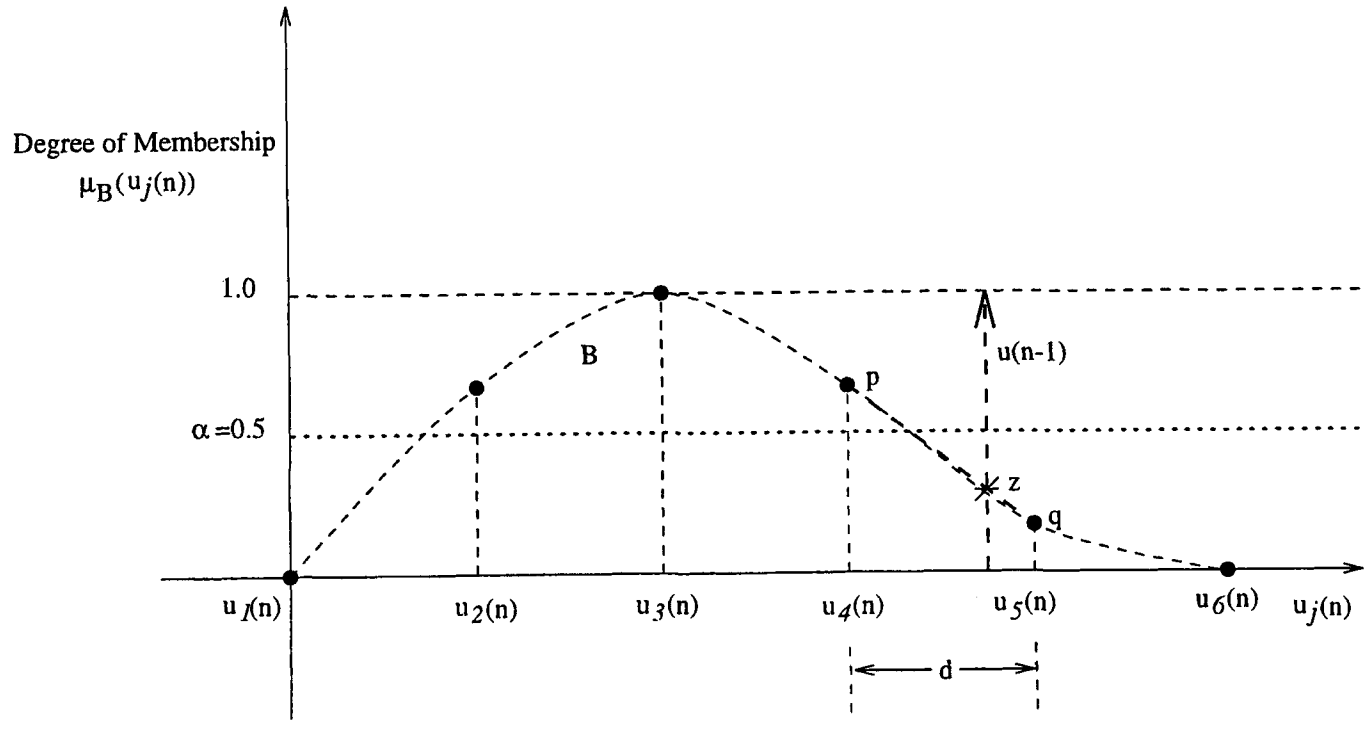


Figure 3.4: Illustrating the CLID Scheme

To decide whether or not to defuzzify and move the actuator, the scheme tests the

following condition:

$$\mu_B(u(n-1)) < \alpha \cdot height(B) \tag{3.15}$$

If the condition is false, no defuzzification takes place. However, if the condition is true, traditional height defuzzification takes place.

The values of $\mu_B(u_j(n))$ are available only at discrete points along the universe of discourse corresponding to the apexes of the fuzzy reference sets defined on the output (j=1,2,$\cdots$,6 in the case under consideration). Since $u(n-1)$ may fall between the apexes, an estimate of $\mu_B(u(n-1))$ is used to test the condition. This estimate, z, is obtained by *linearly interpolating* between values of $\mu_B(u_j(n))$ at the apexes of the fuzzy reference sets.

In Figure 3.4, $p = \mu_B(u_4(n))$ and $q = \mu_B(u_5(n))$. By linear interpolation, z is given by:

$$z = p + \frac{(u(n-1) - u_4(n)) \cdot (q-p)}{d} \tag{3.16}$$

where $d = u_5(n) - u_4(n)$.

The condition tested in Equation 3.15 may therefore be re-written:

$$z < \alpha \cdot height(B) \tag{3.17}$$

The optimal control action applied to the plant at the n^{th} sample is given by:

$$u(n) = \begin{cases} \bar{u}(n) & \text{if} \quad z < \alpha \cdot height(B), \\ u(n-1) & \text{otherwise,} \end{cases} \tag{3.18}$$

where $\bar{u}(n)$ is defined in Equation 3.13.

The Matlab codes for the implementation of each defuzzification scheme are provided in Section C.1.3 of Appendix C.

3.3 The Fuzzy Proximity Function

The second development explored in this chapter is that of the fuzzy proximity function. In Section 2.2, it was argued that, because the entire decision-making process of the control algorithm takes place in the fuzzy domain, it has become necessary to develop functions appropriate for computing with fuzzy numbers. To this end, the *fuzzy proximity* function has been developed.

The fuzzy proximity function is a measure of the similarity between two fuzzy numbers and the concept was developed to cope with some of the anomalies associated with the subtraction of fuzzy numbers. Ordinarily, to calculate the error between the setpoint and the model prediction that appears in a classical cost function, subtraction is used. Therefore, if both are equal, the error reduces to zero and indicates that the model prediction matches the setpoint exactly. However, in the fuzzy domain, subtraction needs to be more carefully evaluated, since the uncertainties of each fuzzy number must now be taken into consideration. The use of the *extension principle* (Klir and Yuan, 1995) and/or α-cut techniques is the preferred method of performing fuzzy addition, subtraction, multiplication and division, or *fuzzy arithmetic* as it has become known (Jager, 1995).

With either of these techniques, it can be shown that the difference between two fuzzy numbers is always fuzzier than either number (Klir and Yuan, 1995). This is of particular significance when one fuzzy number is a proper subset of the next, or when both fuzzy numbers are identical.

To illustrate, consider Figure 3.5. Here, the fuzzy number, B = "around 2" is subtracted from the fuzzy number, A = "about 2", and produces the fuzzy number, C = "about 0". The uncertainties in A and B have a cumulative effect when B is subtracted from A, and results in C being more uncertain than either A or B. So, although C is centred around 0, there is some non-zero possibility that it could have a value of unity, for example.

This is an undesirable characteristic of fuzzy subtraction when used to compare fuzzy numbers. For example, if the fuzzy prediction of the controlled variable is identical to the fuzzy description of the setpoint, any comparison of the two *should* indicate that they are equal. Further, if the fuzzy description of the controlled variable is now less

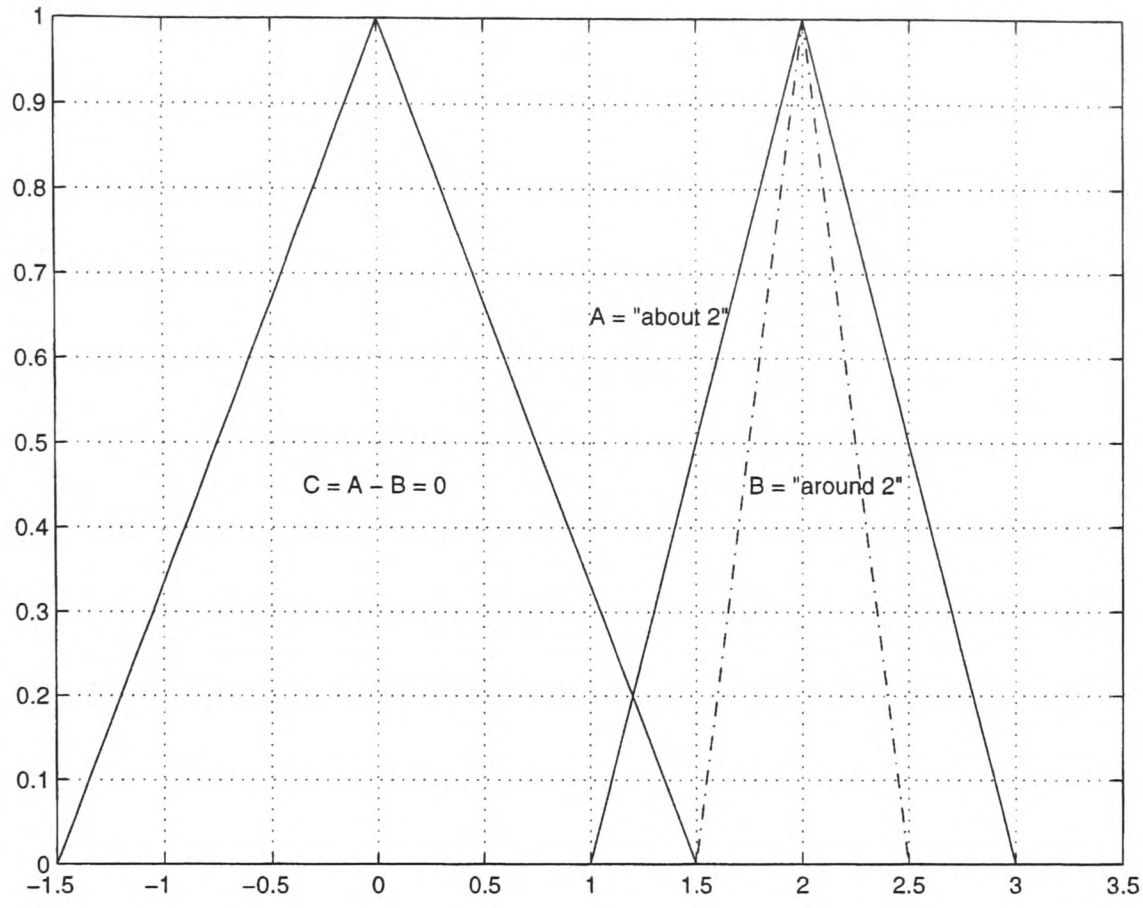


Figure 3.5: Fuzzy Subtraction of Two Fuzzy Numbers: One Contained within the Next

fuzzy than the fuzzy description of the setpoint, and is a subset of this fuzzy setpoint, comparing both should yield zero to indicate zero error between these two quantities. Fuzzy subtraction does not produce this.

The fuzzy proximity function is an attempt to address this inadequacy. It measures the closeness of two fuzzy numbers, by considering the closeness of each element on the universe of discourse of one fuzzy number to elements in the second fuzzy number. Figure 3.6 demonstrates this concept, where the double headed arrow, "$\leftrightarrow$", denotes the fuzzy proximity operator. Note that both A and B remain "about 2" and "around 2", respectively, as they were in Figure 3.5. However, the result of this proximity operation is now a fuzzy singleton at 0, as opposed to a fuzzy number centred around 0.

The fuzzy proximity function is defined as:

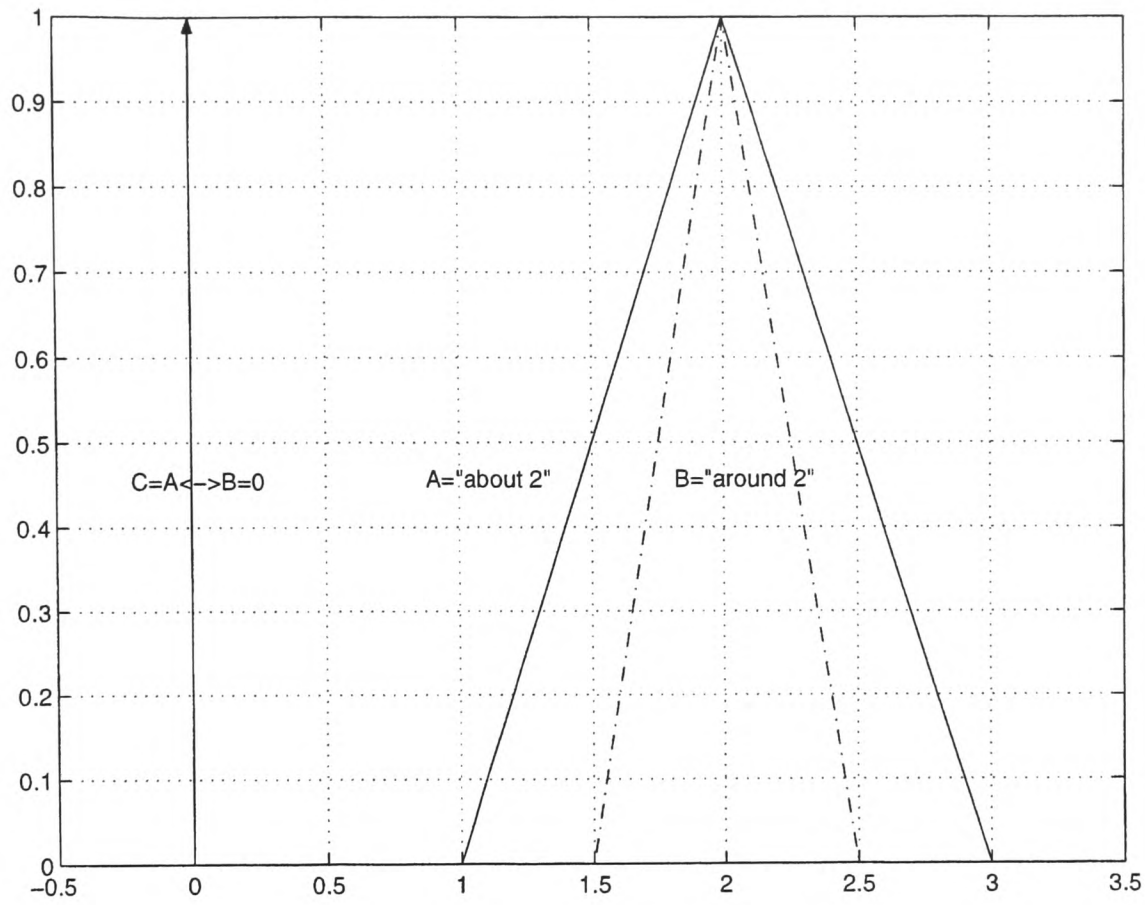


Figure 3.6: Fuzzy Proximity of Two Fuzzy Numbers: One Contained within the Other

$$A \leftrightarrow B = \begin{cases} 0 & A \subseteq B,\ B \subseteq A \\ \sup\limits_{\forall z=|x_A - x_B|} \min(\mu_A(x_A), \mu_B(x_B)) & \text{otherwise.} \end{cases} \tag{3.19}$$

As a rule, the fuzzy proximity of one number to the next is always taken as positive. An explanation of the development of this operator along with an algorithm that performs this function is given in Section 3.3.1, while the Matlab code used for performing the function is given in Section C.1.2 of Appendix C. In addition, Figures 3.7 and 3.8 are examples of the fuzzy proximity of two fuzzy numbers, A and B, where the dashed lines represent the fuzzy number, $A \leftrightarrow B$.

3.3.1 Fuzzy Proximity Algorithm

As outlined above, the fuzzy proximity function has been developed to cope with some of the anomalies normally associated with fuzzy subtraction. The function is realized in

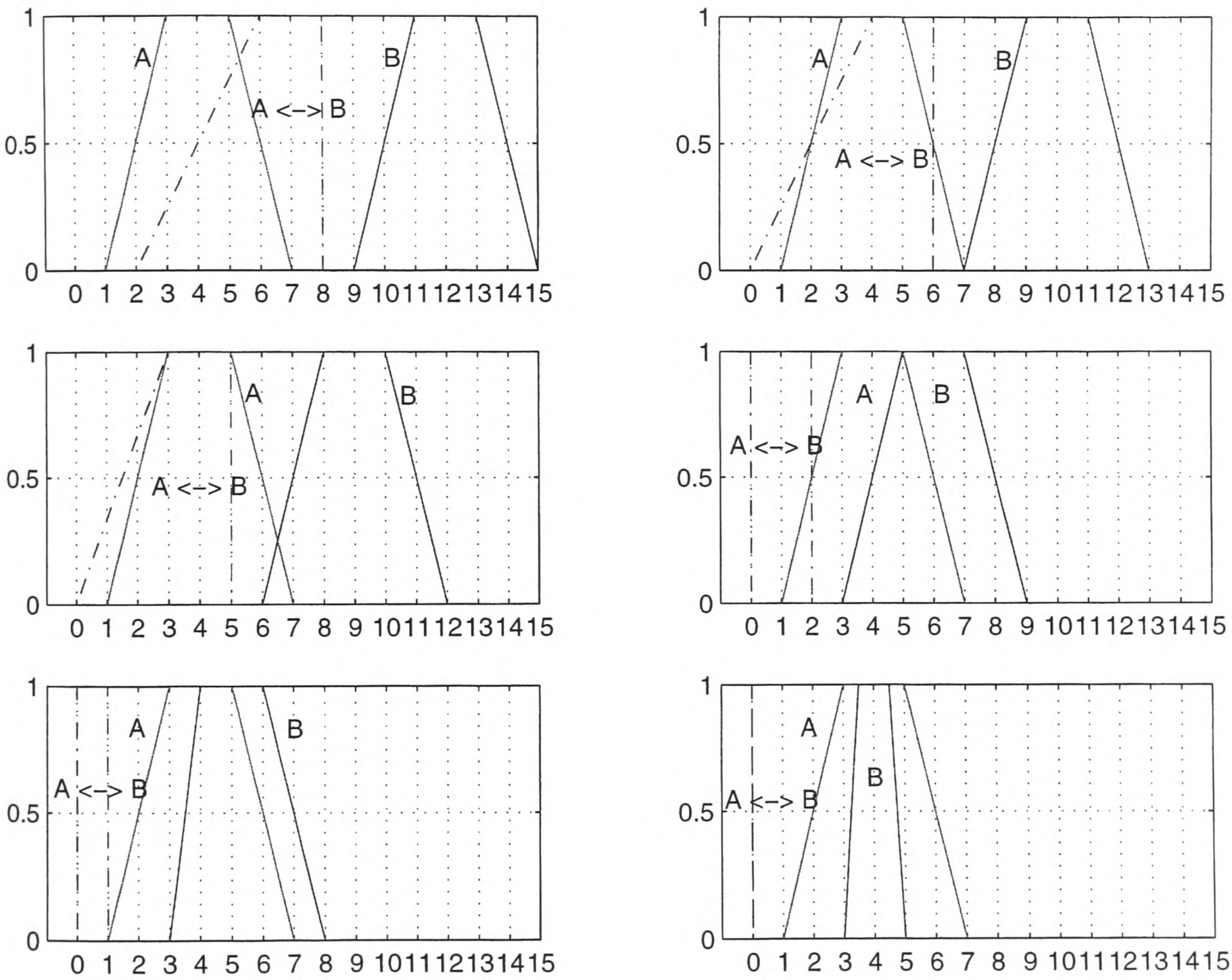


Figure 3.7: Fuzzy Proximity of Two Fuzzy Numbers

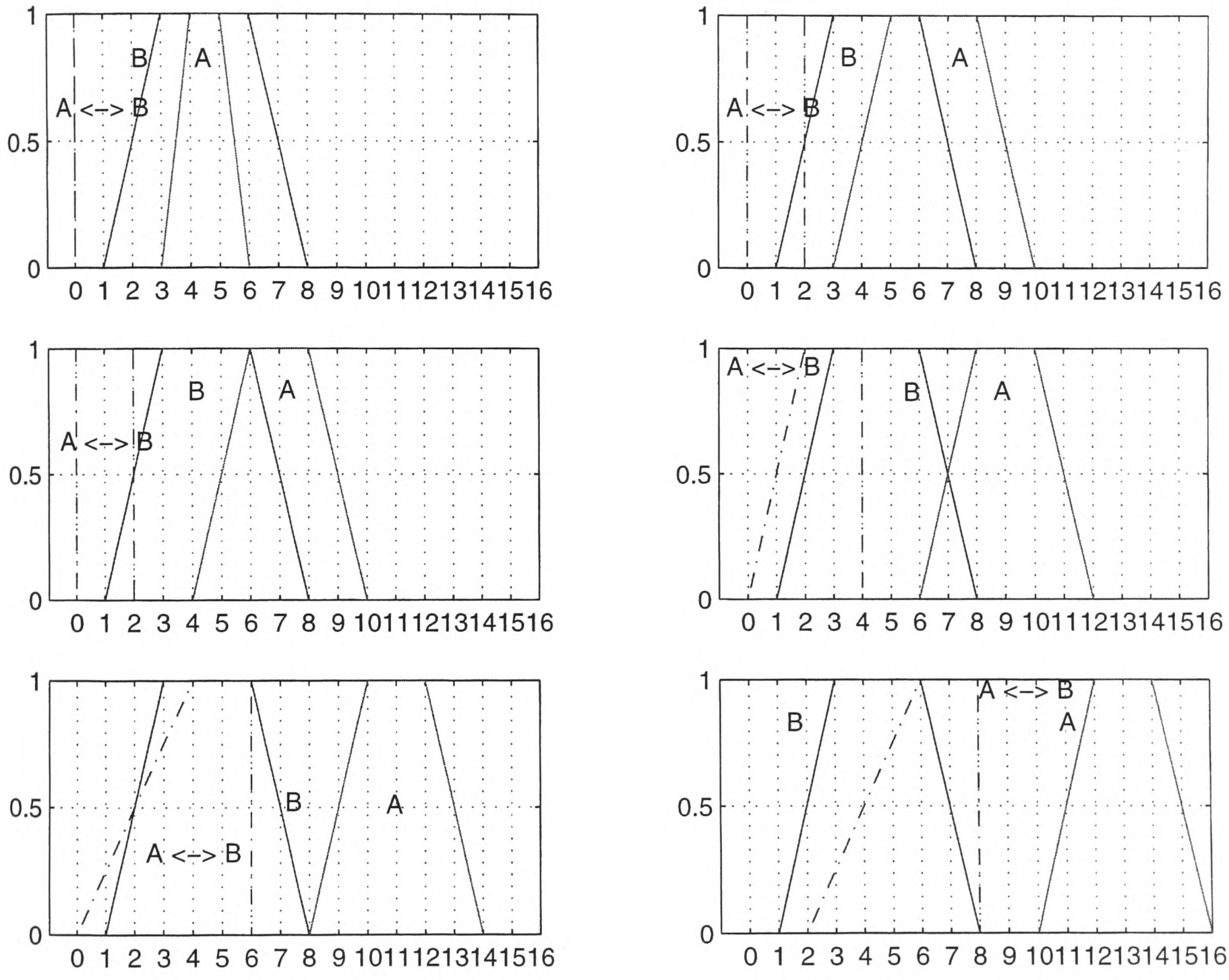


Figure 3.8: Fuzzy Proximity of Two Fuzzy Numbers

Matlab and takes two arguments, W and Y, as inputs. To compute the fuzzy proximity of W to Y, the extension principle (Klir and Yuan, 1995) is employed, and a necessary condition of this algorithm is that at least one element in the W fuzzy set must have a degree of membership of unity (i.e., W must be a normalized fuzzy set). Under this imposed condition, the fuzzy proximity algorithm is as follows:

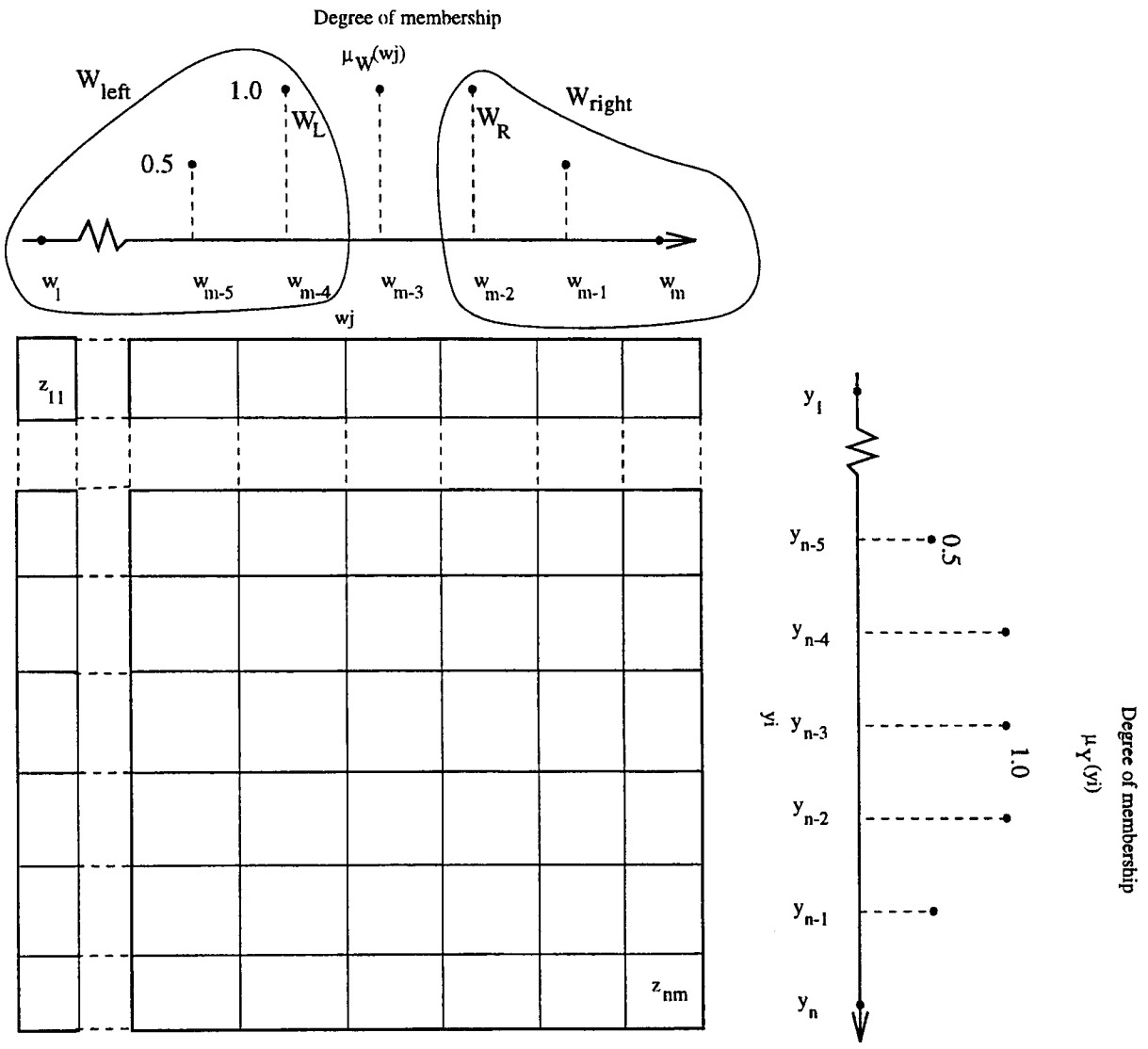


Figure 3.9: The Z space of Fuzzy Proximity

1. Determine the number of elements in the W fuzzy set, m. If an element in this set is represented as w_j, $j = 1 \ldots m$, then the set may be written: $W = \{(w_j, \mu_W(w_j))\}$, where $\mu_W(w_j)$ is the degree of membership of w_j in W.

2. Identify the core of W, as defined in Equation 2.14. The core of the fuzzy set W, denoted $core(W)$, is a crisp set defined as:

$$core(W) = \{x \in X \mid \mu_W(x) = 1\} \tag{3.20}$$

3. Define W_L and W_R as the lower and upper bounds of the core respectively, as shown in Figure 3.9. Therefore, $W_L = min(core(W))$ and $W_R = max(core(W))$. Also, let W_L be the m_l^{th} element in W, and W_R be the m_r^{th} element in W.

4. Define W_{left} as $W_{left} = \{(w_1, \mu_W(w_1)), \ldots, (W_L, 1)\}$. Similarly, define W_{right} as $W_{right} = \{(W_R, 1), \ldots, (w_m, \mu_W(w_m))\}$.

5. Determine the number of elements in the Y fuzzy set, n. If an element in this set is represented as y_i, $i = 1 \ldots n$, then the set may be written: $Y = \{(y_i, \mu_Y(y_i))\}$, where $\mu_Y(y_i)$ is the degree of membership of y_i in Y.

6. Set up an n-by-m matrix, Z. An element in Z is therefore represented by z_{ij}, $i = 1 \ldots n$, $j = 1 \ldots m$. Z represents a space for the universe of the function *proximity*, denoted by the fuzzy set $P = W \leftrightarrow Y$.

7. Initialize Z to 0.

8. Set up a matrix, MV, with the same dimensions as Z. MV represents the space for the membership values in P, of the corresponding space in Z. Therefore, while Z holds proximity values, MV holds the corresponding membership values of these proximities or, more succinctly, $mv_{ij} = \mu_P(z_{ij})$.

9. Initialize MV to 0.

10. If $W_L \leq y_i \leq W_R$, then $Z_{ileft} = 0$ and $MV_{ileft} = \mu_Y(y_i)$, where Z_{ileft} represents the left subset of the i^{th} row vector of the Z matrix (i.e., z_{ij} where $j = 1 \ldots m_l$) and MV_{ileft} represents the left subset of the i^{th} row vector of the MV matrix. Since both the Z and MV matrices were initialized to 0, this step essentially leaves Z at a proximity of 0 while assigning a membership of $\mu_Y(y_i)$ to this proximity. It should be noted that $\mu_Y(y_i)$ is really the result of $min(\mu_Y(y_i), \mu_W(W_L))$, since $\mu_W(W_L) = 1$ by definition.

11. Else if $y_i < W_L$, then $Z_{ileft} = W_{left} - y_i$, $MV_{ileft} = min(\mu_Y(y_i), \mu_W(W_{left}))$.

12. Else $Z_{iright} = y_i - W_{right}$, $MV_{iright} = min(\mu_Y(y_i), \mu_W(W_{right}))$, where Z_{iright} represents the right subset of the i^{th} row vector of the Z matrix (i.e., z_{ij}

where $j = m_r \ldots m$) and MV_{iright} represents the right subset of the i^{th} row vector of the MV matrix.

Since this procedure is repeated for each element in the Y space, each row in Z will represent the fuzzy proximity of that element, y_i, to the elements in the W fuzzy set (each column), while each row in MV will represent the degree of membership or confidence in belief of the corresponding proximities in Z.

By the extension principle, to compute the overall degree of confidence of a particular proximity in the Z space, z_{ij}, $\bar{\mu}_P(z_{ij})$, we need to look over all $Z = z_{ij}$ and take the max of the corresponding $MV = mv_{ij}$:

$$\begin{aligned} for\ \ ii &= 1 : n \cdot m \\ C(ii,1) &= Z(ii) \\ C(ii,2) &= max(MV(find(Z = Z(ii)))) \\ \bar{\mu}_P(z_{ij}) &= C(ii,2) \end{aligned} \tag{3.21}$$

where $find(X = \beta)$ is a Matlab function that returns the indices of the elements of the vector X that are equal to β.

Finally, the algorithm cleans up by:

- removing any redundant rows from C,
- sorting C in ascending order, and
- transferring C to a pre-defined universe structure called PROXCO2.

The entire Matlab code implementation of this algorithm is presented in Section C.1.2 of Appendix C.

3.4 Concluding Remarks

The examination of two issues central to the development of the proposed control scheme has been undertaken in this chapter.

An overview of traditional height defuzzification has been given, and arguments for the investigation of an alternative to this form of defuzzification has been outlined. The concept of conditional defuzzification has been introduced and explained, and four different types of conditional defuzzification schemes have been examined. Incidental to the development of conditional defuzzification is the actuator activity parameter, α, which has also been defined and discussed. The sensitivity of the control scheme to the choice of α and of a particular defuzzification scheme is presented in Chapter 6.

In addition, due to the anomalies encountered when computing with fuzzy numbers, this chapter has also explored the question of an appropriate fuzzy similarity function. The development of the fuzzy proximity function was outlined. It has been shown that the application of this function to the determination of the similarity between two fuzzy numbers has distinct advantages over fuzzy subtraction, for example. An algorithm for performing the fuzzy proximity function has also been detailed, and diagrammatic examples of the fuzzy proximity of two fuzzy numbers given.

The theory developed in Chapters 2 and 3 is applied in Chapters 4 and 5, using the Multiple Fuzzy Cost Functions, Multiple Fuzzy Goals approach.

Chapter 4

Neuro-Fuzzy Control of Thermal Comfort

4.1 Introduction

Although thermal comfort can have a significant effect on the productivity of the occupants of a modern, energy efficient building, the economic cost of low productivity is difficult to define precisely. In addition, the lack of detailed design information and the non-linear behaviour of HVAC systems make them difficult to model accurately. Fortunately, the control performance needed to maintain acceptable thermal comfort is usually less demanding than that required in process control applications. For these reasons, building thermal comfort control (TCC) problems are ideal candidates for the application of the control strategy developed in this thesis (Thompson and Dexter, 2001), (Conte and Fato, 1997).

In this chapter, the neuro-fuzzy predictive control scheme developed in Chapter 2 is applied to the thermal comfort control problem usually encountered in buildings. A simulated zone comfort control problem is presented. This problem represents a scenario where the control objectives of the problem are not clearly defined. It is shown that under such conditions, acceptable control performance may be achieved without the use of detailed modelling and precise (but not necessarily accurate) calculations.

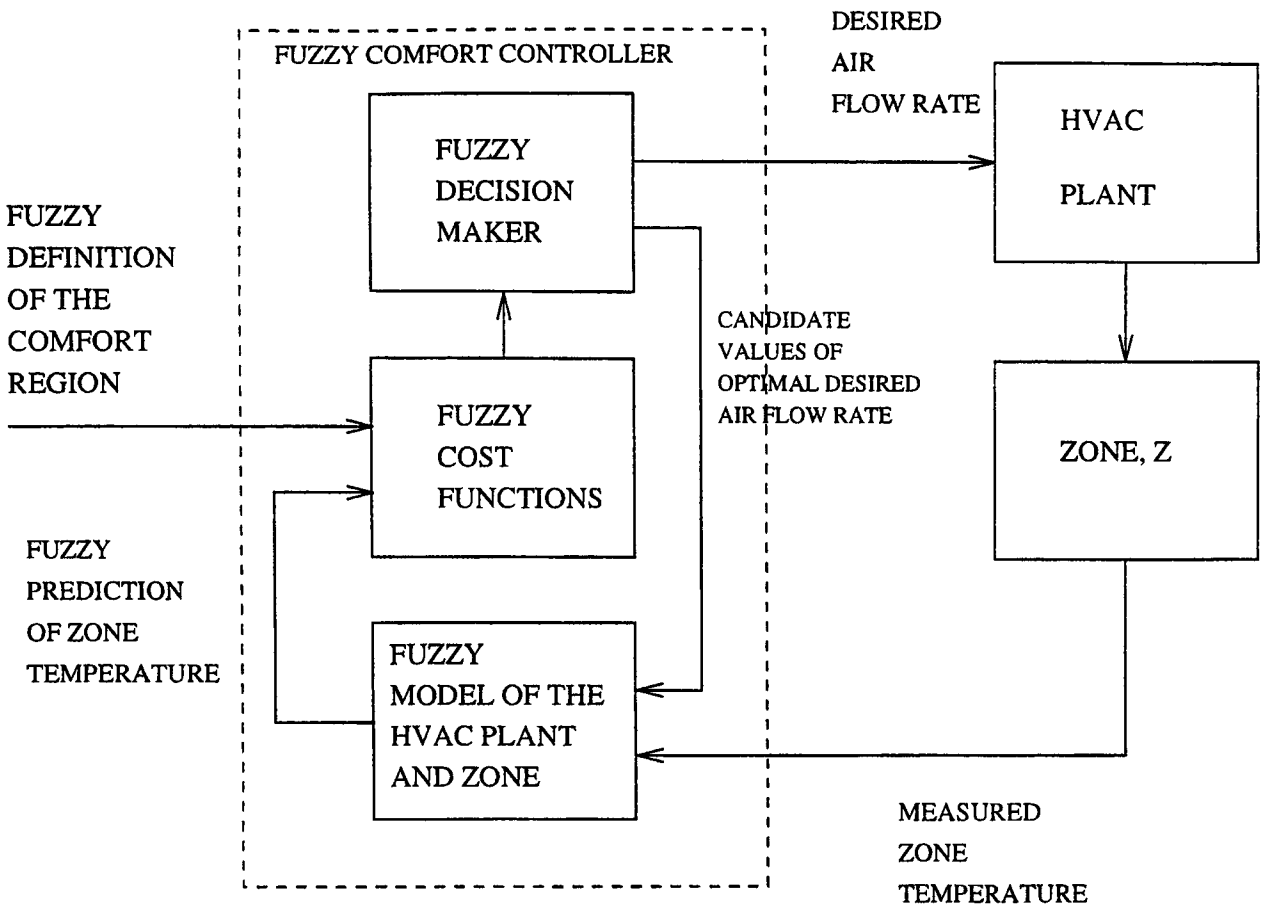


Figure 4.1: Fuzzy Model-Based Comfort Control

4.2 Zone Temperature Control Scheme

A block diagram of the scheme adapted to the control of thermal comfort in a zone is shown in Figure 4.1. In this application, all other factors affecting the thermal comfort of the occupants of a building are assumed constant, as explained in Section 1.3.1, and only the zone temperature is controlled. Control of this temperature is achieved by manipulating the flow of supply air to the zone using a pressure-independent variable air volume (VAV) terminal box. Since there is a non-linear relationship between the VAV box damper position and the supply air mass flow rate, $\dot{m}_a$, entering the zone, the damper is actually driven by a local Proportional-plus-Integral (PI) controller whose setpoint is set by the control signal from the neuro-fuzzy controller. The design of this local PI controller is not considered in this thesis, but is assumed to be such that it satisfactorily tracks the setpoint.

In addition to the thermal comfort criterion, it is also desirable that the energy consumption of the system form a basis upon which the control decisions are made. Thus energy considerations represent a constraint criterion in this system.

For this application of the scheme, the Multiple Fuzzy Cost Functions, Multiple Fuzzy Goals approach of Section 2.4.2 is used. A fuzzy measure of thermal comfort is formed for each value of the control action in the universe of discourse by considering the zone temperature setpoint, w, and a fuzzy prediction of the zone temperature, $\hat{T}_z$. In addition, a fuzzy measure of the energy consumption is also formed for each value of the control action. These cost functions are compared to their respective pre-defined fuzzy goals to determine the extent to which these goals are satisfied (Bellman and Zadeh, 1970). The comparison is made by the Fuzzy Decision-Maker. The results of these respective comparisons are then aggregated and used as the grades of membership of the optimal value of the manipulated variable which, in this case, is the air mass flow rate to the zone under control (Yi and Chung, 1993). The fuzzy optimal current air mass flow rate, $\dot{m}_a(n)$, is then passed to a conditional defuzzification scheme to help reduce actuation wear in the terminal box of the system. The procedure is repeated at each sampling instant.

4.3 Simulink Zone Model

The zone model used in this work is a detailed simulation based on one of the zones from a building located in the south of France. This detailed zone model simulation is courtesy of the *Centre Scientifique et Technique du Batiment* (CSTB) in France, and the interested reader should consult (Husaunndee, 2001) for details of the model.

The zone represents the office spaces of the south facing section of the 2^{nd}, 3^{rd} and 4^{th} storeys of the French building. The thermal behaviour of the zone is quite complex, and the simulation of this thermal behaviour is carried out in a simulation package called Simulink, which is a block diagram dynamic system simulator for Matlab. For this work, there are 5 levels (level 1 to level 5) of schematic details relevant to the zone under consideration. Figure 4.2 is the lowest level (level 5) schematic diagram of the Simulink simulation. The next four higher levels of this Simulink model (levels 4, 3, 2 and 1) are provided in Appendix D, for completeness.

A quick look at Appendix D hints at the level of detail of the simulated model. In Figure 4.2, T_{ext} is the external temperature (which takes on a value equal to the ambient temperature, T_{amb}, in this chapter), **CMV** is a vector representing controlled

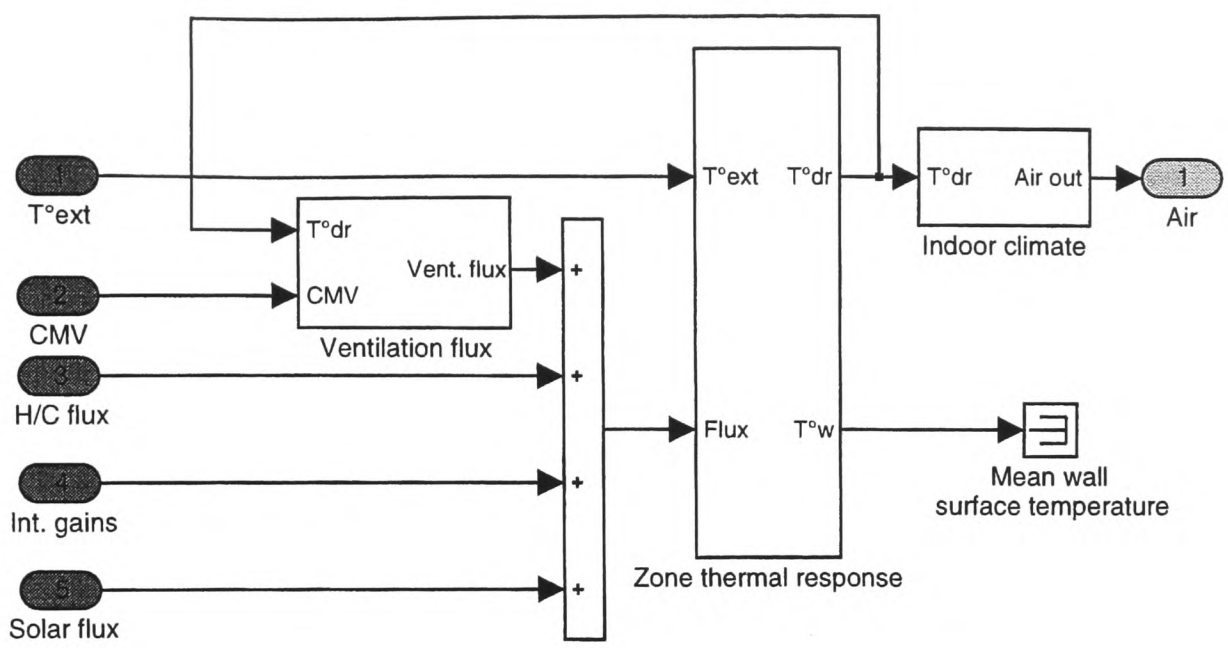


Figure 4.2: Simulink Model: Lowest Level (level 5) of Zone Model

mechanical ventilation and infiltrations, H/C flux represents the heating/cooling loads supplied by the air-conditioning system (the heating load is switched off after the zone is warmed up in preparation for occupancy), Int.gain is the internal gain of the zone and Solar flux represents the load due to solar radiation. For this work, the CMV input to the zone under consideration is disabled. The block labeled "Zone thermal response" is a second-order model of the thermal behaviour of the zone, with an output vector, **Air**, the first element of which is the dry bulb air temperature of the zone.

4.3.1 Simplified Physical Model of the Zone Temperature

To identify the fuzzy relational array which will be used as the core of the neuro-fuzzy model of the zone, it is necessary to capture some input/output training data representative of the general dynamics of the thermal behaviour of this zone. For the identification scheme which is used to identify the fuzzy relational array from this training data, training at the apex of each input fuzzy reference set is desirable (Tan and Dexter, 1996). This would mean maximum activation (firing) of each input fuzzy reference set. This scenario is quite difficult to achieve in practice (particularly because certain combinations of the values of each input are difficult to achieve). One way to

circumvent this problem and ensure apex training is to develop and train on a simplified physical model of the zone's thermal behaviour (Salsbury, 1998). By so doing, there is now control over the values of the variables which are inputs to the neuro-fuzzy model, and these variables can therefore be set to the value on the apex of the input fuzzy reference sets, since both model development and training take place in simulation.

