DATA-BASED METHODS FOR MODELING, CONTROL AND MONITORING OF CHEMICAL PROCESSES

CHENG CHENG

NATIONAL UNIVERSITY OF SINGAPORE
2006

DATA-BASED METHODS FOR MODELING, CONTROL AND MONITORING OF CHEMICAL PROCESSES

CHENG CHENG
(B. Eng., ECUST, China)
(M. Eng., ECUST, China)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF CHEMICAL AND BIOMOLECULAR ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2006

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my research supervisor, Dr. Min-Sen, Chiu for this excellent guidance and valuable ideas. I am indebted to him for providing me advices not only in the academic research but also my daily life. My special thanks to Dr. Chiu for his invaluable time for reading and revising this manuscript.

I am also thankful to Dr. Rangaiah, Dr. Lakshminarayanan, and Dr Wang Qing-Guo for their valuable advices to my research work. Special thanks and appreciation are due to Zhuang Hualiang, Ye Myint Hlaing, Yasuki Kansha, and Ankush Kalmukale for the stimulating discussions that we have had and the help that they have rendered to me. I would like to express my special words of gratitude to Mr. Jimmy Goh for understanding and providing me support when I worked as a part time student in NUS. I would also wish to thank Ms Tay Choon Yen, Mdm Fam Hwee Koong, Mdm Khoh Leng Khim, and Mdm Siew Woon Chee for the efficient and prompt help. I am also indebted to the National University of Singapore for providing me the excellent research facilities and research scholarships.

I cannot find any words to thank my hubby and my parents for their unconditional support, affection and encouragement, without which this research work would not have been possible.

TABLE OF CONTENTS

SUMMARY

"Data rich but information poor" is a common problem for most chemical processes. Therefore, how to extract useful information from data for the purposes of process modeling, control, and monitoring is one of the challenges in chemical industries. In this thesis, a new just-in-time learning (JITL) modeling methodology has been proposed to deal with this problem and the JITL based design methods for controller design and process monitoring have been developed. The main contributions of this thesis are as follows.

First, an enhanced JITL methodology is proposed by using both distance measure and angle measure to evaluate the similarity between two data samples, which is not exploited in the conventional JITL methods. In addition, parametric stability constraints are incorporated into the proposed method to address the stability of local models. Furthermore, a new procedure of selecting the relevant data set is proposed. Simulation studies illustrate that the proposed method gives marked improvement over its conventional counterparts in nonlinear process modeling. It is also demonstrated that the proposed method can be made adaptive online readily by simply adding the new process data to the database.

Second, based on the enhanced JITL technique, a robust controller design methodology is proposed for processes with moderate nonlinearity. Assuming that process nonlinearity is the only source of the model uncertainty, a composite model consisting of a nominal ARX model and JITL, where the former is used to capture the linear process dynamics and the latter to approximate the process nonlinearity, is employed to model the process behaviour in the operating space of interest. The state space realization of the resulting model is then reformulated as an uncertain system, by which the robust stability analysis of this uncertain system under PID control is

developed. Literature examples are employed to illustrate that the proposed methodology can be used to obtain the robust stability region in the parameter space of a PID controller, which assures the closed-loop stability for controlling the nonlinear process in the concerned operating space.

Next, by incorporating the JITL into the controller design, three data-based controller design methods are proposed: adaptive single-neuron (ASN) controller, adaptive IMC controller, and auto-tuning PID controller. ASN controller uses a single neuron to mimic the traditional PID controller. The ASN controller can control the unknown nonlinear dynamic process adaptively through the updating of controller parameters by the adaptive learning algorithm developed and the information provided from the JITL. Adaptive IMC controller integrates the JITL into the IMC framework. The controller parameters are updated not only based on the information provided by the JITL, but also its filter parameter is adjusted online by an adaptive learning algorithm. In the auto-tuning PID controller, a controller database is constructed to store the known PID parameters with their corresponding information vectors, while another database is employed for the standard use by JITL technique for modeling purpose. The PID parameters are automatically extracted from controller database according to the current process dynamics characterized by the information vector at every sampling instant. Moreover, the PID parameters thus obtained can be further fine-tuned, whenever necessary, and the resulting updated PID parameters with their corresponding information vector are stored into the controller database. These controller design methods exploit the current process information from JITL to realize online tuning controller parameters for nonlinear process control. Because of the parsimonious design framework, these adaptive controllers can be implemented online without heavy computational burden. Simulation results demonstrate that the

proposed controllers give better control performance than their conventional counterparts.

Last, by integrating JITL and principal component analysis (PCA) into a JITL-PCA monitoring scheme, a new monitoring method is proposed for dynamic nonlinear process. JITL serves as the process observer to account for the nonlinear dynamic behavior of the process under normal operating conditions. The residuals resulting from the difference between JITL's predicted outputs and process outputs are analyzed by PCA to evaluate the status of the current process operating conditions. Simulation results show that JITL-PCA gives marked improvement over PCA and DPCA in the monitoring of nonlinear static or dynamic systems.

NOMENCLATURE

A_i , B_i , C_i , D_i	State space model parameters
C_A , C_{Af} , C_B , C_C ,	Concentrations
C_I , $C_{I_{in}}$	Initiator concentration
C_m , $C_{m_{in}}$	Monomer concentration
d_i	Distance between $\mathbf{x}_i$ and $\mathbf{x}_q$
E_i , F_i	Model uncertainty part
F , F_I	Flow rate
f	Low-pass filter
f^*	Parameter of polymerization reaction
G	Process
$\tilde{G}$	Model of the process
$\tilde{G}_-^{-1}$	Minimum phase of $\tilde{G}$
K_d , K_i , K_p	Auto-tuning PID controller parameters
$\mathbf{K}(k)$	Auto-tuning PID controller parameter vector
k_{I} , k_{f_m} , k_p , k_{T_c} , k_{T_d}	Parameters of polymerization reaction
k_1 , k_2 , k_3	Kinetic parameters of van de Vusse reactor
$k_{\min}$, $k_{\max}$	Number of minimum and maximum relevant data sets
L	Level of reactor
l	Number of nearest neighbours
P	Regression matrix
$\mathbf{P}_k$	Principal component loading matrix
M_m	Molecular weight of monomer
Q	Statistical Variable of PCA
r	Set-point
s_i	Similarity number
T , T_i	Reactor Temperatures
T^2	Statistical Variable of PCA

u	Process input
V	Reactor volume
W	Weight matrix
$w_1,\ w_2,\ w_3$	Parameters of ASN controller
$\mathbf{x}_i, \mathbf{x}_q$	Information and query vector
y	Process output
y_l	Output of the nominal ARX model
y_{nl}	Nonlinear effect of process
$\hat{y}$	Model prediction
$\mathbf{z}$	Regression vector

Greek Symols

$\alpha_1,\ \alpha_2,\ \beta_1$	ARX model parameters
$\alpha_{l,1},\ \alpha_{l,2},\ \beta_{l,1}$	Nominal ARX model parameters
γ	JITL algorithm parameter
$\delta_1,\ \delta_2,\ \delta_3,\ \delta_a$	Model uncertainty
ε	Threshold of auto-tuning PID algorithm
$\eta,\ \eta_i$	Adaptive learning rate
θ_i	Angle between $\Delta\mathbf{x}_i$ and $\Delta\mathbf{x}_q$
κ	Weight factor of objective functions
λ	IMC filter time constant
ρ	Average density
$\sigma(k)$	Information vector of auto-tuning PID
$\tau_I,\ \tau_D$	PID controller parameters

Abbreviations

ARX	Autoregressive exogenous
CSTR	Continuous stirred tank reactor
DPCA	Dynamic principal component analysis

IMC	Internal model control
JITL	Just-in-time learning
MAE	Mean absolute error
MSE	Mean squared error
MIMO	Multi-input multi-output
NN	Neural network
PCA	Principal component analysis
PID	Proportional-integral-derivative
SISO	Single input single-output

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Motivations

Thanks to the development of advanced control techniques and computer technologies, spectacular progresses have been achieved in process control during the last two decades. However, with the market competition getting more intense than before, growing demands for improving performance of process still stimulate researchers to develop more efficient and reliable methods for process modeling, control and monitoring.

In chemical industries, hundreds or even thousands of variables, such as flow rate, temperature, pressure, levels and compositions are routinely measured and automatically recorded in historical databases for the purposes of process control, online optimization or monitoring. Despite that significant potential benefits may be gained from the database, it is generally not a trivial task to extract useful information and knowledge from the databases. Therefore, most chemical processes face a well-known problem, i.e., “data rich but information poor”. Thus how to extract relevant

information from data to better understand process behavior becomes a significant research topic for chemical industries. On the other hand, an accurate process model can improve the performance of many advanced control and monitoring methods. However, model development represents 75% of the cost of developing advanced process control design (Nelles, 2001). Moreover, for most chemical processes, detailed first-principle models are often unavailable or too costly and tedious to build. In this respect, data-based methods capable of extracting the information from process data for process modeling, control, and monitoring become an attractive alternative.

During last two decades, several data-based methods are proposed for nonlinear system modeling (Pearson, 1999; Nelles, 2001), for example, artificial neural network (ANN) and neuro-fuzzy network, Volterra series or other various orthogonal series models (Nelles, 2001). However, when dealing with large sets of data, these approaches becomes less attractive because of the difficulties in specifying model structure and the complexity of the associated optimization problems, which are usually highly non-convex. Because of these restrictions, most nonlinear controller design methods based on ANNs or neuro-fuzzy networks require complicated control structure and heavy computation. To alleviate the aforementioned problems, the just-in-time learning (JITL) modeling technique (Cybenko, 1996) was recently proposed. It is also known as instance-based learning (Aha et al., 1991), local weighted model (Atkeson et al., 1997), lazy learning (Aha, 1997; Botempi et al., 2001), or model-on-demand (Braun et al., 2001; Hur et al., 2003) in the literature. JITL not only needs lesser a priori knowledge to initialize but also is inherently adaptive and thus it can be readily updated online. In contrast, ANN and neuro-fuzzy network need to be retrained from scratch. This is obviously not desirable if these models are to be used in model based controller design or monitoring method.

However, the existing JITL algorithms do not exploit the available information of angular relationship between two data samples, which may hamper the effectiveness of the existing JITL methods. Thus we aim to develop a new similarity criterion by incorporating the angle measure to improve the modeling accuracy of the JITL method. Furthermore, data-based methods for controller design and process monitoring by incorporating JITL are not well exploited in the literature. This motivates our research efforts to develop new JITL based design methods for adaptive and robust controller designs and process monitoring, which require less computational effort and simpler design framework.

1.2 Contributions

In this thesis, JITL based methods for process modeling, control and monitoring are studied and developed. The main contributions of this thesis are as follows.

First, a new JITL modeling methodology is proposed. In the method, both distance measure and angle measure are used to evaluate the similarity between two data samples, which is not exploited in the conventional methods. In addition, parametric stability constraints are incorporated into the proposed method to address the stability of local models. Furthermore, a new procedure of selecting the relevant data set is proposed. Simulation results demonstrate that the proposed method has better predictive performance than its conventional counterparts.

Second, a robust controller design methodology is proposed based on a composite model that consists of a nominal ARX model and JITL, where the former is used to capture the linear process dynamics and the latter to approximate the nonlinearity of the processes, which is assumed to be the only source of the model

uncertainty. The state space realization of the resulting model is then reformulated as an uncertain system, by which the robust stability analysis of this uncertain system under PID control is developed by using the structured singular value analysis framework.

Next, by incorporating JITL into the controller design, three data-based adaptive controller design methods are proposed. The first design method is an adaptive single- neuron (ASN) controller, which uses a single neuron to mimic the traditional PID controller. The ASN controller can control the unknown nonlinear dynamic process adaptively through the updating of controller parameters by the adaptive learning algorithm developed and the information provided from the JITL. The next proposed design method is an adaptive IMC controller. By incorporating the JITL into IMC framework, the proposed controller parameters are updated not only based on the information provided by the JITL, but also its filter parameter is adjusted online by an adaptive learning algorithm. Last, an auto-tuning PID controller by employing two databases is proposed. A controller database is constructed to contain the known PID parameters and their corresponding information vectors for controller design purpose, while another database is employed for the standard use by JITL for process modeling purpose. During the on-line implementation, the controller database is used to extract the relevant information to obtain new PID parameters based on the current process dynamics characterized by the current information vector. Moreover, the new PID parameters thus obtained can be further updated on-line when the predicted control error is greater than a pre-specified threshold and the resulting updated PID parameters with their corresponding information vector are stored into the controller database. These control design methods exploit the current process information from process model database or/and controller database to realize online

tuning controller parameters for nonlinear process control. Because of the parsimonious design framework, these adaptive controllers can be implemented online without heavy computational burden.

Last, by integrating JITL and principal component analysis (PCA) into a JITL-PCA monitoring scheme, a new monitoring method is proposed for dynamic nonlinear process. JITL serves as the process observer to account for the nonlinear dynamic behavior of the process under normal operating conditions. The residuals resulting from the difference between the JITL's predicted outputs and process outputs are analyzed by PCA to evaluate the status of the current process operating conditions. Simulation results show that JITL-PCA gives marked improvement over PCA and dynamic PCA (DPCA) in the monitoring of nonlinear static or dynamic systems.

1.3 Thesis Organization

The thesis is organized as follows. Chapter 2 comprises the literature review of data-based methods for process modeling, control and monitoring. The comparison between the traditional learning methods and JITL technique is also discussed. In Chapter 3, a new JITL methodology augmented with an angle measure is proposed for nonlinear process modeling. A new methodology for robust controller design of nonlinear processes is developed in Chapter 4. Based on the JITL technique, Chapter 5 presents the ASN controller for nonlinear process control. By incorporating JITL into IMC framework, an adaptive IMC controller for nonlinear process control is developed in Chapter 6. The proposed auto-tuning PID controller is presented in Chapter 7. By integrating JITL and PCA into the proposed JITL-PCA monitoring framework, a new process monitoring methodology is developed in Chapter 8. Finally,

the general conclusions from the present work and suggestions for future work are given in Chapter 9.

Chapter 2

Literature Review

This chapter examines the research work that has been conducted in the field of data-based methods for process modeling, control and monitoring. An overview of the current progress of data-based methods is presented. A newly developed data-based method, just-in-time learning (JITL), will be discussed in detail and the possible applications of JITL for process modeling, control and monitoring will be discussed as well.

2.1 Nonlinear Process Modeling

Process models are undoubtedly fundamentally important for process control and monitoring because the performance of many advanced control and monitoring methods is based on the availability of accurate models. However, most chemical processes are multivariable and nonlinear in nature, and their dynamics can be time varying. Thus, first-principle models are often unavailable due to the lack of complete

physicochemical knowledge of chemical processes. An alternative approach is to develop data-based methods to extract models from process data measured in industrial processes when very little a priori knowledge is available. Recently, various data-based methods for nonlinear process modeling have been proposed (Pearson, 1999; Nelles, 2001). These methods can be classified into two groups. One is standard-learning approach, which usually follows the modeling building procedure: 1) collect data from process; 2) use different methodologies to determine the model structures and initial model parameters; 3) fix the model parameters by optimization techniques (Nelles, 2001). Another attractive data-based approach is just-in-time learning (JITL) technique, which requires little a priori information and needs significantly less effort for online adaptation of the model as compared with standard-learning methods mentioned above. The following subsections will discuss these two approaches for nonlinear process modeling.

2.1.1 Standard-learning methods

The standard-learning approach includes NARMAX (nonlinear autoregressive moving average models with exogenous inputs), Volterra models, Wiener models, Hammerstein models, neural networks, neuro-fuzzy networks, and wavelets. Among these methods, neural networks and neuro-fuzzy networks are the most popular approaches for nonlinear process modeling. Therefore, we will review the other methods briefly in the sequel, followed by a discussion of the neural network and neuro-fuzzy network in detail.

NARMAX models are identified from input/output data using a conventional least-square fitting procedure, which have proved to be versatile and useful empirical models for industrial processes (Pearson, 1999). However, like other empirical

models, it is not easy to choose nonlinear model structure and select nonlinear function approximation for NARMAX models (Pearson, 1995).

Volterra model considers the cross terms between the past inputs in the manner of convolution models. A large number of coefficients are required for modeling purposes. In order to decrease the complexity of Volterra model, Maner et al. (1996) suggested the use of an autoregressive plus Volterra based model in the model-based control strategies.

Wiener and Hammerstein models use special types of nonlinear models composed of linear and nonlinear blocks cascaded in series. For Hammerstein models, the static nonlinear block precedes the linear dynamic element, while Wiener models consist of the linear model followed by the nonlinear block. Various methods have been proposed to identify the Hammerstein models e.g., the iterative method (Narendra and Gallman, 1966), the over-parameterization method (Chang and Luus, 1971; Bai, 1998), and multivariate statistical method (Lakshminarayanan et al., 1995). Identification of Wiener models is more difficult due to the lack of a good representation of the output nonlinearity for identification purpose. The main approach used for Wiener model identification is the stochastic method (Bilings and Fakhouri, 1978; Wigren, 1994). Kalafatis et al. (1997) proposed inverse representation method to identify Wiener models.

Most of the methods mentioned above share the same shortcoming of the lacking of a straightforward procedure to select a nonlinear model structure. In addition, all the methods are global modeling methods, which are difficult to handle large amount of data. (Nelles, 2001)

Neural networks (NNs) can provide an excellent framework for modeling the nonlinear systems because of their capability of approximating any smooth function to an arbitrary degree of accuracy with a certain number of hidden layer neuron (Hornik et al., 1989). The NNs as shown in Figure 2.1 are feedforward neural networks that consist of neurons arranged in layers, which are connected via weight parameters such that the signals at the input are propagated through the network to the output.

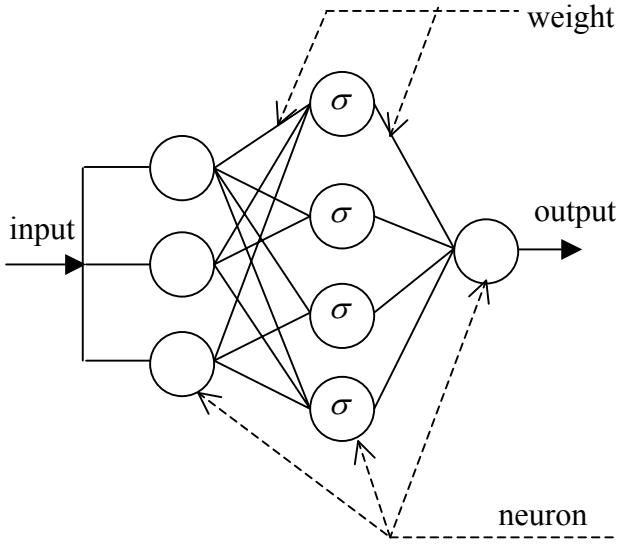


Figure 2.1 Structure of a multi-layer feedforward network

Through the weight parameters, the input of each neuron is computed as the weighted sum of the outputs from the neurons in the preceding layer. The output of each neuron is computed by a transfer function to yield the non-linear behavior of the networks. The most popular functions are the sigmoid function $\sigma(x) = \frac{1}{1-e^{-x}}$ and radial basis function $\sigma(x) = e^{-x^2}$, where x is the input of each neuron. Another

function is wavelet-basis function $\psi_{a,b}(x) = |a|^{-0.5}\,\psi(\frac{x-b}{a})$, where a and b are the dilation and translation parameters respectively, that has multi-resolution capabilities to enhance the modeling capability of the resulting network (Zhang, 1997).

During the training of neural network, the weights are adjusted and learned from a given set of data aiming to achieve the 'best' approximation of the behavior of the system. For modeling the dynamic systems, the output of the neural network can be represented by:

$$\hat{y}(k) = g(y(k-1),\cdots,y(k-n_y),u(k-1-n_d),\cdots,u(k-n_u-n_d)) \tag{2.1}$$

where $\hat{y}(k)$ is the output of neural network at the k-th sampling instant, y is the system output, u is the system input, n_y and n_u are integers related to the system's order, n_d is the time delay, and g is the unknown nonlinear function to be approximated by the neural network.

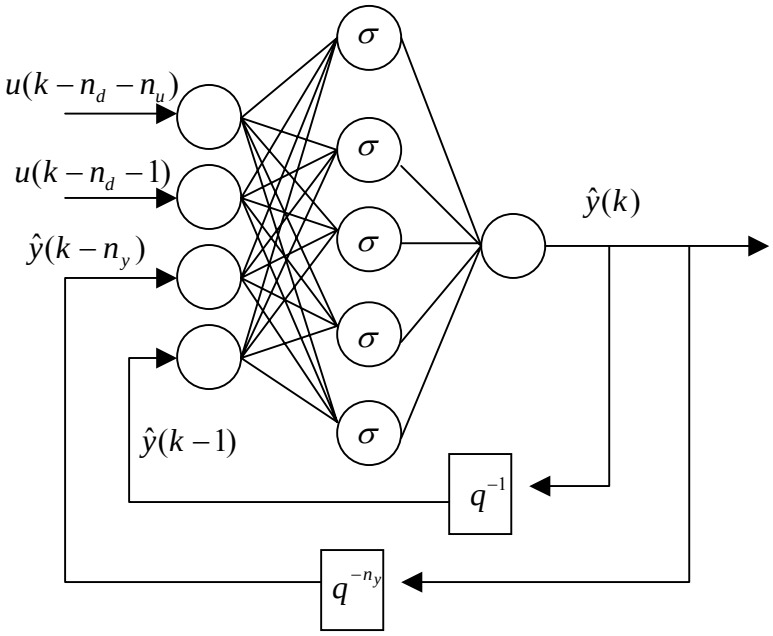


Figure 2.2 Structure of a recurrent neural network

Another commonly used neural network is the recurrent neural network as depicted in Figure 2.2. The advantage of the recurrent network over the feedforward network is its better capability for process long term prediction and thus it is more suitable for predictive control application (Su and McAvoy, 1992, 1997). Mathematically, the output of recurrent network is described by

$$\hat{y}(k) = g(\hat{y}(k-1), \cdots, \hat{y}(k-n_y), u(k-1-n_d), \cdots, u(k-n_u-n_d)) \tag{2.2}$$

Chen et al. (1990) first used neural network for nonlinear dynamic modeling. For chemical engineering, Bhat and McAvoy (1990) employed neural network to model pH neutralization processes. Recent works have focused on improving methods for selection of initial network parameter, the selection of neural network structure, and the stability of the resulting models (Nikravesh, 1997; Shaw, 1997). These various methods attempt to avoid random initialization and trial-and-error efforts, which are usually adopted in determining the network structure and parameters.

Fuzzy set theories and neural network technologies are integrated together to construct the neuro-fuzzy networks. By using the learning capability of the neural networks, neuro-fuzzy networks can identify fuzzy rules and optimize membership function of fuzzy model (Lin and Lee, 1991; Jang, 1993; Jang and Sun, 1995). In the context of neuro-fuzzy network, the fuzzy model commonly used is the Takagi-Sugeno (T-S) fuzzy model (Takagi and Sugeno, 1985). In T-S model, the rule antecedents describe fuzzy region in the input space and the rule consequents are crisp function of the model inputs:

R_i: IF $x_1 = A_{i1}$ AND … AND $x_m = A_{im}$ THEN $y_i = f_i(x_1, x_2, \cdots, x_m)$, $i = 1,2,\cdots,r$

where R_i represents the *i*-th rule, $x_1, \cdots, x_m$ are the inputs of fuzzy system, A_{ij} denotes the fuzzy set used for input x_j in the *i*-th rule, f_i is a crisp function of the

input vector $[x_1, \cdots, x_m]$, and r is the number of rules. Normally, the consequents employ a liner model, i.e., $y_i = \sum_{j=1}^{m} w_j^i x_j + b^i$, where w_j^i and b^i are the model parameters.

The output of the model is calculated by the center of gravity defuzzification as follows:

$$y = \frac{\sum_{i=1}^{r} \mu_i y_i}{\mu_i} \tag{2.3}$$

where μ_i is the membership of the *i*-th rule antecedent.

Applying T-S model to describe dynamic system is equivalent to dividing the operating space of a dynamic system into several local operating regions. Within each local region, one fuzzy rule R_i is used to represent the process behavior. To do so, the consequent of the fuzzy rule employ a linear dynamic model:

R_i : IF operating condition i , THEN

$$y_i(k) = l_i(y(k-1), \cdots, y(k-n_y), u(k-1-n_d), \cdots, u(k-n_u-n_d)) \qquad i = 1,2,\cdots,r \tag{2.4}$$

where l_i is a linear function. The final model output is obtained by Eq (2.3).

In the past decade, neuro-fuzzy networks have been extensively studied. Lin and Lee (1991) proposed a three-phase learning algorithm. In the first phase, the self-organizing map (Kohonen, 1995) is applied to obtain the structure and parameters of the fuzzy model. In the second phase, a competitive learning technique is employed to find the rules. Lastly, backpropagation algorithm is used to fine-tune the model parameters. Jang (1993) developed an Adaptive-Network-Based Fuzzy Inference System (ANFIS) that can construct an input-output mapping based on both human knowledge and input-output data. Zhang and Morris (1999) proposed a recurrent

neuro-fuzzy network by the external feedback of the network's outputs, whereas Mastorocostas and Theocharis (2002) introduced internal feedback to build a recurrent neuro-fuzzy network.

It is noted that neural network and neuro-fuzzy methods suffer from the drawbacks of requiring a priori knowledge to determine the model structures and complicated training strategy to determine the optimal parameters of the models. In addition, both methods are difficult to be updated online when the process dynamics are moved away from the nominal operating space, where the retraining of neural network and neuro-fuzzy network is required. To alleviate these problems, JITL provide an attractive alternative approach, which will be introduced in the next subsection.

2.1.2 Just-in-time learning

Just-in-time learning (JITL) (Cybenko, 1996) was developed as an attractive alternative for modelling nonlinear systems. It is also known as instance-based learning (Aha et al., 1991), local weighted model (Atkeson et al., 1997), lazy learning (Aha, 1997; Bontempi et al., 2001; Bontempi and Birattari, 2005), or model-on-demand (Braun et al., 2001; Hur et al., 2003) in the literature. This approach was originally developed from machine learning field. A detailed survey of lazy learning is given in Aha (1997).

JITL assumes that all available observations are stored in a database, and the models are built dynamically upon query. Compared with other learning algorithms, JITL exhibits three main characteristics. First, the model-building phase is postponed until an output for a given query data is requested. Next, the predicted output for the query data is computed by exploiting the stored data in the database. Finally, the

constructed answer and any intermediate results are discarded after the answer is obtained (Atkeson et al., 1997; Bontempi et al., 2001; Nelles, 2001). Figure 2.3 illustrates the differences between the standard learning and the JITL method. Standard learning methods like NNs and neuro-fuzzy networks are typically trained offline. Thus, all learning data is processed a priori in a batch-like manner. This can become computationally expensive or even impossible for huge amounts of data, and therefore data reduction techniques may have to be applied. Additionally, online adaptation of NN and neuro-fuzzy network models requires model update from scratch, namely both network structure (e.g. the number of hidden layers in the former case and the number of the fuzzy rules in the latter) and model parameters may need to be changed simultaneously. Evidently, this procedure is not only time consuming but also will interrupt the plant operation, if these models are used for other purposes like model based controller design. In contrast, 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 should be noted that the JITL model 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. It is important to point out that the selection of relevant data is carried out individually for each incoming query data. This allows one to change the model architecture, model complexity, and the criteria for relevant data selection online according to the current situation (Nelles, 2001). Potentially, JITL is an attractive data-based approach.

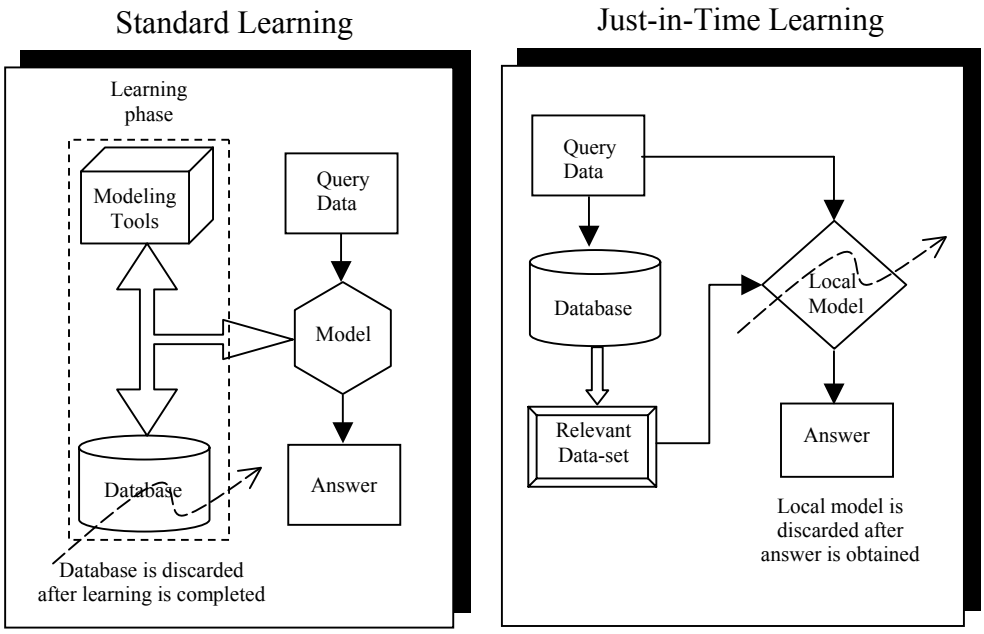


Figure 2.3 Comparison of JITL and standard-learning

2.2 Controller Design for Nonlinear Processes

In chemical and biochemical industries, most processes are inherently nonlinear, however most controller design techniques are based on linear control techniques. The prevalence of linear control strategies is partly due to the fact that, over the normal operating region, many of the processes can be approximated by linear models, which can be obtained by the well-established identification methods and the available input/output process data. In addition, the theories for the stability analysis of linear control systems is quite well developed so that linear control techniques are widely accepted. In contrast, controller design for nonlinear models is considerably more difficult than that for linear models. However, most chemical processes are nonlinear in nature, therefore linear control design methodologies may not be adequate to achieve a good control performance for these processes. This has

led to an increasing interest in the nonlinear controller design for the nonlinear dynamic processes.

In what follows, three control strategies, i.e. robust control, adaptive control, and nonlinear internal model control, capable of providing the improved performance for nonlinear systems are reviewed. Specifically, the mathematical tools introduced in robust control theory will pave the foundation for the proposed robust controller design given in Chapter 4, while three data-based control strategies by incorporating the JITL technique will be developed in Chapters 5 to 7 within the adaptive control and internal model control design frameworks.

2.2.1 Robust control

For most chemical processes, the first principle models are usually unavailable because of the lack of physicochemical knowledge. Therefore controller design has to rely on the models extracted from the process input/output measurements. These models generally have varying degrees of accuracy. If the plant/model mismatch is not taken into account in the controller design, the control performance may become poor and even the closed-loop stability cannot be guaranteed. This robust control problem has motivated the researchers to pursue various robust control designs in the last two decades (Malan et al., 2004). Robust control methodologies aim to design controllers, which maintain closed-loop stability and performance not only for nominal model of the process but also for a set of possible process models that capture the actual process dynamics. Normally, this set of process models is represented by the nominal model and pre-specified uncertainty (or perturbation) description equation, which is used to account for the plant/model mismatch or modeling error between the nominal model and a given set of process models.

The concept of robust control design is briefly given as follows. Without loss of generality, the input uncertainty description is assumed to describe the relation between the actual process $G(s)$ and the process model $G_m(s)$ as follows:

$$G(s) = G_m(s)(I + \Delta_m(s)) \tag{2.5}$$

where $\Delta_m(s)$ is bounded perturbation at the plant input and is constrained by

$$\left\|\Delta_m(j\omega)\right\|_2 = \bar{\sigma}(\Delta_m(j\omega)) < l_m(\omega),\ \forall\omega \tag{2.6}$$

where $\bar{\sigma}$ denotes the maximum singular value and $l(\omega)$ is a frequency dependent worst-case uncertainty weighting function. The choice of spectral norm is not only it describes well the effects of high frequency unmodeled dynamics and nonlinearities, but also it leads to mathematically tractable control problem (Toffner-Clausen, 1996). Small gain theorem shows that for the perturbation structure (2.5), the stability of the closed-loop system is maintained if and only if the feedback system is nominally stable and the feedback controller $K(s)$ satisfies the following robust stability condition:

$$\bar{\sigma}(K(j\omega)G_m(j\omega)(I + K(j\omega)G_m(j\omega))^{-1}) \le l_m(\omega),\ \forall\omega \tag{2.7}$$

However, the above formulation is restricted to the cases when the plant is subject to unstructured perturbation. Consequently, it would yield conservative design when multiple perturbations occur in the feedback control system or robust performance is considered as the design objective. To overcome this problem, a much more general design framework based on the structured singular value (μ) theory was developed (Packard and Doyle, 1993). Consider the feedback structure consisting of the system M and structured perturbation Δ as depicted in Figure 2.4, where the perturbation Δ is generally a norm bounded uncertainty block:

$$\Delta = diag[\delta_1 I_{r1}, \ldots, \delta_S I_{rS}, \Delta_{S+1}, \ldots, \Delta_{S+F}] \tag{2.8}$$

where $\delta_i \in C$, $\Delta_{S+j} \in C^{m_j \times m_j}$, $1 \le i \le S$, $1 \le j \le F$. Furthermore, Δ is constrained by

$$\bar{\sigma}(\Delta) \le \nu \tag{2.9}$$

In order to take into account the structure of the perturbation Δ, the structured singular value $\mu_\Delta(M)$ is defined such that $\mu_\Delta^{-1}(M)$ is equal to the smallest $\bar{\sigma}(\Delta)$ needed to make $(I - M\Delta)$ singular, i.e.

$$\mu_\Delta^{-1}(M) = \min_{\nu}\{\nu \mid \det(I - M\Delta) = 0\} \tag{2.10}$$

If no Δ exists such that $\det(I - M\Delta) = 0$, then $\mu_\Delta(M) = 0$. It is important to note that $\mu_\Delta(M)$ depends both on the matrix M and on the structure of the perturbation Δ. Based on the on-going discussion, the following theorem gives the robust stability condition for the feedback system given in Figure 2.4.

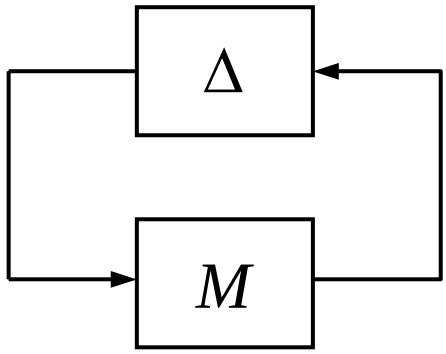


Figure 2.4 The *M*-Δ structure

Theorem 2.1: Assume that the nominal system $M(s)$ is stable, then the closed-loop system in Figure 2.4 is stable for all perturbations Δ if and only if

$$\mu_\Delta(M(j\omega)) < \nu^{-1}, \ \forall \omega \tag{2.11}$$

The definition (2.10) is not useful for computing μ since the optimization problem implied by it does not appear to be easily solvable. Fortunately, the computation of μ can be performed by solving its upper and lower bounds as given in the following inequality:

$$\max_{U} \rho(MU) \le \mu_{\Delta}(M) \le \inf_{D} \overline{\sigma}(DMD^{-1}) \tag{2.12}$$

where U is the set of all unitary matrices with the same block diagonal structure as Δ and D is the set of real positive diagonal matrices with the structure of each block opposite to that of the corresponding block in the perturbation Δ. More detailed discussion on the properties and computation of μ can be found in Morari and Zafiriou (1989) and Packard and Doyle (1993).

Doyle et al. (1989) used structured singular value approach to design robust controller for a CSTR based on the first principle model. In their work, it is assumed that the model uncertainty is due to the nonlinearities of the process, which can be described by the conic sector. Subsequently, the bounds of the conic sector are treated as the uncertainties in the robustness analysis under the structured singular value framework. However, the identification of the conic bounds is cumbersome, which requires careful observation of the nonlinearities to be bounded (Knapp and Budman, 2000). In addition, first-principle models are usually unavailable for most chemical processes. Knapp and Budman (2000) proposed an alternative methodology for the robust analysis for the nonlinear process based on the input and output data. To do so, a nonlinear autoregressive moving (NARMA) model is initially identified from the input and output data. Next, for the purpose of robustness analysis, a minimal state affine model realization of the identified NARMA model is obtained, by which the robust stability and performance conditions are derived as the design constraints for robust controller design. However, to obtain the state affine model from a NARMA

model is not a trivial task because one needs to find a suitable Volterra kernel from a given NARMA model so that a behavior matrix can be developed to obtain a state affine model.

To alleviate the aforementioned drawbacks in the state affine model employed in the work by Knapp and Budman (2000), robust controller design under JITL framework will be investigated in Chapter 4.

2.2.2 Adaptive control

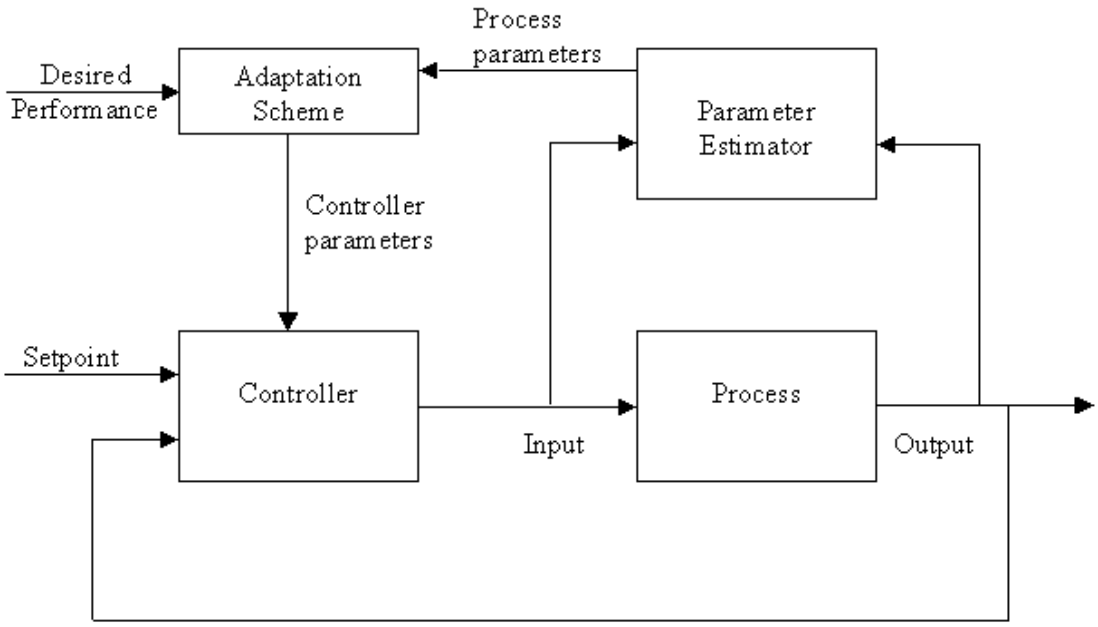


Figure 2.5 Diagram of adaptive control scheme

Research in adaptive control has a long and rich history. The development of adaptive control started in the 1950's with the aim of developing adaptive flight control systems. With the progressing of control theories and computer technology, various adaptive control methodologies were proposed for process control in the last

three decades. Astrom (1983), Seborg et al. (1986) and Astrom and Wittenmark (1995) gave detail reviews of the theories and application of adaptive control. Most adaptive methodologies integrate a set of techniques for automatic adjustment of controller parameters in real time in order to achieve or to maintain a desired level control performance when the dynamic characteristics of the process are unknown or vary in time. The diagram of adaptive control concept is depicted in Figure 2.5.

There are three main technologies for adaptive control: gain scheduling, model reference control, and self-tuning regulators. The purpose of these methods is to find a convenient way of changing the controller parameters in response to changes in the process and environment dynamics.

Gain scheduling is one of the earliest and most intuitive approaches for adaptive control. The idea is to find process variables that correlate well with the changes in process dynamics. It is then possible to compensate for process parameter variations by changing the parameters of the controller as function of the process variables. The advantage of gain scheduling is that the parameters can be changed quickly in response to changes in the process dynamics. It is convenient especially if the process dynamics in a well-known fashion on a relatively few easily measurable variables. Gain scheduling has been successfully applied to nonlinear control design for process industry (Astrom and Wittenmark, 1995). One drawback of gain scheduling is that it is open-loop compensation without feedback. Another drawback of gain scheduling is that the design is time consuming. A further major difficulty is that there is no straightforward approach to select the appropriate scheduling variables for most chemical processes.

Model reference control is a class of direct self-tuners since no explicit estimate or identification of the process is made. The specifications are given in terms

of "reference model" which tells how the process output ideally should respond to the command signal. The desired performance of the closed-loop system is specified through a reference model, and the adaptive system attempts to make the plant output match the reference model output asymptotically.

The third class of adaptive control is self-tuning controller. The general strategy of this controller is to estimate model parameters on-line and then adjust the controller settings based on current parameter estimate (Astrom, 1983). In the self-tuning controller, at each sampling instant the parameters in an assumed dynamic model are estimated recursively from input-output data and controller setting is then updated. The whole control strategy can be divided into three steps: first, information gathering of the present process behavior; second, control performance criterion optimization; and last, adjustment of the controller parameters. The first step implies the continuous determination of the actual condition of the process to be controlled based on measurable process input and output and appropriate approaches selected to identify the model parameters. Various types of model identification can be classified depending on the information gathered and the method of estimation. The last two steps evaluate the control loop performance and the decision as to how the controller will be adjusted or adapted. These characteristics make self-tuning controller very flexible with respect to its choice of controller design methodology and to the choice of process model identification (Seborg et al., 1986).

In the past two decades, many research efforts have focused on the development of intelligent control algorithms that can be applied to complex processes whose dynamics are poorly modeled and/or have severe nonlinearities. (Stephanopoulos and Han, 1996; Linkens and Nyongesa, 1996). Because neural networks have the capacity to approximate any nonlinear function to any arbitrary

degree of accuracy, NNs have received much attention in the area of adaptive control. Perhaps the most significant work of the application of NNs in adaptive control is that of Narendra and Parthasarathy (1990) who investigated adaptive input-output neural models in model reference adaptive control structures. Hernandez and Arkun (1992) studied control-relevant properties of neural network model of nonlinear systems. Jin et al. (1994) used recurrent neural networks to approximate the unknown nonlinear input-output relationship. Based on the dynamic neural model, an extension of the concept of the input-output linearization of discrete-time nonlinear systems is used to synthesize a control technique under model reference control framework. te Braake et al. (1998) provided a nonlinear control methodology based on neural network combined with feedback linearization technique to transform the nonlinear process into an equivalent linear system in order to simplify the controller design problem. Recently, some researchers have constructed stable NN for adaptive control based on Lyapunov's stability theory (Lewis et al., 1996; Polycarpou, 1996; Ge et al., 2002). One main advantage of these schemes is that the adaptive laws are derived based on the Lyapunov synthesis method and therefore guarantee the stability of the control systems.

While neuro-control techniques are suited to control an unknown nonlinear dynamic process, it is generally difficult to present the control law in simple analytical form. Also, a nonlinear optimization routine is required to determine the control input, which may lead to the problems of large computational efforts and poor convergence. For chemical process control, control strategy has to be implemented in real time, so it is desirable to keep the control algorithm as simple as possible. Therefore, it still remains a challenging task to acquire a simple and easy-to-implement adaptive control strategy for nonlinear process control. In this research, adaptive control

strategies based on the JITL technique will be developed to deal with the problem mentioned above.

2.2.3 Nonlinear internal model control (NIMC)

Internal Model Control (IMC) proposed by Garcia and Morari (1982) is a powerful controller design strategy for the open-loop stable dynamic systems (Morari and Zafiriou, 1989). This is mainly due to two reasons. First, integral action is included implicitly in the controller because of the IMC structure. Moreover, plant/model mismatch can be addressed via the design of the robustness filter. IMC design is expected to perform satisfactorily as long as the process is operated in the vicinity of the point where the linear process model is obtained. However, the performance of IMC controller will degrade or even become unstable when it is applied to nonlinear processes with a range of operating conditions.

To extend the IMC design to nonlinear processes, various nonlinear IMC schemes have been developed in the literature. For instance, Economou et al. (1986) provided a nonlinear extension of IMC by employing contraction mapping principle and Newton method. However, this numerical approach to nonlinear IMC design is computationally demanding. Calvet and Arkun (1988) used an IMC scheme to implement their stat-space linearization approach for nonlinear systems with disturbance. A disadvantage of the state-space linearization approach is that an artificial controlled output is introduced in the controller design procedure and cannot be specified a priori. Another drawback of this method is that the nonlinear controller requires state feedback (Henson and Seborg, 1991a). Henson and Seborg (1991b) proposed a state-space approach and used nonlinear filter to account for plant/model mismatch. However, their method relied on the availability of a nonlinear state-space

model, which may be time-consuming and costly to obtain. Doyle et al. (1995) proposed a partitioned model inverse controller to obtain the nonlinear model inversion. This controller synthesis scheme based on Volterra model retains the original spirit and characteristics of conventional 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 'correction'. However, Volterra model derived using local expansion results such as Carleman linearization is accurate for capturing local nonlinearities around an operating point, but may be erroneous in describing global nonlinear behavior (Maner et al., 1995). Another drawback of this method is that parameters of second-order Volterra model are not parsimonious to describe the process nonlinearities. Harris and Palazoglu (1998) proposed another nonlinear IMC scheme based on the functional expansion models instead of Volterra model. However, functional expansion models are limited to fading memory systems and consequently, the resulting controller gives satisfactory performance only for a limited range of operation.

The ability of artificial neural networks to model almost any nonlinear function without a priori knowledge has lead to the investigation of nonlinear IMC schemes using neural networks (NN). In the earlier methods given in Bhat and McAovy (1990) and Hunt and Sbarbaro (1991), two NN were used in the IMC framework, where one NN was trained to represent the nonlinear dynamics of process, which was used as the IMC model, while another NN was trained to learn the inverse dynamics of the process and was employed as the nonlinear controller. Because IMC model and controller were built by separate neural networks, the controller might not invert the steady-state gain of the model and thus steady-state offset might not be

eliminated (Nahas et al., 1992). Moreover, these control schemes do not provide a tuning parameter that can be adjusted to account for plant/model mismatch. Nahas et al. (1992) developed another NN based nonlinear IMC strategy, which consists of a model inverse controller obtained from a neural network and a filter with a single tuning parameter.

However, the above nonlinear IMC designs 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 of process dynamics for the necessary construction of nonlinear process inverses. To overcome these difficulties, Shaw et al. (1997) used recurrent neural network (RNN) within the partitioned model inverse controller synthesis scheme in IMC framework and showed that this strategy provided an attractive alternative for NN-based control application. Maksumov et al. (2002) investigated partitioned model structure consisting of a linear ARX model and a NN model in the IMC framework. However, one 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. In this situation, on-line adaptation of these models requires model update from scratch, namely both network structure (e.g. the number of hidden neurons) 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 in model based controller design.

To alleviate the aforementioned problems, the JITL-based adaptive IMC design strategy will be investigated in Chapter 6. By taking advantage of simple models employed in JITL, the model inverse can be readily obtain for IMC design at

each sampling instant. Therefore, the IMC control strategy can be extended to the nonlinear processes without scarifying the simplicity of the linear IMC design.

2.3 Process Monitoring

Process monitoring is an important aspect of process engineering not only from plant's safety viewpoint, but also for the maintenance of yield and quality of process product. Therefore, there is strong incentive to have tools for process monitoring to ensure the success of the plant operations by recognizing anomalies of the process behavior. In the literature, there are two approaches for process monitoring: data-based methods and model-based methods. In what follows, the basic theories of the two methods will be introduced.

2.3.1 Data-based methods

Multivariate statistical analysis is a popular data-based technology for process monitoring. It employs the normal operation data to build statistical models, which represent the nominal process condition. If there is any fault in the process, the discrepancy will occur between the current measured process data and nominal statistical models. Therefore statistical tools (e.g. parameter estimation, interval estimation and hypothesis test) can be used to detect process faults. The most popular multivariate statistical methods are principal component analysis (PCA) (Piovoso et al., 1992; Nomikos and MacGregor, 1994; Chiang et al., 2001) and partial least squares (PLS) (Kresta et al., 1991; Nomikos and MacGregor, 1995). Wise and Gallagher (1996) provided a survey on the application of PCA and PLS in process monitoring.

In PCA analysis, the measured process variables are collected to form a data matrix whose covariance matrix is diagonalized in a statistically optimal manner by extracting the cross-correlation between the variables in the data matrix to build PCA model. If the measured variables are linearly related and contaminated by errors, the first few significant principal components capture the relationship among the variables, and the remaining principal components reflect only the error. Thus, eliminating the lesser important components reduces the contribution of errors in the measured data and to represented in a compact manner. Applications of PCA rely on its ability to reduce the dimensionality of the data matrix while capturing the underlying relationship between the variables. In addition, T^2 and Q charts are powerful visual tools to help PCA interpret the process trend, which are useful to assist the operators and engineers to understand the current status of the plant operation.

It should be noted that, PCA is based on linear correlation analysis, which limits its application for nonlinear systems. Xu et al. (1992) indicated that in nonlinear problems, PCA's minor components do not always consist of noise or unimportant variance, but they contain important information. To overcome this shortcoming, some researchers devoted to study nonlinear PCA. Kramer (1992) developed a nonlinear PCA method based on the autoassociative neural network. Dong and McAvoy (1996) proposed a nonlinear PCA by combining the principle curves and the autoassoicative neural network. Hiden et al. (1999) used genetic programming to address the same problem. Although these methods can improve performance of PCA to deal with nonlinear and dynamic problems, they sacrifice the simplicity of PCA and make the original statistical meaning of PCA somewhat vague. Other related research attempts to improve the PCA performance in fault detection include the use

of wavelet (Bakshi, 1998; Shao et al., 1999) and independent component analysis (Kano et al., 2004).

In PLS, the data matrices are decomposed into a series of abstract latent variables. However, the difference between constructing a PLS model and PCA model is that the former involves the decomposition of both input variable $X \in R^{n \times m}$ (n samples of m variables) and output variable $Y \in R^{n \times p}$ (n samples of p variables) data blocks so as to maximize the covariance between input and output data. Similar to PCA model, PLS is a linear approach. Therefore various nonlinear PLS techniques were developed, e.g. neural network PLS algorithm (Qin and McAvoy, 1992; Malthouse et al., 1997), and radial basis function network based PLS algorithm (Wilson and Irwin, 2000).

It is important to note that both conventional PCA and PLS are not suited to be applied to dynamic systems. To enhance their applicability to such systems, Ku et al. (1995) proposed a dynamic PCA by adding past value of each input variable to the data matrix. Negiz and Cinar (1997) proposed a state space model based on canonical variate analysis, which calculates linear combination of past value of the system inputs and/or the outputs that are highly correlated with the linear combination of the future values of the outputs of the process.

2.3.2 Model-based methods

Model based approach essentially refers to the analytical or functional redundancy technique (Himmelblau, 1978; Isermann and Belle, 1997; Frank, 1990; Frank et al., 2000). Model based methods detect faults by using the dependencies between different measurable signals. These dependencies are expressed by analytical process model. Based on the measured input and output signals, the monitoring

methods generate residuals between current process condition and model's state parameter/output. If there is any discrepancy between them, the residual will be relatively large, indicating the occurrence of the abnormal conditions. There is a variety of different approaches to the problem of model based fault detection and isolation using analytical redundancy. The most popular methods include parity space approach, dedicated observer approach and innovation based approach, fault detection filter approach, and parameter identification approach. As we discussed before, the analytical models are either too costly to obtain or not available. To overcome this problem, various neural network models have been proposed, for example Elman neural network (Saludes and Fuente, 1999), feedforward neural network (Frank et al., 2000), and neuro-fuzzy network (Patton et al., 2000). However, neural network and neuro-fuzzy networks have limitations when they are employed in on-line application as we discussed before.

Wachs and Lewin (1998) proposed a model-based PCA approach, which was applied to the monitoring of an ethylene compressor with good result. In this approach, the nonlinearity and dynamics of process are accounted for by using known first-principle models, followed by the PCA analysis of the residuals, i.e. the difference between the actual process outputs and model's predicted outputs. However, the difficulty with this procedure is that first-principle models may not be available or too costly to obtain. To alleviate the aforementioned drawbacks in the previous model-based monitoring methods, a model-based monitoring method based on the JITL technique will be investigated in Chapter 8.

Chapter 3

An Enhanced Just-in-Time Learning Technique

Data-based method is an attractive approach to extract information from data to build up process model to deal with the data rich but information poor problem in chemical processes. In this chapter, we will develop an enhanced just-in-time learning technique and apply it to the nonlinear process modeling.

3.1 Introduction

Traditional treatments of the data-based modeling methods focus on global approaches, such as neural networks, fuzzy set, and other kinds of non-linear parametric models (Nelles, 2001). 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. Another fundamental limitation of these methods is that it is

difficult for them to be updated online when the process dynamics are moved away from the nominal operating space. On the other hand, the idea of local modeling is to approximate a nonlinear system with a set of relatively simple local models valid in a certain operating regimes. The T-S fuzzy model (Takagi and Sugeno, 1985) and neuro-fuzzy network (Jang and Sun, 1995; Nelles, 2001) are well-known examples of local modeling approach. However, most local modeling approaches suffer from the drawback of requiring a priori knowledge to determine the partition of operating space and when this information is lacking, complicated training strategy needs to be resorted to determine both optimal model structure and parameters of the local models.

To alleviate the above problems, Just-in-Time Learning (JITL) was recently developed as an attractive alternative for modelling the nonlinear systems. As mentioned in Chapter 2, JITL has no standard learning phase and it only assumes that all available observations are stored in a database and the models are built dynamically upon query. Thus, 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 (Bontempi et al., 2001). In comparison, online adaptation of neural network and neuro-fuzzy models requires model update from scratch, namely both network structure (e.g. the number of hidden neurons in the former case and the number of the fuzzy rules in the latter) and model parameters may need to be changed simultaneously. Evidently, this procedure 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.

In the previous work, distance measures are overwhelmingly used to evaluate the similarity between two data samples (Atkeson et al., 1997; Rhodes and Morari, 1997; Bontempi et al., 2001; Braun et al., 2001). Complementary information available from angular relationship has not been exploited. In addition, the stability of local model is not addressed in the previous work, resulting in unstable local models even when the process is stable. In this chapter, by incorporating the stability constraints, an enhanced JITL methodology based on both angle measure and distance measure is proposed. In addition, a new procedure of selecting the relevant data set is proposed**.** Literature examples are used to illustrate the modeling capability of the proposed method in nonlinear process modeling.

3.2 Just-in-time Learning

There are three main steps in JITL to predict the model output corresponding to the query data: (1) the relevant data samples in the database are searched to match the query data by some nearest neighborhood criterion; (2) a local model is built based on the relevant data; (3) model output is calculated based on the 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.

To facilitate the ensuing developments, the JITL algorithm is described next. Suppose that a database consisting of N process data $(y_i, \mathbf{x}_i)_{i=1\sim N}$, $y_i \in R$, $\mathbf{x}_i \in R^n$, is collected. It is worth pointing out that the vector $\mathbf{x}_i$ is formed by the past values of both process input and process output in modeling a dynamic system, which will become evident in the following discussion. Given a specific query data $\mathbf{x}_q \in R^n$

whose elements are identical to those defined for $\mathbf{x}_i$, the objective of JITL is to predict the model output $\hat{y}_q = f(\mathbf{x}_q)$ according to the known database $(y_i, \mathbf{x}_i)_{i=1\sim N}$. In the literature, distance measure $d(\mathbf{x}_q, \mathbf{x}_i)$, e.g. Euclidean norm $d(\mathbf{x}_q, \mathbf{x}_i) = \|\mathbf{x}_q - \mathbf{x}_i\|_2$, is commonly used to select the relevant data set from the database by evaluating the relevance (or similarity) between the query data $\mathbf{x}_q$ and $\mathbf{x}_i$ in the entire database, i.e. smaller value of distance measure indicates greater similarity between $\mathbf{x}_q$ and $\mathbf{x}_i$. In doing so, a weight w_i is assigned to each data $\mathbf{x}_i$ and it is calculated by the kernel function, $w_i = \sqrt{K(d(\mathbf{x}_q, \mathbf{x}_i)/h)}$, where h is the bandwidth of the kernel function K that normally uses a Gaussian function, $K(d) = e^{-d^2}$. If a linear model is employed to calculate the model output $\hat{y}_q$, the query answer is given by (Atkeson et al., 1997)

$$\hat{y}_q = \mathbf{x}_q^T (P^T P)^{-1} P^T \mathbf{v} \tag{3.1}$$

where $P = W\Phi$, $\mathbf{v} = W\mathbf{y}$, $W \in R^{N\times N}$ is a weight matrix with diagonal elements w_i, $\Phi \in R^{N\times n}$ is the matrix with every row corresponding to $\mathbf{x}_i^T$, and $\mathbf{y} = [y_1, y_2, \cdots, y_N]^T$.

In JITL, PRESS statistic (Myers, 1990) is used to perform leave-one-out cross validation to assess the generalization capability of the model (Atkeson et al., 1997). For a current query data $\mathbf{x}_q$, the leave-one-out cross validation test determines the optimal values of h, h_{opt}, as follows: for a given h, Eq. (3.1) is used to compute the predicted outputs as required in the leave-one-out cross validation test and the corresponding validation error is calculated. This procedure repeats for a number of h and h_{opt} is chosen as the one resulting in the smallest validation error. With h_{opt}

known, the optimal model prediction $\hat{y}_q$ is then computed using Eq. (3.1) for the current query data $\mathbf{x}_q$.

