INTERNAL MODEL CONTROL DESIGN USING JUST-IN-TIME LEARNING TECHNIQUE

KALMUKALE ANKUSH GANESHREDDY

NATIONAL UNIVERSITY OF SINGAPORE

2006

INTERNAL MODEL CONTROL DESIGN USING JUST-IN-TIME LEARNING TECHNIQUE

KALMUKALE ANKUSH GANESHREDDY
(B.E., NITK, Surathkal, India)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF CHEMICAL AND BIOMOLECULAR ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2006

ACKNOWLEDGEMENTS

First, I would like to thank my supervisor Dr. Min-Sen Chiu for his constant support, encouragement, motivation, invaluable guidance and suggestions throughout my research work at National University of Singapore, Singapore. He was always there to listen and to give advice. He showed me different ways to approach a research problem and the need to be persistent to accomplish any goal. My special thanks to Dr. Chiu for his promptness and sparing his invaluable time to read this manuscript. Further, I extend my sincere and deepest gratitude for his kindness, forgiveness, concern and moral support shown throughout my stay here in Singapore.

Besides my supervisor, I would like to thank Prof. Zhao George, Prof. A. K. Ray, Prof. Rangaiah, Prof. Karimi and Prof. Matsuura for teaching me the fundamentals of colloids, mathematical methods, optimization and thermodynamics. My special thanks to Dr. Laksh for his moral support and concern shown throughout my research work at NUS. I would also wish to thank technical and administrative staffs in the Chemical & Biomolecular Engineering Department who have contributed, directly or indirectly, to this thesis. I am also indebted to the National University of Singapore for providing me the excellent research facilities and research scholarship.

Special thanks to my labmates Cheng Cheng, Dr. Jia Li, Ye and Yasuki for actively participating discussion related to my research work and the help that they have rendered to me. I will always relish the warmth and affection that I received from my present and past colleagues Sreenivasareddy, Biswajit, Sathishkumar,

Mranal, Dharmesh, Rampa, Marathe, Bhutani, Avinash, Naveen, Murthy, Srinivas, Sudhakar, Manish, Pavan, Atreyee, Mukta, Xu Bu, Martin, Raghuraj, Balaji, Ganesh, Suresh and Arul. Special words of gratitude to Sreenivasareddy for providing support throughout my research work.

I equally cherish the moments that I spent with Ugandhar, Sreenivasareddy, Sathishkumar, Biswajit, Krishna, Shukla, Abhishek, Pradip, Arindam, Asif, Sateesh, Vempati, Varsha, Prateek and Anurodh. I am immensely thankful to all of them in making me feel at home in Singapore. My wonderful friends other than the mentioned above, to list whose names would be endless, have been a great source of solace for me in times of need besides the enjoyment they had given me in their company.

Last, but not least, I thank my parents and family members for their unconditional support, affectionate love and encouragement, without which this work would not have been possible. I also wish to thank my fiancée Padmaja for her understanding, continuous support and encouragement during the final days of my project work. Also I would like to thank my school and college friends whose moral support helped me cruise through some of tough times. In particular, I am greatly indebted to Vijay Chandrashekharan for getting me interested in coming to Singapore.

TABLE OF CONTENTS

SUMMARY

In this study, two novel IMC design methods using JITL technique, which are capable of controlling dynamic systems that operate over a wide range of operating regimes, are presented. In the first approach, a nonlinear IMC design based on partitioned model inverse is proposed for a class of nonlinear SISO and MIMO systems. Partitioned model consists of a linear model, which is obtained around an operating point, and a nonlinear model, which is identified by JITL algorithm. It is also shown that JITL model in the proposed control strategy can be made adaptive on-line readily by simply adding the new process data to the database. Simulation results confirm that the resultant IMC design is indeed superior to the conventional IMC scheme.

In other approach, a memory-based IMC design approach is proposed for nonlinear systems. The proposed method employs JITL not only to update model parameters but also to adjust the parameters of IMC controller. At each sampling instant, the initial IMC filter parameter is obtained using a controller database. In addition, parameter updating algorithm is developed by employing the steepest descent gradient rule and is used to adjust the initial filter parameter on-line. Simulation results confirm that the performance of proposed memory-based IMC scheme shows a marked improvement over that achieved by the conventional PI/PID controller.

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

A, A_w	Heat exchange area
$\mathbf{C}$	Decentralized controller
$C_A, C_B, c_{A,}\, c_B$	Product concentrations
C_{Af}	Feed concentration of component A
C_I	Initiator concentration
C_m	Monomer concentration
C_p	Average heat capacity
C_{pc}, C_{pw}	Coolant heat capacity
c_{A0}	Inlet concentration of component A
d_i	Distance between $\mathbf{x}_i$ and $\mathbf{x}_q$
E_i	Activation energy
F	Inlet flow rate
F_I	Initiator flow rate
F_L	Low-pass filter
F_N	Second filter in NLIMC
h	Heat transfer coefficient
$[I_f]$	Initiator concentration in feed
k_i	Kinetic parameter
$k_{i,0}$	Arrhenius constant
$k_{\min}, k_{\max}$	JITL algorithm parameters

k_w	Coolant conductivity
L	Linear model of the process
$M, \tilde{\mathbf{G}}, G_m$	Model of the process
M_m	Molecular weight of monomer
M_+	Non-minimum-phase of M
M_-	Minimum-phase of M
m_w	Coolant mass
N	Nonlinear model of the process
$N(0)$	Number of information vectors stored in the initial database
n	Number of nearest-neighbors
P, $\mathbf{G}$	Actual process
p	Order of the second IMC filter
Q	IMC controller
Q_i	Initiator flow rate
Q_c	Cooling water flow rate
Q_m	Monomer flow rate
Q_s	Solvent flow rate
Q_t	Total flow rate
Q_w	Heat exchanger duty
q_1	Acid flow rate
q_2	Buffer flow rate
q_3	Base flow rate
r	Setpoint

s_i	Similarity number
T	Reactor temperature
T_0	Inlet temperature
T_c, T_w	Cooling water outlet temperature
T_{cf}	Cooling water inlet temperature
T_f	Reactor feed temperature
u	Process input
V	Reactor volume
W_a, W_b	Reaction invariants
w_i	Weight corresponding to *i*-th information vector
$\mathbf{x}_i, \mathbf{x}_q$	Information and query vector
y	Process output
$\hat{y}$	JITL model prediction
$\mathbf{z}$	Regression vector

Greek Letters

α, β	IMC filter time constants
γ	JITL algorithm parameter
θ_i	Angle between two vectors
ψ_i	JITL model parameter
ρ	Average density
ρ_c	Coolant density
ΔH_i	Heat of reaction
ε	Difference between setpoint and model output

η	Adaptive learning rate

Abbreviations

CSTR	Continuous stirred tank reactor
IMC	Internal model control
JITL	Just-in-time learning
MIMO	Multi-input multi-ouput
MSE	Mean-squared-error
NAMW	Number-average molecular weight
NLIMC	Nonlinear IMC
NN	Neural Network
RGA	Relative gain array
SISO	Single-input single-output

CHAPTER 1

Introduction

1.1 Motivations

It is well known that virtually all processes of practical importance exhibit some degree of nonlinear behaviour. Nevertheless, the vast majority of controller design techniques used for chemical processes are based on well-established results in linear control theory. For nonlinear systems, in particular, the predominant approach is linearization around an operating point followed by one of the controller design techniques developed for linear systems (e.g., linear optimal control, pole placement, characteristic loci, etc). For chemical processes which exhibit only mildly nonlinear dynamic behavior, the errors incurred by local linearization are small enough so that their effects on stability and performance can be satisfactorily handled by building sufficient robustness into the linear controllers. More recently, increasingly stringent requirements on product quality and energy utilization, as well as on safety and environmental responsibility, demand that a growing number of industrial processes operate in a range of operating points. Under this situation, the process dynamics is forced away from its nominal design condition, which exacerbates the effect of the inherent nonlinear nature of the process. As a result, it can create difficult stability and performance problems and therefore render the linear controllers unacceptable. There

is therefore increased industrial and academic interest in the development and implementation of controllers that will be effective when process nonlinearities cannot be ignored without serious consequences (Calvet and Arkun, 1988; Ogunnaike and Wright, 1996).

To alleviate aforementioned problems, a variety of controller design techniques for nonlinear system have recently been proposed. Among these, IMC is a convenient and powerful controller design strategy for the open-loop stable dynamic systems. The IMC is significant because the stability and robustness properties of the structure can be analyzed and manipulated in a transparent manner, even for nonlinear systems. Thus IMC provides a general framework for nonlinear systems control. Such generality is not apparent in alternative approaches to nonlinear control (Hunt and Sbarbaro, 1991).

In literature, several nonlinear IMC (NLIMC) schemes that incorporate concepts from linear IMC have been developed recently. The initial approaches were using fundamental nonlinear model or nonlinear state-space model as a process model in IMC scheme (Economou et al., 1986a; Calvet and Arkun, 1988; Henson and Seborg, 1991). However, it is generally difficult to get accurate fundamental models of the processes and most of the times are not readily available in industrial practice because of a chronic lack of detailed and extensive knowledge required for their development.

The ability of multilayer feedforward neural networks (NN) to model almost any nonlinear function without a priori knowledge suggests that they may provide a promising approach for modeling nonlinear processes and utilizing them in IMC structure (Nahas et al., 1992). However, when dealing with large sets of data, this approach becomes less attractive because of the difficulties in specifying model

structure and the complexity of the associated optimization problem, which is usually highly non-convex. In addition, the problem of inverting a NN model is encountered. Several methods have been utilized for this inversion. One method involves training a separate NN model (i.e. a NN IMC controller) directly to learn the inverse dynamics. Although successful in some cases, this approach can often lead to steady-state offset because the product of the gains of the NN model and the NN controller does not necessarily yield unity (Nahas et al., 1992).

The above NLIMC control schemes that employ more realistic and often more complex nonlinear process descriptions typically sacrifice the simplicity associated with linear techniques in order to achieve improved performance. This is mainly due to the use of computationally demanding analytical or numerical methods and neural networks to learn the inverse process dynamics for the necessary construction of nonlinear operator inverses.

To overcome these difficulties, a promising NLIMC approach has recently been proposed to yield a flexible nonlinear model inversion (Doyle et al., 1995). This controller synthesis scheme based on partitioned model inverse retains the original spirit and characteristics of conventional (linear) IMC while extending its capabilities to nonlinear systems. In this control scheme, the nonlinear IMC controller consists of a standard linear IMC controller augmented by an auxiliary loop of nonlinear 'corrections'. Harris and Palazoglu (1998) investigated the use of Functional Expansion models in the aforementioned NLIMC scheme. However, expansion models such as Volterra model and Functional Expansion model are limited to fading memory systems and the radius of convergence is not guaranteed for all input magnitudes. In addition, these models share a common drawback in that they can

describe only a specific class of nonlinearity. This limitation restricts the implementation of these models in practice (Xiong and Jutan, 2002).

The problem of modeling a process from observed data has been the object of several disciplines from nonlinear regression to machine learning and system identification. Recent rapid developments of computer technologies enable us to memorize, fast retrieve and read out a large number of data. By effectively utilizing these advantages, Just-In-Time Learning (JITL) was recently developed as an attractive alternative for modeling the nonlinear systems (Cybenko, 1996; Aha et al., 1991; Atkeson et al., 1997; Bontempi et al., 1999, 2001; Cheng and Chiu, 2004).

1.2 Contributions

Inspired by the previous work done in the development of IMC strategy for nonlinear processes and modeling of this type of processes, two IMC design methods capable of controlling dynamic systems that operate over a wide range of operating regimes are developed. The main contributions of this thesis are as follows.

Firstly, a nonlinear IMC design based on partitioned model inverse is proposed for a class of nonlinear single-input and single-output (SISO) and multi-input and multi-output (MIMO) systems. This partitioned model consists of a linear model, which is obtained around an operating point, and a nonlinear model, which is identified by JITL algorithm. It is also shown that JITL model in the proposed control strategy can be made adaptive on-line readily by simply adding the new process data to the database. Simulation results demonstrate that proposed IMC deign gives better performance than the conventional IMC scheme.

Secondly, a memory-based IMC design approach is proposed. The proposed method employs JITL not only to update model parameters but also to adjust the parameters of IMC controller. At each sampling instant, the initial IMC filter

parameter is obtained using a controller database. In addition, parameter updating algorithm is developed by employing the steepest descent gradient method and is used to adjust the initial filter parameter on-line. Simulation results show that the proposed memory-based IMC scheme gives better performance than the benchmark PI/PID controller reported in the literature.

1.3 Thesis Organization

The thesis is organized as follows. Chapter 2 will introduce the basic knowledge on internal model control and review the concept of its extension to nonlinear systems and the recent developments in modeling of nonlinear processes. The detailed JITL algorithm is also presented in Chapter 2. Nonlinear IMC design method using both adaptive and non-adaptive JITL for SISO systems is developed in Chapter 3, while decentralized nonlinear IMC design method is presented in Chapter 4. The proposed memory-based IMC design method is developed in Chapter 5. The general conclusions are summarized in Chapter 6 along with some recommendations for future work in this area. An exhaustive literature is provided at the end of the thesis.

CHAPTER 2

Literature Review

This chapter will give a brief introduction to the research work that has been conducted in the control of nonlinear chemical processes using Internal Model Control (IMC) strategy. Also, recent developments in modeling of nonlinear processes are discussed. Some relevant theoretical background and modeling algorithm required for further development of thesis will also be presented.

2.1 General IMC Structure

Internal Model Control (IMC) structure was proposed by Garcia and Morari (1982). The general IMC structure is illustrated in Figure 2.1, where *P(s)* is the process to be controlled, *M(s)* represents the model of the process, and *Q(s)* is the IMC controller. The disturbance signal is omitted since the effect of disturbance and plant/model mismatch are indistinguishable in the closed loop (Garcia et al., 1989).

The IMC approach has two important advantages: (1) It explicitly takes into account model uncertainty, and (2) it allows the designer to trade-off control system performance against control system robustness to process changes and modeling errors (Seborg et al., 1989). The IMC controller is designed in two steps:

Step 1: The process model is factored as

$$M(s) = M_{+}(s)M_{-}(s) \tag{2.1}$$

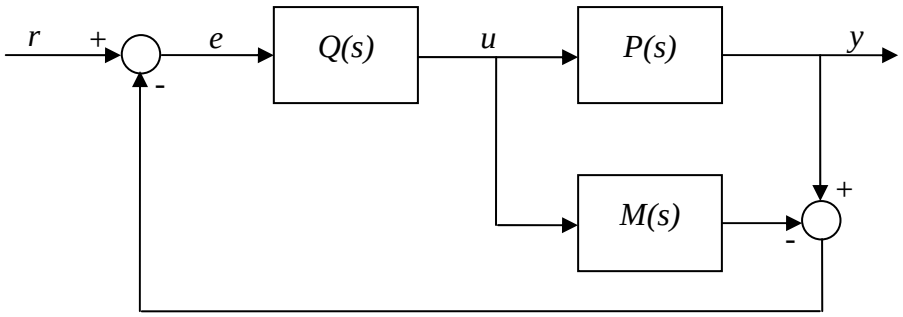

Figure 2.1 General IMC structure

where $M_+(s)$ is an all-pass element containing all the non-minimum-phase dynamics, and $M_-(s)$ contains a minimum-phase portion. In addition, $M_+(s)$ is specified such that its steady state gain is one.

Step 2: The controller is specified as

$$Q(s) = \frac{1}{M_-(s)} F_L(s) \tag{2.2}$$

where $F_L(s)$ is a low-pass filter with a steady-state gain of one.

Typically, this filter is given by

$$F_L(s) = \frac{1}{(\alpha s + 1)^r} \tag{2.3}$$

where α is the desired closed-loop time constant. Parameter *r* is a positive integer that is selected so that *Q(s)* is either a proper or strictly proper transfer function.

2.2 Linear IMC

The IMC scheme has been under intensive research and development in the last two decades due to its simple yet effective framework for system design. The idea inherent in the IMC has been floating around in one form or another for several decades. The IMC enables the transient response and the robustness to be addressed independently. Most of the existing advanced controllers such as linear quadratic

optimal controller, smith predictor and model predictive controller can be equivalently put into the general IMC form (Garcia and Morari, 1982; Fisher, 1991). The advantages of IMC are exploited in many industrial applications (Morari and Zafiriou, 1989).

Although many processes exhibit significant nonlinear behavior, most model-based controller design techniques are based on linear models. The prevalence of linear model-based control strategies is primarily due to two reasons. First, there are well-established methods for the development of linear models from input-output data while practical identification techniques for nonlinear models are still being developed. Furthermore, controller design for nonlinear models is considerably more difficult than for linear models (Nahas et al., 1992).

In available linear model-based control strategies, linear IMC is a convenient and powerful controller design strategy for the open-loop stable dynamic systems (Morari and Zafiriou, 1989). Linear IMC design is expected to perform satisfactorily as long as the plant is operated in the vicinity of the point where the process model is obtained. However, many chemical processes exhibit a certain degree of nonlinearity. Furthermore, different operating conditions are usually necessitated by the external factors such as the persistent load disturbances or the increasingly demand of product diversification and cost reduction, e.g. grade changeover in a polymerization reactor. Under this situation, the process dynamics is forced away from its nominal design condition, which exacerbates the effect of the inherent nonlinear nature of the process. As a result, the performance of linear IMC controller will degrade or even become unstable.

2.3 Nonlinear IMC

The development of a general extension of IMC to nonlinear systems poses serious difficulties due to the inherent complexity of nonlinear systems. For instance, except for very simple SISO systems, the IMC factorization procedure has no well-defined nonlinear analog (Kravaris and Daoutidis, 1990). Also, very few tools exist for the design and analysis of robust nonlinear controllers. Furthermore, linear IMC is based on transfer function models, while nonlinear systems are usually described by nonlinear state-space models. Despite these difficulties, several nonlinear controller design techniques that incorporate concepts from linear IMC have been developed recently. These design methods are reviewed below.

The nonlinear extension of IMC design was proposed by Economou et al. (1986a) for open-loop stable nonlinear systems with stable inverse. Input-output operators were used to show that their nonlinear IMC (NLIMC) technique satisfies the same stability, perfect control and zero offset properties as linear IMC. The controller was based on the inverse of the nonlinear model, and a linear filter was added to account for input constraints and modeling errors. Economou et al. (1986a) augmented the nonlinear controller with a linear filter because design techniques for nonlinear filters that preserve the nominal stability and no offset properties were not available. The stability of the model inverse was analyzed using the small gain theorem. Because the calculation of the required nonlinear gains is nontrivial (Nikolaou and Manousiouthakis, 1989), the stability theorems are difficult to use in practice. Although an input-output approach was used for analysis, the only analytical technique investigated for construction of the model inverse was the state-space approach of Hirschorn (1979). However, the Hirschorn inverse is internally unstable due to pole-zero cancellations at the origin (Kravaris and Kantor, 1990a, b). Hence,

the model inverse was constructed using numerical procedures based on the contraction mapping principle and Newton's method. The Newton method is reliable and efficient, but requires the solution of a linear variational problem. This numerical approach to nonlinear IMC is, therefore, computationally intensive. Moreover, analysis of the resulting iterative procedure is difficult (Economou and Morari, 1985; Li et al., 1990).

Calvet and Arkun (1988) used an IMC scheme to implement their state-space linearization approach for nonlinear systems in the presence of disturbances. A disadvantage of this approach is that an artificial controlled output is introduced in the controller design procedure and therefore is difficult to be specified a priori. Another disadvantage of this method is that the nonlinear controller requires state feedback.

Henson and Seborg (1991) proposed a general extension of linear IMC to nonlinear SISO systems by using global input-output linearization technique. Like the nonlinear IMC approach of Economou et al. (1986a), this new approach was restricted to open-loop stable systems with stable inverses. Also, their method relied on the availability of a nonlinear state-space model, which can be time-consuming and costly to obtain.

The ability of artificial neural networks to model almost any nonlinear function without a priori knowledge has lead to the investigation of nonlinear dynamic systems modeling using neural networks (NN). Several NLIMC schemes using NN have recently been proposed (Bhat and McAvoy, 1990; Hunt and Sbarbaro, 1991). Commonly, a NN model is trained to learn the inverse dynamics of the process and is employed as the nonlinear IMC controller. Because the process is modeled with a separate NN model, the NN controller might not invert the steady-state gain of the model exactly, resulting in steady-state offset. Moreover, these control schemes do

not provide a tuning parameter that can be adjusted to account for plant-model mismatch (Nahas et al., 1992).

To ensure offset-free performance, Nahas et al. (1992) proposed NLIMC strategy that also includes time delay compensation in the form of a Smith predictor. The nonlinear controller consists of a model inverse controller and a robustness filter with a single tuning parameter. In this control strategy, a numerical inversion of neural network process model was proposed instead of training neural networks on the process inverse. However, this numerical inversion is not only computationally demanding but also does not ensure global existence and uniqueness of a solution.

Aoyama et al. (1995) proposed a method using control-affine neural network models. Two neural networks were used in this approach: one for the model of the bias or drift term, and one for the model of the steady-state gain. As the process is approximated by a control-affine model, the inversion of process model is simply obtained by algebraically inverting the process model.

All of the above nonlinear control strategies sacrifice the simplicity associated with linear IMC in order to achieve improved performance. This is mainly due to the use of computationally demanding analytical or numerical methods and neural networks to learn the inverse process dynamics for the necessary construction of nonlinear operator inverses.

Recently, a partitioned model inverse has been proposed to yield a flexible nonlinear model inversion (Doyle et al., 1995). This controller synthesis scheme based on partitioned model inverse retains the original spirit and characteristics of conventional (linear) IMC while extending its capabilities to nonlinear systems. When implemented as part of the control law, the nonlinear controller consists of a standard linear IMC controller augmented by an auxiliary loop of nonlinear ‘corrections’. The

designer is free in the choice of the linear controller, and this element can be chosen as to address the control of nonminimum-phase dynamics. Furthermore, such a scheme has the advantage of providing an extra level in the hierarchical structure available to the control loop operator: instead of having ‘manual’ and ‘automatic’ as the only options, the operator now has the additional option of switching off only the auxiliary nonlinear loop, and downgrade, if necessary, not all the way to manual, but first to the basic linear scheme (Doyle et al., 1995). It is this flexibility that gives partitioned model inverses great promise in nonlinear control schemes. The fact that only a linear inversion is required in the synthesis of this controller is the most attractive feature of this scheme. However, Doyle et al. (1995) employs a Volterra model derived using local expansion results such as Carleman linearization, which is accurate for capturing local nonlinearities around an operating point, but may be erroneous in describing global nonlinear behavior (Maner et al., 1996).

Shaw et al. (1997) also employed a recurrent dynamic neural network within this partitioned model inverse controller synthesis scheme and showed that it provides an attractive alternative for NN-based control applications. Further, Maksumov et al. (2002) presented the first experimental application of this partitioned model inverse controller design strategy using NN as a nonlinear model and a linear ARX model. While the accuracy of NN models offers a potentially significant improvement over linear models, the process control engineer is faced with the daunting tasks of selecting model structure and initializing the optimization routine (Braun et al., 2001). Another fundamental limitation of these types of global approaches for modeling is that it is difficult for them to be updated on-line when the process dynamics are moved away from the nominal operating space.

Harris and Palazoglu (1998) employed a Functional Expansion model in the partitioned model inverse based IMC control scheme. However, these models are limited to fading memory systems and the radius of convergence is not guaranteed for all input magnitudes. Such limitations are typical for expansion models such as Volterra model and Functional Expansion model as discussed by Boyd and Chua (1985) and Schetzen (1980).

2.4 Process Identification

2.4.1 Introduction

In the competitive environment of the chemical and refining process industries, it is mandatory to maximize profit through *optimal process design* and *optimal plant operation*. Optimal process design leads to a high level of process integration in order to increase the efficiency of process energy and material utilization. Optimal plant operation causes frequent changes in feed stocks and production specifications, in order to adapt to changing market conditions. Thus, the trend towards optimal design and operation will significantly increase the complexity of encountered control problems. This development has been realized recently by major companies and as a consequence many companies have drastically increased the investment in development and implementation of advanced control strategies. Practical experience with advanced control, has demonstrated that process identification is the single most time consuming task. Once an adequate dynamic model has been obtained, 80-90% of the implementation is done. Therefore, there is an obvious need for more efficient and reliable methods for industrial process identification.

Conceptually there are three different approaches for process identification:

- White box: The identification is performed based on first-principles.

- Grey box: Both a priori process knowledge and experimental data are used for identification, e.g. only a subset of parameters is estimated from experimental data.
- Black box: The identification is performed exclusively from experimental data.

Just as in the case of control structure selection, proper selection of identification concept depends on the specific problem. In general white box identification leads to relatively complicated nonlinear models, in which parameter values are associated with a significant uncertainty. In the case of chemical processes, this one as well as grey box approach may well be infeasible, due to a lack of understanding of physical phenomena or due to the complexity of the problem (Andersen et al., 1991). On the other hand, black box approach is prepared to describe virtually any dynamics (Ljung, 1999). Hence there has been considerable recent interest in this area.

2.4.2 Data-Based approach

The problem of modeling a process from observed data has been the object of several disciplines from nonlinear regression to machine learning and system identification. In the literature dealing with this problem, three main paradigms have emerged: global, local and local memory-based.

Global modeling method builds a single functional model of the dataset. This has traditionally been the approach taken in neural network modeling, NARMAX models, fuzzy sets, wavelets and other kinds of nonlinear parametric models (Pearson and Ogunnaike, 1997; Su and McAvoy, 1997). These modeling methods compress all available information into a compact model. However, when dealing with large sets of data, this approach becomes less attractive to deal with because of the difficulties in specifying model structure and the complexity of the associated optimization problem, which is usually highly non-convex (Braun et al., 2001). Another drawback

is that, with the data essentially replaced by the model, there are no good methods to update models should new data become available (Cybenko, 1996).

Local modeling is a modular approach where the modules are simple models which focus on different part of the input space. This is the idea of operating regimes which assumes a partitioning of the operating range of the system in order to solve modeling and control problems. Fuzzy inference systems, radial basis functions, neuro-fuzzy network and hierarchical mixture of experts are well-known examples of this approach. It is important to remark that, although these architectures are characterized by an augmented readability, they still are a particular type of functional approximators. Also most local modeling approaches suffer from the drawback of requiring a priori knowledge to determine the partition of operating space (Bontempi et al., 2001).

Local memory-based models are a hybrid approach, leaning more in the direction of local modeling but using the power of global modeling in the local neighbourhood. In global modeling, a relatively simple problem (estimation of the function value) is solved by first solving a much more difficult intermediate problem (function estimation). Memory-based learning, on the other hand, turns out to be a single-step approach where the learning problem is seen as value estimation rather than a function estimation problem. Memory-based techniques are an old idea in classification, regression, and time-series prediction. The idea of memory-based approximators as alternative to global models originated in non-parametric statistics to be later rediscovered and developed in the machine learning fields (Bontempi et al., 2001). Aha et al. (1991) developed instance-based learning algorithms for modeling the nonlinear systems. Subsequent to Aha's work, different variants of instance-base

learning are developed, e.g. locally weighted learning (Atkeson et al., 1997) and Just-In-Time Learning (JITL) (Bontempi et al., 1999, 2001).

Comparing to the traditional methods like neural networks, JITL has no standard learning phase. It merely gathers the data and stores in the database and the computation is not performed until a query data arrives. It should be noted that JITL is only locally valid for the operating condition characterized by the current query data. In this sense, JITL constructs local approximation of the dynamic systems. Therefore a simple model structure can be chosen, e.g. a low order ARX model. Another advantage of JITL is its inherently adaptive nature, which is achieved by storing the current measured data into the database (Cheng and Chiu, 2004).

There are three main steps in JITL to predict the model output corresponding to the query data: (1) relevant data samples in the database are searched to match the query data by some nearest neighborhood criterion; (2) the data is weighted using a kernel or weighting function; (3) a local regression is performed using a linear model to build local model. Model output is calculated based on this local model and the current query data. The local model is then discarded right after the answer is obtained. When the next query data comes, a new local model will be built based on the aforementioned procedure.

In the literature, distance measures are overwhelmingly used in the JITL to evaluate similarity between two data samples. Recently, Cheng and Chiu (2004) developed an enhanced JITL methodology by exploring both distance measure and the complementary information available from the angular relationship. The detail algorithm is given below.

2.4.3 Just-In-Time Learning (JITL) algorithm

The detailed algorithm of the enhanced JITL methodology is described as follows (Cheng and Chiu, 2004). Given a database $(y_i, \mathbf{x}_i)_{i=1-N}$, where the vector $\mathbf{x}_i$ is formed by the past values of both process inputs and output, the parameters $k_{\min}$, $k_{\max}$, weight parameter γ, and a query data $\mathbf{x}_q$ whose elements are identical to those defined for $\mathbf{x}_i$:

Step 1: Compute the distance d_i between $\mathbf{x}_q$ and each $\mathbf{x}_i$, and the angle between $\Delta\mathbf{x}_q$ and $\Delta\mathbf{x}_i$ ($\Delta\mathbf{x}_q = \mathbf{x}_q - \mathbf{x}_{q-1}$ and $\Delta\mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_{i-1}$)

$$d_i = \left\| \mathbf{x}_q - \mathbf{x}_i \right\|_2, \; i = 1 \sim N \tag{2.4}$$

$$\cos(\theta_i) = \frac{\Delta\mathbf{x}_q^{\mathrm{T}} \Delta\mathbf{x}_i}{\left\| \Delta\mathbf{x}_q \right\|_2 . \left\| \Delta\mathbf{x}_i \right\|_2}, \; i = 1 \sim N \tag{2.5}$$

If $\cos(\theta_i) \geq 0$, compute the similarity number s_i

$$s_i = \gamma \sqrt{e^{-d_i^2}} + (1-\gamma)\cos(\theta_i) \tag{2.6}$$

If $\cos(\theta_i) < 0$, the data $(y_i, \mathbf{x}_i)$ is discarded.

Step 2: Arrange all s_i in the descending order. For $l = k_{\min}$ to $k_{\max}$, the relevant data set $(\mathbf{y}_l, \mathbf{\Phi}_l)$, where $\mathbf{y}_l \in \mathbf{R}^{l\times 1}$ and $\mathbf{\Phi}_l \in \mathbf{R}^{l\times n}$, are constructed by selecting *l* most relevant data $(y_i, \mathbf{x}_i)$ corresponding to the largest s_i to the *l*-th largest s_i. Denote $\mathbf{W}_l \in \mathbf{R}^{l\times l}$ a diagonal weight matrix with diagonal elements being the first *l* largest values of s_i, and calculate

$$\mathbf{P}_l = \mathbf{W}_l \mathbf{\Phi}_l \tag{2.7}$$

$$\mathbf{v}_l = \mathbf{W}_l \mathbf{y}_l \tag{2.8}$$

The local model parameters are then computed by

$$\boldsymbol{\psi}_l = \left(\mathbf{P}_l^{\mathrm{T}}\mathbf{P}_l\right)^{-1}\mathbf{P}_l^{\mathrm{T}}\mathbf{v}_l$$

where $\left(\mathbf{P}_l^{\mathrm{T}}\mathbf{P}_l\right)^{-1}$ is calculated by SVD method. Next, the leave-one-out cross validation test is conducted and the validation error is calculated by (Myers, 1990)

$$e_l = \frac{1}{\sum_{j=1}^{l} s_j^2}\sum_{j=1}^{l}\left(s_j \frac{y_j - \boldsymbol{\varphi}_j^{\mathrm{T}}\left(\mathbf{P}_l^{\mathrm{T}}\mathbf{P}_l\right)^{-1}\mathbf{P}_l^{\mathrm{T}}\mathbf{v}_l}{1-\mathbf{p}_j^{\mathrm{T}}\left(\mathbf{P}_l^{\mathrm{T}}\mathbf{P}_l\right)^{-1}\mathbf{p}_j}\right)^2 \tag{2.9}$$

where y_j is *j*-th element of $\mathbf{y}_l$, $\boldsymbol{\varphi}_j$ and $\mathbf{p}_j$ are the *j*-th row vectors of $\boldsymbol{\Phi}_l$ and $\mathbf{P}_l$ respectively.