In addition, developing and training on a simplified zone model has the advantage of being a less system-specific approach than using a more complex model. Model complexity is usually an important consideration when modelling HVAC systems since very complex, specific models need to be significantly altered when adopted for use on a different HVAC system. Models which are able to capture the general thermal behaviour of these systems without significantly sacrificing their validity are generally preferred.

The thermal behaviour of the zone, though complex, may be approximated as shown in Figure 4.3 which is constructed to realize a compromise between accuracy and complexity. Figure 4.3 (a) shows the four major heat loads acting on the zone. Q_a is the heat gain due to the difference in temperature between the ambient air and the zone. If the ambient temperature, T_a, is greater than the zone temperature, T_z, then Q_a will be positive and will flow into the zone. Q_a is therefore given by:

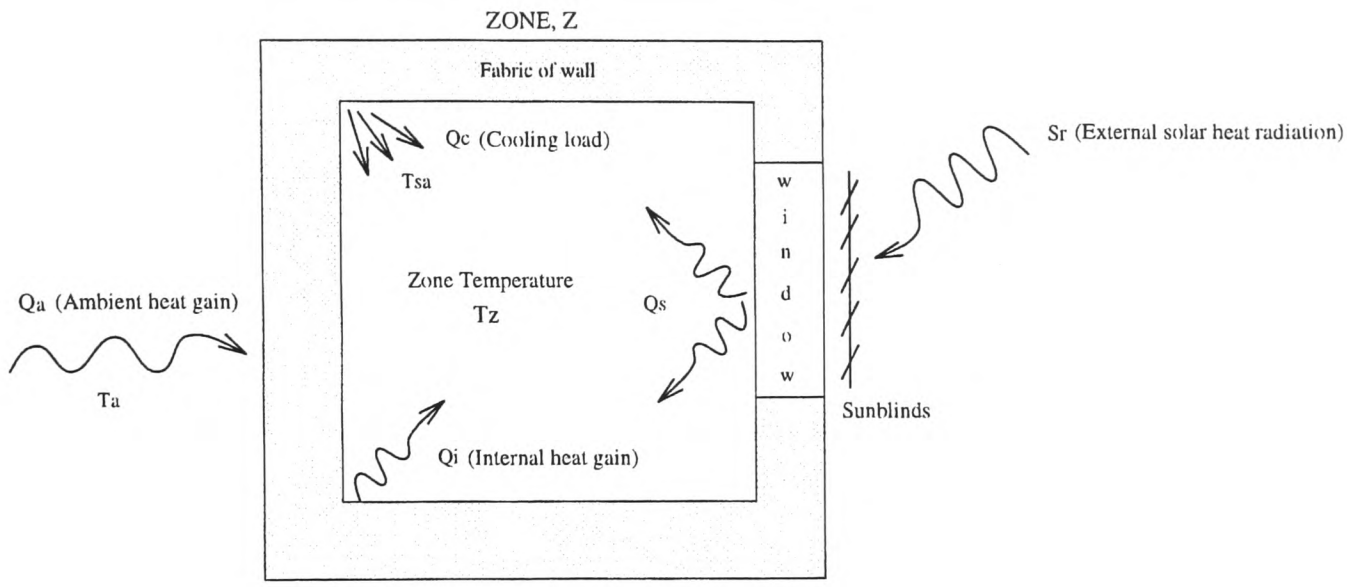
$$Q_a = UA(T_a - T_z) \tag{4.1}$$

where UA is the overall heat loss coefficient by conduction for the zone, in W/K.

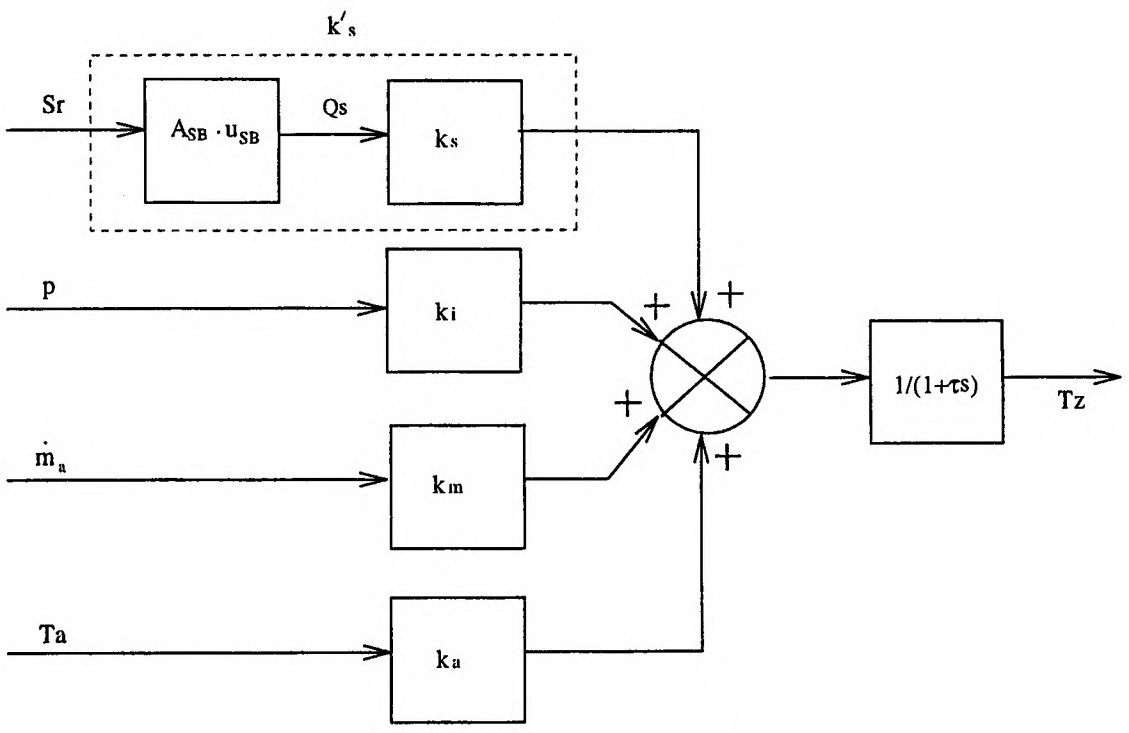
The external solar heat gain acting on the zone is denoted by S_r, which is taken as the net external heat component of the solar radiation, per square metre, on the window. This heat component acts through external sunblinds which may be set from 0% to 100% open, and is therefore scaled by a factor $0 \leq u_{SB} \leq 1.0$, to produce the net solar heat radiation entering the zone, Q_s. Q_s is given by:

$$Q_s = S_r \cdot A_{SB} \cdot u_{SB} \tag{4.2}$$

where A_{SB} is the area of the sunblinds (i.e. the area of the windows), in m^2.



a) Heat Loads Contributing to Temperature of Zone to be Modelled



b) Equivalent Linear Thermal Model of Zone Temperature

Figure 4.3: Simple Linear Thermal Model of Zone

The third heat load acting on the zone is the internal heat gain due to people and equipment, Q_i. In commercial building thermal load design calculations, each person and piece of equipment in a zone is usually assumed to make some constant heat contribution per square metre of floor space of that zone. For this zone, the heat gain due to occupants and related equipment taken together is estimated as 40 W/m^2 (ASHRAE, 1981). Since the floor surface area of the zone is 663.75 m^2, $Q_i = 26,550$ W. In addition, the zone has a presence sensor which sends a binary signal, p, to the building control system whenever there is someone present in the zone. This binary signal is used to formulate a crude estimate of the internal heat gain in this model. Q_i is therefore given by:

$$Q_i = 26,550 \cdot p, \quad p \in \{0,1\} \tag{4.3}$$

The final thermal load considered in this work is the cooling load of the zone, Q_c. This load is the thermal load under control, and is met by varying the air mass flow rate of the cool air entering the zone. If the temperature of this supply air is T_{sa} and the specific heat capacity of the air is denoted as c_{air}, then this thermal load is given by:

$$Q_c = -\dot{m}_a \cdot c_{air} \cdot (T_z - T_{sa}) \tag{4.4}$$

where $\dot{m}_a$ is the air mass flow rate of the supply air, in kg/s. Note that since Q_c is a cooling load, its effect is to remove heat from the zone, and it is therefore negative.

An equivalent linear model of the thermal loads acting on the zone can be constructed by considering the effect of these four loads acting together as in Figure 4.3 (b). Each input in this model acts through a steady state gain and has a corresponding time constant associated with its individual effect on temperature in the zone (later, some assumptions are made about the nature of the individual time constants which allow one general time constant to be used for each input).

The net solar radiation acts both on the walls of the zone and directly on the air inside the zone. The walls absorb this heat and then radiate it into the zone over a longer period of time than the period of time over which the heat radiated into the zone from the more direct heating of the zone air occurs. This shorter time constant associated

with direct heating of the air, τ_{solar}, is similar in magnitude to that associated with the heating effect of the internal heat gain from people and equipment in the zone, $\tau_{internal}$. This is because the internal heat generated also acts directly on the zone air. A similar argument can also be used for the magnitude of the time constant associated with the cooling effect of air mass flow into the zone, τ_{m_a}.

We therefore have:

$$\tau_{solar} \approx \tau_{internal} \approx \tau_{m_a} = \tau_{air} \tag{4.5}$$

where τ_{air} may be considered as the average time constant of direct heat transfer to the zone air due to direct solar radiation and internal heat gain and of direct heat transfer from the zone air due to the cooling load.

The ambient temperature changes very slowly in comparison to the remaining three inputs to the model. In addition, its dominant effect is obtained from heat conduction through the fabric of the walls of the zone. The time constant associated with this effect, τ_{wall}, is therefore much longer than τ_{air} and acts only on T_a. τ_{wall} is associated with the indirect heat transfer from the ambient air to the zone air via the fabric of the walls.

The thermal behaviour of the zone could therefore be appropriately modelled by a second-order system with two time constants, τ_{air} and τ_{wall}.

A simpler first order approximation involving one time constant, τ, is used, where τ is chosen to strike a compromise between the shorter time constant, τ_{air}, and the much longer time constant, τ_{wall}.

To compute the respective steady state gains of the system, we note that, at any point in time, the sum of all heat flows into the zone is equal to the heat capacity of the zone, C_z, times the rate of change of zone temperature with respect to time:

$$\sum_j Q_j = C_z \frac{dT_z}{dt} \tag{4.6}$$

or,

$$Q_a + Q_s + Q_i + Q_c = C_z \frac{dT_z}{dt} \tag{4.7}$$

Therefore, substituting Equations 4.1, 4.2, 4.3 and 4.4 into Equation 4.7, we have:

$$UA(T_a - T_z) + S_r \cdot A_{SB} \cdot u_{SB} + 26550 \cdot p - \dot{m}_a c_{air} \cdot (T_z - T_{sa}) = C_z \frac{dT_z}{dt} \tag{4.8}$$

$$\begin{aligned}
\Rightarrow\ C_z \frac{dT_z}{dt} + \dot{m}_a c_{air} \cdot (T_z - T_{sa}) &= UA(T_a - T_z) + S_r A_{SB} u_{SB} + 26550p \\
\Rightarrow\ C_z \frac{dT_z}{dt} + (\dot{m}_a c_{air} + UA) \cdot T_z &= \dot{m}_a c_{air} T_{sa} + 26550p + S_r A_{SB} u_{SB} + UAT_a \\
\Rightarrow\ C_z \cdot s \cdot T_z + (\dot{m}_a c_{air} + UA) \cdot T_z &= \dot{m}_a c_{air} T_{sa} + 26550p + S_r A_{SB} u_{SB} + UAT_a \\
\Rightarrow\ \left(s \cdot \frac{C_z}{\dot{m}_a c_{air} + UA} + 1 \right) \cdot T_z &= \frac{\dot{m}_a c_{air} T_{sa} + 26550p + S_r A_{SB} u_{SB} + UAT_a}{\dot{m}_a c_{air} + UA} \\
\Rightarrow\ (s \cdot \tau + 1) \cdot T_z &= \frac{c_{air} T_{sa}}{\dot{m}_a c_{air} + UA} \cdot \dot{m}_a + \frac{26550}{\dot{m}_a c_{air} + UA} \cdot p + \\
&\quad \frac{A_{SB} u_{SB}}{\dot{m}_a c_{air} + UA} \cdot S_r + \frac{UA}{\dot{m}_a c_{air} + UA} \cdot T_a
\end{aligned} \tag{4.9}$$

where $\tau = \frac{C_z}{\dot{m}_a c_{air} + UA}$, and s is the Laplace Transform of $\frac{d}{dt}$. Therefore,

$$T_z = \left(\frac{1}{1 + \tau s} \right) \cdot \left[\frac{c_{air} T_{sa}}{\dot{m}_a c_{air} + UA} \cdot \dot{m}_a + \frac{26550}{\dot{m}_a c_{air} + UA} \cdot p + \frac{A_{SB} u_{SB}}{\dot{m}_a c_{air} + UA} \cdot S_r + \frac{UA}{\dot{m}_a c_{air} + UA} \cdot T_a \right] \tag{4.10}$$

At steady state, $s = 0$, and so:

$$T_z = \frac{c_{air} T_{sa}}{\dot{m}_a c_{air} + UA} \cdot \dot{m}_a + \frac{26550}{\dot{m}_a c_{air} + UA} \cdot p + \frac{A_{SB} u_{SB}}{\dot{m}_a c_{air} + UA} \cdot S_r + \frac{UA}{\dot{m}_a c_{air} + UA} \cdot T_a \tag{4.11}$$

This expression gives the steady state zone temperature of the zone as a function of the four inputs to the linear model: the external solar heat radiation per square metre entering the zone, S_r, the presence sensor signal, p, the air mass flow rate of the supply air, $\dot{m}_a$, and the ambient temperature, T_a. The steady state gains of this linear model can now be deduced from considering Equation 4.11.

Since the steady state gain of each input to the linear model is the coefficient of this input, by observation:

$$k'_s = \frac{A_{SB}u_{SB}}{\dot{m}_a c_{air} + UA} \tag{4.12}$$

$$k_i = \frac{26550}{\dot{m}_a c_{air} + UA} \tag{4.13}$$

$$k_m = \frac{c_{air}T_{sa}}{\dot{m}_a c_{air} + UA} \tag{4.14}$$

$$k_a = \frac{UA}{\dot{m}_a c_{air} + UA} \tag{4.15}$$

where k'_s is the steady state gain of the external solar radiation, S_r, k_i is the steady state gain of the presence sensor signal, k_m is the steady state gain of the air mass flow rate, and k_a is the steady state gain of the ambient temperature.

Since the position of the sunblinds can take on only discrete values of either 0%, 20%, 40%, 60%, 80% or 100%, and since A_{SB} is known and constant for the zone under consideration (102 m^2), then the first input to the model, S_r, may be replaced by the net solar radiation entering the zone (see Figure 4.3 (a)), Q_s, by considering Equation 4.2. This substitution makes training on this input easier since no training data then needs to be generated for each of the 6 possible sunblind positions. Instead, training takes place on the universe of discourse of Q_s which would be appropriately set so as to contain all possible values of Q_s given the possible values of u_{SB}.

The steady state gain of Q_s, k_s, is therefore given from Equation 4.12 by:

PARAMETER	VALUE
Window area, A_{sb}	102 m^2
Specific heat capacity of air, c_{air}	1010 J/kgK
Conduction heat loss coefficient, UA	879 W/K
Supply air temperature, T_{sa}	12°C
Sampling time	300 s
Zone time constant, τ	7200 s

Table 4.1: Design Information of Simplified Physical Zone Model

$$k_s = \frac{k'_s}{A_{SB}u_{SB}} = \frac{1}{\dot{m}_a c_{air} + UA} \tag{4.16}$$

Table 4.1 summarizes the design information needed to construct the simplified physical zone model used for generation of the training data.

4.3.2 The Fuzzy Relational Model of the Zone Temperature

The fuzzy relational model used here in the zone thermal comfort control scheme has five inputs and one output (See Figure 4.4). The structure of this non-linear first-order plus time delay auto-regressive model is similar to that proposed in (Bremner and Postlethwaite, 1997) and is given by:

$$T_z(n) = T_z(n-1) \circ \dot{m}_a(n-1-T_D) \circ Q_s(n-1) \circ T_a(n-1) \circ p(n-1) \circ R \tag{4.17}$$

where $T_z(n)$ and $T_z(n-1)$ are the current and previous values of the zone temperature respectively, $\dot{m}_a(n-1-T_D)$ is the delayed setpoint for the air flow through the pressure-independent VAV terminal boxes supplying cold air to the zone (which, in this case, is the control signal computed by the scheme), $Q_s(n-1)$ is an estimate of the previous solar disturbance to the system, $T_a(n-1)$ is the previous ambient temperature, $p(n-1)$ is the Boolean output of a presence sensor that indicates whether the zone was previously occupied, $\circ$ is the fuzzy composition operator (here, sum-of-product

composition is used), T_D is the dead time on the control action expressed as an integer number of sampling intervals, and R is the fuzzy relational array.

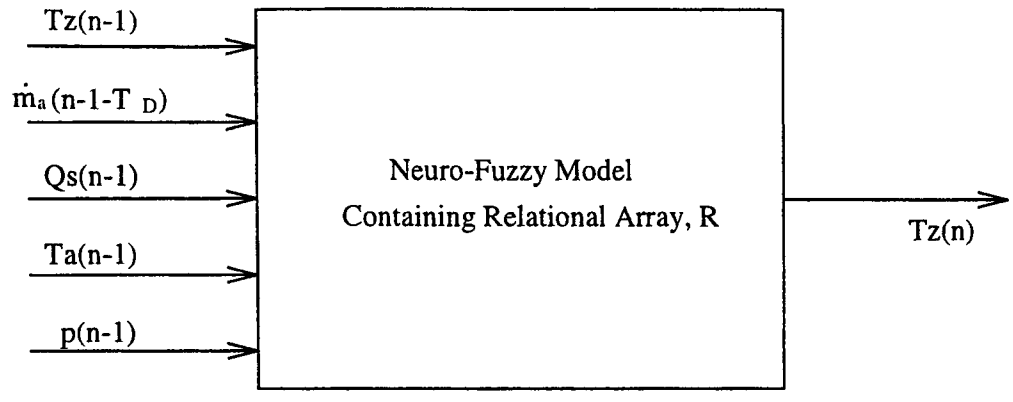


Figure 4.4: Neuro-Fuzzy Model of Zone Temperature

The model has eleven linearly spaced triangular fuzzy reference sets with 50% overlap which describe $T_z(n-1)$. Eleven sets are chosen because, since it is desired that the zone temperature be controllable over a 10°C range (from 19°C to 29°C), if a 10% uncertainty in measuring the zone temperature is assumed, the actual zone temperature could be anywhere in the range ±1.0°C of the measured zone temperature. Eleven sets over a range of 10°C models this 10% uncertainty in the system.

In addition, five linearly spaced triangular fuzzy reference sets with 50% overlap are defined on the universe of discourse of $\dot{m}_a(n-1-T_D)$ and 25 sets for $\hat{T}_z(n)$. The relatively large number of fuzzy sets is needed on the predicted zone temperature to produce a fuzzy output that reflects the uncertainties. Normally, interpretability may not necessarily be a concern when choosing the number of fuzzy reference sets placed on the output universes of fuzzy models, because the output is defuzzified to produce a crisp (and, therefore, interpretable) value. However, if the output of these models are not defuzzified, they remain uninterpretable. In contrast, many output sets are preferred where defuzzification of the model output will not be carried out, for fear of a loss of the information contained in them, but where interpretability is desirable. In such cases, a very uncertain variable has many sets being fired across its universe, whereas a more certain one has less fuzzy sets fired.

Figure 4.5 shows the fuzzy reference sets defined over the respective universe of discourse of the five fuzzy inputs and the fuzzy output.

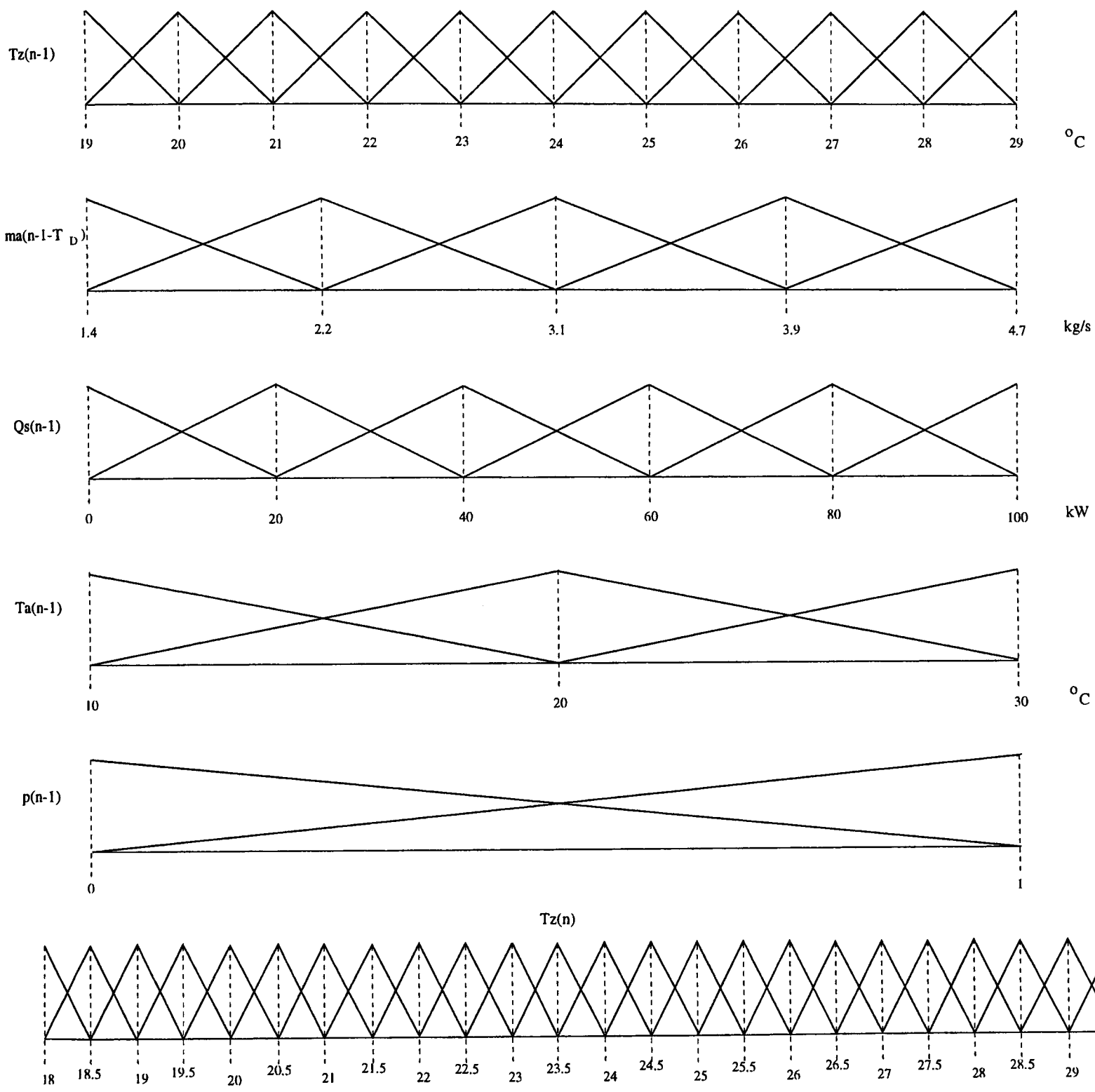


Figure 4.5: Fuzzy Reference Sets Defined on the Input and Output Spaces

4.3.3 Training and Identification of the Fuzzy Model

In this thesis, the fuzzy relational array is identified from data gathered from training on the *simplified thermal model* (detailed above in Section 4.3.1) of the zone under consideration. Table 4.2 lists *part* of the training data gathered from training on this simplified thermal model. The entire training data consists of all possible combinations of the values of the apexes of the fuzzy sets describing the universe of discourse of all five inputs.

$T_z(n)/°\mathrm{C}$	$\dot{m}_a(n)/kgs^{-1}$	$Q_s(n)/W$	$T_a(n)/°\mathrm{C}$	$p(n)$	$\hat{T}_z(n+1)/°\mathrm{C}$
⋮	⋮	⋮	⋮	⋮	⋮
24	1.4	40000	20	1	24.8
24	2.2	40000	20	1	24.5
24	3.1	40000	20	1	24.3
24	3.9	40000	20	1	24.1
24	4.7	40000	20	1	24.0
25	1.4	40000	20	1	25.8
25	2.2	40000	20	1	25.4
25	3.1	40000	20	1	25.2
25	3.9	40000	20	1	25.1
25	4.7	40000	20	1	25.0
⋮	⋮	⋮	⋮	⋮	⋮
28	1.4	100000	30	1	29.9
28	2.2	100000	30	1	29.2
28	3.1	100000	30	1	28.8
28	3.9	100000	30	1	28.6
28	4.7	100000	30	1	27.4
29	1.4	100000	30	1	30.8
29	2.2	100000	30	1	30.2
29	3.1	100000	30	1	29.8
29	3.9	100000	30	1	29.5
29	4.7	100000	30	1	29.3

Table 4.2: Section of Training Data for Zone Model

As an example of the information contained in this table, consider the first row of data. The *if* ... *then* rule described by this row is:

> *If:*
> the **current zone temperature**, $T_z(n)$, is 24°C and the **current air mass flow rate into the zone**, $\dot{m}_a(n)$, is 1.4 kg/s and the **current net solar radiation entering the zone**, $Q_s(n)$, is 40000 W and the **current ambient temperature**, $T_a(n)$, is 20°C and the **zone is currently occupied**, $p(n)$ is 1,
>
> *Then:*
> the **next value of zone temperature**, $\hat{T}_z(n+1)$, is 24.8°C.

This value of 24.8°C is computed from the simplified zone model (which is representing the actual zone) using the expression for steady state zone temperature in Equation 4.11, and by considering the dynamics of the zone with a dominant time constant of 7200 seconds. If training took place by actually interacting with the zone (as opposed to a simulation of it), $\hat{T}_z(n+1) = 24.8$°C would be measured (not computed), but would probably be very difficult to attain given the particular combinations of the values of the five inputs in the antecedent of this rule.

Once the values in the sixth column of this table, $\hat{T}_z(n+1)$, are obtained, they are used to identify the relational array, R. In this work, R is identified using a less computationally intensive modification to the Ridley, Shaw and Kruger (RSK) identification algorithm (Ridley, Shaw and Kruger, 1988). This method has been shown to be superior to other identification techniques (Postlethwaite, 1994), (Sing and Postlethwaite, 1997). Appendix B gives an overview of the RSK method and the modification used here.

A section of the relational array, R, produced by the application of this identification algorithm to the training data in Table 4.2, is shown in Section B.4 of Appendix B.

4.3.4 The Fuzzy Relational Array, R

Consider the section of R shown in Section B.4 of Appendix B. Moving across a particular row of R (i.e. traversing all the columns of R), the predicted zone temperature, $\hat{T}_z(n+1)$, increases from 18°C to 30°C in 0.5°C steps at each column. Additionally, moving down a particular column of R (i.e. traversing all the rows of R), the current air mass flow rate, $\dot{m}_a(n)$, increases from 1.4 kg/s to 4.7 kg/s in 0.82 kg/s steps at each row. Therefore, the predicted zone temperature, $\hat{T}_z(n+1)$, is defined along each row of R (i.e. the number of columns of R is equal to the number of fuzzy reference sets defined on $\hat{T}_z(n+1)$), and the current air mass flow rate, $\dot{m}_a(n)$, is defined along each column of R (i.e. the number of rows of R is equal to the number of fuzzy reference sets defined on $\dot{m}_a(n)$).

The remaining four indices of the relational array are current zone temperature, $T_z(n)$, current net solar radiation entering the zone, $Q_s(n)$, current ambient temperature, $T_a(n)$, and current occupancy status, $p(n)$, respectively. Therefore, an entry of $R(5, 13, 6, 3, 3, 2) = 0.85639$ in the relational array means that if $\dot{m}_a(n) = 4.7$ kg/s (1^{st} index = 5), and $T_z(n) = 24$°C (3^{rd} index = 6), and $Q_s(n) = 60$ kW (4^{th} index = 3), and $T_a(n) = 30$°C (5^{th} index = 3) and $p(n) = 1$ (6^{th} index = 2), then there is 85.64% confidence that the temperature at the next sampling instant (the predicted zone temperature), $\hat{T}_z(n+1) = 24$°C (2^{nd} index = 13).

Prediction Plots

Figures 4.6 and 4.7 show prediction plots of the neuro-fuzzy model of zone temperature, using traditional height defuzzification. In the first plot of Figure 4.6, the effect on the predicted zone temperature, $\hat{T}_z(n+1)$, of varying both the current zone temperature, $T_z(n)$, and the current air mass flow rate, $\dot{m}_a(n)$, is plotted. This is done while holding constant $Q_s(n) = 60\ kW$, $T_a(n) = 25$°C, and $p(n) = 0$. The variable on the z-axis, $\Delta T_z(n+1)$, is defined as $\hat{T}_z(n+1) - T_z(n)$. Notice the non-linearity in the change in zone temperature as the air mass flow rate is decreased.

The second plot of Figure 4.6 shows the effect on the predicted zone temperature, $\hat{T}_z(n+1)$, of varying both the current occupancy status, $p(n)$, and the current air mass

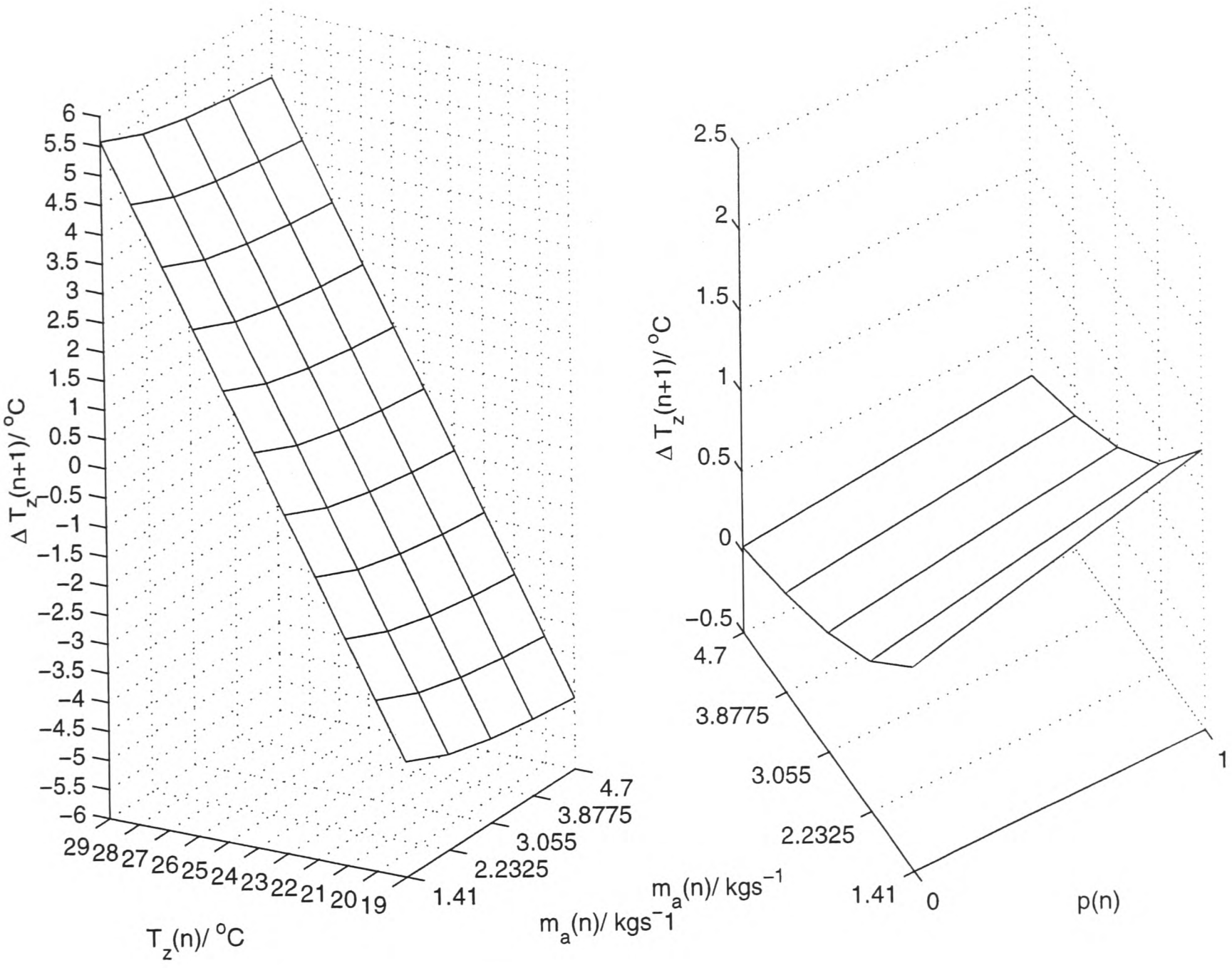


Figure 4.6: Prediction Plots of the Neuro-Fuzzy Model

flow rate, $\dot{m}_a(n)$, while keeping $Q_s(n) = 60\ kW$, $T_a(n) = 25°\text{C}$, and $T_z(n) = 24°\text{C}$ constant. This plot demonstrates that the non-linearity which occurs in the predicted zone temperature as the air mass flow rate is changed is more pronounced at lower air flow rates than at higher flow rates.

Figure 4.7 also has a pair of plots. In the first plot, the effect on the predicted zone temperature, $\hat{T}_z(n+1)$, of varying both the current net solar radiation entering the zone, $Q_s(n)$, and the current air mass flow rate, $\dot{m}_a(n)$, is graphed, while keeping $p(n) = 0$, $T_a(n) = 25°\text{C}$, and $T_z(n) = 24°\text{C}$ constant. Additionally, the second plot shows the effect on the predicted zone temperature, $\hat{T}_z(n+1)$, of varying both the

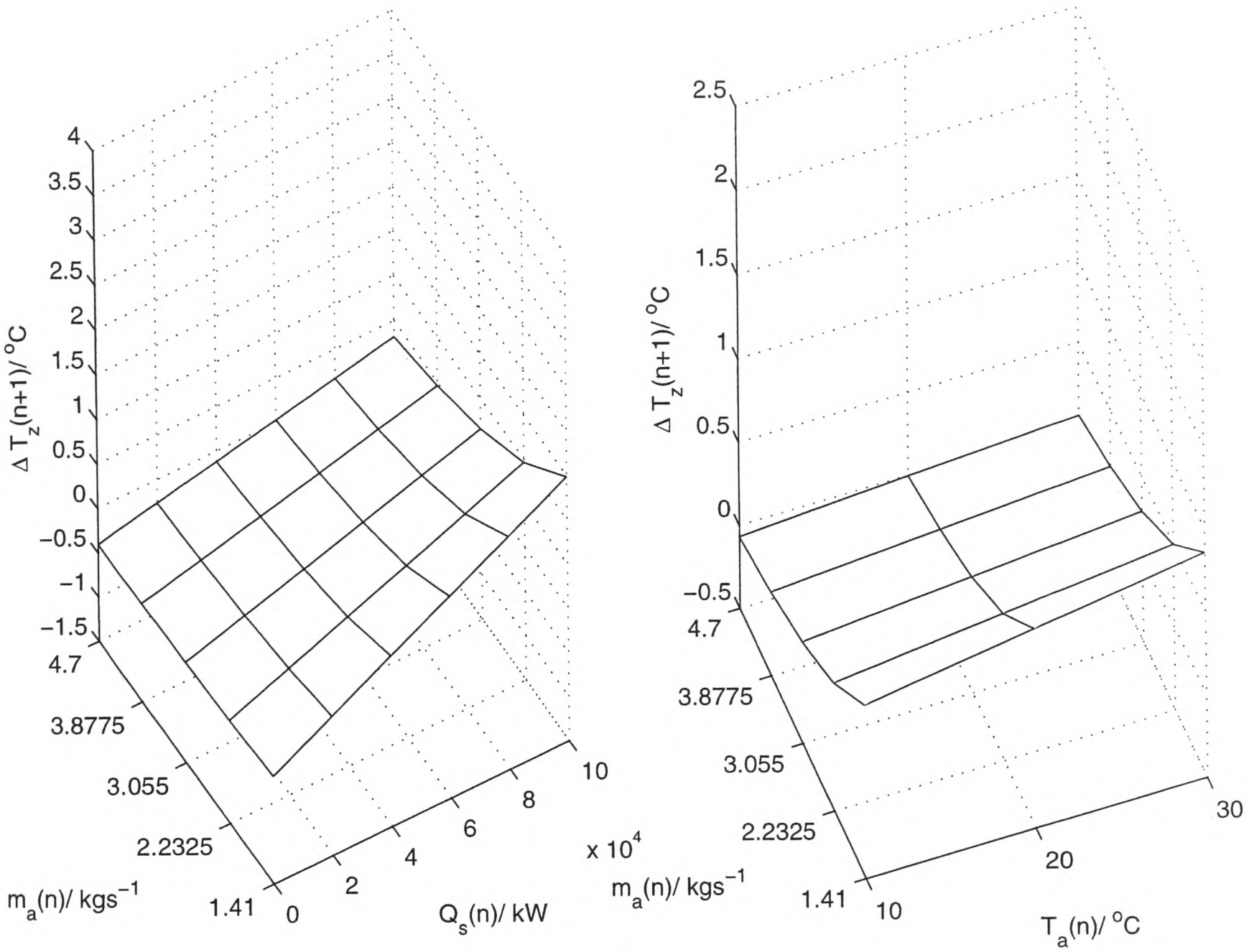


Figure 4.7: Prediction Plots of the Neuro-Fuzzy Model

current ambient temperature, $T_a(n)$, and the current air mass flow rate, $\dot{m}_a(n)$, while keeping $p(n) = 0$, $Q_s(n) = 60\ kW$, and $T_z(n) = 24°\text{C}$ constant.

4.4 Multiple Fuzzy Cost Functions Used in the Zone Temperature Control Scheme

The cost function used in this work is composed of two competing components established with respect to the criteria of comfort and energy consumption, as in Section 2.4.2.

The two-part fuzzy cost function is:

$$J_c(u) = (w(n+2) \leftrightarrow \hat{T}_z(n+2)) \tag{4.18}$$

$$J_e(u) = U(n) \tag{4.19}$$

where J_c is the fuzzy cost of thermal comfort and J_e is the fuzzy cost of energy consumption.

The two-step-ahead predicted zone temperature is used on the assumption that the dead time on the control action, T_D, is of an order of magnitude of one sampling interval, and therefore, considering the inherent unit delay on the control action, any action initiated at the current sample time, n, will not have an effect on the system earlier than time $n+2$. The minimum and maximum costing horizons chosen for $J_c(u)$ are $N_1 = N_2 = 2$ (see Equation 1.1). This choice implies that, although the 1-step-ahead predicted zone temperature, $\hat{T}_z(n+1)$, is used to compute the 2-step ahead prediction, $\hat{T}_z(n+2)$, it is not directly used in the computation of $J_c(u)$.

These two components of the cost function provide a measure of thermal comfort and a measure of energy consumption. The proximity of the 2-step-ahead fuzzy prediction of the zone temperature to the temperature setpoint is used as a measure of the thermal comfort, while a fuzzy cost on the control action attempts to regulate the system energy, on the assumption that system energy is directly proportional to air mass flow rate (i.e. the control action) (Kummert et. al., 1997).

Once the fuzzy cost functions have been established, they are evaluated against their respective fuzzy goals in accordance with the *Multiple Fuzzy Cost Functions, Multiple*

Fuzzy Goals scheme. Aggregation of the two resultant possibility vectors (one vector for each criterion) is achieved via the AND operator which is implemented by taking the *product* of the corresponding elements of both possibility vectors. This ensures that *both* the comfort and energy criteria are considered when deciding on the optimal control action.

4.5 Fuzzy Prediction with a Fuzzy Input

The cost function of Equation 4.18 uses the 2-step-ahead predicted zone temperature, $\hat{T}_z(n+2)$. The neuro-fuzzy model producing this prediction takes as an input, the 1-step-ahead predicted zone temperature, $\hat{T}_z(n+1)$. However, $\hat{T}_z(n+1)$ is *not* a crisp variable, since the output from the first-step-prediction in this model is *not* defuzzified.

To produce the second-step-ahead prediction, therefore, a method of fuzzy prediction using a fuzzy input needs to be investigated.

4.5.1 Developing the Fuzzy-Input Neuro-fuzzy Model

To develop the fuzzy-input neuro-fuzzy model, the fuzzification section of a traditional model is adapted to accept fuzzy inputs (in this case, $\hat{T}_z(n+1)$). Figure 4.8 indicates how this might be achieved. Since the output from the first-step-prediction model is fed through a similar model to produce the second-step-prediction, the universe of discourse of the input remains the same as in the first-step prediction, i.e., it is the universe of discourse of $T_z(n)$. Notice that, whereas in a traditional fuzzification scheme the input is a crisp variable, Figure 4.8 shows the input (indicated by the dashed arrows) as a possibility vector indicating the fuzzy variable, $\hat{T}_z(n+1)$.