As mentioned above, each local model obtained by JITL is only locally valid around the query data, therefore simple model structure can be chosen as local model at each query point. For a dynamic system, ARX model can be chosen as the local model for JITL. The ARX model is given as follows:

$$\hat{y}(k) = \mathbf{z}^T(k-1)\Psi \tag{3.2}$$

where $\hat{y}(k)$ is the model output at the *k*-th sampling instant, $\mathbf{z}(k-1)$ is the regression vector, and Ψ is the model parameter vector as given by:

$$\mathbf{z}(k-1) = [y(k-1), \quad \ldots, \quad y(k-n_y), \quad u(k-n_d-1), \quad \ldots, \quad u(k-n_d-n_u)]^T \tag{3.3}$$

$$\Psi = [\psi_1, \quad \ldots, \quad \psi_{n_y}, \quad \psi_{n_y+1}, \quad \ldots, \quad \psi_{n_y+n_u}]^T \tag{3.4}$$

where n_y and n_u are integers related to the model's order, and n_d is the process time delay. By comparing Eq. (3.1) and (3.2), it is evident that the local model parameters obtained by JITL method is computed as $(P^TP)^{-1}P^T\mathbf{v}$. Furthermore, the vector $\mathbf{x}_i$ in the database and query data $\mathbf{x}_q$ have the same input and output variables as those defined for $\mathbf{z}(k-1)$. For example, for a first-order model with $n_y = n_u = 1$ and $n_d = 0$, the database $(y_i, \mathbf{x}_i)_{i=1\sim N}$ is given by $[y_{id}(k), y_{id}(k-1), u_{id}(k-1)]_{k=1\sim N}$ where $y_{id}(k)$ and $u_{id}(k)$ are the process output and input data collected at the *k*-th sampling instant in the identification test. Similarly, in the prediction phase, the query data $\mathbf{x}_q$ at the (*k*-1)-th sampling instant is arranged in the form of $[y(k-1), u(k-1)]^T$ as the input to the JITL algorithm, from which the predicted output at the next sampling

instant $\hat{y}(k)$ can be computed. Finally, referring from Eq. (3.3), the dimensionality of $\mathbf{x}_i$ and $\mathbf{x}_q$ is equal to $n = n_y + n_u$.

3.3 Enhanced JITL Methodology

In the preceding section, it is evident that the conventional JITL methods only use distance measure to evaluate the similarity between two data samples. However, considering data observations as points in space leads to two types of measures: distance and angle between two vectors. Some researchers have demonstrated the advantage of using additional angle measure in evaluating the similarity/dissimilarity between data in principal component analysis (Raich and Cinar, 1997; Yoon and MacGregor, 2001; Singhal and Seborg, 2002). In this chapter, to enhance the predictive capability of JITL, the following similarity number, s_i, by incorporating the angular relationship, is defined.

$$s_i = \gamma \cdot \sqrt{e^{-d^2(\mathbf{x}_q, \mathbf{x}_i)}} + (1-\gamma) \cdot \cos(\theta_i), \text{ if } \cos(\theta_i) \geq 0 \tag{3.5}$$

where γ is a weight parameter and is constrained between 0 and 1, and θ_i is the angle between $\Delta\mathbf{x}_q$ and $\Delta\mathbf{x}_i$, where $\Delta\mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_{i-1}$, $\Delta\mathbf{x}_q = \mathbf{x}_q - \mathbf{x}_{q-1}$. The value of s_i is bounded between 0 and 1. When s_i approaches to 1, it indicates that $\mathbf{x}_i$ resembles closely to $\mathbf{x}_q$.

It is important to note that Eq. (3.5) will not be used to compute the similarity number s_i between $\mathbf{x}_q$ and $\mathbf{x}_i$ if $\cos\theta_i$ is negative. For simplicity, this point is illustrated in the two-dimensional space as shown in Figure 3.1, where $\Delta\mathbf{x}_q^{\perp}$ denotes the vector perpendicular to $\Delta\mathbf{x}_q$. It is clear that a vector $\Delta\mathbf{x}_i$ lies to the right of $\Delta\mathbf{x}_q^{\perp}$

(say $\Delta\mathbf{x}_1$) is more similar to $\Delta\mathbf{x}_q$ than a vector to the left of $\Delta\mathbf{x}_q^{\perp}$ (say $\Delta\mathbf{x}_2$). To discriminate the directionality between $\Delta\mathbf{x}_q$ and $\Delta\mathbf{x}_i$, cosine function is employed, whose value is positive to the right of $\Delta\mathbf{x}_q^{\perp}$ (e.g. $\cos\theta_1$) and negative to the left of $\Delta\mathbf{x}_q^{\perp}$ (e.g. $\cos(\pi-\theta_2)$). Therefore, a negative cosine function indicates that two vectors $\Delta\mathbf{x}_q$ and $\Delta\mathbf{x}_i$ are dissimilar and hence, $\mathbf{x}_i$ will be discarded and not involved in the subsequent JITL procedure. On the other hand, a positive cosine function requires the subsequent calculation of the proposed similarity number s_i in order to further discriminate the similarity between $\mathbf{x}_q$ and $\mathbf{x}_i$.

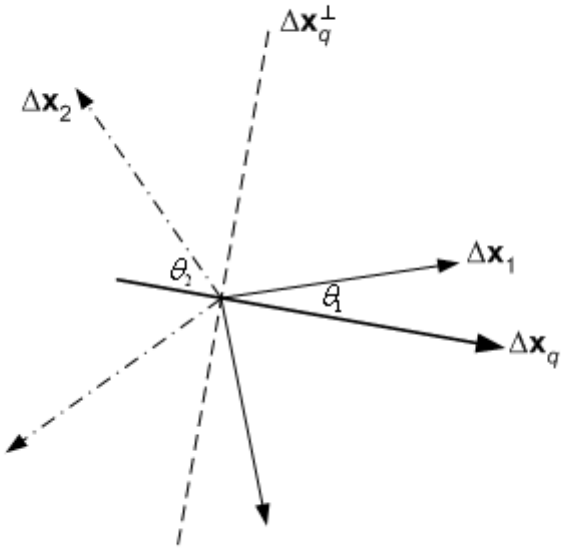


Figure 3.1 Illustration of angle measure

Another shortcoming of the conventional methods is that all the data $\mathbf{x}_i$ in the database are employed in the regression, as shown in Eq. (3.1). This may lead to a large sparse regression matrix $P \in R^{N \times n}$ that is prone to numerical problems. To circumvent this problem, we propose that only a pre-specified number of relevant data with greater resemblance to the query data $\mathbf{x}_q$, as determined by the similarity number defined in Eq. (3.5), are used in the regression. Specifically, two parameters $k_{\min}$ and $k_{\max}$ are chosen such that only the relevant data sets formed by the $k_{\min}$-th relevant data to the $k_{\max}$-th relevant data are used in the regression. Because $k_{\min}$ and $k_{\max}$ are much smaller than the number of data in the entire database, i.e. N, the computational burden is significantly reduced compared to the conventional JITL methods.

Last, it is noted that stability of local model is not taken into account in the conventional JITL methods. As a result, some local models generated by JITL may be unstable even when the database employed and the query points are from stable process. This feature is not desirable, especially when these models are to be employed in the controller design. Thus, for a stable system, the parameters of each ARX model obtained by JITL need to be verified whether they satisfy the stability constraint or not. In case the parameters fail to satisfy the stability constraint, a constrained optimization problem can be incorporated into JITL to obtain the stable model. Similar procedure can be devised to obtain an unstable local model when the process of interest is unstable. The parametric stability constraints imposed on Ψ will be discussed in the ensuing development.

The detailed algorithm of the proposed JITL methodology is described below. Given a database $(y_i, \mathbf{x}_i)_{i=1\sim N}$, the parameters $k_{\min}$, $k_{\max}$, and weight parameter γ, and a query data $\mathbf{x}_q$:

Step 1: Compute the distance and angle between $\mathbf{x}_q$ and each data $(y_i, \mathbf{x}_i)$:

$$d_i = \left\| \mathbf{x}_q - \mathbf{x}_i \right\|_2, i = 1 \sim N \tag{3.6}$$

$$\cos(\theta_i) = \frac{\Delta \mathbf{x}_q^T \Delta \mathbf{x}_i}{\left\| \Delta \mathbf{x}_q \right\|_2 \cdot \left\| \Delta \mathbf{x}_i \right\|_2}, i = 1 \sim N \tag{3.7}$$

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

$$s_i = \gamma \cdot \sqrt{e^{-d_i^2}} + (1-\gamma) \cdot \cos(\theta_i) \tag{3.8}$$

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 ($\mathbf{y}_l$, Φ_l), where $\mathbf{y}_l \in R^{l\times 1}$ and $\Phi_l \in R^{l\times n}$, 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 $W_l \in R^{l\times l}$ a diagonal weight matrix with diagonal elements being the first l largest values of s_i, and calculate:

$$P_l = W_l \Phi_l \tag{3.9}$$

$$\mathbf{v}_l = W_l \mathbf{y}_l \tag{3.10}$$

The local model parameters are then computed by:

$$\Psi_l = (P_l^T P_l)^{-1} P_l^T \mathbf{v}_l \tag{3.11}$$

where $(P_l^T P_l)^{-1}$ is calculated by SVD method. Next, the leave-one-out cross validation test is conducted and the validation error is calculated by (Myers, 1990):

$$e_l = \frac{1}{\sum_{j=1}^{l} s_j^2} \sum_{j=1}^{l} (s_j \frac{y_j - \varphi_j^T (P_l^T P_l)^{-1} P_l^T \mathbf{v}_l}{1 - \mathbf{p}_j^T (P_l^T P_l)^{-1} \mathbf{p}_j})^2 \tag{3.12}$$

where y_j is the j-th element of $\mathbf{y}_l$, ϕ_j^T and $\mathbf{p}_j^T$ are the j-th row vector of Φ_l and P_l, respectively.

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

$$l_{opt} = \arg\min_{l}(e_l) \tag{3.13}$$

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

First-order model:

$$-1 < \psi_1 < 1 \tag{3.14}$$

Second-order model:

$$\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix} < \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{3.15}$$

$$-1 < \psi_1 < 1 \tag{3.16}$$

If $\Psi_{l_{opt}}$ satisfies the stability constraint, the predicted output for query data is computed as

$$(\hat{y}_q)_{l_{opt}} = \mathbf{x}_q^T \Psi_{l_{opt}} \tag{3.17}$$

Otherwise, $\Psi_{l_{opt}}$ is used as the initial value in the following optimization problem subject to appropriate stability constraint, Eq. (3.14) or Eqs. (3.15) and (3.16).

$$\min_{\Psi} \| P_{l_{opt}} \Psi - \mathbf{v}_{l_{opt}} \|_2 \tag{3.18}$$

With the optimal solution $\Psi^*_{l_{opt}}$ obtained from Eq. (3.18), the predicted output for query data is then calculated as $\mathbf{x}_q^T \Psi^*_{l_{opt}}$.

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

One remark about the proposed method is the determination of γ. Typically, the prediction accuracy of the proposed method improves initially when γ decreases from one to a smaller value of γ, after which the prediction accuracy degrades. Owing to the lack of the systematic guideline of determining the optimal value of γ, the following procedure is adopted: the proposed method is applied to the validation data for a number of γ values and the corresponding validation error is calculated. The optimal γ is chosen as the one resulting in the smallest validation error.

Although the aforementioned data-based modeling methodology is developed for the single-input single-out systems, it carries straightforwardly over to the multivariable systems. This is because the modeling of a multivariable system with m outputs can be treated as m multiple-input single-output problems. A chemical reactor with two inputs and three outputs will be presented in the next section to demonstrate the application of the proposed method for nonlinear modeling of multivariable systems.

3.4 Examples

Example 1 Consider the van de Vusse reactor with the following reaction kinetic scheme: $A \to B \to C$, $A \to D$, which is carried out in an isothermal CSTR. The dynamics of the reactor are described by the following equations (Doyle et al., 1995):

$$\frac{dC_A}{dt} = -k_1 C_A - k_3 C_A^2 + \frac{F}{V}(C_{Af} - C_A) \tag{3.19}$$

$$\frac{dC_B}{dt} = k_1 C_A - k_2 C_B - \frac{F}{V} C_B \tag{3.20}$$

where the parameters used are: $k_1 = 50h^{-1}$, $k_2 = 100h^{-1}$, $k_3 = 10$ L/(mol·h), $C_{Af} = 10$ mol/L, and $V = 1$ L. The nominal operation condition is $C_A = 3.0$ mol/L, $C_B = 1.12$ mol/L, and $F = 34.3$ L/h. The concentration of component B, C_B, is the process output and the flow rate, F, is the process input.

A salient feature of the above reactor is that the sign of its steady state gain may change as the operation condition changes (see Figure 3.2). To apply the proposed method, a second-order ARX model is employed as the local model, i.e. the regression vector in Eq. (3.3) is chosen as $\mathbf{z}(k-1) = [C_B(k-1), C_B(k-2), F(k-1)]^T$, and set $k_{\min} = 6$ and $k_{\max} = 60$. the database is generated by introducing uniformly random steps with distribution of [10, 150] and the switching probability of 0.1 at every sampling time to the process input F. Two thousand input/output data as shown in Figure 3.3 are collected to build the database $[C_B(k), \mathbf{z}(k-1)]_{k=1\sim2000}$. Because this system is stable in the operating space under consideration, the local model needs to satisfy the stability constraints given in Eqs. (3.15) and (3.16).

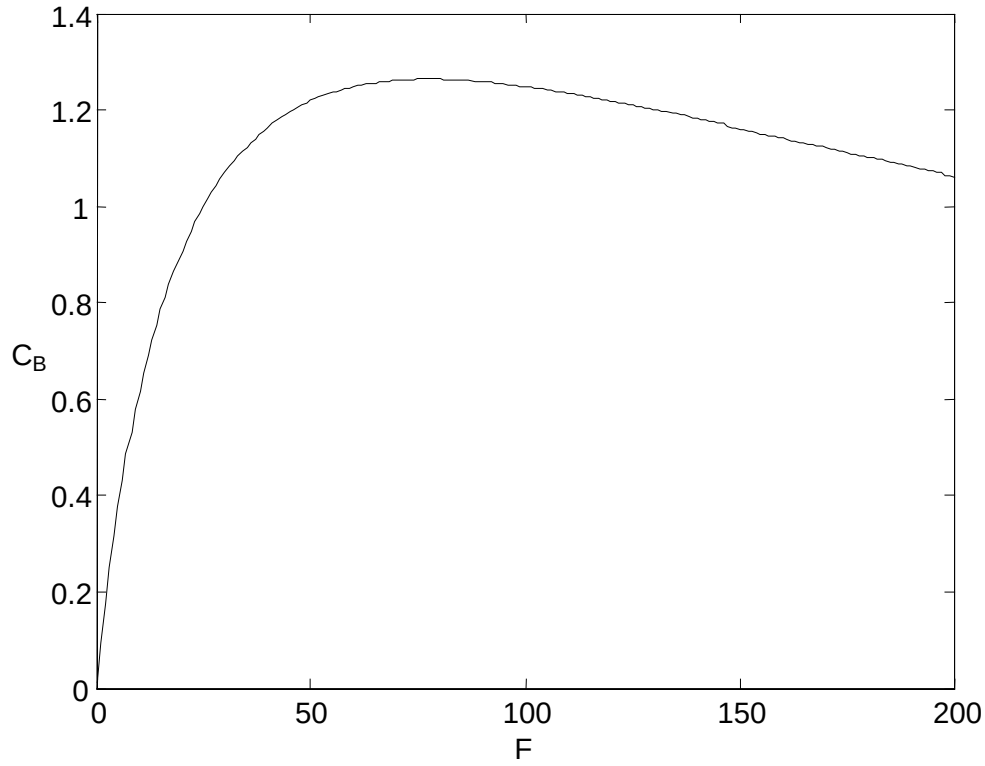


Figure 3.2 Steady-state curve of van de Vusse reactor

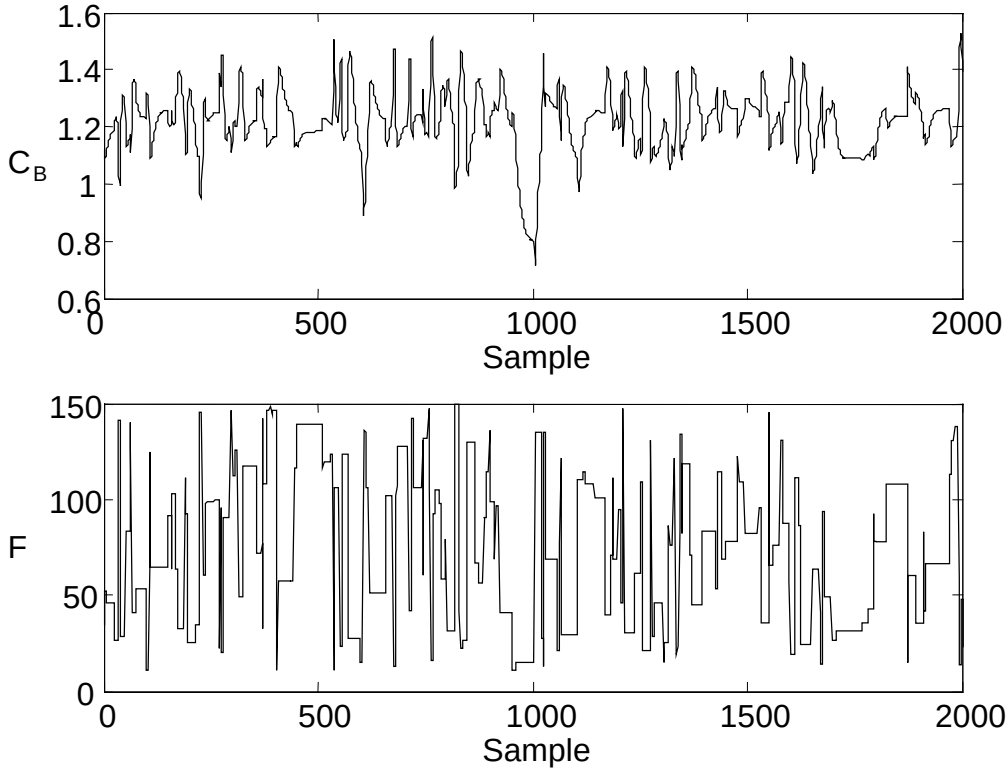


Figure 3.3 Input-output data used for constructing the database (van de Vusse reactor)

To determine the optimal value of the weight parameter γ, Table 3.1 lists the mean-squared-error (MSE) of the validation test for different values of γ. The input signal employed in the validation test is shown in Figure 3.4. As can be seen from Table 3.1, the error decreases initially as γ decreases from 0.98 to 0.95, after which the error starts to increase. Therefore, the optimal γ is chosen to be 0.95. Based on the same database, JITL with distance measure alone is also considered for comparison purpose. The predictive performances of these two methods are compared in Table 3.1 and Figure 3.4. It is evident that the proposed method complemented with angle measure and stability constraint outperforms the conventional JITL.

Table 3.1 Validation error of the proposed method for various values of γ

Distance measure	$\gamma = 0.98$	$\gamma = 0.95$	$\gamma = 0.90$	$\gamma = 0.85$	$\gamma = 0.80$
7.72×10^{-4}	8.11×10^{-5}	7.80×10^{-5}	7.82×10^{-5}	8.03×10^{-5}	9.45×10^{-5}

Figure 3.5 demonstrates the prediction capability of the proposed method with $\gamma = 0.95$ when F is subject to step changes of 15 and -20 respectively. The steady-state errors are 1.12×10^{-3} and 1.52×10^{-4} respectively. This simulation condition is adopted from the work done by Doyle et al. (1995) who constructed a second-order Volterra model to predict this reactor's dynamics. In their paper, the steady state prediction errors for positive and negative step changes are estimated to be 0.016 and 0.056 respectively. Clearly, the proposed method gives much more accurate prediction than the Volterra model. To illustrate the capability of the proposed method to model reactor's dynamics when the value of input F is changed from one side of the extreme point to the opposite side (see Figure 3.2), the step changes of F from

34.3 to 55 and 109 are conducted. Note that these two final values of F correspond to the identical steady state value of C_B. As illustrated in Figure 3.6, the proposed method can predict the actual process very closely, as also evidenced by very small steady-state errors of 6.66×10^{-5} (top curve) and 1.18×10^{-5} (bottom curve).

To test the robustness of the proposed method, both process output and input variables are corrupted by 2% Guassian white noise. Despite that both database and validation data contain the corrupted signals, the proposed method maintains good prediction accuracy in the presence of process noise, as illustrated in Figure 3.7.

Example 2 The application of the proposed method in modeling multivariable systems is illustrated by considering a nonisothermal CSTR with first-order reaction, which can be described by the following equation (You and Nikolaou, 1993):

$$A\frac{dL}{dt} = F_i - K\sqrt{L} \tag{3.21}$$

$$\frac{dC_A}{dt} = \frac{F_i}{V}(C_{Ai} - C_A) - k_0 e^{-E/RT} C_A \tag{3.22}$$

$$\frac{dT}{dt} = \frac{F_i}{V}(T_i - T) + \frac{-\Delta H_R}{c_p \rho} k_0 e^{-E/RT} C_A - \frac{Q}{c_P \rho V}(1+v) \tag{3.23}$$

The parameters used in the simulation and the nominal operating conditions are summarized in Table 3.2. In this example, the variables F_i and C_{Ai} are process inputs, whereas L, C_A, and T are process outputs.

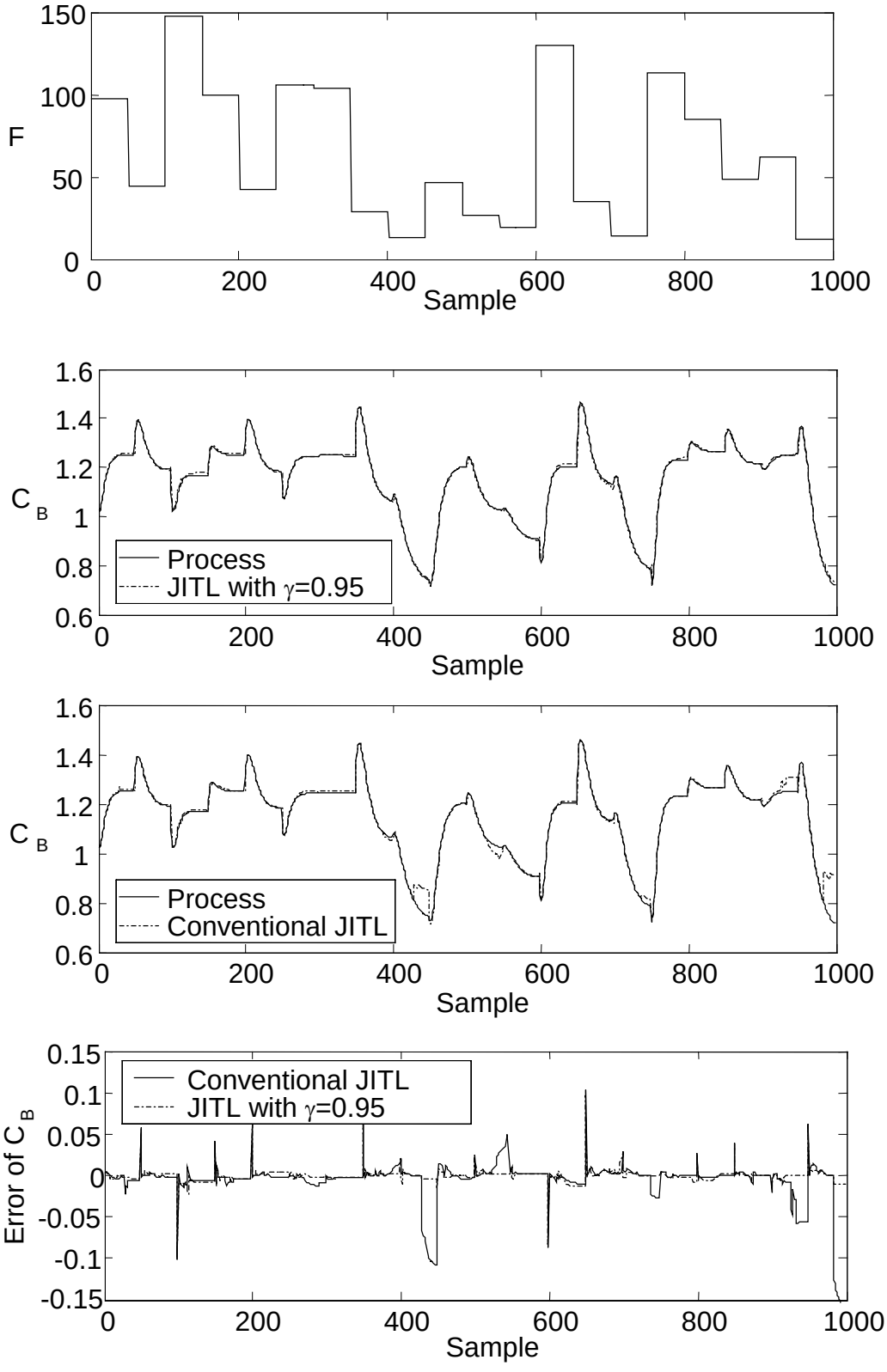


Figure 3.4 Validation result of C_B

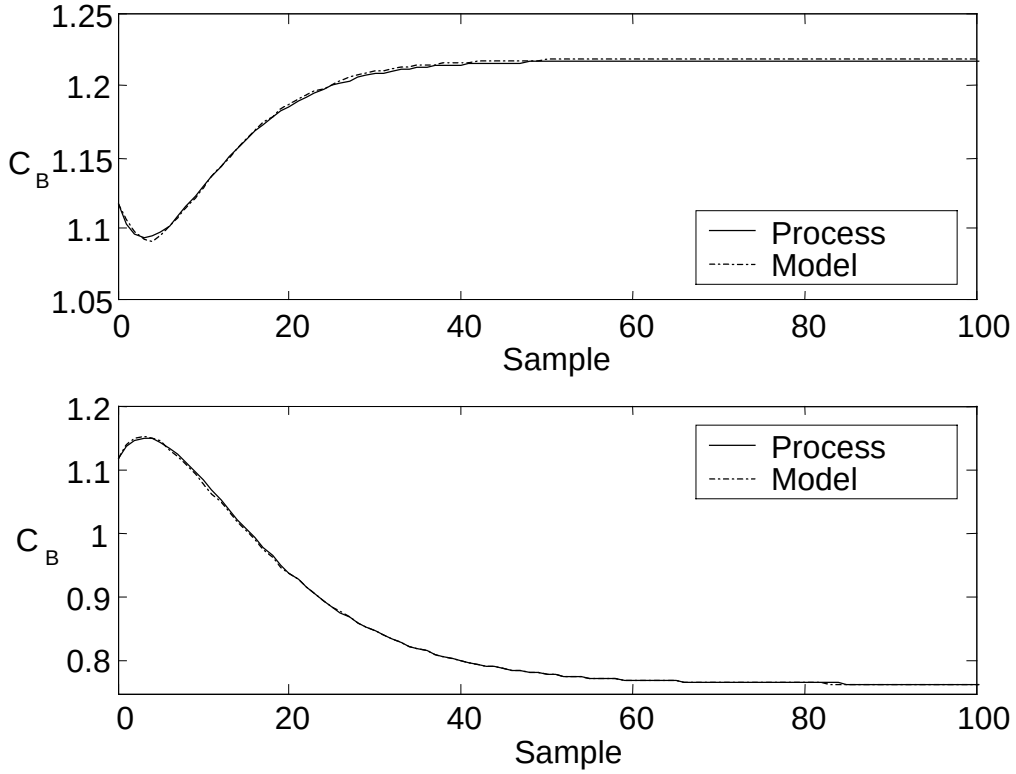


Figure 3.5 Response for step changes from 34.3 to 49.3 (top) and 14.3 (bottom) in F

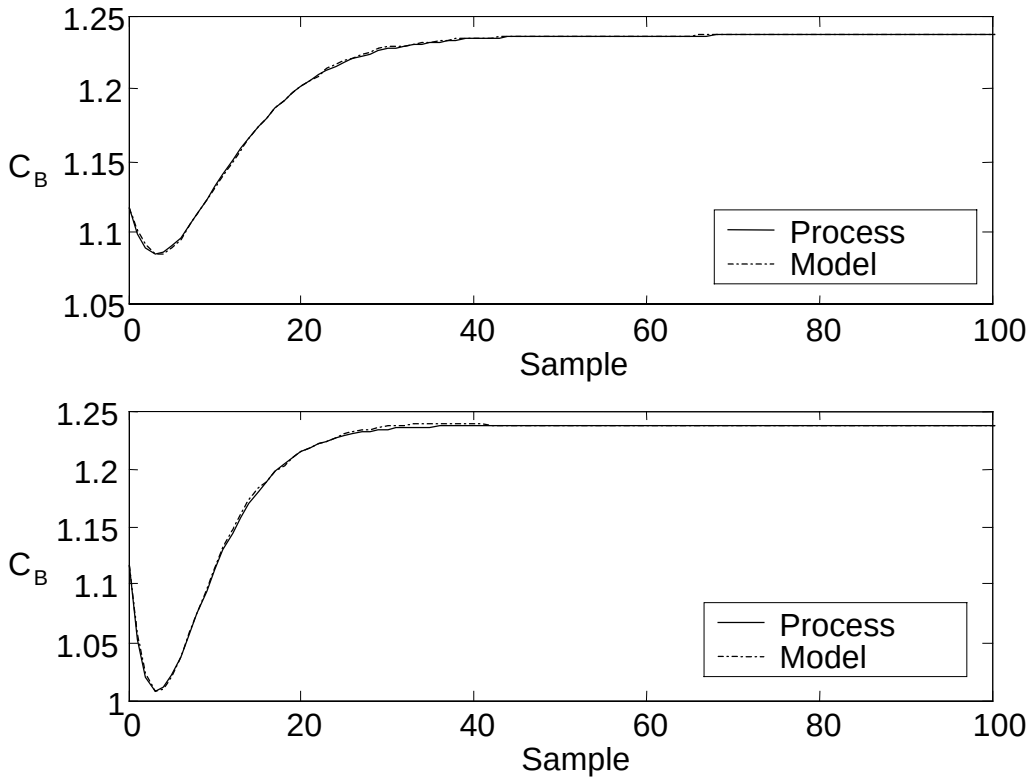


Figure 3.6 Response for step changes from 34.3 to 55 (top) and 109 (bottom) in F

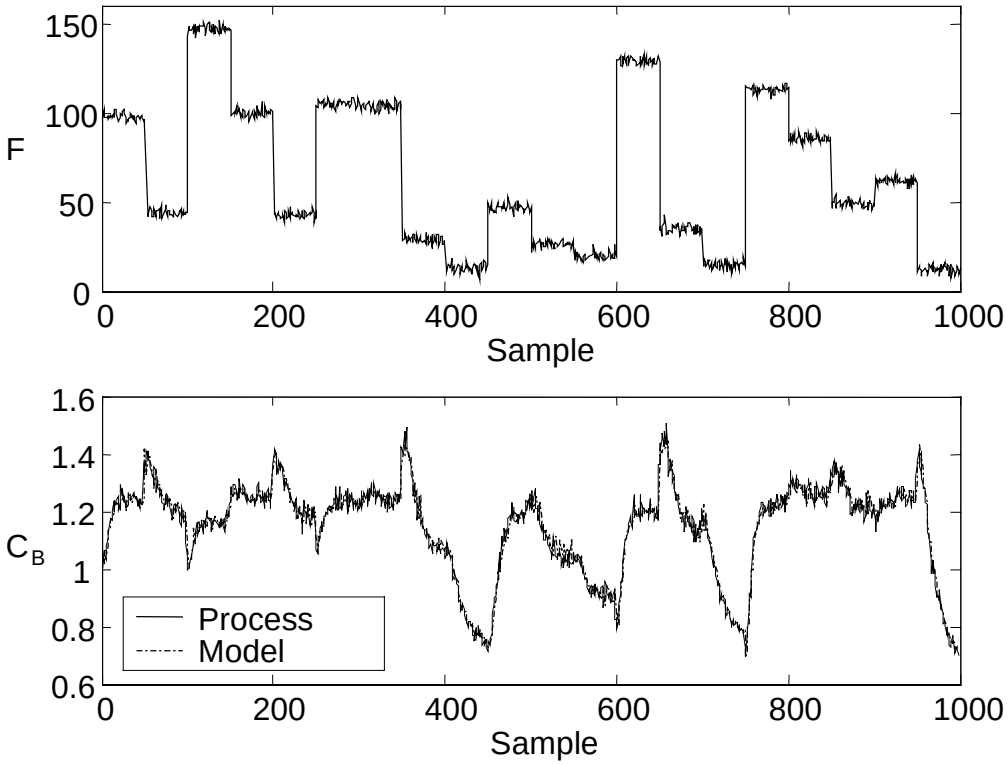


Figure 3.7. Validation result (with noisy process data)

Table 3.2 Parameters and nominal values of CSTR example

Variable	Value	Variable	Value
A	1.000 m^2	v	0
K	0.9715 $m^{5/2}$/h	V	1.360 m^3
k_0	7.08×10^7 1/h	T_i	373.3 K
E/R	8375 K	C_A	393.3 mol/ m^3
ΔH_R	-69755 J/mol	T	547.5 K
c_p	3140 J/kg K	L	1.360 m
ρ	800.8 kg/m^3	F_i	1.133 m^3/h
Q	1.055×10^8 J/h	C_{Ai}	8008 mol/m^3

To proceed with the proposed method, the following regression vectors are chosen:

$$L : \mathbf{z}_1(k-1) = [L(k-1), \quad L(k-2), \quad F_i(k-1)] \tag{3.24}$$

$$C_A : \mathbf{z}_2(k-1) = [C_A(k-1), \quad C_A(k-2), \quad F_i(k-1), \quad C_{Ai}(k-1)] \tag{3.25}$$

$$T : \mathbf{z}_3(k-1) = [T(k-1), \quad T(k-2), \quad F_i(k-1), \quad C_{Ai}(k-1)] \tag{3.26}$$

To generate the database, random step signals with uniform distribution of [1.02 1.25] and [7207 8808] and the switching probability of 0.15 are added to F_i and C_{Ai}, respectively. The input/output data given in Figure 3.8 are then used to construct three databases: $[L(k), \mathbf{z}_1(k-1)]_{k=1\sim2000}$, $[C_A(k), \mathbf{z}_2(k-1)]_{k=1\sim2000}$, and $[(T(k), \mathbf{z}_3(k-1)]_{k=1\sim2000}$ for predicting L, C_A, and T, respectively.

As a result of the open-loop stable nature of this reactor, the parameters of three local models obtained for each query data need to satisfy the stability constraints given in Eqs. (3.15) and (3.16). In addition, $k_{\min} = 6$ and $k_{\max} = 60$ are chosen to predict the output L, whereas $k_{\min} = 8$ and $k_{\max} = 60$ are used for the other two predicted outputs. To determine the optimal value of γ, Table 3.3 summarizes the mean squared errors of the validation test for three process outputs.

Table 3.3 Validation error of the proposed method for various values of γ

	Distance measure	$\gamma = 0.98$	$\gamma = 0.95$	$\gamma = 0.90$	$\gamma = 0.85$	$\gamma = 0.75$	$\gamma = 0.70$
L	3.90×10^{-3}	1.18×10^{-5}	1.11×10^{-5}	1.16×10^{-5}	1.17×10^{-5}	1.17×10^{-5}	1.18×10^{-5}
C_A	211.66	93.50	82.02	68.17	67.02	67.91	68.00
T	1.62	0.92	0.86	0.68	0.64	0.53	0.56

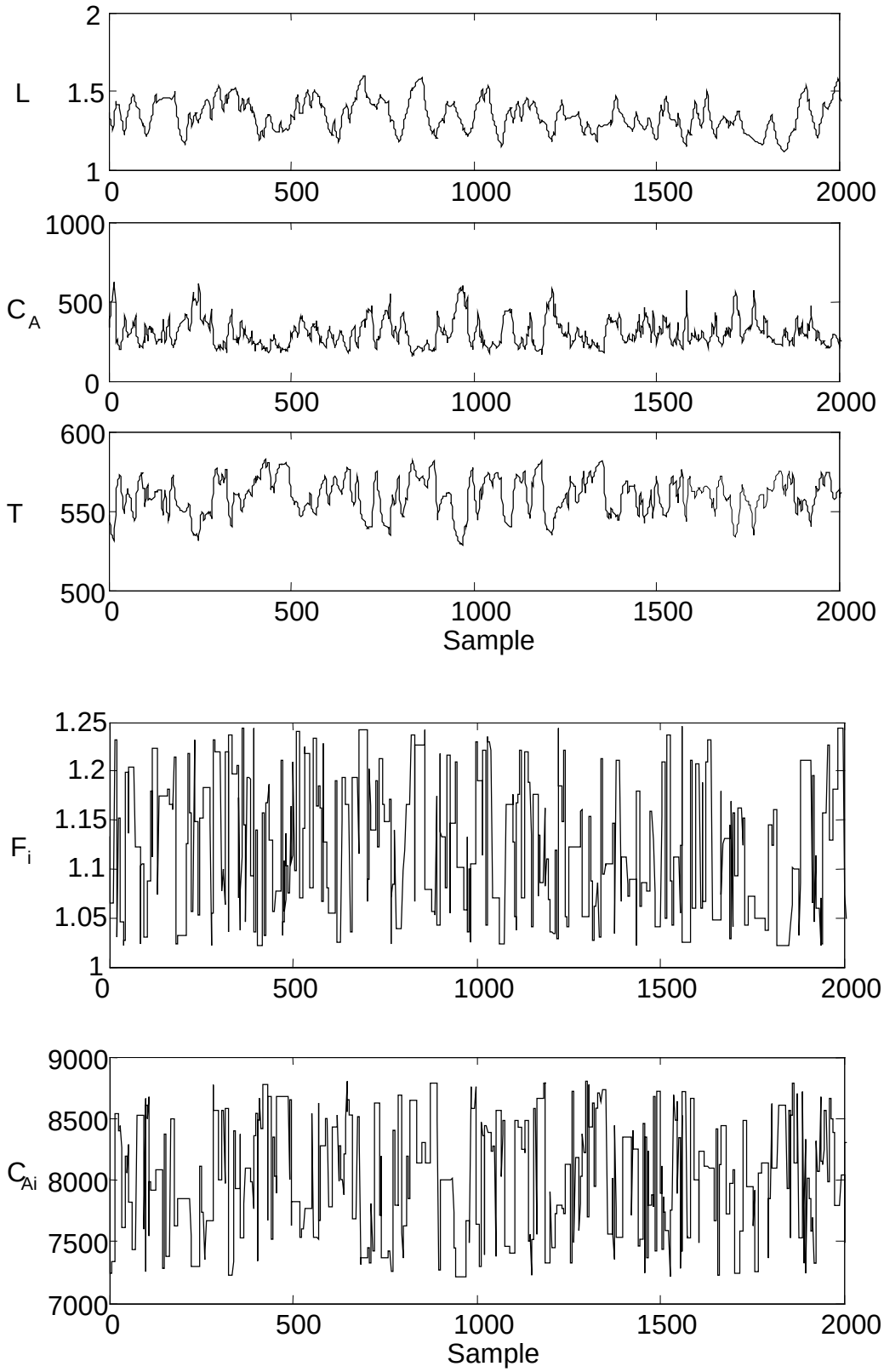


Figure 3.8 Input-output data used for constructing the database (CSTR example)

Figure 3.9 shows the input signal used in the validation test. As can be seen from Table 3.3, $\gamma = 0.95$, $\gamma = 0.85$, and $\gamma = 0.75$ are the best values for the proposed method to predict L, C_A, and T. For comparison purpose, JITL based on the distance measure alone and the same database is also used to model this process. The comparison results given in Table 3.3 and Figures 3.10 to 3.12 show that the proposed method has superior prediction accuracy than the conventional JITL. Again, the robustness of the proposed method is evaluated by introducing 1% Guassian white noise to the measured process variables. As illustrated in Figures 3.13, the proposed method is insensitive to process noise to some extent.

For traditional data-based modeling methods, it is not a trivial task to update the model parameters online. For example, neural networks need to be retrained to adjust the network parameters according to the new operating condition. In the extreme cases, the network structure may even be re-determined to achieve better prediction of the new process dynamics. Evidently, this procedure is not desirable from a computational point of view. In contrast, JITL is inherently adaptive by simply adding the current process data online to the database. For illustration, assume that the heat transfer Q in Eq. (3.23) is suddenly changed $\pm 25\%$ from its nominal value due to the effect of unmeasured disturbance, meaning that the parameter v changes from the nominal value of 0 to 0.25 and –0.25, respectively. Two scenarios are studied: non-adaptive version and adaptive version of the proposed method. In the former, the original databases mentioned previously remain unchanged, whereas in the latter the databases are constantly updated by adding the new available input-output data to the databases at each sampling time. Simulation results in Figures 3.14 and 3.15 show that significantly smaller modeling error is achieved by the adaptive version of the proposed method.

3.5 Conclusion

In this chapter, a data-based methodology for nonlinear process modeling is proposed. The proposed method makes use of both distance measure and angular measure to evaluate the similarity between the query data and data in the database. In addition, a constrained optimization problem is incorporated into the proposed method to address the stability of local model. Simulation studies illustrate that the proposed method gives marked improvement over its conventional counterparts in nonlinear process modeling. It is also demonstrated that the proposed method can be made adaptive online readily by simply adding the new process data to the database. In the subsequent chapters, we will employ this JITL modeling methodology to controller design and process monitoring.

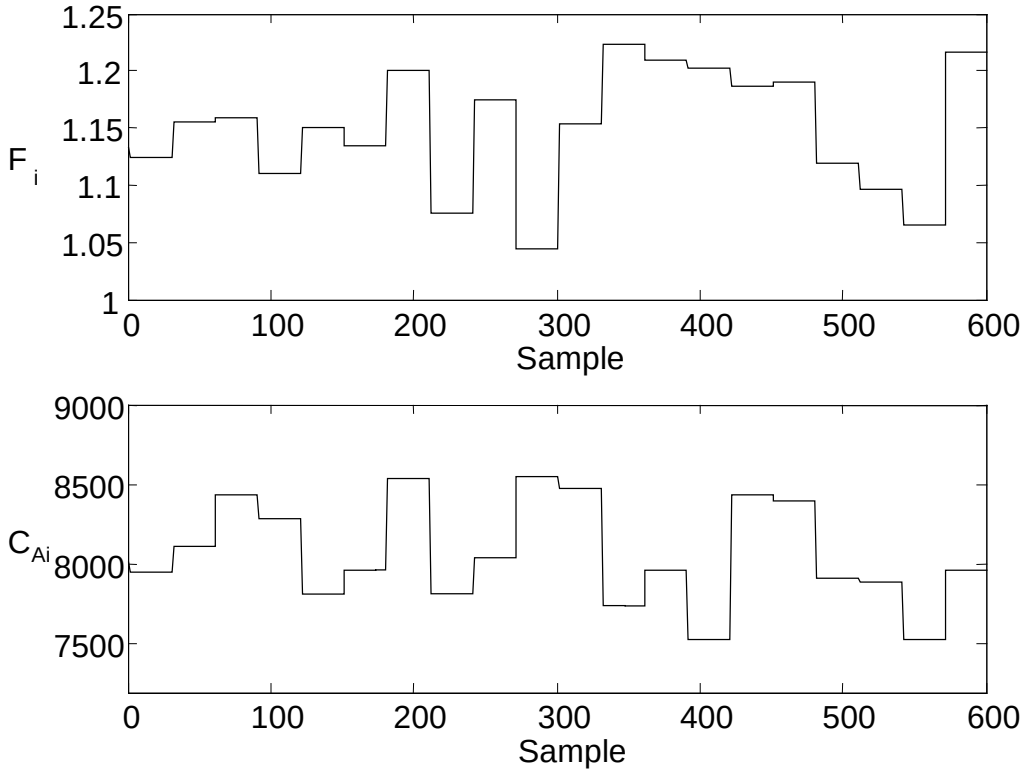


Figure 3.9 Input data used in the validation test of CSTR example

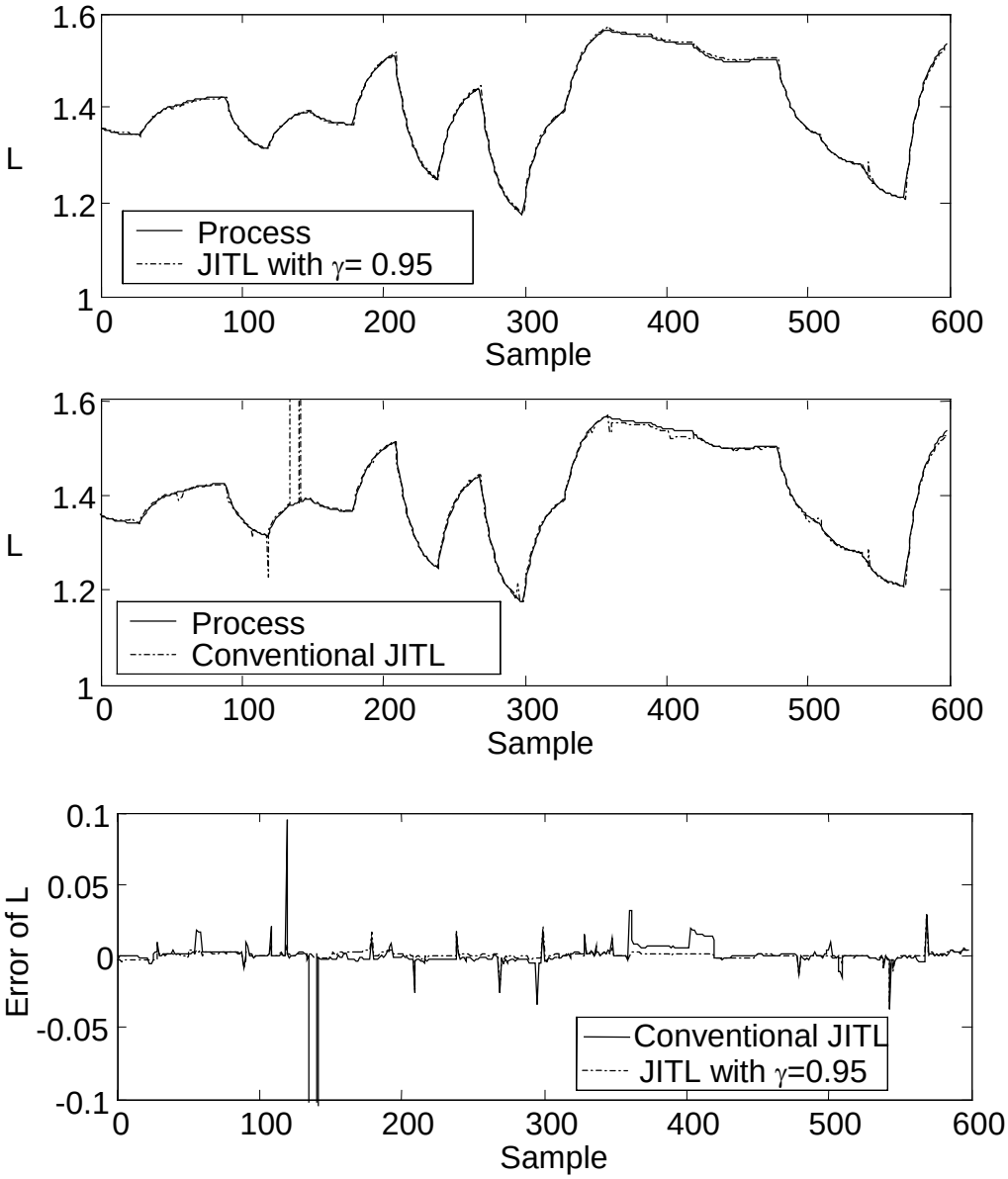


Figure 3.10 Validation results of L

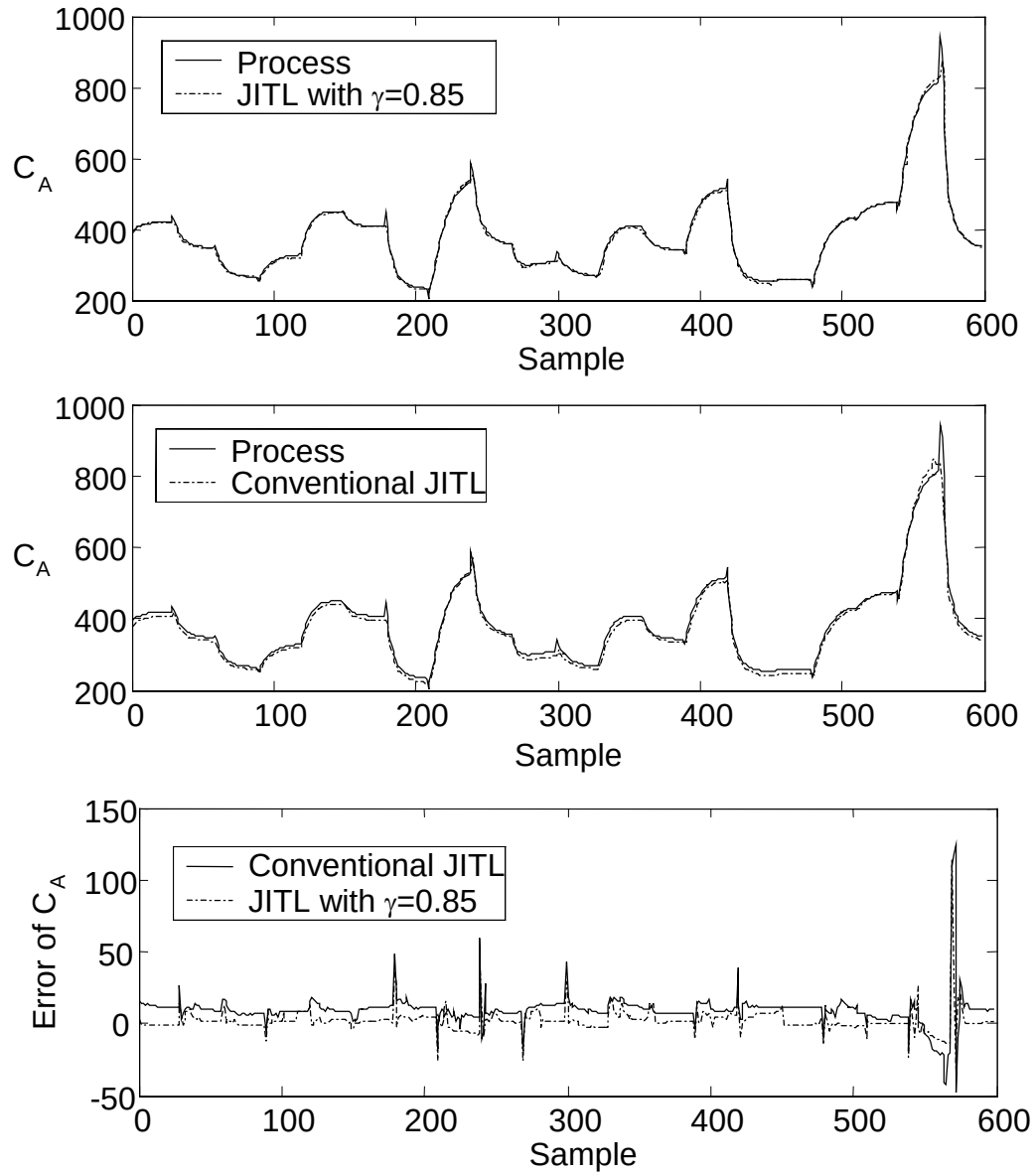


Figure 3.11 Validation results of C_A

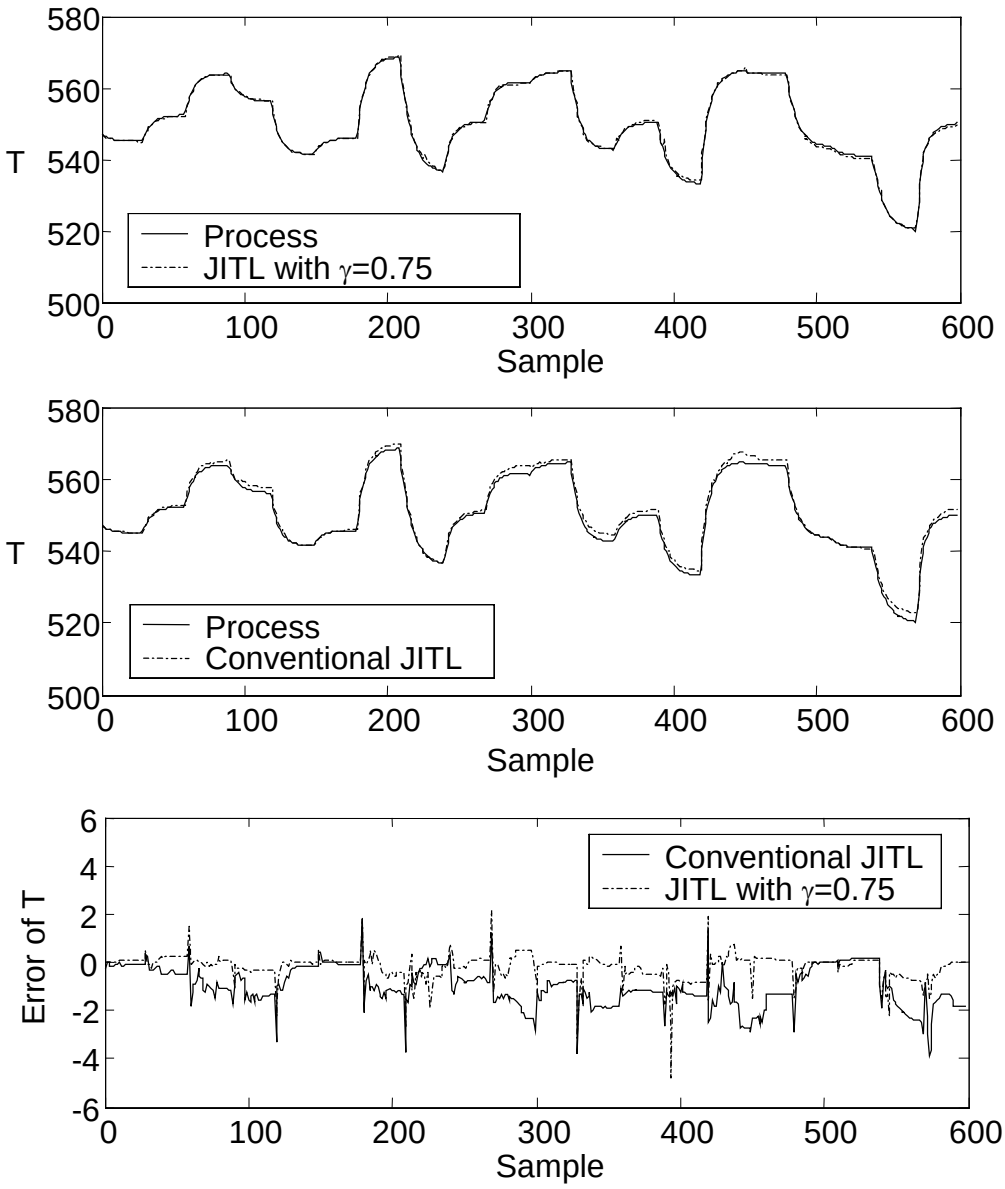


Figure 3.12 Validation results of T

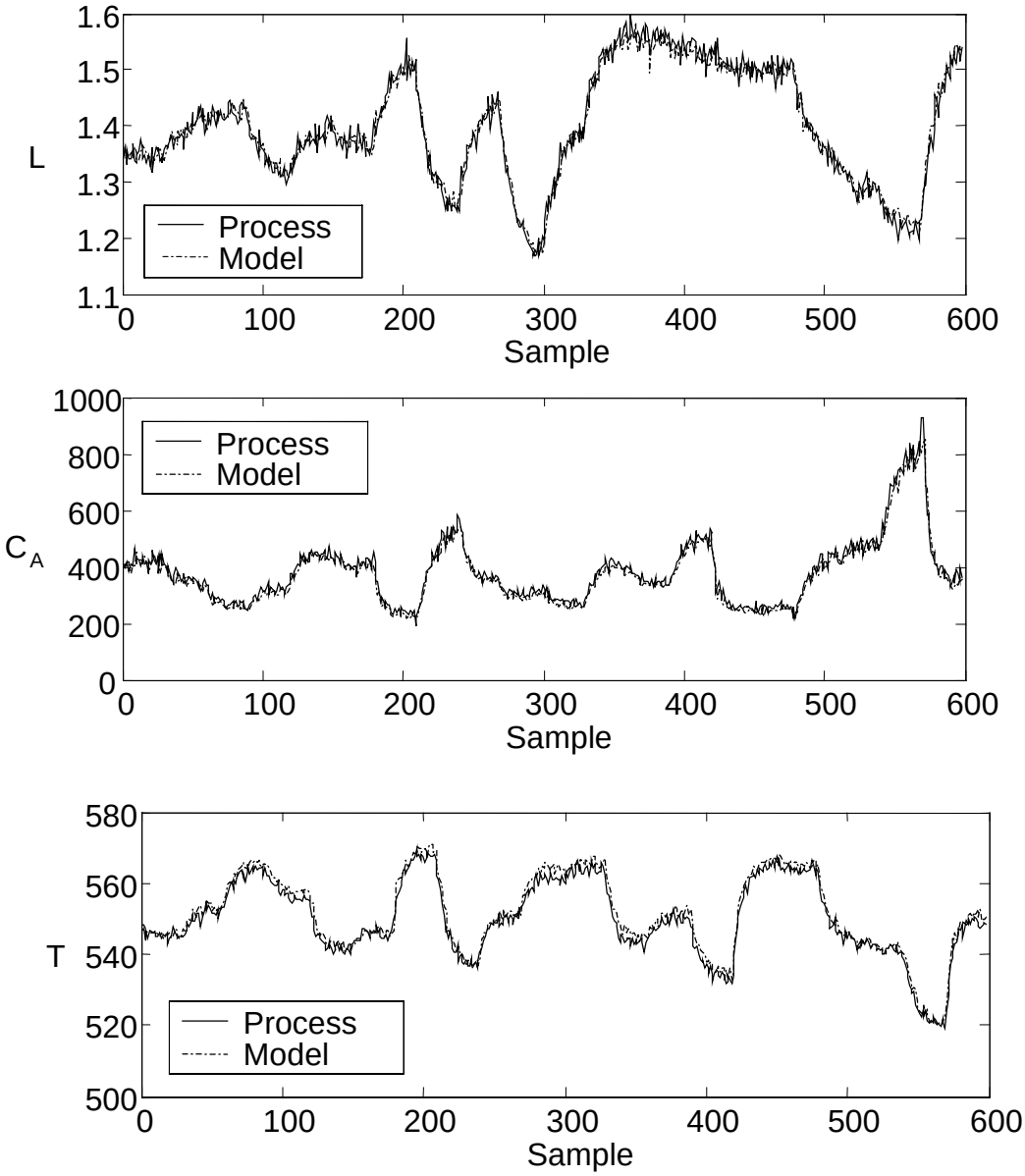


Figure 3.13 Validation result (with noisy process data)

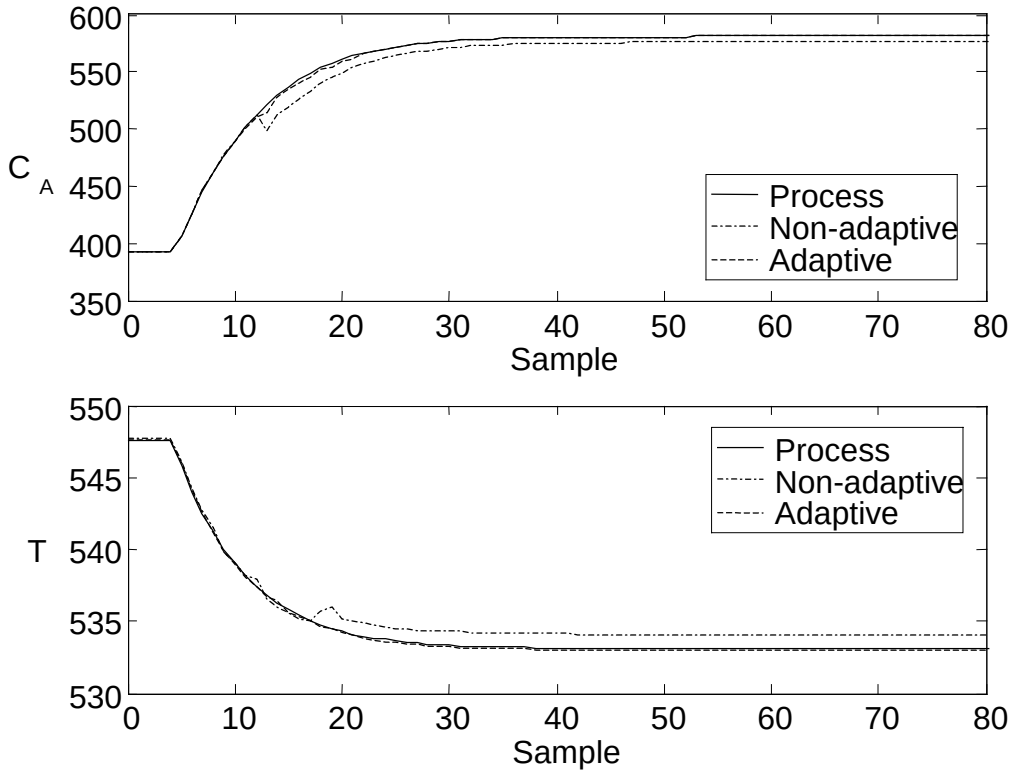


Figure 3.14 Response when v varies from 0 to 0.25

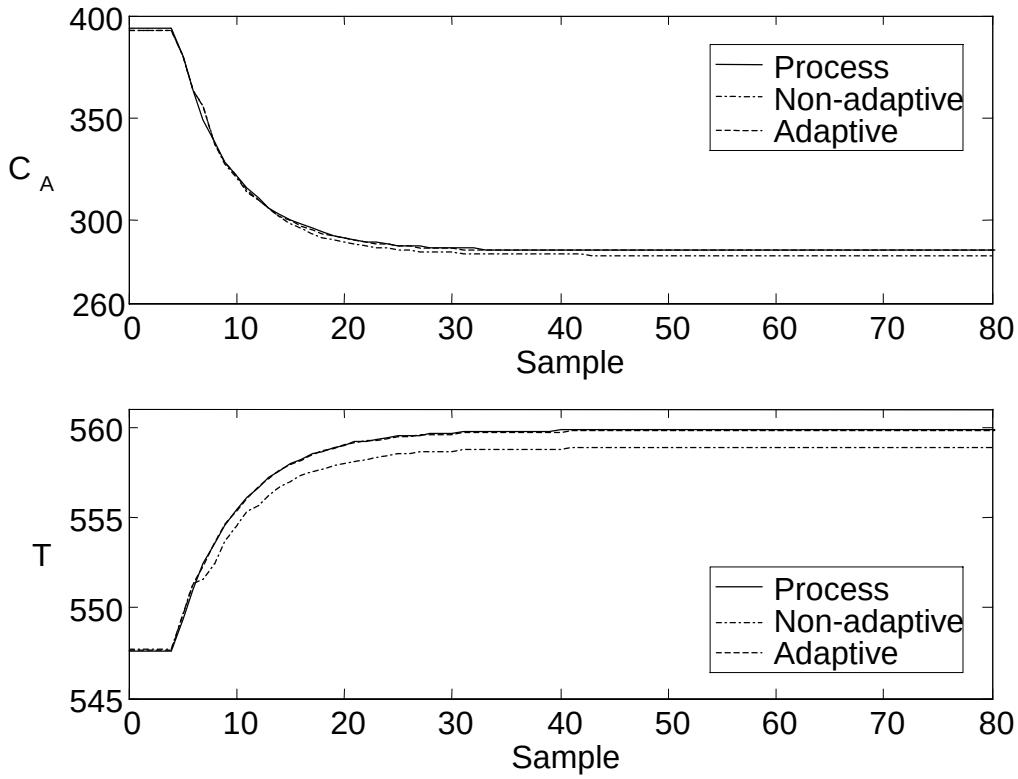


Figure 3.15 Response when v varies from 0 to -0.25

Chapter 4

Robust Controller Design for Nonlinear Processes Using JITL Technique

4.1 Introduction

Accurate models of chemical processes are often difficult to obtain since many of the model parameters are poorly known. Therefore, the common approach to robust controller design is based on empirical models of the process. From the review provided in Chapter 2, it is clear that the robust control problem for linear systems has been solved by representing the process by Laplace transfer function models in which the model parameters are constrained in a pre-specified range (Packard and Doyle, 1993). Doyle et al. (1989) proposed a robust controller design method for a nonlinear CSTR for which a first-principle model is assumed to be available. By assuming that the process/model mismatch is entirely due to the nonlinearities of the process, bounds on the conic sectors to describe the process nonlinearities were developed and used in the standard $M-\Delta$ structure for robust stability analysis. However, the

identification of the conic bounds is not trivial and the resulting robust stability analysis tends to give conservative result, not to mention that first-principle models are generally not available for many chemical processes.

