Fuzzy Model Based Predictive Control of Chemical Processes

Sivasothy Kandiah, B E (Hons), M Eng Sc

Thesis submitted as a requirement for the degree of
Doctor of Philosophy
in the Department of Automatic Control and Systems Engineering
University of Sheffield
July 1996

This thesis is dedicated to the memory of my brother

SIVASEGARAN KANDIAH
(1940 -1994)

ACKNOWLEDGEMENTS

Working for my PhD qualification has been a profound learning experience for me, in more ways than I can put down in a few words, and I wish that I could convey a sense of my excitement as I complete the writing of this thesis. I wish to take this opportunity to thank everyone who has contributed, directly or indirectly, to the fulfilment of my ambition.

When I made my decision to do my PhD study in the field of intelligent control, I had very little knowledge of the field to enable me to screen through research programmes of various universities. I consider myself very fortunate, not only for the opportunity to have worked in a new and exciting area of control engineering, destined to play a very significant role in the years ahead, but also for the opportunity to have worked with Professor Derek Linkens, whose knowledge and experience supervising research students were essential in ensuring the timely completion of this study. Considerable credit for the research achievements reported in this thesis should go to Professor Linkens for his very able guidance of all phases of the study.

I am also grateful to the Government of Malaysia and to my company, Palm Oil Research Institute of Malaysia (PORIM), for the scholarship and for allowing me to take study leave on full pay over a considerable period of time to complete my studies.

I am deeply indebted to my family, whose support, encouragement and understanding made it possible for me to focus my full attention, at a stage in my life when it should have been very difficult, on the task of completing my studies. I hope that my achievement is well-worth their sacrifices during the three years that I have been away in United Kingdom.

I also wish to thank fellow research students and research assistants of the Intelligent Control Group for the enlightening discussions which helped in the formulation of many of the ideas presented in this thesis. I particularly wish to direct my thanks to Dr. M. Mahfouf for sharing his insights on Generalised Predictive Control (GPC), and to Dr. M.F. Abbod and Dr. J.S. Shieh for their many helpful suggestions during the early phases of this study.

FUZZY
MODEL BASED PREDICTIVE CONTROL
OF
CHEMICAL PROCESSES

Sivasothy Kandiah

ABSTRACT

The past few years have witnessed a rapid growth in the use of fuzzy logic controllers for the control of processes which are complex and ill-defined. These control systems, inspired by the approximate reasoning capabilities of humans under conditions of uncertainty and imprecision, consist of linguistic '*if-then*' rules which depend on fuzzy set theory for representation and evaluation using computers. Even though the fuzzy rules can be built from purely heuristic knowledge such as a human operator's control strategy, a number of difficulties face the designer of such systems. For any reasonably complex chemical process, the number of rules required to ensure adequate control in all operating regions may be extremely large. Eliciting all of these rules and ensuring their consistency and completeness can be a daunting task.

An alternative to modelling the operator's response is to model the process and then to incorporate the process model into some sort of model-based control scheme. The concept of Model Based Predictive Control (MBPC) has been heralded as one of the most significant control developments in recent years. It is now widely used in the chemical and petrochemical industry and it continues to attract a considerable amount of research. Its popularity can be attributed to its many remarkable features and its open methodology. The wide range of choice of model structures, prediction horizon and optimisation criteria allows the control designer to easily tailor MBPC to his application. Features sought from such controllers include better performance, ease of tuning, greater robustness, ability to handle process constraints, dead time compensation and the ability to control nonminimum phase and open loop unstable processes. The concept of MBPC is not restricted to single-input single-output (SISO) processes. Feedforward action can be introduced easily for compensation of measurable disturbances and the use of state-space model formulation allows the approach to be generalised easily to multi-input multi-output (MIMO) systems. Although many different MBPC schemes have emerged, linear process models derived from input-output data are often used either explicitly to predict future process behaviour and/or implicitly to calculate the control action even though many chemical processes exhibit nonlinear process behaviour. It is well-recognised that the inherent nonlinearity of many chemical processes presents a challenging control problem, especially where quality and/or economic performance are important demands.

In this thesis, MBPC is incorporated into a nonlinear fuzzy modelling framework. Even though a control algorithm based on a 1-step ahead predictive control strategy has initially been examined, subsequent studies focus on determining the optimal controller output using a long-range predictive control strategy. The fuzzy modelling method proposed by Takagi and Sugeno has been used throughout the thesis. This modelling method uses fuzzy inference to combine the outputs of a number of auto-regressive linear sub-models to construct an overall nonlinear process model. The method provides a more compact model (hence requiring less computations) than fuzzy modelling methods using relational arrays. It also provides an improvement in modelling accuracy and effectively overcomes the problems arising from incomplete models that characterise relational fuzzy models.

Difficulties in using traditional cost function and optimisation techniques with fuzzy models have led other researchers to use numerical search techniques for determining the controller output. The emphasis in this thesis has been on computationally efficient analytically derived control algorithms. The performance of the proposed control system is examined using simulations of the liquid level in a tank, a continuous stirred tank reactor (CSTR) system, a binary distillation column and a forced circulation evaporator system. The results demonstrate the ability of the proposed system to outperform more traditional control systems. The results also show that inspite of the greatly reduced computational requirement of our proposed controller, it is possible to equal or better the performance of some of the other fuzzy model based control systems that have been proposed in the literature.

It is also shown in this thesis that the proposed control algorithm can be easily extended to address the requirements of time-varying processes and processes requiring compensation for disturbance inputs and dead times. The application of the control system to multivariable processes and the ability to incorporate explicit constraints in the optimisation process are also demonstrated.

CONTENTS

ACKNOWLEDGEMENTS i

ABSTRACT ii

CONTENTS iv

1 INTRODUCTION

1.1 Background 1

1.2 Some Current Problems and Needs 4

1.3 Control Techniques Applied in Industry 5

1.4 Model Based Predictive Control 6

1.4.1 General 6

1.4.2 The Predictive Control Concept 7

1.4.3 Linear Process Models 10

1.4.4 Nonlinear Model Based Predictive Control 11

1.5 Intelligent Modelling and Control 13

1.5.1 General 13

1.5.2 Artificial Neural Networks 15

1.5.3 Fuzzy Logic 19

1.5.4 Knowledge-Based Systems 21

1.6 Motivations for the Current Research 22

1.7 Scope of this Work 23

1.8 Thesis Overview 24

2 FUZZY MODELLING

2.1 Introduction 29

2.3 Basic Concepts and Definitions 30

2.2.1 Fuzzy Set 30

2.2.2 Linguistic Variable 31
2.2.3 Set-Theoretic Operation 31
2.2.4 Approximate Reasoning 32
2.3 Fuzzy Logic System 33
2.3.1 Fuzzy Rule Base 34
2.3.2 Fuzzy Inference Engine 35
2.3.3 Fuzzifier 35
2.3.4 Defuzzifier 36
2.3.5 Some Specific Fuzzy Logic Systems 36
2.4 Relational Fuzzy Model 38
2.4.1 Pedrycz Method 39
2.4.2 Xu and Lu Method 40
2.4.3 Graham and Newell Method 41
2.4.4 Ridley, Shaw and Kruger Method 41
2.4.5 Lee, Hwang and Shih Method 42
2.5 Piece-wise Linear Fuzzy Model 43
2.6 Comparison of Fuzzy Modelling Methods 45

3 FUZZY MODEL BASED CONTROL
3.1 Introduction 49
3.2 Graham and Newell Method 51
3.3 Postlethwaite Method 52
3.4 Skrjanc and Matko Method 52
3.5 Other Fuzzy Model Based Controllers 54
3.6 Comparison of Some Fuzzy Model Based Controllers 55

4 IDENTIFICATION OF FUZZY PROCESS MODEL
4.1 Introduction 58
4.2 Fuzzy Process Model 59
4.3 Liquid Level System 61
4.3.1 Mathematical Model 61
4.3.2 Identification of Fuzzy Process Model 63
4.4 Continuous Stirred Tank Reactor System 70

4.4.1 Mathematical Model 70

4.4.2 Identification of Fuzzy Process Model 72

4.5 Conclusions 81

5 ONE-STEP AHEAD PREDICTIVE CONTROLLER

5.1 Introduction 82

5.2 Transformation of Nonlinear Fuzzy Process Model into Linear Model 83

5.3 Controller Formulation 84

5.4 Application to Control of Liquid Level 86

5.4.1 Simulation Results 86

5.4.2 Comparison of Fuzzy Model Based Controllers 92

5.5 Application to Control of CSTR 93

5.6 Conclusions 99

6 LONG-RANGE PREDICTIVE CONTROL: NUMERICAL APPROACH

6.1 Introduction 100

6.2 Making Multi-Step Predictions Using the Fuzzy Process Model 101

6.3 Controller Formulation 104

6.4 Numerical Optimisation Using Fibonacci Search 108

6.5 Application to Control of Liquid Level 111

6.6 Conclusions 114

7 LONG-RANGE PREDICTIVE CONTROL: ANALYTICAL APPROACH

7.1 Introduction 115

7.2 Linearisation By Weighting Fuzzy Model Parameters 116

7.3 Controller Formulation 121

7.3.1 First-Order System 125

7.3.2 Second-Order System 126

7.4 Application to Control of Liquid Level 127

7.5 Application to Control of CSTR 134

7.6 Numerical Approach versus Analytical Approach 142

7.7 Robustness Analysis 143
7.8 Conclusions 148

8 CONTROL OF MULTI-INPUT SINGLE-OUTPUT SYSTEMS
8.1 Introduction 149
8.2 Model Predictions Using MISO Models 150
8.2.1 First-Order MISO System 150
8.2.2 Second-Order MISO System 151
8.3 Controller Formulation 152
8.4 Application to Control of CSTR 153
8.4.1 Identification of MISO Process Models 153
8.4.2 MISO Process Model Based Controllers 162
8.5 Conclusions 162

9 CONTROL OF MULTIVARIABLE SYSTEMS: BINARY DISTILLATION COLUMN
9.1 Introduction 163
9.2 Mathematical Model of Binary Distillation Column 165
9.3 Control Problem 169
9.4 SISO Distillate Composition Control 174
9.4.1 Identification of SISO Process Model 174
9.4.2 SISO Controller 174
9.5 SISO Bottom Composition Control 179
9.5.1 Identification of SISO Process Model 179
9.5.2 SISO Controller 179
9.6 MIMO Control of Distillate and Bottoms Composition 184
9.6.1 Identification of MISO Process Model for Distillate Composition Control 184
9.6.2 Identification of MISO Process Model for Bottoms Composition Control 184
9.6.3 MIMO Controller 185
9.7 Conclusions 195

10 CONTROL OF MULTIVARIABLE SYSTEMS: FORCED CIRCULATION EVAPORATOR SYSTEM

10.1 Introduction 196

10.2 Mathematical Model of Evaporator System 198

10.3 Control Problem 202

10.4 Identification of SISO Process Models 205

10.4.1 SISO Model for Product Composition Control 205

10.4.2 SISO Model for Operating Pressure Control 209

10.4.3 SISO Model for Separator Level Control 212

10.5 SISO Controllers 213

10.6 MISO Controller for Compensation of Feed Flowrate Disturbance 216

10.6.1 Identification of MISO Process Model 216

10.6.2 MISO Process Model Based Controller 216

10.7 MIMO Controller for Product Composition and Operating Pressure 219

10.7.1 Identification of MISO Process Model for Product Composition Control 219

10.7.2 Identification of MISO Process Model for Operating Pressure Control 219

10.7.3 MIMO Controller 224

10.8 Dead Time Compensation 227

10.9 Conclusions 229

11 CONTROL OF SYSTEMS WITH CONSTRAINTS

11.1 Introduction 230

11.2 Extended Control Horizon 231

11.2.1 First-Order System 232

11.2.2 Second-Order System 232

11.2.3 Application to Control of CSTR 233

11.3 Controller Output Clipping 237

11.3.1 Magnitude Constraints 237

11.3.2 Rate Constraints 237

11.3.3 Application to Control of CSTR 238

11.4 Constraint Handling Using the Quadratic Programming Approach 241

11.4.1 General 241

11.4.2 Formulation of Control Problem Incorporating Constraints 241

11.4.3 Solution of the Constrained Optimisation Problem 242

11.4.4 Input Magnitude Constraints 244

11.4.5 Input Rate Constraints 245

11.4.6 Output Magnitude Constraints 245

11.4.7 The Handling of Incompatible Constraints 247

11.4.8 Application to Control of CSTR 247

11.4.9 Application to Control of Binary Distillation Column 253

11.5 Conclusions 256

12 CONCLUSIONS

12.1 Summary of the Thesis 257

12.2 Some Guidelines on Design Parameters 260

12.3 Recommedations for Further Work 262

REFERENCES 264

APPENDIX A ADAPTIVE CONTROL

A.1 Introduction 275

A.2 Self-Tuning Control 277

A.3 Recursive Least Squares Algorithm 279

A.4 Application to Control of Liquid Level 281

A.5 Application to Control of CSTR 290

A.6 Conclusions 295

APPENDIX B CONTROL OF SYSTEMS WITH DEAD TIME

B.1 Introduction 296

B.2 Model Predictions in the Presence of Dead Time 297

B.2.1 First-Order System 297

B.2.2 Second-Order System 298
B.3 Controller Formulation 301
B.4 Application to Control of Liquid Level 302
B.5 Application to Control of CSTR 305
B.6 Conclusions 308

Chapter 1

INTRODUCTION

1.1 Background

As a result of the universal drive for more consistent attainment of high product quality, more efficient use of energy, and tighter safety and environmental regulations, industrial processes have evolved over the past few decades into very complex and highly integrated systems. Such stringent demands naturally create more difficult and challenging control problems for today's industrial process control engineers - problems requiring more sopisticated solutions than can be provided by traditional techniques alone. The search for effective solutions has benefitted tremendously from the availability of powerful microprocessor-based computers at a small fraction of the cost of comparable systems as recently as a decade ago. Thus, of all factors influencing the current state of industrial process control, perhaps none is as significant as the computer, either as a hardware element for implementing advanced control systems, or as a computing device for facilitating the analysis of process behaviour. This is evidenced by the fact that virtually every modern manufacturing facility is now equipped with its own network of dedicated process control computers.

The primary objective in the chemical process industries is to combine chemical processing units (chemical reactors, distillation columns, extractors, evaporators, heat exchangers, etc.) in a rational fashion into a "chemical process" (or a "plant") in order to transform raw materials and input energy into finished products. The basic principles guiding the operation of these chemical processes may be stated as follows:

- They must be operated *safely*.
- Specified *production* rates must be achieved.
- *Product quality* specifications must be met.

Because chemical processes are by nature dynamic (i.e. their variables are always changing with time) the process control system has the responsibility of monitoring, and inducing change in, the appropriate variables related to safety, production rate, and product quality.

Virtually all processes of commercial importance are multivariable, with significant interaction between the input and output variables; they typically exhibit nonlinear characteristics, and often suffer from time delays either as a result of material transport through pipes, or as a result of measurement and analysis delay. Of these processes, perhaps the most important (and definitely the most common) are chemical reactors and distillation columns. The problems posed by the control of these two processes continues to receive both industrial and academic research attention, and as a result there is a growing body of literature on techniques for solving them.

The fundamental premise of process control is that the natural response of all dynamic processes can be modified by the influence of a controller. The objective is therefore to design and implement the controller in such a way that the dynamic response is modified appropriately, in a desired fashion. However, the extent to which the natural response can be modified appropriately will usually be determined by:

- The depth of our knowledge of the intrinsic process characteristics.
- The versatility of the hardware elements available for implementing the controller.
- The nature of inherent process limitations.

Given that not much can be done about inherent process limitations (such as the presence of time delays arising because of the finite time required for material to flow from one processing unit to another); and given that the traditional implementation limitations imposed by rigid analog controller hardware elements has been all but eliminated with the advent of the digital computer; observe therefore that the major challenge of industrial process control lies in the fact that chemical processes are typically huge, complex and poorly understood.

This puts in perspective the current trend of increasing focus on fundamental process understanding. With such process understanding, it will be possible to develop more detailed and accurate process models which enable rational analysis of dynamic process behaviour; these models, and the knowledge obtained from their analyses, in turn facilitate the design of effective control systems. The concept of process model based control is one of the most important control techniques to have emerged in recent years.

Parallel to developments in process model based control is increasing awareness of the benefits of using intelligent modelling and control techniques to tackle difficult process control problems. The term "intelligent control" is usually taken to cover the application of both machine intelligence and control theory in process control (White and Sofge, 1992). It attempts to understand and replicate, using computers, the phenomena that we normally associate with "intelligence", i.e. the generalised, flexible, learning and adaptive capability that we see in the human brain. An intelligent control system is designed so that it can achieve a high goal, while its components, control goals, plant models and control laws are not completely defined (Antsaklis, 1994; Antsaklis, 1995; Harris *et al.*, 1993).

1.2 Some Current Problems and Needs

In spite of the significant strides in process control achieved in recent years, effective control of industrial processes is still being hindered by some important problems and needs. The most important of these are listed below.

On-Line Measurements

There are many processes for which critical process variables cannot be measured on-line. This complicates "solvable" problems since it is significantly more difficult to control (with confidence) what cannot be measured.

Severely Nonlinear Processes

Since most standard control techniques applied in industry are for linear systems, the control of processes which exhibit severely nonlinear behaviour remains a major problem. Standard linear control theory is inadequate; linear MBPC provides poor performance; adaptive control has only limited success; and nonlinear MBPC is still developing.

Modelling and Identification for Control System Design

A significant number of advanced control schemes are model-based; and it is widely recognised that obtaining the process model is the single most time-consuming task in the application of model-based control.

Modelling for Simulation and Operator Training

The development of high-fidelity, low-maintenance models for simulators that can be used for control system evaluation as well as operator training is crucial if more sophisticated control systems are to become a permenant part of the chemical process of the future.

Process Monitoring and Diagnosis

The development of effective paradigms and tools for automatic monitoring and diagnosis of the increasingly complex process operations is important for continuous assessment of overall process and control system performance. The early detection of sensor and/or actuator failure (or other process operating faults), and an effective system for devicing and implementing corrective actions rapidly and effectively, are inevitable if the current and projected future safety and environmental regulations are to be met without jeopardizing economical viability in the attempt.

1.3 Control Techniques Applied in Industry

The proliferation in control literature does not seem to have a significant impact on industrial process control practice. The majority of process control loops are still based on the PID control strategy, and industry as a whole tends to adopt a cautious attitude to any new technology that is proposed until it is proven through extensive testing. As a result, the control techniques most commonly applied on industrial processes today are:

- *Classical PID Control*

  Both analog and digital versions.

  Sometimes augmented with feedforward, cascade, ratio, Smith Predictor schemes for enhanced performance, etc.

  For multivariable systems, the relative gain array (RGA) methodology is used to determine appropriate loop pairing.

- *Statistical Quality Control*

  Techniques such as Shewart charts, CUSUM, etc.

  Principal component analysis (PCA) techniques have recently been introduced for multivariable process data analysis, with potential for control decisions still being explored.

- *Decoupling*
- *Model Based Predictive Control (MBPC)*
- *Intelligent Modelling and Control*

  Neural networks for modelling (and sometimes for control)

  Fuzzy controllers

The last two techniques mentioned above, i.e. MBPC and intelligent modelling and control, will be the focus of our research investigations and will therefore be further explored in Sections 1.4 and 1.5.

1.4 Model Based Predictive Control

1.4.1 General

Model Based Predictive Control (MBPC) was introduced simultaneously by Richalet (1978) and Cutler and Ramaker (1980) in the late 1970s and early 1980s under the names of Model Algorithmic Control (MAC) and Dynamic Matrix Control (DMC), respectively. Generalised Predictive Control (GPC) popularised by Clarke (1987) and his co-workers a few years later is also considered to be in the class of model based predictive controllers. Other algorithmic variations include Extended Prediction Self-Adaptive Control (EPSAC) (De Keyser and Van Cauwenberghe , 1985), Extended Horizon Adaptive Control (EHAC) (Ydstie, 1984), and Unified Predictive Control (UPC) (Soeterboek, 1991). While the details of these various algorithms differ, clearly the main idea is very much the same: the concept of a *moving* or *receding* horizon. MBPC has been heralded as one of the most significant control developments in recent years (Deshpande, 1995). It is now widely used in the chemical and petrochemical industries and it continues to attract a considerable amount of research. Recent industrial applications of MBPC have been reviewed by Richalet (1993). Its popularity can be attributed to its many remarkable features, which include the following:

- Model based predictive controllers are relatively easy to tune. This makes predictive controllers attractive to a wide class of control engineers and even to people who are not control engineers.

- The concept of predictive control is not restricted to single-input single-output (SISO) processes. Predictive controllers can be derived for multivariable processes. Extending predictive controllers for SISO processes to multivariable processes is straightforward.

- In contrast to other model based controllers such as linear quadratic (LQ) and pole placement controllers, predictive controllers can also be derived using nonlinear process models.

- Predictive control is the only methodology that can handle process constraints in a systematic way during the design of the controller. Since in real life process constraints are quite common, this feature in particular is believed to be one of the most attractive aspects of predictive controller design.

- Predictive control is an open methodology. That is, within the framework of predictive control there are many ways to design a predictive controller. As a result, more than ten different predictive controllers, each with different properties, have been proposed in the literature.

- The concept of predictive control can be used to control a wide variety of processes without the designer having to take special precautions. It can be used to control "simple processes" as well as "difficult processes", such as processes with large time delay, processes that are nonminimum phase and processes that are open-loop unstable.

- Feed-forward action can be introduced in a natural way for compensation of measurable disturbances.

- Because predictive controllers make use of predictions, pre-scheduled reference trajectories (for example, used in robot control) or setpoints can be dealt with.

Unavoidably, predictive controller design has some drawbacks. Since predictive controllers belong to the class of model-based controller design methods, a model of the process must be available. In general, in designing a control system two phases can be distinguished: modelling and controller design. Predictive control provides only a solution for the controller design part. A model of the process must be obtained by other methods.

A second drawback is due to the fact that the predictive control concept is an open methodology. It has already been mentioned that, as a result of this, many different predictive controllers can be derived, each having different properties. From the beginning, DMC and MAC were conceived to deal with multivariable constrained problems, while GPC was proposed as an alternate approach for the adaptive control of unconstrained single-input/single-output (SISO) systems. All expositions of DMC and MAC employ a state space description, while the presentations of GPC employ exclusively a transfer function description. Soeterboek (1991) has analysed the similarities and differences between various techniques in the SISO case employing a transfer function approach. Although, at first glance, the differences between controllers seem rather small, these small differences can yield very different behaviour in closed-loop systems. As a result, it can be quite difficult to select which predictive controller is best for solving a particular control problem.

1.4.2 The Predictive Control Concept

The way predictive controllers operate for a SISO system will now be illustrated. To do this, it is first necessary to define the variables U, $\hat{Y}$ and W as follows:

$$U = \left[u(t), \cdots, u(t + H_c - 1)\right]^T \tag{1.1}$$

$$\hat{Y} = \left[\hat{y}(t+1), \cdots, \hat{y}(t + H_p)\right]^T \tag{1.2}$$

$$W = \left[w(t+1),\cdots,w(t+H_p)\right]^T \tag{1.3}$$

where $u(t)$, $y(t)$ and $w(t)$ denote the controller output, the process output and the desired process output at sampling instant t, respectively; H_c is the control horizon; H_p is the prediction horizon; and $\hat{y}(t+1)$ denotes the 1-step ahead model prediction at time t.

The predictive controller calculates a future controller output sequence U such that the predicted output of the process $\hat{Y}$ is as close to the desired process output W as possible. The assumption is usually made that the value of the controller output beyond $u(t+H_c-1)$ is equal to $u(t+H_c-1)$. The desired process output is often called the reference trajectory and is usually the setpoint. However, the response of a first-order or second-order reference model can also be used.

Rather than using the controller output sequence determined in the above way to control the process over the next H_c sampling intervals, only the first element of this controller output sequence, $u(t)$ is used. At the next sampling instant, the whole procedure is repeated using the latest measured information. This is called the *receding horizon* principle. The reason for using the receding horizon approach is that it allows us to compensate for future disturbances or modelling errors. For example, due to a disturbance or modelling error the predicted process output $\hat{y}(t+1)$ at time time t is not equal to the process output $y(t+1)$. Then, it is intuitively clear that at time $t+1$ it is better to start the predictions from the measured process output rather than from the process output predicted at the previous sample. The predicted process output is corrected for disturbances and modelling errors by activating a feed-back mechanism. As a result of the receding horizon approach, the horizon over which the process output is predicted shifts one sample into the future at every sample instant.

The process output is predicted by using a model of the process to be controlled. Any model that describes the relationship between the input and the output of the process can be used. Models which have been used include transfer-function models, impulse response models, step response models and nonlinear models. Further, if the process is subject to disturbances, a disturbance or noise model can be added to the process model, thus allowing the effect of disturbances on the predicted process output to be taken into account.

In order to define how well the predicted process output tracks the desired process output, a criterion (or objective) function is used. Typically such a criterion function is a function of $\hat{Y}$, W and U. Probably the simplest criterion function that can be used for predictive controller design is:

$$J = \sum_{i=1}^{H_p}\left(\hat{y}(t+i) - w(t+i)\right)^2 \tag{1.4}$$

A modified form which allows controller output weighting is also often used:

$$J = \sum_{i=1}^{H_p} \left(\hat{y}(t+i) - w(t+i)\right)^2 + \rho \Delta u(t+i-1)^2 \tag{1.5}$$

where ρ is a weighting factor ($\rho \geq 0$) and Δu is the change in the controller output.

The optimal controller output sequence U_{opt} over the control horizon which minimizes the future tracking error is obtained by minimization of J with respect to U:

$$U_{opt} = \arg\min_{U} J \tag{1.6}$$

When the process has a time delay of say, 2 samples, the approach is no different, except that predicting $\hat{y}(t+1)$ and $\hat{y}(t+2)$ does not make any sense because these values cannot be influenced by the control actions at t and t+1. Then, the cost function (1.4) can be changed into:

$$J = \sum_{i=3}^{H_p} \left(\hat{y}(t+i) - w(t+i)\right)^2 \tag{1.7}$$

Thus when there is time delay the optimal controller output sequence is obtained by the minimization of (1.7) with respect to U.

Clearly, calculating the controller output sequence is an optimisation problem or, more specifically, a minimisation problem. Usually, solving a minimisation problem requires an iterative procedure. However, an analytical solution is available when the criterion is quadratic, the model is linear and there are no constraints. This is the reason why a quadratic criterion function is normally used by predictive controllers. Note that the controller outputs calculated by the minimisation of a criterion function with respect to U are not structured. That is, when minimising the criterion function the controller outputs are not assumed to be generated by a control law in contrast to most other control strategies.

Because in predictive controller design it is not *a priori* assumed that the controller outputs are generated by a control law, constraints on the controller output can be taken into account in a systematic way. For example, if the output of the controller is limited between two values (which is the case in most practical applications), the optimization problem can simply be formulated as:

$$U_{opt} = \arg\min_{U} J \tag{1.8}$$

with U_{opt} subject to

$$\underline{u} \leq u(t+i-1) \leq \overline{u}, \qquad 1 \leq i \leq H_c \tag{1.9}$$

where $\underline{u}$ and $\overline{u}$ represent the lower and upper bound, respectively, on the controller output. Now, the optimisation problem is a constrained optimization problem for which an analytical solution is no longer available and, therefore, an iterative optimization method must be used.

1.4.3 Linear Process Models

Without delving into the details of the control algorithms itself, we will attempt to review the most important linear process models used, highlighting the advantages and disadvantages of each approach.

One of the simplest models that can be used to predict the output of a process is the impulse response model:

$$y(t) = \sum_{j=0}^{\infty} h_j u(t-j-1) \tag{1.10}$$

where h_j is the jth element of the impulse response of the process. Note that an infinite number of impulse response elements are required in this model. Assuming, however, that the impulse response goes to zero asymptotically, the impulse response may be truncated to some finite number of terms. Then equation (1.10) becomes:

$$y(k) = \sum_{j=0}^{n_H-1} h_j u(t-j-1) \tag{1.11}$$

where n_H is the number of impulse response elements h_j that are taken into account. All other elements are assumed to be zero. The model is called the Finite Impulse Response (FIR) model and is used in Model Algorithmic Control (MAC). The main disadvantages of FIR models are that unstable processes cannot be modelled and the FIR model requires many parameters to be known or estimated. On the other hand, prediction of the process output is simple with no complex calculations required, and no assumption needs to be made about the order of the process.

Another process model that is often used in predictive controllers is the Finite Step Response (FSR):

$$y(k) = \sum_{j=0}^{n_S-1} s_j \Delta u(t-j-1) \tag{1.12}$$

where Δ is the differencing operator ($\Delta = 1 - q^{-1}$ with q^{-1} the backward shift operator) and n_S is the number of step response elements s_j that are taken into account. All other elements are assumed to be constant. In the case where the process is not disturbed by noise, the coefficients of the FSR model can be determined simply by applying a unit step to the input of the process. The FSR model has the same advantages and disadvantages as the FIR model and is used in Dynamic Matrix Control (DMC).

A process model that does not have the disadvantages of the above-mentioned models is is the transfer-function model:

$$y(k) = \frac{q^{-d}B(q^{-1})}{A(q^{-1})}u(t-1) \tag{1.13}$$

where d is the time delay of the process in samples ($d \geq 0$) and the polynomials A and B are given by:

$$A(q^{-1}) = 1 + a_1 q^{-1} + \cdots + a_{n_A} q^{-n_A} \tag{1.14}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \cdots + b_{n_B} q^{-n_B} \tag{1.15}$$

where n_A and n_B are the degrees of the polynomials A and B, respectively. In contrast to FIR and FSR models, equation (1.15) can also be used to model unstable processes. Also, a minimal number of parameters are used to describe a linear process. The main disadvantage is that an assumption about the order of the process must be made. Also, prediction of the output of a process described by a transfer function model is more complex than that of a process described by an FIR or FSR model. Transfer function models are used by Generalised Predictive Control (GPC) and Unified Predictive Control (UPC) algorithms.

1.4.4 Nonlinear Model Based Predictive Control

Several attempts have been made to extend MBPC techniques to nonlinear systems. There are essentially three major approaches: i.e. scheduled linearisation (Garcia, 1984), extended linear MBPC (Peterson et al., 1989) and explicit nonlinear MBPC (Brengel and Seider, 1989; Li and Biegler, 1989; Sistu and Bequette, 1990). In the last approach, the MBPC control algorithm is posed as a nonlinear programming (NLP) problem, incorporating an explicit nonlinear model and possibly nonlinear constraints.

Recently, the use of neural network and fuzzy models to represent nonlinear process dynamics has become quite popular. Unlike linear process models which are based on deviation variables, these new modelling approaches allow the use of the absolute values of variables to achieve good representation of nonlinear process dynamics over a wide range. The use of absolute values often introduces problems in the formulation of the controller making it necessary to resort to techniques such as NLP.

1.5 Intelligent Modelling and Control

1.5.1 General

The increased complexity and stringent demands of today's chemical process plants necessitates more sophisticated control systems capable of delivering better and more flexible performance. Increasingly, control systems are required to have high dynamical performance and robust behaviours, and yet be able to cope with complex, uncertain and highly nonlinear process relationships over wide operating envelopes. One approach being advocated for dealing with such challenging process control problems is intelligent modelling and control. Traditionally, intelligent control has embraced classical control theory, neural networks, fuzzy logic, expert systems, and a wide variety of search techniques (such as genetic algorithms and others). Expert systems or knowledge-based systems are probably the best known, having been around for many years now. Considerable attention in recent years is also being focussed in two other areas: artificial neural networks and fuzzy logic. Artificial neural networks were originally developed to emulate the human brain's neuronal-synaptic mechanisms that store, learn and retrieve information on a purely experiential basis, whereas fuzzy logic was developed to emulate human reasoning using linguistic expressions. Zadeh (1994) has coined the term 'soft computing' to differentiate fuzzy logic, neural networks and probabilistic reasoning (includes genetic algorithms, chaos theory and parts of learning theory) from 'hard computing' based on binary logic, crisp systems and numerical analysis. Hard computing has the attributes of precision, and categoricity, whereas soft computing is based on approximation and dispositionality. Although in hard computing, imprecision and uncertainty are undesirable properties, in soft computing the tolerance for imprecision and uncertainty is exploited to achieve an acceptable solution which is low cost, tractable and has high machine intelligence quotient (MIQ).

Fuzzy logic is mainly concerned with imprecision and approximate reasoning, neurocomputing mainly with learning and curve fitting, and probabilistic reasoning with uncertainty and propagation of belief. These are complementary rather than competitive. The experiences gained over the past decade have indicated that it can be more effective to use them in a combined manner, rather than exclusively. This moves us towards a new era where control theory and AI will become far more compatible with each other. This allows arrangements such as that shown in Figure 1.1 (White and Sofge, 1992), where neural tools and fuzzy logic are used as two complementary technologies on one common controller.

It is not our intention here to delve into the details of these emerging and controversial technologies. Such an attempt can cover many volumes and still not be complete. Our aim here is to draw attention to the most important trends and highlight the process control applications, with particular emphasis on neural networks, fuzzy logic and expert systems. Genetic algorithms largely remain a subject for research rather than practical implementation and will therefore not be covered by this review.

Central to many advanced process control applications is the construction of a model. As processes increase in complexity, they become less amenable to direct mathematical modelling based on physical laws. Rather than utilise models based on linear or physico-chemical relationships, intelligent controllers derive predictive models based on experiential evidence, and use such models to design control systems which can:

- Operate in an ill-defined, time-varying environment.
- Adapt to changes in the plant's dynamics as well as the environmental effects.
- Learn significant information in a stable manner.
- Place few restrictions on the plant's dynamics.

Human learning appears to embody elements of all of these properties, and currently researchers are trying to endow machines with such human-like qualities to enable them to operate autonomously with the minimum amount of intervention.

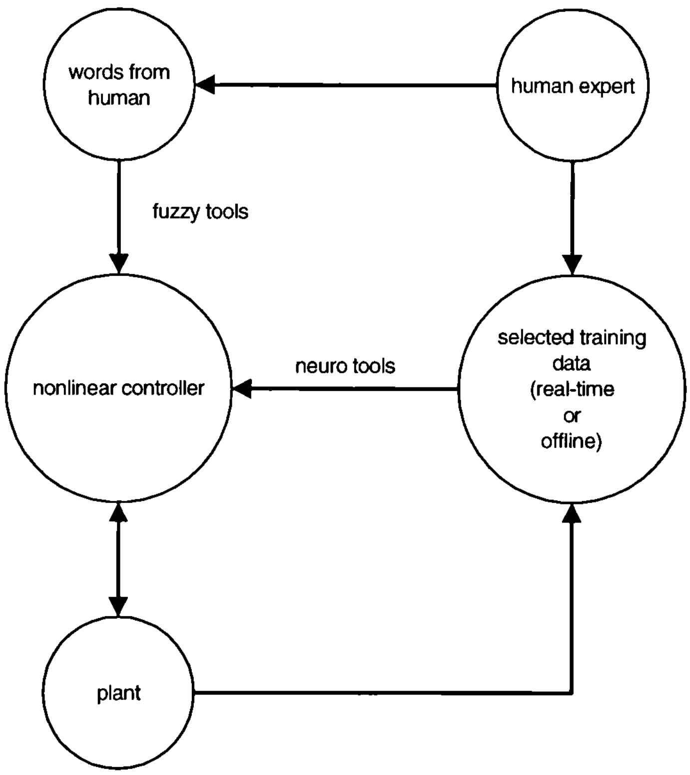


Figure 1.1: A way to combine fuzzy and neural tools

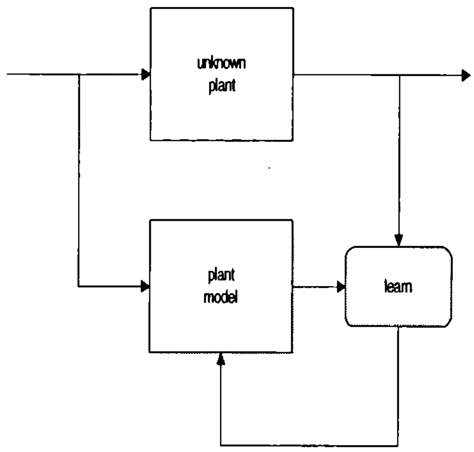
1.5.2 Artificial Neural Networks

The growing literature within the field of chemical engineering describing the use of neural networks has evolved to a diverse range of applications. There are several reasons for these developments. Firstly, recent advances in computer technology and parallel processing have made the use of neural networks more economically feasible than in the past. Second, little or no *a priori* knowledge of the task is required; and thirdly, neural networks have the potential to solve certain types of complex problems that have not been satisfactorily handled by more traditional methods.

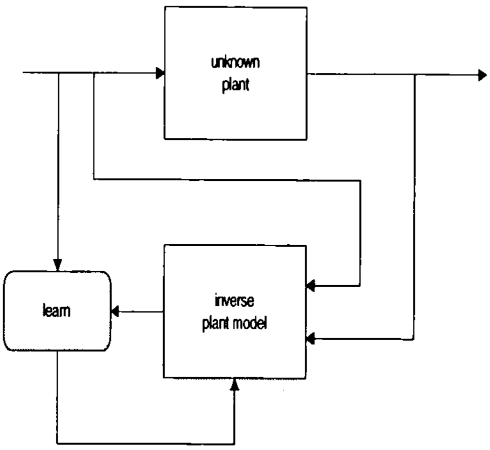
There are four principal architectures which can exploit the modelling capabilities of neural networks: as a basic plant model, an inverse plant model, a specialised inverse plant model and an operator model as shown in Figure 1.2 (Brown and Harris, 1994). For three of the four cases, the desired value of the network's output is directly available and any supervised learning rule can be used to train the weight set of the network. The error in the specialised inverse plant modelling algorithm is formed at the output of the plant, whereas the network's output forms the input to the plant. Therefore some method is required for feeding back the plant output error, in order to train the inverse model. The application of each type of model will now be discussed.

Soft Sensing

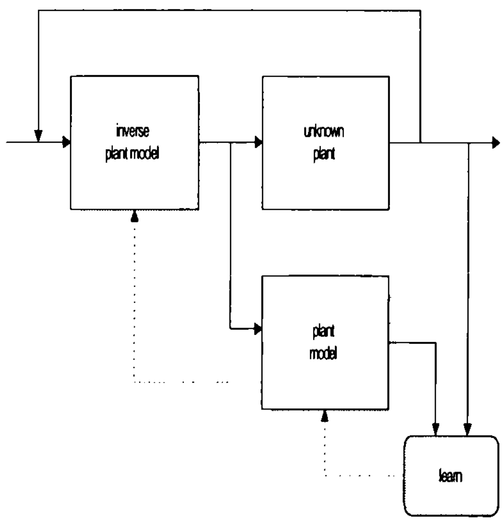
Plant models may be required for a variety of applications. Sensor interpretation is an example of easy, first-generation application of neural networks. Traditionally, the process industries have relied on simple measurements such as flow, pressure and temperature; however to achieve a higher level of efficiency, and to control more complex systems such as bioreactors, it is important to measure parameters such as concentration as well. The fundamental problem is that the key quality variables cannot be measured at a rate which enables their effective regulation. This can be due to limited analyser cycle times or a reliance upon off-line laboratory assays. An obvious solution to such problems could be realised by the use of a model along with secondary process measurements, to infer product quality variables (at the rate at which the secondary variables are available) that are either costly or impossible to measure on-line. Hence, if the relationship between quality measurements and on-line process variables can be captured then the resulting model can be utilised within a control scheme to enhance process regulation. The concept is called *inferential estimation* or *soft sensing* and some promising results have been reported by various researchers (Tham *et al.*, 1991b; Willis, *et al.*, 1991; Di Massimo *et al.*, 1991; McAvoy *et al.*, 1989).



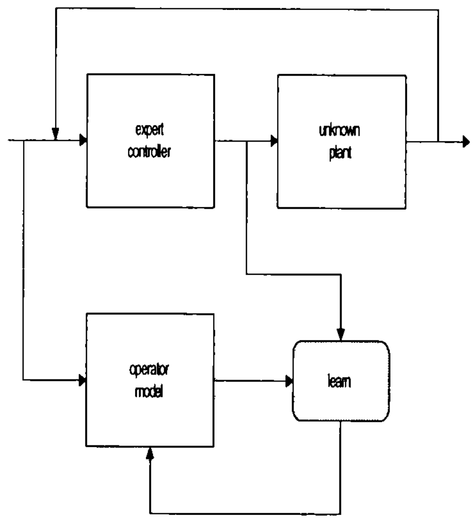
(a) Forward plant modelling



(b) Inverse plant modelling



(c) Specialised inverse plant modelling



(d) Operator modelling

Figure 1.2: Four learning modelling architectures incorporating neural networks.

Control

The modelling of human operators as shown in Figure 1.2(d) can be very useful with some processes (Gingrich *et al.*, 1990). The difference between the best human operators and the average operators is often worth large amounts of money, because of differences in plant efficiency during plant change-over and the like.

Whilst inferential estimation schemes operating in open-loop can be used to assist operators with the availability of fast and accurate product quality estimates, the possibility of closed loop inferential control is also very appealing. The effective elimination of time delays caused by the use of an on-line analyser or the need to perform off-line analysis affords the oppurtunity of tight product control. A potential problem highlighted by Morris *et al.* (1994) is that the model might have been identified using data collected from the plant which may have some of its control loops still closed. The resulting model will then have been identified with correlated data and will not be representative of the underlying process behaviour.

Willis *et al.* (1993) have shown that the use of nonlinear neural network models for tuning PID controllers leads to better controller tuning than that possible using linear models.

Considerable emphasis has been focussed on control strategies based on the Internal Model Control (IMC) structure shown in Figure 1.3 (Hunt and Sbarbaro, 1991; Hunt *et al.*, 1993; Bhat and McAvoy, 1990; Psichogios and Ungar, 1991; Nahas *et al.*, 1992; Dayal *et al.*, 1994). The IMC control structure incorporates models of the plant dynamics and the corresponding inverse. As an important property of IMC, it can be shown that given plant and controller which are input-output stable and having a perfect model of the plant, the closed-loop system will also be input-output stable (Garcia and Morari, 1982; Garcia, 1989; Morari and Zafiriou, 1989). Unfortunately, the neural network model of the inverse process dynamics (the controller) may not have a steady-state gain which is the exact inverse of the steady-state gain of the neural network model of the forward process dynamics. Thus, steady-state offset in the controlled variable cannot be eliminated.

Psichogios and Ungar (1991) and Nahas *et al.* (1992) presented a slightly different IMC control scheme in which the control move is computed by the on-line numerical inversion of the neural network model of the forward process dynamics. Nahas *et al.* (1992) used the predicted process outputs instead of the actual errors as inputs to the neural network when they numerically inverted the neural network model of the forward process dynamics. This modification allows the controller to elimintate steady-state offset in the controlled variable. However, this scheme has potential convergence problems during the on-line numerical inversion of the nonlinear forward model neural network.

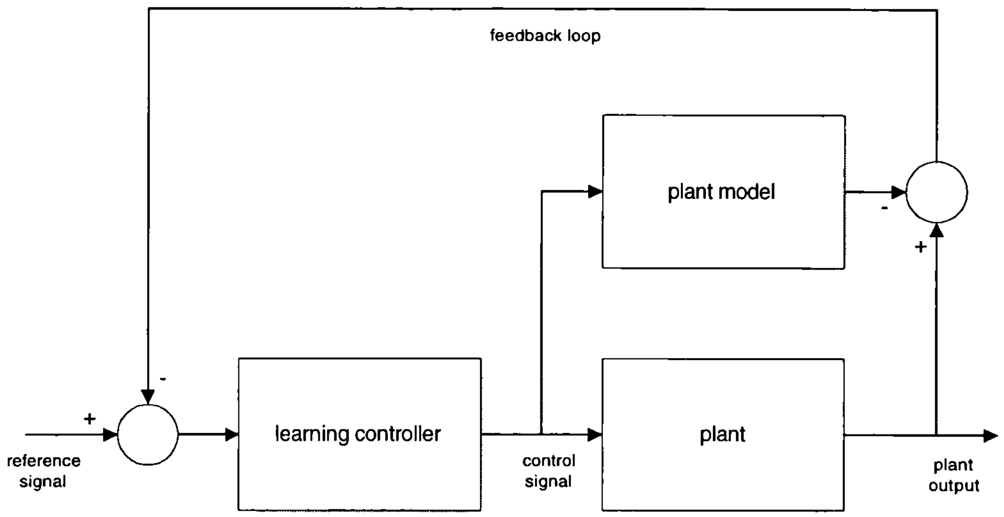


Figure 1.3: An internal model control architecture

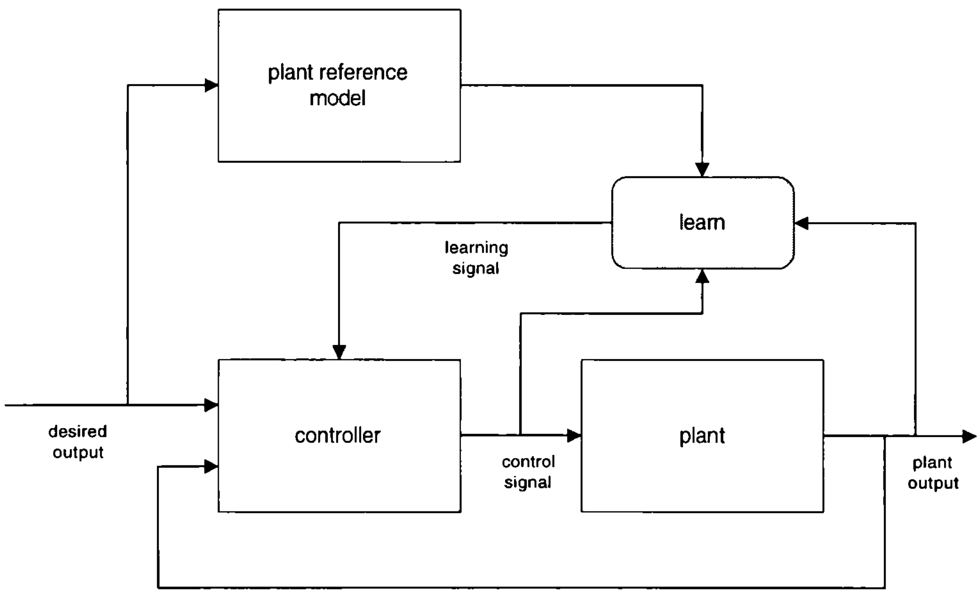


Figure 1.4: A model reference control architecture

Another important approach is that of model reference control. This scheme has been widely used in the linear adaptive control field (Astrom and Wittenmark, 1989), and is shown in Figure 1.4. The control objective is to adjust the control signal in a stable manner so that the plant's output asymptotically tracks the reference model's output (Narendera and Parthasarathy, 1990). The performance of this algorithm depends on the choice of a suitable reference model and the derivation of an appropriate learning mechanism.

Probably the most successful approach to-date (Saint-Donat *et al.*, 1991; Morris *et al.*, 1994) is to use a dynamic nonlinear neural network process model, instead of a dynamic linear process model, in an optimisation based multi-step predictive control scheme. The concept of model based predictive control has been presented in Section 1.4 and will therefore not be further elaborated here. Optimisation using a nonlinear neural network model is generally more difficult to solve than optimization involving a linear model. In fact, Psichogios and Ungar (1991) found that the CPU time required to solve for the control action in multi-step predictive control was 30 times greater than that required for an IMC control scheme with an inverse model neural network controller.

Process Monitoring and Fault Detection

Recently, a number of papers have appeared on the application of neural networks to qualitatively interpret process data and fault detection (Naidu et al., 1990; Kramer and Leonard, 1990; Morris *et al.*, 1994). Theoretically, neural networks can be used to convert quantitative plant data into a qualitative interpretation. For example, neural networks for pattern recognition have been used to detect hot spots in the continuous casting of steel (Tanaka, 1991). In this application, neural networks led to great increase in accuracy over more expensive, conventional methods.

Researchers are still actively studying this application since the large number of possible input patterns can present difficulties.

1.5.3 Fuzzy Logic

At present, there is hardly a topic that attracts more attention in the industrial world and among the general public than fuzzy logic. Over the past several years, fuzzy control has emerged as one of the most active areas of research in control engineering. The pioneering research of Mamdani and his colleagues (Mamdani and Assilan, 1975; Mamdani, 1976) on fuzzy control was motivated by Zadeh's seminal papers on the linguistic approach and system analysis based on the theory of fuzzy sets (Zadeh, 1965; Zadeh, 1973). Applications of fuzzy control in such diverse areas as water quality control, automatic train operation, automatic container crane operation, elevator control, nuclear reactor control, automobile transmission control, etc., have pointed the way for the utilisation of fuzzy control in the context of ill-defined processes that can be controlled by a skilled human operator without the knowledge of their underlying dynamics (Lee, 1990; Ragot and Lamotte, 1993).

Fuzzy control applications in the chemical process industry have also been growing rapidly in recent years, making it difficult to present a comprehensive survey of the wide variety of applications. Notable successes seem to have been achieved with highly nonlinear processes and multivariable processes. Some applications cited in literature include warm water process (Kickert, 1976), water purification process (Sugeno, 1985), continuous stirred tank reactor (King, 1977; Chunyu, 1989; Preub, 1993), heat exchanger (Ostergaard, 1977), activated sludge process (Tong *et al.*, 1980), cement kilns (Umbers and King, 1980), box annealing furnace (Yonekura, 1981), continuous casting plant (Bartolini *et al.*, 1982), alloy-charging process in steel making (Takagi and Sugeno, 1985), blast furnace (Hong, 1985) and pH neutralisation of chemical waste water (Wegmann and Mohr, 1993; Galluzzo *et al.*, 1991).

Although achieving many practical successes, the rule base of the most common fuzzy logic controller is static and has to be developed by trial and error until a good performance is achieved. A major difficulty is found to be the length of time and amount of effort required to develop the rules, especially in the case of nonlinear systems. Also, the rules of the simple fuzzy logic controller do not, in general, contain a temporal component so they cannot cope with process changes over time. In view of these limitations, a considerable amount of research is being carried out into adaptive and model-based fuzzy controllers (model-based fuzzy controllers will be discussed in considerable length in Chapters 2 and 3 of this thesis).

Fuzzy controllers contain a number of sets of parameters that can be altered to modify the controller performance:

- the scaling factors for each variable;
- the fuzzy set representing the meaning of linguistic variables;
- the if-then rules.

Adaptive fuzzy controllers that modify fuzzy set definitions or the scaling factors are often called self-tuning controllers (Yamashita *et al.*, 1988; Tanaka and Sano, 1991; Nomura *et al.*, 1991; Maeda *et al.*, 1991; Batur and Kasparian, 1989). Altering these parameters essentially fine tunes an already working controller. Adaptive fuzzy controllers that modify the rules are called self-organising controllers (Procyk and Mamdani, 1979; Yamazaki and Mamdani, 1982; Daley and Gill, 1986; Linkens and Abbod, 1991; Linkens and Abbod, 1992; Moore and Harris, 1992; Song and Park, 1993; Shah and Rajamani, 1991). They can either modify an existing set of rules, in which case they are similar to self-tuning controllers, or they can start with no rules at all and "learn" their control strategy as they go. Direct self-organising fuzzy logic controllers use observations of the system closed loop performance to directly manipulate the controller fuzzy rule base or relational matrix, without any intermediate process model being produced. In contrast, the indirect self organising fuzzy logic controller (Moore and Harris, 1992) generates a fuzzy relational matrix of the plant and then inverts this model in order to find the control which realises the desired next state.

1.5.4 Knowledge-Based Systems

Knowledge-based systems or expert systems have been used quite successfully for real-time trouble-shooting of complex control systems. Although more difficult and time-consuming to develop than an off-line expert system designed for repair of specific hardware, a real-time diagnostic expert system could be cost-effective compared to the lost production time generally encountered with control system outages. Process control system malfunctions generally result in one of two conditions:

- The system fails and shuts down, resulting in unproductive labour costs and lost revenue.

- The system experiences intermittent component failures that undermine its integrity and leads to poor control, which could result in poor product quality or even unsafe operating conditions.

Either scenario could justify the time and expense to build an expert system to assist the engineer or on-site technician in diagnosing and repairing the problem as quickly as posible. The cost involved in developing such systems can be more easily justified if the system can be distributed to many similar operations within the company.

Another application area is alarm analysis. An expert system can be designed to monitor a process in real-time, interpret alarm conditions, and alert the operator to possible corrective actions. This idea could be extended with a real-time system designed to collect product data, calculate statistical trends, and provide an interpretation. Such a system would enable proactive control of a process - by alerting the operator to statistical trends in key process variables before a system upset occurs. Many control system vendors are already enhancing the operation of their systems with integrated software packages that allow real-time tracking of SPC charts. With widespread implementation, SPC has proven itself to be very valuable methodology for improving quality and reducing manufacturing costs. The real value in capturing these data, however, is a timely and accurate analysis. An expert system could provide constant vigilance and be a valuable assistant to a busy control-room operator, thus bolstering SPC's value (Rock and Guerin, 1992).

1.6 Motivations for the Current Research

The previous sections have reviewed some of the important contemporary challenges of industrial process control, noting the fortunate circumstance of the emergence of cheap and powerful computer systems just in time to facilitate the implementation of the sophisticated control techniques required to take on these challenges. The difficulty posed by severely nonlinear processes was highlighted as was the emergence of MBPC and intelligent modelling and control techniques to tackle difficult process control problems.

MBPC schemes derive some of their industrial appeal from their ability to handle input and output constraints, time delays, non-minimum phase behaviour and multivariable systems. Despite the success enjoyed by MBPCs in industry, there are many processes which pose a challenge for the standard, linear model based algorithms. For example, batch and semi-batch processes are carried out over a wide dynamic range; hence the concept of operation around a steady state becomes invalid. Also, there are some continuous processes which undergo frequent transitions to permit the manufacture of several grades of a basic product. Such processes operate at several steady state levels and may experience start-ups and shutdowns on a daily basis. Lastly, there are some chemical processes (e.g. some polymer reactors) which are so severely nonlinear that small-to-moderate perturbations around the steady state can render a linear model based controller inadequate or even unstable. Thus, there is an incentive to develop extensions of MBPC to tackle nonlinear systems.

A distinguishing characteristic of fuzzy logic is its ability to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness, and low cost solution. It can be used to provide a computing paradigm for modelling highly nonlinear chemical processes when a sufficiently accurate and yet not unreasonably complex model of the process to be controlled is unavailable. The modelling problem, instead of being posed within a strictly analytic framework, is based on empirical acquired knowledge regarding the operation of the process.

The main objective of our research is, therefore, to incorporate MBPC into a nonlinear fuzzy modelling framework. This fusion of ideas drawn from two apparently distinct fields, i.e. MBPC and fuzzy logic, is hoped will lead to a nonlinear MBPC with the same remarkable attributes as linear MBPC while also offering the capability of being applied to highly nonlinear processes.

1.7 Scope of this Work

The scope of the research carried out during the course of the PhD programme and reported in this thesis covers the following:

Fuzzy Modelling

The use of the fuzzy modelling method proposed by Takagi and Sugeno (1985) for modelling highly nonlinear chemical process systems has been examined. This modelling method uses a model representation which is analogous to that used by linear systems and deviates considerably from traditional fuzzy modelling approaches.

System Identification

Many different methods have been proposed in the literature for the identification of fuzzy process models, especially in the case of relational fuzzy models. The use of the standard least squares algorithm as proposed by Takagi and Sugeno (1985) provides a simple and highly efficient method for the identification of the off-line fuzzy model. The effectiveness of this identification method has been investigated in the context of highly nonlinear chemical process systems.

Model Based Predictive Control

Modern control theory opens the door to computer-oriented control strategies based on optimisation methods. The use of such methods with nonlinear fuzzy process models has generally been quite difficult and it has been necessary to resort to numerical approaches. We have attempted to make use of the fuzzy model's analogous structure to linear process models to design a computationally efficient control algorithm.

Adaptive Control

The use of the recursive least squares algorithm for adaptive modelling has been examined. This method may offer an advantage to the traditional approach of rule replacement often used with fuzzy systems.

Dead Time Compensation

In many industrial processes, the phenomenon of dead time is quite common. A method of enabling dead time compensation has therefore been examined.

Multivariable Systems

An important limitation of many relational fuzzy modelling methods lies is in the ability to be easily extended to multivariable systems, thereby facilitating feed forward and decoupling control. A significant portion of the research was therefore focused on multivariable systems.

Constraints Handling

An important feature of MBPC is the ability to incorporate explicit constraints on manipulated and controlled variables in the design of the control algorithm. The use of the quadratic programming approach to enable constraint handling capability in our proposed controller has therefore been examined.

Controller Robustness

It is important, especially in the context of ill-defined processes, that controller robustness is not sacrificed to achieve better performance. The performance of the controller has therefore been evaluated in the presence of noise and under conditions quite different from that used for the identification of the process model.

1.8 Thesis Overview

The twelve chapters of this thesis can be organised into the six sections shown in Table 1.1. The structure of the thesis and the style used for presenting the results of our research makes it unnecessary to read every preceeding chapter. Section 1 provides a review based on published literature on topics related to our research area. The aim of the review is to provide readers with a better perspective of the issues affecting this research study, so that both the significance of the study and the control methods we have used can be better appreciated. Since most of the material presented in this section is of a general nature, it can be skipped by readers familiar to concepts such as model based predictive control, intelligent control and fuzzy modelling. Readers who wish to concentrate on the details of our fuzzy modelling method and our proposed control algorithm can proceed directly to Section 2 of the thesis.

The control methods presented in this thesis have been tested using simulations of well-known chemical processes. Throughout this thesis, emphasis has been given to evaluating the performance of controllers using setpoint changes. This permits comparison of the performances of controllers over wide operating ranges to gauge the improvement, if any, when used to control processes which are significantly nonlinear. Although frequent setpoint changes may be made to permit the manufacture of several grades of a basic product, most chemical engineering control problems, however, involve regulation around a setpoint. Emphasis has therefore also been given to evaluating the performance of controllers using disturbance inputs.

Section	Section Heading	Chapters
1	Literature Review	Chapter 1: Introduction. Chapter 2: Fuzzy Modelling. Chapter 3: Fuzzy Model Based Control.
2	Identification of Fuzzy Process Model	Chapter 4: Identification of Fuzzy Process Model.
3	Basic Control Algorithm	Chapter 5: One-Step Ahead Predictive Controller. Chapter 6: Long-Range Predictive Control: Numerical Approach. Chapter 7: Long-Range Predictive Control: Analytical Approach.
4	Control of Multivariable Systems	Chapter 8: Control of Multi-Input Single-Output Systems. Chapter 9: Control of Multivariable Systems: Binary Distillation Process. Chapter 10: Control of Multivariable Systems: Evaporator System.
5	Control of Systems with Constraints	Chapter 11: Control of Systems with Constraints.
6	Conclusions	Chapter 12: Conclusions

Table 1.1: Structure of this thesis.

Comparison of the performances of different controllers has mostly been based on the integral of absolute error (IAE). Where important, assessment has also been based on visual observations such as rise time, peak overshoot, settling time, oscillatory behaviour, etc.

A chapter-by-chapter overview of the contents of this thesis will now be provided.

Chapter 1

This chapter provides background information on our research project and examines the important contemporary challenges of industrial process control. The concepts of model-based predictive control and intelligent modelling and control are briefly introduced. The emphasis in the second case has been on drawing the attention of the reader to the most important trends and to highlight the process control applications, with particular emphasis to neural networks, fuzzy logic and expert systems.

Chapter 2

Fuzzy modelling is a procedure for developing fuzzy membership functions and fuzzy rules from a given data set. This chapter begins by introducing the basic concepts and definitions involved in fuzzy modelling. These concepts are used to examine various fuzzy modelling approaches that have been proposed in the literature. The chapter finishes with a comparative evaluation of some of these methods.

Chapter 3

The difficulties posed by the traditional approach of designing fuzzy controllers are first examined before looking at the advantages of the model-based control approach. Various fuzzy model based controllers that have been proposed in the literature are then examined and compared.

Chapter 4

This chapter starts by describing the piecewise linear fuzzy modelling proposed by Takagi and Sugeno and also used in this study. The method is then applied to model nonlinear chemical processes using simulations of liquid level and CSTR systems.

Chapter 5

This chapter emphasizes the development of the conceptual framework for a fuzzy model based predictive control strategy based on the fuzzy modelling approach presented in Chapter 4. The prediction horizon used by the controller has been limited to 1-step and the optimal controller output is determined using an analytical approach. It is shown how the nonlinear fuzzy model can be transformed into a linear model to facilitate the design of the control algorithm.

Chapter 6

Even though the 1-step ahead predictive controller has been shown to be viable in Chapter 5, it has not sufficiently addressed the issue of controller robustness. In Chapter 6, a significant step forward in this direction is made by looking at a numerical approach of extending the prediction horizon. Various important issues involved in using the fuzzy model to make multi-step predictions are examined. These issues are important not only for the numerical approach examined here but also for the analytical approach presented in the next chapter.

Chapter 7

In Chapter 6, the focus was on a numerical approach of designing a long-range predictive control algorithm. This chapter, on the other hand, examines an alternative analytical approach of extending the prediction and control horizons used by the controller. The much lower computational requirements of the analytical approach provides it with a distinct advantage over the numerical approach. The question of fuzzy model linearisation, which was partially examined in Chapter 6, is further examined in this chapter. The effectiveness of the linearisation method is compared with the usual linearisation method used in control engineering. An attempt is made to examine the performance of the controller in the presence of noise and under process conditions quite different from that used for the identification of the process model.

The fuzzy model used in all previous studies has been assumed to be a fixed model identified from the open loop response of the process prior to the implementation of the control system. The fuzzy modelling method as proposed by Takagi and Sugeno (1985) also did not consider adaptive modelling. Appendix A examines the use of the recursive least squares algorithm for on-line parameter estimation and control based on the control algorithm presented in this chapter.

It is known that dead times make processes difficult to control. Designing controllers to overcome dead time has always been a serious challenge to control engineers. Appendix B examines how the control algorithm presented in this chapter can be extended to enable dead time compensation.

Chapter 8

All of the previous studies have been based on single-input single-output (SISO) fuzzy models using the manipulated variable as the input. In this chapter, we attempt to extend the fuzzy model to include one or more disturbance inputs. By doing this, it may be possible to achieve a feedforward effect in our controller to the effect of these disturbances. Such a prospect makes multi-input single-output (MISO) fuzzy model based controllers look particularly attractive.

Chapter 9

Control of the distillation process is known to be particularly difficult because of the significant nonlinearity and the severe interaction between process variables. These problems are particularly important in the case of dual composition control. The method used in Chapter 8 is extended in this chapter to allow coupling of the control loops used for composition control.

Chapter 10

A number of the methods presented in earlier chapters are tested in this chapter using a simulation of a forced circulation evaporator system. This is a multivariable control problem consisting of three control loops.

Chapter 11

In all of our earlier research studies, it has been assumed that the only constraints imposed on the control system are minimum and maximum values on the controller output. In this chapter, we show how the presence of constraints can lead to a significantly different output response from that provided by the unconstrained controller. A method of explicitly including constraints in the optimisation process is then examined.

Chapter 12

This chapter summarises the most important findings of this research project and makes recommendations for further research work in the same area.

Chapter 2

FUZZY MODELLING

2.1 Introduction

An integral component of many advanced control strategies is a model. The modelling of real world systems, however, often presents problems. As processes increase in complexity, they become less amenable to direct mathematical modelling based on physical laws, since they may be:

- Distributed, stochastic, nonlinear and time varying.
- Subject to large unpredictable environmental disturbances.
- Have variables that are difficult to measure, have unknown causal relationships, or are too difficult or expensive to evaluate in real-time.

According to Zadeh's Principle of Incompatibility (Zadeh and Chang, 1972), the closer one looks at a real world problem, the fuzzier becomes the solution. As the complexity of a system increases, our ability to make precise and yet significant statements about its behaviour diminishes until a threshold beyond which precision and significance (relevance) become almost mutually exclusive characteristics. This throws light on the role fuzzy logic can play in the modelling of real world systems. In general, a fuzzy logic system is a nonlinear mapping of an input data (feature) vector into a scalar output (the vector output case decomposes into a collection of independent multi-input/single-output systems). The richness of fuzzy logic is that there are enormous numbers of possibilities that lead to lots of different mappings. This richness can be used to provide a computing paradigm for modelling highly nonlinear processes, especially when a sufficiently accurate and yet not unreasonably complex model of the process to be controlled is unavailable. The modelling problem, instead of being posed within a strictly analytic framework, is based on empirical acquired knowledge regarding the operation of the process. This knowledge, cast into linguistic or rule-based form, constitutes the basis of a fuzzy model.

Fuzzy modelling is a procedure for developing fuzzy membership functions and fuzzy rules from a given set of data. This chapter begins by introducing basic concepts and definitions involved in fuzzy modelling. These concepts are used to examine various fuzzy modelling approaches that have been proposed in the literature. The chapter finishes with a comparative evaluation of some of these approaches.

2.2 Basic Concepts and Definitions

Fuzzy logic provides a computationally-oriented system of concepts and techniques for dealing with modes of reasoning which are approximate rather than exact. Approximate reasoning is essential for simulating human decision-making behaviour in an environment of uncertainty and imprecision. By employing the techniques of fuzzy set theory, approximate reasoning (with precise reasoning viewed as a limiting case) can be studied in a formal way.

2.2.1 Fuzzy Set

The concept of a fuzzy set can be viewed as an extension of an ordinary set. In a fuzzy set, an element can be a member of the set with a degree of membership varying between 0 and 1. Thus, a fuzzy set F in a universe of discourse $\mathbf{U} = \{u_i,\ i = 1, \cdots, n\}$ is defined by its membership function:

$$\mu_F : \mathbf{U} \rightarrow [0,1] \tag{2.1}$$

If $\mu_F(u_i)$ are 0 or 1, the fuzzy set is an ordinary set. When $\mathbf{U}$ is continuous, a fuzzy set F can be written concisely as:

$$F = \int_{\mathrm{U}} \mu_F(u) / u \tag{2.2}$$

When $\mathbf{U}$ is discrete, a fuzzy set F is represented as:

$$F = \sum_{i=1}^{n} \mu_F(u_i) / u_i . \tag{2.3}$$

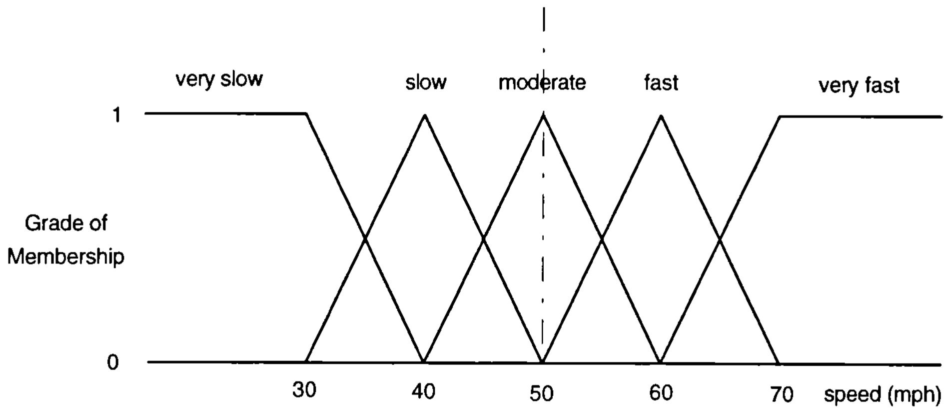


Figure 2.1: Diagrammatic Representation of Fuzzy Sets Used to Define Speed.

2.2.2 Linguistic Variable

A linguistic variable can be regarded either as a variable whose values are fuzzy numbers (fuzzy sets which are normal and convex) or as a variable whose values are defined in linguistic terms. To illustrate, if *speed* is interpreted as a linguistic variable, then the linguistic values of *speed* might be:

$$T(speed) = \{\text{very slow, slow, moderate, fast, very fast}\} \quad (2.4)$$

In a particular context, "very slow" may be interpreted as a speed below about 30 mph, "slow" as a speed about 40 mph, "moderate" as a speed close to 50 mph, "fast" as a speed about 60 mph and " very fast" as a speed above about 70 mph. Figure 2.1 shows this interpretation in the framework of fuzzy sets.

2.2.3 Set-Theoretic Operation

The set-theoretic operations on fuzzy sets are defined via their membership functions. More specifically, let A and B be two fuzzy sets in $\mathbf{U}$ with membership functions μ_A and μ_B respectively.

DEFINITION 1. *Union: the membership function* $\mu_{A\cup B}$ *of the union* $A \cup B$ *is defined pointwise for all* $u \in \mathbf{U}$ *by* $\mu_{A\cup B}(u) = \max\{\mu_A(u), \mu_B(u)\}$.

DEFINITION 2. *Intersection: the membership function* $\mu_{A\cap B}$ *of the intersection* $A \cap B$ *is defined pointwise for all* $u \in \mathbf{U}$ *by* $\mu_{A\cap B}(u) = \min\{\mu_A(u), \mu_B(u)\}$.

DEFINITION 3. *Fuzzy relation: an n-ary fuzzy relation is a fuzzy set in* $\mathbf{U}_1 \times \cdots \times \mathbf{U}_n$ *and is expressed as* $\{[(u_1, \cdots, u_n), \mu_R(u_1, \cdots, u_n)] | (u_1, \cdots, u_n) \in \mathbf{U}_1 \times \cdots \times \mathbf{U}_n\}$.

DEFINITION 4. *Sup-star composition: if R and S are fuzzy relations in* $\mathbf{U}\times\mathbf{V}$ *and* $\mathbf{V}\times\mathbf{W}$*, respectively the sup-star composition of R and S is a fuzzy relation denoted by* $R\circ S$ *and is defined by:*

$$R\circ S=\{[(u,w),\sup_{v}(\mu_R(u,v)*\mu_S(v,w))],\ u\in\mathbf{U},\ v\in\mathbf{V},\ w\in\mathbf{W}\} \qquad (2.5)$$

*where * could be any operator in the class of triangular norms (t-norms), namely minimum, algebraic product, bounded product or drastic product.*

2.2.4 Approximate Reasoning

A forward data-driven inference, named *generalized modus ponens* (GMP), plays an important role in approximate reasoning. An example of such fuzzy inference involving fuzzy sets A, A', B and B' is illustrated as follows:

Premise 1 : x is A' (2.6)

Premise 2 : if x is A then y is B (2.7)

Consequent: y is B' (2.8)

This type of fuzzy inference is based on the compositional rule of inference, given more specifically as follows:

DEFINITION 5. *Sup-star compositional rule of inference: if R is a fuzzy relation in* $\mathbf{U}\times\mathbf{V}$ *(as the premise 2), and x is a fuzzy set in* $\mathbf{U}$ *(as the premise 1), then the sup-star compositional rule of inference asserts that the fuzzy set y in* $\mathbf{V}$ *induced by x is given by:*

$$y=x\circ R \qquad (2.9)$$

where $x\circ R$ *is the sup-star composition of x and R. If the star represents the minimum operator, this definition reduces to Zadeh's compositional rule of inference.*

2.3 Fuzzy Logic System

Most of the fuzzy logic literature deals with mappings from fuzzy sets into fuzzy sets. In many applications of fuzzy logic to engineering problems, we are interested in mappings from numbers into numbers, and not sets into sets. Consequently, our problem is more difficult than the usual fuzzy logic problem. We have to add a front-end "fuzzifier" and a rear-end "defuzzifier" to the usual fuzzy logic model. The result is a fuzzy logic system (FLS).

Figure 2.2 shows the basic configuration of a fuzzy logic system. The four principal components of a fuzzy logic system are a fuzzifier, rule base, inference engine and defuzzifier. The fuzzifier performs a scale mapping which translates the range of values of input variables into corresponding universes of discourse, as well as a function of fuzzification which converts input data into suitable linguistic values. The fuzzy rule base comprises of a collection of fuzzy IF-THEN rules which describes the relationship between input states and output states. The decision-making logic, the inference engine of the system, emulates human decision-making behaviour based on the principles of approximate reasoning. The defuzzifier takes a fuzzy output from the inference engine and determines a crisp output variable. Moreover, the defuzzifier performs a scale mapping, which converts the range of values of output variables into the corresponding application domain.

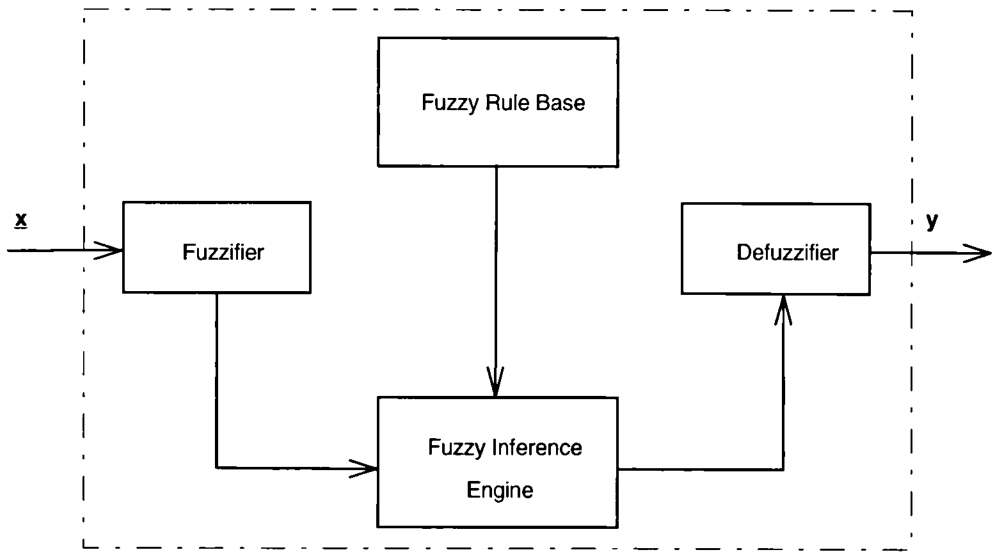


Figure 2.2: Basic Configuration of a Fuzzy Logic System.

2.3.1 Fuzzy Rule Base

The fuzzy rule base consists of a collection of fuzzy IF-THEN rules:

$$R^{(l)}: \text{IF } x_1 \text{ is } F_1^l \text{ and } \cdots \text{ and } x_n \text{ is } F_n^l \text{ THEN } y \text{ is } G^l \tag{2.10}$$

where $\underline{x} = (x_1, \cdots, x_n)^T \in \mathbf{U}$ and $y \in \mathbf{Y}$ are the real inputs and the real output of the fuzzy logic system, respectively, F_i^l and G^l are labels of fuzzy sets in $\mathbf{X}_i$ and $\mathbf{Y}$, respectively, and $l = 1, 2, \cdots, M$. Each fuzzy IF-THEN rule of (2.10) defines a fuzzy implication $F_1^l \times \cdots \times F_n^l \rightarrow G^l$, which is a fuzzy set defined in the product space $\mathbf{U} \times \mathbf{Y}$. Based on generalizations of implications in multivalue logic, many fuzzy implication rules have been proposed in the fuzzy logic literature. Four commonly used fuzzy implication rules are:

- Minimum operation rule of fuzzy implication:

$$\mu_{F_1^l \times \cdots \times F_n^l \rightarrow G^l}(\underline{x}, y) = \min[\mu_{F_1^l \times \cdots \times F_n^l}(\underline{x}), \mu_{G^l}(y)] \tag{2.11}$$

- Product operation rule of fuzzy implication:

$$\mu_{F_1^l \times \cdots \times F_n^l \rightarrow G^l}(\underline{x}, y) = \mu_{F_1^l \times \cdots \times F_n^l}(\underline{x}) \cdot \mu_{G^l}(y) \tag{2.12}$$

- Arithmetic rule of fuzzy implication:

$$\mu_{F_1^l \times \cdots \times F_n^l \rightarrow G^l}(\underline{x}, y) = \min[1,\ 1 - \mu_{F_1^l \times \cdots \times F_n^l}(\underline{x}) + \mu_{G^l}(y)] \tag{2.13}$$

- Max-min rule of fuzzy implication:

$$\mu_{F_1^l \times \cdots \times F_n^l \rightarrow G^l}(\underline{x}, y) = \max\{\min[\mu_{F_1^l \times \cdots \times F_n^l}(\underline{x}), \mu_{G^l}(y)], 1 - \mu_{F_1^l \times \cdots \times F_n^l}(\underline{x})\} \tag{2.14}$$

where $\mu_{F_1^l \times \cdots \times F_n^l}(\underline{x})$ is defined by:

$$\mu_{F_1^l \times \cdots \times F_n^l}(\underline{x}) = \mu_{F_1^l}(x_1) * \cdots * \mu_{F_n^l}(x_n) \tag{2.15}$$

Here the symbol "*" denotes the t-norm, which corresponds to the conjunction "and" in (2.10). The most commonly used operations for the t-norm are:

$$u * v = \begin{cases} \min(u, v) & \text{fuzzy intersection} \\ uv & \text{algebraic product} \\ \max(0, u + v - 1) & \text{bounded product} \end{cases} \tag{2.16}$$

2.3.2 Fuzzy Inference Engine

The fuzzy inference engine performs a mapping from fuzzy sets in **U** to fuzzy sets in **Y**, based upon the fuzzy IF-THEN rules in the fuzzy rule base and the compositional rule of inference. Let A_x be an arbitrary fuzzy set in **U**; then, each $R^{(l)}$ of (2.10) determines a fuzzy set $A_x \circ R^{(l)}$, in **Y** based on the following sup-star compositional rule of inference:

$$\mu_{A_x \circ R^{(l)}}(y) = \sup_{\underline{x} \in U}[\mu_{A_x}(\underline{x}) * \mu_{F_1^l \times \cdots \times F_n^l \to G^l}(\underline{x}, y)] \tag{2.17}$$

where $\mu_{F_1^l \times \cdots \times F_n^l \to G^l}(\underline{x}, y)$ is determined by the fuzzy implication rules of (2.11)-(2.14).

The final fuzzy set, $A_x \circ (R^{(1)}, \cdots, R^{(M)})$, determined by all the M rules in the fuzzy rule base is obtained using fuzzy disjunction:

$$\mu_{A_x \circ (R^{(1)}, \cdots, R^{(M)})}(y) = \mu_{A_x \circ R^{(1)}}(y) \dot{+} \cdots \dot{+} \mu_{A_x \circ R^{(M)}}(y) \tag{2.18}$$

where $\dot{+}$ denotes the t-conorm. The most commonly used operations for $\dot{+}$ are:

$$u \dot{+} v = \begin{cases} \max(u, v) & \text{fuzzy union} \\ u + v - uv & \text{algebraic sum} \\ \min(1, u + v) & \text{bounded sum} \end{cases} \tag{2.19}$$

2.3.3 Fuzzifier

The fuzzifier maps a crisp point $\underline{x} = (x_1, \cdots, x_n)^T \in \mathbf{U}$ into a fuzzy set A_x in **U**. There are (at least) two possible choices of this mapping:

- A_x is a fuzzy singleton with support $\underline{x}$; i.e., $\mu_{A_x}(\underline{x}') = 1$ for $\underline{x}' = \underline{x}$ and $\mu_{A_x}(\underline{x}') = 0$ for all other $\underline{x}' \in \mathbf{U}$ with $\underline{x}' \neq \underline{x}$.
- $\mu_{A_x}(\underline{x}) = 1$ and $\mu_{A_x}(\underline{x}')$ decreases from 1 as $\underline{x}'$ moves away from $\underline{x}$,

In the literature, it seems that only the singleton fuzzifier has been used. However, the nonsingleton fuzzifier may be useful for inputs which are corrupted by noise.

2.3.4 Defuzzifier

The defuzzifier maps fuzzy sets in **Y** to a crisp set in **Y**. There are (at least) two possible choices of this mapping:

- *Maximum defuzzifier*, defined as:

$$y = \arg\sup_{y' \in \mathbf{Y}} [\mu_{A_x \circ (R^{(1)},\cdots,R^{(M)})}(y')] \tag{2.20}$$

where $\mu_{A_x \circ (R^{(1)},\cdots,R^{(M)})}(y')$ is given by (2.18).

- *Centre-average (or centroid) defuzzifier*, defined as:

$$y = \frac{\sum_{l=1}^{M} y^l (\mu_{A_x \circ R^{(l)}}(y^l))}{\sum_{l=1}^{M} (\mu_{A_x \circ R^{(l)}}(y^l))} \tag{2.21}$$

where y^l is the point in **Y** at which $\mu_{G^{(l)}}(y)$ achieves its maximum value and $\mu_{A_x \circ R^{(l)}}(y)$ is given by (2.17).

Note that if we use the center-average defuzzifier, we do not need to calculate the $\mu_{A_x \circ (R^{(1)},\cdots,R^{(M)})}(y)$ of (2.18) (even though this is normally done); we only need to calculate $\mu_{A_x \circ R^{(l)}}(y)$ of (2.17) in the fuzzy inference engine.

2.3.5 Some Specific Fuzzy Logic Systems

From our discussions about the four elements which comprise the FLS shown in Figure 2.2, we see that there are many possibilities to choose from. We must decide on the type of fuzzification (singleton or nonsingleton), functional forms for membership functions (triangular, trapezoidal, Gaussian), parameters of the membership functions (fixed ahead of time, tuned during a training procedure), composition (max-min, max-product), inference (minimum, product), and defuzzifier (maximum, centre-average). This demonstrates the richness of FLS's and there is no such thing as *the* FLS.

If we use the minimim operation in (2.11) and choose * in (2.15) and (2.17) to be a fuzzy intersection, then the inference is called minimum inference. Using the minimum inference, (2.17) becomes:

$$\mu_{A_x \circ R^{(l)}}(y) = \sup_{\underline{x} \in U} \{\min[\mu_{A_x}(\underline{x}), \mu_{F_1^l}(x_1), \cdots, \mu_{F_n^l}(x_n), \mu_{G^l}(y)] \tag{2.22}$$

Fuzzy logic systems with center-average defuzzifier, minimum inference and singleton fuzzifier are of the following form:

$$y(\underline{x}) = \frac{\sum_{l=1}^{M} y^l \left(\min[\mu_{F_1^l}(x_1), \cdots, \mu_{F_n^l}(x_n)] \right)}{\sum_{l=1}^{M} \left(\min[\mu_{F_1^l}(x_1), \cdots, \mu_{F_n^l}(x_n)] \right)} \tag{2.23}$$

where y^l is the point at which μ_{G^l} achieves its maximum value, and we assume that $\mu_{G^l}(y^l) = 1$.

If we use the product operation (2.12) and choose * in (2.15) and (2.17) to be an algebraic product, then the inference is called product inference. Using product inference, (2.17) becomes:

$$\mu_{A_x \circ R^{(l)}}(y) = \sup_{\underline{x} \in U} [\mu_{A_x}(\underline{x}) \mu_{F_1^l}(x_1) \cdots \mu_{F_n^l}(x_n) \mu_{G^l}(y)] \tag{2.24}$$

Fuzzy logic systems with center-average defuzzifier, product inference and singleton fuzzifier are of the following form:

$$y(\underline{x}) = \frac{\sum_{l=1}^{M} y^l \left(\prod_{i=1}^{n} \mu_{F_i^l}(x_i) \right)}{\sum_{l=1}^{M} \left(\prod_{i=1}^{n} \mu_{F_i^l}(x_i) \right)} \tag{2.25}$$

where y^l is the point at which μ_{G^l} achieves its maximum value, and we assume that $\mu_{G^l}(y^l) = 1$.

We can refer to our FLS expressed in the form (2.23) or (2.25) as a *fuzzy basis function expansion*. Doing this is very useful, because it places a FLS into the more global perspective of function approximation.

2.4 Relational Fuzzy Model

Relational fuzzy models preserve the qualitative characteristics of rule-based models but avoid the need for labour-intensive rules development. The following method of representing fuzzy rules is used:

$$R^{(l)} : \text{IF } x_1 \text{ is } F_1^l \text{ and } \cdots \text{ and } x_n \text{ is } F_n^l \text{ THEN } y \text{ is } G^l \text{ with possibility } p^l \tag{2.26}$$

This method of representing fuzzy rules deviates slightly from the form normally used by fuzzy logic systems given by (2.10). This modified rule form allows the relationship between the input and output variables to be represented using relational arrays rather than by a set of rules. The array contains a cell for every possible combination of input and output variables. The content of each cell is a number, with a value between zero and one, which represents the degree of truth of that particular relationship. A value of one indicates that the relationship is at its strongest, and a value of zero indicates that the relationship is at its weakest.

The values inserted in the relational array to produce a relational model is arrived at by identification from i/o data using two general approaches: the linguistic approach and the approach based on resolution of the fuzzy model.

By the linguistic approach, the fuzzy model is constructed from the Cartesian product of i/o data. Linguistic model identification was first proposed by Tong (1978a; 1978b; 1979; 1980) who used "logical examination" to determine which rule best fits a particular data point. Thus, given a data point, the rule with degree of fulfilment greater than some threshold is considered to be a valid rule. Tong used this technique to construct a rule table. The rule table can be thought of as giving the location of the entries in the relational array that equal 1, while all other entries are assumed to equal zero. Tong's results show that a linguistic model may give satisfactory results. Tong's method is, however, difficult to extend to high-dimensional systems and is not suitable for on-line model adaptation.

Pedrycz (1984) proposed a new composition rule and the corresponding fuzzy systems identification algorithm. Most of the relational modelling methods discussed in this thesis are based on the method initially proposed by Pedrycz. To obtain a more accurate model, relational modelling using the linguistic approach normally allows on-line model adaptation.

Various methodologies for the resolution of fuzzy models have been proposed (Czogala and Pedrycz, 1981; Pedrycz, 1983; Lee *et al.*, 1994; Sanchez, 1976; Higashi and Klir, 1984). These approaches suffer from difficulties caused by the fact that the solution is usually not unique and, sometimes, it even does not exist. The computational requirements can also be very high. For these reasons, although several techniques have been proposed, practical applications can hardly be found.

Some relational fuzzy modelling techniques that appear suitable for practical application will now be examined.

2.4.1 Pedrycz Method

Let X_i and Y denote the fuzzified values of each variable x_i and y on universes of discourses $\mathbf{X}_i$ and $\mathbf{Y}$ respectively. Assume each universe of discourse has the same number (=M) of reference fuzzy sets. Then X_i and Y are given by:

$$X_i(l) = \left\{\mu_{F_i^l}(x_i) \middle| l \in \mathbf{m}\right\}, \quad i = 1,2,\cdots,n \text{ and } \mathbf{m} = 1,2,\cdots,M \tag{2.27}$$

$$Y(l) = \left\{\mu_{G^l}(y) \middle| l \in \mathbf{m}\right\}, \quad \mathbf{m} = 1,2\cdots,M \tag{2.28}$$

where X_i and Y are vectors whose elements are the grades of membership of the reference fuzzy sets in their respective universes of discourse.

The predicted output, Y' by a single-output/multiple-input fuzzy relational model using the Pedrycz (1984; 1993) method can be represented by the equation:

$$Y' = \underline{X} \circ R \tag{2.29}$$

where $\underline{X} = (X_1,\cdots,X_n)^T$ and R is a relational array representing the fuzzy relationships between inputs and the output.

Identification of a fuzzy process model involves estimation of the fuzzy relation, R, from process input-output data. For each set of i/o data a family of relational arrays exists which will satisfy the relational equation. Pedrycz has suggested the following scheme for determining R:

$$R_J = \bigcup_{j=0}^{J} (X_1 \times \cdots \times X_n \times Y)_j \text{ where } R_o = 0 \tag{2.30}$$

where $\times$ represents the Cartesian product, which, for this algorithm, is calculated using either the minimum or product compositional operator, and the fuzzy union is interpreted as being the maximum grade of membership.

For prediction, the Cartesian product of the individual grades of membership of the input fuzzy sets is used for max-min composition:

$$Y'(\gamma) = \max_{a \in \mathbf{m}} \max_{b \in \mathbf{m}} \cdots \max_{z \in \mathbf{m}} \min[X_1(a), X_2(b),\cdots, X_n(z), R(a,b,\cdots,z,\gamma)] \tag{2.31}$$

where $Y'(\gamma)$ is the predicted grade of membership corresponding to the output reference set γ.

2.4.2 Xu and Lu Method

Xu and Lu (1987; 1989) have modified Pedrycz's identification method to suit their self-learning algorithm. The prediction method proposed does not directly use the input-set grades of membership for prediction, but instead uses the expression:

$$Y'(\gamma) = \max_{a \in \mathbf{m}} \max_{b \in \mathbf{m}} \cdots \max_{z \in \mathbf{m}} \{R[\lambda_1(a), \lambda_2(b), \cdots, \lambda_n(z), \gamma]\} \tag{2.32}$$

where $\lambda_1(a) \in \lambda_1 = \{l | X_1(l) > q,\ l \in \mathbf{m})\}$, similarly for λ_2, etc., and q is a predefined threshold.

Thus, the predicted output set memberships are determined solely by values in the relational array and are not directly affected by the absolute values of the input memberships. The threshold parameter q limits the relational array entries for prediction to those whose corresponding input variable memberships are larger than the threshold. The q parameter thus gives control of the overall 'fuzziness' of prediction, with low values of q implying high amounts of fuzziness. If the Xu and Lu prediction is used with a q value of zero, every relational array entry will be used for prediction and the predictions produced will be meaningless. If very high values of q are used, prediction success cannot be quaranteed.

Xu and Lu took Pedrycz's identification technique a step further by proposing an algorithm for self learning of the relational model. This self learning was achieved, in an offline form, by making repeated passes through the i/o data series and, within each pass, by modifying the relational array according to:

$$R_j(a,b,\cdots,z,\gamma) = \begin{cases} a_\gamma d_{\gamma j} + (1-a_\gamma) R_{j-1}(a,b,\cdots,z,\gamma) \text{ if } a = \lambda_1', \cdots, z = \lambda_k' \\ R_{j-1}(a,b,\cdots,z,\gamma) \text{ otherwise} \end{cases} \tag{2.33}$$

where λ_1', etc., are the indices to the relational array entries which affected the prediction of the output value, and $d_{\gamma j}$ is defined as:

$$d_{\gamma j} = Y(\gamma) * X_1(a) * \cdots * X_n(z) \tag{2.34}$$

where $*$ represents minimum or product composition. The quantity d_γ is the result Pedrycz's method would identify for the relational array for this particular data subset.

The learning coefficient a_γ is defined to ensure that not only the magnitude of the error in prediction is taken into account, but also the relative contribution to this error from a particular relational cell. The definition used is:

$$a_\gamma = h\beta_\gamma |e_j| \tag{2.35}$$

where h is a tuning constant, $\beta_\gamma (= Y'(\gamma)^2)$ is the relative contribution factor and $e_j (= y_j - y_j')$ is the predictive error.

2.4.3 Graham and Newell Method

This method (Graham and Newell, 1989) is a modified version of the Pedrycz method. Consider a first-order, single-input, single-output fuzzy system which takes the following general form:

$$Y_k' = Y_{k-1} \circ U_{k-d} \circ R \tag{2.36}$$

where the subscript k represents current time, k-1 one sampling time in the past and k-d the dead time of d sampling periods in the past.

The relational matrix at each sampling instant, R' is evaluated as follows:

$$R' = U_{k-d} \times Y_{k-1} \times Y_k \tag{2.37}$$

The overall relational matrix, R is then updated using the following manner:

$$R(i',j',k) = \alpha \cdot R'(i',j',k) + (1-\alpha) \cdot R(i',j',k) \quad k = 1,\cdots,N \tag{2.38}$$

$$R(i,j,k) = \max[R'(i,j,k),\ R(i,j,k)] \text{ for all } i,j,k \text{ except } i = i', j = j' \tag{2.39}$$

where i' and j' denote the positions of the maximum membership values in input vectors U_{k-d} and Y_{k-1}, and α is a scalar constant between 0.5 (good noise rejection) and 1.0 (fastest update). This algorithm can easily be adapted for on-line model updating.

2.4.4 Ridley, Shaw and Kruger Method

Ridley, Shaw and Kruger (Ridley et al, 1988; Shaw and Kruger, 1992) have also adapted Pedrycz's method to make it more suitable for identification in the presence of noise. Given that the entry $R(a,b,\cdots,z,\gamma)$ in the relational array measures the possibility of obtaining an output in reference set $Y(\gamma)$, and given inputs in sets $X_1(a),\cdots,X_n(z)$, a particular array entry is calculated from the equation:

$$R(a,b,\cdots,z,\gamma) = \frac{\sum f_{X_1(a)\times\cdots\times X_n(z)}(k) \cdot Y_k(\gamma)}{\sum f_{X_1(a)\times\cdots\times X_n(z)}(k)} \tag{2.40}$$

where $f_{X_1(a)\times\cdots\times X_n(z)}(k)$ is the product of the possibilities that the individual inputs belong to particular reference sets. This product $f_{X_1(a)\times\cdots\times X_n(z)}(k)$ can be interpreted as the AND combination of a particular group of input reference sets and represents the possibility that the actual input can be represented by this group of sets. The summation is carried out over all of the identification observations. The array Σf, which is used to weight the observations of the possibilities in the output set, therefore represents a 'frequency' measure of how often, and how strongly, a particular set of inputs have occured before. It is this feature which gives this method its excellent resistance to identification noise.

For practical identification purposes, it is easier to create the relational array using one sample of i/o data at a time. It is easy to show that this method can be expressed in the following incremental form:

$$R_m(a,b,\cdots,z,\gamma)=\frac{f_{X_1(a)\times\cdots\times X_n(z)}(m)\cdot Y_m(\gamma)+R_{m-1}(a,b,\cdots,z,\gamma)\sum_{k=1}^{m-1}f_{X_1(a)\times\cdots\times X_n(z)}(k)}{\sum_{k=1}^{m}f_{X_1(a)\times\cdots\times X_n(z)}(k)} \tag{2.41}$$

where R_m and R_{m-1} are the relational array based on m and (m-1) samples of input/output data. This form of the algorithm is also suitable for continuous on-line model identification.

2.4.5 Lee, Hwang and Shih Method

In this method, a two-stage identification procedure which combines the linguistic approach and the approach of numerical resolution of fuzzy relational equation is proposed. In the first stage, a linguistic approach with Pedrycz's algorithm is used to get a fuzzy relation with moderate accuracy. This fuzzy relation is then used as the initial estimate for the second stage in which a recursive algorithm is used to numerically resolve the fuzzy relational equation to obtain a more accurate fuzzy relation. The recursive identification algorithm is based on minimizing a quadratic performance index based on prediction-error. The procedure is particularly suitable for on-line application since all the formulae for computing the fuzzy relation are given in recursive form. Details of this method can be found in the paper by Lee *et al.* (1994).

2.5 Piece-wise Linear Fuzzy Model

Takagi and Sugeno (1985) have proposed a type of fuzzy system where the system output, instead of being a linguistic term, is a numerical function of the input variables. In each of the fuzzy regions of the input space, a function of the following form is used to represent the relationship between the input and output variables:

$$R\text{: } if\ f(x_1\ is\ A_1,\ldots.,x_k\ is\ A_k)\ then\ y = g(x_1,\ldots.,x_k) \tag{2.42}$$

where y is the output, $x_1 - x_k$ are input variables and $A_1 - A_k$ are fuzzy sets with linear membership functions representing a fuzzy subspace in which implication R can be applied for reasoning.

If a linear model structure is assumed, the implications take the following format:

$$\begin{aligned} R^1 &: if\ x_1\ \text{is}\ A_1^1 and \cdots x_k\ is\ A_k^1\ then\ y = p_0^1 + p_1^1 x_1 + \ldots + p_k^1 x_k \\ &\vdots \quad \vdots \\ R^n &: if\ x_1\ \text{is}\ A_1^n and \ldots x_k\ is\ A_k^n\ then\ y = p_0^n + p_1^n x_1 + \ldots + p_k^n x_k \end{aligned} \tag{2.43}$$

Then the output y for the input $(x_1,\cdots,x_k)$ is obtained as:

$$y = \frac{\sum_{i=1}^{n}(A_1^i(x_1)\cap\cdots\cap A_k^i(x_k))(p_0^i + p_1^i x_1 + \cdots + p_k^i x_k)}{\sum_{i=1}^{n}(A_1^i(x_1)\cap\cdots\cap A_k^i(x_k))} \tag{2.44}$$

where $\cap$ is the conjunction operator.