Step 3: According to the validation errors, the optimal *l* is determined by

$$l_{\mathrm{opt}} = \arg\min_{l}\left(e_l\right) \tag{2.10}$$

Step 4: Verify the stability of local model built by the optimal model parameters $\boldsymbol{\psi}_{l_{\mathrm{opt}}}$. Because JITL constructs the local approximation of the dynamic systems, only the stability constraints of first- and second-order models are given as follows:

First-order model:

$$-1 < \psi_1 < 1 \tag{2.11}$$

Second-order model:

$$\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix} < \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{2.12}$$

$$-1 < \psi_2 < 1 \tag{2.13}$$

If $\boldsymbol{\psi}_{l_{\mathrm{opt}}}$ satisfies the stability constraint, the predicted output for query data is computed by

$$\left(\hat{y}_q\right)_{l_{\mathrm{opt}}} = \mathbf{x}_q^{\mathrm{T}}\boldsymbol{\psi}_{l_{opt}} \tag{2.14}$$

Otherwise, $\boldsymbol{\psi}_{l_{\mathrm{opt}}}$ is used as the initial value in the following optimization problem subject to the appropriate stability constraint,

$$\min_{\boldsymbol{\psi}} \left\| \mathbf{P}_{l_{\text{opt}}} \boldsymbol{\psi} - \mathbf{v}_{l_{\text{opt}}} \right\|_2 \tag{2.15}$$

With the optimal solution $\boldsymbol{\psi}^*_{l_{\text{opt}}}$ obtained from Eq. (2.15), the predicted output for query data is then calculated as $\mathbf{x}_q^{\mathrm{T}} \boldsymbol{\psi}^*_{l_{\text{opt}}}$.

Step 5: When the next query data comes, go to step 1.

2.5 Decentralized Control

Decentralized control structures have found wide application in the large scale chemical process industries. The control of MIMO processes using full multivariable centralized control requires too many control loops with increased cost and complexity of design, and difficult implementation, tuning and maintenance problems (Chiu and Arkun, 1992). Though the full multivariable controllers provide better performance, the simpler decentralized controllers are widely used because of the following reasons (Skogestad and Morari, 1989):

- tuning and retuning is simple
- they are easy to understand
- they are easy to make failure tolerant.

Decentralized control involves using a diagonal or block-diagonal controller as shown in Figure 2.2, where $\mathbf{G}(s)$ is the plant and $\mathbf{C}(s)$ is controller.

$$\mathbf{C}(s) = \text{diag}\{c_i(s)\} \tag{2.16}$$

The design of a decentralized control system involves two main steps:

(1) control structure selection, that is, pairing of process inputs and outputs; and

(2) design of a SISO controller for each loop.

The best way to proceed for each of these steps is still an active area of research. The RGA has proven to be an efficient tool for eliminating undesirable

pairings in step 1. For step 2, two classes of design procedures have been reported in the literature. The first class is independent design where each controller element is designed independently of each other (Grosdidier and Morari, 1986; Skogestad and Morari, 1989). The main advantage of this approach is that resulting system is failure tolerant i.e. nominal stability (of the remaining system) is guaranteed if any loop fails. However, this approach is potentially conservative since during the design of a particular controller the information on other controllers is not exploited (Skogestad and Morari, 1989).

The second class is sequential design in which controller design is conducted sequentially (Chiu and Arkun, 1989; Viswanadham and Taylor, 1988). Usually the controller corresponding to a fast loop is designed first. This loop is then closed before the design proceeds with the next controller. This means that the information about the "lower-level" controllers is directly used as more loops are closed; therefore, the method can be less conservative than independent design.

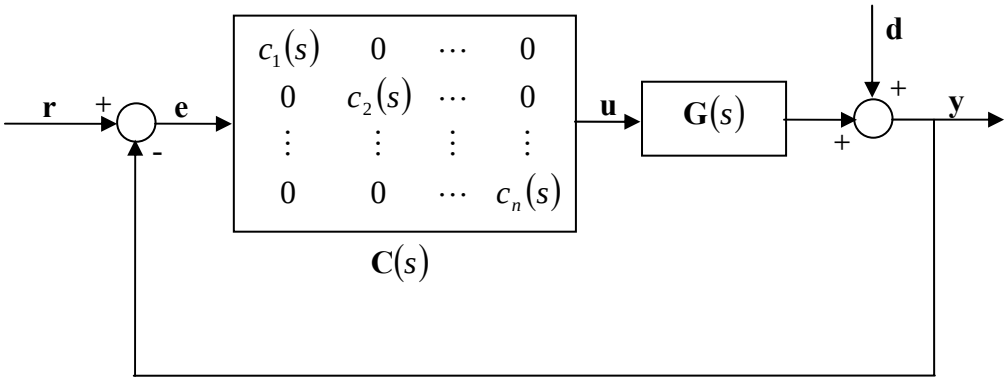

Figure 2.2 Decentralized control structure

CHAPTER 3

Nonlinear Internal Model Control Design for SISO Systems

Traditionally, model based control strategies for chemical processes are to design linear controller based on the linearized model. For open-loop stable dynamic systems, IMC is a convenient and powerful controller design strategy. Although most of chemical processes are nonlinear in nature, the IMC controller is able to perform satisfactorily as long as the plant is operated in the vicinity of the point where the linearization is generated. When the plant is to be operated in a wide range of operating conditions in consequence of large setpoint changes and/or the presence of disturbances, the IMC controller based on nonlinear models can be employed. The IMC structure shown in Figure 2.1 is sufficiently general to allow the use of variety of process models, such as fundamental nonlinear models, as well as NN and black-box models. The difficulty in the use of these models in the IMC strategy arises in the design of IMC controller, which is based on the inverse of the model. As a result, a reliable and efficient method is required to achieve this inversion (Maksumov et al., 2002). In the case of fundamental models, this inversion can be done analytically or numerically. However, generally it is difficult to get accurate fundamental models of the processes and most of the times are not available. In case of black-box model such

as the NN, the problem of inverting a model is encountered. Several methods have been utilized for this inversion. One method involves training a NN directly to learn the inverse dynamics. Although successful in some cases, this approach can often lead to offset because the product of the gains of the model NN and the controller NN does not necessarily yield unity. In literature, numerical inversion techniques have also been employed; however, this approach can be computationally demanding (Maksumov et al., 2002). Other methods in literature have proposed the partitioned model inverse to yield a flexible nonlinear model inversion for Volterra and Functional Expansion models (Doyle et al., 1995; Harris and Palazoglu, 1998). However, Volterra model is derived using local expansion results such as Carleman linearization, which is accurate for capturing local nonlinearities around an operating point, but may be erroneous in describing global nonlinear behavior (Maner et al., 1996). On the other hand, Functional Expansion models are limited to fading memory systems and the radius of convergence is not guaranteed for all input magnitudes. Consequently, the resulting controller gives satisfactory performance only for a limited range of operation (Harris and Palazoglu, 1998).

By utilization of the partitioned model inverse control scheme and Just-In-Time Learning (JITL) technique described in Chapter 2, a nonlinear IMC (NLIMC) design strategy is proposed in this chapter for a class of nonlinear systems that operate over a wide range of operating regimes. Two literature examples are used to illustrate the proposed control strategy and a comparison with the conventional IMC is made.

3.1 Proposed Nonlinear IMC Strategy

In this work, partitioned model is utilized to yield a flexible nonlinear model inversion. Considering a process for which a linear (L) and a nonlinear (N) model are available, the models can be combined into a composite model M as

$$M = L + (N - L) \tag{3.1}$$

Using operator algebra, it is then straightforward to show that the inverse of this composite model is given by

$$M^{-1} = [I + L^{-1}(N - L)]^{-1} L^{-1} \tag{3.2}$$

Note that only the inverse of the linear model is required. Additionally, this inverse can be computed on-line using the feedback loop illustrated in Figure 3.1. In the case of nonlinear systems with non-minimum-phase dynamics, Doyle et al. (1995) have shown that this partitioned model inverse structure is flexible enough to allow for the computation of pseudo inverse, i.e. the inverse of only minimum-phase dynamics of the process, meaning that L^{-1} is replaced by L_{-}^{-1}, where L_{-} denotes the minimum-phase of linear model *L* and hence above equation can be written as

$$M^{-1} = [I + L_{-}^{-1}(N - L)]^{-1} L_{-}^{-1} \tag{3.3}$$

Here, we use this partitioned model inverse structure in IMC control scheme, with linear model *L* obtained around an operating point and nonlinear model obtained by JITL algorithm. The resulting IMC controller, referred to NLIMC henceforth, has the structure illustrated in Figure 3.2, where *Q* is the standard linear IMC controller

$$Q(s) = L_{-}^{-1}(s) F_L(s) \tag{3.4}$$

where F_L is a low-pass filter. Typically, this filter is given by

$$F_L(s) = \frac{1}{(\alpha s + 1)^r} \tag{3.5}$$

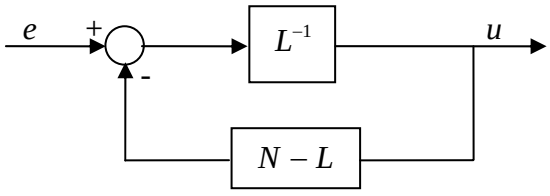

Figure 3.1 Partitioned model inverse

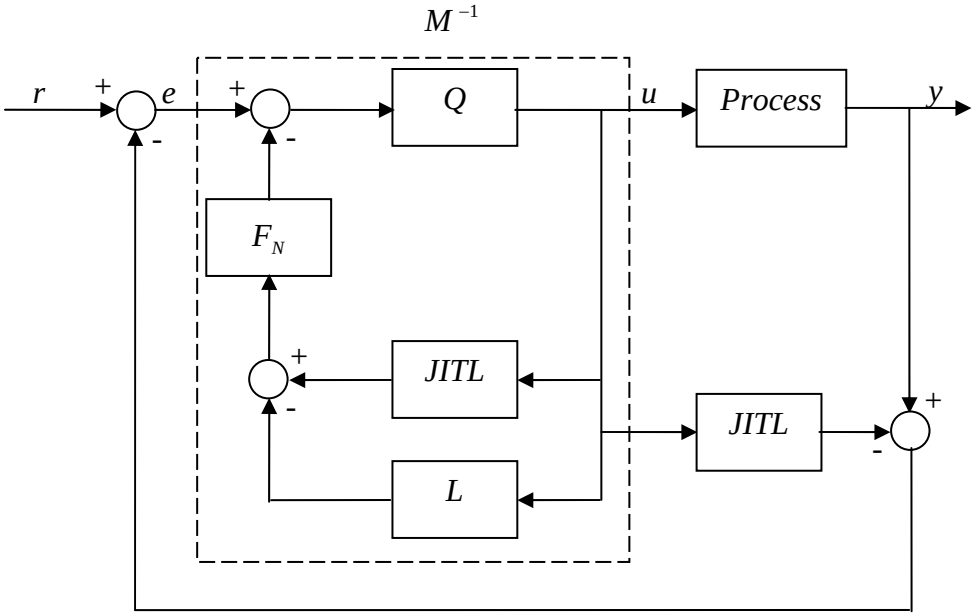

Figure 3.2 NLIMC structure with partitioned controller

where *r* is the relative degree of the system and α acts as a tuning parameter.

The second filter F_N is used to provide robustness for the nonlinear IMC in the same spirit as linear IMC and this filter is chosen as the inverse of conventional robustness filter,

$$F_N(s) = F_L^{-1}(s) \tag{3.6}$$

However, a more practical choice for this filter is given by (Harris and Palazoglu, 1998),

$$F_N(s) = \frac{F_L^{-1}(s)}{(\beta s + 1)^p} \tag{3.7}$$

where β and p are design (tuning) parameters. Typically, $\beta < \alpha$ and p is chosen such that $F_N(s)$ is proper, i.e. $p = r$. The practical considerations for using this $F_N(s)$ are as follows:

- Selection of F_N by Eq. (3.6) can lead to non-causal elements in the control loop. Use of the modified filter circumvents this problem.

- Application of Eq. (3.6) can result in the amplification of noise, while the modified filter has tuning parameters to attenuate excessive noise.
- The modified filter of Eq. (3.7) can be used to stabilize the closed-loop. This is required in cases when the inverse does not exist over the range of operation.

The tuning of Eq. (3.7) leads to two limiting cases. If $\beta \to 0$, the full nonlinear control is achieved. The second case is for $F_N \to 1$, i. e. $\alpha = \beta$. Here, the behavior of the closed-loop approaches that of the linear IMC scheme (Harris et al., 2003).

As mentioned in Chapter 2, JITL has adaptive nature, which is achieved by storing the current measured data into the database. In case of adaptive JITL, initial database is constructed using process data collected in the small range of operating region and subsequently database is updated on-line at each sampling instant, whenever necessary as determined by the following criterion: when the modeling error between the process output and the predicted output by JITL algorithm is greater than the threshold value, the current process data is considered as 'new' data that is not adequately represented by the present database and is thus added to the database. In contrast, non-adaptive JITL algorithm makes use of process data collected over the operating region and this database is kept fixed during on-line application of the proposed NLIMC method. The performance of these two algorithms will be evaluated for two literature examples in the following simulation study. Non-adaptive JITL algorithm in the proposed control strategy is expected to perform well as long as the process is operated in the region for which process data is available to construct database. However, in many chemical processes, different operating conditions are usually necessitated by the external factors such as the increasingly demand of product diversification and cost reduction, e.g. grade changeover in a polymerization

reactor. In this situation, process data required to construct database for this new operating region may not be available. As a result, from the on-line application point of view, adaptive JITL algorithm in the proposed control strategy is preferred over its counterpart.

In comparison, online adaptation of NN and neuro-fuzzy models require model update from scratch, namely both network structure (e.g. the number of hidden neurons in the former case and the number of fuzzy rules in the latter) and model parameters may need to be changed simultaneously. Evidently, this process is not only time-consuming, but also it will interrupt the plant operation, if these models are used for other purposes like model based controller design (Cheng and Chiu, 2004).

3.2 Examples

Example 1: The proposed NLIMC strategy is applied to a polymerization reaction taking place in a jacketed CSTR. The reaction involves free-radical polymerization of methyl methacrylate (MMA) with azo-bis-isobutyronitrile (AIBN) as initiator and toluene as solvent (Maner et al., 1996; Doyle et al., 1995; Congalidis et al., 1989; Daoutidis et al., 1990). A schematic of the process is shown in Figure 3.3. The following simplifying assumptions are made to obtain model for this reactor (Doyle et al., 1995): (a) isothermal operation; (b) perfect mixing; (c) constant heat capacity; (d) no polymer in the inlet stream; (e) no gel effect; (f) constant reactor volume; (g) negligible initiator flow rate (in comparison with monomer flow rate); (h) quasisteady state and long-chain hypothesis. Under these assumptions, the six-state model in Daoutidis et al. (1990) reduces to the following four-state model:

$$\frac{dC_m}{dt} = -\left(k_p + k_{f_m}\right)C_m P_0 + \frac{F\left(C_{m_{in}} - C_m\right)}{V}, \tag{3.8}$$

$$\frac{dC_I}{dt} = -k_I C_I + \frac{F_I C_{I_{in}} - FC_I}{V}, \tag{3.9}$$

$$\frac{dD_0}{dt} = \left(0.5k_{T_c} + k_{T_d}\right)P_0^2 + k_{f_m} C_m P_0 - \frac{FD_0}{V}, \tag{3.10}$$

$$\frac{dD_1}{dt} = M_m\left(k_P + k_{f_m}\right)C_m P_0 - \frac{FD_1}{V}, \tag{3.11}$$

$$y = \frac{D_1}{D_0}. \tag{3.12}$$

where y denotes the number-average molecular weight (NAMW) and

$$P_0 = \left[\frac{2f^* k_I C_I}{k_{T_d} + k_{T_c}}\right]^{1/2}.$$

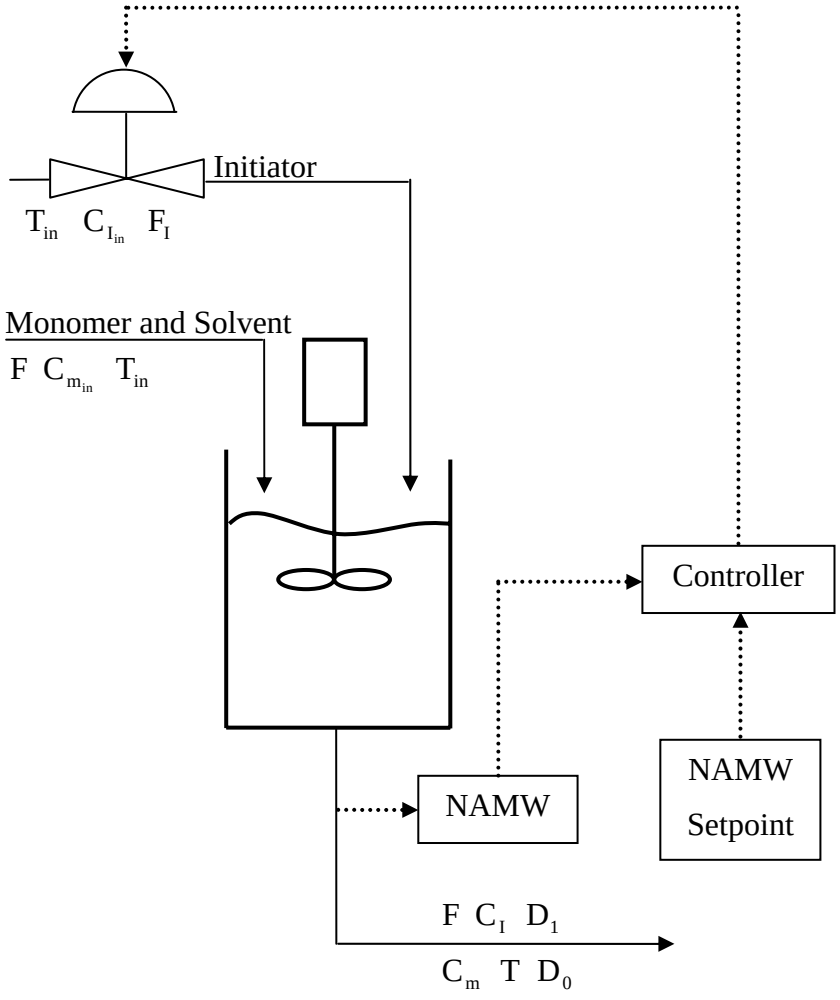

Figure 3.3 Control configuration for polymerization reactor

Table 3.1 Parameters for polymerization reactor

$k_{T_c} = 1.3281 \times 10^{10}\ \mathrm{m^3\ kmol^{-1}\ h^{-1}}$	$F = 1.00\,\mathrm{m^3}$
$k_{T_d} = 1.0930 \times 10^{11}\ \mathrm{m^3\ kmol^{-1}\ h^{-1}}$	$V = 0.1\,\mathrm{m^3}$
$k_I = 1.0225 \times 10^{-1}\ \mathrm{l\,h^{-1}}$	$C_{I_{in}} = 8.0\,\mathrm{kmol\,m^{-3}}$
$k_P = 2.4952 \times 10^{6}\ \mathrm{m^3\ kmol^{-1}\ h^{-1}}$	$M_m = 100.12\,\mathrm{kg\,kmol^{-1}}$
$k_{f_m} = 2.4522 \times 10^{3}\ \mathrm{m^3\ kmol^{-1}\ h^{-1}}$	$C_{m_{in}} = 6.0\,\mathrm{kmol\,m^{-3}}$
$f^* = 0.58$	

Table 3.2 Nominal operating conditions for polymerization reactor

$x_1 = C_m = 5.506774\,\mathrm{kmol\,m^{-3}}$	$x_4 = D_1 = 49.38182\,\mathrm{kg\,m^{-3}}$
$x_2 = C_I = 0.132906\,\mathrm{kmol\,m^{-3}}$	$u = F_I = 0.016783\,\mathrm{m^3 h^{-1}}$
$x_3 = D_0 = 0.0019752\,\mathrm{kmol\,m^{-3}}$	$y = 25000.5\,\mathrm{kg\,kmol^{-1}}$

The model parameters and nominal operating conditions are given in Tables 3.1 and 3.2 respectively. The control objective is to manipulate the volumetric flow rate of the initiator $(u = F_I)$ in order to regulate the process output *y*. The operating space considered is $\mathrm{NAMW} \in [12500, 37500]$. For this example, the sampling time is 0.03 hr and in the following simulation studies, the step change in the process input (open-loop test) or setpoint is made at the time equal to 0.15 hr.

Introducing the values in Table 3.1 for parameters in the modeling equations yields:

$$\dot{x}_1 = 10(6 - x_1) - 2.4568 x_1 \sqrt{x_2}, \tag{3.13}$$

$$\dot{x}_2 = 80u - 10.1022 x_2, \tag{3.14}$$

$$\dot{x}_3 = 0.0024121 x_1 \sqrt{x_2} + 0.112191 x_2 - 10 x_3, \tag{3.15}$$

$$\dot{x}_4 = 245.978 x_1 \sqrt{x_2} - 10 x_4, \tag{3.16}$$

$$y = \frac{x_4}{x_3}. \tag{3.17}$$

From this first-principles reactor model, we now proceed to obtain a linear model via Taylor series approximation approach. By defining the normalized variables $\tilde{z}_i = (x_i - x_{i0})/x_{i0}$, $\tilde{y} = (y - y_0)/y_0$ and $\tilde{u} = (u - u_0)/u_0$, where x_{i0}, y_0 and u_0 are nominal operating values for the corresponding process variables, we obtain a model with state variables that are zero at the nominal operating condition. After taking Taylor series approximation of this normalized model up to the first-order term, we obtain linear state-space representation of the nonlinear system as follows:

$$\dot{\tilde{\mathbf{z}}} = \mathbf{A}\tilde{\mathbf{z}} + \mathbf{b}\tilde{u} \tag{3.18}$$

$$\tilde{y} = \mathbf{c}\tilde{\mathbf{z}} \tag{3.19}$$

where $\tilde{\mathbf{z}} = [\tilde{z}_1 \quad \tilde{z}_2 \quad \tilde{z}_3 \quad \tilde{z}_4]^{\mathrm{T}}$ and the matrices in these equations are given by:

$$\mathbf{A} = \begin{bmatrix} -10.8957 & -0.447837 & 0 & 0 \\ 0 & -10.1022 & 0 & 0 \\ 2.45162 & 8.7744 & -10 & 0 \\ 10 & 5.00001 & 0 & -10 \end{bmatrix},$$

$$\mathbf{b} = [0 \quad 10.1022 \quad 0 \quad 0]^{\mathrm{T}},$$

$$\mathbf{c} = [0 \quad 0 \quad -1 \quad 1].$$

Using the matrices given above, the linear reactor model can be described by the following Laplace transfer function:

$$L(s) = \mathbf{c}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} = \frac{-38.129643(s+10)(s+11.7913)}{(s+10.8957)(s+10.1022)(s+10)^2} \tag{3.20}$$

To proceed with non-adaptive JITL algorithm, first-order ARX model is employed as local model, i.e. the regression vector is chosen as $\mathbf{z}(k-1)=[\tilde{y}(k-1),\tilde{u}(k-1)]^{\mathrm{T}}$. The database is generated by introducing uniformly random steps with distribution of [0.0045 0.078] in process input as shown in Figure 3.4. The JITL algorithm parameters, $k_{\min}=20$, $k_{\max}=60$, and $\gamma=0.95$, are chosen to achieve the smallest mean-squared-error (MSE) in the validation test. To illustrate the predictive performance of JITL algorithm, $\pm 50\%$ step changes in F_I from its nominal value of 0.016783 $\mathrm{m^3\,h^{-1}}$ are considered as shown in Figure 3.5, where the predicted output of the JITL tracks the actual nonlinear plant output very closely and consequently their respective responses are indistinguishable. In comparison, it is evident that linear model given in Eq. (3.20) fails to provide accurate prediction of reactor dynamics in the aforementioned open-loop tests.

To design both IMC controller *Q* and NLIMC controller, the former is chosen as the inverse of linear model $L(s)$ given in Eq. (3.20) with augmentation of the following filter $F_L(s)$:

$$F_L(s)=\frac{1}{(0.2s+1)^2} \tag{3.21}$$

and NLIMC consists of the identical *Q* and the second filter $F_N(s)$ is chosen as

$$F_N(s)=\frac{(0.2s+1)^2}{(0.07s+1)^2} \tag{3.22}$$

To evaluate the performance of two IMC designs, step changes of $\pm 50\%$ in setpoint from its nominal value of 25000.5 kg / kmol are conducted. In addition, the ideal closed-loop transfer function for setpoint change under perfect model assumption, i.e. M_+F_L, is used as the reference trajectory, which is the benchmark performance for both conventional IMC and the proposed NLIMC controllers (Doyle

et al., 1995). In this example, linear model does not contain non-minimum-phase dynamics, and thus reference trajectory is the same as filter response. As can be seen from Figure 3.6, the proposed controller effectively cancels the process nonlinearity and forces the process to behave like a linear process. If the process nonlinearity can be cancelled by the controller entirely, the closed-loop response shown in Figure 3.6 would be identical to the reference trajectory, i.e. process behaviour would exactly match linear model. It is also evident that the NLIMC scheme tracks the reference trajectory more closely than the response obtained by linear IMC scheme. The latter leads to an overshoot response for the positive step change in setpoint, which may be undesirable if product specifications require the molecular weight to be less than $37500\ \mathrm{kg/kmol}$. While linear IMC could be detuned to yield an overdamped response for the positive step change in setpoint, detuning would cause performance deterioration for negative step change in setpoint, i.e. a more sluggish response. Hence, the nonlinear behavior of this process requires a compromise in the tuning of a linear model-based controller. To compare disturbance rejection capability of both IMC designs, unmeasured $\pm 25\%$ step disturbances in inlet initiator concentration $\left(C_{I_{in}}\right)$ are considered. The resulting closed-loop responses at three different operating points are shown in Figures 3.7 and 3.8. In case of operating points other than nominal case in these figures, setpoint changes are made to move process towards new operating point (is not shown in figure) and then unmeasured disturbances are introduced once process reaches steady state. It is evident that the proposed control strategy gives a better performance than that obtained by linear IMC.

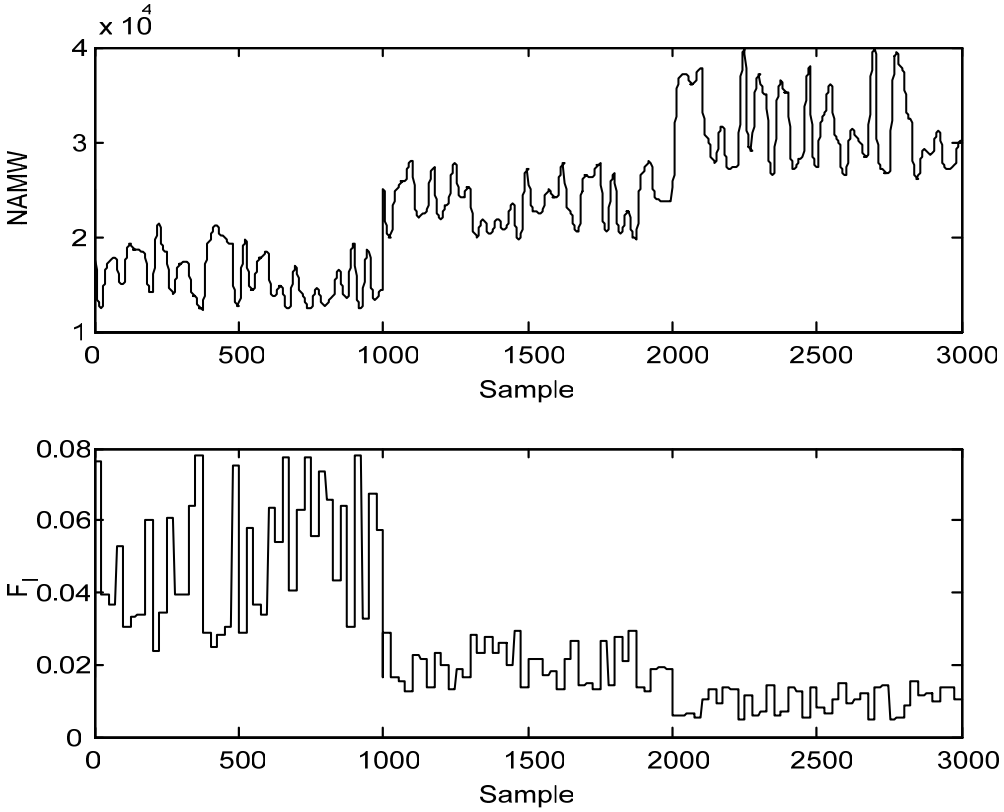

Figure 3.4 Input-ouput data used for constructing the database

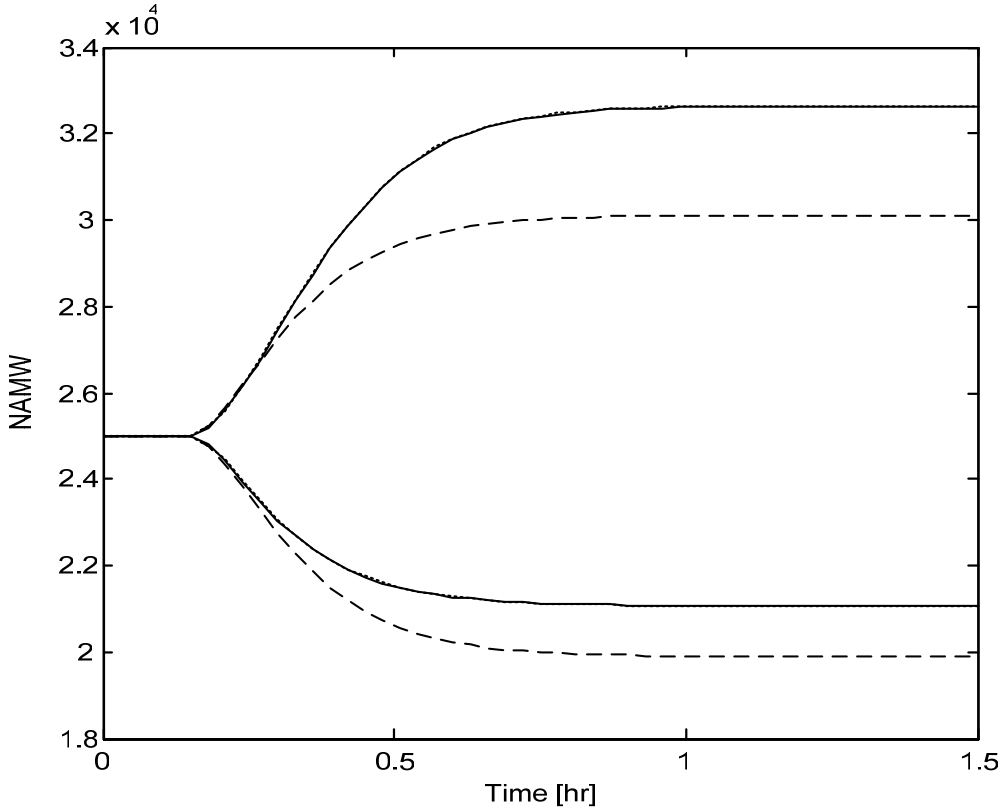

Figure 3.5 Open-loop responses for $\pm 50\%$ step changes in F_I. Solid: actual process; dashed: linear model; dotted: JITL

Table 3.3 summarizes the mean-squared-error (MSE) between the process output and the reference trajectory for both setpoint tracking and disturbance rejection performances aforementioned. It is clear that NLIMC scheme reduces the MSE significantly, relative to linear IMC scheme, by a margin between 68% and 88%.