Knapp and Budman (2000) developed a robust control design methodology for nonlinear processes using empirical state affine models, which can be readily transformed into a suitable form for the robust stability analysis. Although their result provides an attractive alternative to Doyle's work where conic sector bounds on the nonlinearities of a process have been employed, the construction of the state affine models is rather tedious and the computational requirements are high. According to Budman and Knapp (2001), in order to obtain a state affine model, a NARMA model is initially constructed from the available process input and output data. Subsequently, an algorithm developed by Diaz and Desrochers (1988) is employed to find the parameters for a truncated Volterra model based on the NARMA model identified previously. Once the Volterra kernels are obtained, a generalized Hankel matrix (or Behaviour matrix) can be developed to find a state affine model (Sontag, 1979). Obviously, the modeling efforts required to identify a state affine model are extensive and thus hampers the application of robust controller design method developed based on such a model.

To circumvent the aforementioned drawbacks, a robust control design methodology for nonlinear processes using JITL technique is developed in this chapter. Similar to the previous work, it is assumed in this work that the nonlinearity is the only source of the model uncertainty. Furthermore, our work is concerned with modest nonlinearities, i.e. linear dynamics play a dominant role in governing the process output behaviour in the operating range of interest, but the linearization errors may be significant. In the proposed method, a nominal ARX model is used to capture

the linear process dynamics and the inevitable modeling error caused by the nonlinearity is approximated by the JITL. The state space realization of the resulting composite model is then reformulated as an uncertain system, by which the robust stability analysis of this uncertain system under PID control is developed. Several literature examples including the CSTR example aforementioned are used to illustrate the proposed method and a comparison with previous result is made.

4.2 Modeling Methodology

As stated earlier, a composite model consisting of a nominal ARX model and JITL models is used to model the nonlinear process in the operating range of interest, where the former can be identified by using the process input and output data around a nominal operating condition and the latter is used to capture the modeling error caused by the process nonlinearity, i.e. the difference between the predicted output of nominal ARX model and actual process output. Suppose that an input sequence $\{u(k)\}$ is injected into the process and that the corresponding output sequence $\{y(k)\}$ is measured. The following composite model is then used to model the nonlinear mapping from $\{u(k)\}$ to $\{y(k)\}$:

$$\hat{y}(k) = y_l(k) + y_{nl}(k) \tag{4.1}$$

where $\hat{y}(k)$ is the output of the composite model, $y_l(k)$ is the predicted output by nominal ARX model, and $y_{nl}(k)$ is the effect of process nonlinearity, i.e. $y(k) - y_l(k)$, approximated by JITL.

Because JITL normally employs a first-order or second-order ARX model, we shall use a second-order model structure for both nominal ARX model and JITL in the subsequent developments. Therefore, $y_l(k)$ and $y_{nl}(k)$ are represented as follows:

$$y_l(k) = \alpha_{l,1} y(k-1) + \alpha_{l,2} y(k-2) + \beta_{l,1} u(k-1) \tag{4.2}$$

$$y_{nl}(k) = \alpha_1 y(k-1) + \alpha_2 y(k-2) + \beta_1 u(k-1) \tag{4.3}$$

Given the process input and output data $\{u(k),\ y(k)\}$, the plant/model mismatch caused by the nonlinearity, i.e. $y(k) - y_l(k)$, can be calculated once the nominal ARX model given in Eq. (4.2) has been identified. Subsequently, JITL technique can be applied to the known sequence $\{y(k) - y_l(k)\}$ by using the reference dataset constructed from $\{u(k),\ y(k)\}$. Upon the successful implementation of JITL algorithm, denote the range of variation for each model coefficient in Eq. (4.3) by: $\alpha_1 \in \left[\alpha_{1,\min} \quad \alpha_{1,\max}\right]$, $\alpha_2 \in \left[\alpha_{2,\min} \quad \alpha_{2,\max}\right]$, and $\beta_1 \in \left[\beta_{1,\min} \quad \beta_{1,\max}\right]$.

Following the standard practice in robust control theory, the model coefficients in Eq. (4.3) are represented by the following equations:

$$\begin{aligned} \alpha_1 &= \overline{\alpha}_1(1 + r_1\delta_1); \quad |\delta_1| \le 1 \\ \alpha_2 &= \overline{\alpha}_2(1 + r_2\delta_2); \ |\delta_2| \le 1 \\ \beta_1 &= \overline{\beta}_1(1 + r_3\delta_3); \quad |\delta_3| \le 1 \end{aligned} \tag{4.4}$$

where $\delta_i \quad (i = 1 \sim 3)$ is the model uncertainty and other relevant parameters are defined as follows:

$$\begin{aligned} \overline{\alpha}_1 &= \frac{\alpha_{1,\min} + \alpha_{1,\max}}{2};\ r_1 = \frac{\alpha_{1,\max} - \alpha_{1,\min}}{\alpha_{1,\min} + \alpha_{1,\max}} \\ \overline{\alpha}_2 &= \frac{\alpha_{2,\min} + \alpha_{2,\max}}{2};\ r_2 = \frac{\alpha_{2,\max} - \alpha_{2,\min}}{\alpha_{2,\min} + \alpha_{2,\max}} \\ \overline{\beta}_1 &= \frac{\beta_{1,\min} + \beta_{1,\max}}{2};\ r_3 = \frac{\beta_{1,\max} - \beta_{1,\min}}{\beta_{1,\min} + \beta_{1,\max}} \end{aligned} \tag{4.5}$$

By using Eqs. (4.2) to (4.5), Eq. (4.1) can be rewritten as:

$$\begin{aligned} \hat{y}(k) = {} & (\alpha_{l,1} + \overline{\alpha}_1 + \overline{\alpha}_1 r_1 \delta_1) y(k-1) + (\alpha_{l,2} + \overline{\alpha}_2 + \overline{\alpha}_2 r_2 \delta_2) y(k-1) \\ & + (\beta_{l,1} + \overline{\beta}_1 + \overline{\beta}_1 r_3 \delta_3) u(k-1) \end{aligned} \tag{4.6}$$

The state space realization of Eq. (4.6) can be expressed by:

$$\mathbf{x}(k+1) = (A_0 + \begin{bmatrix} A_1 \\ \mathbf{0}_{1\times 2} \end{bmatrix}\delta_1 + \begin{bmatrix} A_2 \\ \mathbf{0}_{1\times 2} \end{bmatrix}\delta_2)\mathbf{x}(k) + B_0 u(k) \tag{4.7}$$

$$\hat{y}(k) = (C_0 + C_1\delta_3)\mathbf{x}(k) + D_0 u(k-1) \tag{4.8}$$

where $\mathbf{x}(k) = \begin{bmatrix} x_1(k) & x_2(k) \end{bmatrix}^T$, $\mathbf{0}_{n\times m}$ denote a $n \times m$ zero matrix and

$$A_0 = \begin{bmatrix} \alpha_{l,1} + \overline{\alpha}_1 & \alpha_{l,2} + \overline{\alpha}_2 \\ 1 & 0 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} \overline{\alpha}_1 r_1 & 0 \end{bmatrix};\ A_2 = \begin{bmatrix} 0 & \overline{\alpha}_2 r_2 \end{bmatrix}$$

$$B_0 = \begin{bmatrix} 1 & 0 \end{bmatrix}^T;\ C_0 = \begin{bmatrix} \beta_{l,1} + \overline{\beta}_1 & 0 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} \overline{\beta}_1 r_3 & 0 \end{bmatrix};\ D_0 = 0$$

To account for the modeling error resulting from the approximation of the nonlinear process by the proposed composite model, an additional additive uncertainty δ_a is added to the model output $\hat{y}(k)$ as follows:

$$y(k) = \hat{y}(k) + l_a x_1(k)\delta_a,\ \ |\delta_a| \le 1 \tag{4.9}$$

where l_a is the magnitude of the worst perturbation as obtained by:

$$l_a = \underset{k}{Max} \left| \frac{y(k) - \hat{y}(k)}{\hat{y}(k)/(\beta_{l,1} + \beta_1)} \right| \tag{4.10}$$

It is noted from Eq. (4.8) that the state variable $x_2(k)$ has no effect on the model output $\hat{y}(k)$. This explains why the additive uncertainty δ_a in Eq. (4.9) is only associated with the state variable $x_1(k)$.

After some algebraic manipulation, the composite model by including the additional uncertainty δ_a, i.e. Eqs. (4.7) and (4.9), can be recast into the standard $M-\Delta$ structure as depicted in Figure 4.1, where

$$E_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix};\ E_2 = \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix};\ E_3 = \mathbf{0}_{4\times 4}$$

$$F_1 = \begin{bmatrix} \overline{\alpha}_1 r_1 & 0 \\ 0 & \overline{\alpha}_2 r_2 \\ \overline{\beta}_1 r_3 & 0 \\ l_a & 0 \end{bmatrix};\ F_2 = \mathbf{0}_{4\times 1}$$

$$\tilde{\mathbf{y}}(k) = \begin{bmatrix} \overline{\alpha}_1 r_1 x_1(k) & \overline{\alpha}_2 r_2 x_2(k) & \overline{\beta}_1 r_3 x_1(k) & l_a x_1(k) \end{bmatrix}^T$$

$$\tilde{\mathbf{u}}(k) = \begin{bmatrix} \overline{\alpha}_1 r_1 x_1(k)\delta_1 & \overline{\alpha}_2 r_2 x_2(k)\delta_2 & \overline{\beta}_1 r_3 x_1(k)\delta_3 & l_a x_1(k)\delta_a \end{bmatrix}^T$$

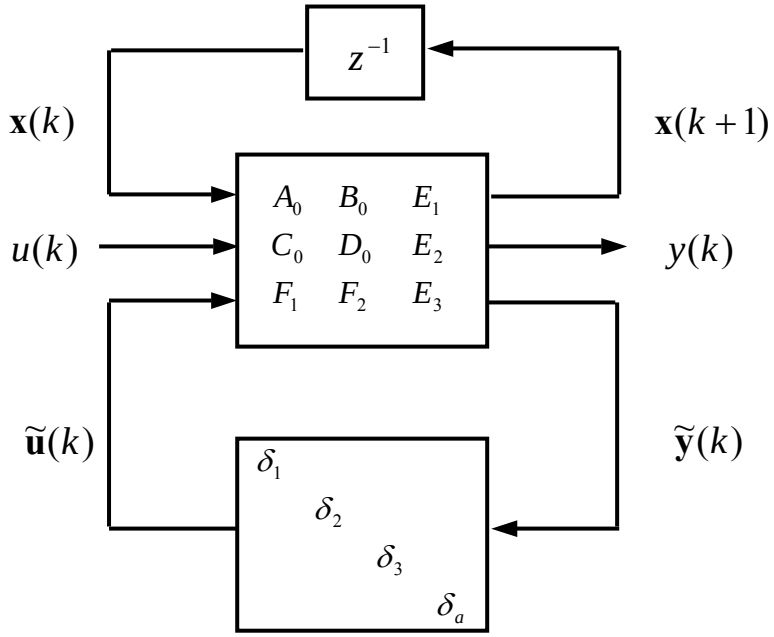


Figure 4.1 *M*-Δ structure for the composite model described by Eqs. (4.7) and (4.9)

The interconnection structure given in Figure 4.1 lays the foundation for the robust stability analysis to be presented in the next section. To this end, it is here to summarize the input and output relationship of the system given in Figure 4.1 as follows:

$$\mathbf{x}(k+1) = A_0\mathbf{x}(k) + B_0u(k) + E_1\tilde{\mathbf{u}}(k) \tag{4.11}$$

$$y(k) = C_0\mathbf{x}(k) + D_0u(k) + E_2\tilde{\mathbf{u}}(k) \tag{4.12}$$

$$\tilde{\mathbf{y}}(k) = F_1\mathbf{x}(k) + F_2u(k) + E_3\tilde{\mathbf{u}}(k) \tag{4.13}$$

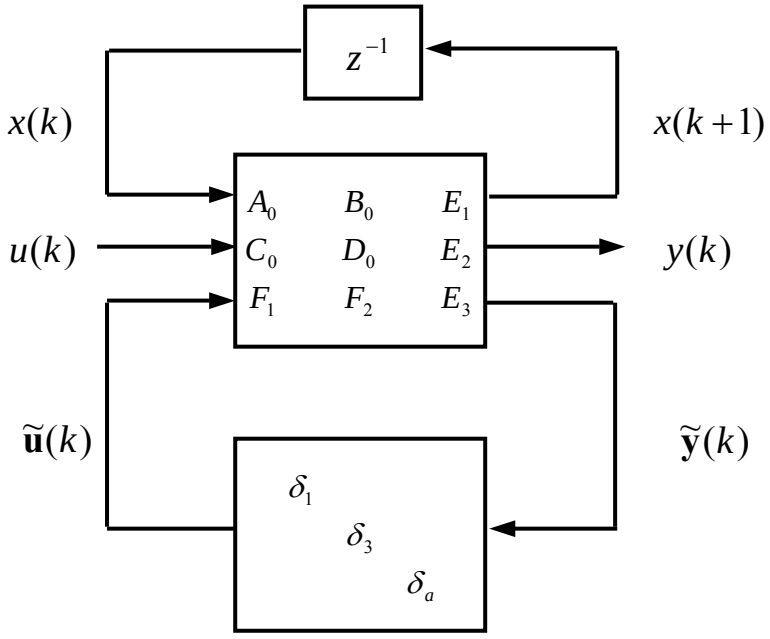


Figure 4.2 *M*-Δ structure for the composite model based on a first-order ARX model

Lastly, when a first-order ARX model is employed for both nominal model and JITL, the corresponding $M-\Delta$ structure can be deducted from Figure 4.1 as shown in Figure 4.2, where

$$A_0 = \alpha_{l,1} + \overline{\alpha}_1;\ \ B_0 = 1$$

$$C_0 = \beta_{l,1} + \overline{\beta}_1;\ \ D_0 = 0$$

$$E_1 = \begin{bmatrix}1 & 0 & 0\end{bmatrix};\ \ E_2 = \begin{bmatrix}0 & 1 & 1\end{bmatrix};\ \ E_3 = \mathbf{0}_{3\times3}$$

$$F_1 = \left[\overline{\alpha}_1 r_1 \quad \overline{\beta}_1 r_3 \quad l_a\right]^T; \quad F_2 = \mathbf{0}_{3\times 1}$$

$$\widetilde{\mathbf{y}}(k) = \left[\overline{\alpha}_1 r_1 x(k) \quad \overline{\beta}_1 r_3 x(k) \quad l_a x(k)\right]^T$$

$$\widetilde{\mathbf{u}}(k) = \left[\overline{\alpha}_1 r_1 x(k)\delta_1 \quad \overline{\beta}_1 r_3 x(k)\delta_3 \quad l_a x(k)\delta_a\right]^T$$

4.3 Robust Stability Analysis

Since PID controller is the most commonly used controller in the process industries, it is considered in the ensuing robust stability analysis. To facilitate the subsequent development, PID controller is represented by the following state space equation:

$$\mathbf{\Psi}(k+1) = A_c \mathbf{\Psi}(k) + B_c e(k) \tag{4.14}$$

$$u(k) = C_c \mathbf{\Psi}(k) + D_c e(k) \tag{4.15}$$

where $\mathbf{\Psi}(k)$ is a 2×1 state variable vector of PID controller, $e(k)$ is the tracking error, i.e. the difference between the set-point and process output, and other model parameters are given by:

$$\begin{aligned} A_c &= \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}; \qquad & B_c &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ C_c &= \begin{bmatrix} \dfrac{k_c}{\tau_I} - k_c\tau_D & k_c\tau_D \end{bmatrix}; \qquad & D_c &= k_c + \dfrac{k_c}{\tau_I} + k_c\tau_D \end{aligned} \tag{4.16}$$

were k_c, τ_I, and τ_D are the PID parameters.

By using Eqs. (4.11) to (4.15), the resulting closed-loop system can be represented by the following augmented state space equation:

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{\Psi}(k+1) \end{bmatrix} = M_{11} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{\Psi}(k) \end{bmatrix} + M_{12} \widetilde{\mathbf{u}}(k) \tag{4.17}$$

$$\tilde{\mathbf{y}}(k) = M_{21}\begin{bmatrix}\mathbf{x}(k)\\ \mathbf{\Psi}(k)\end{bmatrix} + M_{22}\tilde{\mathbf{u}}(k) \tag{4.18}$$

were the matrices M_{11}, M_{12}, M_{21}, and M_{22} are given by:

$$M_{11} = \begin{bmatrix} A_0 - B_0 D_c C_0 & B_0 C_c \\ -B_c C_0 & A_c \end{bmatrix}$$

$$M_{12} = \begin{bmatrix} E_1 - B_0 D_c E_2 \\ -B_c E_2 \end{bmatrix}$$

$$M_{21} = \begin{bmatrix} F_1 & \mathbf{0}_{4\times 2} \end{bmatrix}$$

$$M_{22} = \mathbf{0}_{4\times 4}$$

The uncertain closed-loop system described by Eqs. (4.17) and (4.18) is now amenable for robust stability analysis by referring to Figure 4.3(a), where the delay operator z^{-1} can be treated as a perturbation with magnitude bounded by one and is therefore substituted by an artificial uncertainty δ_z with $|\delta_z| \le 1$. Because $\begin{bmatrix}\mathbf{x}(k+1) & \mathbf{\Psi}(k+1)\end{bmatrix}^T$ is a 4×1 vector, δ_z is thus a repeated perturbation and is appropriately denoted by $\delta_z I_4$ where I_4 stands for a 4×4 identity matrix. Consequently, Figure 4.3 (a) is equivalent to the $M-\Delta$ structure as shown in Figure 4.3(b). With the interconnection matrix $\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$ and the uncertainty structure $\Delta = diag\begin{bmatrix}\delta_z I_4 & \delta_1 & \delta_2 & \delta_3 & \delta_a\end{bmatrix}$ available, robust stability condition for the uncertain closed-loop system aforementioned can be obtained by applying the structured singular value test as given by the Theorem 2.1 in Chapter 2 to obtain:

$$\mu_{\Delta}\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} < 1 \tag{4.19}$$

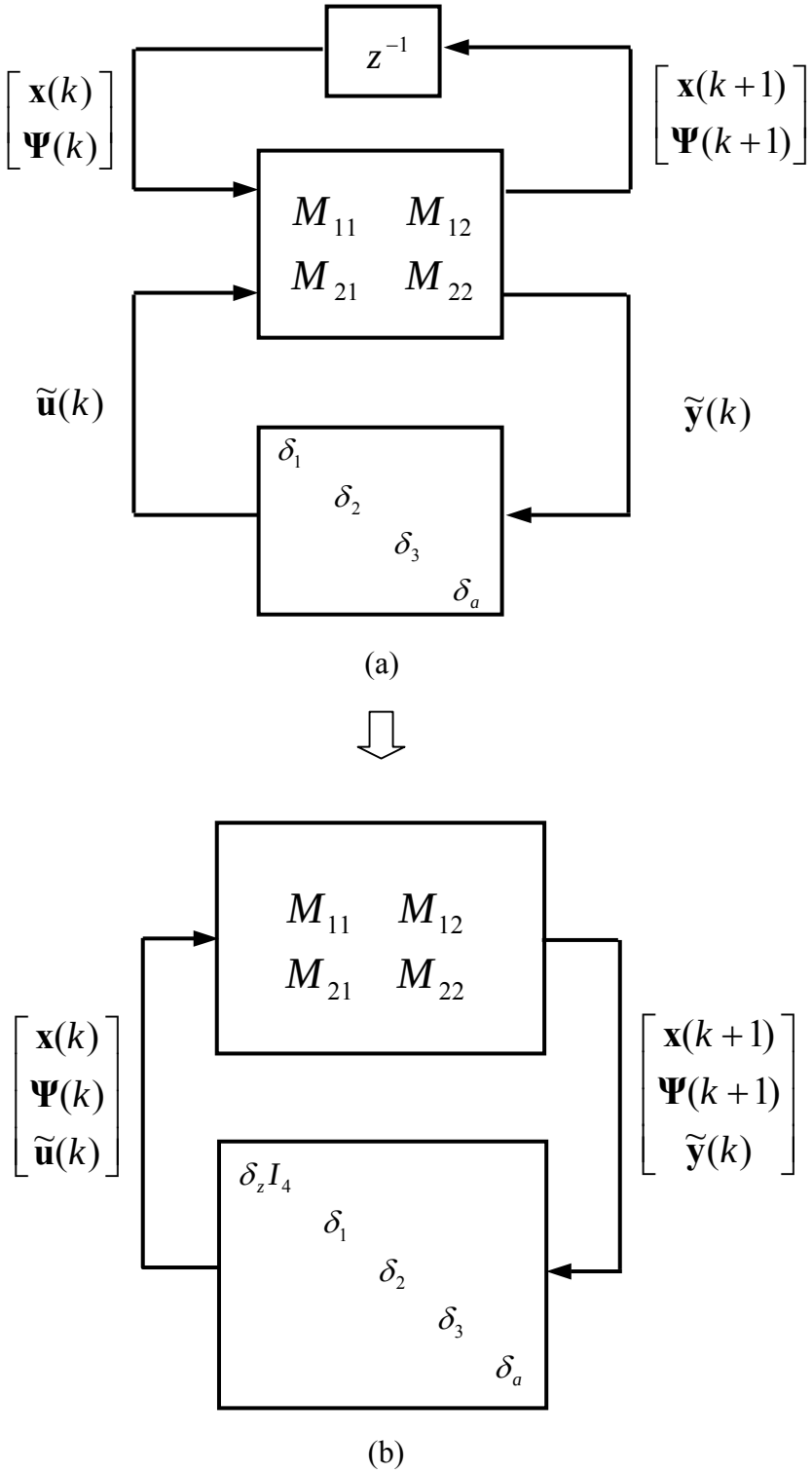


Figure 4.3 *M*-Δ structure for the uncertain closed-loop system

Lastly, it is evident that Eq. (4.19) is applicable to the case where the composite model is constructed by a first-order ARX model and a PI controller is desired. In this situation, the relevant model parameters as provided at the end of last section and the following parameters for PI controller, $A_c = 1$, $B_c = 1$, $C_c = \frac{k_c}{\tau_I}$, and $D_c = k_c + \frac{k_c}{\tau_I}$, are used to obtain the interconnection matrix. In addition, the corresponding perturbation structure is simplified as $diag\left[\delta_z I_2 \quad \delta_1 \quad \delta_3 \quad \delta_a\right]$ as a result of the number of state variable in both composite model and controller being reduced to one.

4.4 Examples

Example 1 Consider the van de Vusse reactor as described by

$$\frac{dC_A}{dt} = -k_1 C_A - k_3 C_A^2 + \frac{F}{V}(C_{Af} - C_A) \tag{4.20}$$

$$\frac{dC_B}{dt} = k_1 C_A - k_2 C_B - \frac{F}{V} C_B \tag{4.21}$$

where the model parameters are detailed in Chapter 3. The control objective of this example is to design a robust PI controller to manipulate the inlet flow rate $F\,(=u)$ to regulate $C_B\,(=y)$ when the operating space is $F \in \left[5 \quad 25\right]$ and $C_B \in \left[0.3783 \quad 1\right]$. To construct the composite model, a first-order ARX model is adopted. By using the process input and output data obtained around the nominal operating condition, the parameters of the nominal ARX model are obtained by $\alpha_{l,1} = 0.7405$ and $\beta_{l,1} = 0.0313$. To model the process nonlinearity by the JITL, a different set of input and output data is generated within the operating space as illustrated in Figure 4.4,

where y_l is the predicted output by the nominal ARX model subject to the same input signal and $y - y_l$ is the modeling error caused by the process nonlinearity. To model the nonlinearity effect by the JITL with $\gamma = 0.9$, $k_{\min} = 10$, and $k_{\max} = 60$ chosen, the resulting model parameters are obtained as $\alpha_1 \in [-0.3822 \quad 0.1317]$ and $\beta_1 \in [-0.0223 \quad 0.0015]$. In addition, to quantify the modeling error of the composite model, the worst perturbation is calculated as $l_a = 0.0012$. Figure 4.5 illustrates that the resulting composite model gives reasonably good prediction in the validation test by using input and output data different from that used in constructing the composite model.

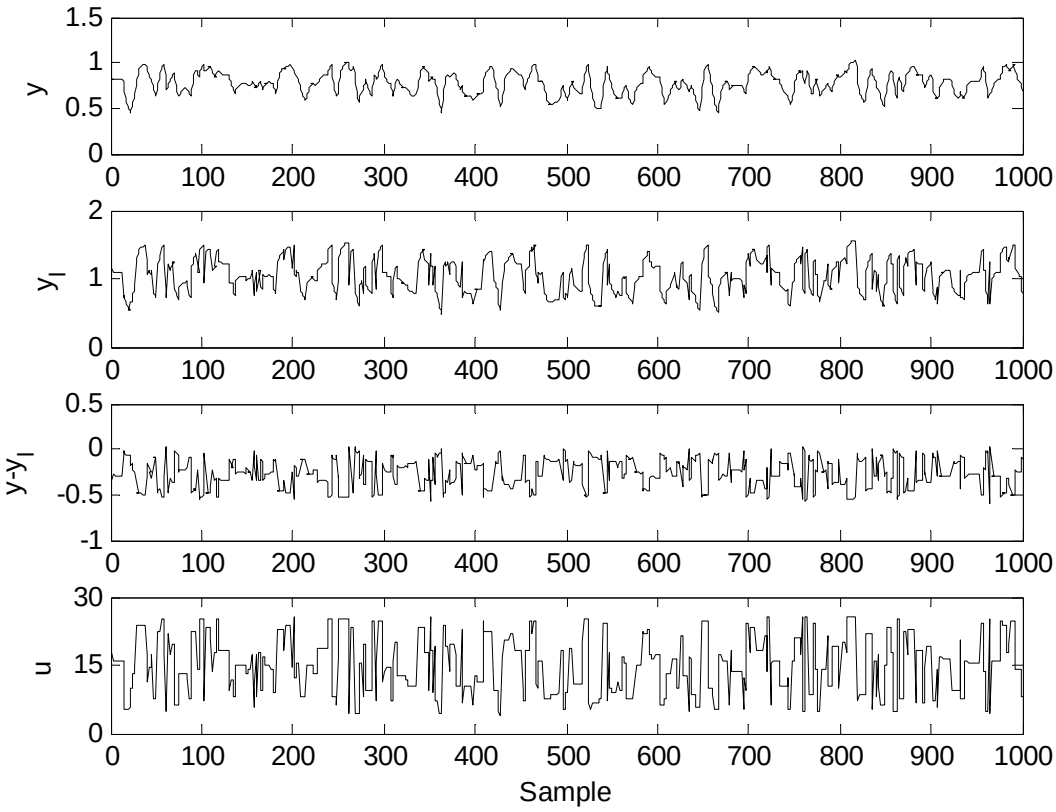


Figure 4.4 Input-output data used for constructing the database for JITL

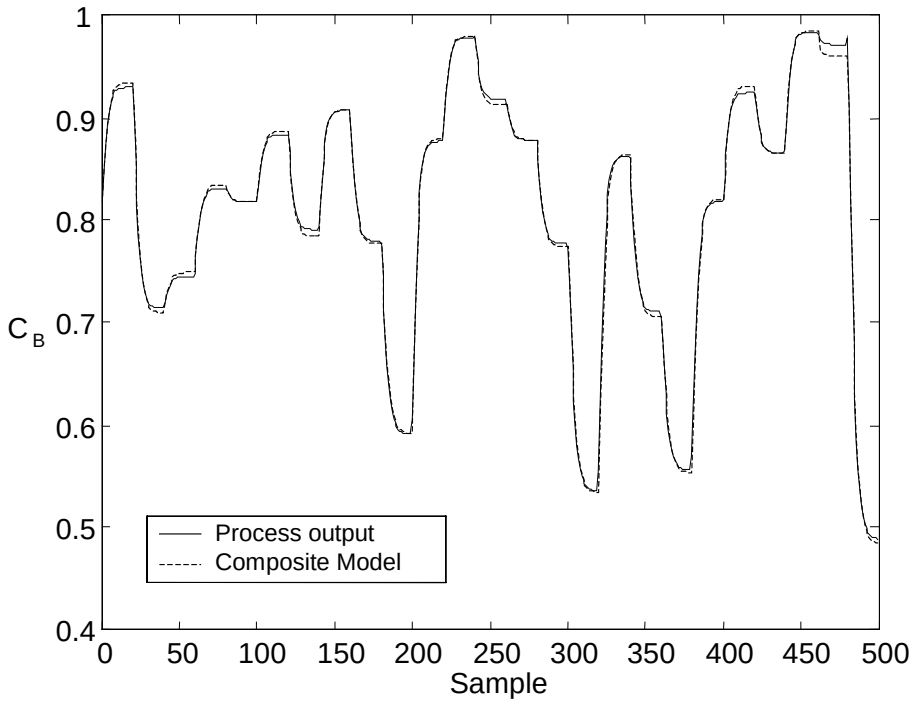


Figure 4.5 Validation result for the composite model

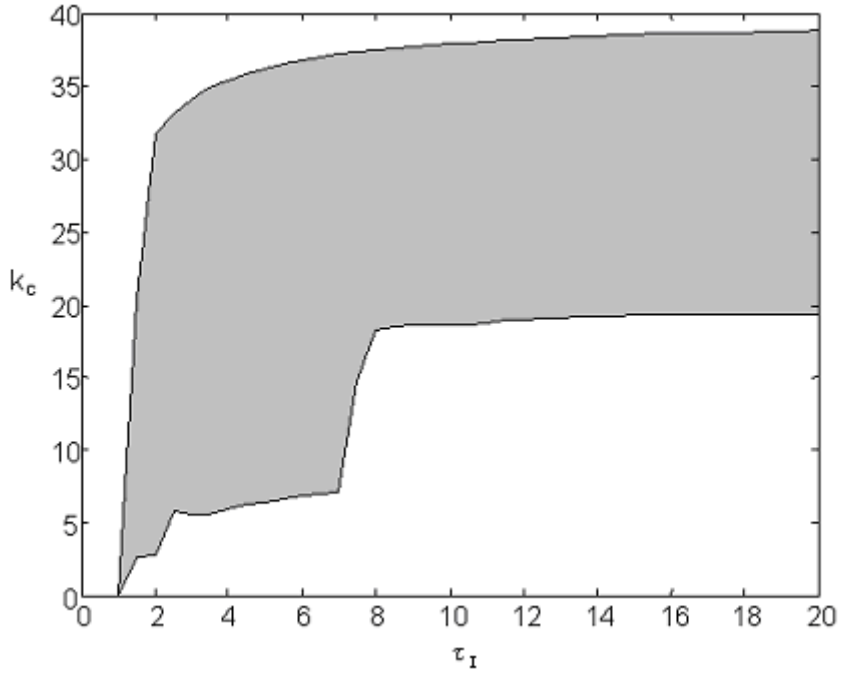


Figure 4.6 Robust stability region (shadow) for van de Vusse reactor

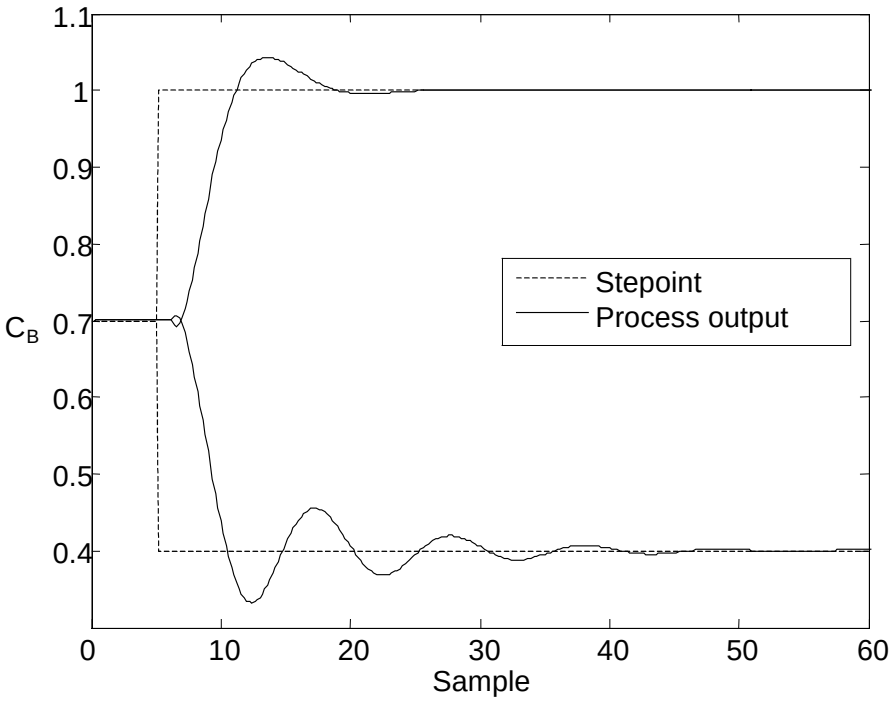


Figure 4.7 Closed-loop responses for set-point changes from $C_B = 0.7$ to 1.0 (top) and 0.4 (bottom) with PI parameters $k_c = 31.8$ and $\tau_I = 2$

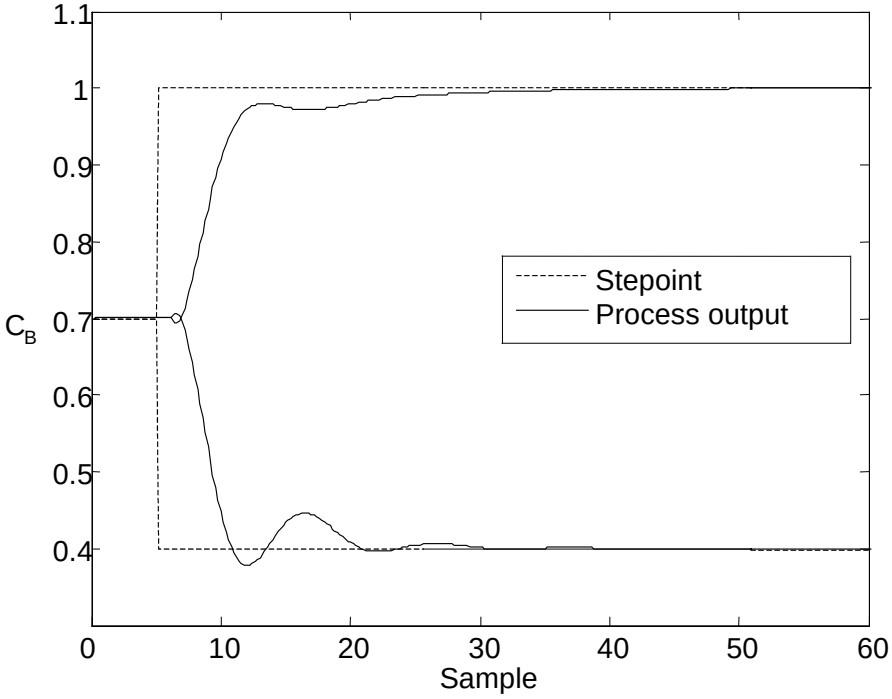


Figure 4.8 Closed-loop responses for set-point changes from $C_B = 0.7$ to 1.0 (top) and 0.4 (bottom) with PI parameters $k_c = 35.4$ and $\tau_I = 4$

To proceed with the proposed robust PI design, Eq. (4.19) is applied to obtain the PI parameters that guarantee the robust stability over the operating space aforementioned. Figure 4.6 shows the resulting robust stability region in $k_c - \tau_I$ space. To verify the proposed analysis result, two viable PI designs are evaluated by the set-point changes covering the entire operating space. The first PI parameters chosen are $k_c = 31.8$ and $\tau_I = 2$, where the former is the maximum allowable k_c for the robust stability criterion Eq. (4.19) when $\tau_I = 2$. Figure 4.7 shows the responses of this controller subject to the set-point changes from $C_B = 0.7$ to $C_B = 1$ and $C_B = 0.4$, respectively. It is evident that this PI controller is able to maintain closed-loop stability over the entire operating space. Similar observation is also obtained for PI controller with $k_c = 36.2$ and $\tau_I = 4$, as illustrated in Figure 4.8. Again, the proportional gain chosen for this controller is the maximum allowable k_c for Eq. (4.19) when $\tau_I = 4$.

Example 2 The second application considered is a distillation process as described by (Gao et al., 2000):

$$y(k) = 0.757y(k-1) + 0.243g(u(k-1)) \tag{4.22}$$

$$g(x) = 1.04x - 14.11x^2 - 16.72x^3 + 562.7x^4 \tag{4.23}$$

The top column composition $y(\%)$ is the process output and the reflux flow rate $u(\text{mol/min})$ is the process input. The operating space under consideration is given by $u \in [-0.05 \quad 0.01]$ and $y \in [-0.0817 \quad 0.009]$. Again, input and output data around the nominal operating condition are generated to identify the nominal ARX model with parameters obtained by $\alpha_{l,1} = 0.7686$ and $\beta_{l,1} = 0.2183$. To construct JITL to model the process nonlinearity, a different set of input and output data as depicted in

Figure 4.9 is generated within the operating space. With the parameters for JITL chosen as $\gamma = 0.95$, $k_{\min} = 8$, and $k_{\max} = 60$, the range of variation of the resulting model parameters is obtained as $\alpha_1 \in [-0.3750 \quad 0.0480]$ and $\beta_1 \in [-0.0133 \quad 0.6614]$. Also, the modeling error of the composite model is calculated as $l_a = 0.0543$. To compare the predictive performance of nominal ARX model and composite model, another set of data different from that given in Figure 4.9 is used in the validation test. As can be seen from Figure 4.10, the composite model gives better prediction than the nominal ARX model over the entire operating space.

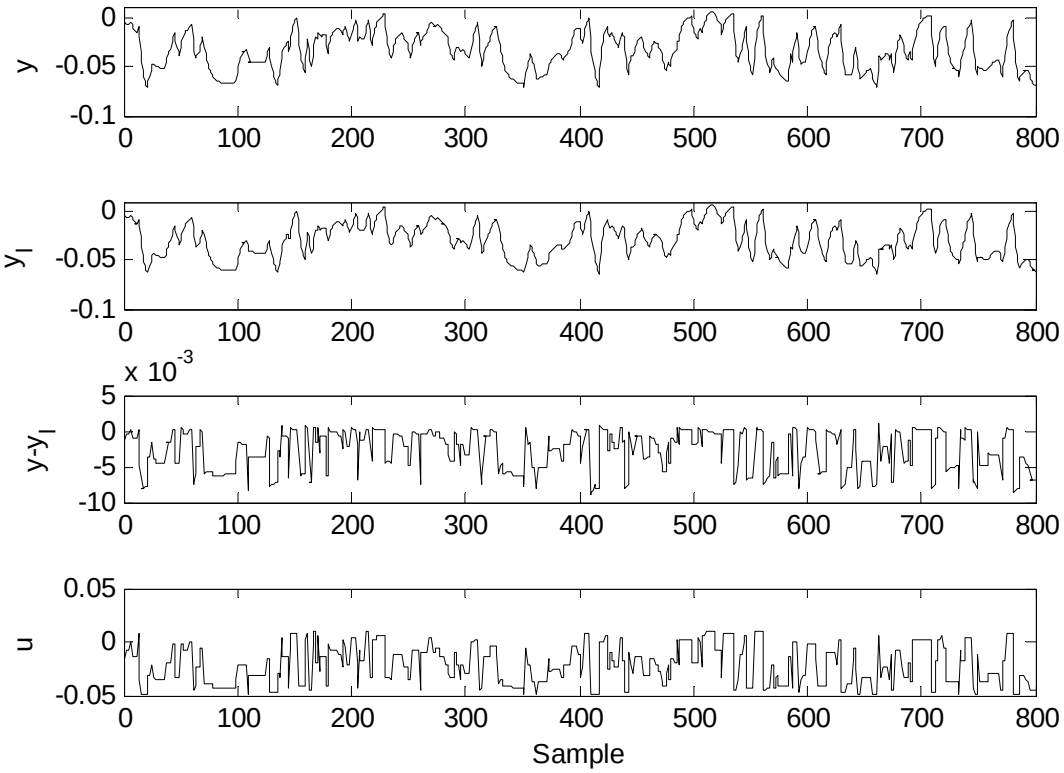


Figure 4.9 Input-output data used for constructing the database for JITL

Figure 4.11 shows the viable PI designs in $k_c - \tau_I$ space that meets the robust stability criterion Eq. (4.19) for the process under PI control. To verify the proposed analysis result, two PI controllers obtained in the robust stability region, i.e. $(k_c \quad \tau_I) = (1.24 \quad 2.5)$ and $(1.32 \quad 4)$, are chosen for the set-point changes from $y = -0.035$ to $y = 0.009$ and $y = -0.08$, respectively. Similar to the previous example, the proportional gains of these two PI controllers are the maximum allowable k_c for Eq. (4.19) when their respective integral time constants are kept constant. It is evident from Figures 4.12 and 4.13 that these two PI controllers give stable responses for the set-point changes ranging over the entire operating space.

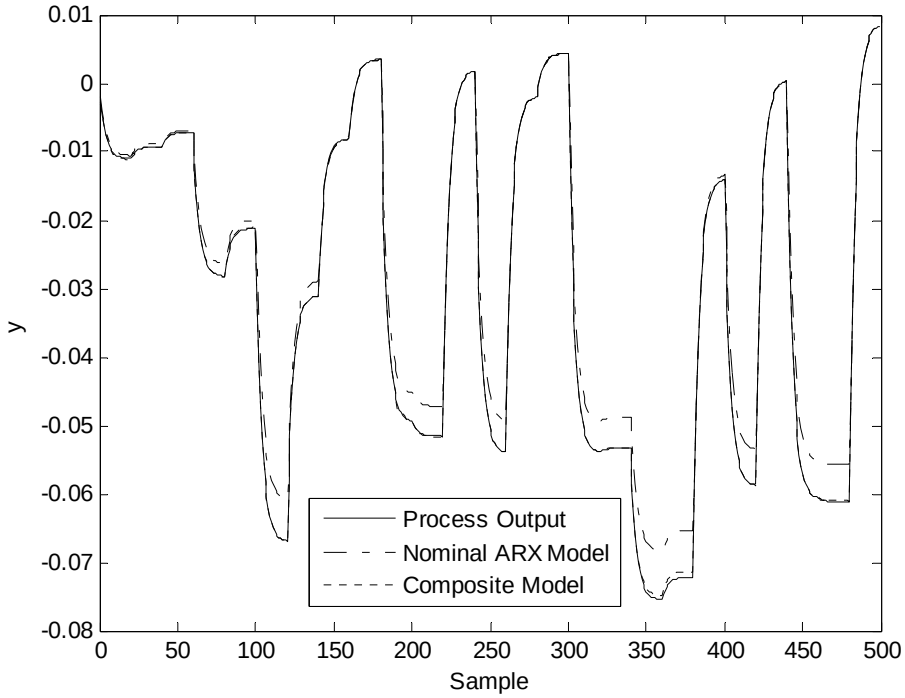


Figure 4.10 Validation results for nominal ARX model and composite model

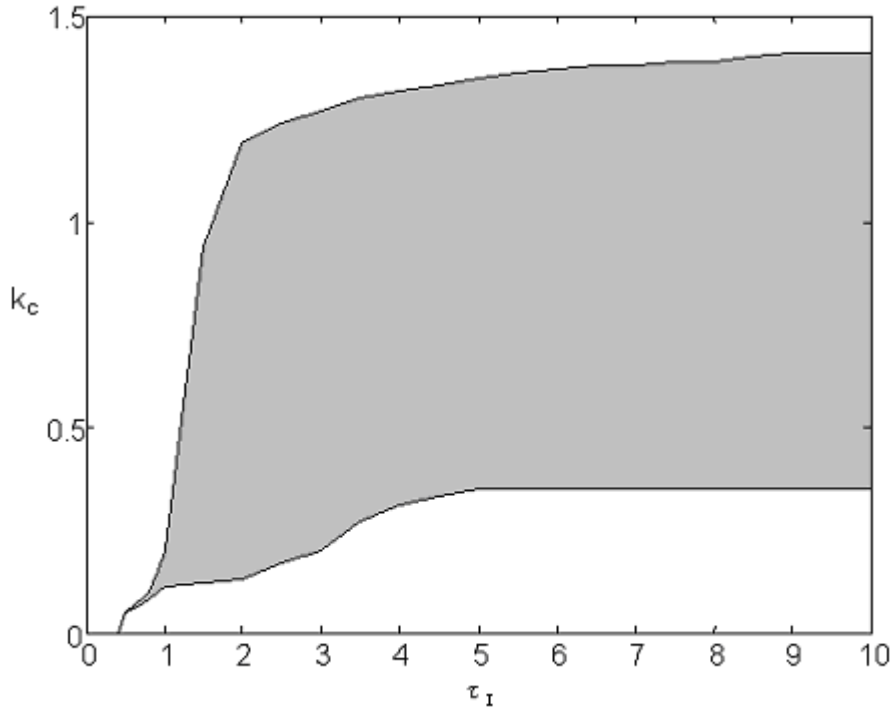


Figure 4.11 Robust stability region (shadow) for distillation process

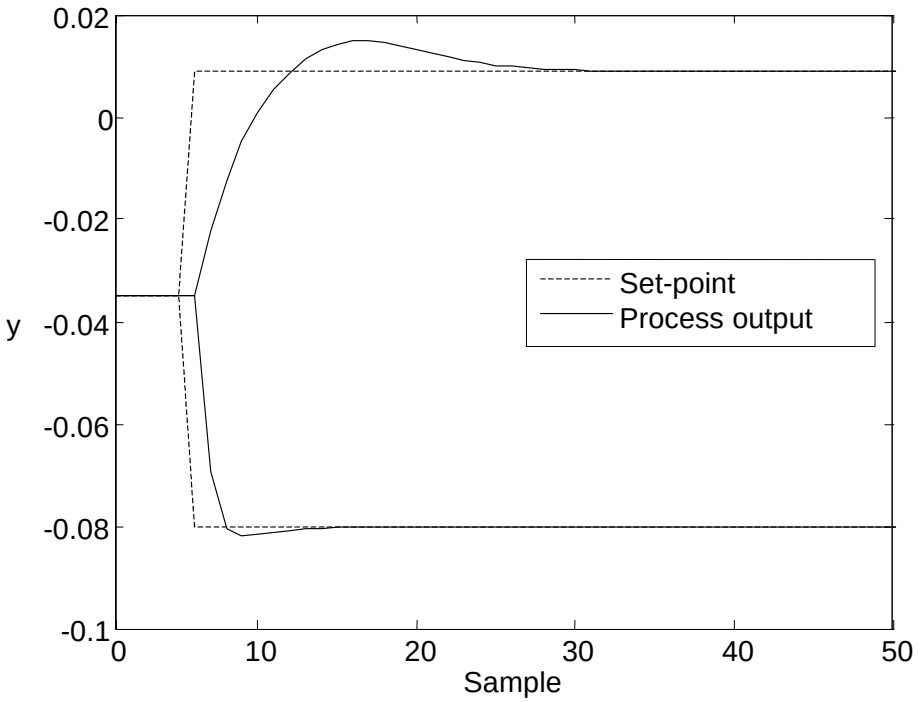


Figure 4.12 Closed-loop responses for set-point changes from $y = -0.035$ to 0.009 (top) and -0.08 (bottom) with PI parameters $k_c = 1.24$ and $\tau_I = 2.5$

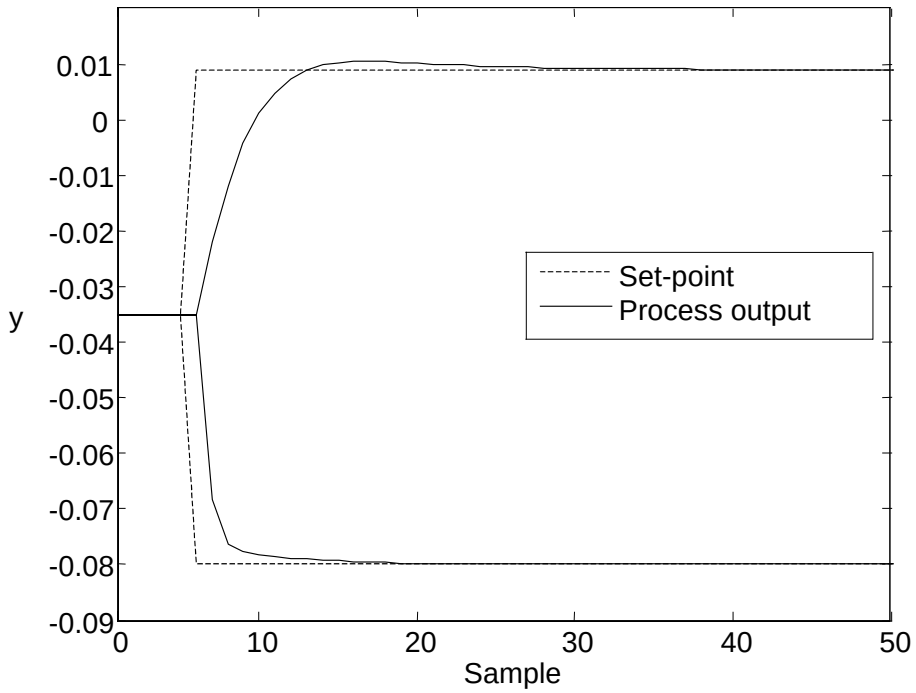


Figure 4.13 Closed-loop responses for set-point changes from $y = -0.035$ to 0.009 (top) and -0.08 (bottom) with PI parameters $k_c = 1.32$ and $\tau_I = 4$

Example 3 Consider the CSTR as described by the following equations (Doyle et al., 1989):

$$\dot{x}_1 = -x_1 + Da(1 - x_1)e^{\frac{x_2}{1+x_2/\nu}} \tag{4.24}$$

$$\dot{x}_2 = -x_2 - b(x_2 - x_c) + BDa(1 - x_1)e^{\frac{x_2}{1+x_2/\nu}} \tag{4.25}$$

where x_1 and x_2 are the dimensionless concentration and temperature of the reactor, and x_c is the cooling temperature selected as manipulated variable while x_1 is the controlled variable. The process has one stable steady state when $Da = 0.072$, $B = 1$, $b = 0.3$, and $\nu = 20$. The following operating space $x_c \in [5 \quad 23]$ and $x_1 \in [0.1969 \quad 0.8781]$ is considered in this example. To construct the composite model, a nominal ARX model with parameters $\alpha_{l,1} = 0.7216$ and $\beta_{l,1} = 0.1231$ is

obtained around the nominal operating condition. With another set of input and output data generated over the operating space, the nonlinearity effect is captured by constructing JITL with its parameter chosen as $\gamma = 0.75$, $k_{\min} = 10$, and $k_{\max} = 60$. The resulting model parameters are obtained by $\alpha_1 \in [-0.3363,\ \ 0.0191]$ and $\beta_1 \in [-0.0217,\ \ 0.1112]$. By using Eq. (4.10), the modeling error of the composite model is obtained as $l_a = 0.0514$.