Let β_i be:

$$\beta_i = \frac{(A_1^i(x_1)\cap\cdots\cap A_k^i(x_k)}{\sum_{i=1}^{n}(A_1^i(x_1)\cap\cdots\cap A_k^i(x_k))} \tag{2.45}$$

Then,

$$y = \sum_{i=1}^{n} \beta_i(p_0^i + p_1^i x_1 + \cdots + p_k^i x_k) \tag{2.46}$$

or

$$y = \sum_{i=1}^{n}(p_0^i\beta_i + p_1^i x_1\beta_i + \cdots + p_k^i x_k\beta_i) \tag{2.47}$$

When a set of input-output data $x_{1j}, x_{2j},\ldots., x_{kj} \to y_j (j = 1,\ldots.,m)$ is given, we can obtain the consequence parameters $p_0^i, p_1^i,\ldots., p_k^i (i = 1,\ldots.,n)$ by the least squares method using the above equation.

Let X (m x $n(k+1)$ matrix), Y (m vector) and P ($n(k+1)$ vector) be:

$$X = \begin{bmatrix} \beta_{11} \cdots \beta_{n1} & x_{11}\beta_{11} \cdots x_{11}\beta_{n1} & \cdots & x_{k1}\beta_{11} \cdots x_{k1}\beta_{n1} \\ \vdots & \vdots & \vdots & \vdots \\ \beta_{1m} \cdots \beta_{nm} & x_{1m}\beta_{1m} \cdots x_{1m}\beta_{nm} & \cdots & x_{km}\beta_{1m} \cdots x_{km}\beta_{nm} \end{bmatrix} \tag{2.48}$$

where:

$$\beta_{ij} = \frac{A_1^i(x_{1j}) \cap \cdots \cap A_k^i(x_{kj})}{\sum_j (A_1^i(x_{ij}) \cap \cdots \cap A_k^i(x_{kj}))} \tag{2.49}$$

$$Y = [y_1, \ldots, y_m]^T \tag{2.50}$$

$$P = [p_0^1, \ldots, p_0^n, p_1^1, \ldots, p_1^n, \ldots, p_k^1, \ldots, p_k^n]^T \tag{2.51}$$

Then the parameter vector P is calculated by:

$$P = (X^T X)^{-1} X^T Y \tag{2.52}$$

It is convenient to transform the matrix P into the following form

$$P_m = \begin{bmatrix} p_0^1 & p_1^1 & \cdots & p_k^1 \\ \vdots & \vdots & \vdots & \vdots \\ p_0^n & p_1^n & \cdots & p_k^n \end{bmatrix} \tag{2.53}$$

which gives a more clear representation of the fuzzy model. Each row in the above matrix gives the parameters of the linear equation in each fuzzy sub-space.

The method as proposed by Takagi and Sugeno involves an iterative search to determine the best model structure, the optimum fuzzy partitioning, and parameter estimation. The overall model fit is assessed using a performance index such as the mean square of the prediction errors based on the test data.

2.6 Comparison of Fuzzy Modelling Techniques

The measures used for comparing fuzzy modelling techniques will depend on the intended application of the fuzzy modelling technique. Since, we are primarily concerned with the use of fuzzy models for model based predictive control, the important attributes are modelling accuracy, on-line learning capability and computational requirements. Also important, is the ability to be easily incorporated in the design of the controller.

The main criticism of rule-based fuzzy models is the amount of time and effort required to develop the fuzzy rule base using just heuristic knowledge. Some information may also be lost when humans express their knowledge using linguistic rules. Wang and Mendel (1992) have suggested supplementing this approach with sampled input-output pairs. An empirical approach is proposed for generating fuzzy rules from numerical data, whereby each generated rule is asigned a degree by an expert that represents his belief of its usefulness. In the case of conflicting rules, i.e., rules that have the same IF part but a different THEN part, only the rule with maximum degree is accepted. The mapping from the input space to the output space of the combined rule base has been proved to be capable of approximating any real continuous function on a compact set to arbitrary accuracy.

Recent research seems to suggest that neural network learning algorithms (Nie and Linkens, 1995; Wang, 1994) and clustering techniques (Kosko, 1992) can be useful for developing the fuzzy rule base. In fact, these methods may be preferred over relational modelling methods since it allows heuristic knowledge to be utilised which is considered to be an important aspect of fuzzy modelling. We are not aware, however, of any study carried out to compare neurofuzzy modelling approaches with relational fuzzy modelling approaches.

A number of comparative studies have been published using the gas furnace data from Box and Jenkins (1970). This data set consists of 296 pairs of input-output observations where the input is the gas flow rate into the furnace and the output is the concentration of carbon dioxide in the exhaust gas. Table 2.1 compares some of the methods discussed above from the point of view of modelling accuracy based on this data set. The performance index used is the mean square error in prediction. It can be observed that the piece-wise linear modelling approach proposed by Takagi and Sugeno offers the best accuracy.

For applications where on-line model modification is essential, methods based on relational models can be considered. A study to compare Pedrycz and Xu and Lu methods has been carried out by Postlethwaite (1991a). The results show that max-product composition produces better results than max-min composition regardless of the identification method used. The results also show that the best performances using both methods are very similar.

Method	Number of reference sets	Mean square error
Tong (1978)	7	0.469
Xu and Lu (wihout learning) (Xu and Lu,1987; Xu , 1989)	5	0.456
Xu and Lu (with learning) (Xu and Lu,1987; Xu , 1989)	5	0.358
Pedrycz (1984)	9	0.320
Pedrycz (1984)	7	0.478
Pedrycz (1984)	5	0.776
Lee, Hwang and Shih (max-min method) (1994)	5	0.653
Lee, Hwang and Shih (max-prod method - 1 iteration) (1994)	5	0.407
Lee, Hwang and Shih (max-prod method - 6 iterations) (1994)	5	0.211
Ridley, Shaw and Kruger (1988)	5	0.289
Takagi and Sugeno (piece-wise linear model) (1985)	2	0.068

Table 2.1: Comparison of the accuracy of fuzzy modelling techniques applied to Box and Jenkins (1970) process data.

Relational modelling approaches generally require significant computational effort, especially if the number of variables and number of reference fuzzy sets used are great. Postlethwaite evaluated the Pedrycz method by using the q-factor concept proposed by Xu and Lu. The q-factor which seemed to give the best results with both methods was the largest tested (i.e. 0.5). This is convenient in the sense that the larger the q-factor, the less the work the prediction algorithm has to do. It also indicates that the trace effects at low grades of membership can reduce the accuracy of the prediction algorithm. The Pedrycz algorithm was also found to be more stable to changes in the q-factor.

Postlethwaite also evaluated a modified version of the Xu and Lu algorithm for self-learning incorporating the Pedrycz method. His results showed that the two approaches performed similarly at high values of q, but at low values of the factor the Xu and Lu algorithm was noticeably inferior. This is likely to be a problem in practical applications since it adds an additional variable which must be considered to develop a good quality model.

Other important observations from the Postlethwaite study are that a relational model developed using self-learning from input/output data can equal the performance of a rule-based model and that relational modelling approaches showed excellent tolerance to noise up to 20 percent level.

The method proposed by Lee, Hwang and Shih (1994) showed a significant improvement in accuracy from 0.407 to 0.211 when the number of iterations using the training data set was increased from 1 to 6. This study also confirms the superiority of max-product composition over max-min composition.

The method proposed by Ridley, Shaw and Kruger (1988) is able to achieve an accuracy of 0.289 with just one iteration. Other tests carried out with this method using data not used for training shows that it has good predictive capability. The computational requirements are also not very great. For these reasons, it appears superior to other relational methods for applications where on-line model learning capability is desired.

Even though relational models are quite easily developed and modified on-line, this advantage must be viewed in the context of their known limitations. Firstly, their use is normally limited to systems with a small number of variables in view of their large size and computing requirements. A first-order relational model of a system consisting of 2-inputs and 1-output, where 7 reference sets are used by each variable, will generate an array consisting of 2401 elements. Another problem posed by relational models is that there is no simple approach for deriving the controller output analytically, making it necessary to resort to numerical approaches which add to the already large computing requirements of the model. Also important are the controller problems that can arise as a result of incomplete rule bases. These problems are further discussed in Chapter 3 after examining the control strategies incorporating relational fuzzy models.

Since the piece-wise linear form of fuzzy rule representation proposed by Takagi and Sugeno contains more information, the number of rules required will typically be several orders of magnitude less than relational fuzzy models. A complex high dimensional nonlinear modelling problem is decomposed into a set of simpler linear models valid within certain operating regimes defined by fuzzy boundaries. Fuzzy inference is then used to interpolate the outputs of the local models in a smooth fashion to get a global model. The computation requirements after identification are quite reasonable. Studies (Sugeno and Yasukawa, 1993) carried out using the Box and Jenkins (1970) gas furnace data have also showed this modelling approach to be generally more accurate than relational modelling approaches. Another advantage of this model is that an output will always be generated whatever the value of the inputs. Hence, it is also more suited for use in controller design than are relational models.

The above fuzzy model representation has more in common with traditional models used in model-based control than fuzzy relational models. This similarity makes it possible to consider adopting at least some model-based control methods to work with the fuzzy model. Also, as demonstrated by Takagi and Sugeno, standard identification techniques such as the method of least squares can be quite easily applied for determining the model parameters. In its original formulation it cannot be developed on-line. More recent studies by Johansen (1994) and Wang and Langari (1994) have shown that recursive parameter estimation techniques can be used for adaptive modelling. Jang (1993) has shown that there is a functional equivalence between this model representation and radial basis function neural networks making it possible for learning algorithms proposed for neural networks to be also applied here.

The piece-wise linear modelling approach does not provide rules that can be expressed linguistically. As such, it may be criticized that this technique would be difficult to use interactively with a human in the loop, making it difficult to update and modify using heuristic knowledge. Techniques based on relational modelling approaches have a similar difficulty.

Chapter 3

FUZZY MODEL BASED CONTROL

3.1 Introduction

Automatic process control algorithms based on the theory of fuzzy sets have been successfully developed over the last decade or so. These algorithms consist of a set of linguistic 'what to do' rules which can be represented and evaluated on a process control computer using fuzzy sets and the associated fuzzy logic. The rationale behind this approach to process control is to be able to control processes which are complex and ill-defined, and consequently for which an accurate mathematical model does not exist. Fuzzy control algorithms can be built from purely heuristic knowledge such as a human operator's control strategy.

A number of difficulties face the designer of any fuzzy control algorithm. Firstly, for any reasonably complex process the number of rules required to ensure adequate control in all operating regions may be extremely large. Simply eliciting all these rules from a source such as an operator is a difficult task. Secondly, when faced with an initial rule set containing anything up to a few thousand rules the designer must ensure that the rules are consistent and complete. Consistency means that no two rules are in conflict such that they have the same antecedents but different consequents. A rule set is complete when every possible input state is represented. Ensuring consistency and completeness is a daunting task for a large rule set. These two problems are the most important in the design of a fuzzy controller.

Two approaches to solving these problems have emerged: self-organising controllers and process model-based controllers. The self-organising controller of Procyk and Mamdani (1979) develops a guaranteed consistent rule set on-line that will satisfy a pre-determined acceptable controller response. This self-organising controller can be started with no rules at all and will quite quickly learn an adequate rule set. However, the designer must specify the desired response in a fashion that can identify a rule that violates the desired response, and then modify the rule so that it is more nearly correct. This requires at least a crude incremental model of the process, and knowledge of the process dead times.

The alternative approach is to model the process rather than the operator and then to incorporate the process model into some sort of model-based control scheme. This approach has the following advantages:

- It is easier to obtain information on how a process responds to particular inputs than it is to record how and why an operator responds to particular situations.
- The modelling procedure and controller tuning are completely separated.
- The model-based control approach provides flexibility and the means to make the controller goal seeking rather than simply responsive.
- The advantages of fuzzy reasoning are maintained.

Model based control has been used very widely in the chemical and petrochemical industries. Because of the difficulties of using or obtaining first-principles models, the usual procedure is to develop a linear model of the plant based on empirical data, and to use that model within an optimisation routine. An alternative approach is to use the same empirical data to develop an intelligent model using neural network or fuzzy modelling techniques. A number of attempts have been made to develop fuzzy model based controllers. Most are based on 1-step ahead predictions using relational fuzzy models which enable on-line model adaptation. A generic model based control strategy consists of the following steps, which is repeated at every sampling instant:

- Measure the plant variables. These include manipulated variables that can be varied to achieve the process goals, measured disturbances that affect the operation of the process but cannot be regulated, and output variables for which we have specified objectives (usually given in terms of set points or inequality constraints).
- Use the measurements and a model of the process to estimate the current process state.
- Calculate new settings of the manipulated variables that are optimal with respect to specified objectives.
- Send the calculated settings to the plant and wait for the beginning of the next cycle. It is assumed that a zero-order hold latches the manipulated variables at these values for the duration of the sampling period.

The internal model serves a dual purpose: it estimates the current state of the process and predicts future plant outputs as a function of past and future (anticipated) inputs and outputs. There are no *a priori* restrictions on the form of this model. Difficulties in using traditional cost function and optimisation search techniques with some types of fuzzy models have led to some unique controller formulations.

This chapter examines some fuzzy model based control approaches that have been proposed in the literature, with particular emphasis to chemical processes. An attempt is also made to compare some of these approaches.

3.2 Graham and Newell Method

Figure 3.1 shows the general structure of the fuzzy model-based controller used by Graham and Newell (1988; 1989). There is a fixed set of possible control actions from which to choose. The fuzzy model is used to predict what the output would be for each of these actions. A decision maker then selects the most favourable action to take, for example the one that results in the smallest error. The selected control action is then applied to the process and the whole procedure repeats itself each sampling interval. The set of possible control actions must be selected to give sufficient, yet fine enough control.

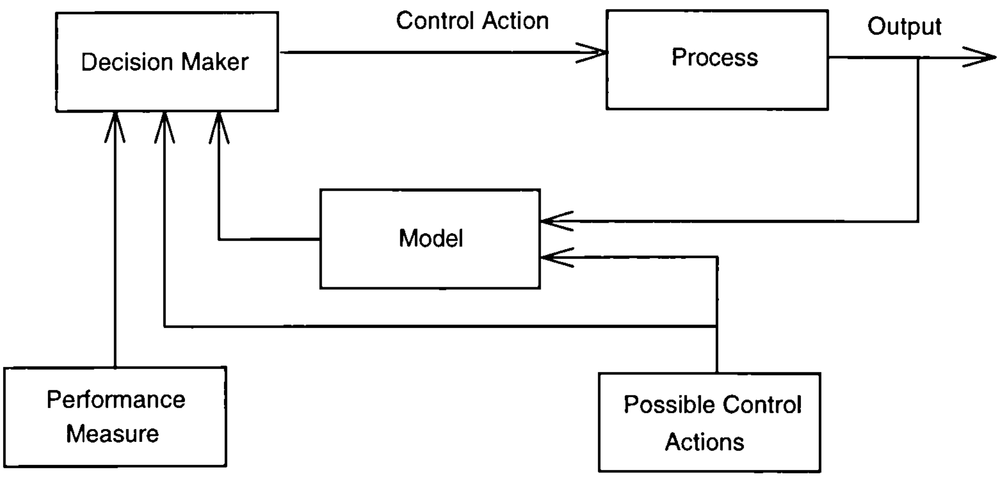


Figure 3.1: Fuzzy Model based Controller Based on Graham and Newell Method

Graham and Newell have used this technique for liquid level control using a simulation as well as a laboratory liquid level rig. For fuzzy modelling, they have tried both rule-based and relational models. Rule-based models were generated using a method similar to that proposed by Tong (1980). The relational fuzzy modelling method used here has been presented in Section 2.4.3. It was observed that relational models tended to perform better than rule-based models. To achieve high resolution in their model for their liquid level rig application, they have used a large number of reference fuzzy sets to describe the process inputs and output. Due to problems with model size, they were forced to resort to hybrid fuzzy/conventional models.

3.3 Postlethwaite Method

Postlethwaite (1991b; 1994; 1996) has used the standard Internal Model Controller (IMC) structure (Figure 1.3). For an open loop stable process and an open loop stable controller, Garcia and Morari (1982; 1989) have shown that the nominal closed-loop system performance of the IMC structure is guaranteed to be stable. Fuzzy model identification and self-learning has been carried out using the method proposed by Ridley, Shaw and Kruger (1988). The use of relational fuzzy models using the IMC structure creates problems in that it is difficult to invert fuzzy models to obtain a controller equation. The controller has therefore been formulated using a cost function and a numerical optimiser. The optimiser uses a direct numerical optimisation search technique called Fibonacci search to minimise the cost function. A method for handling fed-back model error wherein future predictions are adjusted by adding a time-filtered error term has been proposed. The correction mechanism increases the initial rate of rise and eliminates the steady-state error, although this is achieved at the cost of a slightly more oscillatory response.

In a subsequent paper, Sing and Poslethwaite (1996) have used multivariable optimisation methods such as simulated annealing (SA), threshold accepting (TA) and iterative improvement to achieve better modelling accuracy as compared to Ridley, Shaw and Kruger's method. The main limitation of these new methods was found to be the significantly greater computational effort.

3.4 Skrjanc and Matko Method

Skrjanc and Matko (1993) have proposed a fuzzy predictive controller with adaptive gain which uses the piece-wise linear fuzzy modelling approach proposed by Takagi and Sugeno. The information provided by the predictions is used to calculate the value of a multi-step penalty function, J, at sampling instant, k:

$$J = \sum_{i=N_s}^{N_h} [w(k+i) - \hat{y}(k+i)]^2 \tag{3.1}$$

where $\hat{y}$ is the predicted output, w is the reference signal, and N_s and N_h are the minimum and maximum values of the prediction horizon. An alternative formulation to minimise variations in the in the controller output has also been proposed:

$$J = \sum_{i=N_s}^{N_h} [w(k+i) - \hat{y}(k+i)]^2 + \sum_{i=1}^{N_c} \rho \Delta u(k+i-1)^2 \tag{3.2}$$

where N_c is the control horizon and ρ is a weighting constant.

Information about the actual value of the error $e(k) = w(k) - y(k)$ is added to the penalty function. According to the value of $e(k)$, the penalty function is modified as follows:

$$\textit{if } e(k) \geq 0 \textit{ then } J = J \tag{3.3}$$

$$\textit{if } e(k) < 0 \textit{ then } J = -J \tag{3.4}$$

The value of the control signal that would attempt to bring J to zero at the next sampling instant is then determined using fuzzy logic. The fuzzy rules for determining the control signal are based on J and ΔJ and have been expressed in the form of the familiar rule table.

An adaptive mechanism to vary the fuzzy controller output scaling factor based on the sum of the past errors has also been considered.

Even though a multistep fuzzy predictive controller has been proposed using the simplifying assumption that the controller output remains constant over the entire prediction horizon, simulation experiments have been limited to prediction and control horizons of one. The results show that the predictive controller produces a less oscillatory response than a comparable fuzzy logic controller.

3.5 Other Fuzzy Model Based Controllers

A multi-objective predictive fuzzy control scheme that uses a collection of fuzzy rules based on a skilled human operator's experience is described by Yasunobu and Miyamoto (1985). The controller selects the most likely control command based on predictions of control results and direct evaluations of the control objectives. This scheme has been applied to automatic train operation (ATO) systems which control trains by evaluating safety, riding comfort, accuracy of stop gap, running time and energy consumption. Simulation results have revealed significant improvements over conventional PID-based ATO systems and field tests have shown that the controller can operate trains as skilfully as a human expert.

Yamazaki (1993; 1994) has proposed a fuzzy model learning predictive controller by applying a fuzzy inverse relation for deriving desirable control actions along with a model rule learning algorithm for tuning the process model. The process model is represented by a set of fuzzy rules which define explicitly the relationship between control actions and their future process responses. Due to this rule representation, the design of the controller is re-defined as an inverse problem of fuzzy relational equations with one unknown fuzzy variable, i.e. the control action, if the desirable future process response is given in the form of a reference trajectory.

The indirect self-organising controller proposed by Moore and Harris (1992) also involves a process model. The relational fuzzy process model is identified on-line and then inverted to determine the controller rules using a different approach to Yamazaki.

Wang (1994) has examined various approaches including the feedback linearising control strategy and the use of Lyapunov synthesis approach to design direct and indirect adaptive controllers.

Johansen (1994) has proposed an adaptive multivariable controller using the feedback linearisation concept and the piece-wise linear fuzzy modelling method proposed by Takagi and Sugeno.. In this controller, the fuzzy model is numerically inverted and used as the inner level controller. The outer level controller consists of simple integrators. A major drawback of feedback linearisation as compared to long-range predictive control is that it is less suitable for systems with significant nonminimum phase effects.

Nakamori (1991; 1994) has suggested transforming the linear auto-regressive with exogenous inputs (ARX) model structure used by the piece-wise linear fuzzy model into an impulse response form to facilitate the design of a long-range predictive controller.

Braake *et al.* (1994) describe the application of a model based predictive control scheme (MBPC) using fuzzy and neural network models to control the pressure in a laboratory-scale fermenter. The controller uses standard numerical optimisation methods such as the Nelder-Mead algorithm and the piece-wise linear fuzzy modelling approach. It is shown that the process output response of a PI controller is slower than their proposed MBPC scheme, especially for large setpoint changes. Tuning the PI controller was also found to be generally more difficult.

3.6 Comparison of Some Fuzzy Model Based Controllers

Various fuzzy model based controllers have so far been examined. Of particular interest to us are those proposed by Graham and Newell, Postlethwaite, and Skrjanc and Matko. Each one is based on a different fuzzy model identification technique and has used a different approach for determining the controller output based on the predictions from the fuzzy model. It is interesting to try and make a comparison of these three approaches.

Postlethwaite's controller shows a significant performance improvement to the controller proposed by Graham and Newell when used used to control level using a similar simulation. It is not totally clear if the improvement is due to the differences in the control approach only. A major difference is that Graham and Newell have expressed their model in terms of deviation variables (i.e., the fuzzy model relates the current change in error and current change in control action to the change in error at the next sample time) whereas Postlethwaite has used the actual values of the process variables. Graham and Newell's reasons for doing this are to increase the model resolution and thereby have a smaller number of reference fuzzy sets. However, this method of modelling means that if the operating conditions change, the relationship between deviation variables may change markedly as a result of the nonlinear process behaviour. To compensate for this loss of accuracy, on-line model adaptation is essential. Postlethwaite's results show that it is not necessary to define a large number of reference sets to achieve good quality control.

Both Postlethwaite and Graham and Newell have emphasized on-line model building and adaptation using relational fuzzy models. If the identification does not cover a particular combination of inputs, relational models will be completely unable to make any prediction. The problem of model completeness is a major problem with relational model-based controllers. This problem is exacerbated if a large number of reference sets are used to achieve a high resolution in fuzzification. Also, the optimisation method used by Postlethwaite initially searches towards the boundaries of the manipulation, which may be outside the normal operating envelope of the process and thus not included in the input/output data used for identification. The effect of this inability to predict is to render the controller unable to make any sensible controller output choice. There are two ways out of this problem. The first is to ensure that the relational model that the controller uses is complete - but this is not realistic for complex systems. The second is to include additional logic in the controller to allow it to generate a reasonable output when it runs into a problem. This is the approach used by Graham and Newell.

Both studies show that fuzzy model-based controllers perform well in noisy environments. Studies by Graham and Newell show that the non-adaptive controller to be particularly robust to noise. Model adaptation improved the response of the controller when there is no noise, but the controller generally performed worse under noisy conditions especially when the learning rate was set close to 1. This was due to many contradictory and incorrect rules being identified. The policy of completely replacing old rules with new rules leads to a constantly changing model that may contain many incorrect rules. Good control was generally maintained when the learning rate α in equation (2.38) was below 0.5.

In the study on the liquid level rig conducted by Graham and Newell and the study on the laboratory heat exchanger conducted by Postlethwaite, one of the inputs to the fuzzy process model was a disturbance. It is shown in both cases that the fuzzy model-based controller is able to outperform a feedforward/feedback PI controller, especially in noisy environments. By including a disturbance as an input to the fuzzy model, it is possible to compensate for the effect of the disturbance.

The Skrjanc and Matko method has used the piece-wise linear modelling approach proposed by Takagi and Sugeno. The advantages of this modelling approach over relational modelling approaches has been discussed in Section 2.6 and will therefore not be repeated here.

The controller formulations used in each of the above three controllers is different. The aim of each controller is to minimize the error over the next sampling interval. Each controller uses a slightly different approach to achieve this.

A problem posed by relational fuzzy models is that there is no simple approach for deriving the optimal controller output directly from the model. Graham and Newell have used a set of nine possible changes to the control action. Each of these are tested using the process model, and the change that gives the mimimum predicted error in the process output is chosen as the current change in control output. An obvious disadvantage of this approach is that the controller may be severely restricted in its ability to provide fine control. Postlethwaite has used a numerical optimisation technique working in conjunction with the process model to determine the control output that will give the minimum predicted error. While this method seems to be an improvement on the method proposed by Graham and Newell, numerical optimisation techniques rely on the objective function being unimodal (i.e. having a single maximum or minimum). In reality, it is rather difficult to guarantee this with nonlinear process models. The problem with multi-modal cost functions is that the optimum the optimiser finds will depend on the direction of approach. It is conceivable therefore that the controller could converge on different optimum responses at each sampling instant, resulting in considerable oscillation in the manipulated variable. It is also necessary to appreciate the increased computational requirement of the Postlethwaite method. At each sampling instant, 20 optimisation experiments using the relational process model have to be carried out before the controller output can be determined. This may lead to a lower rate of sampling, restricting its use to slow processes only.

Skrjanc and Matko have not attempted to capitalise on the analogous structure of the fuzzy model to linear process models in the design of their controller. Instead, their controller is an extension of the commonly used fuzzy logic controller (FLC). The process model is first used to make predictions. In the next step, the control signal is selected which brings the predicted process output signal back to the reference signal in a way to minimise the area between the reference and output signals. Even though, multi-step versions have been proposed, Skrjanc and Matko have evaluated their controller concept using only a prediction horizon of 1-step. Since this controller's performance has not been evaluated using a similar control application as the other two controllers, it is not possible to comment on its comparative performance. It should be noted, however, that unlike the other controllers, the calculations required at each sampling instant is very minimal. The main problems with this controller are that controller tuning is not likely to be any easier than a normal fuzzy controller; and steady-state errors are likely since the controller structure does not take into account modelling errors. It is also felt that extending the concept to a multi-step prediction horizon is not likely to be easy.

Chapter 4

IDENTIFICATION OF FUZZY PROCESS MODEL

4.1 Introduction

Before a fuzzy model based controller can be designed, it is first necessary to decide on the fuzzy modelling approach to be used. The discussion in Chapter 2 has highlighted two main fuzzy modelling approaches based on the method used to represent the fuzzy rules. Fuzzy models are most commonly expressed using collections of fuzzy *IF-THEN* rules of the following form:

$$IF\ x_1\ is\ B^1\ and \dots and\ x_n\ is\ B^n\ THEN\ y\ is\ C \tag{4.1}$$

where $\underline{x} = (x_1, \dots, x_n)^T$ and y are the input and output linguistic variables respectively, and B^i and C are linguistic values characterised using membership functions. It is claimed that this fuzzy rule representation provides a convenient framework to incorporate human experts' knowledge. Systems consisting of many rules are more conveniently expressed using relational arrays.

Another form of expressing fuzzy rules has been proposed by Takagi and Sugeno (1985) and has fuzzy sets only in the premise part and a linear model as the conclusion:

$$IF\ x_1\ is\ B^1\ and \dots and\ x_n\ is\ B^n\ THEN\ y = c_0 + c_1 x_1 + \cdots + c_n x_n \tag{4.2}$$

where $\underline{x} = (x_1, \dots, x_n)^T$, y and B^i are similarly defined as above, and c_i are real-valued parameters. Since this form of rule representation contains more information, the number of rules required will typically be several orders of magnitude less than fuzzy relational models. A complex high dimensional nonlinear modelling problem is decomposed into a set of simpler linear models valid within certain operating regimes defined by fuzzy boundaries. Fuzzy inference is then used to interpolate the outputs of the local models in a smooth fashion to get a global model. The advantages of this fuzzy model over relational fuzzy models have been pointed out in Chapter 2. The similarity to the traditional linear models used in control engineering makes it particularly attractive, since it may be possible to adopt control strategies based on linear models to work with this model without too much difficulty.

In this chapter we will examine the application of the above fuzzy modelling method to model nonlinear chemical processes using simulations of the level of liquid in a tank and a continuous stirred tank reactor (CSTR) system.

4.2 Fuzzy Process Model

Consider a single-input single-output system which can be modelled using the Takagi-Sugeno modelling approach. Although this modelling approach can accommodate input space partitioning based on more than one variable (i.e., multiple premise variables), we shall limit its usage here to systems where input space partitioning is based on just the current value of the output. It is assumed that the input space is partitioned using p fuzzy partitions and that the system can be represented by fuzzy implications (one in each fuzzy sub-space) of the following form:

$$\begin{aligned} L^i:\ IF\ y(t)\ is\ B^i\ THEN\ y_m(t+1) &= a_1^i y(t) + \cdots + a_j^i y(t-j+1) \\ &+ b_1^i u(t) + \cdots + b_l^i u(t-l+1) + k_i \end{aligned} \tag{4.3}$$

where $y(t)$ and $u(t)$ are the process and controller outputs at time t, $y_m(t+1)$ is the 1-step ahead model prediction at time t, B^i is a fuzzy set representing the fuzzy sub-space in which implication L^i can be applied for reasoning, and $i = 1,...,p$. The fuzzy model parameters can be expressed in the following matrix form:

$$\Phi = \begin{bmatrix} a_1^1 \cdots a_j^1 & b_1^1 \cdots b_l^1 & k_1 \\ \vdots & \vdots & \vdots \\ a_1^p \cdots a_j^p & b_1^p \cdots b_l^p & k_p \end{bmatrix} \tag{4.4}$$

When a set of input-output data is given, the model parameters can be calculated using the method of least squares as explained in Chapter 2. The method as proposed by Takagi and Sugeno involves an iterative search to determine the best model structure, the optimum fuzzy partitioning and parameter estimation. The overall model fit is assessed using a performance index such as the mean square error (MSE) of the prediction errors based on the test data. Details of the modelling approach are available from the paper by Takagi and Sugeno (1985).

The search to determine the optimum fuzzy partitioning points can be initiated by dividing the input space uniformly. Changes are then made and the knowledge acquired from earlier changes used to guide subsequent changes, emphasizing changes which reduce the MSE. When the fuzzy partitioning is close to the optimum, small changes in the fuzzy partitioning points should not lead to significant changes in the fuzzy model parameters or the MSE. The fuzzy model parameters can be verified using a second input/output data set.

It is possible to express the overall fuzzy model output in the following form:

$$y_m(t+1) = \beta\Phi X(t) \tag{4.5}$$

where,

$$X(t) = \left[y(t)\cdots y(t-j+1) \quad u(t)\cdots u(t-l+1) \quad 1\right]^T \tag{4.6}$$

$$\beta = \left[\beta_1\cdots\beta_i\cdots\beta_p\right] \tag{4.7}$$

and

$$\beta_i = \frac{B^i\left[y(t)\right]}{\sum_{i=1}^{p} B^i\left[y(t)\right]} \tag{4.8}$$

$B^i\left[y(t)\right]$ is the grade of membership of $y(t)$ in B^i and β is a vector of the weights assigned to each of the p implications at each sampling instant.

4.3 Liquid Level System

4.3.1 Mathematical Model

The system being modelled consists of the level of liquid in a tank with a manipulated inflow, and an outflow which is dependent on the square root of the level in the tank (Figure 4.1). The dynamic model of this process is just a single, non-linear, differential equation:

$$A\frac{dh}{dt}+\beta\sqrt{h}=F_i \tag{4.9}$$

where,

h = liquid level in the tank (the controlled variable)

A = cross-sectional area of the tank (constant and equal to 10^{-3} m^2)

β = a flow coefficient (equal to 1)

F_i = the inlet flowrate (the manipulated variable).

The open loop response of the liquid level to changes in inlet flowrate introduced 100 seconds after the start of simulation (Figure 4.2) shows clearly the effects of nonlinearity over the range from 0 to 100 cm. By linearising equation (4.9) around the steady-state, and taking deviation variables, the following first-order equation is obtained:

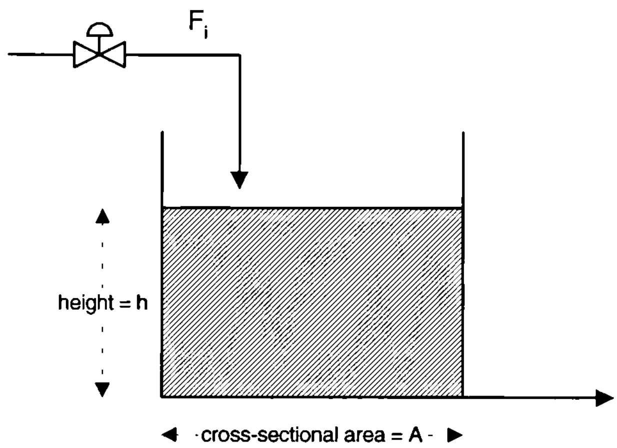


Figure 4.1: Liquid level system.

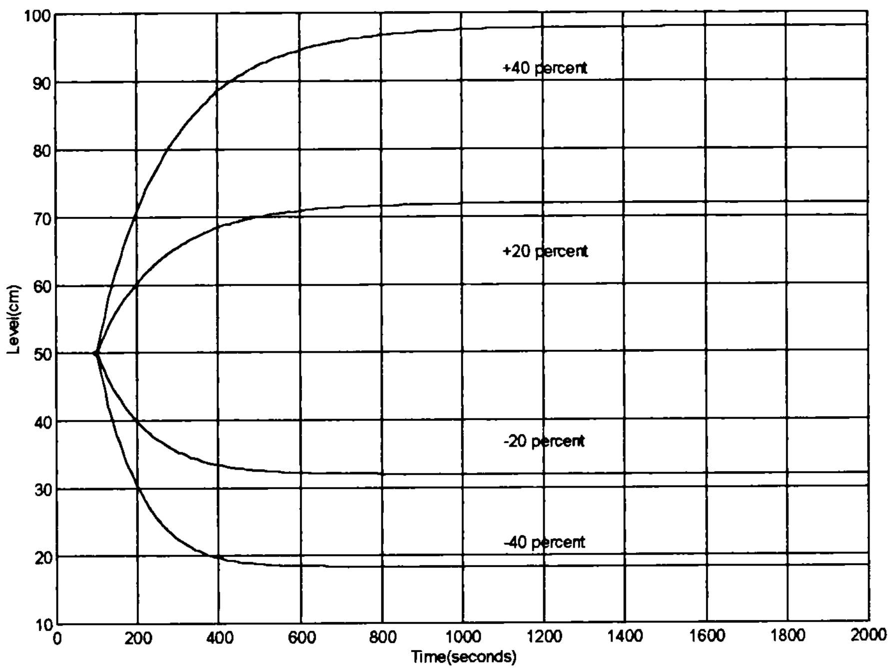


Figure 4.2: Open loop response of the liquid level to step changes in inlet flowrate.

$$\tau_p \cdot \frac{dh'}{dt} + h' = K_p \cdot F_i' \tag{4.10}$$

where,

$$\tau_p = 2A\sqrt{h_s}/\beta \tag{4.11}$$

$$K_p = 2\sqrt{h_s}/\beta \tag{4.12}$$

and the primed variables are deviation variables; h_s is the steady-state height; τ_p is the time constant; and K_p is the process gain. The time constant varies from a value of 63.25 seconds at a height of 10 cm. to 194.94 seconds at a height of 95 cm. The corresponding values of the process gain are 6.32 and 19.49.

4.3.2 Identification of Fuzzy Process Model

In this section, we will examine the application of the fuzzy modelling technique presented in Section 4.2 using a simulation of the liquid level system in Section 4.3.1.

Data for modelling was generated by applying 25 random step changes, each lasting 1000 seconds, in the inlet flowrate as shown in Figure 4.3 while maintaining the level in the tank approximately between 0 and 100 cm. It will be observed in Figure 4.3 that the level in the tank often reaches steady-state conditions, thus ensuring that the low frequency range is sufficiently stimulated. Sampling was carried out at 10 second intervals. A total of 2500 data points were used for identification.

Since the system being studied is approximately a first-order system, the following fuzzy model structure has been assumed:

$$L^i: \text{ if } h(t) \text{ is } A^i \text{ then } h_m(t+1) = a_1^i h(t) + b_1^i F_i(t) + k_i, \quad i = 1,\ldots,p \qquad (4.13)$$

where $h(t)$ and $F_i(t)$ are the level and inlet flowrate at time t, respectively; and it has been assumed that the input space can be partitioned using p fuzzy partitions. The model parameters $a_1^1,\ldots,a_1^p$, $b_1^1,\ldots,b_1^p$ and $k_1,\ldots,k_p$ can be represented by the matrix:

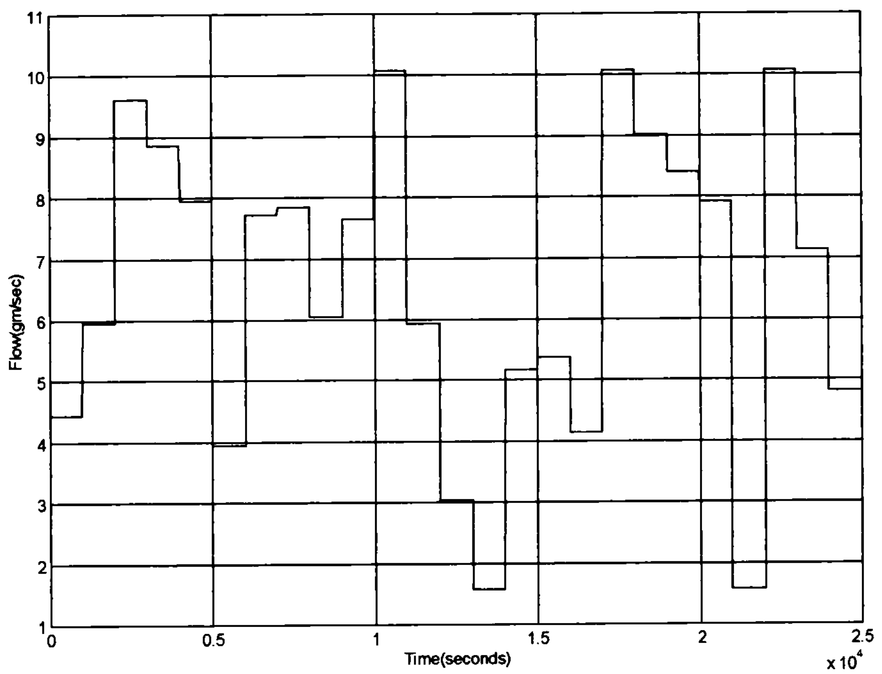
$$\Phi = \begin{bmatrix} a_1^1 & b_1^1 & k_1 \\ \vdots & \vdots & \vdots \\ a_1^p & b_1^p & k_p \end{bmatrix} \qquad (4.14)$$

Table 4.1 examines different methods of fuzzy partitioning the input space into 3 as shown in Figure 4.4. The constant term, k_i in equations (4.13) has been assumed to be zero. The model parameters were calculated using the input/output data used for plotting Figure 4.3. It can be observed that for this particular data set, the third partitioning method in Table 4.1 provides the smallest mean square error (MSE).

Tables 4.2 and 4.3 compare the modelling accuracy of the linear model with the fuzzy models derived using different numbers of fuzzy partitions as shown in Figure 4.5. Also indicated in Figure 4.5 are the important fuzzy partitioning points used by the models. It can be seen that, both with and without the constant term, the fuzzy models provide significantly better modelling accuracy than the linear model. The MSE generally decreases as the number of fuzzy partitions is increased. A point to note, however, is that it is considerably more difficult to experiment with different methods of fuzzy partitioning of the input space with 5-partition models than it is with 2 or 3-partition models.

Comparison of the MSE values in Tables 4.2 and 4.3 shows that there is quite a big improvement in the modelling accuracy when the constant term is included. The improvement is most significant in the case of the linear model and least significant in the case of the 5-partition fuzzy model. As a result, the difference between the lowest and the highest MSE values narrows considerably if the constant term is included.

Figures 4.6 and 4.7 show plots of the prediction error versus time for the linear and 5-partition fuzzy models in Table 4.3 based on the input/output data set used for identification. In both cases the error reaches a maximum immediately after the introduction of a step change. The generally smaller prediction errors with the 5-partition fuzzy model are clearly obvious.



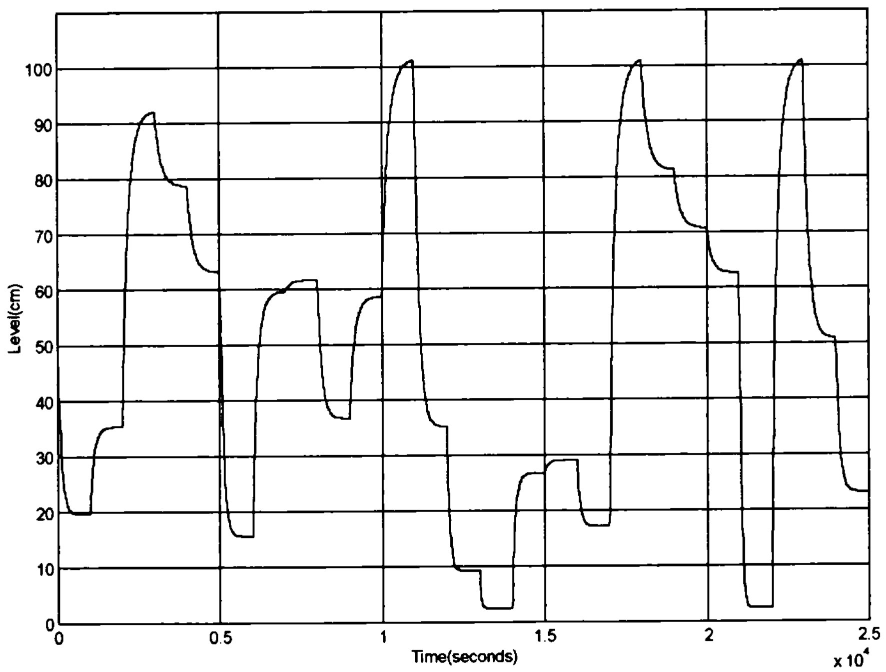


Figure 4.3: Input/output data set utilised in process model identification case study.

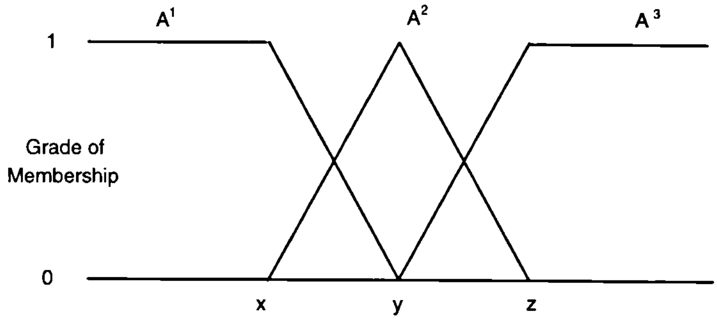


Figure 4.4: Fuzzy partitioning of the input space (i.e. liquid level) used for fuzzy process modelling in Table 4.1.

No.	Fuzzy partitioning of input space+			Model parameters		Mean square error
	x	y	z	a_1^i	b_1^i	
1	25	50	75	0.8870 0.8384 0.9290	0.5187 1.1591 0.6690	0.1292
2	10	50	90	0.8504 0.8509 0.9061	0.5242 1.0871 0.9206	0.0642
3	0	50	100	0.8394 0.8594 0.8944	0.4481 1.0323 1.0706	0.0587
4	0	50	110	0.8394 0.8598 0.9005	0.4489 1.0290 1.0847	0.0594
5	0	50	120	0.8394 0.8598 0.9073	0.4489 1.0290 1.0939	0.0594

+ refer to Figure 4.4

Table 4.1: Effect of fuzzy partitioning of the input space on model parameters and the modelling accuracy ($k_i = 0$).

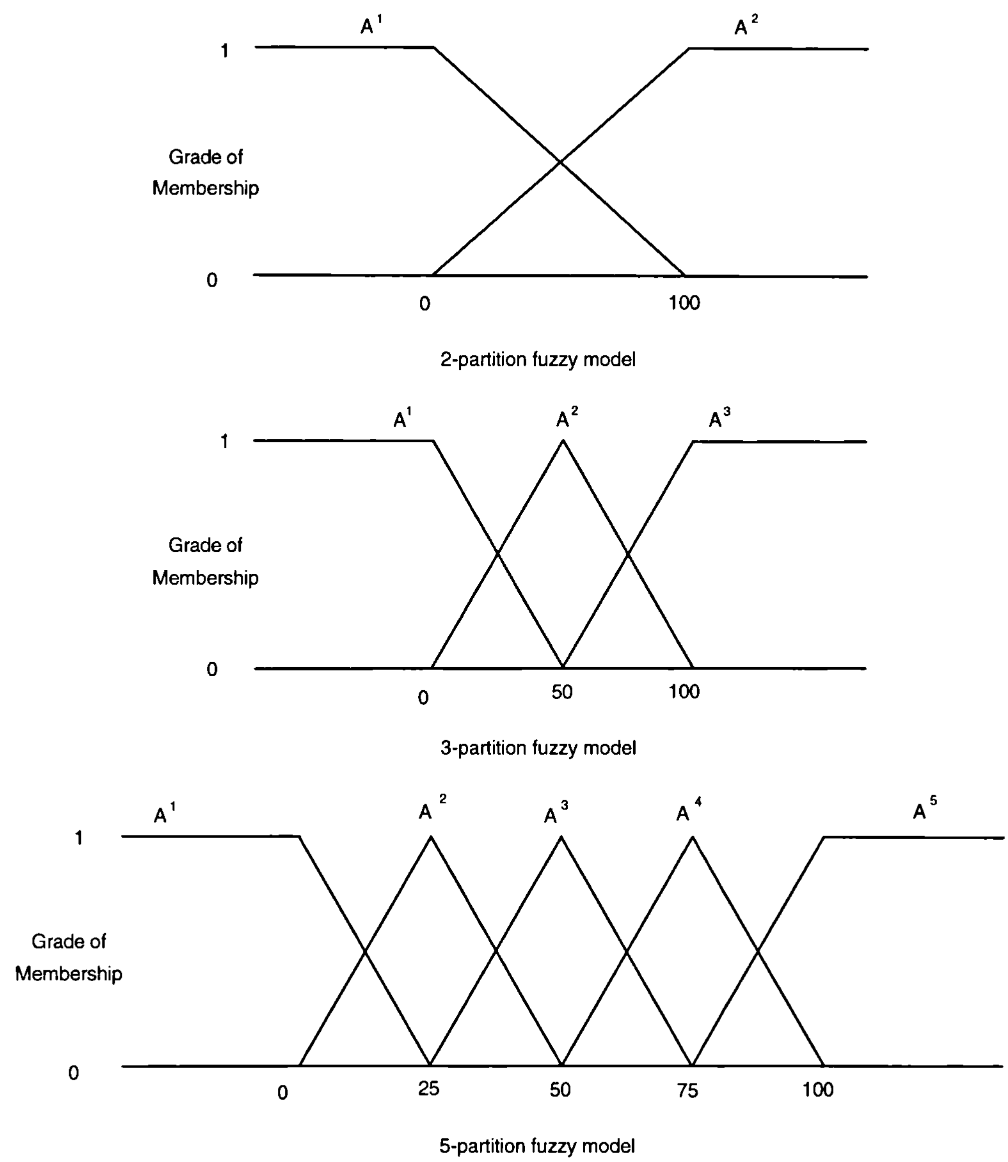


Figure 4.5: Fuzzy partitioning of the input space used for deriving the models in Tables 4.2 and 4.3.

No.	Type of model	Model parameters		MSE
		a_1^i	b_1^i	
1	linear model	0.9735	0.2194	0.4704
2	2-partition fuzzy model	0.8888 0.8554	0.3320 1.4549	0.0856
3	3-partition fuzzy model	0.8394 0.8594 0.8944	0.4481 1.0323 1.0706	0.0587
4	5-partition fuzzy model	0.7755 0.8082 0.8640 0.8900 0.9005	0.4415 0.9951 0.9790 0.9581 0.9953	0.0358

Table 4.2: Effect of the number of fuzzy partitions of the input space on the model parameters and the modelling accuracy ($k_i = 0$).

No.	Type of model	Model parameters			MSE
		a_1^i	b_1^i	k_i	
1	linear model	0.9437	0.7032	-1.8780	0.1824
2	2-partition fuzzy model	0.5772 0.5917	0.7029 1.1831	-1.0819 29.1866	0.0439
3	3-partition fuzzy model	0.7182 0.7649 0.7834	0.8293 0.9982 0.9593	-0.9896 4.7612 12.1092	0.0280
4	5-partition fuzzy model	0.7290 0.7902 0.8066 0.8146 0.8206	0.8729 0.9623 0.9637 0.9726 0.9757	-0.8549 0.4566 2.8614 5.4825 8.1966	0.0243

Table 4.3: Effect of the number of fuzzy partitions of the input space on the model parameters and the modelling accuracy ($k_i \neq 0$).

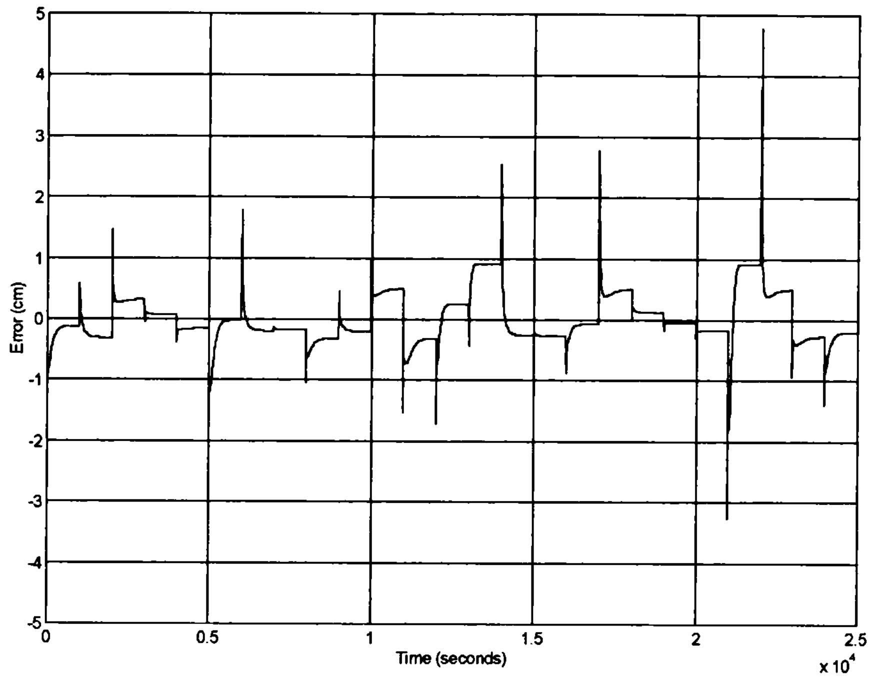


Figure 4.6: Prediction error versus time of the linear model ($k_i \neq 0$).

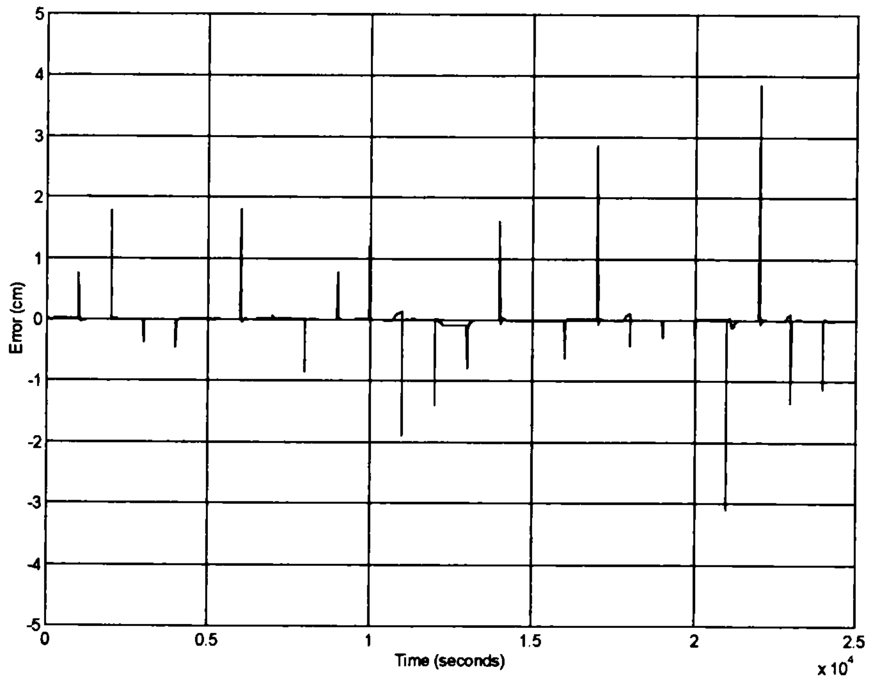


Figure 4.7: Prediction error versus time of the 5-partition fuzzy model ($k_i \neq 0$).

4.4 Continuous Stirred Tank Reactor System

4.4.1 Mathematical Model

A common type of reactor consists of a vessel into which the reactants flow and from which the products are taken. It is stirred to keep the contents as uniform as possible in composition and temperature. Often it is provided with an internal or external means of heat exchange. A schematic diagram of the continuous stirred tank reactor (CSTR) system is shown in Figure 4.8. A single irreversible exothermic reaction (A → B) is assumed to occur in the reactor. To remove the heat of reaction, a cooling jacket surrounds the reactor.

The process model, derived using mass and energy balances, consists of two nonlinear ordinary differential equations:

Mass balance:

$$\frac{dC_A}{dt} = \frac{F_i}{V}(C_{A_i} - C_A) - k_o e^{-E/RT} \cdot C_A \tag{4.15}$$

Energy balance:

$$\frac{dT}{dt} = \frac{F_i}{V}(T_i - T) + \frac{-\Delta H_R}{c_p \rho} \cdot k_o e^{-E/RT} \cdot C_A - \frac{Q}{c_p \rho V} \tag{4.16}$$

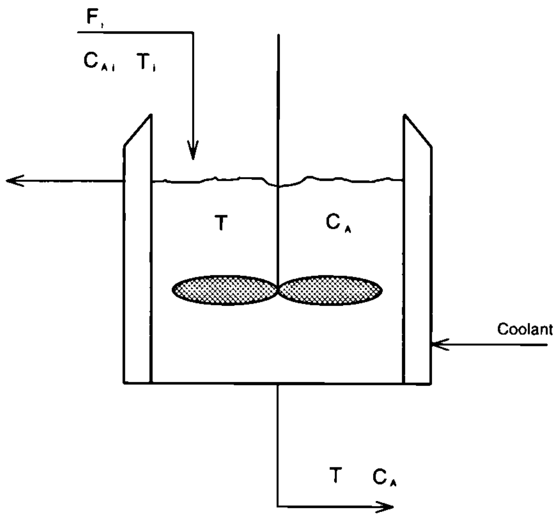


Figure 4.8: The continuous stirred tank reactor system.

where,

V = volume of reactor (1.36 m^3)
F_i = inlet flowrate (1.133 $m^3 \cdot h^{-1}$)
C_{A_i} = concentration of A in inlet stream (8008 moles $\cdot m^{-3}$)
C_A = concentration of A in reactor (393.3 moles $\cdot m^{-3}$)
c_p = specific heat (3140 $J \cdot kg^{-1} \cdot K^{-1}$)
E = activation energy (E/R = 8375 K)
R = gas constant
ΔH_R = reaction heat (-69,775 $J \cdot mole^{-1}$)
Q = heat removed from the CSTR (1.055×10^8 $J \cdot h^{-1}$)
T = reactor temperature (546.7 K)
T_i = temperature of inlet stream (373.3 K)
k_o = reaction rate constant (7.080×10^7 h^{-1})
ρ = density (800.8 $kg \cdot m^{-3}$)

The nominal operating values and constants for this example are given in brackets. A simulation model based on these equations was used in our study. The open loop response of the concentration of A in the reactor to changes in heat removal rate (Figure 4.9) shows that the relationship between these two variables is highly nonlinear.

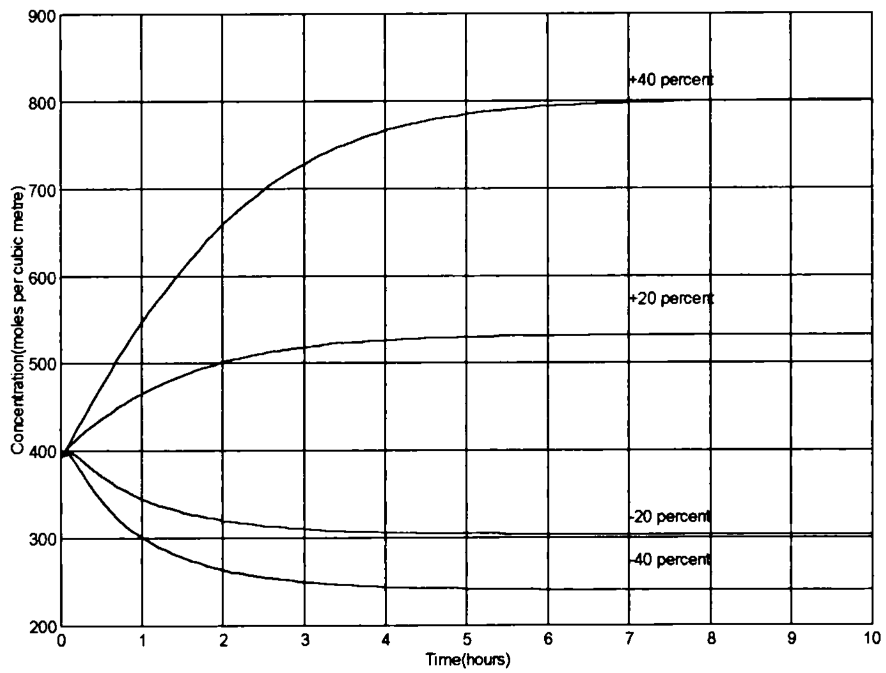


Figure 4.9: Open loop response of the concentration of A in the reactor to step changes in the rate of heat removal.

The process time constant varies from about 1.25 hours for a -25 percent step change in the rate of heat removal from normal steady-state operating conditions, to about 1.75 hours for a +25 percent step change. The process gain to a +25 percent step change is approximately 1.75 times that for a -25 percent step change.

4.4.2 Identification of Fuzzy Process Model

In this section, we examine the application of the fuzzy modelling method presented in Section 4.2 using a simulation of the CSTR system in Section 4.4.1. Data for modelling was generated by applying 50 random step changes, each lasting 6 hours, in the rate of heat removal such that the concentration of A in the reactor remained approximately within 250 and 650 moles per cubic metre (Figure 4.10). Sampling was carried out at 0.1 hour intervals. A total of 3000 data points were used for identification.

The following second order model structure has been assumed:

$$L^i: \ IF\ C_A(t)\ is\ B^i\ THEN\ C_{Am}(t+1) = a_1^i C_A(t) + a_2^i C_A(t-1) + b_1^i Q(t) + b_2^i Q(t-1) + k_i$$

$$\text{for } i = 1,\ldots,p \qquad (4.17)$$

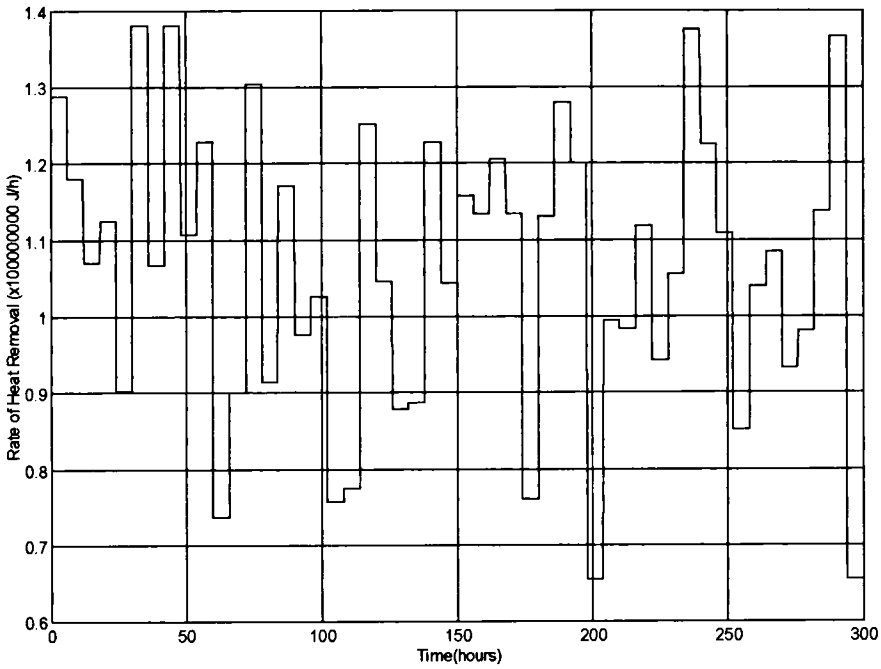
where $C_A(t)$ and $Q(t)$ are the concentration of A in the reactor and the rate of heat removal at time t, respectively.

Tables 4.4 and 4.5 compare the modelling accuracy of the linear model with the fuzzy models derived using different numbers of fuzzy partitions as shown in Figure 4.11. Also indicated in Figure 4.11 are the important fuzzy partitioning points used by the models. It can be observed that, both with and without the constant term, the fuzzy models have a significantly better modelling accuracy than the linear model. As in the case of the level system, the MSE generally decreases as the number of fuzzy partitions is increased.

Comparison of the MSE values in Tables 4.4 and 4.5 shows that unlike the level system, the improvement achieved by including the constant term is quite marginal. The most significant improvement seems to be with the 2-partition fuzzy model. The difference in the MSE values in the case of the linear model and the 5-partition fuzzy model is quite small. Table 4.6 shows the model parameters derived using a second input/output data set. In this case only the 2-partition and 5-partition fuzzy model parameters were derived. The constant term seems to have a similar effect on the MSE values as observed earlier.

Comparison of the model parameters in Tables 4.4, 4.5 and 4.6 will reveal that the model parameters derived without the constant term tend to be more consistent than the model parameters derived with the constant term. This is particularly evident in the case of the values of k_i. It will be observed that the values of k_i for the 5-partition fuzzy model computed using the second data set are about two times the values computed using the first data set.

Figures 4.12 to 4.15 show plots of the prediction error over the first 150 minutes for the models in Table 4.4 based on the first input/output data set used for identification. Very little steady-state error is observed in the case of the 5-partition fuzzy model.



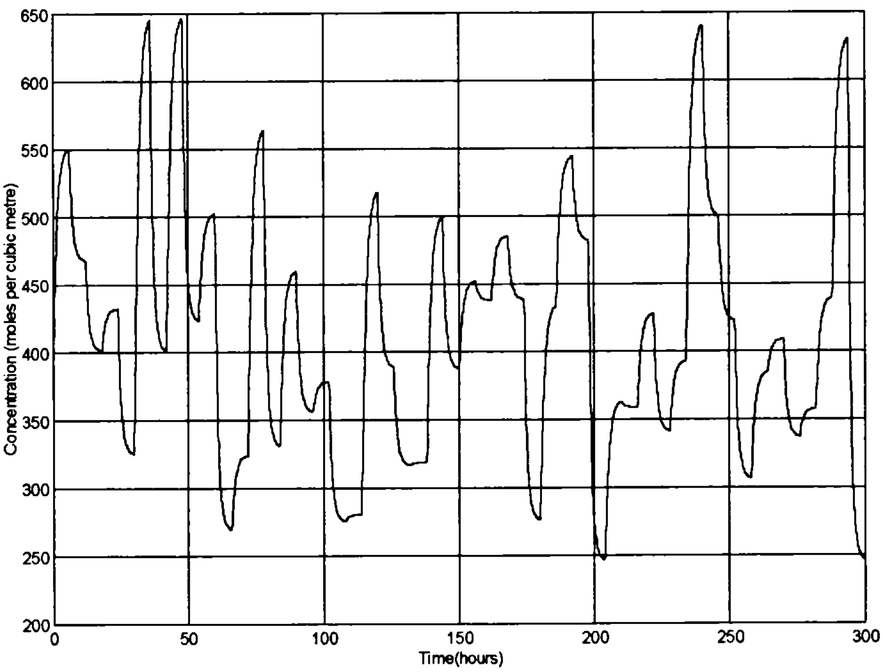


Figure 4.10: First input/output data set utilised in process model identification case study.

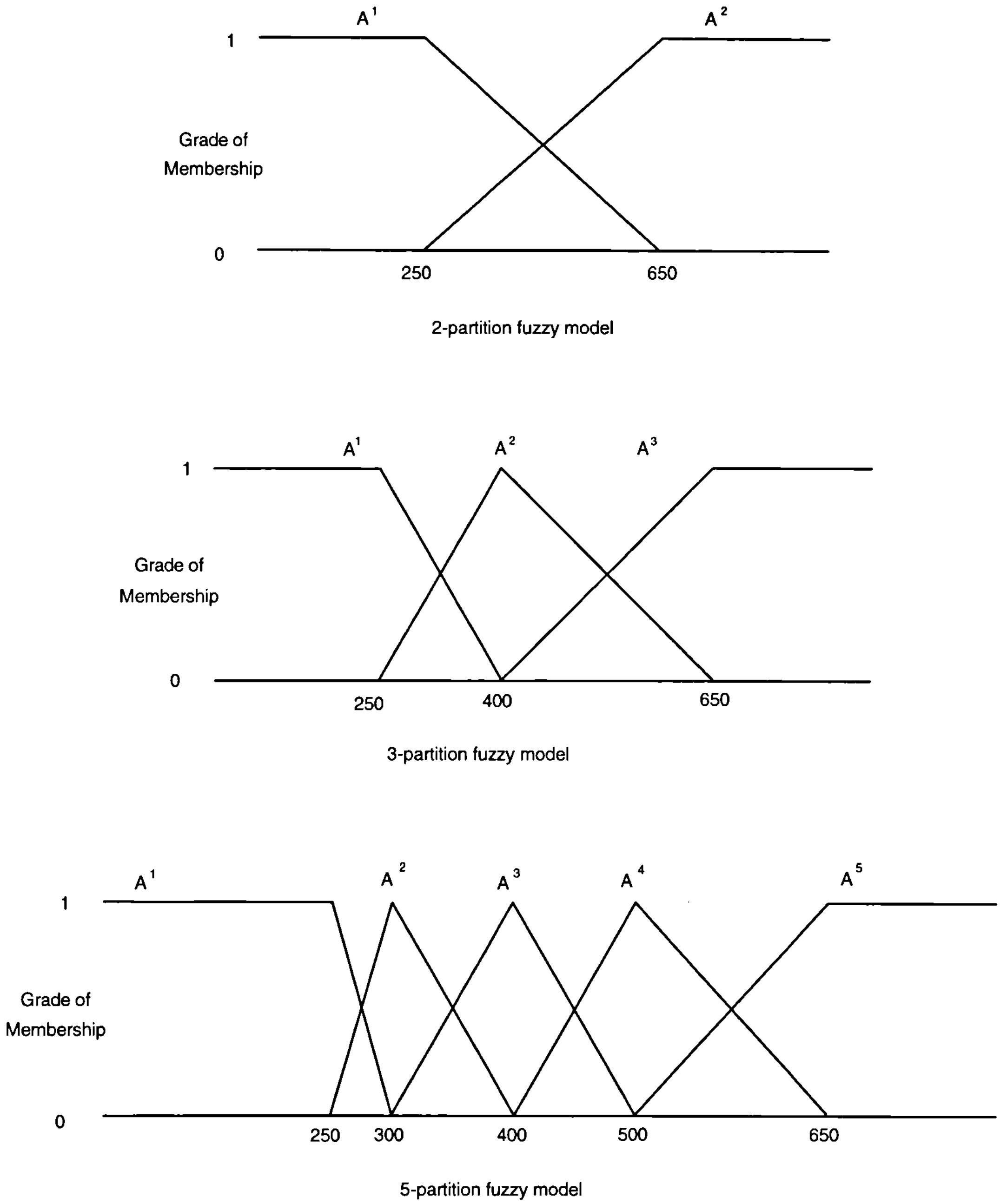


Figure 4.11: Fuzzy partitioning of the input space used for deriving the models in Tables 4.4 to 4.6.

No.	Type of model	Model parameters				Mean square error
		a_1^i	a_2^i	b_1^i	b_2^i	
1	Linear model	1.9076	-0.9137	17.3868	-15.0140	0.4650
2	2-partition fuzzy model	1.1238 1.9168	-0.1873 -0.9503	12.8528 30.0692	9.3424 -14.4170	0.2101
3	3-partition fuzzy model	1.0391 1.2200 1.5937	-0.0954 -0.2984 -0.6667	11.0606 22.5693 31.9760	8.9262 6.5603 2.1853	0.1268
4	5-partition fuzzy model	0.5432 0.8342 1.1712 1.3140 1.4841	0.3749 0.0790 -0.2544 -0.3988 -0.5688	13.9487 17.5735 22.6554 27.2690 32.7530	16.5850 13.3383 8.4176 7.2574 6.9739	0.0904

Table 4.4: Effect of the number of fuzzy partitions of the input space on model parameters and modelling accuracy using first input/output data set ($k_i = 0$).

No.	Type of model	Model parameters					Mean square error
		a_1^i	a_2^i	b_1^i	b_2^i	k_i	
1	Linear model	1.8560	-0.8655	18.0262	-13.2807	-1.1164	0.4492
2	2-partition fuzzy model	0.9782 1.4350	-0.0952 -0.5485	13.0185 34.5816	8.8077 8.6725	14.6469 13.9508	0.0936
3	3-partition fuzzy model	0.6345 1.0868 1.4081	0.1827 -0.2540 -0.5681	13.2058 23.1351 33.0421	14.7918 8.2385 7.8044	26.8443 33.3528 47.5343	0.0877
4	5-partition fuzzy model	0.3250 0.8061 1.0536 1.2073 1.4054	0.4698 0.0119 -0.2285 -0.3788 -0.5733	14.8415 16.9492 22.8091 27.5128 32.7161	18.2535 11.7663 9.3166 8.3492 6.7316	29.0139 30.6802 35.6822 41.7604 54.5354	0.0860

Table 4.5: Effect of the number of fuzzy partitions of the input space on model parameters and modelling accuracy using first input/output data set ($k_i \neq 0$).

No.	Type of model	Model parameters					Mean square error
		a_1^i	a_2^i	b_1^i	b_2^i	k_i	
1	2-partition fuzzy model (no constant term)	1.0989 1.9457	-0.1632 -0.9785	11.9613 31.0281	10.5585 -15.6572	- -	0.2442
2	2-partition fuzzy model (with constant term)	0.7971 1.3234	-0.0290 -0.5498	12.0925 36.2721	11.3078 8.6188	42.2844 85.1453	0.1090
3	5-partition fuzzy model (no constant term)	0.5271 0.8567 1.1797 1.3053 1.5044	0.3903 0.0588 -0.2621 -0.3913 -0.5861	14.1344 17.4147 22.3136 27.4489 32.4450	16.7941 12.7647 8.4092 7.5776 5.9825	- - - - -	0.1049
4	5-partition fuzzy model (with constant term)	0.2048 0.6880 0.9255 1.0961 1.2694	0.4711 0.0062 -0.2252 -0.3916 -0.5602	15.3276 16.6038 22.8239 27.4325 32.7642	17.3939 11.8584 9.4720 7.7317 7.2671	58.9106 68.0183 85.3830 104.6035 133.6841	0.0962

Table 4.6: Effect of fuzzy partitioning of the input space on model parameters and modelling accuracy using second input/output data set.

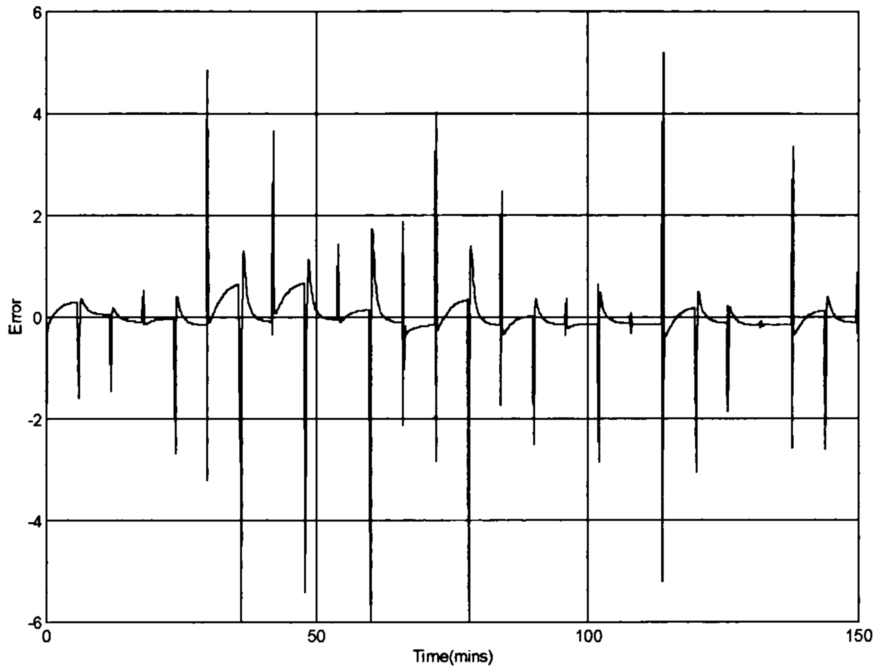


Figure 4.12: Prediction error versus time of the linear model ($k_i = 0$).

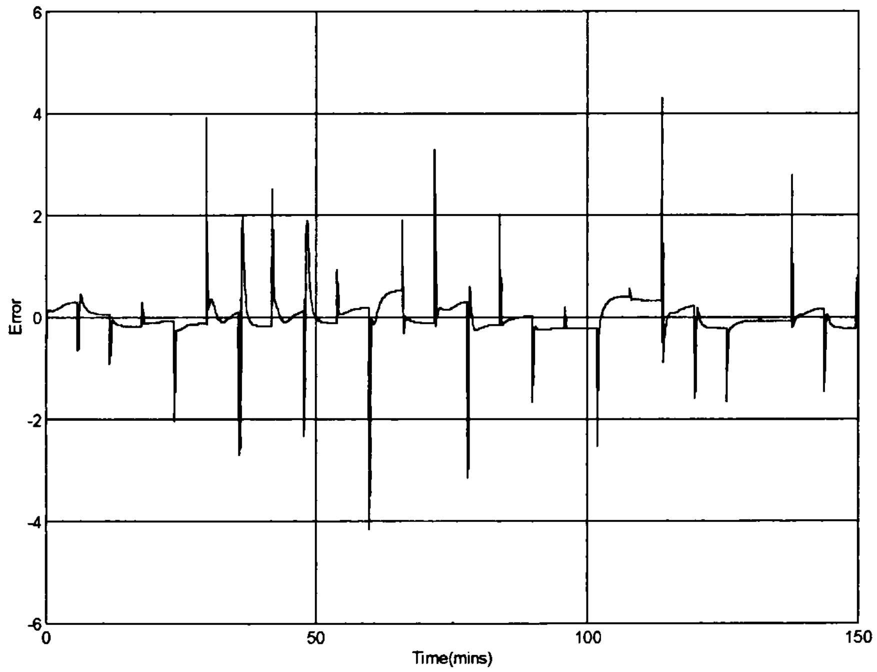


Figure 4.13: Prediction error versus time of the 2-partition fuzzy model ($k_i = 0$).

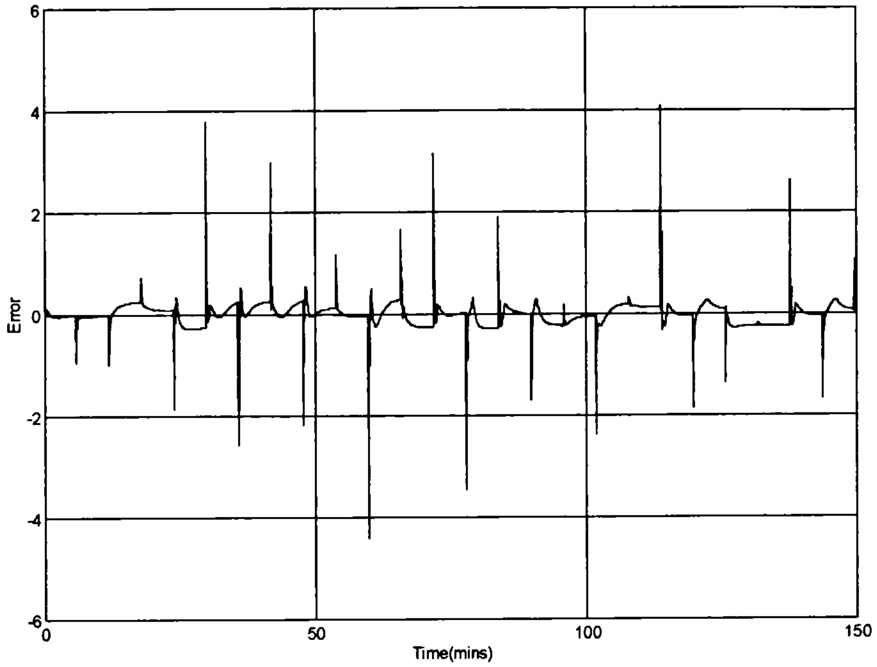


Figure 4.14: Prediction error versus time of the 3-partition fuzzy model ($k_i = 0$).

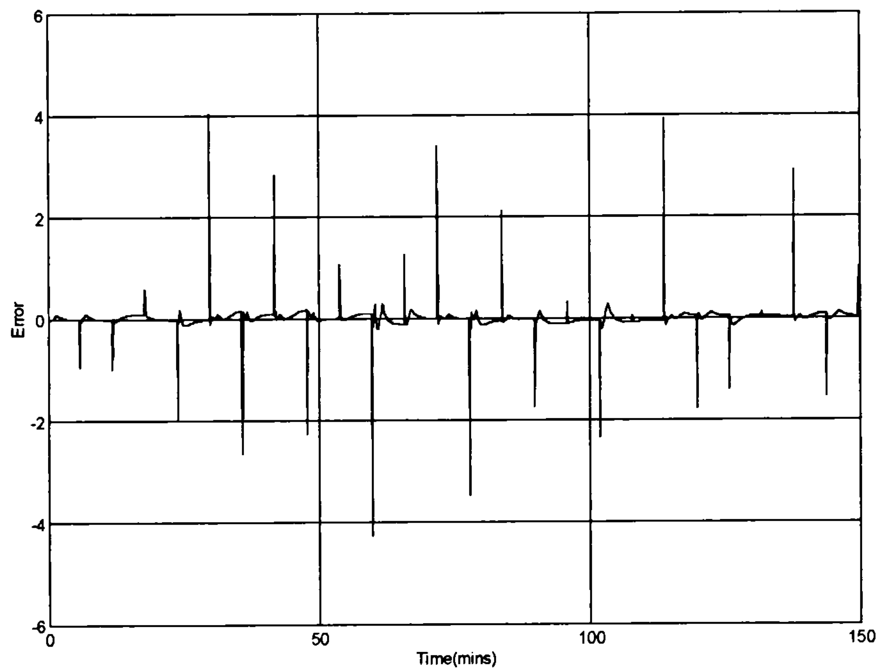


Figure 4.15: Prediction error versus time of the 5-partition fuzzy model ($k_i = 0$).

4.5 Conclusions

In both the examples examined in this chapter, it has been shown that the fuzzy process model provides better modelling accuracy than a single linear process model for representing the dynamics of nonlinear processes. Fuzzy models with up to 5 partitions were examined in our studies. As a general rule, modelling accuracy improves with the number of fuzzy partitions used, but the results seem to show that there is little benefit to be gained by using too many partitions.

The inclusion of the constant term has been shown to lead to better modelling accuracy in the case of the level system. Even though a reduction in MSE values was also noticed in the case of the CSTR system, the improvement in modelling accuracy is questionable since more consistent model parameters were observed if the constant term was not included.

In the next few chapters, we will attempt to design model based controllers which capitalise on the better modelling accuracy of the fuzzy process models investigated in this chapter to also provide better controller performance.

Chapter 5

ONE-STEP AHEAD PREDICTIVE CONTROLLER

5.1 Introduction

This chapter emphasizes the development of the conceptual framework for a fuzzy model based predictive control strategy based on the piecewise linear fuzzy modelling approach proposed by Takagi and Sugeno. To minimise the development effort, the prediction horizon used by the controller is limited to one-step only. The control strategy is based on determining the optimal controller output using an analytical approach. This approach leads to a fuzzy control algorithm which is much more computationally efficient than the numerically based control algorithms proposed by other researchers. The performance of the proposed fuzzy control algorithm is examined using simulations of the level of liquid in a tank and the continuous stirred tank reactor (CSTR) system described in Chapter 4. An attempt is also made to compare the performance with some other fuzzy model-based controllers proposed in the literature as well as the more traditional PID controller.

Chapters 6 and 7 attempt to generalize the controller concept developed in this chapter to multi-step prediction and control horizons.

5.2 Transformation of Nonlinear Fuzzy Process Model into Linear Model

It was shown in Section 4.2 that the overall model output of the single-input single-output system can be expressed in the following form:

$$y_m(t+1) = \beta\Phi X(t) \tag{5.1}$$

where,

$$X(t) = \left[y(t)\cdots y(t-j+1) \quad u(t)\cdots u(t-l+1) \quad 1\right]^T \tag{5.2}$$

$$\Phi = \begin{bmatrix} a_1^1\cdots a_j^1 & b_1^1\cdots b_l^1 & k_1 \\ \vdots & \vdots & \vdots \\ a_1^p\cdots a_j^p & b_1^p\cdots b_l^p & k_p \end{bmatrix} \tag{5.3}$$

$$\beta = \left[\beta_1\cdots\beta_i\cdots\beta_p\right] \tag{5.4}$$

and

$$\beta_i = \frac{B^i[y(t)]}{\sum_{i=1}^{p} B^i[y(t)]} \tag{5.5}$$

The product of β and Φ provides the weighted model parameters at each sampling instant:

$$\Phi' = \left[a_1'\cdots a_j' \quad b_1'\cdots b_l' \quad k'\right] \tag{5.6}$$

$$= \beta\,\Phi \tag{5.7}$$

and the overall model output can be expressed in terms of the weighted model parameters:

$$y_m(t+1) = \Phi' X(t) \tag{5.8}$$

Note that the 1-step ahead model predictions using either the weighted model parameters (equation (5.8)) or the complete fuzzy model (equation (5.1)) will be exactly identical. If the fuzzy model parameters, Φ, have been determined as illustrated in Chapter 4, then advantage can be taken of the ability to represent the 1-step ahead behaviour of the complete fuzzy model at each sampling instant using Φ', essentially transforming the nonlinear fuzzy model into a linear model, to facilitate the design of the controller as discussed in the next section. The linearisation method is discussed further in Section 5.3 after deriving the control algorithm.

5.3 Controller Formulation

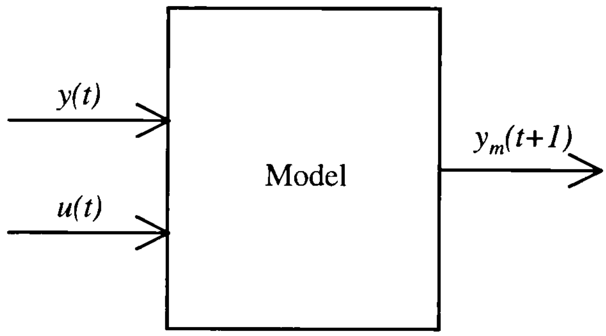


Figure 5.1: Block diagram of model predictions using a first-order model.

Consider 1-step ahead model predictions based on our fuzzy model representation. This is shown in block diagram form in Figure 5.1 in the case of a first-order system without loss of generality. The control strategy that we wish to implement here is to determine, at each sampling instant, the value of controller output, $u(t)$ which will minimize the variance between the predicted process output, $y_m(t+1)$ and the setpoint. This is called the minimum variance control strategy.

We will now attempt to mathematically formulate the above control strategy. The variance in the output of the system calculated using the weighted model parameters at each sampling instant is given by:

$$\begin{aligned} J &= \{[y_m(t+1) + err(t)] - y_{sp}\}^2 \\ &= \{[a_1'y(t)+\cdots+a_m'y(t-m+1) + b_1'u(t)+\cdots+b_n'u(t-n+1) + k' + err(t)] - y_{sp}\}^2 \end{aligned} \tag{5.9}$$

where y_{sp} is the setpoint and *err(t)* is an estimate of the modelling error.

A necessary condition for minimum J is:

$$\frac{dJ}{du} = 0 \tag{5.10}$$

Differentiating the expression for J and using the above condition leads to the following expression for the control signal which minimises the variance in the output of the system:

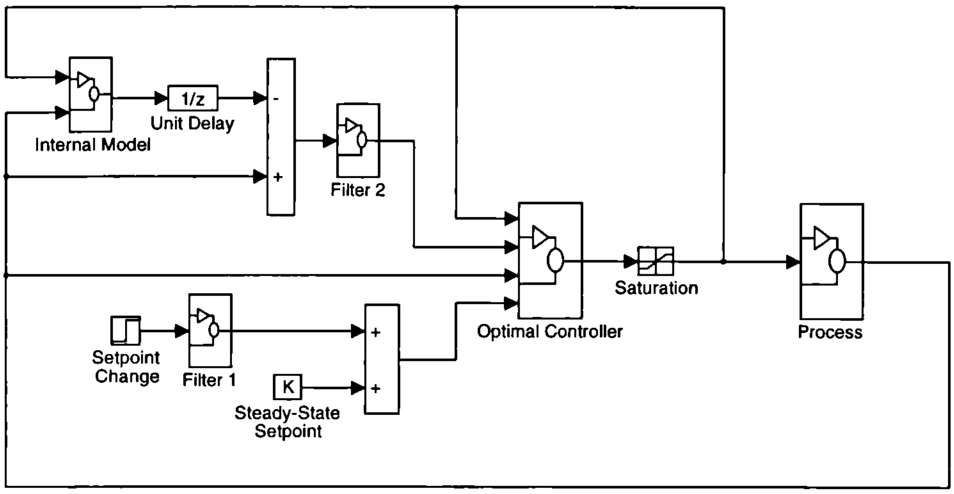


Figure 5.2: Proposed fuzzy model-based predictive controller structure.

$$u(t) = \frac{1}{b_1'}\left\{y_{sp} - \left[a_1'y(t) + \cdots a_m'y(t-m+1) + b_2'u(t-1) + \cdots + b_m'u(t-n+1) + k' + err(t)\right]\right\} \tag{5.11}$$

Note that in our fuzzy model representation, β is only dependent on the process output, $y(t)$ and is not affected by the value of $u(t)$ in Figure 5.1. The weighted model parameters, Φ', calculated using only the value of $y(t)$, can therefore be used for representing the nonlinear model using a linear model at every sampling instant. Linearisation in this manner makes it possible to determine the optimal controller output using an analytical approach. If β had been assumed to be a function of both $y(t)$ and $u(t)$, then it will not be possible to linearise as discussed above, and the only way to determine the 1-step ahead optimal controller output would be by using a numerical approach. It is necessary to make this point clear to highlight the fact that our proposed control strategy is only applicable to the special case of Takagi-Sugeno fuzzy modelling approach where input space partitioning is not dependent on the value of $u(t)$.

Figure 5.2 shows our implementation of the above control scheme. It includes a feedback mechanism to eliminate steady-state errors. The internal model is used to estimate the discrepancy between model and process outputs, $error(t)$ at each sampling instant:

$$error(t) = y(t) - y_m(t) \tag{5.12}$$

where $y_m(t)$ is the 1-step ahead model prediction at time (t-1). The estimate of the error is then filtered to produce $err(t)$ to minimise the instability introduced by modelling error feedback. The concept of a filter to stabilise the closed-loop system against plant-model mismatches follows that proposed by Garcia and Morari (1982) for Internal Model Control. Figure 5.2 also includes a second filter which serves as a reference trajectory for setpoint changes. The action of each filter is given by:

$$y_o(t) = (1 - K_f)y_o(t-1) + K_f y_i(t) \tag{5.13}$$

where $y_o(t)$ is the output from the filter and $y_i(t)$ is the input to the filter at time t, and K_f is the feedback filter gain. K_{f_1} and K_{f_2} are the feedback filter gains of Filters 1 and 2 respectively. Upper and lower bounds are imposed on the controller output to ensure that it does not stray beyond physically realizable values.

5.4 Application to Control of Liquid Level

5.4.1 Simulation Results

In this section, we examine the application of our proposed control strategy using a simulation of the liquid level system presented in Section 4.3.1. Identification of the fuzzy process model has been discussed in Section 4.3.2. The control problem investigated here is setpoint changes over two ranges: the first between 10 and 15 cm, and the second between 90 and 95 cm. The overall output from the controller was limited to the range 0 to 12 gm/sec.

The following first-order model structure was assumed for the liquid level system in Section 4.3.2:

$$L^i\text{: } if\, h(t)\ is\ A^i\ then\ h_m(t+1) = a_1^i h(t) + b_1^i F_i(t) + k_i \tag{5.14}$$

The control signal which minimises the output variance based on the above process model is given by:

$$F_i(t) = \frac{1}{b_1'}\left\{h_{sp} - \left[a_1' h(t) + k' + err(t)\right]\right\} \tag{5.15}$$

Table 5.1 shows the modelling accuracies of the process models used here based on the studies carried out in Chapter 4. Table 5.2 compares the performance of controllers based on the integral of absolute error (IAE) calculated over a time period of 1000 seconds starting from 1500 seconds from the start of a simulation run. The gain of Filter 1 (K_{f_1}) was standardised to 1. The value of the gain of Filter 2 (K_{f_2}) of 0.05 used in our studies corresponds to filter time constant of 190 seconds. As a general rule, lower MSE values in Table 5.1 corresponds with lower IAE values in Table 5.2. The best all-round performance seems to be achieved by the controller using the 5-partition fuzzy model with the constant term.

All subsequent studies carried out on the liquid level system in this thesis are based on models where the constant term is assumed to be non-zero.

Figures 5.3 to 5.10 provide graphical illustration of the performance of the proposed control system when using different process models. The generally better performance of controllers using fuzzy process models as compared to the controller using the linear process model is clearly evident.

Process model	MSE without k_i	MSE with k_i
linear	0.4704	0.1824
2-partition fuzzy	0.0856	0.0439
3-partition fuzzy	0.0587	0.0280
5-partition fuzzy	0.0358	0.0243

Table 5.1: Comparison of modelling accuracies with and without the constant term.

Process model	IAE between 10 and 15 cm.		IAE between 90 and 95 cm.	
	without k_i	with k_i	without k_i	with k_i
linear	2109.1	267.9	1633.5	218.6
2-partition fuzzy	606.4	203.5	264.9	217.7
3-partition fuzzy	218.1	173.2	219.2	198.4
5-partition fuzzy	194.7	189.1	192.2	191.8

Table 5.2: Comparison of 1-step ahead predictive controller performance with and without the constant term.

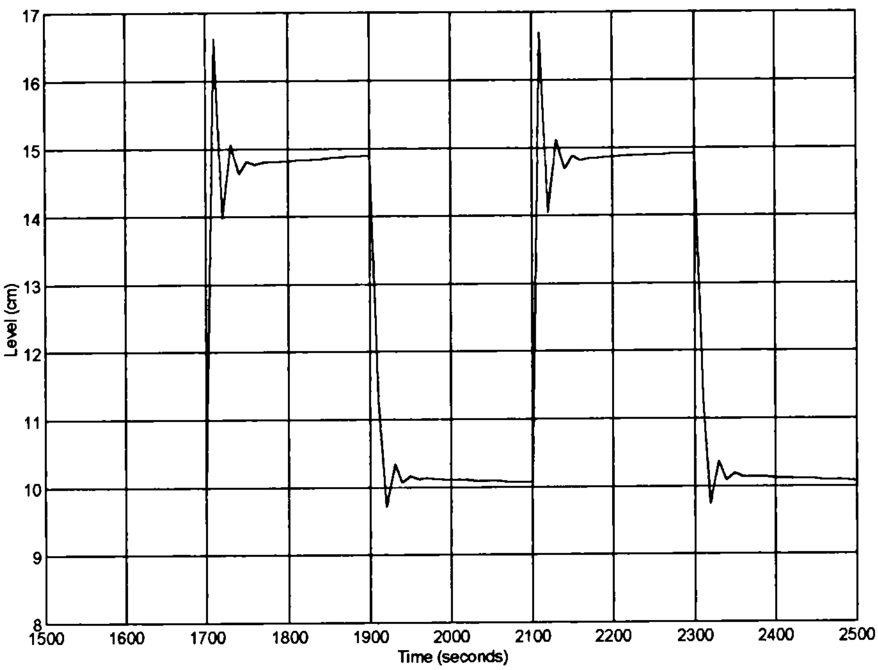


Figure 5.3: Process output response to setpoint changes between 10 and 15 cm. when using controller with linear model.

$(K_{f_1} = 1;\ K_{f_2} = 0.05;\ k_i \neq 0)$

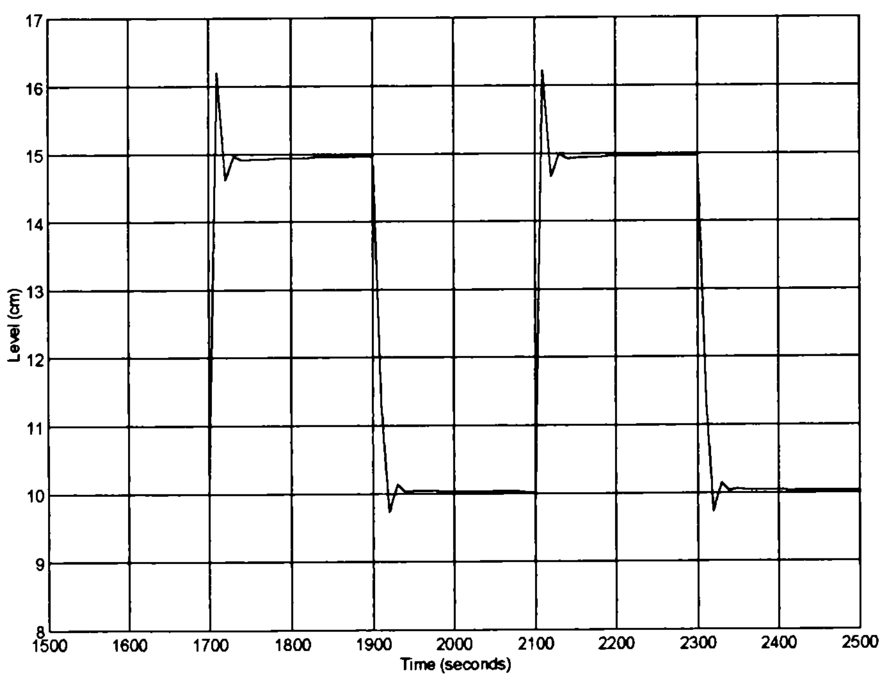


Figure 5.4: Process output response to setpoint changes between 10 and 15 cm. when using controller with 2-partition fuzzy model.