Next, we will implement NLIMC scheme by using adaptive JITL algorithm. In doing so, the initial database is generated by introducing uniformly random steps in process input around its nominal value as shown in Figure 3.9. Again, first-order ARX model is chosen as local model and the same parameter values of $k_{\min}$, $k_{\max}$, and γ are employed. In addition, linear model and two filters used in NLIMC scheme are the same as those chosen for non-adaptive case. The criterion employed to update the database at each sampling instant is to check whether the difference between the predicted output by JITL algorithm and actual process output is within $\pm 5\%$ of the process output. The resulting closed-loop responses for aforementioned setpoint changes and disturbance rejection are shown in Figures 3.10 to 3.12. It is clear that proposed control strategy with adaptive JITL algorithm outperforms linear IMC, as also evidenced by the reduction of MSE as summarized in Table 3.4. Furthermore, the symbol "*" in these figures denotes the sampling instants at which database is updated and the number of new data points added to the initial database during each simulation study are listed in the last column of Table 3.4. It is evident that database is updated only in the transient state of the process. Although there is marginal difference in tracking error, the closed-loop responses obtained by the proposed control strategy with adaptive and non-adaptive JITL are indistinguishable. This shows that the performance of proposed control strategy with adaptive and non-adaptive JITL algorithm is almost similar. But from on-line application point of view, control strategy with adaptive JITL algorithm should be preferred because this control

scheme can perform well enough even though process condition moves towards completely new operating region necessitated by external factors such as market demand of product diversification.

Table 3.3 Comparison of closed-loop performances between IMC and non-adaptive NLIMC

Step change	Tracking error (MSE)		% Decrease in MSE
	IMC	NLIMC	
r = 25000.5 to 37500	4.6711×10^{6}	5.3854×10^{5}	88.47
r = 25000.5 to 12500	3.8276×10^{6}	1.0239×10^{6}	73.25
+25% change in $C_{I_{in}}$ at 15000	3.3353×10^{5}	1.0474×10^{5}	68.60
+25% change in $C_{I_{in}}$ at 25000.5	4.5772×10^{5}	1.0779×10^{5}	76.45
+25% change in $C_{I_{in}}$ at 35000	2.1414×10^{5}	4.9187×10^{4}	77.03
-25% change in $C_{I_{in}}$ at 15000	7.0989×10^{5}	2.2291×10^{5}	68.60
-25% change in $C_{I_{in}}$ at 25000.5	5.6498×10^{5}	1.4774×10^{5}	73.85
-25% change in $C_{I_{in}}$ at 35000	4.4749×10^{5}	1.3046×10^{5}	70.85

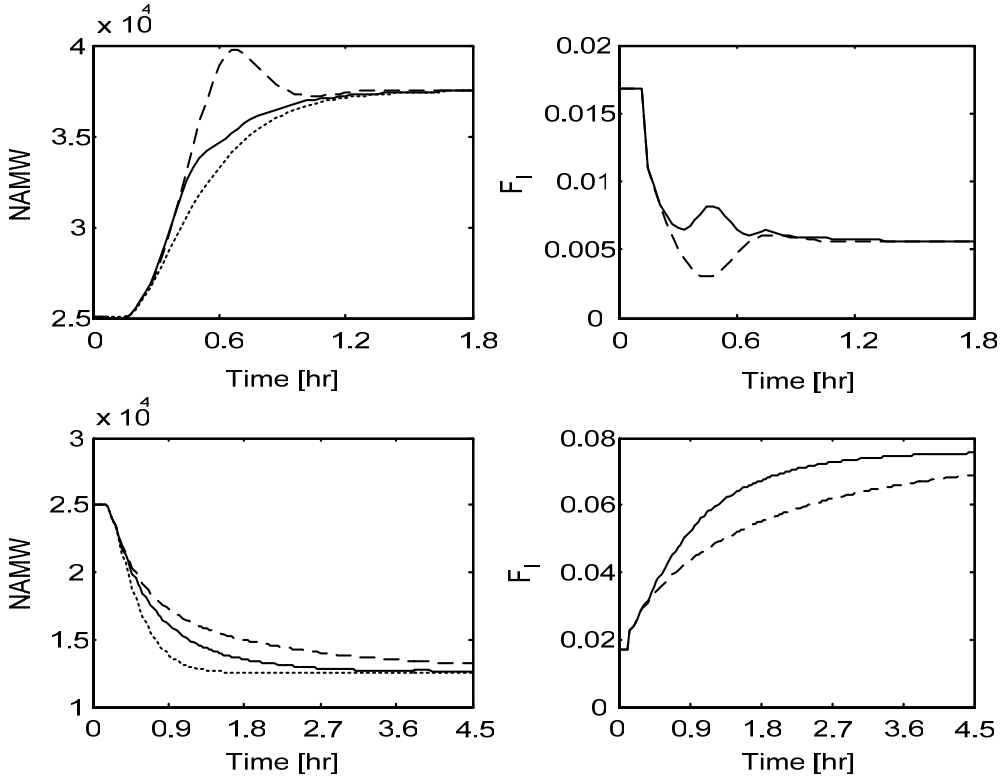

Figure 3.6 Closed-loop responses for $\pm 50\%$ step changes in setpoint. Dotted: reference trajectory; dashed: IMC; solid: NLIMC

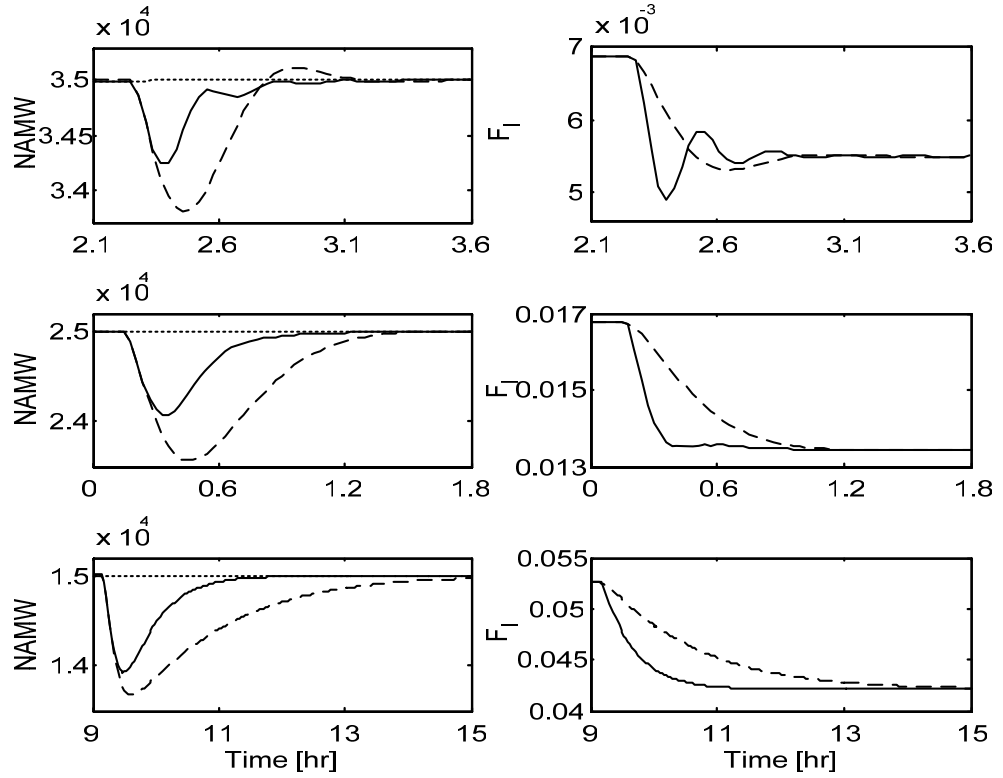

Figure 3.7 Closed-loop responses for $+25\%$ step change in $C_{I_{in}}$. Dotted: reference trajectory; dashed: IMC; solid: NLIMC

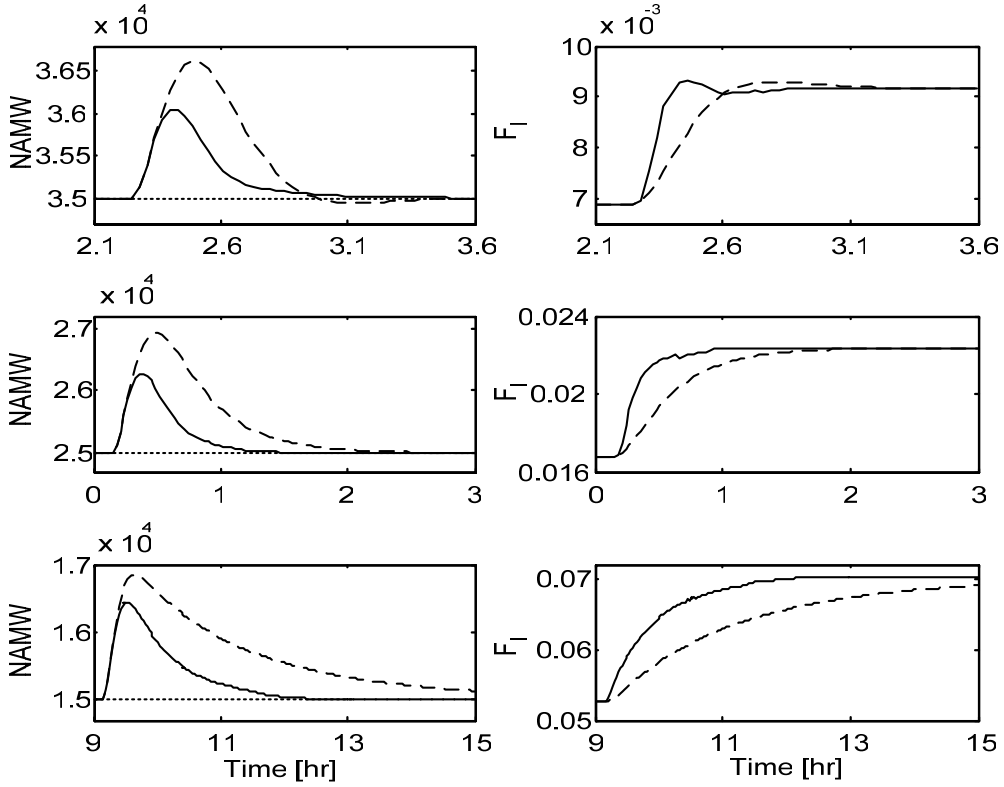

Figure 3.8 Closed-loop responses for -25% step change in $C_{I_{in}}$. Dotted: reference trajectory; dashed: IMC; solid: NLIMC

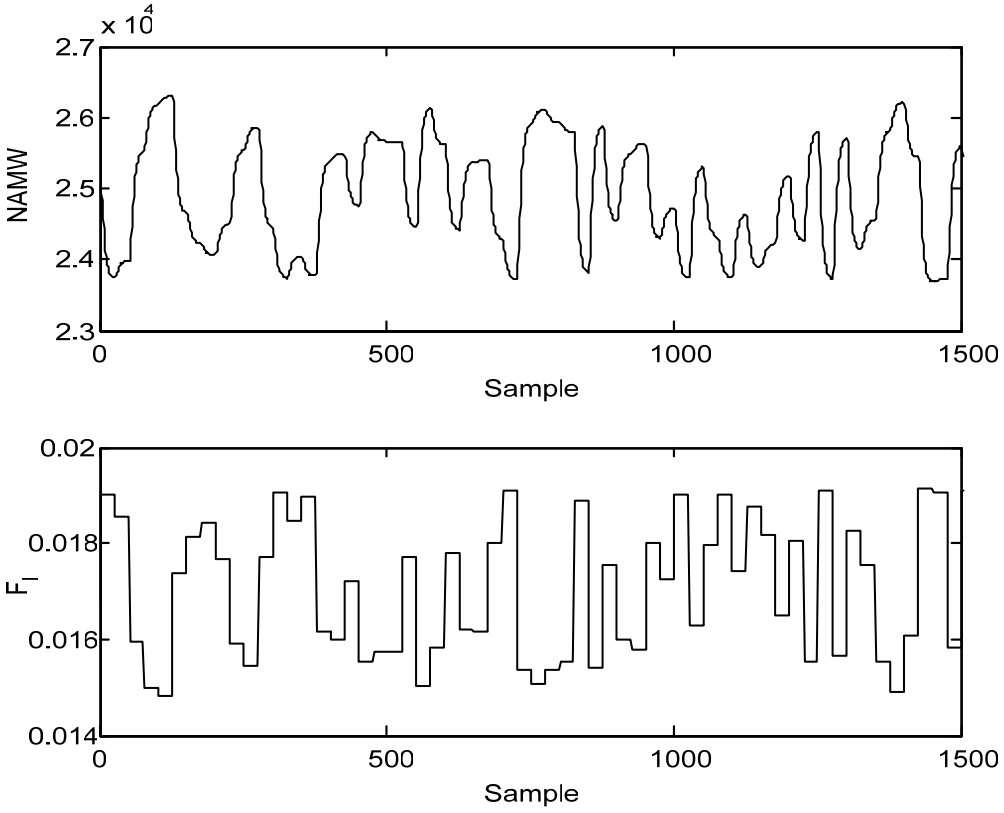

Figure 3.9 Input-ouput data used for constructing the initial database

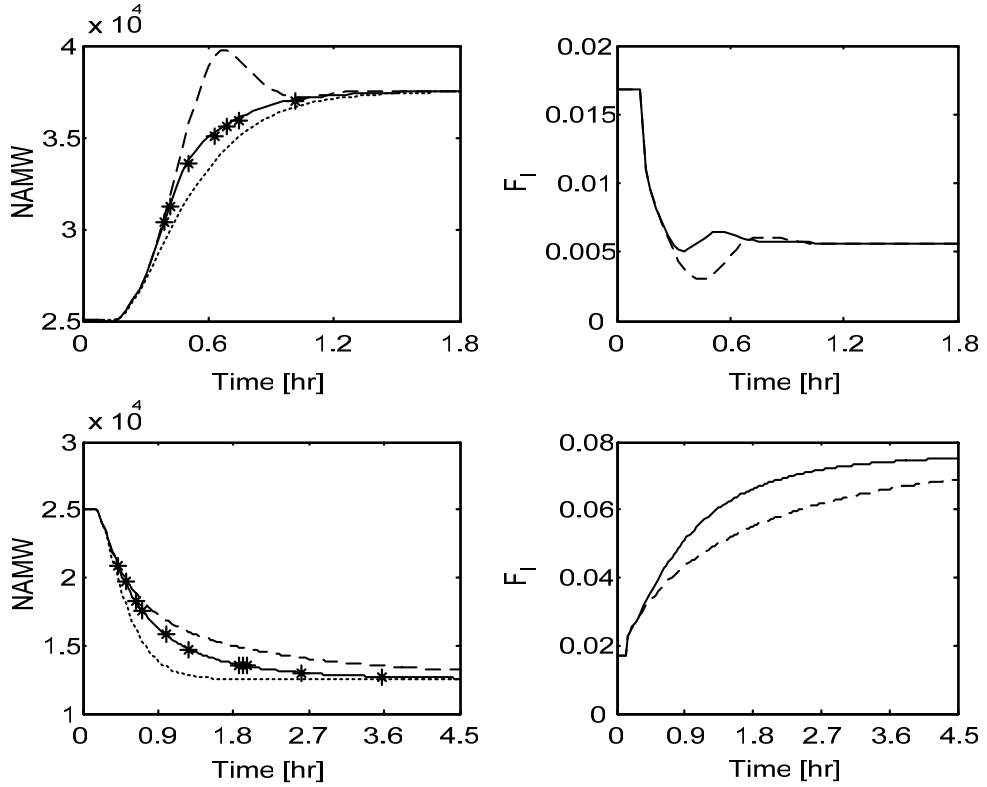

Figure 3.10 Closed-loop responses for $\pm 50\%$ step changes in setpoint. Dotted: reference trajectory; dashed: IMC; solid: NLIMC; star: database update

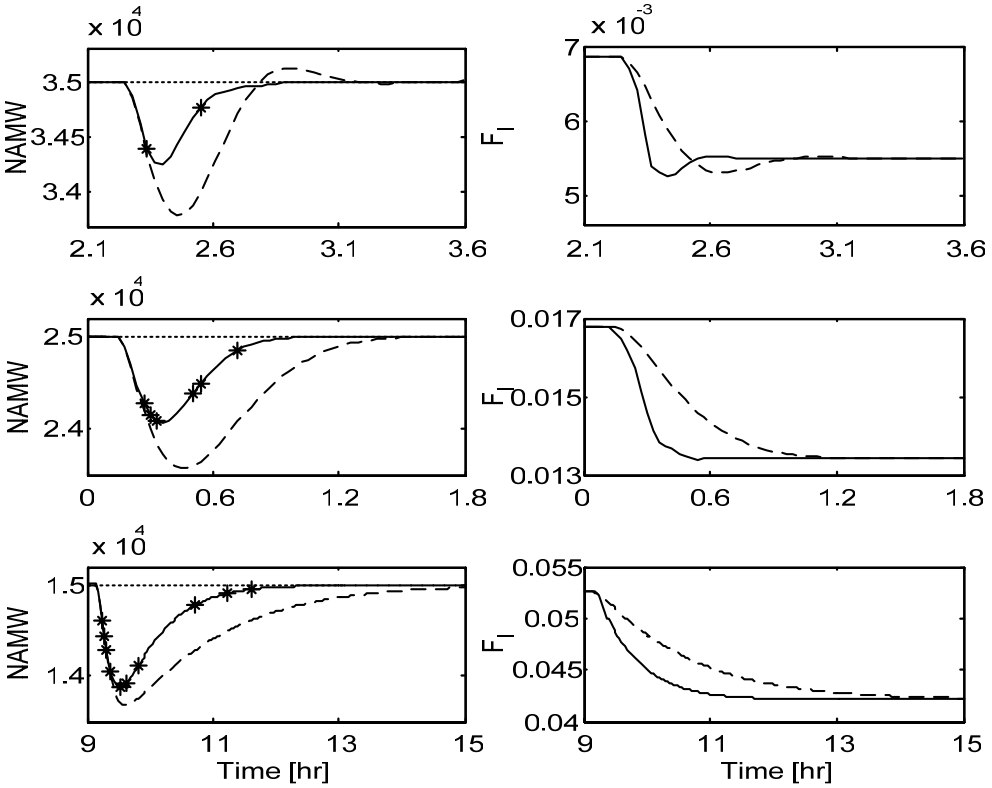

Figure 3.11 Closed-loop responses for $+25\%$ step change in $C_{I_{in}}$. Dotted: reference trajectory; dashed: IMC; solid: NLIMC; star: database update

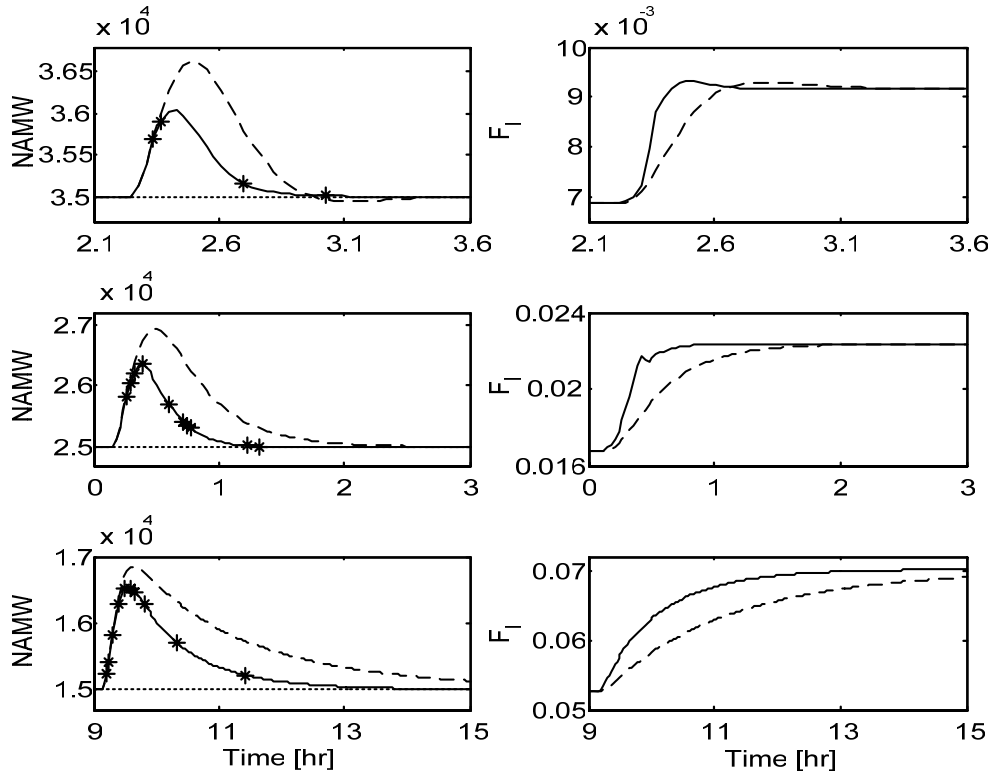

Figure 3.12 Closed-loop responses for -25% step change in $C_{I_{in}}$. Dotted: reference trajectory; dashed: IMC; solid: NLIMC; star: database update

Table 3.4 Comparison of closed-loop performances between IMC and adaptive NLIMC

Step change	Tracking error (MSE)		% Decrease in MSE	No. of data points added
	IMC	NLIMC		
r = 25000.5 to 37500	4.6711×10^{6}	6.2231×10^{5}	86.68	7
r = 25000.5 to 12500	3.8276×10^{6}	1.2395×10^{6}	67.62	11
+25% in $C_{I_{in}}$ at 15000	3.3353×10^{5}	1.3860×10^{5}	58.44	10
+25% in $C_{I_{in}}$ at 25000.5	4.5772×10^{5}	1.1789×10^{5}	74.24	6
+25% in $C_{I_{in}}$ at 35000	2.1414×10^{5}	5.5345×10^{4}	74.15	2
-25% in $C_{I_{in}}$ at 15000	7.0989×10^{5}	2.8862×10^{5}	59.34	10
-25% in $C_{I_{in}}$ at 25000.5	5.6498×10^{5}	1.6000×10^{5}	71.68	10
-25% in $C_{I_{in}}$ at 35000	4.4749×10^{5}	1.3109×10^{5}	70.71	4

Example 2: The second example used for the proposed NLIMC strategy is the van de Vusse reaction kinetic scheme involving following reactions:

$$\mathrm{A} \xrightarrow{k_1} \mathrm{B} \xrightarrow{k_2} \mathrm{C},$$

$$2\mathrm{A} \xrightarrow{k_3} \mathrm{D},$$

which is carried out in an isothermal CSTR. The process is described by the following nonlinear differential equations (Doyle et al., 1995):

$$\dot{C}_A = -k_1 C_A - k_3 C_A^2 + \frac{F}{V}\left(C_{Af} - C_A\right) \tag{3.23}$$

$$\dot{C}_B = k_1 C_A - k_2 C_B - \frac{F}{V} C_B \tag{3.24}$$

where C_A and $C_{\mathbf{B}}$ denote the concentration of components A and B respectively. The model parameters and nominal operating conditions used in the simulation are: $k_1 = 50\,\mathrm{h}^{-1}$, $k_2 = 100\,\mathrm{h}^{-1}$, $k_3 = 10\,\mathrm{l\,mol}^{-1}\,\mathrm{h}^{-1}$, $C_{Af} = 10\,\mathrm{mol\,l}^{-1}$, $V = 1\mathrm{L}$, $x_1 = C_A = 3.0\,\mathrm{mol\,l}^{-1}$, $x_2 = C_B = 1.12\,\mathrm{mol\,l}^{-1}$ and $u = F = 34.31\,\mathrm{h}^{-1}$. The control problem is to regulate the concentration of component B (C_B) by manipulating the inlet flow rate (F). This example has been considered by a number of researchers as a benchmark problem for nonlinear process control algorithms (van de Vusse, 1964; Kantor, 1986). A plot of this reactor's operating locus as shown in Figure 3.13 reveals a salient feature of this system, i.e. a change in sign of steady-state gain at the peak conversion level. In addition, this reactor displays non-minimum phase behavior to the left and minimum phase behavior to the right for the operation conditions of the maximum conversion.The operating space considered here is $C_B \in [0.62, 1.25]$, which exhibits non-minimum phase dynamics. For this example, the sampling time is 0.001 hr and the step change in the process input (open-loop test) or setpoint is made at the time equal to 0.01 hr in the following simulation studies.

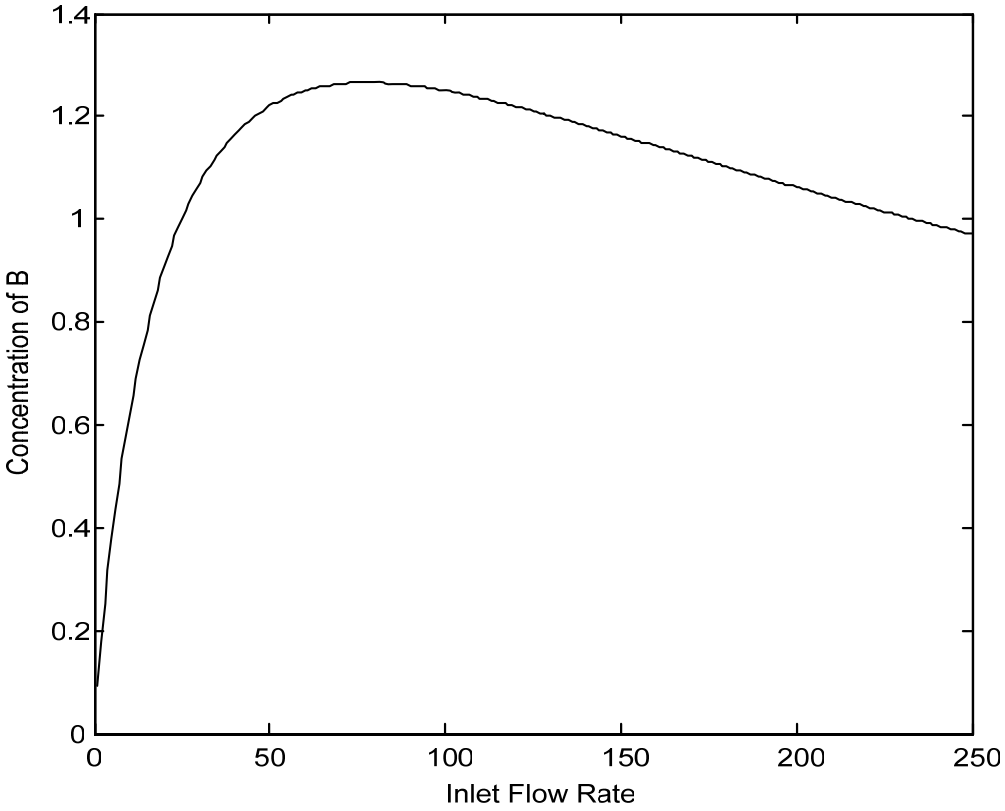

Figure 3.13 Operating locus of van de Vusse reactor

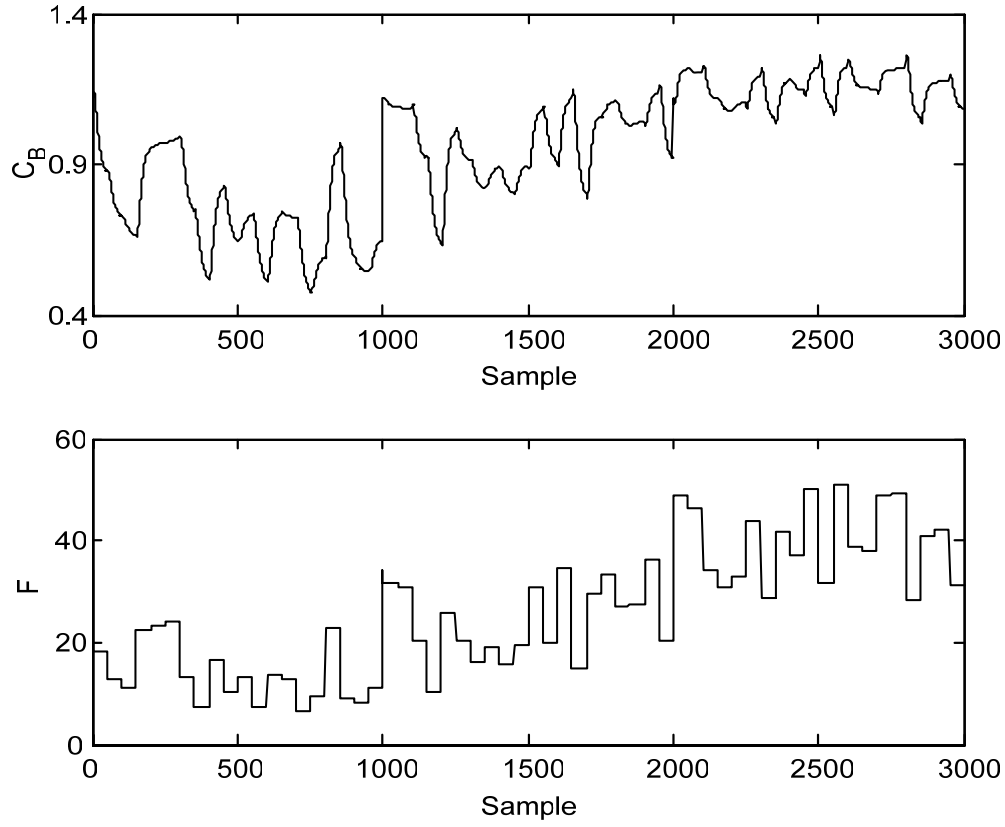

Figure 3.14 Input-output data used for constructing the database

By defining the normalized variables $\tilde{z}_1 = C_A - 3$, $\tilde{y} = \tilde{z}_2 = C_B - 1.12$, and $\tilde{u} = F - 34.3$, we obtain a model with state variables that are zero at the nominal operating condition. After taking Taylor series approximation of this normalized model up to first-order term, we obtain linear state-space representation of the nonlinear system in the form of Eqs. (3.18) and (3.19), where the matrices in those equations are given by:

$$\mathbf{A} = \begin{bmatrix} -144.3 & 0 \\ 50 & -134.3 \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} 7 \\ -1.12 \end{bmatrix},$$

$$\mathbf{c} = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

Therefore, transfer function model can be derived as follows:

$$L(s) = \frac{-1.12(s - 168.2)}{(s + 134.3)(s + 144.3)} \tag{3.25}$$

For non-adaptive JITL algorithm, second-order ARX model is employed as local model i.e. the regression vector is chosen as $\mathbf{z}(k-1) = [\tilde{y}(k-1), \tilde{y}(k-2), \tilde{u}(k-1)]^{\mathrm{T}}$. The database is generated by introducing uniformly random steps with distribution of [6 51] in process input as displayed in Figure 3.14. The JITL algorithm parameters, $k_{\min} = 20$, $k_{\max} = 70$, and $\gamma = 0.9$, are chosen to result in the minimum MSE in the validation test. The predictive performances of JITL algorithm and linear model are compared by introducing step changes of +15 and -20 in *F* from its nominal value as shown in Figure 3.15. Although linear model can display the inverse response associated with the non-minimum phase dynamics and the correct sign of the process gain, its prediction is inferior to that obtained by JITL algorithm, whose prediction resembles closely to the

actual process output and as a result their respective curves are indistinguishable in Figure 3.15.