Similar to the previous two examples, Figure 4.14 shows the robust stability analysis result for this example, while Figures 4.15 and 4.16 verify the viable PI designs that meet the design criterion Eq. (4.19) by choosing two PI controllers with $(k_c \quad \tau_I) = (32.4 \quad 1.1)$ and $(41.4 \quad 3)$, respectively. It is obvious that these two controllers are able to achieve stable responses for the set-point changes in the operating space of interest. Lastly, it is worthy pointing out that this example was studied by Knapp and Budman (2000) who developed a robust stability analysis test based on the state affine model. Compared with the viable PI designs that satisfy the robust stability condition reported in their paper, for example the respective maximum allowable k_c obtained for the $\tau_I = 1.1$ and $\tau_I = 3$ are 18 and 27, it is clear that the proposed method is less conservative than that given in Knapp and Budman, not to mention the tedious modeling procedure required to build a state affine model.

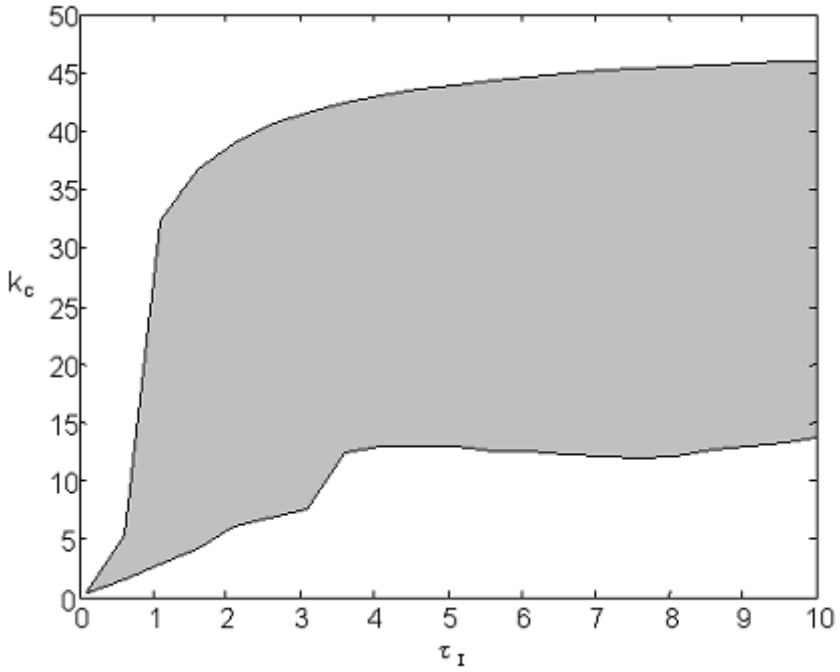


Figure 4.14 Robust stability region (shadow) for CSTR process

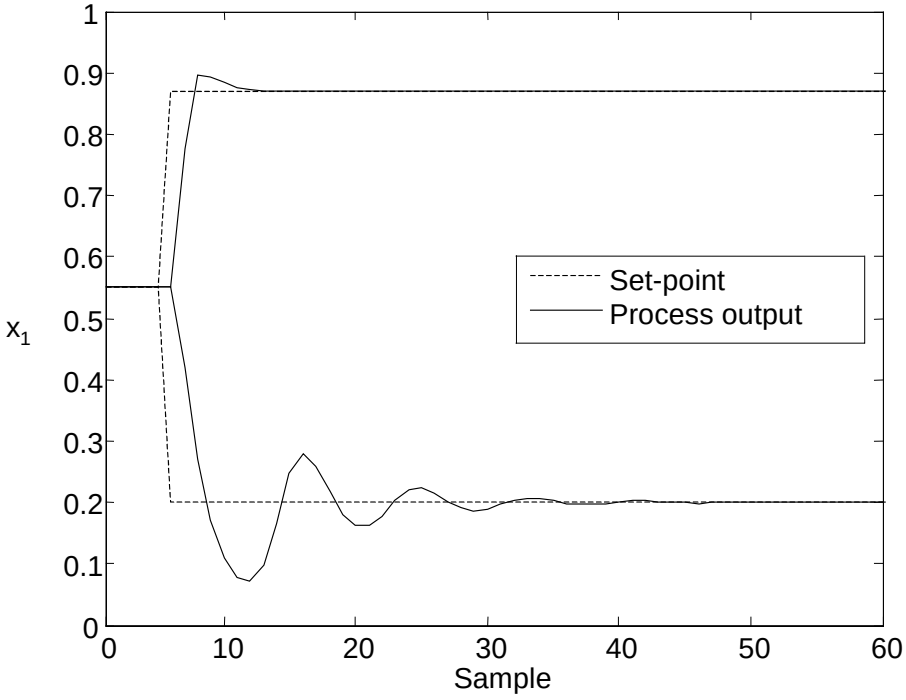


Figure 4.15 Closed-loop responses for set-point changes from $x_1 = 0.55$ to 0.87 (top) and 0.2 (bottom) with PI parameters $k_c = 32.4$ and $\tau_I = 1.1$

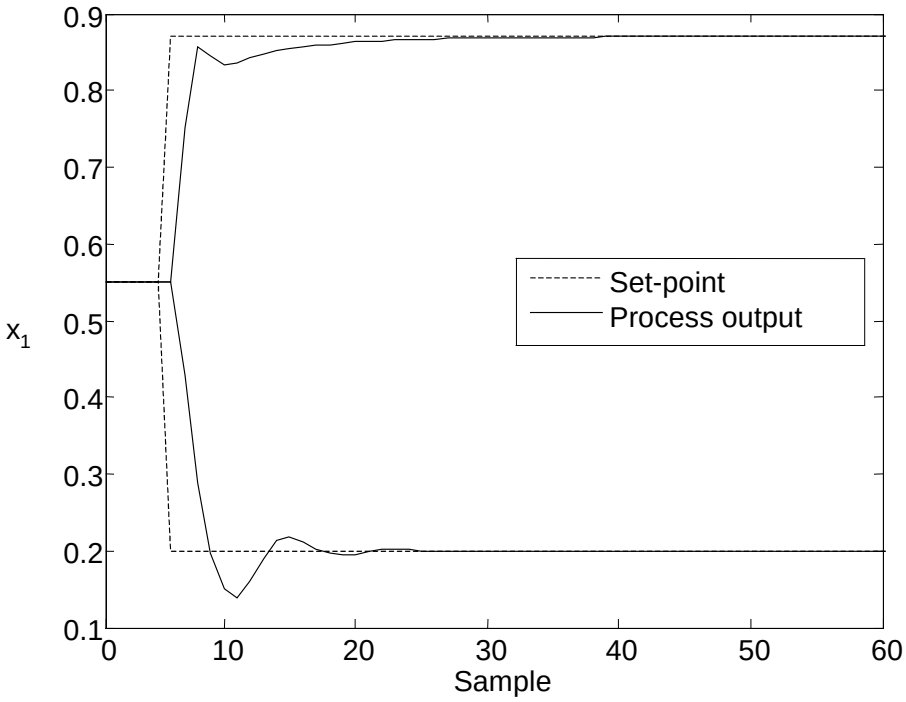


Figure 4.16 Closed-loop responses for set-point changes from $x_1 = 0.55$ to 0.87 (top) and 0.2 (bottom) with PI parameters $k_c = 41.4$ and $\tau_I = 3$

4.5 Conclusion

Based on the assumption that nonlinearity is the only source of the model uncertainty, this chapter develops a new methodology for robust PID controller design. In the proposed method, a composite model is first constructed to model the process dynamics in the operating space of interest. This composite model consists of a nominal ARX model used to capture the linear process dynamics and JITL used to approximate the modeling error caused by process nonlinearity. The state space realizations of the resulting composite model and PID controller are then reformulated as an uncertain system, which can be recast into the standard $M - \Delta$ structure, by which the robust stability analysis by using the structured singular value test can be developed as the design criterion for robust PID controller design. The application of

this methodology is illustrated by three literature examples. Simulation results indicate that the proposed robust stability analysis result can be used to design the robust controllers, which assure the closed-loop stability for controlling the processes with moderate nonlinearity over a given operating space.

Chapter 5

Adaptive Single-Neuron Controller Design

5.1 Introduction

In chemical and biochemical industries, most processes are inherently nonlinear, however most controller design methods are based on linear control techniques. The prevalence of linear control strategies is partly due to the fact that the process dynamics can be approximated by a linear model around the nominal operating region. Furthermore, theories for linear control systems are quite well developed so that linear control techniques are widely accepted. In contrast, controller design for nonlinear models is considerably more difficult than that for linear models. However, for a process that exhibits significant nonlinearities, linear control design methodologies may not be adequate to achieve a good control performance. This has led to an increasing interest in the nonlinear and adaptive controller designs for the nonlinear dynamic processes in the last two decades (Bequette, 1991; Stephanopoulos and Han, 1996; Linkens and Nyongesa, 1996; Ogunnaike and Raymond, 1996). Among various methods, neural network (NN) based control techniques

(Stephanopoulos and Han, 1996; Linkens and Nyongesa, 1996) have become one of the most popular methods due to the ability of neural network to model any nonlinear function to an arbitrary degree of accuracy (Cybenko, 1989). Narendra and Parthasarathy (1990) applied NN for system identification and incorporated it into an adaptive control structure to control nonlinear dynamic systems. Hernandez and Arkun (1992) constructed NN based on the state space realization model for nonlinear system control. Nahas et al. (1992) proposed an internal model control scheme integrated with NN for the control of nonlinear process. te Braake et al. (1998) made use of the NN and feedback linearization technique to transform the nonlinear process into an equivalent linear system in order to simplify the controller design problem.

Most approaches of using neural network for nonlinear process control make use of two neural networks, i.e., one NN is used to model nonlinear dynamics of the process and the other NN acts as a controller. As mentioned in Chapter 2, NN has drawback in on-line nonlinear process modeling. Furthermore, previous NN based control schemes need to deal with the issue of updating a large number of weights. Consequently, a nonlinear optimization routine is required for this purpose, which may lead to the problems of large computing effort and poor convergence. Thus, it is desirable to keep the NN based control scheme as simple as possible.

It is well documented that proportional-integral-derivative (PID) controller has been widely employed for about 80% or more control loops in process industry (Astrom and Hagglund 1995) due to its simple structure and clear physical meaning of controller parameters. However, like other linear control techniques, it is difficult to obtain good control performance for nonlinear systems simply using a fixed-parameter PID controller. To overcome this drawback, researchers proposed various adaptive PID tuning methods based on neural networks because of their learning and

adaptive abilities (Jeon and Lee, 1996). For example, Omatu et al. (1995) and Yeo and Kwon (1999) treated the PID parameters as the output nodes of the neural networks and consequently PID controller can be adjusted on-line at each sampling instant based on certain learning algorithms. Chen and Huang (2004) proposed the use of an off-line neural network to model the nonlinear process and then an instantaneous linearization of this neural network at each sampling instant is conducted to obtain the linearized model. The parameters of linearized NN are subsequently used to tune the PID parameters. Andrasik et al. (2004) also made use of two neural networks for online tuning of PID controller. In their method, a hybrid model consisting of a neural network and a simplified first-principle model is constructed as an estimator, while the second neural network is a neural PID-like controller, which is pre-trained off-line as a black-box model inverse of the controlled process. However, most of the approaches mentioned above are computationally expensive as inevitably required by the associated highly complex learning algorithms developed, which hampers the use of these methods in practical applications.

In this chapter, we present an easy-to-implement controller design strategy for nonlinear processes. To keep the controller structure as simple as possible, a single-neuron controller that mimics a PID controller is considered. Adaptive learning algorithm is derived to adjust controller parameters by using the information from JITL. Literature examples are presented to illustrate the proposed control strategy and a comparison with its conventional counterparts is made.

5.2 JITL Based Adaptive Single-Neuron Controller Design

5.2.1 Control strategy

The proposed adaptive single-neuron (ASN) control scheme is depicted in Figure 5.1, where the JITL model acts as an estimator to provide the most up-to-date process information so that the ASN controller can learn the current dynamics of the process and adapt its parameters to compensate for the changing operating condition.

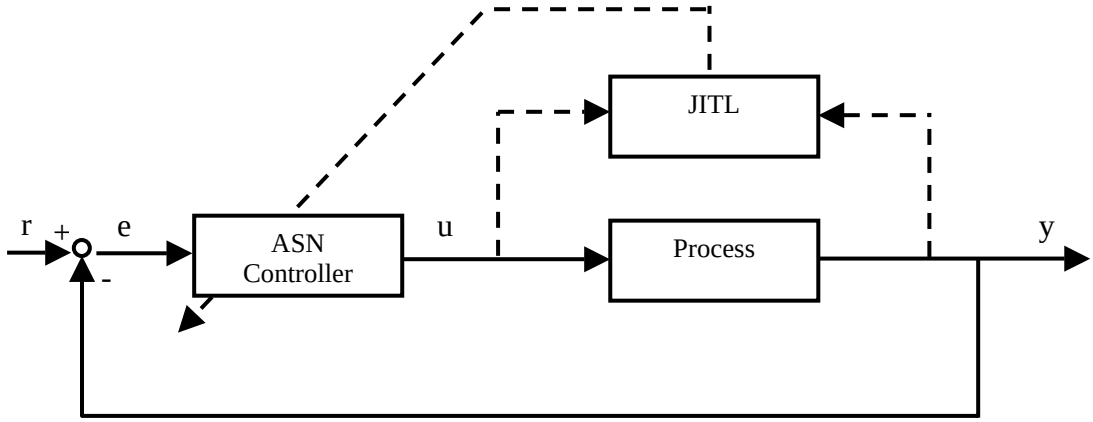


Figure 5.1 JITL based ASN control system

Inspired by the simple and widely used PID structure, a single neuron is adopted in the proposed controller structure as shown in Figure 5.2, where $e(k)$ is the error between process output and its set-point at the k-th sampling instant, $\Delta e(k) = e(k) - e(k-1)$ is the difference between the current and previous error, $\delta e(k) = \Delta e(k) - \Delta e(k-1)$, and w_i $(i = 1 \sim 3)$ are the neuron weights.

From Figure 5.2, control law of the ASN controller is obtained as follows:

$$u(k) = u(k-1) + \Delta u(k)$$

$$\Delta u(k) = w_1(k)e(k) + w_2(k)\Delta e(k) + w_3(k)\delta e(k) \tag{5.1}$$

where $w_1(k)$, $w_2(k)$, and $w_3(k)$ are the ASN controller parameters obtained at the *k*-th sampling instant.

It is obvious that the ASN controller has identical structure with PID controller except that the controller parameters are adjusted on-line. Thus, the proposed ASN controller can be viewed as an adaptive PID controller. In the following section, a learning algorithm integrated with the JITL will be developed to update the parameters of ASN controller at each sampling instant.

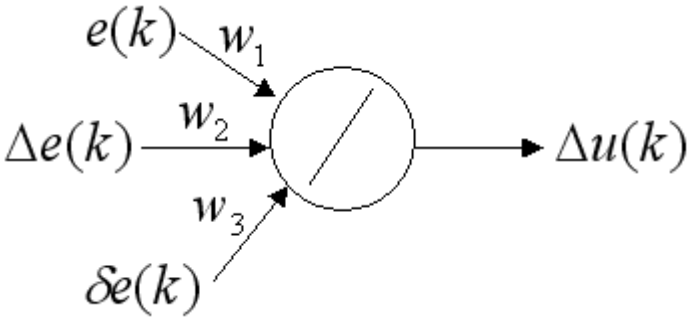


Figure 5.2 ASN controller

5.2.2 Learning algorithm

As mentioned in Chapter 3, the JITL model is able to identify the current process dynamics at each sampling instant, therefore a simple model structure, e.g. low-order ARX models, can be used for each local model at every sampling time.

Therefore, the following second-order ARX model is considered in the ASN controller design.

$$\hat{y}(k+1) = \alpha_1 y(k) + \alpha_2 (y-1) + \beta u(k) \tag{5.2}$$

The control objective is to determine $u(k)$ such that the following quadratic function, consisting of the one-step-ahead tracking error and the change in the control inputs, is minimized

$$Min\ J = (r(k+1) - \hat{y}(k+1))^2 + \kappa(u(k) - u(k-1))^2 \tag{5.3}$$

where $r(k+1)$ is the set-point, $\hat{y}(k+1)$ is one-step-ahead prediction by the JITL model, and κ is a weight parameter. Substituting $\hat{y}(k+1)$ and $u(k)$ by Eqs. (5.2) and (5.1) respectively, it can be seen that Eq. (5.3) is a function of controller parameters, i.e., the control objective is to find the optimal values of $w_i(k)$ to minimize Eq. (5.3).

Since the controller parameters $w_i(k)$ are constrained to be positive or negative, the following mapping function is introduced:

$$w_i(k) = \begin{cases} e^{\varsigma_i(k)}, & \text{if} \quad w_i(k) \geq 0 \\ -e^{\varsigma_i(k)}, & \text{if} \quad w_i(k) < 0 \end{cases}, \quad i = 1 \sim 3 \tag{5.4}$$

where ς_i is a real number. Therefore, the objective is equivalent to finding the optimal ς_i to minimize Eq. (5.3).

To tune the controller parameters at every sampling time, backpropogation method is used to derive the parameter updating equations as follows:

$$\begin{aligned} \varsigma_i(k+1) &= \varsigma_i(k) - \eta_i(k) \frac{\partial J}{\partial \varsigma_i(k)} \\ &= \varsigma_i(k) - \eta_i(k) \frac{\partial J}{\partial w_i(k)} w_i(k) \end{aligned} \tag{5.5}$$

$$\frac{\partial J}{\partial w_i(k)} = \frac{\partial J}{\partial u(k)} \frac{\partial u(k)}{\partial w_i(k)} \tag{5.6}$$

$$\frac{\partial J}{\partial u(k)} = -2(r(k+1) - \hat{y}(k+1))\frac{\partial \hat{y}(k+1)}{\partial u(k)} + 2\kappa(u(k) - u(k-1)) \tag{5.7}$$

for $i = 1 \sim 3$. The derivative $\frac{\partial \hat{y}(k+1)}{\partial u(k)}$ can be obtained from the JITL model, and

$\frac{\partial u(k)}{\partial w_1(k)} = e(k)$, $\frac{\partial u(k)}{\partial w_2(k)} = \Delta e(k)$, $\frac{\partial u(k)}{\partial w_3(k)} = \delta e(k)$.

In Eq. (5.5), the adaptive learning rate $\eta_i(k)$ is determined by the following rules: (i) if the increment of J is more than the threshold, the controller parameters remain unchanged and the learning rate is decreased by a factor l_{dec}, i.e. $\eta_i(k) = l_{dec}\eta_i(k-1)$; (ii) if the increment of J is smaller than the threshold, only the controller parameters are updated; (iii) if the increment of J is negative, the controller parameters are updated and the learning rate is increased by a factor l_{inc}, i.e., $\eta_i(k) = l_{inc}\eta_i(k-1)$. The parameters $l_{dec} = 0.7$ and $l_{inc} = 1.05$ are employed in the simulation studies presented in the next section.

From the on-going discussion, it is evident that the ASN controller maintains simple PID controller structure, which is made possible by the use of only one neuron, and consequently it is much easier to be implemented in practice. On the other hand, the adaptive nature of ASN controller enables the proposed controller to deliver better control performance than the linear controllers. The implementation of the proposed ASN control algorithm is summarized as follows:

1. Given the weight parameter κ, initialize w_i and η_i;
2. Given the current error $e(k)$, compute the manipulated variable $u(k)$ from Eq. (5.1);
3. Update local ARX model by using the most current process data and JITL algorithm and subsequently adjust ς_i by Eq. (5.5);

4. Obtain controller parameters at the next sampling instant using Eq. (5.4). Set $k = k + 1$ and go back to step 2.

To demonstrate the proposed ASN controller, two literature examples are presented in the next section.

5.3 Examples

Example 1 Considering a continuous polymerization reaction that takes place in a jacketed CSTR (Doyle et al., 1995), as depicted in Figure 5.3. In the reactor, an isothermal free-radical polymerization of methyl methacrylate (MMA) is carried out using azo-bis-isobutyronitrile (AIBN) as initiator and toluene as solvent. The control objective is to regulate the product number average molecular weight (NAMW) by manipulating the flow rate of the initiator (F_I), i.e., NAMW is the process output y and F_I is as process input u. Under the following assumptions (Doyle et al., 1995): (i) isothermal operation; (ii) perfect mixing; (iii) constant heat capacity; (iv) no polymer in the inlet stream; (v) no gel effect; (vi) constant reactor volume; (vii) negligible initiator flow rate (in comparison with monomer flow rate); and (viii) quasi-steady state and long-chain hypothesis, the dynamics of the reactor can be described by the following model equations:

$$\frac{\mathrm{d}C_m}{\mathrm{d}t} = -(k_p + k_{fm})C_m P_0 + \frac{F(C_{m_{in}} - C_m)}{V}$$

$$\frac{\mathrm{d}C_\mathrm{I}}{\mathrm{d}t} = -k_\mathrm{I} C_\mathrm{I} + \frac{F_\mathrm{I} C_{\mathrm{I}_{in}} - F C_\mathrm{I}}{V}$$

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

$$\frac{dD_1}{dt} = M_m (k_p + k_{f_m})C_m P_0 - \frac{FD_1}{V}$$

$$y = \frac{D_1}{D_0} \tag{5.8}$$

where $P_0 = \left[\frac{2f * k_\mathrm{I} C_\mathrm{I}}{k_{T_d} + k_{T_c}}\right]^{0.5}$.

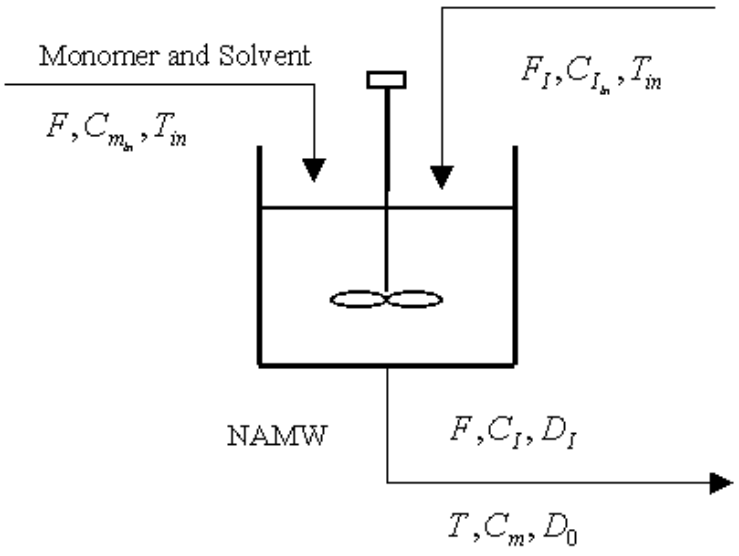


Figure 5.3 Polymerization reactor

The model parameters and steady-state operation condition are given in Tables 5.1 and 5.2.

Table 5.1 Model parameters for polymerization reactor

k_{T_c}	$= 1.3281\times10^{10}$ $m^3/(kmol\ h)$	F	$= 1.00$ m^3/h
k_{T_d}	$= 1.0930\times10^{11}$ $m^3/(kmol\ h)$	V	$= 0.1$ m^3
k_I	$= 1.0225\times10^{-1}$ L/h	$C_{I_{in}}$	$= 8.0$ kmol/m^3
k_p	$= 2.4952\times10^{6}$ $m^3/(kmol\ h)$	M_m	$= 100.12$ kg/kmol
k_{f_m}	$= 2.4522\times10^{3}$ $m^3/(kmol\ h)$	$C_{m_{in}}$	$= 6.0$ kmol/m^3
f^*	$= 0.58$		

Table 5.2 Steady-state operating condition of polymerization reactor

C_m	$= 5.506774$ kmol/m^3	D_1	$= 49.38182$ kmol/m^3
C_I	$= 0.132906$ kmol/m^3	$u = F_I$	$= 0.016783$ m^3/h
D_0	$= 0.0019752$ kmol/m^3	y	$= 25000.5$ kg/kmol

To apply the JITL method for process modeling, input/output data are generated by introducing uniformly random steps with distribution of [0.004, 0.080] and switching probability of 0.25 at every sampling time to the process input F_I. With sampling time of 0.03h, input/output data thus obtained (see Figure 5.4) are used to build the database. The process input and output are scaled by $\tilde{u} = \dfrac{u - 0.016783}{0.01}$ and $\tilde{y} = \dfrac{y - 25000.5}{10000}$ respectively. A second-order ARX model is used as the local model and the parameters chosen for JITL algorithm are as follows: $\gamma = 0.9$, $k_{\min} = 6$, and $k_{\max} = 60$.

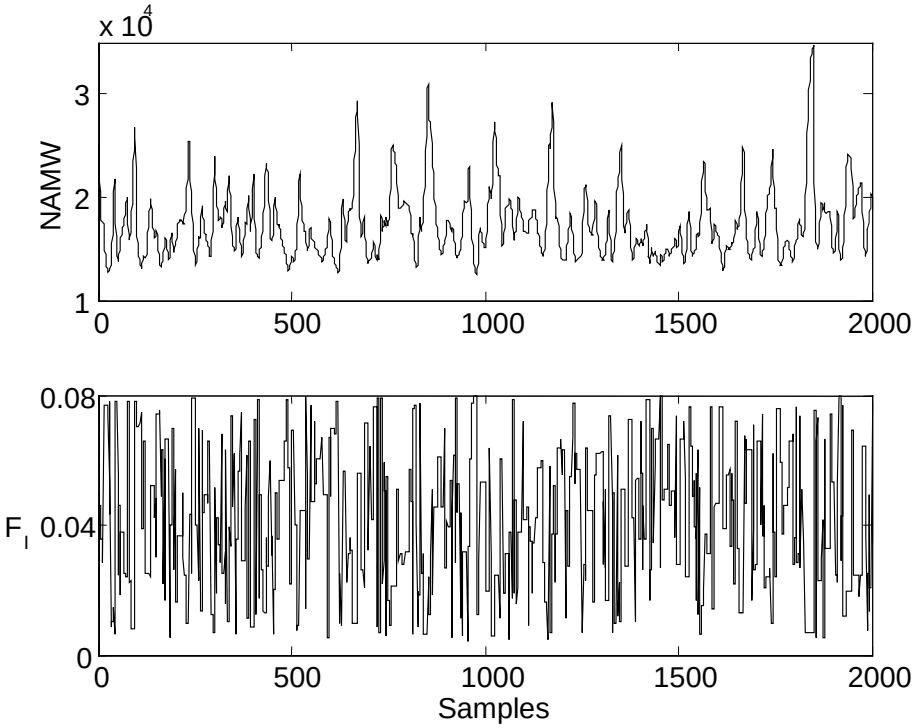


Figure 5.4 Input-output data used for constructing the database for JITL

To proceed with the proposed control strategy, the weight parameter used in the objective function (5.3) is $\kappa = 0.15$ and the initial controller parameters are $w_1 = -0.4$, $w_2 = -1.2$, and $w_3 = -0.5$, with their respective initial learning rates $\eta_1 = 0.9$, $\eta_2 = 0.1$, and $\eta_3 = 0.5$.

For the purpose of comparison, the following IMC controller, which is employed as the benchmark design in the work of Doyle et al. (1995), is designed based on the linear model obtained around the nominal operating condition and a second-order filter with filter time constant equal to 0.2:

$$\frac{-0.6560s^4 - 26.90s^3 - 413.3s^2 - 2822s - 72206}{s^4 + 31.79s^3 + 360.8s^2 + 1723.9s + 2947.8} \tag{5.9}$$

To compare the control performances of the two controllers aforementioned, set-point changes from 25000.5 to 40000 kg/kmol and from 25000.5 to 15000 kg/kmol are conducted, as illustrated in Figure 5.5. For set-point change from 25000.5 to 40000 kg/kmol, IMC controller gives a large overshoot and oscillatory response, while ASN controller arrives at the set-point quickly with smaller overshoot, resulting in 33.2% reduction of the mean absolute error (MAE). On the other hand, the response of IMC controller is much more sluggish than the ASN controller for set-point change from 25000.5 to 15000 kg/kmol. Consequently, ASN controller shows marked improvement over IMC controller, as evidenced by 71.0% reduction of MAE achieved by the ASN controller. Figure 5.6 shows the updating of the ASN controller parameters in the aforementioned closed-loop responses.

Clearly, IMC controller cannot provide satisfactory control performance for this nonlinear process. Note that the IMC controller yields an oscillatory response at the set-point of 40000 kg/kmol, while a sluggish response is observed at the set-point of 15000 kg/kmol. As a result, if one adopts a more aggressive IMC design (as compared with the present IMC design) to avoid sluggish response for set-point change to 15000 kg/kmol, this will inevitably make the servo response for the set-point change to 40000 kg/kmol highly oscillatory. Likewise, a more conservative IMC design can eliminate the oscillatory response in the latter case, but at the expense of even more sluggish response in the former case. Table 5.3 summarizes the tracking errors of these two controllers for various set-point changes. It is evident that the proposed controller gives a better performance over the operating space compared with the IMC controller.

Table 5.3 MAEs of two controllers for various set-point changes

Set-point	IMC	ASN	Improvement
40000	2.80×10^{3}	1.87×10^{3}	33.2%
35000	1.11×10^{3}	6.30×10^{2}	43.2%
30000	5.98×10^{2}	3.12×10^{2}	47.8%
20000	9.96×10^{2}	3.41×10^{2}	65.8%
15000	2.75×10^{3}	7.97×10^{2}	71.0%

To illustrate the disturbance rejection capability of the proposed controller, it is assumed that the inlet initiator concentration $C_{I_{in}}$ is subject to $\pm10\%$ step disturbance respectively. As can be seen from Figure 5.7, ASN controller outperforms IMC controller by reducing the MAE by 62.3% and 61.1%, respectively. Next, to evaluate the robustness of the proposed control strategy, it is assumed that the process kinetic parameter k_1 is subject to -10% modeling error and the resulting servo responses of two controllers are compared in Figure 5.8. It is evident that the proposed controller still maintains superior control performance by achieving 27.0% reduction of MAE for set-point change to 40000 kg/kmol and 66.2% reduction for set-point change to 15000 kg/kmol. Furthermore, to study the effect of process noise on the proposed design, both process input and output are corrupted by 1% Gaussian white noise, which means that the database used for JITL algorithm also contains the corrupted process data. As shown in Figure 5.9, ASN controller can yield reasonably good control performance in the presence of process noise. Lastly, to illustrate the advantage of the JITL compared to the recursive least square (RLS) identification procedure, the proposed ASN design is compared with the ASN design based on a second-order ARX model with parameter adaptation by the RLS modeling technique.

As can be seen from Figure 5.10, the proposed ASN design with the JITL models gives better control performance not only because it takes less time to reach the set-point of 40000 kg/kmol with smaller oscillation, but also it has faster response for the set-point change to 15000 kg/kmol than the ASN design with RLS models, as supported by the reduction of MAE by 5.5% and 23.4%, respectively.

Example 2 Consider the van de Vusse reaction as discussed in Chapter 3, where the control objective is to regulate the concentration of component B (C_B) by manipulating the inlet flow rate F. The nominal operation conditions are given by: $C_{A0} = 3.0$, $C_{B0} = 1.1172$, and $F_0 = 34.3$.

To apply the proposed controller strategy, a second-order ARX model is employed as the local model for JITL algorithm. A database is generated by introducing uniformly random steps with distribution of $[4, 65]$ and switching probability of 0.1 at every sampling instant to the process input F. The parameters used for JITL algorithm are: $\gamma = 0.95$, $k_{\min} = 6$, and $k_{\max} = 60$. In addition, the weight parameter in the objective function (5.3) is chosen as $\kappa = 0.55$ and the initial controller parameters are $w_1 = 0.35$, $w_2 = 2$, and $w_3 = 0.7$, with their initial learning rates specified by $\eta_1 = 0.8$, $\eta_2 = 0.1$, and $\eta_3 = 0.1$, respectively.

For the purpose of comparison, the benchmark IMC controller designed based on the linear model around the nominal operating condition as given in the work of Doyle et al. (1995) is employed. With a first-order filter and filter time constant equal to 0.01, the IMC controller is given by:

$$\frac{89.29(s+134.3)(s+144.3)}{(s+168.2)(s+100)} \tag{5.10}$$

To evaluate the control performance of the two controllers, $+10\%$ and -50% step changes in the set-point of C_B are considered. As can be seen from Figure 5.11, the proposed controller settles down at the new set-point faster than the IMC controller for $+10\%$ set-point change, resulting in 26.4% reduction of MAE. For -50% set-point change, the proposed controller also displays better control performance than the IMC controller, leading to 8.8% reduction of MAE. Figure 5.12 shows the updating of the ASN controller parameters in the aforementioned closed-loop responses.

To evaluate the disturbance rejection capability of the ASN controller, $\pm 10\%$ step disturbances are assumed to occur in the inlet concentration of component A (C_{Af}) respectively. It is evident from Figure 5.13 that ASN controller outperforms IMC controller and consequently MAEs are reduced by 26.7% and 8.1%, respectively. Lastly, the robustness of the proposed controller is also evaluated by assuming -10% modeling error in the kinetic parameter k_3. It can be seen from Figure 5.14 that the proposed controller achieves better performance than the IMC controller, as also evident by the respective 11.9% and 8.2% reduction of MAE for 10% and -50% set-point changes.

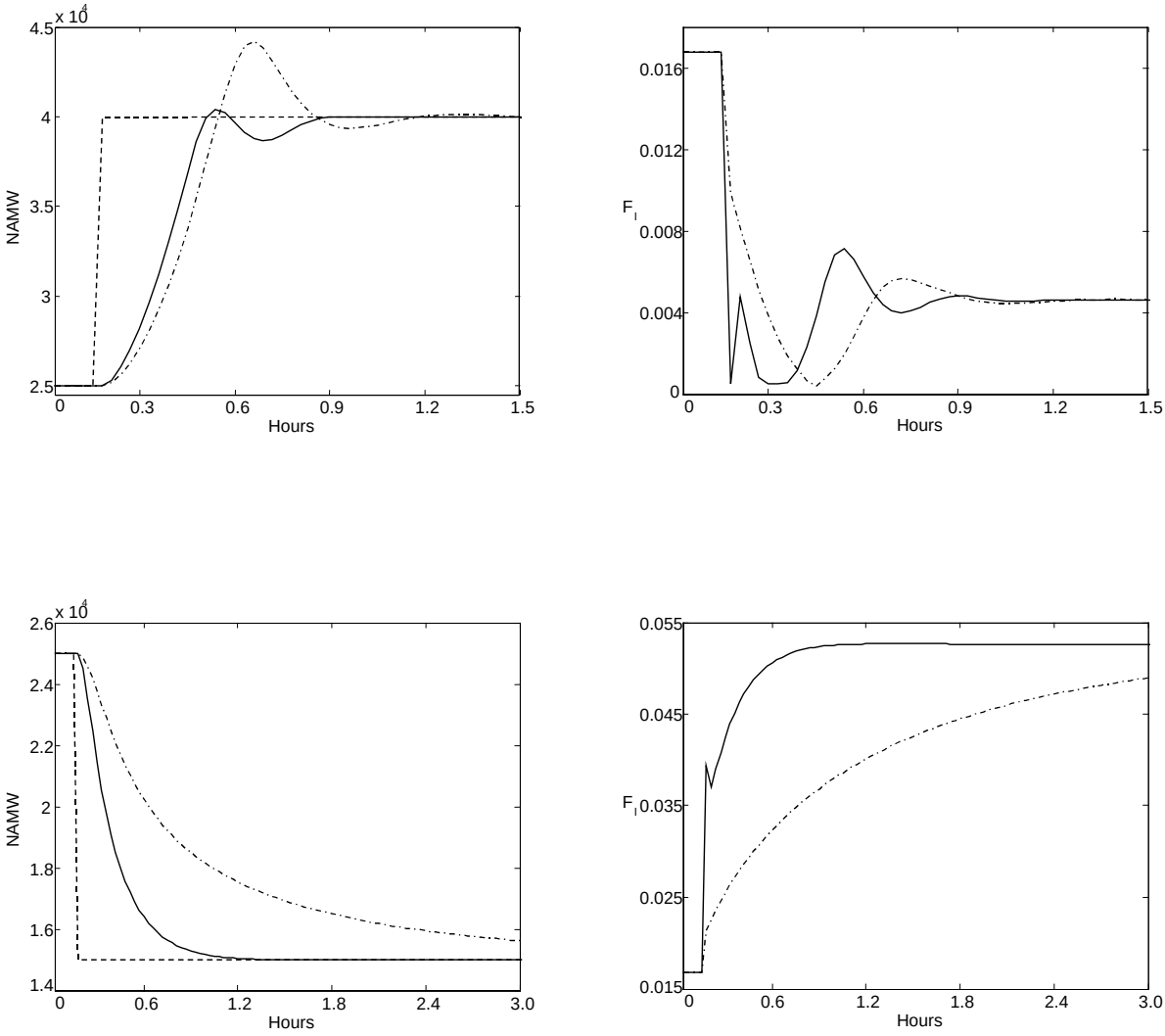


Figure 5.5 Closed-loop responses for set-point changes to 40000 kg/kmol (top) and 15000 kg/kmol (bottom). Dashed: set-point; solid: ASN; dashed-dot: IMC

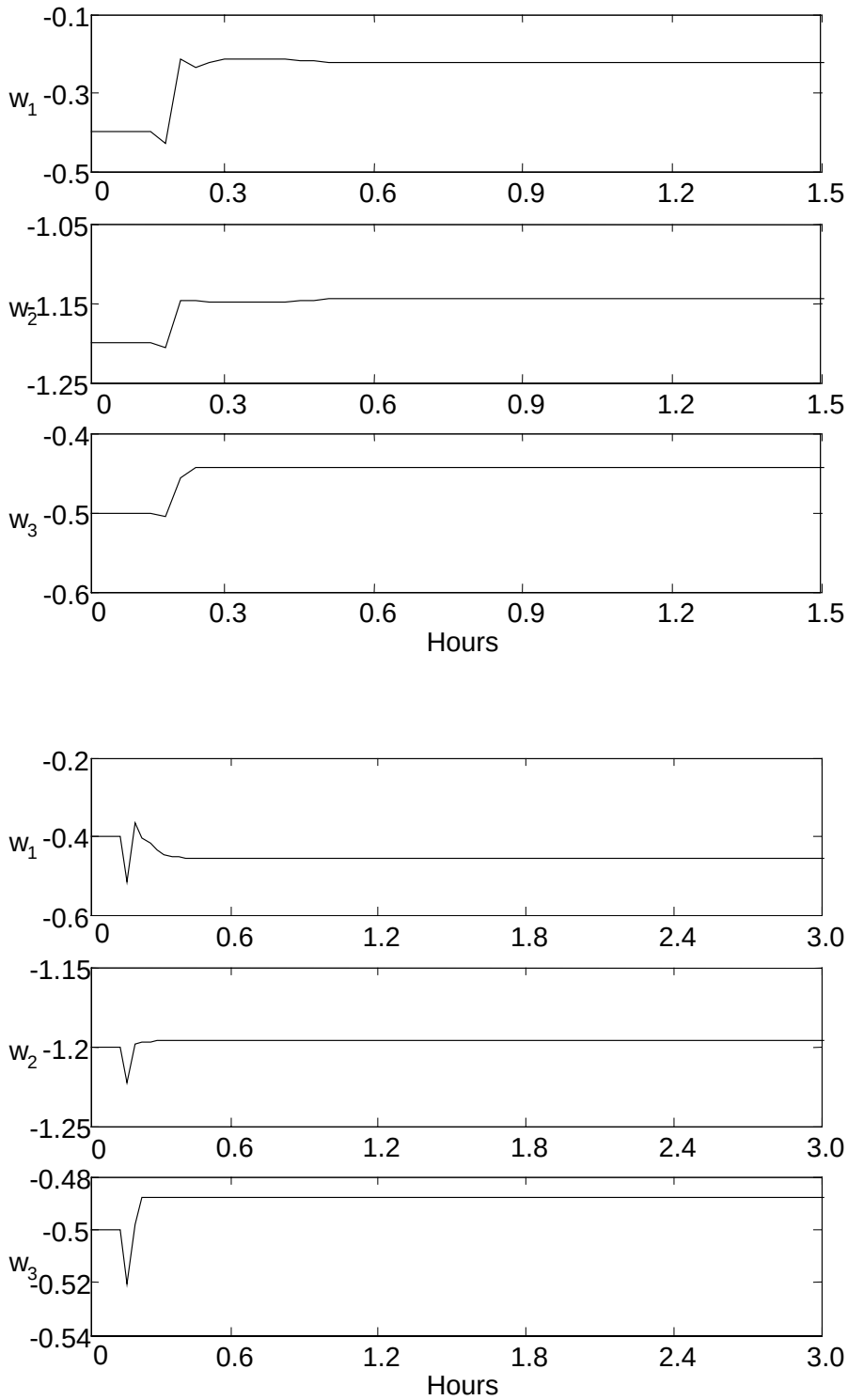


Figure 5.6 Updating of ASN parameters for set-point changes to 40000 kg/kmol (top) and 15000 kg/kmol (bottom)

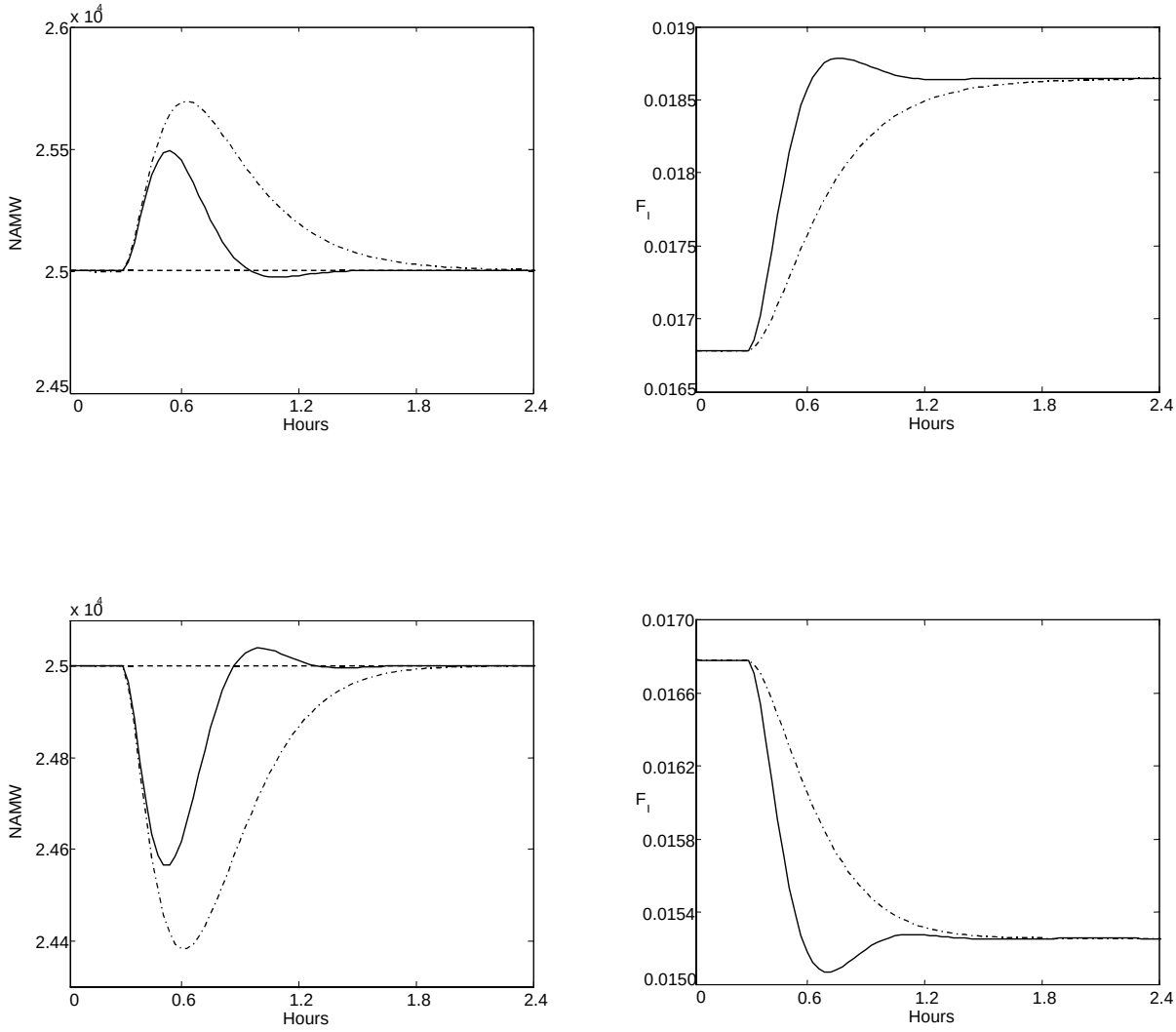


Figure 5.7 Closed-loop responses for -10% (top) and 10% (bottom) step disturbances in $C_{I_{in}}$. Dashed: set-point; solid: ASN; dashed-dot: IMC

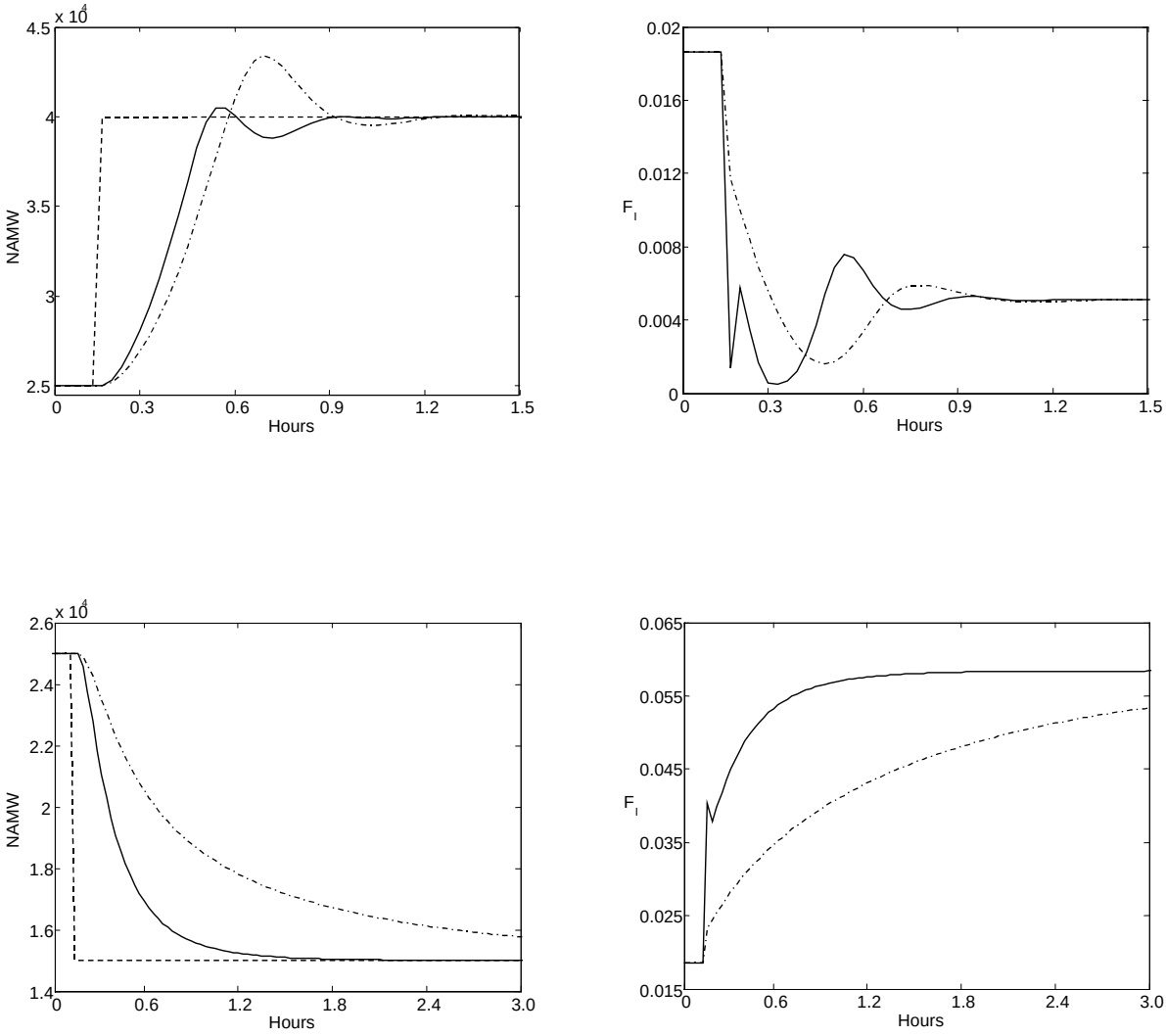


Figure 5.8 Closed-loop responses for set-point changes to 40000 kg/mol (top) and 15000 kg/kmol (bottom) under -10% modeling error in k_I. Dashed: set-point; solid: ASN; dashed-dot: IMC

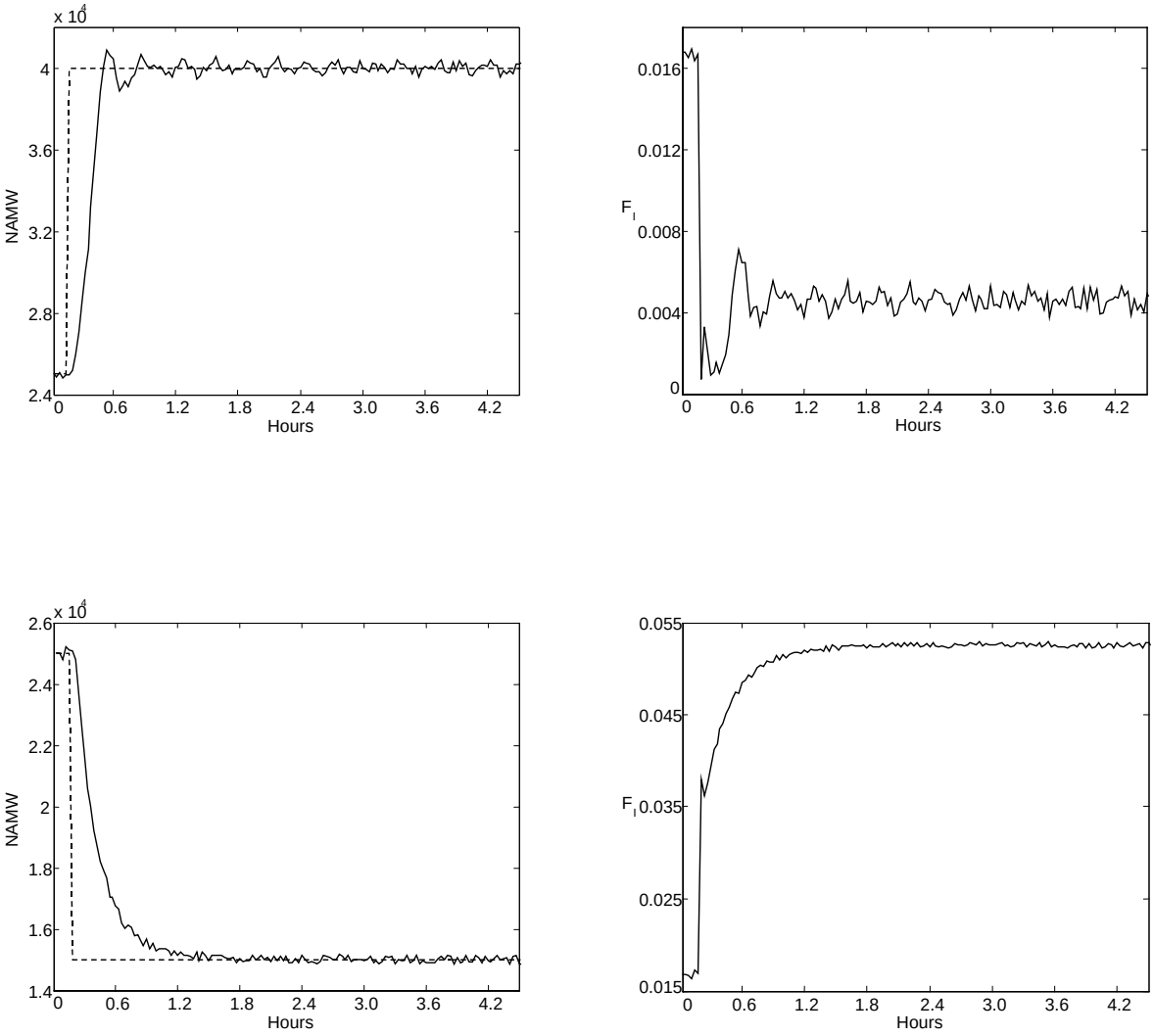


Figure 5.9 Servo responses in the presence of process noise

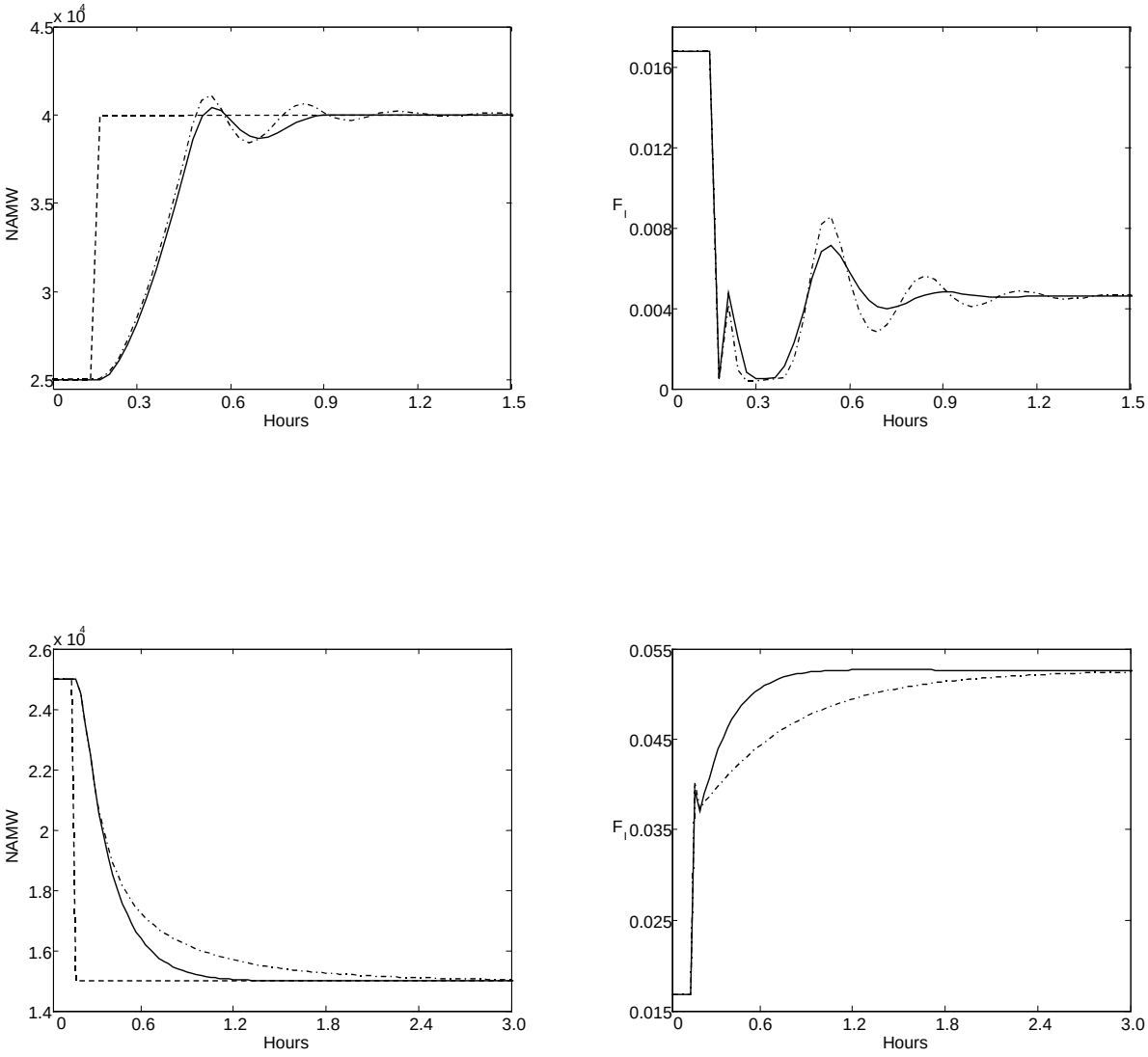


Figure 5.10 Servo response for the ASN design based on JITL and recursive least square (RLS) models. Dashed: set-point; solid: JITL; dashed-dot: RLS