The degree of membership of the fuzzy input, $\hat{T}_z(n+1)$, in the input fuzzy reference sets defined on $T_z(n)$ is computed by taking the maximum of the possibility of *each* reference set defined on $T_z(n)$, given $\hat{T}_z(n+1)$:

$$\mu_{T_z}(\hat{T}_z(n+1)) = max(Poss(T_z \mid \hat{T}_z(n+1))) \tag{4.20}$$

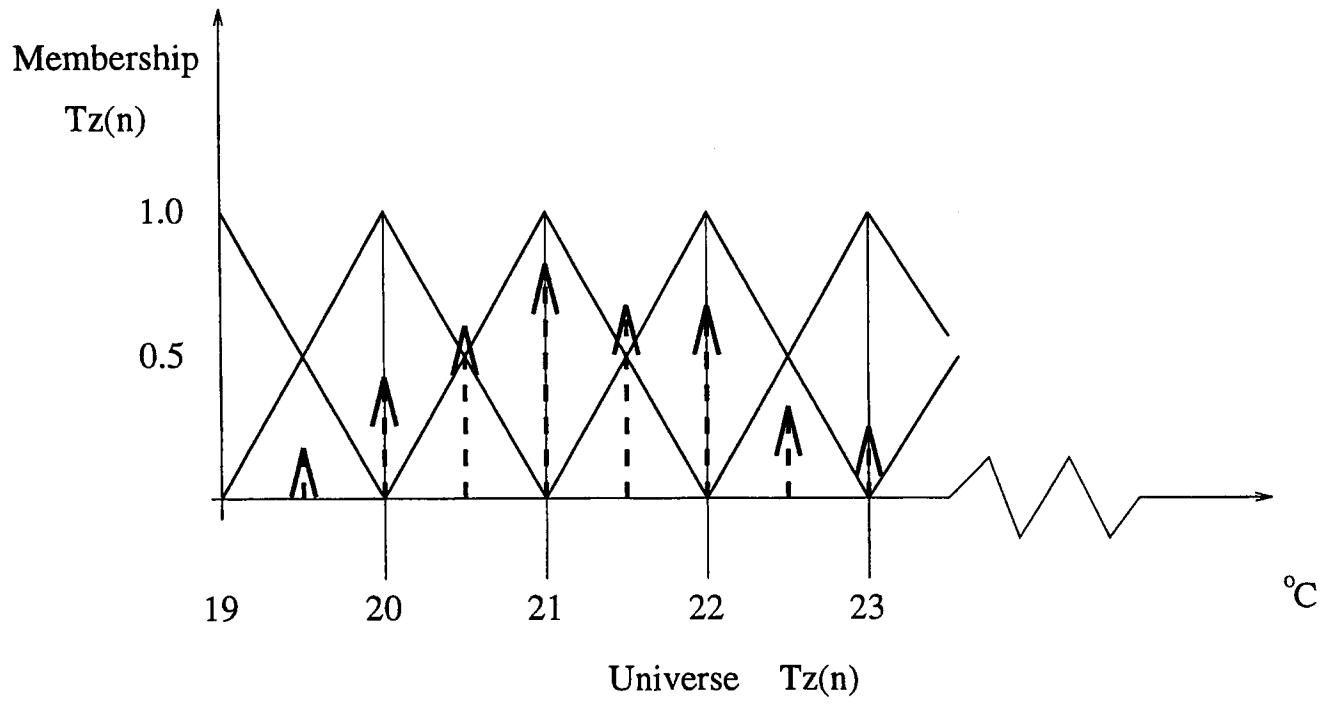


Figure 4.8: Fuzzy-Input Neuro-Fuzzy Prediction

Since $Poss(T_z \mid \hat{T}_z(n+1))$ is given as the $Min(T_z,\ \hat{T}_z(n+1))$, then Equation 4.20 becomes:

$$\mu_{T_z}(\hat{T}_z(n+1)) = max(Min(T_z,\ \hat{T}_z(n+1))) \tag{4.21}$$

From Figure 4.8, it is seen that, for any given fuzzy reference set defined on $T_z(n)$, only three elements in the possibility vector describing $\hat{T}_z(n+1)$ are relevant in the calculation of Equation 4.21. These three elements fall either on the apex of the fuzzy reference set in question, or exactly mid-way between the apexes of adjacent sets. Due to this (which is as a result of the fact that there are twice as many fuzzy reference sets defined on the output predicted zone temperature as there are defined on the input current zone temperature), for the purposes of calculating the minimum values, the fuzzy reference sets defined on $T_z(n)$ may be reduced to the vector $\upsilon = [0.5\ \ 1.0\ \ 0.5\ \ \ldots\ \ 0.5]'$, where the length of υ is equal to the number of fuzzy reference sets defined on the output $\hat{T}_z(n+1)$ (25 in this case).

Equation 4.21 therefore reduces to:

$$\mu_{T_z}(\hat{T}_z(n+1)) = max(min(\upsilon,\ \hat{T}_z(n+1))) \tag{4.22}$$

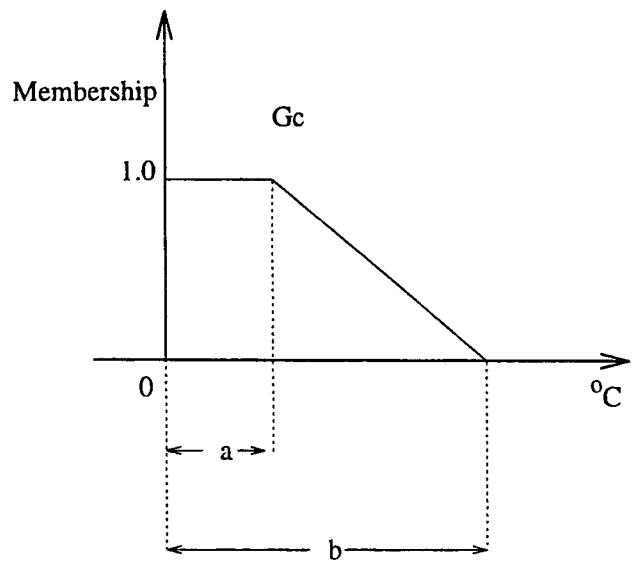
Therefore, whereas due to the structure of the fuzzy reference sets used in this work (linearly spaced 50% overlap sets) a crisp variable would fire only two fuzzy reference sets, a fuzzy variable may fire more than two, since the vector μ_{T_z} from Equation 4.22 may have less than 23 non-zero elements.

4.6 The Fuzzy Goals

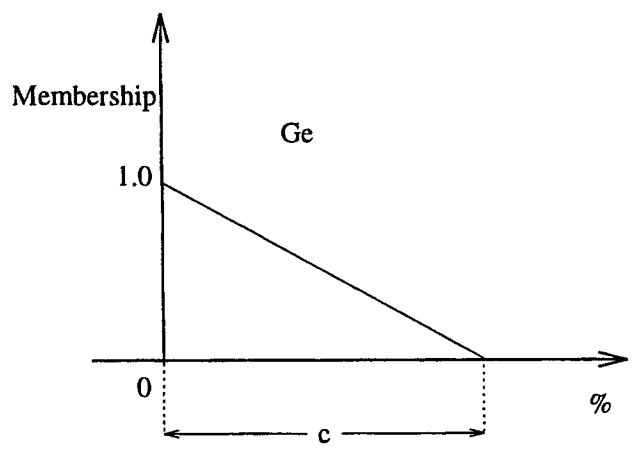
Since both comfort and energy are being taken into consideration here and since the *Multiple Fuzzy Cost Functions, Multiple Fuzzy Goals* approach to evaluating the fuzzy cost function is being used, two fuzzy goals are used in the decision-making process. The first fuzzy goal, G_c, is the comfort goal. It is trapezoidal with the lowest value on the universe of discourse being 0°C and with the length of the parallel sides being a°C and b°C (Figure 4.9a). The width of this fuzzy goal is therefore defined as b, with $core(G_c) = a$.

The fuzzy cost of thermal comfort, $J_c(u)$, is compared to this fuzzy goal and, because of the geometry of G_c, the implication here is that any fuzzy cost of thermal comfort, $J_c(u) = (w(n+2) \leftrightarrow \hat{T}_z(n+2))$, in the interval [0°C, a°C) is equally acceptable. Those in the interval [a°C, b°C] would also be deemed acceptable, though to a linearly decreasing extent.

The second fuzzy goal, G_e, is the energy goal. G_e is triangular, with the lowest value on the universe of discourse representing minimum air flow. Therefore, the figure as shown has $core(G_e) = 0$, and $width(G_e) = c$. The geometry of G_e implies that since minimum air flow would result in the lowest possible energy consumption, we want the control action closest to the minimum air flow as possible, but those greater than the minimum would also be deemed acceptable, again to a linearly decreasing extent. The relative importance of maintaining thermal comfort and of saving energy is captured by the relative widths of the fuzzy goal functions, as discussed in Section 2.4.2.



a) Comfort Fuzzy Goal



b)Energy Fuzzy Goal

Figure 4.9: Fuzzy Goals Defined on Comfort and Energy

In this work, a takes on a value of 1.0°C. The maximum allowable values of b and c are 24.5°C and 120% of maximum air flow, respectively.

Figure 4.10 shows the mechanics of adjusting the goal in order that they remain achievable (Section 2.5.1). In the figure, the energy goal, G_e, is adjusted from $\sigma_e = 1$ to $\sigma_e = 6$ in a series of steps. This scenario would occur if, for example, the initial energy goal were unachievable (given the relative importance of thermal comfort and low energy consumption), and therefore needed to be relaxed.

The increase in goal widths means that larger values of the fuzzy cost of thermal comfort (more discomfort) and larger values of the fuzzy cost of energy consumption (more energy consumption) would now be more acceptable.

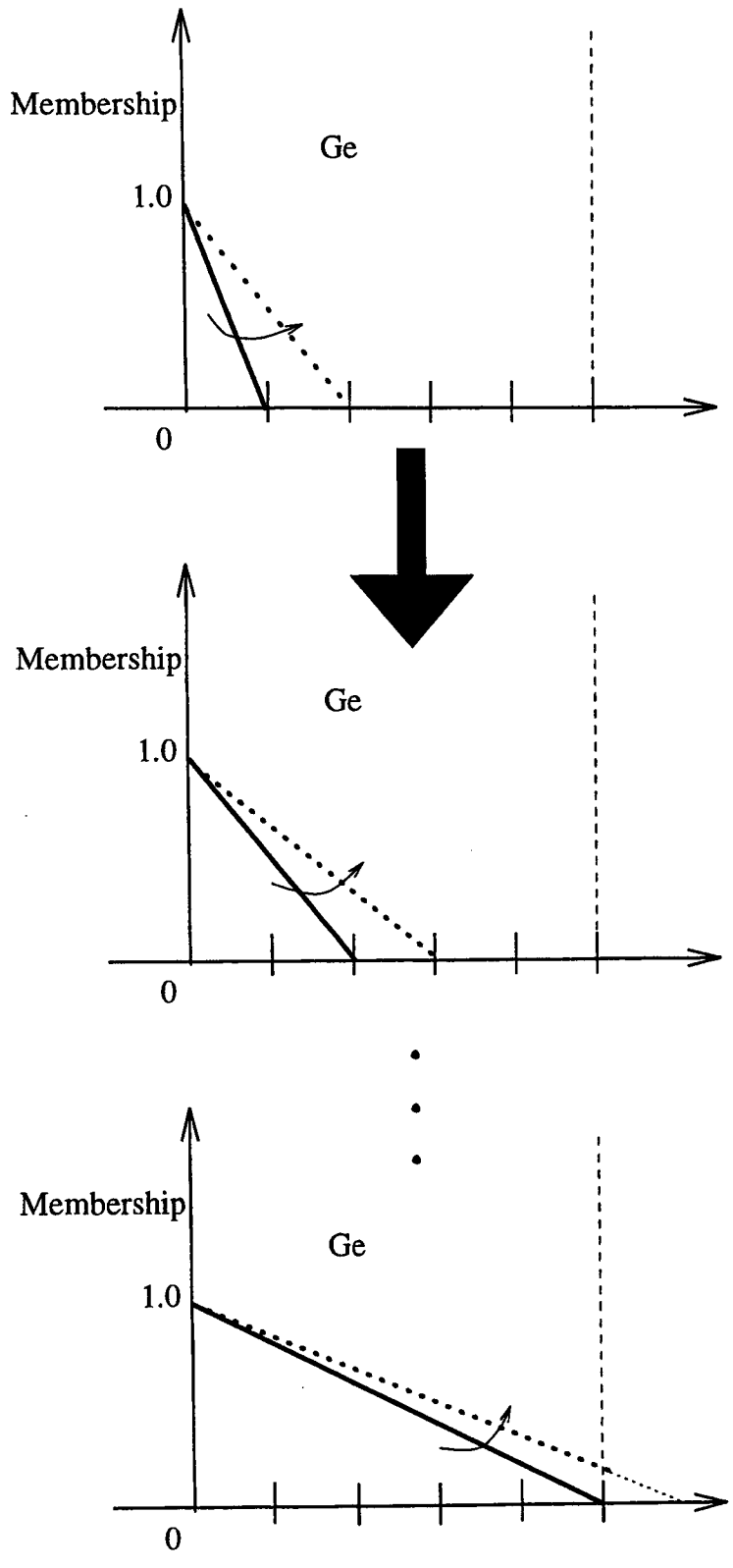


Figure 4.10: Achievable Fuzzy Goal

4.7 Zone Control Results

In this section, simulation results are presented that compare the performance of the comfort control scheme with a traditional PI controller. The simulated zone detailed in this chapter is setup to operate on a typical mid-summer's day (in July), and has a simulation run time of 86,400 seconds (24 hours), with occupancy from 27,000 to 66,600 seconds (07:30 to 18:30). A re-heat coil (not considered in this work) is activated before the start of occupancy to heat the zone up to the comfort region. Once in the region, the coil is switched off, and the cooling controller is allowed to regulate the zone temperature. During occupancy, the presence sensor is set to 1, and the internal gain of 26,550 W is assumed for this period.

Figure 4.11 is a plot of the ambient temperature variation, T_a, observed during simulation, while Figure 4.12 shows the direct normal external solar heat radiation, S_r, over this period.

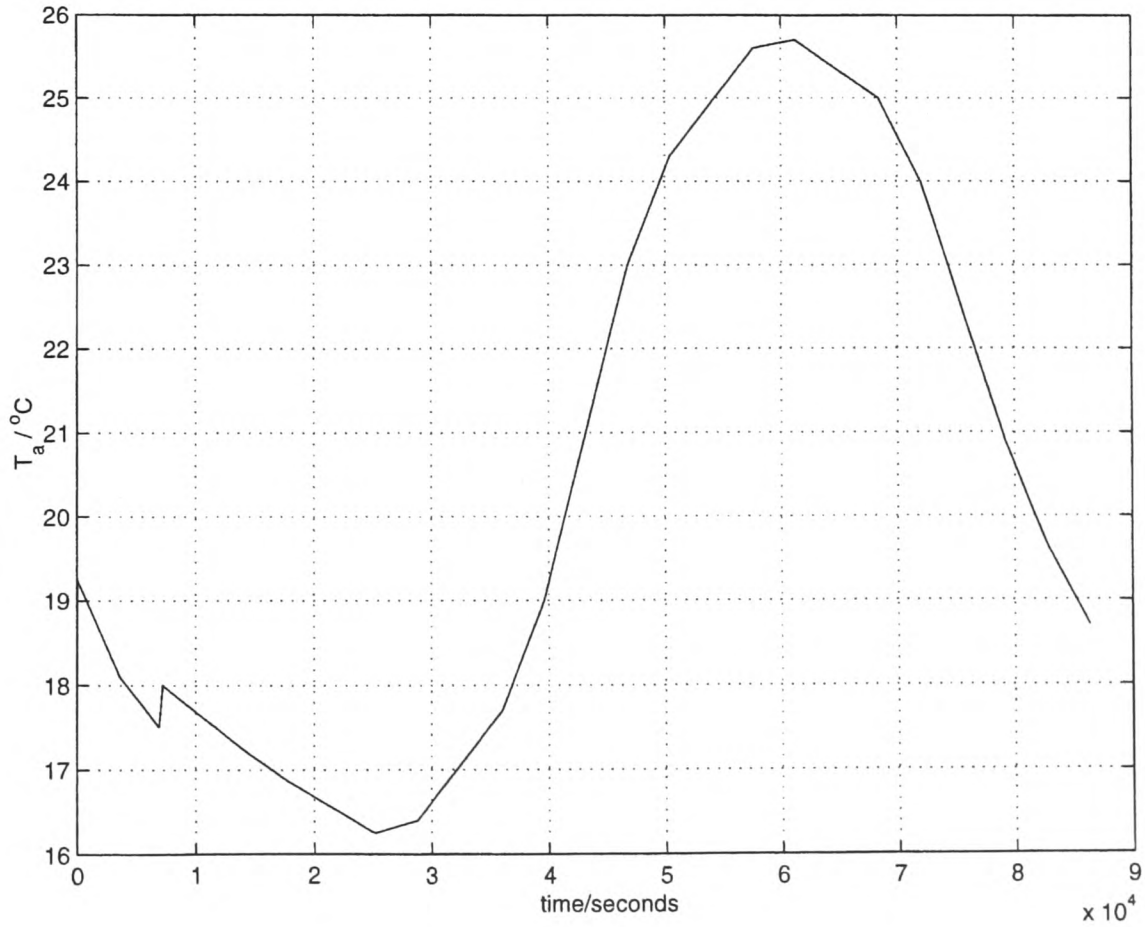


Figure 4.11: Ambient Temperature Variation during Simulation

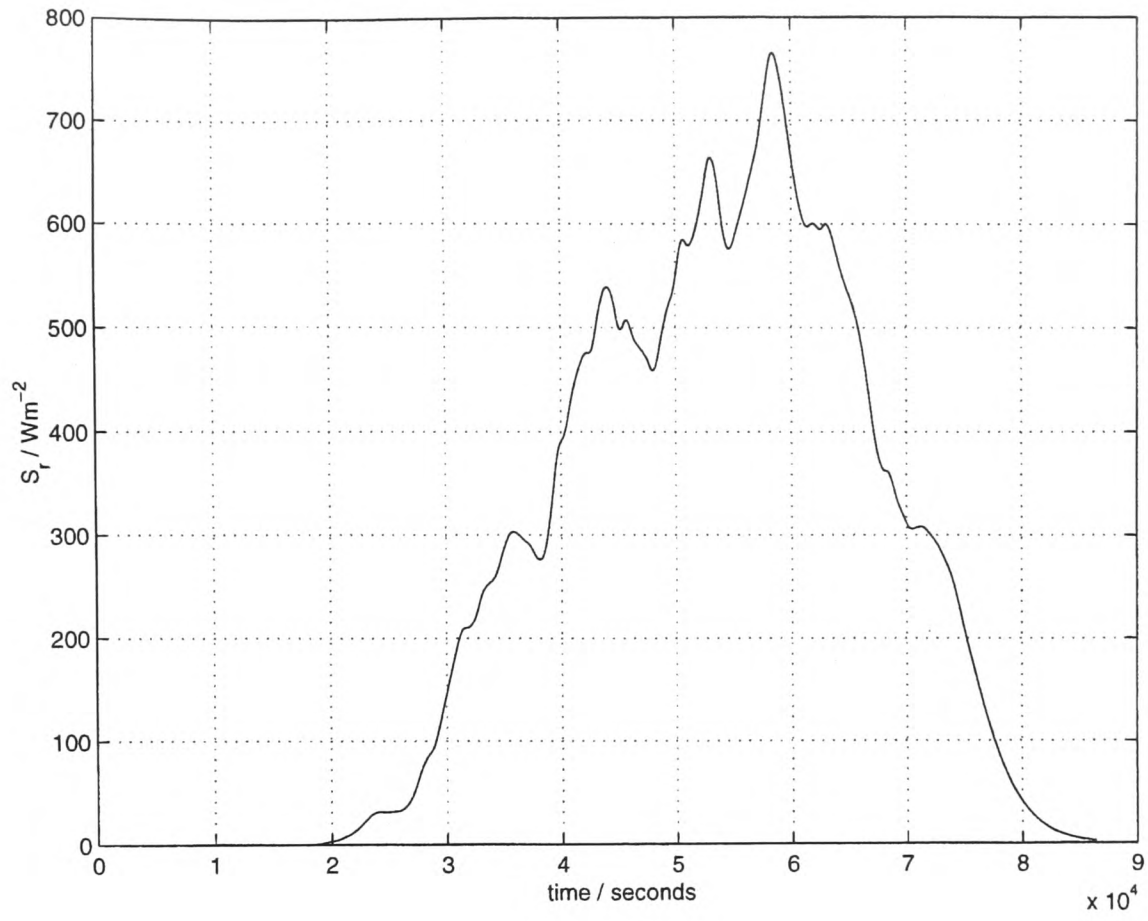
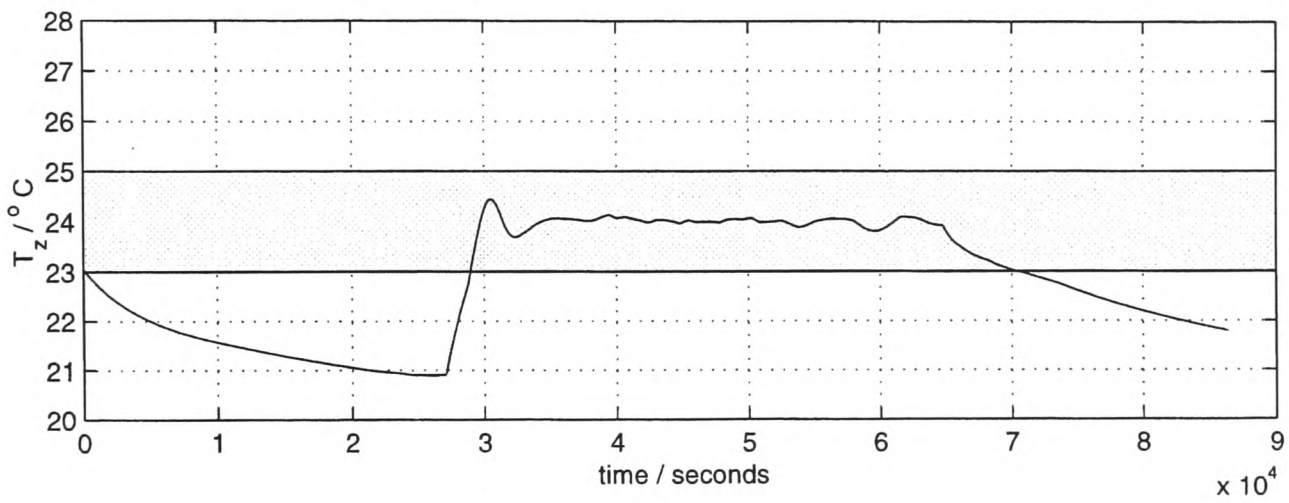


Figure 4.12: Direct Normal External Solar Heat Radiation during Simulation

4.7.1 PI Control of Zone

As a means of gauging the extent to which the proposed control scheme performs relative to established schemes, a PI controller is used to control the zone temperature under consideration. The controller is tuned using a trial-and-error method until the response of the system is satisfactory. This results in a proportional gain, $k_p = 0.05\,\mathrm{kgs}^{-1}\ ^\circ\mathrm{C}^{-1}$ and integrator time constant, $I = 10$ seconds (i.e an integral gain constant of $\frac{k_p}{I} = 0.005\,\mathrm{kgs}^{-1}\ ^\circ\mathrm{C}^{-1}s^{-1}$.

Figure 4.13 shows the resulting control plots. The shaded region from 23°C to 25°C represents the comfort region as suggested by ASHRAE (ASHRAE, 1981). The setpoint of the controller is set at 24°C. During occupancy, this setpoint is reasonably maintained.



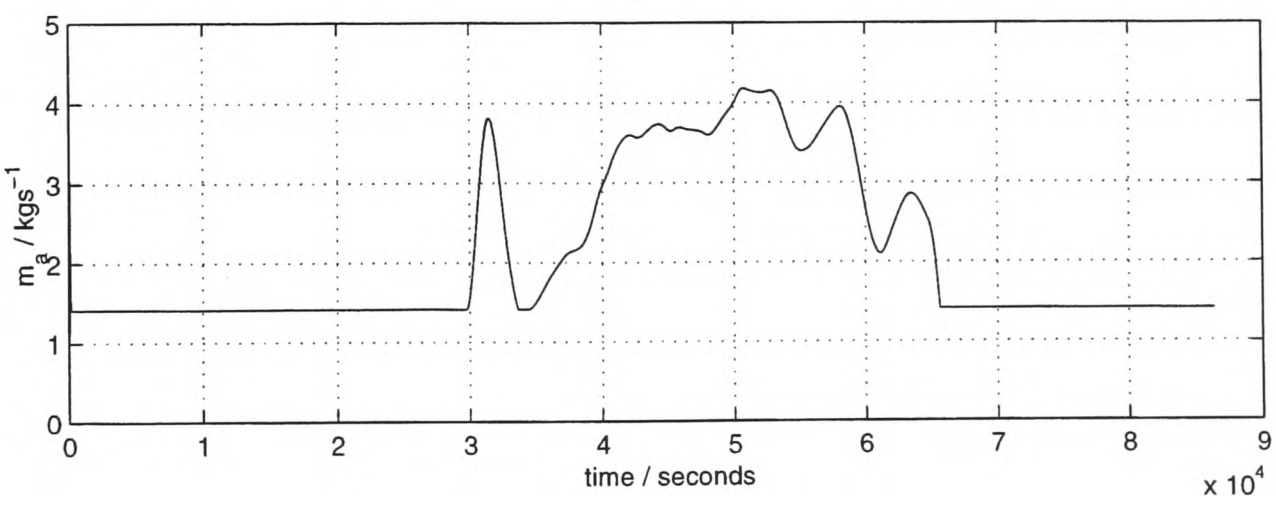


Figure 4.13: PI Control with $k_p = 0.05\,\mathrm{kgs}^{-1}\ ^{\circ}\mathrm{C}^{-1}$, I=10 s

4.7.2 NFMBP Control of Zone Temperature: Applications in Two Different Zones Control Modes

The application of the NFMBPC scheme to the zone comfort control problem is done in two different modes. The first mode is one in which the thermal comfort of the occupants of the zone is a priority, such as would be the case if the zone were being used as a conference room or an open-plan office. The NFMBPC scheme is therefore tuned with a small comfort tolerance (and, therefore, a small $width(G_c)$) of $\sigma_c = 1$, and with a relatively large energy tolerance (and correspondingly larger $width(G_e)$) of $\sigma_e = 10$.

In the second mode, the cost of energy is the major concern. This scenario could occur if the zone were employed as a school classroom, for example, where thermal comfort is not necessarily a great priority, but where conserving energy certainly might be. Here, the comfort tolerance is therefore relaxed, and the energy tolerance tightened: $\sigma_c = 2$ and $\sigma_e = 4$.

In each of these modes, the setpoint of the controller, w, is set to the *neutral temperature* of the zone which is defined as the temperature at which people are neither too hot nor too cold. For this zone, a neutral temperature of $w = 24°C$ is used (ASHRAE, 1981). The CMPD scheme (Section 3.2.4) is used. Figures 4.14 and 4.15 show the results of these simulations.

The *percentage control activity* of the control trajectory, $\mathbf{u}$, of the controller may be defined as:

$$\%\ Activity(\mathbf{u}) = \frac{\frac{1}{n-1}\sum_{j=2}^{n}|\Delta u(j)|}{u_{max} - u_{min}} \cdot 100 \tag{4.23}$$

where there are n elements in the set of control actions (air mass flow rate setpoints) defining the trajectory $\mathbf{u}$, and where $\Delta u(j) = u(j) - u(j-1)$. The values u_{max} and u_{min} are the maximum ($\dot{m}_{a,design}$) and minimum($\dot{m}_{a,min}$) air mass flow rates that the controller can set for the given system. Similarly, the *average percentage energy consumption* of the cooling load due to the control trajectory, $\mathbf{u}$, may be defined as:

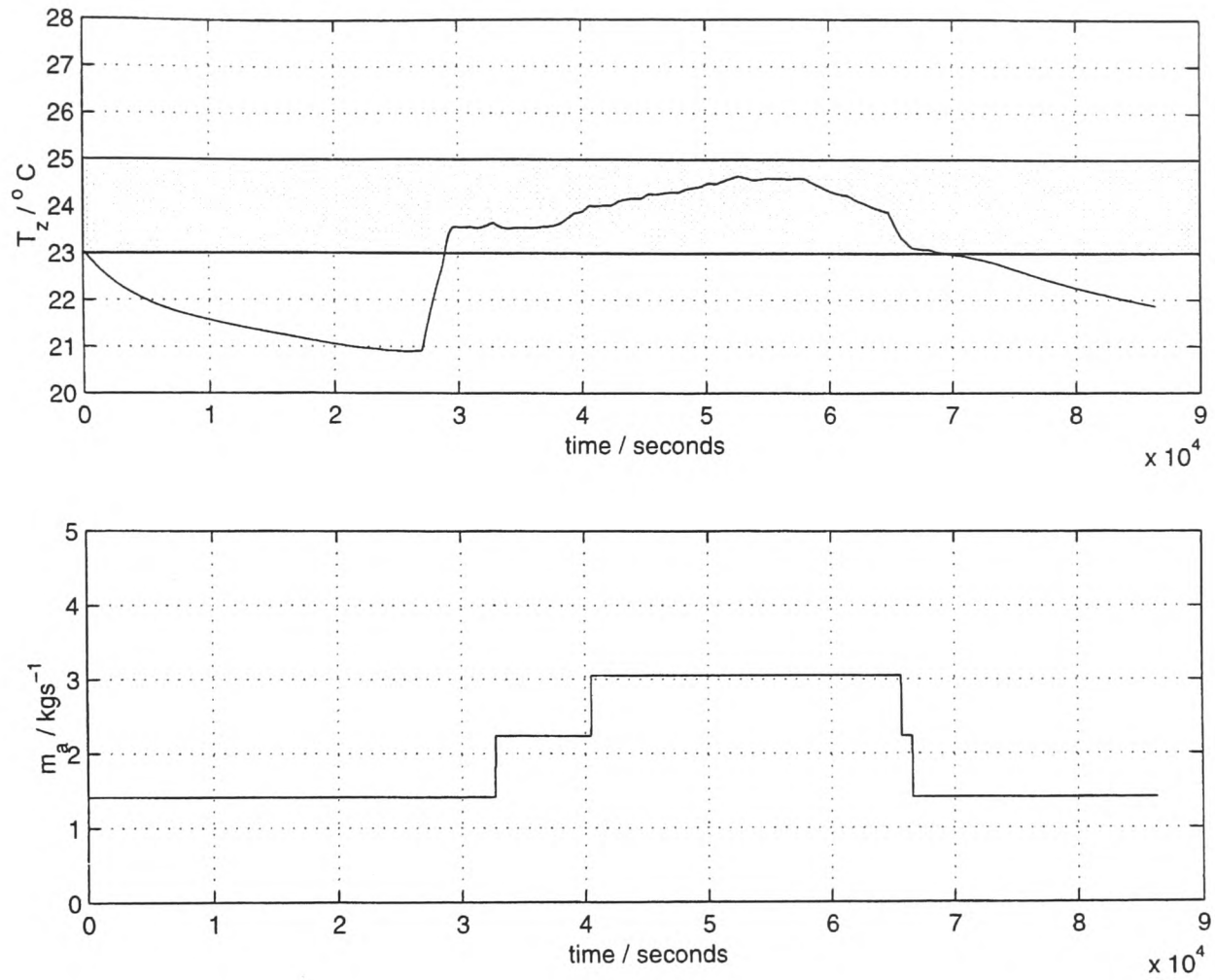


Figure 4.14: NFMBPC Scheme with $\sigma_c = 1$, $\sigma_e = 10$

$$\% \; Encon(\mathbf{u}) = \frac{\frac{1}{n}\sum_{j=1}^{n} u(j)}{u_{max}} \cdot 100 \tag{4.24}$$

This expression assumes that the control trajectory may be used as a proxy for the energy consumption of the system. Although in actuality the energy use of the system is more likely to be a function of the air mass flow rate of the supply air (the control trajectory), the specific heat capacity of the supply air, c_{air}, and the difference in temperature between the supply air and the zone air, for the near-steady-state scenario which characterizes most of the region within which the zone temperature is controlled, the assumption is justifiable for the constant value of c_{air} used.

Defined as in Equation 4.24, an air mass flow rate desired value equal to the design value, $u(j) = u_{max} = \dot{m}_{a,design}$ produces an average energy consumption of 100%. Table 4.3 shows the average energy consumption and percentage control activity for each of the three control schemes outlined above.

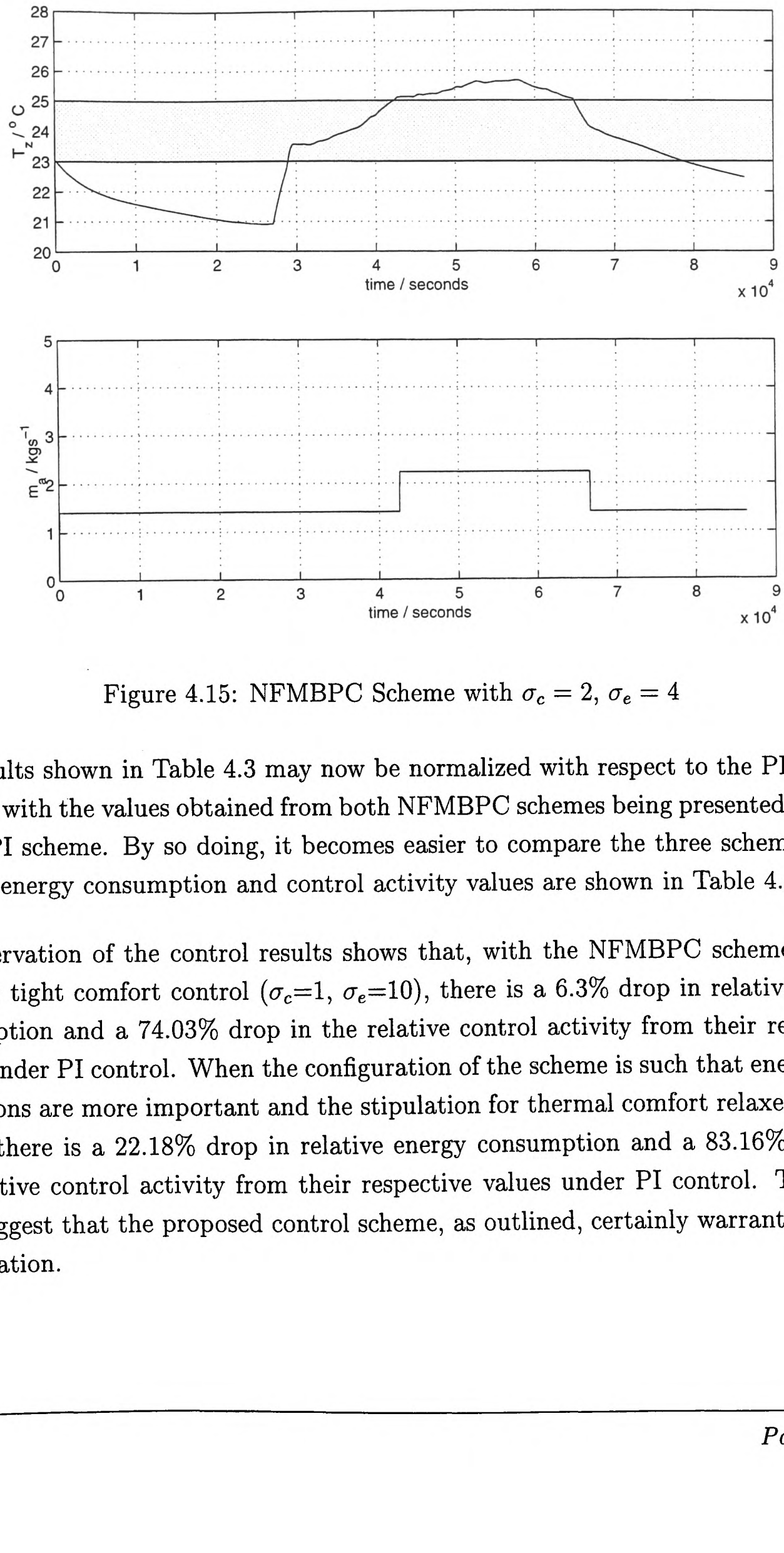


Figure 4.15: NFMBPC Scheme with $\sigma_c = 2$, $\sigma_e = 4$

The results shown in Table 4.3 may now be normalized with respect to the PI control scheme, with the values obtained from both NFMBPC schemes being presented relative to the PI scheme. By so doing, it becomes easier to compare the three schemes. The relative energy consumption and control activity values are shown in Table 4.4.

An observation of the control results shows that, with the NFMBPC scheme configured for tight comfort control (σ_c=1, σ_e=10), there is a 6.3% drop in relative energy consumption and a 74.03% drop in the relative control activity from their respective values under PI control. When the configuration of the scheme is such that energy considerations are more important and the stipulation for thermal comfort relaxed (σ_c=2, σ_e=4), there is a 22.18% drop in relative energy consumption and a 83.16% drop in the relative control activity from their respective values under PI control. These results suggest that the proposed control scheme, as outlined, certainly warrants further investigation.

Control Scheme	Avg. Energy Con./%	Control Activity/%
PI Control, k_p=0.05, I=10	44.78	0.1906
NFMBPC, σ_c=1, σ_e=10	41.96	0.0495
NFMBPC, σ_c=2, σ_e=4	34.85	0.0321

Table 4.3: Average Energy Consumption and Control Activity Associated with the Three Different Control Schemes

Control Scheme	Rel. Energy Con./%	Rel. Control Activity/%
PI Control, k_p=0.05, I=10	100	100
NFMBPC, σ_c=1, σ_e=10	93.70	25.97
NFMBPC, σ_c=2, σ_e=4	77.82	16.84

Table 4.4: Relative Energy Consumption and Control Activity Associated with the Three Different Control Schemes (PI Control = 100%)

4.8 Concluding Remarks

The application of the Neuro-Fuzzy Model-Based Predictive Control Scheme outlined in Chapter 2 to the building thermal comfort control problem has been presented in this chapter. Both the development of the neuro-fuzzy zone model and the application of the control technique to the TCC problem of the zone have been explored.

Firstly, a quick overview of the CSTB zone simulation used in this chapter to represent an actual zone has been given. For the purposes of the "apex training" desired for the application of the modified RSK algorithm used, and in an attempt to develop a less system-specific zone model, a simplified physical zone model was developed. This model characterized the general thermal behaviour of the zone under consideration, succinctly expressed in Equation 4.11.

From the simplified zone model, training data necessary to develop the fuzzy relation array central to the neuro-fuzzy model were gathered, and the neuro-fuzzy model identified using the modified RSK algorithm. The model which has five inputs and one output was designed with 25 fuzzy reference sets on its output, in an attempt at achieving a "truly fuzzy model". Prediction plots generated from the model have also

been presented.

With regards to the application of the control scheme to the TCC problem, since a 2-step-ahead neuro-fuzzy predictor has been used without defuzzification of the model output after the first step, an approach for deriving a fuzzy prediction from a fuzzy input has been outlined. The structure of the fuzzy goals specific to this particular application of the control scheme, as well as an example of the mechanics of maintaining achievable fuzzy goals have also been discussed.

The results presented in this chapter are encouraging. It has been shown that the comfort and energy tolerances of the NFMBPC scheme may be easily selected for the particular control problem under consideration. To achieve satisfactory PI control, the controller would have to be re-tuned for each different control problem. This is an indication of the flexibility of the proposed scheme over conventional PI control.

In addition, the results have also shown that there is a decrease in both energy consumption and control activity when using the NFMBPC scheme over the conventional PI control scheme, which suggests that the application of the proposed scheme to the control of complex systems where the control objective is uncertain, may be valid.

The following chapter examines the performance of the proposed scheme when both the complex system under control is real and the control objective uncertain.

Chapter 5

The Control of Supply Air Temperature using Generic Fuzzy Models

5.1 Introduction

In any given information-poor system, there may be more than one source of fuzziness, as discussed in Section 1.1.1. One source may be uncertainty in measuring certain variables of the system, while another may be uncertainty in how these variables relate to each other. In addition, if the system is to be controlled, yet a third source of uncertainty may be the determination of the control objective for this given system.

In Chapter 4 the neuro-fuzzy predictive control scheme developed in Chapter 2 is applied to the control of zone temperature in a simulated zone. Though the results are encouraging, the system used in the preceding chapter represents one in which only the control objective is fuzzy. Because it is a simulated system, there are no actual uncertainties associated with the behaviour of the system in an inherently fuzzy environment, since no such environment exists.

In this chapter, the scheme is applied to the control of the supply air temperature of an experimental air-handling unit present in the control laboratory in the Engineering Department at the University of Oxford. The Matlab code used in the implementation of the scheme is presented in Section C.1.1 of Appendix C.

Acceptable supply air temperature control can be difficult to achieve in practice due to

the uncertain dynamics of the specific cooling coil subsystem in the HVAC plant, and a lack of knowledge about the prevailing operating conditions. Even for control schemes formulated to accommodate these inherent uncertainties (for example, adaptive control or schemes based on gain scheduling techniques), the resulting control activity of the actuator is often high, giving rise to premature actuator wear.

In addition to an uncertain control objective, this application of the scheme represents a scenario where the system is real and is therefore also subjected to the uncertainties associated with the measurement and relationship of system variables.

5.2 Laboratory Test Rig Used in the Application of Supply Air Temperature Control Scheme

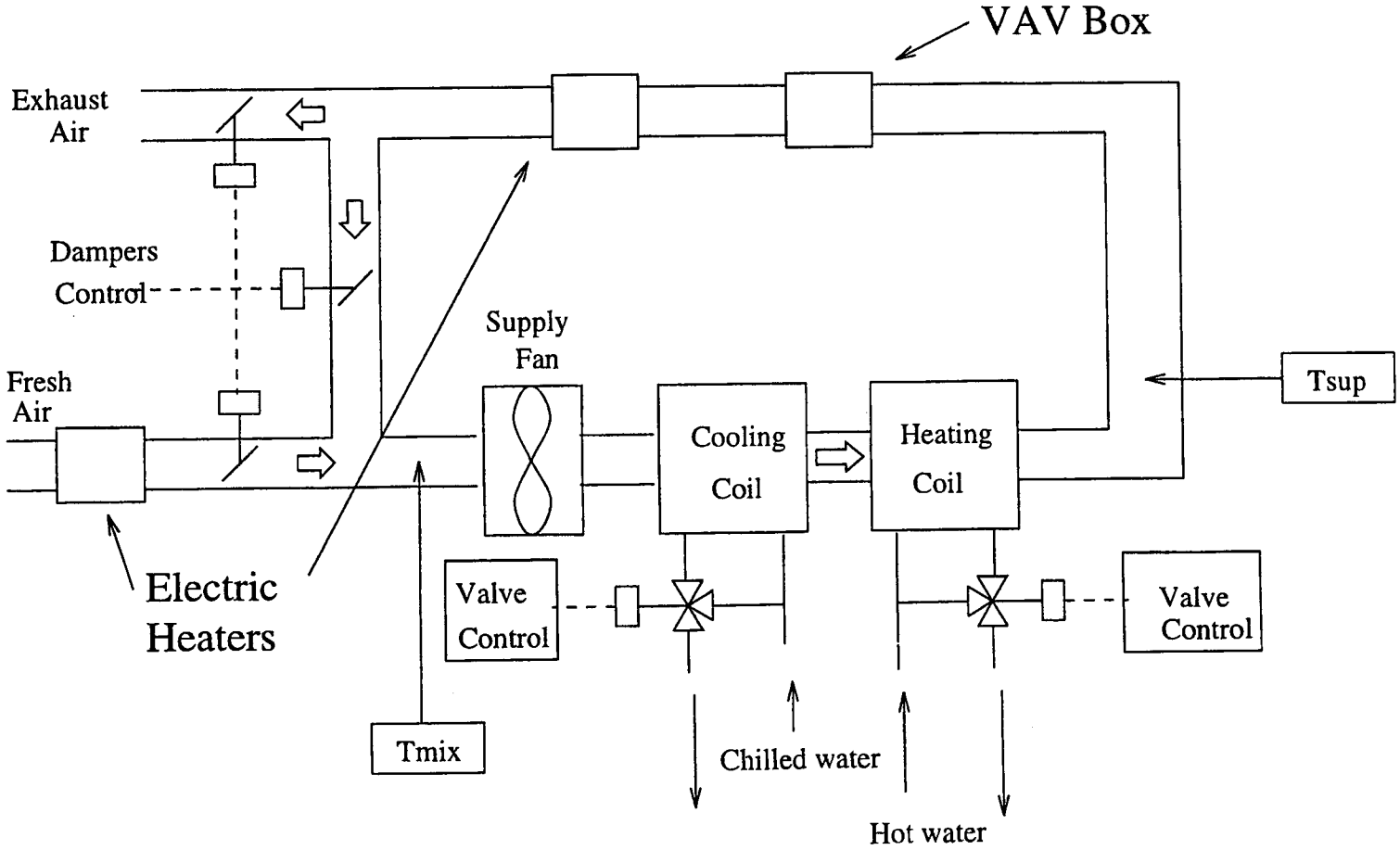


Figure 5.1: Schematic Diagram of the Laboratory Air-Handling Unit

Figure 5.1 is a schematic of the VAV test rig used in the application of the supply air

temperature control scheme. It consists of a fresh air intake duct along with an electric heater, a supply fan, a cooling coil, a heating coil, a VAV box and a second electric heater. An exhaust duct completes the air circuit.