To design two IMC controllers, the IMC controller Q is designed as

$$Q(s) = \frac{(s+134.3)(s+144.3)}{1.12(s+168.2)(0.01s+1)} \tag{3.26}$$

and NLIMC consists of the identical Q and the second filter $F_N(s)$ is chosen as

$$F_N(s) = \frac{0.01s+1}{0.003s+1} \tag{3.27}$$

Next, servo performances of two IMC designs are compared for +0.13 and -0.5 step changes in setpoint as shown in Figure 3.16. It is clear that NLIMC is able to follow the reference trajectory more closely than that obtained by the IMC controller. Disturbance rejection capability is evaluated by introducing $\pm 10\%$ step disturbances in C_{Af}. The closed-loop responses at two different operating points are shown in Figures 3.17 and 3.18. It is evident that performance obtained by the proposed control strategy is better than that of linear IMC. These closed-loop results are also supported by Table 3.5, which gives quantitative summary in terms of MSEs between the process output and the reference trajectory. It is evident that NLIMC scheme has reduced the MSE significantly, relative to linear IMC scheme, by a margin of approximately 36-60%.

Next, NLIMC scheme by using adaptive JITL algorithm is studied. In doing so, the initial database is generated by introducing uniformly random steps in process input around its nominal value as shown in Figure 3.19. Again, second-order ARX model is chosen as local model and the same parameter values of $k_{\min}$, $k_{\max}$, and γ are employed. In addition, linear model and two filters used in NLIMC scheme are the same as those chosen for non-adaptive case. The criterion employed to update the

database at each sampling instant is to check whether the difference between the predicted output by JITL algorithm and actual process output is within $\pm 5\%$ of the process output. The resulting closed-loop responses are compared in Figures 3.20 to 3.22. It is clear that proposed control strategy with adaptive JITL algorithm outperforms liner IMC, as also evidenced by the reduction of MSE as summarized in Table 3.6. Again, the performance of proposed control strategy with adaptive and non-adaptive JITL algorithm is almost identical.

Table 3.5 Comparison of closed-loop performances between IMC and non-adaptive NLIMC

Step change	Tracking error (MSE)		% Decrease in MSE
	IMC	NLIMC	
r = 1.12 to 1.25	2.1585×10^{-4}	1.3801×10^{-4}	36.06
r = 1.12 to 0.62	5.8165×10^{-3}	2.2753×10^{-3}	60.88
+10% change in C_{Af} at 1.12	6.8125×10^{-4}	3.7961×10^{-4}	44.28
+10% change in C_{Af} at 0.62	2.9791×10^{-5}	1.2751×10^{-5}	57.20
-10% change in C_{Af} at 1.12	6.2994×10^{-4}	4.0146×10^{-4}	36.27
-10% change in C_{Af} at 0.62	4.5846×10^{-5}	2.0940×10^{-5}	54.33

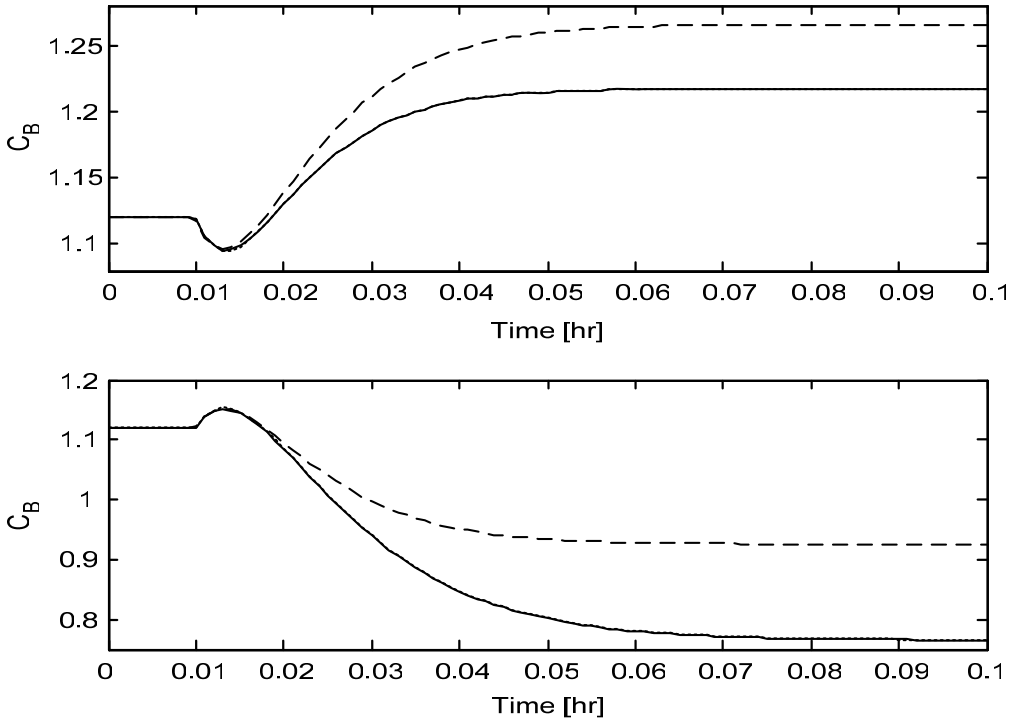

Figure 3.15 Open-loop responses for step changes of +15 (top) and -20 (bottom) in *F*. Solid: actual process; dashed: linear model; dotted: JITL

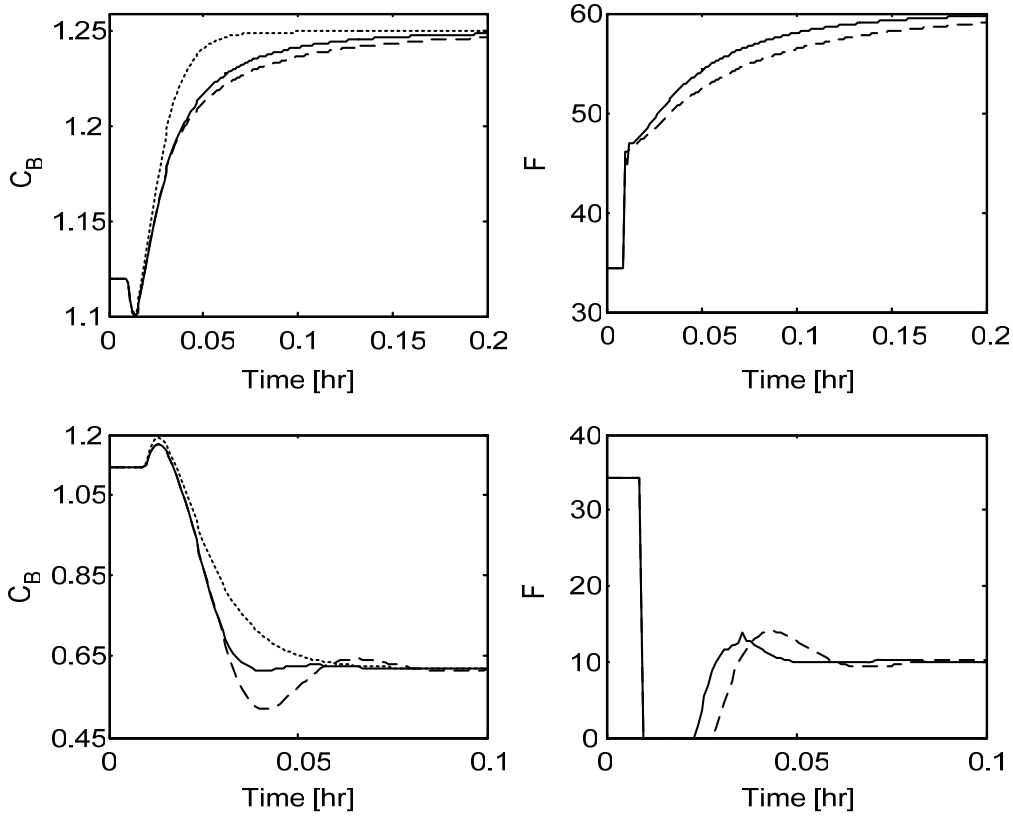

Figure 3.16 Closed-loop responses for step changes of +0.13 (top) and -0.5 (bottom) in setpoint. Dotted: reference trajectory; dashed: IMC; solid: NLIMC

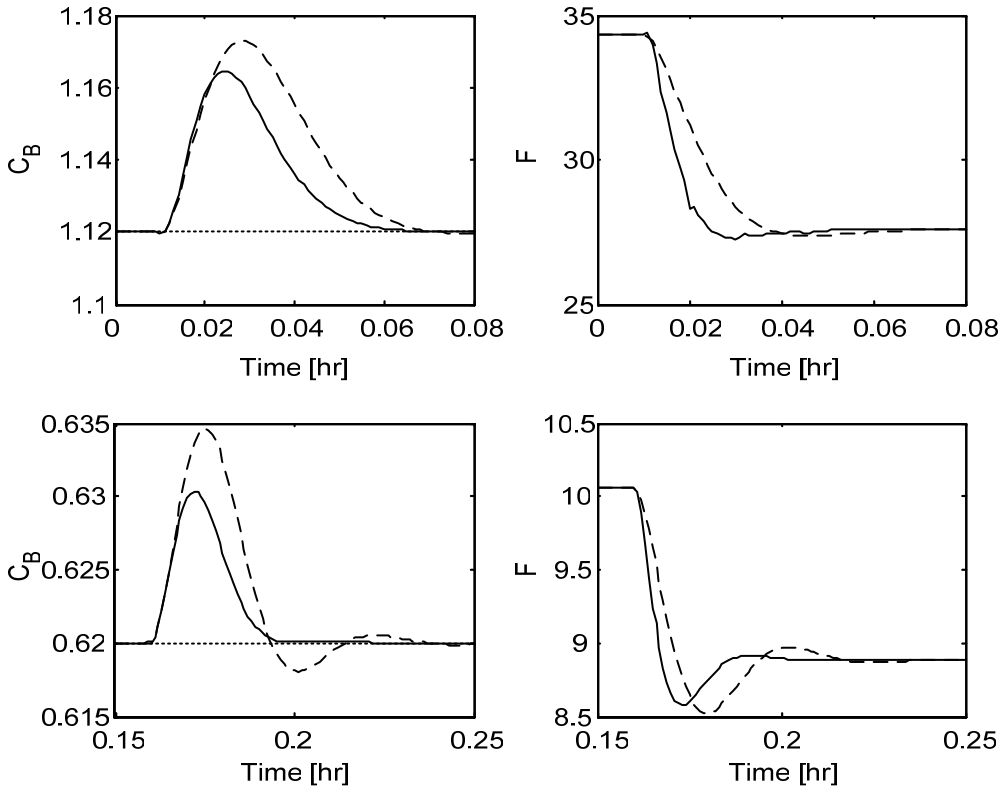

Figure 3.17 Closed-loop responses for +10% step change in C_{Af} . Dotted: reference trajectory; dashed: IMC; solid: NLIMC

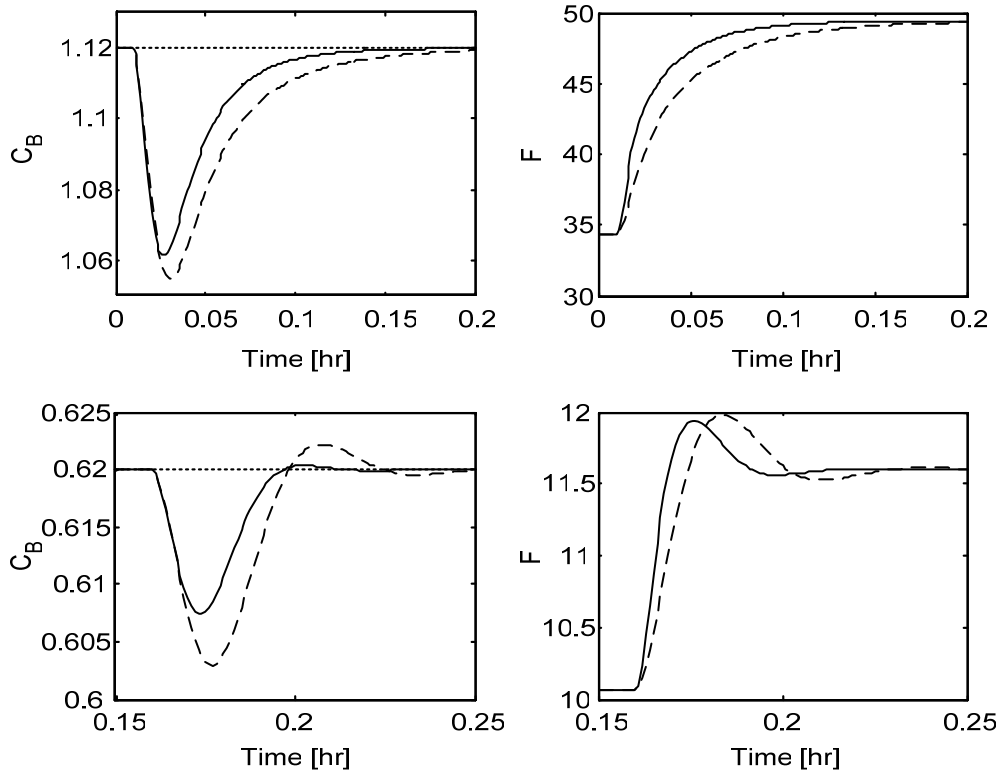

Figure 3.18 Closed-loop responses for -10% step change in C_{Af} . Dotted: reference trajectory; dashed: IMC; solid: NLIMC

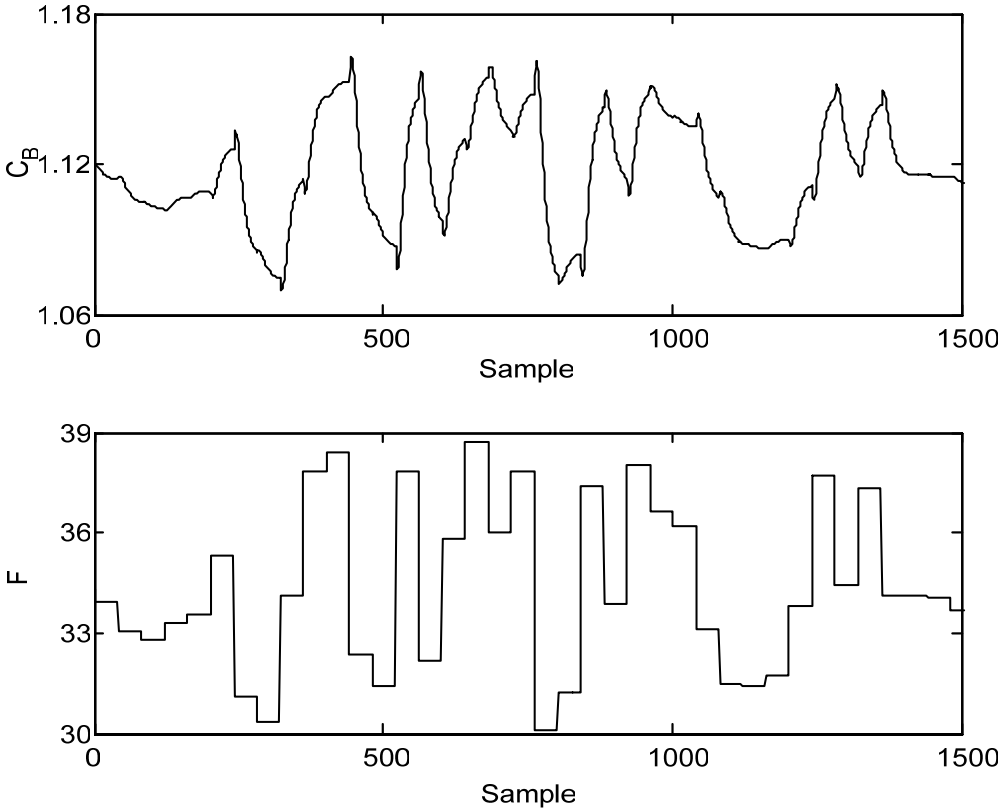

Figure 3.19 Input-output data used for constructing the initial database

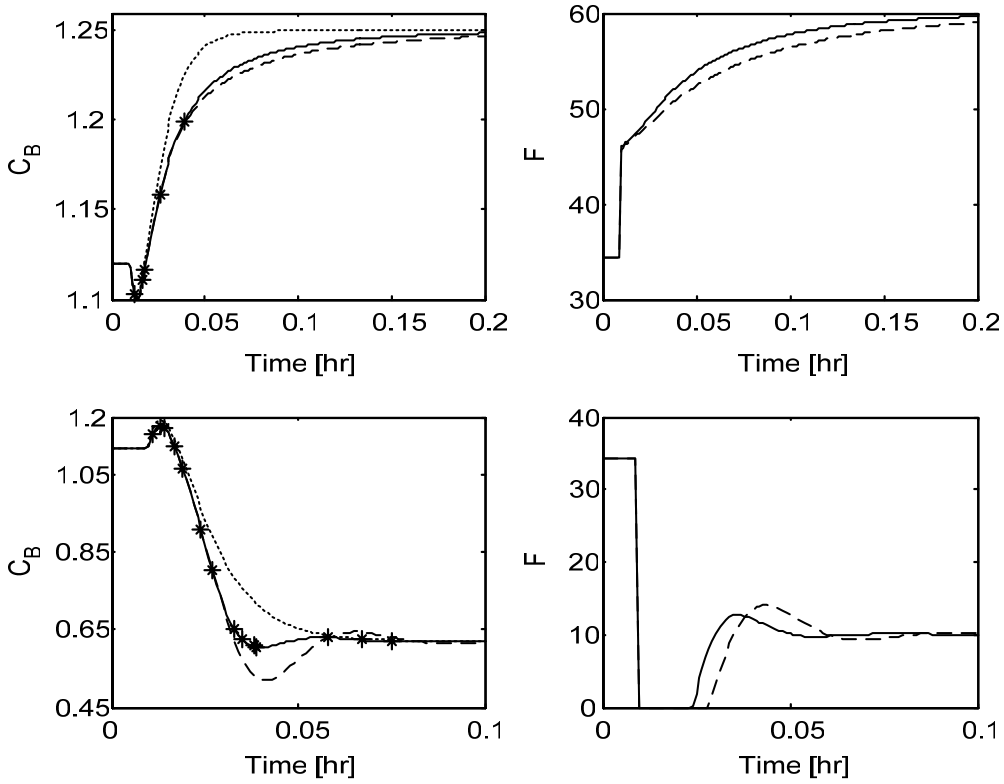

Figure 3.20 Closed-loop responses for step changes of +0.13 (top) and -0.5 (bottom) in setpoint. Dotted: reference; dashed: IMC; solid: NLIMC; star: database update

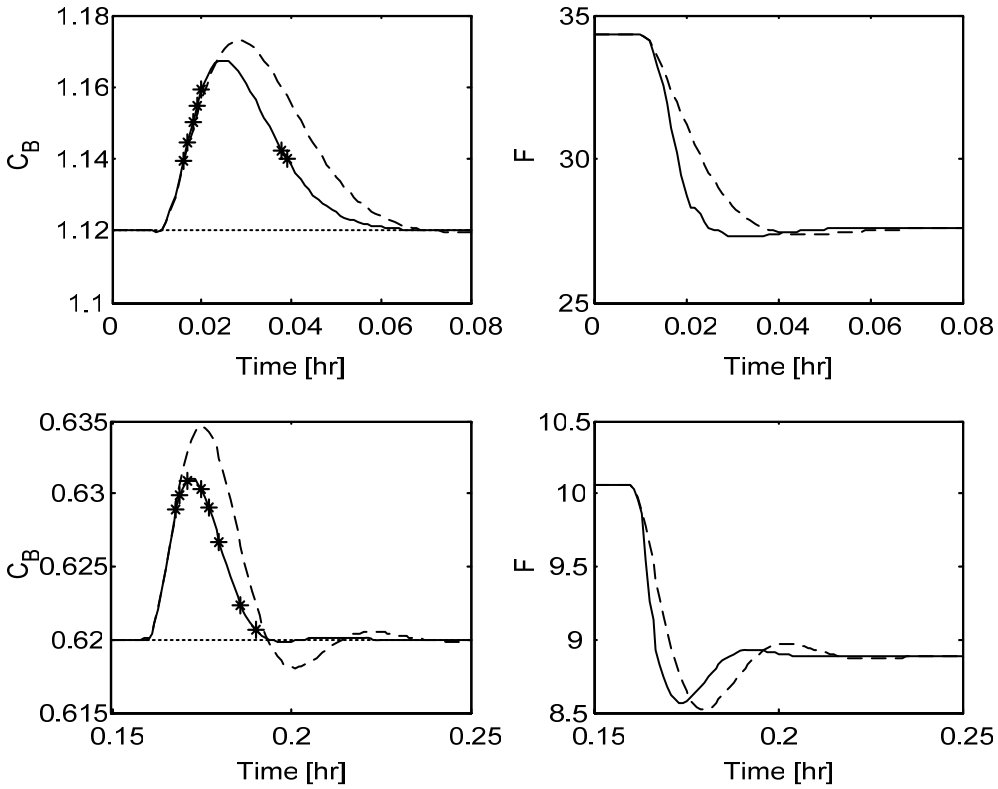

Figure 3.21 Closed-loop responses for $+10\%$ step change in C_{Af} . Dotted: reference trajectory; dashed: IMC; solid: NLIMC; star: database update

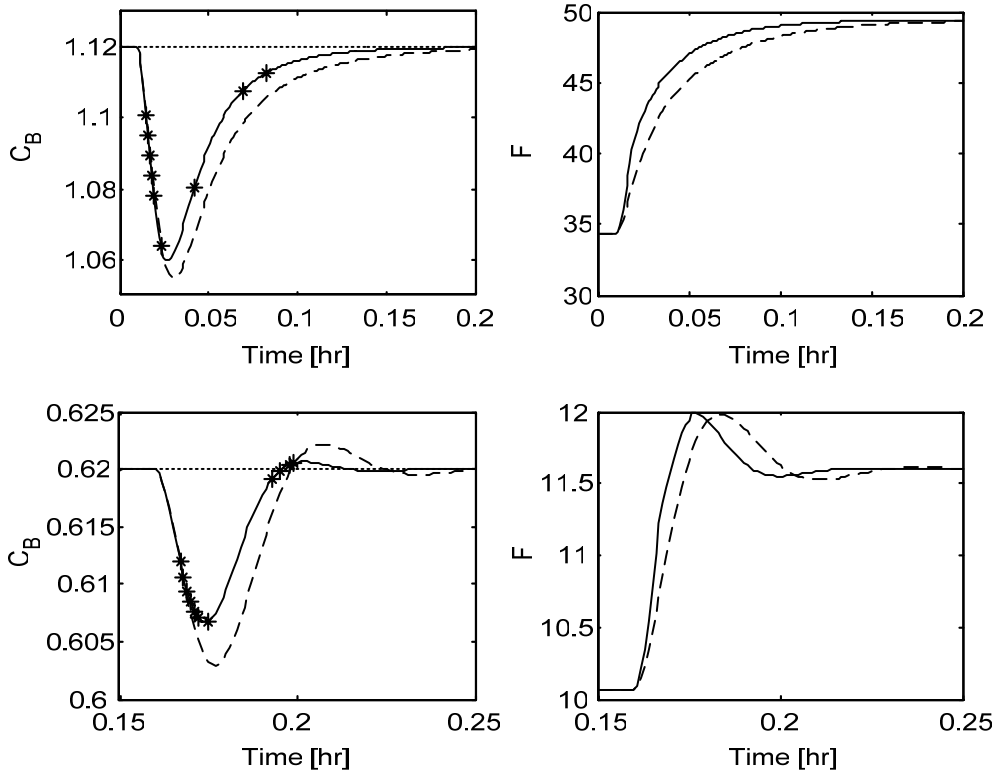

Figure 3.22 Closed-loop responses for -10% step change in C_{Af} . Dotted: reference trajectory; dashed: IMC; solid: NLIMC; star: database update

Table 3.6 Comparison of closed-loop performances between IMC and adaptive NLIMC

Step change	Tracking error (MSE)		% Decrease in MSE	No. of data points added
	IMC	NLIMC		
r = 1.12 to 1.25	2.1585×10^{-4}	1.4953×10^{-4}	30.73	5
r = 1.12 to 0.62	5.8165×10^{-3}	2.6110×10^{-3}	55.11	14
+10% in C_{Af} at 1.12	6.8125×10^{-4}	4.2833×10^{-4}	37.13	7
+10% in C_{Af} at 0.62	2.9791×10^{-5}	1.4589×10^{-5}	51.03	8
-10% in C_{Af} at 1.12	6.2994×10^{-4}	4.3535×10^{-4}	30.89	9
-10% in C_{Af} at 0.62	4.5846×10^{-5}	2.4020×10^{-5}	47.61	11

3.3 Conclusions

A nonlinear IMC design strategy using JITL technique is proposed for a class of nonlinear SISO systems that operate over a wide range of operating regimes. This IMC strategy makes use of conventional linear IMC controller augmented by an auxiliary loop to account for nonlinearities in the system. Simulation results show that proposed control strategy tracks reference trajectory better than its conventional counterpart. It is also shown that JITL model in the proposed control strategy can be made adaptive on-line readily by simply adding the new process data to the database. This adaptive feature of JITL algorithm makes JITL a better candidate than the previously proposed Volterra, Functional Expansion and NN models in the partitioned model inverse based nonlinear IMC scheme.

CHAPTER 4

Nonlinear Internal Model Control Design for MIMO Systems

4.1 Introduction

Decentralized control remains popular in the industry despite the recent developments of advanced controller synthesis procedures leading to full multivariable controllers. The block diagonal structure in decentralized control system invariably leads to performance deterioration when compared to the system under control by a full controller. This sacrifice has to be weighed against the following two factors (Grosdidier and Morari, 1986):

1. Hardware simplicity: If u_i and y_i in Figure 2.2 are physically close but u_i and $y_j (i \neq j)$ are far apart, a full controller could require expensive communication links. Also, the controller hardware costs could be high if an implementation through analogue circuitry is required.

2. Design simplicity: If the subsystem $g_{ij}(s) = 0 (i \neq j)$ in Figure 2.2, then each controller $c_i(s)$ can be designed for the isolated subsystem $g_{ii}(s)$ without any loss of performance. If $g_{ij}(s)(i \neq j)$ is "small", then it should still be possible to design the

controller for the essentially independent subsystems $g_{ii}(s)$ to achieve satisfactory performance. The advantage is that fewer controller parameters need to be chosen than for the full multivariable control system. This is particularly relevant in process control where often many process variables have to be controlled which could lead to an enormously complex controller.

In literature, PID loops have been used in decentralized control scheme. However, as system dimensions increase, the tuning of PID parameters is not trivial task because each PID loop has three parameters. To overcome this drawback, an alternative design approach is proposed by Economou and Morari (1986) within the IMC framework. The resulting decentralized IMC structure is postulated by designing the decentralized IMC controller $\mathbf{Q}(s)$ with respect to the process model $\tilde{\mathbf{G}}(s)$, which is essentially diagonal matrix with entries being the diagonal elements of the process $\mathbf{G}(s)$. Figure 4.1 is the block diagram representation of this structure for $n \times n$ multivariable systems.