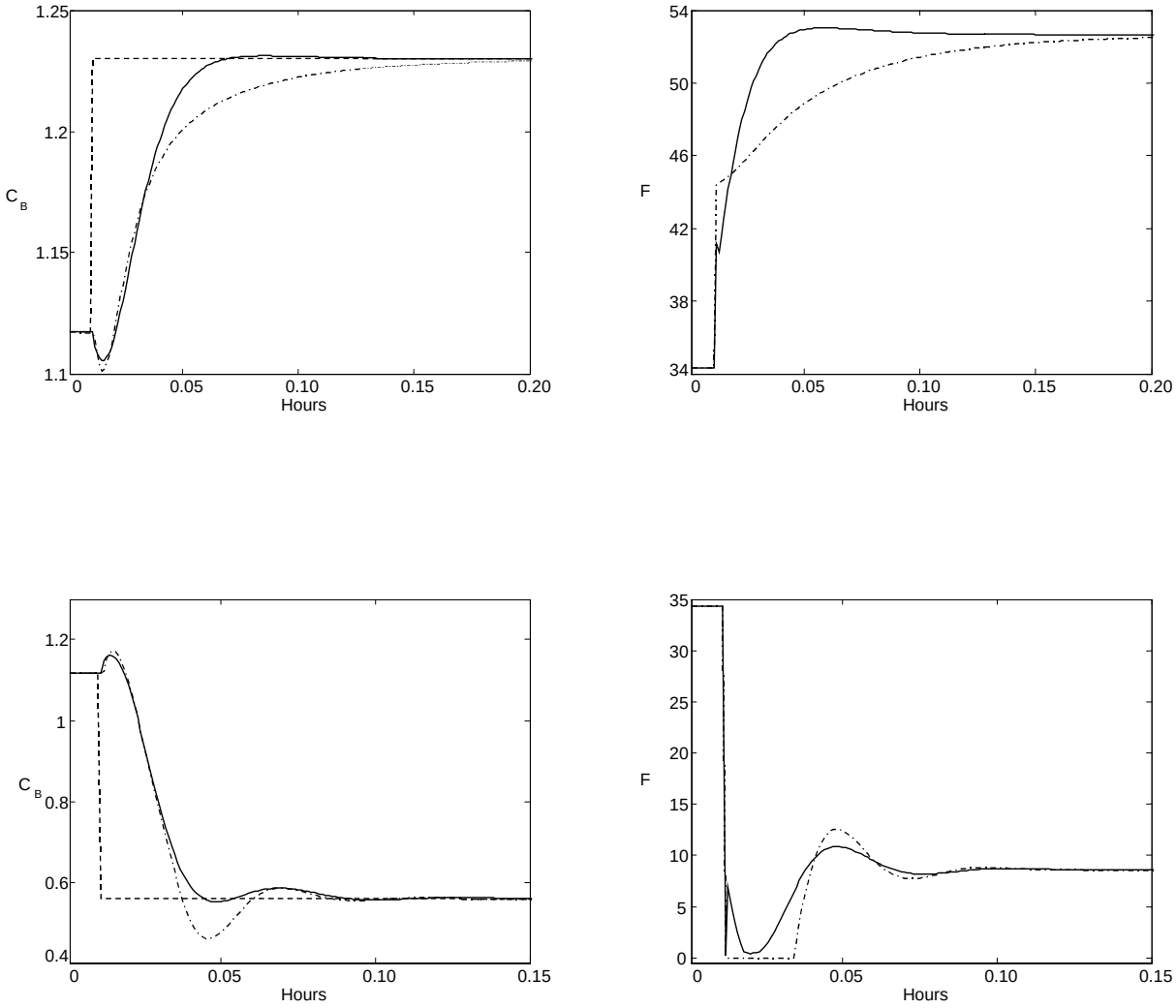


Figure 5.11 Closed-loop responses for $+10\%$ (top) and -50% (bottom) set-point change. Dashed: set-point; solid: ASN; dashed-dot: IMC

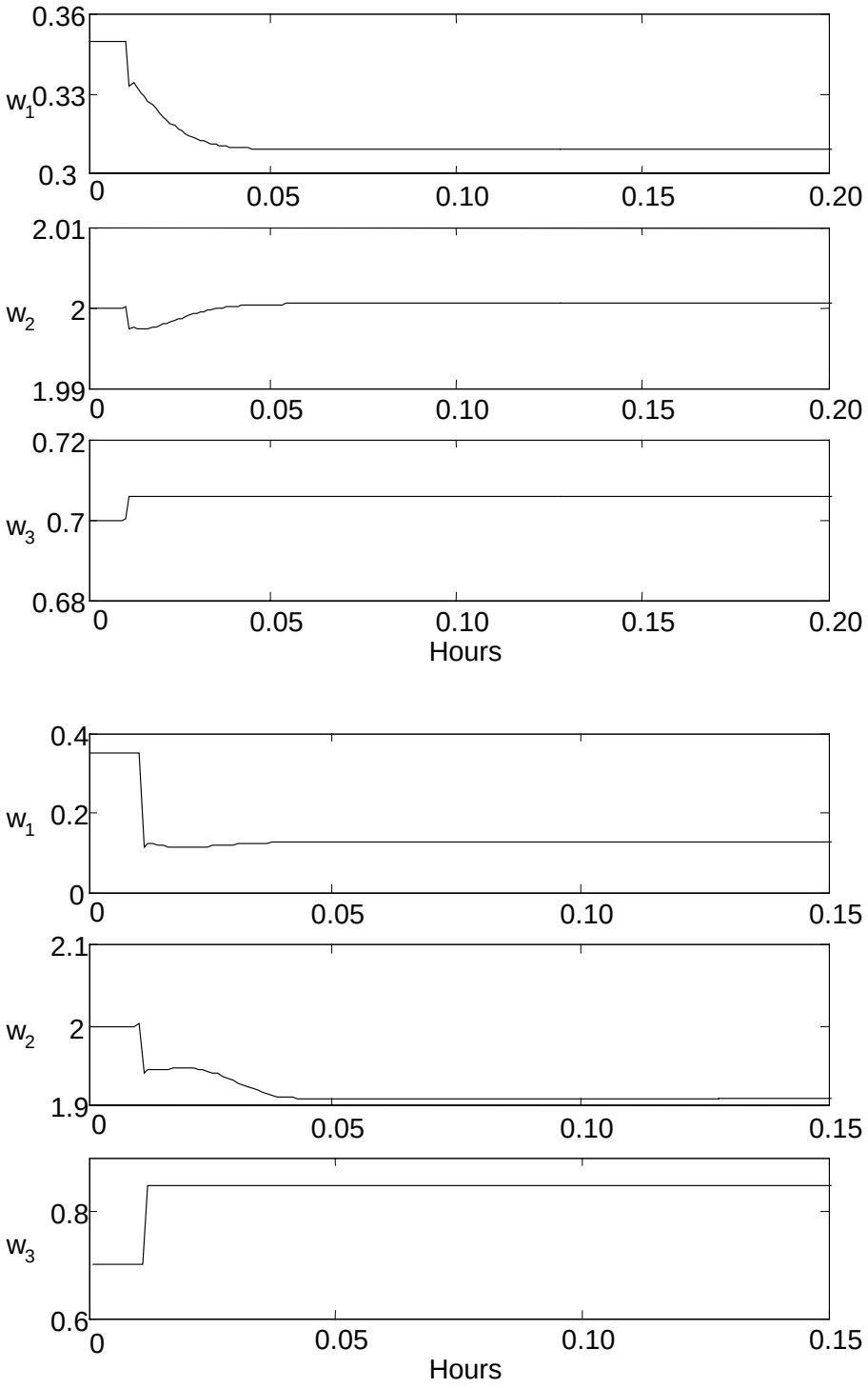


Figure 5.12 Updating of ASN parameters for $+10\%$ (top) and -50% (bottom) set-point changes

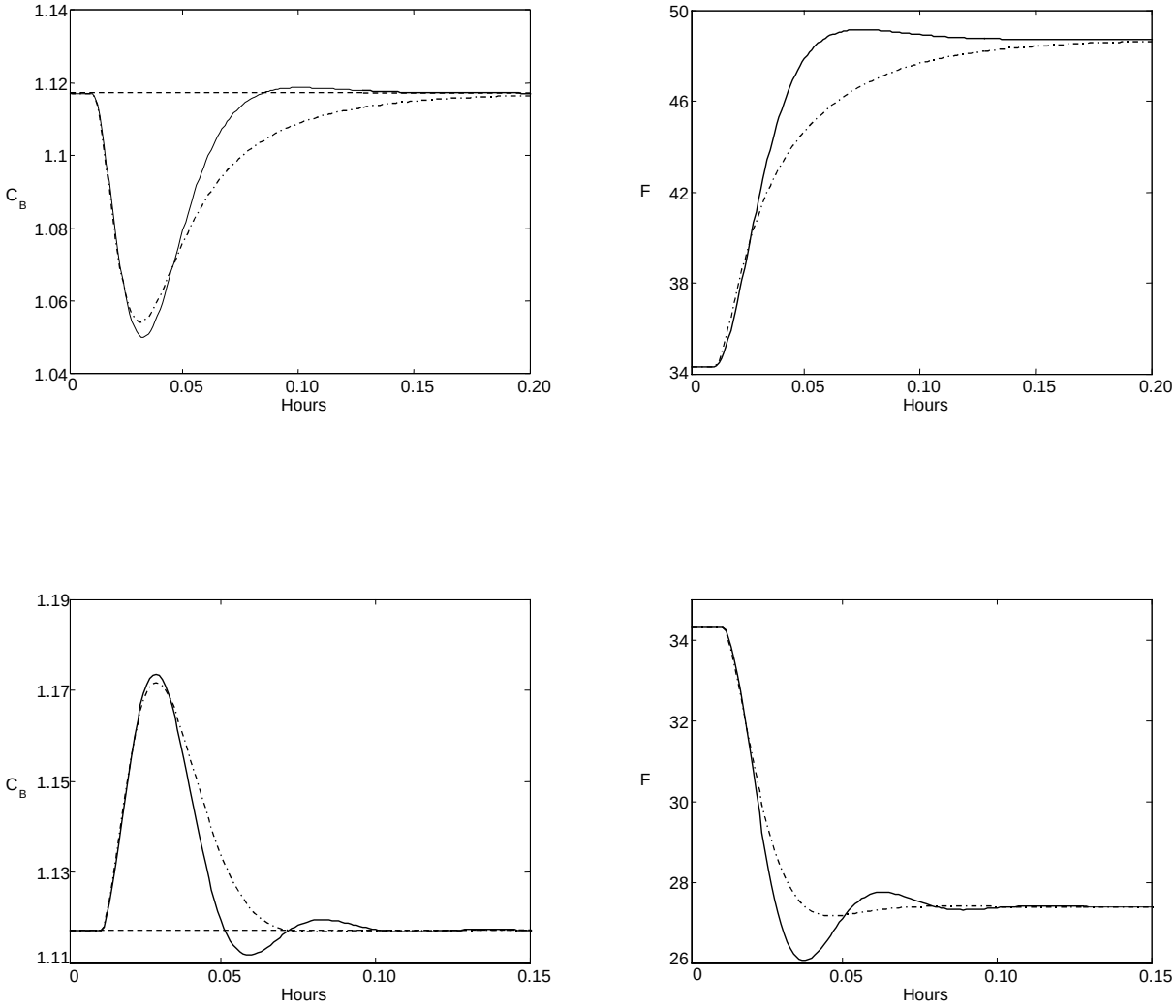


Figure 5.13 Closed-loop responses for -10% (top) and 10% (bottom) step disturbances in C_{Af} . Dashed: set-point; solid: ASN; dashed-dot: IMC

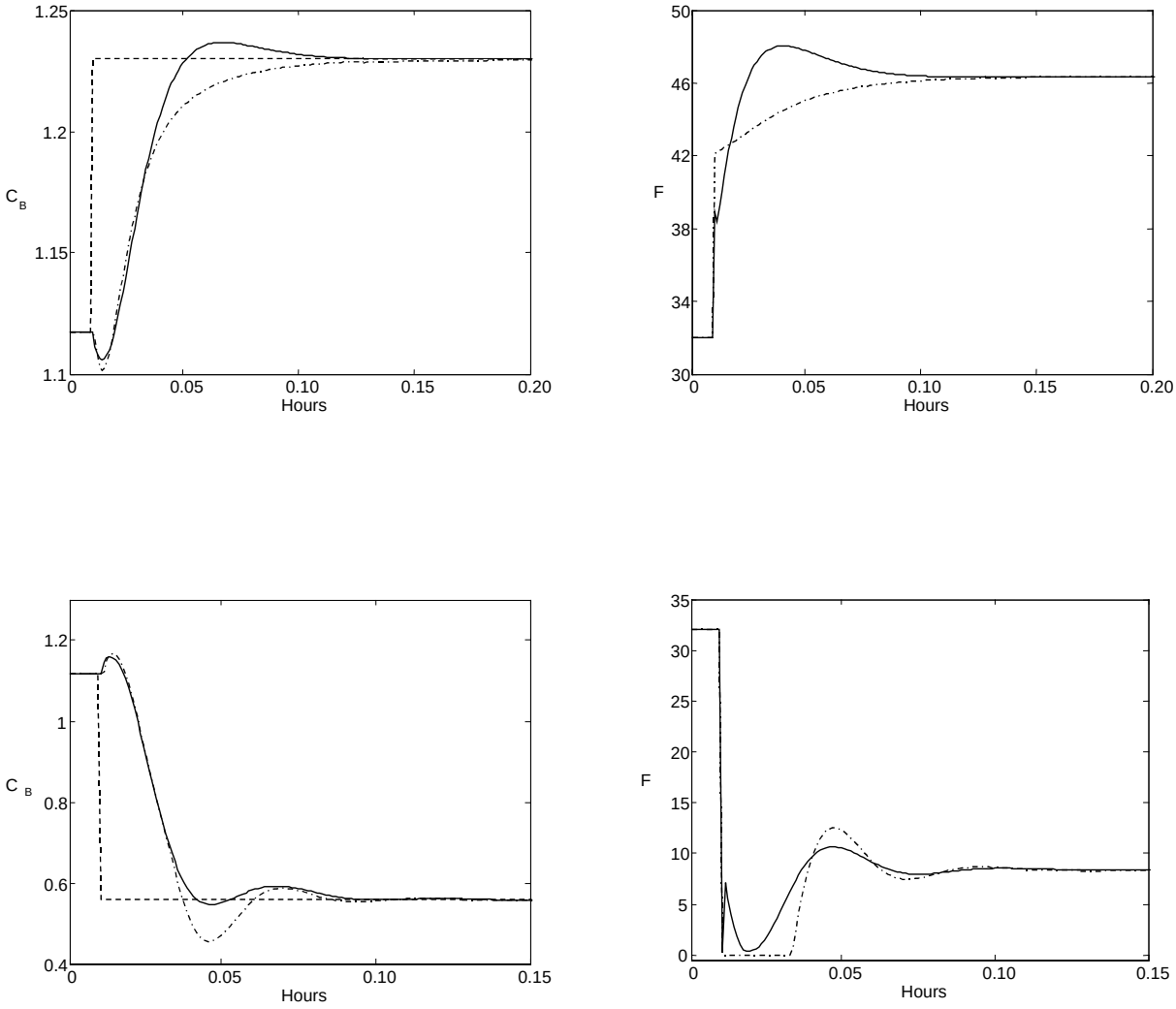


Figure 5.14 Closed-loop responses of 10% (top) and -50% (bottom) set-point changes under -10% modeling error in k_3. Dashed: set-point; solid: ASN; dashed-dot: IMC

5.4 Conclusion

A new adaptive controller, ASN controller, is proposed in this chapter. To mimic the traditional PID controller, a single neuron is employed in the proposed controller design strategy. Incorporated with the neural network's learning ability, the proposed controller can control the unknown nonlinear dynamic process adaptively through the updating of its parameters by the adaptive learning algorithm developed and the information provided from the JITL. Compared with the previous neural network based PID controller designs, ASN controller is more amenable for on-line implementation. Furthermore, the proposed controller retains the PID structure and therefore it is easy for field operators to understand the ASN controller structure, which is in sharp contrast to the neural network based controllers previously developed that represent as a black-box to the operators. Simulation results illustrate that the proposed controller gives better control performance than its conventional counterparts.

Chapter 6

Adaptive IMC Controller Design

6.1 Introduction

From the review of nonlinear IMC design approaches in Chapter 2, it is noted that most nonlinear IMC methods attempt different techniques to transform the original nonlinear control problem to an equivalent linear IMC design problem so that controller design can be easily carried out by the linear IMC design procedure. For instance, Calvet and Arkun (1988) used state-space linearization approach for nonlinear systems in the presence of disturbances. A disadvantage of this method is that an artificial controlled output is introduced in the controller design procedure and therefore it is difficult to be specified a priori (Henson and Seborg, 1991a). Doyle et al. (1995) proposed a partitioned model inverse controller synthesis strategy based on Volterra model that retains the original spirit and characteristics of conventional IMC while extending its capabilities to nonlinear systems. Although Doyle's method can capture the local nonlinearities around an operation point accurately, it may be

erroneous in describing global nonlinear behavior (Maner et al., 1996). Another drawback of this method is that parameters of second-order Volterra model is not parsimonious to describe the process nonlinearities. Harris and Palazoglu (1998) employed functional expansion (Fex) models instead of Volterra model. However, functional expansion model 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. Another popular approach is to integrate neural network into the IMC framework (Bhat and McAvoy, 1990; Hunt and Sbarbaro, 1991; Nahas et al., 1992; Li et al., 2000). In this approach, a neural network is trained to learn the inverse dynamic of the nonlinear process meanwhile another NN is used to design a controller. Although successful in some cases, this approach may lead to offset because the product of the gains of the NN model and the NN controller does not necessarily yield unity. Furthermore, nonlinear optimization is required to update a large number of weights which is not only computationally demanding but also prone to the problem of poor convergence.

To alleviate the aforementioned problems, we propose to incorporate the JITL into the IMC framework to develop an adaptive IMC design methodology that takes advantage of the low-order model employed in JITL, by which the model inverse can be readily obtained for IMC design at each sampling instant. The proposed design strategy shares the same idea with the ASN controller presented in Chapter 5, i.e. it exploits the information provided by the JITL and adjusts the controller parameters on-line by an adaptive learning method. However, three controller parameters and three learning rates need to be initialized in the ASN design, which uses trial and error procedure. In contrast, in the proposed IMC design scheme, the JITL is employed as

the process model and the IMC controller is designed based on the JITL model augmented with a filter. In this manner, the number of controller parameters can be reduced to two, i.e. the IMC filter parameter and its associated learning rate used in the adaptive learning algorithm to be developed in this chapter. Consequently, the proposed IMC design strategy is an attractive alternative because it lessens the efforts of tuning controller parameters compared to the ASN controller. Last, it is worth pointing out that the proposed IMC design can be considered as an adaptive IMC controller because the model obtained by the JITL algorithm is updated at each sampling instant.

6.2 JITL Based Adaptive IMC Design

6.2.1 Linear IMC framework

The block diagram of the IMC structure is shown in Figure 6.1, where G and $\tilde{G}$ denote the open-loop stable process and process model, respectively.

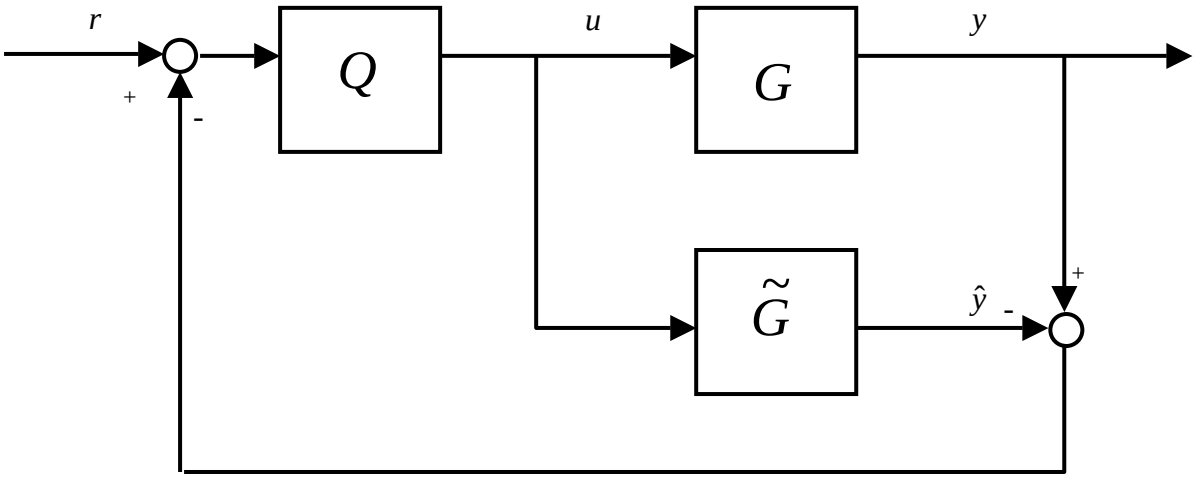


Figure 6.1 Block diagram of IMC structure

The IMC controller, Q, can be designed by the following equation (Morari and Zafiriou, 1989):

$$Q = \tilde{G}_{-}^{-1} f \tag{6.1}$$

where $\tilde{G}_{-}$ is the minimum phase part of $\tilde{G}$ and f is a low-pass filter, which is designed to make the IMC controller Q realizable and to meet the design trade-off between the performance and robustness requirements.

The IMC framework allows the use of a variety of process models, such as first-principle models as well as neural network models. However, the difficult in the use of these models in the IMC framework arises in the design of the controller. Because the IMC controller is based on the inverse of the minimum phase part of the model $\tilde{G}$, a reliable and efficient method is required to achieve this inversion (Maksumov et al. 2002). To this end, the JITL model is embedded in the IMC framework so that the model inverse can be obtained readily, as discussed in the next section.

6.2.2 Proposed adaptive IMC controller design

The proposed adaptive IMC shares the similar design concept with the ASN design, i.e. JITL is used to obtain the local model at each sampling instant and an adaptive learning method is employed to tune the controller parameters on-line. The proposed adaptive IMC scheme is depicted in Figure 6.2, where the process model $\tilde{G}$ is updated by the JITL algorithm on-line. According to the IMC design, controller Q is designed based on the inversion of the minimum phase of process model $\tilde{G}$ augmented with a low-pass filter. In the proposed method, filter parameter of Q is not fixed, instead it is adjusted on-line by the gradient descent algorithm to be developed

in the sequel. As such, the JITL is employed not only to update the model parameters but also to adjust the IMC controller as well.

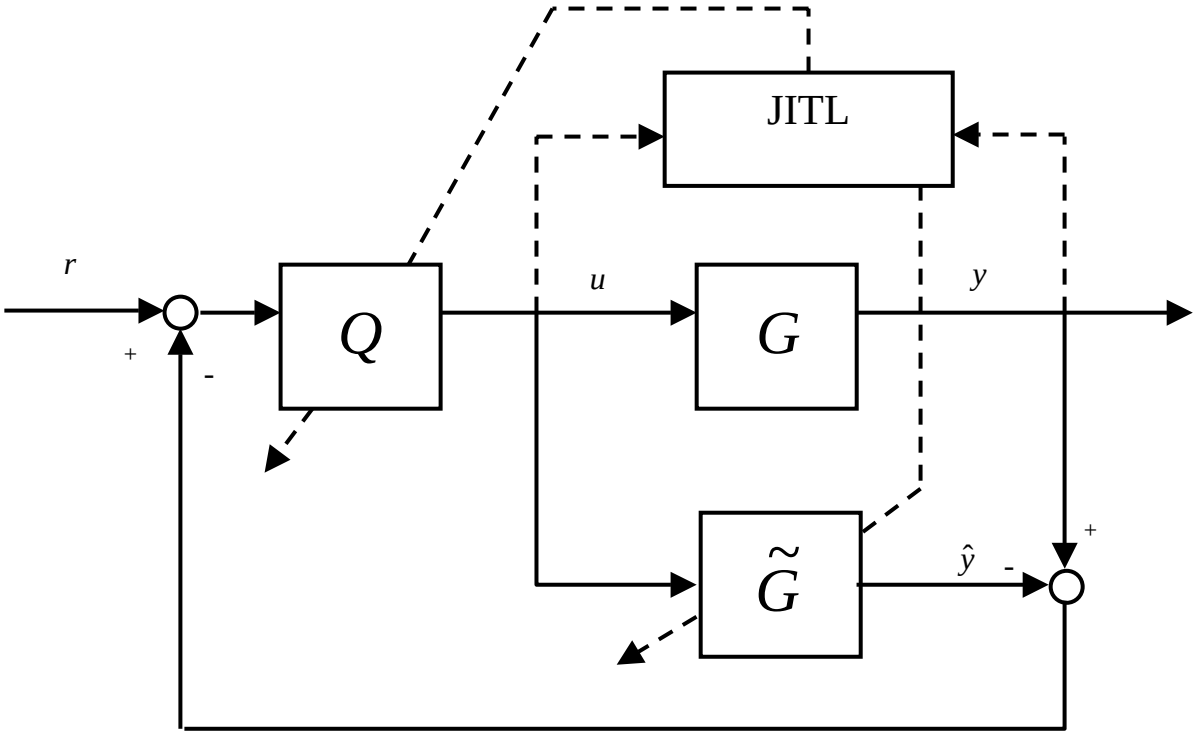


Figure 6.2 JITL based adaptive IMC scheme

Recall that a first-order or second-order ARX model is employed in the JITL algorithm, i.e.

$$y(k) = \alpha_1^k y(k-1) + \alpha_2^k y(k-2) + \beta^k u(k-1) \tag{6.2}$$

where the model parameters α_1^k, α_2^k, and β^k are identified by the JITL at the *k*-th sampling instant. The transfer function model corresponding to Eq. (6.2) is given by:

$$G^k(z^{-1}) = \frac{\beta^k z^{-1}}{1 - \alpha_1^k z^{-1} - \alpha_2^k z^{-2}} \tag{6.3}$$

Using a first-order filter, IMC controller is designed as following:

$$Q^k(z^{-1}) = \frac{1-\alpha_1^k z^{-1} - \alpha_2^k z^{-2}}{\beta^k} \frac{1-\lambda(k)}{1-\lambda(k)z^{-1}} \tag{6.4}$$

where $\lambda(k)$ is the IMC filter parameter obtained at the *k*-th sampling instant.

The controller law resulting from Eq. (6.4) is then given by

$$u(k) = \lambda(k)u(k-1) + \frac{1-\lambda(k)}{\beta^k}(\nu(k) - \alpha_1^k \nu(k-1) - \alpha_2^k \nu(k-2)) \tag{6.5}$$

where $\nu(k) \stackrel{\Delta}{=} r(k) + \hat{y}(k) - y(k)$.

To update the filter parameter on-line, the following objective function is considered:

$$Min\ J = (r(k+1) - \hat{y}(k+1))^2 + \kappa(u(k) - u(k-1))^2 \tag{6.6}$$

where $r(k+1)$ is the set-point, $\hat{y}(k+1)$ is one-step-ahead prediction from the JITL, and κ is the weight parameter.

Because $\lambda(k)$ is constrained between 0 and 1, the following sigmoid function is employed to map the space $[0\ \ 1]$ to the entire real number space:

$$\lambda(k) = \frac{1}{1+e^{-\varsigma(k)}} \tag{6.7}$$

where $\varsigma(k)$ is a real number. In the sequel, an adaptive learning algorithm will be developed to update $\varsigma(k)$ on-line, and the filter parameter $\lambda(k)$ can be easily calculated by Eq. (6.7). Similar to what has been described in Chapter 5, the following summarizes the updating algorithm for $\varsigma(k)$:

$$\begin{aligned} \varsigma(k+1) &= \varsigma(k) - \eta(k)\frac{\partial J}{\partial \varsigma(k)} \\ &= \varsigma(k) - \eta(k)\frac{\partial J}{\partial \lambda(k)}\frac{\partial \lambda(k)}{\partial \varsigma(k)} \end{aligned} \tag{6.8}$$

where $\eta(k)$ is the adaptive learning rate which is determined based on the identical rules discussed in Chapter 5, and

$$\frac{\partial J}{\partial \lambda(k)} = \frac{\partial J}{\partial u(k)} \frac{\partial u(k)}{\partial \lambda(k)} \tag{6.9}$$

$$\frac{\partial J}{\partial u(k)} = -2(r(k+1) - \hat{y}(k+1)) \frac{\partial \hat{y}(k+1)}{\partial u(k)} + 2\kappa(u(k) - u(k-1)) \tag{6.10}$$

$$\frac{\partial u(k)}{\partial \lambda(k)} = u(k-1) - \frac{\nu(k) - \alpha_1^k \nu(k-1) - \alpha_1^k \nu(k-2)}{\beta^k} \tag{6.11}$$

$$\frac{\partial \lambda(k)}{\partial \varsigma(k)} = \lambda(k)(1 - \lambda(k)) \tag{6.12}$$

where $\frac{\partial \hat{y}(k+1)}{\partial u(k)}$ can be obtained from the most updated ARX model obtained by the JITL. Based on the on-going discussion, it can be seen that the proposed adaptive IMC controller exploits more information from the JITL model and fewer tuning parameters are needed compared with the ASN controller.

The implementation of the proposed adaptive IMC controller is summarized as follows:

1. Given the weight parameter κ, initialize the filter parameter and learning rate η;
2. Given $r(k)$, $y(k)$, and $\hat{y}(k)$ at the *k*-th sampling instant, compute $u(k)$ according to Eq. (6.5);
3. Update the linear model by applying the JITL algorithm to the most current process data and subsequently adjust $\varsigma(k)$ according to Eq. (6.8);
4. Calculate IMC filter parameter for the next sampling instant by Eq. (6.7). Set $k = k+1$ and go to step 2.

6.3 Examples

Example 1 The first example considered is the control of a polymerization reaction in a jacketed CSTR discussed earlier in Chapter 5. The model parameters and steady-state operation condition can be found in Tables 5.1 and 5.2. The proposed adaptive IMC controller design is based on the same database and parameters used for the JITL algorithm mentioned in Chapter 5. In addition, the IMC controller provided in Chapter 5 serves the benchmark design for comparison purpose.

To evaluate the servo performances of two controllers, set-point changes from 25000.5 to 40000 kg/kmol and 15000 kg/kmol are considered, as illustrated in Figure 6.3. The initial parameters used for the proposed adaptive IMC design are $\lambda = 0.731$, $\eta = 0.3$, and $\kappa = 0.1$. It is obvious that adaptive IMC design has better performance than that achieved by the IMC controller. For set-point change to 40000 kg/kmol, IMC controller gives a large overshoot and oscillatory response, while the proposed controller arrives at set-point more quickly without overshoot. As a result, the proposed controller reduces MAE by 22.2% compared with the IMC controller. For set-point change to 15000 kg/kmol, the proposed controller reaches set-point much faster than the IMC controller, resulting in significant reduction of MAE, relative to the IMC controller, by approximately 73.5%. Table 6.1 summarizes the tracking errors of these two controllers for various set-point changes. Figure 6.4 shows the updating of the filter parameter in the aforementioned closed-loop responses.

Figure 6.5 compares the disturbance rejection capabilities of two controllers when $\pm 10\%$ step disturbances in $C_{I_{in}}$ are introduced into the process. It is apparent that the proposed controller has superior control performance over the IMC controller by reducing MAE by 65.9% and 64.7%, respectively. To further evaluate the robustness of the proposed controller, it is assumed that there exists -10% modeling

error in the kinetic parameter k_1. As can be seen from Figure 6.6, the proposed controller outperforms the IMC controller, as also evidenced by the reduction of MAE by 25.6% for set-point change to 40000 kg/kmol and 70.2% for set-point change to 15000 kg/kmol. Last, to evaluate the effect of process noise on the proposed design, both process input and output are corrupted by 1% Gaussian white noise. As shown in Figure 6.7, the proposed IMC design can yield reasonably good control performance in the presence of process noise.

Table 6.1 MAEs of two controllers for various set-point changes

Set-point	IMC	Adaptive IMC	Improvement
40000	2.80×10^3	2.18×10^3	22.2%
35000	1.11×10^3	8.90×10^2	19.8%
30000	5.98×10^2	4.25×10^2	28.9%
20000	9.96×10^2	1.97×10^2	80.2%
15000	2.75×10^3	7.29×10^2	73.5%

Example 2 Consider again the control of van de Vusse reactor as studied in Chapter 5. With the initial parameters $\lambda = 0.957$, $\eta = 0.2$, and $\kappa = 0.5$ chosen for adaptive IMC controller, Figure 6.8 shows the servo responses of adaptive IMC and IMC controllers for 10% and -50% set-point changes, respectively. For the former, the setting time of the proposed controller is approximately 59% of that obtained by the IMC controller, resulting in 30.9% reduction of MAE. For -50% set-point change, IMC controller displays oscillatory response, while adaptive IMC controller gives smooth response and reaches the set-point faster than the IMC controller, leading to 14.8% reduction of MAE.

To evaluate the disturbance rejection performance, $\pm 10\%$ step disturbances are assumed to occur in the inlet concentration of component A. The resulting performances of two controllers are compared in Figure 6.9. Again, the performance of the proposed controller shows marked improvement over that obtained by the IMC controller and consequently the resulting MAEs are reduced by 48.8% and 16.2%, respectively. The robustness of the proposed controller is also evaluated by assuming -10% modeling error in the kinetic parameter k_3. Figure 6.10 shows the performance of the two controllers for 10% and -50% set-point changes. It is clear that the proposed controller still achieves better control performance by reducing MAEs by 23.4% and 13.2%, respectively.

6.4 Conclusion

By incorporating the JITL into IMC framework, an adaptive IMC design methodology is developed for nonlinear process control in this chapter. The IMC controller parameters are updated not only based on the information provided by the JITL, but also its filter parameter is adjusted online by an adaptive learning algorithm. Compared with the conventional nonlinear IMC controller design method, it is straightforward for the proposed method to obtain the model inversion based on the JITL modeling technique. Simulation results are presented to demonstrate the advantage of the proposed adaptive IMC controller design over its conventional counterpart.

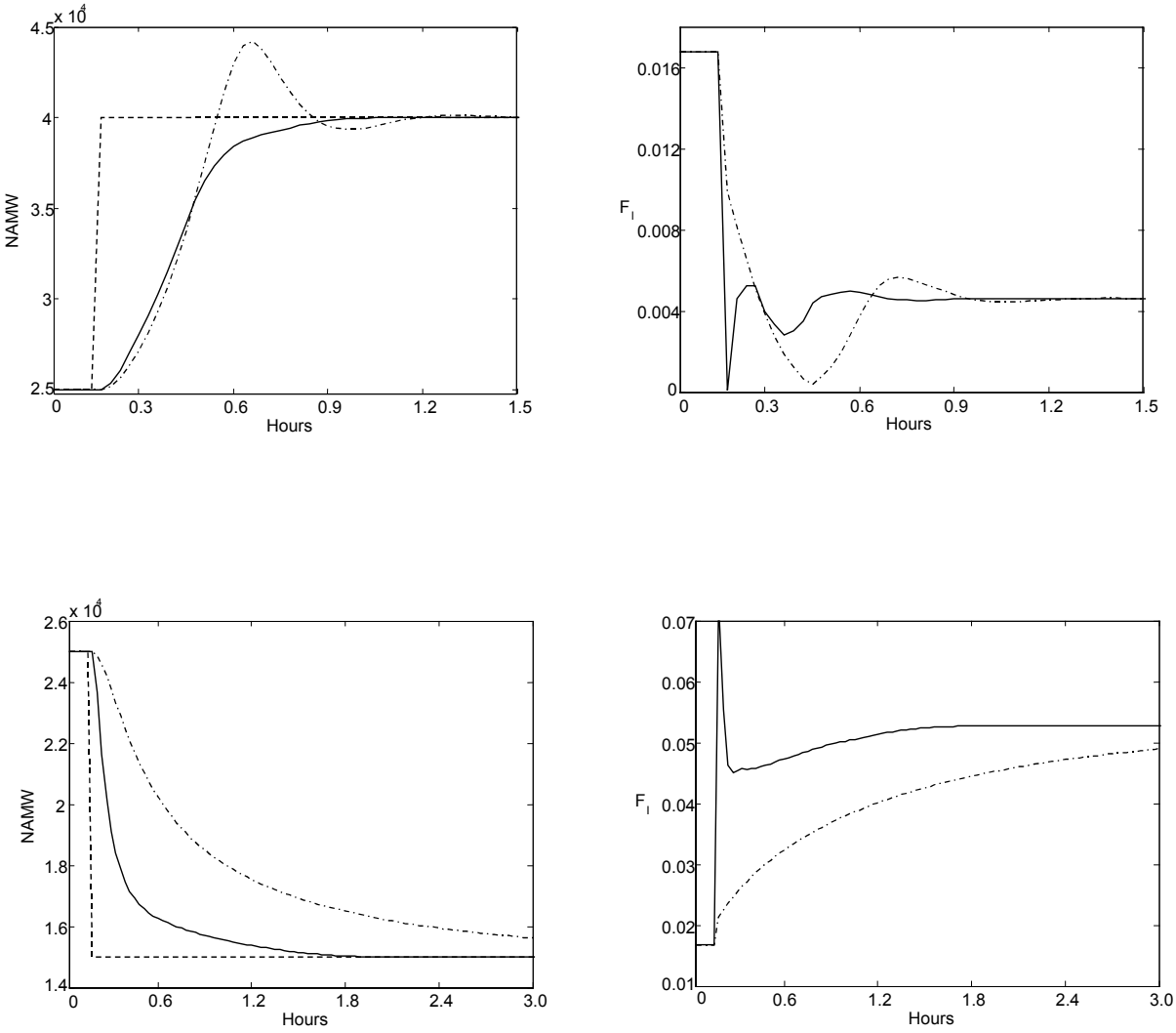


Figure 6.3 Closed-loop responses for set-point changes to 40000 kg/kmol (top) and 15000 kg/kmole (bottom). Dashed: set-point; solid: adaptive IMC; dashed-dot: IMC

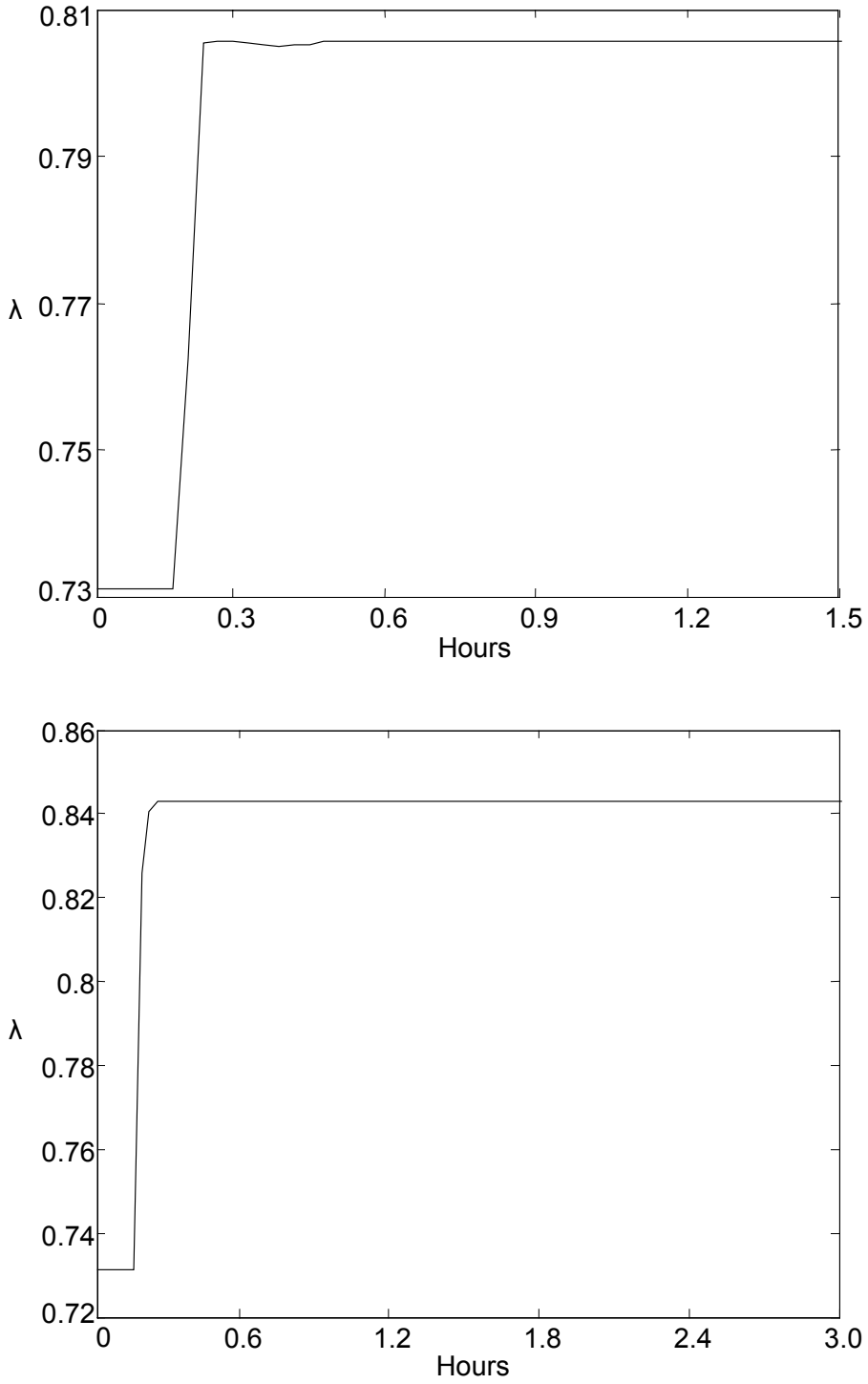


Figure 6.4 Updating of filter parameters for set-point changes to 40000 kg/kmol (top) and 15000 kg/kmol (bottom)

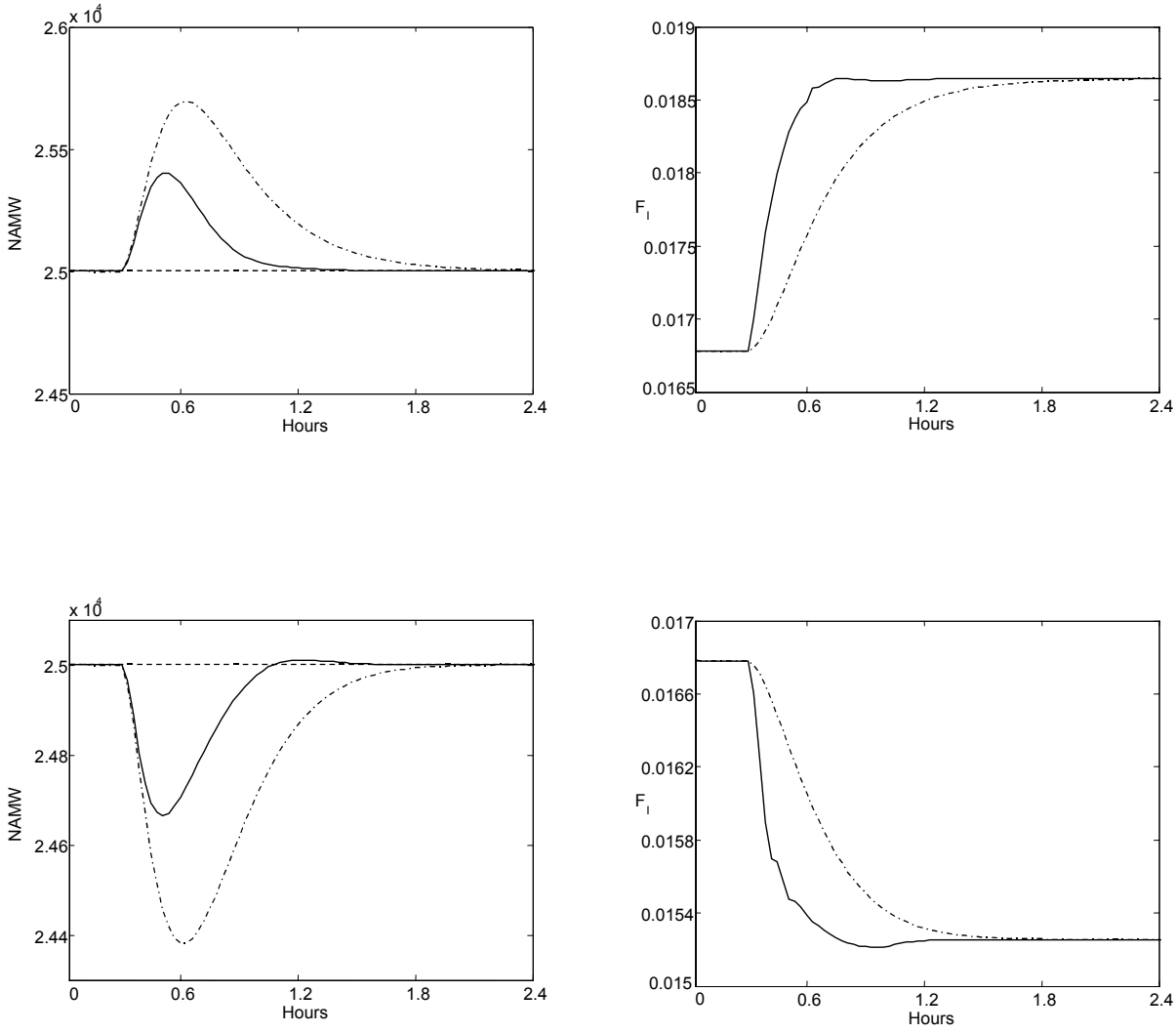


Figure 6.5 Closed-loop responses for -10% (top) and 10% (bottom) step disturbances in $C_{I_{in}}$. Dashed: set-point; solid: adaptive IMC; dashed-dot: IMC

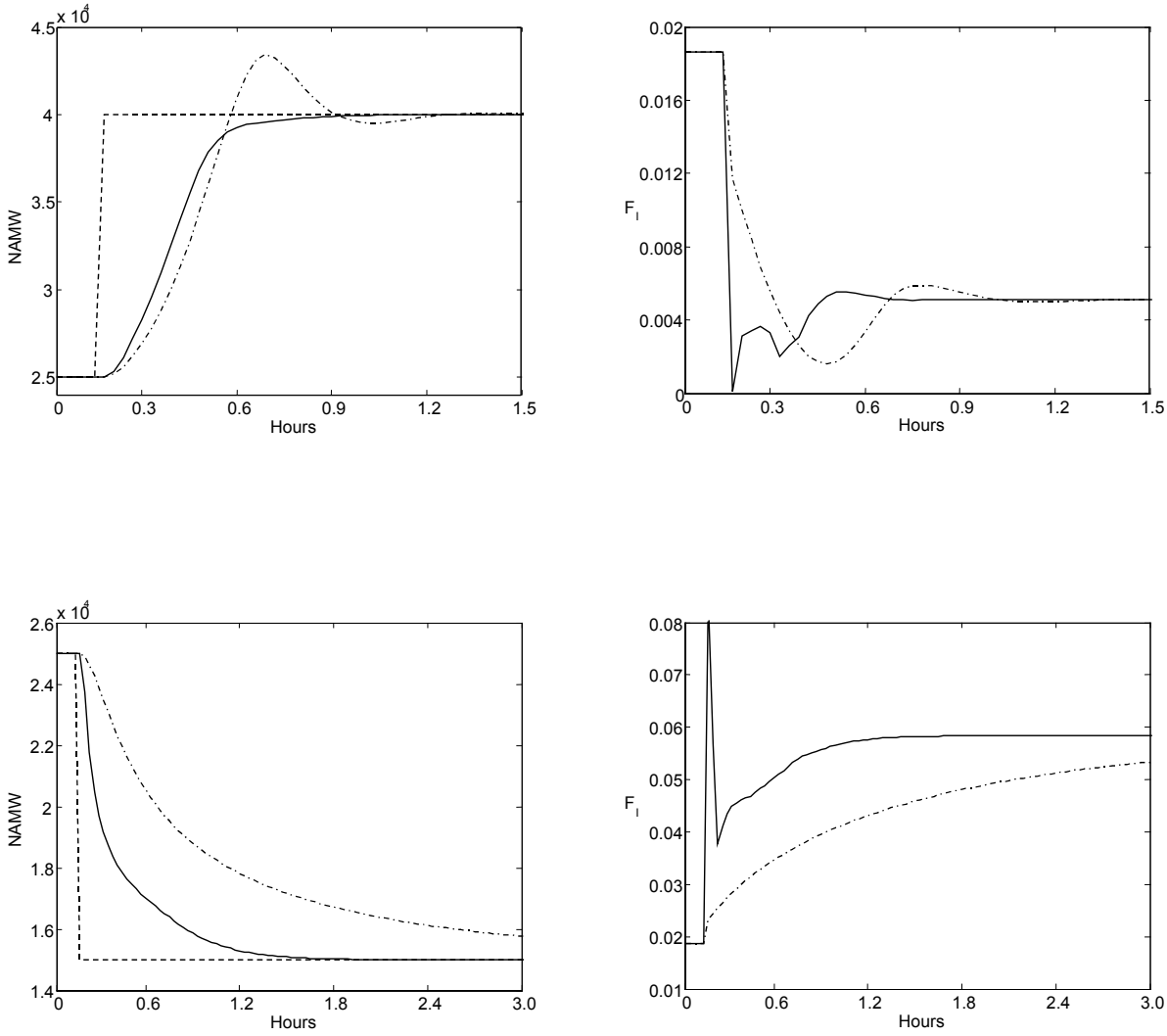


Figure 6.6 Closed-loop responses for set-point changes to 40000 kg/mol (top) and 15000 kg/kmol (bottom) under -10% modeling error in k_I. Dashed: set-point; solid: adaptive IMC; dashed-dot: IMC

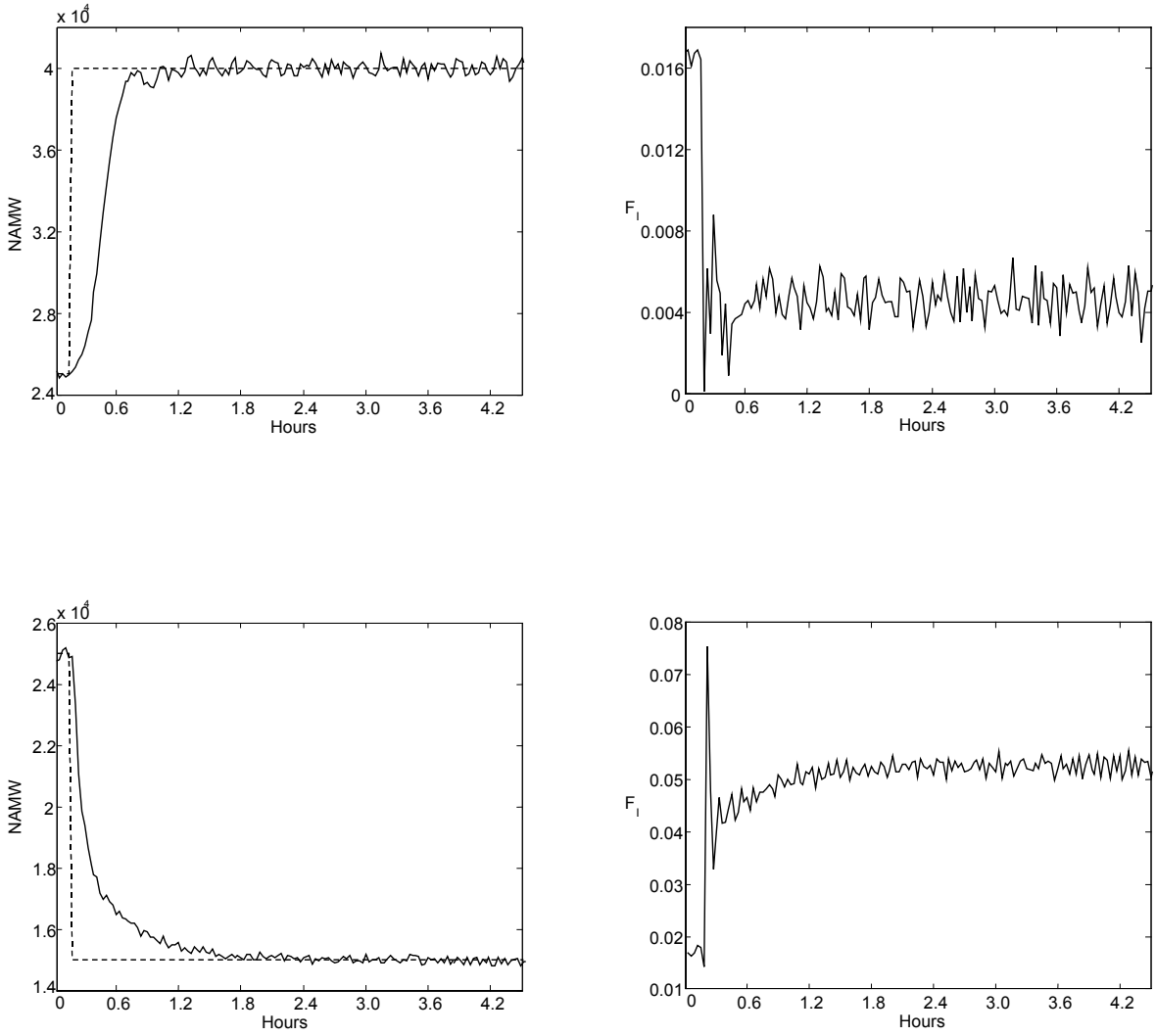


Figure 6.7 Servo responses in the presence of process noise

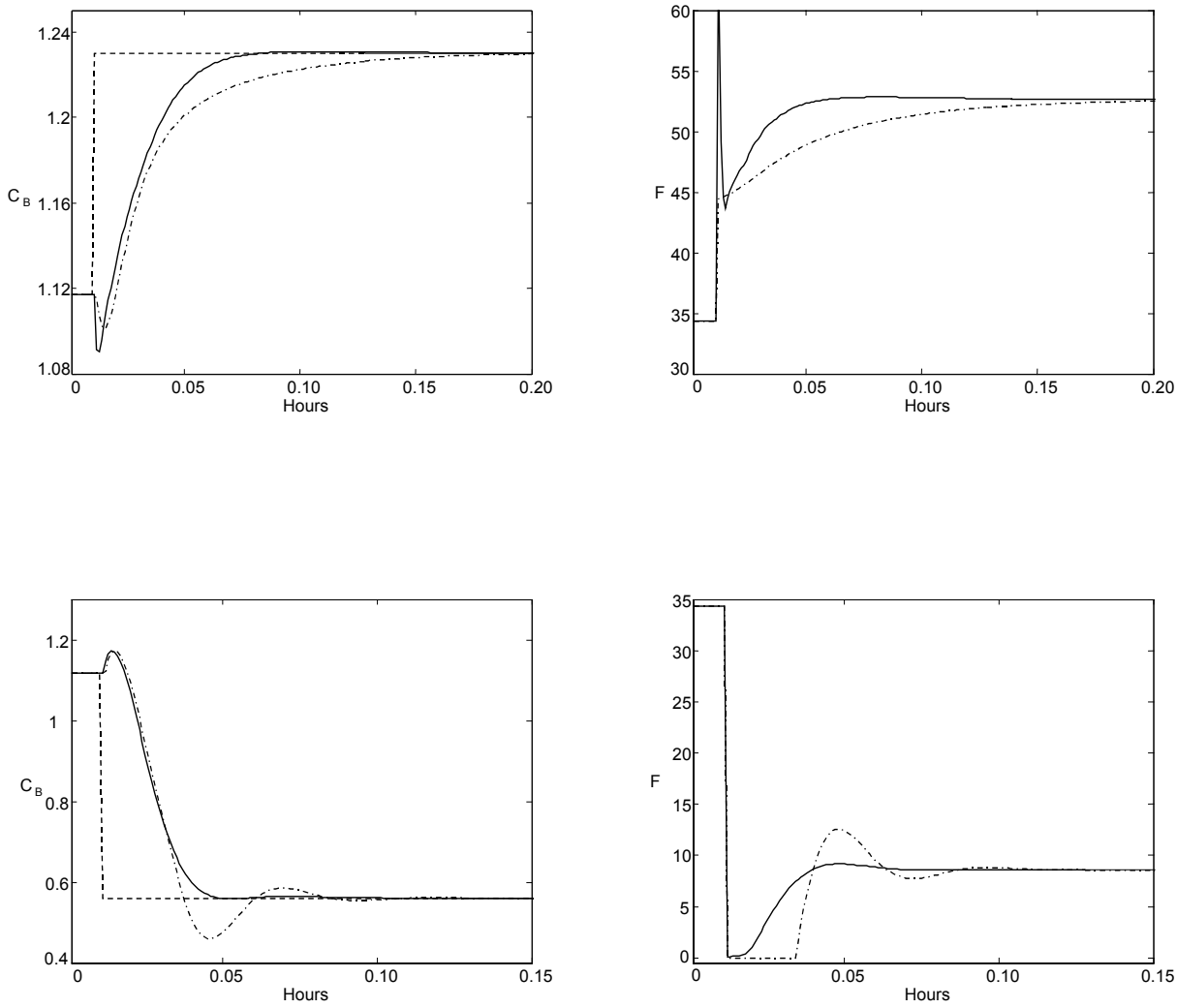


Figure 6.8 Closed-loop responses for $+10\%$ (top) and -50% (bottom) set-point changes. Dashed: set-point; solid: adaptive IMC; dashed-dot: IMC

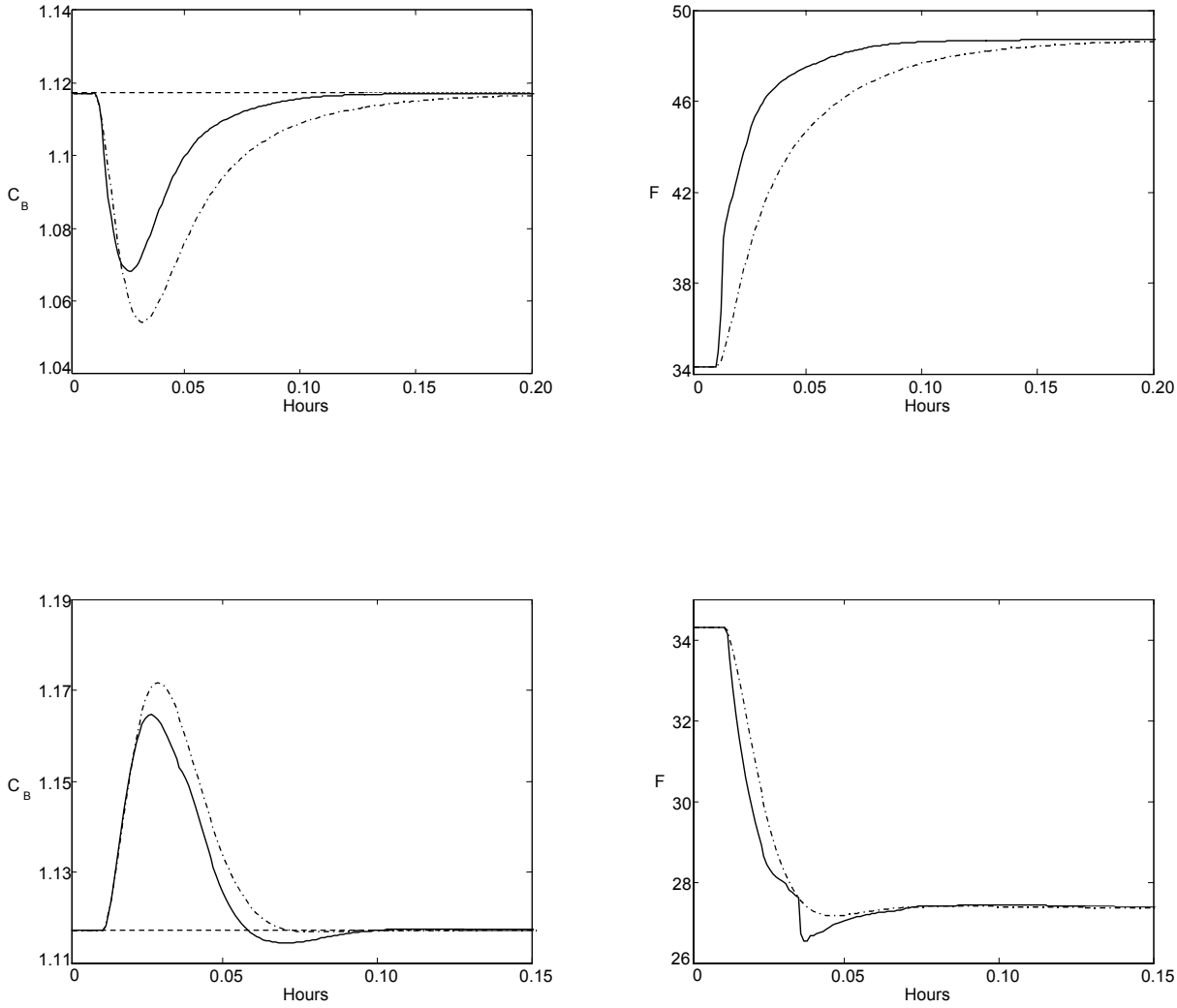


Figure 6.9 Closed-loop responses for -10% (top) and 10% (bottom) step disturbances in C_{Af} . Dashed: set-point; solid: adaptive IMC; dashed-dot: IMC

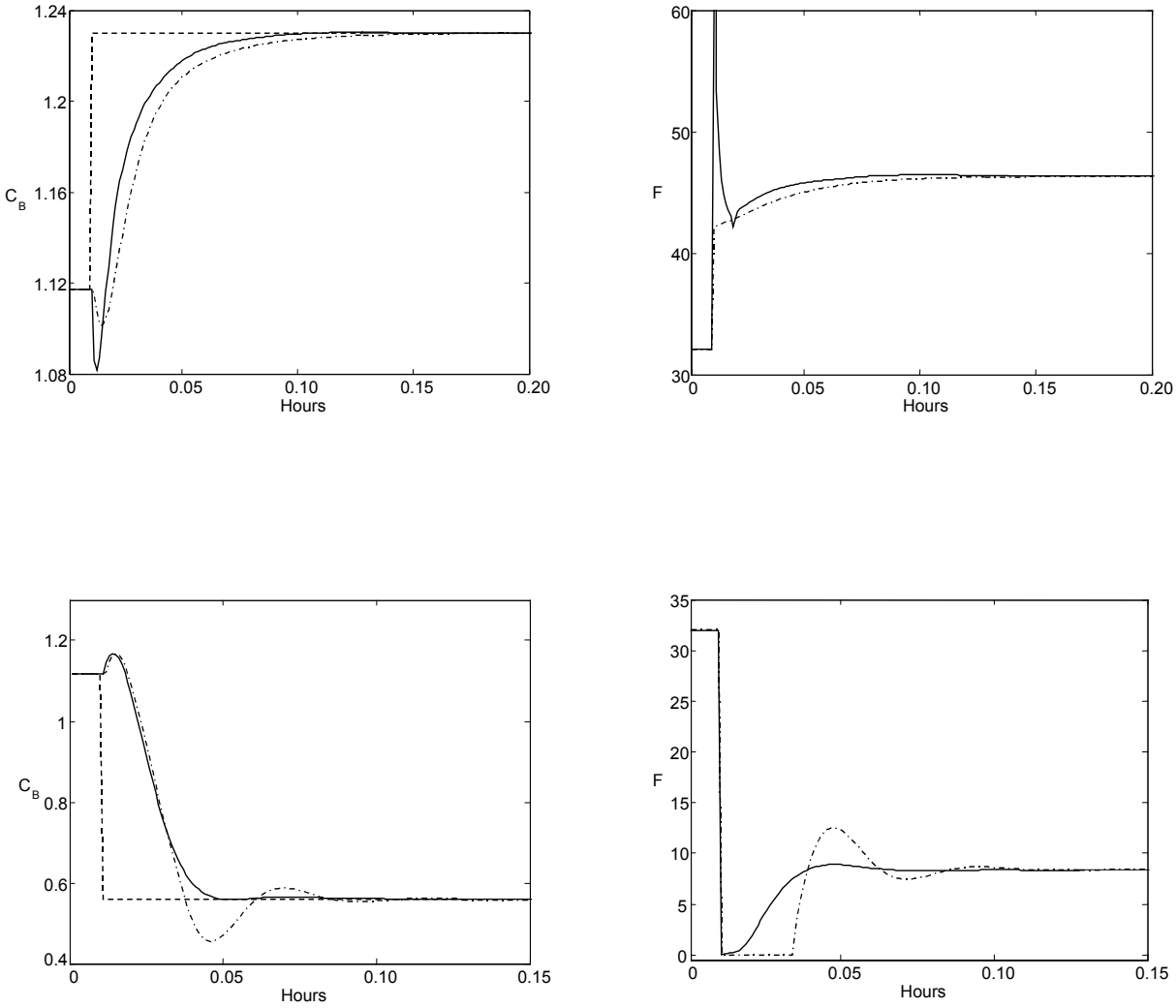


Figure 6.10 Closed-loop responses for 10% (top) and -50% (bottom) set-point changes under -10% modeling error in k_3. Dashed: set-point; solid: adaptive IMC; dashed-dot: IMC

Chapter 7

Auto-Tuning PID Controller Design

7.1 Introduction

It is well known that proportional-integral-derivative (PID) controller has gained widespread use in many process control applications due to its simplicity in structure, robustness in operation, and easy comprehension in its principle. It can be thus said to be the “bread and butter” of control engineering (Astrom and Hagglund, 1995). Numerous tuning methods have already been proposed to achieve this purpose, like Cohen-Coon (C-C), Ziegler-Nichols (Z-N), dominant pole design (Astrom and Hagglund, 1995). Nevertheless, it is difficult for conventional PID algorithms to obtain good control performance for nonlinear processes by simply using the fixed PID parameters. As we discussed in Chapter 5, a lot of artificial intelligence methods like neural networks have been proposed to auto-tune PID parameters on-line to improve the control performance. However, these methods are computationally expensive as inevitably required by the associated highly complex learning algorithms

developed to tune the PID parameters, which hampers the use of these methods in practical applications.