$(K_{f_1} = 1;\ K_{f_2} = 0.05; k_i \neq 0)$

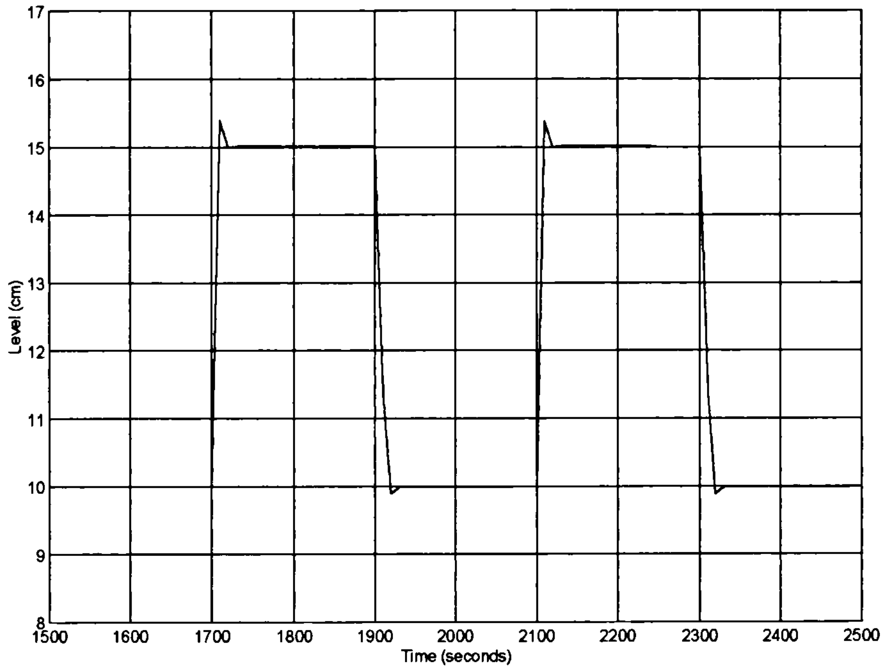


Figure 5.5: Process output response to setpoint changes between 10 and 15 cm. when using controller with 3-partition fuzzy model.
$(K_{f_1} = 1;\ K_{f_2} = 0.05; k_i \neq 0)$

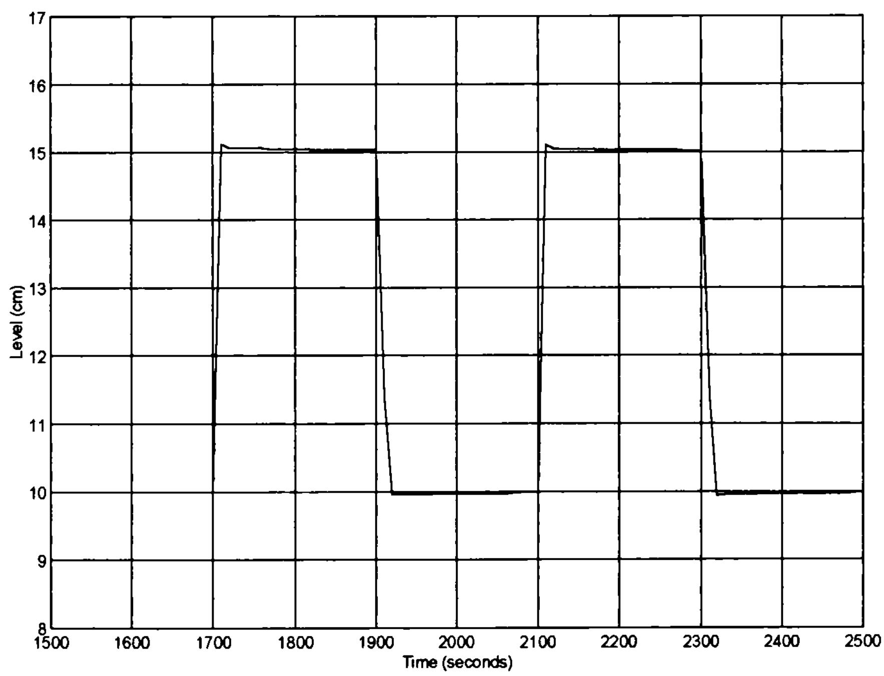


Figure 5.6: Process output response to setpoint changes between 10 and 15 cm. when using controller with 5-partition fuzzy model.
$(K_{f_1} = 1;\ K_{f_2} = 0.05; k_i \neq 0)$

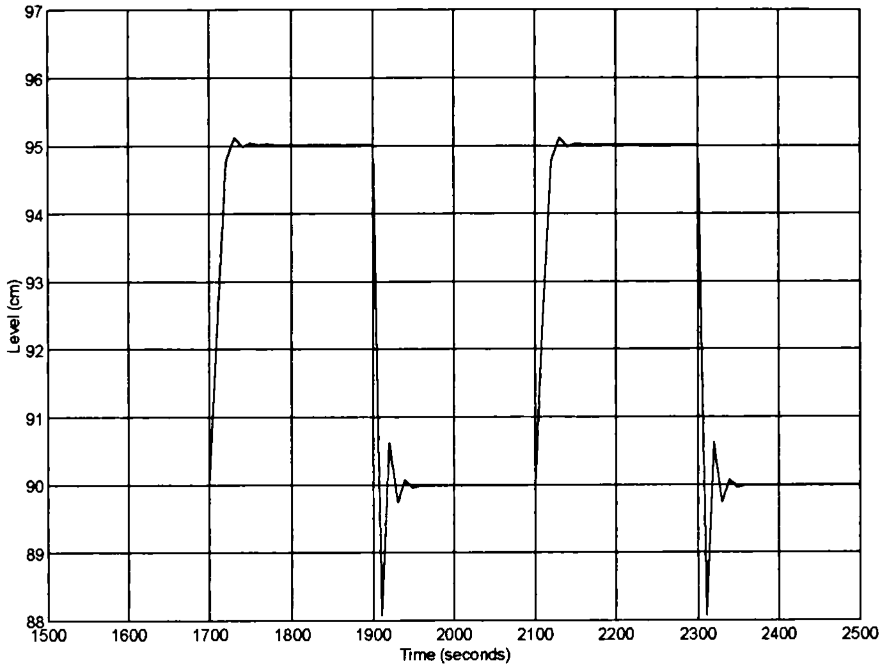


Figure 5.7: Process output response to setpoint changes between 90 and 95 cm. when using controller with linear model.
$(K_{f_1} = 1;\ K_{f_2} = 0.05; k_i \neq 0)$

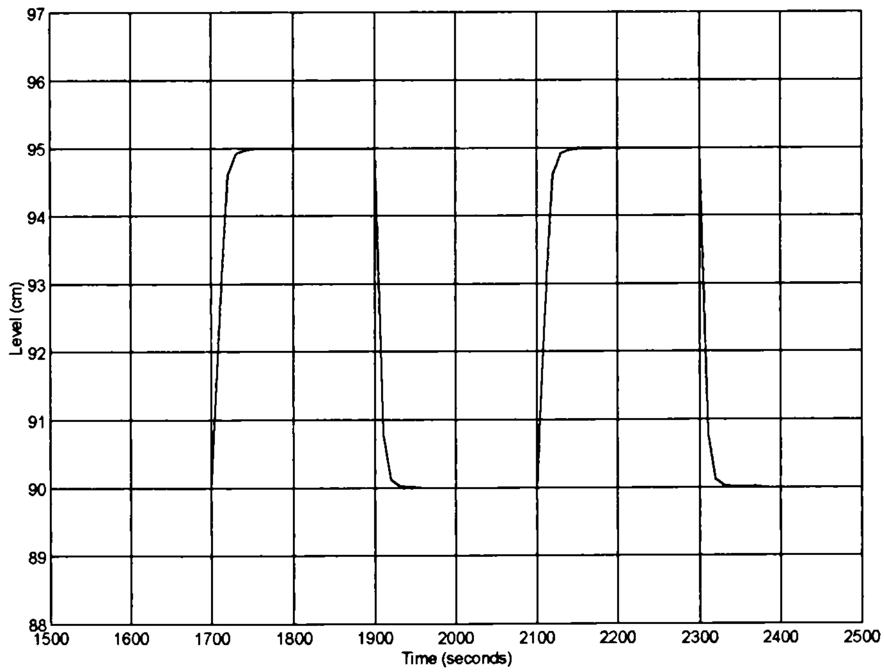


Figure 5.8: Process output response to setpoint changes between 90 and 95 cm. when using controller with 2-partition fuzzy model.
$(K_{f_1} = 1;\ K_{f_2} = 0.05; k_i \neq 0)$

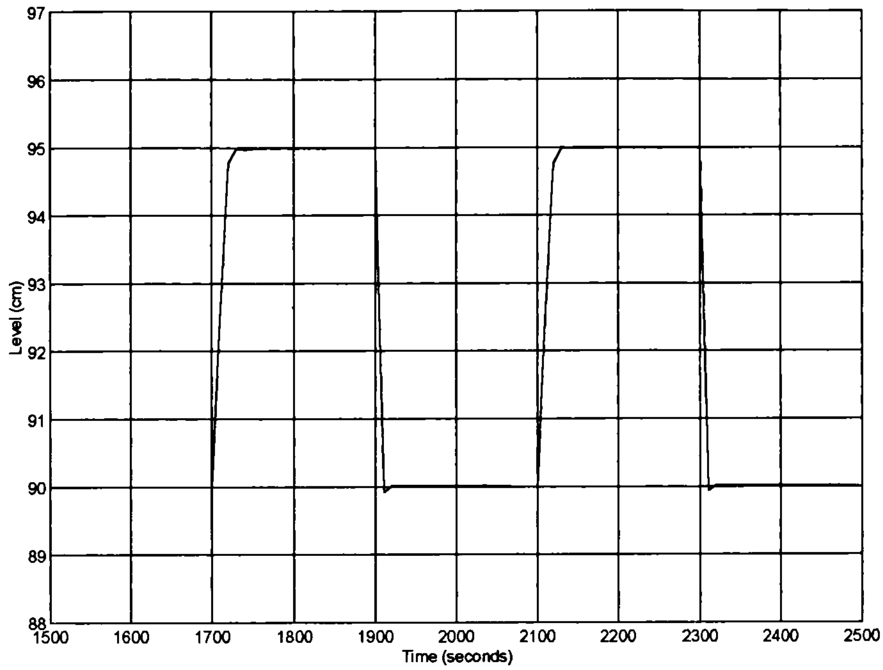


Figure 5.9: Process output response to setpoint changes between 90 and 95 cm. when using controller with 3-partition fuzzy model.
$(K_{f_1} = 1;\ K_{f_2} = 0.05; k_i \neq 0)$

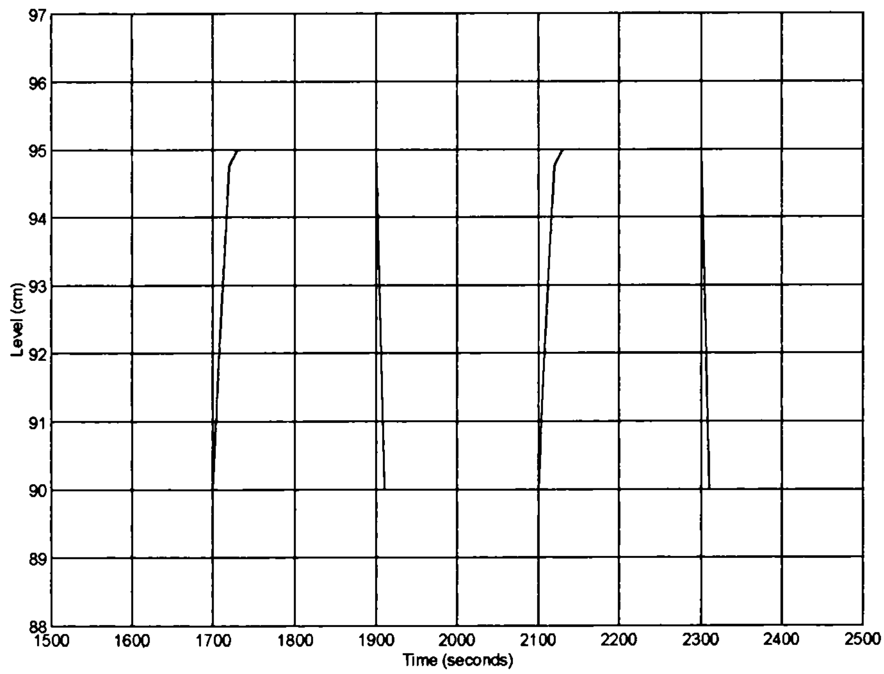


Figure 5.10: Process output response to setpoint changes between 90 and 95 cm. when using controller with 5-partition fuzzy model.
$(K_{f_1} = 1;\ K_{f_2} = 0.05; k_i \neq 0)$

5.4.2 Comparison of Fuzzy Model Based Controllers

Table 5.3 compares the optimal performance of our controller with other fuzzy model based controllers that have been proposed in the literature for the same level control application, i.e. a series of setpoint changes between 10 and 15 cm. and 90 and 95 cm. It will be observed that the best overall performance is obtained from our proposed controller. It should also be noted that the computational requirements of our proposed controller should be much less than fuzzy model based controllers using relational fuzzy models.

Table 5.3 also shows that the performance of our proposed controller is better than a PI controller.

Method	IAE (bottom of tank)	IAE (top of tank)
Graham & Newell (1989)	969	907
Postlethwaite (*with feedback*) (1991; 1994)	524	349
PI control (Postlethwaite, 1994)	426	300
1-step ahead predictive controller (*5-partition fuzzy model,* $K_{f_1}=1$, $K_{f_2}=0.05$, $k_i \neq 0$)	189.1	191.8

Table 5.3: Comparison of IAE values of various liquid level controllers.

5.5 Application to Control of CSTR

In this section, we examine the application of our proposed control strategy using a simulation of the CSTR system presented in Section 4.4.1. Identification of the fuzzy process model has been discussed in Section 4.4.2. The control problem investigated here is setpoint changes in the concentration of A in the reactor of 150 moles.m^{-3} on either side of normal steady-state operation and feed flowrate and feed concentration changes (i.e., load changes) of 20 percent and 5 percent, respectively. The output from the controller was limited to the range 0 to $2.0\times10^{8}\ \mathrm{J\cdot h^{-1}}$.

The following second-order model structure was assumed for the CSTR system in Section 4.4.2:

$$L^i\text{: } IF\ C_A(t)\ is\ B^i\ THEN\ C_{Am}(t+1) = a_1^i C_A(t) + a_2^i C_A(t-1) + b_1^i Q(t) + b_2^i Q(t-1) + k_i$$

$$\text{for } i = 1,\ldots,p \qquad (5.16)$$

The control signal which minimises the output variance based on the above process model is given by:

$$Q(t) = \frac{1}{b_1'}\left\{C_{Asp} - \left[a_1'C_A(t) + a_2'C_A(t-1) + b_2'Q(t-1) + k' + err(t)\right]\right\} \qquad (5.17)$$

Table 5.4 shows the modelling accuracies of the process models used here based on the studies carried out in Chapter 4. Table 5.5 compares the performance of controllers based on the integral of absolute error (IAE) calculated over a time period of 10 hours starting from 10 hours after the start of a simulation run. As a general rule, lower MSE values in Table 5.4 corresponds with lower IAE values in Table 5.5. An exception to this rule seems to be the 5-partition fuzzy model. It will be observed that the improvement in performance by including the constant term in the process model is generally not as significant as in the case of the level system, and deterioration is observed in the case of the 5-partition fuzzy model. Figures 5.11 to 5.14 allow visual comparison of the effect of the constant term on the controller's performance in the case of 3-partition and 5-partition fuzzy models. The best all-round performance is achieved by the controller using the 5-partition fuzzy model without the constant term.

Also evident from Table 5.5 is the generally better performance of controllers using fuzzy process models as compared to the controller using the linear process model.

All subsequent studies carried out on the CSTR system in this thesis are based on models where the constant term is assumed to be zero.

Figures 5.15 and 5.16 show that the output response from the controller using the 5-partition model is good over the whole range of K_{f_1} values even though it is slightly oscillatory at higher values. In Figures 5.17 and 5.18, a small steady-state error is noticed in the response to disturbance changes in the absence of modelling error feedback which can be eliminated at the expense of a slightly more oscillatory response by setting K_{f_2} to a low value of about 0.1. This corresponds to a filter time constant of 0.9 hours. The observations on the effect of K_{f_1} and K_{f_2} are typical of controllers based on IMC and MBPC.

Process model	MSE without k_i	MSE with k_i
linear	0.4650	0.4492
2-partition fuzzy	0.2101	0.0936
3-partition fuzzy	0.1268	0.0877
5-partition fuzzy	0.0904	0.0860

Table 5.4: Comparison of modelling accuracies with and without the constant term.

Process model	IAE to +150 moles.m^{-3} setpoint change		IAE to -150 moles.m^{-3} setpoint change	
	without k_i	with k_i	without k_i	with k_i
linear	293.99	284.53	195.19	182.15
2-partition fuzzy	241.94	43.52	166.86	161.46
3-partition fuzzy	53.43	43.93	167.42	61.34
5-partition fuzzy	44.55	44.02	46.43	252.45

Table 5.5: Comparison of 1-step ahead predictive controller performance with and without the constant term.

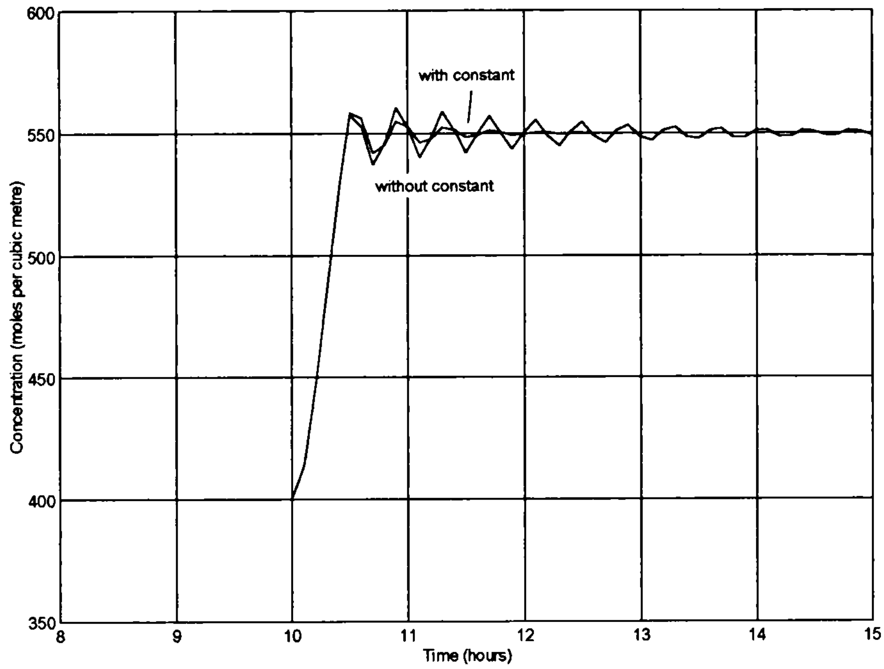


Figure 5.11: Process output response to +150 moles$\cdot$m^{-3} setpoint change when using proposed controller with 3-partition fuzzy process model.
$(K_{f_1} = 1; K_{f_2} = 0.1)$

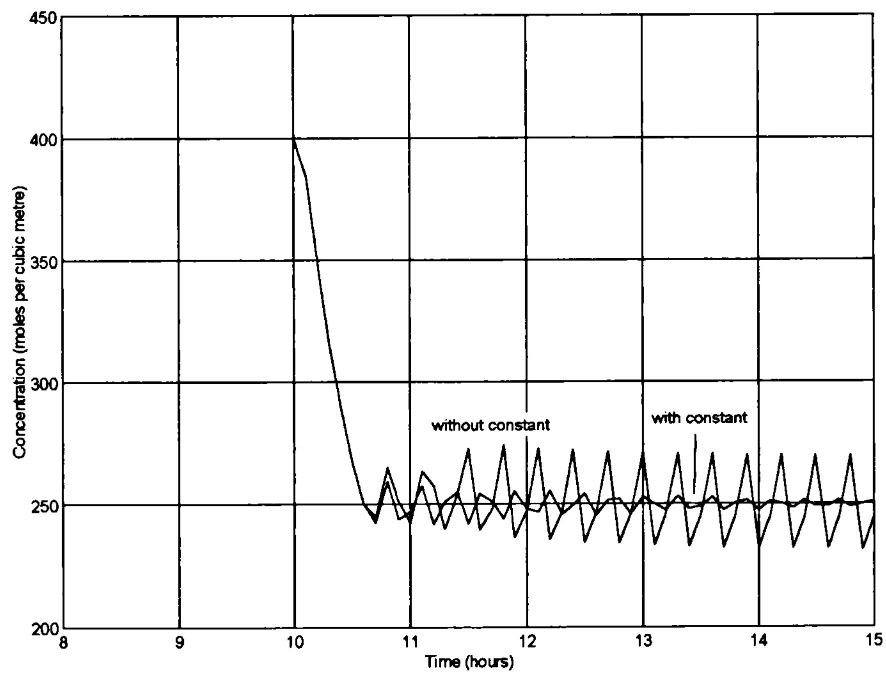


Figure 5.12: Process output response to -150 moles$\cdot$m^{-3} setpoint change when using proposed controller with 3-partition fuzzy process model.
$(K_{f_1} = 1; K_{f_2} = 0.1)$

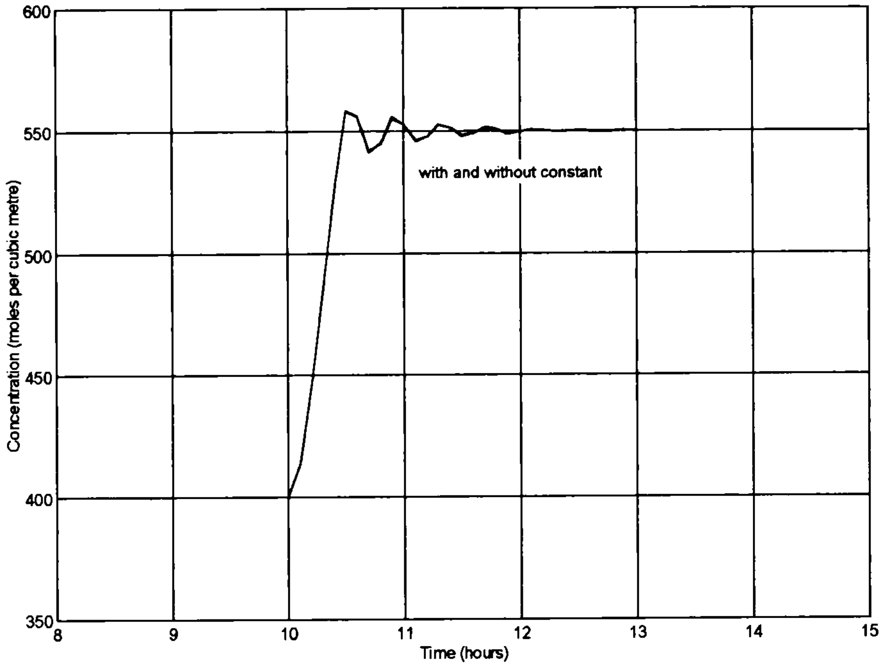


Figure 5.13: Process output response to +150 moles · m^{-3} setpoint change when using proposed controller with 5-partition fuzzy process model.
$(K_{f_1} = 1; K_{f_2} = 0.1)$

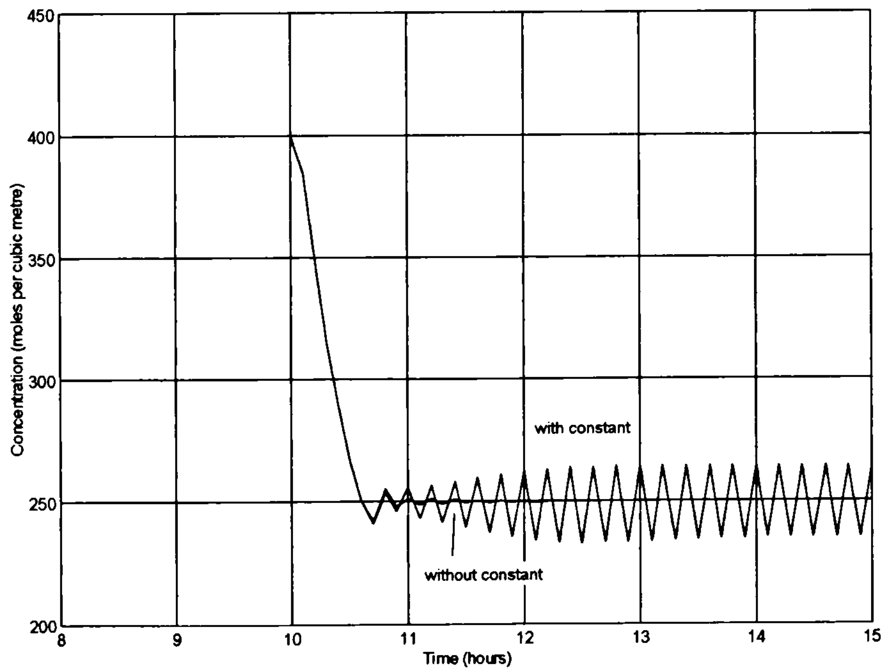


Figure 5.14: Process output response to -150 moles · m^{-3} setpoint change when using proposed controller with 5-partition fuzzy process model.
$(K_{f_1} = 1; K_{f_2} = 0.1)$

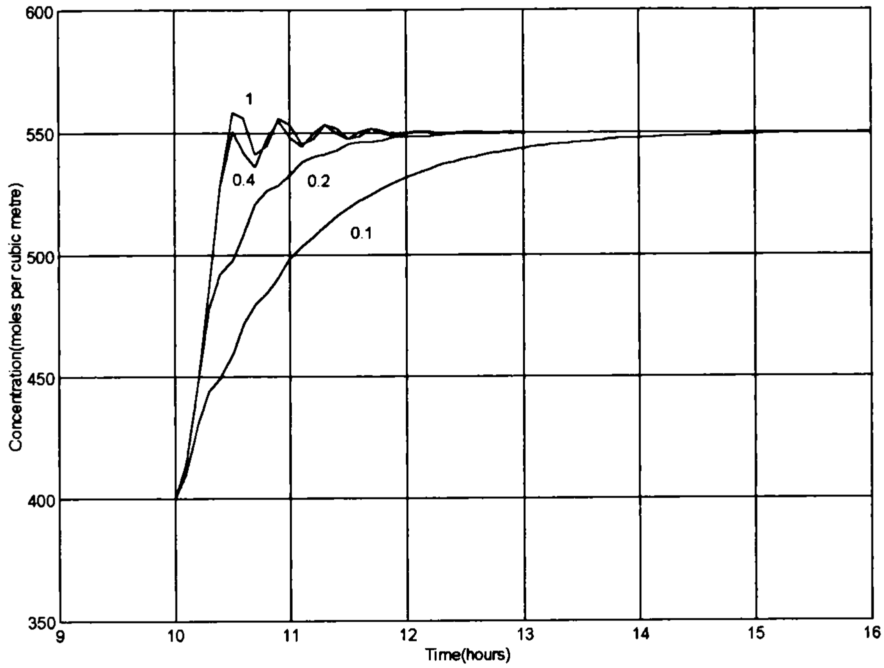


Figure 5.15: Effect of the gain of Filter 1 on process output response to +150 moles $\cdot$ m^{-3} setpoint change when using proposed controller with 5-partition fuzzy process model ($K_{f_2} = 0.1$; $k_i = 0$).

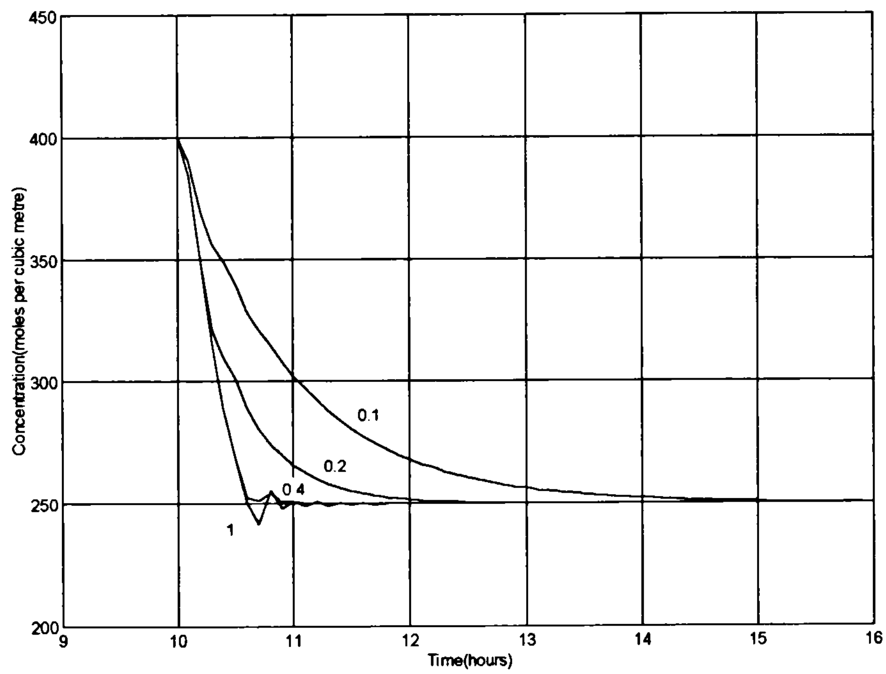


Figure 5.16: Effect of the gain of Filter 1 on process output response to -150 moles $\cdot$ m^{-3} setpoint change when using proposed controller with 5-partition fuzzy process model ($K_{f_2} = 0.1$; $k_i = 0$).

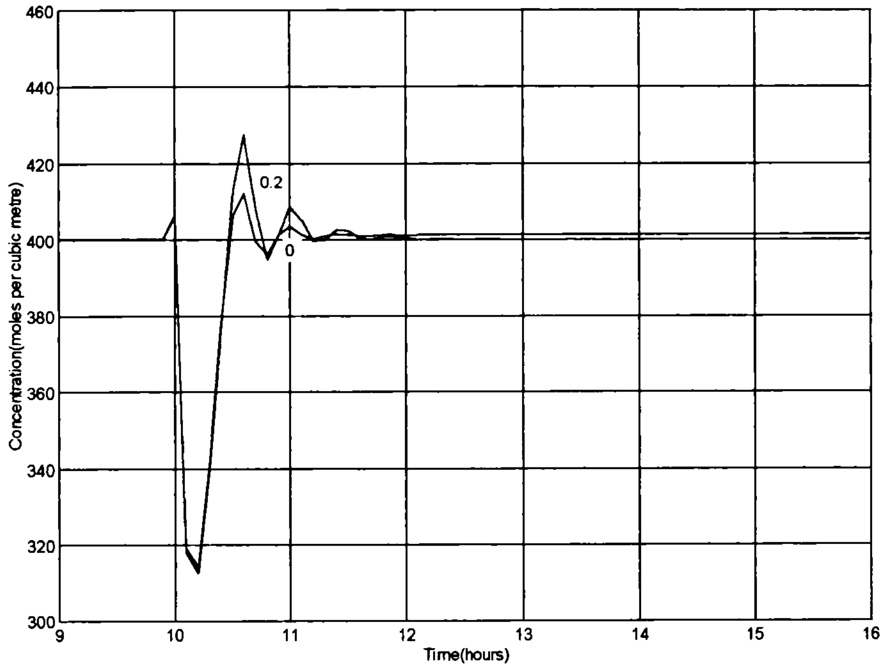


Figure 5.17: Effect of the gain of Filter 2 on process output response to -20 percent change in feed flowrate when using proposed controller with 5-partition fuzzy process model ($K_{f_1} = 1$; $k_i = 0$).

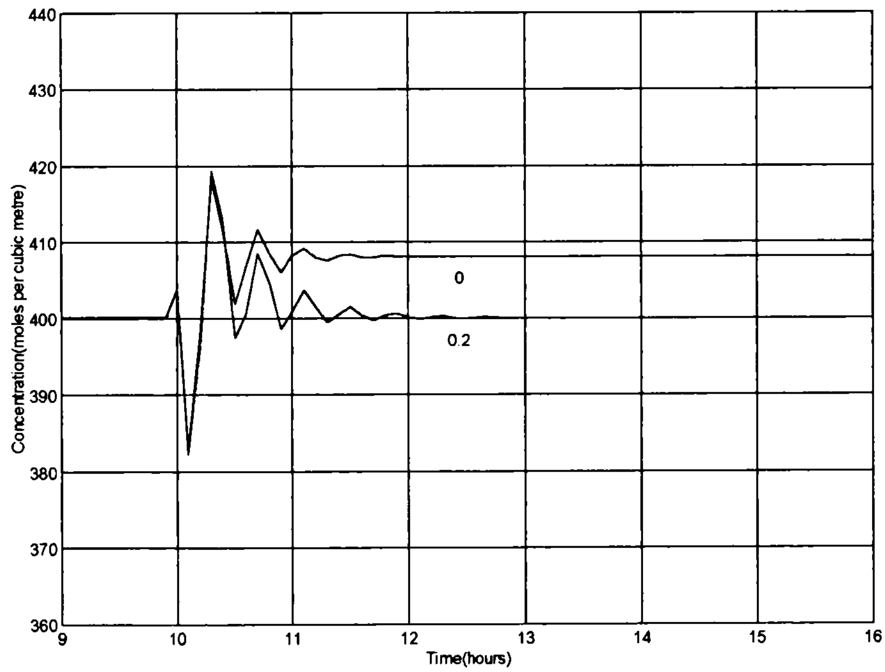


Figure 5.18: Effect of the gain of Filter 2 on process output response to -5 percent change in concentration of A in inlet stream when using proposed controller with 5-partition fuzzy process model ($K_{f_1} = 1$; $k_i = 0$).

5.6 Conclusions

Even though the Takagi-Sugeno fuzzy modelling approach can accommodate input space partitioning based on more than one variable, the two examples examined in Chapter 4 have shown that input space partitioning based on just the current value of the process output can provide significantly better modelling accuracy than a single linear model for representing the nonlinear system dynamics over a wide range. In this chapter, an analytical approach for determining the optimal controller output using the fuzzy modelling approach has also been shown to result in improved controller performance over a similar controller using a single linear process model as well as other fuzzy model-based controllers that have been proposed in literature. The question may be asked, if it is necessary to consider input space partitioning based on more than one variable (for example, the current values of process and controller outputs) to achieve better modelling accuracy and better controller performance. This has not been attempted here mainly because of the problems that will be encountered when attempting to design a control algorithm using an analytical approach for reasons pointed out in Section 5.3. Hence, in all of the examples studied in this thesis, input space partitioning has been based on just the current value of the process output.

The emphasis in this chapter has been on the one-step ahead predictive controller structure. It is a well-known fact that one-step ahead predictive controllers are susceptible to robustness problems. A solution to this problem is to use a multi-step prediction horizon. The issue of robustness is therefore discussed in Chapter 7 after examining two methods of extending the controller to multi-step prediction and control horizons.

Chapter 6

LONG-RANGE PREDICTIVE CONTROL: NUMERICAL APPROACH

6.1 Introduction

In the last chapter, the emphasis was on the development of the conceptual framework for a model based predictive control strategy based on the special case of the piecewise linear fuzzy modelling approach presented in Chapter 4. Even though the controller has been shown to work, it has not sufficiently addressed the issue of controller robustness. One obvious solution to this problem is to extend the prediction horizon used by the controller. This chapter will focus on a numerical approach for achieving this. The Fibonacci search optimisation method used here limits the control horizon to just 1-step. There is, however, no reason why the method cannot be extended to a multi-step control horizon, or permit explicit handling of constraints, if a more advanced optimisation technique is used. The main limitation is the considerable computation capability required.

In Chapter 7, we will present an analytical approach for extending the prediction and control horizons. The much lower computational requirements of the analytical approach provides it with a distinct advantage over the numerical approach. Nevertheless, in the case of high order systems and systems containing dead times, etc. the numerical approach may be easier to implement. The main aims of this chapter are to demonstrate the numerical approach and to allow comparisons to be made between the numerical and analytical approaches. These objectives are sufficiently met by implementing the control strategy using the Fibonacci search optimisation method to the tank liquid level control problem examined in Chapter 5. Some important issues involved in making multi-step predictions using the fuzzy model are examined in this chapter. These issues are important not only for the numerical approach but also for the analytical approach.

6.2 Making Multi-Step Predictions Using the Fuzzy Process Model

Figure 6.1 shows how predictions can be made in a step-by-step manner using a second-order model when a multi-step prediction horizon and a control horizon of 1-step are used. The predicted output from the first-step is given by:

$$y_m(t+1) = \beta \Phi X(t) \tag{6.1}$$

where,

$$X(t) = \begin{bmatrix} y(t) \cdots y(t-j+1) & u(t) \cdots u(t-l+1) & 1 \end{bmatrix}^T \tag{6.2}$$

$$\Phi = \begin{bmatrix} a_1^1 \cdots a_j^1 & b_1^1 \cdots b_l^1 & k_1 \\ \vdots & \vdots & \vdots \\ a_1^p \cdots a_j^p & b_1^p \cdots b_l^p & k_p \end{bmatrix} \tag{6.3}$$

$$\beta = \begin{bmatrix} \beta_1 \cdots \beta_i \cdots \beta_p \end{bmatrix} \tag{6.4}$$

and $$\beta_i = \frac{B^i[y(t)]}{\sum_{i=1}^{p} B^i[y(t)]} \tag{6.5}$$

Equation (6.1) can be generalised to enable multi-step predictions as follows:

$$y_m(t+i) = \beta \Phi X(t+i-1) \tag{6.6}$$

where $X(t+i-1)$ is the input vector to the ith prediction step. When attempting to use of this formula to make predictions in the second and subsequent steps, we are faced with a dilemna. The model we are using is a 1-step ahead prediction algorithm which uses of the current value of the process output to determine β. Adaptation is needed to allow this algorithm to be used for making multi-step predictions. We considered the following two options (refer to Figure 6.1):

1. Calculate β for each prediction step using the y_p value from the previous step.

2. Use the value of β calculated in the first step over the entire prediction horizon.

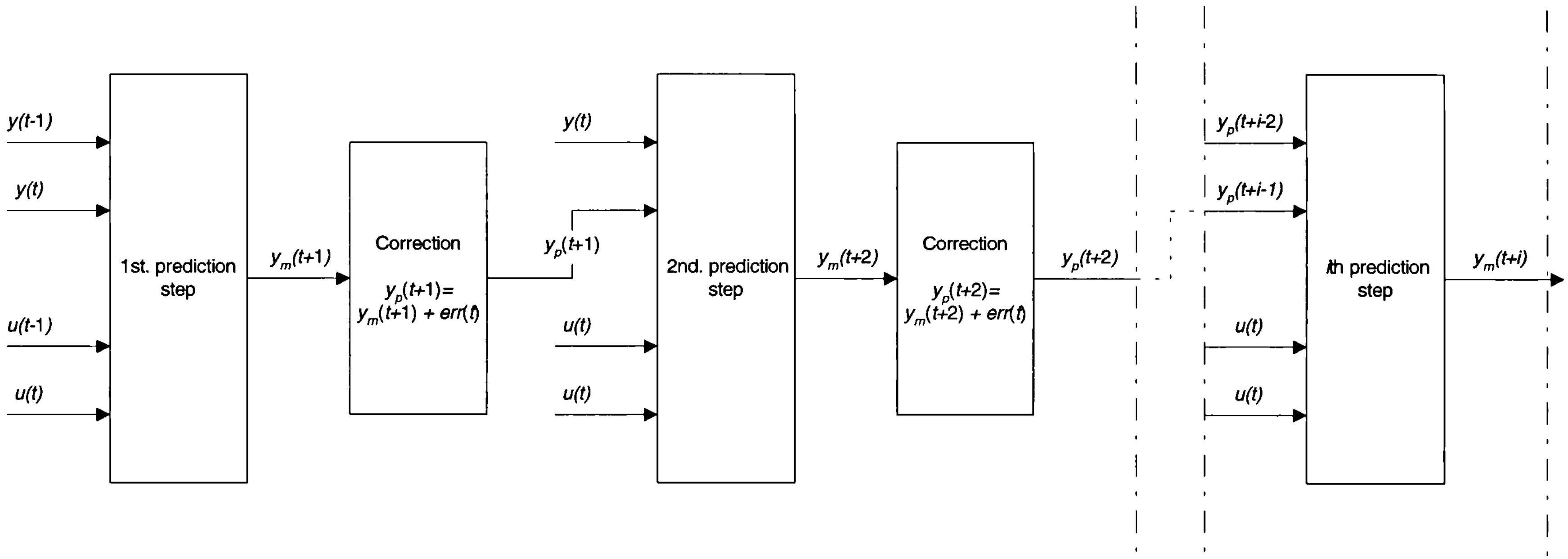


Figure 6.1: Flow diagram of the calculation of model predictions over entire prediction horizon.

The first option appears better, but unfortunately, the model predictions may not be sufficiently accurate to allow estimation of β in the second and subsequent prediction steps. If the y_p values are inaccurate, then the calculated β values will also be inaccurate. It should be noted that model predictions can be highly inaccurate immediately after the introduction of a setpoint change or a load change. Also, the more distant the prediction, the more susceptible it is to modelling inaccuracies. We believe that calculating β using inaccurate y_p values can have a magnifying effect on modelling errors.

The second option (i.e. setting β constant) reduces the amount of computations needed, since it is not necessary to calculate β separately for each prediction step. It also implies that a single linear model can be used for making predictions over the entire prediction horizon. This method of linearization is explored further in Chapter 7.

Experiments carried out using both approaches showed that the second approach leads to better controller peformance.

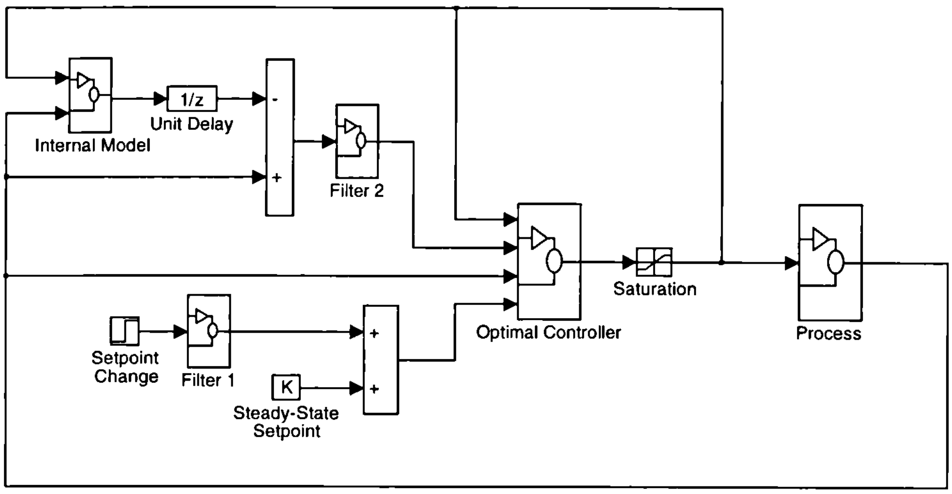


Figure 6.2: Proposed fuzzy model-based predictive controller structure.

6.3 Controller Formulation

Figure 6.2 shows the controller structure that is to be implemented here. Even though it is similar to the 1-step ahead predictive controller structure used in Chapter 5, it is reproduced here to make the description of the controller presented in this chapter complete. The main difference between the two controllers lies in the approach used to determine the optimal controller output. The objective here is to select a set of future control moves (control horizon) in order to minimize a function based on a desired output trajectory over a prediction horizon. A general mathematical formulation is:

$$\min_{u(t),\ldots,u(t+H_c-1)} J = \sum_{i=1}^{H_p} [y_p(t+i) - w(t+i)]^2 \tag{6.7}$$

subject to:

$$u(t+i) = u(t+H_c-1) \text{ for all } i > t+H_c-1 \tag{6.8}$$

$$y_{\min} \le y_p(t+i) \le y_{\max} \tag{6.9}$$

$$u_{\min} \le u(t+i) \le u_{\max} \tag{6.10}$$

$$|u(t+i) - u(t+i-1)| \le \Delta u_{\max} \tag{6.11}$$

where $u(t)$, $y(t)$ and $w(t)$ denote the controller output, process output and the setpoint at sampling instant t, respectively; H_c is the control horizon; H_p is the prediction horizon; and $y_p(t+i)$ denotes the ith-step ahead model prediction at time t. The objective function is the sum of squares of the residuals between the model predicted outputs and the setpoint values over the prediction horizon of H_p time steps. The optimisation decision variables are the control actions H_c time steps into the future; beyond H_c time steps it is assumed that the control action is constant.

The formulation also incorporates upper and lower bounds on the output and input (second and third set of constraints, respectively) and bounds on the allowed control action change between successive sampling intervals (fourth set of constraints). Other operating constraints, unique to the system being controlled, may also be included. Note that both the absolute and velocity constraints on the controller output are explicitly included in the above formulation. Velocity constraints can alternatively be accommodated by including them in the objective function formulation:

$$\min_{u(t),\ldots,u(t+H_c-1)} J = \sum_{i=1}^{H_p}[y_p(t+i) - w(t+i)]^2 + \sum_{i=1}^{H_c}\lambda[u(t+i) - u(t+i-1)]^2 \qquad (6.12)$$

where λ is a weighting factor in the interval (0,1) which penalises excessive changes in the controller output.

Predictions of the controlled output are made explicitly using the process model. Predictions over the entire prediction horizon of H_p time steps is achieved iteratively using earlier predictions where necessary.

To ensure good setpoint tracking, a mechanism for estimation and feedback of the modelling error similar to that used in the 1-step ahead predictive controller is also required here. The filtered estimate of the error, $err(t)$, is used to correct the predictions obtained from the model:

$$y_p(t+i) = y_m(t+i) + err(t), \qquad i = 1,\ldots,R \qquad (6.13)$$

The correction is carried out immediately after each prediction, and it is the corrrected predictions which are then used for further predictions using the iterative process discussed above. An assumption is made that $err(t)$ is constant over the entire prediction horizon. The sequence of corrected predictions is also used in the cost function minimisation.

Even though the control action can be calculated for a number of sampling times into the future, only the first one is actually implemented; the whole procedure is repeated again at the next sampling interval using the latest measured information. This is called the receding horizon principle. Using this approach, the horizon over which the process output is predicted shifts one sample into the future at every sample instant. This allows us to compensate for future disturbances or modelling errors.

There are a number of methods available to minimize the cost function, J. An efficient and accurate solution to this problem is not only dependent on the size of the problem in terms of the number of constraints and decision variables but also on characteristics of the objective function and constraints. Most algorithms employ some form of search technique to scan the feasible space of the objective function until an extremum point is located. The search is generally guided by calculations on the objective function and/or the derivatives of this function. A method which has been used for solving the multi-dimensional optimisation problem involving a nonlinear objective function and multiple constraints as presented above is successive quadratic programming (SQP).

The above discussion provides the conceptual framework for a numerical optimisation based multi-step predictive controller irrespective of the type of model used for prediction. To evaluate this control strategy using the fuzzy process model, we shall limit the control horizon to 1-step, i.e., the control signal is assumed to be constant over the entire prediction horizon. It is also assumed that the only important constraints are the upper and lower bounds on the controller output. Making these assumptions allows us to reduce the optimisation problem from a multi-dimensional search to a one-dimensional search which can be easily handled using the Fibonacci search method presented in the next section.

Figure 6.3 shows the steps involved in determining the optimal controller output at each sampling instant for the proposed fuzzy model-based controller. The figure incorporates the optimisation search procedure to be discussed in the next section. The number of iterations used by the optimisation search procedure (i.e. k in Figure 6.3) was standardised to 20.

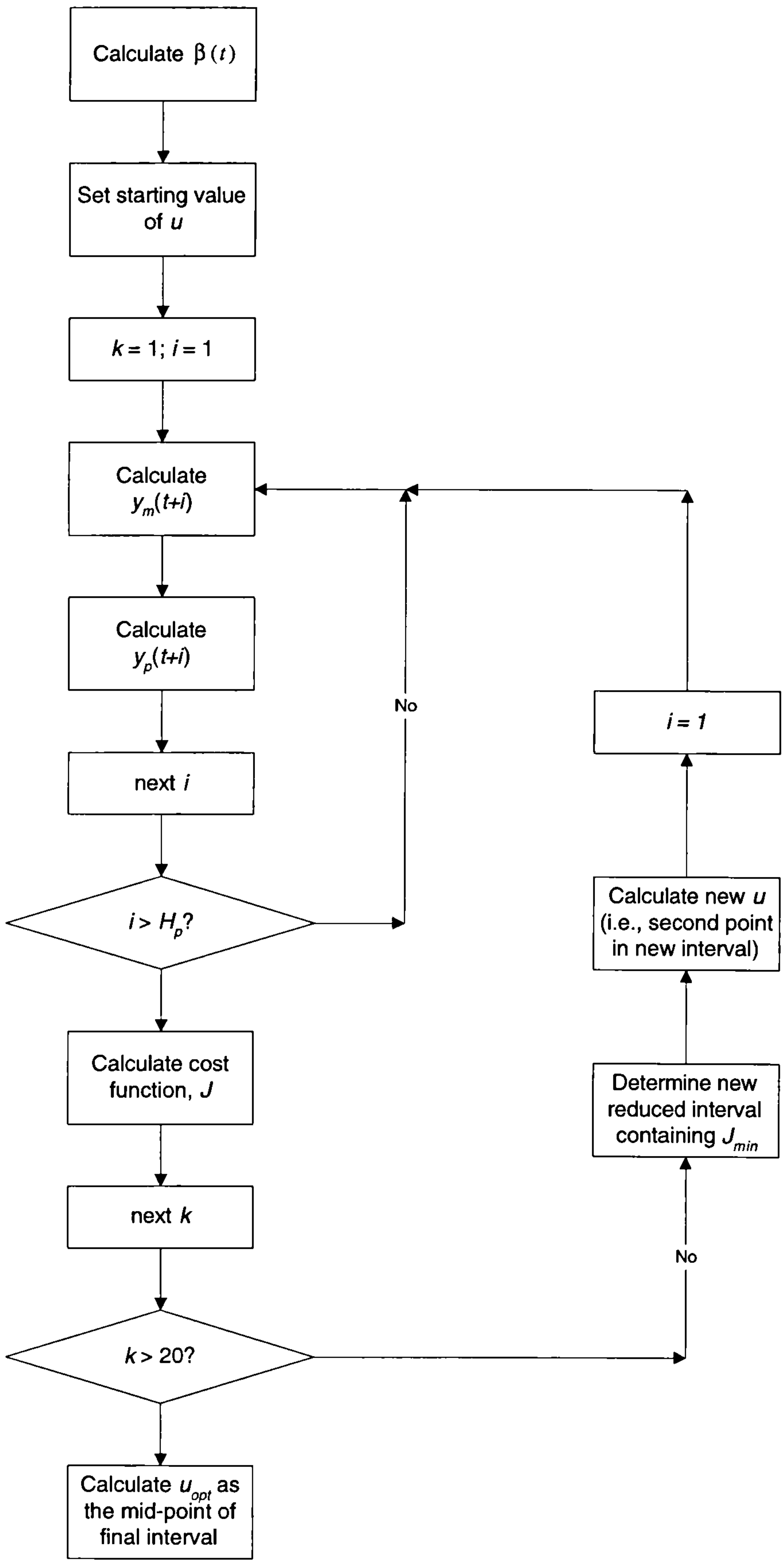


Figure 6.3: Flow diagram of the computations required at each sampling instant to determine the optimal controller output.

6.4 Numerical Optimisation Using Fibonacci Search

The discussion here will focus on finding a solution to a one-dimensional optimisation problem which can be represented by:

$$y = f(x) \tag{6.14}$$

subject to

$$x_1 \leq x \leq x_2 \tag{6.15}$$

where y is the value of the objective function, x is the independent variable, and $x_1 \leq x \leq x_2$ is the constrained range of x for the problem.

We begin by making the assumption that the objective function is unimodal over the bounded range of the independent variable. A unimodal function is shown in Figure 6.4 where there is a single minimum in the bounded range (x_1, x_2). Two points x_3 and x_4 are chosen within this interval such that:

$$x_1 < x_3 < x_4 < x_2 \tag{6.16}$$

Since the function has been assumed to be unimodal in the interval (x_1, x_2), we deduce that if $f(x_3) \geq f(x_4)$, then the minimum lies in the interval (x_3, x_2), whilst if $f(x_3) \leq f(x_4)$, the minimum lies in the interval (x_1, x_4). The second of these situations is illustrated in Figure 6.4.

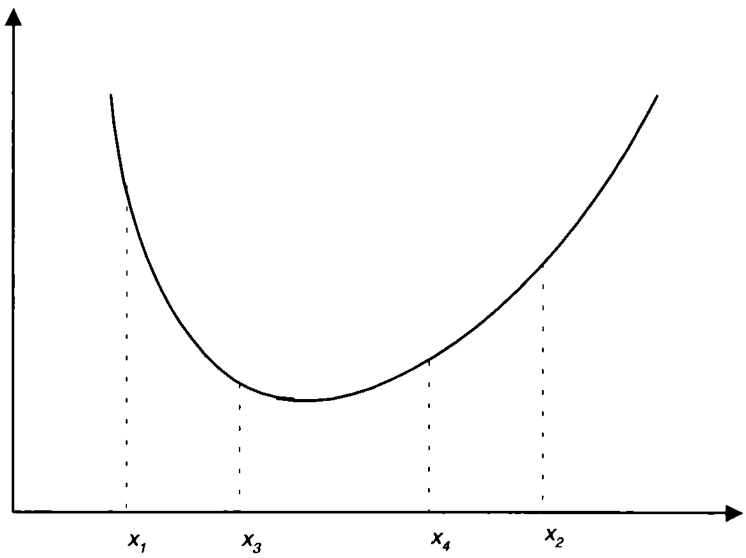


Figure 6.4: Example of a unimodal function.

Further reduction of the interval containing the minimum can only be achieved if more information, in the form of additional function-values is obtained. This process can then be continued until the interval containing the minimum has been reduced to a specified size.

Many plausible ways of selecting the points x_3 and x_4 have been proposed such *as 2-point equal interval search* and *method of bisecting* or *dichotomous search*. They are normally based on symmetric selection of these two points in order to obtain the same interval reduction factor regardless of which of the two possible sub-intervals is found to contain the minimum. In the 2-point equal interval search, x_3 and x_4 are chosen as the points of trisection of the interval (x_1,x_2). This procedure could then be repeated with the resulting reduced interval. It will be noted that the function has already been evaluated at the mid-point of this reduced interval, but this function evaluation is of no use in reducing the interval further by trisection. A more efficient search scheme would be to place x_3 and x_4 in positions such that the one enclosed within the reduced interval would constitute one of the two experiments within that interval. The method known as *Fibonacci search* is such a search scheme.

The Fibonacci search method is so-named on account of the use made of the sequence of positive integers known as the *Fibonacci numbers*. These are defined by the relations:

$$F_0 = F_1 = 1 \tag{6.17}$$

$$F_n = F_{n-1} + F_{n-2}, \qquad n \geq 2, \tag{6.18}$$

and therefore the sequence begins 1, 1, 2, 3, 5, 8, 13, 21, 34,... If N is the total number of function evaluations to be performed, the test points for the ith iteration are:

$$x_3^i = \frac{F_{N-1-i}}{F_{N+1-i}}(x_2^i - x_1^i) + x_1^i \tag{6.19}$$

and
$$x_4^i = \frac{F_{N-i}}{F_{N+1-i}}(x_2^i - x_1^i) + x_1^i \tag{6.20}$$

for $i = 1,2,\ldots,N\text{-}1$, where (x_1^1, x_2^1) is the initial interval.

However, the use of these rules makes the last two test points coincident at the midpoint of the interval (x_1^{N-1}, x_2^{N-1}). Therefore in order to determine in which half of the range (x_1^{N-1}, x_2^{N-1}) the minimum actually lies, we displace one of these final test points by an arbitrarily small amount ε.

It can easily be shown that the *i*th iteration reduces the interval containing the minimum by a factor (F_{N-i}/F_{N+1-i}), and from this it follows that after N function evaluations (that is, N-1 iterations), the length of the final interval is:

$$\frac{1}{2}\cdot\frac{F_2}{F_3}\cdot\frac{F_3}{F_4}\cdots\frac{F_{N-1}}{F_N}(x_2^1 - x_1^1)+\varepsilon = \frac{1}{F_N}(x_2^1 - x_1^1)+\varepsilon \tag{6.21}$$

at most.

Hence, if the minimum is required to an accuracy of δ, then N must be chosen so that:

$$F_N \geq \frac{(x_2^1 - x_1^1)}{\delta - \varepsilon} > F_{N-1} \tag{6.22}$$

Table 6.1 shows the number of function evaluations needed to achieve a certain accuracy using the Fibonacci search method. Accuracy is defined as the ratio of the size of the final interval to the size of the initial interval. An important property of this method is that this accuracy will be attained in N function evaluations for any unimodal function, and so in a sense Fibonacci search is the most efficient one-dimensional search procedure, since no other method can guarantee an interval reduction factor as large as F_N in N function evaluations. However many other methods, whilst not guaranteeing this accuracy, have been found in practice to be more efficient.

To complete this discussion of the Fibonacci search procedure, it remains to demonstrate that each iteration, except the first, requires just one function evaluation (versus 2 in equal interval and dichotomous search). This can be readily seen by supposing for example that for the *i*th iteration, the interval containing the minimum is (x_1^i, x_4^i). Then for the (i+1)th iteration:

$$x_1^{i+1} = x_1^i \tag{6.23}$$

$$x_2^{i+1} = x_4^i \tag{6.24}$$

and from Equations (6.19) and (6.20):

$$x_4^{i+1} = x_3^i \tag{6.25}$$

A parallel argument covers the case when the sub-interval containing the minimum at the *i*th iteration is (x_3^i, x_2^i).

The final output of this numerical search is an interval in which the minimum lies. It is assumed that the optimal value lies at the mid-point of this interval.

Accuracy	Number of experiments
0.1	6
0.01	11
0.001	16
0.0001	20

Table 6.1: Number of iterations required to achieve a given accuracy using the Fibonacci search.

6.5 Application to Control of Liquid Level

In this section, we will examine the application of the proposed control system to a simulation of the liquid level system presented in Section 4.3.1 and used in Section 5.4.1 to evaluate the 1-step ahead predictive controller. In the process models used here, the value of the constant term in equation (5.14) has been assumed to be a non-zero value. The control problem investigated here is exactly identical to Section 5.4.1. The gain of Filter 1 (K_{f_1}) has been standardised to 1 to prevent interference with the effect of the prediction horizon on the performance of the controller.

Figures 6.5 to 6.8 show the effect of the prediction horizon on the performance of controllers using linear and 5-partition fuzzy process models. It will be observed that the output response becomes less oscillatory and more sluggish as the number of steps in the prediction horizon is increased.

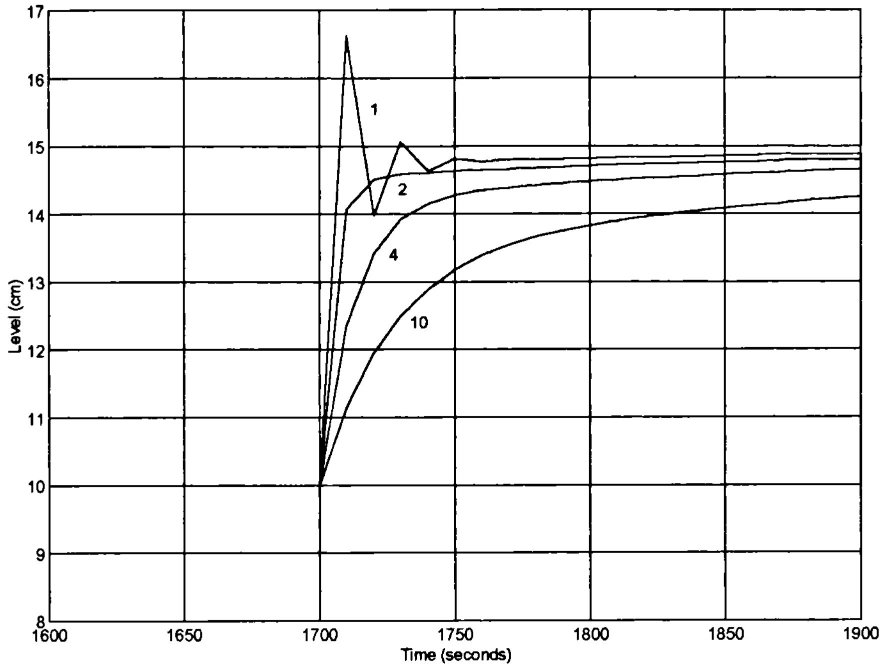


Figure 6.5: Effect of number of steps in prediction horizon on process output response to setpoint changes between 10 and 15 cm. when using proposed controller with linear model ($K_{f_1} = 1$; $K_{f_2} = 0.05; k_i \neq 0$).

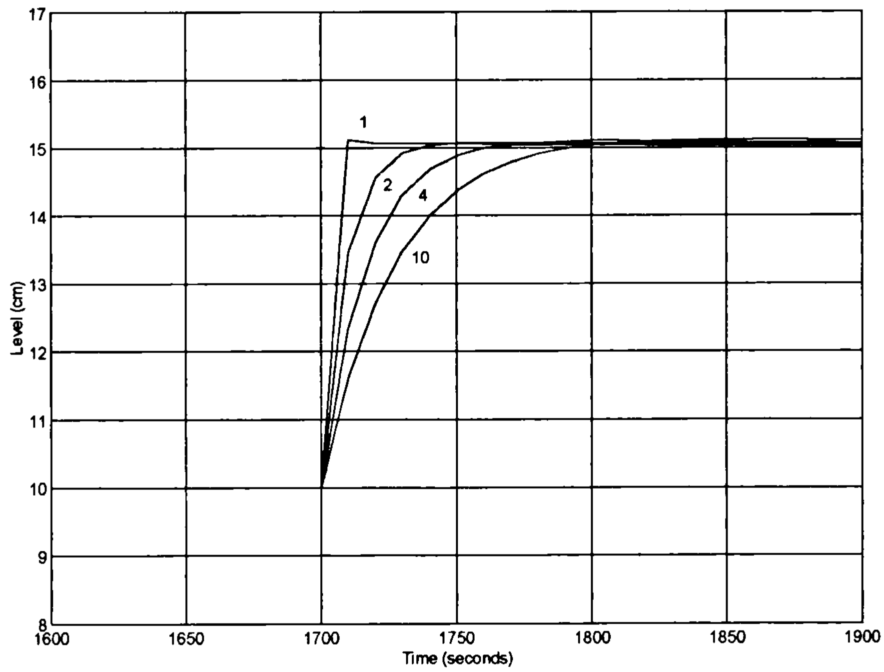


Figure 6.6: Effect of number of steps in prediction horizon on process output response to setpoint changes between 10 and 15 cm. when using proposed controller with 5-partition fuzzy model ($K_{f_1} = 1$; $K_{f_2} = 0.05; k_i \neq 0$).

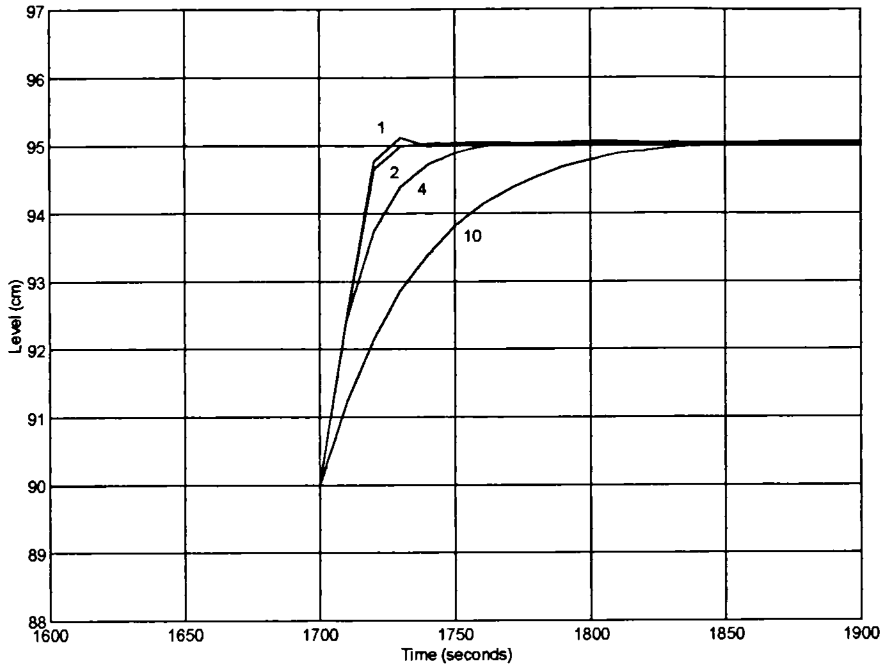


Figure 6.7: Effect of number of steps in prediction horizon on process output response to setpoint changes between 90 and 95 cm. when using proposed controller with linear model ($K_{f_1} = 1$; $K_{f_2} = 0.05; k_i \neq 0$).

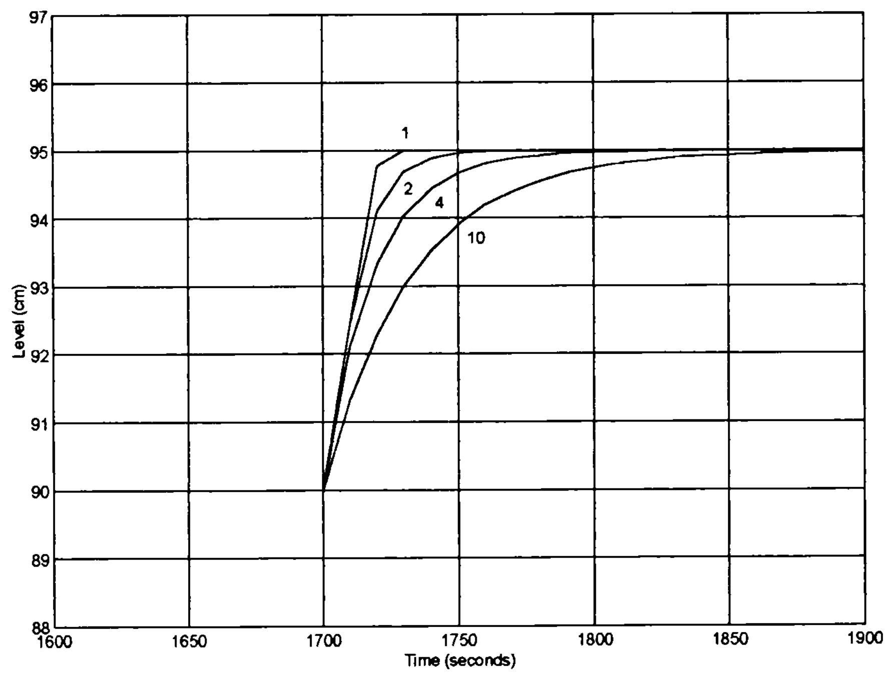


Figure 6.8: Effect of number of steps in prediction horizon on process output response to setpoint changes between 90 and 95 cm. when using proposed controller with 5-partition fuzzy model ($K_{f_1} = 1$; $K_{f_2} = 0.05; k_i \neq 0$).

6.6 Conclusions

It has been shown in this chapter that a multi-step predictive controller can be designed based on the fuzzy process model presented in Chapter 4. Investigations carried out using the tank level simulation shows that the output response becomes less oscillatory and more sluggish as the number of steps in the prediction horizon is increased.

A detailed analysis of the performance of the controller has been deferred until after the analytically-based multi-step predictive controller is presented in Chapter 7.

Chapter 7

LONG-RANGE PREDICTIVE CONTROL: ANALYTICAL APPROACH

7.1 Introduction

Although the long-range predictive controller described in Chapter 6 has been shown to work quite well with the optimal controller output determined numerically using the Fibonacci search method, the computation time can be quite considerable, especially if the prediction horizon is very long. Also, trying to determine the optimal controller output for a control horizon of greater than 1-step will involve a multivariable search technique such as successive quadratic programming (SQP). In this chapter, we will examine an alternative analytical approach for deriving the optimal controller output in the multi-step predictive controller. To facilitate the design of the controller, it is necessary to linearise the fuzzy model at every sampling instant by the weighting method discussed in Chapter 5. Section 7.2 compares the effectiveness of this linearisation method with the usual linearisation method used in control engineering.

The performances of controllers using analytical and numerical approaches will be compared. An attempt will also be made to examine the performance of the controller in the presence of noise and when there are changes in process conditions (i.e., robustness tests).

7.2 Linearisation By Weighting Fuzzy Model Parameters

Even though the fuzzy model consists of a number of linear sub-models, the overall model output is non-linear. To facilitate the design of a multi-step predictive controller using an analytical approach, we will now examine a simple method of linearising the fuzzy model about the current operating point.

It has been shown in Chapter 6 that the value of β calculated for the first prediction step can be used over the entire prediction horizon. This is convenient since it implies that the linear model determined by weighting the fuzzy model parameters at each sampling instant can be used to represent the fuzzy model for making predictions over the entire prediction horizon. Recall from Chapter 5, that the weighted model parameters are given by:

$$\Phi' = \left[a_1' \cdots a_j' \quad b_1' \cdots b_l' \quad k'\right] \tag{7.1}$$

$$= \beta \, \Phi \tag{7.2}$$

where Φ' denotes the weighted model parameters, and Φ denotes the matrix providing the fuzzy model parameters.

Assume that the fuzzy partitioning of the input space shown in Figure 7.1 is used for deriving the fuzzy model. At point x_1, the model output is given exactly by the fuzzy implication used in sub-space A^3, and at point x_2, the model output is given exactly by the fuzzy implication used in sub-space A^4. Let us represent the model outputs at x_1 and x_2 as y_1 and y_2, respectively. Assume that the nonlinear fuzzy model output is monotonically increasing over the region x_1-x_2. We wish to approximate the behaviour of the nonlinear model over x_1-x_2 using a linear model.

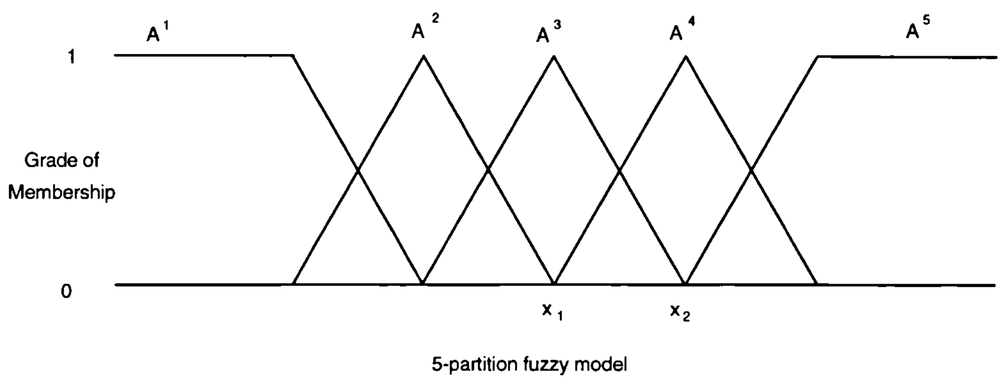


Figure 7.1: Fuzzy partitioning of the input space used for deriving fuzzy model.

The output, y_3 of the linear model derived at any point, x_3 in the region x_1 - x_2 by weighting the parameters of the fuzzy implications in A[3] and A[4] will satisfy the following relationship:

$$y_1 \le y_3 \le y_2 \tag{7.3}$$

This establishes the upper and lower limits of the output of the linear model at any point x_3. Also, if the linear model derived at x_3 is used with any other value of x in the region x_1 - x_2, the above relationship should still hold.

One way to improve the accuracy of the linear model in the region around x_3 is to reduce the interval x_1 - x_2 by increasing the number of fuzzy partitions. This will also reduce the interval $(y_2 - y_1)$.

To compare this method of linearisation with the traditional method of linearisation, we consider first a simple 2-partition fuzzy model described by the following two rules:

IF x is B^1 THEN $y = 2x$ (7.4)
IF x is B^2 THEN $y = 4x$

where B^1 and B^2 denote the fuzzy partitions shown in Figure 7.2. The overall non-linear model from $x = 0$ to $x = 10$ is given by:

$$\begin{aligned} y_1 &= \left(\frac{10-x}{10}\right)\cdot 2x + \left(\frac{x}{10}\right)\cdot 4x \\ &= 0.2x^2 + 2x \end{aligned} \tag{7.5}$$

Since the non-linear model is available, it is possible to linearize by determining the gradient. Linearisation at the point $x = 5$ leads to,

$$\begin{aligned} y_2 &= 15 + \left(\frac{dy}{dx}\right)_{x=5}(x-5) \\ &= 4x - 5 \end{aligned} \tag{7.6}$$

Linearisation by weighting the model parameters at $x = 5$ as described above leads to,

$$\begin{aligned} y_3 &= 0.5(2x) + 0.5(4x) \\ &= 3x \end{aligned} \tag{7.7}$$

From Figure 7.3 which shows plots of the 3 models, it will be noticed that the 2 linearised models can provide quite different results.

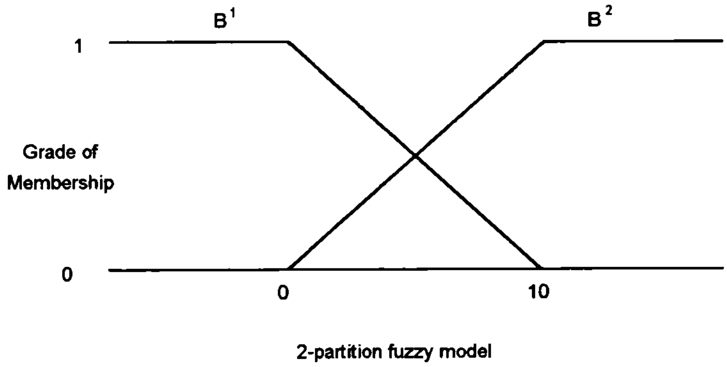


Figure 7.2: Fuzzy partitioning of the input space used by the fuzzy model.

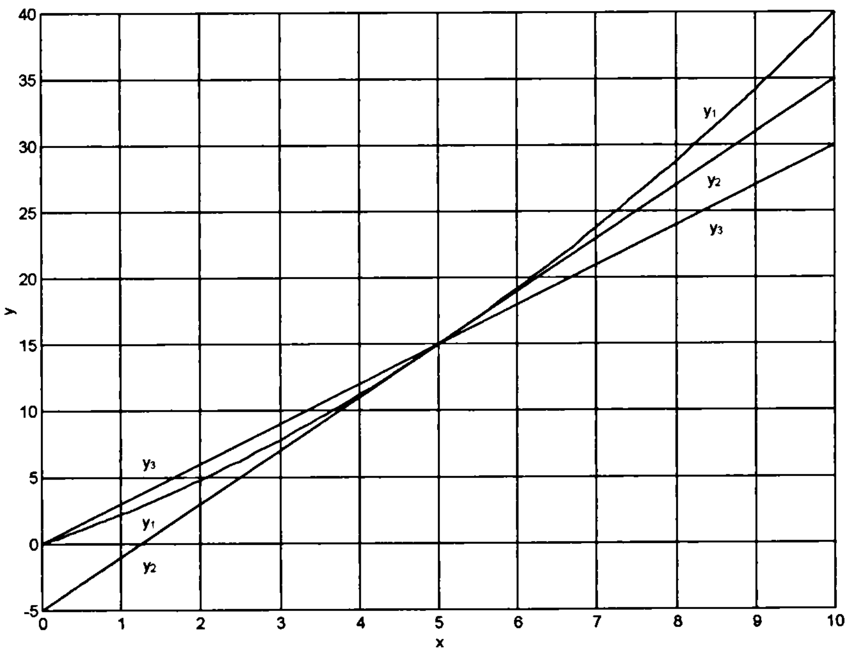


Figure 7.3: Plot of nonlinear model and linearised models at $x = 5$.

Consider instead the 5-partition fuzzy model described by the following fuzzy rules:

$$
\begin{aligned}
&\text{IF } x \text{ is } B^1 \text{ THEN } y = 2x \\
&\text{IF } x \text{ is } B^2 \text{ THEN } y = 2.5x \\
&\text{IF } x \text{ is } B^3 \text{ THEN } y = 3x \\
&\text{IF } x \text{ is } B^4 \text{ THEN } y = 3.5x \\
&\text{IF } x \text{ is } B^5 \text{ THEN } y = 4x
\end{aligned} \tag{7.8}
$$

where B^1 to B^5 denote the fuzzy partitions shown in Figure 7.4. The overall model from $x = 0$ to $x = 2.5$ and the two linearised models at $x = 1.25$ are:

$$y_1 = 0.2x^2 + 2x \tag{7.9}$$

$$y_2 = 2.5x - 0.3125 \tag{7.10}$$

$$y_3 = 2.25x \tag{7.11}$$

Figure 7.5 shows plots of these 3 models. In this case, there is only a small difference between the 2 linearised models. Also, it will be noticed that the linear models approximate the non-linear model quite closely. Increasing the number of fuzzy partitions will lead to a more gradual transition in fuzzy model parameters between adjacent fuzzy partitions. Hence, by increasing the number of fuzzy partitions, the difference between the two linearisation methods can be made negligible and good local approximation of the nonlinear model can be achieved by using either of the two linear models.

It has been shown above that better linearised modelling accuracy by weighting the fuzzy model parameters can be achieved by increasing the number of fuzzy partitions. This improvement in accuracy should be reflected in better performance by the controller designed using the linearised model. In practice, it is found that the controller's performance tends to converge as the number of fuzzy partitions is increased, and there is little benefit to be gained by using too many fuzzy partitions. In our experiments, it was generally not necessary to go beyond 5 fuzzy partitions.

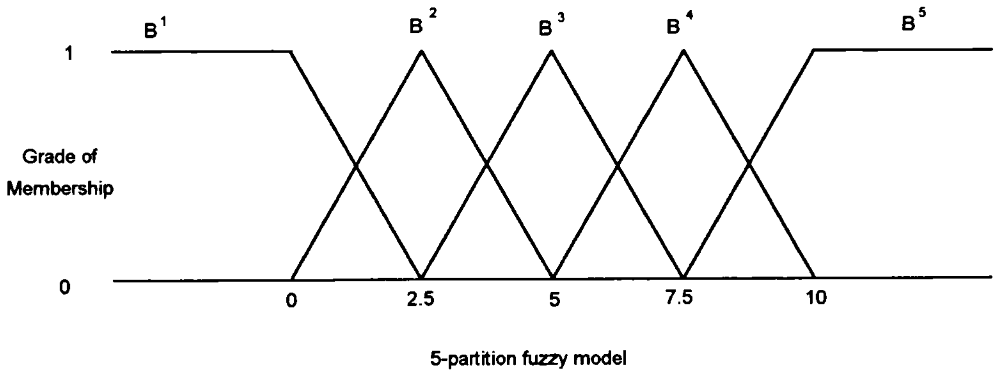


Figure 7.4: Fuzzy partitioning of the input space used by the fuzzy model.

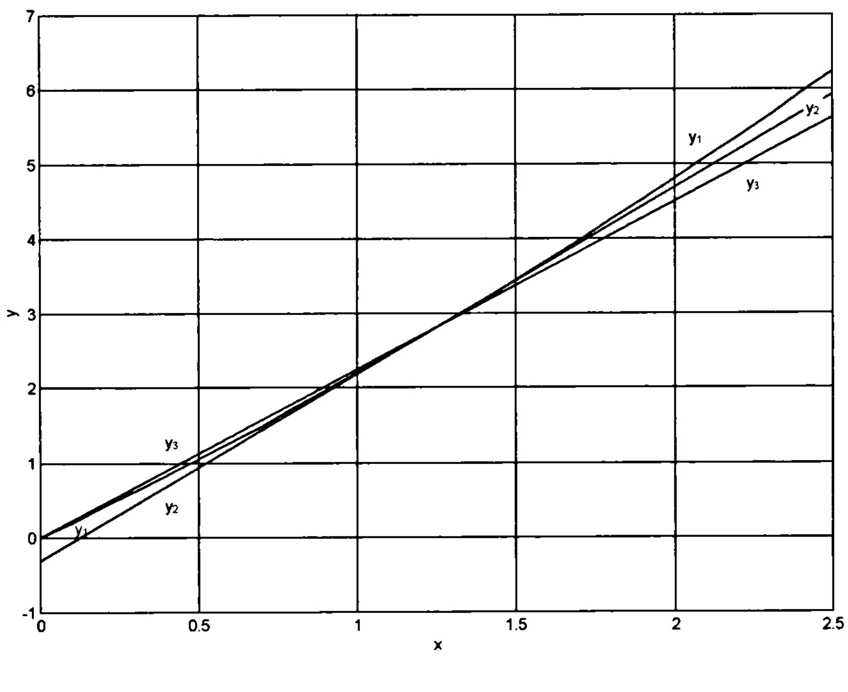


Figure 7.5: Plot of nonlinear model and linearised models at $x = 1.25$.

7.3 Controller Formulation

Some unique problems need to be addressed when attempting to develop a MBPC strategy incorporating the above fuzzy process model. MBPC strategies such as Dynamic Matrix Control (DMC) (Cutler and Ramaker, 1978) and Generalised Predictive Control (GPC) (Clarke *et al.*, 1987) are based on models which linearize processes locally. Even though the fuzzy model is essentially a nonlinear model, it is also possible to linearise as shown in Section 7.2. There are, however, some important differences. Linear models are normally used with deviation variables in control loops, whereas the fuzzy model requires the absolute values of variables. Also, MBPC strategies such as GPC use the incremental form of model representation to enable an integrating action to overcome the offset problem due to modelling inaccuracies. A method of expressing the fuzzy model in incremental form was not readily obvious. Hence, an MBPC strategy which works with the linear ARX model structure where the absolute values of variables can be used and which incorporates a feedback mechanism to overcome the offset due to modelling inaccuracies is needed. A control strategy satisfying these requirements based on analytical derivation of the 1-step ahead optimal controller output was proposed in Chapter 5. The controller to be described below extends this concept using a long-range predictive control strategy. The approach used to derive our control strategy follows that used for GPC by Clarke *et al.* (1987) because of the similarities between the fuzzy process model and the transfer function process model used in GPC.

Consider a single-input single-output discrete time linear system described by the general ARX model structure. The model predictions over a prediction horizon of n time steps is given by:

$$y_p(t+1) = a_1 y(t) + \ldots + a_j y(t+1-j) + b_1 u(t) + \ldots + b_l u(t+1-l) + k + err(t)$$

$$y_p(t+2) = a_1 y(t+1) + \ldots + a_j y(t+2-j) + b_1 u(t+1) + \ldots + b_l u(t+2-l) + k + err(t)$$

$$\vdots \qquad\qquad \vdots$$

$$y_p(t+n) = a_1 y(t+n-1) + \ldots + a_j y(t+n-j) + b_1 u(t+n-1) + \ldots + b_l u(t+n-l) + k + err(t) \tag{7.12}$$

where $err(t)$ is an estimate of the modelling error and $y_p(t+1)$ to $y_p(t+n)$ are the model predictions after corrections for modelling error at time t. It has been assumed that the modelling error is constant over the entire prediction horizon. If the control horizon is m, the values of $u(t)$ following $u(t+m-1)$ can be assumed to be constant and equal to $u(t+m-1)$.

Using the above equations, predictions over the entire prediction horizon are achieved iteratively using earlier predictions where required. Alternatively, the above equations can be transformed into the following form using back-substitutions:

$$
\begin{aligned}
y_p(t+1) &= p_{11}y(t)+\ldots+p_{1j}y(t+1-j)+p_{1(j+1)}u(t-1)+\ldots+p_{1(j+l-1)}u(t+1-l) \\
&\quad +q_{11}u(t)+r_1k+r_1err(t) \\
y_p(t+2) &= p_{21}y(t)+\ldots+p_{2j}y(t+1-j)+p_{2(j+1)}u(t-1)+\ldots+p_{2(j+l-1)}u(t+1-l) \\
&\quad +q_{21}u(t)+q_{22}u(t+1)+r_2k+r_2err(t) \\
&\vdots \\
y_p(t+m) &= p_{m1}y(t)+\ldots+p_{mj}y(t+1-j)+p_{m(j+1)}u(t-1)+\ldots+p_{m(j+l-1)}u(t+1-l) \\
&\quad +q_{m1}u(t)+\ldots+q_{mm}u(t+m-1)+r_mk+r_merr(t) \\
&\vdots \\
y_p(t+n) &= p_{n1}y(t)+\ldots+p_{nj}y(t+1-j)+p_{n(j+1)}u(t-1)+\ldots+p_{n(j+l-1)}u(t+1-l) \\
&\quad +q_{n1}u(t)+\ldots+q_{nm}u(t+m-1)+r_nk+r_nerr(t)
\end{aligned}
\tag{7.13}
$$

or,

$$Y(t) = \mathbf{P}X(t) + \mathbf{Q}U(t) + \mathbf{R}\left[k + err(t)\right] \tag{7.14}$$

where,

$$Y(t) = \left[y_P(t+1) \quad \cdots \quad y_P(t+n)\right]^T \tag{7.15}$$

$$X(t) = \left[y(t) \quad y(t-1) \quad \cdots \quad y(t+1-j) \quad u(t-1) \quad \cdots \quad u(t+1-l)\right]^T \tag{7.16}$$

$$U(t) = \left[u(t) \quad \cdots \quad u(t+m-1)\right]^T \tag{7.17}$$

$$\mathbf{P} = \begin{bmatrix} p_{11} & \cdots & p_{1(j+l-1)} \\ \vdots & & \vdots \\ p_{n1} & \cdots & p_{n(j+l-1)} \end{bmatrix} \tag{7.18}$$

$$\mathbf{Q}=\begin{bmatrix} q_{11} & 0 & 0 & 0 & 0 \\ q_{21} & q_{22} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ q_{m1} & q_{m2} & q_{m3} & \cdots & q_{mm} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ q_{n1} & q_{n2} & q_{n3} & \cdots & q_{nm} \end{bmatrix} \tag{7.19}$$

$$\mathbf{R}=\begin{bmatrix} r_1 & \cdots & r_n \end{bmatrix}^T \tag{7.20}$$

$Y(t)$ denotes the model predictions over the prediction horizon, $X(t)$ is a vector of past plant and controller outputs and $U(t)$ is a vector of future controller outputs. If the coefficients of **P**, **Q** and **R** can be determined then the transformation can be completed. The number of columns in **P** is determined by the ARX model structure used to represent the system whereas the number of columns in **Q** is determined by the length of the control horizon. If the control horizon is 1-step, then **Q** will consist of only one column. The number of rows is fixed by the length of the prediction horizon. The calculation of the coefficients of **P**, **Q** and **R** for first-order and second-order systems for a control horizon of 1-step are discussed later on in this chapter.

In order to define how well the predicted process output tracks the setpoint, the following cost function is used:

$$J=\sum_{i=1}^{n}[y_p(t+i)-w(t+i)]^2 \tag{7.21}$$

$$=\left[Y(t)-W(t)\right]^T\left[Y(t)-W(t)\right] \tag{7.22}$$

$$=\left[\mathbf{P}X(t)+\mathbf{Q}U(t)+\mathbf{R}(k+err(t))-W(t)\right]^T\left[\mathbf{P}X(t)+\mathbf{Q}U(t)+\mathbf{R}(k+err(t))-W(t)\right] \tag{7.23}$$

where,

$$W(t)=\left[w(t+1) \quad \cdots \quad w(t+n)\right]^T \tag{7.24}$$

$W(t)$ is a vector of the setpoints over the prediction horizon. The optimal controller output sequence can be found by minimising the above cost function. An analytical solution is available if the cost function is quadratic, the model is linear and there are no constraints. A necessary condition for minimum J is:

$$\frac{\partial J}{\partial U}=0 \tag{7.25}$$

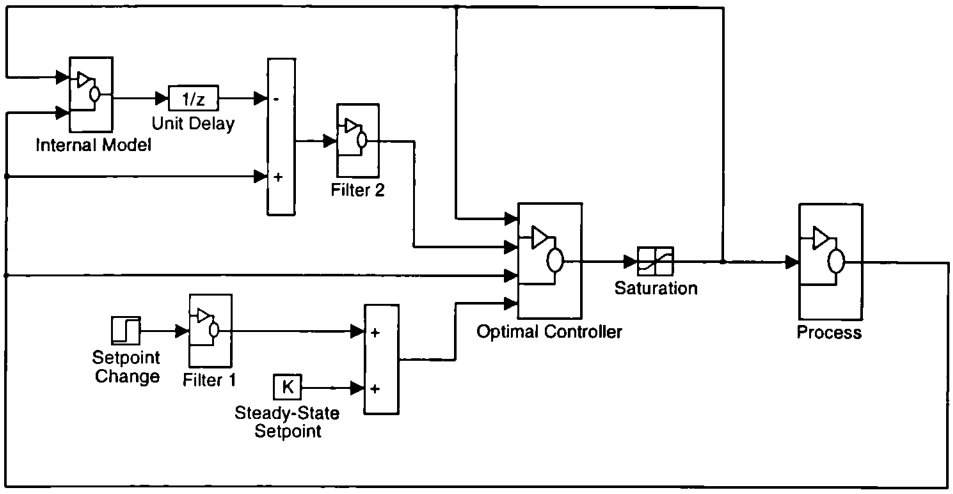


Figure 7.6: Proposed fuzzy model-based predictive controller structure.

Differentiating the expression for J and using the above condition leads to the following optimal solution:

$$U(t) = [\mathbf{Q}^T\mathbf{Q}]^{-1}\mathbf{Q}^T\left[W(t) - \mathbf{P}X(t) - \mathbf{R}(k + err(t))\right] \tag{7.26}$$

Adopting the above control strategy to work with the fuzzy model only requires substituting the normal linear ARX model parameters with the weighted model parameters determined at each sampling instant from the fuzzy model using the method proposed above. Hence, the overall control strategy consists of the following 3 steps carried out in succession at each sampling instant:

1. Calculate the weighted model parameters vector, $\mathbf{\Phi}'$ using the fuzzy process model and the current value of the process output.

2. Calculate the coefficients of **P**, **Q** and **R** matrices using $\mathbf{\Phi}'$.

3. Calculate the optimal controller output sequence, $U(t)$ using equation (7.26).

Even though the control sequence over the entire control horizon will have been calculated, only the first one is actually implemented; the whole procedure is repeated again at the next sampling interval using the latest measured information. This is called the receding horizon principle. Using this approach, the horizon over which the process output is predicted shifts one sample into the future at every sampling instant.

Implementation of the above control scheme uses a similar approach to the one-step ahead predictive controller (Figure 7.6) and will therefore not be further elaborated here.

7.3.1 First-Order System

A first-order system can be described by the following ARX model structure:

$$y_p(t+1) = a_1 y(t) + b_1 u(t) + k + err(t) \tag{7.27}$$

Model predictions over a prediction horizon of n-steps based on a control horizon of 1-step is given by the equation:

$$Y(t) = \mathbf{P}X(t) + \mathbf{Q}U(t) + \mathbf{R}[k + err(t)] \tag{7.28}$$

where,

$$Y(t) = \begin{bmatrix} y_p(t+1) & \cdots & y_p(t+n) \end{bmatrix}^T \tag{7.29}$$

$$X(t) = [y(t)] \tag{7.30}$$

$$U(t) = [u(t)] \tag{7.31}$$

$$\mathbf{P} = \begin{bmatrix} p_1 & \cdots & p_n \end{bmatrix}^T \tag{7.32}$$

$$\mathbf{Q} = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}^T \tag{7.33}$$

$$\mathbf{R} = \begin{bmatrix} r_1 & \cdots & r_n \end{bmatrix}^T \tag{7.34}$$

Back-substitution during model transformation leads to the following recursive formulae for determining the coefficients of **P**, **Q** and **R**:

$$\begin{aligned} p_1 &= a_1 \\ p_i &= a_1 \cdot p_{i-1}, \text{ for } i = 2,\ldots,n \end{aligned} \tag{7.35}$$

$$\begin{aligned} q_1 &= b_1 \\ q_i &= a_1 \cdot q_{i-1} + b_1, \text{ for } i = 2,\ldots,n \end{aligned} \tag{7.36}$$

$$\begin{aligned} r_1 &= 1 \\ r_i &= a_1 \cdot r_{i-1} + 1, \text{ for } i = 2,\ldots,n \end{aligned} \tag{7.37}$$

7.3.2 Second-Order System

A second-order system can be described by the following ARX model structure:

$$y_p(t+1) = a_1 y(t) + a_2 y(t-1) + b_1 u(t) + b_2 u(t-1) + k + err(t) \tag{7.38}$$

Model predictions over a prediction horizon of n-steps based on a control horizon of 1-step is given by the equation:

$$Y(t) = \mathbf{P}X(t) + \mathbf{Q}U(t) + \mathbf{R}[k + err(t)] \tag{7.39}$$

where,

$$Y(t) = \begin{bmatrix} y_p(t+1) & \cdots & y_p(t+n) \end{bmatrix}^T \tag{7.40}$$

$$X(t) = \begin{bmatrix} y(t) & y(t-1) & u(t-1) \end{bmatrix}^T \tag{7.41}$$

$$U(t) = \begin{bmatrix} u(t) \end{bmatrix} \tag{7.42}$$

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ \vdots & \vdots & \vdots \\ p_{n1} & p_{n2} & p_{n3} \end{bmatrix} \tag{7.43}$$

$$\mathbf{Q} = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}^T \tag{7.44}$$

$$\mathbf{R} = \begin{bmatrix} r_1 & \cdots & r_n \end{bmatrix}^T \tag{7.45}$$

Back-substitution during model transformation leads to the following recursive formulae for determining the coefficients of **P**, **Q** and **R**:

$$\begin{array}{lll} p_{11} = a_1 & p_{12} = a_2 & p_{13} = b_2 \\ p_{21} = a_1 \cdot p_{11} + a_2 & p_{22} = a_1 \cdot p_{12} & p_{23} = a_1 \cdot p_{13} \\ p_{i1} = a_1 p_{(i-1)1} + a_2 p_{(i-2)1} & p_{i2} = a_1 p_{(i-1)2} + a_2 p_{(i-2)2} & p_{i3} = a_1 p_{(i-1)3} + a_2 p_{(i-2)3}, \\ & & \text{for } i = 3,\ldots,n \end{array} \tag{7.46}$$

$$\begin{array}{l} q_1 = b_1 \\ q_2 = a_1 \cdot q_1 + b_1 + b_2 \\ q_i = a_1 \cdot q_{(i-1)} + a_2 \cdot q_{(i-2)} + b_1 + b_2, \text{ for } i = 3,\ldots,n \end{array} \tag{7.47}$$

$$\begin{array}{l} r_1 = 1 \\ r_2 = a_1 + 1 \\ r_i = a_1 \cdot r_{i-1} + a_2 \cdot r_{i-2} + 1, \text{ for } i = 3,\ldots,n \end{array} \tag{7.48}$$

Hence, the coefficients of **P**, **Q** and **R** can be determined quite easily for any length of prediction horizon in the case of first-order and second-order systems.

7.4 Application to Control of Liquid Level

The gain of Filter 1 (K_{f_1}) has been standardised to 1 so that the effect of the prediction horizon on the performance of the controller could be studied free of the influence of this parameter. The gain of Filter 2 (K_{f_2}) was set to the value 0.05 based on studies carried out on the 1-step ahead predictive controller in Chapter 5 which showed that this value of K_{f_2} leads to close to optimum performance. The control horizon has been standardised to 1-step; and the process models have included the constant term, k_i since studies in Chapter 5 have shown that this leads to better controller performance.

Analysis of the IAE values in Tables 7.1 and 7.2 and Figures 7.7 to 7.14 will reveal all-round good performance from the controller using the 5-partition fuzzy model when the prediction horizon is about 2-steps. The IAE achievable with this controller is close to the minimum achievable IAE at both levels under noisy conditions. The performance when using the 2-partition and 3-partition fuzzy models is only marginally worse than the 5-partition fuzzy model.

It is also clearly evident that fuzzy model based controllers generally perform better than the controller using the linear model. A considerable amount of time is needed to reach the steady-state level at the lower level when using the linear model.

Two important general observations can be made from this application study. Firstly, it will be noticed that the output response becomes less oscillatory and more sluggish as the number of steps in the prediction horizon increases. Secondly, the difference in the performance of controllers using fuzzy process models seems to more significant when the prediction horizon is small. These observations are discussed further in Section 7.5.

It will be observed in Figure 7.10 that a considerable amount of time is needed for the steady-state offset correction. Even though better modelling accuracy is achieved with the 5-partition fuzzy model, there may be slight variations in modelling accuracy over the range from 10 to 15 cm., especially under steady-state operating conditions. Variations in the modelling accuracy are probably smaller near the top of the tank. The time needed for offset correction can be reduced by increasing the value of K_{f_2}.

Prediction horizon	Linear model		2-partition fuzzy model		3-partition fuzzy model		5-partition fuzzy model	
	IAE without noise	IAE with noise+	IAE without noise	IAE with noise+	IAE without noise	IAE with noise+	IAE without noise	IAE with noise+
1	267.9	623.0	203.5	558.2	173.2	503.8	189.1	489.1
2	338.0	507.6	214.2	472.8	212.3	440.3	249.6	434.1
3	454.0	543.3	245.4	456.5	278.4	437.9	324.6	445.9
4	566.9	622.0	272.0	452.0	330.3	459.4	384.4	474.0
5	667.1	704.8	293.8	450.9	371.5	480.6	432.2	500.8
10	1026.9	1031.4	342.5	464.4	485.3	550.5	568.7	594.2
20	1370.5	1359.5	365.8	475.2	553.9	602.4	655.8	667.7

\+ Uniformly distributed noise in the range -1 and +1 was added to the output from the simulation before being used by the controllers.