Fresh air is admitted from outside and its temperature controlled at 25°C by means of a separate PI loop controlling the first electric heater. This is done to simulate the ambient air temperature of a typical summer day, and a heater is required since the experiments are performed in the winter months. Since the air is not humidified at this point, the cooling coil subsystem is essentially performing only sensible cooling as opposed to the more likely scenario of both sensible and latent cooling which would have prevailed on an actual day in summer.

The air is circulated around the circuit by the supply fan which is controlled to keep the supply air static pressure constant. The air is passed over the cooling coil, after which it is passed over the heating coil (which is always switched off and is therefore not used during these experiments). The temperature of the air after the heating coil is the supply air temperature, and, since the heating coil is not used, this temperature may be taken as being equal to the cooling coil outlet air temperature for the purposes of this work. The supply air passes through a second heater which simulates the internal heat gains of an occupied zone, the temperature of which will be determined by the supply air temperature. The air is now considered return air, a fraction of which may be re-circulated and mixed with fresh incoming air, and the remainder of which exits the system as exhaust air. The positions of the dampers controlling fresh air intake, the fraction of air re-circulated and the remaining fraction which exits the system, are fixed and remain so for the duration of the experiments. The exhaust-air damper is set to 100%, so that no air is re-circulated (return-air damper set to 0%). However, due to faults and imperfections on the test rig in the form of leakage, there is actually some return air at these settings, and its temperature is therefore controlled at 23°C by a separate control loop.

5.2.1 The Cooling Coil Subsystem

A schematic diagram of the cooling coil subsystem is shown in Figure 5.2, and consists of rows of copper tubes through which water, at temperature T_{cws} °C, is pumped. Inlet

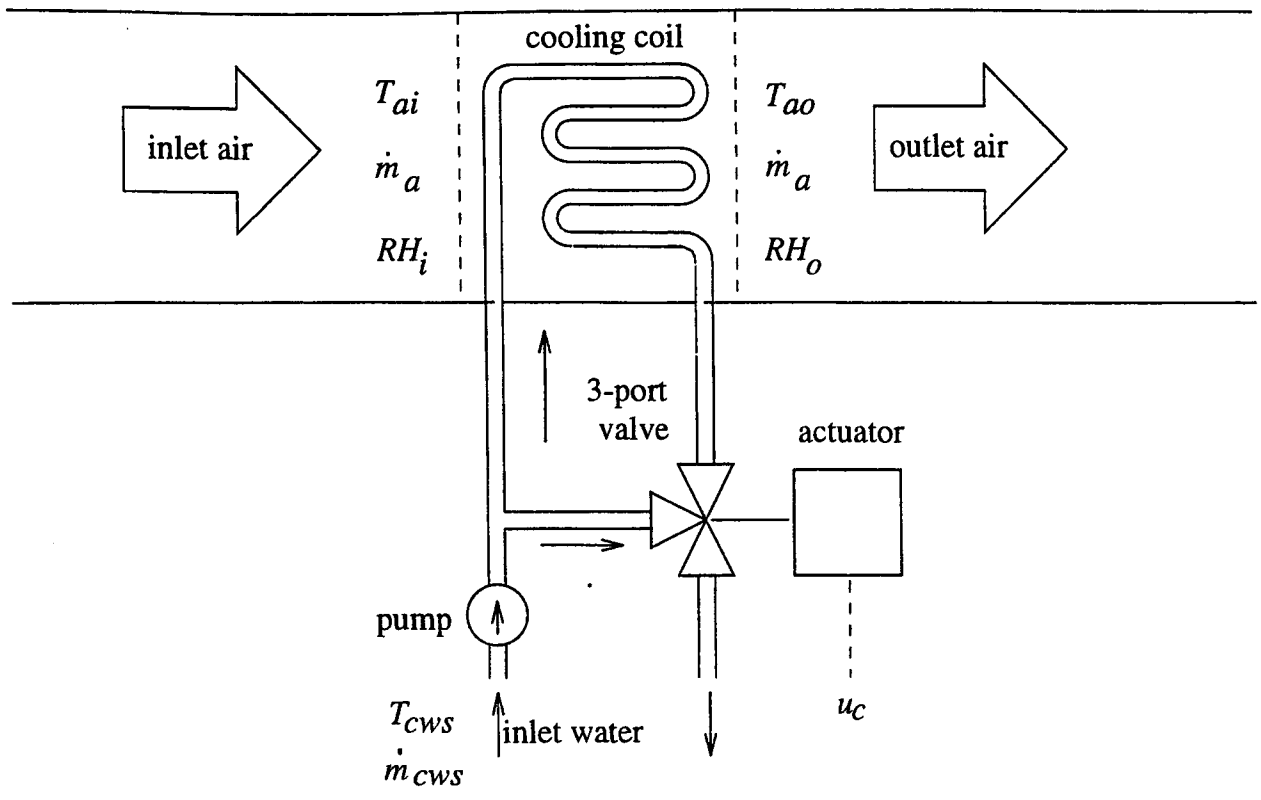


Figure 5.2: Schematic Diagram of the Cooling Coil Subsystem of the Test Rig

air at temperature T_{ai} °C, air mass flow rate, $\dot{m}_a$ kgs^{-1}, and inlet air relative humidity RH_i%, is passed over the cooling coil which is immersed in the air stream of the air to be cooled. Cooling takes place by way of the heat exchange between the warmer inlet air and the cooler chilled water flowing through the coil. The cooled outlet air has temperature T_{ao} °C and outlet air relative humidity RH_o%. Table 5.1 outlines the definition of the symbols used in the description of the cooling coil subsystem.

The actuator in this subsystem is controlled by a signal, $0 \leq u_c \leq 10V$, derived from the control scheme, and drives the stem of the three-port valve which regulates the chilled water supply mass flow rate through the coil, $\dot{m}_{cws}$. When $u_c = 0V$, the actuator causes the stem of the valve to stop the flow of chilled water through the coil, and no cooling of the inlet air takes place. Conversely, when $u_c = 10V$, there is maximum flow of chilled water through the coil $(\dot{m}_{cws,design})$.

5.3 Implementation of Control Scheme

The control scheme is implemented in Matlab which reads in values of the relevant system variables, computes the appropriate value of u_c, and outputs this computed

SYMBOL	DEFINITION
T_{ai}	inlet air temperature (^oC)
T_{ao}	outlet air temperature (^oC)
T_{cws}	chilled water supply temperature (^oC)
$\dot{m}_a$	air mass flow rate (kg/s)
$\dot{m}_{cws}$	chilled water supply mass flow rate (kg/s)
RH_i	inlet air relative humidity (%)
RH_o	outlet air relative humidity (%)
u_c	valve control signal (-)

Table 5.1: Variables Associated with the Cooling Coil Subsystem

value to the actuator. Since the system variables are analog but the algorithm is implemented on a digital computer, a 12-bit, -5V to 5V Analog-to-Digital Converter (ADC) and a 12-bit, 0V to 10V Digital-to-Analog Converter (DAC) are used in the implementation of the scheme.

The temperature sensors and actuator operate in the 0V to 10V range, and the signals representing the system variables are conditioned to ensure a high signal-to noise ratio, and that they fall within the specified ranges of the ADC and DAC. This conditioning entails appropriate amplification and filtering performed by dedicated circuitry.

5.4 Description of Experiments Carried out on Test Rig

The series of experiments carried out on the test rig are designed to get a comparison of the control performance of a conservatively tuned PI controller and the neuro-fuzzy predictive controller outlined in Chapter 2. These experiments are carried out under both **low air mass flow rate** and **high air mass flow rate** scenarios. In each case, inlet air is admitted through the fresh air damper and controlled at 25 °C. The air mass flow rate of the inlet air across the coil is then manually set to (and maintained at) either low (0.14 kgs^{-1}) or high (0.35 kgs^{-1}) depending on the particular experiment

being performed. The controller is switched on and attempts to control the supply air temperature at about 18 °C and 12.5 °C, respectively. Each experiment is run for 5400 seconds (1.5 hours).

5.5 PI Control of Supply Air Temperature

A PI controller is used as a benchmark for the control of the supply air temperature. Figure 5.3 is a Simulink diagram of the PI controller used in commissioning. The error between the supply air temperature, T_{sa}, and the supply air temperature setpoint, $w_{T_{sa}}$, is multiplied by a gain, G, and then added to the positive feedback of the output cooling coil control signal, u_c, fed through first-order dynamics, with time constant I. The output from this summation is passed through a limiter (to ensure that no constraints on the control action are violated), and through a memory device. This positive feedback form of the controller is given in (Clarke 1985).

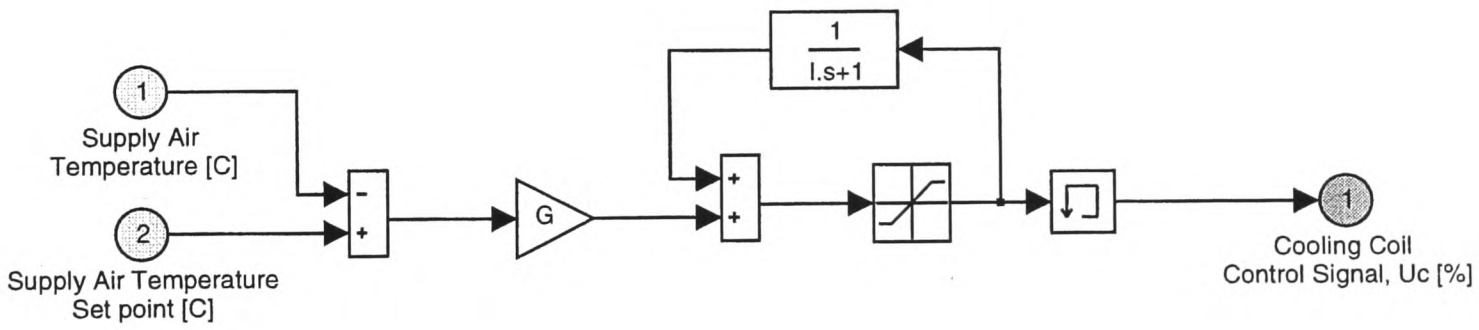


Figure 5.3: PI Controller for Supply Air Temperature Control

5.6 Modelling the Cooling Coil Subsystem

A neuro-fuzzy model of the cooling coil subsystem on the test rig is developed in an attempt to capture and replicate the general dynamics of the coil. The development of the model requires training to ensure that the general dynamics of the coil are accurately modelled. Real-time training of the model is not an attractive option, since this would require time-consuming random step tests covering all possible operating

conditions on the cooling coil. Since it is so difficult and time consuming to generate good training data from real-time training on the actual cooling coil under consideration, the behaviour of the cooling coil is simulated using a software package known as HVACSim+ which is used to furnish the training data. The package was developed to simulate the non-linear dynamic behaviour of HVAC systems (Park, Clark and Kelly, 1985). The coil neuro-fuzzy model is also realized in Matlab, which communicates with HVACSim+ via a Matlab interface present in HVACSim+.

Since the precise behaviour and the detailed design data of the cooling coil on the experimental air-handling unit are unknown, the coil's behaviour is modelled by use of a *generic* HVACSim+ model. The model developed is generic in the sense that it is designed so that the design parameters of the cooling coil on the rig would be a subset of the design parameters of the coil models used to construct the generic model. In other words, using any available *a priori* knowledge about the actual cooling coil on the rig, the HVACSim+ cooling coil models used in the construction of the generic model are selected such that their range of behaviour will be similar to, and will likely include, the behaviour of the cooling coil on the rig. If this is the case, it is assumed that the behaviour of the cooling coil under consideration will lie somewhere within the general behaviour of the generic model.

The generic model uses three different types of HVACSim+ cooling coil simulations (models): D1, D2 and D3. All three coils have a design air mass flow rate of 1.0 kg/s, and the design chilled water temperature for D1, D2 and D3 are 5.0, 7.0 and 9.0 °C respectively.

The generic model developed here, by nature of its design, is inherently fuzzy since it attempts to capture and faithfully replicate the general thermodynamic behaviour of a *class* of cooling coils. The 'amount of fuzziness' built into the model is dependent on the relationship between a number of coil model design parameters. More specifically, if:

- D1, D2 and D3 have similar time constants,
- no dehumidification of the inlet air occurs ($RH_i = RH_o$),
- for a given T_{ai} and $\frac{\dot{m}_a}{\dot{m}_{a,design}}$, the T_{ao} vs. u_c characteristic of the coils are similar, and

	Coil Design		
Design parameters	D1	D2	D3
No. of rows of tube	3	5	10
Length (finned sec.) in dir. of flow (m)	0.1143	0.1905	0.3810
Flow resistance on air side (m)	0.548441e-01	0.890056e-01	0.208625
Coil water flow resistance (0.001 kg.m)	31.1917	50.7668	98.0850
By-pass water flow resistance (0.001 kg.m)	31.1917	50.7668	98.0850
Valve capacity (m^3/hr @ 1 bar)	6.44525	5.05257	6.63497
Fin Spacing (m)	0.211667e-02	0.211667e-02	0.181429e-02

Table 5.2: Design Parameters for each Cooling Coil Design

- the control valves are sized so that their authority is 0.5,

then the generic model will be such that the class of coils it describes will be small (i.e., it will inherently have a small amount of fuzziness associated with its design). This is so since, in such a case, the parameters of each coil design included in the generic model are not very different from the parameters of the other coil designs in the model.

The design parameters and detailed coil data for D1, D2 and D3 are shown on Tables 5.2 and 5.3 respectively.

5.7 Developing the Generic Neuro-Fuzzy Model of the Cooling Coil

The basic approach to the development of the generic neuro-fuzzy model involves combining nine different fuzzy relational models identified from simulations on the three coil designs as shown in Figure 5.4.

Training and identification is done via a series of simulations as shown in Figure 5.5. This process produces nine different fuzzy relational arrays (neuro-fuzzy models), $R_1 \ldots R_9$; one for each combination of coil design and simulated air mass flow rate used during training. Air mass flow rate values of 40%, 70% and 100% of design value are used

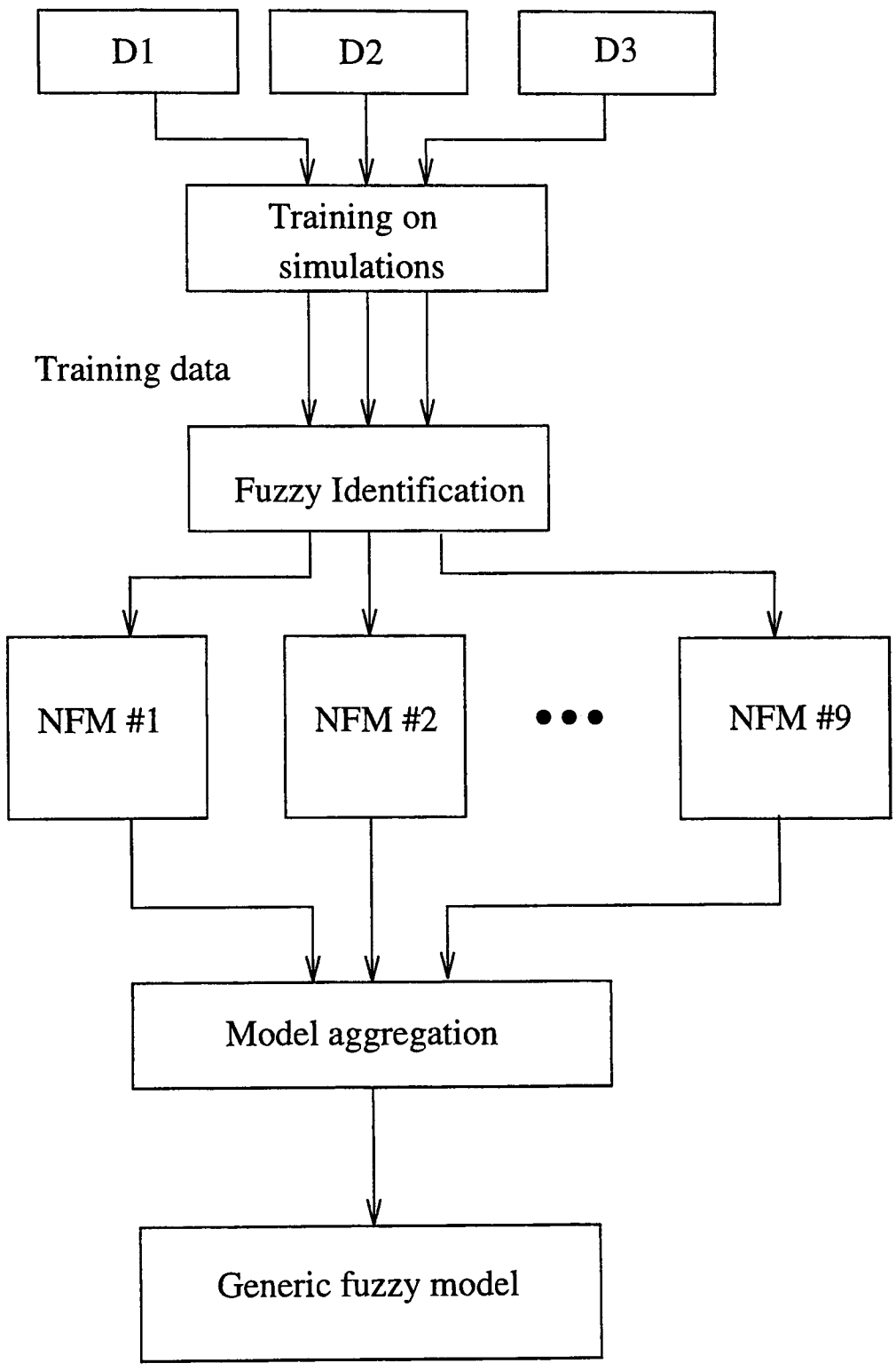


Figure 5.4: Generating Generic Fuzzy Model of the Cooling Coil

PARAMETER	VALUE
Number of tubes per row	16.00
Number of parallel water circuits	4.00
Height of finned section (m)	0.6090
Width of finned section (m)	0.6100
Tube outside diameter (m)	0.1270e-01
Tube wall thickness (m)	0.4300e-03
Tube material	Copper
Fin thickness (m)	0.2400e-03
Fin material	Aluminum
Valve type	equal % (flow) / linear (bypass)
Valve curvature parameter (0 = linear)	20
Valve rangability	100
Valve leakage (fractional flow)	0.1000e-03

Table 5.3: Detailed Coil Data for the Cooling Coil Subsystems of D1, D2 and D3

to get a more robust model covering flows from minimum to maximum value. The generic model is computed as the maximum over the nine fuzzy relational models, $R_{generic} = max(R_1, R_2, \ldots, R_9)$.

5.7.1 Structure of the Neuro-Fuzzy Model

The structure of the neuro-fuzzy model used in this section of the work is shown in Figure 5.6. Since the model is generic, its inputs are normalized, and it produces a normalized output. These normalized values are as follows:

- normalized cooling coil actuator control signal, $u = \frac{u_{actual}}{100}$
- normalized air mass flow rate, $\dot{m}_a = \frac{\dot{m}_{a,actual} - \dot{m}_{a,min}}{\dot{m}_{a,max} - \dot{m}_{a,min}}$
- normalized outlet air temperature, $T_{ao} = \frac{T_{ao,actual} - T_{ao,min}}{T_{ao,max} - T_{ao,min}}$

SIMULATED AIR MASS FLOW RATE USED DURING TRAINING

COIL DESIGN	0.4 kg/s	0.7 kg/s	1.0 kg/s
D1	Simulation # 1	Simulation # 2	Simulation # 3
D2	Simulation # 4	Simulation # 5	Simulation # 6
D3	Simulation # 7	Simulation # 8	Simulation # 9

Figure 5.5: Training on D1, D2 and D3

The model has three inputs and one output given by:

$$\hat{T}_{ao}(n+1) = T_{ao}(n) \circ u(n) \circ \dot{m}_a(n) \circ R \tag{5.1}$$

where $\hat{T}_{ao}(n+1)$ and $T_{ao}(n)$ are the predicted and current values of normalized outlet air (supply air) temperature respectively, $\dot{m}_a(n)$ is the current value of normalized air mass flow rate through the pressure-independent VAV terminal box, $u(n)$ is the currently applied normalized cooling coil actuator control signal, $\circ$ is the fuzzy composition operator, and R is the fuzzy relational array.

For each experiment, the value of the air mass flow rate across the coil changes very slowly, and can therefore be regarded as being constant for the duration of the experiment. Therefore, even though the air mass flow rate across the coil is an input to the neuro-fuzzy model, it is assumed to have negligible dynamics during the experiment.

The controller is designed for an outlet air temperature range of 7 to 30 °C (0 to 1 on the normalized universe of discourse). The model has eleven linearly spaced

triangular fuzzy reference sets with 50% overlap which describe T_{ao}. Eleven sets are chosen because, since it is desired that the supply air temperature be controllable over a 23°C range (from 7°C to 30°C), if a 10% uncertainty in measuring the supply air temperature is assumed, the actual supply air temperature could be anywhere in the range ±2.3°C of the measured zone temperature. Eleven sets over a range of 23°C models this 10% uncertainty in our system.

Five fuzzy reference sets are defined on the normalized current actuator control signal, $u(n)$, three fuzzy reference sets on the normalized current air mass flow rate, $\dot{m}_a(n)$, and twenty-six fuzzy reference sets are defined on the normalized predicted outlet air temperature, $\hat{T}_{ao}(n+1)$. The fuzzy reference sets are shown in Figure 5.7.

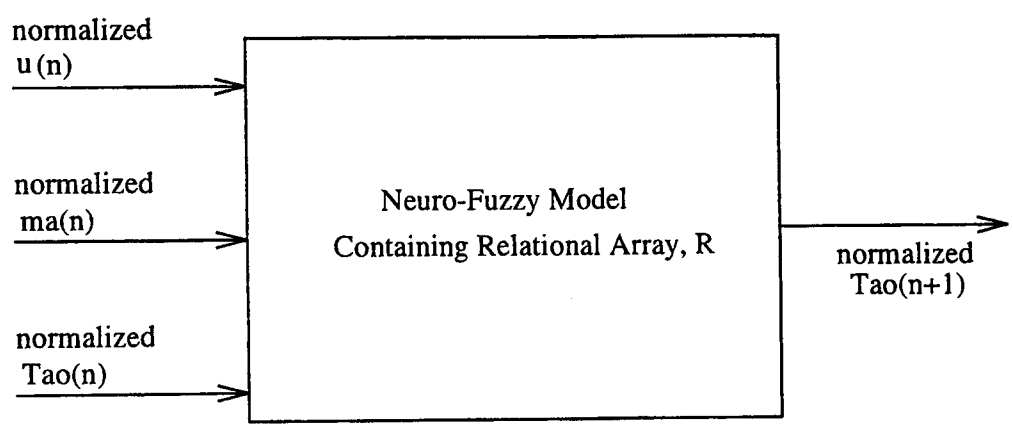


Figure 5.6: Neuro-Fuzzy Model of Supply Air Temperature

5.7.2 Training and Identification of the Neuro-Fuzzy Model using Generic HVACSim+ Data

The modified RSK identification scheme described in Section B.3 of Appendix B is used to generate the fuzzy relational array, R. The scheme requires training at the apex of each input fuzzy reference set, as mentioned in Section 4.3.1. In an attempt to achieve the necessary 100% activation of each combination of input sets, a random training scheme with data filtering is employed.

Training on each HVACSim+ coil model simulation is achieved as follows: First, the value of the inlet air temperature is set to, and held constant at 25 °C (to simulate the

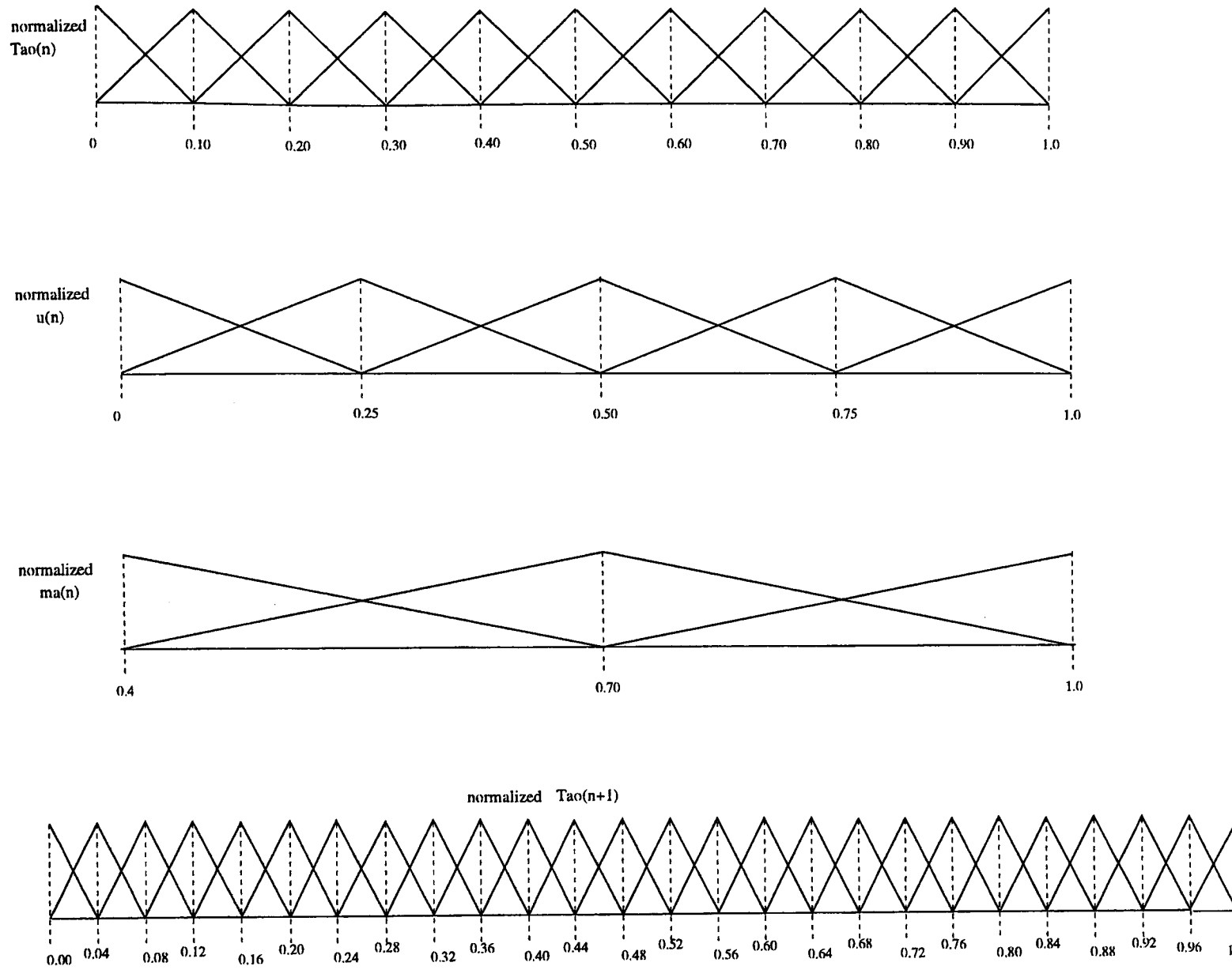


Figure 5.7: Fuzzy Reference Sets defined on the Normalized Input and Output Spaces

ambient temperature on a typical summer day). Next, the value of the air mass flow rate across the coil is set to, and held constant at, 0.4 kgs^{-1}.

Then, for the case where the total time delay on the control action = 1 sample (i.e. the inherent time delay), a fuzzy relational model is identified by applying a pseudo-random sequence of values of normalized control action, $\mathbf{u}(n)$, to the HVACSim+ coil model under consideration, and observing the corresponding sequence of values of the normalized outlet air temperature one sample period later, $\mathbf{T_{ao}}(n+1)$.

Since eleven fuzzy reference sets are used to describe the universe of discourse of the current value of outlet air temperature, $T_{ao}(n)$, and five fuzzy reference sets are used to describe the universe of discourse of the current value of control action, $u(n)$, there are a total of 55 different combinations of inputs in each simulation (no dynamic training

takes place on $\dot{m}_a$). For each of the 55 combination of inputs, the data points of interest are the values of $T_{ao}(n)$ and $u(n)$ which maximally activate their particular respective fuzzy reference set (since an activation of 100% can not be guaranteed for all fuzzy sets, what is needed is an activation as close to 100% as possible).

Those values of $\mathbf{u}(n)$ **AND** $\mathbf{T_{ao}}(n)$ which maximally activate the respective fuzzy sets defined on them are stored. Together with the corresponding values of $\mathbf{T_{ao}}(n+1)$, these values form a triplet representing maximum activation of a particular combination of inputs and the resulting output. It is on this triplet of values, $g^* = (u(n), T_{ao}(n), T_{ao}(n+1))$, that the modified RSK algorithm is used in the identification of the fuzzy relational model (See Appendix B). To generate the remaining eight fuzzy relational models, the process is repeated for an air mass flow rate of 0.7 kgs^{-1}, then 1.0 kgs^{-1}, before being applied to the remaining two coil designs (D2 and D3). The maximum is taken over all nine relational arrays to produce the generic fuzzy relational model (Section 5.7). Training is done for a simulation run time of 301500 seconds with 30-second sample intervals. This creates 10050 samples to observe during training.

Post-processing of the resultant fuzzy relational array is necessary because of 'holes' which are present in the training data. These holes exist where there is not a smooth piece-wise linear relationship between the model inputs and output interpolated from the training data, because the training data does not sufficiently cover all operating regions during training. Though the particular combinations of inputs and output that characterize the operating regions corresponding to these holes may indeed be possible during the normal course of operation of the rig, they are unlikely to occur.

Post-processing is done on the possibility values of the predicted output as shown in Figure 5.8, and involves converting a non-convex fuzzy set into a convex fuzzy set (See Figure 1.5 in Section 1.5.2). Figure 5.8 shows two plots of the predicted outlet air temperature derived from the generic cooling coil model, when the current outlet air temperature is 19 °C, the air mass flow rate across the coil is 0.4 kg/s, and when the current control signal is 25%. The plot on the left shows the unprocessed model output. The resultant holes in the predicted outlet air temperature plot are visible since the plot is not monotonically increasing (non-convex). The plot on the right is the post-processed output, and is one possible convex set equivalent to the plot on the left. The post processing Matlab code is given in Section C.1.4 of Appendix C.

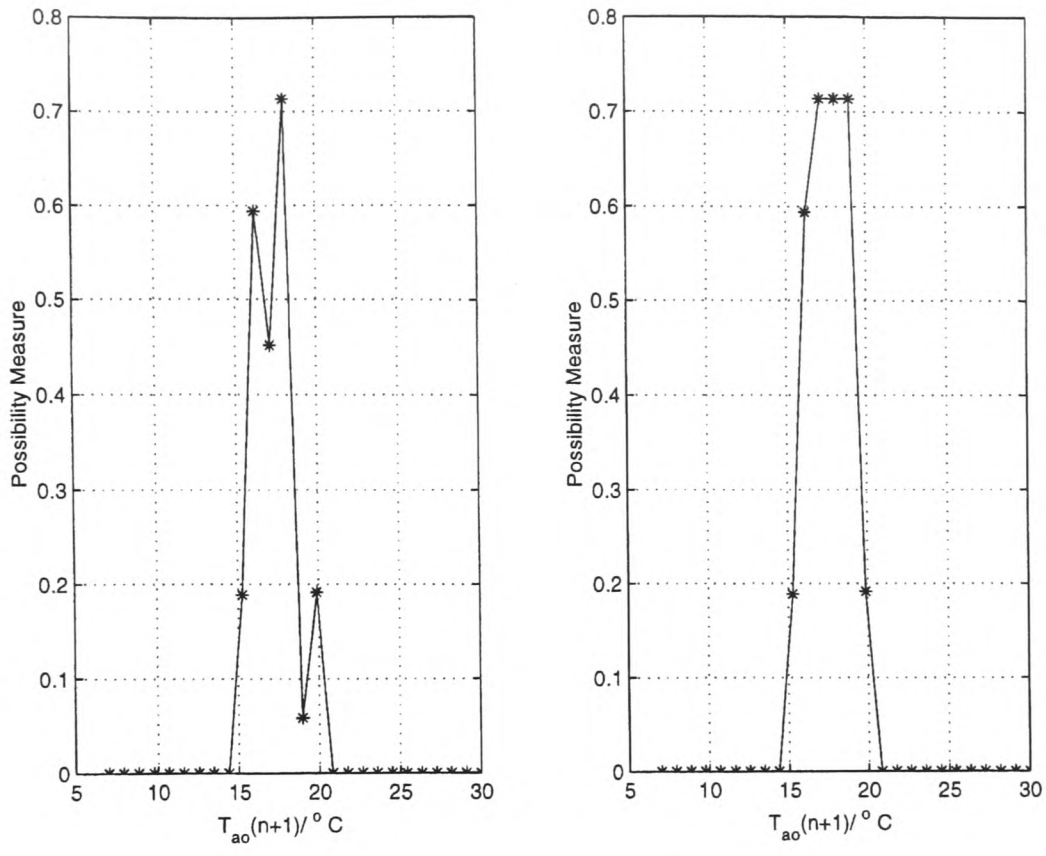


Figure 5.8: Post-processing of Fuzzy Predicted Outlet Air Temperature

A section of the pseudo-random sequence of control actions used in training, $\mathbf{u}(n)$, is shown in Figure 5.9, and is generated by passing uniformly distributed random numbers on the interval $(0.0, 1.0)$ through a second order digital filter. Matlab's rand function is used to generate the necessary random values, and the digital filter is given by:

$$\begin{aligned} p(n) &= e^{-t/\tau} \cdot p(n-1) + (1 - e^{-t/\tau}) \cdot (\mathsf{rand} - 0.5) \\ q(n) &= e^{-t/\tau} \cdot q(n-1) + k \cdot (1 - e^{-t/\tau}) \cdot p(n) \\ u(n) &= q(n) + 0.5, \qquad\qquad u(n) \in (0, 1) \end{aligned} \tag{5.2}$$

where n is the sample number, t is the sampling time (30 seconds), τ, the filter time constant, is of the same order as the time constant of the coil (estimated as 60 seconds), k is the filter gain and rand is a Matlab function which generates uniformly distributed random numbers in the interval $(0.0, 1.0)$. The filter gain must be large enough so that the control action adequately covers the interval $(0.0, 1.0)$, but small enough to prevent over-saturation. A value of $k = 7.5$ is used.

Filtering is done to increase the efficiency of the training process. Since the generic

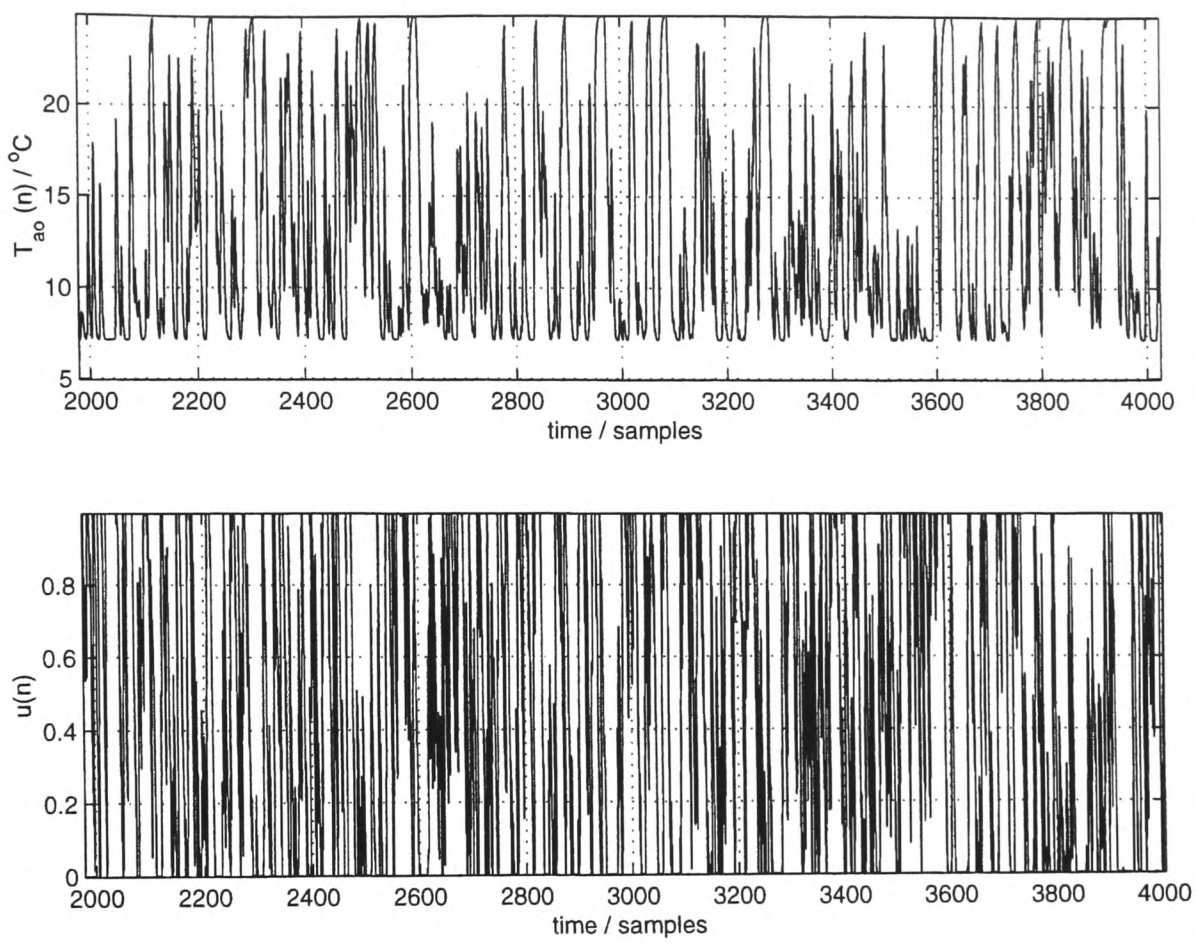


Figure 5.9: Pseudo-Random Control Sequence used and Corresponding Outlet Air Temperature observed during Training

cooling coil model will itself low pass filter the control action, only lower frequencies are relevant in the training process. Filtering reduces training time which would otherwise have been much longer since training would then have included the processing of superfluous high frequencies. Starting with a current inlet air temperature of 25 °C, and with an air mass flow rate of 0.4 kg/s, Figure 5.9 shows a section of the random sequence of control actions generated during training (from the 2000^{th} to the 4000^{th} sample), and the corresponding values of T_{ao} (outlet air temperature) observed.

The random training Matlab code is presented in Section C.1.5 of Appendix C. For the case where the minimum air mass flow rate of 0.4 kg/s is employed in training on coil design $D1$ (i.e. design air mass flow rate of 1.0 kg/s and design chilled water temperature of 5 °C), the values of maximum activation for the 55 combinations of fuzzy sets are presented in Table 5.4.

The lack of activation of the reference sets that occur at the normalized outlet air temperature of 0.90 across all normalized control actions, and at 1.00 at normalized

	Normalized u(n)				
Normalized Tao(n)	**0.00**	**0.25**	**0.50**	**0.75**	**1.00**
0.00	83.9%	93.8%	94.3%	93.2%	94.6%
0.10	99.9%	90.9%	97.4%	97.1%	99.8%
0.20	99.4%	93.3%	98.8%	96.5%	100.0%
0.30	99.9%	94.3%	96.6%	96.5%	99.3%
0.40	99.9%	94.3%	91.4%	93.8%	98.1%
0.50	100.0%	96.8%	92.0%	92.3%	99.0%
0.60	99.6%	93.0%	96.5%	96.0%	90.0%
0.70	99.9%	97.7%	95.6%	95.5	77.7%
0.80	82.6%	78.4%	74.6%	70.1%	26.6%
0.90	0.0%	0.0%	0.0%	0.0%	0.0%
1.00	0.0%	0.0%	87.6%	12.4%	0.0%

Table 5.4: Maximum Percentage Activation of Fuzzy Input Reference Sets by Training Data: Training at 0.4 kg/s on Coil Design D1

control actions of 0.00, 0.25 and 1.00 are a result of a lack of adequate training data in these regions. Neuro-fuzzy model predictions in these regions would benefit from post-processing of the model output, as described above.

5.7.3 The Fuzzy Goal Function

There is one fuzzy goal function, G, in this application of the control scheme. It is normalized since the fuzzy decision-maker compares it to normalized values of fuzzy proximities representing the fuzzy cost functions. The goal is trapezoidal and defined on the interval [0 1]. Figure 5.10 shows G with a particular value of $\sigma_T = 4$ as an example.

The universe of discourse of G is described on a *normalized absolute temperature difference* space. If the temperature difference between any two crisp temperatures is denoted ΔT, then the absolute value is $|\Delta T|$, and the normalized absolute value is given by:

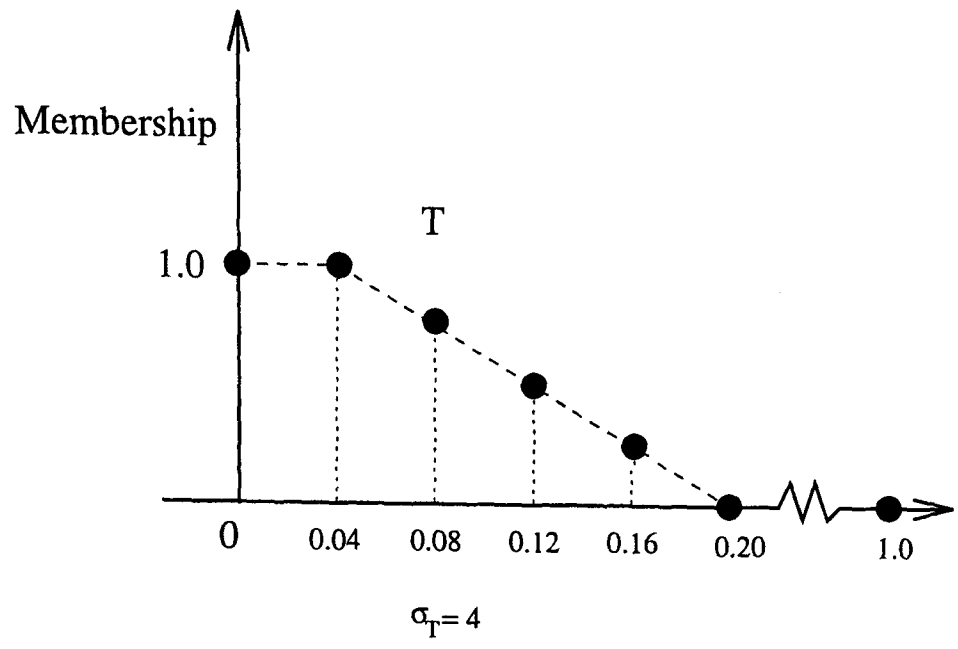


Figure 5.10: An Example of the Fuzzy Goal Function Defined on Supply Air Temperature

$$|\Delta T|_{normalized} = \frac{|\Delta T|_{actual} - |\Delta T|_{min}}{|\Delta T|_{max} - |\Delta T|_{min}} \tag{5.3}$$

With respect to G, since the minimum and maximum outlet air temperatures, T_{ao}, accommodated are 7 °C (280.15 K) and 30 °C (303.15 K) respectively, $|\Delta T|_{min} = 0$ K, and $|\Delta T|_{max} = 23$ K.