Compared with the traditional learning methods like neural networks, the JITL technique just stores data in the database and then based on the new observation, the relevant information from database is extracted to construct the predicted output for the new observation. In Chapters 5 and 6, we have investigated two nonlinear controller designs by incorporating the JITL technique. However, in the previous studies, JITL only serves as the process model in the controller design, which influences the controller performance in an indirect manner. In this chapter, we explore the spirit of the JITL technique to construct a database for controller itself as well so that this controller database can have direct impact on the controller design, which is in sharp contrast with the previously design methods developed in Chapters 5 and 6. A similar design concept was discussed in the memory-based PID controller design proposed by Yamamoto et al. (2004). In the proposed method for auto-tuning PID controller, a controller database is constructed to store the known PID parameters with their corresponding information vectors, while another database is employed for the standard use by JITL for the modeling purpose. During the on-line implementation of this auto-tuning method, the controller database is used to extract the relevant information to obtain new PID parameters based on the current process dynamics characterized by the information vector. Moreover, the PID parameters thus obtained can be further updated on-line when the predicted control error is greater than a pre-specified threshold and the resulting updated PID parameters with their corresponding information vector are stored into the controller database. Finally, it is worthy pointing out that the initialization of the present controller design requires less trial and error effort than that for the controller designs proposed in Chapters 5 and 6.

This is because the initial controller database can be easily constructed from the closed-loop data available in the historical operating data.

In the following sections, we first analyze how to select the information vector for the controller database and the design procedure to obtain the controller parameters from the controller database, then a learning strategy is provided to update the controller parameters and controller database, whenever necessary, to realize the auto-tuning PID parameters on-line. Lastly, literature examples are presented to illustrate the proposed control strategy.

7.2 Auto-Tuning PID Controller Design

7.2.1 Information vector selection

Different from the controllers developed in Chapters 5 and 6, the proposed auto-tuning PID design as depicted in Figure 7.1 requires not only the database used for JITL for the modeling purpose but also the controller database to be exploited by the on-line tuning algorithm to extract the relevant information in order to compute PID parameters at every sampling instant.

Undoubtedly, the objective to explore the controller database is to design PID parameters based on the current closed-loop dynamics characterized by the corresponding information vector. In what follows, we shall first discuss how to select the information vector and for that matter the content of the controller database.

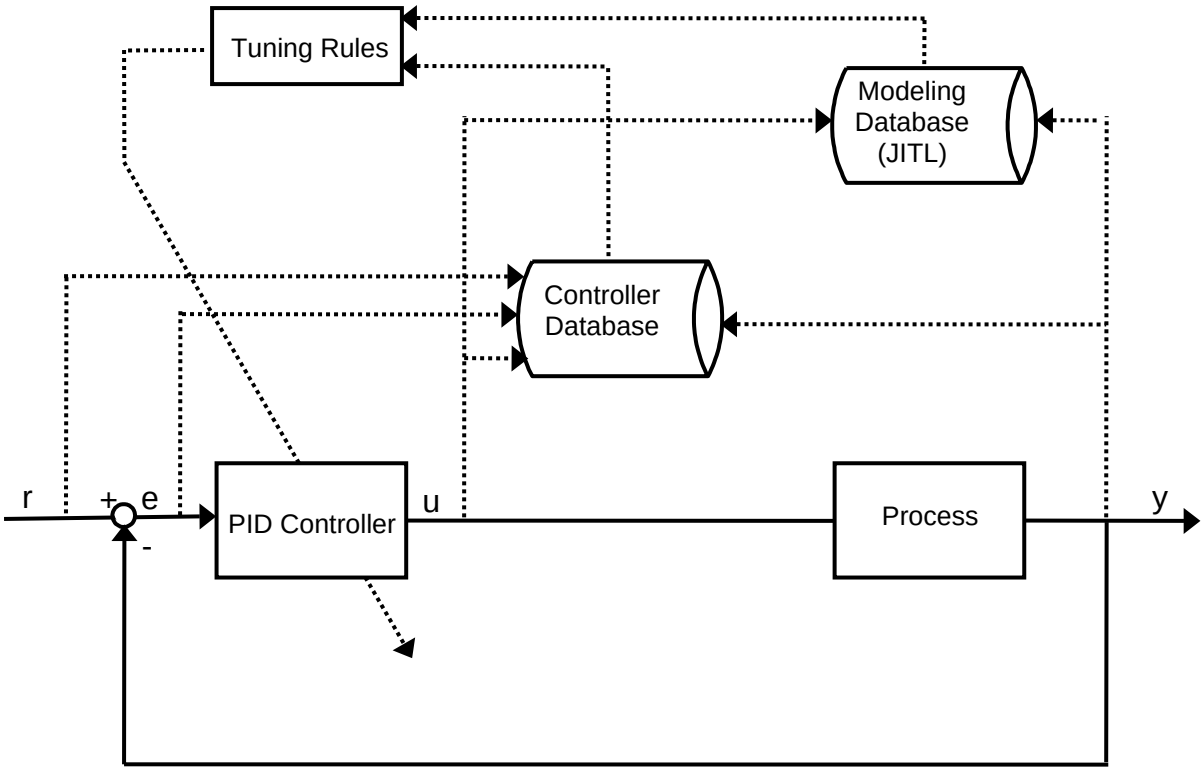


Figure 7.1 Block diagram of the auto-tuning PID design

Consider the following continuous-time expression for PID controller:

$$u(t) = K_p[e(t) + \frac{1}{\tau_i}\int_0^t e(t)d\tau + \tau_d \frac{d}{dt}e(t)] \tag{7.1}$$

where $u(t)$ is the process input, $e(t)$ is the error, K_p is the proportional constant, τ_i is the integral time, and τ_d is the derivative time. The velocity form of PID algorithm can be obtained from Eq. (7.1) as follows:

$$u(k) = u(k-1) + K_p(e(k) - e(k-1)) + K_i e(k) + K_d(e(k) - 2e(k-1) + e(k-2)) \tag{7.2}$$

where $u(k)$ and $e(k)$ are discrete time signals at the k-th sampling instant, $K_i = K_P \frac{T}{\tau_i}$, $K_d = K_p \frac{\tau_d}{T}$, and T is the sampling time.

From Eq. (7.2), it is clear that the controller output $u(k)$ is the function of PID parameters and process input, i.e.

$$u(k) = g(\mathbf{K}(k),\ e(k), e(k-1), e(k-2), u(k-1)) \tag{7.3}$$

where $g(\cdot)$ denotes a linear function and

$$\mathbf{K}(k) = [K_p(k),\ K_i(k),\ K_d(k)] \tag{7.4}$$

Suppose that the nonlinear process under control can be described as following:

$$y(k+1) = h(y(k), \ldots,\ y(k-n_y+1), u(k), \ldots, u(k-n_u+1)) \tag{7.5}$$

where $h(\cdot)$ denotes a nonlinear function, n_y and n_u are integers related to the model orders.

By combining Eqs. (7.3) and (7.5), the controller parameter $\mathbf{K}$ is the function of the following equation:

$$\begin{aligned}\mathbf{K}(k) = F(&y(k+1), y(k), \ldots,\ y(k-n_y+1),\\ &u(k-1), \ldots,\ u(k-n_u+1), e(k), e(k-1), e(k-2))\end{aligned} \tag{7.6}$$

where $F(\cdot)$ denotes a nonlinear function.

Consequently, the information vector which is used to construct the controller database is defined as following:

$$\begin{aligned}\sigma(k) = [&y(k+1), y(k), \ldots, y(k-n_y+1),\\ &u(k-1), \ldots, u(k-n_u+1), e(k), e(k-1), e(k-2)]\end{aligned} \tag{7.7}$$

However, since $y(k+1)$ in Eq. (7.7) is not available at the current sampling instant, $y(k+1)$ is replaced by $r(k+1)$. Eq. (7.7) is thus rewritten as:

$$\begin{aligned}\sigma(k) = [&r(k+1), y(k), \ldots, y(k-n_y+1),\\ &u(k-1), \ldots, u(k-n_u+1), e(k), e(k-1), e(k-2)]\end{aligned} \tag{7.8}$$

Based on the on-going analysis, it is clear that the controller parameter $\mathbf{K}(k)$ is a function of the information vector $\sigma(k)$ defined in Eq. (7.8). As a result, the data

pair in the controller database is chosen as $(\mathbf{K}(k), \sigma(k))$. In the next subsection, we will discuss how the initial controller database is constructed off-line and the procedures to obtain the PID parameters from the controller database on-line. In addition, a criterion to update the controller database on-line will be addressed.

7.2.2 Controller design

The initial controller database can be easily constructed from the closed-loop data when the process is under PID control around the nominal operating condition. For example, a PID controller is designed to give good control performance around the nominal operating condition and subsequently successive set-point changes around the nominal operating condition are conducted, from which the resulting closed-loop data can be measured to construct the information vectors $(\sigma(i))_{i=1\sim N_0}$, where N_0 is the number of data points in the initial controller database. Alternatively, the available historical closed-loop data can be used for the same purpose. Because a fixed-parameter PID controller is employed in constructing the initial controller database, we have $\mathbf{K}(1) = \mathbf{K}(2) = \cdots = \mathbf{K}(N_0)$.

With the available initial controller database, auto-tuning PID design can be conducted as discussed in the sequel. At each sampling instant, the following measure between the current information vector $\sigma(k)$ obtained from the closed-loop system and that in the controller database is calculated:

$$s_i = \sqrt{e^{-\|\sigma(k)-\sigma(i)\|^2}}, \; i = 1 \sim N \tag{7.9}$$

where $\|\cdot\|$ is an Euclidean norm and N is the number of data points in the current controller database.

To extract PID parameters from controller database, those relevant information vectors or nearest neighbours resemble the current information vector $\sigma(k)$ are identified. In this work, l nearest neighbours are selected corresponding to the largest s_i to the l-th largest s_i. Subsequently, a weight for each neighbour is calculated using the following equation:

$$\omega_i = \frac{s_i}{\sum_{i=1}^{l} s_i}, \qquad \sum_{i=1}^{l} \omega_i = 1 \tag{7.10}$$

The PID parameters are then obtained by using the following formula:

$$\mathbf{K}^0(k) = \sum_{i=1}^{l} \omega_i \mathbf{K}(i) \tag{7.11}$$

and the corresponding controller output is obtained by using Eq. (7.2) as:

$$\hat{u}(k) = u(k-1) + K_p^0(e(k) - e(k-1)) + K_i^0 e(k) + K_d^0(e(k) - 2e(k-1) + e(k-2)) \tag{7.12}$$

Because the initial controller database is constructed by using the local data around the nominal operating condition, it may not provide adequate information to adjust PID parameters effectively when the operating condition is away from the nominal one. In this situation, the PID parameters $\mathbf{K}^0(k)$ and $\hat{u}(k)$ for that matter need further refinement and there is also a need to update the controller database by expanding it to include the current information vector and PID parameters. To determine whether $\mathbf{K}^0(k)$ is satisfactory or not, the following criterion is introduced:

$$\left|\frac{y_r(k+1) - \hat{y}(k+1)}{y_r(k+1)}\right| < \varepsilon \tag{7.13}$$

where $y_r(k+1)$ is the reference trajectory at the next sampling instant $k+1$, $\hat{y}(k+1)$ is the predicted output by the JITL by using $\mathbf{K}^0(k)$, and ε is the threshold. Evidently, $\mathbf{K}^0(k)$ is considered to give good control performance if the above inequality is

satisfied and thus there is no need for further refinement. On the other hand, when the above inequality does not hold, $\mathbf{K}^0(k)$ can be refined further by the steepest descent method to be given in what follows.

Similar to what was done in Chapters 5 and 6, the following quadratic function is used as the objective function for the updating law of PID parameters:

$$MinJ = (y_r(k+1) - \hat{y}(k+1))^2 + \kappa(\hat{u}(k) - u(k-1))^2 \tag{7.14}$$

where κ is a weight parameter.

Since the PID parameters are constrained to be positive or negative, to automatically incorporate the constraints to the objective function, the following mapping function is introduced:

$$K_x(k) = \begin{cases} e^{\varsigma_x(k)}, & \text{if} \quad K_x(k) \geq 0 \\ -e^{\varsigma_x(k)}, & \text{if} \quad K_x(k) < 0 \end{cases}, \quad x = p,i,d \tag{7.15}$$

where ς_x is a real number. Similar to Eq. (7.15), $\varsigma_x^0(k)$ denotes the corresponding mapping variable with respect to the PID parameters $\mathbf{K}^0(k)$. The following updating algorithm can be derived to improve the design of $\varsigma_x^0(k)$:

$$\begin{aligned} \varsigma_x^{new}(k) &= \varsigma_x^0(k) - \eta_x \frac{\partial J}{\partial \varsigma_x^0(k)} \\ &= \varsigma_x^0(k) - \eta_x \frac{\partial J}{\partial K_x^0(k)} K_x^0(k) \end{aligned} \tag{7.16}$$

$$\frac{\partial J}{\partial K_x^0(k)} = \frac{\partial J}{\partial u(k)} \frac{\partial u(k)}{\partial K_x^0(k)} \tag{7.17}$$

$$\frac{\partial J}{\partial u(k)} = -2(y_r(k+1) - \hat{y}(k+1)) \frac{\partial \hat{y}(k+1)}{\partial u(k)} + 2\kappa(\hat{u}(k) - u(k-1)) \tag{7.18}$$

for $x = p,i,d$. In above equations, the derivative $\frac{\partial \hat{y}(k+1)}{\partial u(k)}$ can be obtained from the JITL model, η_x are the respective learning rate for ς_x^0, and

$$\frac{\partial u(k)}{\partial K_p(k)} = e(k) - e(k-1) \tag{7.19}$$

$$\frac{\partial u(k)}{\partial K_i(k)} = e(k) \tag{7.20}$$

$$\frac{\partial u(k)}{\partial K_d(k)} = e(k) - 2e(k-1) + e(k-2) \tag{7.21}$$

After $\varsigma_x^{new}(k)$ is calculated by Eq. (7.16), the corresponding new PID parameters are obtained from Eq. (7.15). Furthermore, these new PID parameters and their corresponding information vector are stored into controller database. The implementation of the proposed control algorithm is summarized as follows:

1. Given the threshold ε, weight parameter κ, learning rate η_x, number of nearest-neighbour l, and the initial controller database;
2. At each sampling instant, calculate PID parameters $\mathbf{K}^0(k)$ according to Eq. (7.11) and the corresponding controller output $\hat{u}(k)$ by Eq. (7.12), by which $\hat{y}(k+1)$ is obtained from JITL model;
3. Evaluate the fitness of $\mathbf{K}^0(k)$ by the criterion given in Eq. (7.13). If this criterion is satisfied, $\hat{u}(k)$ is implemented to the process and set $k = k+1$ and go to step 2. Otherwise, go to step 4;
4. Update $\varsigma_x^0(k)$ by Eq. (7.16) and calculate the resulting new PID parameters using Eq. (7.15), by which controller output is obtained and implemented to the process. In addition, the controller database is updated by adding new PID parameters and their corresponding information vector. Set $k = k+1$ and go to step 2.

7.3 Examples

Example 1 The first example considered is the control of polymerization reaction as studied in Chapters 5 and 6. The model parameters and steady-state operation condition can be found in Tables 5.1 and 5.2. In addition, the same database and parameters for the JITL used in Chapters 5 and 6 are employed.

To construct the initial controller database, the information vector $\sigma(k) = [r(k+1), y(k),\ y(k-1),\ u(k-1),\ e(k),\ e(k-1), e(k-2)]$ is specified because a second-order ARX model has been employed by the JITL. Furthermore, a PID controller with parameters, $K_p = -0.8$, $K_i = -0.25$, and $K_d = -0.3$, is designed to give good control performance for $\pm 10\%$ set-point changes around the nominal operating condition, as shown in Figure 7.2. This PID controller is then used in a closed-loop experiment consisting of multiple set-point changes as illustrated in Figure 7.3, from which one hundred information vectors (i.e. $N_0 = 100$) are obtained to construct the initial controller database. In addition, the following parameters are chosen in the proposed design: the threshold $\varepsilon = 0.05$, the weight parameter $\kappa = 0.2$, the number of nearest-neighbour $l = 5$, the learning rates $\eta_p = 0.15$, $\eta_i = 0.4$, and $\eta_d = 0.3$, and the reference trajectory given by:

$$y_r(k) = 0.95r(k) + 0.05y_r(k-1) \tag{7.22}$$

To evaluate the performance of the proposed controller, set-point change from 25000.5 to 40000 kg/kmol is initially conducted, followed by the -20% step disturbance in $C_{I_{in}}$. For the comparison purpose, the control performance of the PID controller aforementioned is also shown in Figure 7.4. It is clear that the PID controller gives a large overshoot while the proposed controller reaches to the set-point quickly with less overshoot, resulting in 8.2% reduction of the MAE. For

disturbance rejection, the proposed controller gives better control performance by reducing the MAE by 23.1% . Figure 7.5 compares the performances of two controllers for set-point change from 25000.5 to 15000 kg/kmol and the subsequent -20% step disturbance in $C_{I_{in}}$. Evidently, the proposed controller not only reaches the set-point much faster but also displays better disturbance rejection performance than the PID controller, leading to 27.5% and 35.5% reduction of MAE, respectively. The updating of the parameters of the proposed controller in the aforementioned closed-loop responses are shown in Figure 7.6.

Although the PID controller is designed to provide good control performance around the nominal operating condition as shown in Figure 7.2, it is not capable of producing the same level of good control performance for the set-point changes given in Figures 7.4 and 7.5, as evidenced by an oscillatory response for the set-point change to 40000 kg/kmol and a sluggish response for the set-point change to 15000 kg/kmol. As a result, if more aggressive PID controllers are designed to avoid sluggish response for set-point change to 15000 kg/kmol, this will inevitably make the servo response for the set-point change to 40000 kg/kmol highly oscillatory. Likewise, more conservative PID controller design can eliminate the oscillatory response in the latter case, but at the expense of even more sluggish response in the former case.

Next, to evaluate the robustness of the proposed control controller, it is assumed that the process kinetic parameter k_I is subject to -10% modeling error and the resulting servo responses of two controllers are compared in Figure 7.7. It is evident that the proposed controller still maintains better control performance by achieving 8.3% reduction of MAE for set-point change to 40000 kg/kmol and 33.2% reduction for set-point change to 15000 kg/kmol. Furthermore, to study the effect of process noise on the proposed design, both process input and output are corrupted by

1% Gaussian white noise. As can be seen from Figure 7.8, the proposed controller can yield reasonably good control performance in the presence of process noise. Lastly, Figure 7.9 compares the performances of the proposed controller and the benchmark IMC controller designed in Chapters 5 and 6. Obviously, the proposed controller outperforms the IMC controller by reducing 34.6% and 62.6% of MAE for the set-point changes to 40000 kg/kmol and 15000 kg/kmol, respectively.

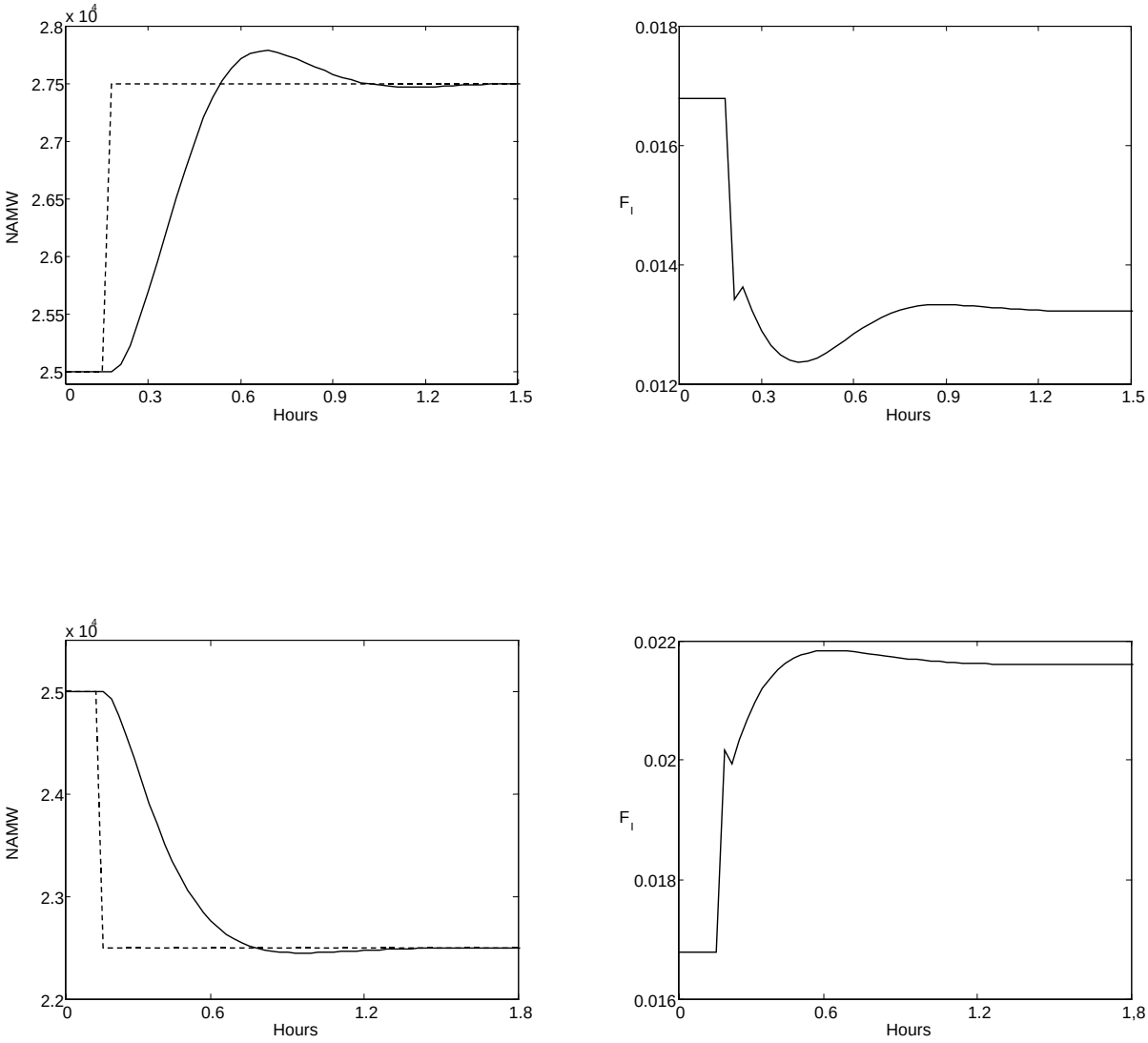


Figure 7.2 Servo responses of the PID controller for ±10% set-point changes. Dashed: set-point; solid: PID

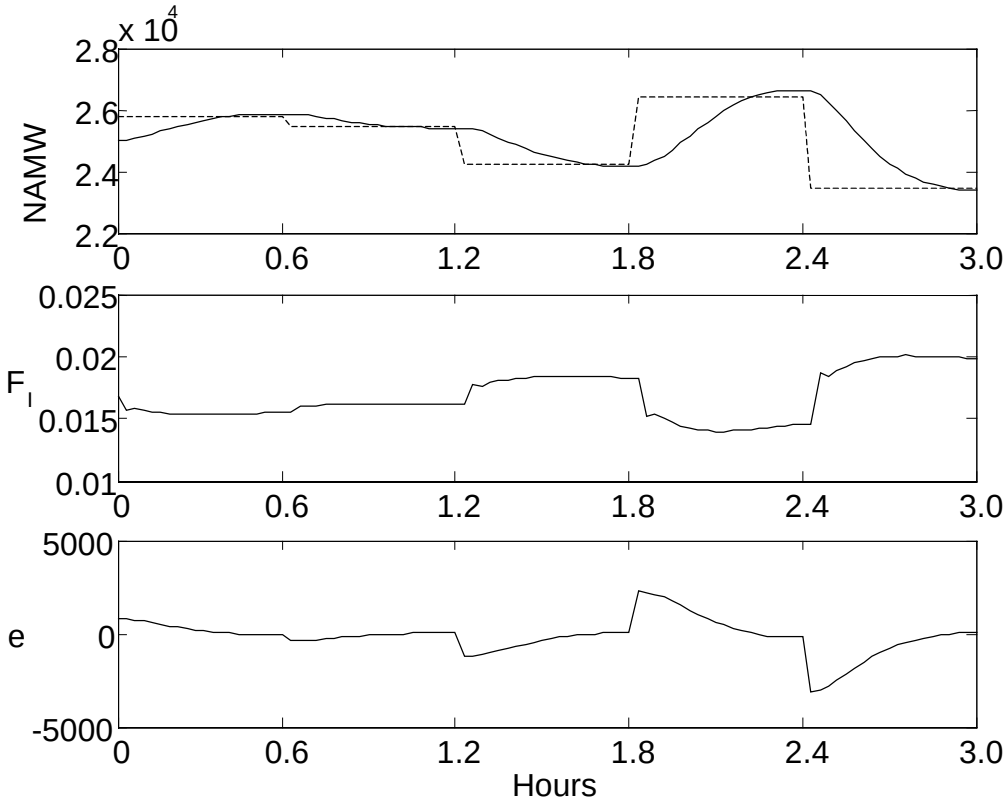


Figure 7.3 Data used for constructing the initial controller database

Example 2 Consider the control problem of the van de Vusse reaction (Doyle et al., 1995) as discussed in the previous chapters. The same nominal operation conditions given in Chapters 5 and 6 are considered, i.e. $C_{A0} = 3.0$, $C_{B0} = 1.1172$, and $F_0 = 34.3$. To apply the proposed control strategy, the identical database and parameters for the JITL used in Chapters 5 and 6 are employed. Furthermore, initial controller database is generated by a PID controller with parameters, $K_p = 2$, $K_i = 0.2$, and $K_d = 1$, which gives good control performance in the vicinity of the nominal operation condition, as depicted in Figure 7.10. Because a second-order ARX model is employed for the JITL, the information vector of the controller database is

given by $\sigma(k) = [r(k+1),\ y(k),\ y(k-1),\ u(k-1), e(k), e(k-1), e(k-2)]$ with other parameters chosen as follows: $\varepsilon = 0.05$, $\kappa = 0.1$, $N_0 = 100$, $l = 5$, $\eta_p = 0.3$, $\eta_i = 0.45$, $\eta_d = 0.5$, and the following reference trajectory:

$$y_r(k) = 0.9r(k) + 0.1y_r(k-1) \tag{7.23}$$

To compare the performances of the proposed controller and the PID controller aforementioned, 10% set-point change followed by -2% step disturbance in C_{Af} is conducted. It is apparent from Figure 7.11 that the proposed controller reaches to the set-point much faster than the PID controller, as supported by the reduction of MAE by 24.3% . Furthermore, the proposed controller has better disturbance rejection capability than the PID controller, resulting in 34.3% reduction of MAE. Figure 7.12 compares the performances of two controllers for -50% set-point change and -20% step disturbance in C_{Af} . The proposed controller yields less oscillatory response and has shorter setting time than the PID controller, resulting in 9.6% and 12.2% reduction of MAE, respectively. The updating of the parameters of the proposed controller in the aforementioned closed-loop responses are shown in Figure 7.13.

The robustness of the proposed controller is also evaluated by assuming -10% modeling error in the kinetic parameter k_3. It can be seen from Figure 7.14 that the proposed controller achieves better performance than the PID controller, leading to the respective 17.3% and 9.8% reduction of MAE for 10% and -50% set-point changes. Lastly, the performances of the proposed controller and the IMC controller designed in Chapters 5 and 6 are compared in Figure 7.15. Again, the proposed controller shows better control performance than the IMC controller by

reducing MAE by 14.9% and 9.1% for 10% and −50% set-point changes, respectively.

7.4 Conclusion

A new auto-tuning PID controller is proposed for nonlinear process control in this chapter. In the proposed design strategy, the spirit of JITL technique is exploited for controller design directly, i.e. a controller database is constructed to contain the known PID parameters and their corresponding information vectors for controller design purpose, while another database is employed for the standard use by JITL algorithm for process modeling purpose. During the on-line implementation, the controller database is used to extract the relevant information to obtain new PID parameters based on the current process dynamics characterized by the current information vector. Moreover, the PID parameters obtained can be further updated when the predicted control error is greater than a pre-specified threshold and the resulting PID parameters with their corresponding information vector are stored into the controller database. The proposed method is evaluated through the simulation studies to show better control performance than its conventional counterpart.

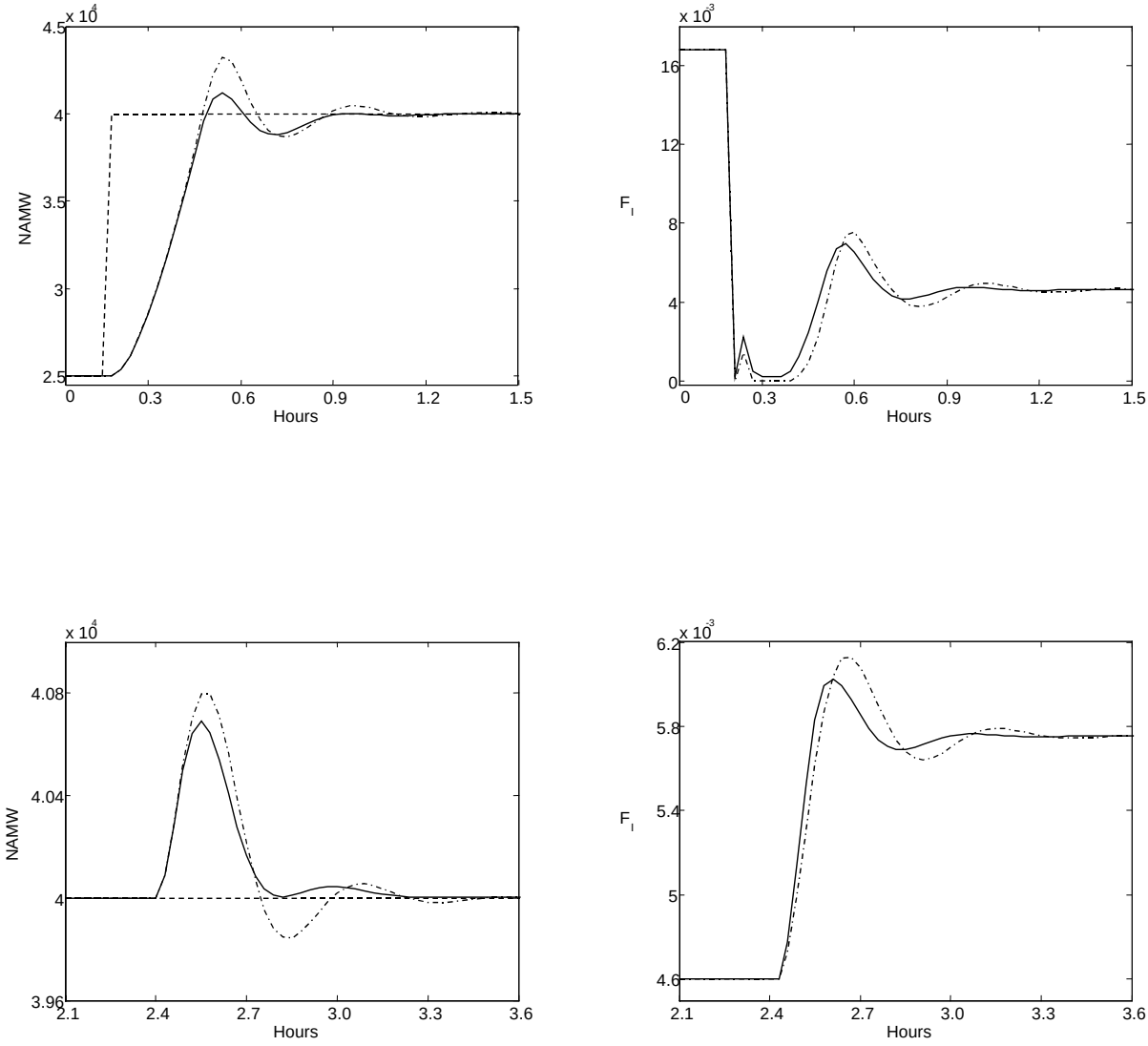


Figure 7.4 Closed-loop responses for set-point change to 40000 kg/kmol (top) and -20% step disturbance in $C_{I_{in}}$ (bottom). Dashed: set-point; solid: the proposed method; dashed-dot: PID

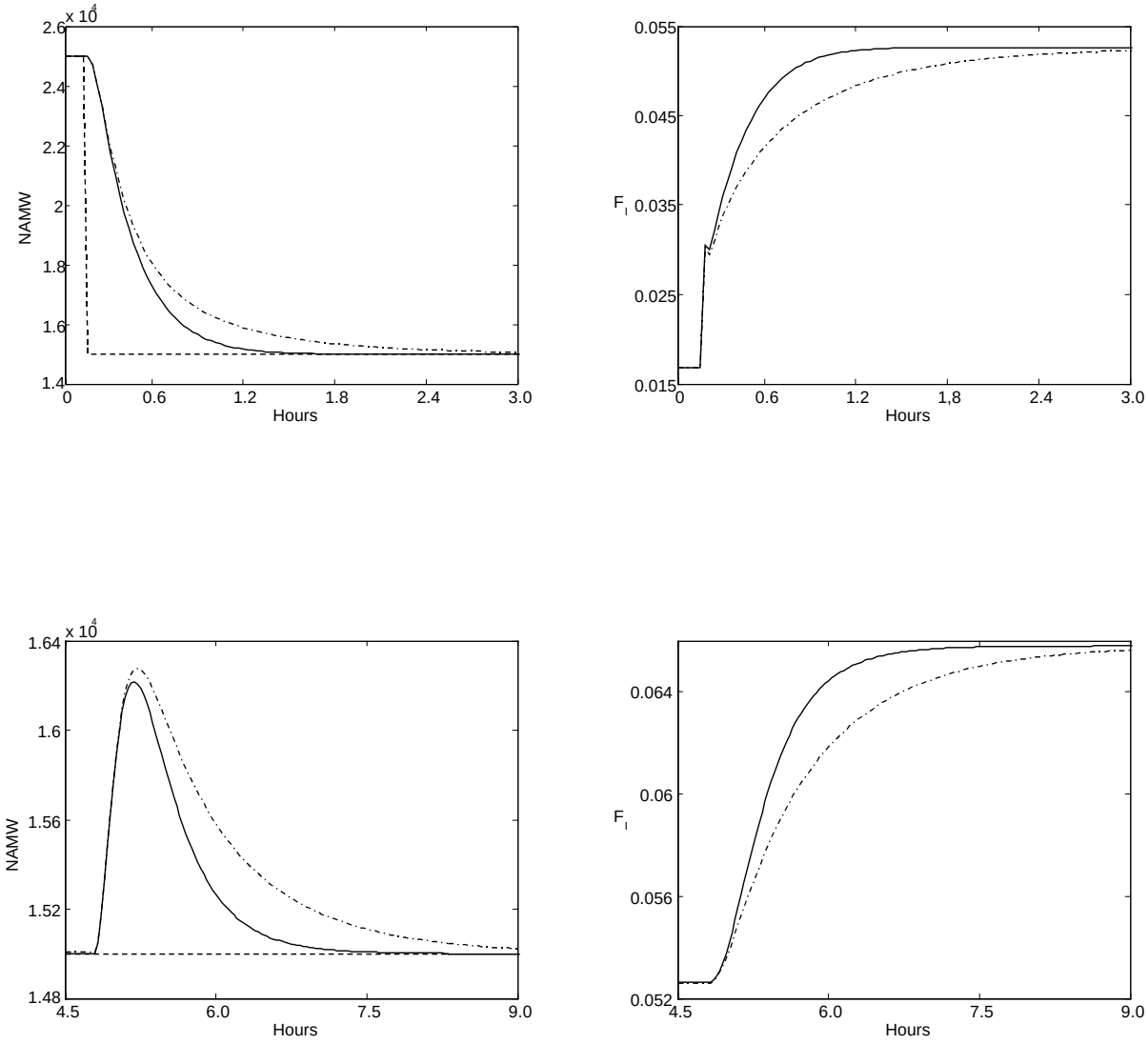
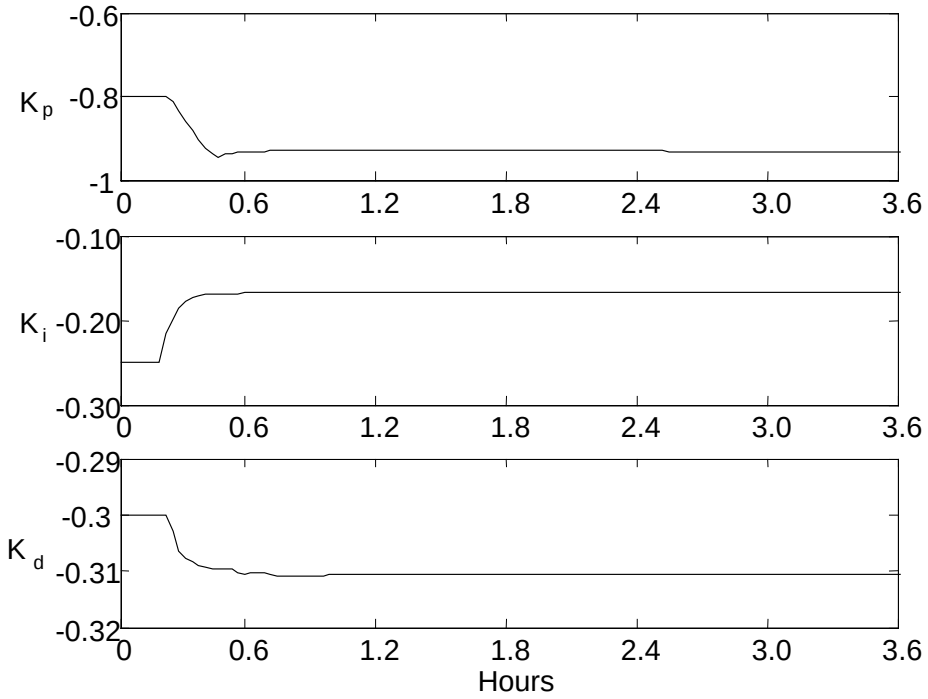


Figure 7.5 Closed-loop responses for set-point change to 15000 kg/kmol (top) and -20% step disturbance in $C_{I_{in}}$ (bottom). Dashed: set-point; solid: the proposed method; dashed-dot: PID



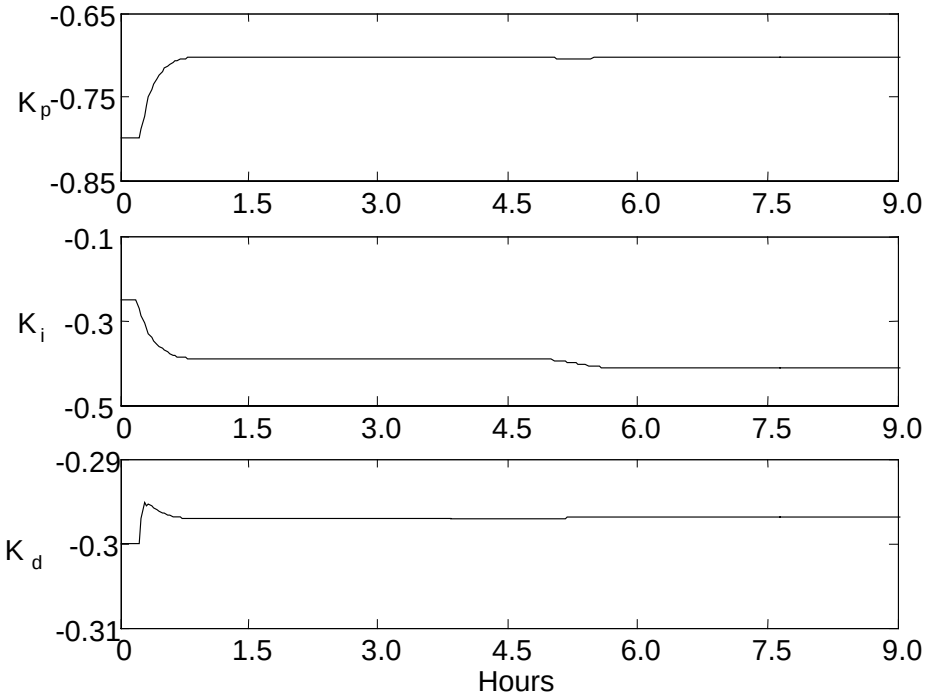


Figure 7.6 Updating of PID parameters for the closed-loop responses given in Figure 7.4 (top) and Figure 7.5 (bottom)

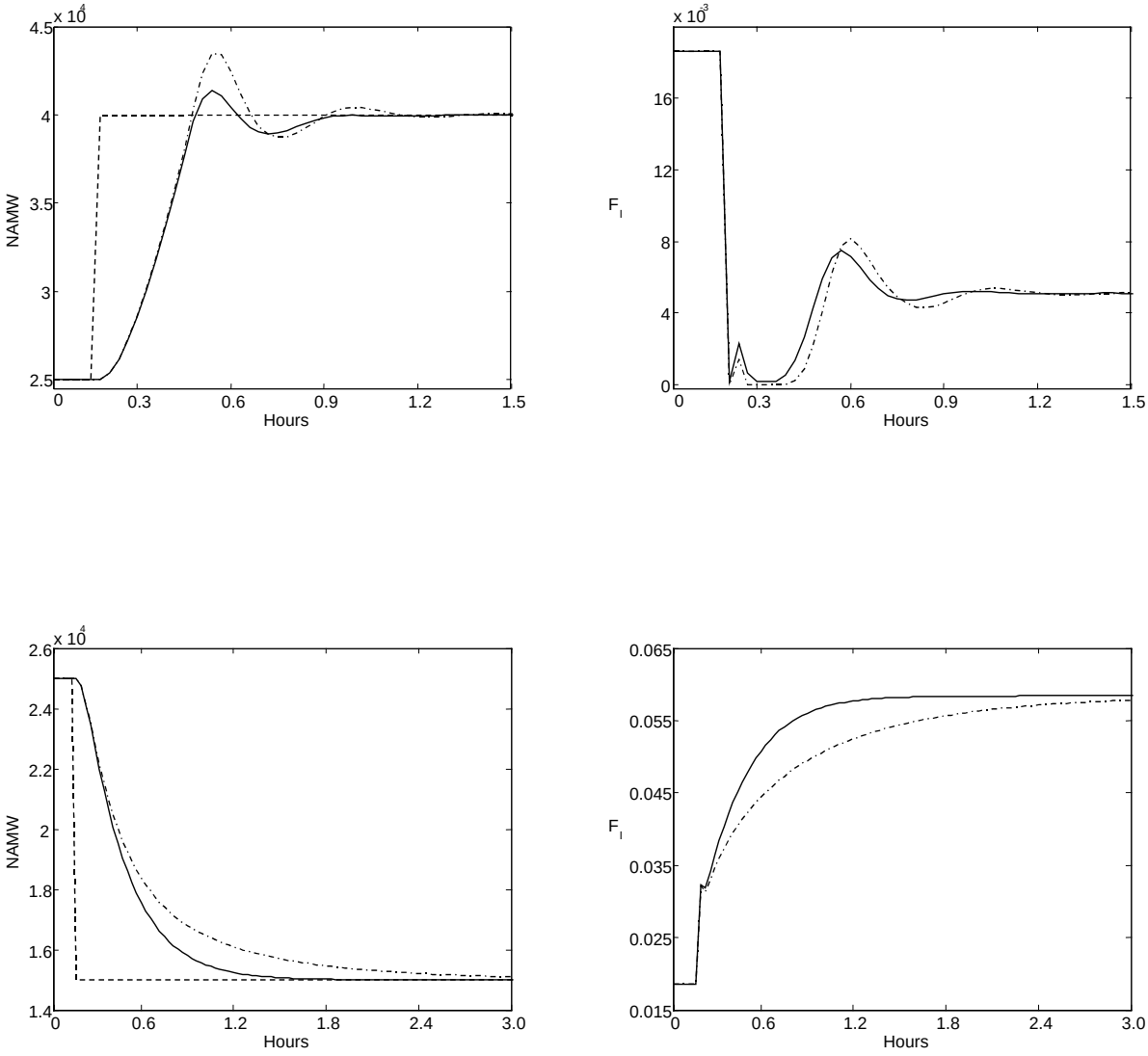


Figure 7.7 Closed-loop responses for set-point changes to 40000 kg/mol (top) and 15000 kg/kmol (bottom) under -10% modeling error in k_I. Dashed: set-point; solid: the proposed method; dashed-dot: PID

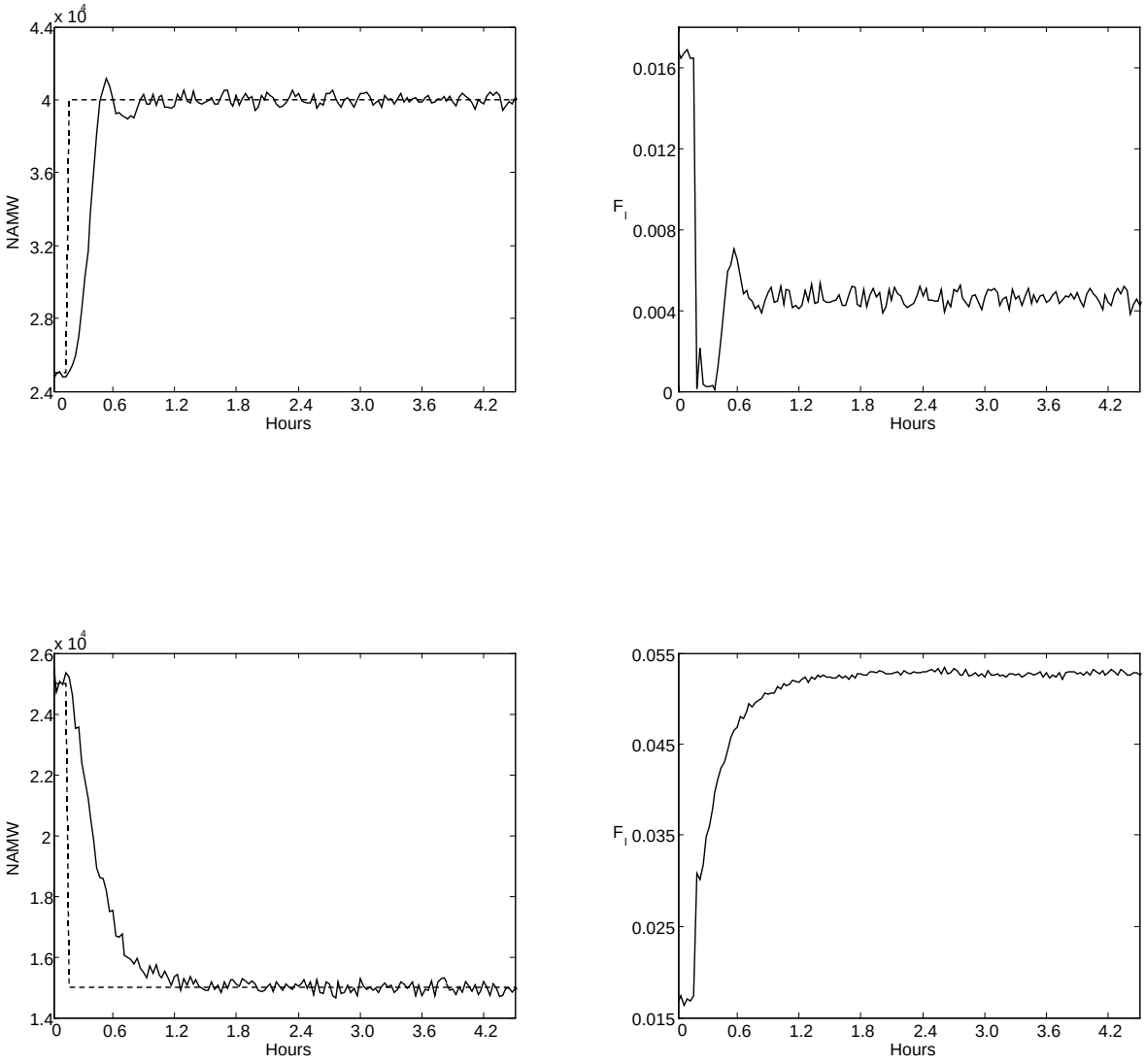


Figure 7.8 Servo responses in the presence of process noise

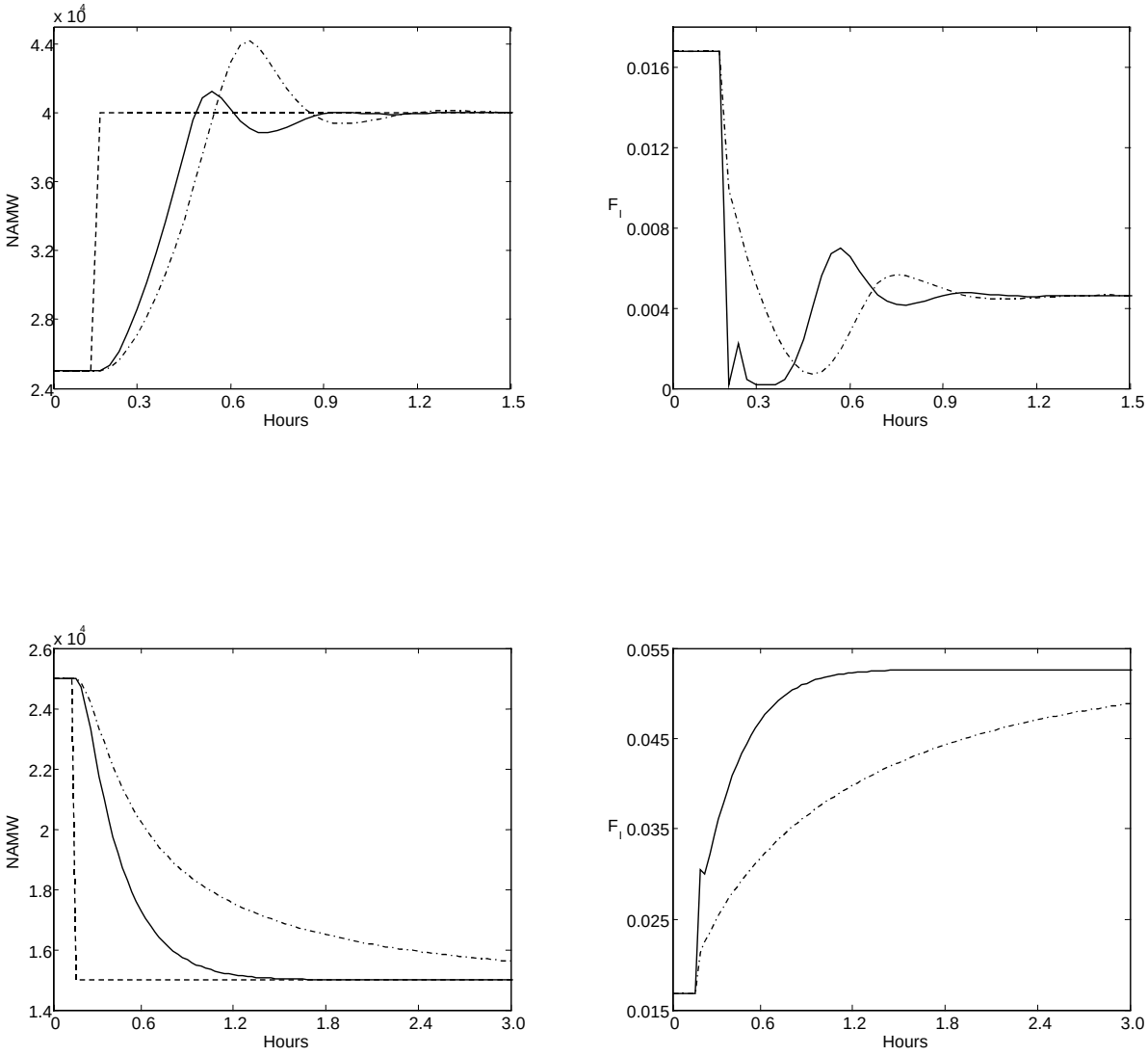


Figure 7.9 Comparison between the proposed design and IMC controller for set-point changes to 40000 kg/mol (top) and 15000 kg/kmol (bottom). Dashed: set-point; solid: the proposed method; dashed-dot: IMC