++ Gain of Filter 1 = 1; gain of Filter 2 = 0.05; IAE was derived using a series of setpoint changes between 10 cm. and 15 cm. over a period of 1000 seconds starting from 1500 seconds after the start of a simulation.

Table 7.1: Effect of the prediction horizon on the performance of the long-range predictive controller to setpoint changes between 10 cm. and 15 cm.

Prediction horizon	Linear model		2-partition fuzzy model		3-partition fuzzy model		5-partition fuzzy model	
	IAE without noise	IAE with noise+	IAE without noise	IAE with noise+	IAE without noise	IAE with noise+	IAE without noise	IAE with noise+
1	218.6	658.2	217.7	502.1	198.4	542.1	191.8	538.7
2	223.9	498.1	284.4	479.1	269.9	480.2	267.6	477.6
3	287.6	473.2	329.6	489.7	334.2	488.6	336.7	491.6
4	346.9	487.8	362.7	504.9	396.3	520.5	411.4	531.9
5	419.4	526.4	392.1	521.1	453.6	555.4	478.0	574.0
10	729.9	757.0	459.2	561.6	626.6	680.6	695.0	735.9
20	1146.5	1133.0	491.9	583.6	744.9	774.4	861.3	870.1

\+ Uniformly distributed noise in the range -1 and +1 was added to the output from the simulation before being used by the controllers.

++ Gain of Filter 1 = 1; gain of Filter 2 = 0.05; IAE was derived using a series of setpoint changes between 90 cm. and 95 cm. over a period of 1000 seconds starting from 1500 seconds after the start of a simulation.

Table 7.2: Effect of the prediction horizon on the performance of the long-range predictive controller to setpoint changes between 90 cm. and 95 cm.

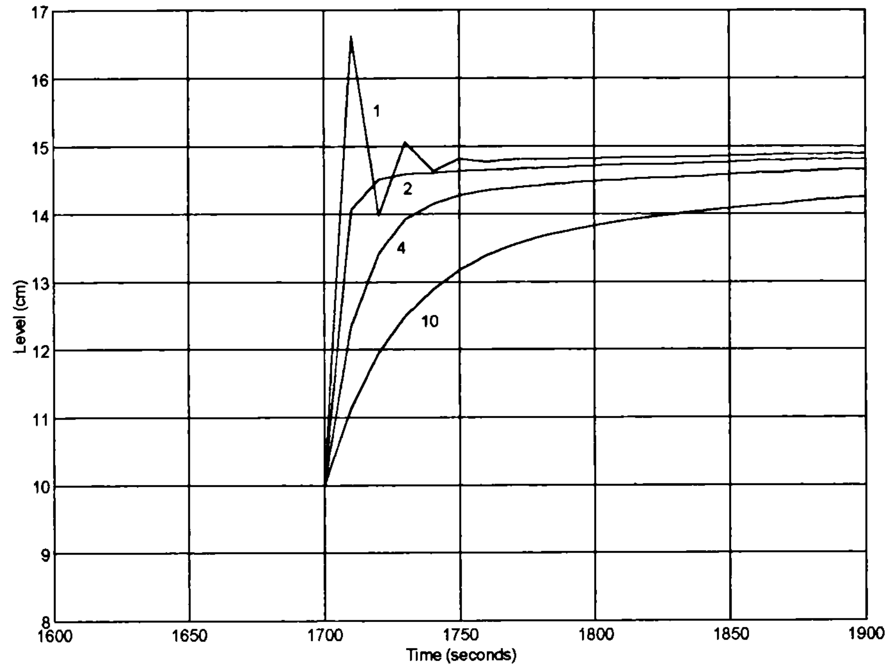


Figure 7.7: Effect of number of steps in prediction horizon on process output response to setpoint changes between 10 and 15 cm. when using proposed controller with linear model ($K_{f_1} = 1$; $K_{f_2} = 0.05; k_i \neq 0$).

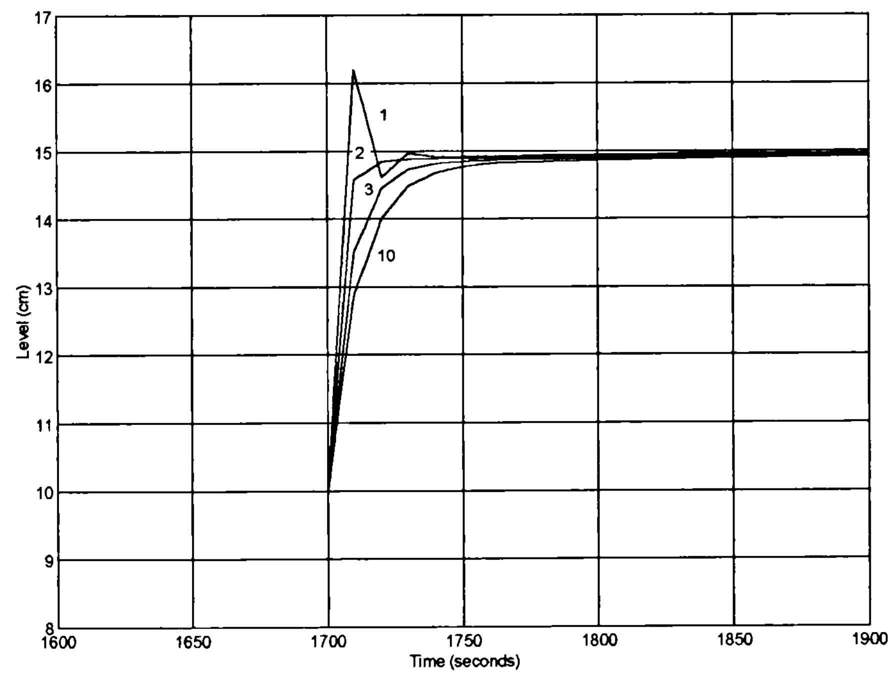


Figure 7.8: Effect of number of steps in prediction horizon on process output response to setpoint changes between 10 and 15 cm. when using proposed controller with 2-partition fuzzy model ($K_{f_1} = 1$; $K_{f_2} = 0.05; k_i \neq 0$).

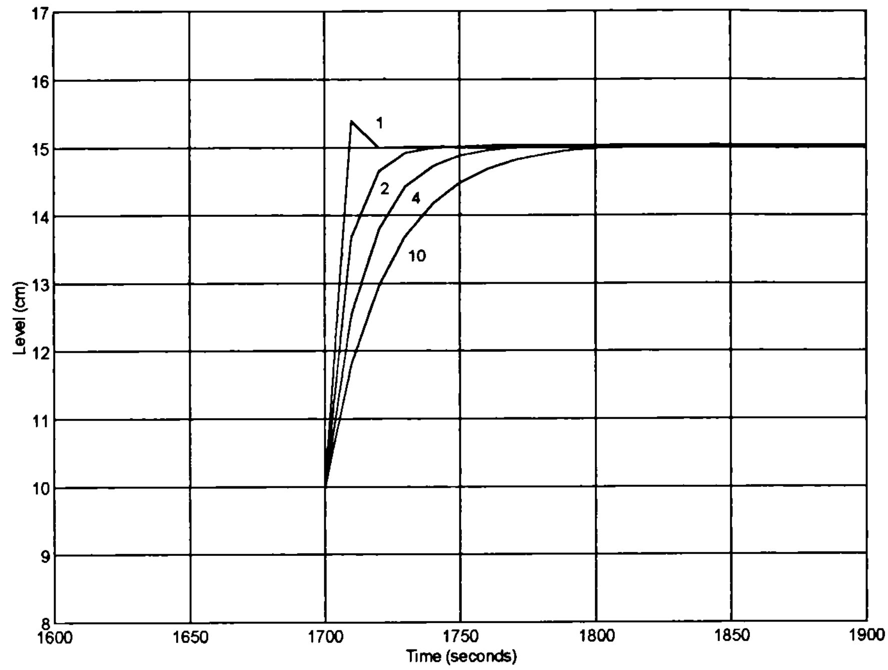


Figure 7.9: Effect of number of steps in prediction horizon on process output response to setpoint changes between 10 and 15 cm. when using proposed controller with 3-partition fuzzy model ($K_{f_1} = 1;\ K_{f_2} = 0.05; k_i \neq 0$).

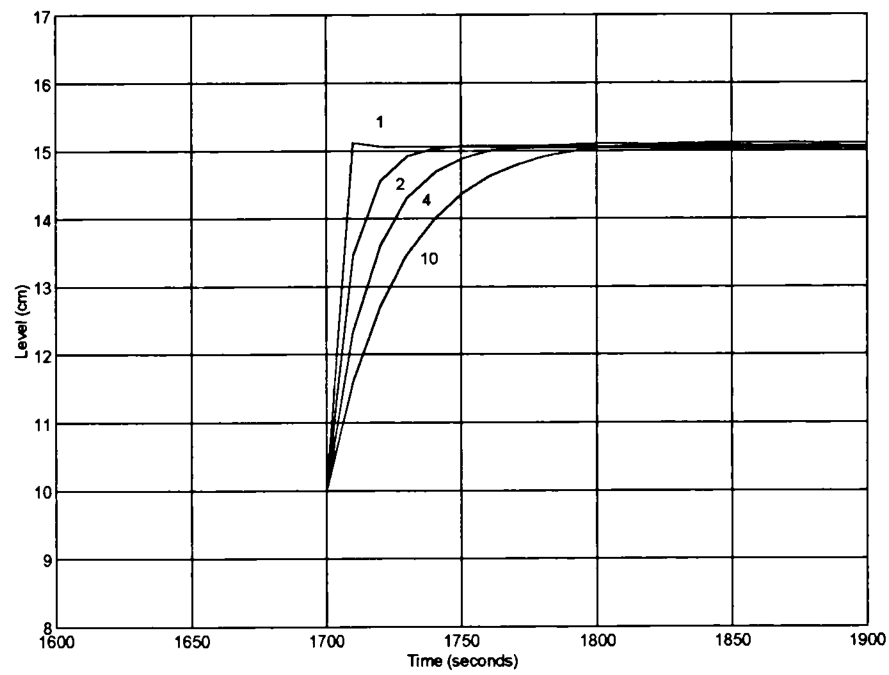


Figure 7.10: Effect of number of steps in prediction horizon on process output response to setpoint changes between 10 and 15 cm. when using proposed controller with 5-partition fuzzy model ($K_{f_1} = 1;\ K_{f_2} = 0.05; k_i \neq 0$).

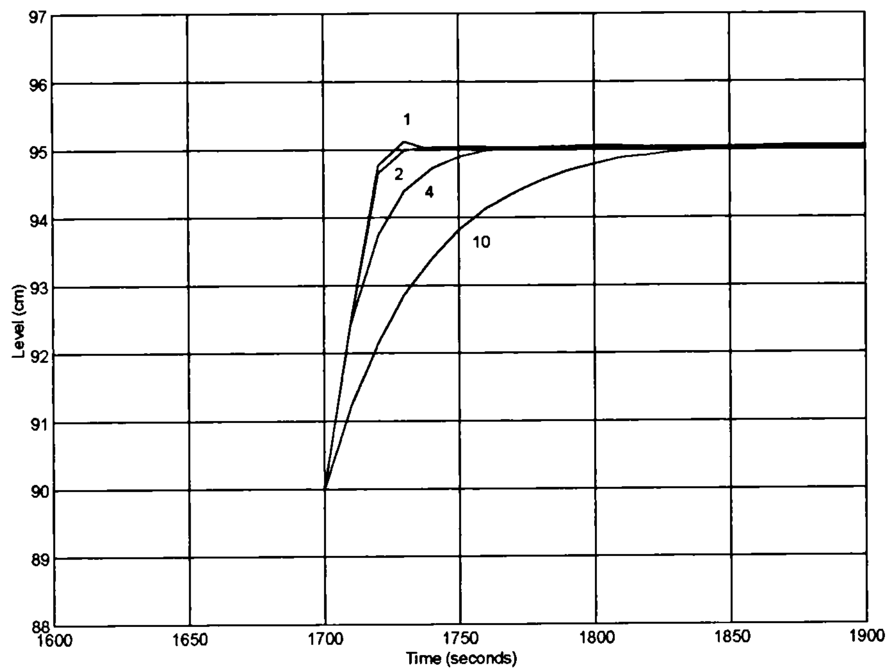


Figure 7.11: Effect of number of steps in prediction horizon on process output response to setpoint changes between 90 and 95 cm. when using proposed controller with linear model ($K_{f_1} = 1$; $K_{f_2} = 0.05; k_i \neq 0$).

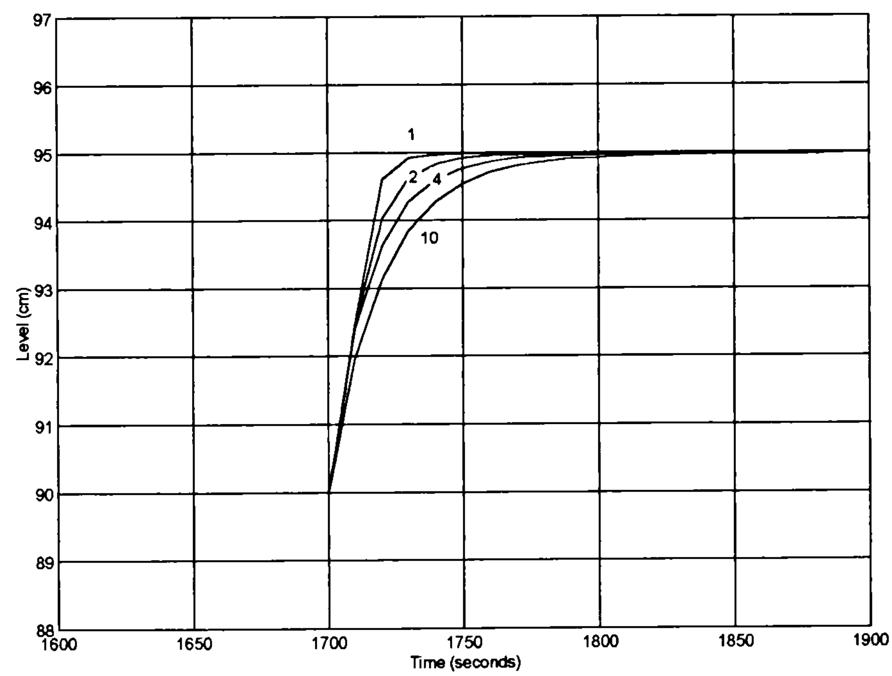


Figure 7.12: Effect of number of steps in prediction horizon on process output response to setpoint changes between 90 and 95 cm. when using proposed controller with 2-partition fuzzy model ($K_{f_1} = 1$; $K_{f_2} = 0.05; k_i \neq 0$).

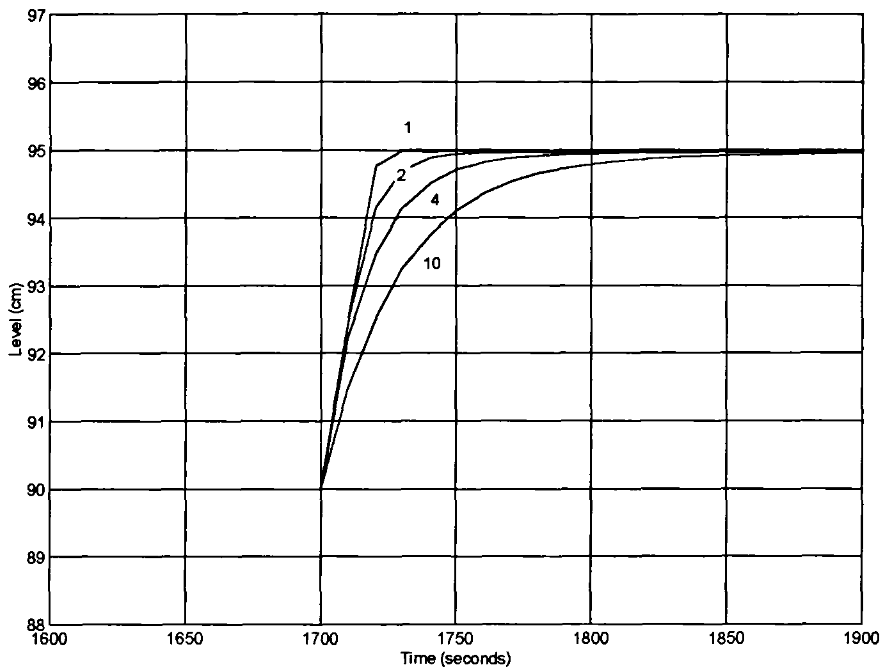


Figure 7.13: Effect of number of steps in prediction horizon on process output response to setpoint changes between 90 and 95 cm. when using proposed controller with 3-partition fuzzy model ($K_{f_1} = 1$; $K_{f_2} = 0.05; k_i \neq 0$).

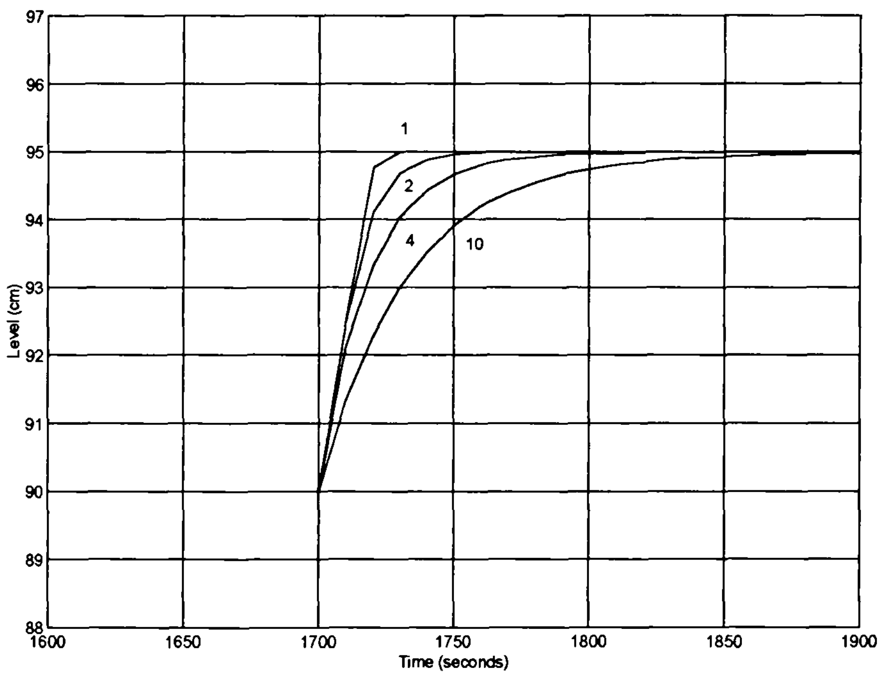


Figure 7.14: Effect of number of steps in prediction horizon on process output response to setpoint changes between 90 and 95 cm. when using proposed controller with 5-partition fuzzy model ($K_{f_1} = 1$; $K_{f_2} = 0.05; k_i \neq 0$).

7.5 Application to Control of CSTR

The gain of Filter 1 (K_{f_1}) has been standardised to 1 so that the effect of the prediction horizon on the performance of the controller could be studied free of the influence of this parameter. The control horizon has been standardised to 1-step; and the value of the constant term, k_i used by the process models has been assumed to be zero, since studies in Chapter 4 and 5 have showed that this is sufficient. To examine the effect of noise on the performance of the controller, uniformly distributed noise in the range between -20 and +20 moles.m^{-3} was added to the output from the simulation before being used by the controller.

Analysis of the IAE values in Tables 7.3 and 7.4 and Figures 7.15 to 7.20 will reveal that the controller using the 5-partition model provides the best performance to both +150 and -150 moles$\cdot$m^{-3} setpoint changes. It will also be observed that the controller using the 3-partition model performs better than controller using the 2-partition model which in turn performs better than the controller using the linear model. Figure 7.21 shows the process output response when this controller is used.

As in the case of the liquid level controller, increasing the prediction horizon leads to a less oscillatory and more sluggish response. The optimum performance under noisy conditions is achieved in this case with a prediction horizon of about 5-steps.

It will also be observed that the controller using the 5-partition fuzzy model performs significantly better than other controllers when the prediction horizon is small, but the difference in the performance of the controllers gets less when the length of the prediction horizon is greater. In fact, using the 3-partition model leads to only marginally worse performance if the prediction horizon is more than 1-step. This observation leads to two important conclusions. Firstly, the use of a more accurate model, achieved by increasing the number of fuzzy partitions, reduces the prediction horizon needed for optimal performance. The shorter rise times (i.e., quicker response to setpoint changes) with smaller prediction horizons generates lower IAE values. Secondly, the improvement in controller performance achieved by increasing the number of fuzzy partitions beyond a certain amount, 3 in the case of the CSTR model, is quite marginal. Similar observations were made when the controller was used in the level control application in Section 7.4. These observations reflect the relationship between the number of fuzzy partitions and modelling accuracy in Tables 4.2 and 4.5.

Tables 7.3 and 7.4 also reveal that the performance of the controller using a single linear model is generally very poor, reflecting its poor representation accuracy of nonlinear systems over a wide range.

Increasing the number of steps in the prediction horizon also introduces greater sluggishness in the response to load changes as observed from Figures 7.22 and 7.23.

Prediction horizon	Linear model		2-partition fuzzy model		3-partition fuzzy model		5-partition fuzzy model	
	IAE without noise	IAE with noise+	IAE without noise	IAE with noise+	IAE without noise	IAE with noise+	IAE without noise	IAE with noise+
1	313.1	265.1	232.3	237.8	81.9	165.1	44.9	154.6
2	381.8	299.4	150.1	195.3	42.3	123.1	42.4	112.2
3	355.8	273.3	52.9	158.2	42.4	106.5	42.4	99.6
5	347.7	366.2	55.5	127.8	47.0	95.8	45.1	90.5
6	-	404.0	-	122.6	-	95.0	-	90.9
10	386.3	545.9	73.9	126.5	64.6	106.5	62.3	102.8

\+ Uniformly distributed noise in the range -20 and +20 was added to the output from the simulation before being used by the controllers.

++ Gain of Filter 1 = 1; gain of Filter 2 = 0.1; IAE was derived over a period of 10 hours starting from 10 hours after the start of a simulation.

Table 7.3: Effect of the prediction horizon on the performance of the long-range predictive controller to +150 moles $\cdot$ m^{-3} setpoint change.

Prediction horizon	Linear model		2-partition fuzzy model		3-partition fuzzy model		5-partition fuzzy model	
	IAE without noise	IAE with noise+	IAE without noise	IAE with noise+	IAE without noise	IAE with noise+	IAE without noise	IAE with noise+
1	207.4	230.6	180.2	177.1	167.1	165.7	46.0	158.7
2	222.4	287.6	56.7	152.3	44.6	145.3	43.6	117.1
3	238.9	310.7	48.0	137.4	45.4	124.1	44.6	104.9
5	274.6	346.4	56.3	117.7	50.7	106.3	49.7	98.3
6	-	360.9	-	115.0	-	104.3	-	99.2
10	330.1	426.9	84.2	123.6	78.0	116.8	76.2	118.1

\+ Uniformly distributed noise in the range -20 and +20 was added to the output from the simulation before being used by the controllers.

++ Gain of Filter 1 = 1; gain of Filter 2 = 0.1; IAE was derived over a period of 10 hours starting from 10 hours after the start of a simulation.

Table 7.4: Effect of the prediction horizon on the performance of the long-range predictive controller to -150 moles $\cdot$ m^{-3} setpoint change.

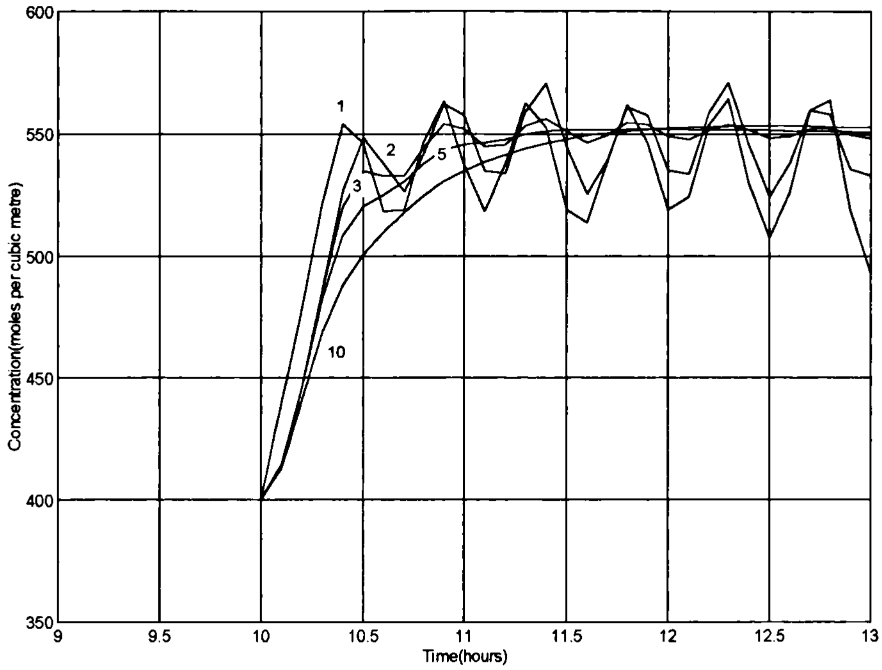


Figure 7.15: Effect of number of steps in prediction horizon on CSTR output response to +150 moles.m^{-3} setpoint change when using proposed controller with 2-partition fuzzy model ($K_{f_1} = 1; K_{f_2} = 0.1;\ k_i = 0$).

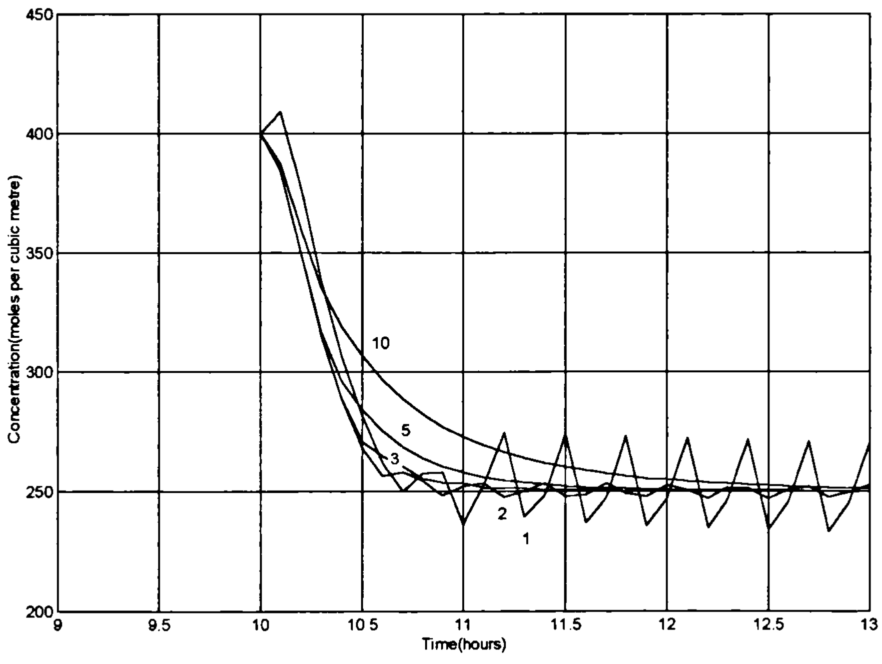


Figure 7.16: Effect of number of steps in prediction horizon on CSTR output response to -150 moles.m^{-3} setpoint change when using proposed controller with 2-partition fuzzy model ($K_{f_1} = 1; K_{f_2} = 0.1;\ k_i = 0$).

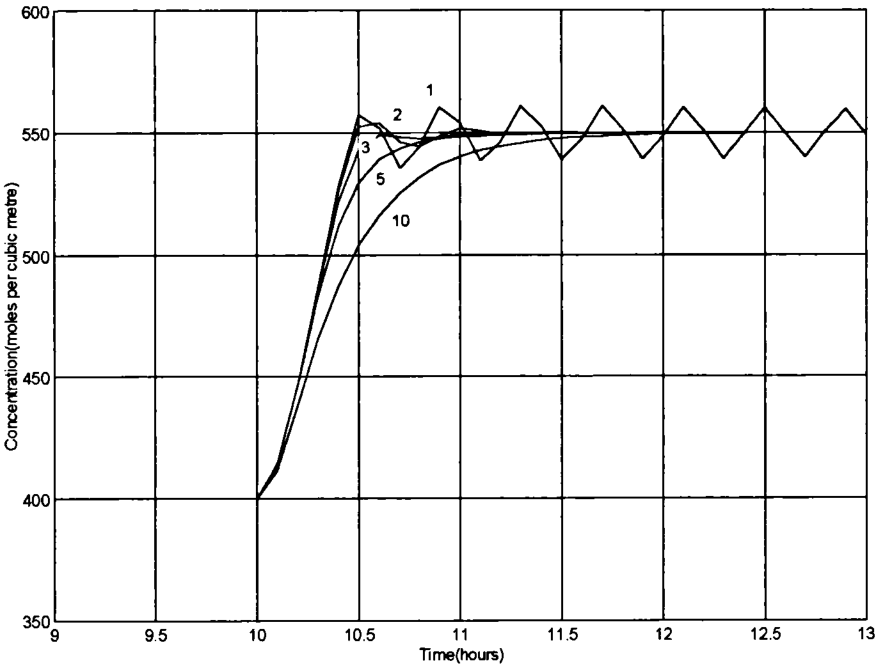


Figure 7.17: Effect of number of steps in prediction horizon on CSTR output response to +150 moles.m^{-3} setpoint change when using proposed controller with 3-partition fuzzy model ($K_{f_1} = 1; K_{f_2} = 0.1;\ k_i = 0$).

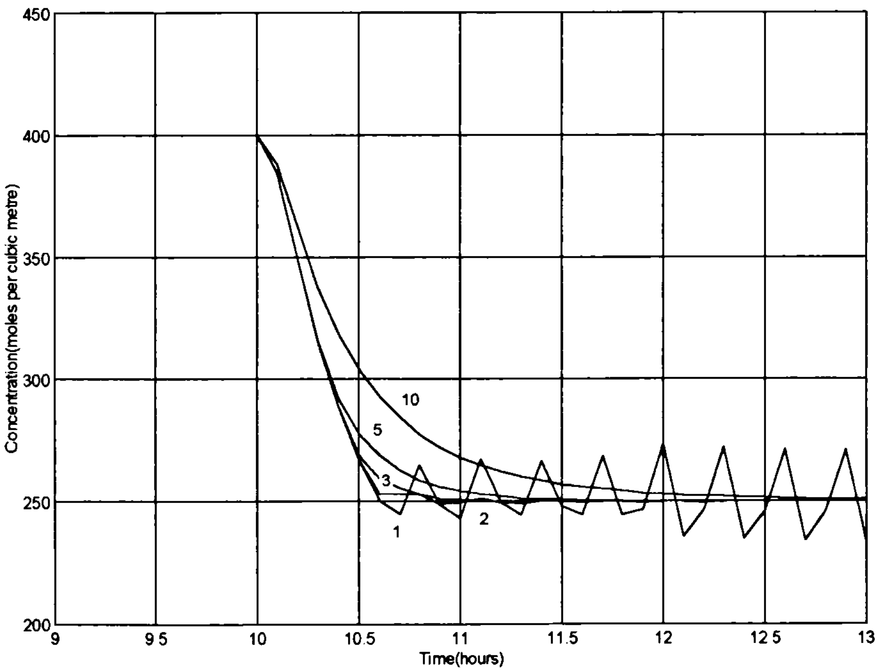


Figure 7.18: Effect of number of steps in prediction horizon on CSTR output response to -150 moles.m^{-3} setpoint change when using proposed controller with 3-partition fuzzy model ($K_{f_1} = 1; K_{f_2} = 0.1;\ k_i = 0$).

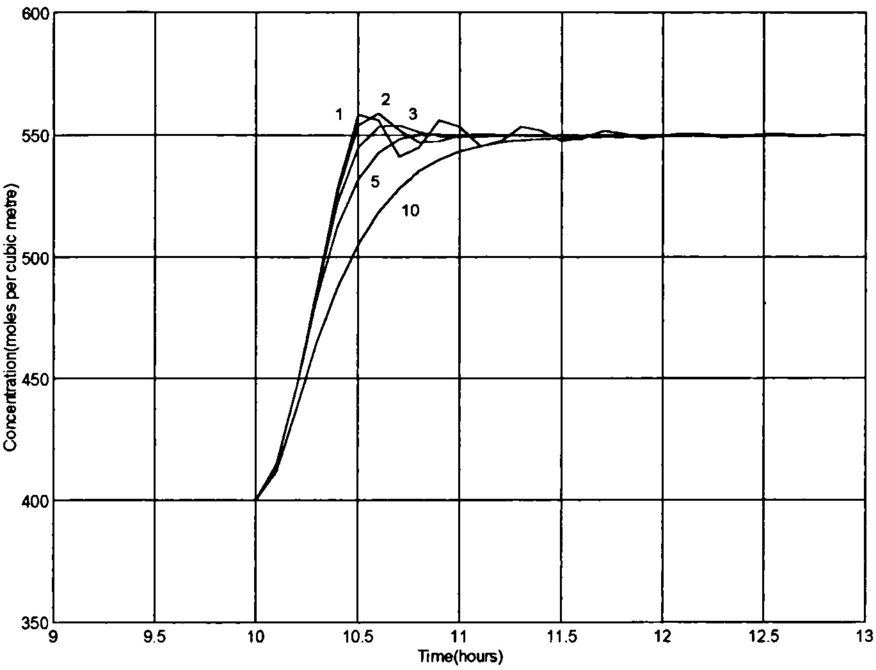


Figure 7.19: Effect of number of steps in prediction horizon on CSTR output response to +150 moles.m^{-3} setpoint change when using proposed controller with 5-partition fuzzy model ($K_{f_1} = 1; K_{f_2} = 0.1;\ k_i = 0$).

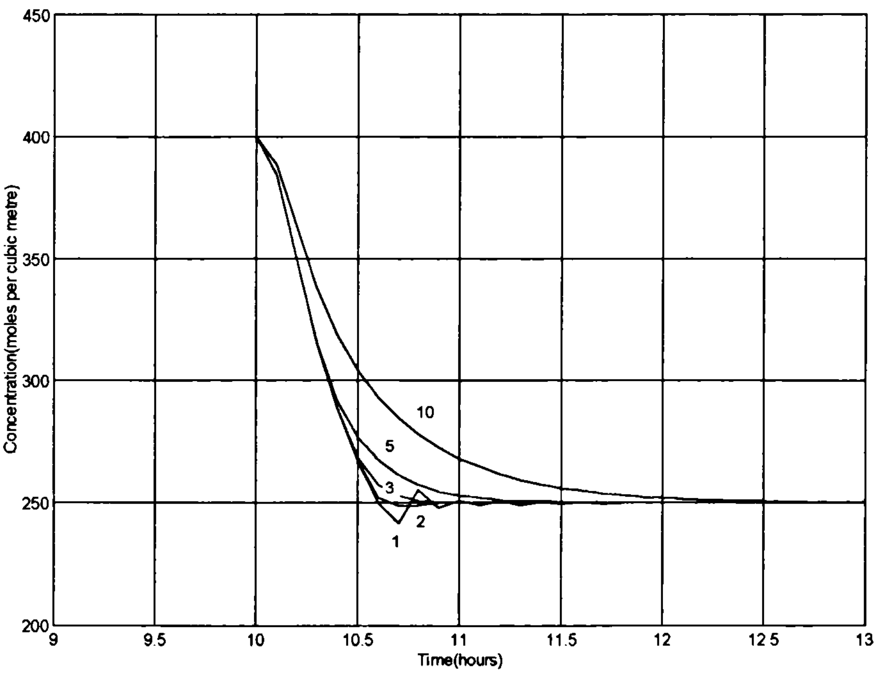


Figure 7.20: Effect of number of steps in prediction horizon on CSTR output response to -150 moles.m^{-3} setpoint change when using proposed controller with 5-partition fuzzy model ($K_{f_1} = 1; K_{f_2} = 0.1;\ k_i = 0$).

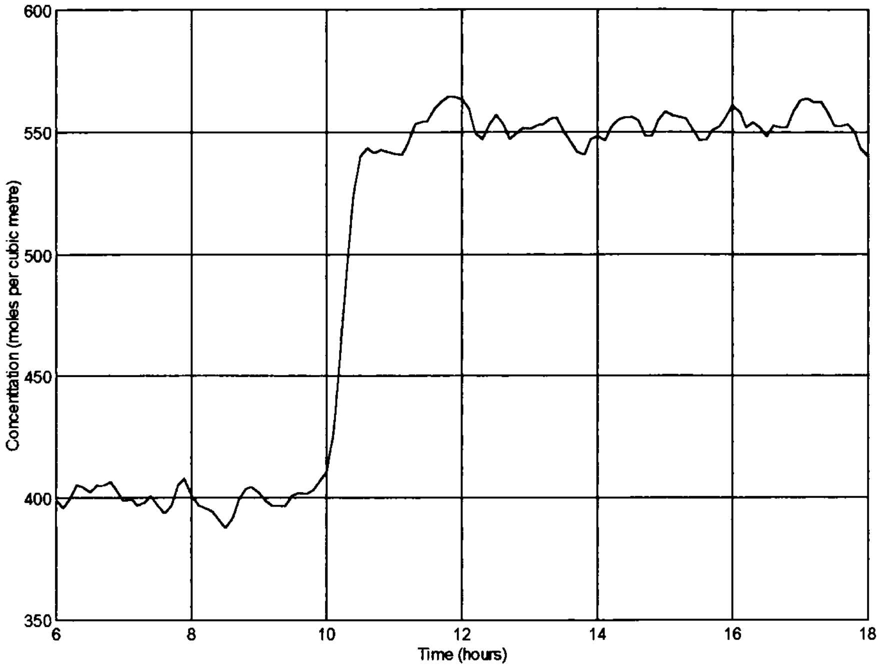


Figure 7.21: Process output response to +150 moles.m^{-3} setpoint change when using 5-steps ahead predictive controller with 5-partition fuzzy model in the presence of noise ($K_{f_1} = 1; K_{f_2} = 0.1;\ k_i = 0$).

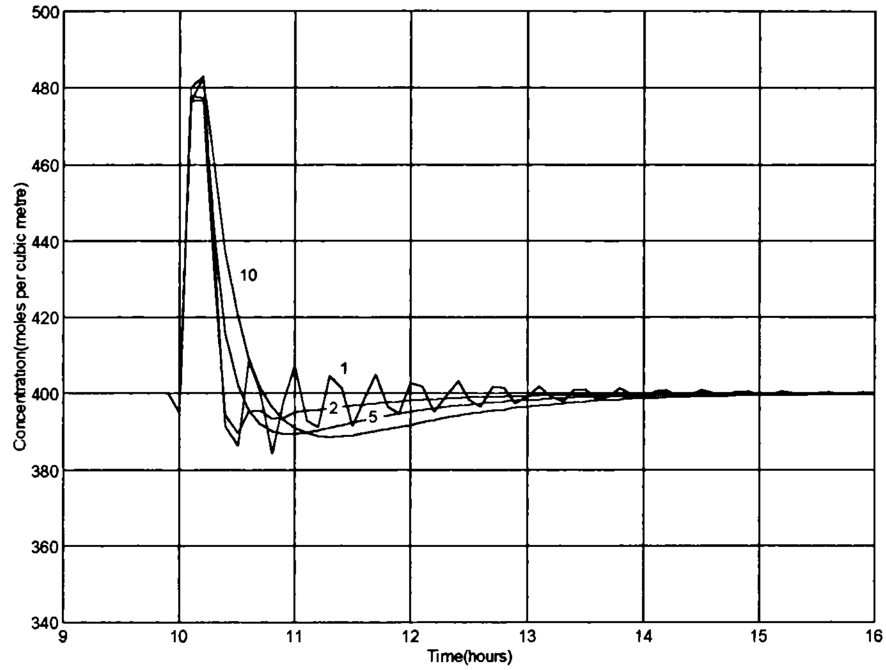


Figure 7.22: Effect of number of steps in prediction horizon on CSTR output response to +20 percent change in feed flowrate disturbance when using proposed controller with 5-partition fuzzy model ($K_{f_1} = 1; K_{f_2} = 0.1;\ k_i = 0$).

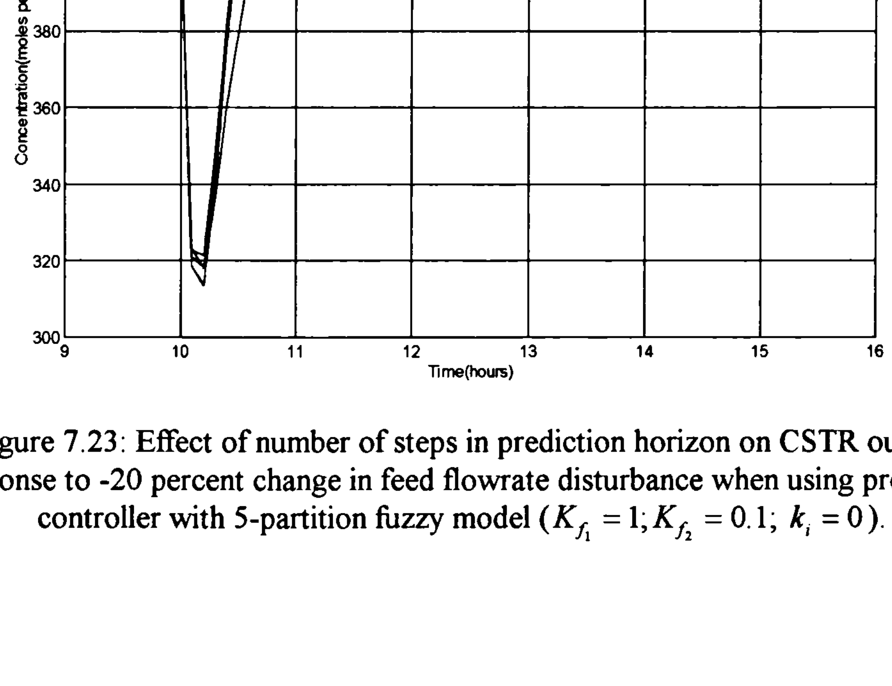


Figure 7.23: Effect of number of steps in prediction horizon on CSTR output response to -20 percent change in feed flowrate disturbance when using proposed controller with 5-partition fuzzy model ($K_{f_1} = 1; K_{f_2} = 0.1;\ k_i = 0$).

7.6 Numerical Approach versus Analytical Approach

Comparison of Figures 6.5 to 6.8 with Figures 7.7, 7.10, 7.11 and 7.14 shows that the process output responses using both approaches are almost identical. This is despite the fact that the computation time required by the analytical approach is significantly less than the numerical approach. Table 7.5 gives an indication of the approximate time required to run a typical simulation of the tank liquid level control problem on a 66 MHz 80486 microcomputer system when the controllers use the 5-partition fuzzy model.

Prediction horizon (number of steps)	Time required to complete simulation (seconds)	
	Numerically derived controller output	Analytically derived controller output
Tank level simulation (no control)	10	10
1	110	25
2	133	26
5	207	28
10	317	31
20	558	38

Table 7.5: Comparison of the times required to complete typical simulation runs over 2500 seconds on a 66 Mhz 80486 microcomputer.

7.7 Robustness Analysis

In order for a control system to function properly, it should not be unduly sensitive to small changes in the process or to inaccuracies in the process model if a model is used to design the control system. A control system that satisfies this requirement is said to be robust or to exhibit a satisfactory degree of robustness.

Although changes in real processes can be attributed to a multitude of causes, we shall limit our analysis of the robustness of the controller used for the CSTR system to changes in just one variable. In the development of the mathematical model, an assumption was made that the volume of the reactor (V) remained constant at 1.36 m^3. Accumulations of solids in the reactor, changes in level, etc., can cause variations in the volume. In this study, we have therefore focused on the effect of variations in the volume of the reactor on the performance of the controller. The value of V in the mathematical model of the CSTR system was varied from 1.088 to 1.632 m^3 and the performance of the controller evaluated using the 5-partition fuzzy model that was generated when V was 1.36 m^3.

Table 7.6 shows that sudden and significant performance deterioration of the one-step ahead predictive controller can take place especially when V is below the normal value. A very small change in V can lead to poor controller performance in the case of feed flowrate changes.

It will be observed from Tables 7.7 to 7.9 that the robustness of the controller improves when the prediction horizon is greater than 1 and is able to cope with the significant changes (more than what is usually encountered in practice) in the volume of the reactor investigated in this study. It will also be observed that there is little change in the IAE to feed concentration changes (Table 7.7) and a gradual change in the IAE to feed flowrate changes (Table 7.8) and setpoint changes (Table 7.9) if the prediction horizon is greater than 1-step.

Volume of reactor (V)	Percentage of normal volume of reactor	IAE to +5 percent concentration change+	IAE to +20 percent feed flowrate change+	IAE to +150 moles.m^{-3} setpoint change+
1.088	80	150.81	190.67	133.63
1.156	85	89.03	160.43	84.99
1.224	90	27.84	122.40	47.38
1.292	95	12.56	99.79	44.40
1.360	100	11.43	31.93	44.87
1.428	105	11.03	26.23	48.05
1.496	110	10.92	24.95	50.39
1.564	115	10.85	24.46	54.39
1.632	120	10.92	24.38	58.03

\+ IAE was derived over a period of 10 hours starting from 10 hours after the start of a simulation run.

++ Gain of Filter 1 = 1; gain of Filter 2 = 0.1

Table 7.6: Effect of changes in the volume of reactor on the performance of the 1-step ahead predictive controller using 5-partition fuzzy model.

Prediction horizon	IAE to flowrate change+						
	V = 1.088 (80 percent)	V = 1.224 (90 percent)	V = 1.292 (95 percent)	V = 1.360 (100 percent)	V = 1.428 (105 percent)	V = 1.496 (110 percent)	V = 1.632 (120 percent)
1	192.23	122.35	99.79	31.94	26.23	24.95	24.39
2	29.67	27.08	27.16	27.64	28.29	29.00	30.51
3	28.02	28.70	29.70	30.81	31.93	33.02	35.28
4	28.93	30.86	32.31	33.84	35.56	37.47	41.04
5	30.32	32.70	34.91	37.27	39.43	41.52	45.62

+ Gain of Filter 1 = 1; gain of Filter 2 = 0.1; V: Volume of reactor; IAE was derived over a period of 10 hours starting from 10 hours after the start of a simulation run.

Table 7.8:Effect of change in the volume of reactor on the performance of the long-range predictive controller using 5-partition fuzzy model to +20 percent feed flowrate change.

Prediction horizon	IAE to concentration change+						
	V = 1.088 (80 percent)	V = 1.224 (90 percent)	V = 1.292 (95 percent)	V = 1.360 (100 percent)	V = 1.428 (105 percent)	V = 1.496 (110 percent)	V = 1.632 (120 percent)
1	150.85	27.84	12.56	11.43	11.03	10.92	10.93
2	17.36	17.37	17.49	17.65	17.81	17.93	18.14
3	23.55	23.71	23.77	23.88	24.04	24.18	24.58
4	29.12	29.24	29.30	29.37	29.57	29.80	30.22
5	34.04	34.15	34.22	34.27	34.56	34.78	35.20

+ Gain of Filter 1 = 1; gain of Filter 2 = 0.1; V: Volume of reactor; IAE was derived over a period of 10 hours starting from 10 hours after the start of a simulation run.

Table 7.7:Effect of change in the volume of reactor on the performance of the long-range predictive controller using 5-partition fuzzy model to +5 percent feed concentration change.

Prediction horizon	IAE to setpoint change+						
	V = 1.088 (80 percent)	V = 1.224 (90 percent)	V = 1.292 (95 percent)	V = 1.360 (100 percent)	V = 1.428 (105 percent)	V = 1.496 (110 percent)	V = 1.632 (120 percent)
1	134.26	47.38	44.40	44.88	48.06	50.39	58.03
2	32.02	36.05	39.11	42.37	45.30	49.51	57.91
3	32.01	36.47	39.07	42.42	46.02	49.62	58.24
4	32.85	38.03	40.50	43.18	46.63	50.55	58.97
5	36.02	39.81	42.59	45.15	48.09	51.45	59.68

+ Gain of Filter 1 = 1; Gain of Filter 2 = 0.1; V: Volume of reactor; IAE was derived over a period of 10 hours starting from 10 hours after the start of a simulation run.

Table 7.9: Effect of change in the volume of reactor on the performance of the long-range predictive controller using 5-partition fuzzy model to +150 moles.m^{-3} setpoint change.

7.8 Conclusions

This chapter has focussed on the development of an analytical approach for determining the long-range predictive controller output. It has been shown to result in a significant reduction in computational requirements over the numerical approach presented in Chapter 6. The use of multi-step prediction horizon has been shown to lead to a more robust control strategy. The length of the prediction horizon needed for optimum performance is comparable to that required in GPC. In the two examples, increasing the number of fuzzy partitions generally led to better modelling accuracy and better controller performance even though the improvement achieved by increasing the number of fuzzy partitions beyond 3 was marginal.

Even though the controller has been shown to perform well when used to control nonlinear processes, the emphasis in this chapter has been on time-invariant single-input single-output (SISO) processes. More and more, control systems are expected to be able to address the requirements of time-varying proceses, multivariable processes and processes with dead times. The ability to explicitly handle constraints on process inputs and outputs is also considered important. Appendices A and B show how the control algorithm presented in this chapter can be extended to enable adaptive control and deadtime compensation, while the remaining chapters of this thesis focus on the control of multivariable processes and explicit handling of constraints. It is shown that extensions to the basic algorithm presented in this chapter can be quite easily handled.

Chapter 8

CONTROL OF MULTI-INPUT SINGLE-OUTPUT SYSTEMS

8.1 Introduction

All previous studies were based on single-input single-output process models using the manipulated variable as the only input. In this chapter, we will attempt to extend the process model to include one or more disturbance inputs. By doing this, it may be possible to achieve a feedforward control effect in our controller to these disturbances. Such a prospect makes the use of multi-input single-output (MISO) process model based controllers appear attractive (Tham *et al.*, 1989; Montague *et al.*, 1991; Vagi *et al.*, 1991; Tham *et al.*, 1991a; Montague *et al.*, 1992; Postlethwaite, 1994). Modifications to the control algorithm proposed in Chapter 7 to allow use with MISO process models is first examined. A comparative evaluation of the performance of the proposed controller using SISO and MISO models is then attempted. The work in this chapter also constitutes a first step in the development of a controller for MIMO systems which will be presented in Chapters 9 and 10.

All derivations in this section are based on 2-inputs 1-output process models. Extension to process models consisting of more inputs can, however, be quite easily achieved. It has also been assumed that the disturbance input remains constant over the entire prediction horizon.

8.2 Model Predictions Using MISO Models

8.2.1 First-Order MISO System

A first-order system with one disturbance input can be described by the following ARX model structure:

$$y_p(t+1) = a_1 y(t) + b_1 u(t) + c_1 v(t) + k + err(t) \tag{8.1}$$

where:

$y(t)$ = process output at time t.

$y_p(t+1)$ = one-step ahead model prediction at time t.

$u(t)$ = manipulated variable at time t.

$v(t)$ = disturbance variable at time t.

$err(t)$ = estimate of the modelling error at time t.

Model predictions over a prediction horizon of n-steps based on a control horizon of 1-step is given by the equation:

$$Y(t) = \mathbf{P}X(t) + \mathbf{Q}U(t) + \mathbf{R}[k + err(t)] \tag{8.2}$$

where,

$$Y(t) = \begin{bmatrix} y_p(t+1) & \cdots & y_p(t+n) \end{bmatrix}^T \tag{8.3}$$

$$X(t) = \begin{bmatrix} y(t) & v(t) \end{bmatrix}^T \tag{8.4}$$

$$U(t) = \begin{bmatrix} u(t) \end{bmatrix} \tag{8.5}$$

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} \\ \vdots & \vdots \\ p_{n1} & p_{n2} \end{bmatrix} \tag{8.6}$$

$$\mathbf{Q} = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}^T \tag{8.7}$$

$$\mathbf{R} = \begin{bmatrix} r_1 & \cdots & r_n \end{bmatrix}^T \tag{8.8}$$

The coefficients of **P**, **Q** and **R** are given by:

$$p_{11} = a_1$$
$$p_{i1} = a_1 \cdot p_{(i-1)1}, \text{ for } i = 2,\ldots,n \tag{8.9}$$

$$p_{12} = c_1$$
$$p_{i2} = a_1 \cdot p_{(i-1)2} + c_1, \text{ for } i = 2,\ldots,n \tag{8.10}$$

$$q_1 = b_1$$
$$q_i = a_1 \cdot q_{i-1} + b_1, \text{ for } i = 2,\ldots,n \tag{8.11}$$

$$r_1 = 1$$
$$r_i = a_1 \cdot r_{i-1} + 1, \text{ for } i = 2,\ldots,n \tag{8.12}$$

8.2.2 Second-Order MISO System

A second-order system with 1 disturbance input can be described by the following ARX model structure:

$$y_p(t+1) = a_1 y(t) + a_2 y(t-1) + b_1 u(t) + b_2 u(t-1) + c_1 v(t) + c_2 v(t-1) + k + err(t) \tag{8.13}$$

Model predictions over a prediction horizon of n-steps based on a control horizon of 1-step is given by the equation:

$$Y(t) = \mathbf{P}X(t) + \mathbf{Q}U(t) + \mathbf{R}[k + err(t)] \tag{8.14}$$

where,

$$Y(t) = \begin{bmatrix} y_p(t+1) & \cdots & y_p(t+n) \end{bmatrix}^T \tag{8.15}$$

$$X(t) = \begin{bmatrix} y(t) & y(t-1) & u(t-1) & v(t) & v(t-1) \end{bmatrix}^T \tag{8.16}$$

$$U(t) = \begin{bmatrix} u(t) \end{bmatrix} \tag{8.17}$$

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{15} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{n1} & p_{n2} & p_{n3} & p_{n4} & p_{n5} \end{bmatrix} \tag{8.18}$$

$$\mathbf{Q} = [q_1 \ \cdots \ q_n]^T \tag{8.19}$$

$$\mathbf{R} = \begin{bmatrix} r_1 & \cdots & r_n \end{bmatrix}^T \tag{8.20}$$

The coefficients of **P**, **Q** and **R** are given by:

$$\begin{aligned} &p_{11} = a_1 && p_{12} = a_2 && p_{13} = b_2 \\ &p_{21} = a_1 \cdot p_{11} + a_2 && p_{22} = a_1 \cdot p_{12} && p_{23} = a_1 \cdot p_{13} \\ &p_{i1} = a_1 p_{(i-1)1} + a_2 p_{(i-2)1} && p_{i2} = a_1 p_{(i-1)2} + a_2 p_{(i-2)2} && p_{i3} = a_1 p_{(i-1)3} + a_2 p_{(i-2)3}, \\ &&&&& \text{for } i = 3,\ldots,n \end{aligned} \tag{8.21}$$

$$\begin{aligned} &p_{14} = c_1 && p_{15} = c_2 \\ &p_{24} = a_1 \cdot p_{14} + c_1 + c_2 && p_{25} = a_1 \cdot p_{15} \\ &p_{i4} = a_1 p_{(i-1)4} + a_2 p_{(i-2)4} + c_1 + c_2 && p_{i5} = a_1 p_{(i-1)5} + a_2 p_{(i-2)5}, \text{ for } i = 3,\ldots,n \end{aligned} \tag{8.22}$$

$$\begin{aligned} &q_1 = b_1 \\ &q_2 = a_1 \cdot q_1 + b_1 + b_2 \\ &q_i = a_1 \cdot q_{i-1} + a_2 \cdot q_{i-2} + b_1 + b_2, \text{ for } i = 3,\ldots,n \end{aligned} \tag{8.23}$$

$$\begin{aligned} &r_1 = 1 \\ &r_2 = a_1 + 1 \\ &r_i = a_1 \cdot r_{i-1} + a_2 \cdot r_{i-2} + 1, \text{ for } i = 3,\ldots,n \end{aligned} \tag{8.24}$$

8.3 Controller Formulation

The optimal controller output is similarly derived as for SISO systems and can be shown to be given by the following equation:

$$U(t) = [\mathbf{Q}^T\mathbf{Q}]^{-1}\mathbf{Q}^T\left[W(t) - \mathbf{P}X(t) - \mathbf{R}(k + err(t))\right] \tag{8.25}$$

The performance of the above control scheme will now be examined using a simulation example.

8.4 Application to Control of CSTR

8.4.1 Identification of MISO Process Models

Figures 8.1 to 8.3 show the open loop response of the product concentration to step changes in some important disturbances. In every case, it is clear that the relationship between the two variables is nonlinear. In Figures 8.2 and 8.3, it will be noticed that the initial response is in a direction opposite to the final response (i.e., inverse response). Such behaviour is the nett result of two opposing effects. A second order model structure with respect to all inputs will be assumed. Since the open loop response between product concentration and feed flowrate will be poorly represented by second order system dynamics, we will only examine MISO models incorporating feed temperature and feed concentration as disturbance inputs.

For process model identification, 50 random step changes, each lasting 6 hours, were applied to each of the inputs to be used in the model. Only the 5-partition fuzzy model was evaluated and fuzzy partitioning of the input space for the MISO model was maintained the same as for the SISO model. Figures 8.4 and 8.5 show the input/output data sets used to identify the MISO process models which incorporate feed temperature and feed concentration, respectively.

The following second-order MISO model incorporating feed temperature was used:

$$L^i\text{: } IF\ C_A(t)\ is\ B^i\ THEN\ C_{Am}(t+1) = a_1^i C_A(t) + a_2^i C_A(t-1) + b_1^i Q(t) + b_2^i Q(t-1) + c_1^i T_i(t) + c_2^i T_i(t-1),\ \text{for } i = 1,\ldots,p. \tag{8.26}$$

where $C_A(t)$, $Q(t)$ and $T_i(t)$ are the concentration of A in the reactor, the rate of heat removal and the feed temperature at time t, respectively. As with the SISO case, the value of the constant term, k_i was assumed to be zero. The model parameters were:

$$\Phi_{temp} = \begin{bmatrix} 1.4677 & +0.2463 & 11.5018 & 20.2810 & -0.4476 & -0.0901 \\ 1.8916 & -0.1567 & 15.3678 & 8.2796 & -0.4188 & -0.2246 \\ 1.9908 & -0.2587 & 22.5238 & 7.9216 & -0.6424 & -0.2291 \\ 2.0558 & -0.3276 & 28.0754 & 10.7149 & -0.7976 & -0.3053 \\ 2.3317 & -0.6004 & 32.2693 & 5.4443 & -0.9643 & -0.4514 \end{bmatrix} \tag{8.27}$$

The MISO model incorporating feed concentration was assumed to take the following form:

$$L^i: \; IF\; C_A(t)\; is\; B^i\; THEN\; C_{A_m}(t+1) = a_1^i C_A(t) + a_2^i C_A(t-1) + b_1^i Q(t) + b_2^i Q(t-1) + c_1^i C_{A_i}(t) + c_2^i C_{A_i}(t-1), \text{ for } i = 1,\ldots,p. \tag{8.28}$$

where $C_A(t)$, $Q(t)$ and $C_{A_i}(t)$ are the concentration of A in the reactor, the rate of heat removal and the feed concentration at time t, respectively. The model parameters were:

$$\Phi_{conc} = \begin{bmatrix} 0.8476 & +0.4579 & 12.8899 & 23.3445 & 0.0319 & -0.0446 \\ 1.4514 & -0.1190 & 15.7681 & 9.0652 & 0.0202 & -0.0352 \\ 1.5815 & -0.2497 & 22.5863 & 8.2536 & 0.0111 & -0.0318 \\ 1.6557 & -0.3273 & 28.1708 & 10.5341 & 0.0097 & -0.0361 \\ 1.9618 & -0.6337 & 31.9225 & 3.7432 & 0.0043 & -0.0373 \end{bmatrix} \tag{8.29}$$

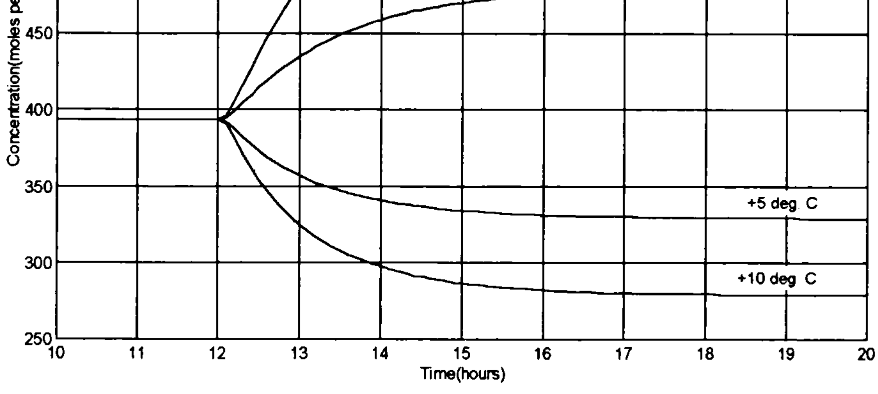


Figure 8.1: Open loop response of the concentration of A in the reactor to step changes in feed temperature.

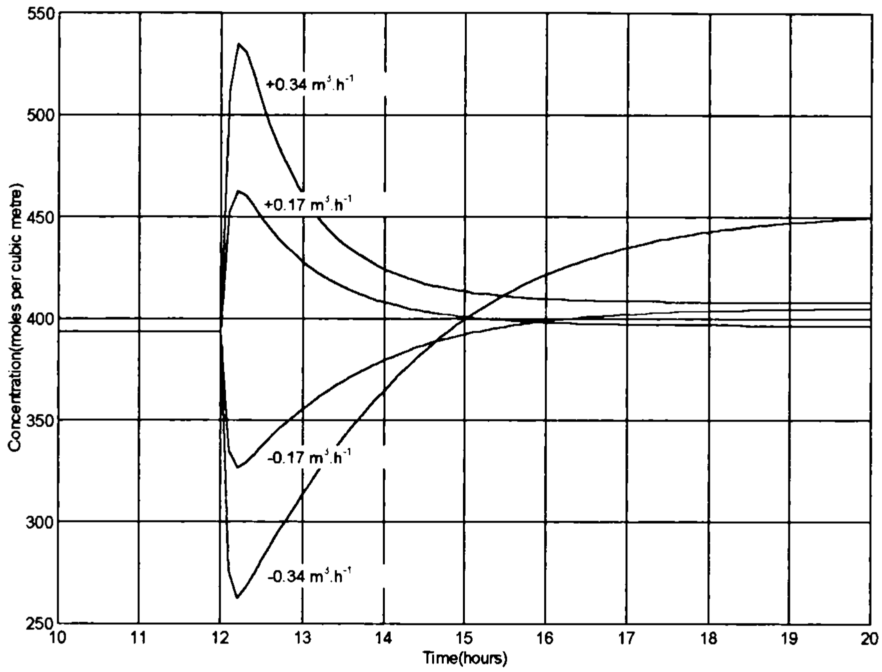


Figure 8.2: Open loop response of the concentration of A in the reactor to step changes in feed flowrate.

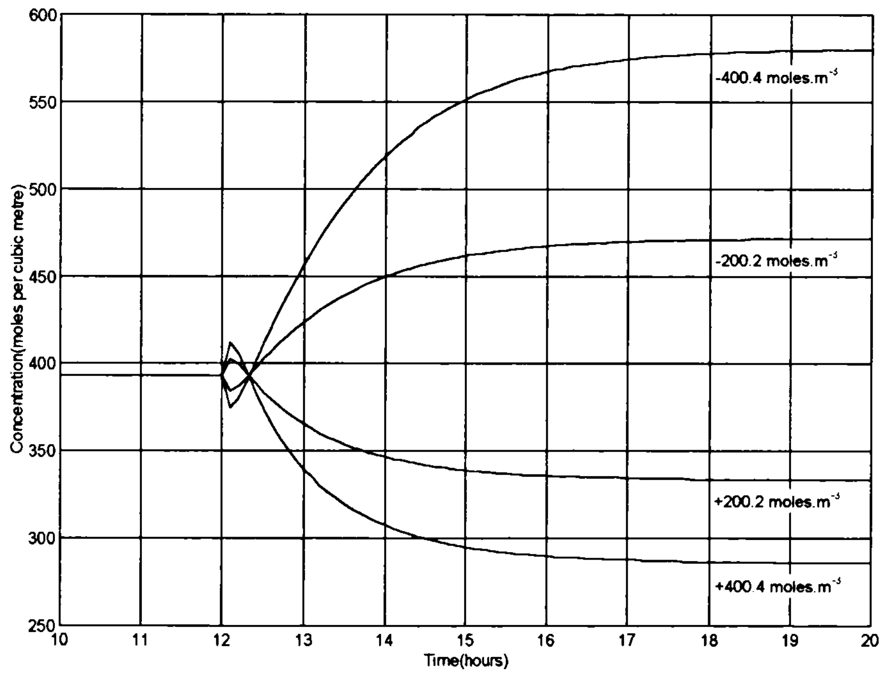


Figure 8.3: Open loop response of the concentration of A in the reactor to step changes in the concentration of A in the feed stream.

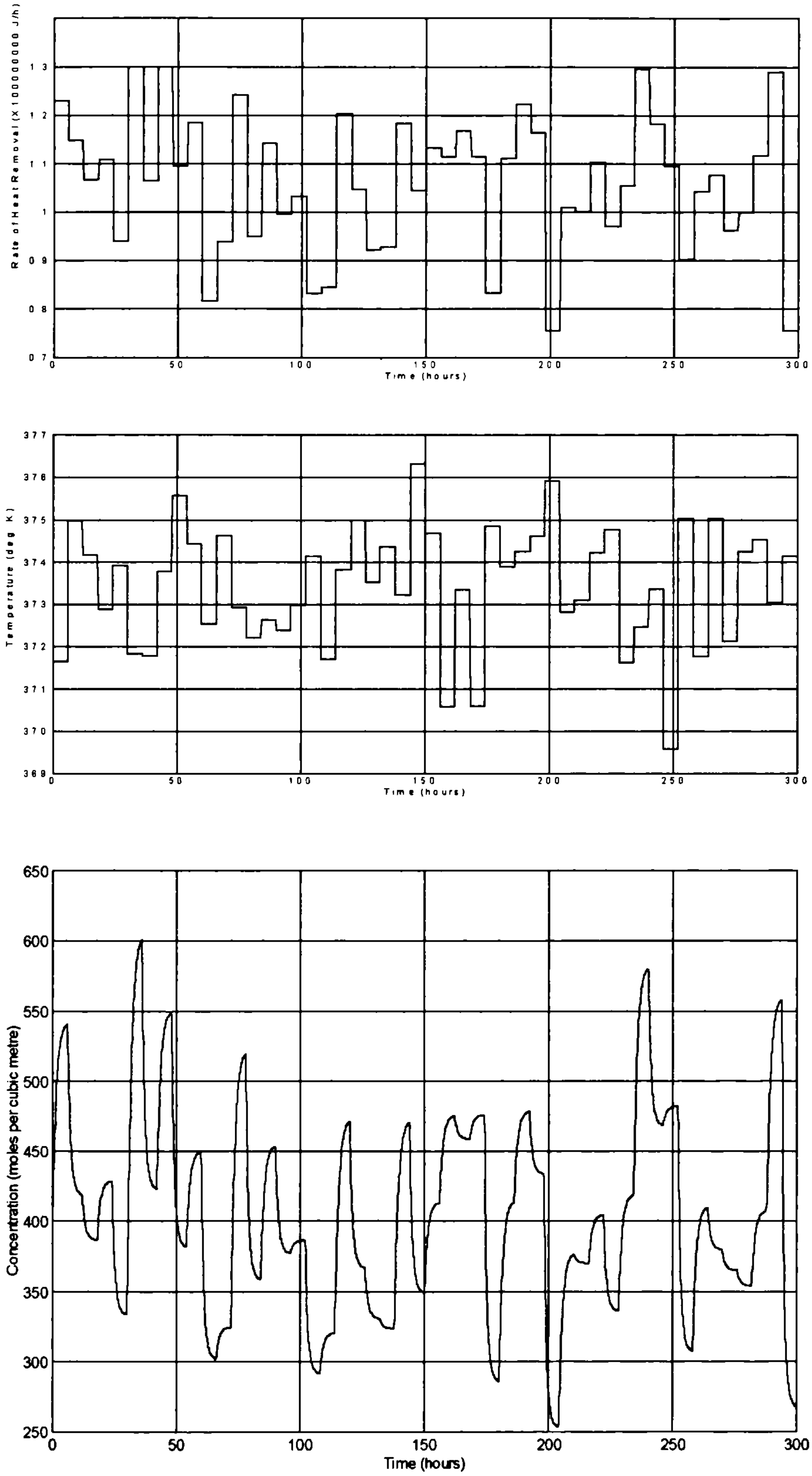


Figure 8.4: Input/output data used for identification of MISO process model incorporating feed temperature as a disturbance input.

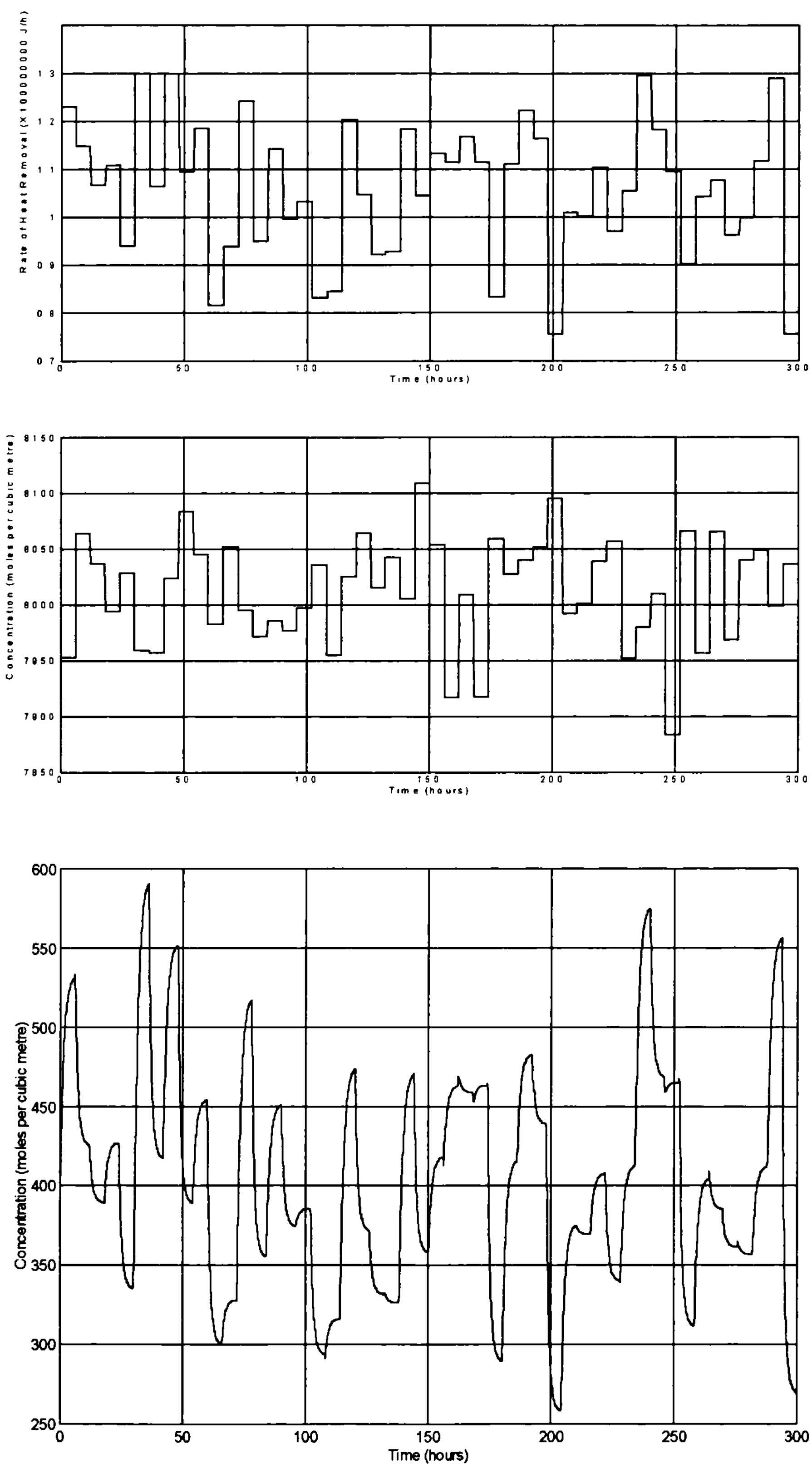


Figure 8.5: Input/output data used for identification of MISO process model incorporating feed concentration as a disturbance input.

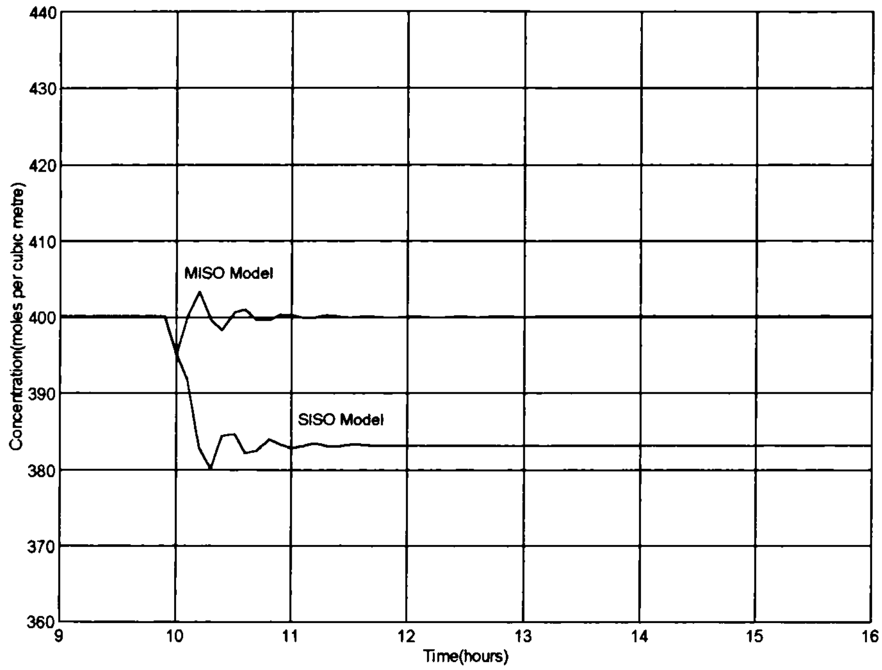


Figure 8.6: Comparison of the process output response to +20 deg. C feed temperature change when using 1-step ahead predictive controller with SISO model and MISO model incorporating feed temperature ($K_{f_1} = 1; K_{f_2} = 0$).

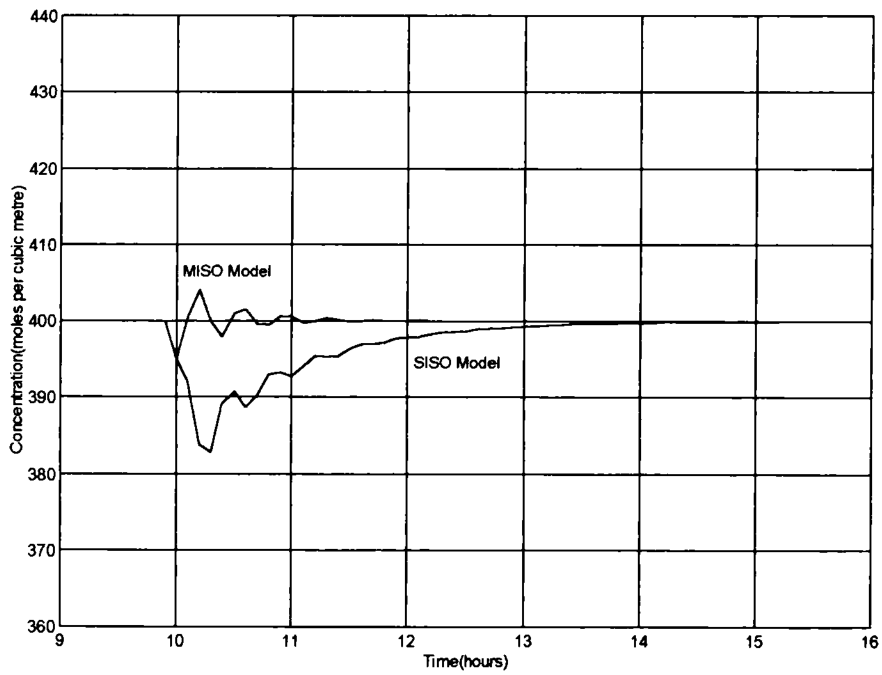


Figure 8.7: Comparison of the process output response to +20 deg. C feed temperature change when using 1-step ahead predictive controller with SISO model and MISO model incorporating feed temperature ($K_{f_1} = 1; K_{f_2} = 0.1$).

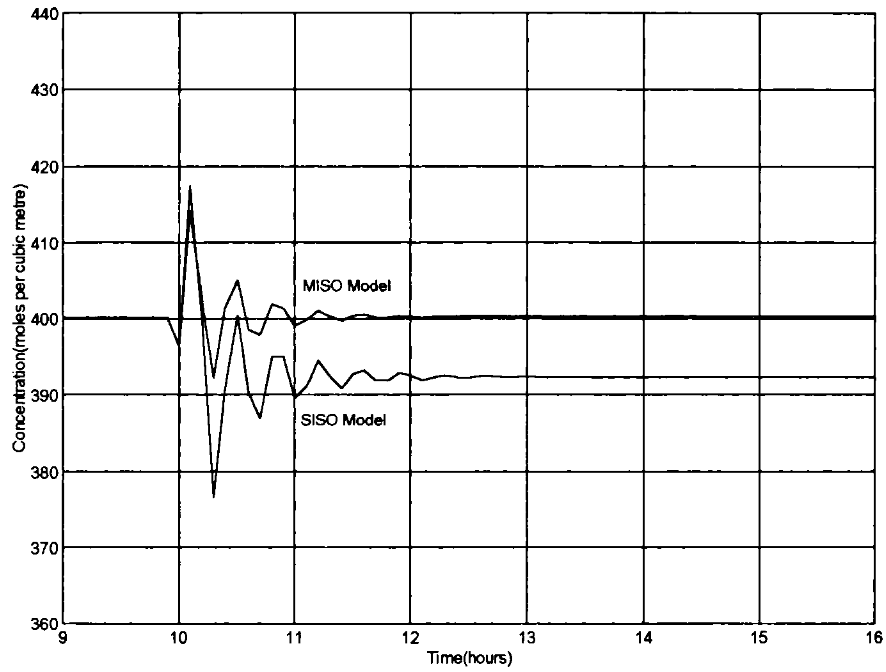


Figure 8.8: Comparison of the process output response to +5 percent feed concentration change when using 1-step ahead predictive controller with SISO model and MISO model incorporating feed concentration ($K_{f_1} = 1; K_{f_2} = 0$).

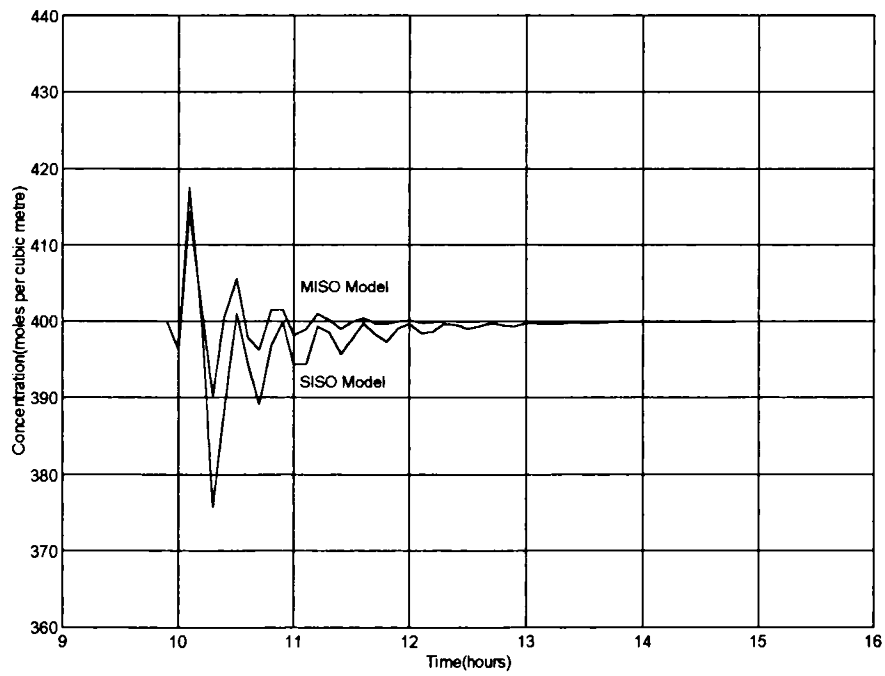


Figure 8.9: Comparison of the process output response to +5 percent feed concentration change when using 1-step ahead predictive controller with SISO model and MISO model incorporating feed concentration ($K_{f_1} = 1; K_{f_2} = 0.1$).

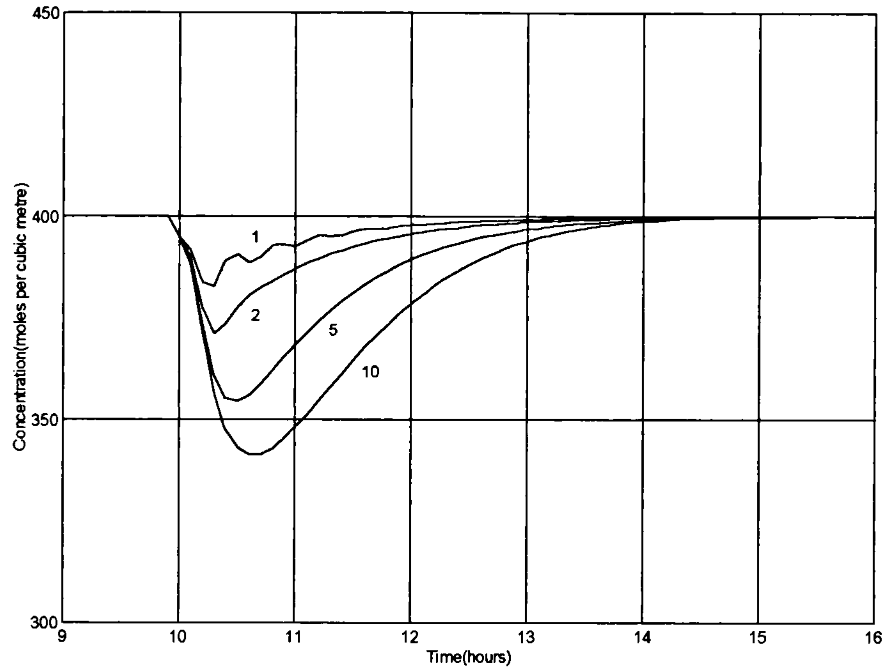


Figure 8.10: Effect of number of steps in prediction horizon on process output response to +20 deg. C feed temperature change when using controller with SISO model ($K_{f_1} = 1; K_{f_2} = 0.1$).

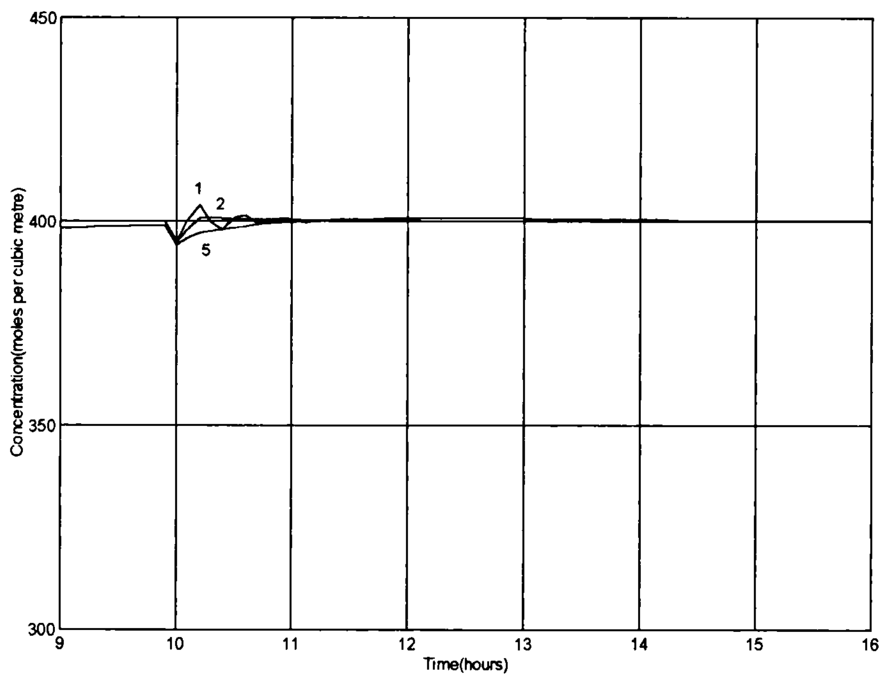


Figure 8.11: Effect of number of steps in prediction horizon on process output response to +20 deg. C feed temperature change when using controller with MISO model incorporating feed temperature ($K_{f_1} = 1; K_{f_2} = 0.1$).

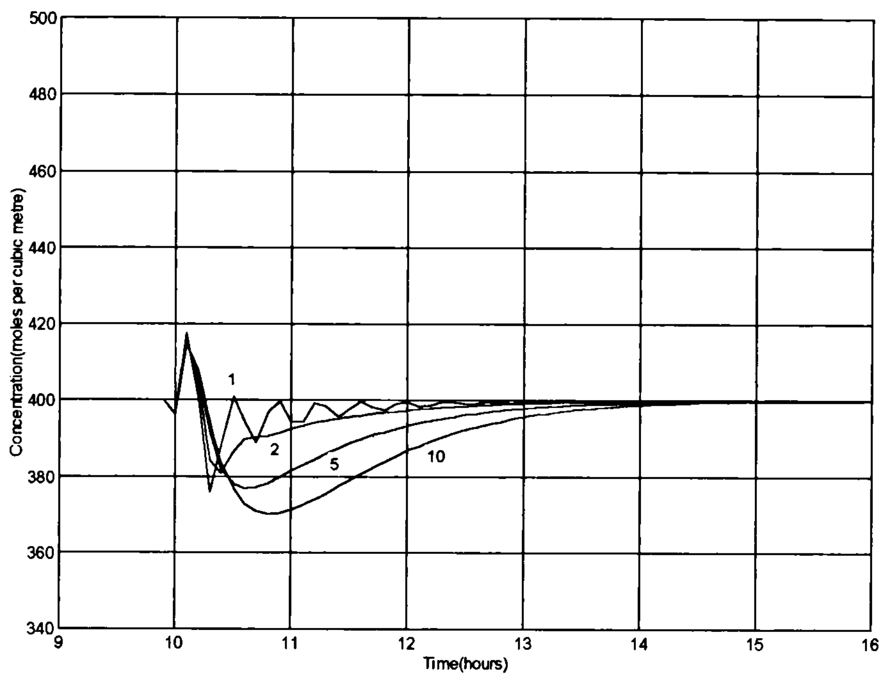


Figure 8.12: Effect of number of steps in prediction horizon on process output response to +5 percent feed concentration change when using controller with SISO model ($K_{f_1} = 1; K_{f_2} = 0.1$).

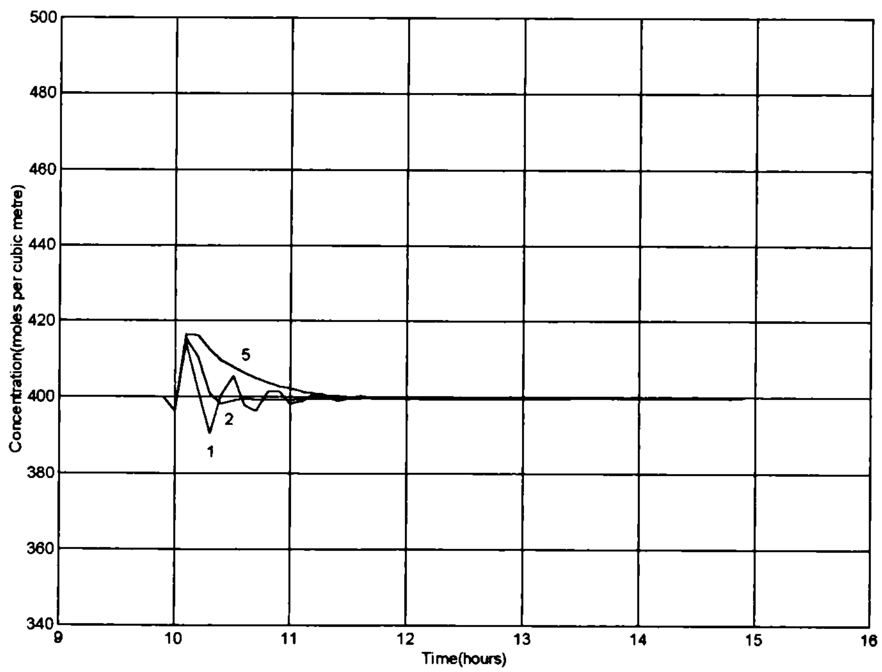


Figure 8.13: Effect of number of steps in prediction horizon on process output response to +5 percent feed concentration change when using controller with MISO model incorporating feed concentration ($K_{f_1} = 1; K_{f_2} = 0.1$).

8.4.2 MISO Process Model Based Controllers

The values of the gains of Filters 1 and 2 were set to 1 and 0.1, which earlier studies using the SISO model have shown to result in good one-step ahead predictive controller performance.

Figures 8.6 to 8.9 compare the output responses of the one-step ahead predictive controller using 2-input MISO models with the controller using the SISO model to step changes in the disturbance included in the model. The better performance with the MISO models is clearly obvious.

Figures 8.10 to 8.13 show the performance of the multi-step predictive controller to the disturbances included in the MISO model. The generally better performance from the controllers using MISO models is once again clearly obvious.

It was difficult to get good controller performance when more than 1 disturbance input was included in the MISO model. This may be due to the inverse response between product concentration and feed concentration (Figure 8.3).

8.5 Conclusions

The long-range predictive control algorithm presented in Chapter 7 has been extended in this chapter to cover MISO systems. By using the CSTR system as an example, it has also been shown that it is possible to extend the fuzzy model to include an additional process input, where the additional input is one of the significant process disturbances. Doing this, results in a significant improvement in the output response to the effect of this disturbance, both with the one-step ahead predictive controller as well as the long-range predictive controller.

Most industrial processes have a number of disturbances. If it is possible to develop a good dynamic model based on more than one disturbance, then the approach used in this chapter can be extended to multiple disturbances. Also, even though we have focused our attention non-adaptive process models, it is probably worthwhile investigating the use of adaptive MISO models to achieve even better performance.

Subsequent chapters will focus on extending the approach used in this chapter to multi-input multi-output (MIMO) systems.

Chapter 9

CONTROL OF MULTIVARIABLE SYSTEMS: BINARY DISTILLATION COLUMN

9.1 Introduction

Next to the CSTR, the distillation column is probably the most important process studied in chemical engineering literature. It is estimated that there are about 40,000 distillation columns in the U.S. consuming about 3 percent of the total U.S. energy usage. Distillation is used in many chemical processes for separating feed streams and for purification of final and intermediate product streams. Figure 9.1 is a schematic representation of a binary distillation column. Feed is separated into an overhead product or "distillate" and a bottoms product or "bottoms". Heat is transferred into the process in the reboiler (typically a tube-and-shell heat exchanger) to vapourise some of the liquid from the base of the column. The vapour coming from the top of the column is liquified in another tube-and-shell heat exchanger called the condenser and liquid from the condenser drops into the reflux drum. The distillate is removed from this drum. In addition, some liquid, called "reflux", is fed back to the top of the column.

The column itself consists of a number of stages (usually trays) which promote mass transfer of light components into the vapour flowing up the column and of heavy components into the liquid flowing down the column. When the liquid reflux stream enters the column, it is vapourised by the vapour it contacts. Because it gives up heat to boil the reflux stream, a portion of the vapour is condensed. This condensate falls to the tray below, where a portion of it is revapourised by vapour rising from that stage. This process continues down the column. Increasing the flow of the external reflux increases the internal reflux, i.e., the flow of liquid down the column. Similarly, increasing the boilup rate increases the flow of vapour up the column. The liquid reflux and the vapour boil-up are required to achieve the separation or "fractionation" of chemical components. The energy needed to make the separation is approximately the heat added to the reboiler.

A mathematical model of a 20-tray binary distillation column is provided in the text by Luyben (1990). A simplified version of this model which neglects the dynamics introduced by tray fluid mechanics is presented in Section 9.2. This model was used in all of our studies.

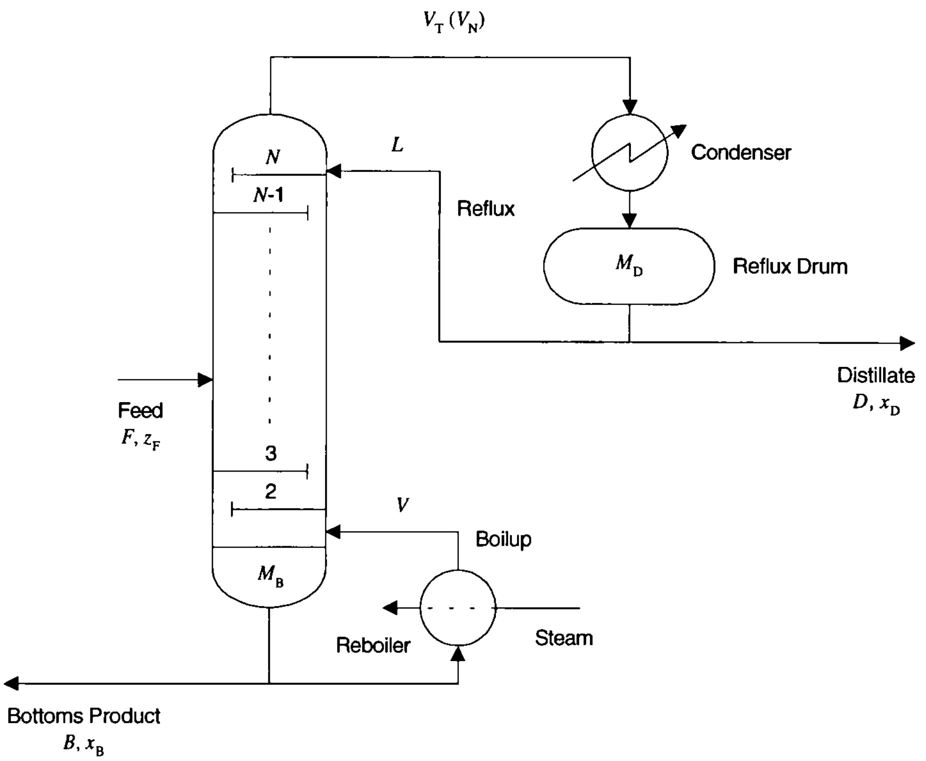


Figure 9.1: A binary distillation column.

9.2 Mathematical Model of Binary Distillation Column

The column is shown in Figure 9.1. The variable names, descriptions, steady-state values, and the engineering units used in deriving the mathematical model are given in Tables 9.1, 9.2 and 9.3. The distillation stages are numbered with the reboiler as Stage 1 and the condenser as Stage 22.