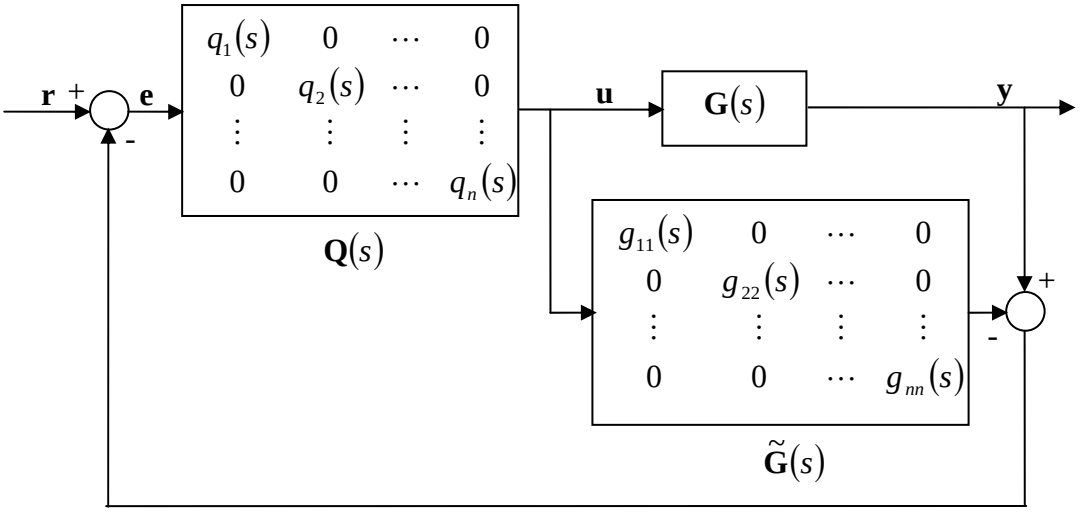

Figure 4.1 Decentralized IMC structure

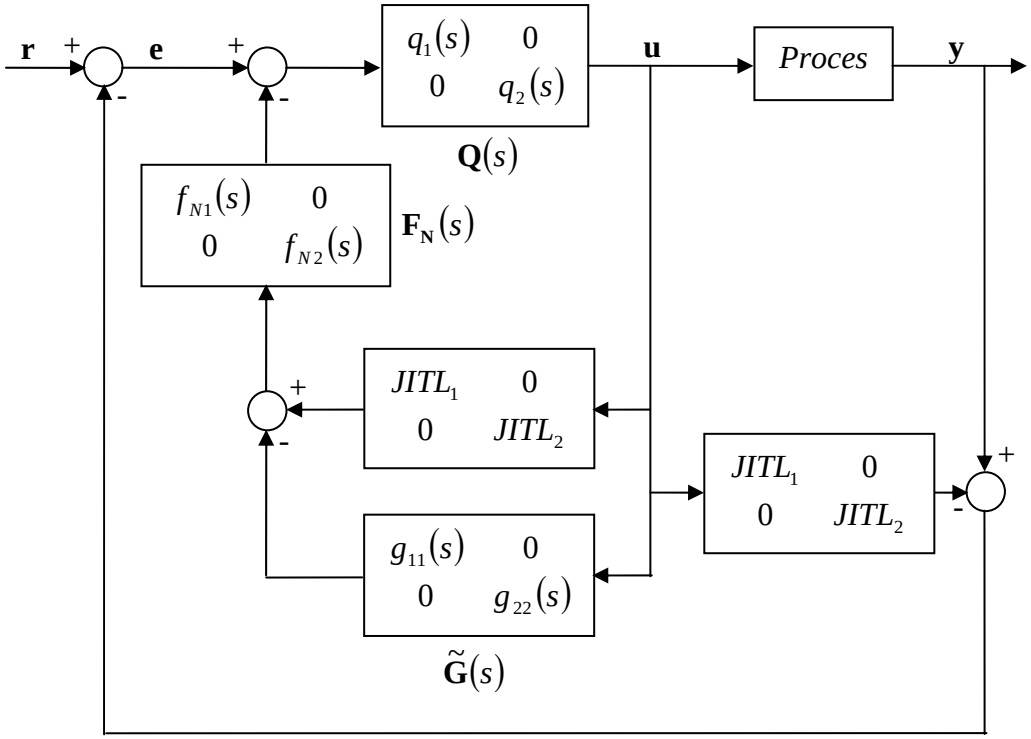

Figure 4.2 Decentralized NLIMC structure

The IMC controller design for each loop in multiloop environment is similar to SISO IMC controller design and is briefly described as follows: Assume that a particular control structure has been selected which assigns each system input to one output. Furthermore, the inputs and outputs have been renumbered so that the corresponding transfer function elements appear on the diagonal of the system transfer matrix (Economou and Morari, 1986b).

Let $\mathbf{G}(s)$ be the transfer function matrix for a $n \times n$ multivariable system:

$$\mathbf{G}(s) = \begin{bmatrix} g_{11}(s) & g_{12}(s) & \cdots & g_{1n}(s) \\ g_{21}(s) & g_{22}(s) & \cdots & g_{2n}(s) \\ \vdots & \vdots & \vdots & \vdots \\ g_{n1}(s) & g_{n2}(s) & \cdots & g_{nn}(s) \end{bmatrix} \tag{4.1}$$

Then the decentralized IMC model is

$$\tilde{\mathbf{G}}(s) = \text{diag}[g_{11}(s), g_{22}(s), \ldots, g_{nn}(s)] \tag{4.2}$$

and the decentralized IMC controller is

$$\mathbf{Q}(s) = \text{diag}[q_1(s), q_2(s), \ldots, q_n(s)] \tag{4.3}$$

with

$$q_i = g_{ii-}^{-1}(s) f_{Li}(s), \qquad i = 1, 2, \ldots, n \tag{4.4}$$

It is important to note that in the transition from the system $\mathbf{G}(s)$ to the model $\tilde{\mathbf{G}}(s)$, the off-diagonal elements were dropped. As a consequence, an a priori exactly known "modeling error" is introduced. The IMC filter $f_{Li}(s)$ has to be designed to preserve the robust stability in the presence of this modeling error.

Economou and Morari (1986b) utilized linear models in their decentralized IMC structure, which is able to perform satisfactory in the small neighborhood of the operating point where the linear model is obtained. When the plant is to be operated in a range of operating conditions in consequence of large setpoint changes and/or the presence of disturbances, there is need of decentralized IMC controller designed based on nonlinear models. Therefore, the objective of this chapter is to extend the nonlinear IMC design strategy developed in Chapter 3 to MIMO systems that operate over a range of operating regimes. Like in SISO case, JITL algorithm is used for modeling purpose. Two literature examples are used to illustrate the proposed control strategy and a comparison with the conventional decentralized linear IMC is made.

4.2 Decentralized Nonlinear IMC Strategy

For convenience sake, the proposed decentralized nonlinear IMC (decentralized NLIMC) design for 2×2 systems is discussed in this section. Figure 4.2 illustrates the decentralized NLIMC structure, where $g_{11}(s)$ and $g_{22}(s)$ are the respective linear models of the subsystems (corresponding to the input-output pairing chosen) obtained around an operating point, whereas $JITL_1$ and $JITL_2$ are the

corresponding nonlinear models for the same subsystems. The IMC controllers $q_1(s)$ and $q_2(s)$ are designed according to Eq. (4.4) and filters $f_{N1}(s)$ and $f_{N2}(s)$ are chosen for each loop in similar way as SISO case to provide robustness. Note that the issue of control structure selection, i.e. input-output pairing, is tackled by conventional pairing criterion like Relative Gain Array (RGA).

4.3 Examples

Example 1: The proposed decentralized NLIMC strategy is applied to the free-radical solution polymerization of styrene in a jacketed CSTR. A schematic of this process is shown in Figure 4.3. The CSTR has three feed streams - pure styrene monomer, azo-bis-isobutyronitrile (AIBN) initiator dissolved in benzene and pure benzene as solvent. As the control objective, we are interested in controlling the number-average molecular weight (NAMW) as well as the reactor temperature (T) by manipulating both the initiator (Q_i) and cooling water flow rates (Q_c). The six-state nonlinear model for this reactor can be written as follows (Maner et al., 1996):

$$\frac{d[I]}{dt} = \frac{Q_i[I_f] - Q_t[I]}{V} - k_d[I], \tag{4.5}$$

$$\frac{d[M]}{dt} = \frac{Q_m[M_f] - Q_t[M]}{V} - k_p[M][P], \tag{4.6}$$

$$\frac{dT}{dt} = \frac{Q_t(T_f - T)}{V} + \frac{(-\Delta H_r)}{\rho C_p} k_p[M][P] - \frac{hA}{\rho C_p V}(T - T_c), \tag{4.7}$$

$$\frac{dT_c}{dt} = \frac{Q_c(T_{cf} - T_c)}{V_c} + \frac{hA}{\rho_c C_{pc} V_c}(T - T_c), \tag{4.8}$$

$$\frac{dD_0}{dt} = 0.5 k_t[P]^2 - \frac{Q_t D_0}{V}, \tag{4.9}$$

$$\frac{dD_1}{dt} = M_m k_p[M][P] - \frac{Q_t D_1}{V}, \tag{4.10}$$

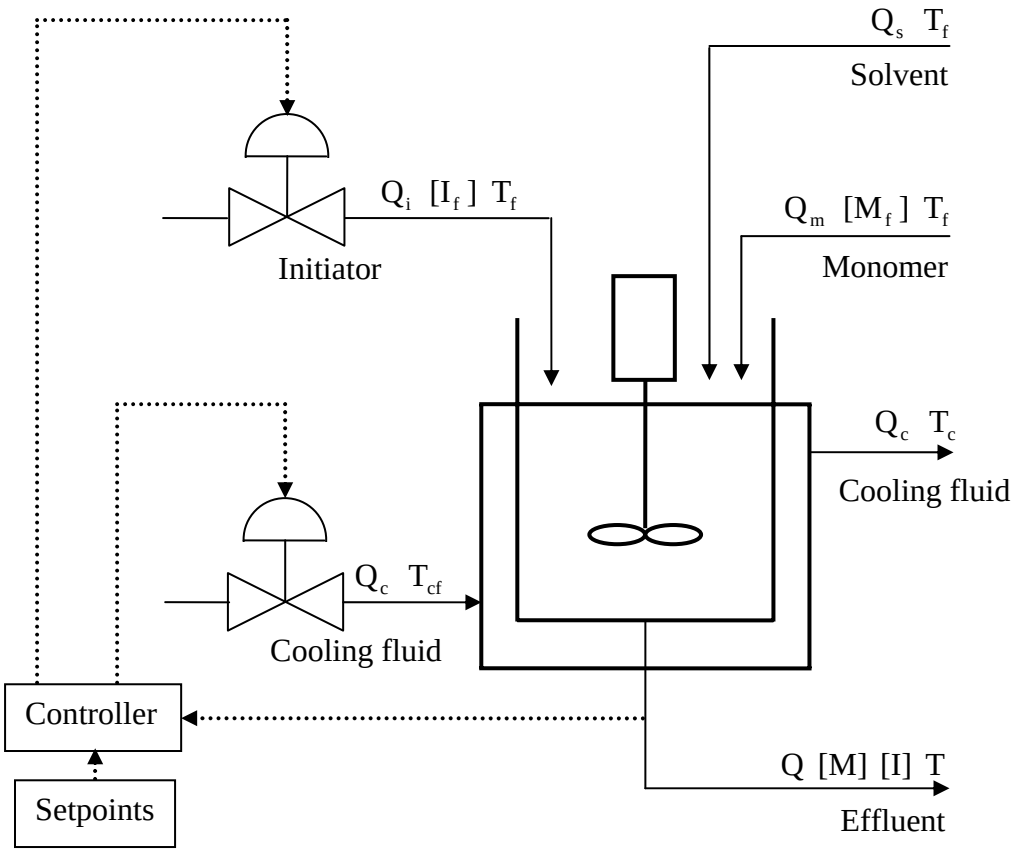

Figure 4.3 Control configuration for MIMO polymerization reactor

where $\text{NAMW} = \dfrac{D_1}{D_0}$, and

$$[P] = \left[\frac{2fk_d[I]}{k_t}\right]^{0.5},$$

$$k_i = A_i \exp(-E_i / T), \qquad i = d, p, t$$

$$Q_t = Q_i + Q_s + Q_m .$$

The model parameters and nominal operating conditions for this reactor are given in Tables 4.1 and 4.2 respectively. The sampling time for this example is 1 h and in the following simulation studies, the step change in the process input (open-loop test) or setpoint is made at the time equal to 10 h. In addition, we shall denote two process outputs as $y_1 = \text{NAMW}$ and $y_2 = T$, while two manipulated variables as

$u_1 = Q_i$ and $u_2 = Q_c$. Also, the following dimensionless variables are introduced: $\tilde{y}_i = (y_i - y_{i0})/y_{i0}$ and $\tilde{u}_i = (u_i - u_{i0})/u_{i0}$, where y_{i0} and u_{i0} are nominal operating values for the corresponding process variables.

Table 4.1 Model parameters for 2×2 polymerization reactor

$f = 0.6$	$\rho_c C_{pc} = 966.3 \,\text{cal K}^{-1}\,\text{l}^{-1}$
$A_d = 2.142 \times 10^{17}\,\text{l h}^{-1}$	$Q_s = 459 \,\text{l h}^{-1}$
$E_d = 14897 \,\text{K}$	$Q_m = 378 \,\text{l h}^{-1}$
$A_t = 4.5 \times 10^{12}\,\text{l mol}^{-1}\,\text{h}^{-1}$	$V = 3000 \,\text{L}$
$E_t = 843 \,\text{K}$	$V_c = 3312.4 \,\text{L}$
$A_p = 3.816 \times 10^{10}\,\text{l mol}^{-1}\,\text{h}^{-1}$	$[I_f] = 0.5888 \,\text{mol l}^{-1}$
$E_p = 3557 \,\text{K}$	$[M_f] = 8.6981 \,\text{mol l}^{-1}$
$-\Delta H_r = 16700 \,\text{cal mol}^{-1}$	$T_f = 330 \,\text{K}$
$hA = 2.52 \times 10^5\,\text{cal K}^{-1}\,\text{h}^{-1}$	$T_{cf} = 295 \,\text{K}$
$\rho C_p = 360 \,\text{cal K}^{-1}\,\text{l}^{-1}$	$M_m = 104.14 \,\text{g mol}^{-1}$

Table 4.2 Nominal operating conditions for 2×2 polymerization reactor

$[I] = 6.6832 \times 10^{-2}\,\text{mol l}^{-1}$	$D_1 = 16.110 \,\text{g l}^{-1}$
$[M] = 3.3245 \,\text{mol l}^{-1}$	$Q_i = 108 \,\text{l h}^{-1}$
$T = 323.56 \,\text{K}$	$Q_c = 471.6 \,\text{l h}^{-1}$
$T_c = 305.17 \,\text{K}$	$y_1 = 58481 \,\text{g mol}^{-1}$
$D_0 = 2.7547 \times 10^{-4}\,\text{mol l}^{-1}$	$y_2 = 323.56 \,\text{K}$

Introducing the model parameters in Eqs. (4.5) to (4.10), linear model was obtained for normalized variables after taking Taylor series approximation at nominal operating point. RGA analysis is conducted using steady-state information of this linear model. Based on RGA criteria, NAMW is controlled by Q_i and T is controlled by Q_c in the decentralized controller design.

In the decentralized IMC scheme, IMC model is obtained by:

$$\tilde{\mathbf{G}}(s)=\begin{bmatrix} g_{11}(s) & 0 \\ 0 & g_{22}(s) \end{bmatrix} \tag{4.11}$$

where

$$g_{11}(s)=\frac{-0.05s^4-0.065s^3-0.0302s^2-0.006s-0.00041}{s^6+1.818s^5+1.358s^4+0.5318s^3+0.1146s^2+0.01278s+0.00057}$$

$$g_{22}(s)=\frac{-0.0011s^2-0.0007s-0.000111}{s^4+1.188s^3+0.5104s^2+0.0924s+0.00573}$$

To proceed with JITL modeling, the following regression vectors are chosen:

$$y_1:\mathbf{z}_1(k-1)=\left[\tilde{y}_1(k-1),\tilde{y}_1(k-2),\tilde{u}_1(k-1)\right]^{\mathrm{T}}, \tag{4.12}$$

$$y_2:\mathbf{z}_2(k-1)=\left[\tilde{y}_2(k-1),\tilde{y}_2(k-2),\tilde{u}_1(k-1),\tilde{u}_2(k-1)\right]^{\mathrm{T}}. \tag{4.13}$$

The databases are generated by introducing uniformly random steps with distribution of [32 208] and [304 1090] in process inputs u_1 and u_2 respectively, as shown in Figure 4.4. Then two separate databases are constructed using this data for JITL algorithm to predict y_1 and y_2. The JITL algorithm parameters used are: $k_{min}=15$, $k_{\max}=60$, and $\gamma=0.9$ are chosen to predict y_1, whereas $k_{min}=20$, $k_{\max}=80$, and $\gamma=0.85$ are used to predict y_2. The nonlinear behavior of this reactor can be observed from open-loop responses for step changes of ± 27 l/h in Q_i from its nominal value of 108 l/h shown in Figure 4.5. For number-average molecular weight, the predicted output of the JITL tracks the actual nonlinear plant output very closely

and as a result their responses are indistinguishable in both simulations. In comparison, it is evident that linear model given in Eq. (4.11) fails to provide accurate prediction of reactor dynamics in the aforementioned open-loop tests. Similar observation also applies to the second output, temperature.

To design both decentralized IMC controller $\mathbf{Q}(s)$ and NLIMC controller, the former is chosen as the inverse of linear model $\tilde{\mathbf{G}}(s)$ given in Eq. (4.11) with augmentation of the following filter $\mathbf{F}_{\mathbf{L}}(s)$:

$$\mathbf{F}_{\mathbf{L}}(s) = \begin{bmatrix} \dfrac{1}{(3s+1)^2} & 0 \\ 0 & \dfrac{1}{(3s+1)^2} \end{bmatrix} \tag{4.14}$$

and decentralized NLIMC scheme consists of identical $\mathbf{Q}(s)$ and the second filter in auxiliary loop for nonlinear correction is chosen as

$$\mathbf{F}_{N}(s) = \begin{bmatrix} \dfrac{(3s+1)^2}{(1.3s+1)^2} & 0 \\ 0 & \dfrac{(3s+1)^2}{(1.3s+1)^2} \end{bmatrix} \tag{4.15}$$

To evaluate the performance of two decentralized IMC designs, step change in the setpoint of NAMW from 58481 to 80000 g/mol is conducted. It is clear from Figure 4.6 that decentralized NLIMC outperforms the decentralized linear IMC because the decentralized NLIMC effectively cancels process nonlinearity and tracks the reference trajectory more closely. Decentralized linear IMC may be detuned to obtain similar response. However, improved performance for positive stepoint changes would be achieved at the expense of performance deterioration for negative stepoint changes. Closed-loop simulation results for other setpoint changes are shown in Figures 4.7 to 4.9. It is evident that similar trends as discussed earlier are achieved. To evaluate the disturbance rejection performances of both control schemes,

unmeasured disturbances of $\pm 20\%$ step changes in the concentration of initiator in feed stream $([I_f])$ are considered. As can be seen from Figures 4.10 and 4.11, the decentralized NLIMC is able to reject the disturbances more effectively than linear IMC controller. Table 4.3 summarizes the MSE between the process output and the reference trajectory for the closed-loop responses considered above. It is evident that, decentralized NLIMC scheme has reduced the MSE significantly, relative to decentralized linear IMC scheme.

Table 4.3 Comparison of closed-loop performances between decentralized IMC and NLIMC

Step change	Tracking error (MSE)			
	Decentralized IMC		Decentralized NLIMC	
	y_1	y_2	y_1	y_2
r_1=58481 to 80000	1.1068×10^{6}	1.2130×10^{-1}	1.3385×10^{5}	2.9500×10^{-2}
r_1=58481 to 50000	3.7782×10^{5}	3.4100×10^{-2}	1.2519×10^{5}	9.6000×10^{-3}
r_2 =323.56 to 325	4.2385×10^{5}	8.9674×10^{-4}	1.1952×10^{5}	1.3096×10^{-4}
r_2 =323.56 to 320	1.3529×10^{6}	3.3650×10^{-1}	4.1268×10^{5}	1.2400×10^{-1}
+20% change in $[I_f]$	1.0931×10^{6}	4.8660×10^{-3}	2.9994×10^{5}	7.4339×10^{-4}
-20% change in $[I_f]$	1.7755×10^{6}	5.6411×10^{-3}	4.5902×10^{5}	8.5870×10^{-4}

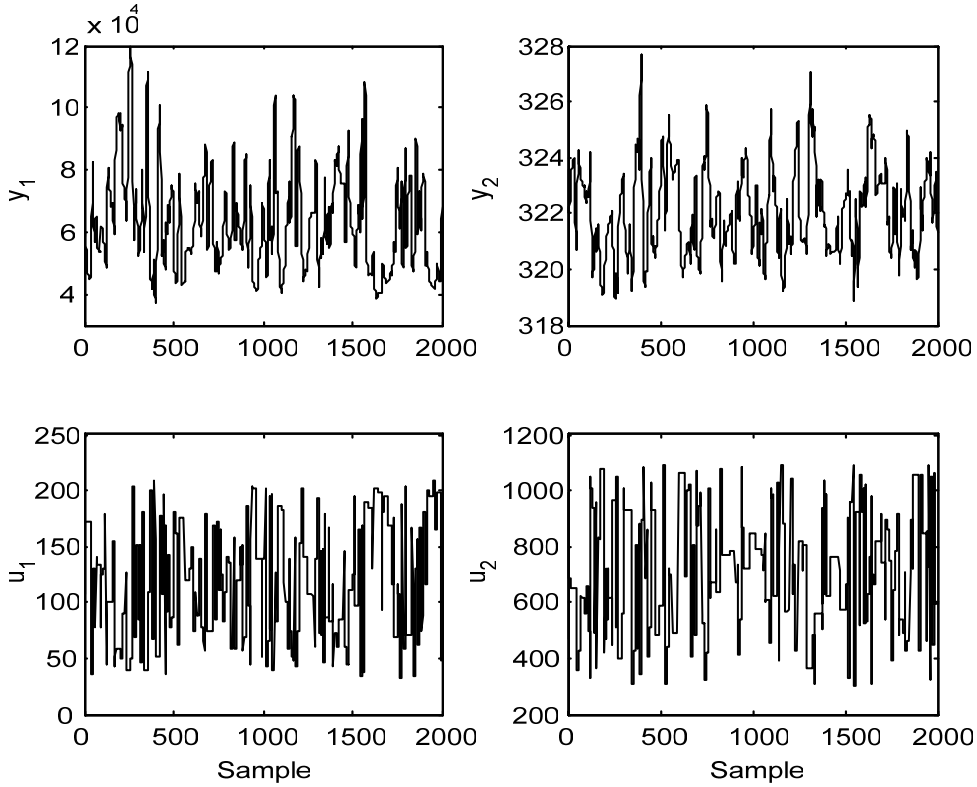

Figure 4.4 Input-output data used for constructing the database

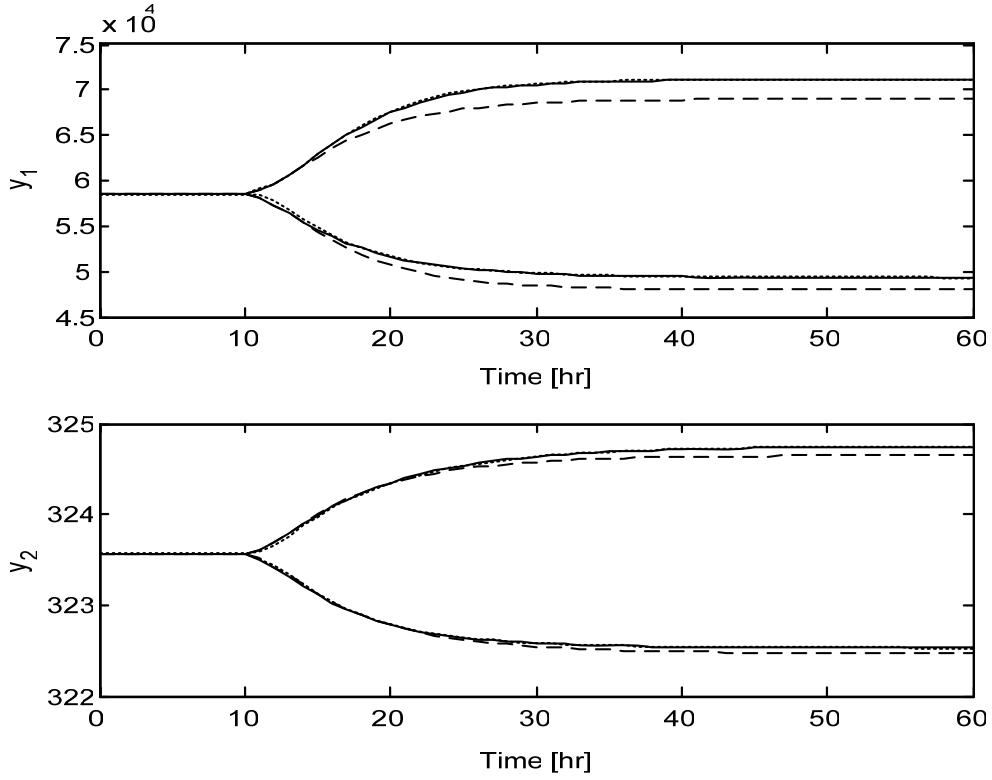

Figure 4.5 Open-loop responses for $\pm 25\%$ step changes in Q_i from its nominal value. Solid: actual process; dashed: linear model; dotted: JITL

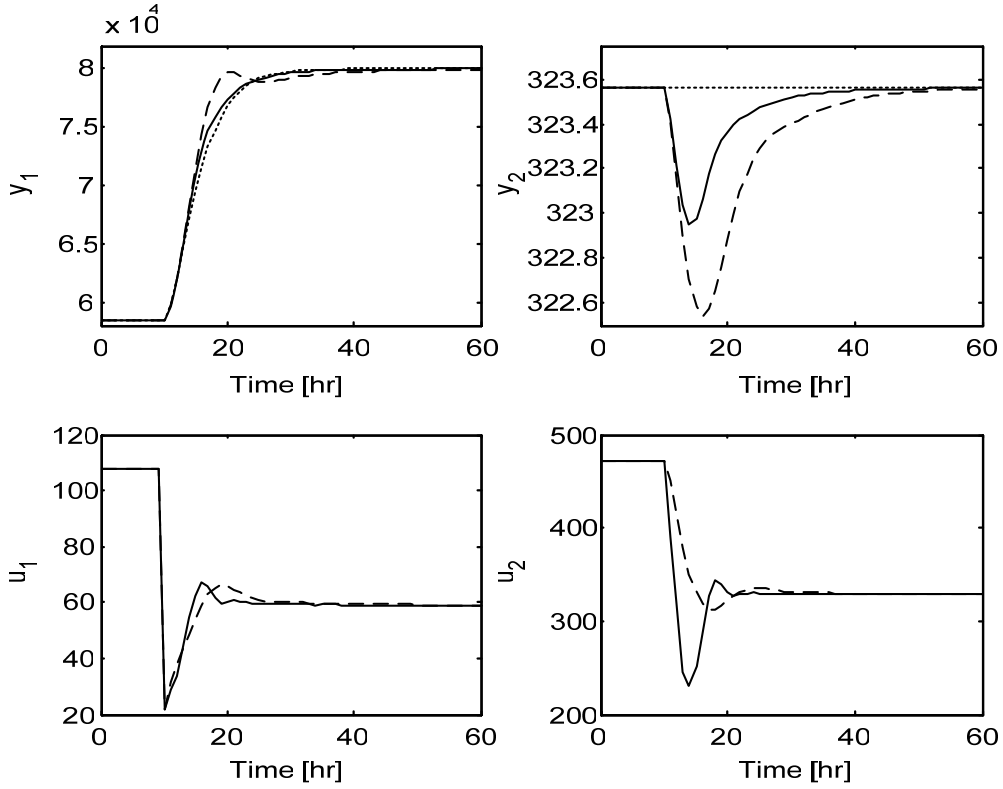

Figure 4.6 Closed-loop responses for setpoint change from 58481 to 80000 in y_1.
Dotted: reference trajectory; dashed: IMC; solid: NLIMC

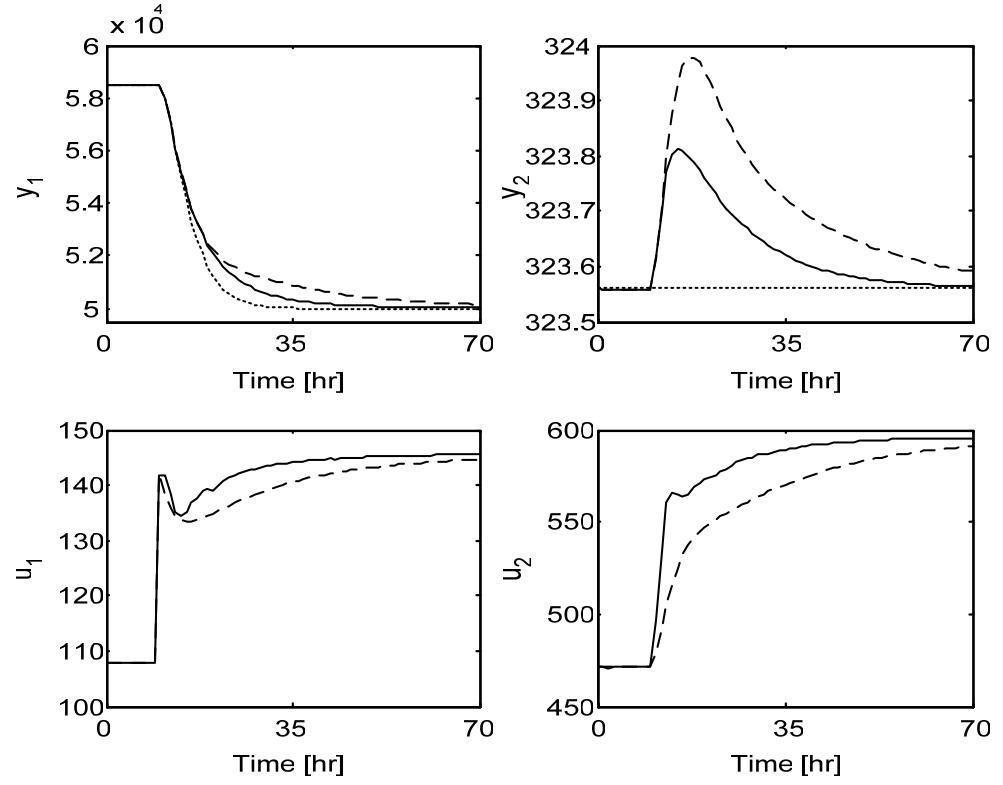

Figure 4.7 Closed-loop responses for setpoint change from 58481 to 50000 in y_1.
Dotted: reference trajectory; dashed: IMC; solid: NLIMC

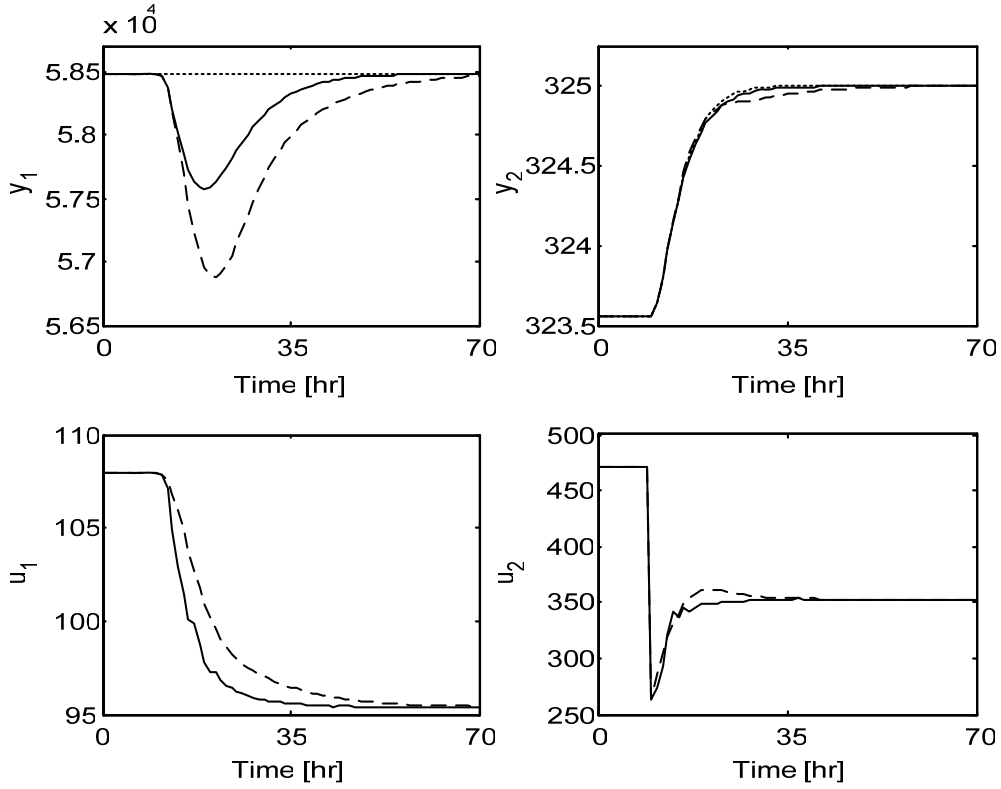

Figure 4.8 Closed-loop responses for setpoint change from 323.56 to 325 in y_2.
Dotted: reference trajectory; dashed: IMC; solid: NLIMC