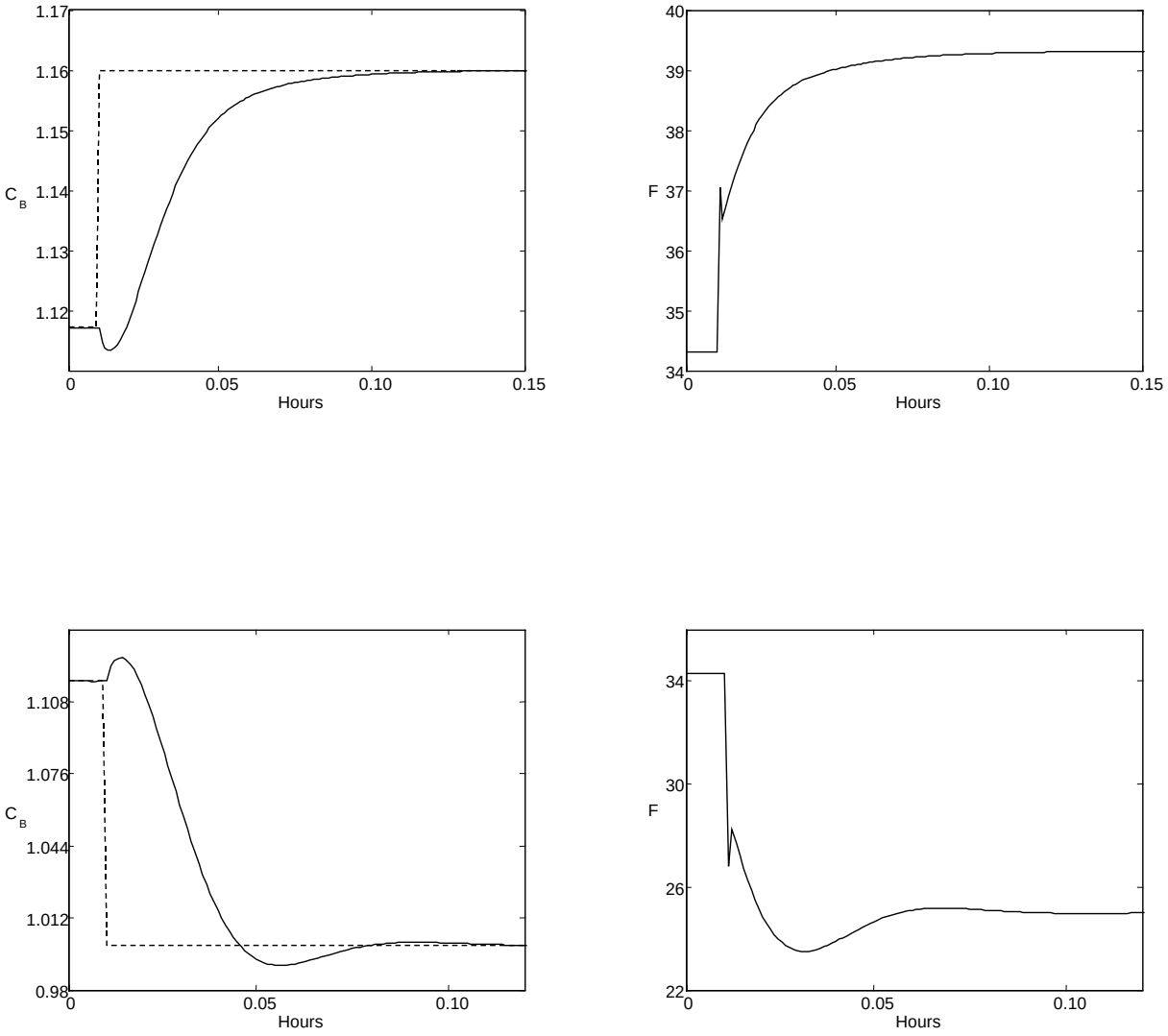


Figure 7.10 Servo responses of the PID controller around nominal operating condition. Dashed: set-point; solid: PID

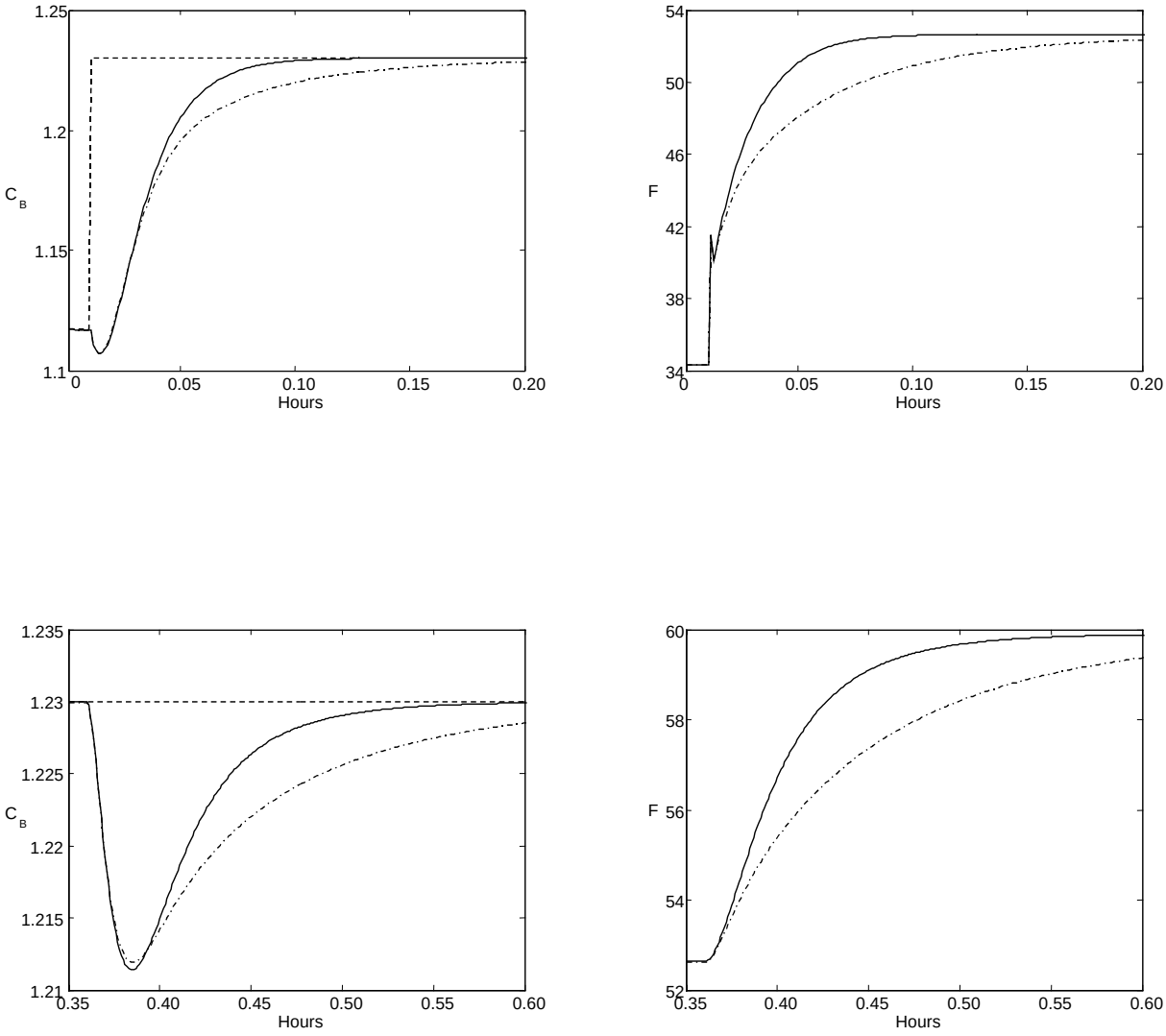


Figure 7.11 Closed-loop responses for 10% set-point change (top) and -2% step disturbance in C_{Af} (bottom). Dashed: set-point; solid: the proposed method; dashed-dot: PID

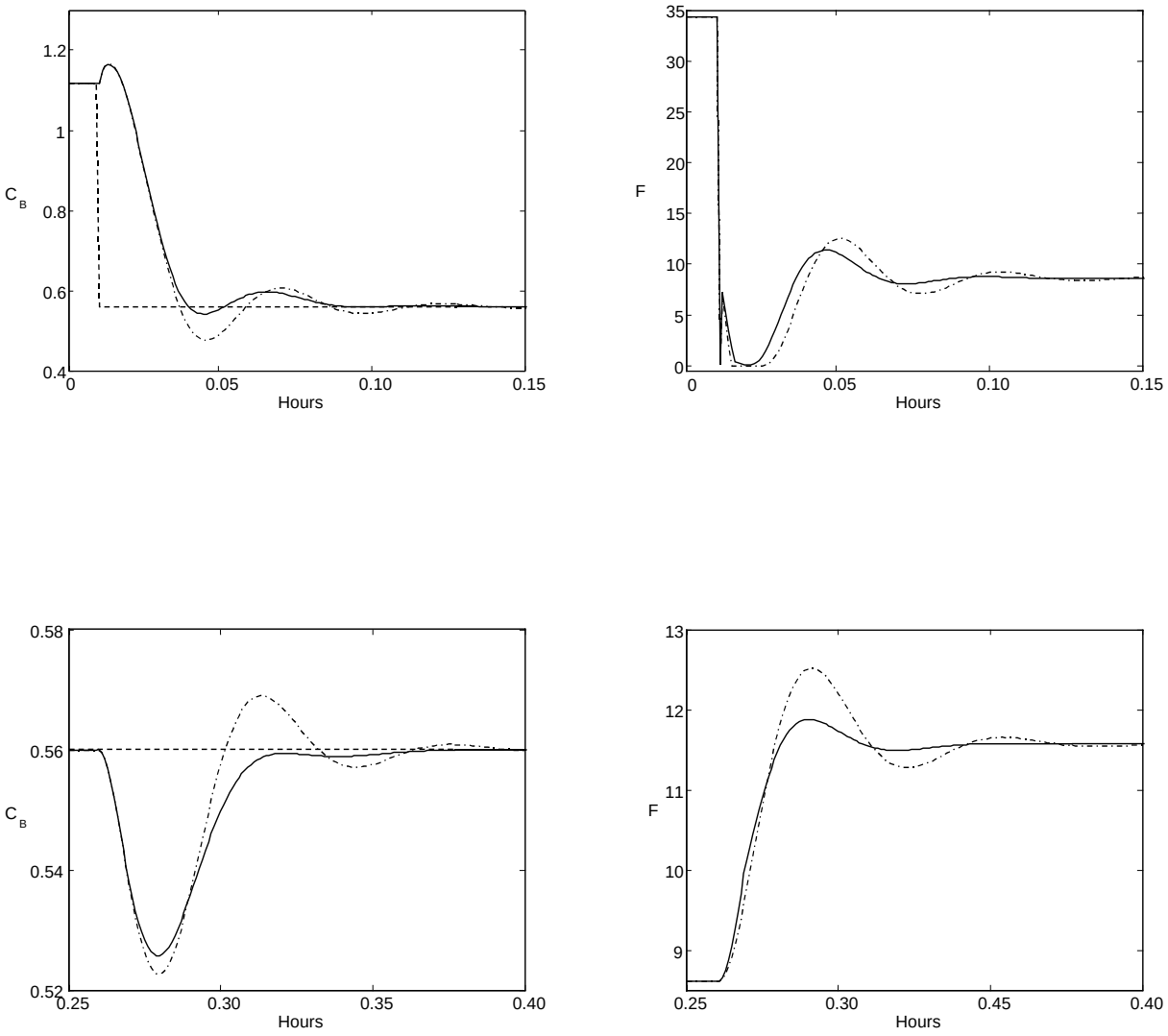


Figure 7.12 Closed-loop responses for -50% set-point change (top) and -20% step disturbance in C_{Af} (bottom). Dashed: set-point; solid: the proposed method; dashed-dot: PID

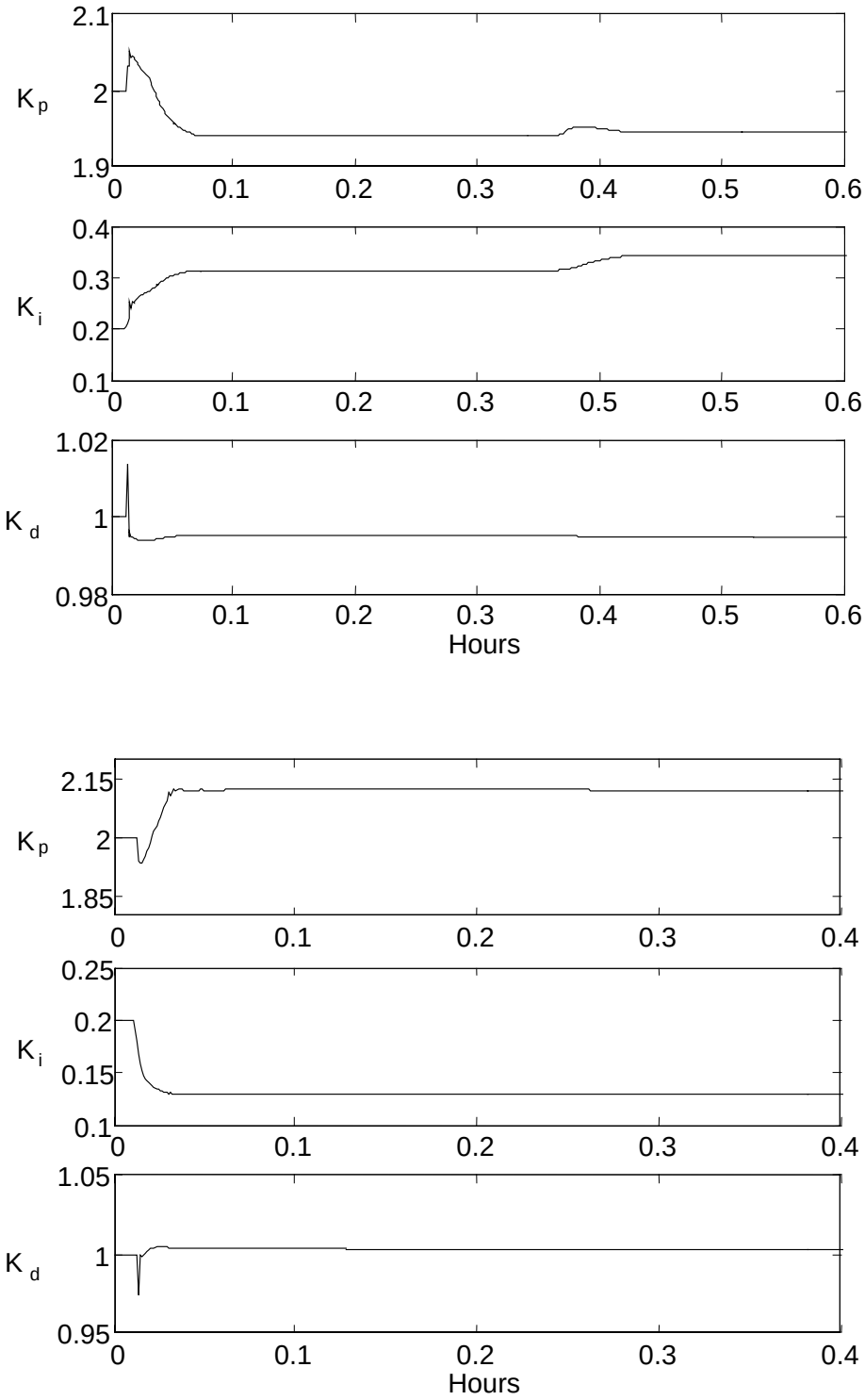


Figure 7.13 Updating of PID parameters for the closed-loop responses given in Figure 7.11 (top) and Figure 7.12 (bottom)

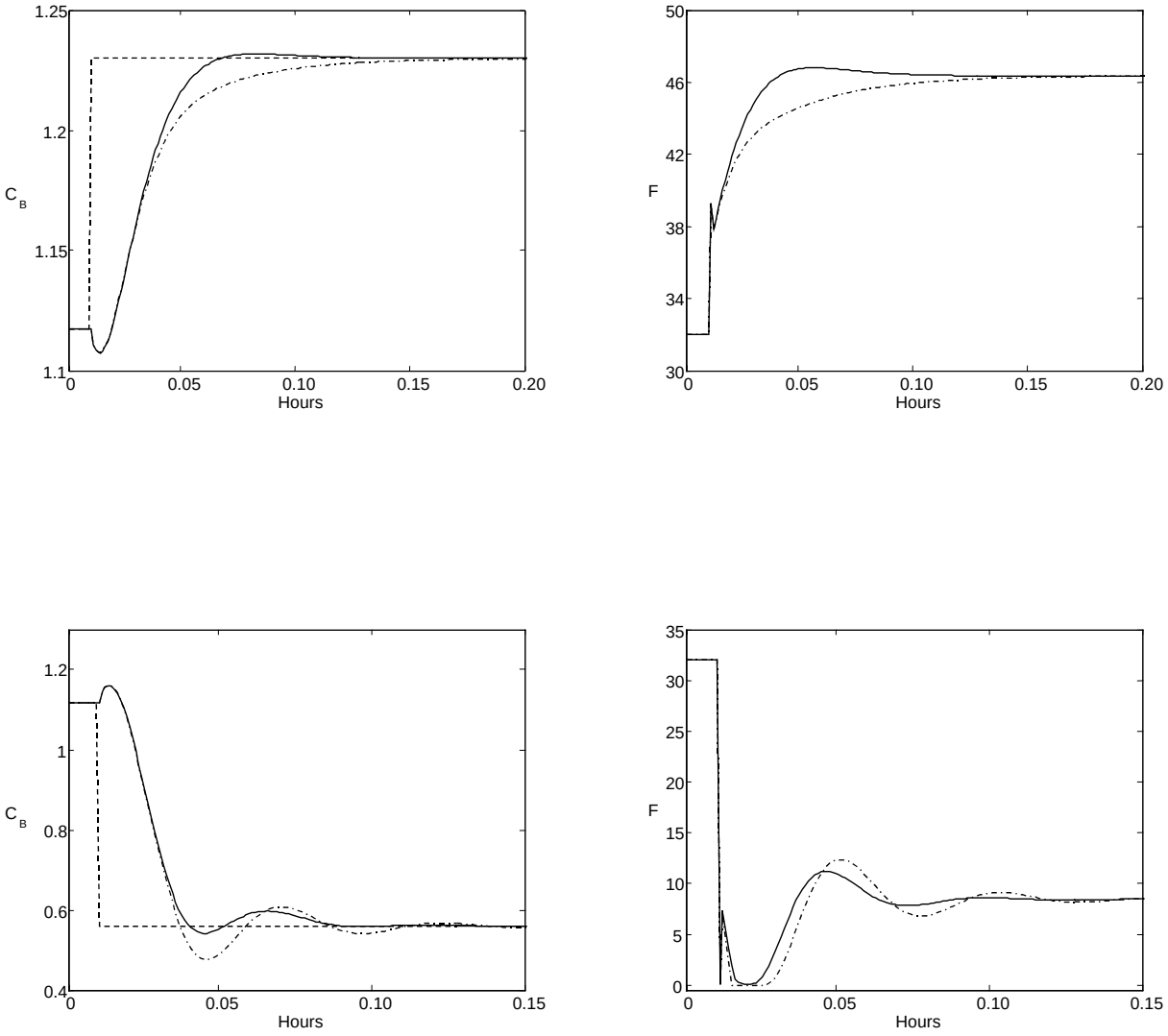


Figure 7.14 Closed-loop responses for 10% (top) and -50% (bottom) set-point changes under -10% modeling error in k_3. Dashed: set-point; solid: the proposed method; dashed-dot: PID

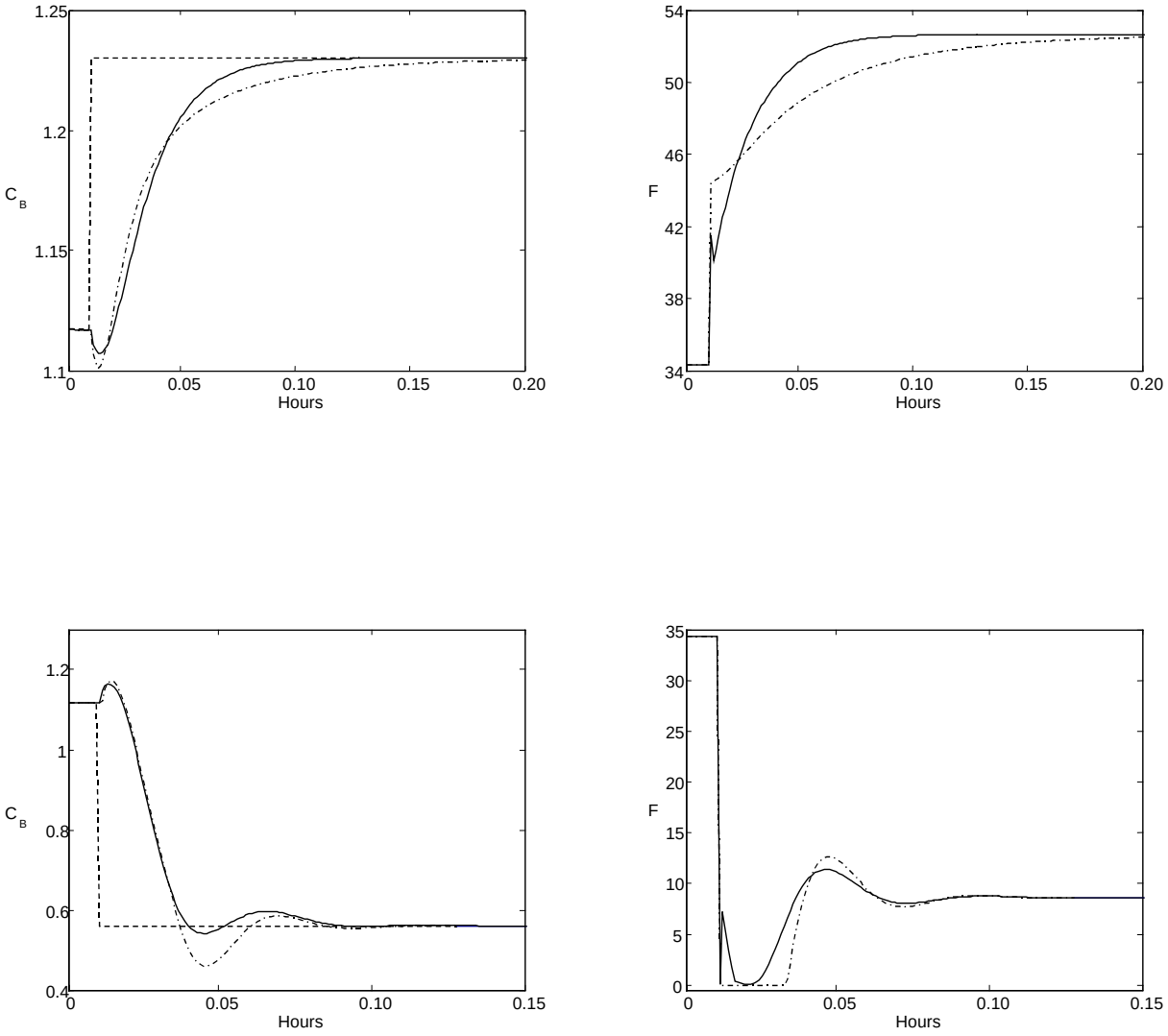


Figure 7.15 Comparison between the proposed design and IMC controller for 10% (top) and -50% (bottom) set-point changes. Dashed: set-point; solid: the proposed method; dashed-dot: IMC

Chapter 8

JITL-PCA Based Process Monitoring

In this chapter, a new method is proposed for monitoring nonlinear static or dynamic systems. In the proposed method, JITL and PCA are integrated to construct JITL-PCA monitoring scheme, where JITL servers as the process model to account for the nonlinear and dynamic behavior of the process under normal operating conditions. The residuals resulting from the difference between JITL's predicted outputs and process outputs are analyzed by PCA to evaluate the status of the current process operating conditions.

8.1 Introduction

Process monitoring is an important aspect of process engineering not only from plant's safety viewpoint, but also for the maintenance of yield and quality of process products. Therefore there is strong incentive to have efficient tools for process monitoring to ensure the success of the plant operations by recognizing anomalies of the behavior. Various methods have been developed for fault detection and diagnosis

in the past three decades, e.g. multivariate statistical methods, model-based methods, qualitative knowledge based methods, artificial intelligence, and various integrated methods (Yoon and MacGregor, 2000; Venkatasubramanian et al., 2003a, 2003b, 2003c) In this study, we will focus on multivariate statistical and model-based methods.

One of the popular multivariate statistical methods employed for fault detection is principal component analysis (PCA). From the review provided in Chapter 2, we know that conventional PCA is originally developed for the static system and therefore its application to the monitoring of nonlinear and dynamic systems is limited (Xu et al., 1992; Ku et al., 1995). To alleviate this drawback, various methods to extend linear static PCA for nonlinear and dynamic process monitoring are proposed, as discussed in Chapter 2. For model-based monitoring methods, Wachs and Lewin (1998) and Rotem et al. (2000) presented a model-based PCA approach, which was applied to the monitoring of an ethylene compressor with good result. In this approach, the nonlinearity and dynamics of process are accounted for by using known first-principle models, followed by the PCA analysis of the residuals, i.e. the difference between the actual process outputs and model's predicted outputs. However, the difficulty with this procedure is that a first-principle model may not be available or too costly to obtain. To circumvent this problem, Chen and Liao (2002) used neural networks as the process model in the model-based PCA. It is noted that, however, neural network suffers from the drawbacks of requiring a priori knowledge to determine the network structures and complicated training strategy to determine the optimal parameters of the network. Furthermore, neural networks are difficult to be updated online when the process operating conditions are changed.

In this chapter, we develop a new monitoring method called JITL-PCA that integrates JITL and PCA for nonlinear static or dynamic process monitoring. In the JITL-PCA framework, JITL is used as the process models to remove the nonlinear or dynamic information from the raw process data, while the resulting residuals between the actual process outputs and JITL's predicted outputs are used in the PCA analysis to draw the monitoring charts. Two nonlinear examples are used to illustrate the utility of JITL-PCA in monitoring the nonlinear systems and a comparison with the conventional PCA and dynamic PCA is made.

8.2 PCA and Model-Based PCA

The principle of PCA is to find combinations of variables that capture the largest amount of information in a data set. Let N scaled observations $\mathbf{x}_i \in R^{n\times 1}$ generate a data matrix $\mathbf{X} \in R^{N\times n}$ whose i-th row vector is $\mathbf{x}_i^T$. The covariance matrix of $\mathbf{X}$ is defined as:

$$\Sigma = \frac{\mathbf{X}^T\mathbf{X}}{N-1} \tag{8.1}$$

Denote λ_j ($j = 1,\ldots,n$) the eigenvalues of the matrix Σ that are arranged in descending order to determine the principal components (PCs), and their corresponding eigenvectors are the principal component loadings, $\mathbf{p}_j$. If the first k PCs are selected, the prediction of the PCA model for a new observation data $\mathbf{x}_{new}$ is given by

$$\hat{\mathbf{x}}_{new} = \mathbf{P}_k\mathbf{t} \tag{8.2}$$

where $\mathbf{P}_k = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_k]$ and $\mathbf{t} = \mathbf{P}_k{}^T\mathbf{x}_{new}$ is the score vector. The resulting residual is defined as:

$$\mathbf{r} = \mathbf{x}_{new} - \hat{\mathbf{x}}_{new} = (\mathbf{I} - \mathbf{P}_k \mathbf{P}_k^T)\mathbf{x}_{new} \tag{8.3}$$

Two statistical variables T^2 and Q can be calculated from **t** and **r** respectively (Jackson and Mudholkar, 1979; Kresta et al., 1991; MacGregor and Kourti, 1995):

$$T^2 = \mathbf{t}^T \Lambda_k^{-1} \mathbf{t} \tag{8.4}$$

$$Q = \mathbf{r}^T \mathbf{r} \tag{8.5}$$

where Λ_k is a diagonal matrix constructed by the first k eingenvalues of Σ. The control limit of T^2 is calculated by:

$$T_{\lim}^2 = \frac{k(N-1)}{N-k} F(k, N-1, \alpha) \tag{8.6}$$

where $F(k, N-1, \alpha)$ is a F distribution with k and $N-1$ degrees of freedom with significance level α (MacGregor and Kourti, 1995), and the control limit of Q is obtained by (Jackson and Mudholkar, 1979):

$$Q_{\lim} = \theta_1 [\frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2}]^{1/h_0} \tag{8.7}$$

where $\theta_i = \sum_{j=k+1}^{n} \lambda_j^i$ for $i = 1, 2, 3$, and $h_0 = 1 - \frac{2\theta_1 \theta_3}{3\theta_2^2}$, c_α is the normal deviate corresponding to the upper $(1-\alpha)$ percentile. For a new observation $\mathbf{x}_{new}$, if its associated T^2 and Q are smaller than their respective control limits $T_{\lim}^2$ and $Q_{\lim}$, the system can be considered working under the normal condition with $100(1-\alpha)\%$ confidence. Otherwise, some faults may occur in the system.

Algebraically, PCs are linear combinations of independent random variable, and geometrically PCs represent a new coordinate system obtained by rotating the original coordinate. Therefore, PCA is based on linear and static assumption and it

may not extract adequate information for nonlinear or dynamic systems. To alleviate this shortcoming, model-based PCA as depicted in Figure 8.1 uses a nominal process model to account for the process nonlinear or dynamic behavior under normal operating condition. If the model is accurate enough, the residual between actual process and model will be relatively insensitive to the variations resulting from the nonlinearity or dynamics of the normal process (Rotem et al., 2000) As a result, PCA is more sensitive to detect the process variation caused by the process faults. In the next section, a new model-based PCA method by incorporating the JITL will be developed.

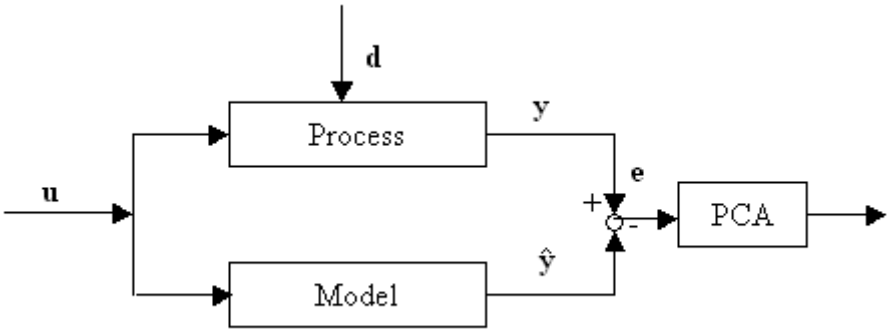


Figure 8.1 Model-based PCA monitoring scheme

8.3 JITL-PCA for Process Monitoring

Although JITL with ARX model has been shown to be efficient in modeling nonlinear systems, it is not suitable to be employed in the proposed JITL-PCA monitoring framework due to its lack of proper sensitivity to the process faults, which will be discussed in more detail in the ensuing discussion. Instead, it is more desirable to employ the finite impulse response (FIR) model as the local model in the proposed

JITL-PCA monitoring scheme to achieve better monitoring accuracy. With a FIR local model, the *j*-th predicted output $\hat{y}_j(k)$ for a multivariable system with m_1 inputs and m_2 outputs is given by:

$$\hat{y}_j(k) = \mathbf{z}_j(k-1)^T \Psi_j \tag{8.8}$$

where the regression vector $\mathbf{z}_j(k-1)$ and model parameter vector Ψ_j are defined as:

$$\mathbf{z}_j(k-1) = [u_1(k-1),\ \ldots,\ u_1(k-n_{j,u\_1}),\ \ldots,\ u_{m_1}(k-1),\ \ldots,\ u_{m_1}(k-n_{j,u\_m1}),\ 1]^T \tag{8.9}$$

$$\Psi_j = [\psi_{j,1},\ \ldots,\ \psi_{j,n_{u\_1}},\ \ldots,\ \psi_{j,n_{u\_1}+\ldots+n_{u\_(m1-1)}+1},\ \ldots,\ \psi_{j,n_{u\_1}+\ldots+n_{u\_m1}},\ b_j]^T \tag{8.10}$$

For monitoring of the static systems, low order polynomial models can be chosen as the local model for JITL:

$$\hat{y}_j = \mathbf{z}_j^T \Psi_j \tag{8.11}$$

where

$$\mathbf{z}_j = [u_1,\ \ldots,\ u_1^{n_{j,1}},\ \ldots,\ u_{m_1},\ \ldots,\ u_{m_1}^{n_{j,m1}},\ 1]^T \tag{8.12}$$

$$\Psi_j = [\psi_{j,1},\ \ldots,\ \psi_{j,n_{j,1}},\ldots,\ \psi_{j,n_{j,1}+\ldots+n_{j,m1-1}+1},\ \ldots,\ \psi_{j,n_{j,1}+\ldots+n_{j,m1}},\ b_j]^T \tag{8.13}$$

Having discussed the local model structure to be employed in the proposed JITL-PCA monitoring scheme, we are in the position to discuss JITL-PCA monitoring scheme for multivariable static or dynamic systems, as shown in Figure 8.2, where each JITL is used as the process model to predict each process output. The residuals between the process outputs and predicted outputs are calculated and subsequently used to build the PCA model.

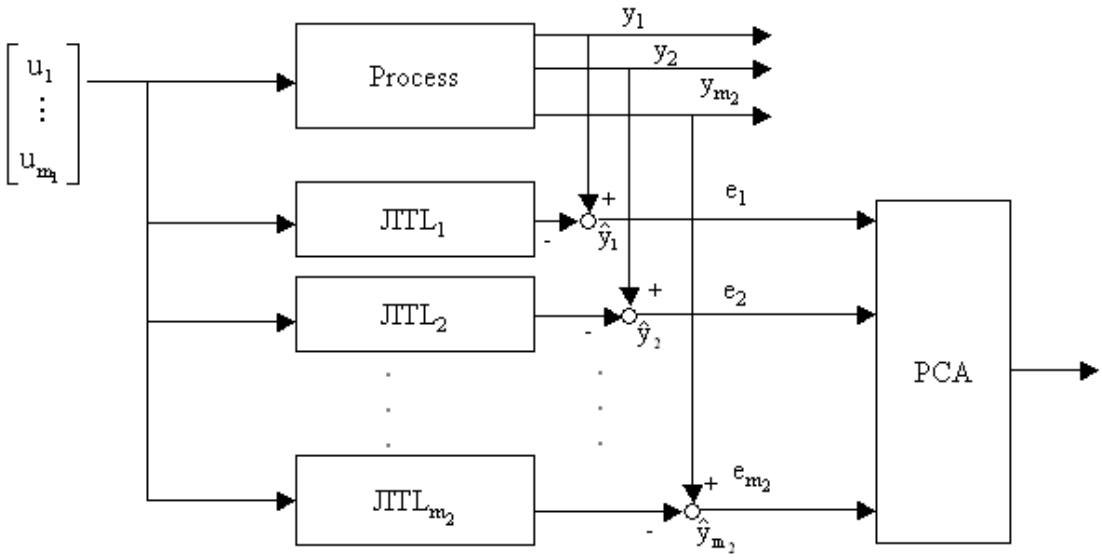


Figure 8.2 JITL-PCA monitoring scheme

The following summarizes the construction procedure of JITL-PCA and its application to process monitoring:

(1) Process data under normal condition are collected to generate the database for JITL modeling purpose;

(2) Another set of N process data under normal condition is generated, where the measured process outputs are used to form the output matrix $\mathbf{Y}$. By using the appropriate regression vector, Eq. (8.10) or (8.13), the JITL algorithm is applied to obtain the model outputs and the predicted output matrix $\hat{\mathbf{Y}}$ is constructed;

(3) The residual matrix $\mathbf{E} = \mathbf{Y} - \hat{\mathbf{Y}}$ is calculated and the number (k) of PCs is determined by the aforementioned PCA procedure. The resulting PCA model is then integrated with JITL to form the JITL-PCA monitoring scheme;

(4) When a new observation comes during the monitoring phase, the scaled residual $\tilde{\mathbf{e}}_{new}$ between the actual process outputs and JITL predicted outputs are calculated and the statistic variables T^2 and Q are obtained by:

$$T^2 = \tilde{\mathbf{e}}_{new}^T \mathbf{P}_k \Lambda_k^{-1} \mathbf{P}_k^T \tilde{\mathbf{e}}_{new} \tag{8.14}$$

$$Q = \tilde{\mathbf{e}}_{new}^T (I - \mathbf{P}_k \mathbf{P}_k^T) \tilde{\mathbf{e}}_{new} \tag{8.15}$$

If T^2 and Q are below their respective control limits, Eqs. (8.6) and (8.7), the system is considered to be under normal operating condition; otherwise some process faults may have occurred.

Based on the on-going discussion, it is evident that the effectiveness of the JITL-PCA monitoring scheme hinges on the sensitivity of JITL algorithm to the different types of process faults. For example, if process fault only affects the process outputs, e.g. process parameter drift or process output sensor bias, it is essential that JITL prediction should be insensitive to the values of process outputs as much as it can. Therefore, if one applies JITL with ARX model structure, which makes use of past values of both process outputs and process inputs, JITL will attempt to mimic the faulty process outputs and hence the resulting residuals cannot truly reflect the influence of process fault on the process output. As a result, the monitoring capability of JITL-PCA is compromised. In contrast, for JITL with FIR model structure, where only the past values of process inputs are employed, the predicted outputs are not influenced by the occurrence of the aforementioned process fault. Consequently, the resulting residuals are able to reflect closely the deviation of process outputs from those obtained in the absence of process fault, resulting in better monitoring capability of the JITL-PCA. On the other hand, if process fault only affects the process inputs,

e.g. input sensor bias, it is crucial that JITL prediction is sensitive to the process fault so that noticeable difference between the predicted outputs and process outputs can be observed. Again, FIR model is preferred over ARX model to be incorporated by JITL-PCA due to the reason previously discussed, i.e. distinct monitoring characteristics caused by the different regression vectors employed in these two model structures. In the next section, an example will be used to illustrate the different fault sensitivity between FIR and ARX models.

8.4 Examples

Example 1 Considering a nonlinear static system as given by (Dong and McAvoy, 1996):

$$\begin{aligned} y_1 &= u + \varepsilon_1 \\ y_2 &= u^2 - 3u + \varepsilon_2 \\ y_3 &= -u^3 + 3u^2 + \varepsilon_3 \end{aligned} \tag{8.16}$$

where u is the system input, y_1, y_2, y_3 are system outputs, $\varepsilon_1, \varepsilon_2, \varepsilon_3$ are independent random noise $N(0, 0.01)$, and system's operating space is $u \in [0.01\ 1]$. To demonstrate the monitoring capability of JITL-PCA, two system faults are considered.

The first fault assumes a small deviation occurred in y_2 as described by:

$$\begin{aligned} y_1 &= u + \varepsilon_1 \\ y_2 &= 1.1u^2 - 3u + \varepsilon_2 \\ y_3 &= -u^3 + 3u^2 + \varepsilon_3 \end{aligned} \tag{8.17}$$

The following equations describe the second fault:

$$y_1 = u + \varepsilon_1$$

$$y_2 = u^2 - 3u + \varepsilon_2 \tag{8.18}$$

$$y_3 = -1.1u^3 + 3u^2 + \varepsilon_3$$

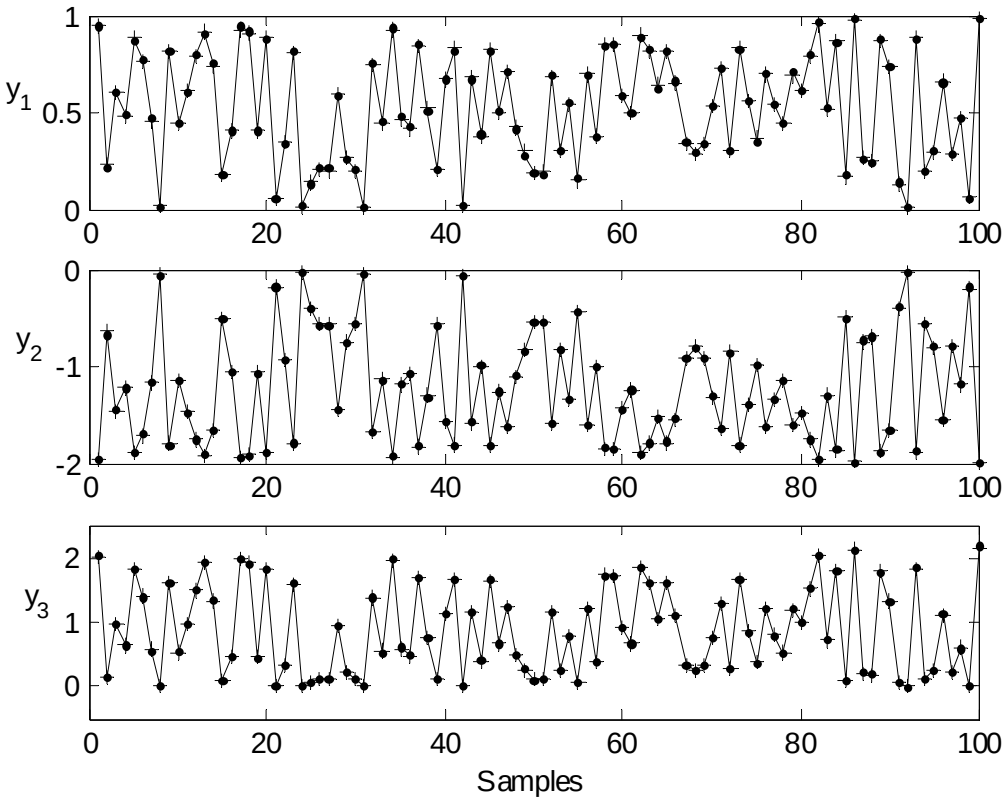


Figure 8.3 Modeling result of JITL. (•): actual output; (+): model output

To proceed with the JITL algorithm, one hundred process data under normal condition, i.e. Eq. (8.16), are generated to construct the databases and $k_{\min} = 6$ and $k_{\max} = 30$ are chosen. In addition, a first-order polynomial model, i.e. the regression vector $\mathbf{z}_1 = \mathbf{z}_2 = \mathbf{z}_3 = u$, is employed as the local model to predict the three outputs. To build the PCA model, another one hundred normal process data as illustrated in Figure 8.3 are generated and used to calculate the residuals between the actual system outputs and JITL's predicted outputs, which are subsequently used to construct the PCA model. For this new process data, which is not included in the databases of JITL,

JITL is able to predict three outputs with excellent accuracy, as shown in Figure 8.3. For this process, the variances captured by the three PCs are 1.17, 0.99, and 0.84 respectively, therefore three PCs are selected to build the PCA model used in JITL-PCA.

The monitoring result of the proposed method for the first fault is illustrated in Figure 8.4(a), where the first one hundred samples are the normal operating data and the next one hundred samples are the faulty data as simulated by Eq. (8.17). The dashed line is the 95% control limit and the solid line is the 99% control limit. As can be seen, for the normal samples, the T^2 values of three samples exceed the 95% control limit and none exceeds the 99% control limit, whereas for the faulty data, the T^2 values of 52 samples exceed the 95% control limit and 45 samples exceed the 99% control limit. This indicates that after one hundred samples something had changed in the systems or/and the occurrence of the unmeasured disturbances. For comparison purpose, Figure 8.4(b) shows the T^2 plot of the PCA analysis with three PCs for the identical set of process data used in the JITL-PCA. For the normal samples, the T^2 values of five samples exceed the 95% control limit and none exceeds the 99% control limit, while for the faulty data, only 14 samples' T^2 values exceed the 95% control limit and four samples exceed the 99% control limit.

Figure 8.5 compares the monitoring result of JITL-PCA and PCA for the second fault. For JITL-PCA, 44 samples exceed the 95% control limit and 35 samples exceed the 99% control limit under the faulty condition. In contrast, PCA analysis reveals that nine samples exceed 95% control limit and only one sample exceeds 99% control limit, indicating that PCA fails to detect the second fault. As clearly shown from both Figures 8.4 and 8.5 that the proposed method is more effective than PCA in monitoring the nonlinear system.

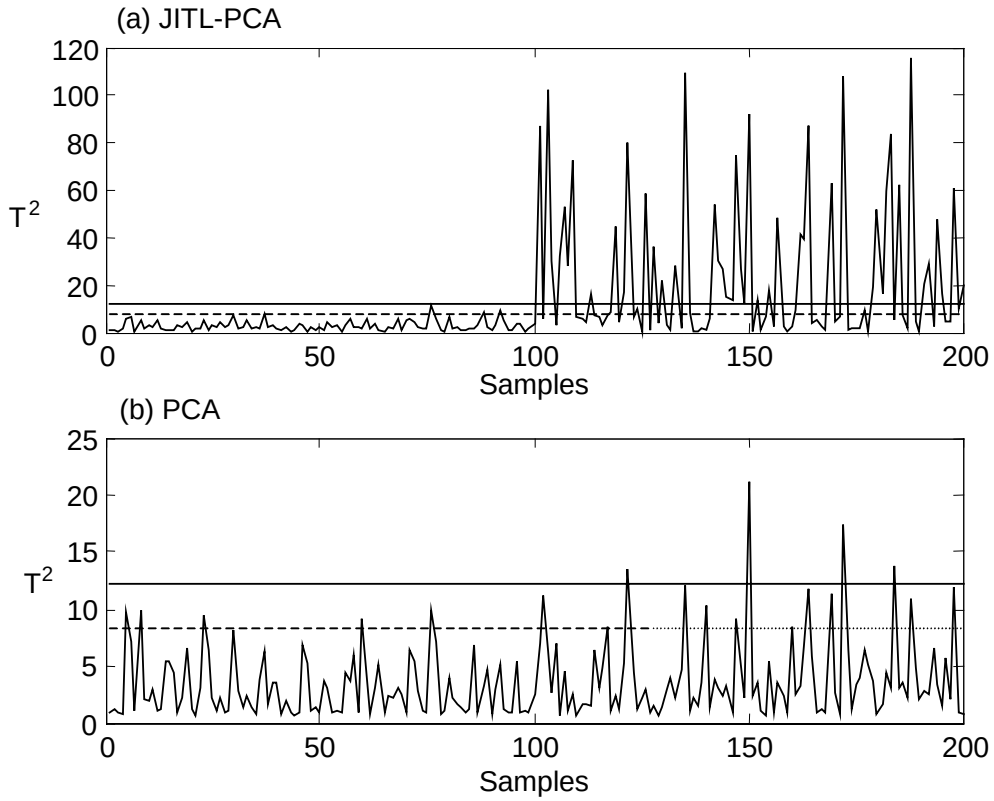


Figure 8.4 Monitoring result of the fault 1: (a) JITL-PCA; (b) PCA

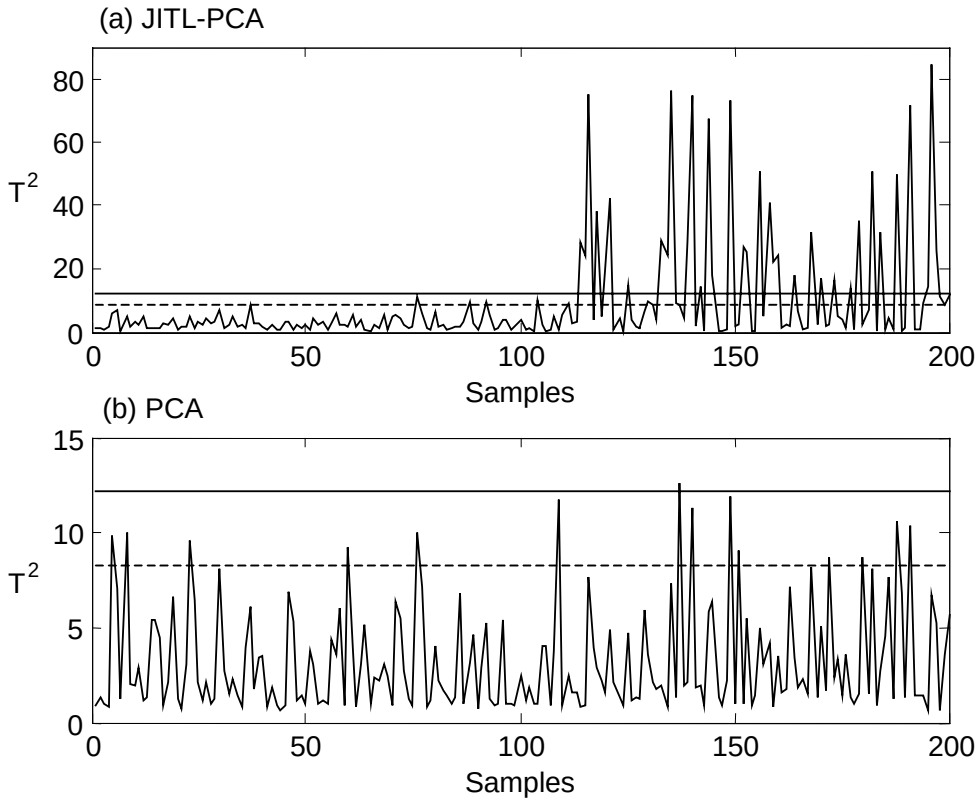


Figure 8.5 Monitoring result of the fault 2: (a) JITL-PCA; (b) PCA

In general, it is not a trivial task for neural network based modeling methods to update its model online. For example, when the system operating space is changed to the new operating space, neural network requires model update from scratch. In contrast, JITL is inherently adaptive by simply adding the current system data online to the database. For illustration purpose, assume that the system's operating space shifts from $u \in [0.01\ 1]$ to $u \in [0.7\ \ 1.2]$, while the system nonlinearity under normal condition and faulty conditions, i.e. Eqs. (8.16) to (8.18), remain unchanged. To learn the new process nonlinearity, the original database of JITL is augmented by the addition of new process data obtained in the new operating space $u \in [0.7\ \ 1.2]$ at each sampling time.

To illustrate this point, Figure 8.6(a) shows that the initial T^2 values of some data samples exceed the control limit, indicating that the database still closely resembles the original operating space $u \in [0.01\ 1]$. However, the T^2 value eventually falls below the control limit, showing that the database of JITL has been updated with sufficient number of new process data and hence is capable of modeling the system dynamics in the new operating space. Figure 8.6 (parts (b)-(d)) and Table 8.1 show the monitoring results of JITL-PCA with this new database for three scenarios, i.e. normal condition and two faulty conditions under new operating space. Evidently, JIT-PCA can monitor the system in the new operating space effectively.

Table 8.1 Summary of JITL-PCA monitoring result in the new operating space

	Normal	Fault 1	Fault 2
95%	4	92	75
99%	0	88	66

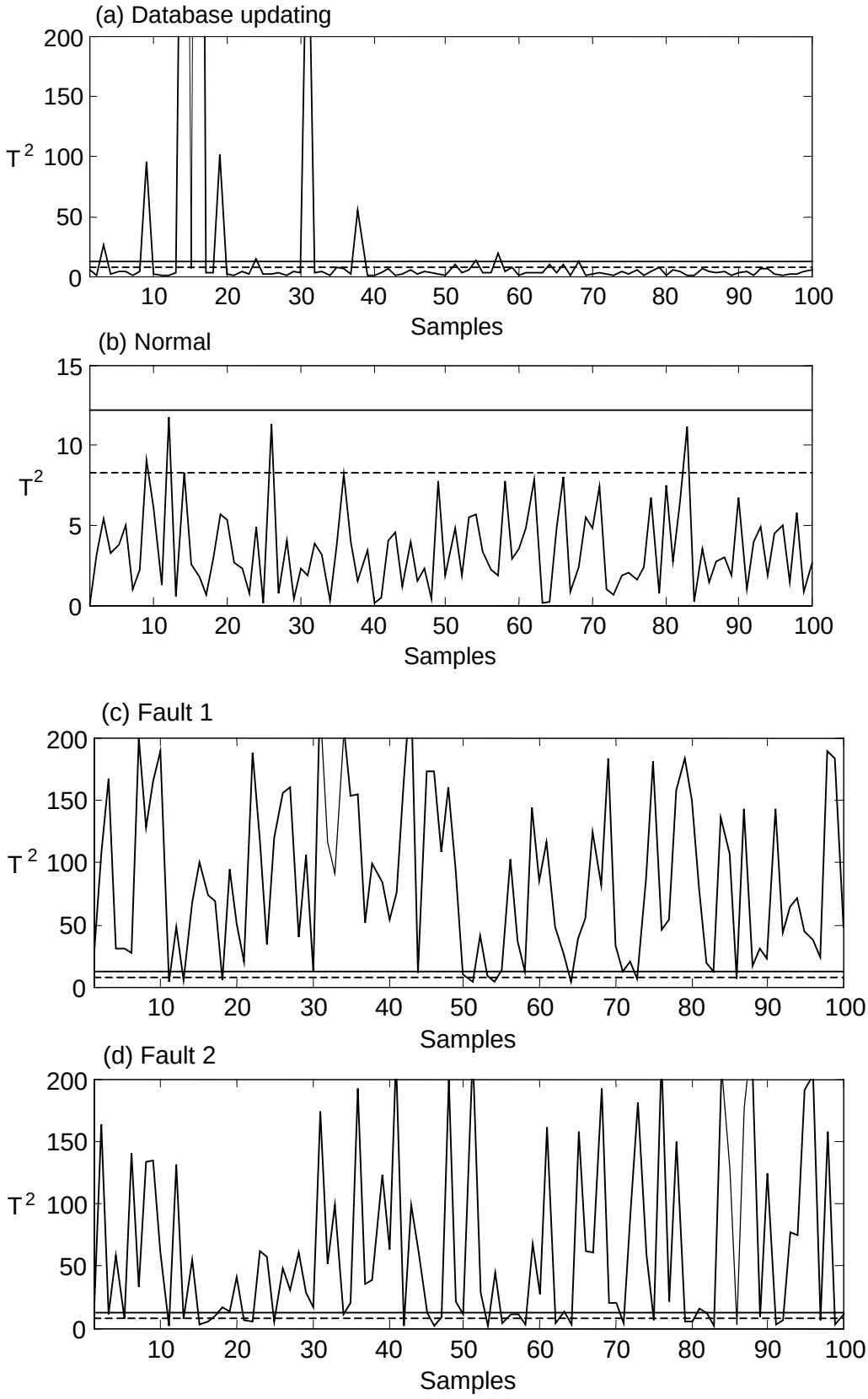


Figure 8.6 Monitoring result of JITL-PCA in the new operating space

Example 2 Two continuous stirred tank reactors (CSTRs) in series with an intermediate feed as depicted in Figure 8.7 are used to demonstrate the application of the proposed monitoring technique for nonlinear dynamic process. The reaction system involves an exothermic and irreversible reaction $A \rightarrow B$ in both reactors. In addition, an undesired side-reaction $B \rightarrow C$ takes place in the two reactors. Pure component A is fed to the first reactor and the mixer.

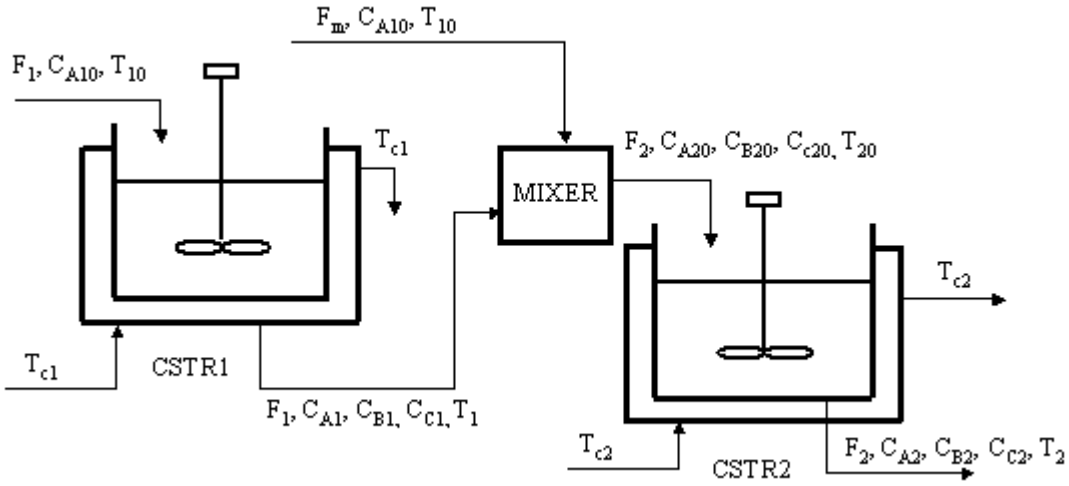


Figure 8.7 Two CSTRs in series with an intermediate feed

The process can be described by the following equations (Loeblain and Perkins, 1998):

$$\frac{dC_{A1}}{dt} = \frac{F_1}{V_1}(C_{A10} - C_{A1}) - k_{A1}C_{A1}$$

$$\frac{dC_{B1}}{dt} = -\frac{F_1}{V_1}C_{B1} + k_{A1}C_{A1} - k_{B1}C_{B1}$$

$$\frac{dC_{C1}}{dt} = -\frac{F_1}{V_1}C_{C1} + k_{B1}C_{B1}$$

$$\frac{dT_1}{dt} = \frac{F_1}{V_1}(T_{10} - T_1) + \frac{-\Delta H_A}{c_p \rho} k_{A1} C_{A1} + \frac{-\Delta H_B}{c_p \rho} k_{B1} C_{B1} - \frac{U_1}{c_P \rho V_1}(T_1 - T_{c1})$$

$$\frac{dC_{A2}}{dt} = \frac{F_2}{V_2}(C_{A20} - C_{A2}) - k_{A2} C_{A2} \tag{8.19}$$

$$\frac{dC_{B2}}{dt} = \frac{F_2}{V_2}(C_{B20} - C_{B2}) + k_{A2} C_{A2} - k_{B2} C_{B2}$$

$$\frac{dC_{C2}}{dt} = \frac{F_2}{V_2}(C_{C20} - C_{C2}) + k_{B2} C_{B2}$$

$$\frac{dT_2}{dt} = \frac{F_2}{V_2}(T_{20} - T_2) + \frac{-\Delta H_A}{c_p \rho} k_{A2} C_{A2} + \frac{-\Delta H_B}{c_p \rho} k_{B2} C_{B2} - \frac{U_2}{c_P \rho V_2}(T_2 - T_{c2})$$

$$k_{Ai} = k_{A0} \exp(-\frac{E_A}{RT_i}),\ k_{Bi} = k_{B0} \exp(-\frac{E_B}{RT_i}),\ i = 1, 2$$

The process parameters are given in Table 8.2 and the range of normal operating condition is: $F_1 \in [0.28\ \ 0.42]$ (m^3/min), $T_{10} \in [290\ \ 310]$ (K), $F_m \in [0.16\ \ 0.24]$ (m^3/min). The inlet concentration to the first tank, $C_{A10} = 20 + e$ (kmol/m^3), where e is the white noise with 0.2 variance, is considered as the unmeasured system noise. Additionally, measurement noises are assumed to be 1% of the respective measurement ranges for all measured variables. The sampling time is four minutes. To test the monitoring capability of the proposed method for nonlinear dynamic system, ten process faults including process parameter faults and the input/output sensor faults as summarized in Table 8.3 are considered.