The following assumptions have been made to facilitate deriving the mathematical model:

- binary mixture
- constant pressure
- constant relative volatility
- no vapour holdup (immediate vapour response, $dV_T = dV_B$)
- constant liquid holdup M_i on all trays (immediate liquid response, $dL_T = dL_B$)
- vapour and liquid at equilibrium and perfect mixing on all stages

Material balances for change in holdup of light component on each tray, $i = 2,\ldots,N\ (i \neq N_F,\ i \neq N_F + 1)$:

$$M_i \frac{dx_i}{dt} = L_{i+1}x_{i+1} + V_{i-1}y_{i-1} - L_i x_i - V_i y_i \tag{9.1}$$

Above feed location, $i = N_F + 1$:

$$M_i \frac{dx_i}{dt} = L_{i+1}x_{i+1} + V_{i-1}y_{i-1} - L_i x_i - V_i y_i + F_V y_F \tag{9.2}$$

Below feed location, $i = N_F$:

$$M_i \frac{dx_i}{dt} = L_{i+1}x_{i+1} + V_{i-1}y_{i-1} - L_i x_i - V_i y_i + F_L x_F \tag{9.3}$$

Reboiler, $i = 1$:

$$M_B \frac{dx_i}{dt} = L_{i+1}x_{i+1} + V_i y_i - Bx_B, \quad x_B = x_1 \tag{9.4}$$

Total condenser, $i = N + 1$:

$$M_D \frac{dx_D}{dt} = V_{i-1} y_{i-1} - L_i x_D - D x_D, \quad x_D = x_{N+1} \tag{9.5}$$

Vapour-liquid equilibrium on each tray ($i = 1, \ldots, N$) based on constant relative volatility:

$$y_i = \frac{\alpha x_i}{1 + (\alpha - 1) x_i} \tag{9.6}$$

Flow rates assuming constant molar flows:

$$i > N_F \text{ (above feed)}: \; L_i = L, \;\; V_i = V + F_V \tag{9.7}$$

$$i \leq N_F \text{ (below feed)}: \; L_i = L + F_L, \;\; V_i = V \tag{9.8}$$

$$F_L = q_F F, \qquad F_V = F - F_L \tag{9.9}$$

$$D = V_N - L = V + F_V - L \qquad \text{(constant condenser holdup)} \tag{9.10}$$

$$B = L_2 - V_1 = L + F_L - V \qquad \text{(constant reboiler holdup)} \tag{9.11}$$

Compositions x_F and y_F in the liquid and vapour phase of the feed are obtained by solving the flash equations:

$$F z_F = F_L x_F + F_V y_F \tag{9.12}$$

Variable	Description	Value	Units
N	Number of equilibrium (theoretical) stages	21	-
N-1	Number of trays	20	-
N+1	Total number of stages including condenser	22	-
N_F	Feed stage location	11	-
F	Feed rate	1	kmol/min
z_F	Mole fraction of light component in feed	0.5	-
q_F	Mole fraction of liquid in feed	1	-
D	Distillate flow	0.5	kmol/min
V	Boilup	1.7801	kmol/min
V_i	Vapour flow from the *i*th stage	-	kmol/min
L	Reflux flow	1.2801	kmol/min
L_i	Liquid flow from the *i*th stage	-	kmol/min
B	Bottom flow	0.5	kmol/min
M_i	Tray hold-up	0.5	kmol
M_D	Condenser hold-up	0.5	kmol
M_B	Reboiler hold-up	0.5	kmol
x	Mole fraction of light component in liquid	-	-
y	Mole fraction of light component in vapour	-	-
x_D	Distillate composition	0.98	-
x_B	Bottoms composition	0.02	-
α	Relative volatility	2	-

Table 9.1: Distillation column variables and operating conditions during steady-state.

Subscript	Description
i	Distillation stage number
F	Feed
D	Distillate
B	Bottom
T	Top
V	Vapour phase
L	Liquid phase

Table 9.2: Subscripts used in modelling distillation process

Stage	x_i	Stage	x_i
1	0.02000	12	0.51526
2	0.03500	13	0.56295
3	0.05719	14	0.61896
4	0.08885	15	0.68052
5	0.13180	16	0.74345
6	0.18622	17	0.80319
7	0.24951	18	0.85603
8	0.31618	19	0.89995
9	0.37948	20	0.93458
10	0.43391	21	0.96079
11	0.47688	22	0.98000

Table 9.3: Distillation stage compositions under steady-state conditions.

9.3 Control Problem

Figure 9.2 shows the important feedback loops of a binary distillation column. Acceptable operation of a binary distillation column normally requires the following control objectives:

- Control of the composition of the distillate
- Control of the composition of the bottoms product
- Control of the liquid hold-up in the reflux drum
- Control of the liquid hold-up at the base of the column

The first two control objectives characterise the two product streams, whereas the other two objectives are required for operational feasibility (i.e., to prevent flooding and drying up of the reflux drum and the base of the column). The dynamic responses of Control Loops 3 and 4 in Figure 9.2 are usually much faster than the dynamic responses of Control Loops 1 and 2. In the development of our mathematical model, the dynamics introduced by Control Loops 3 and 4 have therefore been neglected and the hold-up of liquid in the reflux drum and the base of the column has been assumed to be constant.

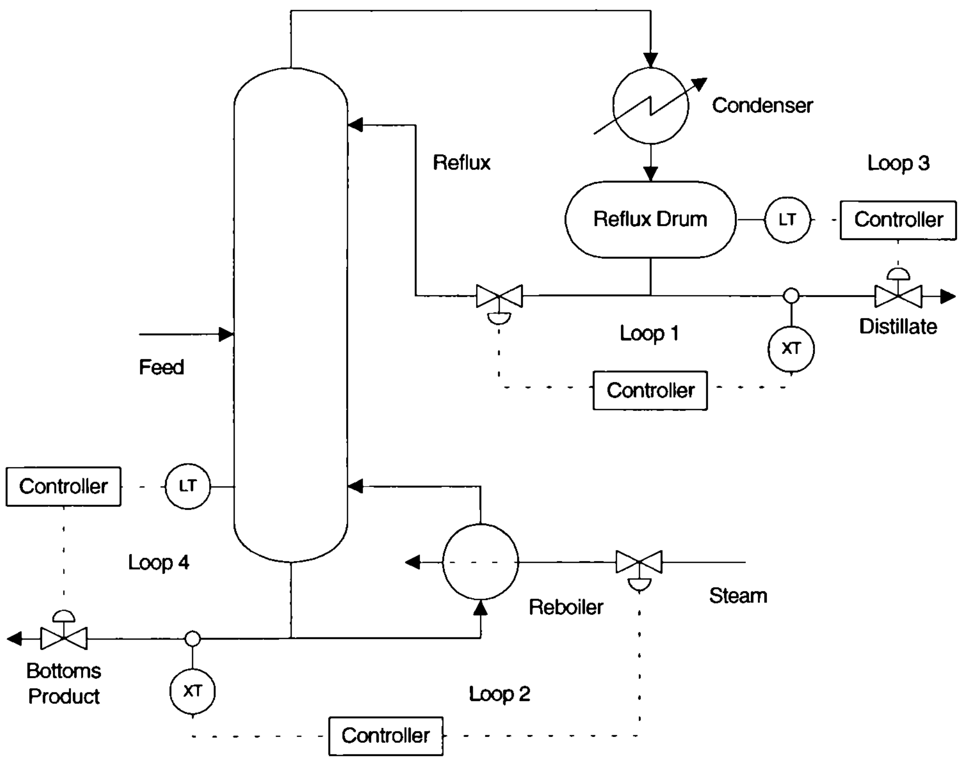


Figure 9.2: Control of a binary distillation column.

Distillation control is known to be difficult because of the following characteristics:

- The inherent nonlinearity of columns particularly for producing high-purity products.
- The severe coupling present for dual composition control.
- The variation in process gains due to process and operating changes.
- Large disturbances in feed flow rate and feed composition.

These problems are particularly important for dual composition control. When single-ended control is used, coupling is eliminated and the resulting control problem is greatly simplified. Unfortunately, single-ended control results in suboptimal operation in many cases due to issues of energy usage and product recovery.

In Figure 9.2, the quality of the distillate is controlled by manipulating the reflux rate, whereas the bottom product quality is controlled by manipulating the boilup rate. Control of the boilup rate is normally exercised by varying the steam flowrate to the reboiler. In our mathematical model, we have neglected the dynamics of the heat transfer processes in the condenser and the reboiler. In commercial-scale columns, the dynamic response of these heat exchangers is usually much faster than the response of the column itself. In some systems, however, the dynamics of these peripheral equipment are important and must be included in the model. The open loop response of the output variables to both manipulated variables is shown in Figures 9.3 to 9.6. It will be noticed that there is significant interaction and the time constants and the steady-state gains are variable, all of which make control particularly difficult.

The control problem investigated here will be setpoint changes in both distillate and bottoms composition as well as 10 percent step changes in feed composition and feed flowrate. Our primary aim here is to demonstrate the application of our controller to a difficult-to-control multivariable system where process nonlinearity and interaction are important considerations.

In the past few years, we have witnessed a proliferation of advanced multivariable control systems for the distillation process. Linear process model-based predictive controllers such as Dynamic Matrix Control (DMC) (Cutler and Ramaker, 1980) appear to be the most widely accepted methods for industrial application because of the relative ease of developing linear process models. Benefits reported include improved product qualities, yields and energy savings. These strategies can perform satisfactorily on nonlinear processes when the processes operate in the vicinity of the desired steady-state operating conditions. However, when a wide range of process operations with tight specifications on product purities is required, the nonlinearities become more critical and these controllers often need to be detuned to compensate for process/model mismatch and control performance is sacrificed.

The controller used in all of our investigations is the fuzzy model-based multi-step predictive controller with the control horizon set to 1-step. Before proceeding to design the multivariable controller itself, we will initially examine SISO model-based control of distillate and bottoms composition. Distillate composition has been paired with reflux flowrate, while bottoms composition has been paired with the boilup rate as illustrated in Figure 9.2. When designing each SISO model-based controller, it will be assumed that the other control loop is left open and the manipulated variable of that loop held constant at its normal steady-state value. In the next stage, these single loop controllers will be coupled to form a multivariable controller. This is done by expanding the process model used by each loop to include the manipulated variable of the other loop as a model input, essentially transforming the SISO process models to MISO process models.

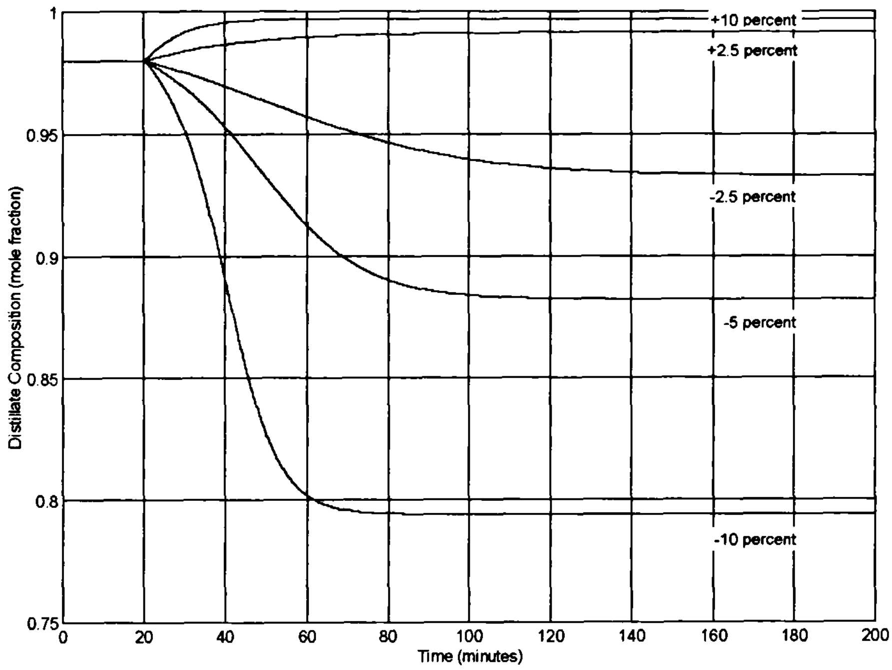


Figure 9.3: Open loop response of the distillate composition to step changes in the reflux flowrate (L).

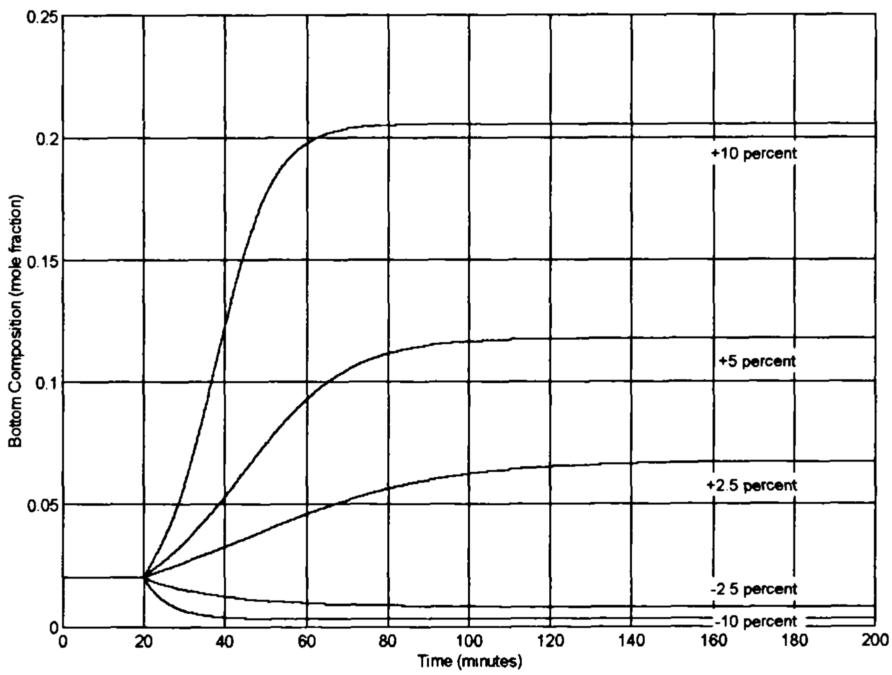


Figure 9.4: Open loop response of the bottoms composition to step changes in the reflux flowrate (L).

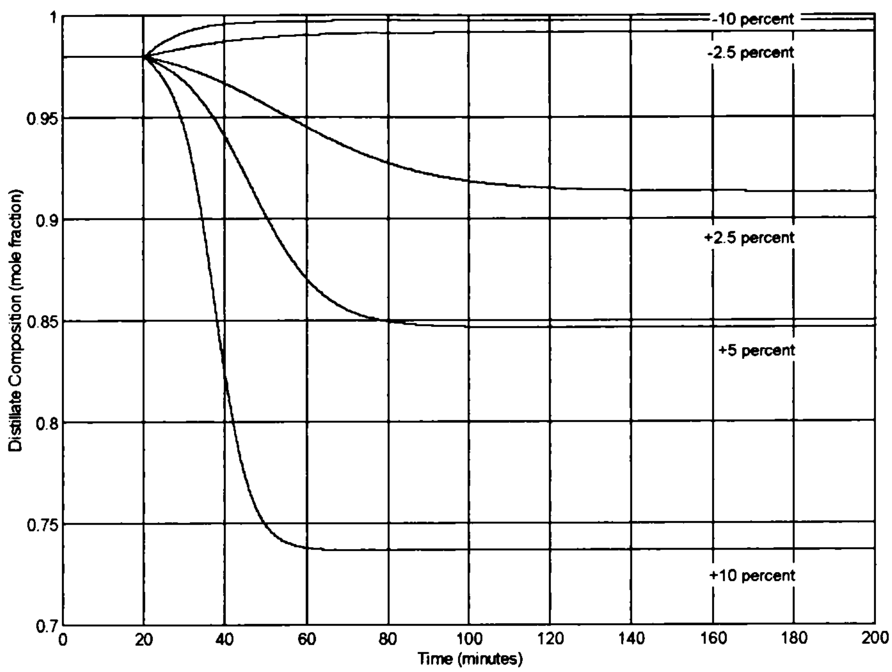


Figure 9.5: Open loop response of the distillate composition to step changes in the boilup flowrate (V).

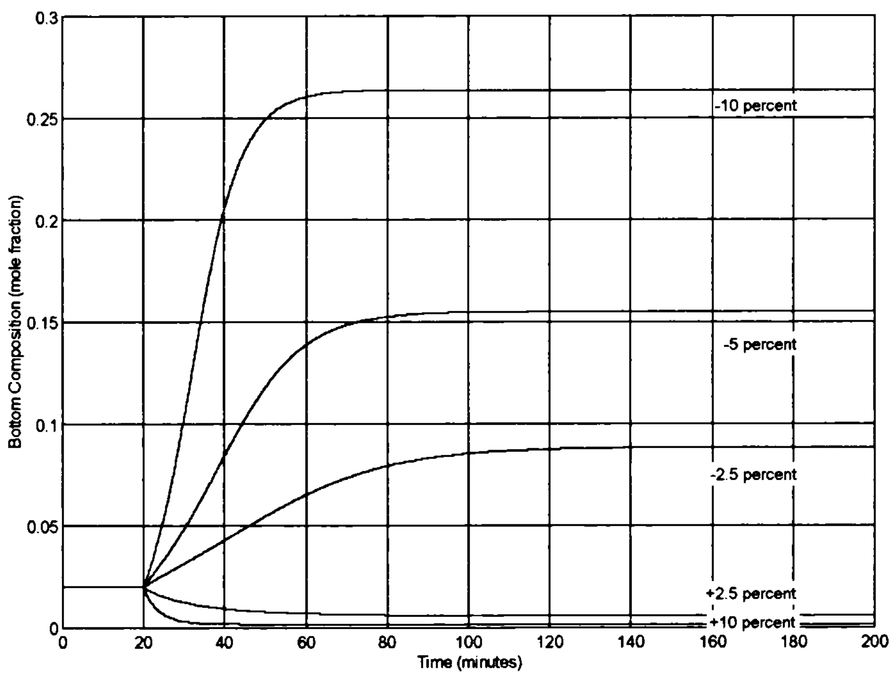


Figure 9.6: Open loop response of the bottoms composition to step changes in the boilup flowrate (V).

9.4 SISO Distillate Composition Control

9.4.1 Identification of SISO Process Model

Data for modelling were generated by applying 40 uniformly distributed random step changes, each lasting 50 minutes, in the reflux rate (L) as shown in Figure 9.7 while maintaining the distillate composition approximately between 0.92 and 1 mole fraction of lighter component. The model was updated at 0.1 minute intervals using the Runge-Kutta third order numerical integration method. Sampling for control was carried out at 2-minute intervals and the output from the controller was maintained constant over the sampling interval using a zero-order hold. A total of 1000 data points were used for identification.

Since the open-loop response between distillate composition and reflux flowrate (Figure 9.3) can be approximated by a first-order system, the following model structure has been assumed:

$$R^i:\ if\, x_D(t)\ is\ A^i\ then\, x_D(t+1) = a_1^i x_D(t) + b_1^i L(t) + k_i,\quad i = 1,\dots,p \qquad (9.13)$$

where $x_D(t)$ and $L(t)$ are the distillate composition and the reflux flowrate at time t, respectively.

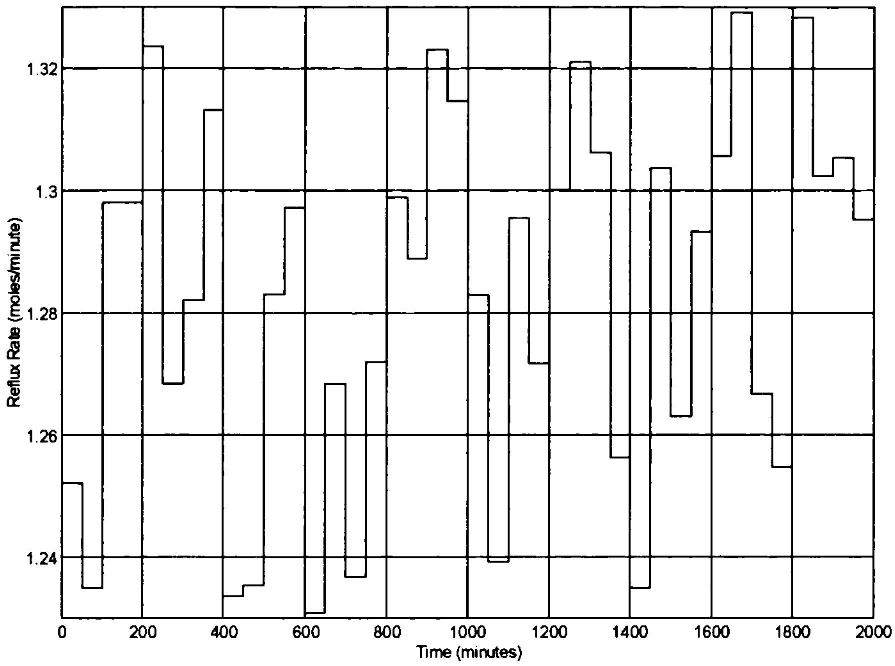
Different methods of fuzzy partitioning the input space as shown in Figure 9.8 were examined. Table 9.4 shows the model parameters calculated using the input/output data shown in Figure 9.7. It will be observed that the fuzzy process models provide an improvement in modelling accuracy over the linear process model.

9.4.2 SISO controller

Studies showed very little difference in performance of controllers using different fuzzy models reflecting the small difference in their modelling accuracies in Table 9.4. A comparison of Figures 9.9 and 9.10 shows, however, that there is a significant difference between the performances of controllers using linear and 3-partition fuzzy models when examined over the range of 0.93 to 0.99 mole fraction of lighter component. The output response to the controller using the linear model ranges from over-damped to under-damped whereas the output response to the controller using the fuzzy model is quite consistent.

No.	Type of model	Model parameters			MSE
		a_1^i	b_1^i	k_i	
1	linear model	0.9621	0.0396	-0.0138	54.142 x 10^{-8}
2	2-partition fuzzy model	0.8899 0.9383	0.1249 0.0072	-0.0541 0.0517	2.548 x 10^{-8}
3	3-partition fuzzy model	0.9092 0.9349 0.9558	0.1147 0.0684 0.0041	-0.0590 -0.0240 0.0384	1.834 x 10^{-8}
4	5-partition fuzzy model	0.9274 0.9560 0.9580 0.9762 0.9685	0.1034 0.1016 0.0667 0.0299 0.0074	-0.0616 -0.0861 -0.0441 -0.0149 0.0213	1.287 x 10^{-8}

Table 9.4: Effect of the number of fuzzy partitions of the input space on the model parameters and the modelling accuracy of SISO process model used for distillate composition control.



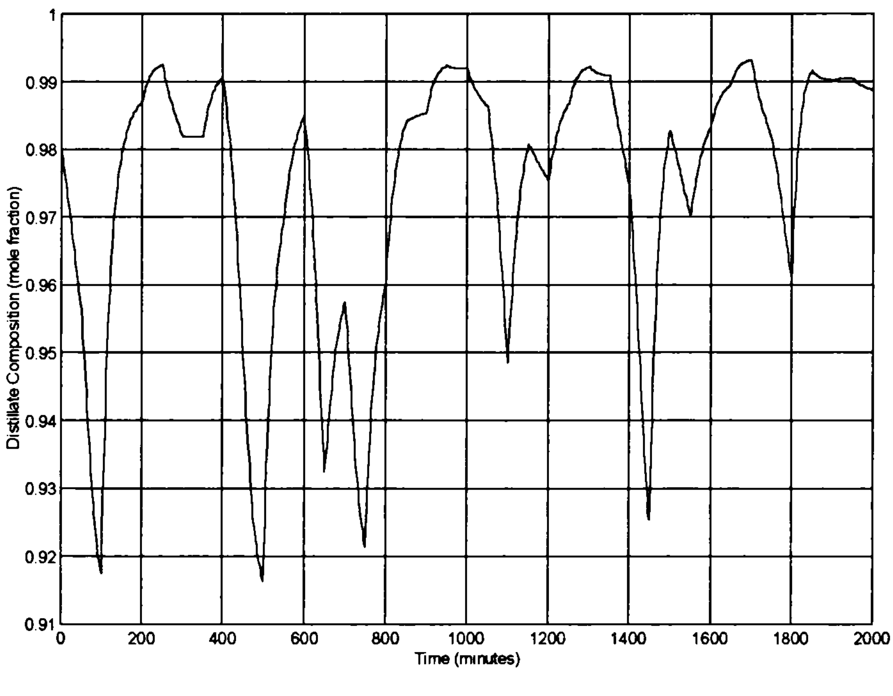


Figure 9.7: Input/output data utilised for identification of the SISO process model for distillate composition control loop.

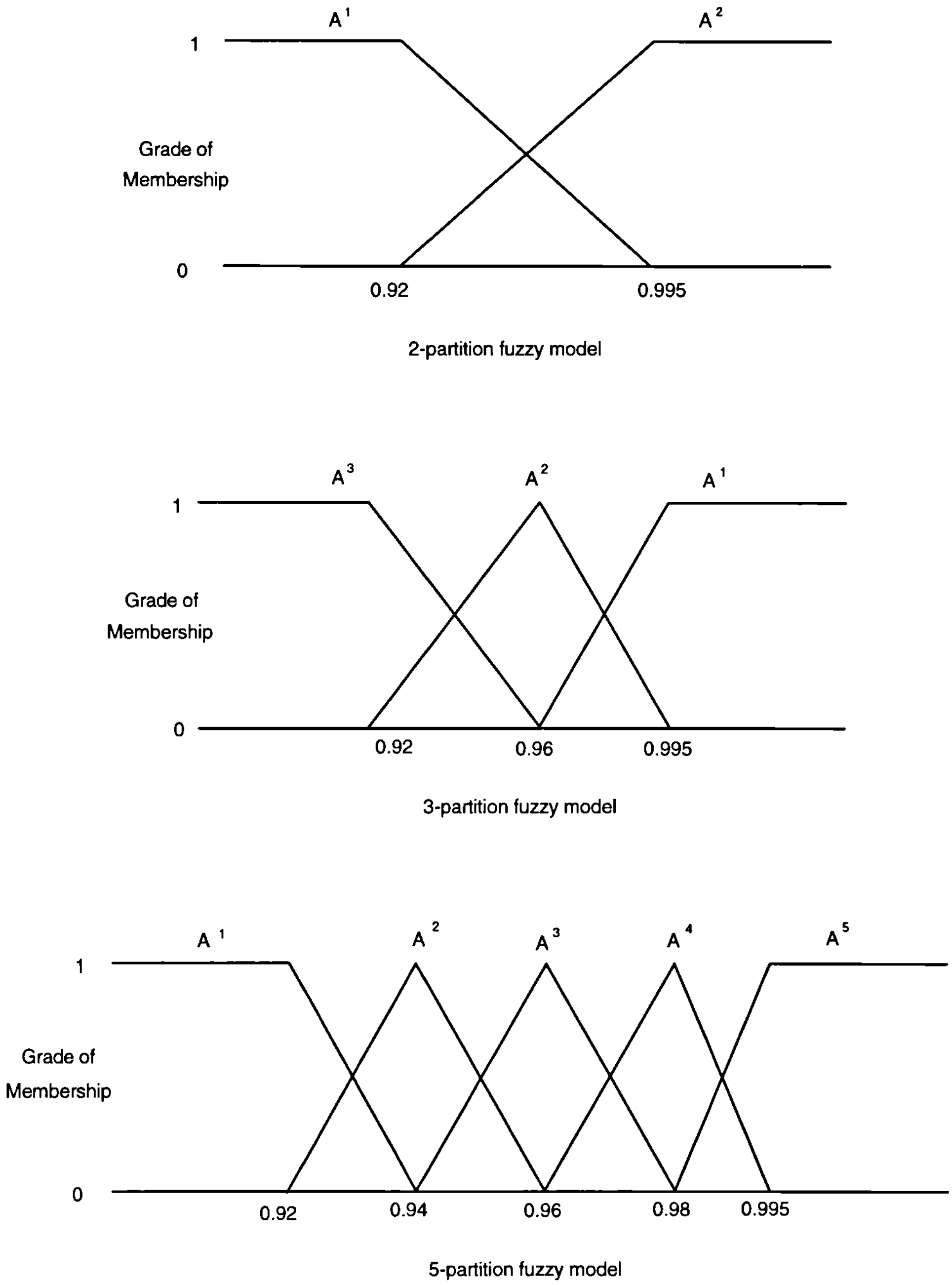


Figure 9.8: Fuzzy partitioning of the input space used for deriving the SISO process models in Table 9.4.

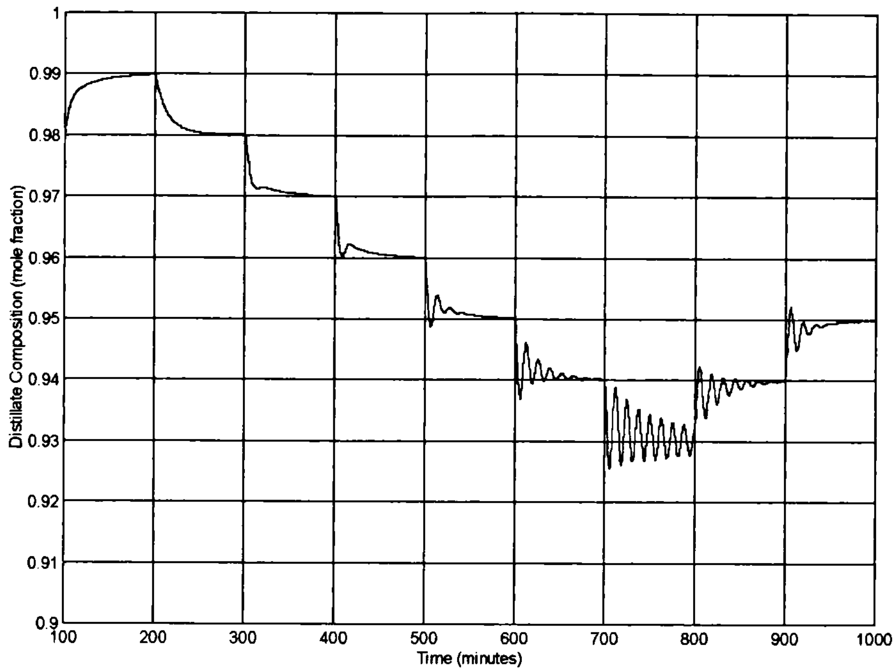


Figure 9.9: Process output response to setpoint changes in distillate composition when using 5-steps ahead predictive controller with linear SISO process model ($K_{f_1} = 1; K_{f_2} = 0.1$).

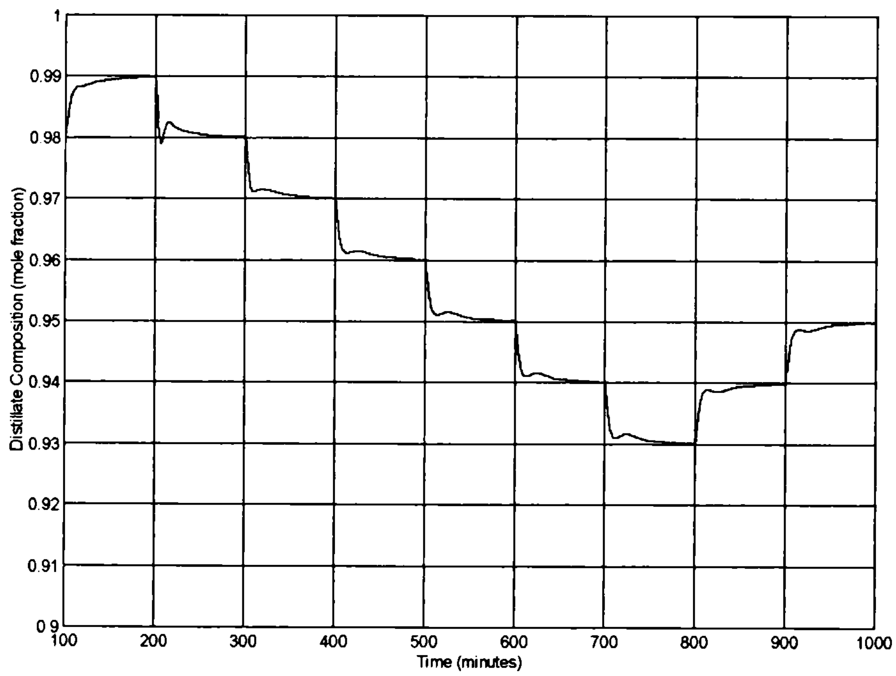


Figure 9.10: Process output response to setpoint changes in distillate composition when using 5-steps ahead predictive controller with 3-partition fuzzy SISO process model ($K_{f_1} = 1; K_{f_2} = 0.1$).

9.5 SISO Bottom Composition Control

9.5.1 Identification of SISO Process Model

The approach used was quite similar to that used for SISO distillate composition control as described above. Data for modelling were generated by applying 40 uniformly distributed random step changes, each lasting 50 minutes, in the boilup rate (V) as shown in Figure 9.11 while maintaining the bottoms composition approximately between 0 and 0.08 mole fraction of lighter component. Sampling for control was once again carried out at 2-minute intervals and a total of 1000 data points were used for identification.

Since the open-loop response between bottoms product composition and boilup rate (Figure 9.6) can be approximated by a first-order system, the following model structure has been assumed:

$$R^i:\ if\, x_B(t)\ is\ A^i\ then\, x_B(t+1) = a_1^i x_B(t) + b_1^i V(t) + k_i,\quad i = 1,\ldots,p \qquad (9.14)$$

where $x_B(t)$ and $V(t)$ are the bottoms composition and the boilup rate at time t, respectively.

Different methods of fuzzy partitioning the input space as shown in Figure 9.12 were examined. Table 9.5 shows the model parameters calculated using the input/output data shown in Figure 9.11. Once again, it will be observed that the fuzzy process models provide an improvement in modelling accuracy over the linear process model.

9.5.2 SISO controller

As in the case of distillate composition control, it was found that there is very little difference in the performances of controllers using 3 and 5-partition fuzzy models reflecting the small difference in the modelling accuracies of these two models. The performance of the controller when using the 2-partition fuzzy model is only slightly worse. A comparison of Figures 9.13 and 9.14 show that there is a significant difference between the performance of the controllers using linear and 3-partition fuzzy models when examined over the wider range of 0.01 to 0.07 mole fraction of lighter component. In fact, setpoint change to 0.07 mole fraction using the linear model was found to lead to instability and has not been shown in Figure 9.13. The output response to the controller using the linear model in Figure 9.13 varies from from over-damped to under-damped whereas the output response to the controller using the fuzzy model in Figure 9.14 is quite consistent inspite of the wider range examined.

No.	Type of model	Model parameters			MSE
		a_1^i	b_1^i	k_i	
1	linear model	0.9510	-0.0496	0.0897	85.441 x 10^{-8}
2	2-partition fuzzy model	0.8858 0.8116	-0.0089 -0.1622	0.0168 0.2976	2.176 x 10^{-8}
3	3-partition fuzzy model	0.9227 0.9038 0.8492	-0.0105 -0.0795 -0.1629	0.0197 0.1442 0.2956	1.938 x 10^{-8}
4	5-partition fuzzy model	0.9142 0.9499 0.9395 0.9136 0.8654	-0.0140 -0.0376 -0.0827 -0.1166 -0.1669	0.0262 0.0679 0.1483 0.2096 0.3012	1.741 x 10^{-8}

Table 9.5: Effect of the number of fuzzy partitions of the input space on the model parameters and the modelling accuracy of SISO process model used for bottoms composition control.

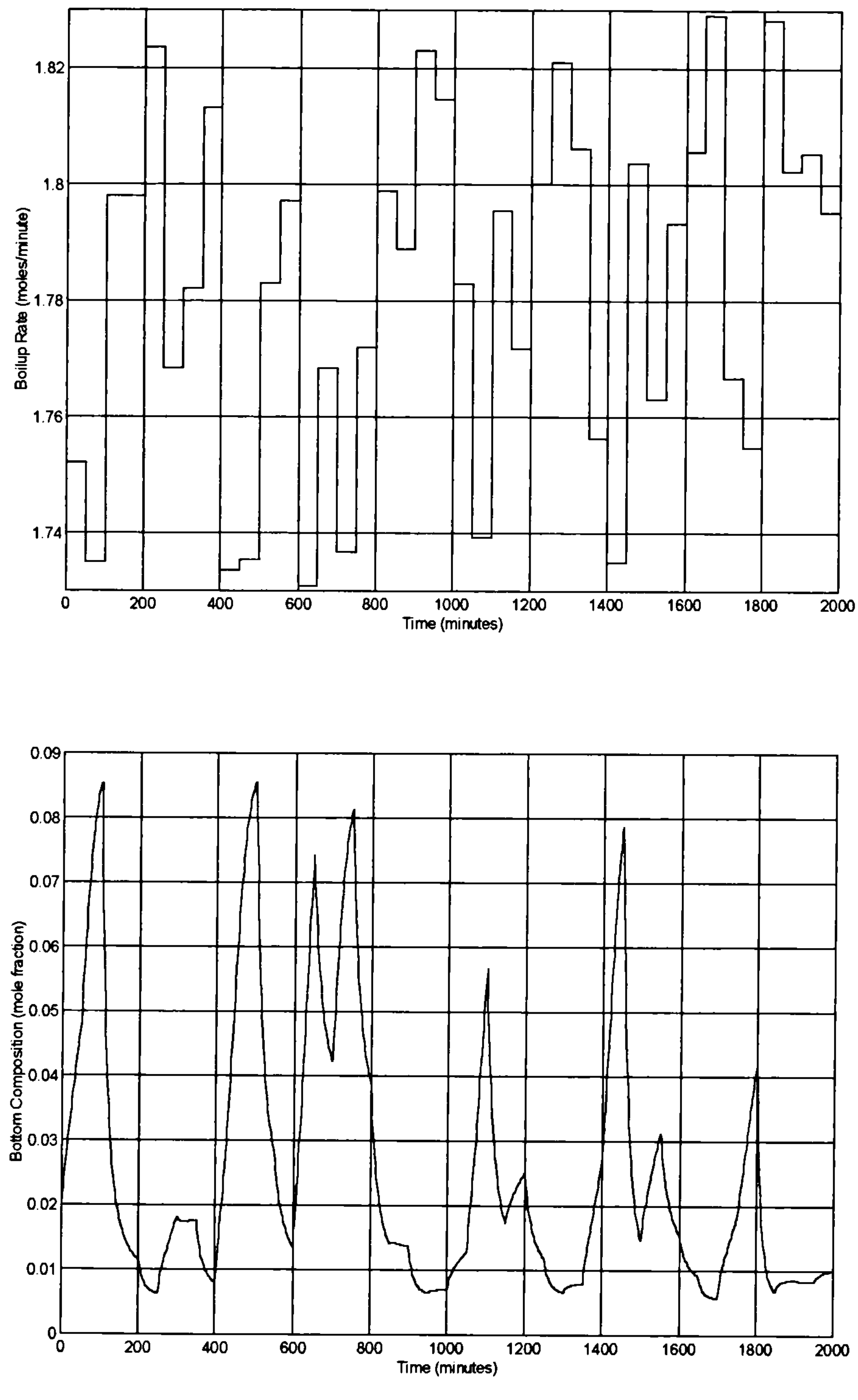


Figure 9.11: Input/output data utilised for identification of the SISO process model for bottoms composition control loop.

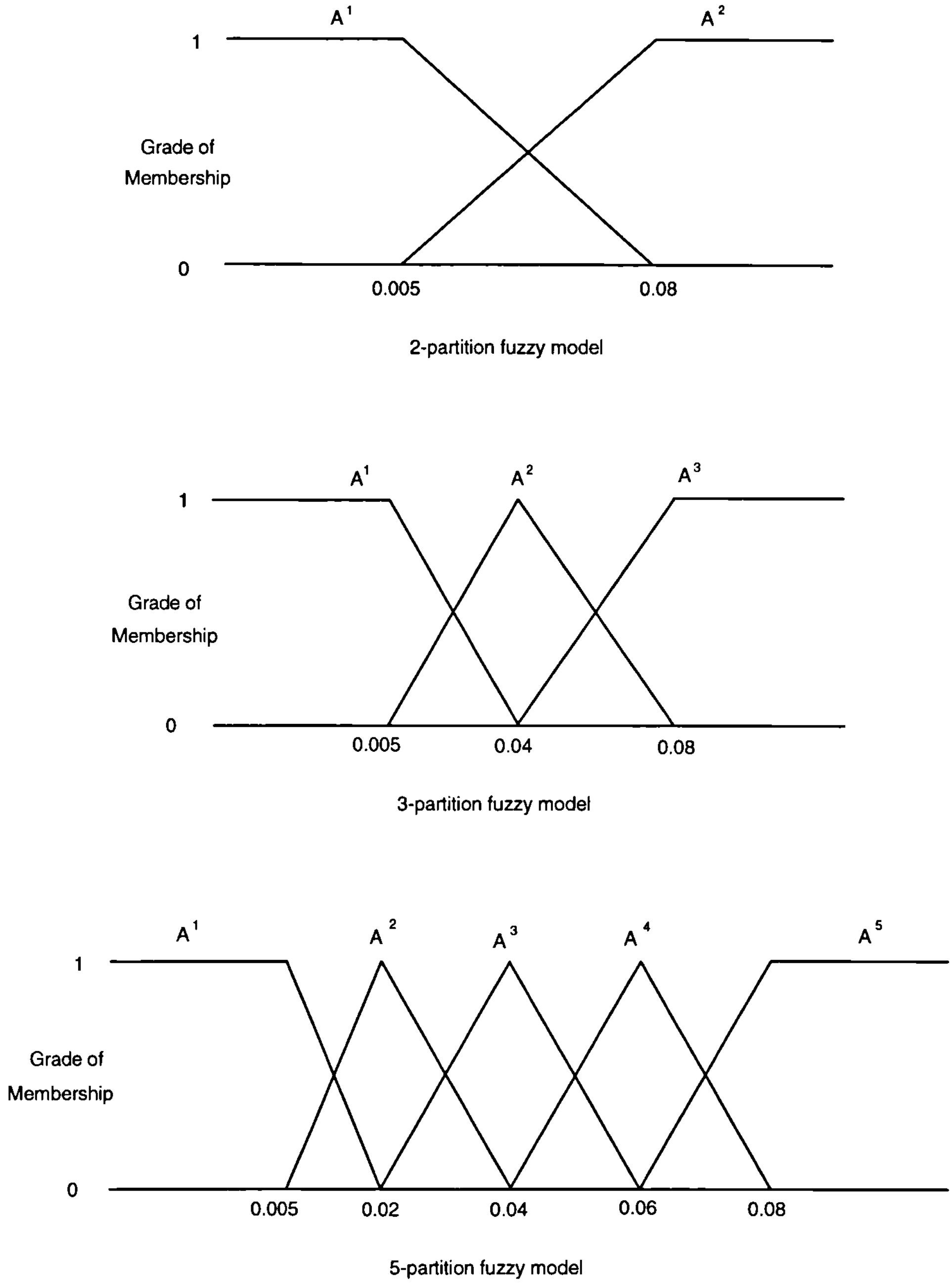


Figure 9.12: Fuzzy partitioning of the input space used for deriving the SISO process models in Table 9.5.

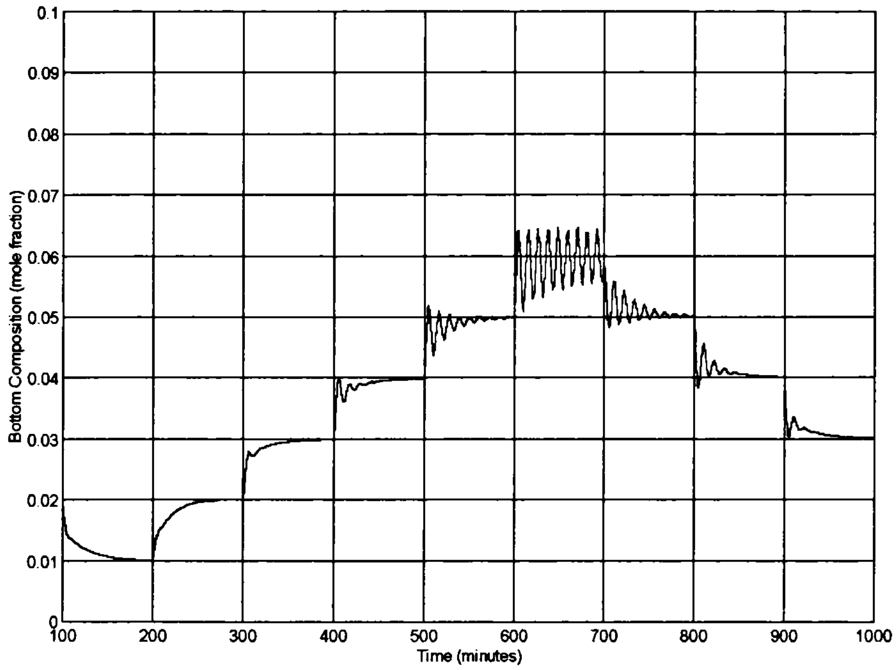


Figure 9.13: Process output response to setpoint changes in bottoms composition when using 5-steps ahead predictive controller with linear SISO process model ($K_{f_1} = 1; K_{f_2} = 0.1$).

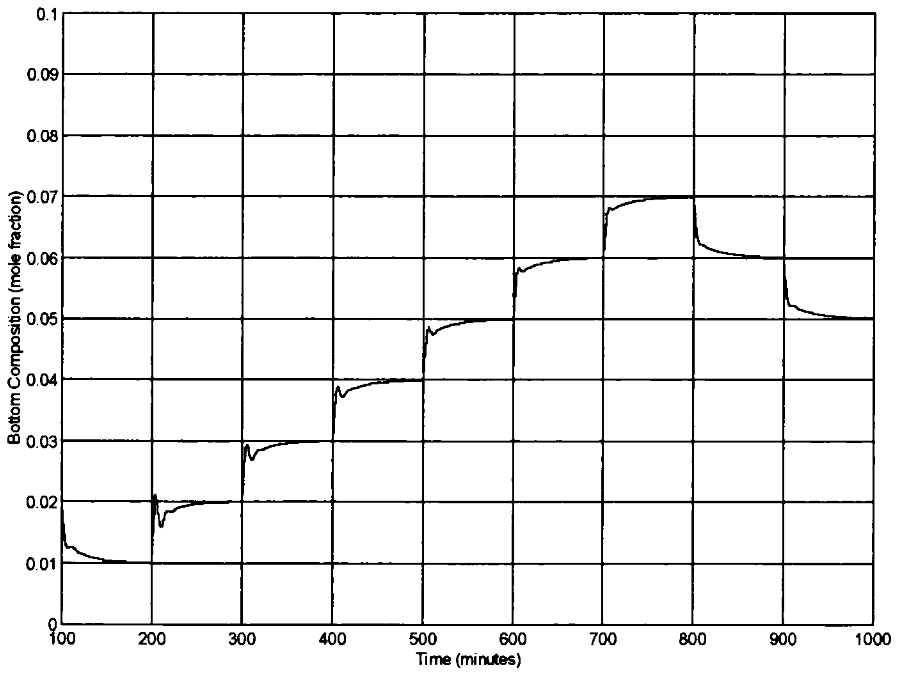


Figure 9.14: Process output response to setpoint changes in bottoms composition when using 5-steps ahead predictive controller with 3-partition fuzzy SISO process model ($K_{f_1} = 1; K_{f_2} = 0.1$).

9.6 MIMO Control of Distillate and Bottoms Composition

9.6.1 Identification of MISO Process Model for Distillate Composition Control

Data for modelling were generated by applying 40 uniformly distributed random step changes, each lasting 50 minutes, in the reflux rate (L) and the boilup rate (V) as shown in Figure 9.15 while maintaining the distillate composition approximately between 0.92 and 1 mole fraction of lighter component. It will be observed that the magnitude of the step changes applied to the reflux flowrate were much bigger than the changes applied to the boilup rate. Sampling for control was once again carried out at 2-minute intervals and a total of 1000 data points were used for identification.

Since the open-loop response of the distillate composition to changes in the reflux rate and the boilup rate (Figures 9.3 and 9.5) can be approximated by first-order systems, the following MISO model structure has been assumed:

$$R^i\text{: } if\, x_D(t) \text{ } is\, A^i \text{ } then\, x_D(t+1) = a_1^i x_D(t) + b_1^i L(t) + c_1^i V(t) + k_i,\ \ i = 1,\ldots,p \qquad (9.15)$$

where $x_D(t)$, $L(t)$ and $V(t)$ are the distillate composition, the reflux flowrate and the boilup rate at time t, respectively.

Fuzzy partitioning of the input space was maintained the same as in Figure 9.8. Table 9.6 shows the model parameters calculated using the input/output data shown in Figure 9.15. Here again, it will be observed that the fuzzy process models provide a significant improvement in modelling accuracy over the linear process model.

9.6.2 Identification of MISO Process Model for Bottoms Composition Control

Data for modelling were generated by applying 40 uniformly distributed random step changes, each lasting 50 minutes, in both the boilup rate (V) and the reflux flowrate (L) as shown in Figure 9.16 while maintaining the bottoms composition approximately between 0 and 0.08 mole fraction of lighter component. It will be observed that the magnitude of the step changes applied in the boilup rate were much bigger than the changes applied to the reflux rate. Sampling for control was once again carried out at 2-minute intervals and a total of 1000 data points were used for identification.

Since the open-loop response of the bottoms composition to changes in the boilup rate and the reflux rate (Figures 9.6 and 9.4) can be approximated by first-order systems, the following MISO model structure has been assumed:

$$R^i: \; if \; x_B(t) \; is \; A^i \; then \; x_B(t+1) = a_1^i x_B(t) + b_1^i V(t) + c_1^i L(t) + k_i, \;\; i = 1,\ldots,p \quad (9.16)$$

where $x_B(t)$, $L(t)$ and $V(t)$ are the bottoms composition, the reflux flowrate and the boilup rate at time *t*, respectively.

Fuzzy partitioning of the input space was maintained the same as in Figure 9.12. Table 9.7 shows the model parameters calculated using the input/output data shown in Figure 9.16. It will be observed here again that the fuzzy process models provide a significant improvement in modelling accuracy over the linear process model.

9.6.3 MIMO Controller

Even though the design of a MIMO controller can be approached by using a single objective function covering all the control loops, this will not be attempted here because it requires the use of large matrices to manage the computations. We will instead use the alternative approach of decomposing a MIMO model based controller into MISO model based controllers and computing the optimal controller output of each MISO controller using separate objective functions. The design of a fuzzy model based multi-step predictive controller for MISO systems has been presented in Chapter 8 and will therefore not be repeated here. By including the manipulated variables of the other control loops as model inputs, it may be possible to achieve a feed-forward control effect to changes in these variables. The concept involved is similar to compensation of disturbance inputs discussed in Chapter 8. Doing this with both loops in the case of the distillation control problem will minimize the interaction between the loops.

Figures 9.17 to 9.20 show the process output responses to setpoint changes in the distillate composition. The prediction horizon used by the MISO controllers for distillate composition and bottoms composition was set to 7-steps and 5-steps, respectively. The plot of bottoms composition has been displaced upwards (i.e., plot shows $x_B + 0.9$) to allow plotting alongside with distillate composition. As in the case of the SISO controller, it will be observed that the response of the distillate composition to setpoint changes varies from over-damped to under-damped when a linear model is used. Better performance is achieved with fuzzy process models.

Similar observations can also be made regarding setpoint changes in the bottoms composition (Figures 9.21 to 9.24). In this case the plot of distillate composition has been displaced downwards (i.e., plot shows $x_D - 0.9$) to allow plotting alongside with bottoms composition.

An interesting observation is the generally poorer response with 5-partition fuzzy process models to setpoint changes in distillate composition in the range 0.92 to 0.94 and to setpoint changes in the bottoms composition in the range 0.06 to 0.08. This can probably be attributed to the fact that identification of the 5-partition fuzzy models was based on very few sample points in these regions (refer to Figures 9.15 and 9.16). The 2-partition and 3-partition fuzzy models probably provide better modelling accuracy than the 5-partition fuzzy model in these regions.

Figure 9.25 shows that the performance of a controller using the 3-partition fuzzy model is generally better to feed composition changes than a comparable controller using a linear process model. The plot of bottoms composition has been displaced upwards (i.e., plot shows $x_B + 0.95$) to allow plotting alongside with distillate composition. Similar observations to the above can be made from Figure 9.26 in the case of feed flowrate changes.

No.	Type of model	Model parameters				MSE
		a_1^i	b_1^i	c_1^i	k_i	
1	linear model	0.9676	0.0365	-0.0327	0.0431	48.632 x 10^{-8}
2	2-partition fuzzy model	0.9553 1.0097	0.1265 0.0063	-0.1217 -0.0040	0.1004 -0.0109	2.734 x 10^{-8}
3	3-partition fuzzy model	0.9501 0.9661 0.9758	0.1216 0.0646 0.0055	-0.1199 -0.0585 -0.0042	0.1080 0.0548 0.0241	2.484 x 10^{-8}
4	5-partition fuzzy model	0.9331 0.9142 0.9298 0.9428 0.9285	0.1108 0.1024 0.0634 0.0296 0.0081	-0.1103 -0.1014 -0.0572 -0.0266 -0.0046	0.1200 0.1328 0.0889 0.0655 0.0684	2.134 x 10^{-8}

Table 9.6: Effect of the number of fuzzy partitions of the input space on the model parameters and the modelling accuracy of MISO process model used for distillate composition control.

No.	Type of Model	Model Parameters				MSE
		a_1^i	b_1^i	c_1^i	k_i	
1	linear model	0.9486	-0.0481	0.0434	0.0314	99.282 x 10^{-8}
2	2-partition fuzzy model	0.8913 0.8256	-0.0090 -0.1618	0.0071 0.1468	0.0081 0.1080	3.106 x 10^{-8}
3	3-partition fuzzy model	0.8712 0.8600 0.8299	-0.0106 -0.0787 -0.1632	0.0092 0.0688 0.1545	0.0083 0.0564 0.1002	2.909 x 10^{-8}
4	5-partition fuzzy model	0.8440 0.8874 0.8713 0.8573 0.8337	-0.0163 -0.0354 -0.0831 -0.1174 -0.1643	0.0107 0.0321 0.0738 0.1013 0.1679	0.0171 0.0241 0.0574 0.0845 0.0845	2.492 x 10^{-8}

Table 9.7: Effect of the number of fuzzy partitions of the input space on the model parameters and the modelling accuracy of MISO process model used for bottoms composition control.

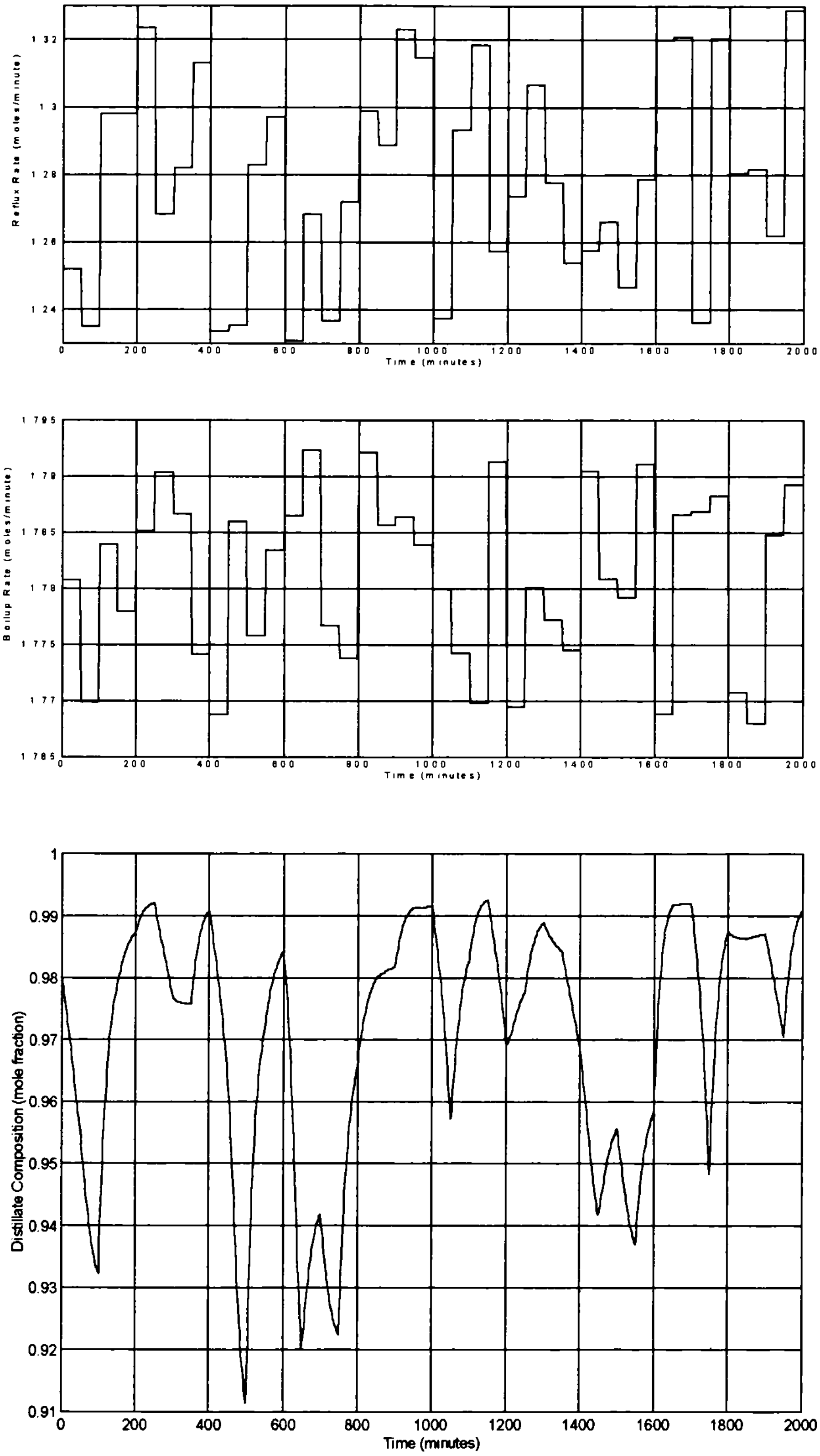


Figure 9.15: Input/output data utilised for identification of the MISO process model for distillate composition control loop.

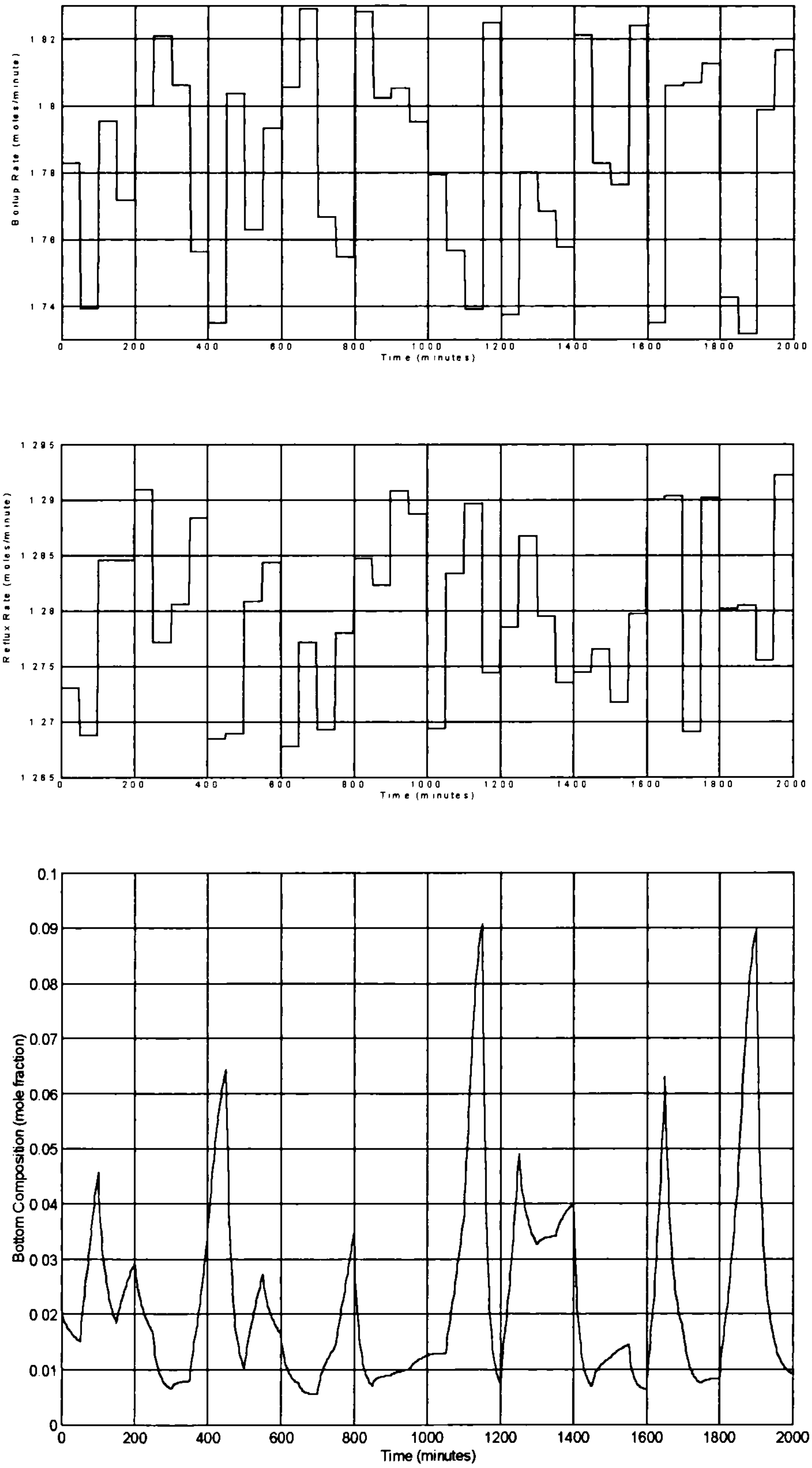


Figure 9.16: Input/output data utilised for identification of the MISO process model for bottoms composition control loop.

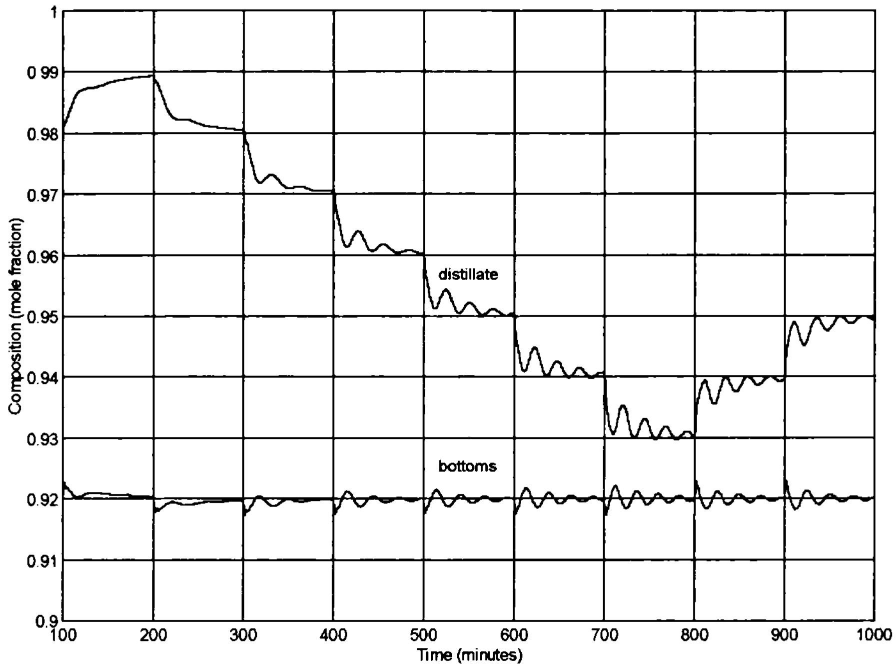


Figure 9.17: Process output responses to setpoint changes in distillate composition when using proposed controller with linear MISO process models ($K_{f_1} = 1; K_{f_2} = 0.1$).

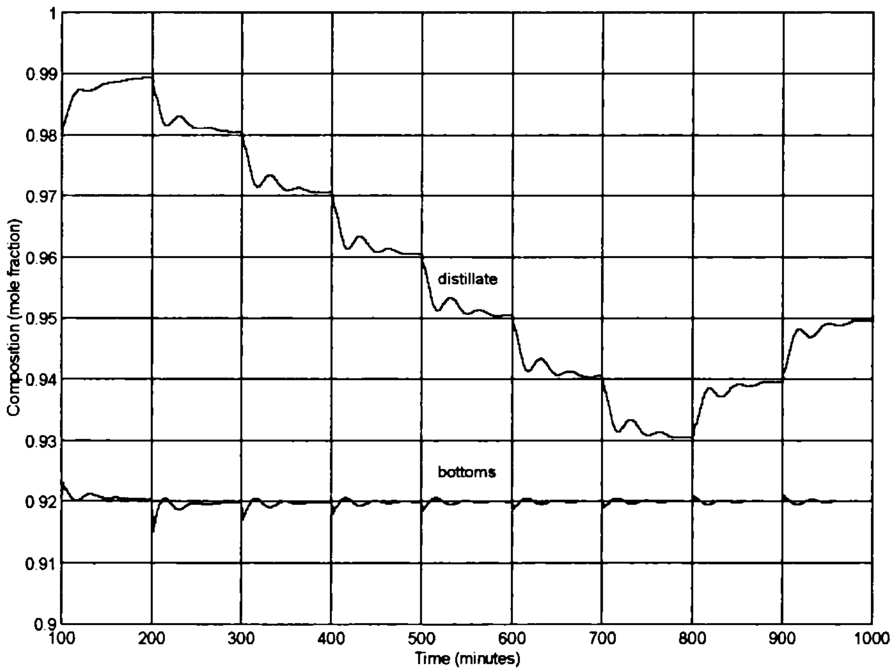


Figure 9.18: Process output responses to setpoint changes in distillate composition when using proposed controller with 2-partition fuzzy MISO process models ($K_{f_1} = 1; K_{f_2} = 0.1$).

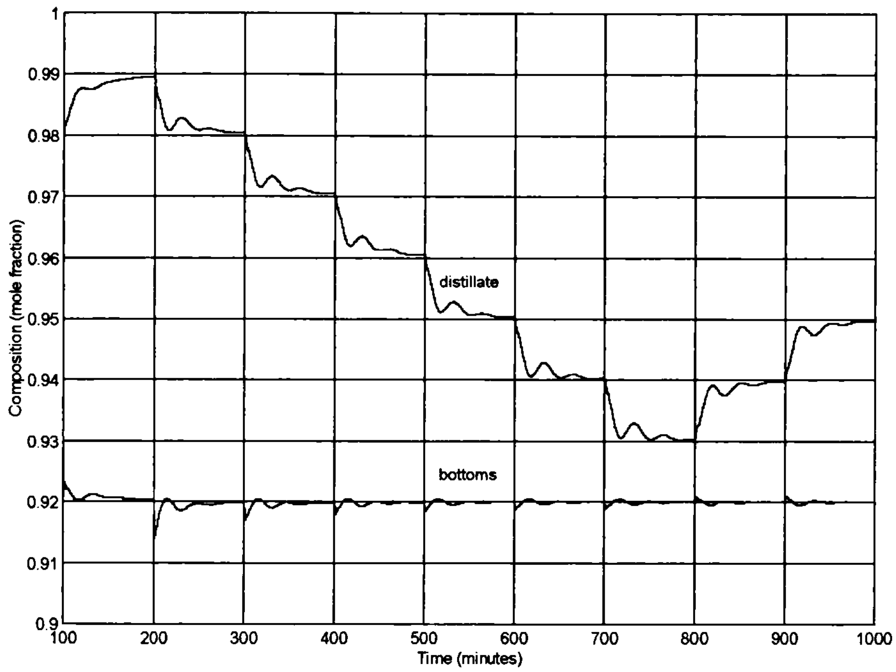


Figure 9.19: Process output responses to setpoint changes in distillate composition when using proposed controller with 3-partition fuzzy MISO process models ($K_{f_1} = 1; K_{f_2} = 0.1$).

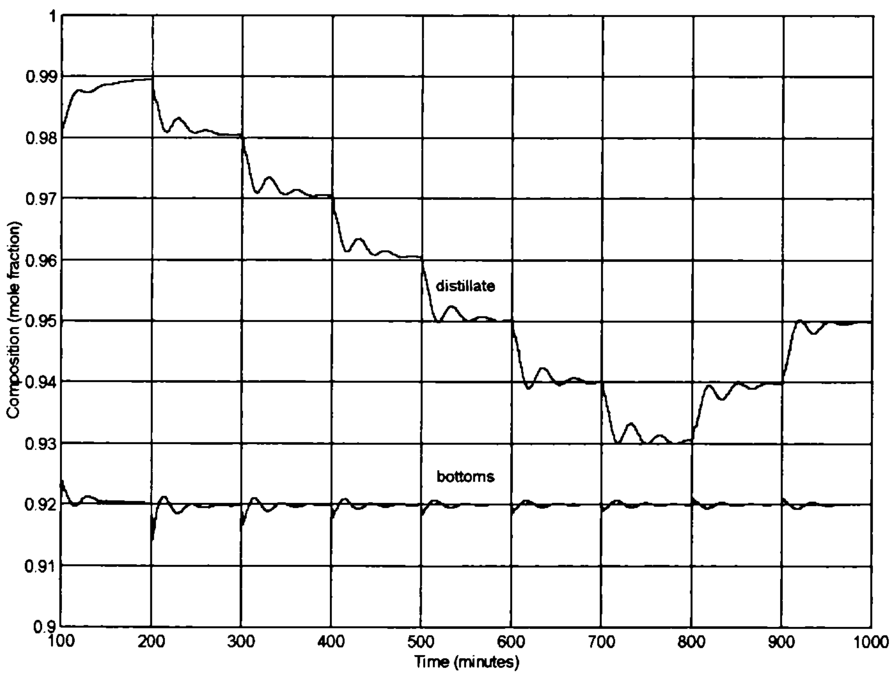


Figure 9.20: Process output responses to setpoint changes in distillate composition when using proposed controller with 5-partition fuzzy MISO process models ($K_{f_1} = 1; K_{f_2} = 0.1$).

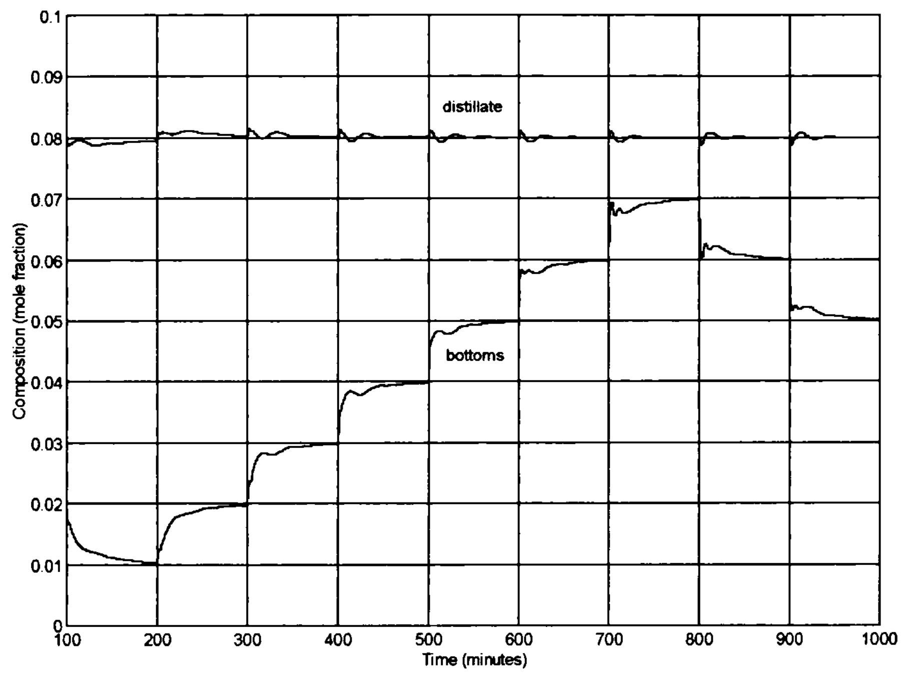


Figure 9.21: Process output responses to setpoint changes in bottom composition when using proposed controller with linear MISO process models ($K_{f_1} = 1; K_{f_2} = 0.1$).

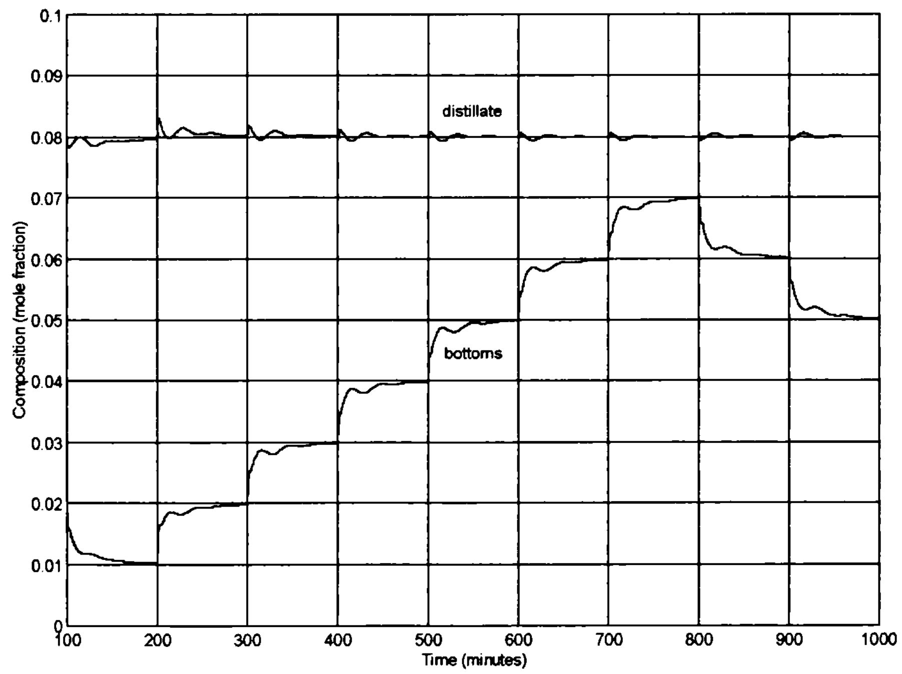


Figure 9.22: Process output responses to setpoint changes in bottom composition when using proposed controller with 2-partition fuzzy MISO process models ($K_{f_1} = 1; K_{f_2} = 0.1$).

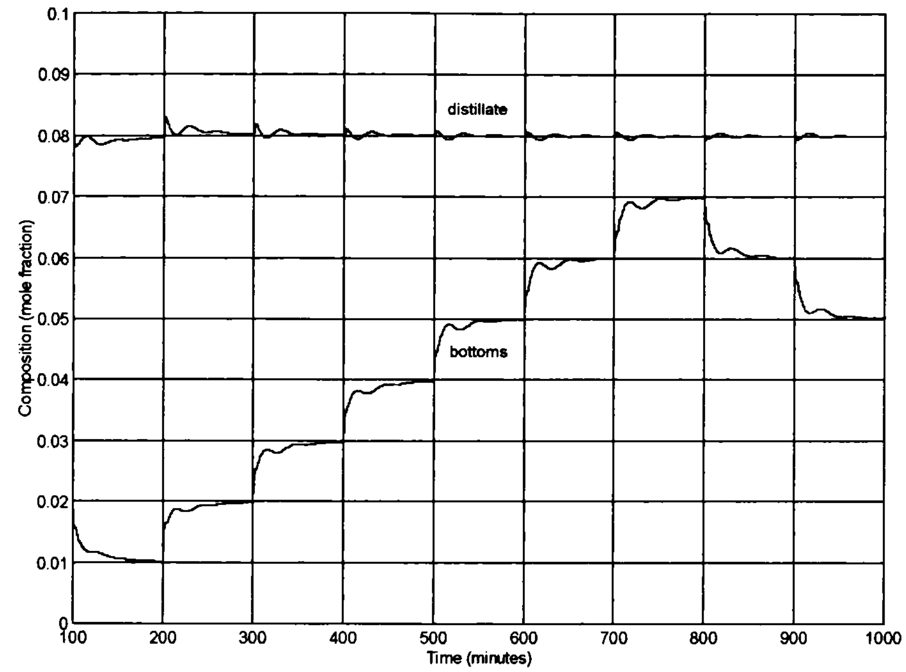


Figure 9.23: Process output responses to setpoint changes in bottom composition when using proposed controller with 3-partition fuzzy MISO process models ($K_{f_1} = 1; K_{f_2} = 0.1$).

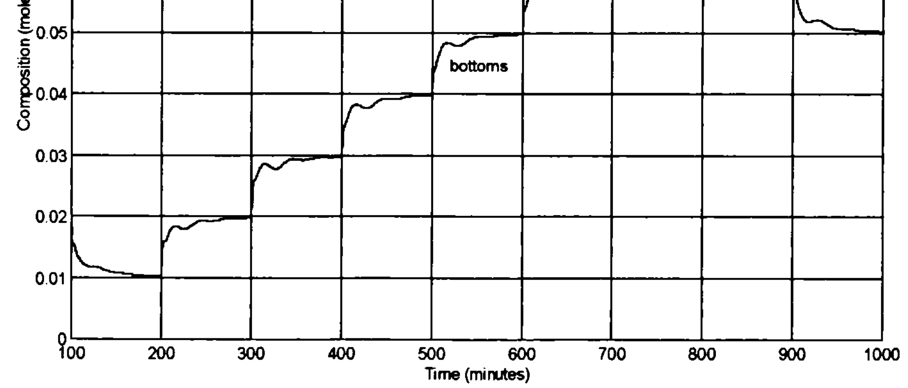


Figure 9.24: Process output responses to setpoint changes in bottom composition when using proposed controller with 5-partition fuzzy MISO process models ($K_{f_1} = 1; K_{f_2} = 0.1$).

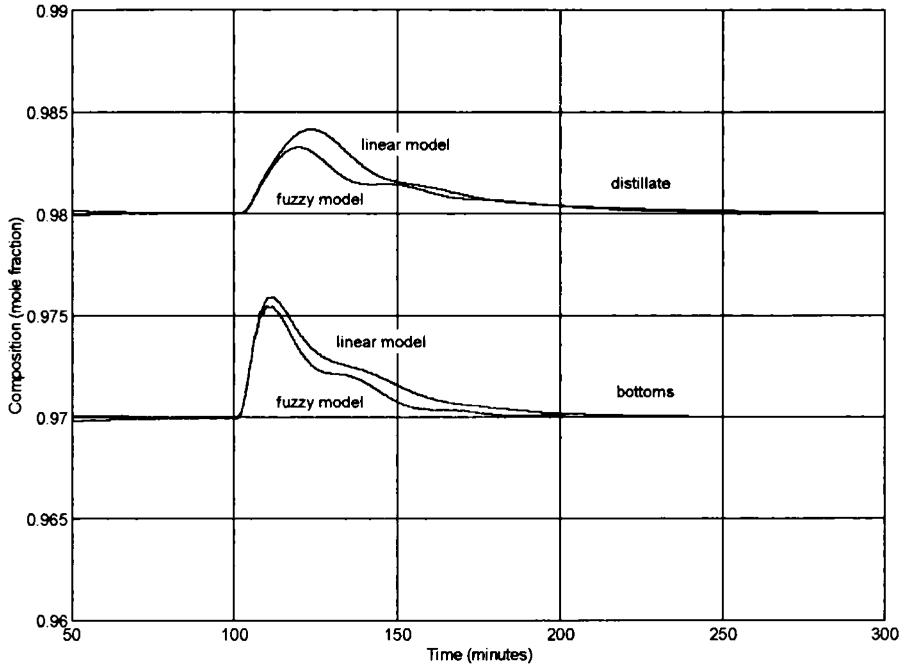


Figure 9.25: Comparison of process output responses to +10 percent feed composition change when using proposed controller with linear and 3-partition fuzzy MISO process models ($K_{f_1} = 1; K_{f_2} = 0.1$).

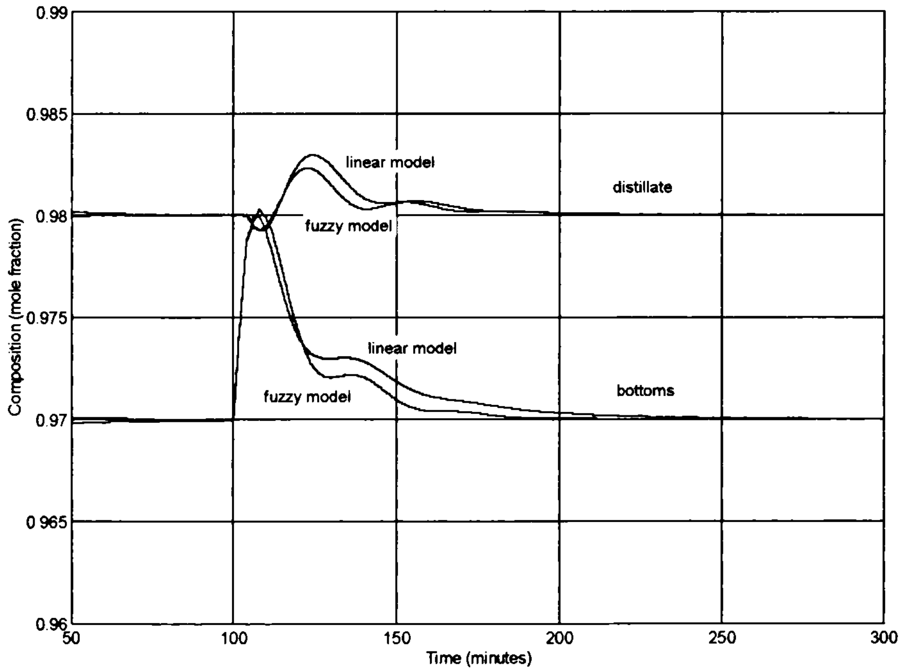


Figure 9.26: Comparison of process output responses to +10 percent feed flowrate change when using proposed controller with linear and 3-partition fuzzy MISO process models ($K_{f_1} = 1; K_{f_2} = 0.1$).

9.7 Conclusions

The modelling accuracy of fuzzy process models was found to be much better than linear process models. The preliminary studies carried out using SISO process models showed that the fuzzy model-based controller is able to cope quite well with the significant nonlinearities of distillate composition control and bottoms composition control over wide ranges. In both applications, the fuzzy model-based controllers outperformed comparable controllers using linear process models. It was also found that there was little difference in the performance of the fuzzy model-based controllers.

This study has also shown that the design of a 2-input 2-output MIMO fuzzy model-based predictive controller can be approached by decomposing it into two MISO fuzzy model-based controllers and computing the optimal controller outputs of the MISO fuzzy model-based controllers sequentially using separate objective functions. The resulting controller has been shown to be able to cope with process nonlinearities and interaction between loops. Here again, the performance of fuzzy model-based controllers has been shown to be superior to comparable controllers using linear process models.

In the design of our controller, we have not considered the measurement delays which can range from about 3-minutes in the case of binary distillation columns, to about 20-minutes in the case of multi-component distillation columns. Such delays can probably be better handled by using the dead time compensation scheme presented in Appendix B.

Chapter 10

CONTROL OF MULTIVARIABLE SYSTEMS: FORCED CIRCULATION EVAPORATOR SYSTEM

10.1 Introduction

An often used evaporator, known as the forced circulation evaporator, is shown in Figure 10.1. In this evaporator feed is mixed with a high volumetric flowrate of recirculating liquor and is pumped into a vertical heat exchanger. The heat exchanger is heated with steam which condenses on the outside of the tube walls. The liquor which passes up the inside of the tubes, boils and then passes to a separation vessel. In this vessel, liquid and vapour are separated. The liquid is recirculated with some being drawn off as product. The vapour is condensed by cooling using water as the coolant. Newell and Lee (1989) have derived a mathematical model of the forced circulation evaporator described above (Section 10.2). A simulation based on this mathematical model was used in all of our studies. The model was updated at 1-minute intervals using the Runge-Kutta third-order numerical integration method.

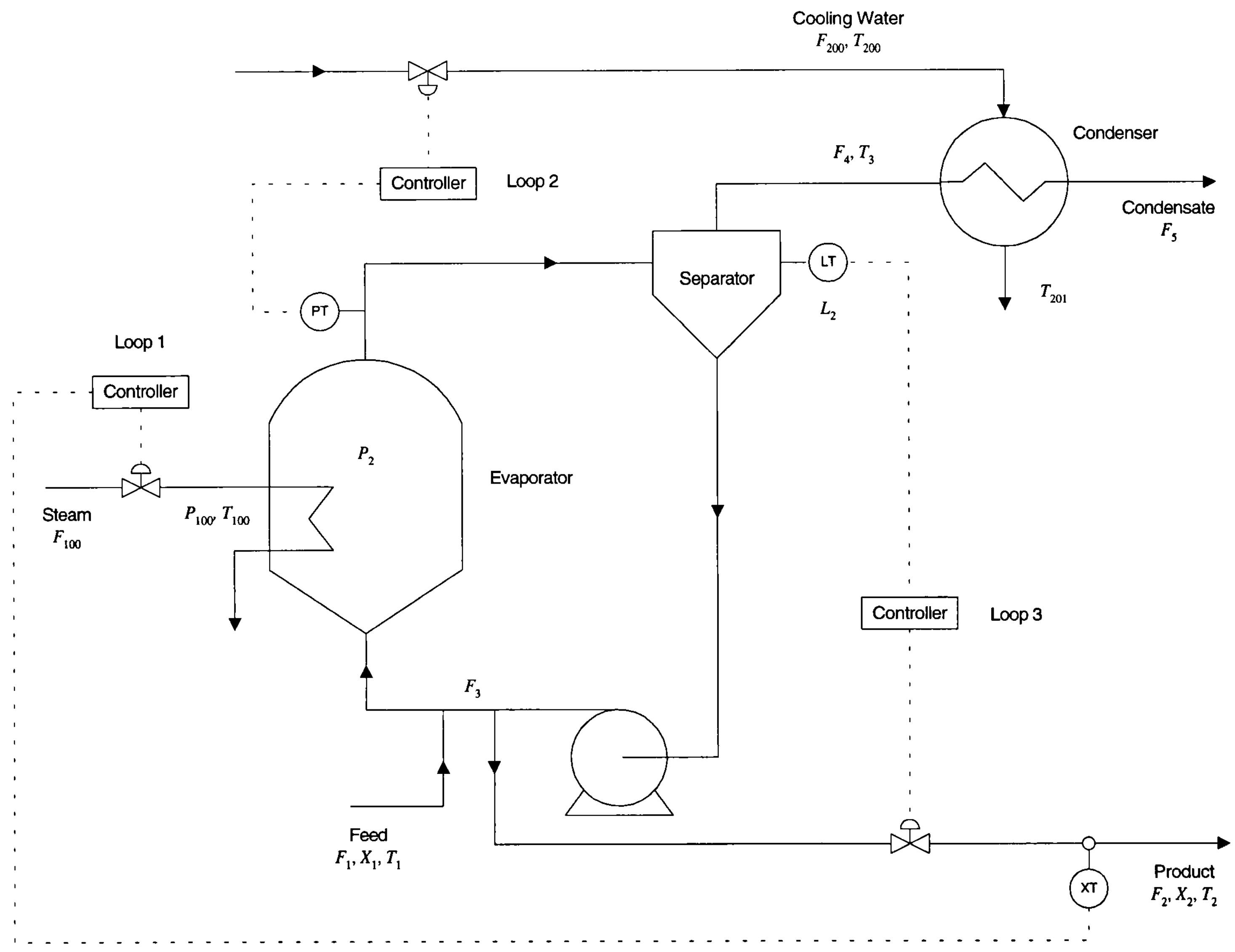


Figure 10.1: The forced circulation evaporator system.

10.2 Mathematical Model of Evaporator System

A forced circulation evaporator system is shown in Figure 10.1. This unit process, a common operation in many plants, represents a difficult control problem. The simulation as described below has been used in a number of other investigations (Newell and Lee, 1989; Lee and Sullivan, 1988; Wang and Cameron, 1994)

The variable names, descriptions, standard steady-state values, and engineering units are shown in Table 10.1 and Figure 10.1.

Process Liquid Mass Balance

A mass balance on the total process liquid (solvent and solute) in the system yields:

$$\frac{dL_2}{dt} = (F_1 - F_2 - F_4) / \rho A \qquad (10.1)$$

where ρ is the liquid density and A is the cross-sectional area of the separator. The product ρA is assumed to be constant at 20 kg/metre.

Process Liquid Solute Mass Balance

A mass balance on the solute in the process liquid phase yields:

$$\frac{dX_2}{dt} = (F_1 X_1 - F_2 X_2) / M \qquad (10.2)$$

where M is the amount of liquid in the evaporator and is assumed to be constant at 20 kg.

Process Vapour Mass Balance

A mass balance on the process vapour in the evaporator will express the total mass of the water vapour in terms of the pressure that exists in the system:

$$\frac{dP_2}{dt} = (F_4 - F_5) / C \qquad (10.3)$$

where C is a constant that converts the mass of vapour into an equivalent pressure and is assumed to have a value of 4 kg/kPa. This constant can be derived from the ideal gas law.

Process Liquid Energy Balance

The process liquid is assumed to always exist at its boiling point and to be perfectly mixed (assisted by the high circulation rate). The liquid temperature is:

$$T_2 = 0.5616 P_2 + 0.3126 X_2 + 48.43 \tag{10.4}$$

which is a linearization of the saturated liquid line for water about the standard steady-state value and includes a term to approximate boiling point elevation due to the presence of the solute.

The vapour temperature is:

$$T_3 = 0.507 P_2 + 55.0 \tag{10.5}$$

which is a linearization of the saturated liquid line for water about the standard steady-state value.

The dynamics of the energy balance are assumed to be very fast so that:

$$F_4 = (Q_{100} - F_1 C_p (T_2 - T_1)) / \lambda \tag{10.6}$$

where C_p is the heat capacity of the liquor and is assumed constant at a value of 0.07 kW/K(kg/min) and λ is the latent heat of vapourization of the liquor and is assumed to have a constant value of 38.5 kW/(kg/min).

The sensible heat change between T_2 and T_3 for F_4 is considered small compared to the latent heat. It is assumed that there are no heat losses to the environment or heat gains from the energy input of the pump.

Heater Steam Jacket

Steam pressure P_{100} is a manipulated variable which determines steam temperature under assumed saturated conditions. An equation relating steam temperature to steam pressure can be obtained by approximating the saturated steam temperature-pressure relationship by local linearization about the steady-state value:

$$T_{100} = 0.5138 P_{100} + 90.0 \tag{10.7}$$

The rate of heat transfer to the boiling process liquid is given by:

$$Q_{100} = UA_1 (T_{100} - T_2) \tag{10.8}$$

where UA_1 is the overall heat transfer coefficient times the heat transfer area and is a function of the total flowrate through the tubes in the evaporator:

$$UA_1 = 0.16(F_1 + F_3) \tag{10.9}$$

The steam flowrate is calculated from:

$$F_{100} = Q_{100} / \lambda_s \tag{10.10}$$

where λ_s is the latent heat of steam at the saturated conditions, assumed constant at a value of 36.6 kW/(kg/min).

The dynamics within the steam jacket have been assumed to be very fast.

Condenser

The cooling water flowrate F_{200} is a manipulated variable and the inlet temperature T_{200} is a disturbance variable. The cooling water energy balance yields:

$$Q_{200} = UA_2(T_3 - 0.5(T_{200} + T_{201})) \tag{10.11}$$

where UA_2 is the overall heat transfer coefficient times the heat transfer area, which is assumed constant with a value of 6.84 kW/K.

These two equations can be combined to eliminate T_{201} to give explicitly:

$$Q_{200} = \frac{UA_2(T_3 - T_{200})}{[1 + UA_2/(2C_pF_{200})]} \tag{10.12}$$

It follows that:

$$T_{201} = T_{200} + Q_{200}/(F_{200}C_p) \tag{10.13}$$

The condensate flowrate is:

$$F_5 = Q_{200}/\lambda \tag{10.14}$$

where λ is the latent heat of vapourization of water assumed constant at 38.5 kW/K(kg/min).

The dynamics within the condenser have been assumed to be very fast.

Slave Controllers

It has been assumed that controllers have been used to regulate the flow of the manipulated variables. The setpoints of these flow controllers is provided by the controllers shown in Figure 10.1. The dynamics introduced by these slave controllers on the manipulated variables F_2, P_{100} and F_{200} have been approximated by first-order lags with time constants of 1.2 minutes.

Variable	Description	Value	Units
F_1	Feed flowrate	10.0	kg/min
F_2	Product flowrate	2.0	kg/min
F_3	Circulating flowrate	50.0	kg/min
F_4	Vapour flowrate	8.0	kg/min
F_5	Condensate flowrate	8.0	kg/min
X_1	Feed composition	5.0	percent
X_2	Product composition	25.0	percent
T_1	Feed temperature	40.0	deg C
T_2	Product temperature	84.6	deg C
T_3	Vapour temperature	80.6	deg C
L_2	Separator level	1.0	metres
P_2	Operating pressure	50.5	kPa
F_{100}	Steam flowrate	9.3	kg/min
T_{100}	Steam temperature	119.9	deg C
P_{100}	Steam pressure	194.7	kPa
Q_{100}	Heater duty	339.0	kW
F_{200}	Cooling water flowrate	208.0	kg/min
T_{200}	Cooling water inlet temperature	25.0	deg C
T_{201}	Cooling water outlet temperature	46.1	deg C
Q_{200}	Condenser duty	307.9	kW

Table 10.1: Evaporator Variables

10.3 Control Problem

Figure 10.1 shows the important feedback loops of a forced circulation evaporator system. Pressure and level are controlled by manipulating the cooling water flowrate and product discharge rate respectively. Product composition is controlled by manipulating the steam pressure. To allow open loop tests to be carried out on the system, it is first necessary to control the level in the separation vessel. This is to ensure that the remainder of the evaporator system is open-loop bounded. A PID controller was therefore used for Loop 3 in Figure 10.1.

The open loop response of product composition and operating pressure to step changes in steam pressure and cooling water flowrate introduced 50 minutes after the start of a simulation is shown in Figures 10.2 to 10.5. It can be seen that the relationship between input and output variables is generally nonlinear and there is interaction between variables. Figure 10.6 shows the open loop response between product composition and steam pressure with a second PID controller used for Loop 2.

It is important that the product composition is maintained as close as possible to the desired value. The problem is investigated here using setpoint changes in the product composition in the range between 15 and 35 percent and feed flowrate changes (i.e. load changes) of 10 percent. Good performance is also desired from the controllers used for level and operating pressure. These secondary controllers should be able to cope with the above-mentioned changes taking into account the process nonlinarities and interaction between loops highlighted above.

The controller that will be emphasized in our investigations is the fuzzy model-based multi-step predictive controller with the control horizon set to 1-step. SISO model-based control of all three loops mentioned above will first be attempted. Next, we will attempt to extend the SISO model used for product composition control to include feed flowrate as an additional input and then evaluate the controller based on this MISO process model using feed flowrate changes to gauge the extent of improvement possible through feed-forward compensation of disturbance inputs. In the third stage, we will attempt to design a 2-input 2-output multivariable controller for product composition and operating pressure which allows coupling of control loops to improve performance. Coupling will be achieved by expanding the SISO process models used by each loop to include the the manipulated variable of the other loop as a model input. The optimal controller output of the two MISO process model-based controllers will be computed sequentially using separate objective functions. Close examination of Figures 10.2, 10.4 and 10.6 provides visible evidence of dead time which is likely to introduce problems in the control of product composition. In the final stage, we will therefore examine the improvement that is possible through dead time compensation.

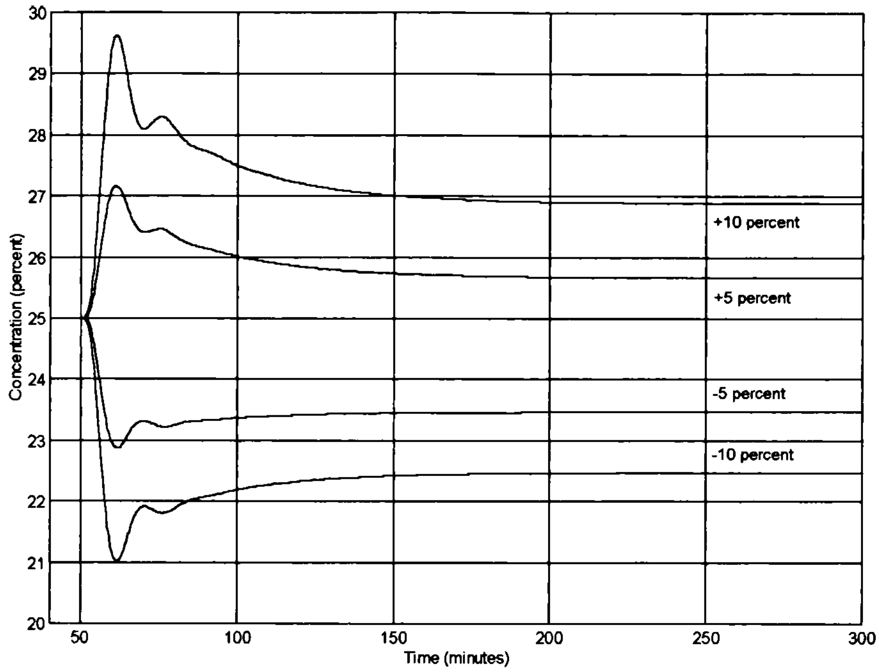


Figure 10.2: Open loop response of the product composition to step changes in steam pressure.