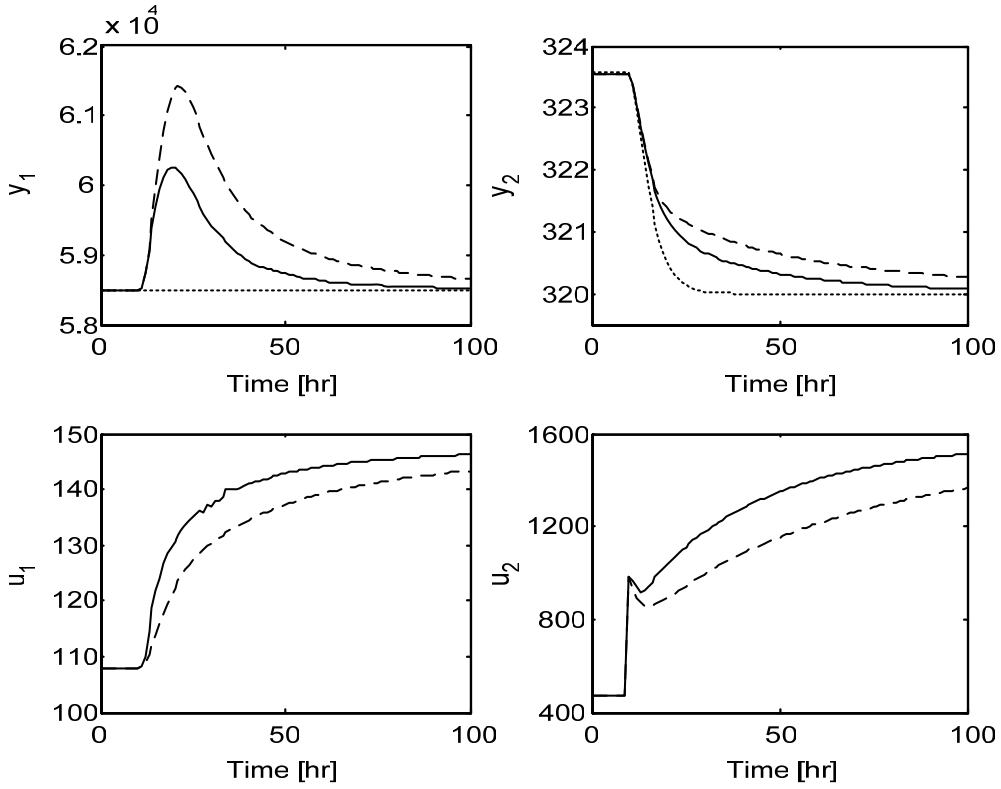

Figure 4.9 Closed-loop responses for setpoint change from 323.56 to 320 in y_2.
Dotted: reference trajectory; dashed: IMC; solid: NLIMC

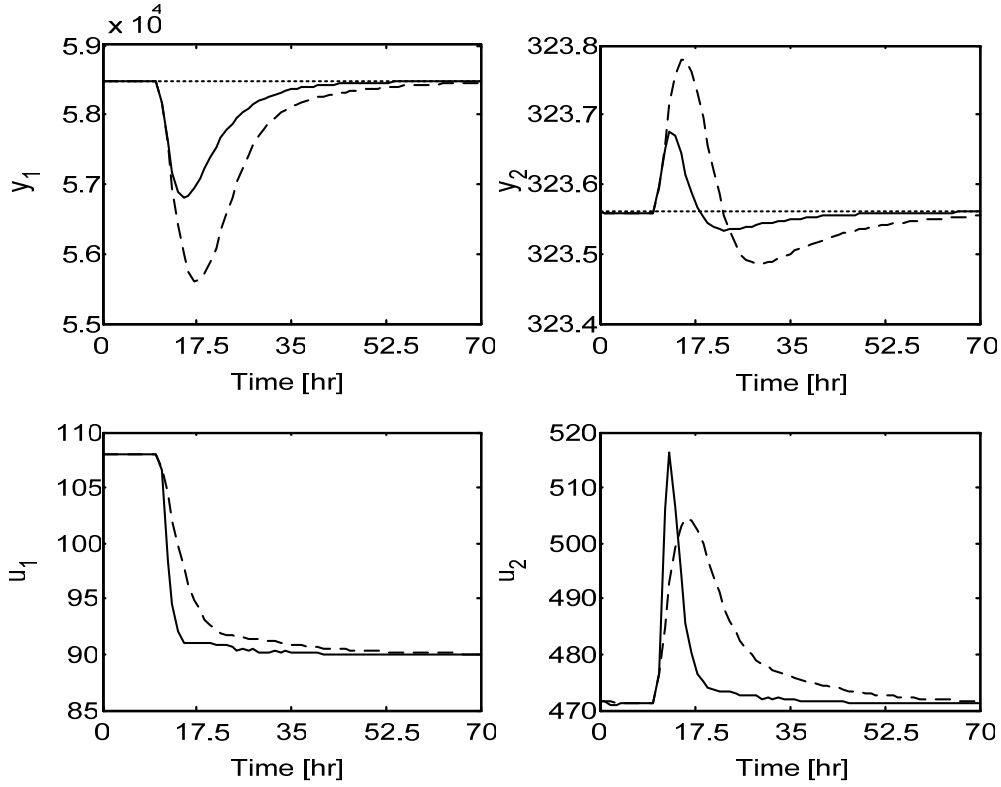

Figure 4.10 Closed-loop responses for $+20\%$ step change in $[I_f]$. Dotted: reference trajectory; dashed: IMC; solid: NLIMC

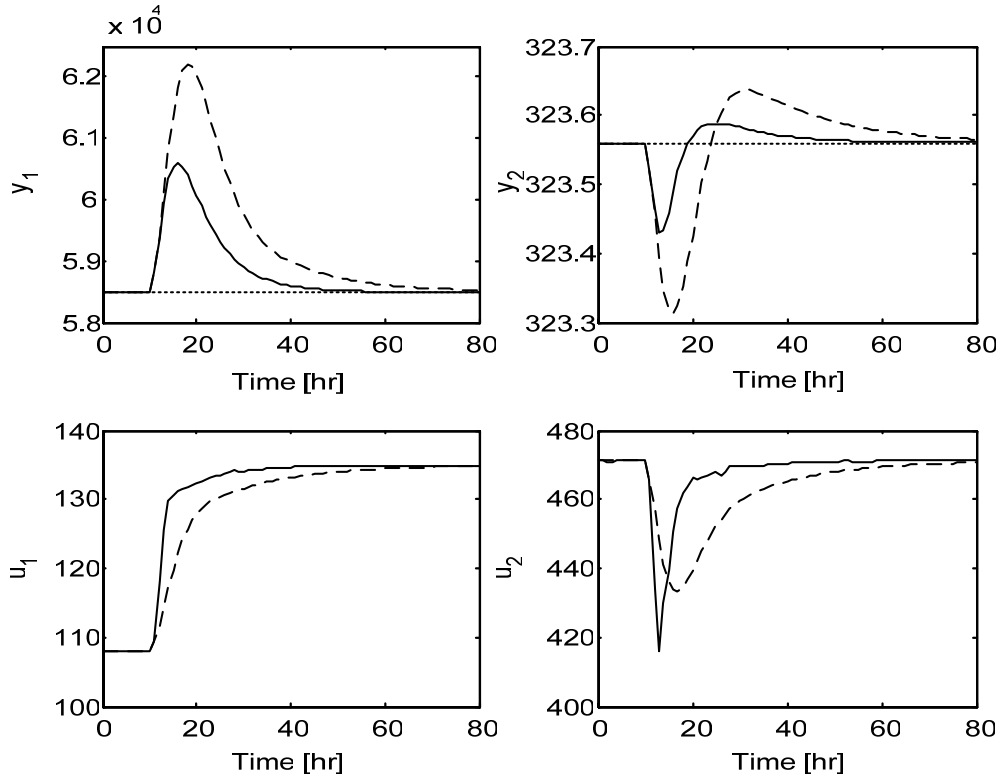

Figure 4.11 Closed-loop responses for -20% step change in $[I_f]$. Dotted: reference trajectory; dashed: IMC; solid: NLIMC

Example 2: The second application of proposed control strategy focuses on the control of a nonisothermal CSTR, with the kinetics governed by the van de Vusse reactions,

$$\mathrm{A} \xrightarrow{k_1} \mathrm{B} \xrightarrow{k_2} \mathrm{C},$$

$$2\mathrm{A} \xrightarrow{k_3} \mathrm{D}.$$

This reaction scheme describes the production of cyclopentenol (B) from cyclopentadiene (A) by acid catalyzed electrophilic addition of water in dilute solution. The unwanted by-products are dicyclopentadiene (D) and cyclopentanediol (C) (Harris and Palazoglu, 1998; Engell and Klatt, 1993). The above reaction takes place in a jacket-cooled, perfectly mixed CSTR, where the coolant is introduced by an external heat exchanger in which a certain amount of heat is removed. Here the control problem focuses on controlling the outlet concentration of B $\left(y_1 = c_B\right)$ as well as the reactor temperature $\left(y_2 = T\right)$ by manipulating both reactor flow rate $\left(u_1 = F\right)$ and heat exchanger duty $\left(u_2 = Q_w\right)$.

The balance equations for the concentrations of cyclopentadiene, c_A, and cyclopentenol, c_B, are:

$$\frac{dc_A}{dt} = \frac{F}{V}\left(c_{A0} - c_A\right) - k_1(T)c_A - k_3(T)c_A^2 \tag{4.16}$$

$$\frac{dc_B}{dt} = -\frac{F}{V}c_B + k_1(T)c_A - k_2(T)c_B \tag{4.17}$$

The energy balance for the reactor and external heat exchanger yields the following differential equations:

$$\begin{aligned}\frac{dT}{dt} = &-\frac{1}{\rho C_P}\left[k_1(T)c_A\Delta H_1 + k_2(T)c_B\Delta H_2 + k_3(T)c_A^2\Delta H_3\right] \\ &+ \frac{F}{V}\left(T_0 - T\right) + \frac{k_w A_w}{\rho C_p V}\left(T_w - T\right)\end{aligned} \tag{4.18}$$

$$\frac{dT_w}{dt} = \frac{1}{m_w C_{pw}}\left[Q_w + k_w A_w (T - T_w)\right] \tag{4.19}$$

In all of the above differential equations, the reaction rate coefficients k_1, k_2 and k_3 depend on the reactor temperature via Arrhenius's equation:

$$k_i(T) = k_{i,0}\exp\left[\frac{E_i}{T}\right] \tag{4.20}$$

The complete notation and relevant parameters are given in Table 4.4 and nominal operating conditions are in Table 4.5. For this example, the sampling time is 0.002 h and in the following simulation studies, the step change in the process input (open-loop test) or setpoint is made at the time equal to 0.2 h. In addition, the following dimensionless variables are defined: $\tilde{y}_i = (y_i - y_{i0})/y_{i0}$ and $\tilde{u}_i = (u_i - u_{i0})/u_{i0}$, where y_{i0} and u_{i0} are nominal operating values for the corresponding process variables.

As described in the previous example, linear model was obtained by Taylor series approximation and RGA analysis was conducted using steady state information of this model. Based on RGA criteria, outlet concentration of B $(y_1 = c_B)$ is controlled by the reactor flow rate $(u_1 = F)$ and reactor temperature $(y_2 = T)$ is controlled by the external heat exchanger duty $(u_2 = Q_w)$.

In the decentralized IMC scheme, IMC model is given by:

$$\tilde{\mathbf{G}}(s) = \begin{bmatrix} g_{11}(s) & 0 \\ 0 & g_{22}(s) \end{bmatrix} \tag{4.21}$$

where

$$g_{11}(s) = \frac{-0.09s^3 + 0.2897s^2 + 1430s + 10750}{s^4 + 279s^3 + 26220s^2 + 939400s + 9623000}$$

$$g_{22}(s)=\frac{3.083s^2+479.5s+18430}{s^4+279s^3+26220s^2+939400s+9623000}$$

The following two regression vectors are chosen in JITL algorithm:

$$y_1:\mathbf{z}_1(k-1)=[\tilde{y}_1(k-1),\tilde{y}_1(k-2),\tilde{u}_1(k-1)]^{\mathrm{T}}, \tag{4.22}$$

$$y_2:\mathbf{z}_2(k-1)=[\tilde{y}_2(k-1),\tilde{y}_2(k-2),\tilde{u}_1(k-1),\tilde{u}_2(k-1)]^{\mathrm{T}}. \tag{4.23}$$

Table 4.4 Model parameters for cyclopentenol reactor

V	Reactor volume	10 L
T_0	Inlet temperature	403.15 K
C_p	Average heat capacity	30.1 kJ/kg/K
c_{A0}	Inlet concentration of A	5.1 mol/L
ρ	Average density	0.9342 kg/L
k_w	Coolant conductivity	4032 kJ/h/m^2/K
C_{pw}	Coolant heat capacity	2.0 kJ/kg/K
m_w	Coolant mass	5.0 kg
A_w	Heat exchange area	0.215 m^2
$k_{1,0}$	Arrhenius constant	1.287×10^{12} h^{-1}
$k_{2,0}$	Arrhenius constant	1.287×10^{12} h^{-1}
$k_{3,0}$	Arrhenius constant	9.043×10^{9} L/mol/h
E_1	Normalized activation energy	-9758.3 K
E_2	Normalized activation energy	-9758.3 K
E_3	Normalized activation energy	-8560 K
ΔH_1	Heat of reaction	4.3 kJ/mol
ΔH_2	Heat of reaction	-11 kJ/mol
ΔH_3	Heat of reaction	-41.85 kJ/mol

Table 4.5 Nominal operating conditions for cyclopentenol reactor

c_A	1.235 mol/L	T_w	402.1 K
c_B	0.900 mol/L	F	188.3 L/h
T	407.3 K	Q_w	-4496 kJ/h

The input-output data is generated by introducing uniformly random steps with distribution of [50 300] and [-3000 -6500] in process inputs u_1 and u_2 respectively, as shown in Figure 4.12. The input-output data given in Figure 4.12 is then used to construct two databases for JITL algorithm to predict y_1 and y_2. The JITL algorithm parameters $k_{\min} = 15$, $k_{\max} = 100$, and $\gamma = 0.98$ are chosen to predict the first output y_1, whereas $k_{\min} = 10$, $k_{\max} = 80$, and $\gamma = 0.95$ are used to predict y_2. To compare predictive performances of JITL algorithm and linear model, the respective step changes of -100 l/h and -1000 kJ/h in u_1 and u_2 from their nominal values are considered. As can be seen from Figure 4.13, linear model gives erroneous prediction for both transient and steady-state behavior of this reactor in this open-loop test. In contrast, JITL is able to predict the reactor dynamics much more accurately.

To design both decentralized IMC controller $\mathbf{Q}(s)$ and NLIMC controller, the former is chosen as the inverse of linear model $\tilde{\mathbf{G}}(s)$ given in Eq. (4.21) with augmentation of the following filter $\mathbf{F_L}(s)$:

$$\mathbf{F}_L(s) = \begin{bmatrix} \dfrac{1}{0.5s+1} & 0 \\ 0 & \dfrac{1}{(0.04s+1)^2} \end{bmatrix} \tag{4.24}$$

and decentralized NLIMC consists of the aforementioned linear IMC controller with the second filter in auxiliary loop chosen as

$$\mathbf{F}_N(s) = \begin{bmatrix} \dfrac{0.5s+1}{0.1s+1} & 0 \\ 0 & \dfrac{(0.04s+1)^2}{(0.01s+1)^2} \end{bmatrix} \tag{4.25}$$

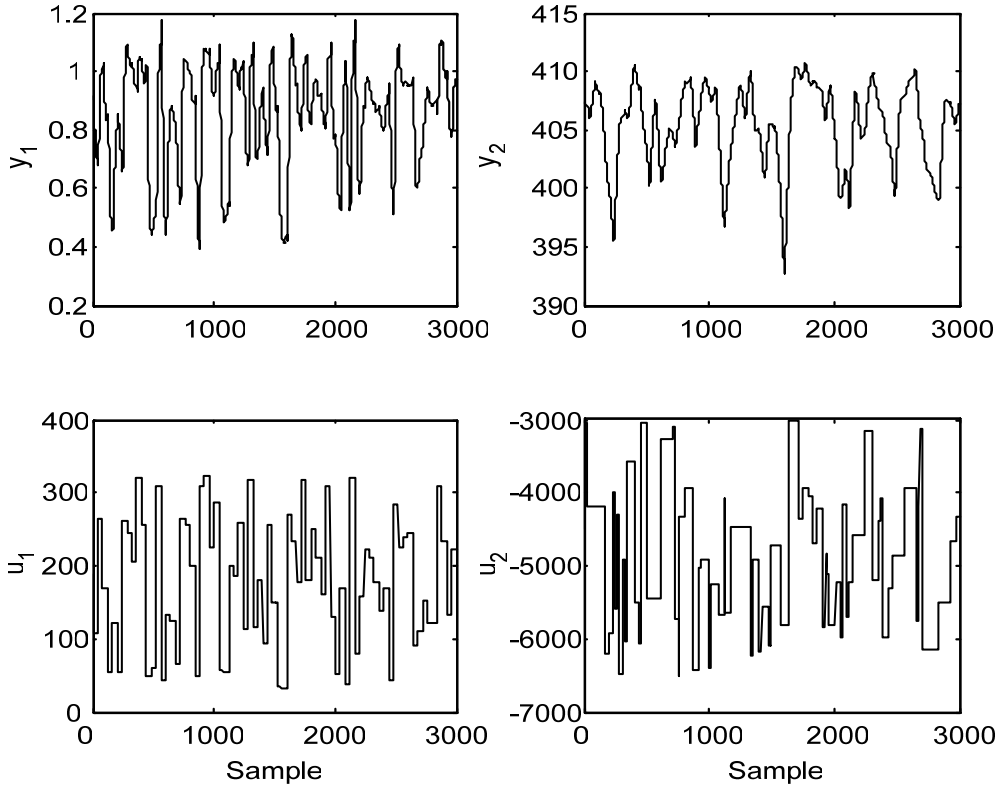

Figure 4.12 Input-output data used for constructing the database

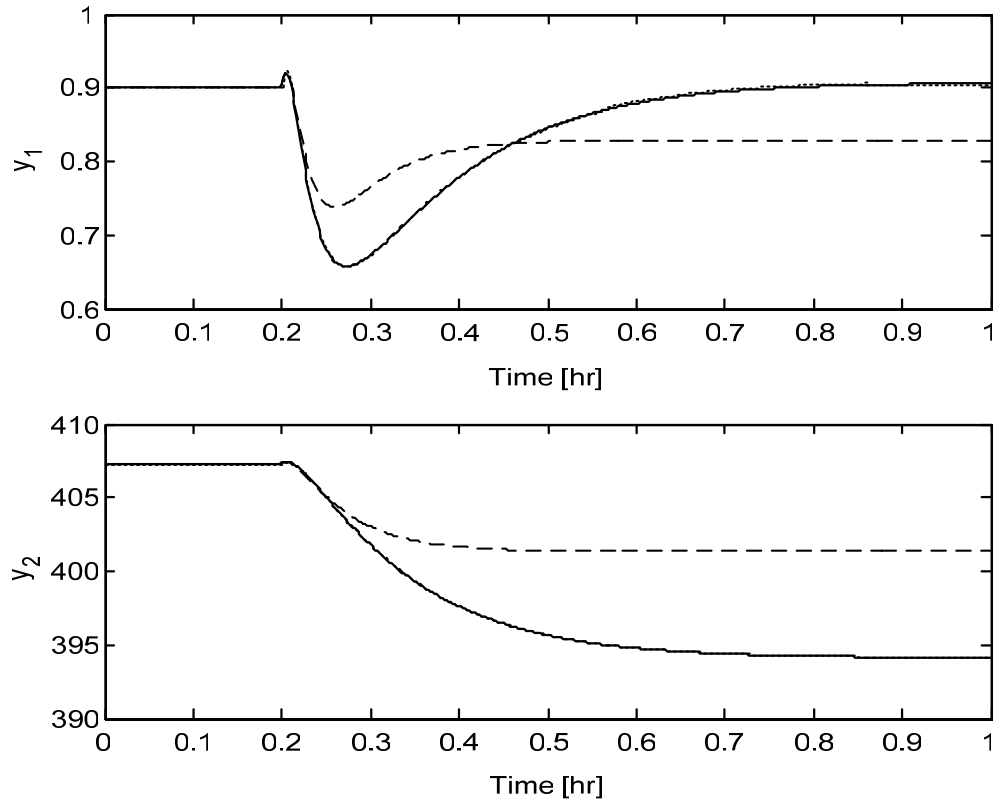

Figure 4.13 Open-loop responses for step changes of -100 and -1000 in u_1 and u_2 respectively. Solid: actual process; dashed: linear model; dotted: JITL

Next, servo performances of two control schemes for different setpoint changes in y_1 and y_2 are compared in Figures 4.14 to 4.17. It is evident that decentralized NLIMC outperforms the decentralized linear IMC not only because the former tracks the trajectory closely (also evidenced by the reduction of MSE between 85% and 95% as given in Table 4.6), but also it can cope with the process interaction more effectively (also evidenced by the reduction of MSE between 74% and 97% as given in Table 4.6). Disturbance rejection capability of these two controllers is evaluated by introducing step disturbances in the inlet concentration of cyclopentadiene (c_{A0}) and the corresponding closed-loop responses are shown in Figures 4.18 and 4.19. It is clear that decentralized NLIMC has better disturbance rejection performance compared to its linear counterpart, as also evidenced by the reduction of MSE between 71% and 93% in Table 4.6.

4.4 Conclusions

A decentralized nonlinear IMC deign strategy using JITL technique is proposed for a class of nonlinear MIMO systems that operate over a range of operating conditions. This IMC strategy makes use of conventional decentralized IMC controller augmented by an auxiliary loop to account for nonlinearities in the system. Simulation results are presented to demonstrate the advantages of proposed control scheme over its conventional counterparts.

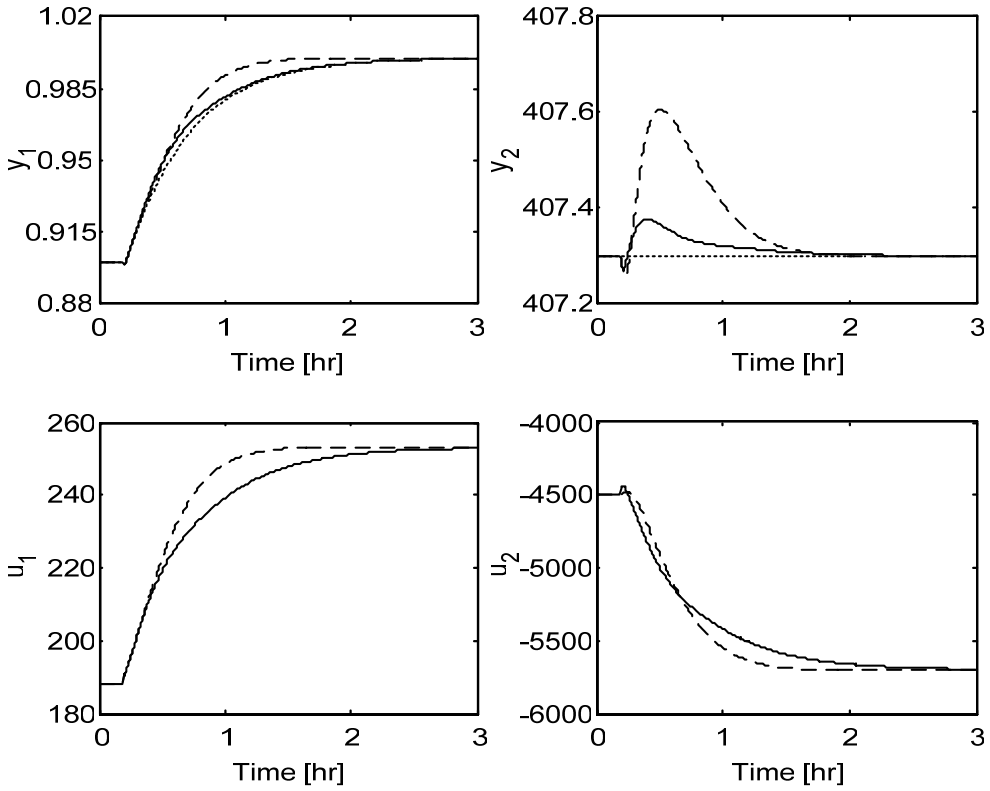

Figure 4.14 Closed-loop responses for setpoint change from 0.9 to 1.0 in y_1. Dotted: reference trajectory; dashed: IMC; solid: NLIMC

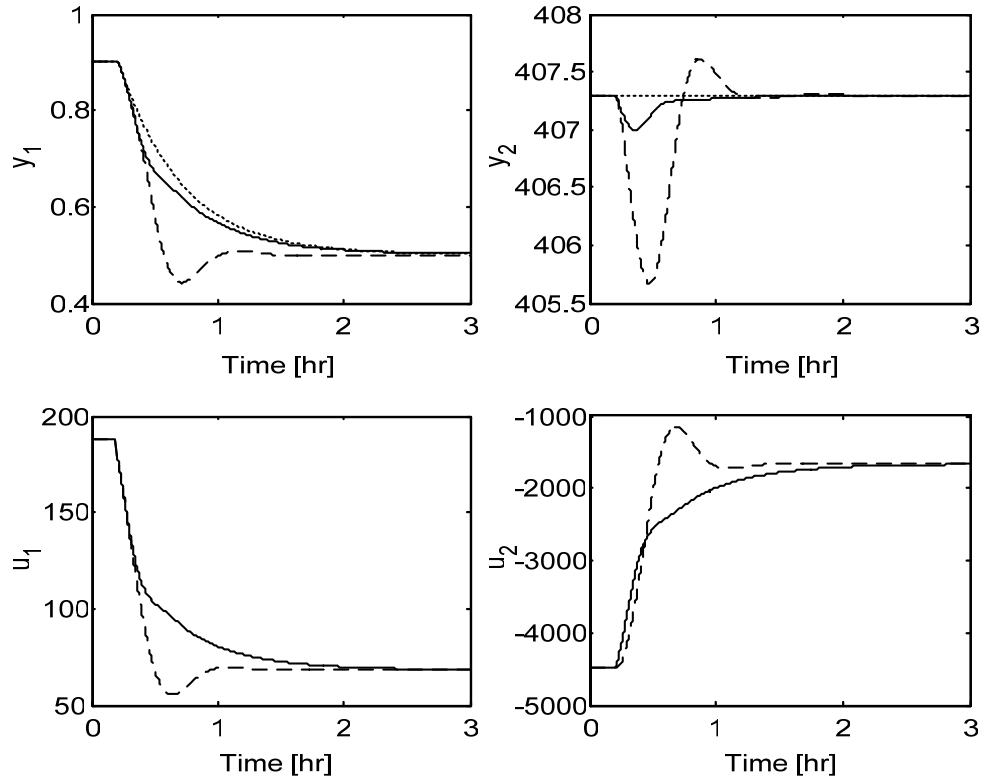

Figure 4.15 Closed-loop responses for setpoint change from 0.9 to 0.5 in y_1. Dotted: reference trajectory; dashed: IMC; solid: NLIMC

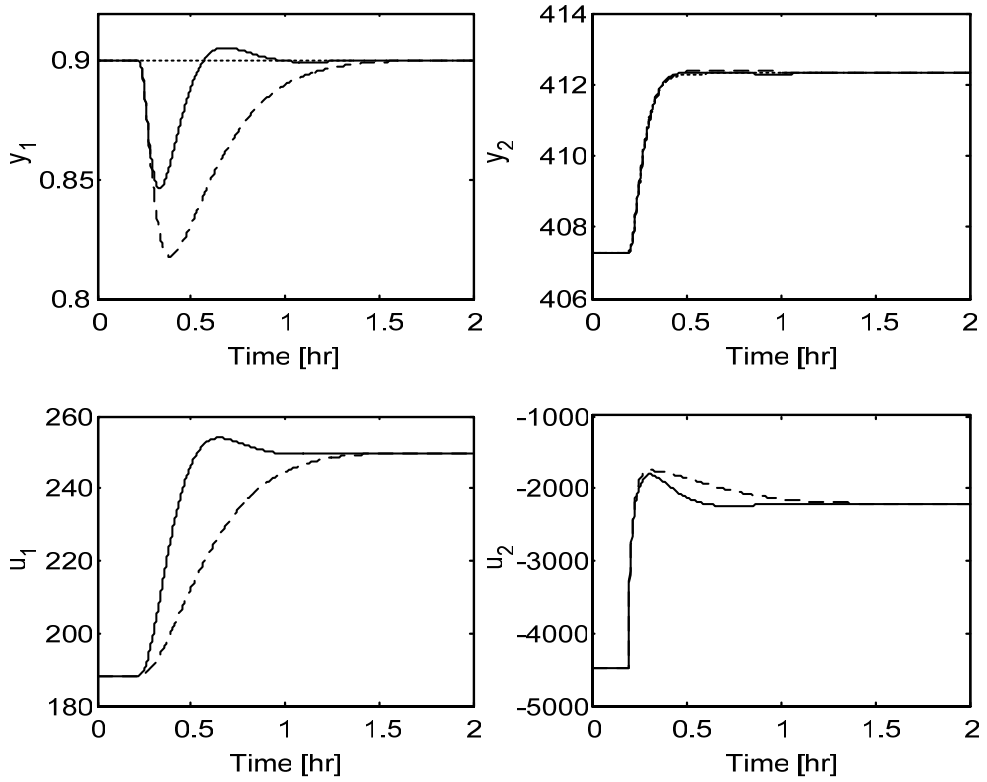

Figure 4.16 Closed-loop responses for setpoint change from 407.3 to 412.3 in y_2.
Dotted: reference trajectory; dashed: IMC; solid: NLIMC