The universe of discourse of the normalized goal, $UOD(G)_{normalized}$ is therefore given by:

$$UOD(G)_{normalized} = \frac{UOD(G)_{unnormalized} - 0}{23} \tag{5.4}$$

This therefore means that the smallest change in $UOD(G)_{normalized}$ as determined from Figure 5.10, $\Delta UOD(G)_{normalized}$=0.04, corresponds to an unnormalized goal difference of 0.04 x 23 $\approx$ 0.9 K (the core of the goal, as defined in Section 2.5) .

The fuzzy cost function computed here, $J_{sd}(u)$, is a measure of normalized setpoint deviation (the proximity of the normalized predicted outlet air temperature to the normalized setpoint). This fuzzy cost of setpoint deviation, $J_{sd}(u)$, is compared to G and, *for the case where* $\sigma_T = 1$, the implication here is that we are willing to accept any

normalized fuzzy cost of setpoint deviation, $J_{sd}(u) = (w(n+1) \leftrightarrow \hat{T}_{oa}(n+1))$, in the interval $[0, 0.04)$, but those in the interval $[0.04, 0.08]$ would also be deemed acceptable, though to a linearly decreasing extent.

In an unnormalized sense, the geometry of G implies that we are willing to accept any setpoint deviation up to about 0.9 K, but deviations up to $0.04(1 + \sigma_T)$ x 23 K will be tolerated to a linearly decreasing extent. For σ_T=1, this tolerance translates to a temperature difference of about 1.8 K.

5.7.4 Fuzzy Predicted Temperatures from the Generic Model

To gain a better appreciation of the fuzzy predicted temperatures generated by the neuro-fuzzy model, some typical results are presented in this section.

Figure 5.11 is the fuzzy (one-step-ahead) predicted temperature generated from the neuro-fuzzy model when the current air mass flow rate across the coil, $\dot{m}_a(n)$, is set to the minimum value of 40% of the coil design value, or 0.4 kg/s, and when the current outlet air temperature, $T_{ao}(n) = 14$ °C. Notice, firstly, that there is fuzziness or uncertainty associated with any given predicted temperature, which is characterized by a fuzzy set. Secondly, as expected, as the value of the control action increases from 0% to 100% (i.e., from no cooling to maximum cooling), the value of the predicted temperature, though still uncertain, decreases. The small circle on the $u(n) - T_{ao}(n+1)$ plane indicates the *crisp* value of predicted outlet air temperature that would have been generated by the neuro-fuzzy model, had *traditional height defuzzification* taken place.

Figure 5.12 is a plot of the fuzzy (one-step-ahead) predicted temperature when the current air mass flow rate across the coil, $\dot{m}_a(n)$, is now set to the coil design value of 1.0 kg/s (the current outlet air temperature, $T_{ao}(n)$, remains 14 °C). Notice that, even though uncertainty about the predicted temperatures persists, the drop in temperature associated with the increase in control action is less than is the case when the air mass flow rate is at 40% of the design value.

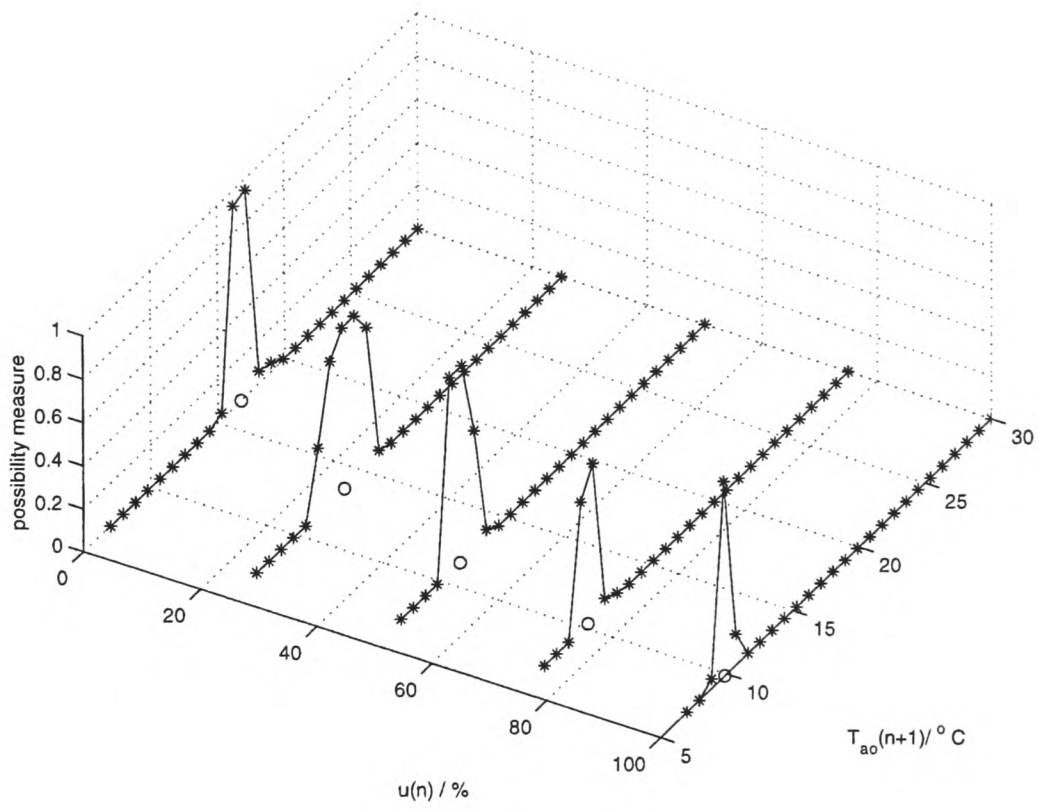


Figure 5.11: Fuzzy Predicted Temperature when $\dot{m}_a(n) = 0.4$ kg/s and $T_{ao}(n) = 14$ °C

5.8 Results of Experiments on the Test Rig

5.8.1 Intermediate Experimental Results: Justification for the Inclusion of Normalized Air Mass Flow Rate in Model

Initially, the generic model was developed to exclude the current normalized air mass flow rate, $\dot{m}_a(n)$, as an input, in an attempt at developing a more generic model.

The structure of the model attempted is given by:

$$\hat{T}_{ao}(n+1) = T_{ao}(n) \circ u(n) \circ R \tag{5.5}$$

where the variables take on their usual meanings as stated in Section 5.7.1.

Figures 5.13 and 5.14 are the results obtained from the application of the NFMBPC scheme when normalized air mass flow rate is **not** included in the model structure. Figure 5.13 is the High Flow Rate plot of supply air temperature and control action,

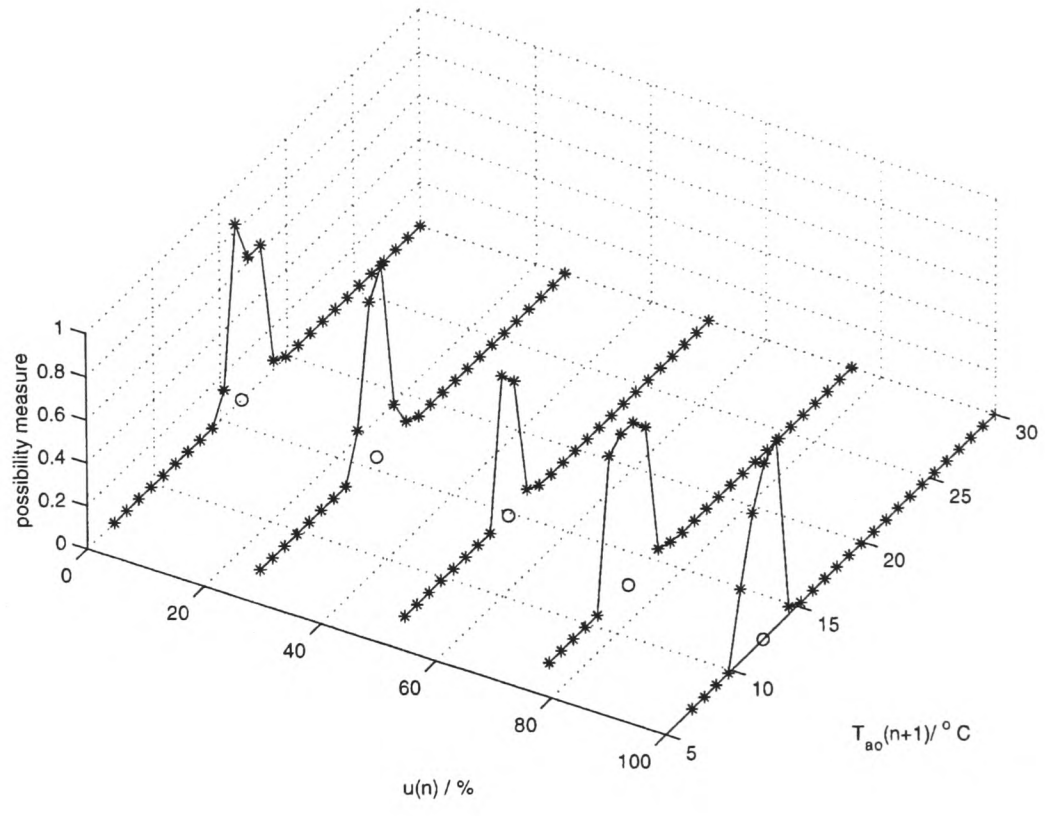


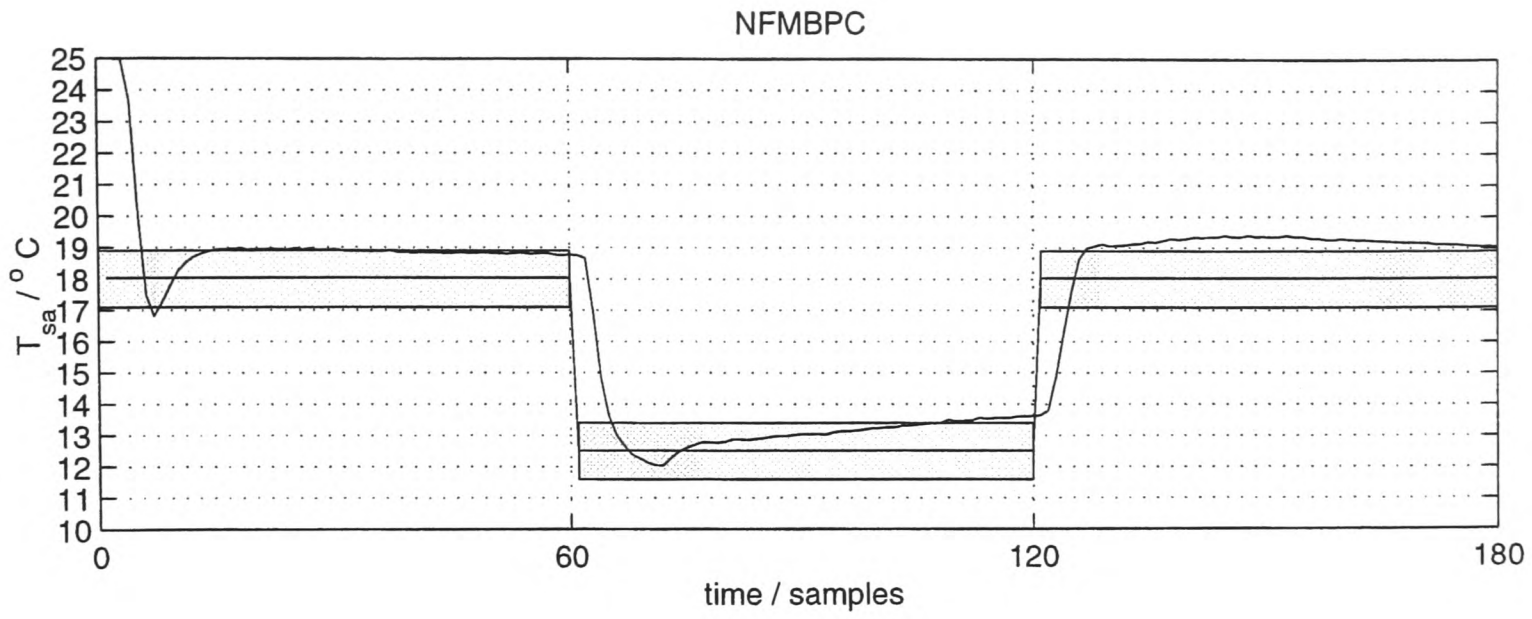
Figure 5.12: Fuzzy Predicted Temperature when $\dot{m}_a(n) = 1.0$ kg/s and $T_{ao}(n) = 14$ °C

while Figure 5.14 is the Low Flow Rate plot.

As observed from the figures, the performance of the controller is unacceptable at both flow rates, but especially under the more difficult low air flow rate scenario. This is because the exclusion of air mass flow rate renders a predictive model that is too generic for the purposes of this work. Therefore, although the controller is not very active, its performance is compromised. Similar intermediate experimental results led to the conclusion that the normalized air mass flow rate needed to be included as an input to the neuro-fuzzy model. The final model used is therefore the one presented in Equation 5.1.

5.8.2 Final Experimental Results

Figures 5.15, 5.16, 5.17 and 5.18 show results from tests performed on the rig. The shaded region represents the area in which the supply air temperature is deemed equally acceptable. Figures 5.15 and 5.16 are **high flow rate** plots of the supply air temperature and control action of both the PI controller and the neuro-fuzzy controller. In



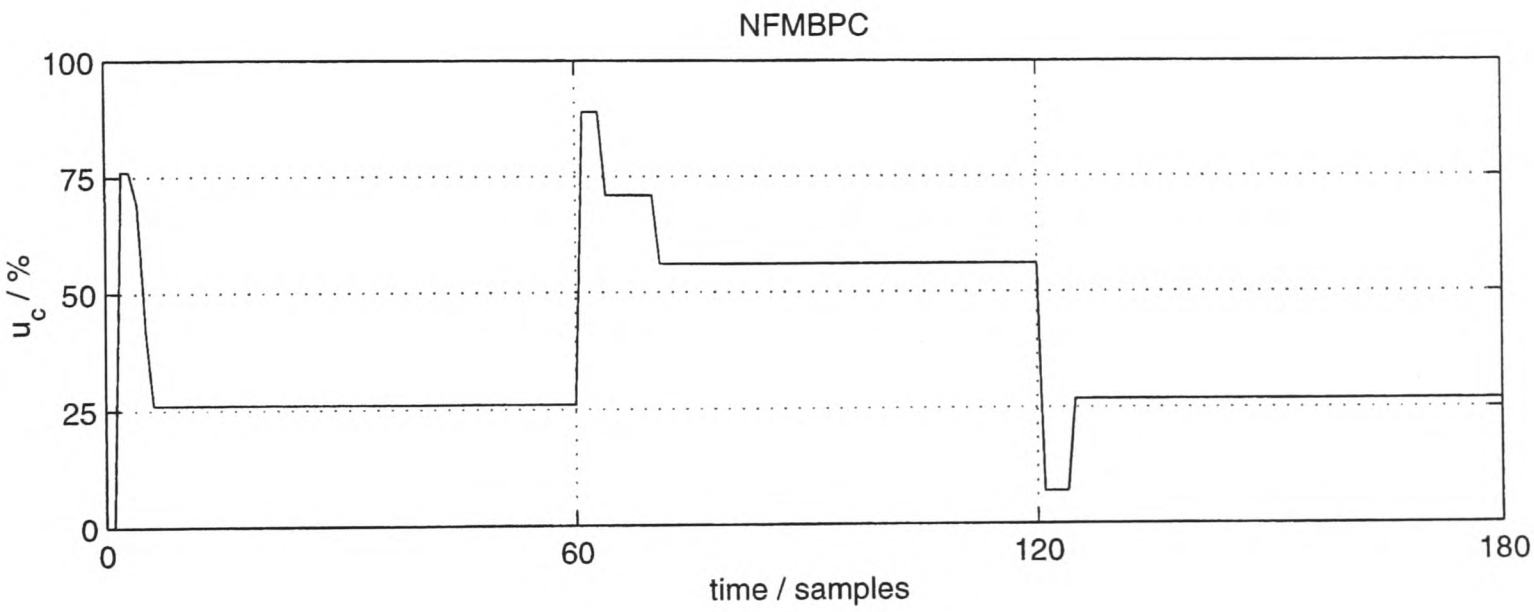
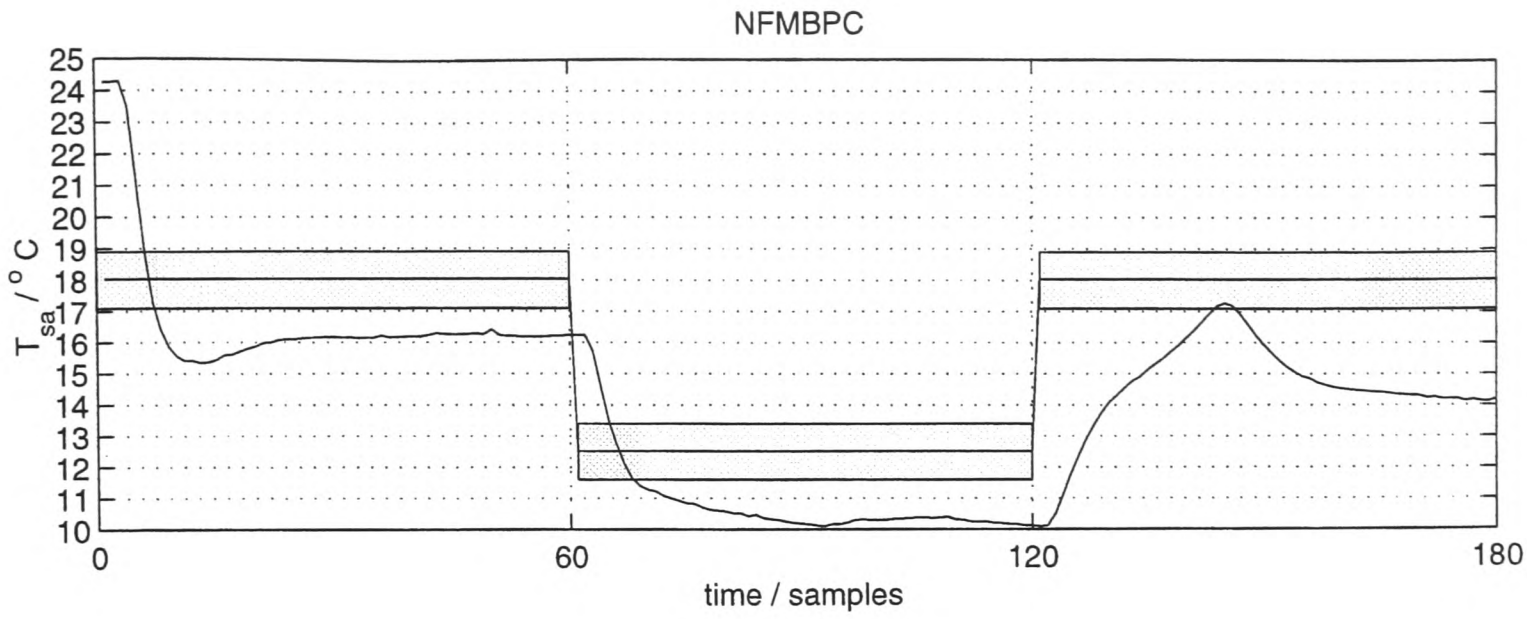


Figure 5.13: Intermediate Results: Supply Air Temperature and Control Action Plots at **High Air Flow Rate.**

contrast, Figures 5.17 and 5.18 are **low air flow rate** plots of the supply air temperature and control action for both controllers.

The PI controller is tuned with a proportional gain, $k_p = 5.0\ ^\circ\mathrm{C}^{-1}$ and integral time constant, $I = 60$ seconds (i.e an integral gain constant of $\frac{k_p}{I} = 0.083\ ^\circ\mathrm{C}^{-1}s^{-1}$). The neuro-fuzzy model-based predictive controller has parameters $\alpha = 0.8$, $\sigma_T = 1$ ($core = 0.9\ ^\circ\mathrm{C}$), and uses the Conditional Linear-Interpolation Defuzzification Scheme (CLID Scheme) outlined in Section 3.2.7. As explained earlier, under uncertainty (such as might be the case if there were hysterisis in the control valve, for example), conditional defuzzification schemes may prove more robust than conventional defuzzification



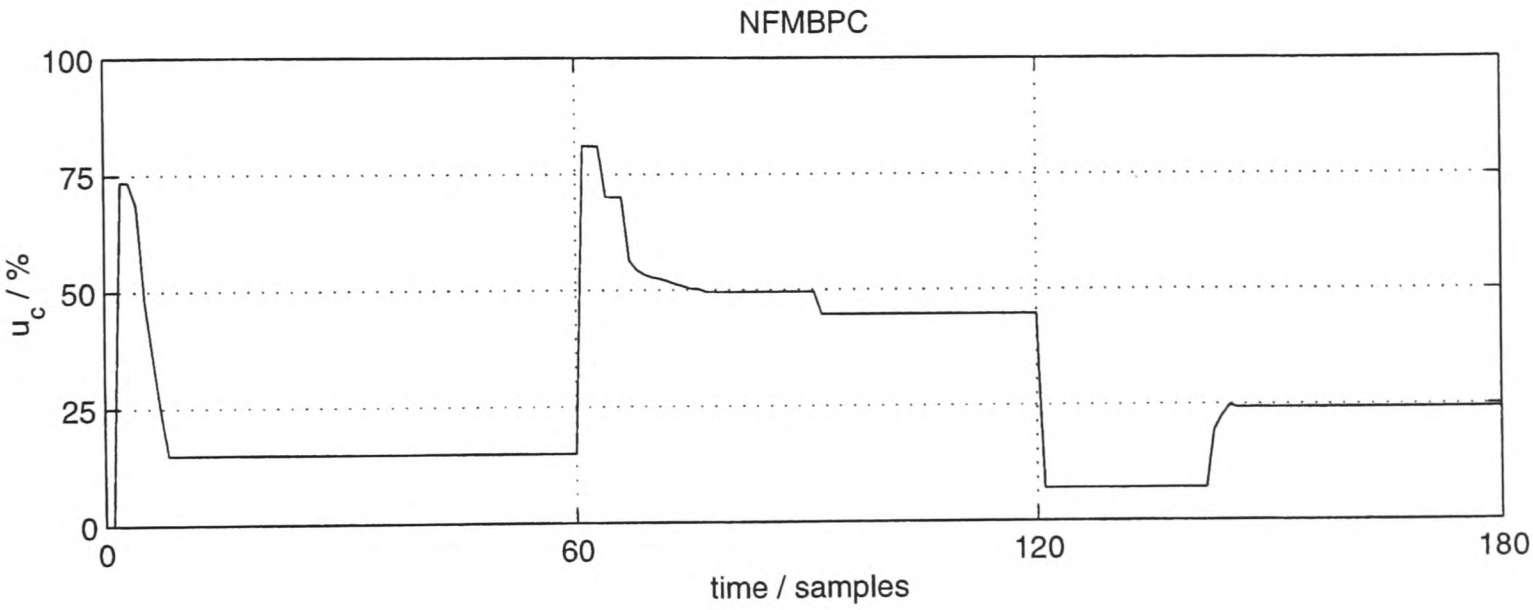
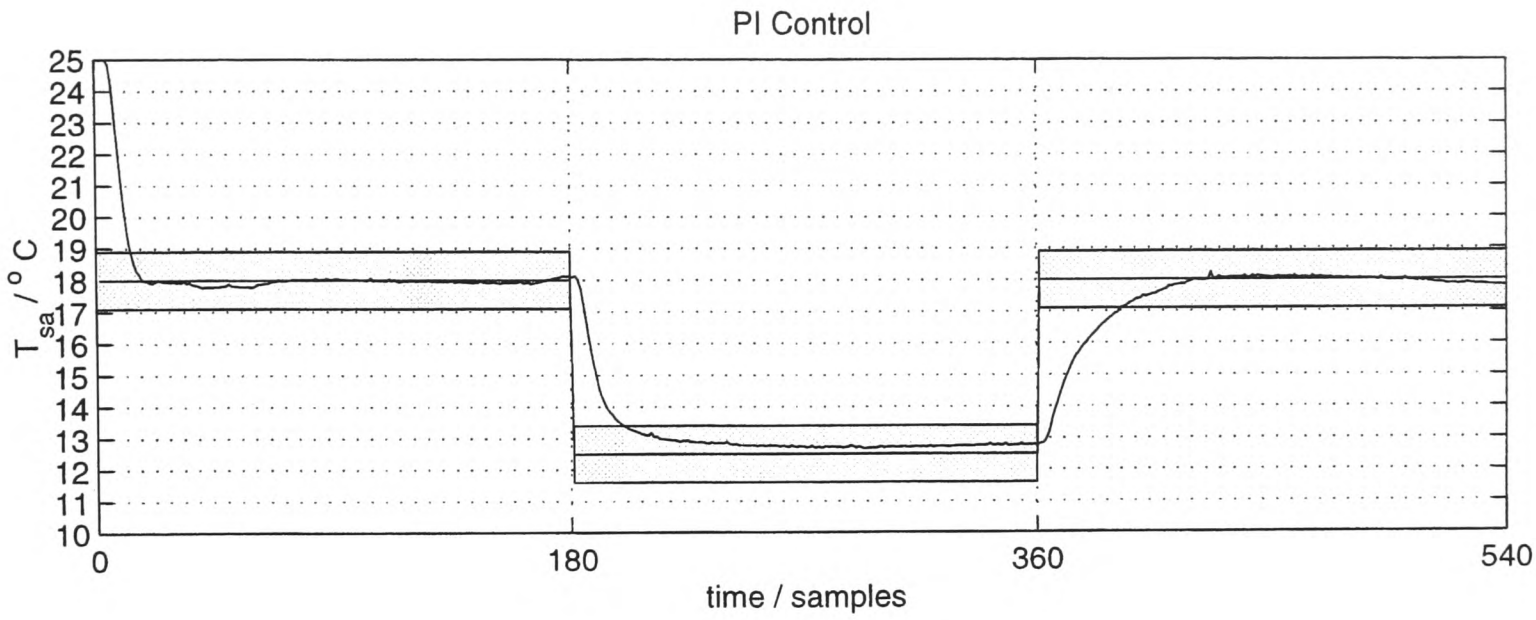


Figure 5.14: Intermediate Results: Supply Air Temperature and Control Action Plots at **Low Air Flow Rate.**

schemes.

Table 5.5 summarizes the relative control activity observed for each scheme, under both low and high flow rate scenarios. Since the low air flow rate PI control scheme was observed to have the most actuator activity, it is assigned a value of 100% and the remaining three scenarios are normalized with respect to it. In addition, for convenience, the table also shows the reduction in actuator activity obtained from using the NFMBPC scheme as opposed to the PI controller.

Observing Figure 5.15, it should first be understood that, due to the inherent nature of the design of the NFMBPC scheme, the steady-state performance of the PI controller



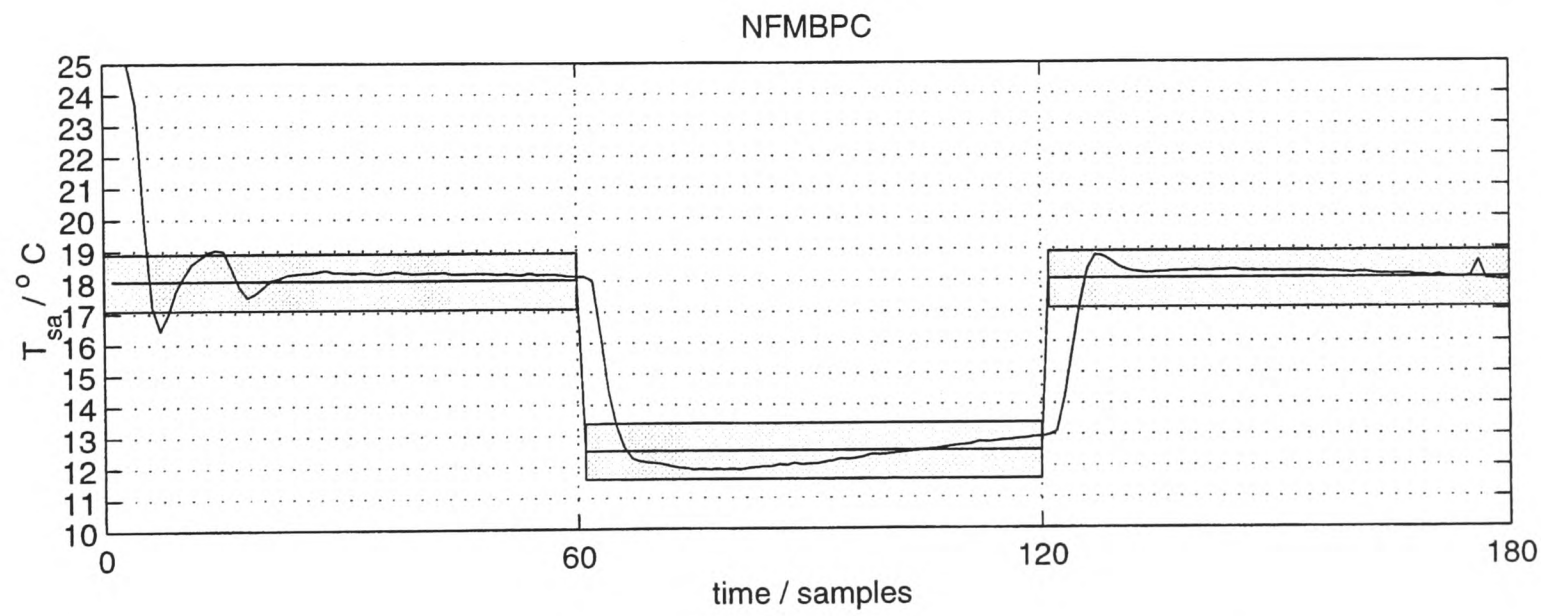


Figure 5.15: Supply Air Temperature Plots: PI Control vs. NFMBP Control of Supply Air Temperature at **High Air Flow Rate**.

is expected to be superior to the steady-state performance of the NFMBPC scheme. This is because the PI control scheme has an integrator directly incorporated into the control loop. Notice, however, that, while this steady-state performance is superior in the PI scheme, the transient response is sluggish compared to the NFMBPC scheme. In addition, since the control objective is defined such that we are equally willing to tolerate a ± 0.9 °C drift in steady-state temperature, the performance of the NFMBPC scheme is superior when the control activities of each scheme, as depicted in Figure 5.16, are considered. Considering both figures together, it is seen that the NFMBPC scheme results in less actuator activity for a similar steady-state performance.

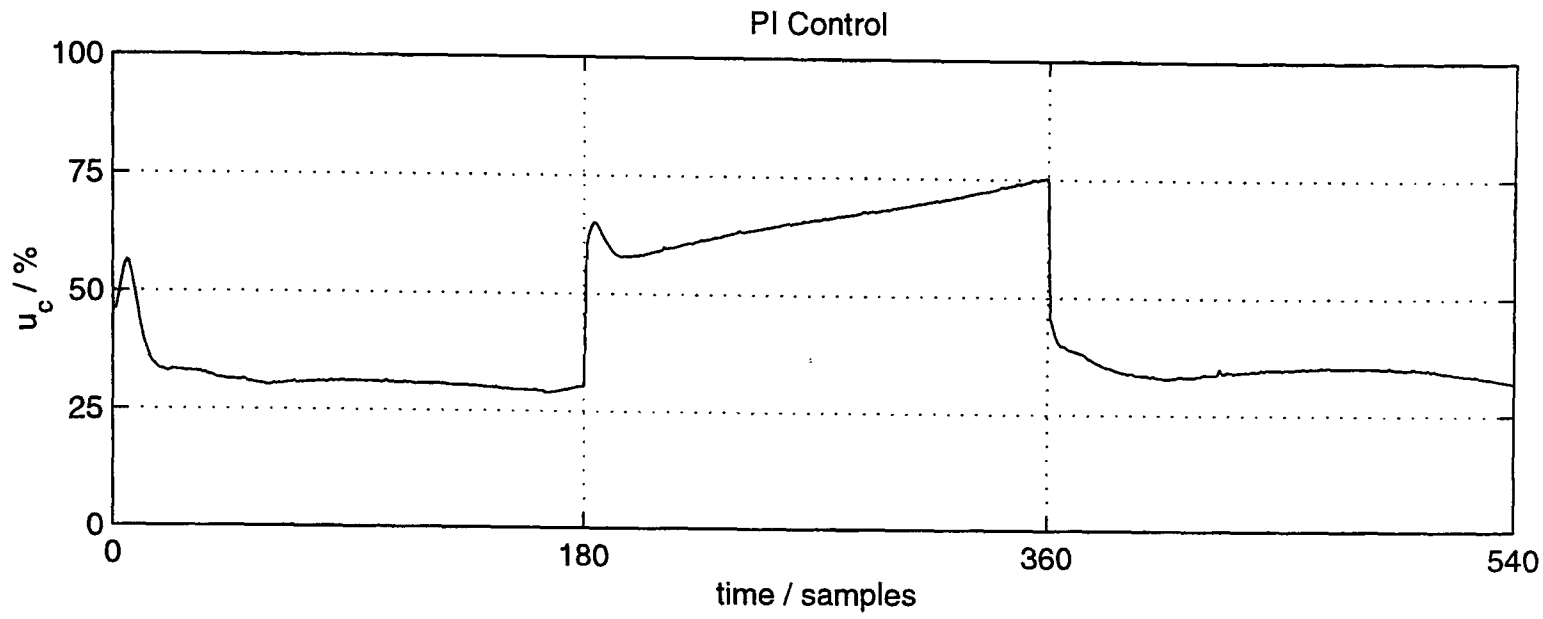


Figure 5.16: Control Activity Plots: PI Control vs. NFMBP Control of Supply Air Temperature at **High Air Flow Rate.**

The figures also show that the NFMBPC scheme has superior robustness handling capabilities in the presence of uncertainty when compared to the PI control scheme, an observation also made in (Gouda, Danaher and Underwood, 2000) where PID and Fuzzy Logic control are compared. During the experiments, since the chilled water supply temperature, T_{cws}, is not controlled, the cycling of the chiller causes T_{cws} to increase in value. This increase in chilled water temperature, though not significant enough to cause an appreciable degradation in controller performance, nonetheless causes the actuator to open the control valve in the case of the PI controller (in Figure 5.16, u_c changes from about 58% at the 200^{th} sample to about 75% at the 360^{th} sample). However, the NFMBPC scheme allows the outlet air temperature to drift slightly upwards

	PI Control/%	**NFMBPC/%**	**Reduction in Activity/%**
Low Flow	100	29.30	70.70
High Flow	61.60	40.40	21.20

Table 5.5: Relative Control Activity Associated with Both Control Schemes

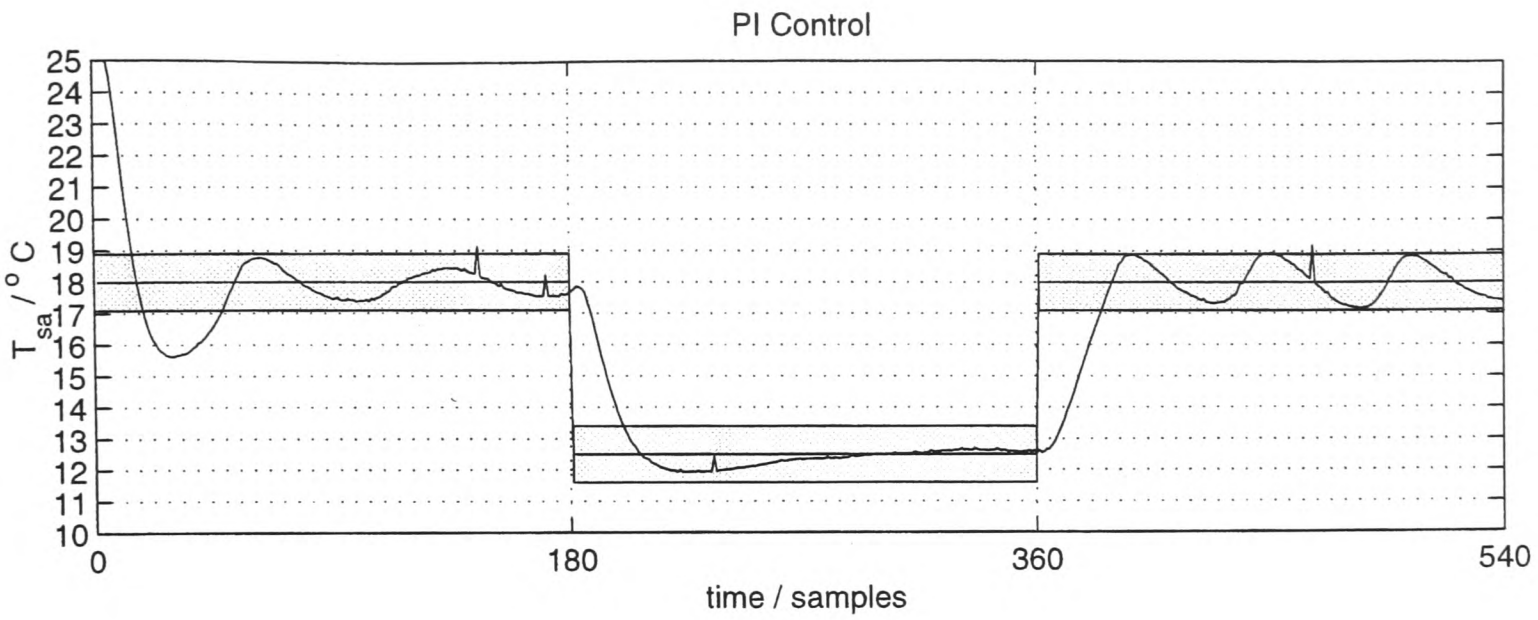
due to the warmer chilled water temperature, since the scheme recognizes that the supply air temperature is still within acceptable limits and therefore cannot justify any movement in the control valve position.

At low air mass flow rates, controlling the supply air temperature becomes more difficult due to the higher system gains under these conditions. Therefore, relatively small changes in control action have a correspondingly large effect on the supply air temperature.

Figures 5.17 and 5.18 compare the steady-state and transient performance of both schemes while controlling the supply air temperature at a low air mass flow rate. The spikes observed on the supply air temperature and control action plots of the PI control scheme are thought to be due to an imperfection in a region of the control valve assembly corresponding to the lower control actions which result at higher gains. These spikes are absent from the NFMBPC scheme since the control actuator is not as active in this scheme, and therefore does not enter this region for any appreciable length of time during the normal operation of the controller.

The most obvious observation from the plots is that the PI controller causes the control valve to oscillate with a resultant oscillation in the supply air temperature around the setpoint. While the oscillations could most likely be eliminated by re-tuning the PI controller, in practice it is still very much a matter of trial-and error that would produce acceptable results. These figures show that, using *the same* generic NFMBP controller, the control valve is again less active when compared to the PI control scheme. In addition, except during the transient period caused by the step in setpoint, the supply air temperature is controlled to within the desired control limits, as stipulated by the control objective. The fact that this is achieved with less control activity suggests that the NFMBPC scheme also outperforms the PI controller at low air mass flow rates.

With regards to the transient behaviour of both schemes, in the case of the NFMBPC



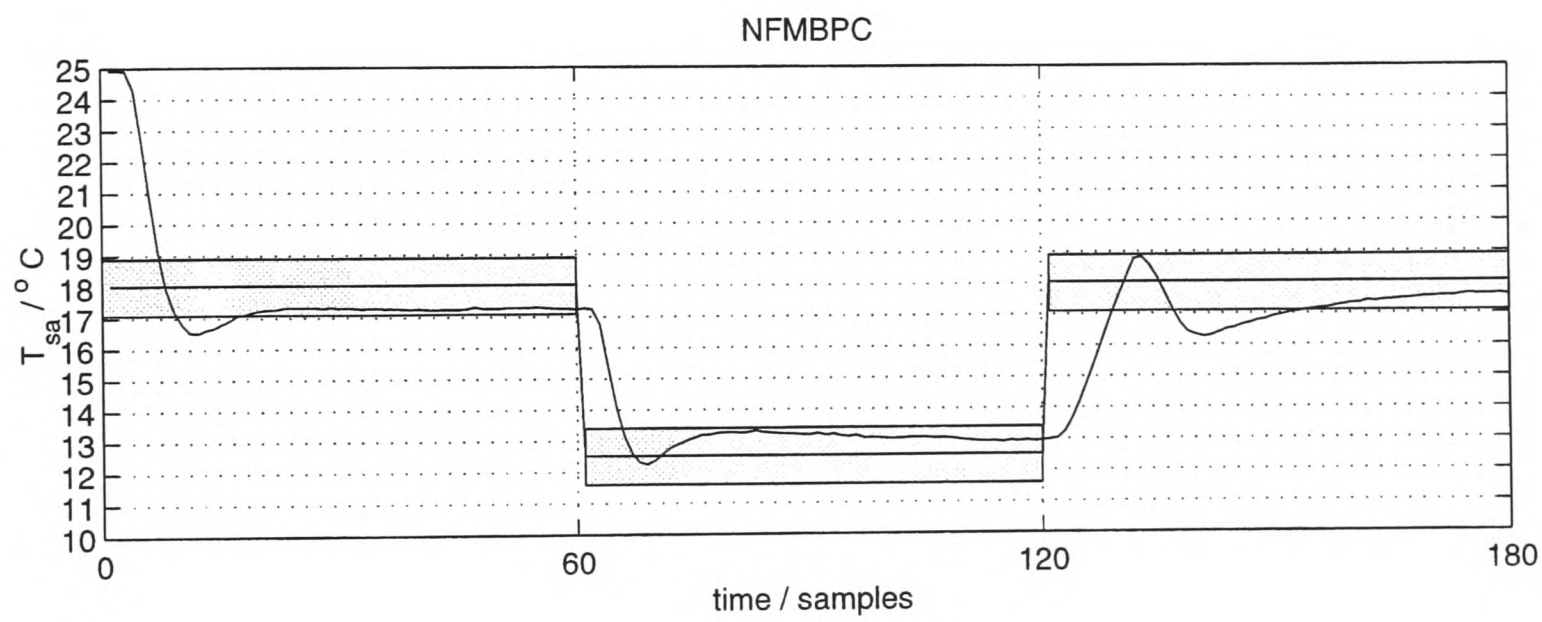


Figure 5.17: Supply Air Temperature Plots: PI Control vs. NFMBP Control of Supply Air Temperature at **Low Air Flow Rate**.

scheme, the transient response during a step down in setpoint is slightly better than that when controlling using the PI controller. The transient responses of both schemes are similar during an increase in setpoint.