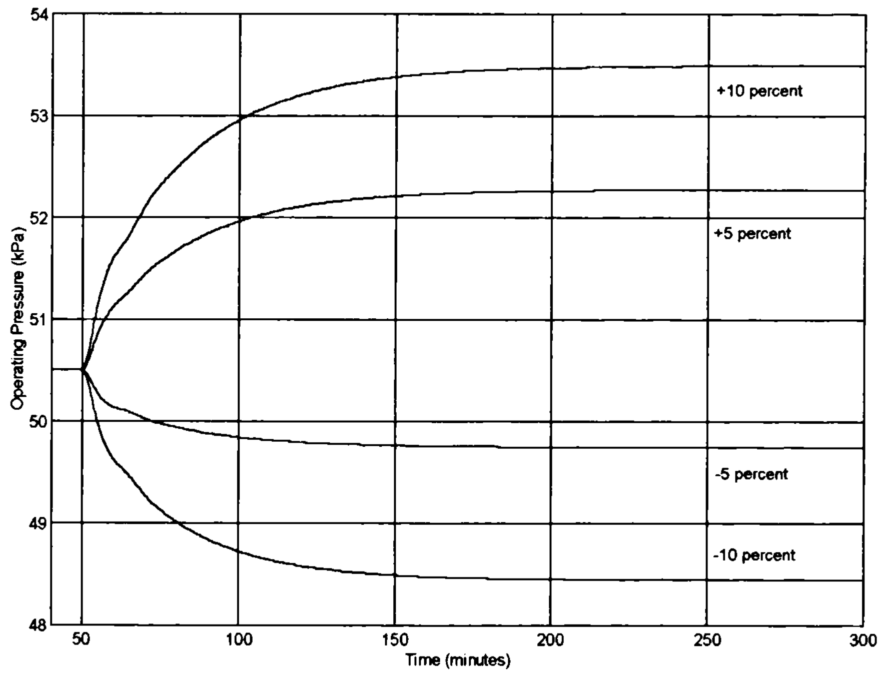


Figure 10.3: Open loop response of the operating pressure to step changes in steam pressure.

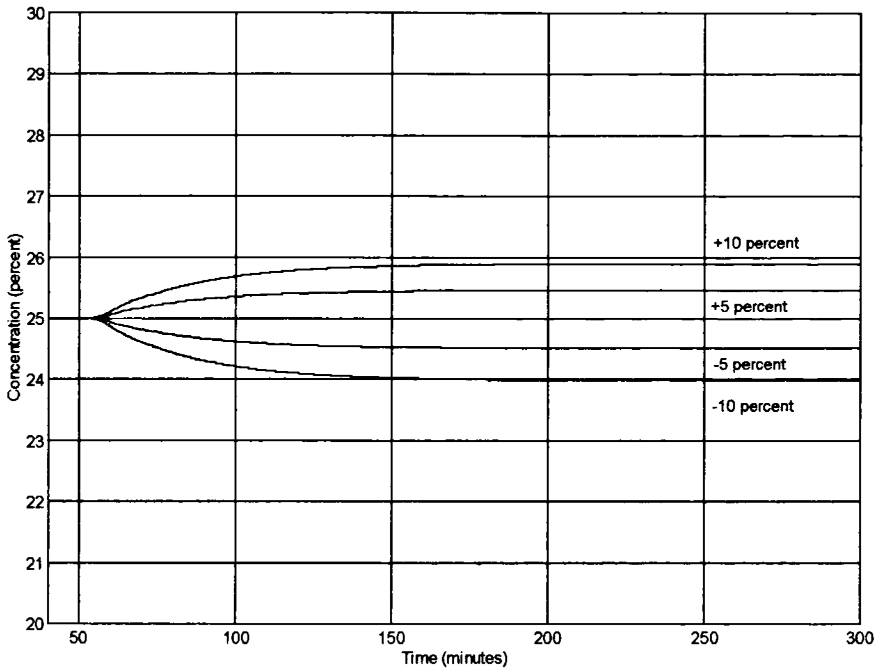


Figure 10.4: Open loop response of the product composition to step changes in cooling water flowrate.

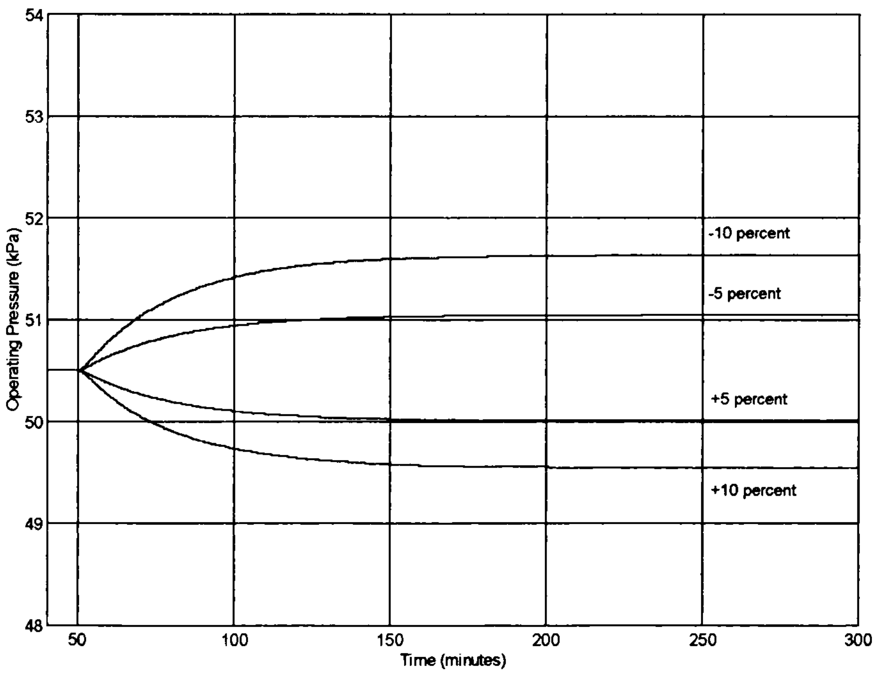


Figure 10.5: Open loop response of the operating pressure to step changes in cooling water flowrate.

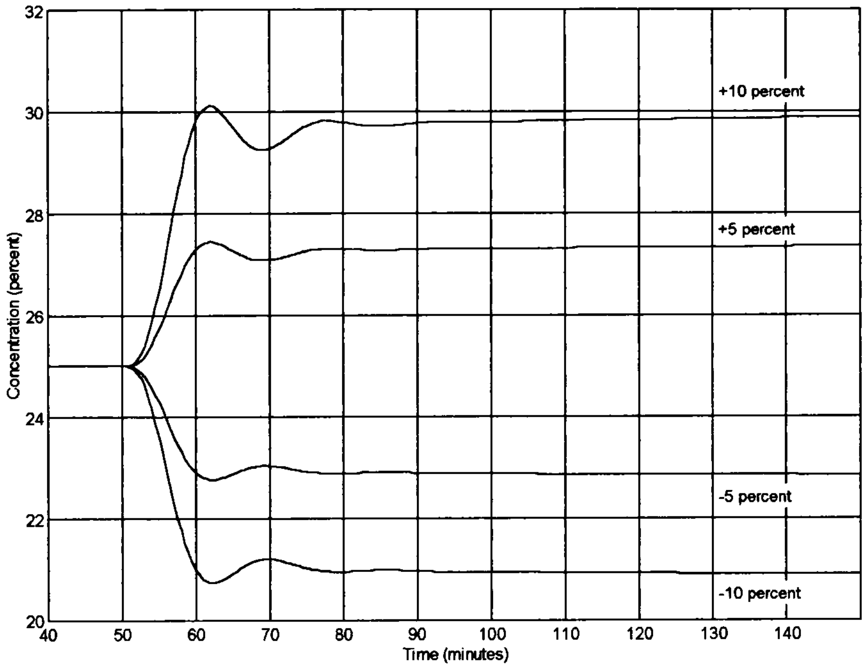


Figure 10.6: Open loop response of the product composition to step changes in steam pressure with Control Loops 2 and 3 implemented.

10.4 Identification of SISO Process Models

10.4.1 SISO Model for Product Composition Control

Data for modelling was generated by applying 50 random step changes, each lasting 20 minutes, in the steam pressure such that the product composition remained approximately within 15 and 35 percent (Figure 10.8). Sampling was carried out at 1-minute intervals. A total of 1000 data points were used for the identification.

An assumption has been made that the input/output relationship between product composition and steam pressure can be approximated by the following second-order model structure:

$$\begin{aligned} R^i:\ if\, X_2(t)\ is\ A^i\ then\, X_2(t+1) &= a_1^i X_2(t) + a_2^i X_2(t-1) \\ &+ b_1^i P_{100}(t) + b_2^i P_{100}(t-1) + k_i, \quad i = 1,\ldots,p \end{aligned} \tag{10.15}$$

where $X_2(t)$ and $P_{100}(t)$ are the product composition and steam pressure at time t, respectively.

Table 10.2 compares the modelling accuracy of the linear model with fuzzy models derived using different numbers of fuzzy partitions as shown in Figure 10.7. Figure 10.7 also shows the fuzzy partitioning points used by the models. It will be noticed from Table 10.2 that the inclusion of a constant term leads to considerable improvement in the modelling acuracy of the linear process model, but little improvement is achieved in the case of the fuzzy process models. Also evident from Table 10.2 is the slightly better modelling accuracy of the fuzzy process models over the linear process model which includes the constant term, and the small difference between the modelling accuracies of fuzzy process models. This is probably attributable to the fact that the system being modelled is not highly nonlinear as is evident from the open-loop response (Figure 10.6).

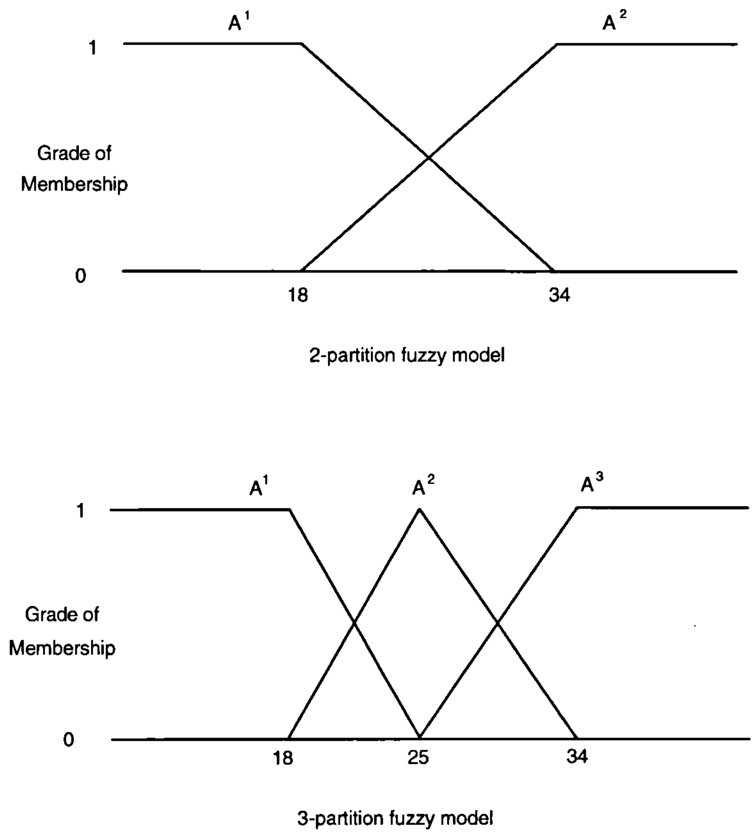
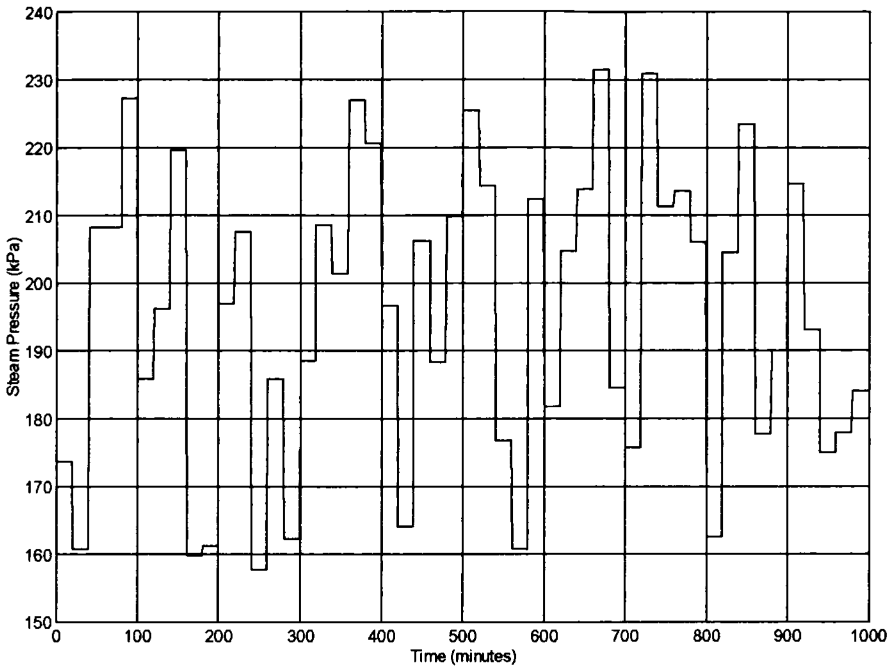


Figure 10.7: Fuzzy partitioning of the input space used for deriving the SISO process models in Table 10.2.



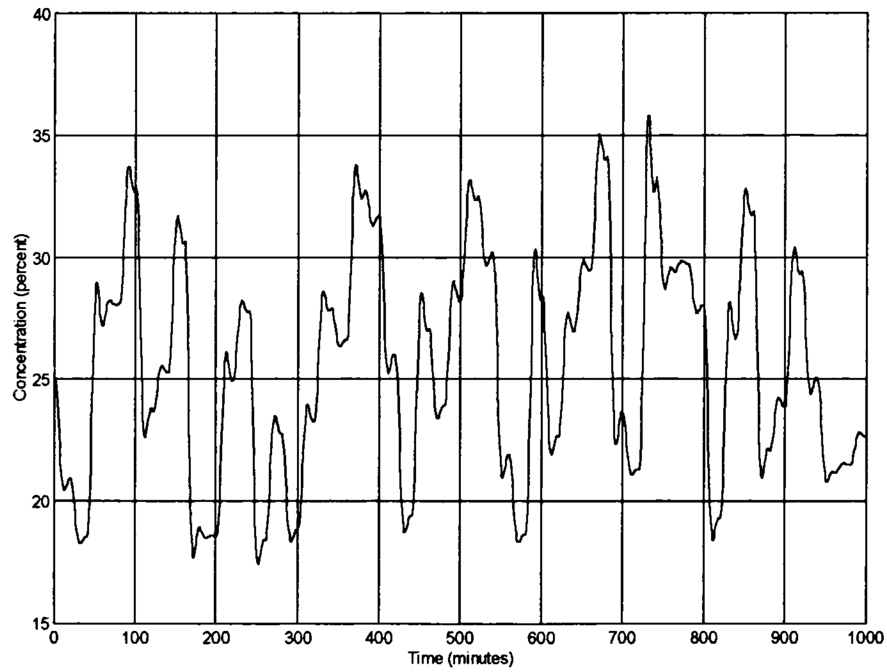


Figure 10.8: Input/output data utilised for identification of the SISO process model for product composition control loop.

No.	Type of model	Model parameters					Mean square error
		a_1^i	a_2^i	b_1^i	b_2^i	k_i	
1	Linear model	1.8448	-0.8877	-0.0006	0.0063	-	0.0112
2	Linear model (with constant term)	1.7369	-0.7956	0.0008	0.0111	-0.8343	0.0059
3	2-partition fuzzy model	1.6683 1.7219	-0.7571 -0.8236	0.0009 0.0009	0.0093 0.0139	- -	0.0052
4	2-partition fuzzy model (with constant term)	1.5647 1.6021	-0.7672 -0.8173	0.0006 0.0012	0.0088 0.0142	2.1779 3.7467	0.0050
5	3-partition fuzzy model	1.6596 1.6986 1.6912	-0.7338 -0.7998 -0.7914	0.0002 0.0018 0.0003	0.0084 0.0112 0.0144	- - -	0.0050
6	3-partition fuzzy model (with constant term)	1.5465 1.5491 1.5930	-0.7484 -0.7816 -0.8110	0.0000 0.0019 0.0004	0.0076 0.0125 0.0128	2.4492 3.0327 4.3573	0.0047

Table 10.2: Effect of the number of fuzzy partitions of the input space on model parameters and modelling accuracy of SISO process model used for product composition control.

10.4.2 SISO Model for Operating Pressure Control

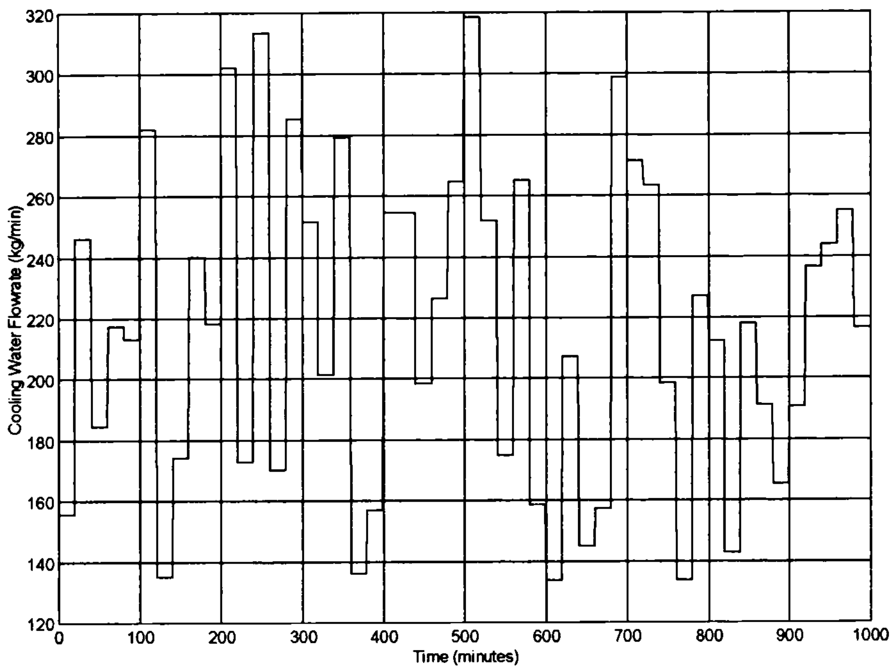
Data for modelling was generated by applying 50 random step changes, each lasting 20 minutes, in the water flowrate such that the operating pressure remained approximately within 45 and 55 kPa (Figure 10.9) and sampling was carried out at 1-minute intervals. A total of 1000 data points were used for identification.

Since the open-loop response between operating pressure and water flowrate (Figure 10.5) can be approximated by a first-order system, the following model structure has been assumed:

$$R^i: \ if\, P_2(t)\ is\ A^i\ then\ P_2(t+1) = a_1^i P_2(t) + b_1^i F_{200}(t) + k_i,\ \ i = 1,\ldots,p \qquad (10.16)$$

where $P_2(t)$ and $F_{200}(t)$ are the operating pressure and water flowrate at time t, respectively.

Table 10.3 compares the modelling accuracy of the linear model with fuzzy models derived using different numbers of fuzzy partitions as shown in Figure 10.10. Figure 10.10 also shows the fuzzy partitioning points used by the models. The performance of the different models appears to follow similar trends to the SISO models used for product composition earlier, and therefore needs no further elaboration here.



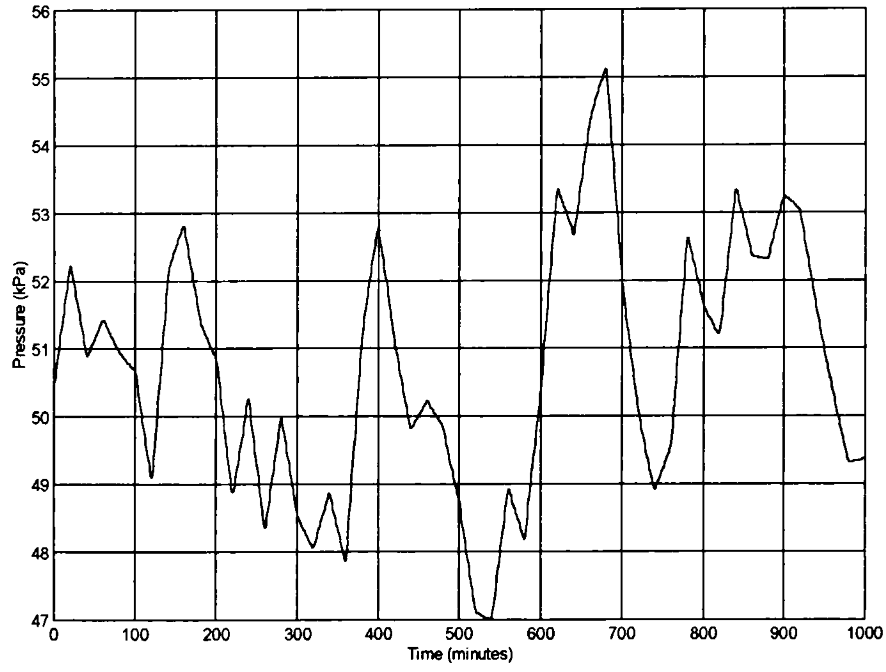


Figure 10.9: Input/output data utilised for identification of the SISO process model for operating pressure control loop.

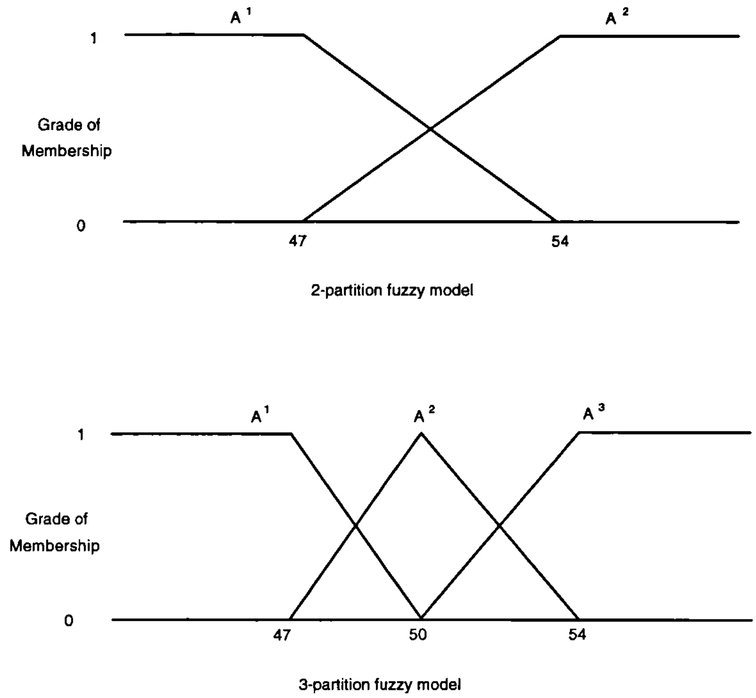


Figure 10.10: Fuzzy partitioning of the input space used for deriving the SISO process models in Table 10.3.

No.	Type of model	Model parameters			MSE
		a_1^i	b_1^i	k_i	
1	linear	1.0052	-0.0012	-	0.00150
2	linear (with constant term)	0.9858	-0.0015	1.0473	0.00043
3	2-partition fuzzy	1.0068 1.0062	-0.0012 -0.0018	- -	0.00037
4	2-partition fuzzy (with constant term)	0.9981 0.9960	-0.0012 -0.0018	0.4018 0.5517	0.00037
5	3-partition fuzzy	1.0056 1.0070 1.0059	-0.0010 -0.0016 -0.0017	- - -	0.00036
6	3-partition fuzzy (with constant term)	0.9729 0.9702 0.9824	-0.0010 -0.0016 -0.0017	1.5187 1.8477 1.2778	0.00035

Table 10.3: Effect of number of fuzzy partitions on model parameters and modelling accuracy of SISO process model used for operating pressure control.

10.4.3 SISO Model for Separator Level Control

Identification of the process model for separator level control presents some special problems. The liquid level control loop is not self-regulatory and therefore open loop tests cannot be carried out to determine the process model as in the previous two cases. The following first-order model structure has been assumed (i.e., no constant term was included):

$$R^i: \ if \ L_2(t) \ is \ A^i \ then \ L_2(t+1) = a_1^i L_2(t) + b_1^i F_2(t), \ i = 1,\dots,p \tag{10.17}$$

where $L_2(t)$ and $F_2(t)$ are the separator level and product flowrate at time t, respectively.

The problem of identifying the model parameters has been approached by first implementing a PID controller for this control loop. A series of random step changes were introduced while ensuring that the level in the tank remained in the range 0 to 2 metres, and input/output data was collected from the closed loop response at 1-minute intervals. The setpoint changes were introduced sufficiently frequently to prevent the system spending too much time at steady-state conditions. The data was used for identification of a crude 3-partition fuzzy process model. Figure 10.11 shows the fuzzy partitioning points used by the model. This crude process model was then used to implement an adaptive fuzzy model-based predictive controller which used the recursive least squares algorithm for fine-tuning the model parameters. Details of this method are presented in Appendix A and will therefore not be elaborated further here. Random step changes in the range 0 to 2 m were once again used during the fine-tuning. Adaptation was stopped when the controller's performance was considered to be sufficiently good. The final model parameters were as follows:

$$\Phi = \begin{bmatrix} 1.2750 & -0.0748 \\ 1.0867 & -0.0470 \\ 1.0542 & -0.0550 \end{bmatrix} \tag{10.18}$$

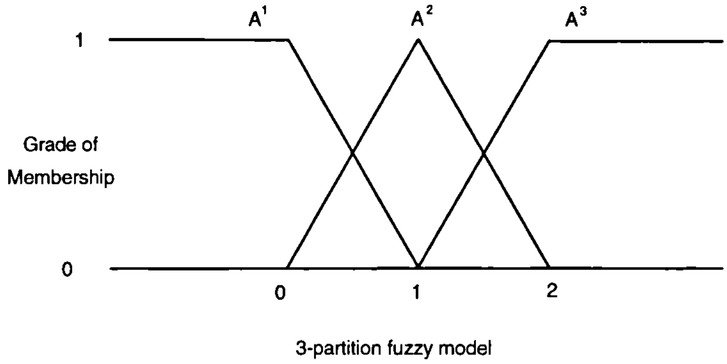


Figure 10.11: Fuzzy partitioning of the input space used for deriving the SISO process model for level control loop.

10.5 SISO Controllers

All studies in this section are based on fuzzy process models without the constant term since better controller performance seemed to be achieved without including this term.

Figures 10.12 to 10.15 examine the performance of the controllers to a series of setpoint changes in the product composition on both sides of the steady-state level. The number of steps in the prediction horizon of the controllers, n_i, is indicated in brackets together with the values of the feedback filter gain of Filter 2, $K_{f_{2i}}$. The subscript, i, is used to indicate the loop number as shown in Figure 10.1. The prediction horizon of the controllers have been set to approximately the optimum values identified from earlier studies on individual loops. The big prediction horizon needed is quite characteristic of systems with dead time. The plot of operating pressure has been displaced downwards (i.e., plot shows $P_2 - 205$) and the plot of separator level has been displaced upwards (i.e., plot shows $L_2 + 19$) to allow plotting alongside with product composition.

There is only a slight difference between the performances of controllers using 2-partition and 3-partition fuzzy models. Close examination reveals that the performance may be slightly better with the 2-partition fuzzy model when the product composition is about 25 percent and higher. Figure 10.14 shows very poor performance when using the linear model without the constant term for Loops 1 and 2. The process model used by Loop 3 was still the same 3-partition fuzzy model. Figure 10.15 shows that considerable improvement can be achieved by using the linear model with the constant term included. It is obvious, however, that the output response below 25 percent is quite sluggish and the time required for the steady-state offset correction can be quite considerable.

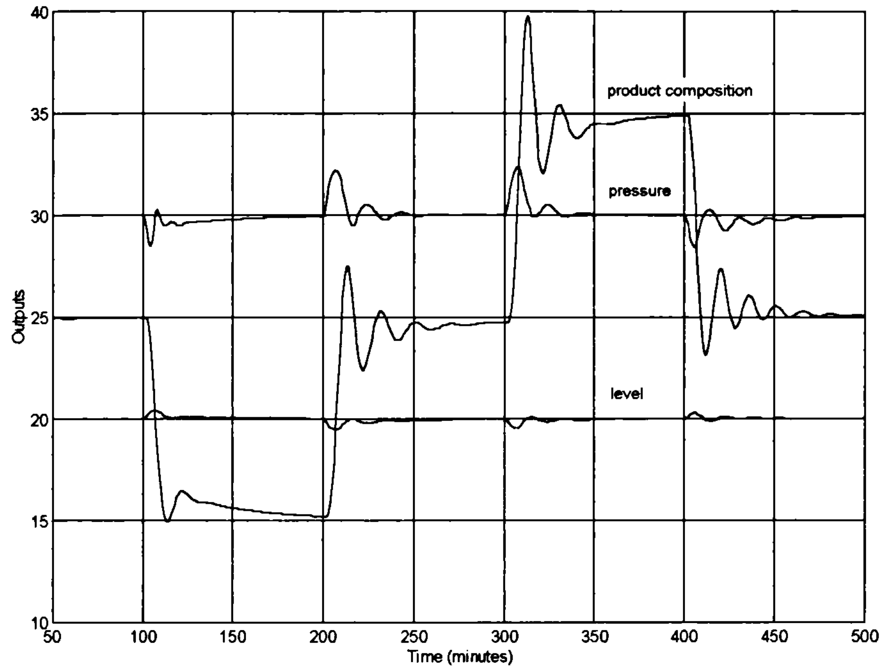


Figure 10.12: Process output responses to setpoint changes in product composition when using 2-partition fuzzy SISO process models ($n_1 = 10; n_2 = 10; n_3 = 7; K_{f_{21}} = 0.025; K_{f_{22}} = 0.05; K_{f_{23}} = 0.05$).

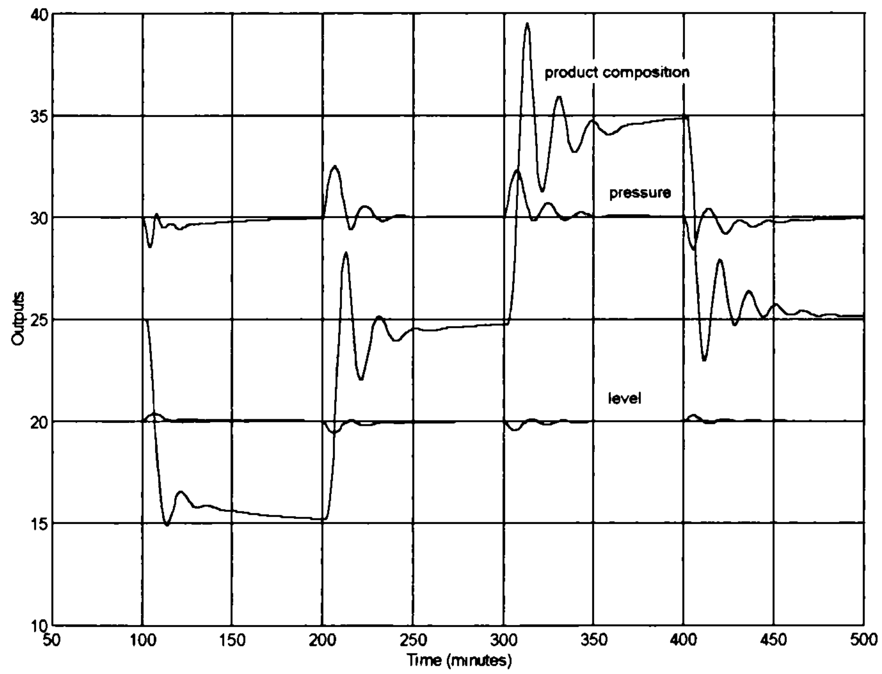


Figure 10.13: Process output responses to setpoint changes in product composition when using 3-partition fuzzy SISO process models ($n_1 = 10; n_2 = 10; n_3 = 7; K_{f_{21}} = 0.025; K_{f_{22}} = 0.05; K_{f_{23}} = 0.05$).

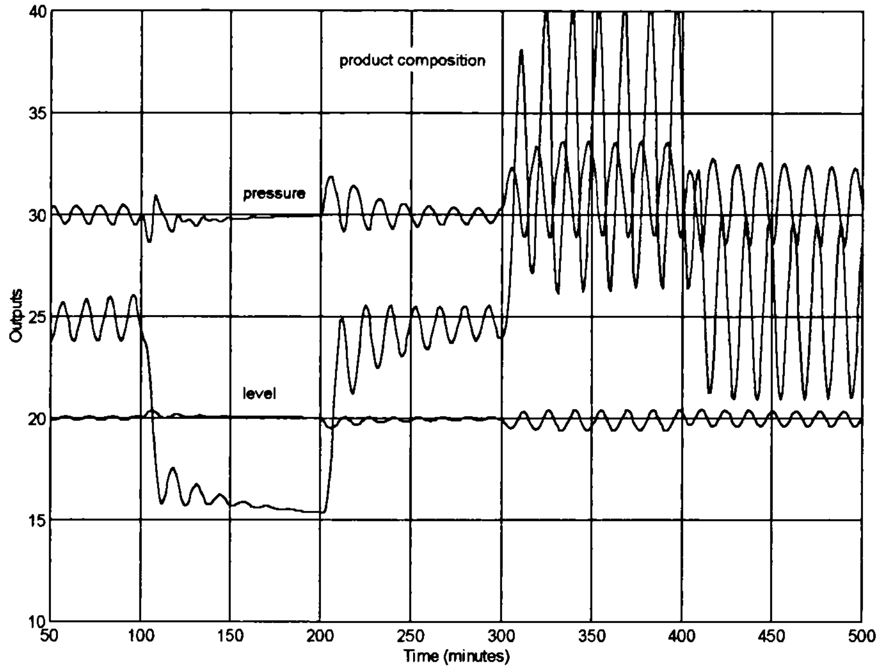


Figure 10.14: Process output responses to setpoint changes in product composition when using linear SISO models without constant for Loops 1 & 2 $(n_1 = 10; n_2 = 10; n_3 = 7; K_{f_{21}} = 0.025; K_{f_{22}} = 0.05; K_{f_{23}} = 0.05)$.

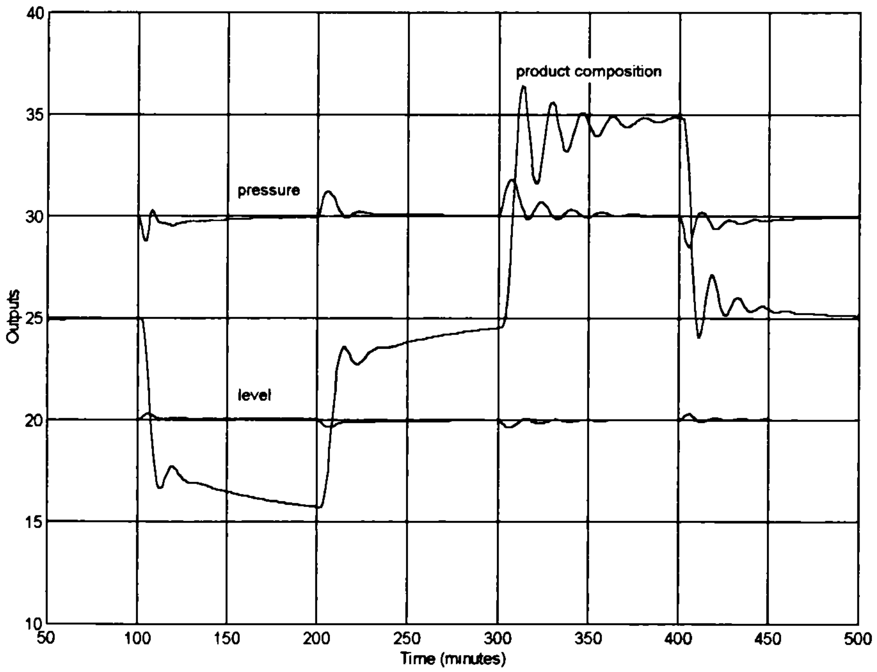


Figure 10.15: Process output responses to setpoint changes in product composition when using linear SISO models with constant for Loops 1 & 2 $(n_1 = 10; n_2 = 10; n_3 = 7; K_{f_{21}} = 0.025; K_{f_{22}} = 0.05; K_{f_{23}} = 0.05)$.

10.6 MISO Controller for Compensation of Feed Flowrate Disturbance

10.6.1 Identification of MISO Process Model

We next address the development of a MISO process model-based controller for compensation of disturbance inputs by extending the SISO process model used for product composition control.

Data for modelling was generated by applying 50 random step changes, each lasting 20 minutes, in the steam pressure and feed flowrate as shown in Figure 10.16 such that the product composition remained approximately within 15 and 35 percent. The fluctuations allowed in feed flowrate had a comparatively smaller effect on the product concentration as compared to the fluctuations in steam pressure. Sampling was carried out at 1-minute intervals. A total of 1000 data points were used for the identification.

It was assumed that the input/output response can be approximated by the following second-order model structure:

$$R^i: \ if\, X_2(t)\ is\ A^i\ then\, X_2(t+1) = a_1^i X_2(t) + a_2^i X_2(t-1) + b_1^i P_{100}(t) + b_2^i P_{100}(t-1) + c_1^i F_1(t) + c_2^i F_1(t-1), \quad i = 1,\ldots,p \tag{10.19}$$

where $X_2(t)$, $P_{100}(t)$ and $F_1(t)$ are the product composition, steam pressure and feed flowrate at time t, respectively. Only the 3-partition fuzzy process model was examined and the fuzzy partitioning was maintained the same as for the SISO process model (Figure 10.7). The model parameters were:

$$\Phi = \begin{bmatrix} 1.7910 & -0.7622 & -0.0006 & 0.0092 & -0.0350 & -0.1452 \\ 1.8079 & -0.7947 & +0.0027 & 0.0105 & -0.0325 & -0.2567 \\ 1.7974 & -0.8118 & -0.0010 & 0.0155 & -0.0594 & -0.2391 \end{bmatrix} \tag{10.20}$$

10.6.2 MISO Process Model-Based Controller

The performance of the controller using the process model identified above was examined to gauge the improvement, if any, to feed flowrate changes. The controller parameters were maintained the same as for the SISO model-based controller. Comparison of Figures 10.17 and 10.18 shows considerable improvement by using the MISO process model instead of the SISO process model.

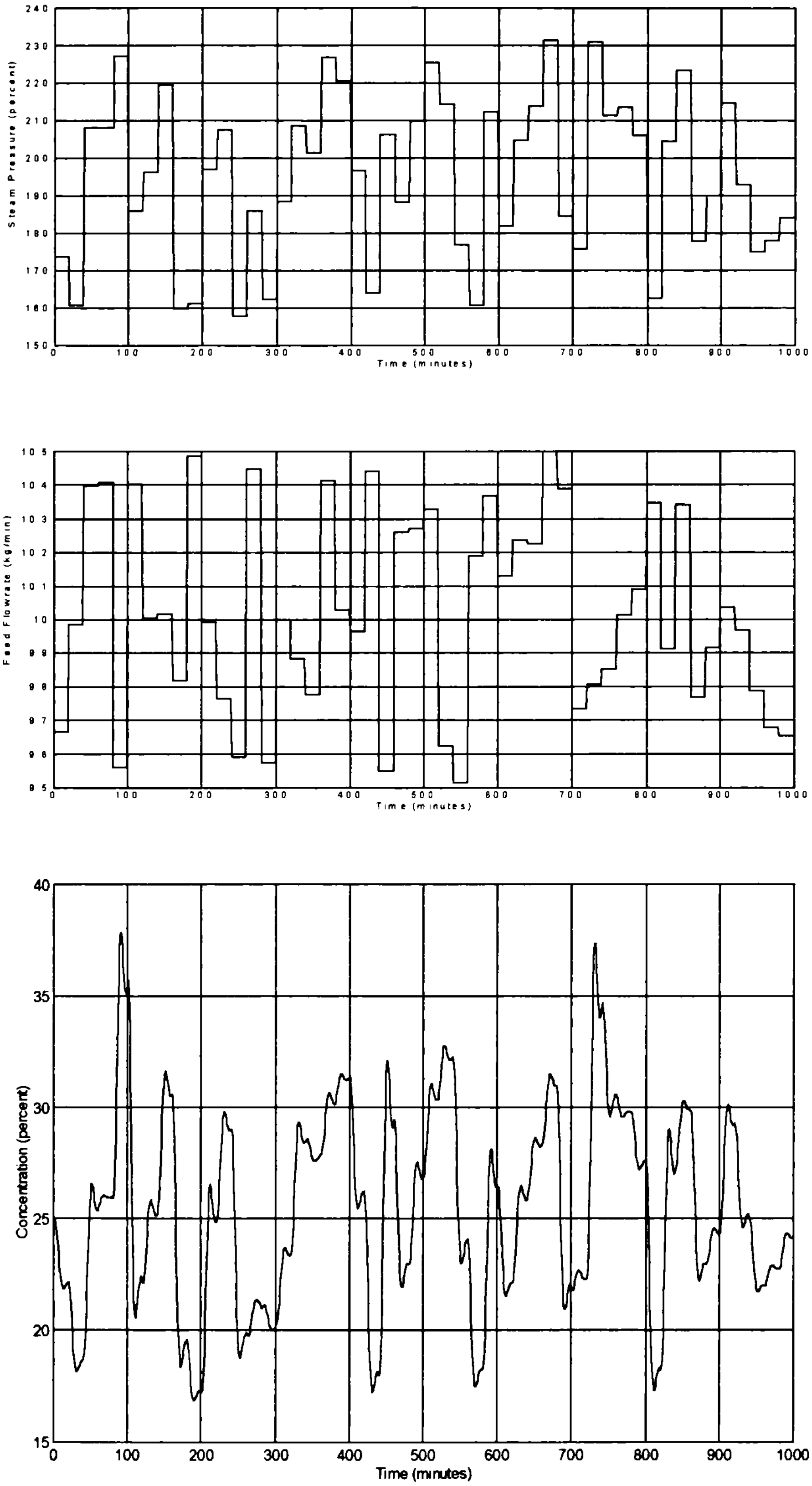


Figure 10.16: Input/output data utilised for identification of the MISO process model for feed flowrate disturbance compensation.

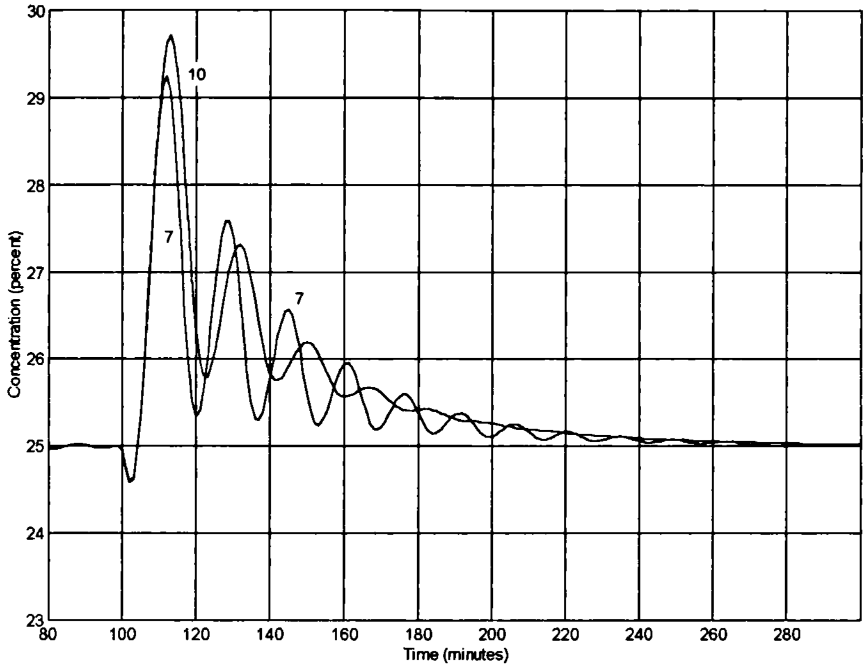


Figure 10.17: Effect of prediction horizon on product composition response to -10 percent change in feed flowrate when using 3-partition fuzzy SISO process model in Loop 1 $(n_2 = 10; n_3 = 7; K_{f_{21}} = 0.025; K_{f_{22}} = 0.05; K_{f_{23}} = 0.05)$.

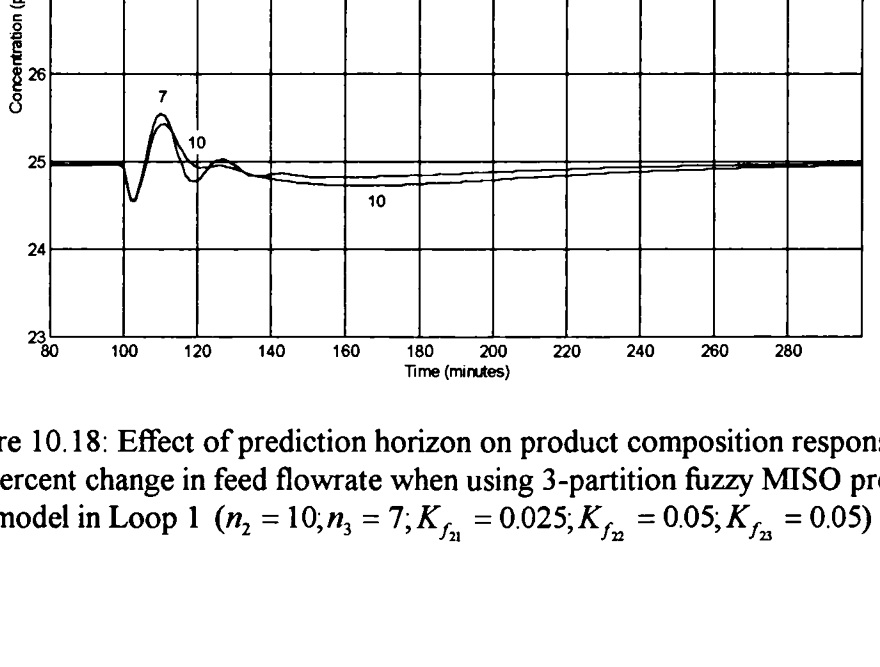


Figure 10.18: Effect of prediction horizon on product composition response to -10 percent change in feed flowrate when using 3-partition fuzzy MISO process model in Loop 1 $(n_2 = 10; n_3 = 7; K_{f_{21}} = 0.025; K_{f_{22}} = 0.05; K_{f_{23}} = 0.05)$.

10.7 MIMO Controller for Product Composition and Operating Pressure

10.7.1 Identification of MISO Process Model for Product Composition Control

The SISO process model for product composition control was expanded to include the manipulated variable used for operating pressure control, i.e., the cooling water flowrate.

Data for modelling was generated by applying 50 random step changes, each lasting 20 minutes, in the steam pressure and cooling water flowrate as shown in Figure 10.19 such that the product composition remained approximately within 15 and 35 percent. The fluctuations allowed in water flowrate had a comparatively smaller effect on the product concentration as compared to the fluctuations in steam pressure. Sampling was carried out at 1-minute intervals. A total of 1000 data points were used for the identification.

It was assumed that the input/output response can be approximated by using the following second-order model structure:

$$R^i: \ if\, X_2(t) \ is\, A^i \ then\, X_2(t+1) = a_1^i X_2(t) + a_2^i X_2(t-1) + b_1^i P_{100}(t) + b_2^i P_{100}(t-1) + c_1^i F_{200}(t) + c_2^i F_{200}(t-1), \quad i = 1,\ldots,p \tag{10.21}$$

where $X_2(t)$, $P_{100}(t)$ and $F_{200}(t)$ are the product composition, the steam pressure and the cooling water flowrate at time t, respectively.

Table 10.4 compares the modelling accuracy of the linear model with fuzzy models using the same fuzzy partitioning as for SISO process models (Figure 10.7). Here again, the performance of the different process models appears to follow a similar trend to the SISO process models used for product composition earlier.

10.7.2 Identification of MISO Process Model for Operating Pressure Control

The SISO process model for operating pressure control was expanded to include the manipulated variable used for product composition control, i.e., the steam pressure.

No.	Type of model	Model parameters							Mean square error
		a_1^i	a_2^i	b_1^i	b_2^i	c_1^i	c_2^i	k_i	
1	Linear	1.8124	-0.8758	0.0011	0.0086	-0.0014	0.0002	-	0.0079
2	Linear (with constant term)	1.7766	-0.8463	0.0016	0.0103	-0.0004	0.0010	-0.6727	0.0058
3	2-partition fuzzy	1.7441 1.7514	-0.8293 -0.8617	0.0016 0.0018	0.0079 0.0132	-0.0002 -0.0007	0.0009 0.0013	- -	0.0054
5	3-partition fuzzy	1.6287 1.8201 1.6678	-0.7133 -0.9076 -0.7883	0.0003 0.0029 0.0006	0.0098 0.0078 0.0156	-0.0002 -0.0007 0.0000	0.0004 0.0013 0.0010	- - -	0.0049

Table 10.4: Effect of the number of fuzzy partitions of the input space on model parameters and modelling accuracy of MISO process model used for product composition control.

Data for modelling was generated by applying 50 random step changes, each lasting 20 minutes, in the cooling water flowrate and steam pressure as shown in Figure 10.20 such that the operating pressure remained approximately within 45 and 55 percent. The fluctuations allowed in steam pressure had a comparatively smaller effect on the operating pressure as compared to the fluctuations in water flowrate. Sampling was carried out at 1-minute intervals. A total of 1000 data points were used for the identification.

It was assumed that the input/output response can be approximated by using the following first-order model structure:

$$R^i:\ if\ P_2(t)\ is\ A^i\ then\ P_2(t+1) = a_1^i P_2(t) + b_1^i F_{200}(t) + c_1^i P_{100}(t),\quad i = 1,\dots,p \qquad (10.22)$$

where $P_2(t)$, $F_{200}(t)$ and $P_{100}(t)$ are the operating pressure, the cooling water flowrate and the steam pressure at time *t*, respectively.

Table 10.5 compares the modelling accuracy of the linear model with fuzzy models using the same fuzzy partitioning as for SISO process models (Figure 10.10). Here again, the performance of the different process models appears to follow a similar trend to the SISO process models used for product composition earlier.

No.	Type of model	Model parameters				MSE
		a_1^i	b_1^i	c_1^i	k_i	
1	linear	0.9841	-0.0013	0.0056	-	0.00087
2	linear (with constant)	0.9694	-0.0014	0.0045	0.9707	0.00060
3	2-partition fuzzy	0.9861 0.9909	-0.0011 -0.0017	0.0052 0.0039	- -	0.00056
4	3-partition fuzzy	0.9959 0.9855 0.9981	-0.0011 -0.0014 -0.0017	0.0027 0.0054 0.0020	- - -	0.00054

Table 10.5: Effect of the number of fuzzy partitions of the input space on model parameters and modelling accuracy of MISO process model used for operating pressure control.

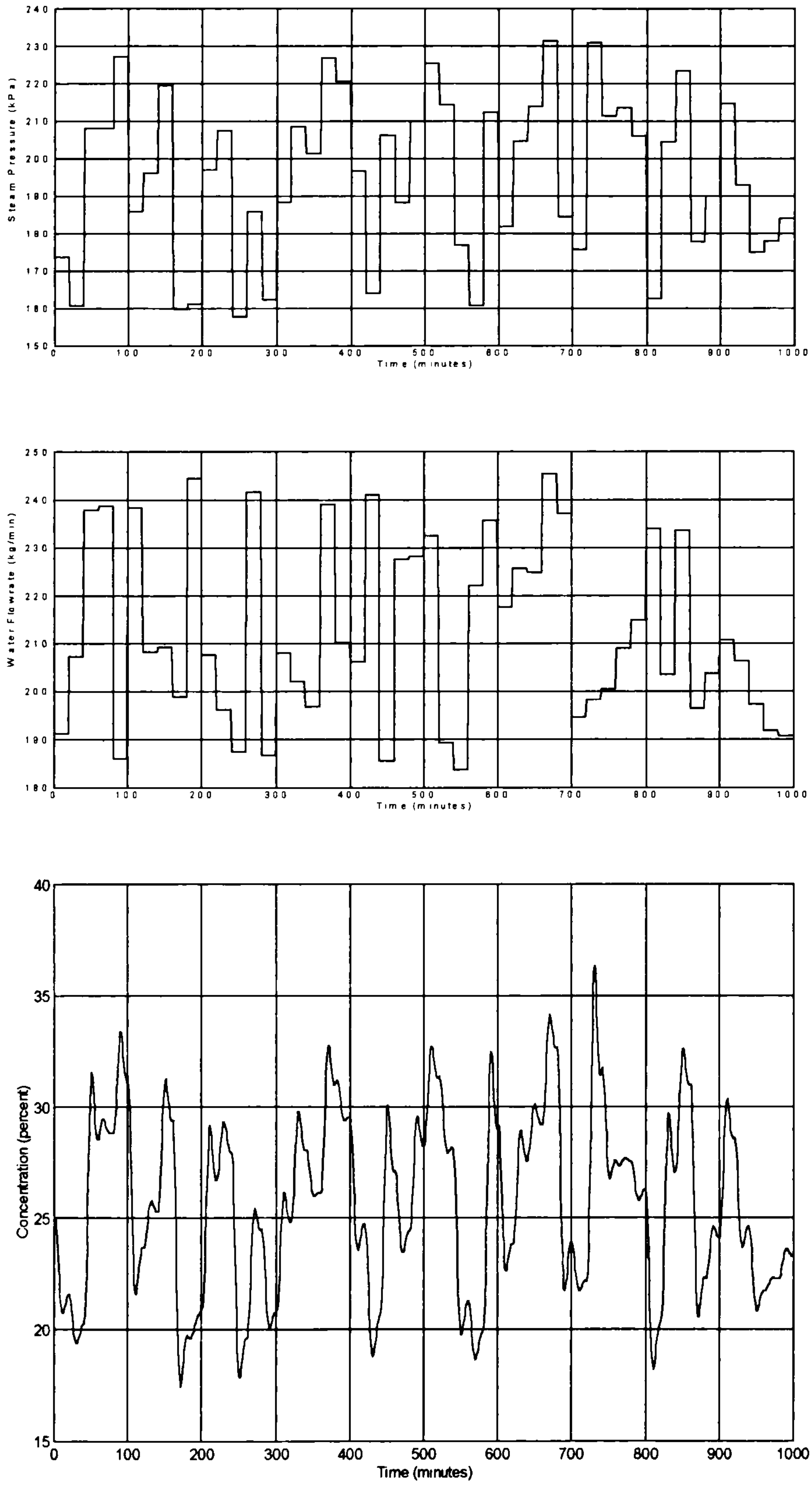


Figure 10.19: Input/output data utilised for identification of the MISO process model for product composition control loop.

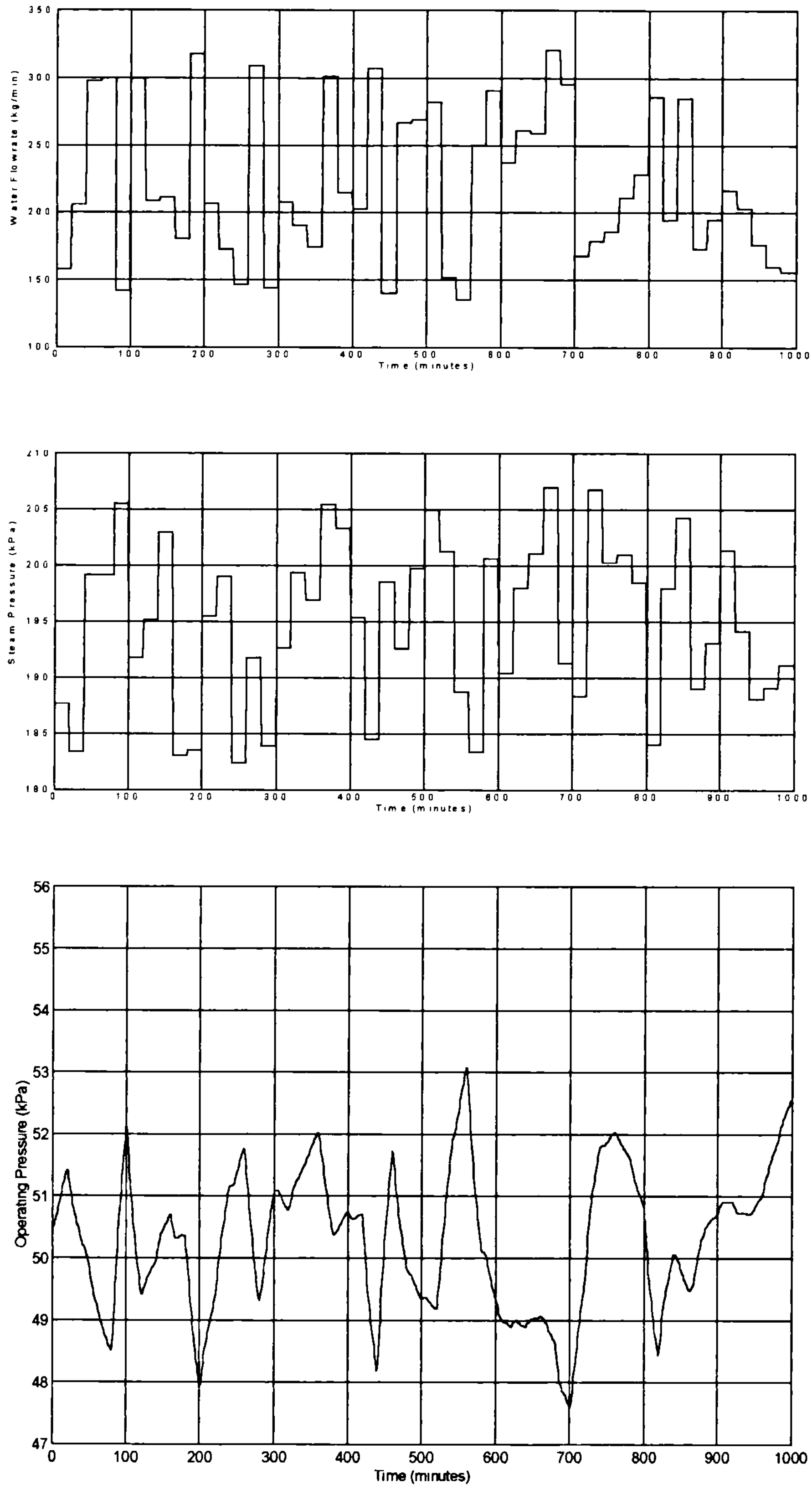


Figure 10.20: Input/output data utilised for identification of the MISO process model for operating pressure control loop.

10.7.3 MIMO Controller

As mentioned earlier, the design of the MIMO controller has been approached by decomposing it into two MISO process model-based controllers and computing the optimal controller output for the MISO process model-based controllers sequentially using separate objective functions.

Figures 10.21 and 10.22 examine the performance of our proposed controller to a series of setpoint changes in the product composition on both sides of the steady-state level when using the MISO process models identified above for Loops 1 and 2. The parameters used by the controller was maintained the same as for the SISO controllers in earlier studies. The process model used by Loop 3 was the 3-partition fuzzy SISO process model identified under closed loop earlier. The plot of operating pressure has been displaced downwards (i.e., plot shows $P_2 - 205$) and the plot of separator level has been displaced upwards (i.e., plot shows $L_2 + 19$) to allow plotting alongside with output composition. The performance of controller is slightly better with the 3-partition fuzzy model as compared to the 2-partition fuzzy model. Significant improvement over the controllers using SISO process models (Figures 10.12 and 10.13) will also be observed. Figure 10.23 shows that quite good performance can also be achieved by replacing the fuzzy process models used by Loops 1 and 2 with linear process models with the constant term included. The output response below 25 percent is, however, observed to be quite sluggish and the time required for the steady-state offset correction can be quite considerable. Comparison with Figure 10.22 shows that the output composition response is generally quicker and the steady-state error generally smaller with the 3-partition fuzzy process model.

Comparison of Figures 10.17 and 10.24 shows that there is also an improvement in the response to feed flowrate changes as compared to control using SISO process models.

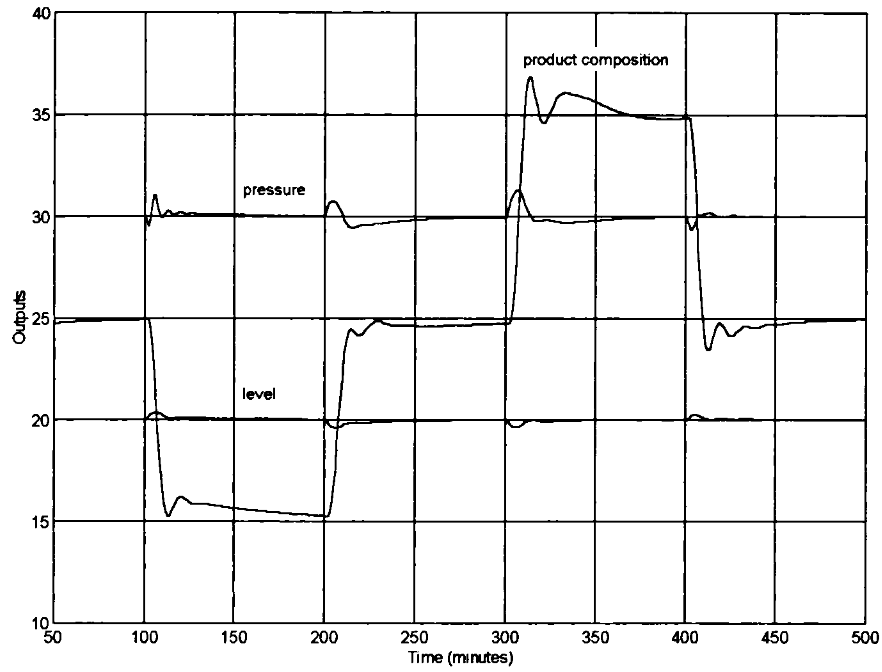


Figure 10.21: Process output responses to setpoint changes in product composition when using 2-partition fuzzy MISO process models in Loops 1 & 2 ($n_1 = 10; n_2 = 10; n_3 = 7; K_{f_{21}} = 0.025; K_{f_{22}} = 0.05; K_{f_{23}} = 0.05$).

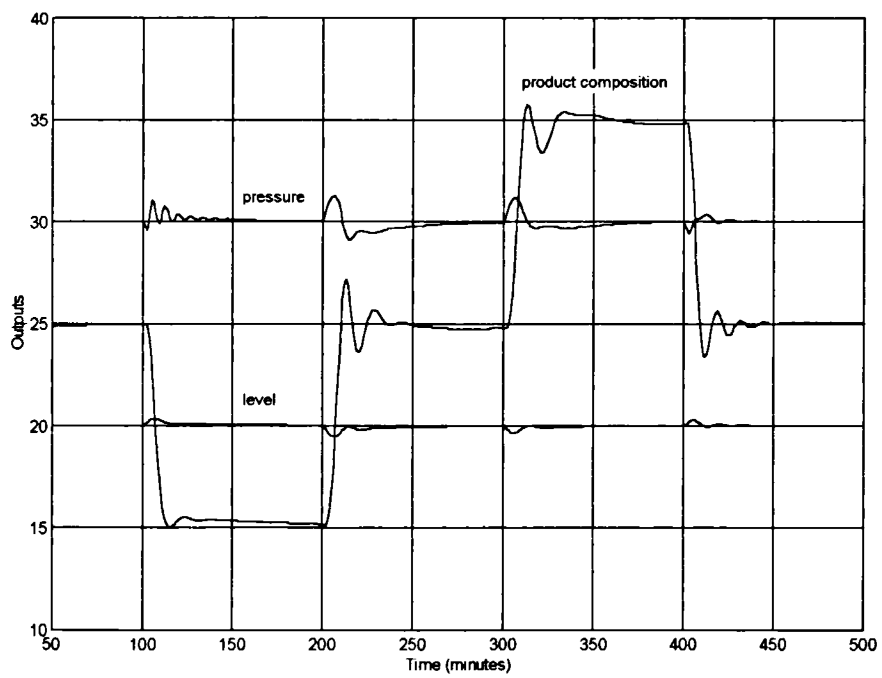


Figure 10.22: Process output responses to setpoint changes in product composition when using 3-partition fuzzy MISO process models in Loops 1 & 2 ($n_1 = 10; n_2 = 10; n_3 = 7; K_{f_{21}} = 0.025; K_{f_{22}} = 0.05; K_{f_{23}} = 0.05$).

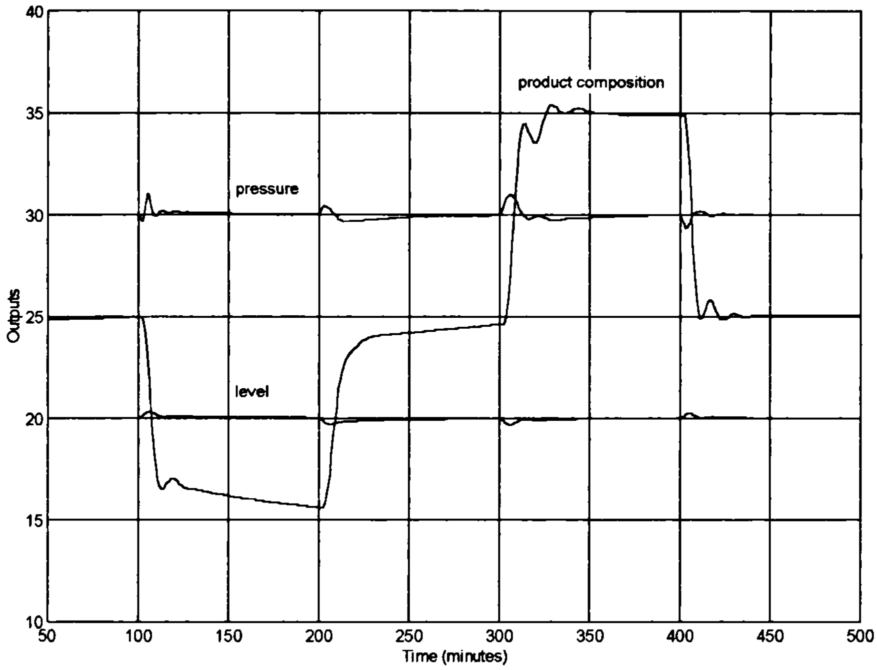


Figure 10.23: Process output responses to setpoint changes in product composition when using linear MISO process models in Loops 1 & 2 $(n_1 = 10; n_2 = 10; n_3 = 7; K_{f_{21}} = 0.025; K_{f_{22}} = 0.05; K_{f_{23}} = 0.05)$.

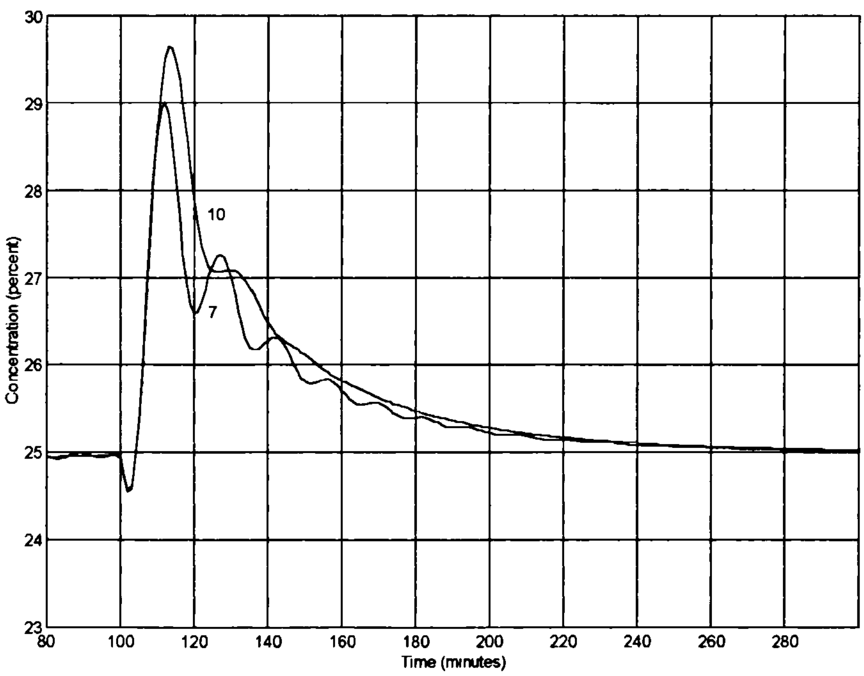


Figure 10.24: Effect of prediction horizon on product composition response to -10 percent change in feed flowrate when using 3-partition fuzzy MISO process models in Loops 1 & 2 $(n_2 = 10; n_3 = 7; K_{f_{21}} = 0.025; K_{f_{22}} = 0.05; K_{f_{23}} = 0.05)$.

10.8 Dead Time Compensation

The presence of dead time in the system was highlighted from the open-loop response of the system. The following second-order SISO model with a dead time of d sampling intervals can therefore be used to better represent the relationship between product composition and steam pressure:

$$\begin{aligned} R^i:\ if\, X_2(t)\ is\ A^i\ then\ X_2(t+1) = a_1^i X_2(t) + a_2^i X_2(t-1) + b_1^i P_{100}(t-d) \\ + b_2^i P_{100}(t-(d+1)),\ \ i = 1,\dots,p \end{aligned} \tag{10.23}$$

where $X_2(t)$ and $P_{100}(t)$ are the product composition and steam pressure at time t, respectively; and the sampling interval is 1-minute. Table 10.6 compares the 3-partition fuzzy process model identified by assuming different values of dead time ranging from 0 to 2 sampling intervals. It will be observed that the minimum MSE is obtained when the dead time is assumed to be equal to 1-sampling interval. This model was then used in a control scheme which allowed dead time compensation. Details of the dead time compensation scheme are presented elsewhere and will not be elaborated further here.

Figures 10.25 and 10.26 compare the performance of the controller to a series of setpoint changes with and without dead time compensation using smaller prediction horizons than the usual 10-steps used in earlier studies. It will be observed that dead time compensation makes it is possible to achieve good performance using smaller prediction horizons.

No.	Dead time	Model parameters				MSE
		a_1^i	a_2^i	b_1^i	b_2^i	
1	No dead time	1.6596 1.6986 1.6912	-0.7338 -0.7998 -0.7914	0.0002 0.0018 0.0003	0.0084 0.0112 0.0144	0.0050
2	1-sampling interval	1.5865 1.6388 1.6247	-0.6865 -0.7605 -0.7536	0.0046 0.0078 0.0087	0.0069 0.0079 0.0102	0.0032
3	2-sampling intervals	1.5302 1.5877 1.5900	-0.6567 -0.7273 -0.7423	0.0084 0.0128 0.0149	0.0062 0.0052 0.0073	0.0044

Table 10.6: Effect of the assumed dead time on model parameters and modelling accuracy of the 3-partition fuzzy SISO process model used for product composition control.

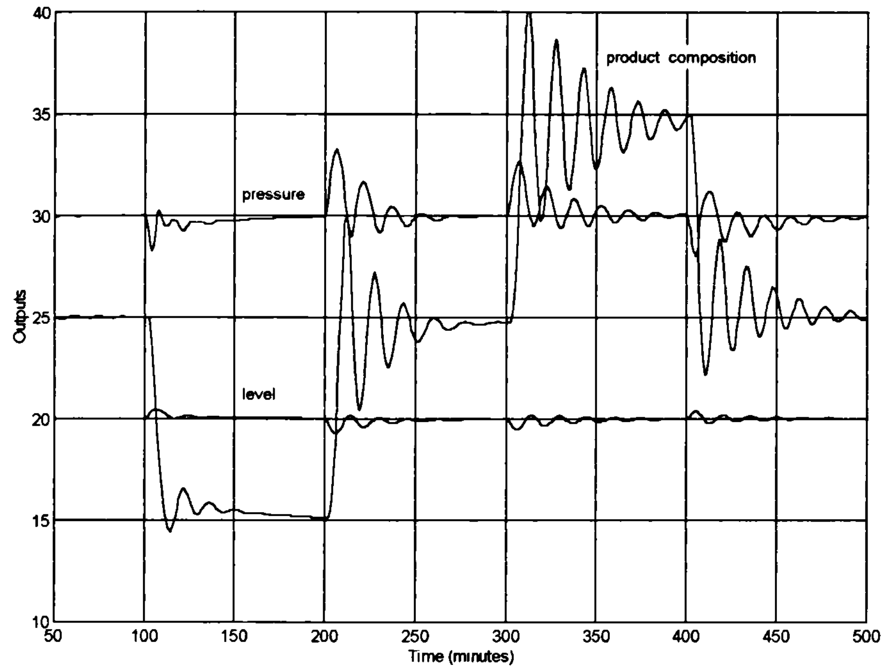


Figure 10.25: Process output responses to setpoint changes in product composition when using proposed controller without dead time compensation ($n_1 = 7; n_2 = 10; n_3 = 7; K_{f_{21}} = 0.025; K_{f_{22}} = 0.05; K_{f_{23}} = 0.05$).

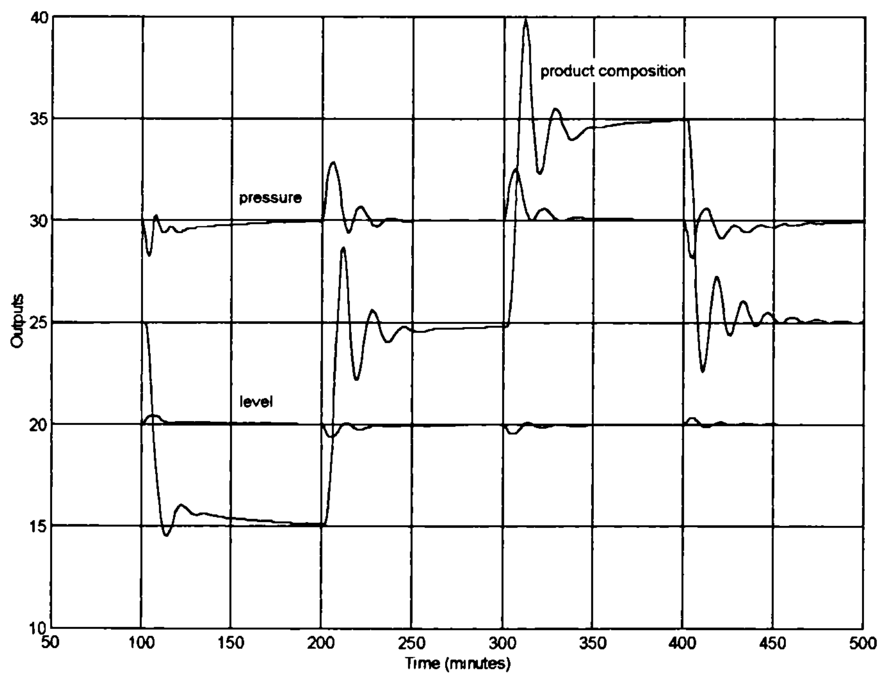


Figure 10.26: Process output responses to setpoint changes in product composition when using proposed controller with dead time compensation ($n_1 = 7; n_2 = 10; n_3 = 7; K_{f_{21}} = 0.025; K_{f_{22}} = 0.05; K_{f_{23}} = 0.05$).

10.9 Conclusions

The study has examined the performance of the proposed controller in the context of a multivariable nonlinear system with significant process interaction and dead times. The modelling accuracy of fuzzy process models was found to be not significantly better than linear process models with a constant term. This is most likely attributable to the system being only slightly nonlinear as compared to other systems that we have examined such as the distillation process and CSTR system. Evidence of this is provided by the generally good performance of the controllers using linear process models. Nevertheless, it has still been shown that the use of our proposed fuzzy model-based predictive controller leads to some improvement over comparable controllers using linear process models.

The proposed controller can be easily extended to MISO and MIMO systems. By including a process disturbance as a model input, feedforward compensation of disturbances was shown to be feasible. Also, it was shown that the design of a 2-input 2-output multivariable controller can be approached by decomposing it into two MISO fuzzy model-based controllers. The resulting controller was shown to be able to cope with process nonlinearities and interaction between loops.

Chapter 11

CONTROL OF SYSTEMS WITH CONSTRAINTS

11.1 Introduction

Thus far, it has been assumed that the only constraints imposed on the control system are minimum and maximum values on the process input (and hence the controller output). Constraints, however, can be present on the inputs, states and the outputs of the process. Practical systems can have many operating constraints, some of which may be required to protect it from entering unsafe or uneconomic operating regions. Loss of control is experienced when a manipulated variable reaches either limit of travel. In most control loops, the limits are the ends of the stroke of a control valve, representing physical limits of its capacity. However, in certain cases, hard limits must be set short of full stroke to favour equipment constraints downstream of the valve. A valve supplying fuel to a burner, for example, cannot be shut off completely without extinguishing its flame. Therefore fuel valves typically have a low limit which cannot be passed by the controller; shutdown is accomplished with a stop-valve or other on-off mechanism. In addition to magnitude constraints, another type of constraint often encountered are rate constraints. Here, the change of the controller output per sample is limited between two values once again imposed by physical limitations of the plant.

The above-mentioned constraints are called *hard constraints* because no violations of the bounds is allowed at any time. On the other hand, *soft constraints* may be imposed in the design of control systems primarily as a way of improving the performance of the control system. Violation of a soft constraint is permitted, but the overall effect of imposing the constraint will be an improvement in the performance of the control system mainly brought about by a reduction in the controller activity. This may be required, for example, when the number of steps in the control horizon is set to more than one.

Hard constraints can be handled most easily by clipping the output from the controller. It is shown in this chapter that clipping can lead to a significantly different process output response from that provided by the unconstrained controller. When there are multiple constraints such as when the control horizon is more than one, a method must be found for dealing with these constraints explicitly. The most established method for dealing with problems of this nature is quadratic programming (QP). A disadvantage of QP methods is that they are rather complex and computationally intensive. This issue may not be important in the process industry where sampling times may be in the order of seconds or even minutes.

Only hard constraints are considered in this study. The performance of the unconstrained controller is initially examined when using an extended control horizon. Next, the effect of clipping the output from the unconstrained controller using magnitude and rate constraints is examined. The discussion then focuses on extending the analytical control algorithm to enable explicit constraint handling capability using quadratic programming. Both input and output constraints are considered and an attempt made to compare different methods of constraint handling.

11.2 Extended Control Horizon

All previous studies have been carried out using a control horizon of 1-step. The most common type of constraints, however, arise from the use of control horizons consisting of more than 1-step. In this section, therefore, we will examine how the use of an extended control horizon affects the unconstrained controller's performance.

It has been shown previously that the unconstrained optimal controller output sequence in our proposed fuzzy model-based controller is given by:

$$U(t) = [\mathbf{Q}^T\mathbf{Q}]^{-1}\mathbf{Q}^T\left[W(t) - \mathbf{P}X(t) - \mathbf{R}(k + err(t))\right] \tag{11.1}$$

The coefficients of **P** and **R** matrices are not affected by the control horizon used. The coefficients of **Q**, however, depend on the control horizon and are given by:

$$\mathbf{Q}=\begin{bmatrix} q_{11} & 0 & \cdots & 0 \\ q_{21} & q_{22} & \ddots & \\ \vdots & & \ddots & 0 \\ q_{m1} & q_{m2} & \cdots & q_{mm} \\ \vdots & \vdots & \vdots & \vdots \\ q_{n1} & q_{n2} & \cdots & q_{nm} \end{bmatrix} \tag{11.2}$$

where the prediction horizon is n steps and control horizon is m steps. Back-substitution during model transformation leads to the following recursive formulae for determining the coefficients of the $\mathbf{Q}$ matrix for different lengths of the control horizon.

11.2.1 First-Order System

If the control horizon is set to 1,

$$\begin{aligned} &q_1 = b_1 \\ &q_i = a_1 \cdot q_{i-1} + b_1, \text{ for } i = 2,\ldots,n \end{aligned} \tag{11.3}$$

If the control horizon is set to 2,

$$\begin{aligned} &q_{11} = b_1 && q_{12} = 0 \\ &q_{i1} = a_1 \cdot q_{(i-1)1} && q_{i2} = a_1 \cdot q_{(i-1)2} + b_1, \text{ for } i = 2,\ldots,n \end{aligned} \tag{11.4}$$

If the control horizon is set to 5,

$$\begin{array}{lllll} q_{11} = b_1 & q_{12} = 0 & q_{13} = 0 & q_{14} = 0 & q_{15} = 0 \\ q_{21} = a_1 \cdot q_{11} & q_{22} = b_1 & q_{23} = 0 & q_{24} = 0 & q_{25} = 0 \\ q_{31} = a_1 \cdot q_{21} & q_{32} = a_1 \cdot q_{22} & q_{33} = b_1 & q_{34} = 0 & q_{35} = 0 \\ q_{41} = a_1 \cdot q_{31} & q_{42} = a_1 \cdot q_{32} & q_{43} = a_1 \cdot q_{33} & q_{44} = b_1 & q_{45} = 0 \\ q_{i1} = a_1 \cdot q_{(i-1)1} & q_{i2} = a_1 \cdot q_{(i-1)2} & q_{i3} = a_1 \cdot q_{(i-1)3} & q_{i4} = a_1 \cdot q_{(i-1)4} & q_{i5} = a_1 \cdot q_{(i-1)5} + b_1, \end{array}$$

$$\text{for } i = 5,\ldots,n \tag{11.5}$$

11.2.2 Second-Order System

If the control horizon is set to 1,

$$\begin{aligned} &q_1 = b_1 \\ &q_2 = a_1 \cdot q_1 + b_1 + b_2 \\ &q_i = a_1 \cdot q_{(i-1)} + a_2 \cdot q_{(i-2)} + b_1 + b_2, \text{ for } i = 3,\ldots,n \end{aligned} \tag{11.6}$$

If the control horizon is set to 2,

$$\begin{aligned}
&q_{11} = b_1 && q_{12} = 0 \\
&q_{21} = a_1 \cdot q_{11} + b_2 && q_{22} = b_1 \\
&q_{31} = a_1 \cdot q_{21} + a_2 \cdot q_{11} && q_{32} = a_1 \cdot q_{22} + b_1 + b_2 \\
&q_{i1} = a_1 \cdot q_{(i-1)1} + a_2 \cdot q_{(i-2)1} && q_{i2} = a_1 \cdot q_{(i-1)2} + a_2 \cdot q_{(i-2)2} + b_1 + b_2, \text{ for } i = 4,\ldots,n
\end{aligned} \tag{11.7}$$

If the control horizon is set to 3,

$$\begin{aligned}
&q_{11} = b_1 && q_{12} = 0 && q_{13} = 0 \\
&q_{21} = a_1 \cdot q_{11} + b_2 && q_{22} = b_1 && q_{23} = 0 \\
&q_{31} = a_1 \cdot q_{21} + a_2 \cdot q_{11} && q_{32} = a_1 \cdot q_{22} + b_2 && q_{33} = b_1 \\
&q_{41} = a_1 \cdot q_{31} + a_2 \cdot q_{21} && q_{42} = a_1 \cdot q_{32} + a_2 \cdot q_{22} && q_{43} = a_1 \cdot q_{33} + b_1 + b_2 \\
&q_{i1} = a_1 \cdot q_{(i-1)1} + a_2 \cdot q_{(i-2)1} && q_{i2} = a_1 \cdot q_{(i-1)2} + a_2 \cdot q_{(i-2)2} && q_{i3} = a_1 \cdot q_{(i-1)3} + a_2 \cdot q_{(i-2)3} + b_1 + b_2, \\
& && && \text{for } i = 5,\ldots,n
\end{aligned} \tag{11.8}$$

Hence, the coefficients of **Q** can be determined quite easily for any length of control horizon in the case of first-order and second-order systems.

11.2.3 Application to Control of CSTR

Examination of Figures 11.1 to 11.6 will reveal how using a control horizon of more than 1-step affects the unconstrained controller. Only the 5-partition fuzzy model was used in all of the studies on the CSTR system in this chapter. The number of steps in the prediction and control horizons used by the controller, H_p and H_c, respectively, are indicated in brackets together with the values of the feedback filter gains used by the controllers.

Comparison of Figures 11.1 and 11.3 shows that the deviation of the process output from the steady-state level is less when the control horizon is 2-steps. On the other hand, it will be observed that the improved performance is achieved with a marked increase in controller activity (Figures 11.2 and 11.4). In Figure 11.4, it will be observed that the rate of heat removal drops to zero. Such extreme conditions are seldom tolerated with real reactors.

Figures 11.5 and 11.6 compare the process output responses when using control horizons of different lengths. The performance of the controller using a control horizon of 2-steps appears optimum in both cases. It must be reiterated, however, that that the improved performance is achieved at the expense of a more active controller which can severely affect the controller's robustness.

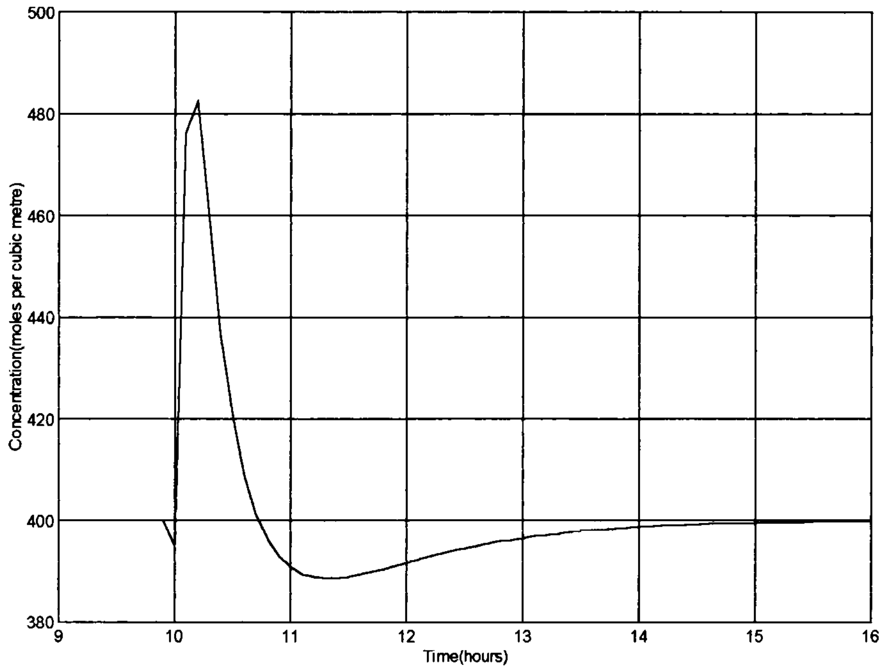


Figure 11.1: Process output response to +20 percent change in feed flowrate when using unconstrained controller.
($K_{f_1} = 1$; $K_{f_2} = 0.1$; $H_p = 10$; $H_c = 1$; 5-partition fuzzy model)

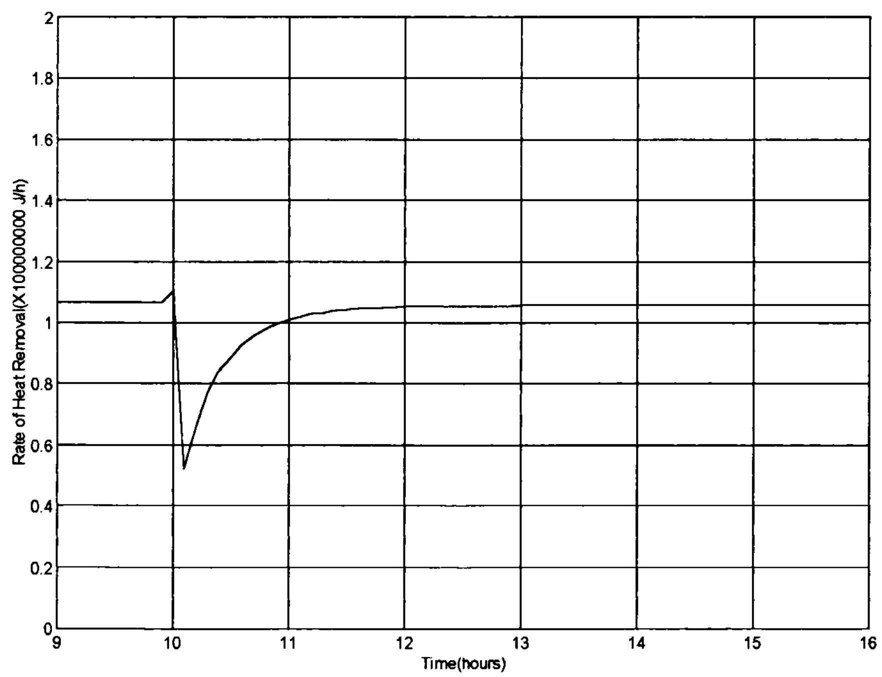


Figure 11.2: Controller output response to +20 percent change in feed flowrate when using unconstrained controller.
($K_{f_1} = 1$; $K_{f_2} = 0.1$; $H_p = 10$; $H_c = 1$; 5-partition fuzzy model)

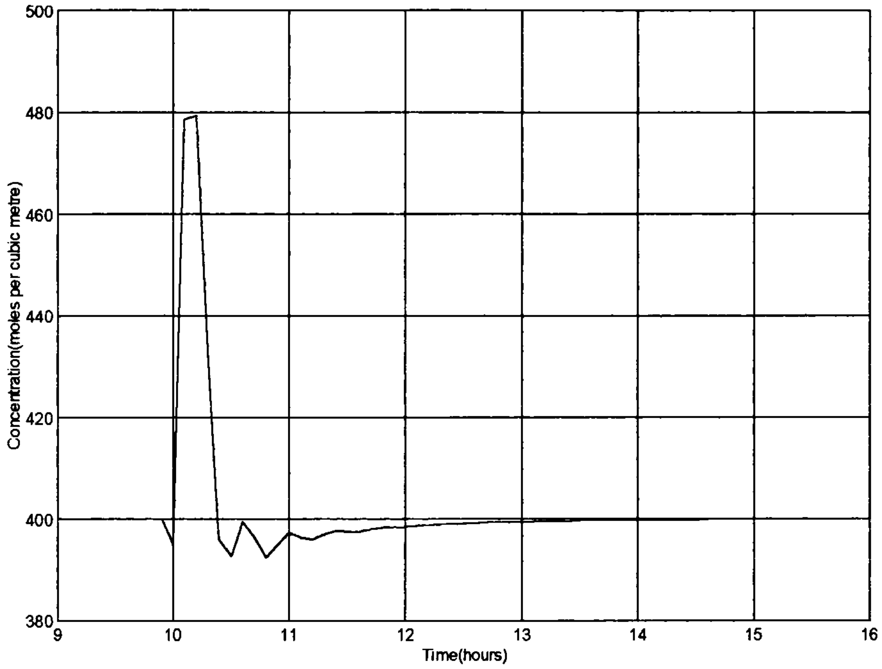


Figure 11.3: Process output response to +20 percent change in feed flowrate when using unconstrained controller.
($K_{f_1} = 1$; $K_{f_2} = 0.1$; $H_p = 10$; $H_c = 2$; 5-partition fuzzy model)

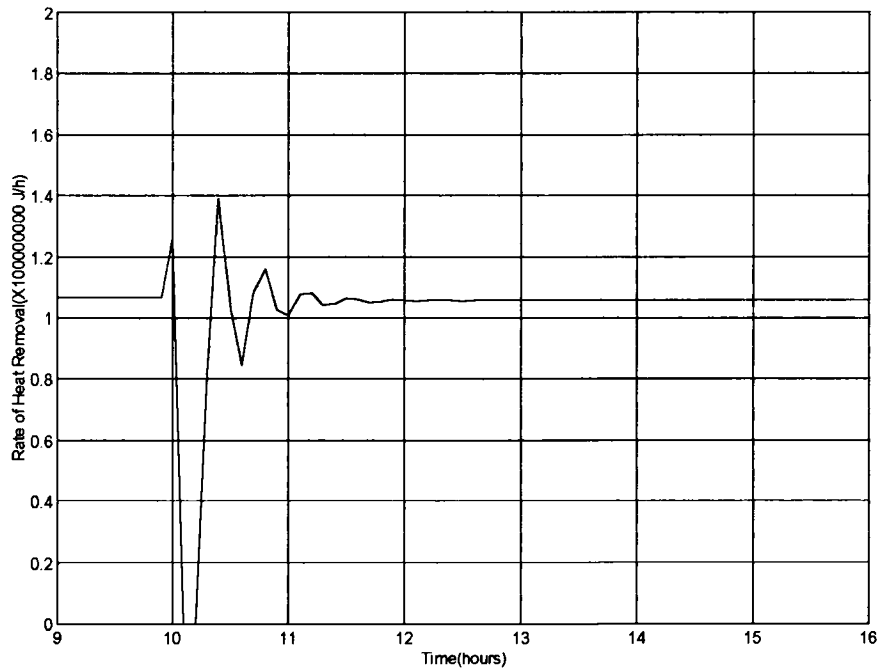


Figure 11.4: Controller output response to +20 percent change in feed flowrate when using unconstrained controller.
($K_{f_1} = 1$; $K_{f_2} = 0.1$; $H_p = 10$; $H_c = 2$; 5-partition fuzzy model)

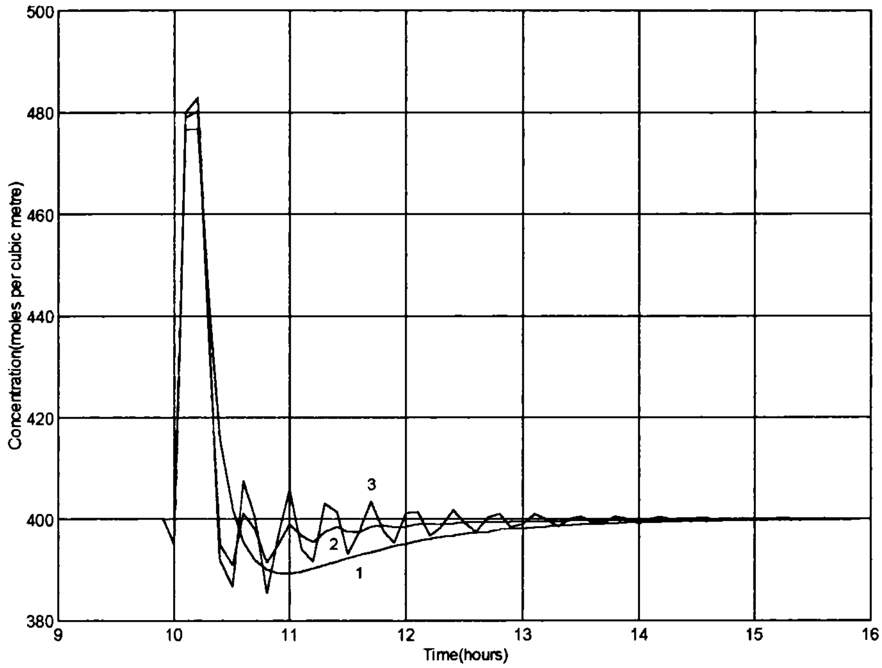


Figure 11.5: Effect of the control horizon on process output response to +20 percent change in feed flowrate when using unconstrained controller. (K_{f_1} = 1; K_{f_2} = 0.1; H_p = 5; 5-partition fuzzy model)

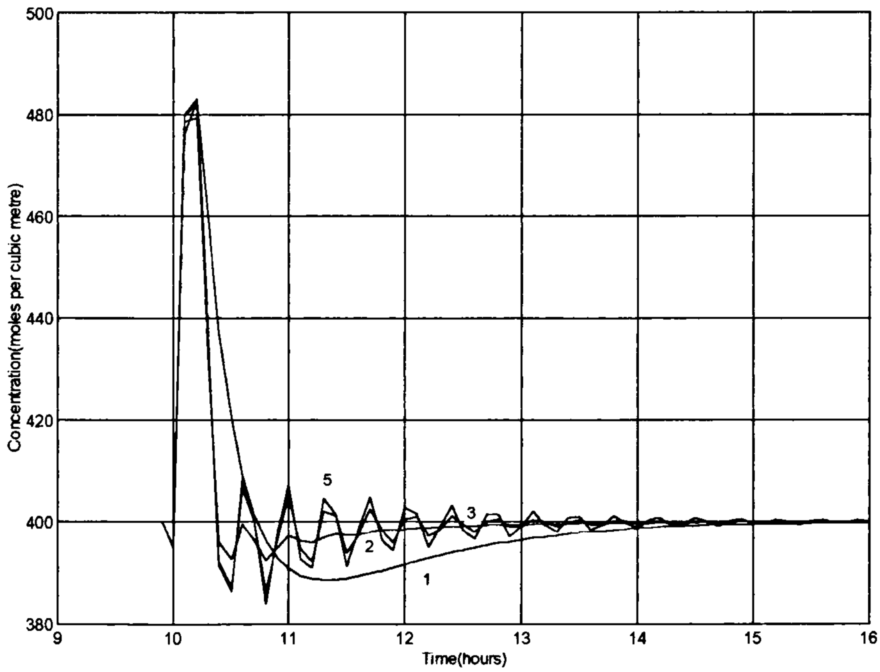


Figure 11.6: Effect of the control horizon on process output response to +20 percent change in feed flowrate when using unconstrained controller. (K_{f_1} = 1; K_{f_2} = 0.1; H_p = 10; 5-partition fuzzy model)

11.3 Controller Output Clipping

In this section, we will examine how clipping the output from the unconstrained controller affects the control system's performance. Clipping may be achieved at the software level within the control system itself, but it is more likely to be imposed externally by the physical limitations of the equipment used such as control valves. The action of clipping the controller output using magnitude and rate constraints is examined below.