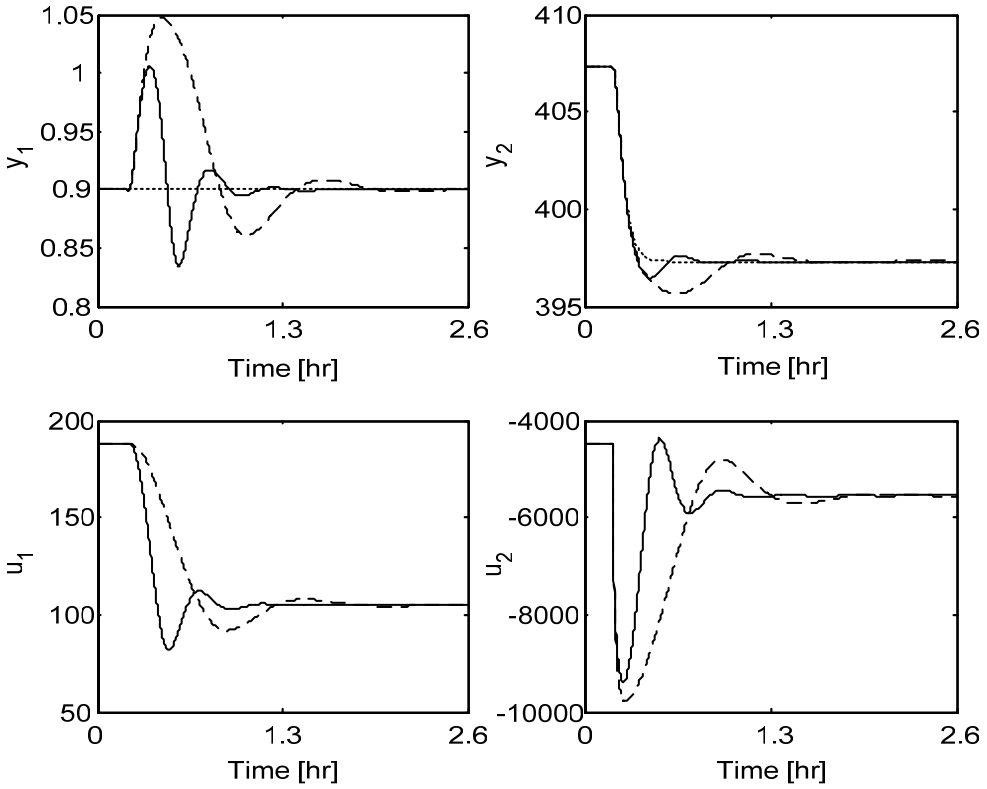

Figure 4.17 Closed-loop responses for setpoint change from 407.3 to 397.3 in y_2.
Dotted: reference trajectory; dashed: IMC; solid: NLIMC

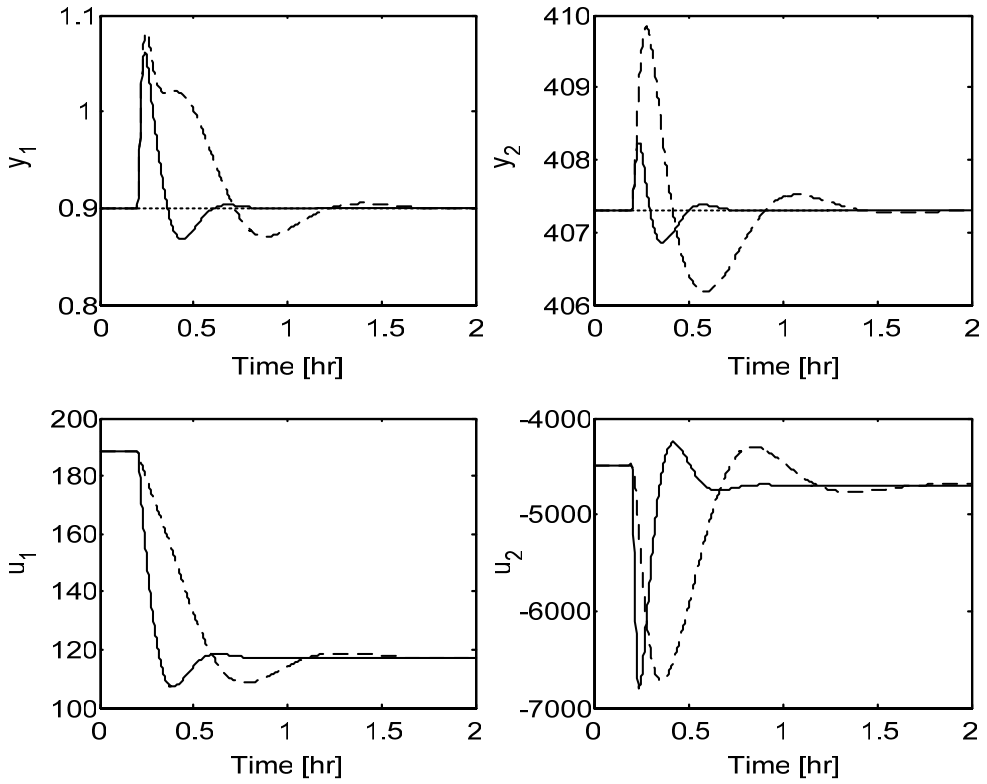

Figure 4.18 Closed-loop responses for step change of +1.5 in c_{A0}. Dotted: reference trajectory; dashed: IMC; solid: NLIMC

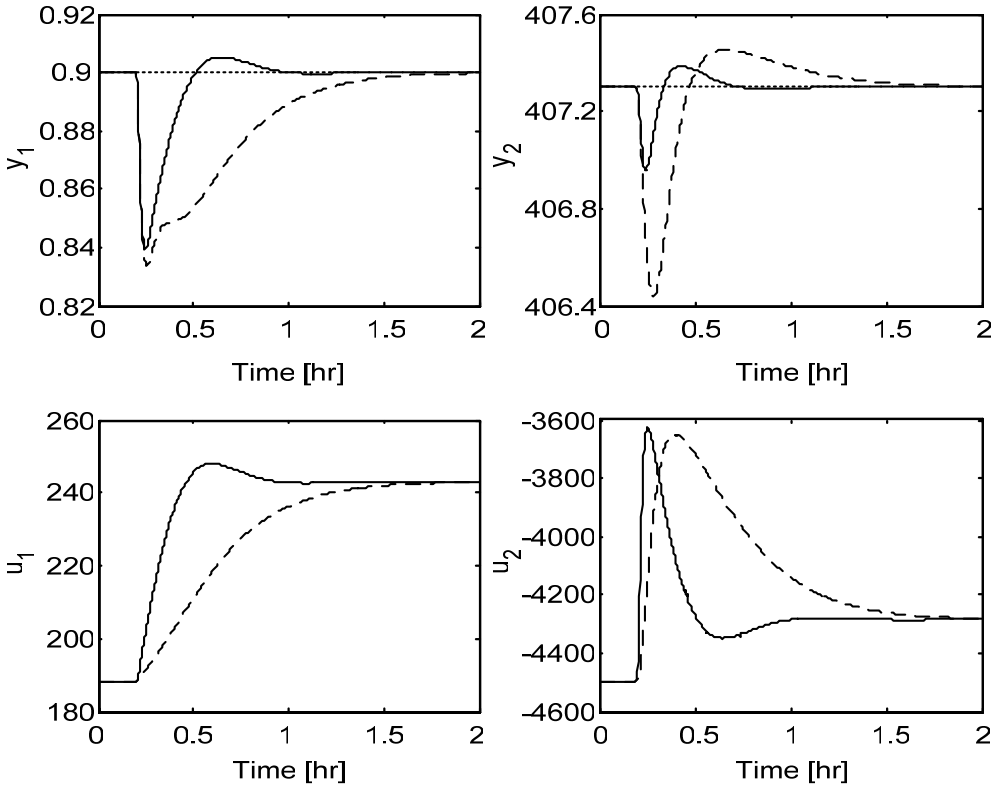

Figure 4.19 Closed-loop responses for step change of -0.5 in c_{A0}. Dotted: reference trajectory; dashed: IMC; solid: NLIMC

Table 4.6 Comparison of closed-loop performances between decentralized IMC and NLIMC

Step change	Tracking error (MSE)			
	Decentralized IMC		Decentralized NLIMC	
	y_1	y_2	y_1	y_2
r_1=0.9 to 1.0	3.9568×10^{-5}	1.3118×10^{-2}	3.6038×10^{-6}	6.8081×10^{-4}
r_1=0.9 to 0.5	5.8422×10^{-3}	2.1430×10^{-1}	3.0384×10^{-4}	6.1092×10^{-3}
r_2 =407.3 to 412.3	9.6047×10^{-4}	2.7000×10^{-3}	1.8788×10^{-4}	3.2899×10^{-4}
r_2 =407.3 to 397.3	2.7508×10^{-3}	3.7960×10^{-1}	7.2226×10^{-4}	5.5400×10^{-2}
c_{A0} =5.1 to 6.6	2.9537×10^{-3}	4.4190×10^{-1}	8.3504×10^{-4}	2.7411×10^{-2}
c_{A0} =5.1 to 4.6	6.2202×10^{-4}	4.3724×10^{-2}	1.7696×10^{-4}	3.7388×10^{-3}

CHAPTER 5

Memory-Based Internal Model Control Design

5.1 Introduction

The IMC scheme has been under intensive research and development in the last two decades due to its simple yet effective framework for system design. The idea inherent in the IMC has been floating around in one form or another for several decades. There are quite many nonlinear IMC design methods proposed in the literature using nonlinear models. The difficulty in the use of these models in the IMC design strategy arises in the design of IMC controller, which is based on the inverse of the model. Detailed discussions on this issue have been provided in Chapters 2 and 3.

One attractive feature of IMC is its ability to address the transient response and the robustness issues in a transparent manner. Typically, an IMC filter is used to make a compromise between the robustness and performance requirement. Generally this IMC filter parameter is kept fixed. However, since most process systems have nonlinearities, it may be difficult to obtain good control performance for such systems simply using the fixed filter parameter.

The objective of this chapter is to propose a novel memory-based IMC design to address the above mentioned two control problems. Recent rapid developments of

computer technologies enable us to memorize, fast retrieve and read out a large number of data. By effectively utilizing these advantages, Just-In-Time Learning (JITL) technique described in Chapter 2 offer an attractive alternative for modeling the nonlinear systems. Comparing to the traditional methods, JITL has no standard learning phase. It merely gathers the data and stores them in the database and the computation is not performed until a query data arrives. It is worth noting that JITL is only locally valid for the operating condition characterized by the current query data, meaning that JITL constructs local approximation of the dynamic systems. Therefore, a simple model structure can be chosen, e.g. a low order ARX model. As a result, it is quite easy to obtain inverse of such models. Hence first control problem is a trivial task by using JITL algorithm as a process model.

The second problem is addressed here by making IMC filter parameter adaptive. At each sampling instant, the proposed method initializes IMC filter parameter using a controller database. Subsequently, the filter parameter is updated on-line in proportion to control error and the resulting filter parameter is stored in the controller database. Therefore, the most current filter parameter corresponding to the characteristics of the controlled object is newly stored. Finally, the effectiveness of the newly proposed control scheme is examined on a polymerization reactor, which was discussed in Chapter 3, and a pH neutralization process. The simulation results of these two processes demonstrate the ability of the proposed control strategy to outperform the benchmark PI/PID controller reported in the literature.

5.2 Memory-Based IMC Strategy

Consider the class of discrete-time nonlinear systems that can be described accurately by the JITL technique discussed in Chapter 2, where each local model

obtained by JITL is only locally valid around the query data, therefore a low-order ARX model can be chosen as local model at each query point, i.e.

$$y(k) \cong \hat{y}(k) = f(\mathbf{z}(k-1)) = \mathbf{z}^{\mathrm{T}}(k-1)\boldsymbol{\psi}, \tag{5.1}$$

where $y(k)$ denotes the system output, $\hat{y}(k)$ is the predicted output by JITL algorithm, $f(\cdot)$ denotes a linear function, $\mathbf{z}(k)$ is the regression vector, and $\boldsymbol{\psi}$ is the model parameter vector. The regression vector and model parameter vector are defined by the following equations:

$$\mathbf{z}(k-1) = \left[y(k-1),\ldots,y(k-n_y),u(k-1),\ldots,u(k-n_u)\right]^{\mathrm{T}}, \tag{5.2}$$

$$\boldsymbol{\psi} = [\psi_1,\ldots,\psi_{n_y},\psi_{n_y+1},\ldots,\psi_{n_y+n_u}]^{\mathrm{T}}, \tag{5.3}$$

where n_y and n_u are integers related to the model's order.

The above mentioned JITL model serves as IMC model in our proposed control scheme. For illustration purpose, consider $n_y = 2$ and $n_u = 1$ in the aforementioned JITL model. Then its equivalent transfer function can be expressed as follows:

$$G_m(z) = \frac{\hat{y}(z)}{u(z)} = \frac{\psi_3 z^{-1}}{1-\psi_1 z^{-1}-\psi_2 z^{-2}} \tag{5.4}$$

At each sampling instant, the controller $Q(z)$ is obtained by inverting $G_{m-}(z)$ (the invertible part of $G_m(z)$) as required in IMC scheme and then multiplying by a first-order filter to improve robustness of the controller as well as to ensure physical realizability of controller. Thus JITL is employed not only to update model parameters but also to adjust the parameters of IMC controller. Following these steps, the controller transfer function using model described in Eq. (5.4) can be derived as:

$$Q(z) = \frac{u(z)}{e(z)} = \frac{1-\psi_1 z^{-1}-\psi_2 z^{-2}}{\psi_3} f(z), \tag{5.5}$$

where $f(z)$ is a first order filter with tuning parameter α and as given by

$$f(z) = \frac{1-\alpha}{1-\alpha z^{-1}} \tag{5.6}$$

The control law from Eq. (5.5) becomes:

$$u(k) = \alpha u(k-1) + \frac{1-\alpha}{\psi_3}\left[e(k) - \psi_1 e(k-1) - \psi_2 e(k-2)\right], \tag{5.7}$$

where $e(k)$ denotes the controller input defined by

$$e(k) = r(k) - y(k) + \hat{y}(k) \tag{5.8}$$

where $r(k)$ denotes the reference signal and $\hat{y}(k)$ denotes the JITL prediction. As mentioned earlier, it is quite difficult to obtain a good control performance due to nonlinearities, if filter parameter α in Eq. (5.7) is fixed. Therefore, here we propose a new control scheme, which can adjust this parameter in an on-line manner corresponding to characteristics of the system. Thus, instead of Eq. (5.7), the following control law with variable filter parameter at each sampling instant is employed:

$$u(k) = \alpha(k) u(k-1) + \frac{1-\alpha(k)}{\psi_3}\left[e(k) - \psi_1 e(k-1) - \psi_2 e(k-2)\right] \tag{5.9}$$

Now, Eq. (5.9) can be rewritten as the following relations:

$$u(k) = g(\phi'(k)) \tag{5.10}$$

$$\begin{aligned}\phi'(k) := [&\alpha(k), r(k), r(k-1), r(k-2),\\ &y(k), y(k-1), y(k-2), u(k-1)]\end{aligned} \tag{5.11}$$

where $g(\cdot)$ denotes a linear function. By substituting Eq. (5.10) and Eq. (5.11) into Eq. (5.1) and Eq. (5.2), the following equation can be derived:

$$y(k+1) = f(\mathbf{z}(k)) = f\left([y(k), y(k-1), u(k)]^{\mathrm{T}}\right) = h\left(\tilde{\phi}(k)\right) \tag{5.12}$$

$$\begin{aligned}\tilde{\phi}(k) := [&y(k), y(k-1), y(k-2), \alpha(k),\\ &r(k), r(k-1), r(k-2), u(k-1)]\end{aligned} \tag{5.13}$$

where $h(\cdot)$ denotes a linear function. Therefore, $\alpha(k)$ is given by the following equations:

$$\alpha(k) = F\left(\bar{\phi}(k)\right) \tag{5.14}$$

$$\begin{aligned}\bar{\phi}(k) := [&y(k+1), y(k), y(k-1), y(k-2),\\ &r(k), r(k-1), r(k-2), u(k-1)]\end{aligned} \tag{5.15}$$

where $F(\cdot)$ denotes a linear function. Since the future output $y(k+1)$ included in Eq. (5.15) cannot be obtained at the sampling instant *k*, $y(k+1)$ is replaced by $r(k+1)$. Therefore, the information vector $\bar{\phi}(k)$ included in Eq. (5.15) is rewritten as follows:

$$\begin{aligned}\bar{\phi}(k) := [&r(k+1), r(k), r(k-1), r(k-2),\\ &y(k), y(k-1), y(k-2), u(k-1)]\end{aligned} \tag{5.16}$$

Based on the on-going analysis, a new memory-based IMC control scheme is designed based on JITL technique, and its controller design algorithm is summarized step by step as follows:

Step 1 Generation of initial controller database

To generate the initial controller database, closed-loop data around the nominal operating point are obtained with a priori designed IMC controller, whose filter parameter (α^0) is kept fixed and tuned to give satisfactory control performance. Each element of the initial controller database is then generated as follows:

$$\Phi(j) := [\bar{\phi}(j), \alpha(j)], \qquad j = 1, 2, \cdots, N(0) \tag{5.17}$$

where $\bar{\phi}(j)$ is given by Eq. (5.16) and $N(0)$ denotes the number of information vectors stored in the initial controller database. Note that all filter parameters included in the initial controller database are identical, i.e. $\alpha(1) = \alpha(2) = \cdots = \alpha(N(0)) = \alpha^0$.

Step 2 Selection of nearest-neighbors

At each sampling instant *k*, distances between the query data $\bar{\phi}(k)$ and information vectors $\bar{\phi}(j)$ are calculated using the following 2-norm measure:

$$d_j = \left\| \bar{\phi}(k) - \bar{\phi}(j) \right\|_2, \qquad j = 1, 2, \cdots, N(k) \tag{5.18}$$

where $N(k)$ denotes the number of information vectors stored in the current controller database at *k*-th sampling instant. Then similarity number for each information vector is calculated as follows:

$$s_j = \sqrt{e^{-d_j^2}} \tag{5.19}$$

Subsequently, nearest-neighbors are chosen from all of the information vectors with *n* largest similarity numbers.

Step 3 Construction of IMC filter parameter

Using the nearest-neighbors selected in step 2, IMC filter parameter is initialized based on the following equation:

$$\alpha^{old}(k) = \sum_{i=1}^{n} w_i \alpha(i), \tag{5.20}$$

where w_i denotes the weight corresponding to the *i*-th information vector $\bar{\phi}(i)$ in the selected nearest-neighbors, and is calculated by:

$$w_i = \frac{s_i}{\sum_{i=1}^{n} s_i} \tag{5.21}$$

Step 4 Data adjustment

To further refine the filter parameter obtained in step 3, an updating algorithm is developed in the sequel to adjust the IMC filter at each sampling instant. In doing so, an objective function is defined as

$$J(k+1) := \frac{1}{2}\varepsilon(k+1)^2 \tag{5.22}$$

$$\varepsilon(k+1) = r(k+1) - \hat{y}(k+1). \tag{5.23}$$

By using the steepest descent gradient method, a parameter tuning algorithm is derived as follows:

$$\alpha^{new}(k) = \alpha^{old}(k) - \eta \frac{\partial J(k+1)}{\partial \alpha(k)} \tag{5.24}$$

where η denotes the adaptive learning rate, which is made adaptive by following rules: (i) if the increment of J is more than threshold, the learning rate is decreased by a factor l_{dec}, i.e. $\eta(k) = l_{dec}\eta(k-1)$; (ii) if the new error is smaller than old error, the learning rate is increased by a factor l_{inc}, i.e. $\eta(k) = l_{inc}\eta(k-1)$; (iii) otherwise learning rate is kept same as previous sampling instant.

Moreover, partial derivative in Eq. (5.24) is developed as follows:

$$\begin{aligned} \frac{\partial J(k+1)}{\partial \alpha(k)} &= \frac{\partial J(k+1)}{\partial \varepsilon(k+1)} \frac{\partial \varepsilon(k+1)}{\partial \hat{y}(k+1)} \frac{\partial \hat{y}(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial \alpha(k)} \\ &= -\varepsilon(k+1) \frac{\partial \hat{y}(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial \alpha(k)} \end{aligned} \tag{5.25}$$

where $\frac{\partial \hat{y}(k+1)}{\partial u(k)} = \psi_3$ can be readily obtained from the JITL and $\frac{\partial u(k)}{\partial \alpha(k)}$ can be obtained from Eq. (5.9). The block diagram of the proposed control scheme is shown in Figure 5.1.

This new filter parameter obtained by Eq. (5.24) along with the corresponding information vector $\bar{\phi}(k)$ is stored in the controller database as given in Eq. (5.17) and when next query arrives, go to step 2.

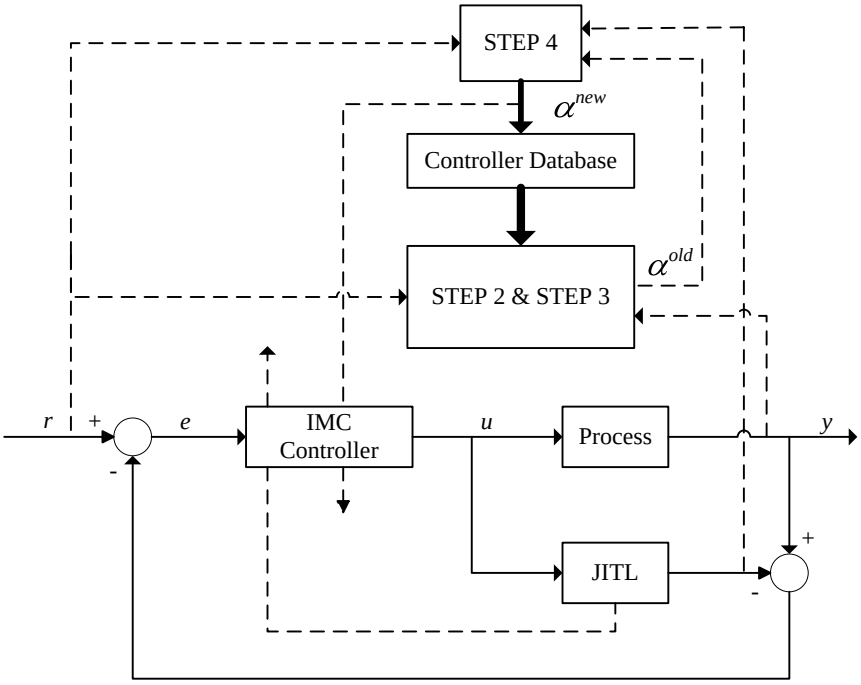

Figure 5.1 Memory-based IMC control scheme

5.3 Examples

Example 1: The first example considered here is polymerization reaction taking place in a jacketed CSTR, which was discussed earlier in Chapter 3. Again control objective is to manipulate the volumetric flow rate of the initiator $(u = F_I)$ in order to regulate the number-average molecular weight $(y = \text{NAMW})$. The operating space and sampling time are the same as those chosen in NLIMC scheme. The identical non-adaptive JITL algorithm including database, regression vector, and algorithm parameters, as discussed in Chapter 3 is also used here for modeling purpose.

To proceed with proposed memory-based IMC design strategy, initial controller database is generated by introducing local setpoint changes in the IMC scheme with fixed filter parameter and filter parameter, $\alpha = 0.84$, is found to give satisfactory setpoint performance. Thus, this parameter is included in the initial information vectors. Furthermore, the user-specified parameters included in the proposed method are determined as shown in Table 5.1. For the comparison purpose,

a PI controller with parameters $K_c = -1.371$ and $\tau_I = 0.225\,\text{h}$ (Maner and Doyle, 1997) is also designed.

The control performances of these two controllers for step changes of $\pm 50\%$ in setpoint from its nominal value of 25000.5 kg/kmol are shown in Figure 5.2. Although the closed-loop responses obtained by these two controllers are very close for positive step change in the setpoint, memory-based IMC design scheme brings the reactor to the new operating point without causing any significant overshoot or undershoot. For a negative step change in the setpoint, it can be seen that the proposed control scheme brings the process into the new setpoint much faster as compared to PI controller. The corresponding trajectory of filter parameter in the memory-based IMC design is also illustrated in Figure 5.2. It may be argued that the PI controller could be tuned more aggressively so that the 12500 kg/kmol setpoint would be reached more quickly. However, this approach would cause poorer control performance for the positive setpoint change. In fact, PI controller is already tuned quite aggressively as evidenced by the underdamped response and manipulated variable profile for the positive setpoint change. Tuning PI controller more aggressively would yield severe oscillation to those shown in Figure 5.2. Hence, the nonlinear behaviour of this process requires a compromise in the tuning of a linear controller for servo control.

Table 5.1 User-specified parameters in the proposed method (polymerization reactor)

Initial number of data	$N(0) = 150$
Number of nearest-neighbors	$n = 6$
Initial learning rate	$\eta = 0.8$

Another important measure of control-system performance is the ability to reject unmeasured disturbances. To compare disturbance rejection capability of both controller designs, unmeasured $\pm 25\%$ step disturbances in inlet initiator concentration $\left(C_{I_{in}}\right)$ are considered and corresponding closed-loop responses are shown in Figure 5.3. It is evident that memory-based IMC design scheme gives improved performance for disturbance rejection than that obtained by PI controller. Table 5.2 summarizes the MSE for both setpoint tracking and disturbance rejection performances aforementioned. It is clear that memory-based IMC design scheme reduces the MSE significantly, relative to PI controller, by a margin between 6% and 54%.

Table 5.2 Comparison of closed-loop performances between PI and memory-based IMC controllers

Step change	Tracking error (MSE)		% Decrease in MSE
	PI	Memory-based IMC	
r = 25000.5 to 37500	1.5445×10^{7}	1.4527×10^{7}	5.94
r = 25000.5 to 12500	1.9844×10^{7}	9.2094×10^{6}	53.59
+25% change in $C_{I_{in}}$	3.5892×10^{5}	2.4088×10^{5}	32.89
-25% change in $C_{I_{in}}$	4.6295×10^{5}	2.3784×10^{5}	48.63

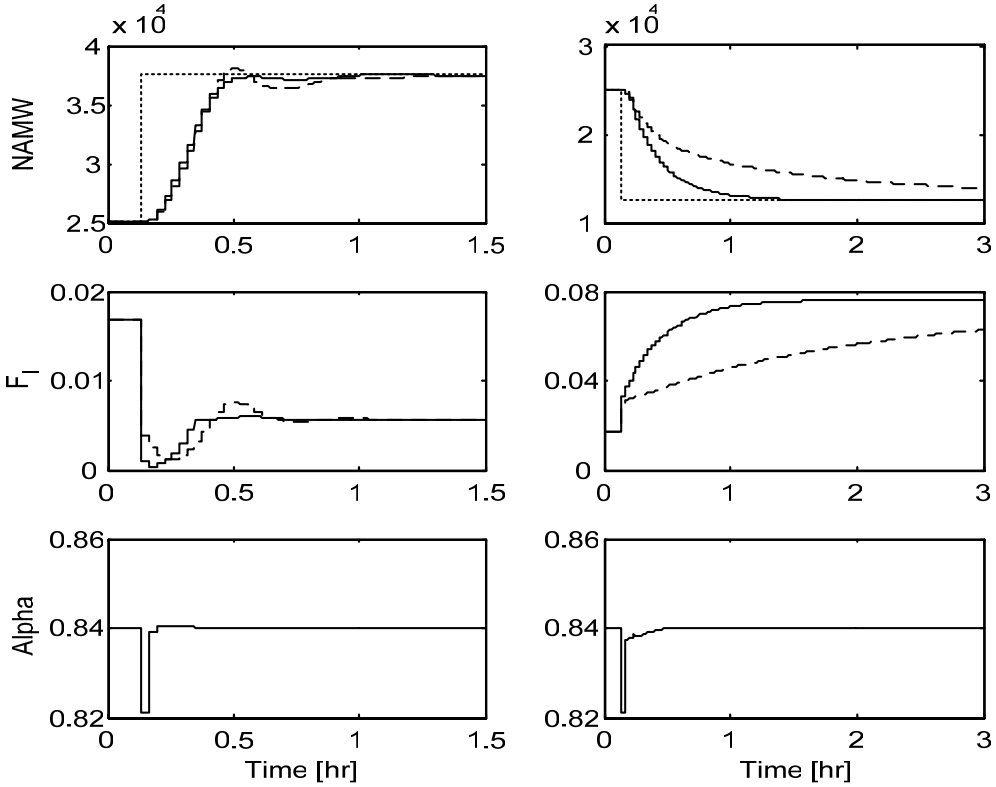

Figure 5.2 Closed-loop responses for $\pm 50\%$ step changes in setpoint. Dotted: setpoint; dashed: PI; solid: memory-based IMC

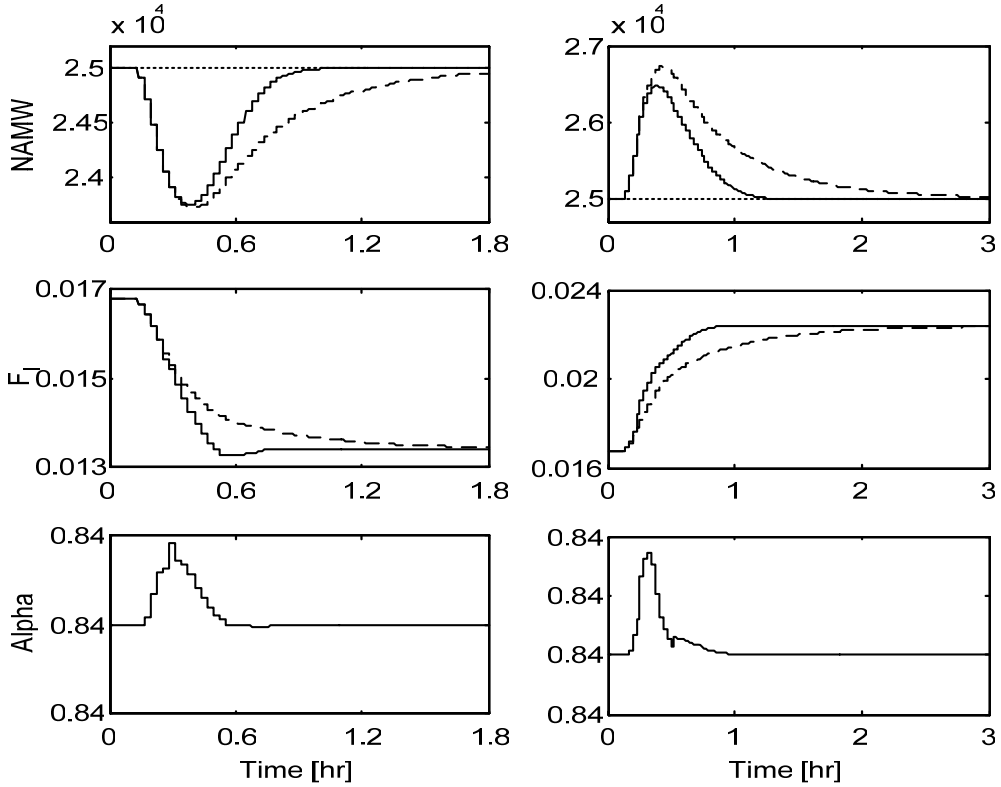

Figure 5.3 Closed-loop responses for $+25\%$ (left) and -25% (right) step changes in $C_{I_{in}}$. Dotted: setpoint; dashed: PI; solid: memory-based IMC

Example 2: The proposed control strategy is applied to a pH neutralization process. A schematic of this process is shown in Figure 5.4. Acid, buffer and base streams are mixed in a tank as shown in figure and effluent pH is measured. Three inlet streams are:

Acid stream: 0.003M HNO_3

Buffer stream: 0.03M $NaHCO_3$

Base stream: 0.003M NaOH, 0.00005M $NaHCO_3$

The process model is derived by defining the following reaction invariants (Nahas et al., 1992; Aoyama et al., 1995):

$$W_a \equiv [H^+] - [OH^-] - [HCO_3^-] - 2[CO_3^{2-}] \tag{5.26}$$

$$W_b \equiv [H_2CO_3] + [HCO_3^-] + [CO_3^{2-}] \tag{5.27}$$

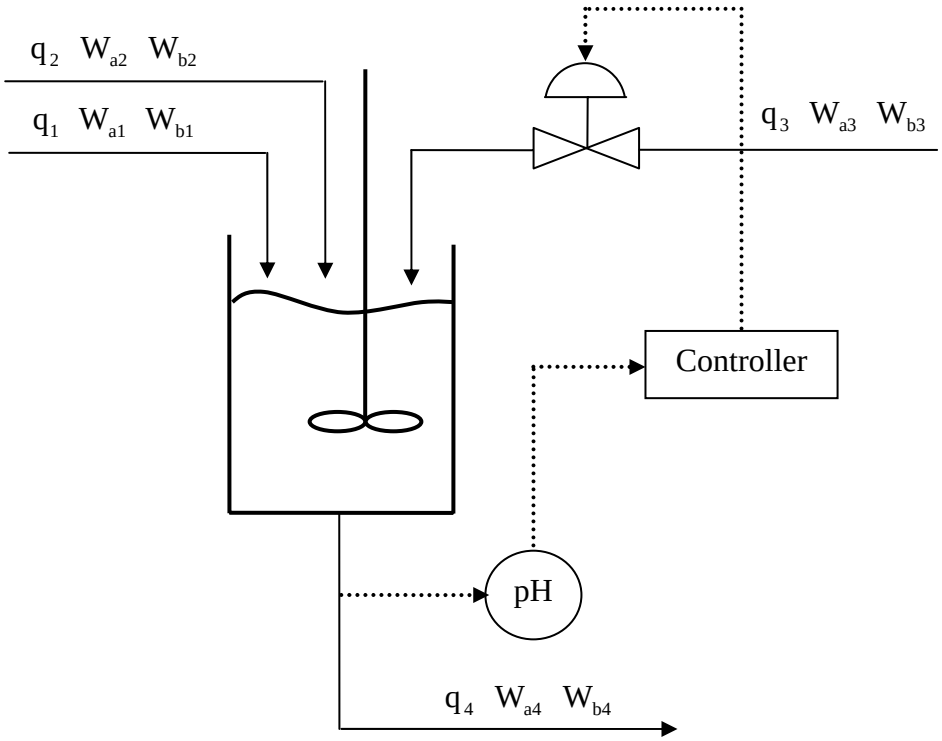

Figure 5.4 pH neutralization system

The first invariant represents a charge balance, while the second represents a balance on the carbonate ion. Unlike pH, these reaction invariants are conserved quantities. The resulting process model consists of three nonlinear ordinary differential equations and a nonlinear output equation for the pH:

$$\dot{h} = \frac{1}{A}\left(q_1 + q_2 + q_3 - C_V h^{0.5}\right), \tag{5.28}$$

$$\dot{W}_{a4} = \frac{1}{Ah}\left[(W_{a1} - W_{a4})q_1 + (W_{a2} - W_{a4})q_2 + (W_{a3} - W_{a4})q_3\right], \tag{5.29}$$

$$\dot{W}_{b4} = \frac{1}{Ah}\left[(W_{b1} - W_{b4})q_1 + (W_{b2} - W_{b4})q_2 + (W_{b3} - W_{b4})q_3\right], \tag{5.30}$$

$$W_{a4} + 10^{\mathrm{pH}-14} + W_{b4}\frac{1 + 2\times 10^{\mathrm{pH}-\mathrm{pK}_2}}{1 + 10^{\mathrm{pK}_1-\mathrm{pH}} + 10^{\mathrm{pH}-\mathrm{pK}_2}} - 10^{-\mathrm{pH}} = 0, \tag{5.31}$$

where h is the liquid level, W_{a4} and W_{b4} are the reaction invariants of the effluent stream, and q_1, q_2 and q_3 are the acid, buffer and base flow rate, respectively.