5.9 Concluding Remarks

The application of the proposed scheme to the control of supply air temperature on a actual HVAC rig has been presented. A description of the HVAC rig including the

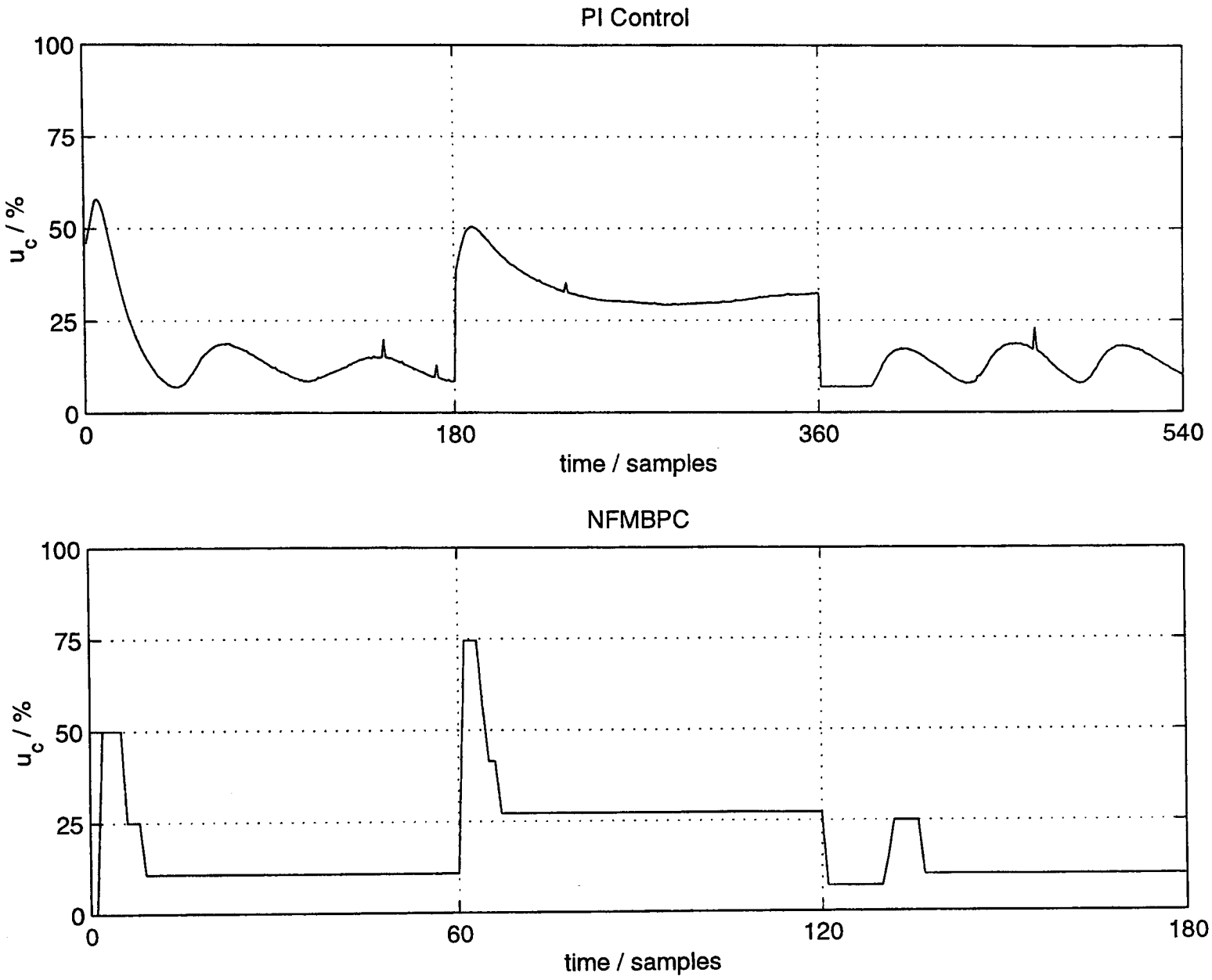


Figure 5.18: Control Activity Plots: PI Control vs. NFMBP Control of Supply Air Temperature at **Low Air Flow Rate**.

cooling coil subsystem to which the control scheme is applied, was also outlined. The test rig was configured for comparison testing (PI control versus NFMBP control) at both high and, the more difficult, low flow rates.

The neuro-fuzzy model of the cooling coil subsystem was trained on a *generic* HVACSim+ coil model, and the control results using such a model imply that generic neuro-fuzzy models may be suitable for the control of complex systems. Detailed in this chapter is a technique for developing generic neuro-fuzzy models for such systems, including an outline of the random training algorithm used to gather relevant training data, along with a discussion and example of why post-processing of the model output

is necessary.

Typical plots of fuzzy predicted temperatures generated from the neuro-fuzzy model have been presented. The plots show the fuzziness associated with each predicted temperature, in addition to the fact that, even though uncertainty about the predicted temperatures persists at the higher air flow rate, the drop in temperature associated with the increase in control action at this higher flow rate is less than is the case when the air mass flow rate is at 40% of the design value.

From the intermediate results presented, it has been shown that the inclusion of the normalized air mass flow rate was necessary to improve the performance of the controller. The results show that structural identification remains an important task, even if the model is generic in nature, as failure to correctly structurally identify the model may lead to a situation where it is too generic. This is an example of the compromise usually made between complexity and accuracy when working with generic models.

The final results of the application of the control scheme demonstrate the possibility of using generic fuzzy models to adequately *control* complex information-poor systems (as opposed to, say, performing fault diagnosis on these systems, as in (Bourdouxhe and Seutin, 1998)), where the control objective is uncertain. It has been shown that, under such conditions, the NFMBPC control scheme offers superior robustness handling capabilities compared to the PI control scheme. Finally, the results demonstrate one of the more attractive features employing the scheme with generic fuzzy models - the fact that the same generic model may be used in a class of control problems without necessarily re-tuning the controller to obtain acceptable results (as would not be the case under PI control). This is demonstrated by the results obtained when controlling in the high gain (low flow rate) scenario outlined above.

Although the results of Chapters 4 and 5 have been encouraging, very little has been said about the impact of the selection of application dependent parameters, such as choosing σ_c and/or σ_e for a given control problem. Chapter 6 investigates the effect of such choice on the control results of the scheme.

Chapter 6

Selection of Application Dependent Parameters

6.1 Introduction

In this chapter, the sensitivity of the control algorithm to various parameters specific to the proposed control scheme are examined. This examination takes place with the aid of both the detailed Simulink zone simulation used in Section 4.3 and detailed in Appendix D, and with a simple Matlab zone simulation subroutine, the code for which is presented in Section C.1.6 of Appendix C.

6.2 The Effect of Different Defuzzification Schemes on the Value of the Control Signal

The value of the final control signal applied to actuator is clearly affected by the defuzzification scheme used. In particular, for a given value of α (Section 3.2.3), the values of the percentage actuator control activity and the average percentage energy consumption of the cooling load due to the control trajectory (as defined in Equations 4.23 and 4.24) depend on the choice of defuzzification scheme used. This has implications for the control performance of the controller. In this section, therefore, the effect of employing each defuzzification scheme in the overall control scheme is examined and illustrated with typical control plots.

6.2.1 Simple Matlab Zone Simulation for Illustration of the Effect of Different Defuzzification Schemes

To illustrate the effect of different defuzzification schemes on actuator activity and cooling load energy consumption, a simple Matlab zone simulation is developed. Conceptually, it is similar to the Simulink zone model of Section 4.3 in Chapter 4, but is a greatly over-simplified representation of this zone.

The zone under consideration has an occupancy period of 8 hours (from 0900 hrs to 1700 hrs) with a sinusoidal solar gain of maximum value of 40 kW active from 1000 hrs to 1600 hrs. To simulate more realistically the solar gain, Gaussian distributed random cloud activity is included in the solar gain model. The zone has an internal heat gain (due to people, equipment etc.) of 20 kW and, due to the fabric of the walls, has an additional 20 kW of heat gain from conduction. These three gains together constitute a disturbance to the system. Given these disturbances, the control problem is to maintain the zone temperature in the acceptable comfort region around the setpoint of 21.5 °C while taking energy considerations into account.

During occupancy, thermal comfort in the zone is important, and the comfort tolerance, σ_c, is therefore very low. During lunchtime (1201 hrs to 1400 hrs) when the zone is unoccupied, however, energy consideration, and not comfort, is priority. The comfort tolerance is accordingly higher while the energy consumption tolerance, σ_e, is reduced.

The comfort and energy consumption tolerance profile of the zone therefore looks like:

Time of Day	Comfort Tolerance Profile	Energy Tolerance Profile
0900 - 1200	σ_c=1	σ_e=16
1201 - 1400	σ_c=2	σ_e=4
1401 - 1700	σ_c=1	σ_e=16

Table 6.1: Comfort and Energy Tolerance Profile of Matlab Simulated Zone

6.2.2 Control Plots under Different Defuzzification Schemes

Figures 6.1, 6.2, 6.3, 6.4 and 6.5 below illustrate the control plots under different defuzzification schemes, for $\alpha = 0.8$. Table 6.2 summarizes the percentage actuator control activity and the average percentage energy consumption of the cooling load due to the control trajectory.

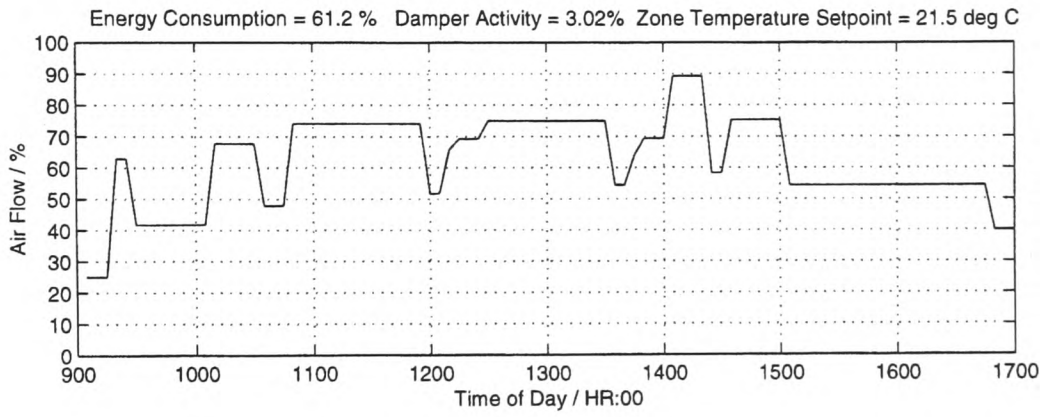


Figure 6.1: Control Plots using the CLID Scheme, $\alpha = 0.8$

An observation of the results shows that although the THD scheme (Figure 6.3) uses the least energy, it results in the highest control activity. In contrast, the CHD scheme (Figure 6.2) results in the least control activity, but the most energy consumption. The CMPD scheme (Figure 6.4) results in better overall performance than both the THD and CHD Schemes. In this particular application of the schemes, it should also be noted that the CNND scheme (Figure 6.5) has identical results to the CMPD scheme. This occurs when, for each sample step, the nearest neighbour to the previously applied control action is the member of the set of optimal control actions which has the maximum possibility. Under these conditions, both the CMPD and CNND schemes produce different results if the value of α is different in each scheme.

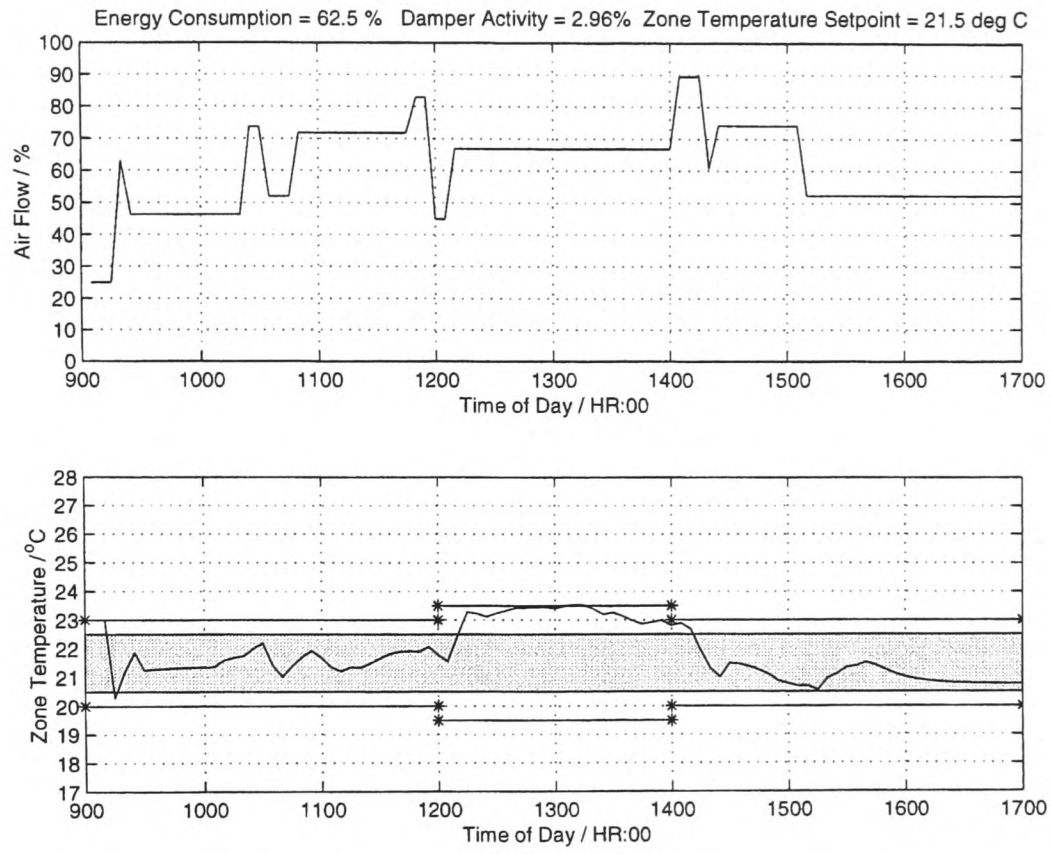


Figure 6.2: Control Plots using the CHD Scheme, $\alpha = 0.8$

The CLID scheme (Figure 6.1) produces the most attractive energy consumption and control activity results for the simulations carried out.

6.3 Sensitivity of Control Activity to the choice of the Actuator Activity Parameter, α

For any given defuzzification scheme outlined in this thesis, the control activity of the actuator is sensitive to the actuator activity parameter, α. In general, as α is increased, the activity of the actuator also increases. Conversely, a decrease in α will have a similar effect on the actuator activity. As implied earlier in Section 3.2.3, α may be thought of as the likelihood of defuzzifying at a particular sample step. Clearly, if this likelihood is higher for a particular scheme, then, on average, more defuzzifications will take place while the controller is switched on, than would have been the case for a lower value of α. This would correspond to an increased controller activity.

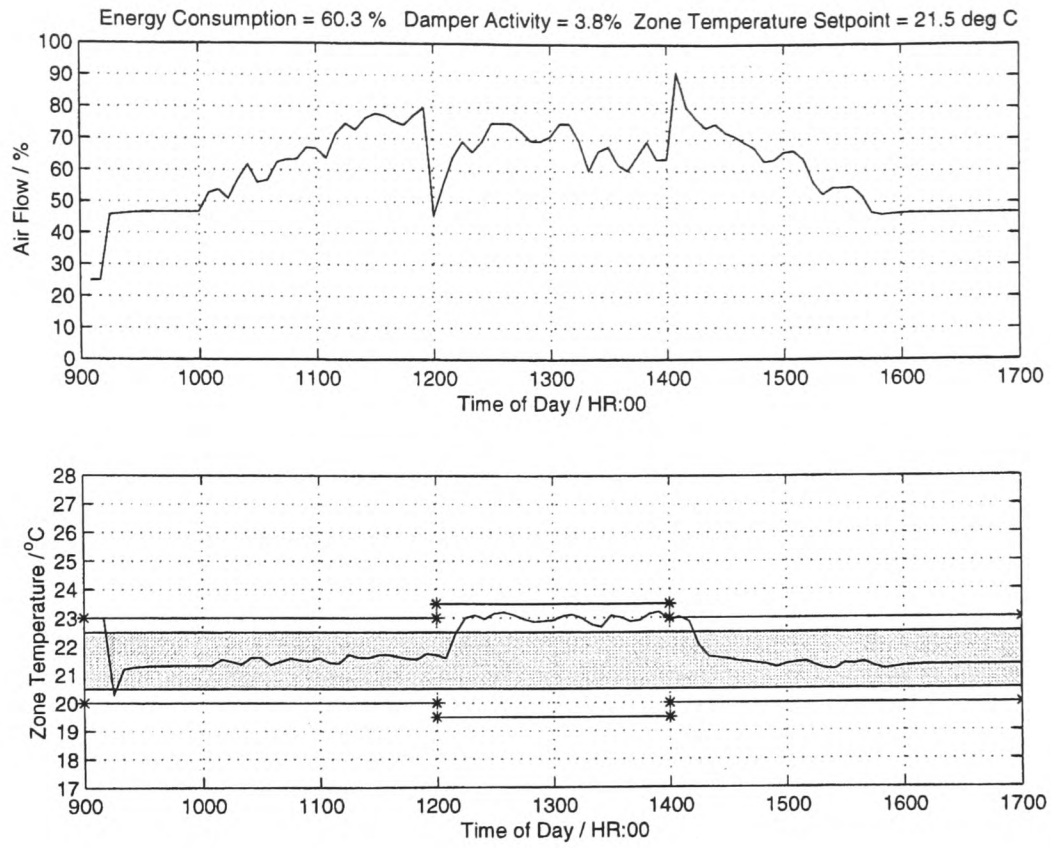


Figure 6.3: Control Plots using the THD Scheme, $\alpha = 0.8$

Using the simulated system of Chapter 4, Figures 6.6, 6.7 and 6.8 illustrate, while Tables 6.3 and 6.4 summarize the results.

The three figures show the results of a simulated system in which both σ_c and σ_e are kept constant at 1 and 4 respectively, and in which the CMPD defuzzification scheme is used. The value of α is decreased from 0.8 (in Figure 6.6) to 0.6 (in Figure 6.7), and then to 0.4 (in Figure 6.8). A corresponding decrease in controller activity is observed from the decrease in α. In Table 6.4, the relative decrease of control activity obtained by using $\alpha = 0.6$ is quantified as 25.90%, down to 74.10% of the control activity using $\alpha = 0.8$. In addition, there is a *further* drop in control activity of 26.05% when $\alpha = 0.4$ is used instead of $\alpha = 0.6$.

Also, observing Figure 6.8, it should be noted that the zone temperature is not controlled strictly within the comfort region, as dictated by the core of the fuzzy goal defined on thermal comfort. This is understandable given the values of σ_c and σ_e, and the low actuator activity parameter of $\alpha = 0.4$. Under this scenario, the controller is attempting to maintain a comfortable zone temperature with relatively low energy consumption, but with restrictions on the amount of activity on the actuator. The

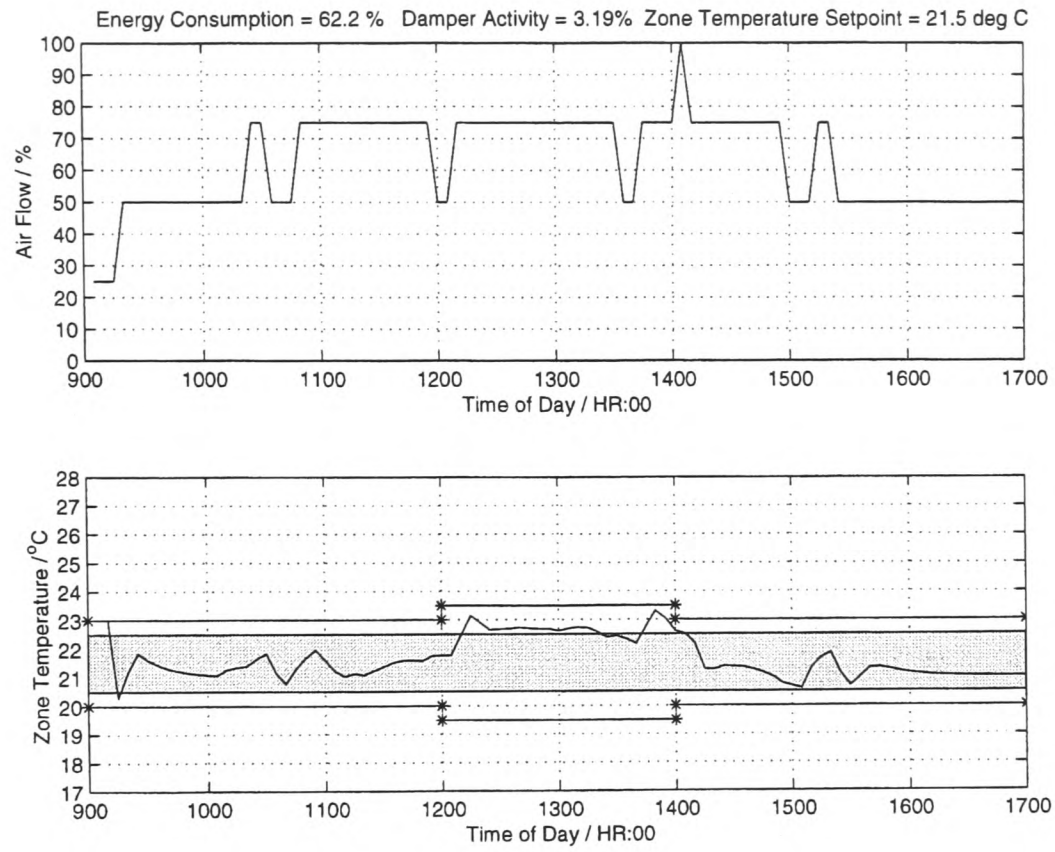


Figure 6.4: Control Plots using the CMPD Scheme, $\alpha = 0.8$

best the controller can do is to allow the zone temperature to drift slightly outside of the core (but still within the width) of the fuzzy goal defined on thermal comfort.

6.4 Choosing σ_c and σ_e

In this section, the effect on the energy consumption and control activity of choosing the thermal comfort and energy tolerance, σ_c and σ_e is examined. To achieve an appropriate comparison, the simulated system of Chapter 4 is set up with $\alpha = 0.8$ and with the CMPD scheme. The values of σ_c and σ_e are set to 1 and 3 respectively, as shown in Figure 6.9. Under this scenario, thermal comfort tolerance is very tight (a high degree of thermal comfort is required), while energy consumption tolerance is also very tight (a relatively low energy consumption is required). Figures 6.10 and 6.11 then show the results of the progression of σ_e (from 3) to 4 and then to 10, while σ_c is kept constant at 1.

Notice that, moving from Figure 6.9 to 6.11, the control action tends to assume higher

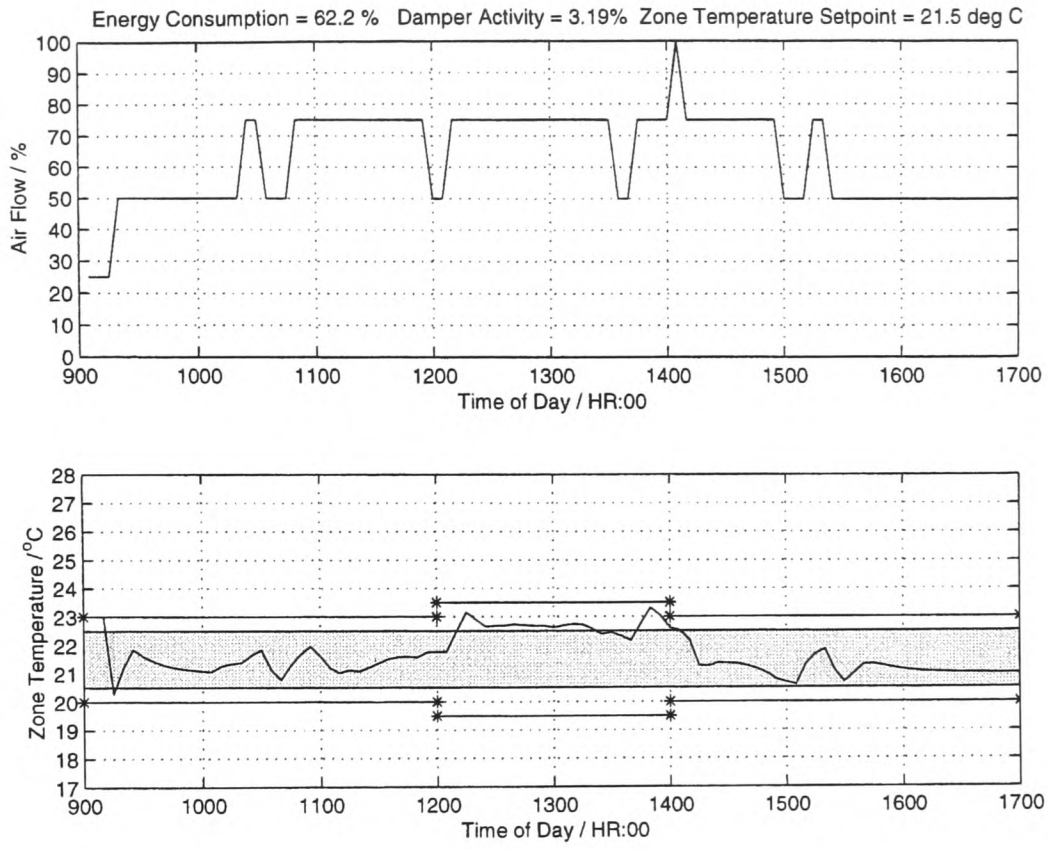


Figure 6.5: Control Plots using the CNND Scheme, $\alpha = 0.8$

values for longer periods of time as σ_e is increased. This is consistent with a relaxation of the energy consumption tolerance so that higher energy consumption is more acceptable. The first three rows of Table 6.5 show the relative energy consumption and control activity obtained from varying σ_e. The relative control activity in each of these cases is the same (100%), however, the relative energy consumption increases 12.35% when σ_e is increased from 3 to 4, and a further 6.53% when σ_e is increased from 4 to 10.

The value of σ_c is then set to 2 (with σ_e remaining at 10). The resulting values of relative energy consumption and relative control activity are recorded in the fourth row of Table 6.5. This shows a decrease of 16.94% in the relative energy consumption, and a decrease of 35.15% in the relative control activity when σ_c is changed from 1 to 2 (with a resulting relaxation in the thermal comfort tolerance). In addition, since thermal comfort is now no longer as important as it was with σ_c=1, the zone temperature as shown in Figure 6.12 is allowed to drift outside of the core of the fuzzy goal.

Defuzzification Scheme	Avg. Energy Con./%	Control Activity/%
CLID	61.20	3.02
CMPD	62.20	3.19
CNND	62.20	3.19
CHD	62.50	2.96
THD	60.30	3.80

Table 6.2: Control Activity and Energy Consumption typical of Different Defuzzification Schemes

Control Scheme	Avg. Energy Con./%	Control Activity/%
σ_c=1, σ_e=4, $\alpha = 0.8$	41.78	0.0668
σ_c=1, σ_e=4, $\alpha = 0.6$	40.56	0.0495
σ_c=1, σ_e=4, $\alpha = 0.4$	35.52	0.0321

Table 6.3: Varying α: **Average** Energy Consumption and Control Activity

6.5 Concluding Remarks

The sensitivity of the control performance to the choice of certain application dependent parameters has been presented. More specifically, it has been shown that the choice of a defuzzification scheme, the value of the actuator activity parameter, α, and the choice of the criteria tolerances, σ_c and σ_e, have an appreciable effect on controller performance.

A simple Matlab/Simulink zone simulation was developed to aid in the illustration of the effects on the system energy consumption and control activity of the various defuzzification schemes. It has been shown that, while the THD defuzzification scheme consumes the least system energy, it produces the most control activity of the five schemes examined. In contrast, the converse has been shown to be true of the CHD scheme. The CLID scheme produced the most attractive set of results in the tests carried out, and it is surmised that the main reason for this has been because of the effect of the linear interpolation that characterizes the scheme.

The effect of α on the defuzzified output has also been studied. It has been shown that

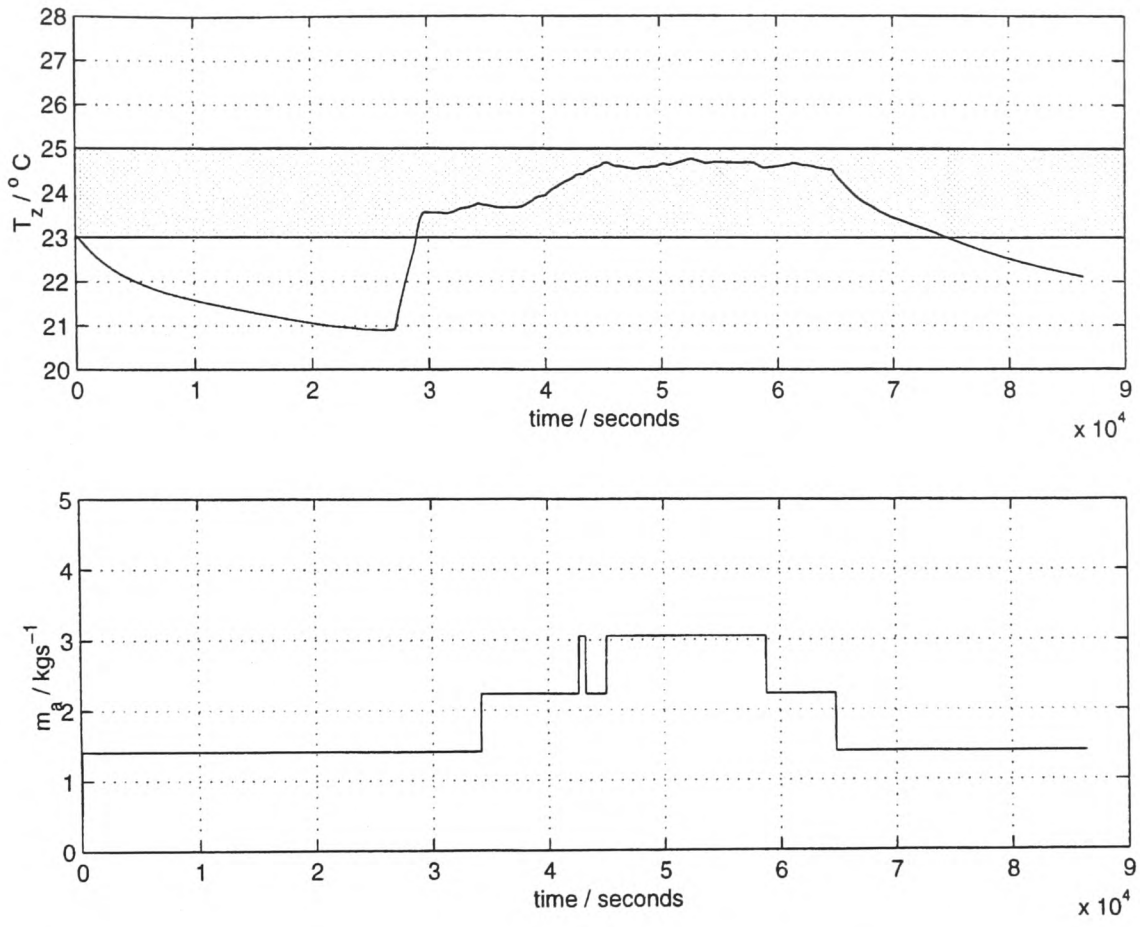


Figure 6.6: $\sigma_c = 1$, $\sigma_e = 4$, $\alpha = 0.8$

Control Scheme	Rel. Energy Con./%	Rel. Control Activity/%
σ_c=1, σ_e=4, $\alpha = 0.8$	100	100
σ_c=1, σ_e=4, $\alpha = 0.6$	97.08	74.10
σ_c=1, σ_e=4, $\alpha = 0.4$	85.02	48.05

Table 6.4: Varying α: **Relative** Energy Consumption and Control Activity

increasing α generally increases the control activity of the controller.

Finally, results showing the effect of choosing σ_c and σ_e have been presented. The results show a general increase in relative energy consumption when σ_e is increased, and a decrease in relative control activity when σ_c is increased. These results are consistent with the proposed theory.

Although encouraging results have been obtained from Chapters 4 and 5, and despite a closer look at the selection of application dependent parameters undertaken in this chapter, there are still a number of practical issues which require further consideration.

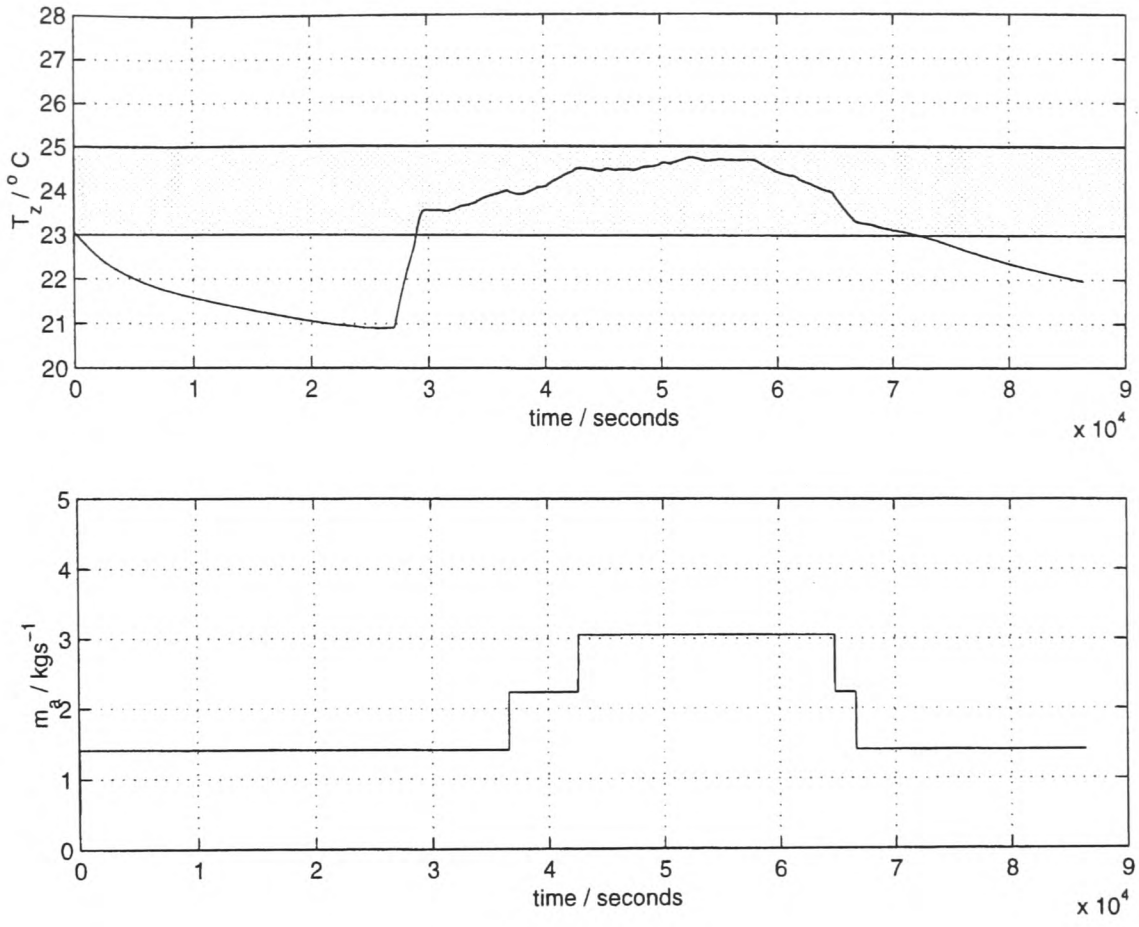


Figure 6.7: $\sigma_c = 1$, $\sigma_e = 4$, $\alpha = 0.6$

Suggestions for future work are made in the following chapter.