11.3.1 Magnitude Constraints

Magnitude constraints can be expressed using the following equation:

$$\underline{u}(t) \le u(t) \le \overline{u}(t) \tag{11.9}$$

where,

$u(t)$ = process input (i.e. controller output) at time t.

$\underline{u}(t)$ = low level magnitude constraint.

$\overline{u}(t)$ = high level magnitude constraint.

11.3.2 Rate Constraints

Rate constraints can be expressed using the following equations:

$$-\Delta\underline{u}(t) \le u(t) - u(t-1) \le +\overline{\Delta u}(t) \tag{11.10}$$

or, $$u(t-1) - \Delta\underline{u}(t) \le u(t) \le u(t-1) + \overline{\Delta u}(t) \tag{11.11}$$

where,

$\Delta\underline{u}(t)$ = absolute value of low level rate constraint.

$\overline{\Delta u}(t)$ = absolute value of high level rate constraint.

and $u(t)$ is similarly defined as above.

11.3.3 Application to Control of CSTR

It has been assumed in the following studies that $\pm 0.1 \times 10^8$ J.h^{-1} rate constraints have been imposed on the controller output in addition to the normally imposed magnitude constraints. As in earlier studies, only the 5-partition fuzzy model was examined.

Figures 11.7 and 11.8 examine the effect of the prediction horizon on the performance of systems with and without rate constraints when a control horizon of 1-step is used. It will be observed that the maximum deviation is generally greater with rate constraints. A significant performance deterioration is observed in the presence of rate constraints when the prediction horizon is small but the use of prediction horizons of more than 5-steps seems to lead to considerable improvement. The optimum response seems to be achieved with a prediction horizon of about 5-steps.

Figures 11.9 and 11.10 examine the effect of the control horizon on the performance of systems with and without rate constraints, respectively. It will be observed that the use of control horizons of more than 1-step leads to a more oscillatory response when rate constraints are present.

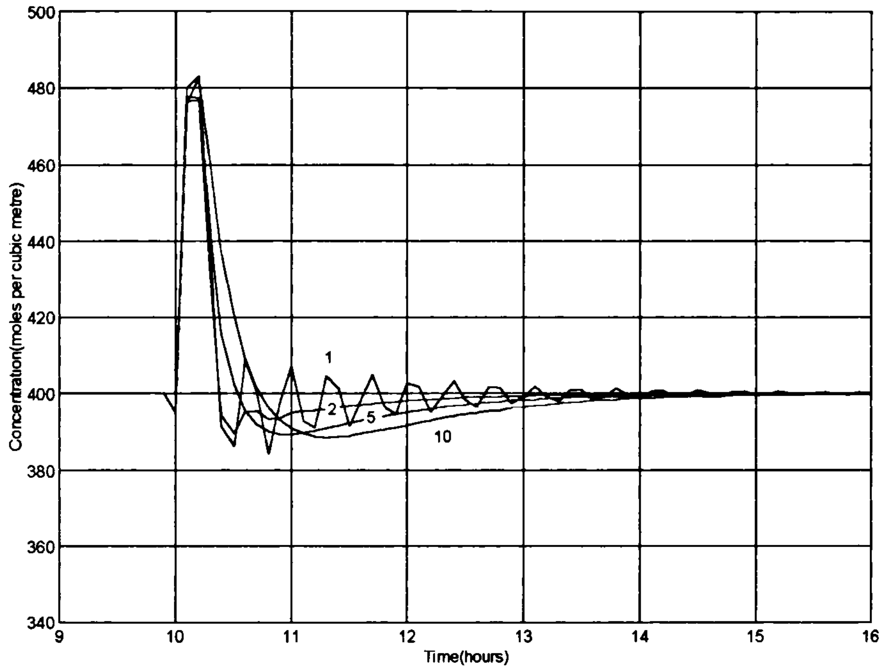


Figure 11.7: Effect of prediction horizon on process output response to +20 percent change in feed flowrate with no rate constraints.
($K_{f_1} = 1$; $K_{f_2} = 0.1$; $H_c = 1$; 5-partition fuzzy model)

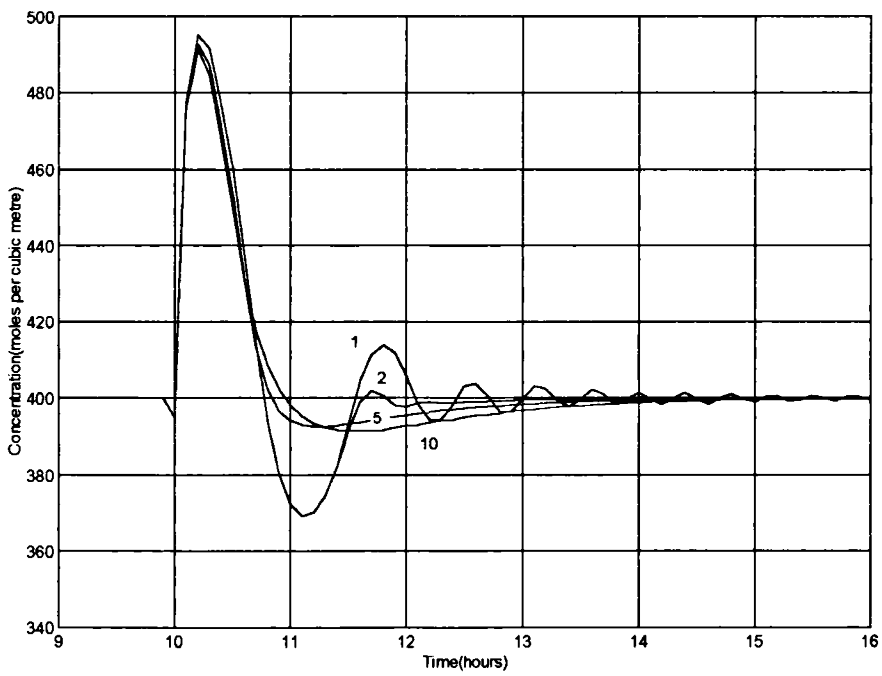


Figure 11.8: Effect of prediction horizon on process output response to +20 percent change in feed flowrate with rate constraints.
($K_{f_1} = 1$; $K_{f_2} = 0.1$; $H_c = 1$; 5-partition fuzzy model)

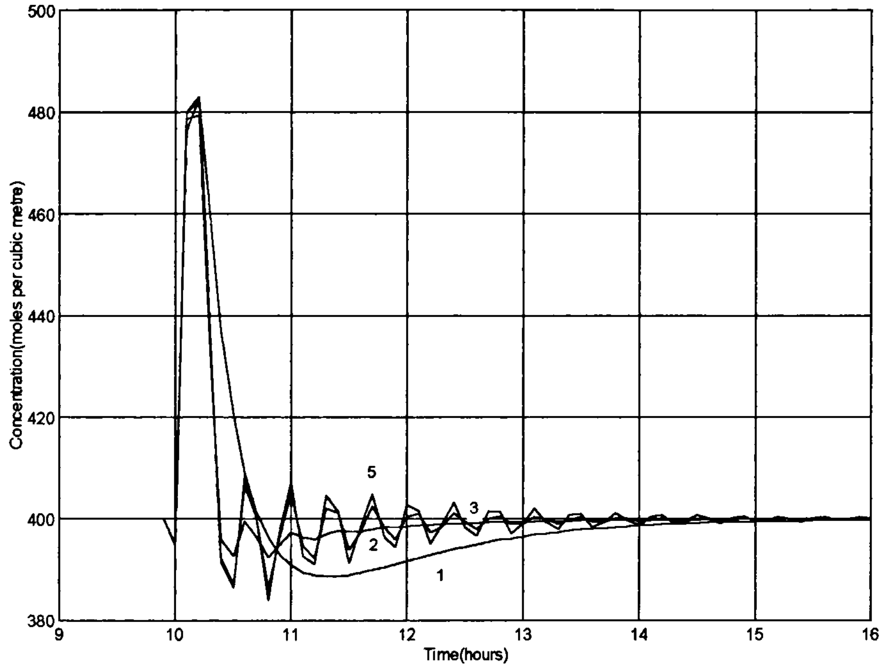


Figure 11.9: Effect of the control horizon on process output response to +20 percent change in feed flowrate with no rate constraints.
($K_{f_1} = 1$; $K_{f_2} = 0.1$; $H_p = 10$; 5-partition fuzzy model)

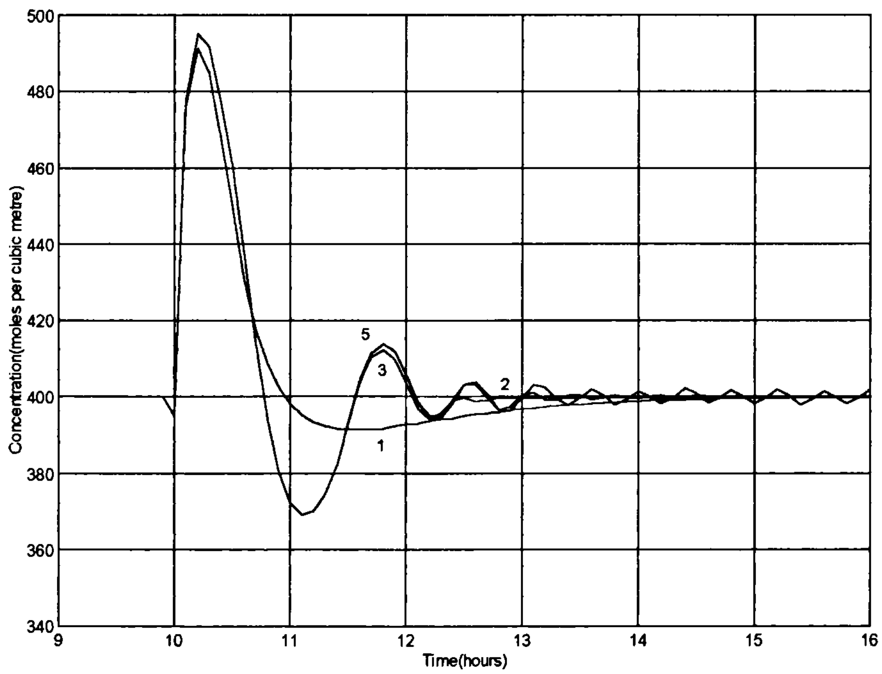


Figure 11.10: Effect of the control horizon on process output response to +20 percent change in feed flowrate with rate constraints.
($K_{f_1} = 1$; $K_{f_2} = 0.1$; $H_p = 10$; 5-partition fuzzy model)

11.4 Constraint Handling Using the Quadratic Programming Approach

11.4.1 General

It has been shown in the last section that the use of a control horizon of more than 1-step can lead to undesirable output response in the presence of externally applied constraints. In this section, therefore, we will examine a method of explicitly handling multiple constraints in the formulation of the optimal controller algorithm. Various methods are available for this purpose, including Rosen's gradient projection algorithm (Soeterboek, 1991) and quadratic programming (QP) (Garcia and Moreshedi, 1986; Tsang and Clarke, 1988; Wilkinson *et al.*, 1994).

QP finds solutions to problems that can be stated in the following form:

$$\min_{x} \left\{ \frac{1}{2} x^T \mathbf{H} x + c^T x \right\} \text{ subject to } \mathbf{A} x \leq b \tag{11.12}$$

where the Hessian matrix $\mathbf{H}$ and vector c are the set of coefficients of the quadratic objective function. The matrix $\mathbf{A}$ and vector b, are the coefficients of the linear constraints. The vector x is a set of independent variables.

Non-negative least squares (NNLS) can be used to solve QP problems which can be mathematically stated in the following form:

$$\min_{x} \|\mathbf{A}x - b\|^2 \text{ subject to } x \geq 0 \tag{11.13}$$

Note that the NNLS problem is a subset of the QP problem. For badly conditioned problems, QP may give more accurate answers than NNLS. For problems greater than order 20, QP may also be faster than NNLS; otherwise NNLS is generally more efficient. The Kuhn-Tucker conditions form the basis for the development of the NNLS computational algorithm. These conditions must be satisfied at the constrained optimum.

The method presented below uses the NNLS computational algorithm and follows closely the method proposed by Wilkinson *et al.* (1994) based on Generalised Predictive Control.

11.4.2 Formulation of Control Problem Incorporating Constraints

Recall that the optimal controller output is given by minimising the following cost function:

$$
\begin{aligned}
J &= \sum_{i=1}^{n}[y_p(t+i) - w(t+i)]^2 \\
&= [Y(t) - W(t)]^T [Y(t) - W(t)] \\
&= [\mathbf{P}X(t) + \mathbf{Q}U(t) + \mathbf{R}(k + err(t)) - W(t)]^T [\mathbf{P}X(t) + \mathbf{Q}U(t) + \mathbf{R}(k + err(t)) - W(t)]
\end{aligned}
\tag{11.14}
$$

If the constraints can be expressed using linear inequalities in the form $\mathbf{C}U(t) \geq h$, inspection of equation (11.14) shows that the constrained optimisation problem being considered is a least squares problem involving linear inequality constraints (LSI):

$$\min_U \|\mathbf{A}U(t) - b\|^2 \text{ subject to } \mathbf{C}U(t) \geq h \tag{11.15}$$

where,

$\mathbf{A} = \mathbf{Q}$

$b = W(t) - \mathbf{P}X(t) - \mathbf{R}(k + err(t))$

$\mathbf{C}$ = matrix providing dynamic information on constraints

h = vector giving limiting values for the constraints

11.4.3 Solution of the Constrained Optimisation Problem

In order to obtain a solution which satisfies the constraint set, the LSI problem above is first converted into a least distance programming (LDP) problem.

Let k denote the rank of the $m \times n$ matrix A in equation (11.15). Consider the partitioned singular value decomposition (SVD) of A,

$$\mathbf{A} = \mathbf{B}\begin{bmatrix} \mathbf{S} & 0 \\ 0 & 0 \end{bmatrix}\mathbf{V}^T = [\mathbf{B}_1 \vdots \mathbf{B}_2]\begin{bmatrix} \mathbf{S} & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix} \tag{11.16}$$

where $\mathbf{B}$ is a $m \times m$ orthogonal matrix, $\mathbf{V}$ is a $n \times n$ orthogonal matrix, and $\mathbf{S}$ is a $k \times k$ diagonal matrix consisting of the singular values of $\mathbf{A}$.

By an orthogonal transformation of variables, $U = \mathbf{V}_1 p$, the LSI objective function can be rewritten as:

$$\begin{aligned}\phi(x) &= \|b - \mathbf{A}U\|^2 \\ &= \left\| \begin{bmatrix} \mathbf{B}_1^T b \\ \mathbf{B}_2^T b \end{bmatrix} - \begin{bmatrix} \mathbf{S}p \\ 0 \end{bmatrix} \right\|^2 \\ &= \|\tilde{b}_1 - \mathbf{S}p\|^2 + \|\tilde{b}_2\|^2 \end{aligned} \tag{11.17}$$

where,

$$\tilde{b}_i = \mathbf{B}_i^T b, \qquad i = 1,\ 2 \tag{11.18}$$

A further change of variables transforms the objective function to:

$$\phi(x) = \|z\|^2 + \|\tilde{b}_2\|^2, \text{ where } z = \mathbf{S}p - \tilde{b}_1 \tag{11.19}$$

Except for the constant $\|\tilde{b}_2\|^2$, the original LSI problem is then equivalent to the following LDP formulation:

$$\min \|z\|, \text{ subject to } \tilde{\mathbf{C}}z \geq \tilde{h} \tag{11.20}$$

where,

$$\tilde{\mathbf{C}} = \mathbf{C}\mathbf{V}_1\mathbf{S}^{-1} \tag{11.21}$$

and $$\tilde{h} = h - \tilde{\mathbf{C}}\tilde{b}_1 \tag{11.22}$$

In solving the LDP representation, $\tilde{d}$ is first computed in order to solve the NNLS form,

$$\min \|\mathbf{Q}\tilde{d} - v\|, \text{ subject to } \tilde{d} \geq 0 \tag{11.23}$$

where,

$$\mathbf{Q} = \begin{bmatrix} \tilde{\mathbf{C}}^T \\ \tilde{h}^T \end{bmatrix} \text{ and } v = [0, \ldots,\ 0,\ 1]^T \tag{11.24}$$

The residual vector,

$$r = \mathbf{Q}\tilde{d} - v, \tag{11.25}$$

where $r = [r_1, \ldots,\ r_{n+1}]^T$ is next computed and the LDP solution is obtained as,

$$\hat{z}_i = -r_i / r_{n+1}, \quad i = 1, \ldots, n \tag{11.26}$$

Finally, the solution of the original LSI problem is then recovered from,

$$\hat{U} = \mathbf{V}_1\mathbf{S}^{-1}(\hat{z} + b_1), \qquad \hat{z} = [\hat{z}_1, \ldots, \hat{z}_n]^T. \tag{11.27}$$

In some cases, constraints may be incompatible and this is indicated by $\|r\| = 0$ in equation (11.25).

To summarise, the LSI optimisation problem is initially transformed into the LDP form using singular value decomposition . The LDP algorithm examines if the set of constraints is consistent, and if consistent finds a solution using the NNLS algorithm. Further details, including proofs, are available from the text by Lawson and Hanson (1974).

Application of the LDP and NNLS algorithms requires that the constraints be expressed explicitly using linear inequalities in the form $\mathbf{C}U(t) \geq h$. We next discuss the mathematical formulation of constraints into the form suitable for applying these algorithms.

11.4.4 Input Magnitude Constraints

Input magnitude constraints can be expressed using:

$$\underline{u}(t+i-1) \leq u(t+i-1) \leq \overline{u}(t+i-1) \qquad i = 1, \ldots, H_c \tag{11.28}$$

where $\underline{u}(t+i-1)$ and $\overline{u}(t+i-1)$ are the lower and upper limits corresponding to to the i-th step in the control horizon, and H_c is the length of the control horizon. It is possible to express the above constraints in the following alternative form to facilitate their use here,

$$\begin{aligned} u(t+i-1) &\geq \underline{u}(t+i-1) \\ \text{and } -u(t+i-1) &\geq -\overline{u}(t+i-1) \end{aligned}, \qquad i = 1, \ldots, H_c \tag{11.29}$$

The above constraints can be expressed in the following matrix form:

$$\mathbf{C}U(t) \geq h: \quad \left[\begin{array}{cccc} -1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 \\ \hline 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{array}\right] \begin{bmatrix} u(t) \\ \vdots \\ u(t+H_c-1) \end{bmatrix} \geq \left[\begin{array}{c} -\overline{u} \\ \vdots \\ -\overline{u} \\ \hline \underline{u} \\ \vdots \\ \underline{u} \end{array}\right] \tag{11.30}$$

where **C** is a $2H_c \times H_c$ matrix, and U and h are $H_c \times 1$ vectors. Even though the constraints $\underline{u}(t+i-1)$ and $\overline{u}(t+i-1)$ can be allowed to vary over the control horizon, it has been assumed to be constant and is denoted as $\underline{u}$ and $\overline{u}$.

11.4.5 Input Rate Constraints

Input rate constraints can be expressed using,

$$-\underline{\Delta u}(t+i-1) \leq u(t+i-1) - u(t+i-2) \leq +\overline{\Delta u}(t+i-1), \qquad i = 1,\ldots,H_c \quad (11.31)$$

where $\underline{\Delta u}(t+i-1)$ and $\overline{\Delta u}(t+i-1)$ are the absolute values of lower and upper limits corresponding to the i-th step in the control horizon. It is possible to express the above constraints in the following alternative form,

$$\begin{array}{rl} & u(t+i-1) - u(t+i-2) \geq -\underline{\Delta u}(t+i-1) \\ \text{and} & -[u(t+i-1) - u(t+i-2)] \geq -\overline{\Delta u}(t+i-1) \end{array}, \qquad i = 1,\ldots,H_c \quad (11.32)$$

The above constraints can be expressed in the following matrix form:

$$CU(t) \geq h: \begin{bmatrix} -1 & 0 & \cdots & & 0 \\ 1 & -1 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \\ \vdots & \ddots & \ddots & -1 & 0 \\ 0 & \cdots & 0 & 1 & -1 \\ \hline 1 & 0 & \cdots & & 0 \\ -1 & 1 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \\ \vdots & \ddots & \ddots & 1 & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} u(t) \\ \vdots \\ u(t+H_c-1) \end{bmatrix} \geq \begin{bmatrix} -\Delta u - u(t-1) \\ -\Delta u \\ \vdots \\ -\Delta u \\ \hline -\Delta u + u(t-1) \\ -\Delta u \\ \vdots \\ -\Delta u \end{bmatrix} \quad (11.33)$$

where **C** is a $2H_c \times H_c$ matrix, and U and h are $H_c \times 1$ vectors. Even though the constraints $\underline{\Delta u}(t+i-1)$ and $\overline{\Delta u}(t+i-1)$ can be allowed to vary over the control horizon, it has been assumed to be constant and of equal magnitude (i.e. Δu).

11.4.6 Output Magnitude Constraints

Output magnitude constraints can be expressed using future output predictions:

$$\underline{y_p}(t+i) \leq y_p(t+i) \leq \overline{y_p}(t+i), \qquad i = 1,\ldots,H_p \quad (11.34)$$

or, $$\underline{Y} \leq Y(t) \leq \overline{Y} \quad (11.35)$$

where $\underline{Y}$ and $\overline{Y}$ denote vectors of the minimum and maximum values of output magnitude constraints over the prediction horizon.

Using the following expression for future output predictions,

$$Y = PX(t) + QU(t) + R(k + err(t)) \tag{11.36}$$

leads to,

$$\underline{Y} - \beta \le QU(t) \le \overline{Y} - \beta \tag{11.37}$$

where,

$$\beta = PX(t) + R(k + err(t)) \tag{11.38}$$

The above constraints can be re-expressed in the following form to facilitate their use here:

$$\begin{aligned} QU(t) &\ge \underline{Y} - \beta \\ \text{and} \quad -QU(t) &\ge -(\overline{Y} - \beta) \end{aligned} \tag{11.39}$$

It is possible to express the above constraints in the following matrix form:

$$\mathbf{C}U(t) \ge h: \begin{bmatrix} -q_{11} & 0 & \cdots & 0 \\ -q_{21} & -q_{22} & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ -q_{m1} & -q_{m2} & \cdots & -q_{mm} \\ \vdots & \vdots & \vdots & \vdots \\ -q_{n1} & -q_{n2} & \cdots & -q_{nm} \\ \hline q_{11} & 0 & \cdots & 0 \\ q_{21} & q_{22} & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ q_{m1} & q_{m2} & \cdots & q_{mm} \\ \vdots & \vdots & \vdots & \vdots \\ q_{n1} & q_{n2} & \cdots & q_{nm} \end{bmatrix} \begin{bmatrix} u(t) \\ \vdots \\ u(t + H_c - 1) \end{bmatrix} \ge \begin{bmatrix} -(\overline{y_p} - \beta) \\ \vdots \\ -(\overline{y_p} - \beta) \\ \hline (\underline{y_p} - \beta) \\ \vdots \\ (\underline{y_p} - \beta) \end{bmatrix} \tag{11.40}$$

where $\mathbf{C}$ is a $2H_p \times H_c$ matrix (note: $H_p = n$ and $H_c = m$ in the above matrix), and U and h are $H_c \times 1$ vectors. As in the case of input magnitude constraints, it has been assumed that $\underline{y_p}(t+i)$ and $\overline{y_p}(t+i)$ are constant over the prediction horizon.

If all three types of constraints are being considered, the overall dynamic constraint information matrix will be:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \mathbf{C}_3 \end{bmatrix} \quad \text{and} \quad h = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} \tag{11.41}$$

where $\mathbf{C}_1$, $\mathbf{C}_2$ and $\mathbf{C}_3$ are matrices providing dynamic constraint information on input magnitude, input rate and output constraints and h_i denotes the respective vector providing limiting values of the constraints.

11.4.7 The Handling of Incompatible Constraints

It was pointed out earlier that, in some cases, constraints may be incompatible. A common cause is that, in satisfying an output constraint, a control is required that violates one or more of the input constraints. To overcome this, it is useful to include a scheme for removing constraits in a predefined manner, starting with the most distant output magnitude constraints, until all the remaining constraints are satisfied.

11.4.8 Application to Control of CSTR

Earlier studies have shown that the best performance with the CSTR system is achieved with a control horizon of 2-steps. All studies in this section have, therefore, used a control horizon of 2-steps.

We consider the application of the controller using a 5-partition fuzzy process model in the presence of the following input magnitude and input rate constraints:

Input Magnitude Constraints:

$$0 \le u(t+i-1) \le 2 \times 10^8, \qquad i = 1,\ldots,H_c \tag{11.42}$$

Input Rate Constraints:

$$-0.1 \times 10^8 \le u(t+i-1) - u(t+i-2) \le 0.1 \times 10^8, \qquad i = 1,\ldots,H_c \tag{11.43}$$

where $u(t)$ is the controller output, and H_c is the length of the control horizon.

The output magnitude constraints were set as follows, unless otherwise specified:

Output Magnitude Constraints:

$$250 \le y_p(t+i) \le 550, \qquad i = 1,\ldots,H_p \tag{11.44}$$

where $y_p(t+i)$ is the predicted process output; and H_p is the length of the prediction horizon.

Figures 11.11 to 11.14 examine the effect of load and setpoint changes on the controller incorporating input constraints when using a prediction horizon of 10-steps. Very little controller activity is observed in Figures 11.12 and 11.14. This is particularly obvious if Figure 11.12 is compared with Figure 11.4. Hence, controller robustness is less likely to be a problem as compared to the unconstrained controller using a similar control horizon. Figures 11.15 and 11.16 compare the process output responses achieved by controller output clipping and explicitly handling the constraints using the QP approach. The significant improvement in performance by using the second approach is quite obvious.

Figure 11.17 examine the effect of incorporating output constraints together with input constraints on 150 moles.m^{-3} setpoint changes. The prediction horizon used by the controller has been reduced from 10-steps to 6-steps. It is obvious that output constraints can limit the maximum amount of setpoint change allowed.

Figure 11.18 compare the process output responses from a controller incorporating just input constraints with a controller incorporating both input and output constraints. Also shown is the performance from the controller where the controller output is clipped using constraints. The prediction horizon used by all the controllers has been standardised to 6-steps. It is obvious that the best overall response is achieved in this case by incorporating both input and output constraints. It must be stated, however, that this is true only in the case of small prediction horizons. Comparison of Figures 11.16 and 11.18 shows that the performance of the controller incorporating input and output constraints can be matched by a controller incorporating input constraints only if the prediction horizon is extended to 10-steps.

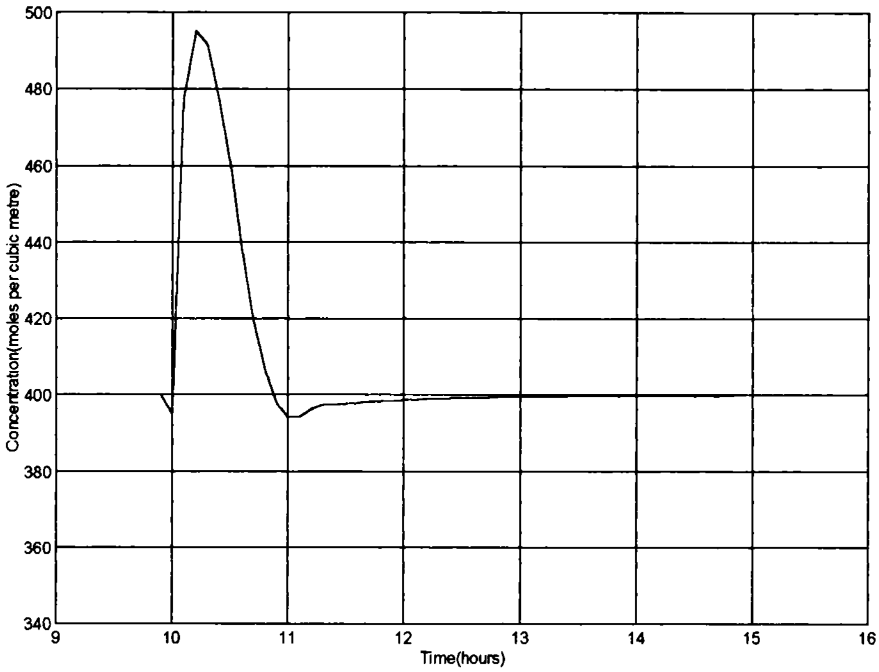


Figure 11.11: Process output response to +20 percent change in feed flowrate with input rate and magnitude constraints.
($K_{f_1} = 1$; $K_{f_2} = 0.1$; $H_p = 10$; $H_c = 2$; 5-partition fuzzy model)

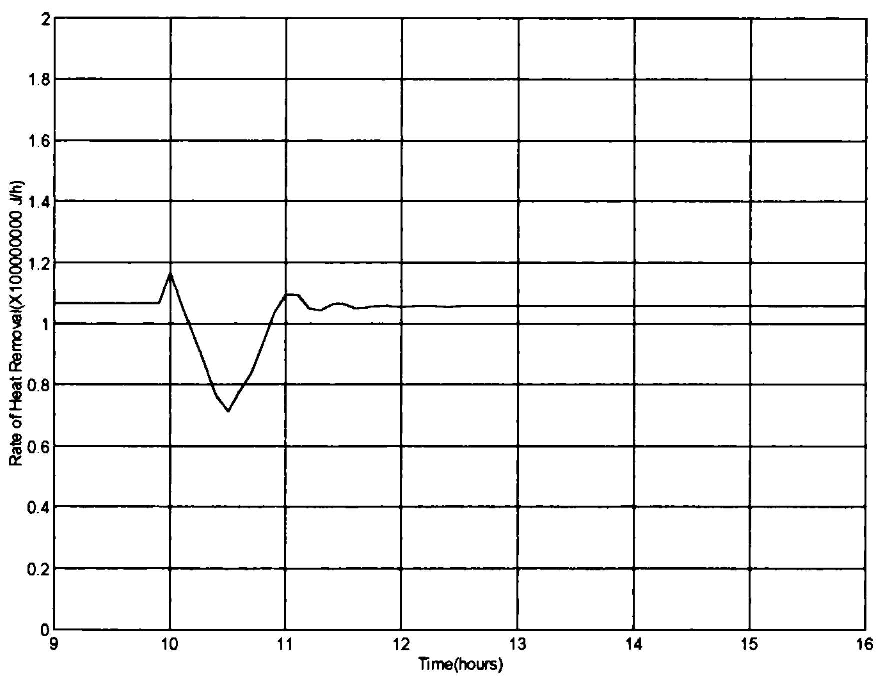


Figure 11.12: Controller output response to +20 percent change in feed flowrate with input rate and magnitude constraints.
($K_{f_1} = 1$; $K_{f_2} = 0.1$; $H_p = 10$; $H_c = 2$; 5-partition fuzzy model)

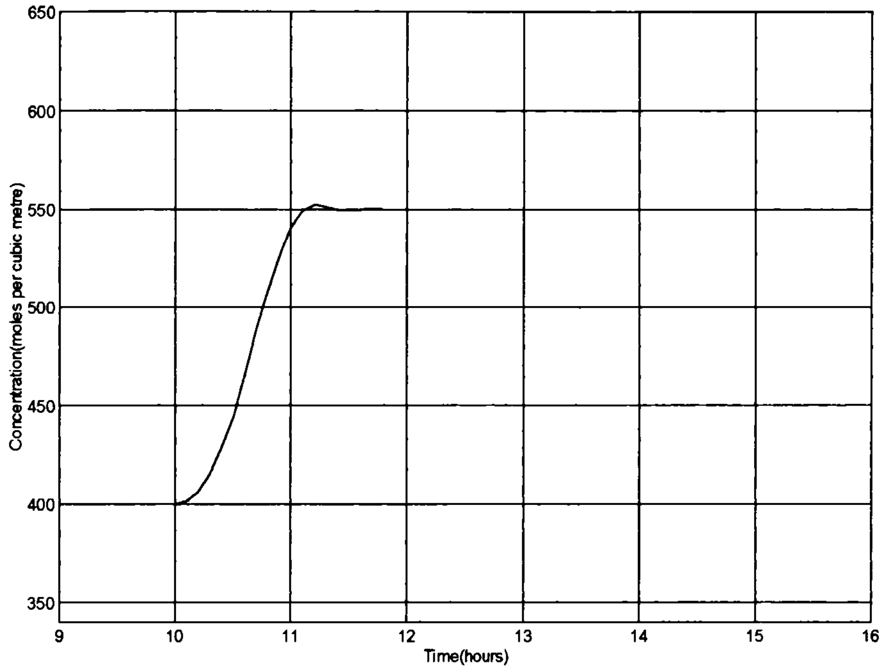


Figure 11.13: Process output response to +150 moles.m^{-3} setpoint change with input rate and magnitude constraints.
(K_{f_1} = 1; K_{f_2} = 0.1; H_p = 10; H_c = 2; 5-partition fuzzy model)

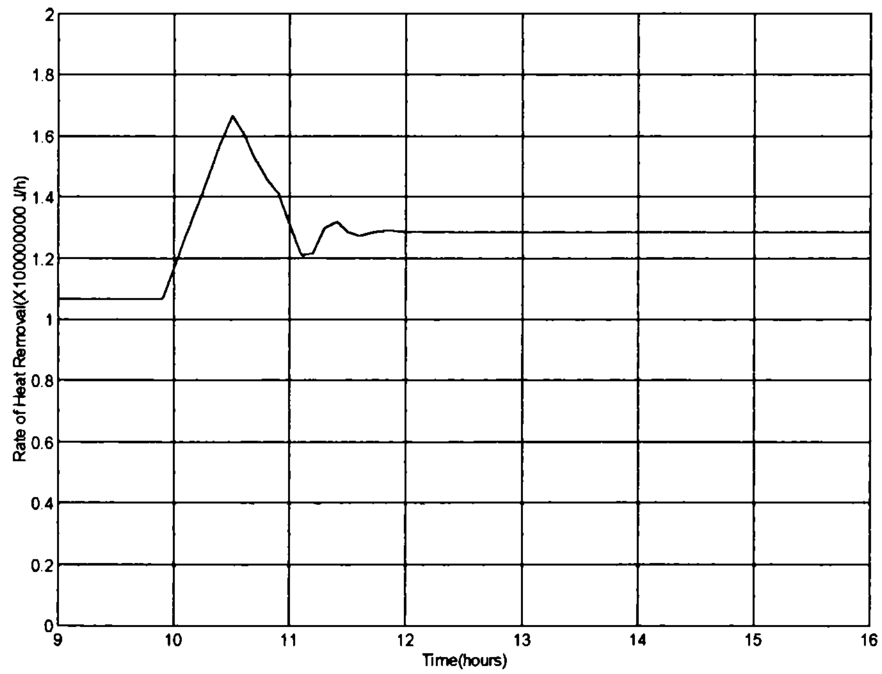


Figure 11.14: Controller output response to +150 moles.m^{-3} setpoint change with input rate and magnitude constraints.
(K_{f_1} = 1; K_{f_2} = 0.1; H_p = 10; H_c = 2; 5-partition fuzzy model)

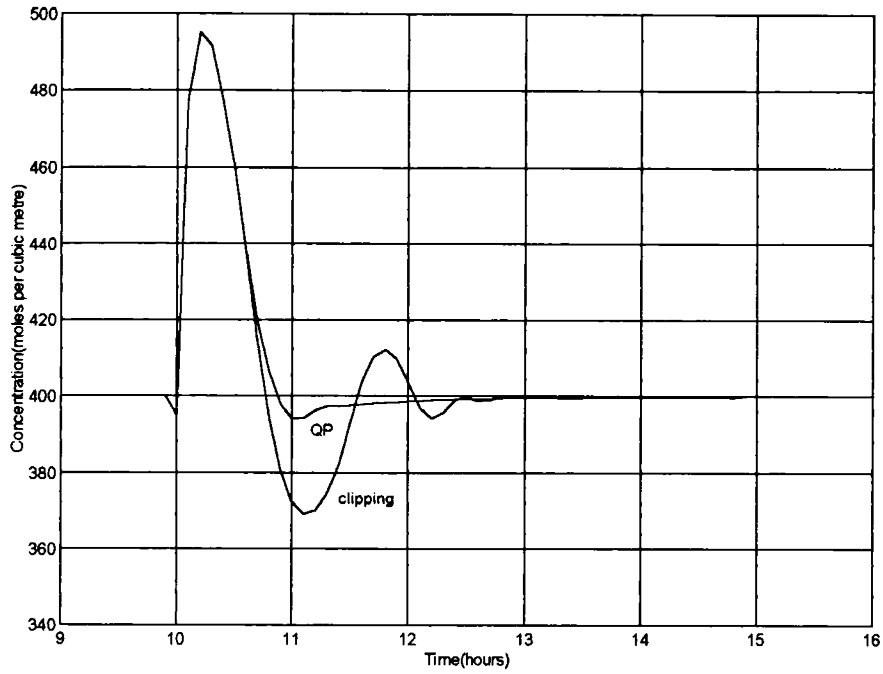


Figure 11.15: Comparison of process output responses to +20 percent change in feed flowrate with input rate and magnitude constraints.
($K_{f_1} = 1$; $K_{f_2} = 0.1$; $H_p = 10$; $H_c = 2$; 5-partition fuzzy model)

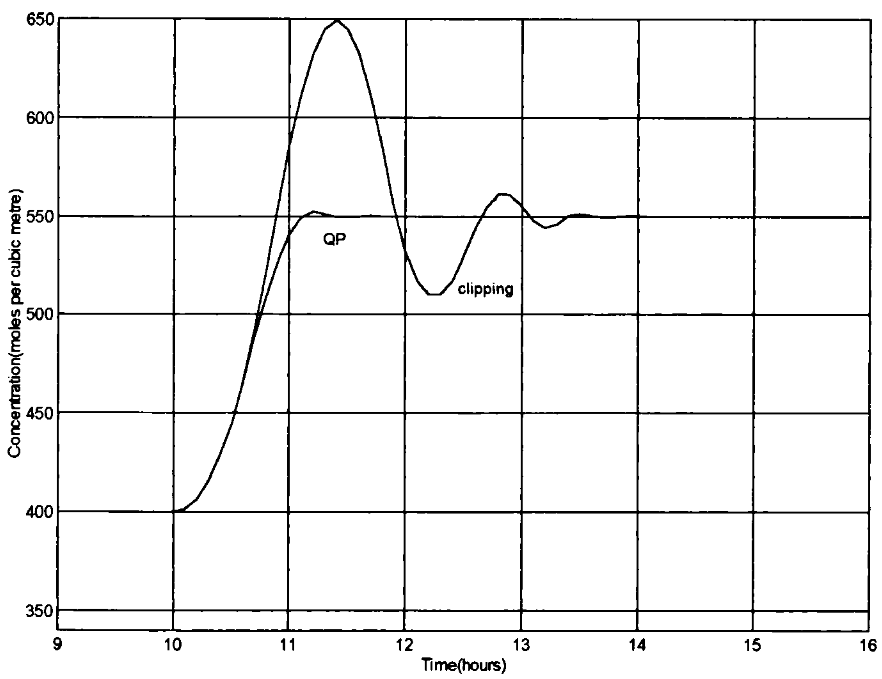


Figure 11.16: Comparison of process output responses to +150 moles.m^{-3} setpoint change with input rate and magnitude constraints.
($K_{f_1} = 1$; $K_{f_2} = 0.1$; $H_p = 10$; $H_c = 2$; 5-partition fuzzy model)

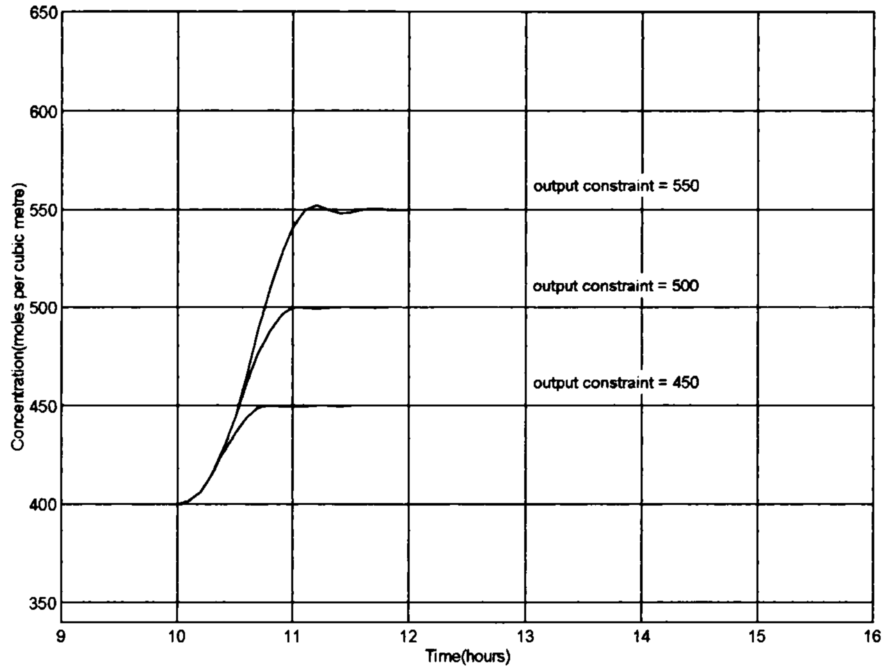


Figure 11.17: Process output response to +150 moles.m^{-3} setpoint change with input rate and magnitude and output constraints.
(K_{f_1} = 1; K_{f_2} = 0.1; H_p = 6; H_c = 2; 5-partition fuzzy model)

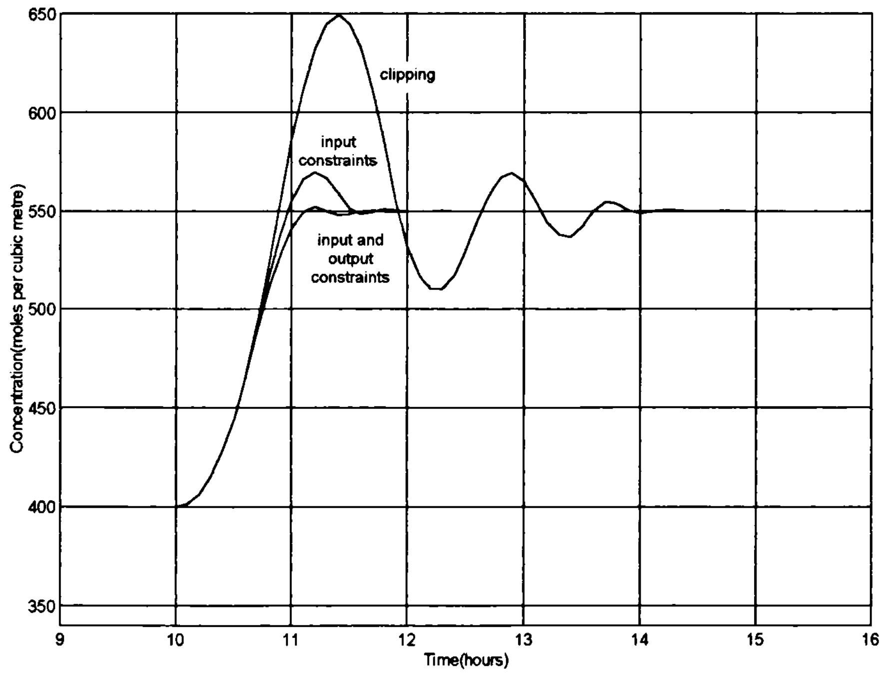


Figure 11.18: Comparison of process output responses to +150 moles.m^{-3} setpoint change.
(K_{f_1} = 1; K_{f_2} = 0.1; H_p = 6; H_c = 2; 5-partition fuzzy model)

11.4.9 Application to Control of Binary Distillation Column

The starting point for these studies was the MIMO controller described in Section 9.6. The controller used for bottoms product composition control was maintained the same, while the controller used for distillate composition was modified to enable constraint handling capability. As with the CSTR system, a control horizon of 2-steps was used and the prediction horizon was changed to 10-steps. Only the 3-partition MISO process models given in Tables 9.6 and 9.8 were evaluated. The other controller parameters were maintained the same as in Section 9.6. We considered the application of the controller in the presence of the following constraints:

Input Magnitude Constraints:

$$0 \le u(t+i-1) \le 5, \qquad i = 1,\ldots,H_c \tag{11.42}$$

Input Rate Constraints:

$$-0.02 \le u(t+i-1) - u(t+i-2) \le 0.02, \qquad i = 1,\ldots,H_c \tag{11.43}$$

Output Magnitude Constraints:

$$0.97 \le y_p(t+i) \le 0.99, \qquad i = 1,\ldots,H_p \tag{11.44}$$

where $u(t)$ is the controller output; $y_p(t+i)$ is the predicted process output; and H_p and H_c are the lengths of the prediction and control horizons, respectively.

Figures 11.19 to 11.21 show the process output responses when using different constraint handling methods. The plot of bottoms product composition has been displaced upwards (i.e. plot shows $x_B + 0.94$) to allow plotting alongside with distillate composition. It will be observed that controller output clipping (Figure 11.19) results in the worst output responses. The extent of the improvement achieved by incorporating constraints in the optimisation process is more obvious from Figure 11.22.

The effect of including the additional output magnitude constraints in the optimisation process is clear from Figure 11.23. It will be noticed that the output response becomes more sluggish as the constraints are reached and no difference is observed in the region in between the constraints.

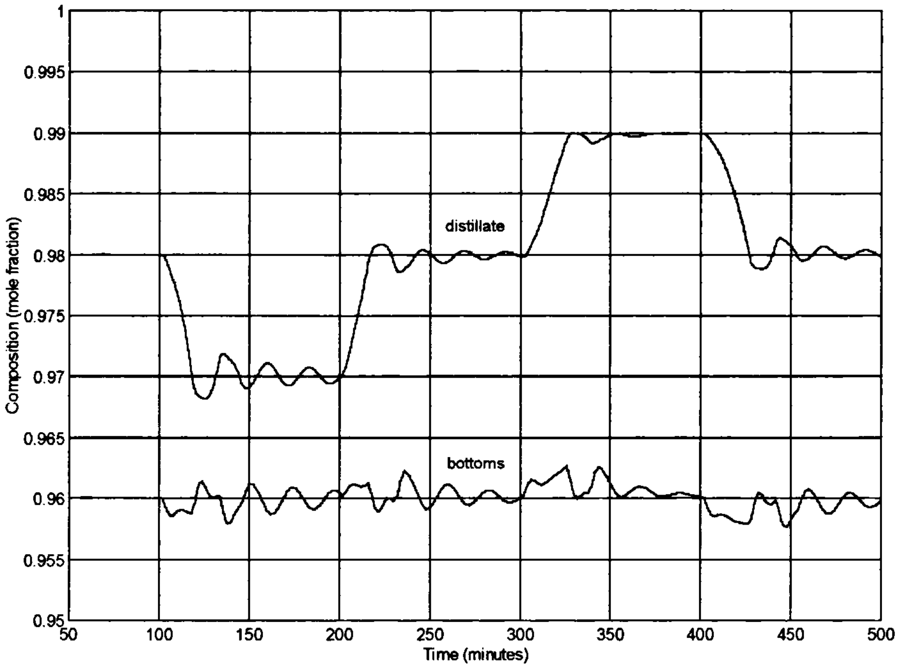


Figure 11.19: Process output responses to ± 0.01 setpoint changes in distillate composition with controller output clipping.

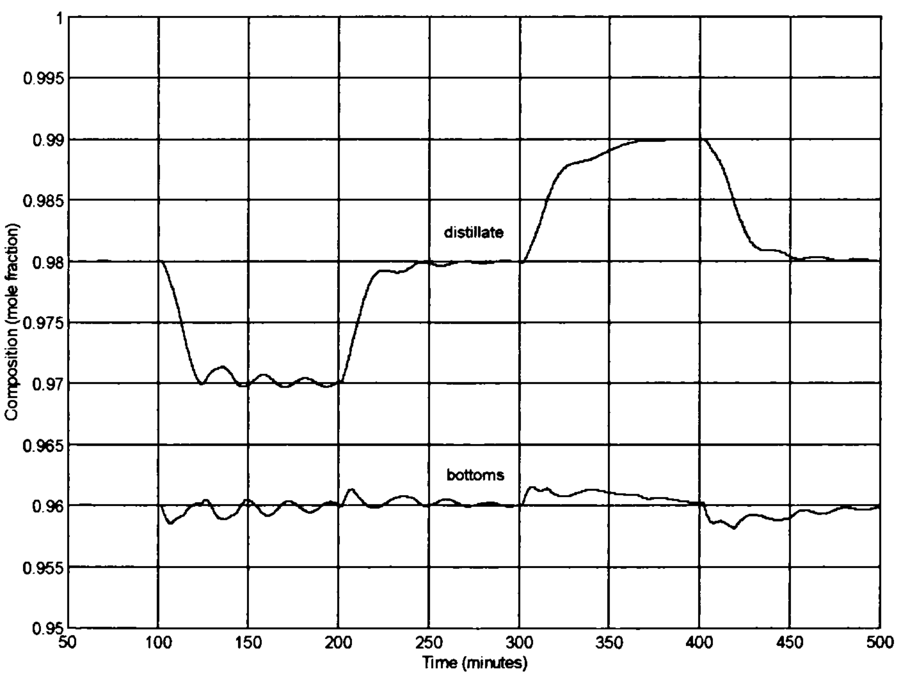


Figure 11.20: Process output responses to ± 0.01 setpoint changes in distillate composition with input magnitude and rate constraints.

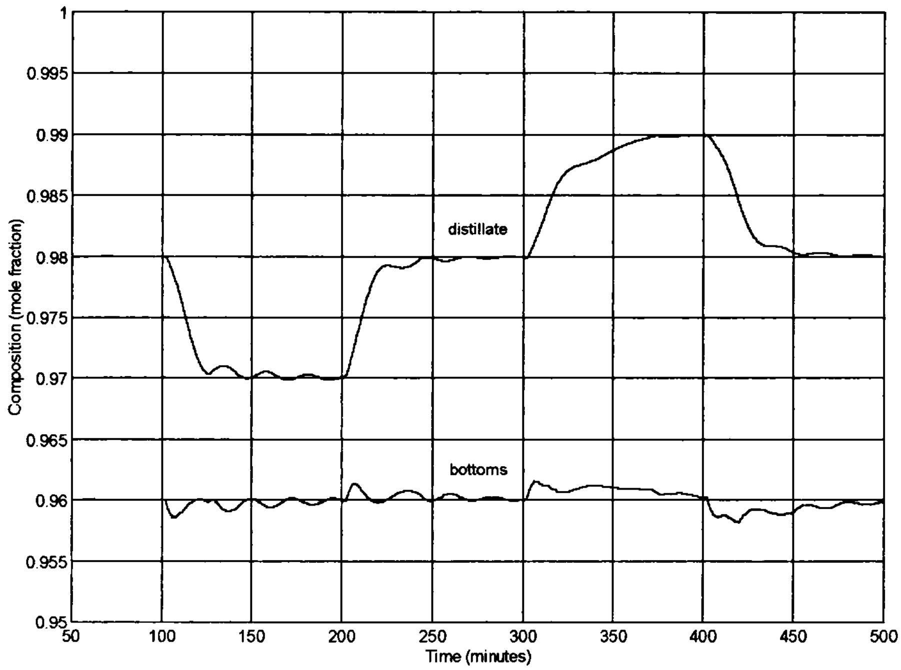


Figure 11.21: Process output responses to ± 0.01 setpoint changes in distillate composition with input magnitude and rate and output constraints.

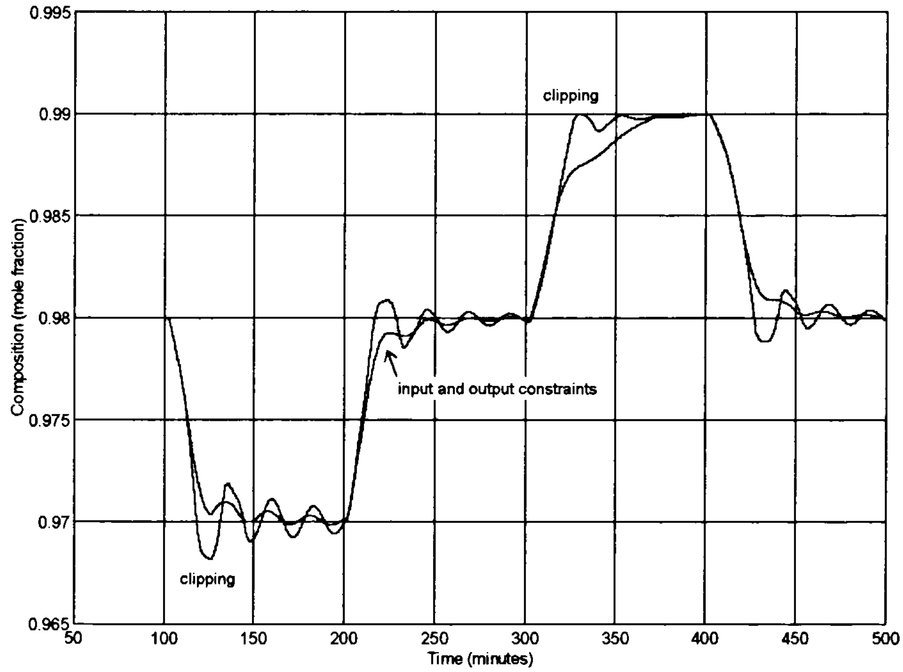


Figure 11.22: Comparison of process output responses to ± 0.01 setpoint changes in distillate composition using different methods of constraint handling.

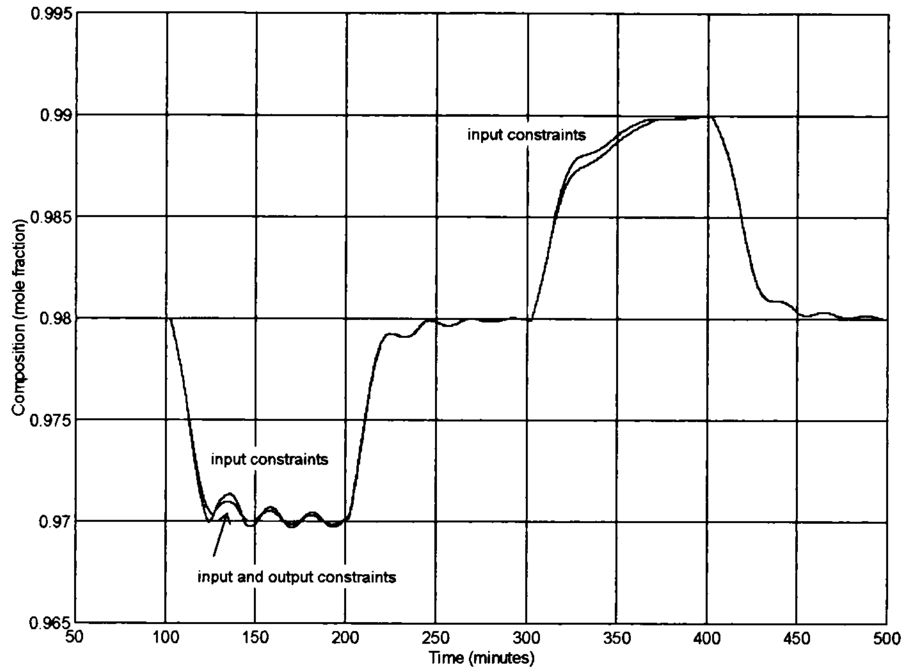


Figure 11.23: Comparison of process output responses to ± 0.01 setpoint changes in distillate composition using different methods of constraint handling.

11.5 Conclusions

It has been shown in this chapter that the presence of constraints can significantly alter the process output response of the unconstrained controller. A method of explicitly including the constraints in the optimisation process has therefore been considered.

Since the most common constraints are constraints on the controller output, the chapter has emphasized the use of an extended control horizon. The control algorithm derived in Chapter 7 was initially extended to a control horizon of greater than 1-step. The effect of an extended control horizon on the unconstrained controller's performance was then examined. It was shown to lead to an increase in control activity.

Finally, the use of the QP method of explicitly handling constraints was shown to lead to a significant improvement in the output response compared to merely clipping the controller output using the constraints. The flexibility of the QP method was also demonstrated by showing that process output constraints can also be easily incorporated in the optimisation process.

Chapter 12

CONCLUSIONS

12.1 Summary of this Thesis

The increased complexity and stringent demands of today's chemical process plants necessitates more sophisticated control systems capable of delivering better and more flexible performance. An approach that has been popularly used when dealing with challenging process control problems is MBPC. Despite their populrity, there are many processes which pose a challenge for the standard, linear model based algorithms. These include batch and semi-batch processes which are often carried out over a wide dynamic range, continuous processes which undergo frequent transitions to permit the manufacture of several grades of a basic product, and severely nonlinear processes for which linear model based controllers may be inadequate. Although fuzzy logic controllers have been used to tackle such highly nonlinear processes, a major limitation is the amount of time and effort needed to develop the fuzzy rule base using purely heuristic knowledge such as a human operator's control strategy. An alternative to modelling the operator's response is to use a fuzzy logic system as a computing paradigm for modelling nonlinear process dynamics.

In this thesis, it has been shown how a fuzzy process model can be incorporated into a MBPC scheme. Even though a control algorithm based on a 1-step ahead predictive control strategy was initially proposed in Chapter 5, the emphasis in later chapters has been on determining the optimal controller output using a long-range predictive control strategy. Numerous investigations have shown that the proposed approach has many of the remarkable attributes of linear MBPC, while offering better performance with highly nonlinear processes.

The fuzzy modelling method proposed by Takagi and Sugeno has been used throughout this thesis. This modelling method uses fuzzy inference to combine the outputs of a number of auto-regressive linear sub-models to construct an overall nonlinear process model. The accuracy of this modelling method when applied to model chemical processes was examined using simulations of two well known examples of nonlinear chemical processes in Chapter 4. It was shown that the fuzzy process model provides better modelling accuracy than a single linear process model for representing the dynamics of nonlinear processes over a wide range.

The fundamental issue of designing control algorithms which can capitalise on the better modelling accuracy of the fuzzy process model to also provide better controller performance was examined in Chapters 5 to 7. Difficulties in using traditional cost function and optimisation techniques with fuzzy models have led other researchers to use numerical search techniques for determining the controller output. Even though a numerical approach was also demonstrated in Chapter 6, the emphasis in this thesis has been on computationally efficient analytically derived control algorithms. A control algorithm based on an analytical approach for determining the 1-step ahead predictive controller output was first proposed in Chapter 5. This control system was shown to be able to outperform a similar control system using a single linear process model. The results also show that inspite of the greatly reduced computational requirement of our proposed control system, it is possible to equal or better the performance of some of the other numerically based fuzzy model based control systems that have been proposed in the literature.

It is a well-known fact that one-step ahead predictive controllers are susceptible to robustness problems. A solution to this problem is to use a multi-step prediction horizon. Two methods of extending the controller to multi-step prediction and control horizons were examined in Chapters 6 and 7. The first method was based on a numerical approach. It was first shown how multi-step predictions can be made using the fuzzy process model. The Fibonacci search method was then used to determine the optimal controller output numerically. Investigations carried out using the tank level simulation showed that the output response becomes less oscillatory and more sluggish as the number of steps in the prediction horizon is increased.

Limitations of the numerical approach are the significant computational requirements and the difficulty in extending the approach to a control horizon of more than 1-step. Hence, in Chapter 7 we focussed on an analytical approach for determining the long-range predictive controller output. The analytical approach was shown to result in a significant reduction in computational requirements over the numerical approach presented earlier, while providing a very similar process output response.

In Chapter 7, it was also shown that the use of a multi-step prediction horizon leads to a more robust control strategy. In the two examples examined in this chapter, increasing the number of fuzzy partitions used by the fuzzy model led to better modelling accuracy and to better controller performance, especially when the prediction horizon used by the controller is small. The improvement achieved by increasing the number of fuzzy partitions beyond 3 was, however, quite marginal.

Adaptive control based on recursive least squares estimation of fuzzy model parameters using the control algorithm presented in Chapter 7 is demonstrated in Appendix A. In Appendix B, a dead time compensation scheme is demonstrated for first-order and second-order systems with dead times up to 2-sampling intervals.

Attention in Chapter 8 focused on extending the proposed predictive control algorithm to MISO systems. By using the CSTR system as an example, it was shown that it is possible to extend the fuzzy model to include an additional process input, where the additional input is one of the significant process disturbances. Doing this, resulted in a significant improvement in the output response to the effect of this disturbance, both with the one-step ahead predictive controller as well as the long-range predictive controller.

Subsequent research effort focused on extending the approach used for MISO systems to MIMO systems. It was shown that the design of a 2-input 2-output MIMO fuzzy model-based predictive controller can be approached by decomposing it into two MISO fuzzy model-based controllers. The optimal controller outputs of the MISO fuzzy model-based controllers were computed sequentially using separate objective functions. The resulting controller was shown to be able to cope with process nonlinearities and interaction between loops. Here again, the performance of the fuzzy model-based controller has been shown to be superior to a comparable controller using a linear process model.

In Chapter 11, it was shown that the presence of constraints can significantly alter the process output response of the unconstrained controller. A method of explicitly including the constraints in the optimisation process was therefore considered using the QP method. The use of this method was shown to lead to a significant improvement in the output response compared to merely clipping the controller output using the constraints. The flexibility of the QP method was also demonstrated by showing that process output constraints can also be easily incorporated in the optimisation process.

The results of our research study point to a model based predictive controller which has the following features:

- Can be used with highly nonlinear processes operating over a wide range.
- Uses long-range predictive control strategy to achieve better robustness.
- Fuzzy model can be identified using the least squares algorithm.
- Can be used with MISO and MIMO processes.
- Allows explicit handling of constraints on process inputs and outputs.
- Allows dead time compensation.

- Process model can adapt to process changes.

12.2 Some Guidelines on Design Parameters

The proposed controller requires a number of choices to be made by the designer. The values of the parameters used by the controller also need fine-tuning by the designer. Many of these decisions are best arrived at by trial and error studies. However, based on the limited amount of experience acquired from simulation studies, some guidelines are provided below to assist the designer in making some of these decisions.

Model Structure

The model sructure is best determined by trial and error studies. For most systems, representation using either first-order or second-order system dynamics appears to be sufficient.

The effect of including the constant term in the model is also best determined by trial and error studies.

Number of Fuzzy Partitions of the Input Space

Three to five partitions should be sufficient for most applications.

Number of Steps in Control Horizon

For applications which do not require explicit handling of constraints by the control algorithm, a control horizon of 1-step should be sufficient. The use of a control horizon of more than 1-step can lead to a more active controller output response which could have a negative effect on robustness.

Number of Steps in Prediction Horizon

Optimal performance is generally achieved with a prediction horizon of less than 10-steps. In the case of SISO systems with little or no noise, 2 to 5-steps seems to be sufficient. A longer prediction horizon is needed if the system contains significant amount of noise, dead time, or when there is interaction between control loops in the case of multivariable systems.

Dead time makes processes difficult to control making it necessary to use prediction horizons in excess of 10-steps. In such cases, better performance can be achieved using shorter prediction horizons with the dead time compensation scheme.

Gain of Filter 1

Studies seem to indicate that it is sufficient to set the value of this parameter to 1, especially in the case of the controller using a multi-step prediction horizon.

Gain of Filter 2

This parameter should preferably be set to a value less than 0.1. A high value will lead to a better response to setpoint changes under noise-free conditions, but studies seem to indicate poorer performance under noisy conditions and to unmodelled disturbances.

12.3 Recommendations for Further Work

Even though we have tried to ensure a sufficient level of exhaustiveness when carrying out this research study, given the limited amount of time for completing the study, it seems inevitable that compromises have to be made, some of which may be more important than others. Also, given the exploratory nature of the research carried out into the completely new control method proposed in this thesis, it appears logical that its performance should first be verified using simulated processes. Since the results of the exploratory tests using simulated processes appear encouraging, the next logical step forward is to test the controller using real processes. This can initially be done using a laboratory rig or a pilot plant, and subsequently on industrial-scale process plants.

Throughout this thesis, identification of the fuzzy process model was based on the trial and error method proposed by Takagi and Sugeno (1985). Even though this approach is conceptually clear, and was quite easy to apply in the case of the idealised processes that were examined in this thesis, considerably more difficulty may be encountered in the case of multivariable processes where relationships between inputs and outputs are poorly understood, and where the input/output data used for identification is corrupted with noise. Studies based on real procesess should throw more light on this issue. A number of papers have been published claiming more systematic approaches to model structure identification when using the fuzzy modelling approach used in this thesis (Sugeno and Kang, 1988; Nakamori and Ryoke, 1994). It should be interesting to evaluate the usefulness of such approaches in the context of our proposed control method.

It is the ultimate goal of research into intelligent control systems to be able to design a system that can autonomously achieve a high goal using the phenomena that we normally associate with "intelligence", i.e. the generalised, flexible, learning and adaptive capability that humans possess. In this thesis, only one chapter has addressed the issue of adaptive control, inspite of its importance. Considerably more research in this direction is needed, with emphasis on techniques such as UD factorisation to achieve better numerical robustness. The possibility also exists for using neural-network learning algorithms (Jang, 1993) and genetic algorithms (Linkens and Nyongesa, 1995a; 1995b) for fine-tuning the fuzzy partitioning of the input space to achieve better modelling accuracy.

Linearisation of the fuzzy process model was necessary to enable multi-step predictions and to enable the design of an analytical long-range predictive control algorithm. While the method of linearisation by weighting the fuzzy model parameters has been shown to lead to better controller performance over a similar controller using a single linear process model to represent a highly nonlinear process over a wide operating range, there may be scope for further improvement if a better method of linearisation can be found.

Even though the Takagi-Sugeno fuzzy modelling method allows fuzzy partitioning of the input space based on more than one variable, input space partitioning throughout this thesis has been based on just the current value of the process output to facilitate the design of controllers using an analytical approach. It is probably worthwhile carrying out investigations into the design of controllers which can capitalise on input space partitioning based on more than one variable.

Although we have shown how the coefficients of the matrices arising from model transformation during the formulation of the long-range predictive control algorithm can be determined, this has been limited to first-order and second-order processes. A more general approach which can accomodate higher-order processes, dead time, etc. will be useful. MBPC methods such as GPC and UPC use the Diaphontine equation to achieve this.

REFERENCES

1. Antsaklis, P.J. (1994). Defining intelligent control. *IEEE Control Systems Magazine*, 14(3): 4-5, 58-66.

2. Antsaklis, P.J. (1995). Intelligent learning control. *IEEE Control Systems Magazine*, 15(6): 5-7.

3. Astrom, K.J. & Wittenmark, B. (1973). On self-tuning regulators. *Automatica*, 9: 185-199.

4. Astrom, K.J. & Wittenmark, B. (1989). *Adaptive Control*, Addison Wesley, Reading, MA.

5. Bartolini, G., Casalino, G., Davoli, F., Mastretta, M., Minciardi, R. & Morten, E. (1982). Development of performance adaptive fuzzy controllers with application to continuous casting plants. In: R. Trappl (ed.). *Cybernatics and Systems Research*, North-Holland,Amsterdam, pp. 721-728.

6. Batur, C. & Kasparian, V. (1989). A real-time fuzzy self-tuning control. *Proceedings of IEEE International Conference on Systems Engineering*, pp. 1810-1815.

7. Bhat, N.V., Minderman, P.A., McAvoy, T.J. & Wang, N.S. (1990). Modelling chemical process systems via neural computation. *IEEE Control Systems Magazine*, April.

8. Bierman, G.J. (1977). *Factorisation Methods for Discrete Sequential Estimation*, Academic Press. New York.

9. Braake, H.T., Babuska, R. & Can, E.V. (1994). Fuzzy and neural models in predictive control. *Journal A*, 35(3): 44-51.

10. Brengel, D.D. & Seider, W.D. (1989). Multistep nonlinear predictive controller. *Ind. Eng. Chem. Res.*, 28(12): 1812-1822.

11. Brown, M. & Harris, C. (1994). *Neurofuzzy Adaptive Modelling and Control*, Prentice Hall International, U.K.

12. Box, G.E.P. & Jenkins, G.M. (1970). *Time Series Analysis, Forecasting and Control*, Holden Day, San Francisco.

13. Chunyu, H.T., Toguchi, K. Shenoi, S. & Fan, L.T. (1989). A technique for designing and implementing fuzzy logic control. *Proceedings of the 1989 American Control Conference*, Vol. 3, pp. 2754-5.

14. Clarke, D.W., Mohtadi, C. & Tuffs, P.S. (1987a). Generalised predictive control - Part I: the basic algorithm. *Automatica*, 23(2): 137-148.

15. Clarke, D.W., Mohtadi, C. & Tuffs, P.S. (1987b). Generalised predictive control - Part II: extensions and interpretations. *Automatica*, 23(2): 149-160.

16. Cutler, C. R. and Ramaker, B L (1980). Dynamic matrix control - a computer control algorithm, *Proc. Joint Automatic Control Conference*, San Francisco, California.

17. Czogala, E. & Pedrycz, W. (1981). On identification in fuzzy systems and its applications in control problems. *Fuzzy Sets and Systems*, 6: 73-83.

18. Daley, S. & Gill, K.F. (1986). A design study of a self-organising fuzzy logic controller. *Proc. Inst. Mech. Eng.*, 200: 59-69.

19. Dayal, B.S., Taylor, P.A. & Macgregor, J.F. (1994). The design of experiments, training and implementation of nonlinear controllers based on neural networks. *Canadian J. of Chemical Engineering*, 72: 1066-1079.

20. De Keyser, R.M.C., & Van Cauwenberghe, A.R. (1985). Extended prediction self-adaptive control, *Proc. of IFAC Identification and System Parameter Estimation*, York, U.K., pp. 1255-1260.

21. Deshpande, P.B., Caldwell, J.A. and Yerraprogoda, S.S. (1995). Should you use constrained model predictive control? *Chemical Engineering Progress*, 91(5): 65-72.

22. Di Massimo, C., Willis, M.J., Montague, G.A., Tham, M.T. and Morris, A.J. (1991). Bioprocess model building using artificial neural networks. *Bioprocess Engineering*, 7: 77-82.

23. Fortescu, T.R., Kershenbaum, L.S. & Ydstie, B.E. (1981). Implementation of self-tuning regulators with variable forgetting factors. *Automatica*, 17: 831-835.

24. Galluzzo, M., Cappellani, V. & Garofalo, U. (1991). Fuzzy control of pH using NAL. *International Journal of Approximate Reasoning*, 5(6): 505-519.

25. Garcia, C.E. & Morari, M. (1982). Internal Model control. 1. A unifying review and some new results. *Ind. Eng. Chem. Process Des. Dev.*, 21: 308-323.

26. Garcia, C.E. (1984). Quadratic/dynamic matrix control of nonlinear processes: an application to a batch reaction process. *AIChE Annual Meeting,* San Francisco, CA.

27. Garcia, C.E., Prett, D.M. & Morari, M. (1989). Model predictive control: theory and practice - a survey. *Automatica*, 25(3): 335-348.

28. Garcia, G.E. & Moreshedi, A.M. (1986). Quadratic programming solution of dynamic matrix control (QDMC). *Chemical Engineering Communication*, 46: 73-87.

29. Gingrich, C., Kuespert, K. & McAvoy, T.J. (1990). Modelling human operators using neural networks. *Proceedings of Instrument Society of America Conference*, October 1990.

30. Goodwin, G.C. & Sin, K.S. (1984). *Adaptive Filtering, Prediction and Control*, Prentice-Hall, Englewood Cliffs.

31. Graham, B. & Newell, R. (1988). Fuzzy identification and control of a liquid level rig. *Fuzzy Sets and Systems*, 26: 255-273.

32. Graham, B. & Newell, R. (1989). Fuzzy adaptive control of a first-order process. *Fuzzy Sets and Systems*, 31: 47-65.

33. Harris, C.J. Moore, C.G. & Brown, M. (1993). *Intelligent Control: Aspects of Fuzzy Logic and Neural Networks*, World Scientific Series in Robotics and Automated Systems - Vol. 6.

34. Higashi, M. & Klir, G.J. (1984). Identification of fuzzy relation systems. *IEEE Trans. on Systems, Man, and Cybernatics,* 14: 349-355.

35. Hong, Z. (1985). The applications of fuzzy and artificial intelligence method in building up blast furnace smelting process model. In: Sugeno, M. (ed.), *Industrial Applications of Fuzzy Control*, North-Holland, Amsterdam.

36. Hunt, K.J. & Sbarbaro, D. (1991). Neural networks for nonlinear internal model control. *IEE Proceedings-D*, 138(5): 431-438.

37. Hunt, K.J., Sbarbaro, D., Zbikowski, R. & Gawthrop, P.J. (1992). Neural networks for control systems - a survey. *Automatica*, 28(6): 1083-1112.

38. Jang, J. S. R. (1993). Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Trans. on Neural Networks*, 4(1): 156-159.

39. Johansen, T. A. (1994). Fuzzy model based control: stability, robustness, and performance issues. *IEEE Trans. on Fuzzy Systems*, 2(3): 221-234.

40. Kalman, R.N. (1958). Design of a self-optimising control system. *Trans. ASME*, 80(Series D): 468.

41. Kaminski, P.G., Bryson, A.E. & Schmidt, S.F. (1971). Discrete square-root filtering: a survey of current techniques. *IEEE Trans. Automatic Control*, 16: 727-735.

42. Kickert, *et al.* (1976). Application of a fuzzy controller in a warm water plant. *Automatica*, 12: 301.

43. King, P.J. & Mamdani, E.H. (1977). The application of fuzzy control systems to industrial processes. *Automatica*, 13: 235-242.

44. Kosko, B. (1992). *Neural Networks and Fuzzy Systems*, Prentice-Hall, Englewood Cliffs.

45. Kramer, M.A. & Leonard, J.A. (1990). Diagnosis using back-propagation neural networks - analysis and criticism. *Computers in Chem. Engineering*, 14: 1323-1338.

46. Lawson, C.L. & Hanson, R.J. (1974). *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs.

47. Lee, P.L. & Sullivan, G.R. (1988). Generic model control (GMC). *Computers in Chem. Eng.*, 12(6): 573-580.

48. Lee, C.C. (1990). Fuzzy logic in control systems: fuzzy logic controller, Parts I & II. *IEEE Trans. on Systems, Man, & Cybernetics*, 20(2): 404-435.

49. Lee, Y.C., Hwang, C. & Shih, Y.P. (1994). A combined approach to fuzzy model identification. *IEEE Trans. on Systems, Man, and Cybernatics*, 24(5): 736-743.

50. Li, W.C. & Biegler, L.T. (1990). Newton-type controllers for constrained nonlinear processes with uncertainty. *Ind. Eng. Chem. Res.*, 29(8):1647-1657.

51. Linkens, D.A. & Abbod, M.F. (1991). Self-organising fuzzy logic control for real-time processes. *Proc. IEE Control '91 Conf.*, Edinburgh, pp. 971-976.

52. Linkens, D.A. & Abbod, M.F. (1992). Self-organising fuzzy logic control and the selection of its scaling factors. *Trans. Inst. MC*, 14(3): 114-125.

53. Linkens, D.A. & Nyongesa, H. (1995a). Genetic algorithms for fuzzy control: Part 1, off-line system development and applications. *IEE Proceedings-D*, 142: 161-176.

54. Linkens, D.A. & Nyongesa, H. (1995a). Genetic algorithms for fuzzy control: Part 1, on-line system development and applications. *IEE Proceedings-D*, 142: 177-185.

55. Luyben, W.L. (1990). *Process Modelling, Simulation and Control for Chemical Engineers*, McGraw-Hill Publishing Company.

56. Maeda, A., Someya, R. and Funabashi, M. (1991). A self-tuning algorithm for fuzzy membership functions using a computational flow network. *Proc. of the IFSA '91*, Brussels.

57. Mamdani, E.H. & Assilan, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man-Machine Studies*, 7(1): 1-13.

58. Mamdani, E.H. (1976). Advances in the linguistic synthesis of fuzzy controllers. *Int. J. Man-Machine Studies*, 8(6): 669-678.

59. McAvoy, T.J. et al. (1989). Interpreting biosensor data via backpropagation. *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, IEEE, New York.

60. Montague, G.A., Willis, M.J., Morris, A.J. and Tham, M.T. (1991). Artificial neural network based multivariable predictive control. *ANN'91*, Bournemouth.

61. Montague, G.A., Tham, M.T., Willis, M.J. and Morris, A.J. (1992). Predictive control of distillation columns using dynamic neural networks. *Third IFAC Symposium DYCORD+'92*, Maryland, USA.

62. Moore, C.G. & Harris, C.J. (1992). Indirect adaptive fuzzy control. *Int. J. of Control*, 56(2): 441-468.

63. Morari, M. and Zafiriou, E. (1989). *Robust Process Control*, Englewood Cliffs: Prentice-Hall.

64. Morris, A.J., Montague, G.A. & Willis, M.J. (1994). Artificial neural networks: studies in process modelling and control. *Trans. IChemE*, 72(A).

65. Nahas, E.P., Henson, M.A. & Seborg, D.E. (1992). Nonlinear internal model control strategy for neural network models. *Computers in Chem. Engineering*, 16: 1039-1057.

66. Naidu, S., Zafirou, E. & McAvoy, T. (1990). Application of neural networks with detection of sensor failure during the operation of a control system. *IEEE Control Systems*, 10: 49-55.

67. Nakamori, Y., Suzuki, K. & Yamanaka, T. (1991). Model predictive control using fuzzy dynamic models. *Proceedings of IFSA'91, Brussels, Vol. Engineering*, pp135-138.

68. Nakamori, Y. & Ryoke, M. (1994). Identification of fuzzy prediction models through hyperellipsoidal clustering. *IEEE Trans. on Systems, Man, and Cybernetics*, 24(8): 1153-1173.

69. Nakamori, Y. (1994). Fuzzy modelling for adaptive process control. In Kandel, A. & Langholz, G. (eds.). *Fuzzy Control Systems*, CRC Press, Boca Rotan, 296-314.

70. Narendra, K.S. & Parthasarathy, K. (1990). Identification and control of dynamic systems using neural networks. *IEEE Trans. Neural Networks*, 1(1): 4-27.

71. Newell, R.B. & Lee, P.L. (1989). *Applied Process Control: A Case Study*, Prentice-Hall, New York.

72. Nie, J. & Linkens, D.A. (1995). *Fuzzy Neural Control: Principles, Algorithms and Applications*, Prentice-Hall, U.K.

73. Nomura, H., Hayashi, I. & Wakami, N. (1991). A self-tuning method of fuzzy control by descent method. *Proc. of the IFSA '91*, Brussels, pp. 155-158.

74. Ostergaard, J.J. (1977). Fuzzy logic control of a heat exchanger process. In: Gupta, M.M., Saridas, G.N. & Gaines, B.R. (Eds.). *Fuzzy Automata and Decision Processes*, North-Holland, Amsterdam.

75. Pedrycz, W. (1983). Numerical and applicational aspects of fuzzy relational equations. *Fuzzy Sets and Systems*, 11: 1-18.

76. Pedrycz, W. (1984). An identification algorithm in fuzzy relational systems. *Fuzzy Sets and Systems*, 13: 153-167.

77. Pedrycz, W. (1993). *Fuzzy Control and Fuzzy Systems*, Research Studies Press, John Wiley & Sons, New York.

78. Peterka, V. (1970). Adaptive digital regulation of noisy systems. *Second IFAC Symp. on Identification and Process Parameter Estimation*, Praque.

79. Peterson, T., Hernandez, E., Arkun, Y. and Schork, F.J. (1989). Nonlinear predictive control of a semi-batch polymerization reactor by an extended DMC. *Proc. of the 1989 American Control Conf.*, pp. 1534-1539.

80. Postlethwaite, B. (1991a). Empirical comparison of methods of fuzzy relational identification. *IEE Proceedings-D*, 138(3): 197-206.

81. Postlethwaite, B. (1991b). A model-based fuzzy controller. *Proceedings Intern. AMSE Conference "Signals & Systems"*, Warsaw, Poland, Vol. 1, pp. 15-26.

82. Postlethwaite, B. (1994). A model-based fuzzy controller. *Trans. IChemE*, 72(A): 38 - 46.

83. Postlethwaite, B. (1996). Building a model-based fuzzy controller. *Fuzzy Sets and Systems*, 79: 3-13.

84. Preub, H.P. (1993). Fuzzy control in process automation. *Int. J. Syetems Science*, 24(10): 1849-1861.

85. Procyk, T.J. & Mamdani, E.H. (1979). A linguistic self-organising process controller. *Automatica*, 15: 15-30.

86. Psichogios, D.C. & Ungar, L.H. (1991). Direct and indirect model based control using artificial neural networks. *Ind. Eng. Chem. Res.*, 30: 2564-2573.

87. Ragot, J. & Lamotte, M. (1993). Fuzzy logic control. *Int. J. Syetems Science*, 24(10): 1825-1848.

88. Richalet, J., Rault, A., Testud, J.L. & Papon, J. (1978). Model predictive heuristic control: applications to industrial processes. *Automatica*, 14: 413.

89. Richalet, J. (1993). Industrial applications of model based predictive control. *Automatica*, 29(5): 1251-1274.

90. Ridley, J.N., Shaw, I.S. and Kruger, J.J. (1988). Probabilistic Fuzzy Model for Dynamic Systems. *Electronic Letters*, 24(14): 890-892.

91. Rock D. & Guerin, D. (1992). Applying AI to statistical process control. *AI Expert*, 7(9): 30-35.

92. Saint-Donat, J., Bhat, N.V. & McAvoy, T.J. (1991). Neural net based model predictive control. *Int. J. Control*, 54: 1453-1468.

93. Sanchez, E. (1976). Resolution of composite relation equations. *Information & Control*, 30: 38-48.

94. Shah, I.M. & Rajamani, R.K. (1991). Self organising control of pH in a stirred tank reactor. *Expert Systems in Mineral and Metal Processing. Proceedings of the IFAC Workshop*, pp. 131-7.

95. Shaw, I.S. & Kruger, J.J. (1992). New fuzzy learning model with recursive estimation for dynamic systems. *Fuzzy Sets & Systems*, 48: 217-229.

96. Sing, C.H. & Postlethwaite, B. (1996). Fuzzy relational model based control applying stochastic and iterative methods for model identification. *Trans. IChemE*, 74(A): 70-76.

97. Sistu, P.B. & Bequette, B.W. (1990). Process identification using nonlinear programming techniques. *Proc. of the 1990 American Control Conference*, San Diego.

98. Skrjanc, I. & Matko, D. (1993). Fuzzy predictive controller with adaptive gain. *Proc. of Conf. on Advances in Model-Based Predictive Control*, Oxford, England, vol. 1, pp. 89 - 100.

99. Soeterboek, R. (1991). *Predictive Control: A Unified Approach*, Prentice-Hall International, U.K.

100. Song, J.J. & Park, S. (1993). A fuzzy dynamic learning controller for chemical process control. *Fuzzy Sets & Systems*, 54: 121-133.

101. Sugeno, M. (1985). An introductory survey of fuzzy control. *Information Sciences*, 36: 59-83.

102. Sugeno, M. & Kang, G.T. (1986). Fuzzy modelling and control of multilayer incinerator. *Fuzzy Sets and Systems*, 18: 329-346.

103. Sugeno, M. & Kang, G.T. (1988). Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28: 15-33.

104. Sugeno, M. & Yasukawa, T. (1993). A fuzzy-logic based approach to qualitative modelling. *IEEE Trans. on Fuzzy Systems*, 1(1): 7-29.

105. Takagi, T. & Sugeno, M. (1985). Fuzzy identification of systems and its application to modelling and control. *IEEE Trans. on Systems, Man, and Cybernetics*, 15(1): 116-132.

106. Tanaka, K. & Sano, M. (1991). A new tuning method of fuzzy controllers. *Proc. of the IFSA '91*, Brussels, pp. 207-210.

107. Tanaka, T. *et al.* (1991). A trouble forecasting system by a multi-neural network on a continuous casting process of steel production. In: T.Kohonen *et al.* (eds.). *Artificial Neural Networks, Vol. 1*, North-Holland, Amsterdam.

108. Tham, M.T., Morris, A.J. & Montague, G.A. (1988). Generalisation of GMV self-tuning controller synthesis. *IFAC Symposium on Identification and System Parameter Estimation*, Beijing, China.

109. Tham, M.T., Vagi, F., Morris, A.J. and Wood, R.K. (1991a). Multivariable multirate self-tuning control: a distilllation column case study. *Proc. IEE Part D*, 138(1): 9-24.

110. Tham, M.T., Morris, A.J., Montague, G.A. & Lant, P.A. (1991b). Soft sensers for process estimation and inferential control. *J. Proc. Control*, 1: 3-14.

111. Tong, R.M. (1978a). Analysis and control of fuzzy systems using finite discrete relations. *Int. J. of Control*, 27(3): 431-440.

112. Tong, R.M. (1978b). Synthesis of fuzzy models for industrial processes - some recent results. *Int. J. Gen. Systems*, 4: 143-162.

113. Tong, R.M. (1979). The construction and evaluation of fuzzy models. In: Gupta, M.M., Ragade, R.K. & Yager, R.R. (eds.). *Advances in Fuzzy Set Theory and Applications*, North-Holland, Amsterdam.

114. Tong, R.M. (1980). The evaluation of fuzzy models derived from experimental data. *Fuzzy Sets and Systems*, 4: 1-12.

115. Tong, *et al.* (1980). Fuzzy control of the activated sludge wastewater treatment plant. *Automatica*, 16: 695.

116. Tsang, T.T.C. & Clarke, D.W. (1988). Generalised predictive control with input constraints. *IEE Proceedings-D*, 135(6): 451-460.

117. Umbers, I.G. & King, P.I. (1980). An analysis of human decision making in cement kiln control and the implication for automation. *Int. J. Man-Machine Studies*, 12(1): 11-23.

118. Vagi, F., Tham, M.T., Morris, A.J., Wood, R.K. (1991). Multivariable adaptive control of a binary distillation column. *Canadian Journal of Chemical Engineering*, 69(4): 997-1009.

119. Wang, F.Y. & Cameron, I.T. (1994). Control studies on a model evaporation process - constrained state driving with conventional and higher relative degree systems. *J. Proc. Cont.*, 4(2): 59-75.

120. Wang, L. and Langari, R. (1994). Identification of time-varying fuzzy systems. *Proc. of the First NAFIPS/IFIS/NASA '94 International Joint Conference*, San Antonio, Texas, 365-369.

121. Wang, L.X. & Mendel, J.M. (1992). Generating fuzzy rules by learning from example. *IEEE Trans. on Systems, Man, and Cybernatics*, 22(6): 1414-1427.

122. Wang, L.X. (1994). *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*, Prentice Hall, Englewood Cliffs.

123. Wegmann, H. & Mohr, D. (1993). Fuzzy logic, first application with SIMATIC S5. *Automatisierungstechnische Praxis*, 35(3): 167-171.

124. White, A. & Sofge, D.A. (1992). *Handbook of Intelligent Control: Fuzzy, Neural and Adaptive Approaches*, Van Nostrand.

125. Wilkinson, D.J., Morris, A.J. & Tham, M.T. (1994). Multivariable constrained predictive control (with application to high performance distillation). *Int. J. Control,* 59(3)841-862.

126. Willis, M.J., Di Massimo, C., Montague, G.A., Tham, M.T. and Morris, A.J. (1991). Artificial neural networks in process engineering. *IEE Proceedings-D,* 138(3): 256-266.

127. Willis, M.J. & Montague, G.A. (1993). Auto-tuning PI(D) controllers with artificial neural networks. *Proc. IFAC World Congress,* Sydney, Australia, 4: 61-64.

128. Xu, C.W. & Lu, Y.Z. (1987). Fuzzy model identification and self-learning for dynamic systems. *IEEE Trans. on Systems, Man, and Cybernatics,* SMC-17(4): 683-689.

129. Xu, C.W. (1989). Fuzzy systems identification. *IEE Proceedings,* 4(136): 146-150.

130. Yamashita, Y., Matsumoto, S. & Suzuki, M. (1988). Start-up of a catalytic reactor by fuzzy controller. *J. of Chem. Eng. of Japan,* 21(3): 277-281.

131. Yamazaki, T. & Mamdani, E.H. (1982). On the performance of a rule-based self-organising controller. *Proc. of IEEE Conf. on Applications of Adaptive and Multivariable Control,* Hull, England, pp. 50-55.

132. Yamazaki, T. (1993). The formulation of a simple fuzzy predictive controller with fuzzy modelling. *First Asian Fuzzy Systems Symposium,* Singapore.

133. Yamazaki, T. (1994). The formulation of a simple fuzzy model tuning predictive controller for MIMO systems. *IEEE World Congress on Computational Intelligence,* Orlando, Florida.

134. Yasunobu, S. & Miyamoto, S. (1985). Automatic train operation by predictive fuzzy control. In: M.Sugeno (ed.). *Industrial Application of Fuzzy Control,* North-Holland, Amsterdam, pp. 1-18.

135. Ydstie, B.E. (1984). Extended horizon adaptive control, *IFAC World Congress,* Budapest, Hungary.

136. Yonekura, M. (1981). The application of fuzzy sets theory to temperature of box annealing furnaces using simulation techniques. *Proceedings of the Eighth IFAC World Congress,* Tokyo.

137. Zadeh, L. A. (1965). Fuzzy sets. *Information and Control,* 8: 338-353.

138. Zadeh, L.A. & Chang, S.L. (1972). On fuzzy mapping and control. *IEEE Trans. on Systems, Man, and Cybernatics.*

139. Zadeh, L.A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. on Systems, Man, & Cybernetics*, 3: 28-44.

140. Zadeh, L.A. (1994). Berkeley initiative in soft computing. *IEEE Industrial Electronics Society Newsletter*, 41(3): 8-10.

Appendix A

ADAPTIVE CONTROL

A.1 Introduction

The model used by the fuzzy model-based predictive controller described in Chapter 7 has been assumed to be a fixed model identified from the open loop response of the process prior to the implementation of the control system. Physical systems are, however, generally subject to long term changes due to wear and tear on the physical plant parts as well as the sensors. The fixed process model will begin to fail and manifest itself through imprecise control. In this section we will therefore examine a method for online modification of the model.

Adaptive control systems may be needed either for coping with changes in process characteristics over time or for coping with process nonlinearities. Even though the nonlinear modelling capability of the fuzzy process model has been shown to lead to significant performance improvement over a linear process model in the non-adaptive controller, we also wish to examine if adaptation can lead to improvement in the performance of the controller when applied to control nonlinear time-invariant processes.

Various schemes have been proposed for online learning/adaptation in fuzzy logic based controllers. One of the earliest attempts was made by Procyk and Mamdani (1979) with their concept of self-organising fuzzy logic controller (SOFLC) which combines system identification with control. The SOFLC is a hierarchical controller which has two levels. The basic level is a simple fuzzy logic controller. The second level is the self-organising level which supervises the basic level by modifying the existing rule base and generating new rules based on performance feedback.

The use of neuro-fuzzy approaches have gained in popularity over the past few years. Various schemes (Wang, 1994; Jang, 1993; Nie and Linkens, 1995) have been proposed for implementing a fuzzy inference system into a neural network architecture and applying back-propagation gradient descent algorithms using the mean squared error (MSE) performance function for the tuning of the model parameters. Gradient descent is a technique in which each parameter being adapted is altered according to that parameter's current effect on the error. For such schemes to be useful in in real-time applications, it is essential that the rate of convergence is sufficiently fast. Also, gradient descent algorithms do not guarantee that the absolute minimum error solution can be found.

Methods for online adaptation of relational fuzzy models have also been proposed. Although adaptation using relational models is generally quite straight-forward, this advantage must be viewed in the context of known limitations of relational modelling methods such as significant computational requirements and susceptibility to problems arising from incomplete models.

On the other hand, adaptive techniques in filtering, prediction and control have been extensively studied for more than a decade in the case of linear discrete-time systems. As in the case of controller design, it seems natural that we should examine this body of knowledge in our search for a suitable model adaptation scheme. In contrast to off-line methods for parameter estimation, on-line estimation requires that the parameter estimates be recursively updated within the time limit imposed by the sampling period. This necessitates the use of relatively simple algorithms. Many different estimation schemes are available including stochastic approximation, least squares, extended and generalised least squares, instrumental variables, and maximum likelihood. A thorough discussion of such techniques is available in the text by Goodwin and Sin (1984). Probably the most commonly used algorithm is the least squares algorithm. Since this algorithm has also been used for off-line fuzzy model identification, it appears to be the most logical choice for on-line parameter estimation.

A.2 Self-Tuning Control

The adaptive control method used by our controller is based on the widely accepted *self-tuning* concept of identifying the system using measured input and output data and then forming an appropriate controller using the identified system. This is illustrated in broad terms in Figure A.1. A self-tuning controller has 3 main elements. There is a standard feedback law in the form of a difference equation which acts upon a set of values such as the measured output and feedforward signals, the current setpoint, etc., and which produces the new control action. A recursive parameter estimator monitors the plant's input and output and computes an estimate of the plant dynamics in terms of a set of parameters in a prescribed structural model. The parameter estimates are fed into the control design algorithm which then provides a new set of coefficients for the feedback law. The control design algorithm simply accepts current estimates and ignores their uncertainties using the *certainty-equivalent* principle. The rationale for using this approach is that, although there may be poor control during the tuning phase, this can be minimised by taking special precautions, but the overall algorithm is considerably simplified. A self-tuner which includes both an estimate of a standardised process model and a control design stage is called an *explicit* self-tuner. A further modification is possible by omitting the control-design stage in an *implicit* self-tuner where the process model is reformulated such that the estimator directly produces the coefficients of the required control law.

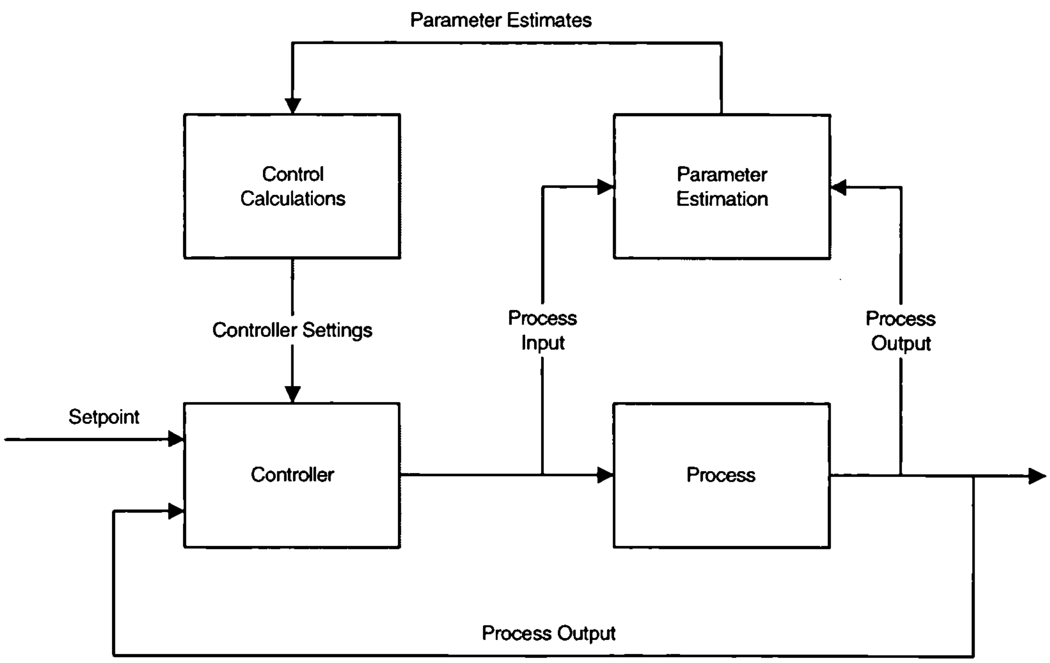


Figure A.1: Explicit self-tuning control system.