The model parameters and nominal operating conditions are given in Table 5.3. The control objective is to manipulate the base flow rate ($u = q_3$) in order to regulate the pH in the tank, i.e. $y = \mathrm{pH}$. The operating space considered is $\mathrm{pH} \in [4\ \ 7]$. The process sampling time is chosen as 0.25 min and the step change in the process input (open-loop test) or setpoint is made at the time equal to 1 min in the following simulation studies. The manipulated input appears linearly in the model equations for reaction invariants; however, the relationship between reaction invariants and effluent pH is expressed by a highly nonlinear Eq. (5.31). The highly nonlinear dynamics can be observed from open-loop responses obtained by considering $\pm 10\%$ step changes in q_3 from its nominal value as shown in Figure 5.5. It can be seen that the process gain varies by more than 250% for these small input changes.

To proceed with non-adaptive JITL algorithm, first-order ARX model is employed as local model, i.e. the regression vector is chosen as $\mathbf{z}(k-1)=[\tilde{y}(k-1),\tilde{u}(k-1)]^{\mathrm{T}}$, where $\tilde{y}$ and $\tilde{u}$ are the respective normalized process output and input as defined by $\tilde{y}=(y-y_0)/y_0$ and $\tilde{u}=(u-u_0)/u_0$, where y_0 and u_0 are nominal operating values for the corresponding process variables. The database is generated by introducing uniformly random steps with distribution of [580 1035] in process input as displayed in Figure 5.6. The JITL algorithm parameters, $k_{min}=30$, $k_{\max}=60$, and $\gamma=0.95$, are chosen to achieve the smallest MSE in the validation test.

In the proposed memory-based IMC design strategy, initial controller database is generated by introducing local setpoint changes in the IMC scheme with fixed filter parameter $\alpha=0.82$, which is found to give satisfactory control performance. Thus, this parameter is included in the initial information vectors. Furthermore, the user-specified parameters included in the proposed method are determined as shown in Table 5.4.

Table 5.3 Model parameters and nominal operating conditions for pH system

$A=207\,\mathrm{cm}^2$	$W_{b2}=3\times10^{-2}\ \mathrm{M}$
$C_V=525\,\mathrm{ml\,cm^{-1}\,min^{-1}}$	$W_{b3}=5\times10^{-5}\ \mathrm{M}$
$\mathrm{pK}_1=6.35$	$q_1=996\,\mathrm{ml\,min^{-1}}$
$\mathrm{pK}_2=10.25$	$q_2=33\,\mathrm{ml\,min^{-1}}$
$W_{a1}=3\times10^{-3}\ \mathrm{M}$	$q_3=936\,\mathrm{ml\,min^{-1}}$
$W_{a2}=-3\times10^{-2}\ \mathrm{M}$	$h=14.0\,\mathrm{cm}$
$W_{a3}=-3.05\times10^{-3}\ \mathrm{M}$	$\mathrm{pH}=7.0$
$W_{b1}=0\,\mathrm{M}$	

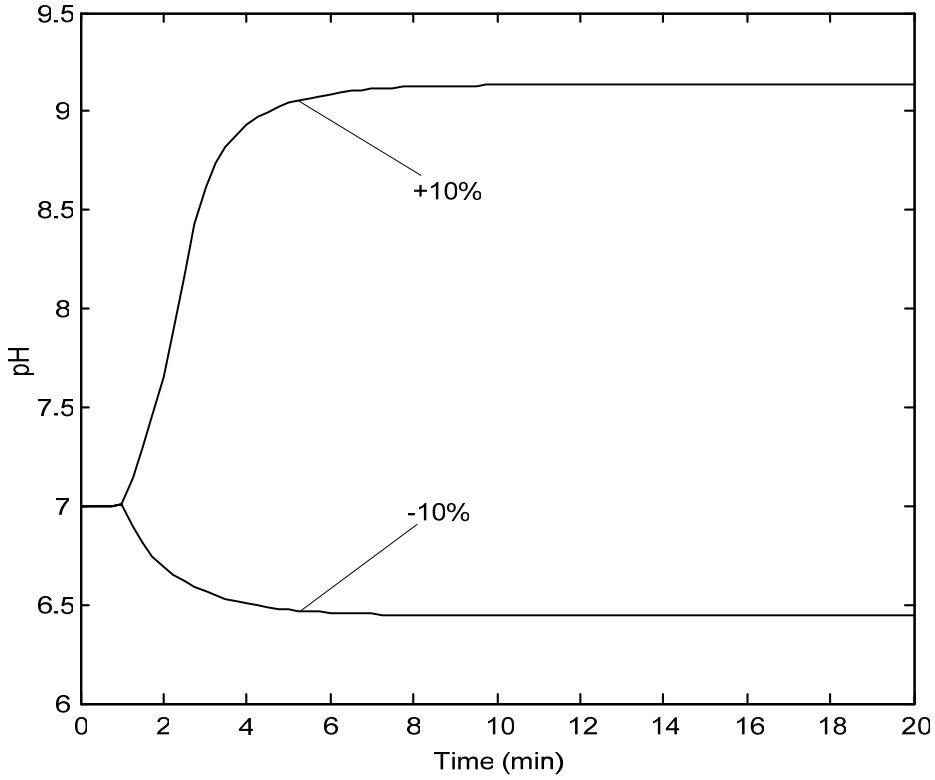

Figure 5.5 Open-loop responses of pH neutralization system for step changes in the base flow rate (q_3)

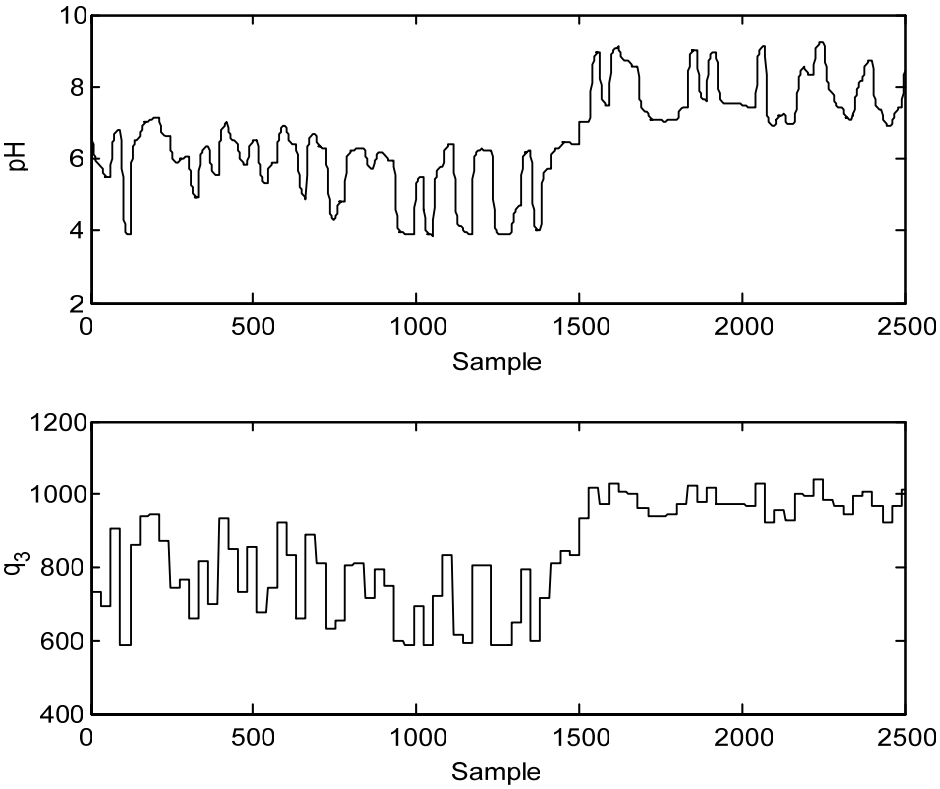

Figure 5.6 Input-output data used for constructing the database

Table 5.4 User-specified parameters in the proposed method (pH neutralization system)

Initial number of data	$N(0) = 100$
Number of nearest-neighbors	$n = 6$
Initial learning rate	$\eta = 0.9$

Again, the performance of the memory-based IMC is compared to that of a PID controller. The parameters of PID controller are tuned to provide a compromise for two setpoint changes from 7 to 4 and from 7 to 9, resulting in the values: $K_c = 20.0\,\text{min}\,\text{ml}^{-1}$, $\tau_I = 1.0\,\text{min}$ and $\tau_D = 0.2\,\text{min}$ (Aoyama et al., 1995).

The two controllers are compared for a step change in the setpoint from 7 to 4 and from 7 to 9, respectively. The setpoint tracking performance of two controllers and trajectory of filter parameter are illustrated in Figure 5.7. The memory-based IMC controller yields a fast response for both setpoint changes. In contrast, the PID controller is very sluggish. The setpoint tracking performance can be improved by increasing K_c, but this leads to an undesirable oscillatory behavior for changes in buffer flow rate. In Figure 5.8, these two controllers are compared for unmeasured step disturbances of +27 $\text{ml}\,\text{min}^{-1}$ and -33 $\text{ml}\,\text{min}^{-1}$ in the buffer flow rate, respectively. It is clear that the proposed control scheme provides faster disturbance rejection. A quantitative summary of closed-loop performances for both setpoint tracking and disturbance rejection in terms of MSE is given in Table 5.5. It is evident that proposed control strategy reduces the MSE significantly, relative to PID controller, by a margin between 61% and 88%.

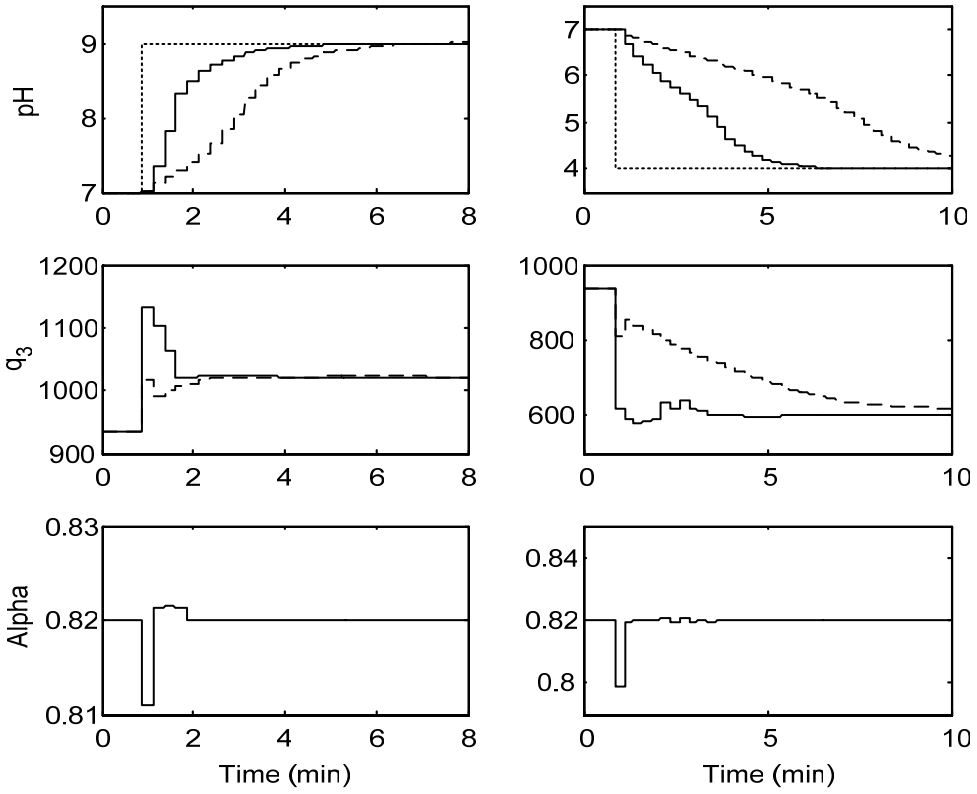

Figure 5.7 Closed-loop responses for step changes in setpoint. Dotted: setpoint; dashed: PID; solid: memory-based IMC

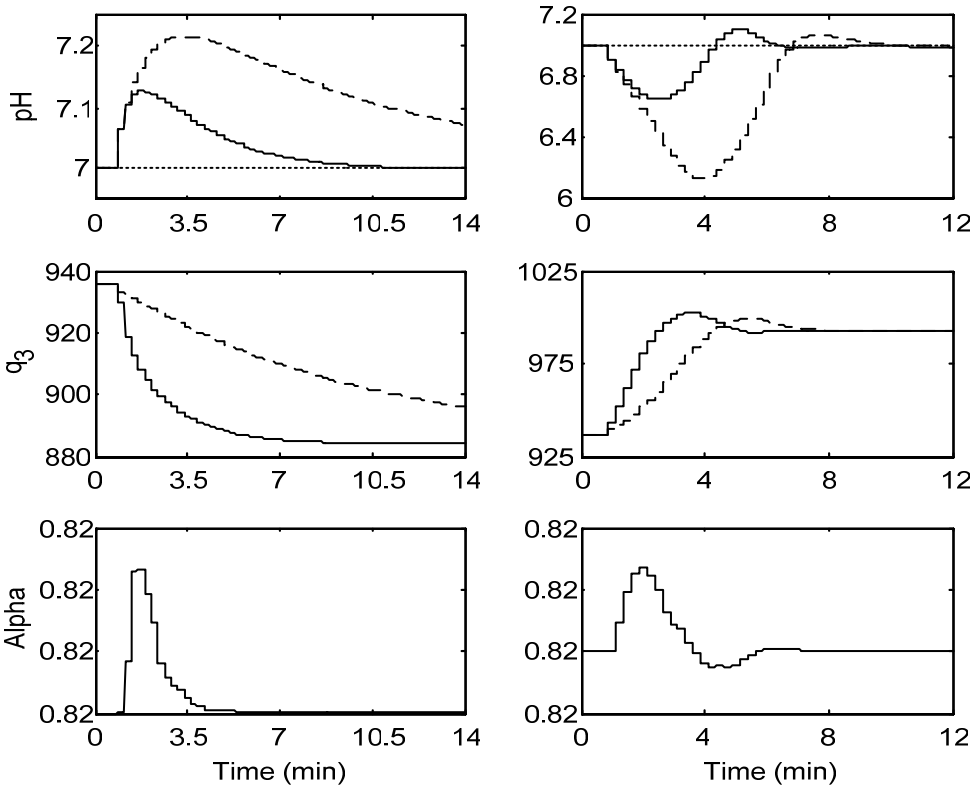

Figure 5.8 Closed-loop responses for step changes of +27 (left) and -33 (right) in q_2. Dotted: setpoint; dashed: PID; solid: memory-based IMC

Table 5.5 Comparison of closed-loop performances between PID and memory-based IMC controllers

Step change	Tracking error (MSE)		% Decrease in MSE
	PID	Memory-based IMC	
r = 7 to 9	0.7226	0.2783	61.49
r = 7 to 4	3.1731	1.1538	63.64
-33 step change in q_2	0.1594	0.0187	88.27
+27 step change in q_2	0.0217	0.0029	86.64

5.4 Conclusions

A memory-based IMC design strategy is proposed for a class of nonlinear systems that can be modelled accurately by JITL technique. At each sampling instant, the initial IMC filter parameter is obtained using a controller database. Furthermore, parameter updating algorithm is developed by employing the steepest descent gradient method and is used to adjust the initial filter parameter on-line. The proposed control strategy is evaluated through simulation studies to show better controller performance than the benchmark PI/PID controller reported in the literature.

CHAPTER 6

Conclusions

In this research work, two novel IMC design methods are proposed for a class of nonlinear systems that can be modelled accurately by JITL technique. Firstly, a nonlinear IMC design method is developed to extend the conventional IMC design to a certain class of SISO nonlinear systems that operate over a wide range of operating regimes. This IMC strategy makes use of conventional linear IMC controller augmented by an auxiliary loop to account for nonlinearities in the system. As a result, on-line application of the proposed control strategy requires the computation of only auxiliary loop using JITL technique. In addition, the adaptive capability of JITL is illustrated. This adaptive feature of JITL algorithm makes JITL a better candidate than the previously proposed Volterra, Functional Expansion and NN models in the partitioned model inverse based nonlinear IMC scheme. Furthermore, this control strategy is extended to MIMO nonlinear systems that operate over a range of operating regime. Simulation results confirm that the proposed control strategy tracks the reference trajectory, which is the benchmark performance for IMC design, better than its conventional counterparts.

In other approach, a memory-based IMC design using JITL is proposed for SISO nonlinear systems. Since a simple ARX model can be chosen in JITL, the inverse of this model can be easily obtained to get control law as required in IMC

design. Hence, JITL is employed to update model parameters as well as control law. Furthermore, IMC filter parameter is initialized using the most current controller database at each sampling instant, and is subsequently adjusted on-line by using steepest descent gradient method and current process information. Simulation results illustrate that the proposed memory-based IMC design method has better setpoint tracking and disturbance rejection performance than the PI/PID controller reported in the literature.

The suggested future work includes following points. First, like many previous work in IMC design, the proposed IMC control strategies do not take the input constraint into account. Therefore, an extension of the current work to model predictive control framework may provide a possible solution to this design issue. Next, similar to other model-based controller design methods, IMC design involves two design steps i.e. identification of a model and controller design based on this model. Our methods are no exception since the IMC scheme has been employed, despite that our methods are data-based ones. Thus, one research direction that warrants further investigation is to exploit *model-free* data-based controller design directly from process input-output data such that the controller design can do without the need of identifying a process model and can be performed in one single step.

REFERENCES

Aha, D. W., D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, pp. 37-66, 1991.

Andersen, H. W., K. H. Rasmussen, and S. B. Jorgensen, "Advances in process identification," *Proceedings of the fourth international conference on chemical process control*, pp. 237-269, 1991.

Aoyama, A., F. J. Doyle, and V. Venkatasubramanian, "Control-affine fuzzy neural network approach for nonlinear process control," *J. Proc. Cont.*, vol. 5, no. 6, pp. 375-386, 1995.

Atkeson, C. G., A. W. Moore, and S. Schaal, "Locally weighted learning," *Artificial Intelligence Review*, 11, pp. 11-73, 1997.

Bhat, N. V., and T. J. McAvoy, "Use of neural nets for dynamic modeling and control of chemical process systems," *Comp. & Chem. Engng.*, vol. 14, pp. 573-582, 1990.

Bontempi, G., H. Bersini, and M. Birattari, "The local paradigm for modeling and control: from neuro-fuzzy to lazy learning," *Fuzzy Sets and Systems*, vol. 121, pp. 59-72, 2001.

Bontempi, G., M. Birattari, and H. Bersini, "Lazy learning for local modeling and control design," *Int. J. Control*, vol. 72, no. 7/8, pp. 643-658, 1999.

Boyd, S., and L. Chua, "Fading memory and the problem of approximating nonlinear operators with Volterra series," *IEEE Trans. Circuits Systems*, vol. 32, pp. 1150-1161, 1985.

Braun, M. W., D. E. Rivera, and A. Stenman, "A model-on-demand identification methodology for nonlinear process systems," *Int. J. Control*, vol. 74, no. 18, pp. 1708-1717, 2001.

Calvet, J., and Y. Arkun, “Feedforward and feedback linearization of nonlinear systems and its implementation using internal model control (IMC),” *Ind. Eng. Chem. Res.*, vol. 27, pp.1822-1831, 1988.

Cheng, C., and M. S. Chiu, “A new data-based methodology for nonlinear process modeling,” *Chem. Eng. Sci.*, 59, pp. 2801-2810, 2004.

Chiu, M. S., and Y. Arkun, “Parametrization of all stabilizing decentralized IMC controllers and a sequential stabilization procedure,” in *Proc. of ACC*, Pittsburgh, PA, pp. 554-559, 1989.

Chiu, M. S., and Y. Arkun, “A methodology for sequential design of robust decentralized control systems,” *Automatica*, vol. 28, no. 5, pp. 997-1001, 1992.

Congalidis, J., J. Richards, and W. H. Ray, “Feedforward and feedback control of a copolymerization reactor,” *AIChE J.*, 35, pp. 891-907, 1989.

Cybenko, G., “Just-in-Time learning and estimation,” in *Identification, Adaptation, Learning: The science of learning models from data*, S. Bittanti, G. Picci, Eds. New York: Springer, pp. 423-434, 1996.

Daoutidis, P., M. Sorousch, and C. Kravaris, “Feedforward/feedback control of multivariable nonlinear processes,” *AIChE J.*, 36, pp. 1471-1484, 1990.

Doyle, F. J., B. A. Ogunnaike, and R. K. Pearson, “Nonlinear model-based control using second-order Volterra models,” *Automatica*, vol. 31, no. 5, pp. 697-714, 1995.

Economou, C. G., and M. Morari, “Newton control laws for nonlinear controller design,” in *Proc. IEEE Conf. on Decision Control*, Ft. Lauderdale, p. 1361, 1985.

Economou, C. G., M. Morari, and B. O. Palsson, “Internal model control. 5. Extension to nonlinear systems,” *Ind. Eng. Chem. Proc. Des. Dev.*, vol. 25, p. 403-411, 1986a.

Economou, C. G., and M. Morari, “Internal model control. 6. Multiloop design,” *Ind. Eng. Chem. Proc. Des. Dev.*, vol. 25, p. 411-419, 1986b.

Engell, S., and K. U. Klatt, “Nonlinear control of a nonminimum phase CSTR,” in *Proc. of ACC*, pp. 2941-2945, 1993.

Fisher, D. G., “Process control: an overview and personal perspective,” *The Canadian J. Chemical Engineering*, vol. 69, no. 1, pp. 5-26, 1991.

Garcia, C. E., and M. Morari, “Internal model control. 1. A unifying review and some new results,” *Ind. Eng. Chem. Proc. Des. Dev.*, vol. 21, no. 2, pp. 308-323, 1982.

Garcia, C. E., D. M. Prett, and M. Morari, “Model predictive control: theory and practice - a survey,” *Automatica*, vol. 25, no. 3, pp. 335-348, 1989.

Grosdidier, P., and M. Morari, “Interaction measures for systems under decentralized control,” *Automatica*, vol. 22, no. 3, pp. 309-319, 1986.

Harris, K. R., and A. Palazoglu, “Studies on the analysis of nonlinear processes via functional expansions – III: controller design,” *Chem. Eng. Sci.*, vol. 53, no. 23, pp. 4005-4022, 1998.

Harris, K. R., and A. Palazoglu, “Control of nonlinear processes using functional expansion models,” *Comp. & Chem. Engng.*, 27, pp. 1061-1077, 2003.

Henson, M. A., and D. E. Seborg, “An internal model control strategy for nonlinear systems,” *AIChE J.*, vol. 37, no. 7, pp. 1065-1078, 1991.

Hirschorn, R. M., “Invertibility of nonlinear control systems,” *SIAM J. Cont. Optimiz.*, vol. 17, p. 289, 1979.

Hunt, K. J., and D. Sbarbaro, “Neural networks for nonlinear internal model control,” *IEE Proc. Pt. D.*, vol. 138, no. 5, pp. 431-438, 1991.

Kantor, J., “Stability of state feedback transformations for nonlinear systems – some practical considerations,” in *Proc. of ACC*, pp. 1014-1016, 1986.

Kravaris, C., and J. C. Kantor, “Geometric methods for nonlinear process control: I. background,” *Ind. Eng. Chem. Res.*, vol. 29, pp. 2295-2310, 1990a.

Kravaris, C., and J. C. Kantor, “Geometric methods for nonlinear process control: II. controller synthesis,” *Ind. Eng. Chem. Res.*, vol. 29, pp. 2310-2323, 1990b.

Kravaris, C., and P. Daoutidis, “Nonlinear state feedback control of second-order nonminimum-phase nonlinear systems,” *Comp. & Chem. Engng.*, vol. 14, pp. 439-449, 1990.

Li, W. C., L. T. Biegler, C. G. Economou, and M. Morari, “A constrained pseudo-newton control strategy for nonlinear systems,” *Comp. & Chem. Engng.*, vol. 14, p. 451, 1990.

Ljung, L., *System Identification Theory for the User*. Second ed., Prentice Hall, 1999.

Maksumov, A., D. J. Mulder, K. R. Harris, and A. Palazoglu, “Experimental application of partitioned model-based control to pH neutralization,” *Ind. Eng. Chem. Res.*, vol. 41, no. 4, pp. 744-750, 2002.

Maner, B. R., F. J. Doyle, B. A. Ogunnaike, and R. K. Pearson, “Nonlinear model predictive control of a simulated multivariable polymerization reactor using second-order Volterra models,” *Automatica*, vol. 32, no. 9, pp. 1285-1301, 1996.

Maner, B. R., and F. J. Doyle, “Polymerization reactor control using autoregressive-Plus Volterra-based MPC,” *AIChE J.*, vol. 43, pp. 1763-1784, 1997.

Morari, M., and E. Zafiriou, *Robust Process Control*. Prentice-Hall, Englewood Cliffs, NJ, 1989.

Myers, R. H., *Classical and Modern Regression with Applications*. PWS-Kent Publ., Boston, MA, 1990.

Nahas, E. P., M. A. Henson, and D. E. Seborg, “Nonlinear internal model control strategy for neural network models,” *Comp. & Chem. Engng.*, vol. 16, no. 12, pp. 1039-1057, 1992.

Nikolaou, M., and V. Manousiouthakis, “A hybrid approach to nonlinear system stability and performance,” *AIChE J.*, vol. 35, p. 559, 1989.

Ogunnaike, B. A., and R. A. Wright, “Industrial application of nonlinear control,” *Proceedings of the fifth international conference on chemical process control*, pp. 46-59, 1996.

Pearson, R. K., and B. A. Ogunnaike, “Nonlinear process identification,” in *Nonlinear Process Control*, M. A. Henson, D. E. Seborg, Eds. Upper Saddle River, NJ: Prentice Hall, 1997.

Schetzen, M., *The Volterra and Wiener Theories of Nonlinear Systems*. Wiley, New York, 1980.

Seborg, D. E., T. F. Edgar, and D. A. Mellichamp, *Process Dynamics and Control*. Wiley, New York, 1989.

Shaw, A. M., F. J. Doyle, and J. S. Schwaber, “A dynamic neural network approach to nonlinear process modeling,” *Comp. & Chem. Engng.*, vol. 21, no. 4, pp. 371-385, 1997.

Skogestad, S., and M. Morari, “Robust performance of decentralized control systems by independent design,” *Automatica*, vol. 25, no. 1, pp. 119-125, 1989.

Su, H. T., and T. J. McAvoy, “Artificial neural networks for process identification and control,” in *Nonlinear Process Control*, M. A. Henson, D. E. Seborg, Eds. Upper Saddle River, NJ: Prentice Hall, 1997.

van de Vusse, J., “Plug-flow type reactor versus tank reactor,” *Chem. Eng. Sci.*, vol. 19, pp. 994-997, 1964.

Viswanadham, N., and J. H. Taylor, “Sequential design of large-scale decentralized control systems,” *Int. J. Control*, vol. 47, pp. 257-279, 1988.

Xiong, Q., and A. Jutan, “Grey-box modeling and control of chemical processes,” *Chem. Eng. Sci.*, vol. 57, pp. 1027-1039, 2002.