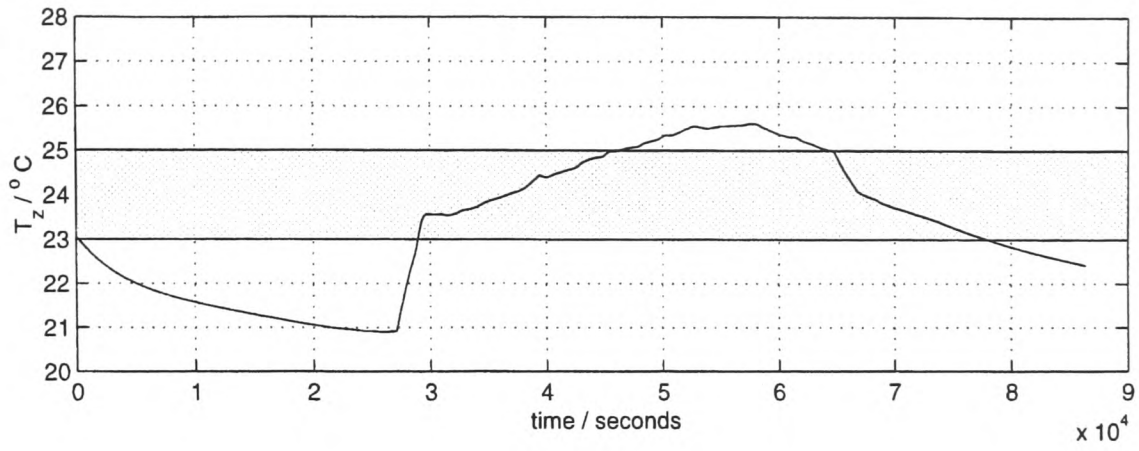


Figure 6.8: $\sigma_c = 1$, $\sigma_e = 4$, $\alpha = 0.4$

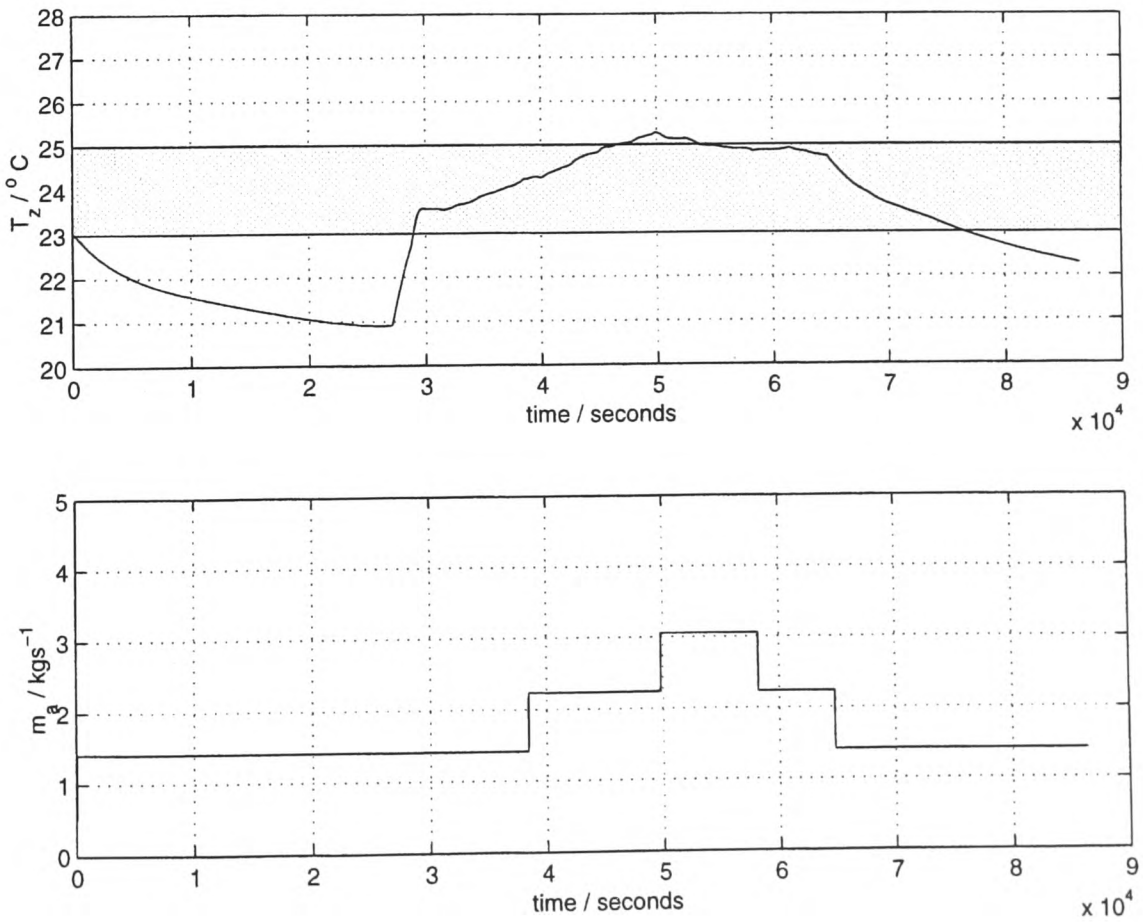


Figure 6.9: $\sigma_c = 1$, σ_e =3, $\alpha = 0.8$

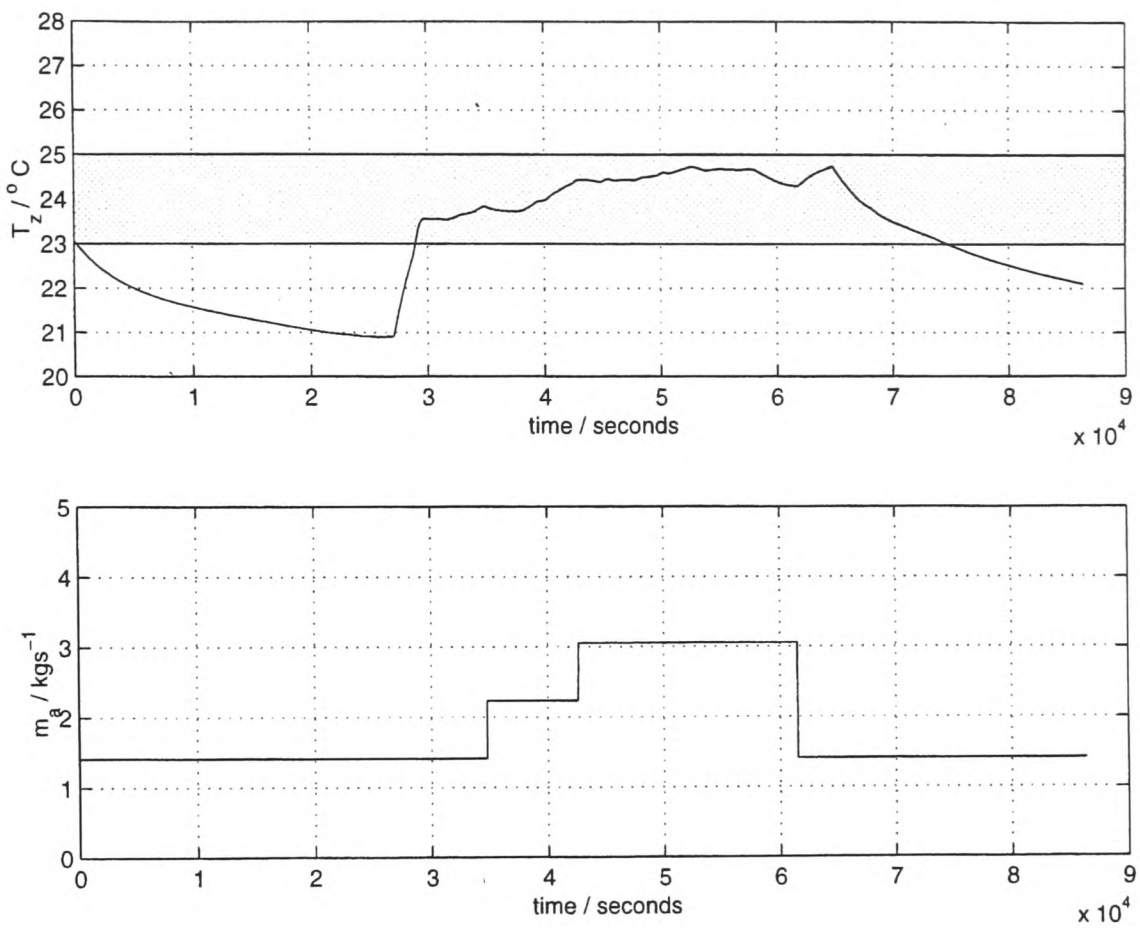


Figure 6.10: $\sigma_c = 1$, $\sigma_e = 4$, $\alpha = 0.8$

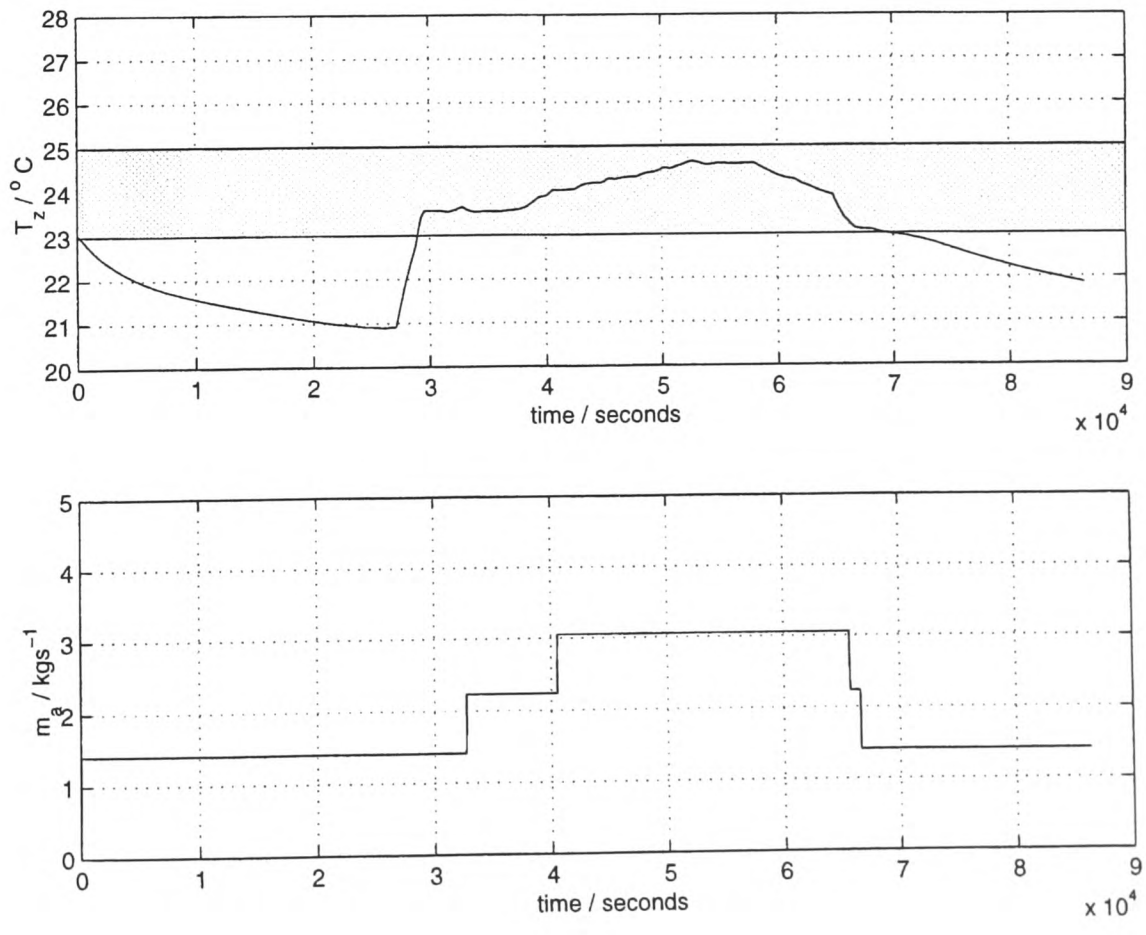


Figure 6.11: $\sigma_c = 1$, $\sigma_e = 10$, $\alpha = 0.8$

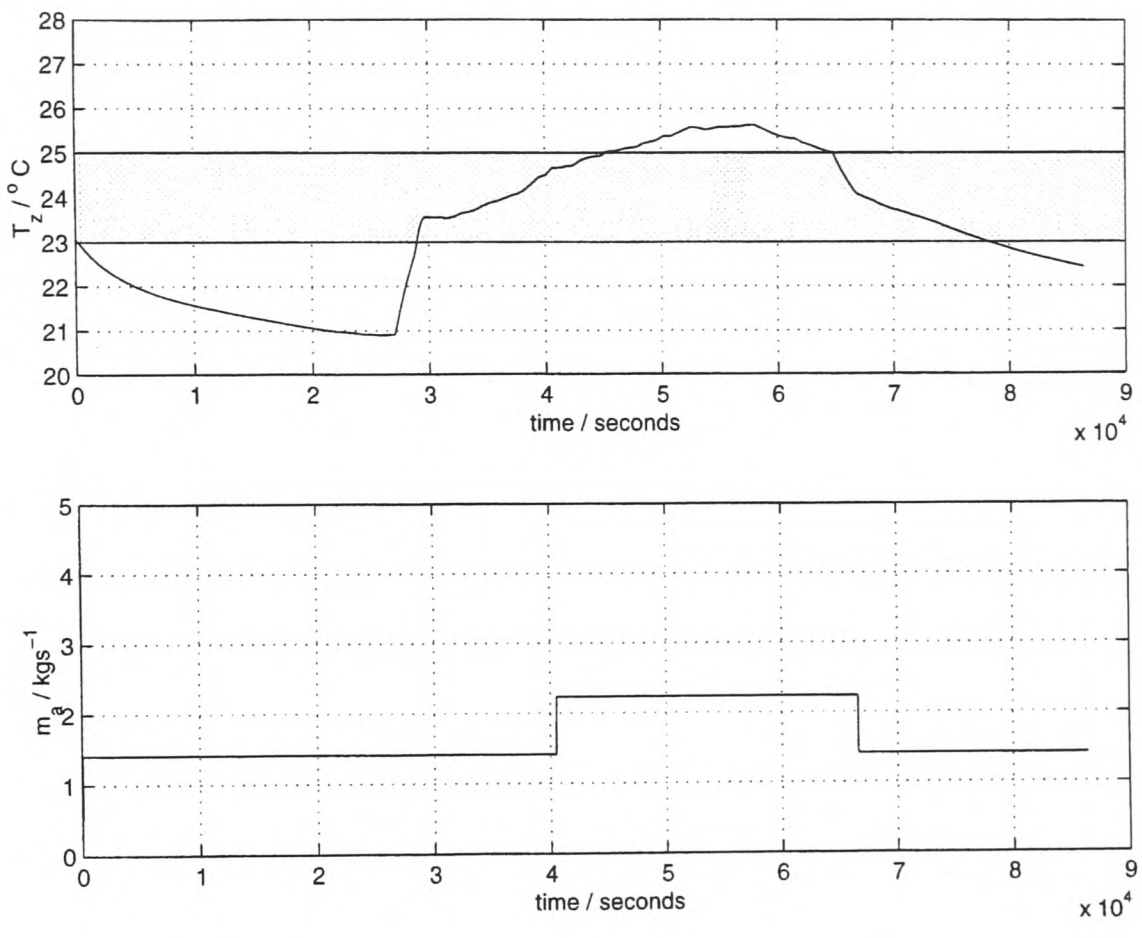


Figure 6.12: $\sigma_c = 2$, $\sigma_e = 10$, $\alpha = 0.8$

Control Scheme	**Rel. Energy Con./%**	**Rel. Control Activity/%**
σ_c=1, σ_e=3, $\alpha = 0.8$	81.12	100
σ_c=1, σ_e=4, $\alpha = 0.8$	93.47	100
σ_c=1, σ_e=10, $\alpha = 0.8$	100	100
σ_c=2, σ_e=10, $\alpha = 0.8$	83.06	64.85

Table 6.5: Varying σ_c and σ_e: **Relative** Energy Consumption and Control Activity

Chapter 7

Conclusions and Future Work

7.1 Conclusions

A neuro-fuzzy model-based predictive control scheme applicable to the control of information-poor systems has been developed. The scheme is particularly useful when the control objective of the controller is uncertain, since it attempts to formulate the control problem, including the control objective, in an inherently fuzzy environment.

The thesis has focused on the development and evaluation of a new neuro-fuzzy model-based predictive control scheme, where certain variables used in the optimization remain in the fuzzy domain. The method requires no training data from the actual plant under consideration, since the design of the controller assumes that detailed knowledge of the plant is unavailable. Instead, a simple neuro-fuzzy model of the general behaviour of the system under control is developed. The utilization of generic neuro-fuzzy models to control complex systems has also been investigated. It has been shown that such models are sufficient to provide acceptable control of the information-poor systems described in this work. The ideas of a fuzzy goal function and a fuzzy cost function have also been introduced and an algorithm for performing fuzzy optimization (decision-making) developed.

An attempt at an alternative to traditional height defuzzification has been outlined. The conditional defuzzification scheme developed was premised on the assumption that it was probably not necessary to defuzzify the controller output at each sample step in the control algorithm, as is currently the case in a number of neuro-fuzzy control schemes. Instead, certain conditions on the previously applied control signal (or a signal whose value was a function thereof) were tested to determine the extent to which this output signal should be changed, if any at all. The new value of control signal was assumed only if this test were true. The work done on conditional defuzzification has

shown that it is possible to develop a more intelligent defuzzification scheme; one more appropriate for, and applicable to, decision-making systems under uncertainty.

The development of the fuzzy proximity function has also been explored. This was attempted as one possible solution to some of the problems encountered with the subtraction of fuzzy numbers, as used in this thesis. It has been shown that the application of the function provides more acceptable results for the control problems considered, than would have been the case if fuzzy subtraction were used.

Results of the application of the control scheme to the control of thermal comfort in a simulated zone have been presented. This application represented a scenario where the control objective is uncertain. It has been shown that, under these prevailing conditions, acceptable control results are possible with the scheme outlined. In particular, the use of a large number of output fuzzy reference sets, in conjunction with the philosophy of not defuzzifying the model output used for prediction, has been shown to be a successful approach.

Additionally, the proposed control scheme was also applied to the control of the supply air temperature of an experimental air-handling unit in the laboratory. In addition to an uncertain control objective, this application represented a scenario where the system was real and was therefore subjected to the uncertainties associated with the measurement of and relationship between system variables. An approach for developing and incorporating into the scheme, robust generic neuro-fuzzy models, has also been outlined. The approach is based on generic neuro-fuzzy models designed to replicate the general thermodynamic behaviour of a class of cooling coil subsystems, in which is included as a subset, the actual cooling coil subsystem under consideration. The use of generic neuro-fuzzy models in the control of complex information-poor systems has therefore been applied to a practical problem. Experimental results confirmed that this approach was able to adequately control the system without the need to re-design the generic model, or re-tune the controller (as might have been the case under PI control) for changes in the behaviour of the system. The robustness of the approach has therefore been demonstrated.

From the research, it has been concluded that precious resources (as measured by actuator activity, for example) need not be wasted when controlling information-poor systems. In addition, it has also been shown that a very precise (and sometimes

not necessarily accurate) control value computed at each sample step is unnecessary when the control objective is itself uncertain. Rather, by defining the system and its environment in the fuzzy domain, the fuzzy decision algorithms developed here have been employed to get an "acceptable" control performance.

In order to further explore the potential of the scheme outlined in this work, possible directions for future research are suggested in the following section.

7.2 Suggestions for Future Work

Further consideration needs to be given to a few aspects of the scheme as outlined.

Firstly, the work done in this thesis has used the Multiple Fuzzy Cost Functions, Multiple Fuzzy Goals approach of Section 2.4.2. The application of the proposed scheme to a practical problem using the Single Fuzzy Cost Function, Single Fuzzy Goal approach of Section 2.4.1 needs to be investigated. In order to be adequately attempted, more work should be done on developing a general, robust fuzzy arithmetic framework which allows the fuzzy numbers used in this approach to be more easily mathematically manipulated.

In Chapter 4, the application of the proposed control scheme to the control of zone thermal comfort demonstrated the successful use of the fuzzy-input neuro-fuzzy model of Section 4.5.1. A system-specific solution to the problem of producing a fuzzy prediction from fuzzy inputs was outlined. Further work is needed to develop a more generalized theory of fuzzy-input neuro-fuzzy models.

Additionally, the tests carried out on the HVAC laboratory rig did not involve dynamic training on the air mass flow rate input, $\dot{m}_a$, of the neuro-fuzzy model. This was because an assumption was made that, during the course of the experiment, the rate of change of $\dot{m}_a$ was negligible. While justifiable for the laboratory system considered, in many HVAC applications this air mass flow rate fluctuates and represents a disturbance to the system. Further work involving training the neuro-fuzzy model on this dynamic input is needed to adequately control such systems.

Also, the algorithms developed here were fairly easily implemented. However, in addition to its ease of implementation, an important aspect of any control scheme is its computational intensity (demand). Although the computational demand of the algorithms was sufficiently low to satisfy the requirements of the computer hardware used to implement them, it is believed that this computational demand could be further reduced by more efficient (vector) coding of the algorithms. Also, the implementation of a long term memory function which would aid in the optimization (decision-making) process requires further investigation.

The technique developed in this thesis employs off-line training of the neuro-fuzzy model. However, because of the modified RSK algorithm used in this work to identify the neuro-fuzzy model, it is possible to fairly easily adapt the technique for on-line training. A further development to the work might involve re-coding the training algorithms so that they would now gather training data from the actual plant under control during the normal operation of the plant, and use this training data to recursively update the generic plant model on-line.

Further experiments are also necessary to explore the robustness properties of the proposed control technique with respect to rooms of different thermal capacities. This would involve developing additional thermal models to replicate a spectrum of the thermal behaviour of different buildings, and re-applying the control technique of Chapter 4 to each thermal model to get an indication of, and appreciation for, the overall robustness of the approach, as outlined.

Finally, the thesis has concentrated on cooling and cooling coil subsystems. An interesting development of the work would be to take an integrated system-wide approach to the control problem of HVAC systems, obtaining generic neuro-fuzzy models of additional air-conditioning subsystems, including fans and heating coils. It is believed that, in such a case, models would need to be generated only once for each class of subsystem design, rather than for each specific subsystem under test. The entire approach could then be implemented in a general Building Energy Management System, for example.

Appendix A

Fuzzy Arithmetic

A.1 Introduction

In this appendix, an overview of operations involving fuzzy numbers and intervals currently found in the literature is presented.

A.2 Fuzzy Arithmetic

Fuzzy arithmetic, as used in this work, refers to arithmetic operations carried out on fuzzy numbers and fuzzy intervals; both of which are described in the following section.

A.2.1 Fuzzy Numbers and Fuzzy Intervals

The work in this thesis makes use of a particular type of fuzzy set which has its universe of discourse defined on the set $\mathbb{R}$ of real numbers. Additionally, as outlined in Section 1.5.1, the membership functions of the fuzzy sets used here are of the form

$$A : \mathbb{R} \rightarrow [0, 1] \tag{A.1}$$

and, therefore, have an associated quantitative meaning. This is usually the case in fuzzy control.

If these fuzzy sets capture an intuitive conception of approximate numbers or intervals,

such as "numbers that are close to a given real number" or "numbers that are around a given interval of real numbers", which is the case here (see the introductory section of Chapter 2), then these fuzzy sets may be viewed as fuzzy numbers or fuzzy intervals (Klir and Yuan, 1995).

A fuzzy set A on $\mathbb{R}$ must satisfy at least the following three properties, in order to be considered a fuzzy number:

- A must be a normal fuzzy set;
- ${}^{\alpha}A$, the α-cut of A, must be a closed interval for every $\alpha \in (0, 1]$;
- the support of A (see Equation 2.8), $supp(A)$, must be bounded.

where the α-cut of A, ${}^{\alpha}A$, is defined as:

$$ {}^{\alpha}A = \{x \in X \mid \mu_A(x) \geq \alpha\},\ X \in \mathbb{R} \tag{A.2} $$

Figure A.1 is an illustration of crisp and fuzzy numbers and interval.

A.2.2 Arithmetic Operations on Fuzzy Numbers and Fuzzy Intervals

The four basic arithmetic operations on fuzzy numbers and intervals, addition, subtraction, multiplication and division, may be performed by employing either of two techniques: (i) the extension principle or (ii) α-cuts. The extension principle is used in this work and is treated next. However, for completeness, the α-cut technique is also presented.

The Use of the Extension Principle

The Extension Principle may be summed up as follows:

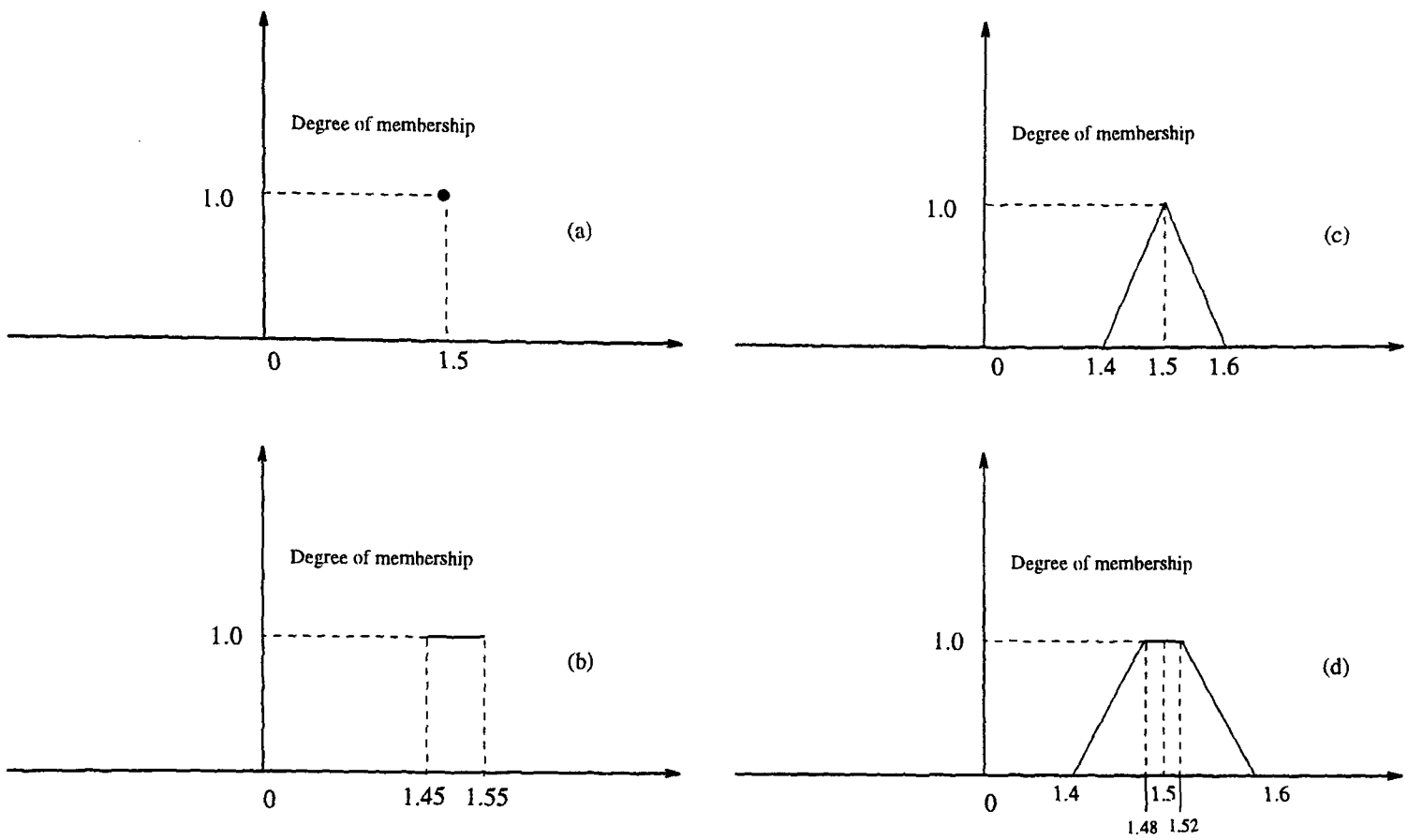


Figure A.1: A Comparison of (a) a Real Number and (b) a Crisp Interval with (c) a Fuzzy Number and (d) a Fuzzy Interval

Let A be a fuzzy set defined on X,

$$A(x;i) = \{(x_1, \mu_A(x_1)), (x_2, \mu_A(x_2)), \ldots, (x_n, \mu_A(x_n))\},\ x_i \in X,\ i = 1 \ldots n \qquad \text{(A.3)}$$

and let the image of A under a mapping $f(\cdot)$ be a fuzzy set B,

$$B(x;i) = \{(y_1, \mu_B(x_1)), (y_2, \mu_B(x_2)), \ldots, (y_n, \mu_B(x_n))\} \qquad \text{(A.4)}$$

where $y_i = f(x_i)$ and $i = 1 \ldots n$. Then, if $f(\cdot)$ is a many-to-one mapping,

$$\mu_B(y) = \max_{x=f^{-1}(y)} \mu_A(x) \qquad \text{(A.5)}$$

With the extension principle, arithmetic operations on real numbers are extended to

fuzzy numbers.

If A and B are two fuzzy numbers and $*$ represents any of the four basic arithmetic operations, then a fuzzy set on $\mathbb{R}$, $A * B$, may be defined as:

$$(A * B)(z) = \sup_{z=x*y} \min [A(x), B(y)] \tag{A.6}$$

for all $z \in \mathbb{R}$. More specifically:

$$(A + B)(z) = \sup_{z=x+y} \min [A(x), B(y)] \tag{A.7}$$
$$(A - B)(z) = \sup_{z=x-y} \min [A(x), B(y)] \tag{A.8}$$
$$(A \cdot B)(z) = \sup_{z=x \cdot y} \min [A(x), B(y)] \tag{A.9}$$
$$(A/B)(z) = \sup_{z=x/y} \min [A(x), B(y)] \tag{A.10}$$

Digital implementation of this principle results in making convolution-type point-wise computations on the universes of A and B and logical type operations on the corresponding membership grades.

The Use of the α-Cut Technique

Since each fuzzy number can fully and uniquely be represented by its α-cuts and since the α-cuts of a fuzzy number are closed intervals of real numbers for all $\alpha \in (0,1]$, then arithmetic operations on fuzzy numbers (and intervals) may be carried out by operations on their α-cuts. These operations therefore reduce to operations on closed intervals and are subject to a branch of mathematics called *interval analysis.*

If A and B are two fuzzy numbers and $*$ represents any of the four basic arithmetic operations, then a fuzzy set on $\mathbb{R}$, $A * B$, may be defined, by way of its α-cut, ${}^{\alpha}(A * B)$, as

$$^{\alpha}(A * B) =^{\alpha} A *^{\alpha} B \tag{A.11}$$

for any $\alpha \in (0,1]$. Note that when $* = /$, it's a requirement that $0 \notin {}^{\alpha}B$, for all $\alpha \in (0,1]$.

Let ${}^{\alpha}A = [a,b]$ and let ${}^{\alpha}B = [d,e]$, then ${}^{\alpha}A *^{\alpha} B = [a,b] * [d,e]$, which, in turn, is given more specifically by:

$$\begin{aligned} [a,b] + [d,e] &= [a+d, b+e] && \text{(A.12)} \\ [a,b] - [d,e] &= [a-e, b-d] && \text{(A.13)} \\ [a,b] \cdot [d,e] &= [min(ad,ae,bd,be), max(ad,ae,bd,be)] && \text{(A.14)} \end{aligned}$$

and, provided that $0 \notin [d,e]$,

$$\begin{aligned} [a,b] \,/\, [d,e] &= [a,b] \cdot [1/e, 1/d] \\ &= [min(a/d, a/e, b/d, b/e), max(a/d, a/e, b/d, b/e)] && \text{(A.15)} \end{aligned}$$

Though this technique may be applied to both continuous and discrete fuzzy sets, digital implementation results in discretization errors in the construction of $A \cdot B$ and A/B if either A or B is discrete. Since discrete fuzzy sets are used in this work, this technique has been avoided.

Appendix B

Fuzzy Relational Arrays and The RSK Training and Identification Scheme

B.1 Introduction

Identification of the fuzzy relational array has been the topic of much research in the past. If a Multiple-Input-Single-Output (MISO) system has r fuzzy reference sets defined on each of the input spaces and on the output space, then in identifying the relational array, R, we are in essence determining the entries in an array of order r^{n+1} connecting the n inputs of the MISO system to the one output. Two methods often used in practice are: that proposed by Xu and Lu (Xu and Lu, 1987) and that proposed by Ridley, Shaw and Kruger (Ridley, Shaw and Kruger, 1988).

B.2 The RSK Identification Scheme

Consider a MISO relational model consisting of n inputs $(x_1, \ldots, x_n)$ and one output, y, and where the input and output spaces are characterized by $r_1, \ldots, r_n$ and j fuzzy reference sets respectively. Then the entry $R_{s_{11},\ldots,s_{nk},q}$ in the fuzzy relation measures the possibility of obtaining an output y in set q $(q \leq j)$ from inputs $x_1, \ldots, x_n$ in sets $s_{11}, \ldots, s_{nk}$ respectively $(k \leq r_n)$. The RSK algorithm estimates this entry from observations by taking a weighted average as follows:

$$R_{s_{11},\ldots,s_{nk},q} = \frac{\sum_g f_{s_{11},\ldots,s_{nk}}(g) \cdot \mu_q(g)}{\sum_g f_{s_{11},\ldots,s_{nk}}(g)} \tag{B.1}$$

where $f_{s_{11},\ldots,s_{nk}}(g)$ is the product $\mu_{s_{11}}(g),\ldots,\mu_{s_{nk}}(g)$ and where the summation runs over the set of relevant observations, g. This set of observations, g, consists of individual data points (observations) $g_1, g_2 \ldots g_p$, where the p^{th} observation g_p is the data point $(x_{1,p}, x_{2,p}, \ldots x_{n,p}, y_p)$. This is analogous to estimating a probability from a table of observed frequencies (Ridley, Shaw and Kruger, 1988).

B.3 The Modified RSK Identification Scheme

A modification to the RSK algorithm (Tan, 1997) involves choosing for *each combination of input sets*, that one data point, $g^* \in g$, which maximizes the product $\mu_{s_{11}}(g),\ldots,$ $\mu_{s_{nk}}(g)$, rather than running the summations over all relevant observations, g. This is equivalent to pre-filtering the data so that only data close to the apex of each fuzzy reference set is considered, since for such data $\mu_{s_{11}}(g),\ldots,\mu_{s_{nk}}(g) \approx 1$ (maximum activation). This modification removes the need to pass *all* data points through *all* the input fuzzy reference sets and is an attempt to minimize modelling errors at the apex of the fuzzy reference sets in the presence of unevenly distributed training data. Since the original RSK algorithm assumes evenly distributed training data and this scenario is quite difficult to achieve in practice, choosing that data point at the apex ensures that the modelling error at that point is minimized. In addition the modified algorithm is less computationally intensive and more appropriate for on-line identification, should this be decided upon as future modifications to this work.

Using this scheme, since now only one observation, g^*, is used in the identification process, the relational array reduces to:

$$R_{s_{11},\ldots,s_{nk},q} = \frac{f_{s_{11},\ldots,s_{nk}}(g^*) \cdot \mu_q(g^*)}{f_{s_{11},\ldots,s_{nk}}(g^*)} \tag{B.2}$$

$$= \mu_q(g^*) \tag{B.3}$$

Interestingly, it should also be noted that since Xu and Lu identification of fuzzy relational array (Xu and Lu, 1987) produces a relational array given by:

$$R_{s_{11},\ldots,s_{nk},q} = \max_{g} f_{s_{11},\ldots,s_{nk}}(g) \cdot \mu_q(g) \tag{B.4}$$

then, for the noise-free case, pre-filtering the training data would produce a modified Xu and Lu identification algorithm given by:

$$R_{s_{11},\ldots,s_{nk},q} = f_{s_{11},\ldots,s_{nk}}(g^*) \cdot \mu_q(g^*) \tag{B.5}$$

$$\approx \mu_q(g^*) \tag{B.6}$$

since $f_{s_{11},\ldots,s_{nk}}(g^*) \approx 1$.

B.4 A Section of the Fuzzy Relational Array used in Chapter 4

```
R(:,:,6,3,3,2) =

  Columns 1 through 6

    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

  Columns 7 through 12

    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	0.14361

Columns 13 through 18

0.00000	0	0.58218	0.41782	0	0
0.00000	0.73178	0.26822	0	0	0
0.39961	0.60039	0	0	0	0
0.83606	0.16394	0	0	0	0
0.85639	0	0	0	0	0

Columns 19 through 24

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

Column 25

0
0
0
0
0

Appendix C

Matlab Codes used in the Implementation of the Control Scheme

C.1 Matlab Codes

In this appendix, various Matlab codes used in the implementation of the proposed control scheme are presented. The codes have been written for Matlab version 5.1, and should be able to be run on any higher version of the software.

C.1.1 Matlab Implementation of the Control Scheme

The following is a printout of the Matlab code used to implement the control algorithm of Chapter 5. The general structure remains true for the version of the code used in Chapter 4:

```
%Neuro-fuzzy Predictive Control algorithm for OXrig
% modelled from pirig5.m
% one step ahead prediction with control horizon = 1 (Td=0, Tsample=30 sec)

clear clf
```

```
%Declare and initialize variables
run_time=90*60;                          % run time in seconds
sample_time=30;                          % sampling interval in sec.
num_samp=run_time/sample_time;           % number of samples
rawTsa=23*ones(1,num_samp);              % supply air temperature to be controlled
u=0*ones(1,num_samp);                    % control action in percentage valve open

alpha=0.8;                               % fuzzy actuation parameter

in1=11;                                  % # of sets defined on Tao(n)
in2=5;                                   % # of sets defined on u(n-1)
in3=5;                                   % # of sets defined on Tai(n)
in4=3;                                   % # of sets defined on ma(n)
out=26;                                  % # of sets defined on Tao(n+1)

unnorm_Tsa_low=7;                        % min unnormalized outlet air temperature/ deg C
unnorm_Tsa_high=30;                      % max unnormalized outlet air temperature/ deg C
unnorm_Tsa_range=unnorm_Tsa_high-unnorm_Tsa_low;

ul=0;            % min control signal
uh=1.0;          % max control signal

% load goal functions and UODs

load GFcrig2; load R_allgeneric; load Rgeneric; R=R_allgeneric;
u=0*ones(1,num_samp);

% Set up one-step-ahead proximity:
    proximity2 = [[0 1;
    0.0400 0;
    0.0800 0;
    0.1200 0;
    0.1600 0;
    0.2000 0;
    0.2400 0;
    0.2800 0;
    0.3200 0;
    0.3600 0;
    0.4000 0;
```

```
    0.4400 0;
    0.4800 0;
    0.5200 0;
    0.5600 0;
    0.6000 0;
    0.6400 0;
    0.6800 0;
    0.7200 0;
    0.7600 0;
    0.8000 0;
    0.8400 0;
    0.8800 0;
    0.9200 0;
    0.9600 0;
    1.0000 0]];

% Tsa setpoint trajectory in deg. C

W_rawTsa=18.04*ones(4,2,num_samp);
W_rawTsa(:,2,:)=ones(4,1,num_samp);


first=30*60/sample_time; second=60*60/sample_time; for a=1:first
   W_rawTsa(:,1,a)=18.04*ones(4,1);
end for a=first+1:second
   W_rawTsa(:,1,a)=12.52*ones(4,1);
end for a=second+1:num_samp
    W_rawTsa(:,1,a)=18.04*ones(4,1);
end

FOFc(26,2,in2)=zeros; POS=IN_SET2(2:in2+1);

frame = 200;                                    %# of samples before
graph is redrawn. graphcount = 0;

% Real time control algorithm
for i=2:num_samp
   rowc=1;
   i;
   START = clock;
```

```
   Vx(:,i)=adc_matlab;                          % read integer inputs from ADC (0<=inp.<=4095)
%   pause(sample_time);
   for k=1:8                                    % channel number
      Vx(k,i)=Vx(k,i)*(10/4096) - 5;            % conversion from integer to volts (-5<=V<=5)
    end
    Tx(i)=5*Vx(1,i);                            % conversion to supply air temperature (T=5V) on channel 1
    Tmix(i)=5*Vx(3,i);                          % conversion to mix air temperature (T=5V) on channel 3
    ma_x(i)=2*100/8.53*(Vx(2,i)-0.735);         % conversion to percentage mass air flow rate on channel 2
  pitot_flow(i)=(2*Vx(4,i)).^0.5;

     for j=1:100
     y=1;
    end
     rawTsa(i)=Tx(i);                           % assign 1st channel of ADC to rawTsa

     Tsa(i)=(rawTsa(i)-unnorm_Tsa_low)/unnorm_Tsa_range;
     W_Tsa(:,:,i)=W_rawTsa(:,:,i);
     W_Tsa(:,1,i)=(W_rawTsa(:,1,i)-unnorm_Tsa_low)/unnorm_Tsa_range;

%    ************************
    for control_step=1:1:in2;
       uprime=((uh-ul)*(control_step-1))/(in2-1) +ul;
       %[rawTsa(i) u(i) uprime]

       ym_np1(1:out,1:2)=fuzinfcoil(Tsa(i),uprime,IN_SET1,IN_SET2,OUT_SET,R,in1,in2,out);
     % set up ym as a 3-D matrix with # of rows
     % equal to # of output sets, 2 columns (1 for universe and 1 for membership)
     %and # of pages equal to # of samples.
    YM1=ym_np1(:,1)'*ym_np1(:,2)/sum(ym_np1(:,2));
     YM1=23*YM1+7;

       control=[[ul:(uh-ul)/(in2-1):uh]' zeros(in2,1)];

       control(control_step,2)=1;


       proximity1=sfp5(W_Tsa(:,:,i),ym_np1(:,:));

       FOFc(:,:,control_step)=sfadd(proximity1,proximity2);
end
```

```
TAc(1:in2)=sum(FOFc(:,2,1:in2));

RAc(1:in2)=zeros(1,in2); RA(1:in2)=zeros(1,in2);

while isempty(find(RA~=0))

    for control_step=1:in2
        CAc(control_step)=sum(min(GFcrig2(rowc,:),FOFc(:,2,control_step)'));
    end

  CA1c=CAc(TAc>0);
    TA1c=TAc(TAc>0);

    RAc=CA1c./TA1c;
   RA=RAc;
    rowc=rowc+1;
end RA

u(i)=fuzact5(alpha, RA,u(i-1),IN_SET2,POS,uh,ul,in2); if u(i)<0.07
   u(i)=0.07;
end

u(i)

    Vu(i)=10*u(i);                                    % convert % to voltage

    integer_u(i)=round(409.5*Vu(i));                  % convert voltage to integer for current control action
    dac_matlab(0,integer_u(i));                       % write current control action to channel 0 of DAC

%-- Draw graph
  graphcount = graphcount + 1;
  if (rem(graphcount,frame) == 1)
    clf;
    tmp = fix(graphcount/frame);
    figa = axes('XLim',[tmp*frame (tmp+1)*frame],'YLim',[5 25]');
    set(figa,'NextPlot','add','Box','on','XGrid','on','YGrid','on');
    lineh1 = line('EraseMode','none','Color',[1 0 0.75],'LineStyle','-');
    lineh2 = line('EraseMode','none','Color',[0 0 0.75],'LineStyle',':');
    drawnow;
```

```
 end;
 if i > 2
   set(lineh1,'XData',[i-1 i],'YData',[rawTsa(i-1) rawTsa(i)]);
   set(lineh2,'XData',[i-1 i],'YData',[W_rawTsa(1,1,i-1) W_rawTsa(1,1,i)]);
   drawnow;
 end
 while (etime(clock,START) < sample_time)
 end
end
save RIGVAR6 Tmix rawTsa W_rawTsa u ma_x pitot_flow alpha rowc
run_time sample_time
dac_matlab(0,0)                                    %fully close the valve
```

C.1.2 Fuzzy Proximity Matlab Code

The following is a printout of the Matlab code used to implement the fuzzy proximity algorithm:

```
function PROXCO2=fp5(W,Y)

rrrx=size(W); [rx,cx]=find(W(:,2)==1);

rrry=size(Y);

Z=zeros(rrry(1),rrrx(1)); MV=zeros(rrry(1),rrrx(1));

W_L=W(min(rx),1); W_R=W(max(rx),1); Wleft=W(1:min(rx),:);
Wright=W(max(rx):rrrx(1),:);

for i=1:rrry(1)
    if W_L<=Y(i,1) & Y(i,1)<=W_R
        Z(i,1:min(rx))=0;
        MV(i,1:min(rx))=Y(i,2);

    elseif Y(i,1)<W_L
        Z(i,1:min(rx))=(Wleft(:,1)-Y(i,1))';
```

```
        MV(i,1:min(rx))=(min(Wleft(:,2),Y(i,2)*...
    ones(length(Wleft(:,2)),1)))';

    else    Z(i,max(rx):rrrx(1))=(Y(i,1)-Wright(:,1))';
        MV(i,max(rx):rrrx(1))=(min(Wright(:,2),Y(i,2)*...
    ones(length(Wright(:,2)),1)))';
    end
end

for ii=1:rrrx(1)*rrry(1)
    C(ii,1)=Z(ii);
    C(ii,2)=max(MV(find(Z==Z(ii))));
end

PROXCO=C; [b,i,j]=unique(PROXCO(:,1)); PROXCO=[b PROXCO(i,2)];

q=find(PROXCO(:,1)<0); PROXCO(q,:)=[];

PROXCO2=zeros(20,2); PROXCO2(:,1)=[0:0.5:9.5]';

for i=1:20
    [r,c]=find(PROXCO(:,1)==PROXCO2(i,1));
    if isempty(r)==1
        PROXCO2(i,2)=0;
    else
        PROXCO2(i,2)=PROXCO(r,2);

    end
end
```

C.1.3 Matlab Codes used in the Implementation of the different Defuzzification Schemes

The following is a printout of the Matlab codes used to implement the various defuzzification schemes outlined in Chapter 3:

Traditional Height Defuzzification Matlab Code

```
% Traditional Height (full) Defuzzificztion

function u=fuzact4(alpha,RA,u_prev,IN_SET2,POS,uh,ul,in2)

u=RA*IN_SET2(2:in2+1)/sum(RA);
```

Conditional Maximum Possibility Defuzzification Matlab Code

```
function u=fuzact2(alpha,RA,u_prev,IN_SET2,POS,uh,ul,in2)

%if u_prev==0.07;
 %  u_prev=0;                          % correct for the deadband on the actuator
%end

index=find(IN_SET2==u_prev);
index=index-1;                        %determine index of previous applied control

% decide whether or not to move
if RA(index)>alpha*max(RA)
   u=u_prev;                          % if the similarly indexed RA value is within band, don't change control
else

   % now that we've decided to move, where do we go?
   [y,i]=max(RA);
   u=IN_SET2(i+1);

end
```

Conditional Nearest Neighbour Defuzzification Matlab Code

```
function u=fuzact2(alpha,RA,u_prev,IN_SET2,POS,uh,ul,in2)

%if u_prev==0.07;
 %  u_prev=0;
%end                                    % correct for the deadband on the actuator

index=find(IN_SET2==u_prev);
index=index-1;                         %determine index of previous applied control

% decide whether or not to move
if RA(index)>alpha*max(RA)
   u=u_prev;                           % if the similarly indexed RA value is within band, don't change control
else

   % now that we've decided to move, where do we go?
   INDEX2=find(RA>=alpha*max(RA));     % isolate all indices > alpha * max(RA).
%This is the set of indices of the candidate set of u

   if isempty(INDEX2)==1
      u=u_prev;                        % if no such index exist, stay where you are
   else
      [value,pointer]=min(abs(INDEX2-index));  % determine closest member of this candidate set of u to u_prev
      pointer2=INDEX2(pointer); % point to the index of this closest member in the set of candidate indices
      pointer2=pointer2+1;      % prepare to point to index of IN_SET2
      u=IN_SET2(pointer2);      % select asociated u from UOD
   end
end
```

Conditional Height Defuzzification Matlab Code

```
function u=fuzact4(alpha,RA,u_prev,IN_SET2,POS,uh,ul,in2)

[a,index]=max(op_tfs(IN_SET2,u_prev,1));      %find closest apex to u(n-1)
%index=index-1;                               %determine index of previous applied control

% decide whether or not to move
```

```
if RA(index)>=alpha*max(RA)
   u=u_prev;                    % if the similarly indexed RA value is within band, don't change control
else

   % now that we've decided to move, where do we go?
   u=RA*IN_SET2(2:in2+1)/sum(RA);
end
```

Conditional Linear-Interpolation Defuzzification Matlab Code

```
% Conditional Linear Interpolation Defuzzification

function u=fuzact5(alpha,RA,u_prev,IN_SET2,POS,uh,ul,in2)

DOM=op_tfs(IN_SET2,u_prev,1);    %find deg. of mem. of u(n-1) in IN_SET2
pos_DOM=find(DOM)                %find index of apexes on either side of u(n-1)

if size(pos_DOM)==[1 1]
   %u=IN_SET2(pos_DOM(1)+1)
   y_prev=RA(pos_DOM(1));

else
   y_prev=RA(pos_DOM(1))+(u_prev-IN_SET2(pos_DOM(1)+1))*(RA(pos_DOM(2))-RA(pos_DOM(1)))/25
end

    % decide whether or not to move
    if y_prev>=alpha*max(RA)
    u=u_prev; % if the similarly indexed RA value is within band, don't change control
    else

    % now that we've decided to move, where do we go?
    u=RA*IN_SET2(2:in2+1)/sum(RA);
    end
```

C.1.4 Matlab Code used in the Implementation of the Post-Processing Algorithm

The following is a printout of the Matlab code used to implement the post-processing algorithm of Chapter 5:

```
% This function patches holes in a fuzzy output of dimension 26-by-2

function patchout2=patchout2(y)

col=2; for row=2:24
   l=y(row,col)-y(row-1,col);
   r=y(row+1,col)-y(row+2,col);
   if l<0 & r<0
     y(row,col)=y(row,col)+abs(l);
     y(row+1,col)=y(row+1,col)+abs(l);
  end
end patchout2=y;
```

C.1.5 Random Training Matlab Code

The following is a printout of the Matlab code used to implement the random training algorithm of Chapter 5:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This algorithm is the random training scheme for the test rig cooling coil. This scheme uses three %
%{\sf HVACSim+} coil models to generate dynamic training data. The data is then used to identify   %
%three R's for each coil model. The nine R's are then MAXed to get the generic R.                  %
%                                                                                                  %
%                                                                                                  %
%                                                                                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
clear

tsamp=30;                          % sampling time in sec.
dur=301500;%15075;                 % duration of training run in sec.
time_const=60;                     % filter time constant

u=zeros(1,10050);                  % initialize normalized control signal
Tai=zeros(1,10050);                % initialize normalized inlet air temperature
ma=zeros(1,10050);                 % initialize normalized air mass low rate

%filter variables
k=7.5;                             % filter gain
a=exp(-tsamp/time_const);          % filter  parameter
p=zeros(1,10050);
q=zeros(1,10050);                  % filter intermediate variables

in1=11;                            % # of sets defined on Tao(n)
in2=5;                             % # of sets defined on u(n-1)
out=26;                            % # of sets defined on Tao(n+1)

PROD=zeros(in1*in2,1);             % PRODuct vector (holds Kronecker tensor products)
COMP=zeros(in1*in2,1);             % COMParison vector (holds maximum activation for each rule combinat
                                   % the elements of COMP tend to 1 during the simulation
COILDATA=0*ones(in1*in2,3);        % Training data matrix [Tao(n) u(n-1) Tao(n+1)]


unnorm_Tao_low=7;                  % min unnormalized outlet air temperature/ deg C
unnorm_Tao_high=30;                % max unnormalized outlet air temperature/ deg C
unnorm_Tao_range=unnorm_Tao_high-unnorm_Tao_low;

unnorm_u_low=0;                    % min unnormalized control valve stem position/%
unnorm_u_high=1;                   % max unnormalized control valve stem position/%
unnorm_u_range=unnorm_u_high-unnorm_u_low;

rawCOILDATA=unnorm_Tao_low*ones(in1*in2,3);
rawCOILDATA(:,2)=unnorm_u_low*ones(in1*in2,1);

IN_SET1=zeros(in1+2,1);            % structure and initialize fuzzy reference sets on Tao
IN_SET2=zeros(in2+2,1);            % structure and initialize fuzzy reference sets on u
```

```
step1=1/(in1-1); for j=-1:in1
    IN_SET1(j+2)=j*step1;                          % fuzzy reference sets for INPUT 1
end

step2=1/(in2-1); for j=-1:in2
    IN_SET2(j+2)=j*step2;                          % fuzzy reference sets for INPUT 2
end


step6=1/(out-1); for j=-1:out
    OUT_SET(j+2)=j*step6;                          % fuzzy reference sets for OUTPUT
end OUT_SET=OUT_SET';

rawTao(1)=unnorm_Tao_high; rawu(1)=unnorm_u_high;


%initialize HVACSIM+ and setup sampling and duration parameters
N=init_hvac4([tsamp dur],[0 30 1],[1 5 7]);                %initialize u to 0, Tai to 30 and ma to 1

%read in raw current outlet air temperature from HVACSim+
rawTao(2)=fread_hvac4([1]);                    %   unormalized current outlet air temperature
Tao(2)=(rawTao(2)-unnorm_Tao_low)/unnorm_Tao_range;    % normalized current outlet air temperature

%
mu_Taon=op_tfs(IN_SET1,Tao(2),1);              % fuzzify normalized current temperature
%
for n=2:N+1
    rand('state',sum(100*clock))               % reset random generator seed
    p(n)=a*p(n-1)+(1-a)*(rand-0.5);
    q(n)=a*q(n-1)+k*((1-a)*p(n));

  u(n)=q(n)+0.5;                               % generate random normalized
                                               %control signal and filter it
  if u(n)>1
     u(n)=1;
  end
  if u(n)<0
     u(n)=0;
```

```
  end

  rawTai(n)=25;
rawma(n)=0.4;
 %
    mu_un1=op_tfs(IN_SET2,u(n-1),1);                    % fuzzify current control signal
 rawu(n)=unnorm_u_range*u(n)+unnorm_u_low;              %convert from normalized to raw values


  fsend_hvac4([rawu(n) rawTai(n) rawma(n)],[1 5 7]);    % send raw crisp control signal,
                                                       %inlet air temperature,air mass
                                                       %flow rate to HVACSim+
  %read in raw 'next' outlet air temperature from HVACSim+
  rawTao(n+1)=fread_hvac4([1]);
                                                       % unormalized 'next' outlet air temperature
  Tao(n+1)=(rawTao(n+1)-unnorm_Tao_low)/unnorm_Tao_range; % normalized 'next' outlet air temperature
  %
  mu_Taon1=op_tfs(OUT_SET,Tao(n+1),1);                 % fuzzify normalized 'next' temperature
 %
 mu_un=op_tfs(IN_SET2,u(n),1);                         % fuzzify previous control signal

 PROD=kron(mu_Taon,mu_un);                             % assumes 1 sample delay on u
 for j=1:in1*in2
    if PROD(j)>COMP(j)
      COMP(j)=PROD(j);
      COILDATA(j,1:3)=[Tao(n) u(n) Tao(n+1)];
      rawCOILDATA(j,1:3)=[rawTao(n) rawu(n) rawTao(n+1)];
   end
 end
 mu_Taon=op_tfs(IN_SET1,Tao(n+1),1);

end


%COMP
training_efficiency=sum(COMP)/(in1*in2)*100

MODDATA=reshape(COILDATA(:,3),in2,in1); for i=1:in1
    for j=1:in2
```

```
            Rcoil(j,i,1:out)=op_tfs(OUT_SET,MODDATA(j,i),1);
        end
end

save ALLCOILDATA30_2 Tao rawTao u rawu Tai rawTai ma rawma
COILDATA rawCOILDATA COMP training_efficiency in1 in2 out IN_SET1
IN_SET2 OUT_SET Rcoil figure(1);clf subplot(2,2,1)
     subplot(2,2,1)
     plot(rawTao,'r-')
     hold on
     xlabel('time / samples')
      ylabel('T_{ao} / ^{o} C')
      grid on;zoom on
     subplot(2,2,3)
     plot(rawu,'g-')
     hold on
     xlabel('time / samples')
     ylabel('control')
     grid on; zoom on
     subplot(2,2,4)
     plot(rawTai ,'m-')
     hold on
     xlabel('time / samples')
      ylabel('T_{ai} / ^{o} C')
     grid on; zoom on
     subplot(2,2,2)
     plot(rawma ,'c-')
     hold on
     xlabel('time / samples')
      ylabel('m_a / %')
     grid on; zoom on
```

C.1.6 Simple Matlab Zone Simulation/Controller used in Chapter 6

The following is a printout of the Matlab code used in the simple zone and NFMBP controller of Chapter 6.

```
clear

global POSSI yhat POSSI=[]; yhat=[];
alpha=0.8 ;                  %fuzzy actuation parameter

load GFc2.dat                    %load comfort goals
load GFe3.dat;                   %load energy goals

number=96;       %number of samples to plot
observe=3;       %sample # to observe more closely

choice = 0; % (1 = max membership for u; defuz. for u otherwise)
k=-0.1; kc=10; delta_k=0;

T=5; Tau=15; delta_Tau=0; Taus=0;

W=zeros(4,2,number+1);

first=32; second=64;

for a=1:first
   W(1:4,1:2,a)=[21.5 1; 21.5 1; 21.5 1; 21.5 1];
end

for a=first+1:second
   W(1:4,1:2,a)=[21.5 0; 21.5 1; 21.5 1; 21.5 0];
end

for a=second+1:number+1
    W(1:4,1:2,a)=[21.5 0; 21.5 1; 21.5 1; 21.5 0];
end W(1:4,1:2,number+2)=W(1:4,1:2,number+1);
```

```
in1=11;                         % # of sets defined on yout(n)
in2=5;                          % # of sets defined on u(n-1)
in3=3;                          % # of sets defined on d(n-1)
out=25;                         % # of sets defined on ym(n+1)

ul=0; uh=100;

yout_low=k*uh;      %since k is neg
yout_high=k*ul; ym_low=yout_low-1; ym_high=yout_high+1;

solgain_max=40;                 % maximum solar gain
intgain=20;                     % internal heat gain

d_low=0; d_high=solgain_max;

FOFALL=[]; YALL=0*ones(5,in2); ZALL=0*ones(5,in2);


flops(0) yout=-5*ones(1,number); yp=23*ones(1,number);
clouds=0*ones(1,number); solgain=0*ones(1,number);
%yhat=0*ones(1,number);
e=zeros(1,number); u=25*ones(1,number); N=zeros(1,number);

a1=exp(-T/Tau); b=k*(1-a1);
a1model=exp(-T/(Tau+delta_Tau*Tau/100));
bmodel=(k+delta_k*k/100)*(1-a1model);

as1=exp(-T/Taus); bs=1-as1;

IN_SET1=zeros(in1+2,1); IN_SET2=zeros(in2+2,1);
IN_SET3=zeros(in3+2,1);

step1=(yout_high-yout_low)/(in1-1); for j=-1:in1
    IN_SET1(j+2)=yout_low+j*step1;
end

step2=(uh-ul)/(in2-1); for j=-1:in2
    IN_SET2(j+2)=ul+j*step2;
end
```

```
step3=(ym_high-ym_low)/(out-1); for j=-1:out
    OUT_SET(j+2)=ym_low+j*step3;
end

step4=(d_high-d_low)/(in3-1); for j=-1:in3
    IN_SET3(j+2)=d_low+j*step4;
end

OUT_SET=OUT_SET';

j2=1; for j5=2:in3+1 for j1=1:in1
    DATA(j2:j2+in2-1,1)=IN_SET1(j1+1)*ones(in2,1);
    j2=j2+in2;
end end

j4=1; for j5=2:in3+1 for j3=1:in1
    DATA(j4:j4+in2-1,2)=IN_SET2(2:in2+1);
    j4=j4+in2;
end end

%p=0*zeros(1,number);
%q=0*zeros(1,number);

j4=1; for j5=2:in3+1 for j3=1:in1*in2
    DATA(j4:j4+in1*in2-1,3)=IN_SET3(j5);
end j4=j4+in1*in2; end

DATA(1:in1*in2*in3,4)=a1model*DATA(1:in1*in2*in3,1)+bmodel*(DATA(1:in1*in2*in3,2)-DATA(1:in1*in2*in3,3)-intgain);

%OUT_SET=[-12:1:2]';
ym1=0*ones(size(OUT_SET,1)-2,2,number); ym2=ym1;

ise=0;

MODDATA=reshape(DATA(:,4),in2,in1,in3); for i=1:in1
    for j=1:in2
        for m=1:in3

        R(j,i,m,1:out)=op_tfs(OUT_SET,MODDATA(j,i,m),1);
```

```
        end
    end
end

ise=0; POS=IN_SET2(2:in2+1); tic
rand('state',7)     %initialize state of random number generator so that same sequence is always generated
for n=3:number
   if n > 11 & n < 85
        clouds(n)=kc*(rand-0.5);
        solgain(n)=as1*solgain(n-1)+bs*(solgain_max*sin(2*pi*(n-12)/144)+clouds(n));
        if solgain(n)>50
            solgain(n)=50;
        elseif solgain(n)<0
            solgain(n)=0;
        end


    else
        solgain(n)=0;
   end
end solgain(number+1)=solgain(number); for n=3:number
    N(n)=n;
    yout(n)=a1*yout(n-1)+b*(u(n-2)+(-solgain(n)-intgain));
        yp(n)=24+yout(n);
        FOFc(50,2,in2)=zeros;
        FOFe(in2,2,in2)=zeros;
        for x=1:1:in2;
            uprime=((uh-ul)*(x-1))/(in2-1) +ul;

ym1(1:out,1:2,n+1)=fuzinf92(yout(n),u(n-1), solgain(n),IN_SET1,IN_SET2,IN_SET3,out,OUT_SET,R,n,observe,in1,in2,in3);
% set up ym as a 3-D matrix with # of rows
% equal to # of output sets, 2 columns (1 for universe and 1 for membership) and # of pages equal to # of samples.
YM1=ym1(:,:,n+1);


ym2(1:out,1:2,n+2)=fuzinf99(ym1(:,:,n+1),uprime,solgain(n+1),IN_SET1,IN_SET2,IN_SET3,out,OUT_SET,R,n,observe,in1,in2,i

YM2=ym2(:,:,n+2);
 control=[[0:100/(in2-1):(in2-1)*100/(in2-1)]' zeros(in2,1)];
 control(x,2)=1;
```

```
proximity1 = [[0.0     1;
     0.5     0;
     1.0     0;
     1.5     0;
     2.0     0;
     2.5     0;
     3.0     0;
     3.5     0;
     4.0     0;
     4.5     0;
     5.0     0;
     5.5     0;
     6.0     0;
     6.5     0;
     7.0     0;
     7.5     0;
     8.0     0;
     8.5     0;
     9.0     0;
     9.5     0]];

    proximity2=fp5(W(:,:,n+2),ym2(:,:,n+2));

    FOFc(:,:,x)=fadd92(proximity1,proximity2);
    FOFe(:,:,x)=control;                % !!! MUST have same dicretization as GFe3 !!!


 end

 TAc(n,1:in2)=sum(FOFc(:,2,1:in2));
 TAe(n,1:in2)=sum(FOFe(:,2,1:in2));
  if n>=36 & n<=60
     rowc(n)=2;                              %start at row # 1 in GF
  rowe(n)=4;
  else
     rowc(n)=1;
     rowe(n)=16;
end
```

```
 initial_rowc(n)=rowc(n);
initial_rowe(n)=rowe(n);

rel_import=initial_rowe(n)/initial_rowc(n);

RAc(n,1:in2)=zeros(1,in2);
 RAe(n,1:in2)=zeros(1,in2);
 RA(n,1:in2)=zeros(1,in2);

 while isempty(find(RA(n,:)~=0))

     for x=1:in2
         CAc(n,x)=sum(min(GFc2(rowc(n),:),FOFc(:,2,x)'));
         CAe(n,x)=sum(min(GFe3(rowe(n),:),FOFe(:,2,x)'));
     end

     CA1c=CAc(n,TAc(n,:)>0);
     TA1c=TAc(n,TAc(n,:)>0);
     CA1e=CAe(n,TAe(n,:)>0);
     TA1e=TAe(n,TAe(n,:)>0);

     RAc(n,:)=CA1c./TA1c;
     RAe(n,:)=CA1e./TA1e;
     RA(n,:)=RAc(n,:).*RAe(n,:);                    % product as fuzzy AND

     rowc(n)=rowc(n)+1;
   % rowe(n)=rowe(n)+1;
   rowe(n)=rel_import*(rowc(n)+1);
     [rowc(n)-1,rowe(n)-1,n];
 end

 u(n)=fuzactht(alpha, RA(n,:),u(n-1),IN_SET2,POS,uh,ul,in2);
 du(n)=u(n)-u(n-1);
end

flops=flops time_min_sec=[floor(toc/60)
round(60*(toc/60-floor(toc/60)))]
%energy_consumption=round(number/12*sum(u)/number)
energy_consumption=60.30;%(1/96)*sum(u)
```

```
activity=3.80;%(1/95)*(sum(abs(du)))            % since 1st two values are always zero
initial_rowc(n) initial_rowe(n) alpha N(1)=1; N(2)=2;

figure(1);clf subplot(2,1,1);plot(900+100*N/12,u,'b-');axis([900
900+100*number/12 0 100]);grid on;zoom on c=gca;
set(c,'XTick',[900:100:100*(9+number/12)]); %will only put ticks on y axis for values btw. 1 and 24
set(c,'YTick',[0:10:100]);          %will only put ticks on y axis for values btw. 0 and 100
xlabel('Time of Day / HR:00')
ylabel('Air Flow / %')
title(['  Energy Consumption = ',num2str(energy_consumption), '


subplot(2,1,2)
REGION=[W(3,1,n)+1.0 W(3,1,n)+1.0];     %comfort region

colormap([.8 .8 .8]) area([900 900+100*number/12],REGION,
W(2,1,n)-1.0);hold on

plot([900 1200],[W(3,1,n) W(3,1,n)]+[initial_rowc(4)/2+1.0
initial_rowc(4)/2+1.0],'g*-');hold on plot([900 1200],[W(3,1,n)
W(3,1,n)]-[initial_rowc(4)/2+1.0
initial_rowc(4)/2+1.0],'g*-');hold on

plot([1200  1400],[W(3,1,n) W(3,1,n)]+[initial_rowc(40)/2+1.0
initial_rowc(40)/2+1.0],'g*-');hold on plot([1200  1400],[W(3,1,n)
W(3,1,n)]-[initial_rowc(40)/2+1.0
initial_rowc(40)/2+1.0],'*g-');hold on

plot([1400 1700],[W(3,1,n) W(3,1,n)]+[initial_rowc(90)/2+1.0
initial_rowc(90)/2+1.0],'g*-');hold on plot([1400 1700],[W(3,1,n)
W(3,1,n)]-[initial_rowc(90)/2+1.0
initial_rowc(90)/2+1.0],'g*-');hold on

plot(900+100*N/12,yp,'r-') axis([900 900+100*number/12 17 28])
d=gca;
set(d,'XTick',[900:100:100*(9+number/12)]);                %will only put ticks on y axis for values btw. 1 and 24
set(d,'YTick',[17:28]); grid on zoom on xlabel('Time of Day /
HR:00') ylabel('Zone Temperature / ^{o}C')
%print -deps ../RTthesis/figs/thdplot.eps
```

Appendix D

Simulink Zone Model of Thermal Behaviour of Zone

D.1 Four High-Level Schematics of Simulink Zone Model

In this appendix, the four levels above the schematic of Figure 4.2 (Section 4.3) are given. The highest level (level 1) is shown first in Figure D.1 and represents the overall building control system. Next, the middle level (level 2) is given in Figure D.2. This shows the entire six zones of the complete model, along with the controllers used to control each zone. Only zone 3 (and, therefore, controller 3) of this system is used in Chapter 4. Level 3 shown in Figure D.3 is a look under the block diagram labeled "Zone 3" of Figure D.2. Finally, level 4 shown in Figure D.4 represents the penultimate level for the purposes of this work. It is a more detailed look at the systems contributing to level 3.

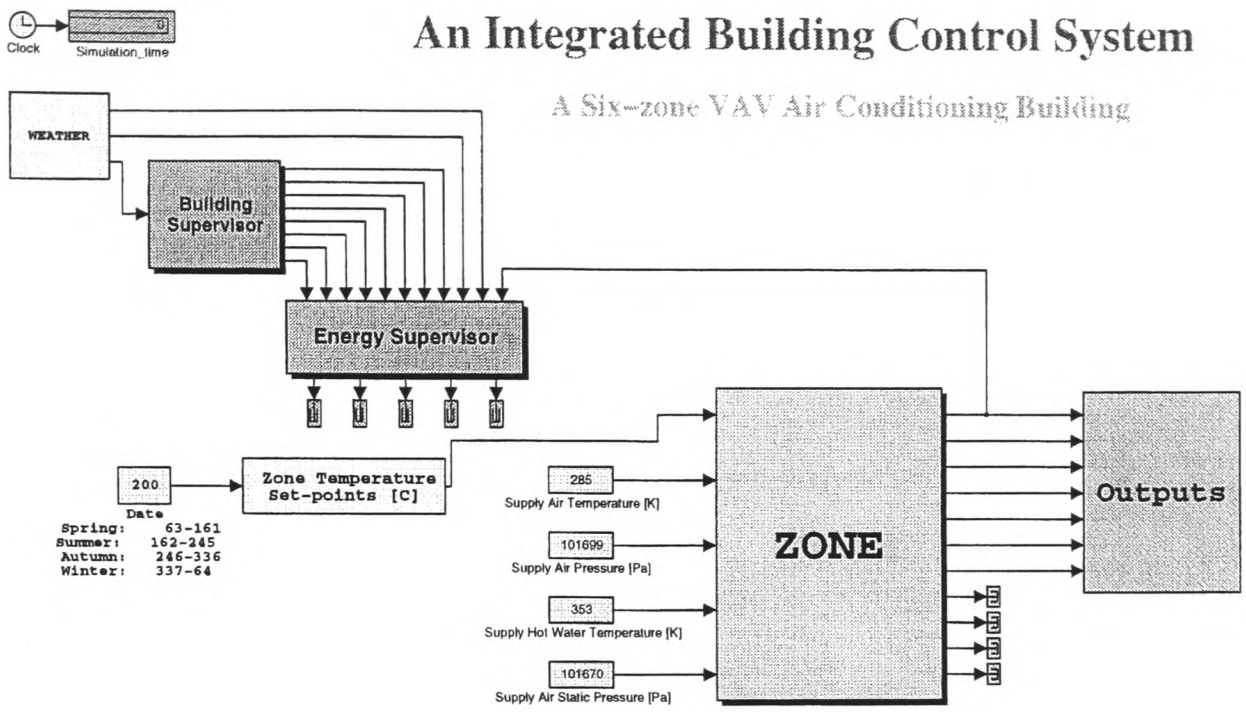


Figure D.1: Simulink Model: Upper Level (level 1) of Zone Model

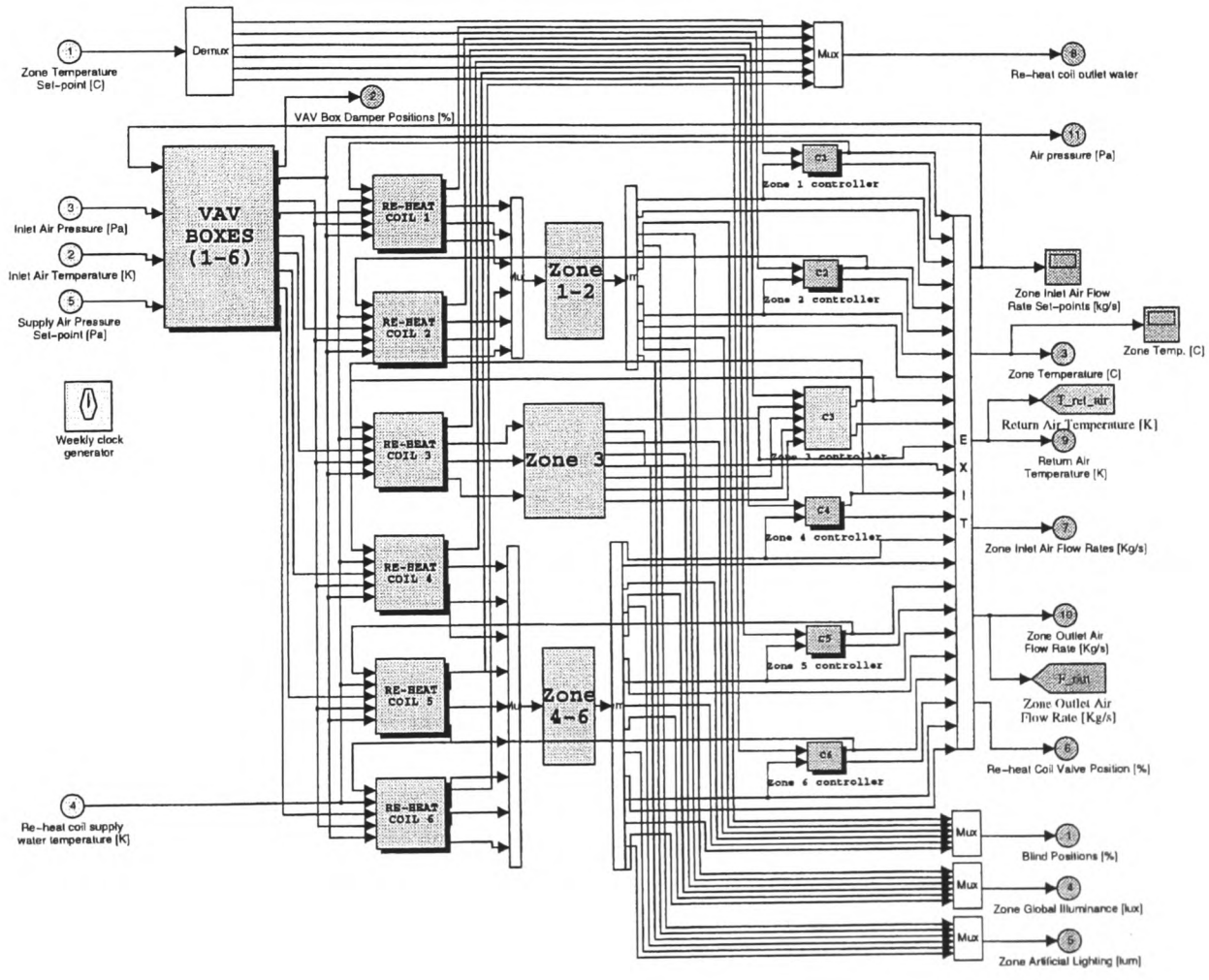


Figure D.2: Simulink Model: Middle Level (level 2) of Zone Model. Zone 3 is the one of interest in this work.

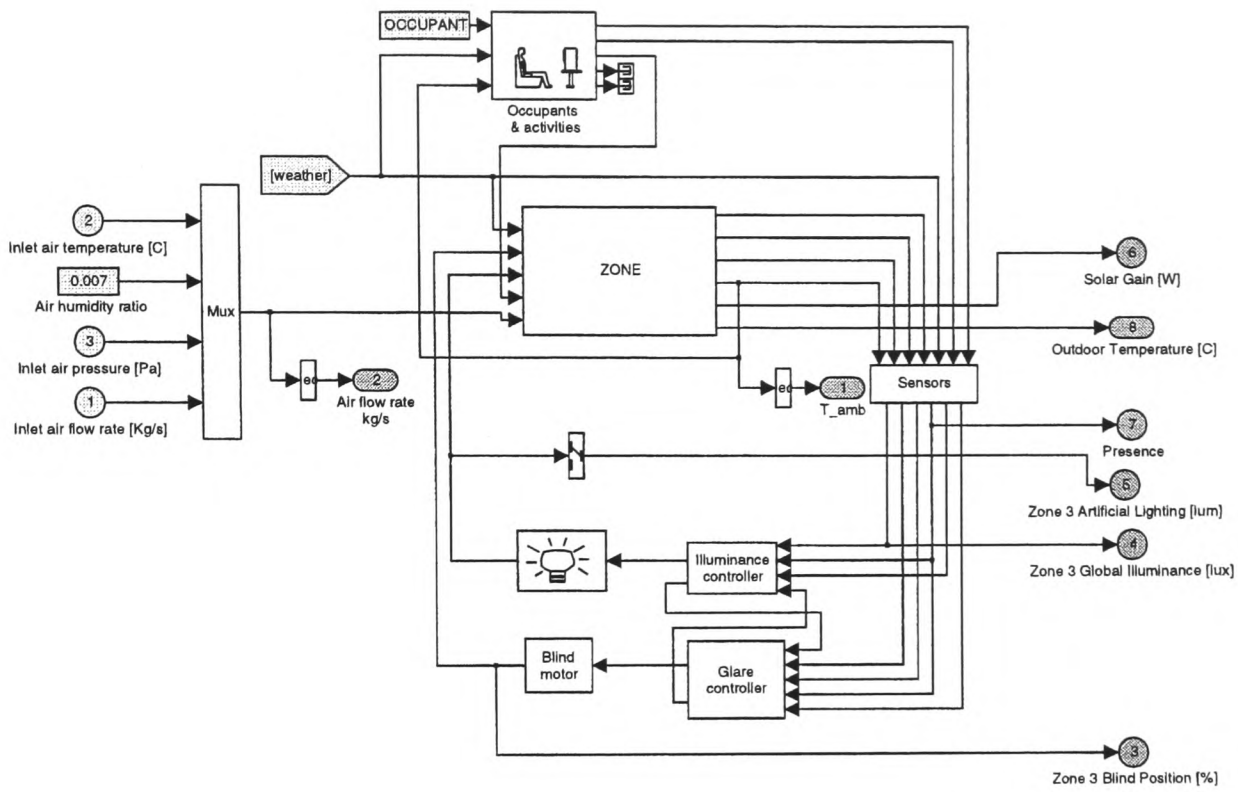


Figure D.3: Simulink Model: Lower Level (level 3) of Zone Model

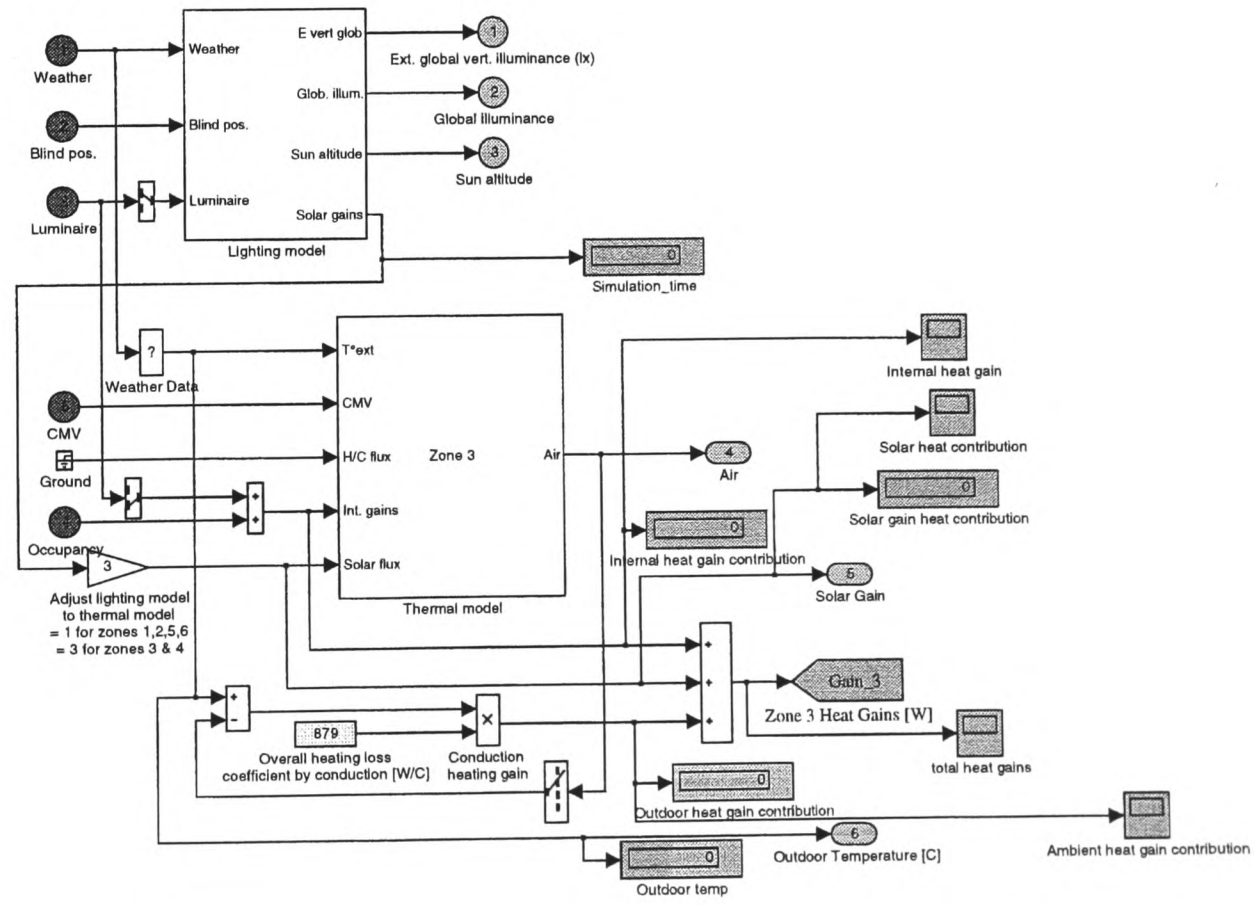


Figure D.4: Simulink Model: Penultimate (Sub) Level (level 4) of Zone Model

References

ASHRAE Handbooks: Fundamentals

Astrom, K.J., Wittenmark, B., (1989), Adaptive Control (1st Edition), pp. 198-208, Addison-Wesley, 1989.

Babuska, R., Sousa, J., Verbruggen, H., B., (1995), Model-Based Design of Fuzzy Control Systems, Proceedings EUFIT' 95, August 28-31, Aachen, Germany, 1995.

Batur, C., Srinivasan, A., Chan, C.C., (1995), Fuzzy Model Based Fuzzy Predictive Controllers, Journal of Intelligent and Fuzzy systems, Vol. 3, pp. 117-130, 1995.

Bellman, R.E., Zadeh, L.A., (1970), Decision-Making in a Fuzzy Environment, Management Science, Vol. 17, No. 4, pp. 141-164, 1970.

Bezdek, J., (1993), Editorial: Fuzzy Models-What are They, and Why?, IEEE Transactions on Fuzzy Systems, Vol. 1, No. 1, February 1993.

Bourdouxhe, J.P., Seutin H., (1998), Fault Detection and Diagnosis of an Dir-Handling Unit of a Big Office Building, Proc. SSB'98, Paper P44, Liege, Belgium, 1998.

Bremner, H., Postlethwaite, B., (1997), An Application of Model-Based Fuzzy Control

to an Industrial Grain Dryer, Trans. Inst. Measurement and Control, 19(4), pp. 185-191, 1997.

Bruant, M., Guarracino, G., Michel, P., (1994), Design and Tuning of a Fuzzy Controller for Indoor Air Quality and Thermal Comfort Management, International Journal of Solar Energy Vol.21, no.2-3 pp. 81-109, 1994.

Cao, S., Rees, N., Feng, G., (1997), Analysis and Design for a Class of Complex Control Systems Part I: Fuzzy Modelling and Identification, Automatica, Vol. 33, No. 6, pp. 1017-1028, 1997.

Clarke D.W., (1985), PID Algorithms and their Computer Application, Trans. Inst. Measurement and Control, 6(6), pp. 305-316, 1985.

Clarke, D.W., Mohtadi, C., Tuffs, P.S., (1987),Generalized Predictive Control - Part I. The Basic Algorithm, Automatica, Vol. 23, No. 2, pp. 137-148, 1987.

Conte, E., Fato, I., (1997), The Fuzzy Set Theory for Analysing Thermal Comfort, CD-ROM of Proc. Clima 2000, Brussels, Paper No. 104, 1997.

Curtiss, P.S, Kreider, J.F., Brandemeuhl, M.J., (1993), Adaptive Control of HVAC Processes using Predictive Neural Networks, ASHRAE Trans. 99(1), 1993.

Curtiss, P.S., Kreider, J.F., Brandemuehl, M.J., (1994), Energy Management in Central HVAC Plants using Neural Networks, Trans. ASHRAE 100(1), 1994.

De Keyser, R.M.C., (1985), Adaptive Microcomputer Control of Residence Heating,

CIBOW '97: Recent Advances in the Control and Operations of Building HVAC Systems. SINTEF, Trondheim, pp. 154-164, 1985.

Dexter, A.L., Benouarets, M., (1997), Model-based Fault Diagnosis using Fuzzy Matching, Trans. IEEE on Systems, Man, and Cybernetics - Part A, Vol. 27, No 5, pp. 673-682, 1997.

Dexter, A.L., Haves, P., (1989), A Robust Self-Tuning Controller for HVAC Applications, ASHRAE Trans., Vol. 95, Pt. 2, pp. 431-438, 1989.

Dounis, A.I., et al., (1994), Thermal Comfort Degradation by a Visual Comfort Fuzzy-Reasoning Machine under Natural Ventilation, Applied Energy, Vol. 48, No. 4, 1994.

Driankov, D., Hellendoorn, H., Reinfrank, M, (1996), An Introduction to Fuzzy Control, pp. 103-144, Springer-Verlag: New York, 1996.

Fanger, P.O., (1982), Thermal Comfort Analysis and Applications in Environmental Engineering, ed. Robert E. Krieger, Publishing Co., Melbourne, FL, pp. 143-148, 1982.

Funakoshi, S., Matsuo, K., (1995), PMV-Based Train Air Conditioning Control System, for inclusion in ASHRAE Transactions, Vol. 101, Pt. 1, 1995.

Geng, G., Dexter, A.L., (1990), Fuzzy Gain-Scheduling Control of a Non-linear HVAC System, 28th Annual Allerton Conference on Communication, Control and Computing, University of Illinois at Urbana-Champaign, October 3-5, 1990.

Glorennec, P., (1991), Application of Fuzzy Control for Building Energy Management, Proc. of the 2nd Int. Building Performance Simulation Association (IBPSA) Conference, Nice, France, 1991.

Gouda, M.M., Danaher, S., Underwood, C.P., (2000), Fuzzy Logic Control Versus Conventional PID Controller for Controlling Indoor Temperature of a Building Space, CASD 2000 Conference, Salford University, UK, 2000.

Gouda, M.M., Danaher, S. and Underwood, C.P., (2001), Thermal Comfort Based Fuzzy Logic Controller, Int. J of BSER and T, 22(4), pp 237-253, 2001.

Grace, A., (1992), Optimization Toolbox for Use with Matlab, pp. 3:9-3:13, 1992.

Graham, B.P., Newell, R.B., (1989), Fuzzy Adaptive Control of a First Order Process, Fuzzy Sets and Systems 31 (1989), pp. 47-65, 1989.

Hao, J., Vandewalle, J., Tan, S., (1994), Predictive Control of Nonlinear Systems Based on Identification by Back Propagation Networks, International Journal of Neural Systems, Vol. 5, No. 4 (December 1994), pp. 335-344, 1994.

Howell, J., (1994), Model-Based Fault Detection in Information Poor Plants, Automatica, Vol. 30, No. 6, pp. 929-943, 1994.

Huang, S., Nelson, R., (1994), Rule Development and Adjustment Strategies for a Fuzzy Logic Controller for an HVAC System: Part One-Analysis; Part Two-Experiment, ASHRAE Transactions, Vol. 100, Pt. 1, 1994.

Huang, S., Nelson, R., (1999), Development of a Self-Tuning Fuzzy Logic Controller, Trans. ASHRAE 105(1), 1999.

Hunt, K., Sbarbaro, D., (1991), Neural Networks for Internal Model Control, IEE Proceedings-D, Vol. 138, No. 5, September, 1991.

Husaunndee, A., (2001), *Modelisation Du Batiment (Enveloppe), Energie, Environnement Interieur Et Automatismes*, 2001.

Isermann, R., (1995), On Fuzzy Logic Applications for Automatic Control, Supervision and Fault diagnosis, Third European Congress on Fuzzy and Intelligent Technologies, EUFIT '95, Aachen, Germany, Aug. 28-31, 1995.

Jager, R., (1995), Fuzzy Logic in Control, PhD Thesis, Delft University, 1995.

Karr, C. L., Gentry, E.J., (1993),Fuzzy Control of pH Using Genetic Algorithms, IEEE Transactions on Fuzzy Systems, Vol. 1, No. 1, February 1993.

Kaymak, U., Sousa, J.M., Verbruggen, H.B., (1996), Influence of Decision Functions in Fuzzy Predictive Control, EUFIT '96, September 2-5, 1996.

Klir, G.J., Yuan, B., (1995), Fuzzy Sets and Fuzzy Logic, Prentice Hall, 1995.

Kummert, M., Andre, P., Nicolas, J., (1997), Optimised Thermal Zone Controller for Integration within a Building Energy Management System, CD-ROM of Proc. Clima 2000, Brussels, Paper No. 329, 1997.

Lee, K.M., Scong, K.A., Kwang, H., (1992), Fuzzy Matching and Fuzzy Comparisonin Fuzzy Expert Systems, Proceedings of the 2nd International Conference on Fuzzy Logic and Neural Networks, Japan, July 17-22, pp. 313-316, 1992.

Leigh, J.R., (1992), Applied Digital Control, pp 452-490, Prentice Hall, 1992.

Ling, K.V., Dexter, A.L., (1994), Expert Control of Air-conditioning Plant, Automatica, Vol. 30 , No. 5, pp. 761-773, 1994.

Mamdani, E., (1974), Applications of Fuzzy Algorithms for Simple Dynamic Plant, IEE Proceedings, Vol. 121, No. 3, pp. 1585-1588, 1974.

Mamdani, E., Assilian, S., (1975), An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller, International Journal of Man, Machine Studies, Vol. 7, No. 1, pp. 1-13, 1975.

Mauris, G., Benoit, E., Foulloy, L., (1997), Fuzzy Sensors: An Overview, Fuzzy Information Engineering, pp. 1-30, John Wiley and Sons, Inc., (Ed. Dubois et al.), 1997.

Nesler, C.G., (1986), Adaptive Control of Thermal Processes in Buildings, IEEE Control Systems Mag., August, 9-13, 1986.

Norman, P., Naveed, S., (1985), An Expert System Supervisor for a Rotary Cement Kiln, Proceedings of IEEE Symposium on Expert Systems in Process Control, University of Salford, 1985.

Park, C., Clark, D.R., Kelly, G.E., (1985), An Overview of HVACSIM+, A Dynamic Building/HVAC Control Systems Simulation Program, Proc. Energy Building Simulation Conference, Seattle, Washington, US Department of Energy, pp. 175-185, Aug., 1985.

Pedrycz, W., (1984), An Identification Algorithm in Fuzzy Relational Systems, Fuzzy Sets and Systems, Vol. 13, pp. 153-167, 1984.

Peitsman, H.C., Soethout L.L., (1997), ARX Models and Real-Time Model-Based Diagnosis, Trans. ASHRAE 103(1), 1997.

Postlethwaite, B., (1994), A Model-Based Fuzzy Controller, Transactions IChemE, Vol. 72, Part A, pp. 39-41, 1994.

Radke, F., Isermann, R., (1987), A Parameter-Adaptive PID-Controller with Stepwise Parameter Optimization, Automatica, Vol. 23, pp. 449-457, 1987.

Ridley, J.N., Shaw, I.S., Kruger, J.J., (1988), Probabilistic Fuzzy Model for Dynamic Systems, Electronics Letters July 7, Vol. 24, No. 14, 1988.

Salsbury, T.I., (1998), A Temperature Controller for VAV Air-Handling Units Based on Simplified Physical Models, IJHVAC&R Research 4(3), pp. 265-302, 1998.

Sing, C.H., Postlethwaite, B., (1996), Fuzzy Relational Model-Based Control Applying Stochastic and Iterative Methods for Model Identification, Transactions IChemE, Vol. 74, Part A, pp. 70-71, 1996.

Sing, C.H., Postlethwite, B., (1997), pH Control: Handling Nonlinearity and Dead-time with Fuzzy Relational Model-Based Control, IEE Proc.-Control Theory Appl., Vol.144, No. 3, May 1997.

So, A.T.P., Chan, W.L., Tse, W.L., (1997), Self-Learning Fuzzy Air Handling System Controller, Proc. CIBSE A: Building Serv. Eng. Res. Technol., Vol. 18(2), pp. 99-108, 1997.

Sousa, J.M., (1998), A Fuzzy Approach to Model-Based Control, pp. 89-188, ISBN:90-9011506-4, 1998.

Sousa, J.M., Babuska, R., Verbruggen H.B., (1997), Fuzzy Predictive Control Applied to an Air-Conditioning System, Control Engineering Practice 5(10), pp. 1395-1406, 1997.

Sugeno, M., (1985), Industrial Applications of Fuzzy Control, Elsevier (North-Holland): Amsterdam, 1985.

Takagi, T., Sugeno, M., (1995), Fuzzy Identification of Systems and its Application to Modeling and Control, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 15, No.1 pp. 116-132, 1995.

Tan, W.W., (1997), Self-learning Neurofuzzy Control of Non-linear Systems, PhD. Thesis, University of Oxford, 1997.

Tan, W.W., Dexter, A.L., (1996), Properties of a Fuzzy Relational Model, University of Oxford, Department of Engineering Science, Report No. OUEL 2119/96,

1996.

Terano, T., Asai, K., Sugeno, M., (1994), Applied Fuzzy Systems, Academic Press: Boston, 1994.

Thompson, R., Dexter, A.L., (2001), Thermal Comfort Control Based on Fuzzy Decision Making, Intelligent Control, European Society for Fuzzy Logic and Technology (EUSFLAT) 2001, Leicester, UK, Sept. 5 -7, 2001, paper 121, 2001.

Virk, V.S., Cheung, J.M., Loveday, D.L., (1991), The Development of Adaptive Control Techniques for BEMS, IEE Conference, Control 91, Edinburgh, pp. 329-334, 1991.

Wallenborg, A.O., (1991), A New Self-Tuning Controller for HVAC Systems, ASHRAE Trans., Vol. 97, Pt. 1, pp. 19-25, 1991.

Wang, L.X., Mendel, J.M., (1992), Generating Fuzzy Rules by Learning from Examples, Transactions on Systems, Man, and Cybernetics, Vol. 22, No.6, pp. 1414-1427, 1992.

Wang, Y.G., Shi, Z.G., Cai, W.J., (2001), PID Autotuner and its Application in HVAC Systems, Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148) Part vol.3 pp. 2192-6 vol.3, 2001.

Willis, M., Massimo, C., Montague, G., Tham, M., Morris, A., (1991), Artificial Neural Networks in Process Engineering, IEE Proceedings-D, Vol. 138, No. 3, May 1991.

Xu, C.W., Lu, Y.Z., (1987), Fuzzy Model Identification and Self-Learning for Dynamic Systems, Transactions on Systems, Man, and Cybernetics, Vol. SMC-17, No. 4, July/August 1987.

Yi, S., Chung, M., (1993), Identification of Fuzzy Relational Model and its Application to Control, Fuzzy Sets and Systems 59, pp. 25-33, 1993.