The idea of a self-tuning system is not new. The realization of the idea into practical algorithms, however, is a recent development made possible by low-cost digital computing equipment. For example, Kalman (1958) described a self-optimizing controller in 1958, but the algorithm was impractical at the time due to digital computer limitations of cost, speed and size. The current interest in self-tuning control was first stimulated by (amongst others) the Czech researcher Peterka (1970) who showed how system identification and controller synthesis could be combined into one iterative procedure for process control. Analysis of the asymptotic properties of a direct self-tuning regulator was first given in 1973 by Astrom and Wittenmark (1973) who also coined the phrase 'self-tuning'. Their regulator was based on least-squares estimation and minimum-variance control. Since then, several other self-tuning control methods have emerged including linear quadratic (LQ), pole placement and generalised predictive control (GPC) all of which are based on linear process models.

Figure A.2 shows how the concept of an explicit self-tuner has been adapted to fit the nonlinear fuzzy proess model used in this study.

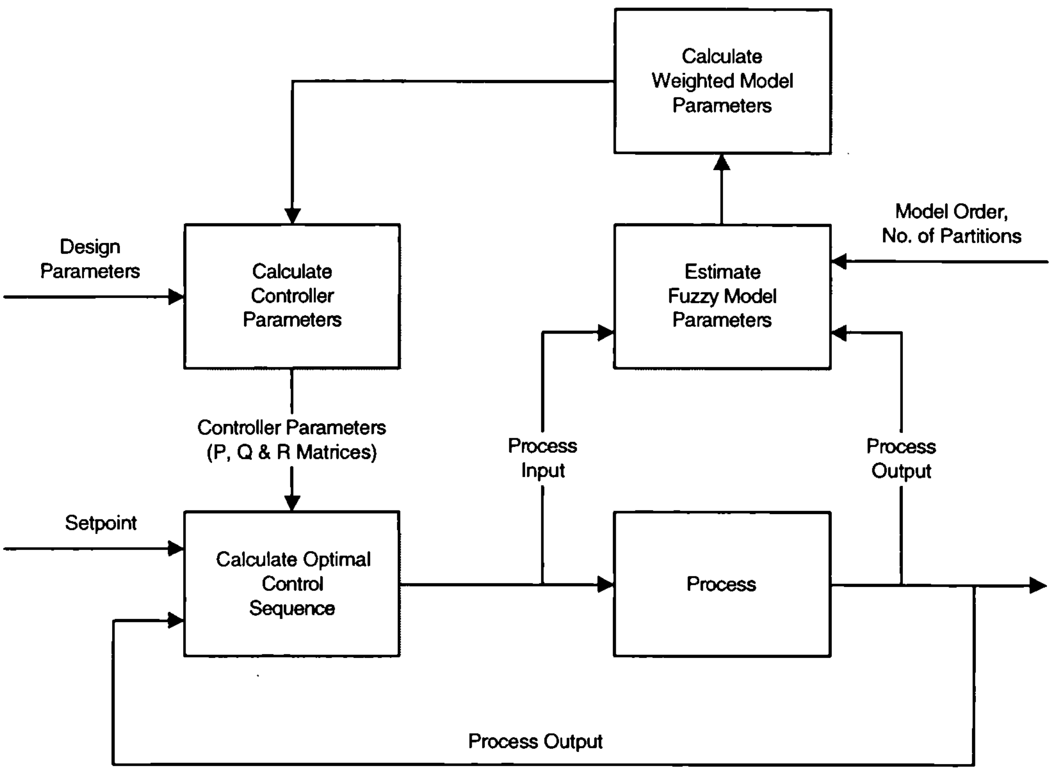


Figure A.2: Adaptive fuzzy model based predictive controller

A.3 Recursive Least Squares Algorithm

For the purpose of facilitating the application of parameter estimation techniques, the equation (4.5) can be re-formulated in the following form:

$$y_m(t+1) = \theta(t) \cdot \varphi(t) \tag{A.1}$$

where the regressor vector $\varphi(t)$ is determined from $X(t)$ and $\beta(t)$ and is given by:

$$\varphi(t) = \begin{bmatrix} \beta_1 y(t) \\ \vdots \\ \beta_1 u(t-l+1) \\ \beta_1 \\ \vdots \\ \beta_p y(t) \\ \vdots \\ \beta_p u(t-l+1) \\ \beta_p \end{bmatrix} \tag{A.2}$$

and the parameter vector $\theta(t)$ is determined from $\Phi(t)$ and is given by:

$$\theta(t) = \left[a_1^1 \cdots a_j^1 \quad b_1^1 \cdots b_l^1 \quad k_1 \quad \cdots \quad a_1^p \cdots a_j^p \quad b_1^p \cdots b_l^p \quad k_p\right] \tag{A.3}$$

The least squares estimates of model parameters is found by minimising the following cost function, J with respect to $\theta(t)$:

$$J = \sum_{t=1}^{N} [y(t) - \theta(t) \cdot \varphi(t)]^2 \tag{A.4}$$

where N is the number of data points and $y(t)$ is the measured value of the output. To be useful in self-tuning control the parameter estimation scheme should be iterative, allowing the estimated model of the system to be updated at each sampling interval as new data become available. The model based on past information (summarised in $\theta(t-1)$) is used to obtain an estimate $y_m(t)$ of the current output. This is then compared with the observed output $y(t)$ to generate an error $\varepsilon(t)$. This is turn generates an update to the model which corrects $\theta(t-1)$ to the new value $\theta(t)$. This recursive 'predictor-corrector' form allows significant saving in computation. Instead of recalculating the least squares estimate in its entirety, requiring the storage of all previous data, it is both efficient and elegant to merely store the 'old' estimate calculated at time t-1, denoted by $\theta(t-1)$, and to obtain the 'new' estimate $\theta(t)$ by an updating step involving the new observation only. Details of the derivation of the recursive least squares (RLS) algorithm are available from texts by Astrom and Wittenmark (1989) and

Welstead and Zarrop (1991). The algorithm is presented below in the form suitable for implementation in self-tuning systems:

$$
\begin{aligned}
&\theta(t) = \theta(t-1) + K(t)\cdot\varepsilon(t) \\
&K(t) = \frac{P(t-1)\cdot\varphi(t)}{\lambda + \varphi^T(t)\cdot P(t-1)\cdot\varphi(t)} \\
&P(t) = \frac{1}{\lambda}\left[I - K(t)\cdot\varphi^T(t)\right]P(t-1) \\
&\varepsilon(t) = y(t) - \varphi^T(t)\cdot\theta(t-1)
\end{aligned}
\tag{A.5}
$$

where $P(t)$ is called the covariance matrix of the estimation error, $\varepsilon(t)$ is the prediction error, I is the identity matrix, λ is the forgetting factor, and $K(t)$ is the Kalman (feedback) filter gain. The parameter estimates obtained using the above algorithm are unbiased only if the system has no random noise or the noise is thought to be white.

The forgetting factor, λ, $0 \le \lambda \le 1$, is an exponential weighting factor which weights past values of the process information. A small value of λ implies fast forgetting of past data, whereas a value close to unity implies slow forgetting of past data. Thus, the forgetting factor can be used to influence the rate of adaptation of the parameter estimates. In practice, $0.95 \le \lambda \le 0.99$ is normally used. An enhancement to the above RLS algorithm is to allow the exponential forgetting factor λ to vary. This variation should allow the parameter estimates to adapt more rapidly when the process is disturbed and less rapidly when the process remains at setpoint under good control. A method for achieving this is outlined by Fortescue, Kershenbaum and Ydstie (1981).

To start the above algorithm, it is necessary to initialize P and θ. Large initial values of P means that the user expresses little confidence in $\theta(0)$, while small values mean that $\theta(0)$ is good and therefore, no big fluctuations around this value are anticipated.

It is well known in numerical analysis that considerable accuracy may be lost when a least-squares problem is solved by forming and solving the normal equations. The reason is that the measured values are squared unnecessarily. In particular, rounding errors can accumulate and cause errors to occur in both the estimated parameters and the covariance matrix. Apart from the loss of accuracy, the calculated covariance matrix may become negative semi-definite with consequent divergence of the parameter estimates. This may happen in computers with limited computational accuracy where the round-up error at each recursion is large. In a similar vein, numerical problems can occur during estimator runs involving thousands of recursions in which normally negligible round-up errors are allowed to accumulate. It should be stressed that in most circumstances the standard recursion is completely adequate. Despite this, most users of self-tuning algorithms and estimators will employ the numerically robust RLS versions, such as square-root decomposition (Kaminski *et al.*, 1971) or UD factorization

(Bierman, 1977) as a matter of good engineering practice. Since the nature of our investigations is intended to be exploratory, we will begin by using the RLS algorithm in the basic form as presented above and will only consider enhancements if problems are encountered.

A.4 Application to Control of Liquid Level

To examine how the performance of the adaptive controller is affected by changes in process conditions, the cross-sectional area, A of the tank used in the liquid level simulation (refer to Section 4.3.1) was varied.

Figures A.3 to A.10 show the adaptive controller's performance when the cross-sectional area of the tank was maintained the same as that used during off-line model identification (i.e. A = 10 cm^2). Model adaptation was commenced 1900 seconds after the start of a simulation run. The response up to 1900 seconds therefore shows the performance of the non-adaptive controller using the model identified from the open-loop response and provides a reference for comparing the performance of the controller after adaptation. Calculation of the P matrix was initiated at the start of a simulation run even though the switch-over from non-adaptive to adaptive control occured much later. It was found that initiating the calculation of the P matrix much earlier enabled a smooth transition from non-adaptive to adaptive control.

Comparison of the process output responses up to 1900 seconds shows that generally better performance can be achieved by using fuzzy process models before model adaptation is initiated. Adaptation allows fine-tuning of model parameters to local operating conditions. An improvement in performance is noticed in the case of the 1-step ahead predictive controller using the linear process model (Figures A.3 and A.7). The improvement is, at most, marginal with fuzzy process models (Figures A.4 to A.6 and Figures A.8 to A.10). Close examination will reveal that adaptation can introduce sluggishness in the output response increasing both the initial rise time as well as the time needed for offset correction at the new steady-state level resulting in a more pronounced overshoot. This can lead to an overall deterioration in performance which is most significant in the case of 3 and 5-partition fuzzy models.

An important observation is that there is very little difference in the performance of 1-step ahead predictive controllers when using different process models after adaptation is completed to the setpoint changes used in our study. Figures A.11 and A.12 show that the difference in the performance of controllers using fuzzy and linear process models after adaptation becomes more apparent when the prediction horizon is increased.

It was found that the use of more than two partitions can affect the robustness of the controller. Adaptation repeatedly carried out very close to the fuzzy partitioning points caused wide fluctuations and drift in the parameter estimates. This is probably due to the influence of round-up errors when the membership function is very close to zero. Such problems can probably be avoided by switching off the adaptation in the vicinity of the fuzzy partitioning points.

Figures A.13 and A.14 examine the performance of the controller using the 2-partition fuzzy model under conditions similar to that used for initial model identification. Adaptation has improved the performance of the 1-step ahead predictive controller in the range 10 to 15 cm. but has caused a slight deterioration in the performance in the range 90 to 95 cm. It will also be noticed that very good performance at both levels is provided by the 2-steps ahead predictive controller when using a non-adaptive model. The combined use of a fuzzy model and a prediction horizon of 2-steps seems sufficient to cope with the process nonlinearities when process conditions during operation are identical to that during identification of the off-line model and no real advantage is gained by using an adaptive controller.

Figure A.15 examines the performance of our proposed controller when there is a 10 percent change in the cross-sectional area of the tank. An improvement is achieved in the performance of the 1-step ahead predictive controller after adaptation, but it will also be observed that the performance of the 2-steps ahead predictive controller using the non-adaptive model seems sufficient to cope with the small changes in the cross-sectional area and, therefore, no real advantage will be gained through adaptive control.

Significant changes in the cross-sectional of 50 percent (Figure A.16) are probably better handled through model adaptation than by extending the prediction horizon.

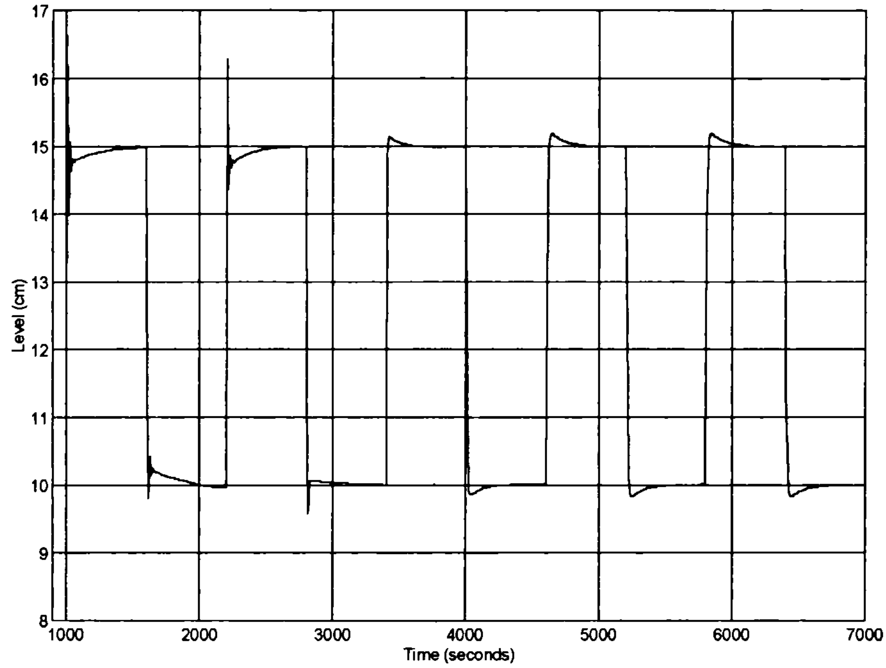


Figure A.3: Process output response to setpoint changes between 10 and 15 cm. when using 1-step ahead predictive controller with linear model. ($K_{f_1} = 1; K_{f_2} = 0.05;\ \lambda = 0.98; A = 10$ cm^2)

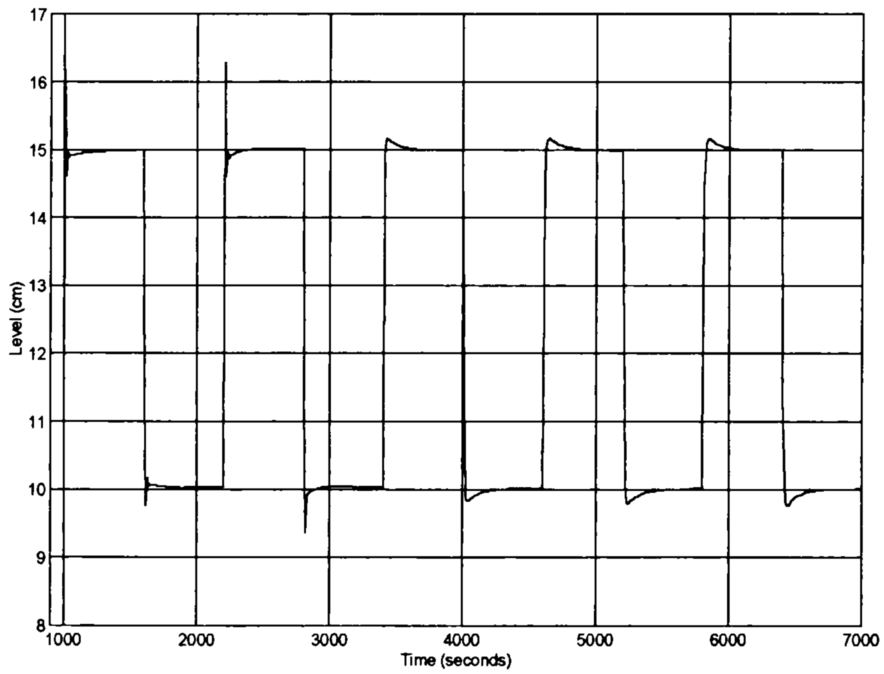


Figure A.4: Process output response to setpoint changes between 10 and 15 cm. when using 1-step ahead predictive controller with 2-partition fuzzy model. ($K_{f_1} = 1; K_{f_2} = 0.05;\ \lambda = 0.98; A = 10$ cm^2)

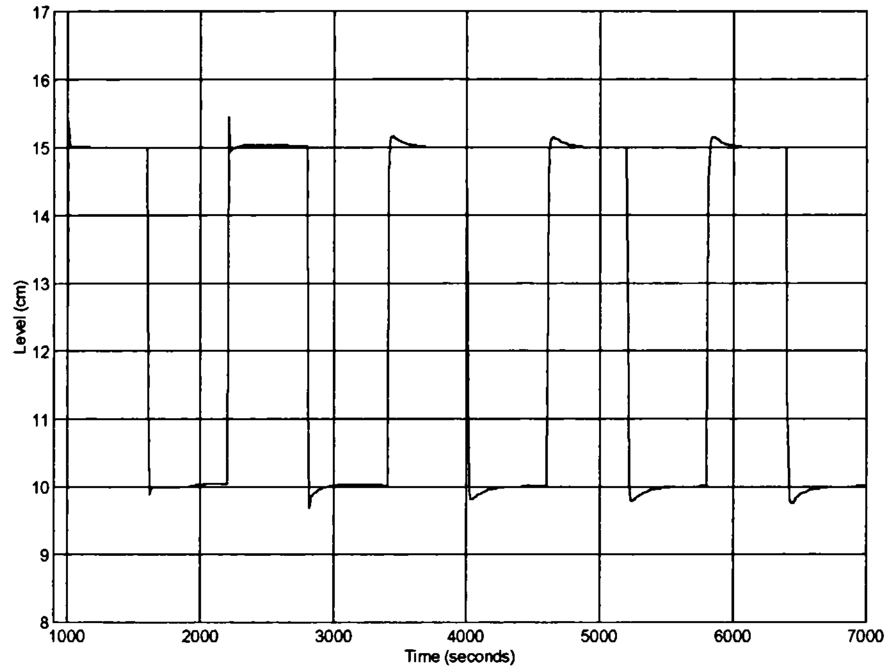


Figure A.5: Process output response to setpoint changes between 10 and 15 cm. when using 1-step ahead predictive controller with 3-partition fuzzy model. ($K_{f_1} = 1; K_{f_2} = 0.05;\ \lambda = 0.98; A = 10$ cm^2)

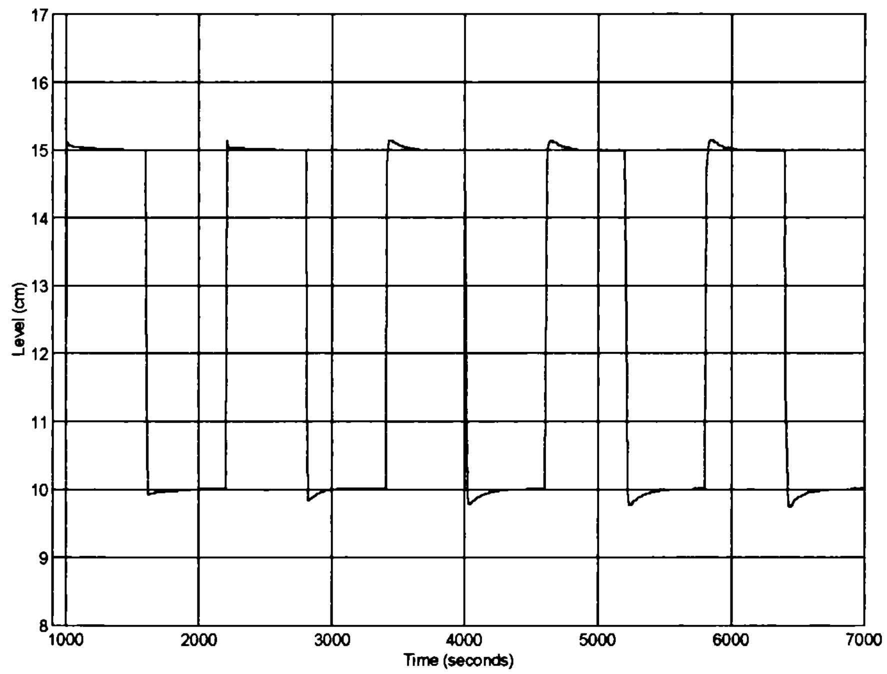


Figure A.6: Process output response to setpoint changes between 10 and 15 cm. when using 1-step ahead predictive controller with 5-partition fuzzy model. ($K_{f_1} = 1; K_{f_2} = 0.05;\ \lambda = 0.98; A = 10$ cm^2)

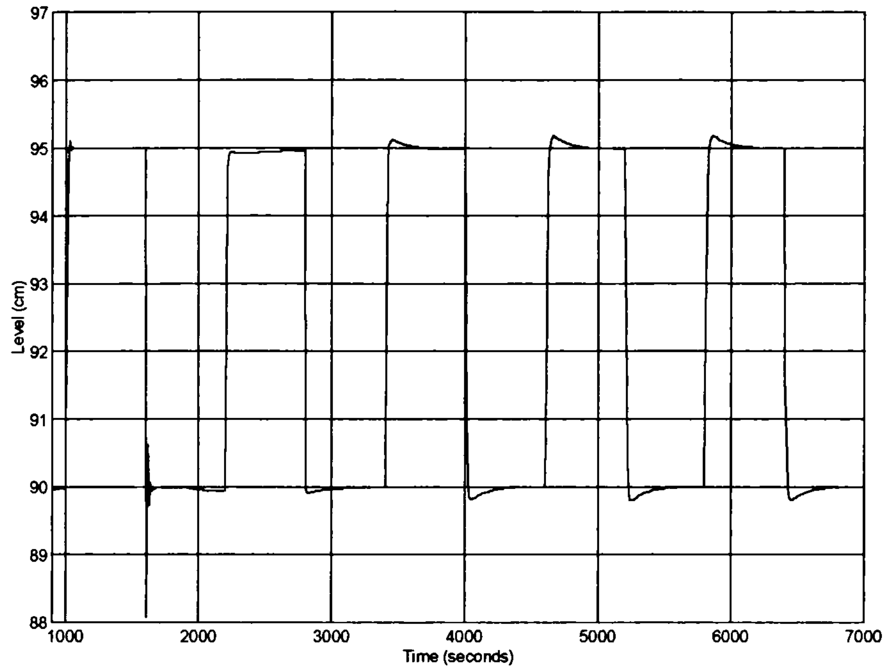


Figure A.7: Process output response to setpoint changes between 90 and 95 cm. when using 1-step ahead predictive controller with linear model. ($K_{f_1} = 1; K_{f_2} = 0.05;\ \lambda = 0.98; A = 10\ \text{cm}^2$)

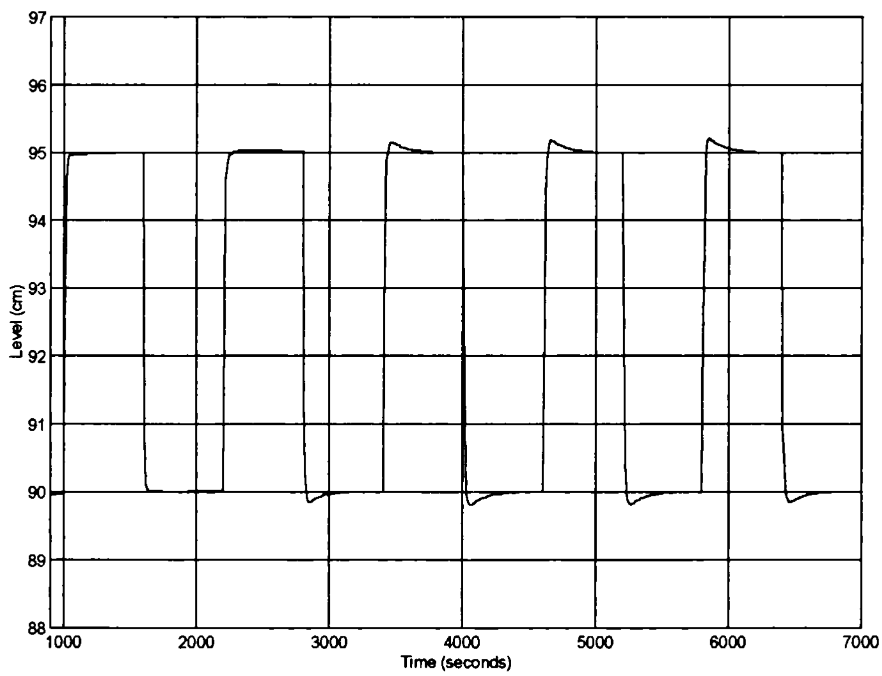


Figure A.8: Process output response to setpoint changes between 90 and 95 cm. when using 1-step ahead predictive controller with 2-partition fuzzy model. ($K_{f_1} = 1; K_{f_2} = 0.05;\ \lambda = 0.98; A = 10\ \text{cm}^2$)

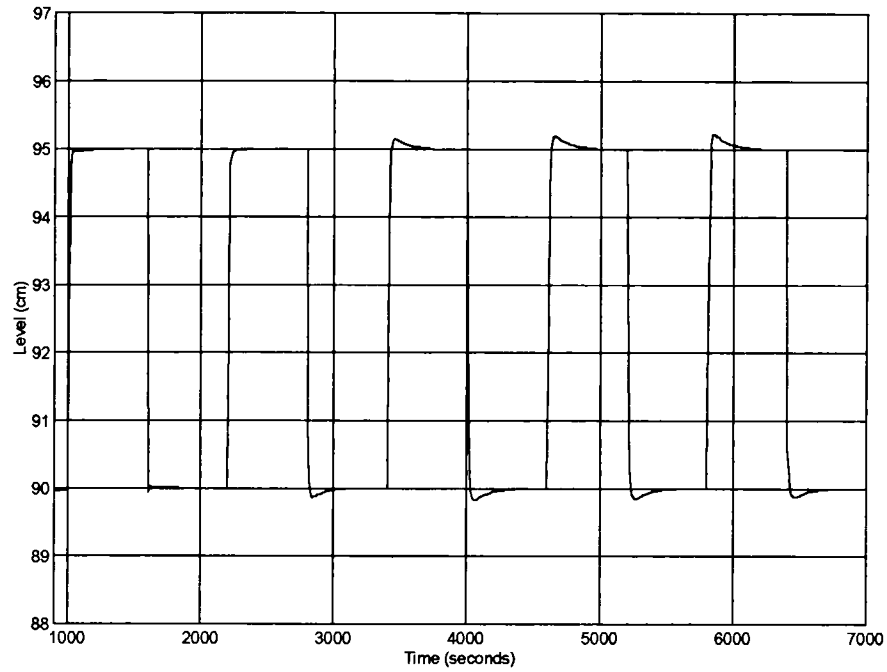


Figure A.9: Process output response to setpoint changes between 90 and 95 cm. when using 1-step ahead predictive controller with 3-partition fuzzy model. ($K_{f_1} = 1; K_{f_2} = 0.05;\ \lambda = 0.98; A = 10$ cm^2)

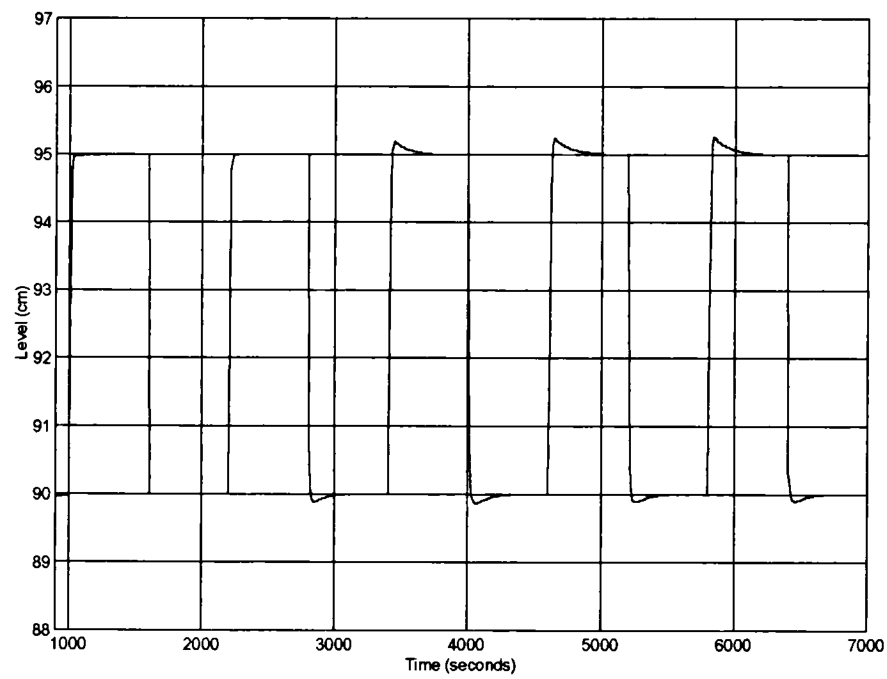


Figure A.10: Process output response to setpoint changes between 90 and 95 cm. when using 1-step ahead predictive controller with 5-partition fuzzy model. ($K_{f_1} = 1; K_{f_2} = 0.05;\ \lambda = 0.98; A = 10$ cm^2)

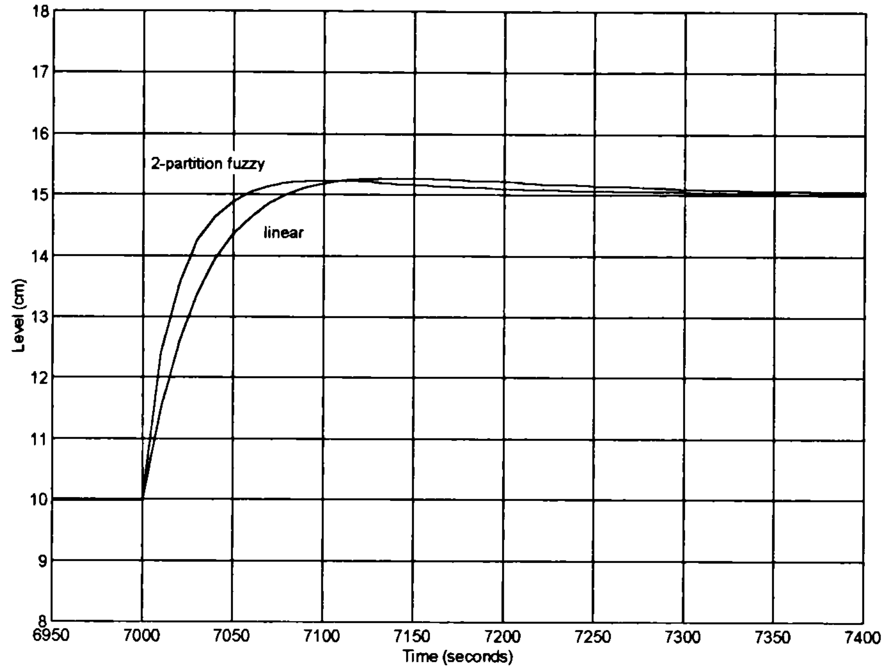


Figure A.11: Process output responses to setpoint changes between 10 and 15 cm. when using 3-steps ahead predictive controller with linear and fuzzy models. ($K_{f_1} = 1; K_{f_2} = 0.05$; $\lambda = 0.98; A = 10$ cm²)

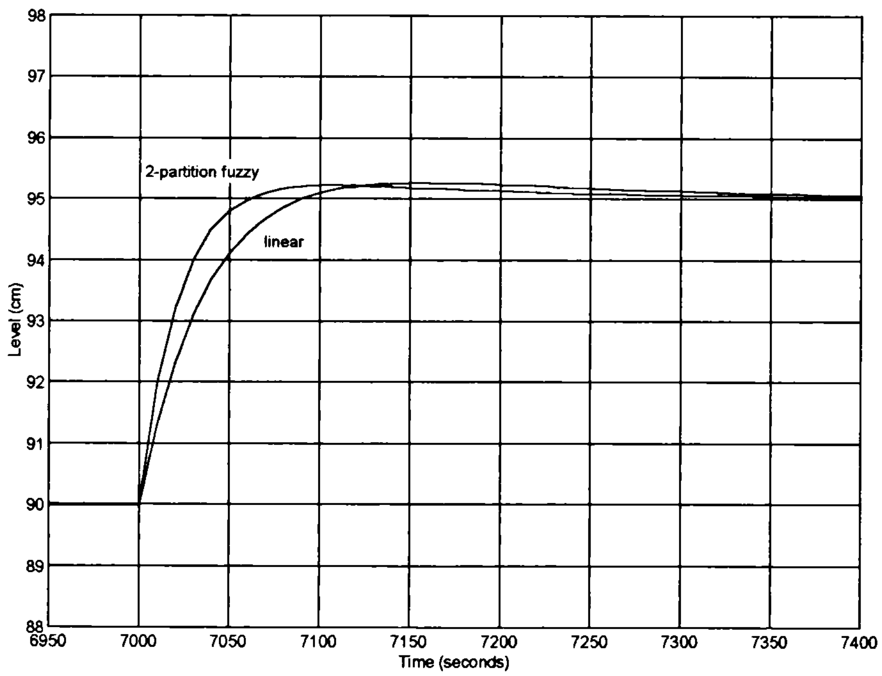


Figure A.12: Process output responses to setpoint changes between 90 and 95 cm. when using 3-steps ahead predictive controller with linear and fuzzy models. ($K_{f_1} = 1; K_{f_2} = 0.05$; $\lambda = 0.98; A = 10$ cm²)

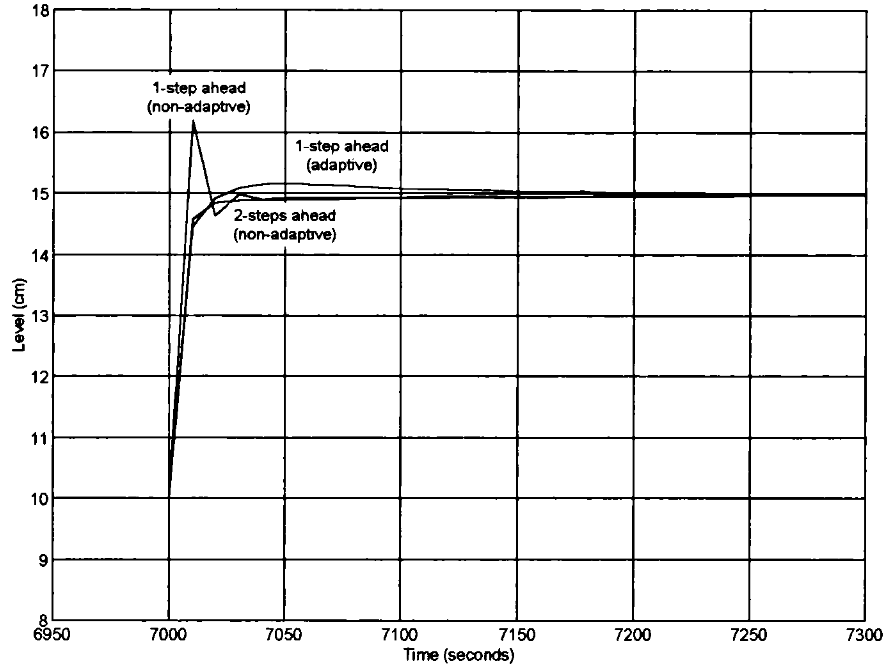


Figure A.13: Comparison of process output responses to setpoint change between 10 and 15 cm. when using proposed controller with 2-partition fuzzy model. ($K_{f_1} = 1; K_{f_2} = 0.05;\ \lambda = 0.98; A = 10\ \text{cm}^2$)

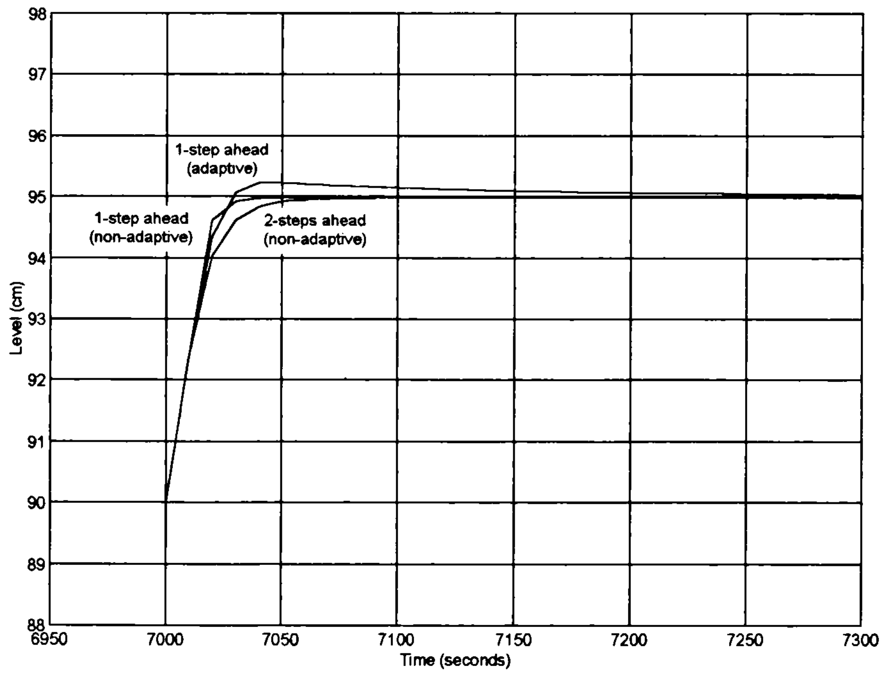


Figure A.14: Comparison of process output responses to setpoint change between 90 and 95 cm. when using proposed controller with 2-partition fuzzy model. ($K_{f_1} = 1; K_{f_2} = 0.05;\ \lambda = 0.98; A = 10\ \text{cm}^2$)

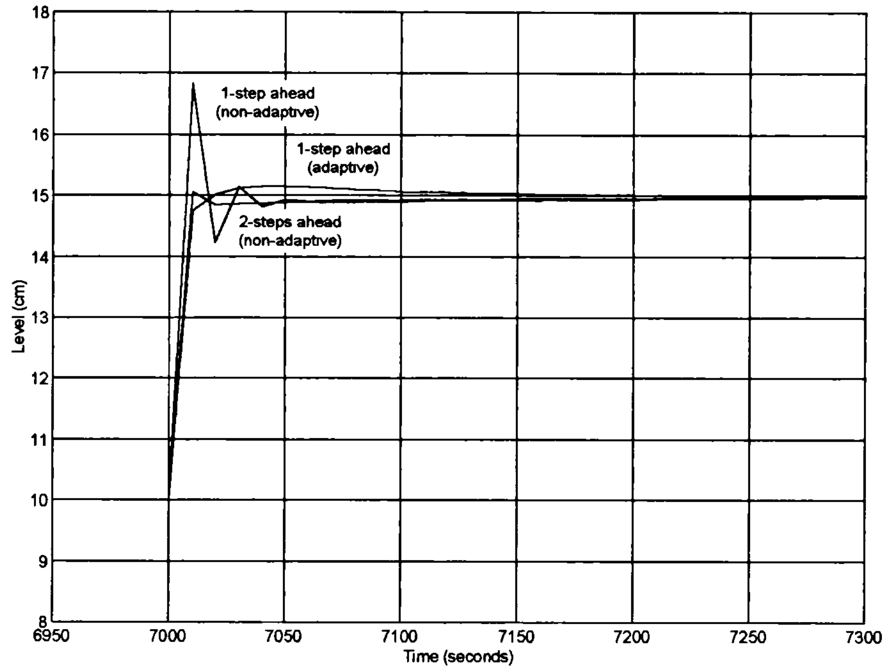


Figure A.15: Comparison of process output responses to setpoint change between 10 and 15 cm. when using proposed controller with 2-partition fuzzy model. ($K_{f_1} = 1; K_{f_2} = 0.05$; $\lambda = 0.98; A = 9$ cm^2)

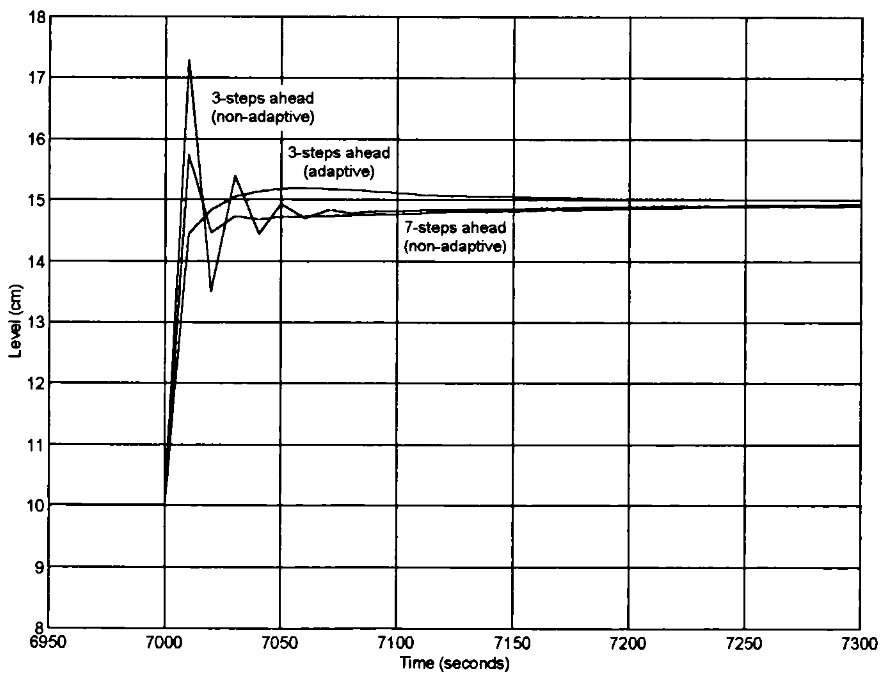


Figure A.16: Comparison of process output responses to setpoint change between 10 and 15 cm. when using proposed controller with 2-partition fuzzy model. ($K_{f_1} = 1; K_{f_2} = 0.05$; $\lambda = 0.98; A = 5$ cm^2)

A.5 Application to Control of CSTR

Figures A.17 to A.24 show the adaptive controller's performance when using different process models under process conditions similar to that used during off-line model identification. Model adaptation was initiated 27.5 hours after the start of a simulation run to enable comparative assessment of the controller with and without adaptive modelling capability. Calculation of the *P* matrix was initiated at the start of a simulation run even though the switch-over from non-adaptive to adaptive control occured much later.

As in the case of the liquid level simulation study, adaptation allows fine-tuning of model parameters to local operating conditions. An improvement in performance is observed with all process models in the case of 1-step ahead predictive controller (Figures A.17 to A.20). The improvement is most significant in the case of the linear process model (Figure A.17) and least significant in the case of the 5-partition fuzzy process model (Figure A.20).

An improvement is also noticed in the case of the linear process model (Figure A.21) and the 2-partition fuzzy process model (Figure A.22) when the prediction horizon is increased to 2-steps. Close examination of Figures A.23 and A.24 will reveal that adaptation can introduce sluggishness in the output response increasing both the initial rise time as well as the time needed for offset correction at the new steady-state concentration resulting in a more pronounced overshoot. This leads to an overall deterioration in performance in the case of 3 and 5-partition fuzzy models when the prediction horizon is more than 1-step.

It will be observed from Figures A.17 to A.24 that the performance of controllers after adaptation is completed is generally better with fuzzy process models as compared to the linear process model. Also, the performance of predictive controllers using fuzzy process models is almost identical after adaptation is completed.

The use of more than 2 fuzzy partitions was found to affect the robustness of the controller similarly as in the case of the liquid level simulation study.

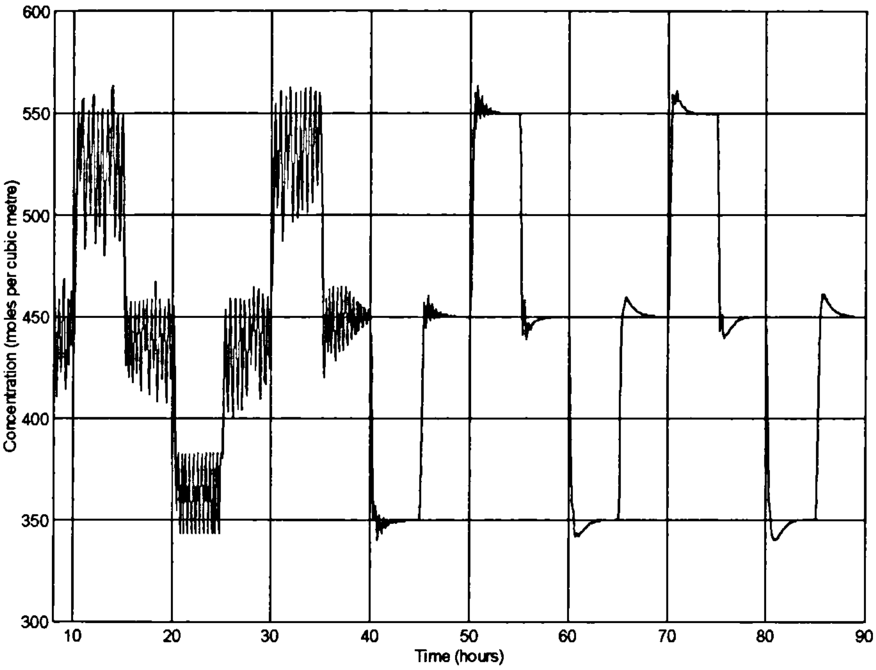


Figure A.17: Process output response to setpoint changes when using 1-step ahead predictive controller with linear process model.
($K_{f_1} = 1; K_{f_2} = 0.1$; $\lambda = 0.987$; $V = 1.36\ m^3$)

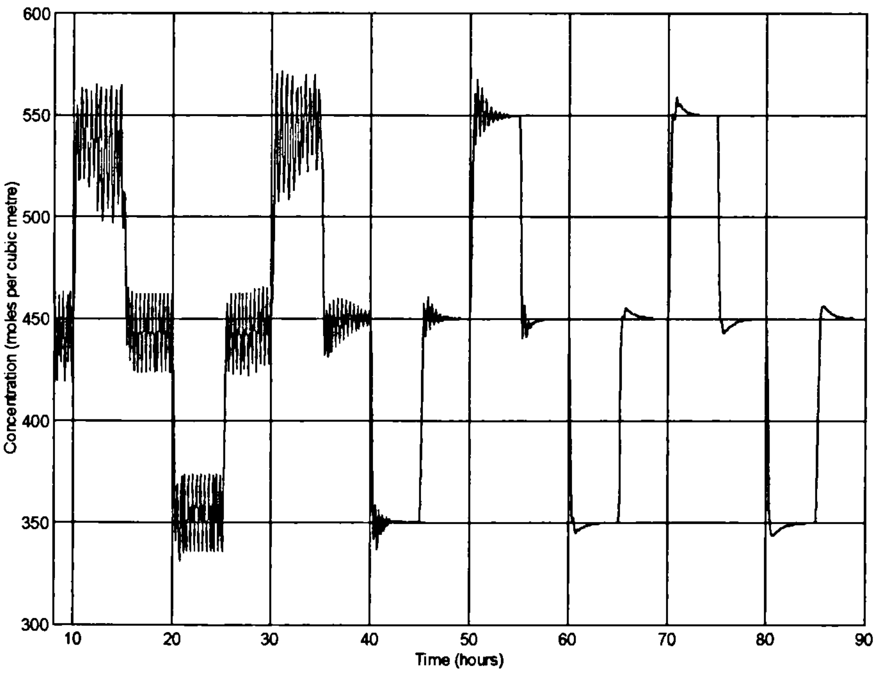


Figure A.18: Process output response to setpoint changes when using 1-step ahead predictive controller with 2-partition fuzzy model.
($K_{f_1} = 1; K_{f_2} = 0.1$; $\lambda = 0.987$; $V = 1.36\ m^3$)

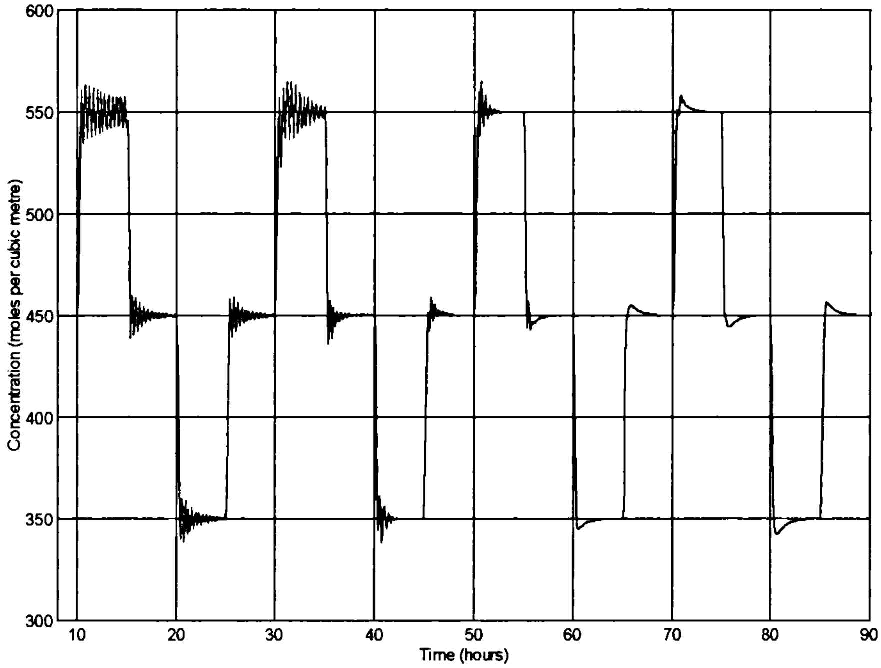


Figure A.19: Process output response to setpoint changes when using 1-step ahead predictive controller with 3-partition fuzzy model. ($K_{f_1} = 1; K_{f_2} = 0.1;\ \lambda = 0.987;\ V = 1.36\ m^3$)

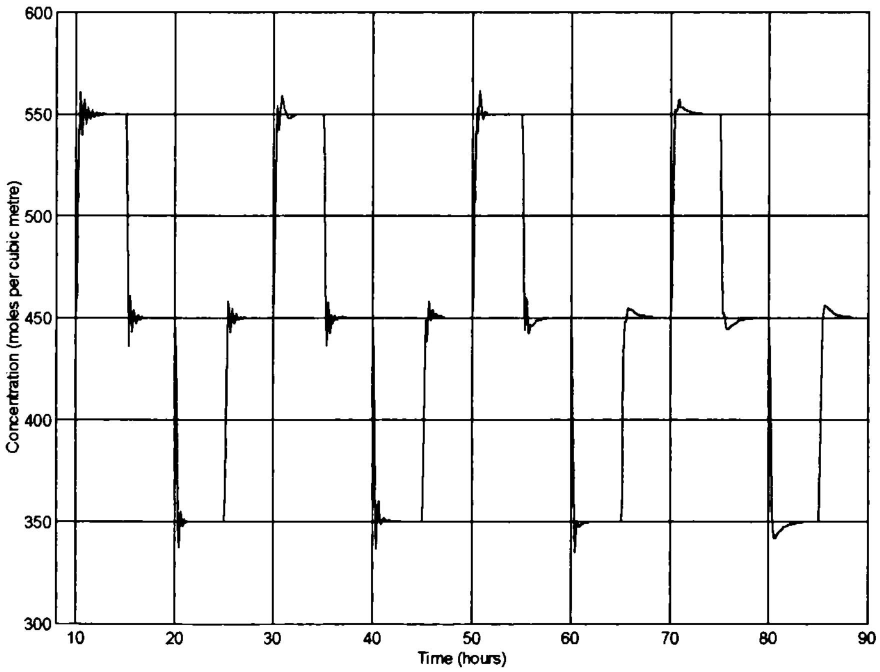


Figure A.20: Process output response to setpoint changes when using 1-step ahead predictive controller with 5-partition fuzzy model. ($K_{f_1} = 1; K_{f_2} = 0.1;\ \lambda = 0.987;\ V = 1.36\ m^3$)

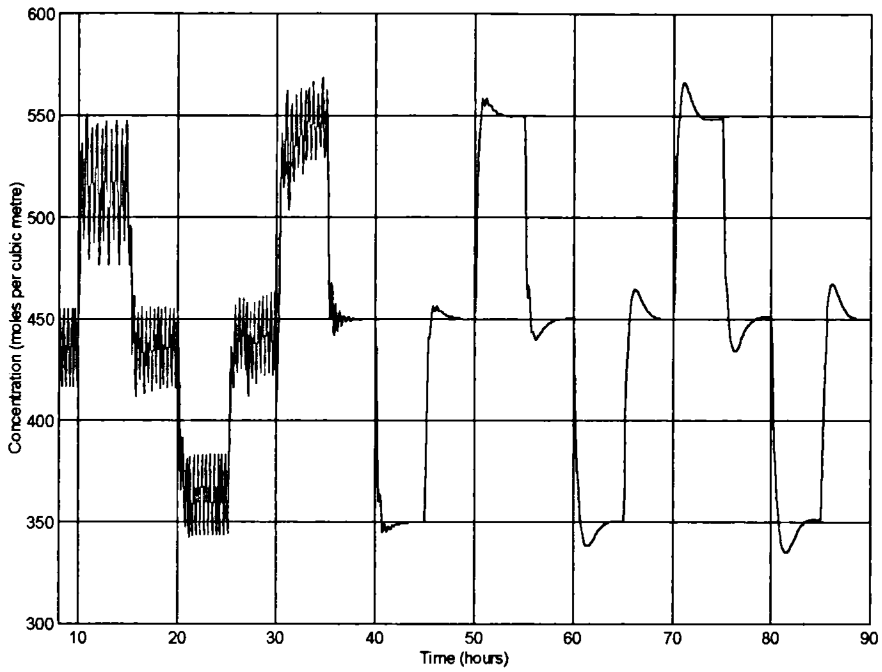


Figure A.21: Process output response to setpoint changes when using 2-steps ahead predictive controller with linear process model.
($K_{f_1} = 1; K_{f_2} = 0.1$; $\lambda = 0.987$; $V = 1.36$ m^3)

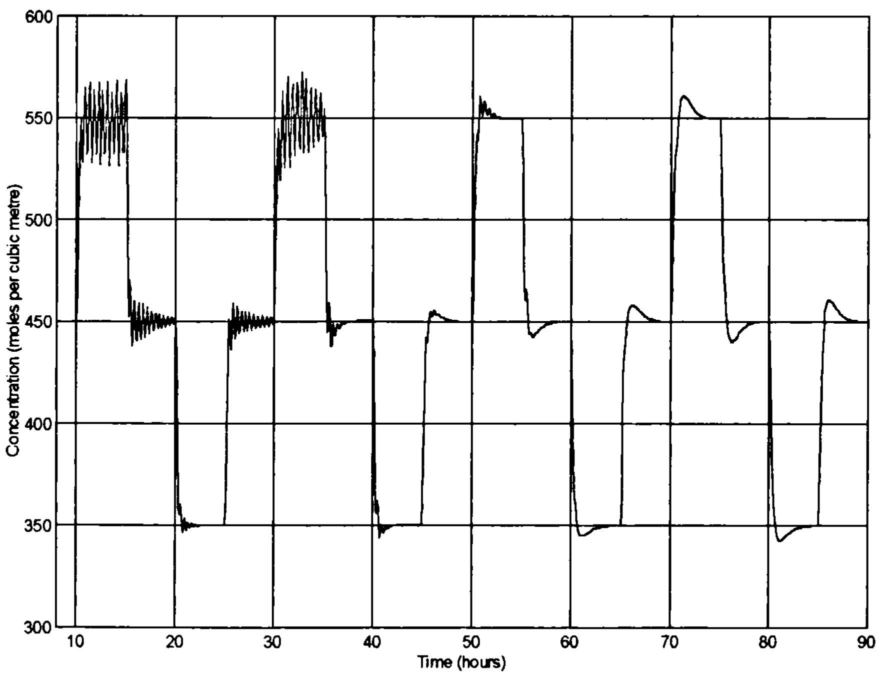


Figure A.22: Process output response to setpoint changes when using 2-steps ahead predictive controller with 2-partition fuzzy model.
($K_{f_1} = 1; K_{f_2} = 0.1$; $\lambda = 0.987$; $V = 1.36$ m^3)

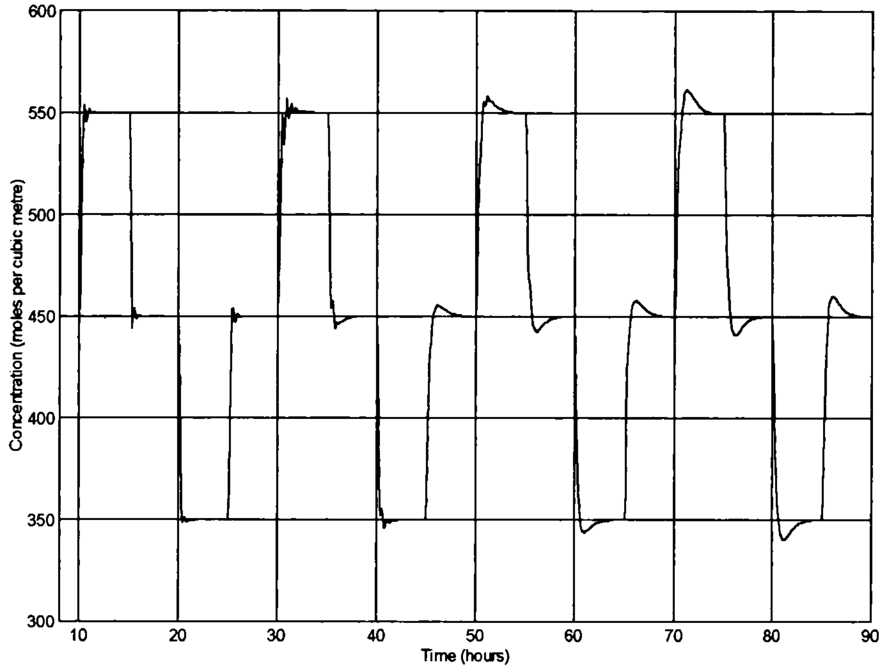


Figure A.23: Process output response to setpoint changes when using 2-steps ahead predictive controller with 3-partition fuzzy model.
($K_{f_1} = 1; K_{f_2} = 0.1$; $\lambda = 0.987$; $V = 1.36$ m³)

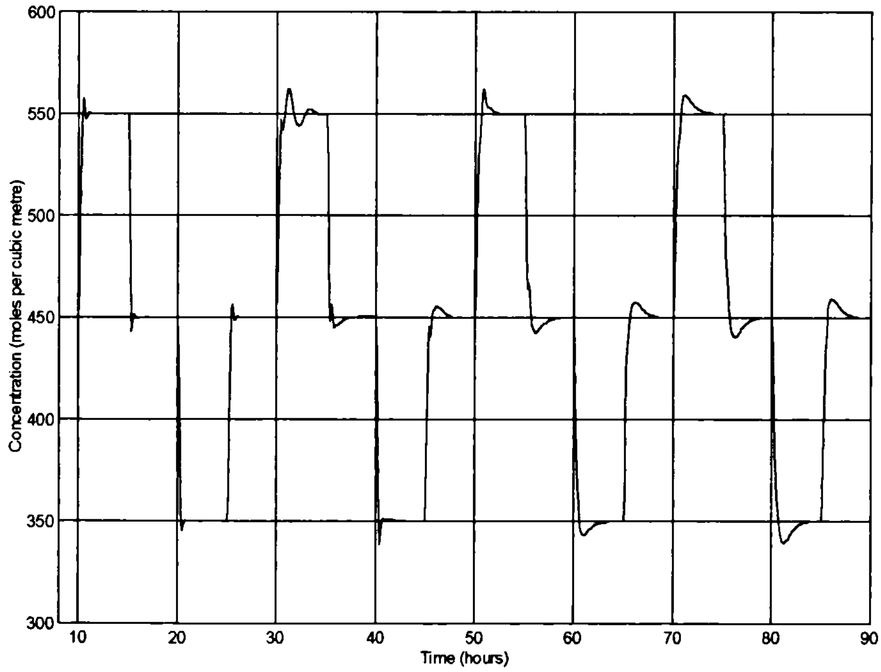


Figure A.24: Process output response to setpoint changes when using 2-steps ahead predictive controller with 5-partition fuzzy model.
($K_{f_1} = 1; K_{f_2} = 0.1$; $\lambda = 0.987$; $V = 1.36$ m³)

A.6 Conclusions

The two simulation examples show that adaptive control using the Takagi-Sugeno fuzzy modelling approach based on recursive estimation of model parameters using the least-squares algorithm to be viable. It was found that there was little or no difference in the performance of controllers after adaptation is completed when using different fuzzy process models. Both studies, however, showed an improvement in performance over linear process models, especially when the prediction horizon is increased.

The use of more fuzzy partitions leads to better performance in non-adaptive control systems but even here the improvement tends to become less significant when too many fuzzy partitions are used. It has also been shown in this study that comparable or better performance can be achieved without model adaptation when process conditions are similar to or only slightly different from that used during identification of the off-line model. Hence the decision as to whether to use an adaptive controller or a non-adaptive controller with possibly more fuzzy partitions will depend on the anticipated process changes. If we are uncertain we should start with a non-adaptive controller and switch over to an adaptive controller if the performance is found to be poor.

Robustness issues have not been sufficiently addressed in this study. It was observed that adaptation repeatedly carried out close to the fuzzy partitioning points caused wide fluctuations and drift in the parameter estimates. Such problems can be minimised by using variable step sizes during adaptation or by switching off the adaptation in the vicinity of the fuzzy partitioning points. Using techniques such as square root decomposition or UD factorization may also help to improve the robustness. It is also advisable to monitor the modelling error and trigger the model adaptation process only if the error falls outside a certain threshold. The use of too many fuzzy partitions will make it much more difficult to manage model adaptation over a wide range to ensure that the model correctly reflects process changes. The stability of a system where local models concurrently provide the very different process behaviour before and after a significant process change may be questionable. To avoid such problems, it is suggested that adaptive control using fuzzy process models should be limited to 2 fuzzy partitions.

Appendix B

CONTROL OF SYSTEMS WITH DEAD TIME

B.1 Introduction

It is known that dead time makes processes difficult to control. Designing controllers to overcome time delays has always been a serious challenge to control engineers. The widely used PID controllers can be very ineffective in overcoming this problem.

In this section, we will consider a method for enabling dead time compensation in the fuzzy model-based predictive controller proposed in Chapter 7. It is assumed in the design of the compensation scheme that the dead time can be approximated in terms of number of sampling intervals. The study will examine the effect of dead time on the performance of our proposed fuzzy model-based predictive control algorithm, with and without the dead time compensation scheme.

B.2 Model Predictions in the Presence of Dead Time

B.2.1 First-Order System

A first-order system with dead time can be described by the following ARX model structure:

$$y_p(t+1) = a_1 y(t) + b_1 u(t-d) + k + err(t) \tag{B.1}$$

where:

$y(t)$	=	process output at time t.
$y_p(t+1)$	=	one-step ahead model prediction at time t.
$u(t)$	=	manipulated variable at time t.
d	=	dead time expressed in number of sampling intervals.
$err(t)$	=	estimate of the modelling error at time t.

Model predictions over a prediction horizon of n-steps based on a control horizon of 1-step is given by the equation:

$$Y(t) = \mathbf{P}X(t) + \mathbf{Q}U(t) + \mathbf{R}[k + err(t)] \tag{B.2}$$

where,

$$Y(t) = \begin{bmatrix} y_p(t+1) & \cdots & y_p(t+n) \end{bmatrix}^T \tag{B.3}$$

$$X(t) = \begin{bmatrix} y(t) & u(t-d) & \cdots & u(t-1) \end{bmatrix}^T \tag{B.4}$$

$$U(t) = \begin{bmatrix} u(t) \end{bmatrix} \tag{B.5}$$

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & 0 & 0 & \cdots & 0 \\ p_{21} & p_{22} & p_{23} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{(d-1)1} & p_{(d-1)2} & p_{(d-1)3} & p_{(d-1)4} & \cdots & 0 \\ p_{d1} & p_{d2} & p_{d3} & p_{d4} & \cdots & p_{d(d+1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{n1} & p_{n2} & p_{n3} & p_{n4} & \cdots & p_{n(d+1)} \end{bmatrix} \quad (B.6)$$

$$\mathbf{Q} = \begin{bmatrix} 0 & \cdots & 0 & q_{d+1} & \cdots & q_n \end{bmatrix}^T \quad (B.7)$$

$$\mathbf{R} = \begin{bmatrix} r_1 & \cdots & r_n \end{bmatrix}^T \quad (B.8)$$

The coefficients of **P**, **Q** and **R** are given by:

$$\begin{aligned} &p_{11} = a_1 \\ &p_{12} = p_{23} = p_{34} = \ldots = p_{d(d+1)} = b_1 \\ &p_{ij} = a_1 \cdot p_{(i-1)j}, \text{ for all other non-zero entries} \end{aligned} \quad (B.9)$$

$$\begin{aligned} &q_{d+1} = b_1 \\ &q_i = a_1 \cdot q_{i-1} + b_1, \text{ for } i = d+2, \ldots, n \end{aligned} \quad (B.10)$$

$$\begin{aligned} &r_1 = 1 \\ &r_i = a_1 \cdot r_{i-1} + 1, \text{ for } i = 2, \ldots, n \end{aligned} \quad (B.11)$$

B.2.2 Second-Order System

A second-order system with a dead time of d sampling intervals can be described by the following ARX model structure:

$$y_p(t+1) = a_1 y(t) + a_2 y(t-1) + b_1 u(t-d) + b_2 u(t-(d+1)) + k + err(t) \quad (B.12)$$

Model predictions over a prediction horizon of n-steps based on a control horizon of 1-step is given by the equation:

$$Y(t) = \mathbf{P}X(t) + \mathbf{Q}U(t) + \mathbf{R}[k + err(t)] \quad (B.13)$$

where,

$$Y(t) = \left[y_p(t+1) \quad \cdots \quad y_p(t+n)\right]^T \tag{B.14}$$

$$X(t) = \left[y(t) \quad y(t-1) \quad u(t-(d+1)) \quad u(t-d) \quad \cdots \quad u(t-1)\right]^T \tag{B.15}$$

$$U(t) = \left[u(t)\right] \tag{B.16}$$

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & 0 & 0 & \cdots & 0 \\ p_{21} & p_{22} & p_{23} & p_{24} & p_{25} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{(d-1)1} & p_{(d-1)2} & p_{(d-1)3} & p_{(d-1)4} & p_{(d-1)5} & p_{(d-1)6} & \cdots & 0 \\ p_{d1} & p_{d2} & p_{d3} & p_{d4} & p_{d5} & p_{d6} & \cdots & p_{d(d+3)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{n1} & p_{n2} & p_{n3} & p_{n4} & p_{n5} & p_{n6} & \cdots & p_{n(d+3)} \end{bmatrix} \tag{B.17}$$

$$\mathbf{Q} = \left[0 \quad \cdots \quad 0 \quad q_{d+1} \quad \cdots \quad q_n\right]^T \tag{B.18}$$

$$\mathbf{R} = \left[r_1 \quad \cdots \quad r_n\right]^T \tag{B.19}$$

The coefficients of **P**, **Q** and **R** are given by:

$$\begin{aligned} p_{11} &= a_1 \\ p_{12} &= a_2 \\ p_{13} &= b_2 \\ p_{14} &= p_{25} = p_{36} = \ldots = p_{d(d+3)} = b_1 \end{aligned} \tag{B.20}$$

$$\begin{aligned} p_{21} &= a_1 \cdot p_{11} + a_2 \\ p_{22} &= a_1 \cdot p_{12} \\ p_{23} &= a_1 \cdot p_{13} \\ p_{24} &= p_{35} = p_{46} = \ldots = p_{(d+1)(d+3)} = a_1 \cdot b_1 + b_2 \end{aligned} \tag{B.21}$$

$$p_{ij} = a_1 \cdot p_{(i-1)j} + a_2 \cdot p_{(i-2)j}\text{, for all other non-zero entries} \tag{B.22}$$

$$\begin{aligned} q_{d+1} &= b_1 \\ q_{d+2} &= a_1 \cdot q_{d+1} + b_1 + b_2 \end{aligned} \tag{B.23}$$

$$q_i = a_1 \cdot q_{i-1} + a_2 \cdot q_{i-2} + b_1 + b_2 \text{, for } i = d+3,\ldots,n$$

$$\begin{aligned} r_1 &= 1 \\ r_2 &= a_1 + 1 \\ r_i &= a_1 \cdot r_{i-1} + a_2 \cdot r_{i-2} + 1 \text{, for } i = 3,\ldots,n \end{aligned} \tag{B.24}$$

The following observations can be made from the above two systems:

- It is possible to derive **P** for systems with no dead time using the formulae for systems with dead time. For a first-order system, **P** reduces to a one-column matrix given by the first column; and for a second-order system, **P** reduces to a 3-column matrix given by the first 3 columns.

- $X(t)$ contains more terms as compared to the system with no dead time. The increase in the number of terms in $X(t)$ is equal to the number of sampling intervals in the dead time. This increase is also reflected by an increase in the number of columns in **P**.

- The coefficients in these new **P** columns follow a simple pattern which facilitates their determination. For example, the coefficients in the second column can be determined by shifting the coefficients in the first column one row down. This is equivalent to applying the following general rule to determine the coefficients of all the columns except the first column:

$$\begin{aligned} p_{1j} &= 0 \\ p_{ij} &= p_{(i-1)(j-1)} \end{aligned} \tag{B.25}$$

- The coefficients in the first d rows of **Q** is '0'.

- **R** remains the same as for the system with no dead time.

B.3 Controller Formulation

Since the earliest values of the process output that can be influenced by control action taken at time t occurs at time $(t+d+1)$, the cost function used to determine the optimal controller output will be defined as follows:

$$\begin{aligned} J &= \sum_{i=d+1}^{n}[y_p(t+i)-w(t+i)]^2 \\ &= [Y'(t)-W'(t)]^T[Y'(t)-W'(t)] \\ &= [\mathbf{P}'X(t)+\mathbf{Q}'U(t)+\mathbf{R}'(k+err(t))-W'(t)]^T[\mathbf{P}'X(t)+\mathbf{Q}'U(t)+\mathbf{R}'(k+err(t))-W'(t)] \end{aligned} \tag{B.26}$$

where,

$$Y'(t) = [y_p(t+d+1) \quad \cdots \quad y_p(t+n)]^T \tag{B.27}$$

$$W'(t) = [w(t+d+1) \quad \cdots \quad w(t+n)]^T \tag{B.28}$$

$\mathbf{P}'$, $\mathbf{Q}'$ and $\mathbf{R}'$ are derived from **P**, **Q** and **R** by deleting the first d rows and $W'(t)$ is the desired trajectory of setpoints. It is important that the prediction horizon should be at least 1-step more than the dead time. This will ensure that $\mathbf{P}'$, $\mathbf{Q}'$ and $\mathbf{R}'$ will have a minimum of 1 row. It will also be noticed that the coefficients in $\mathbf{Q}'$ start in the same way as in **Q** for the system with no time delay.

The optimal controller output sequence can be found by minimising the above cost function. A necessary condition for minimum J is:

$$\frac{\partial J}{\partial U} = 0 \tag{B.29}$$

Differentiating the expression for J and using the above condition leads to the following optimal solution:

$$U(t) = [\mathbf{Q}'^T\mathbf{Q}']^{-1}\mathbf{Q}'^T\{W'(t)-\mathbf{P}'X(t)-\mathbf{R}'[k+err(t)]\} \tag{B.30}$$

The performance of the above dead time compensation scheme will now be examined using two simulation examples.

B.4 Application to Control of Liquid Level

A first-order model structure has been assumed as in previous studies and the value of the constant term, k_i, has been assumed to be non-zero. The 5-partition fuzzy model has been used in all studies because of the better modelling accuracy. Sampling was carried out at 10-second intervals.

Figures B.1, B.2 and B.4 show the performance of the controller without the dead time compensation scheme at the lower level when there is no dead time, a dead time of 1-sampling interval and a dead time of 2-sampling intervals, respectively. The plots show the output from the simulation of the liquid level system. The effect of dead time will lead to a shift in the process output by an amount equal to the dead time. It will be observed that dead time can lead to significant performance deterioration of the control system, especially if the prediction horizon is small. It will also be observed that the performance deterioration becomes worse as the amount of dead time increases. One possible solution to this problem is to use prediction horizons of more than 10-steps. This, however, introduces considerable sluggishness in the output response which may be undesirable.

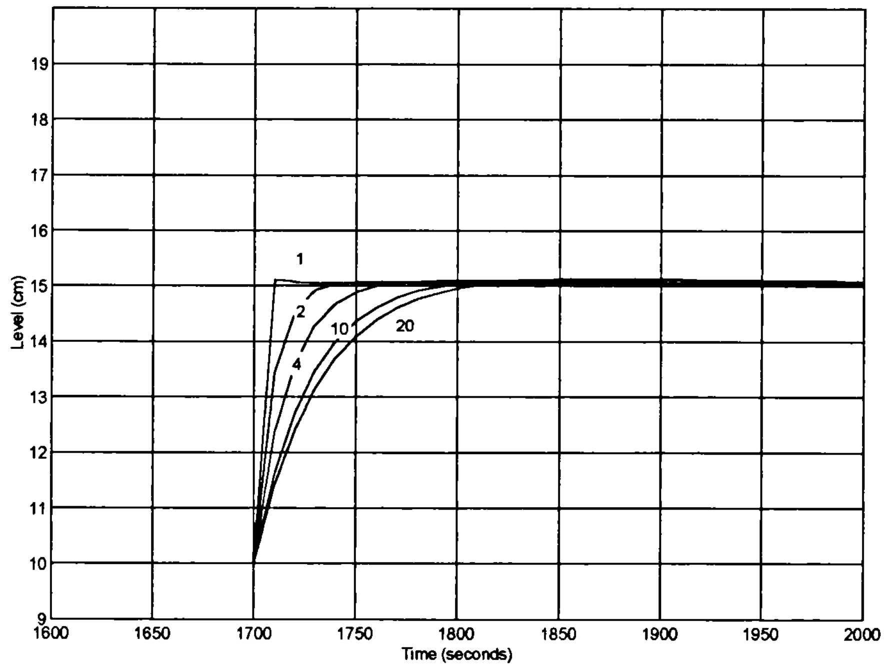


Figure B.1: Effect of number of steps in prediction horizon on process output response to setpoint changes between 10 and 15 cm. when using proposed controller with 5-partition fuzzy model ($K_{f_2} = 0.05; d = 0$).

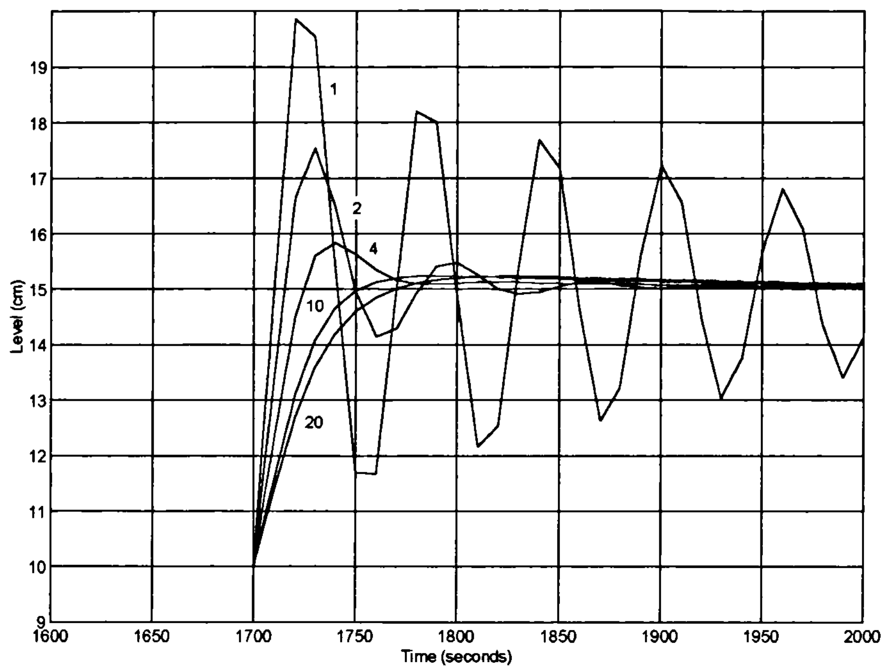


Figure B.2: Effect of number of steps in prediction horizon on process output response to setpoint changes between 10 and 15 cm. when using proposed controller without dead time compensation $(K_{f_2} = 0.05; d = 1)$.

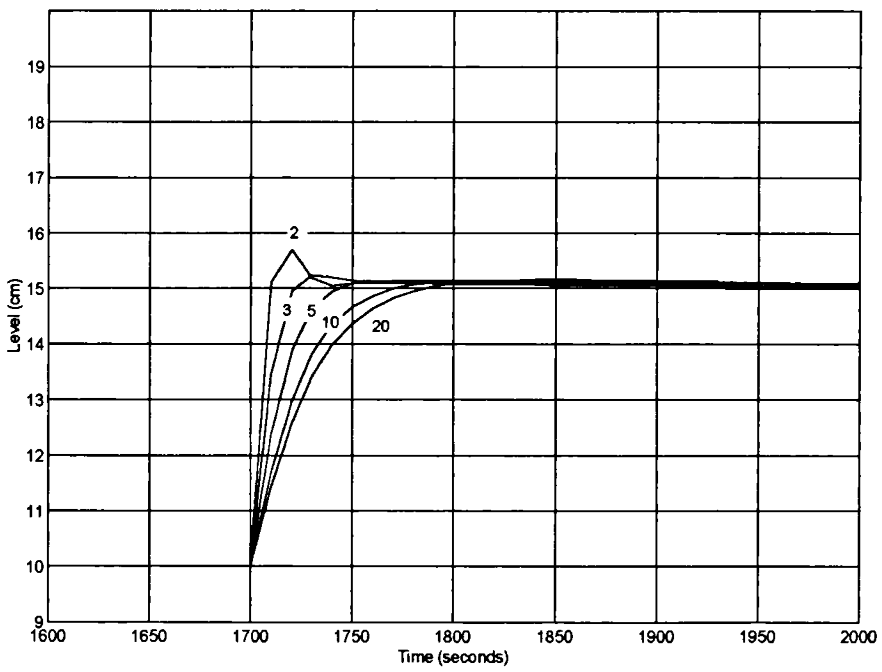


Figure B.3: Effect of number of steps in prediction horizon on process output response to setpoint changes between 10 and 15 cm. when using proposed controller with dead time compensation $(K_{f_2} = 0.05; d = 1)$.

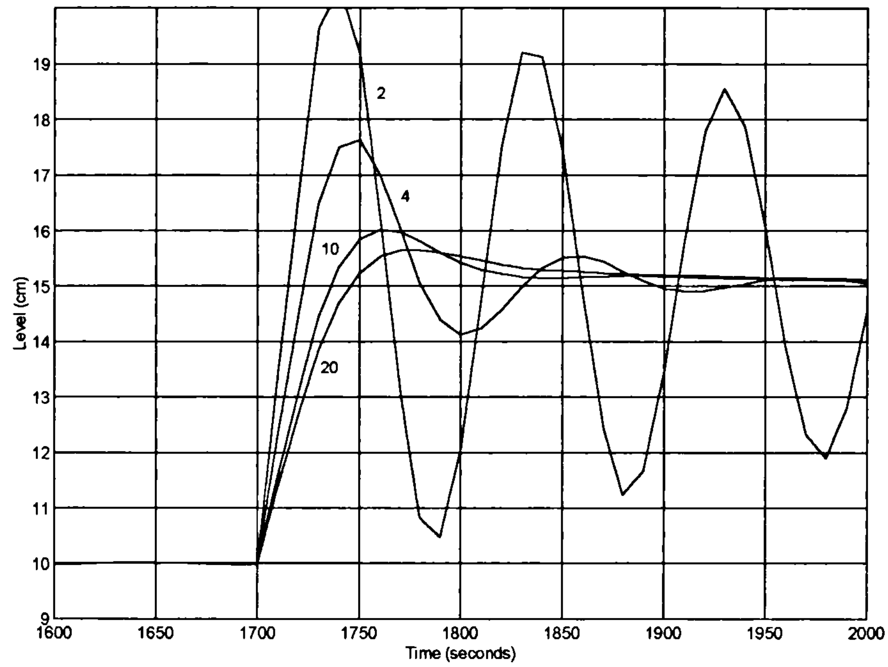


Figure B.4: Effect of number of steps in prediction horizon on process output response to setpoint changes between 10 and 15 cm. when using proposed controller without dead time compensation $(K_{f_2} = 0.05; d = 2)$.

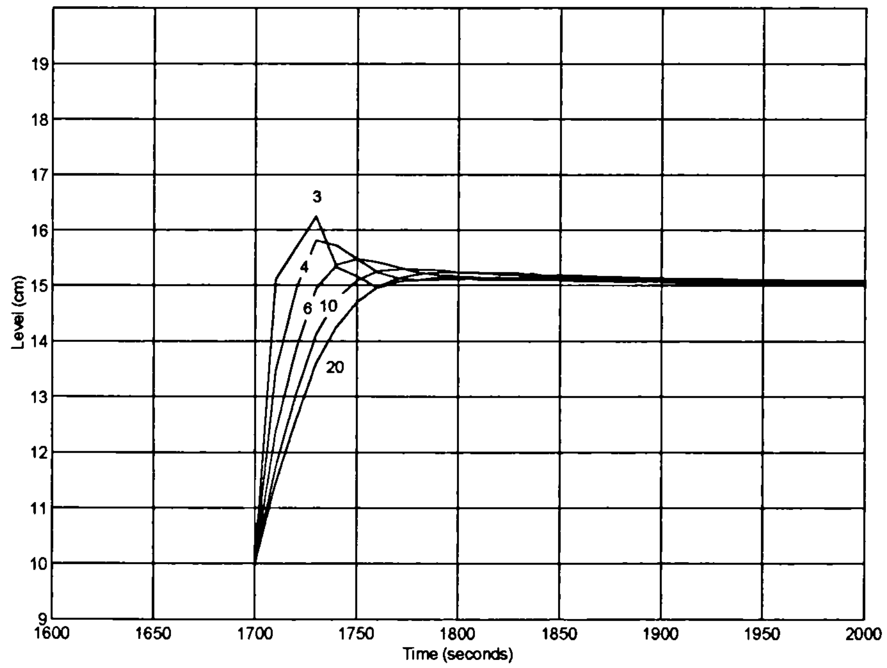


Figure B.5: Effect of number of steps in prediction horizon on process output response to setpoint changes between 10 and 15 cm. when using proposed controller with dead time compensation $(K_{f_2} = 0.05; d = 2)$.

The advantage of the dead time compensation scheme is apparent from Figures B.3 and B.5. As in the case of the controller without the dead time compensation scheme, the plots show the output from the simulation of the liquid level system. Significant improvement in performance when using small prediction horizons can be noticed. Comparison of Figures B.1, B.3 and B.5 show that the presence of dead time generally leads to performance deterioration, even with dead time compensation. The extent of the deterioration gets worse as the amount of dead time increases.

B.5 Application to Control of CSTR

A second-order model structure has been assumed and the value of the constant term, k_i, has been assumed to be zero. The 5-partition fuzzy model has once again been used because of the better modelling accuracy and sampling was carried out 0.1 hour intervals.

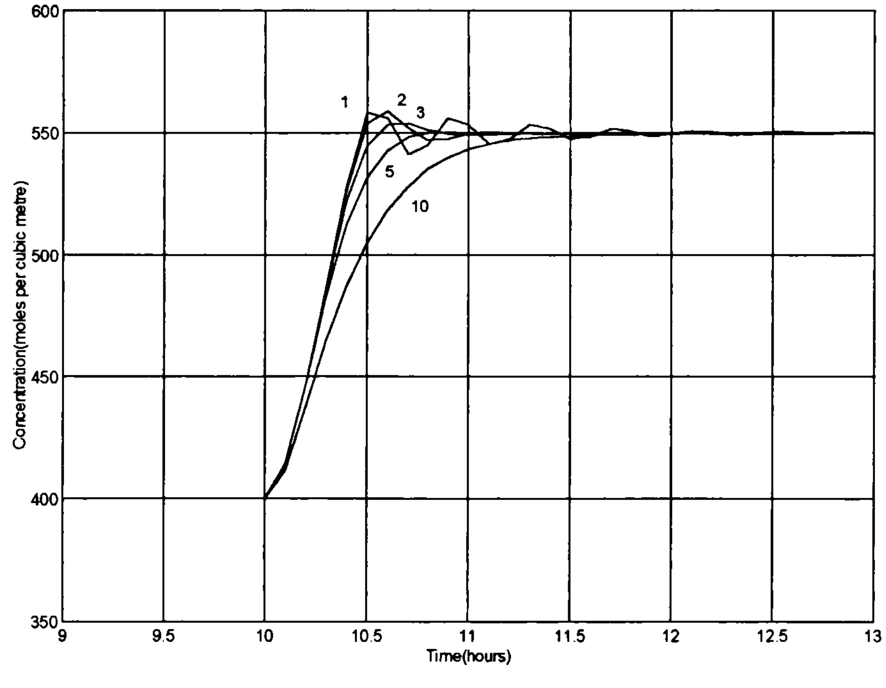


Figure B.6: Effect of number of steps in prediction horizon on process output response to +150 moles $\cdot$ m^{-3} setpoint change when using proposed controller with 5-partition fuzzy model ($K_{f_1} = 1; K_{f_2} = 0.1; d = 0$).

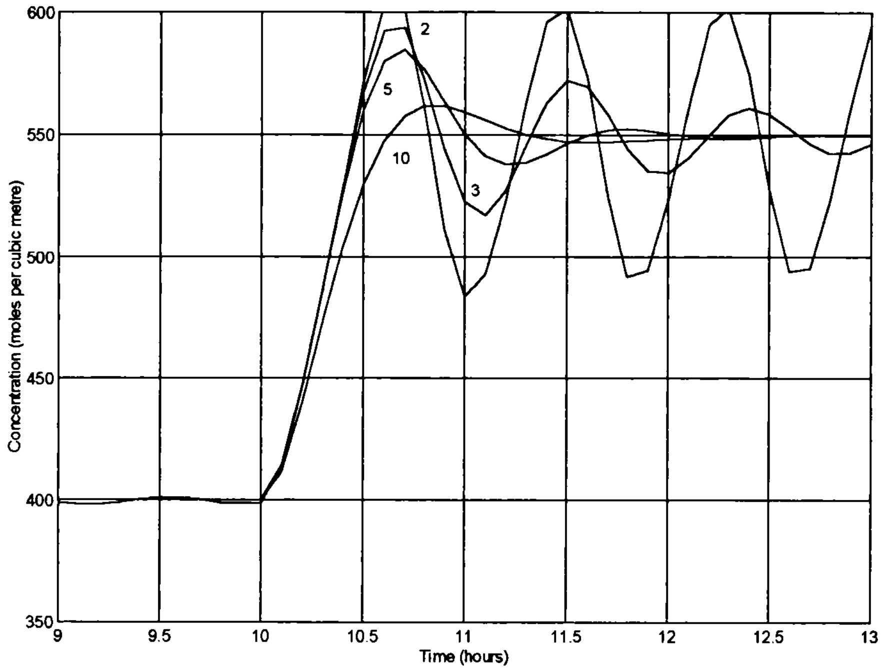


Figure B.7: Effect of number of steps in prediction horizon on process output response to +150 moles·m^{-3} setpoint change when using proposed controller without dead time compensation ($K_{f_1} = 1; K_{f_2} = 0.1; d = 1$).

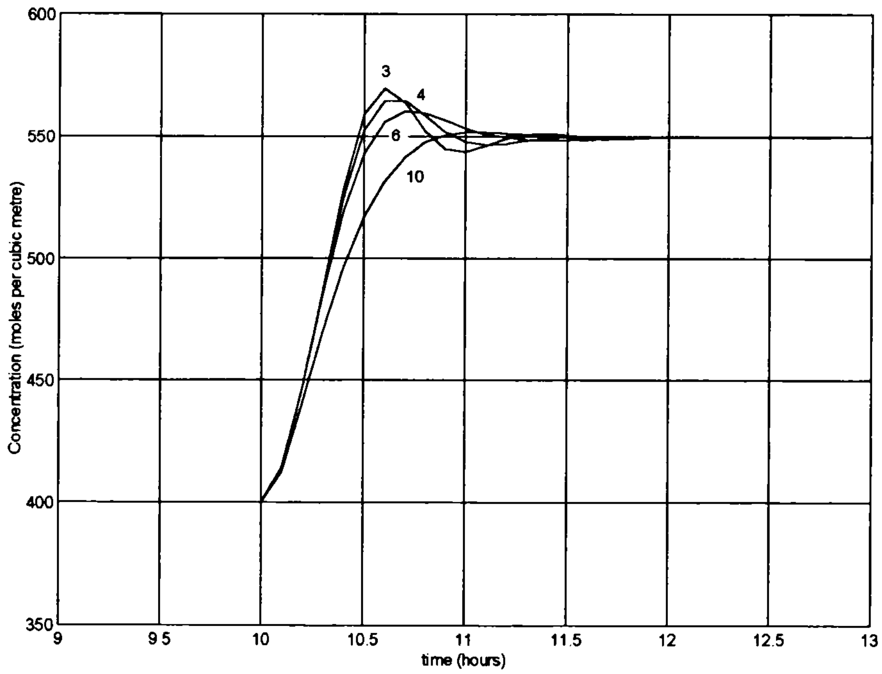


Figure B.8: Effect of number of steps in prediction horizon on process output response to +150 moles·m^{-3} setpoint change when using proposed controller with dead time compensation ($K_{f_1} = 1; K_{f_2} = 0.1; d = 1$).

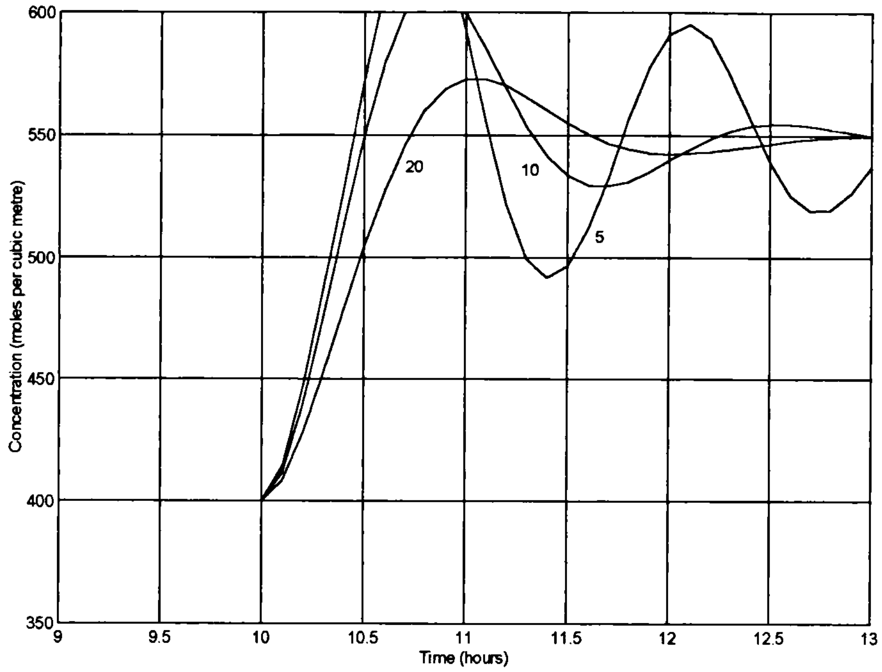


Figure B.9: Effect of number of steps in prediction horizon on process output response to +150 moles$\cdot m^{-3}$ setpoint change when using proposed controller without dead time compensation ($K_{f_1} = 1; K_{f_2} = 0.1; d = 2$).

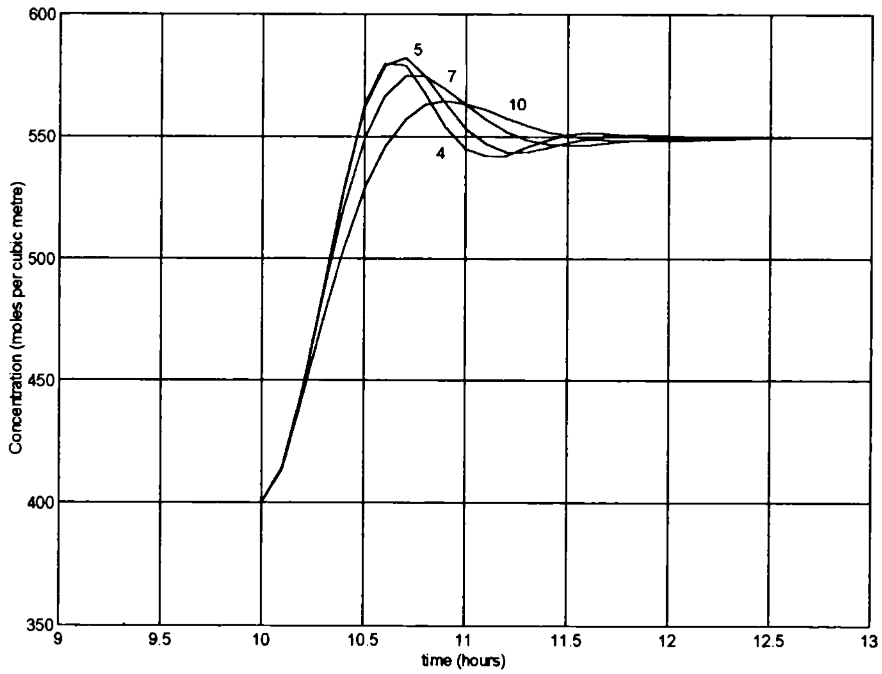


Figure B.10: Effect of number of steps in prediction horizon on process output response to +150 moles$\cdot m^{-3}$ setpoint change when using proposed controller with dead time compensation ($K_{f_1} = 1; K_{f_2} = 0.1; d = 2$).

Figures B.6, B.7 and B.9 show the performance of the controller without the dead time compensation scheme to +150 moles.m^{-3} setpoint change when there is no dead time, a dead time of 1-sampling interval and a dead time of 2-sampling intervals, respectively. As in the case of the tank liquid level simulation studies, the plots show the output from the simulation of the CSTR system and does not include the effect of dead time. The observations here are quite similar to the observations for the liquid level system. One possible solution to the problem of dead time is to use prediction horizons of about 20-steps. This, however, introduces considerable sluggishness in the output response which may be undesirable.

The advantage of the dead time compensation scheme is apparent by comparing Figures B.8 and B.10 with Figures B.7 and B.9, respectively. Once again, the plots show the output from the simulation of the CSTR system. Significant improvement in performance when using small prediction horizons can be noticed. Comparison of Figures B.6, B.8 and B.10 show that the presence of dead time generally leads to performance deterioration, even with dead time compensation. As with the liquid level system, the extent of the deterioration gets worse as the amount of dead time increases.

B.6 Conclusions

Both application examples have shown that dead time can lead to performance deterioration, with the extent of deterioration becoming worse as the amount of dead time increases. One possible solution to the problem is to increase the number of steps in the prediction horizon. This, however, can introduce considerable sluggishness in the output response. Besides, no real advantage will be gained by using a more accurate model, since the main advantage of using a more accurate model is the better performance achieved when the prediction horizon is small.

A dead time compensation scheme has been demonstrated using first-order and second-order systems with relatively small dead times of up to 2-sampling intervals. In each case, it was found that the most significant performance improvement occurs when the prediction horizon is small. Further, it was found that even with the dead time compensation scheme, the performance of the system was still worse than a system with no dead time.

A dead time of 2-sampling intervals corresponds to approximately 20 percent of the time constant of the processes investigated. Such dead times can arise from transportation lags such as the flow of liquids through pipes. In many practical control problems, however, the dead time-to-time constant ratio can be much greater than 20 percent.