Table 8.2 Parameters of example 2

Variable	Value	Variable	Value
k_{A0}	2.7×10^{8}/min	V_1	5 m^3
k_{B0}	160/min	V_2	5 m^3
E_A / R	6000 K	T_{c1}	300 K
E_B / R	4500 K	T_{c2}	300 K
$\Delta H_A /(c_p\rho)$	-5 m^3 K/kmol	$U_1 /(c_p\rho V_1)$	0.35 /min
$\Delta H_B /(c_p\rho)$	-5 m^3 K/kmol	$U_2 /(c_p\rho V_2)$	0.35 /min

Table 8.3 Fault description for example 2

Fault	Description
Fault 1	5% decrease in the heat transfer coefficient of CSTR1
Fault 2	5% decrease in the heat transfer coefficient of CSTR2
Fault 3	5% decrease in the kinetic parameter k_{A0} of CSTR1
Fault 4	5% decrease in the kinetic parameter k_{A0} of CSTR2
Fault 5	± 0.01 m^3/min measurement bias of F_1
Fault 6	± 2 K measurement bias of T_{10}
Fault 7	± 0.01 kmol/ m^3 measurement bias of C_{A1}
Fault 8	± 2 K measurement bias of T_1
Fault 9	± 0.01 kmol/ m^3 measurement bias of C_{A2}
Fault 10	± 2 K measurement bias of T_2

As mentioned previously, different model structure employed by JITL has distinct sensitivity to the process faults. To illustrate this point, both ARX and FIR models are used to predict the output T_1 in the presence of Fault 1. To do so, the following regression vectors are chosen for ARX and FIR models respectively:

$$\hat{T}_1(ARX):\mathbf{z}(k-1)=[T_1(k-1),\quad T_1(k-2),\quad F_1(k-1),\quad T_{10}(k-1)] \tag{8.20}$$

$$\hat{T}_1(FIR):\mathbf{z}(k-1)=[F_1(k-1),\cdots,F_1(k-10),T_{10}(k-1),\ldots,T_{10}(k-10),1] \qquad (8.21)$$

As clearly shown in Figure 8.8, the residual between the process output and the predicted output by the JITL with FIR model is larger than that obtained from the JITL with ARX model, indicating that FIR model is more sensitive to the fault than the ARX model. This observation confirms the previous discussion given in the last section. Because ARX model will degrade the effectiveness of the JITL-PCA monitoring scheme, FIR model is used in the subsequent JITL-PCA monitoring scheme.

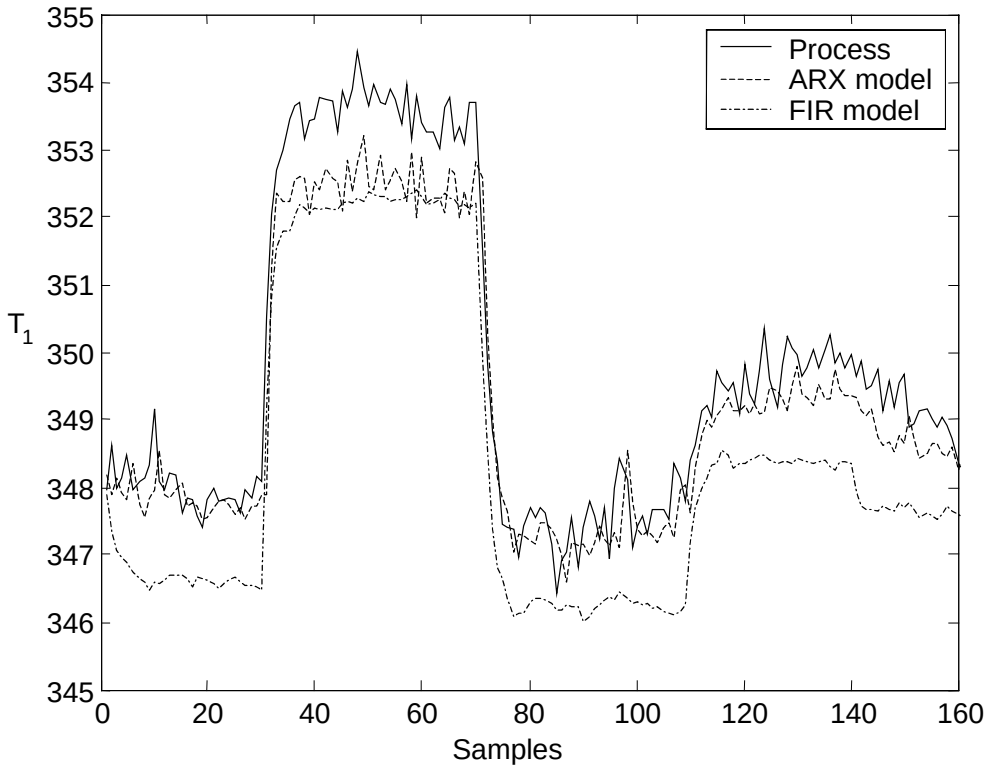


Figure 8.8 Comparison between FIR model and ARX model under the fault 1

To do so, four process outputs T_1, C_{A1} , C_{A2}, and T_2, are predicted by JITL using the regression vectors Eqs. (8.21) to (8.24):

$$\hat{C}_{A1} : \mathbf{z}(k-1) = [F_1(k-1), \cdots, F_1(k-10), T_{10}(k-1), \ldots, T_{10}(k-10), 1] \tag{8.22}$$

$$\hat{C}_{A2} : \mathbf{z}(k-1) = [F_2(k-1), \cdots, F_2(k-10), \hat{T}_{20}(k-1), \ldots, \hat{T}_{20}(k-10), 1] \tag{8.23}$$

$$\hat{T}_2 : \mathbf{z}(k-1) = [F_2(k-1), \cdots, F_2(k-10), \hat{T}_{20}(k-1), \ldots, \hat{T}_{20}(k-10), 1] \tag{8.24}$$

where $\hat{T}_{20}$ is the predicted inlet temperature of the second tank, $\hat{T}_{20} = (\hat{T}_1 F_1 + T_{10} F_m)/F_2$. Three thousand process data under normal conditions are used to build the database for JITL modeling purpose. In addition, the parameters $\gamma = 0.7$, $k_{\min} = 60$ and $k_{\max} = 120$ are chosen for four predicted outputs. Figure 8.9 shows that JITL with FIR model has good modeling accuracy in the validation test. Another one thousand process data are collected and the resulting residuals are used to build the PCA model. Because the variances captured by the four PCs are 1.87, 0.93, 0.6 and 0.59, respectively, four PCs are selected to build the PCA model for JITL-PCA.

For comparison purpose, dynamic PCA (DPCA) (Ku et al., 1995) is applied by using 27 measured variables, $F_1(k-i)$, $T_{10}(k-i)$, $F_2(k-i)$, $T_{20}(k-i)$, $C_{A1}(k-i)$, $T_1(k-i)$, $C_{A2}(k-i)$, $T_2(k-i)$, and $F_m(k-i)$ ($i = 1 \sim 3$) and the same process data used in constructing the JITL-PCA scheme. Eight PCs, which explain 99% variance, are selected to build the DPCA model. Table 8.4 compares the monitoring performances of JITL-PCA and DPCA. Figures 8.10 to 8.19 also illustrate the monitoring results where the first one hundred samples are the normal operating data, followed by one hundred faulty data. It is evident that JITL-PCA outperforms DPCA for monitoring the nonlinear dynamic systems.

Table 8.4 Monitoring results of JITL-PCA and DPCA

	JITL-PCA		DPCA			
	T^2		Q		T^2	
	95%	99%	95%	99%	95%	99%
Fault 1	99	94	35	21	6	6
Fault 2	81	62	39	17	11	5
Fault 3	96	78	9	6	2	2
Fault 4	78	63	17	7	5	4
Fault 5	73	44	15	7	6	3
Fault 6	94	82	10	8	10	5
Fault 7	100	99	89	68	6	5
Fault 8	100	100	100	99	3	3
Fault 9	99	95	99	84	17	6
Fault 10	100	99	89	54	54	18

8.5 Conclusion

A new model-based monitoring scheme, JITL-PCA, is proposed for monitoring the nonlinear static or dynamic systems. In this framework, JITL is used to model the nonlinear and dynamic information of the process and PCA is employed to monitor the residuals in order to evaluate whether the current process is under normal working condition or not. Literature examples are used to illustrate the utility of JITL-PCA in monitoring the nonlinear systems and a comparison with the conventional PCA and dynamic PCA is made. Simulation results show that JITL-PCA gives marked improvement over PCA and DPCA in the monitoring of nonlinear static or dynamic systems.

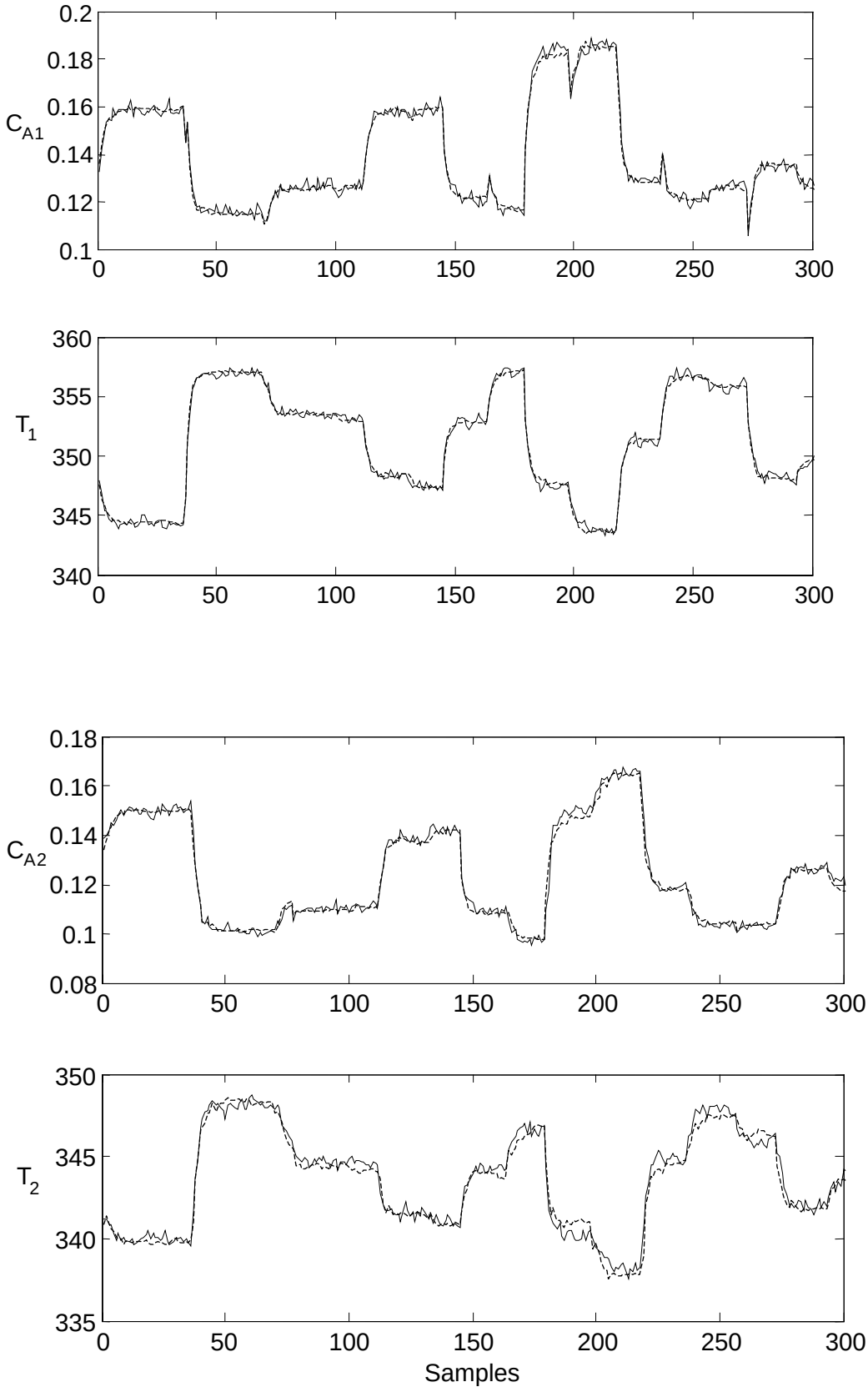


Figure 8.9 Modeling result of JITL under normal condition.
Solid line: actual output; dashed line: model output

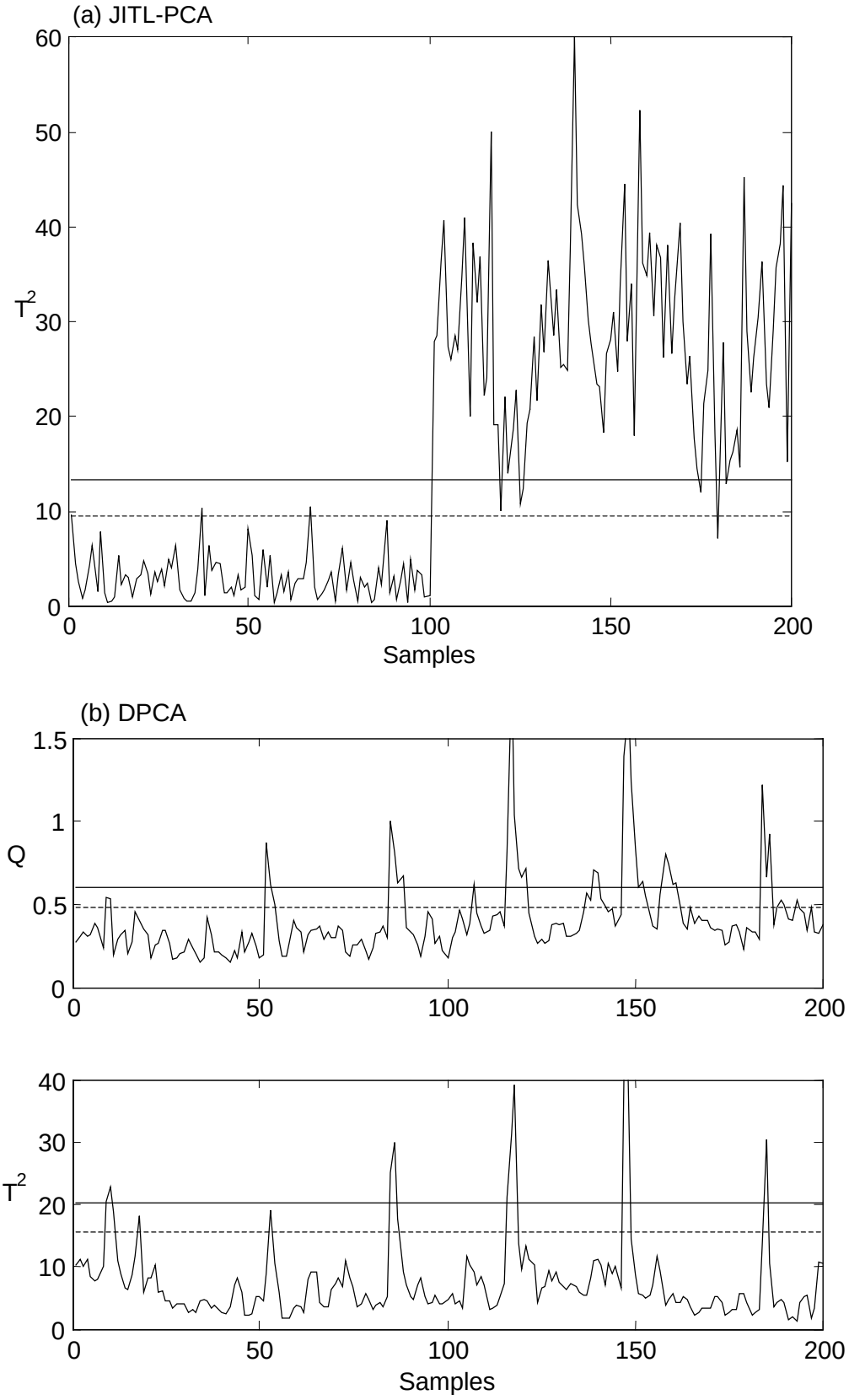


Figure 8.10 Monitoring result of fault 1: (a) JITL-PCA; (b) DPCA

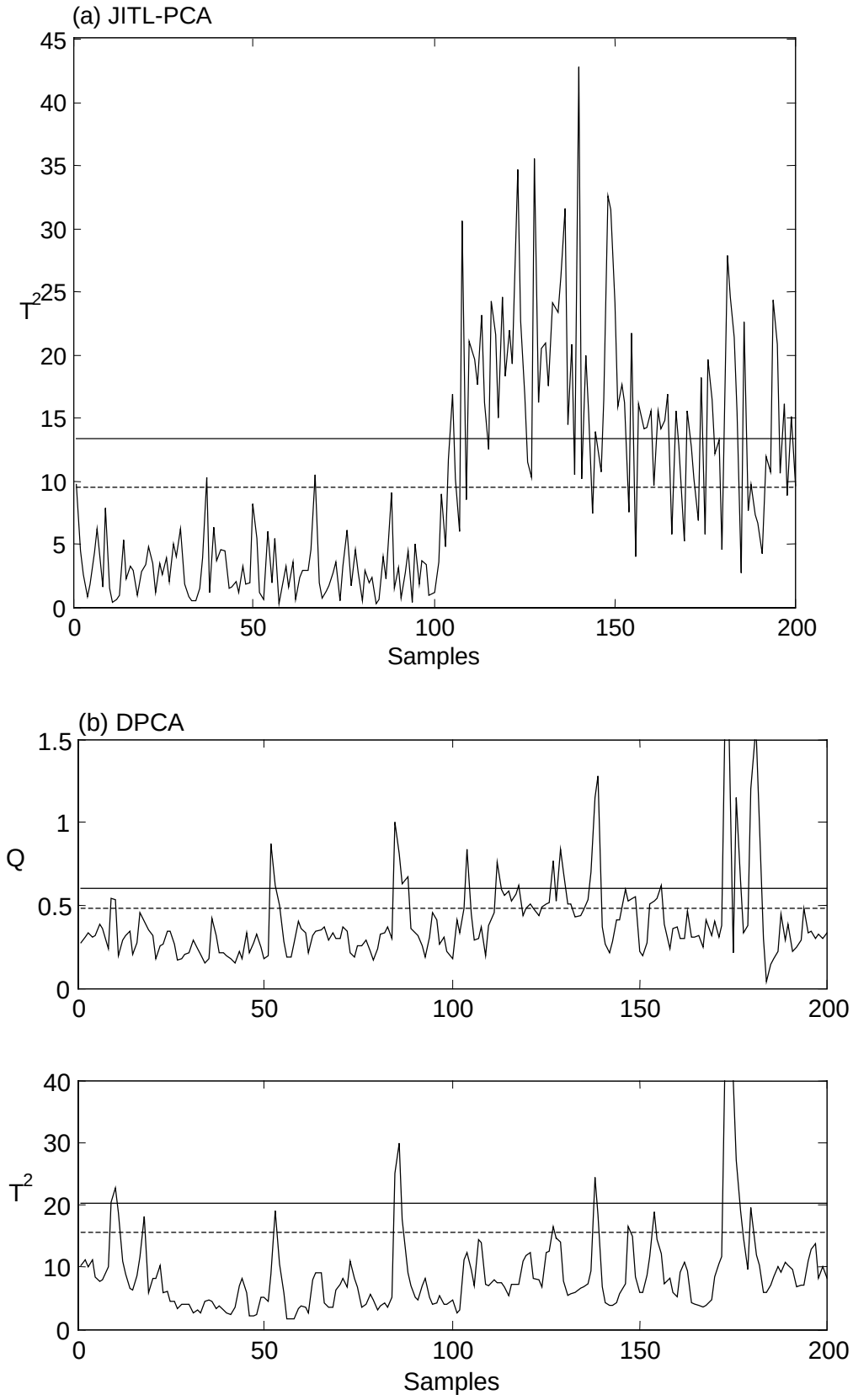


Figure 8.11 Monitoring result of fault 2: (a) JITL-PCA; (b) DPCA

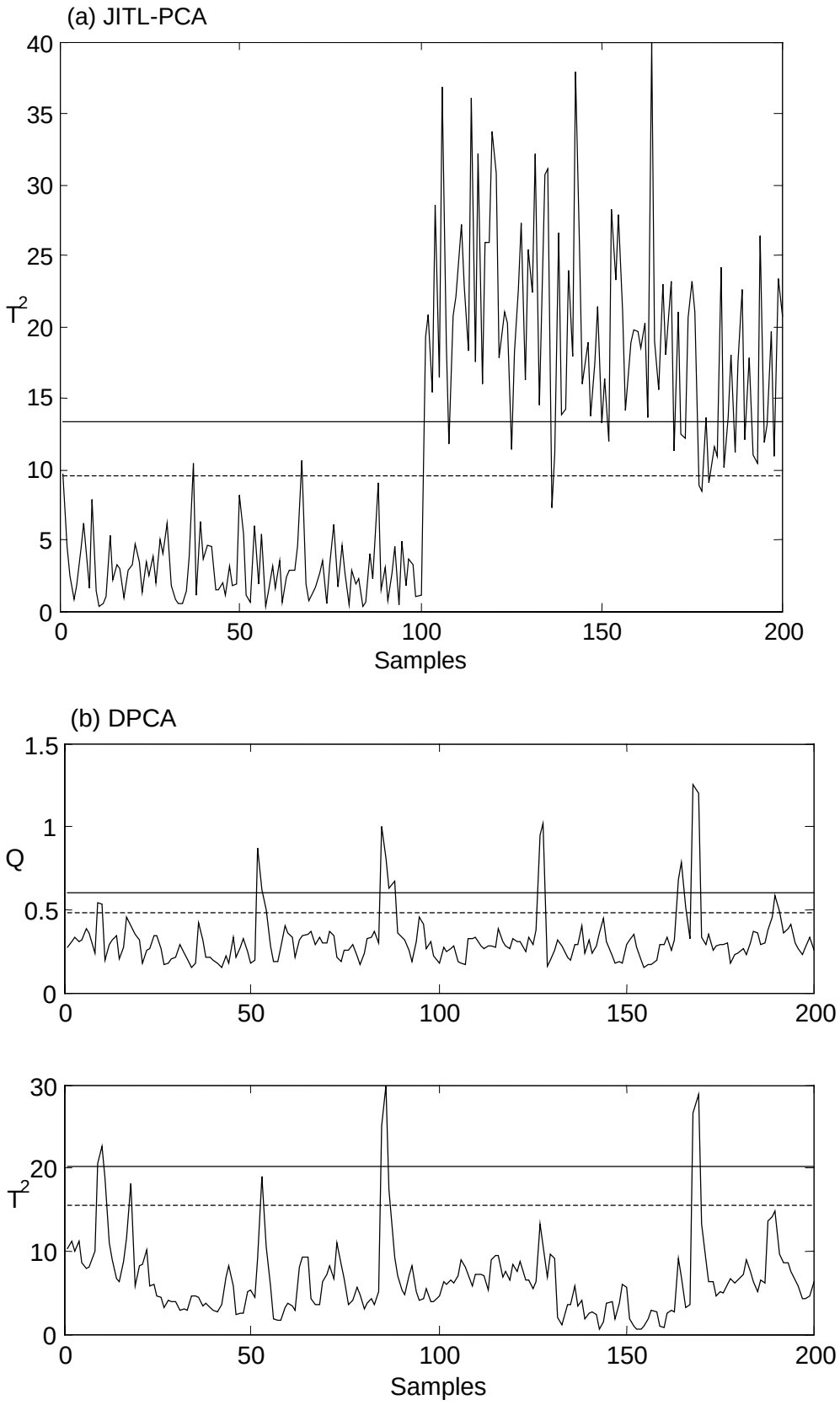


Figure 8.12 Monitoring result of fault 3: (a) JITL-PCA; (b) DPCA

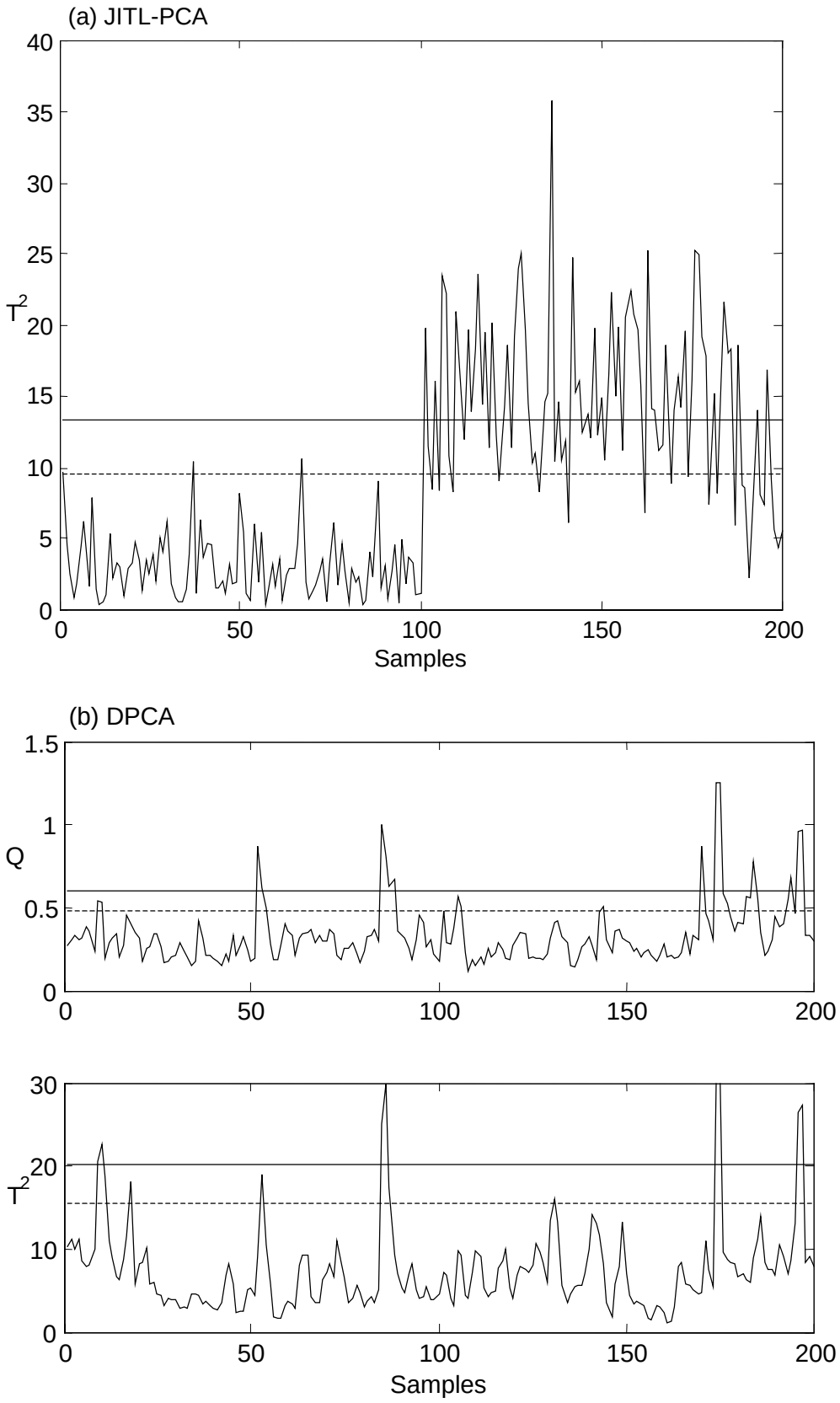


Figure 8.13 Monitoring result of fault 4: (a) JITL-PCA; (b) DPCA

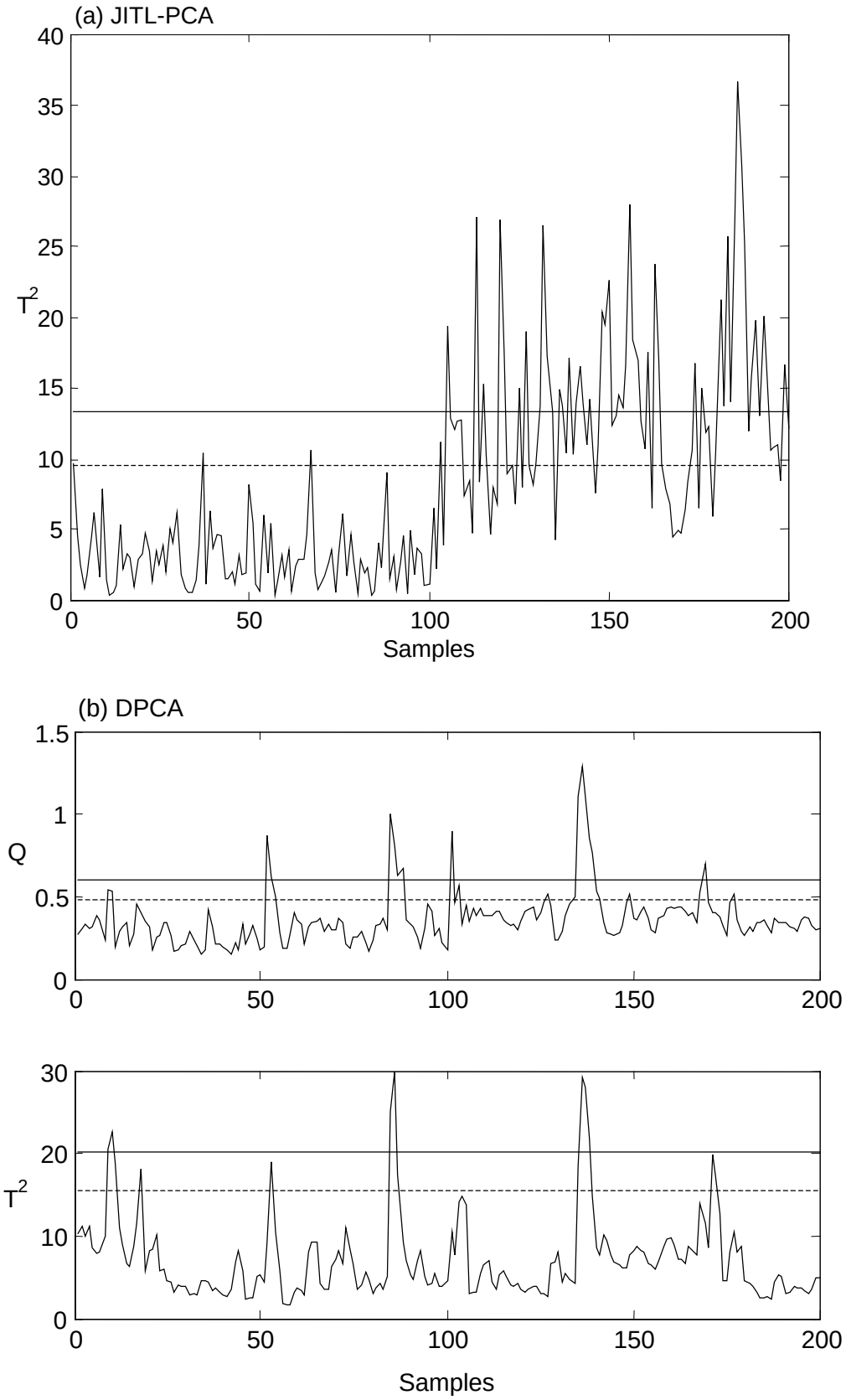


Figure 8.14 Monitoring result of fault 5: (a) JITL-PCA; (b) DPCA

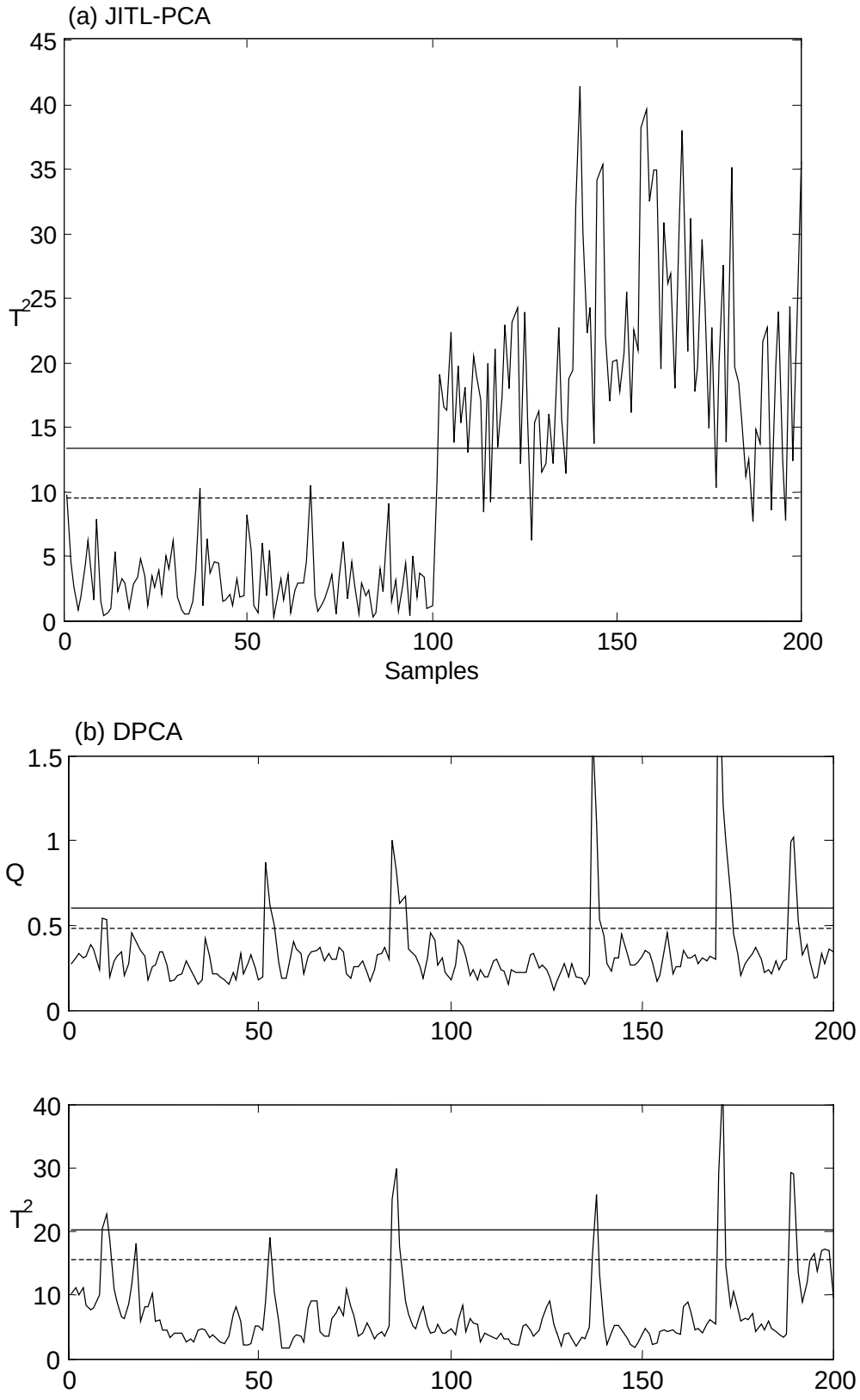


Figure 8.15 Monitoring result of fault 6: (a) JITL-PCA; (b) DPCA

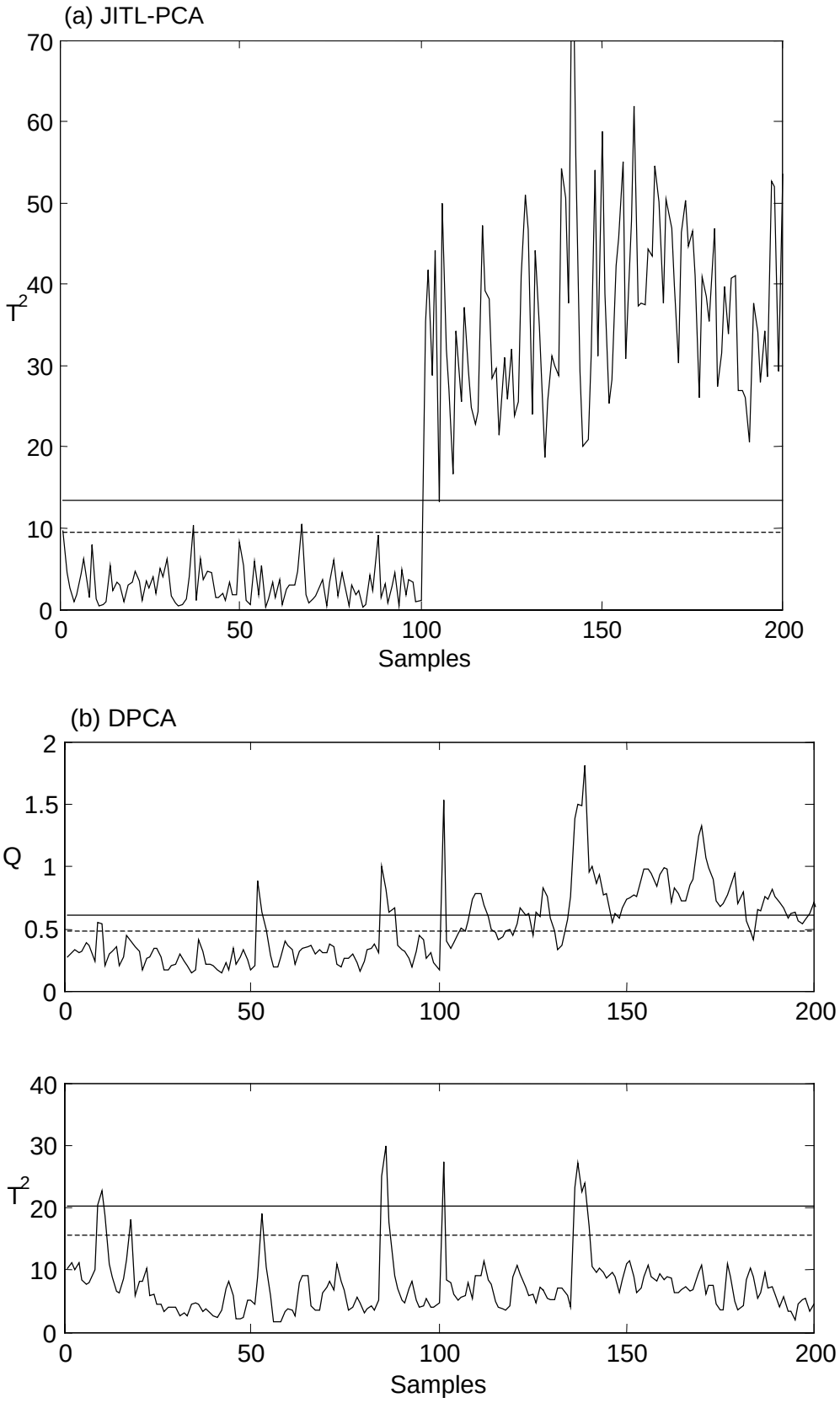


Figure 8.16 Monitoring result of fault 7: (a) JITL-PCA; (b) DPCA

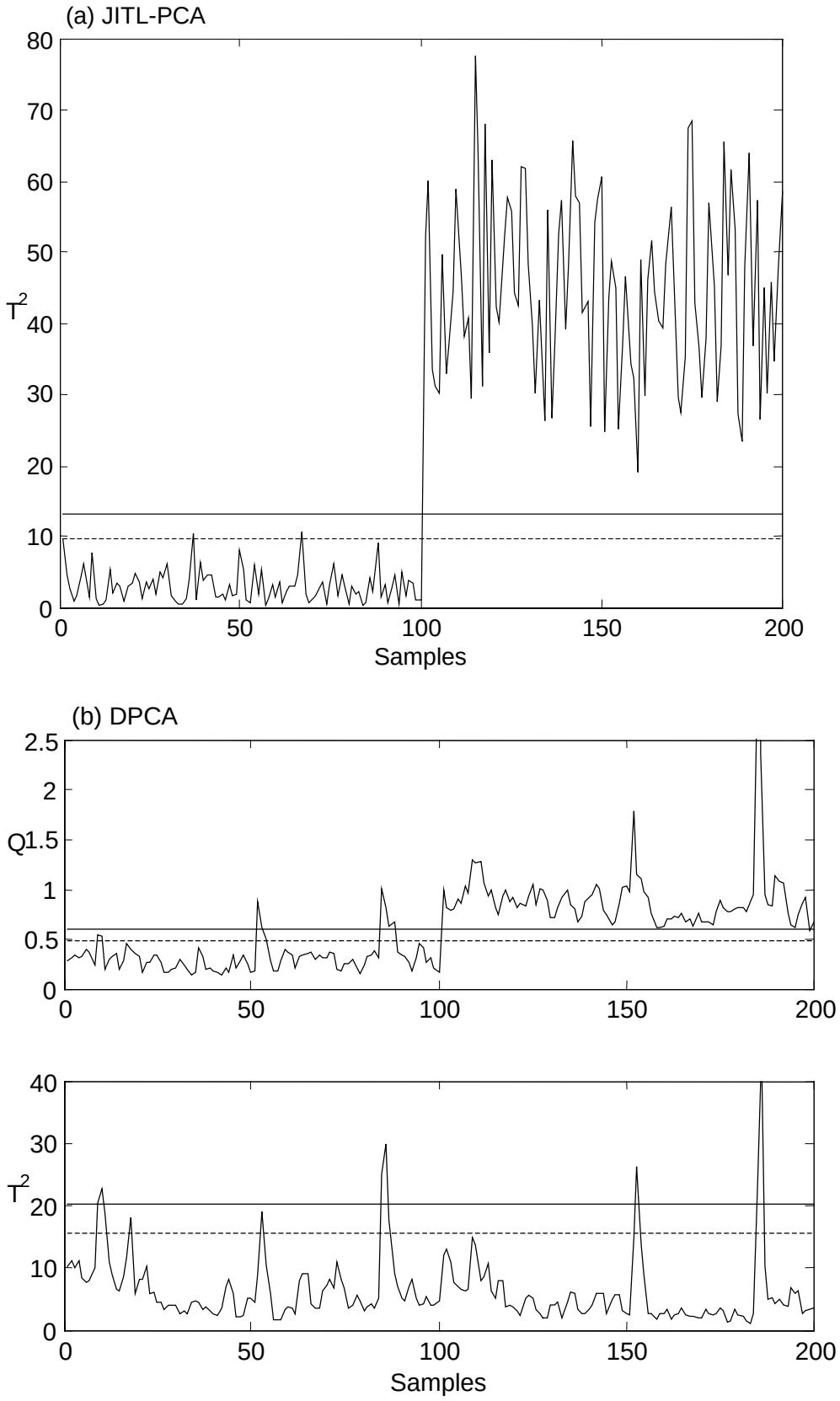


Figure 8.17 Monitoring result of fault 8: (a) JITL-PCA; (b) DPCA

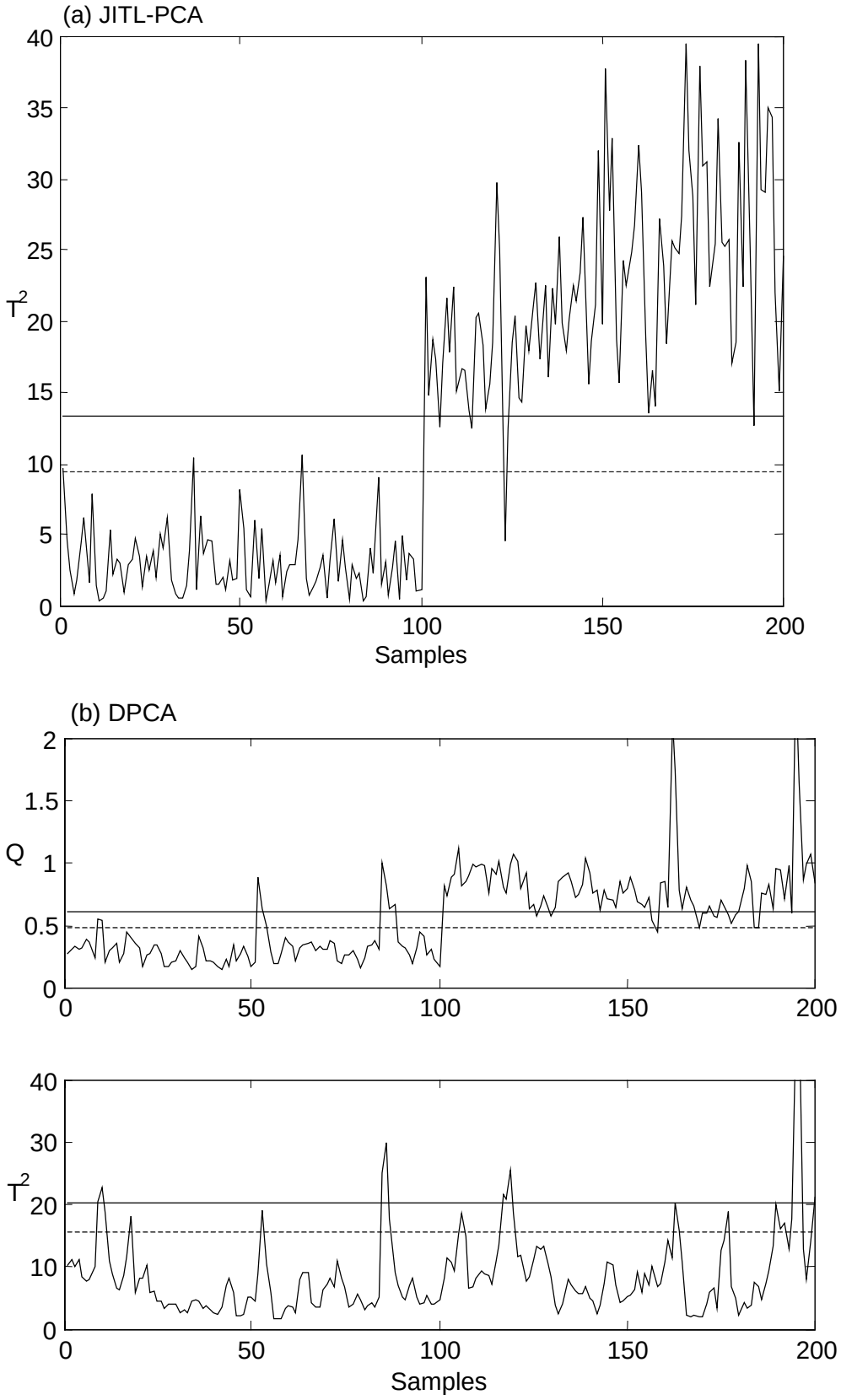


Figure 8.18 Monitoring result of fault 9: (a) JITL-PCA; (b) DPCA

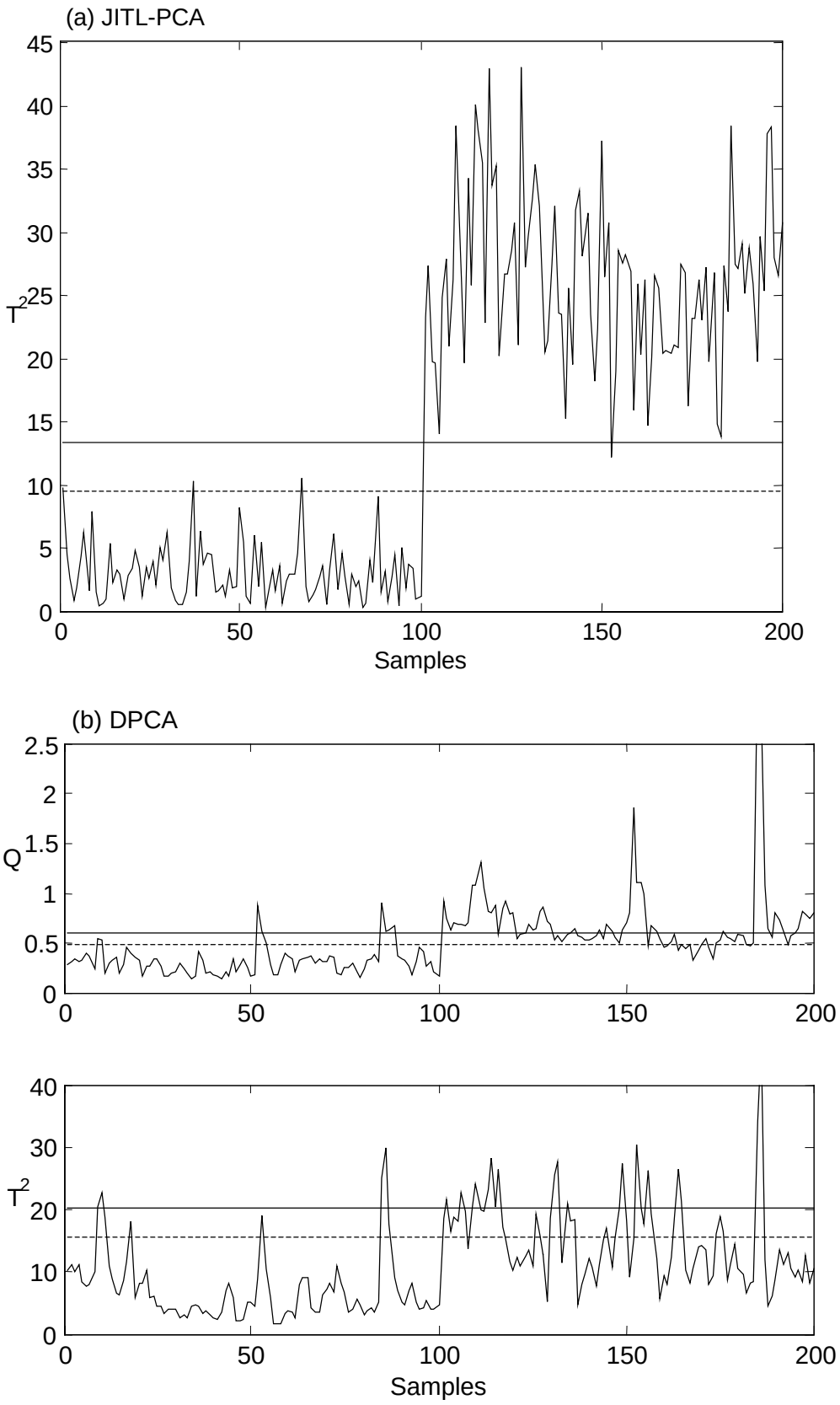


Figure 8.19 Monitoring result of fault 10: (a) JITL-PCA; (b) DPCA

Chapter 9

Conclusions and Further Work

9.1 Conclusions

In chemical processes, a well-known problem called ‘data rich but information poor’ is constantly encountered by the engineers dealing with the assignments of nonlinear process modeling, control, and monitoring. To circumvent this problem, this thesis investigates the use of JITL technique as a modeling framework, by which robust and adaptive controller designs and a process monitoring methodology for nonlinear processes are developed. Unlike the standard learning methods, JITL has no standard learning phase and the models are built dynamically upon query at each sampling instant. In this sense, 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. These advantages make JITL attractive in nonlinear process modeling, modeling and monitoring.

In this thesis, an enhanced JITL technique is developed by combining the angle measure into the distance measure to evaluate similarity between two data

samples to improve the modeling accuracy. In addition, parametric stability constraints are incorporated into the proposed method to address the stability of local models. Moreover, a new procedure of selecting the relevant data set is proposed. Simulation results show that the proposed JITL has better modeling accuracy than its conventional counterparts.

By using the enhanced JITL technique, several control strategies are proposed. Firstly, a robust controller design methodology is proposed based on a composite model consisting of a nominal ARX model and JITL, where the former is used to capture the linear process dynamics and the latter to approximate the process nonlinearity, which is assumed to be the only source of the model uncertainty. The state space realizations of the resulting model and PID controller are then reformulated as an uncertain system, which can be recast into the standard $M-\Delta$ structure, by which the robust stability analysis by using the structured singular value test can be developed as the design criterion for robust PID controller design. Literature examples are employed to illustrate that the proposed methodology can be used to obtain the robust stability region in the parameter space of a PI controller, which assures the closed-loop stability for controlling the nonlinear process in the concerned operating space.

Secondly, by integrating JITL into the controller design, three data-based adaptive control strategies are developed, meaning that adaptive single-neuron (ASN) controller, adaptive IMC controller, and auto-tuning PID controller. These controllers take advantages of the information provided by JITL to realize online control parameter tuning for nonlinear process. Because of the parsimonious design framework, these adaptive controllers can be implemented online without heavy

computational burden. The simulation results illustrate that the proposed methods have better control performance than their counterparts.

Lastly, this thesis proposes a JITL-PCA scheme for nonlinear dynamic process monitoring. JITL serves as the process observer to account for the nonlinear dynamic characteristics of the process under normal operating conditions. The residuals resulting from the difference between JITL and process outputs are analyzed by PCA to evaluate the status of the current process operating conditions. Simulation results show that JITL-PCA outperforms PCA and DPCA in the monitoring of the nonlinear processes.

9.2 Suggestions for Further Work

There are few open questions that need to be further studied. Some possible topics for future research are listed below.

JITL is one of the powerful techniques for learning from observed data and for gaining insight on the local behaviour of nonlinear systems. However there are still some concerns for adoption of JITL for process modeling and control. One is the well known problem of the curse of dimensionality, that is the sparseness of data in situation of high dimension of the query space. JITL shares this problem with all other nonlinear modeling techniques. But, unlike other approaches, JITL can take advantage from its feature of updating the database continuously (Atkeson et al., 1997). Therefore, the technique to detect the local regions where data are missing and need to collect additional samples could be found useful in this context. Another concern is how can JITL model real systems fast enough when the size of the database grows. This problem is actually a dataset-searching problem. When working with huge datasets, it would be more desirable to utilize a database management

system (DBMS). Then, how such a DBMS could be integrated with the estimation procedure clearly needs further investigation.

Virtually all practical control systems are subject to hard constraints on their manipulated inputs. In this respect, model predictive controller (MPC) is now widely recognized as one of the few design methods of handling constraints in a systematic manner. However, MPC techniques are based on linear models and thus not very well-suited for the control of nonlinear systems. Thus how to integrate JITL technique into MPC design for nonlinear system warrants future investigation.

For the data-based control strategies developed in Chapters 5 to 7, the controller parameters are updated on-line by the steepest descent method with the aim to minimize the tracking error at the next sampling instant. In this regard, it is of theoretical interest for future research to derive a more rigorous updating algorithm for controller parameters by the Lyapunov method such that the convergence of predicted tracking error at the next sampling instant is guaranteed.

Last, how to integrate process monitoring and controller design is an open topic that warrants further investigation. Specifically, when fault is identified and diagnosed in a control system, how can the monitoring information be applied in the redesign of controller to maintain acceptable control performance? This problem is closely related to the fault tolerant controller design and is a challenging problem given the complexity of chemical processes. Moreover, from the demand of process model's point of view, there is a fundamental conflict between the tasks of controller design and process monitoring. For example, in adaptive controller design, it is desired to have a model that is able to track parameter variations quickly for better control performance. Under this situation, it expects small residuals between the plant outputs and the predicted outputs despite that all the changes occur in the system. On

the other hand, a process model insensitive to the changing parameters is preferred in process monitoring application because large residuals can be obtained in the presence of faults, leading to higher successful rate of detecting system abnormalities.

References

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

Aha, D. W. (1997). Lazy learning. *Artificial Intelligence Review*, **11**, 7-10.

Andrasik, A., Meszaros, A., and de Azevedo, S. F. (2004). On-line tuning of a neural PID controller based on plant hybrid modeling. *Computers and Chemical Engineering*, **28**, 1499-1590.

Åström, K. J. (1983). Theory and application of adaptive control – A survey. *Automatica*, **19**, 471-486.

Åström, K. J. and Hagglund, T. (1995). PID controllers: theory, design and tuning, 2nd Edition. International Society for Measurement and Control, North Carolina.

Åström, K. J. and Wittenmark, B. (1995). Adaptive control, 2nd Edition. Addison-Wesley.

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

Bakshi, B. R. (1998). Multiscale PCA with application to multivariate statistical process monitoring. *AIChE Journal*, **44**, 1596-1610.

Bai, E. W. (1998). An optimal two-stage identification algorithm for Hammerstein-Winner nonlinear system. *Automatica*, **34**, 333-338.

Bequette, B. W. (1991). Nonlinear control of chemical processes: a Review. *Industrial and Engineering Chemistry Research*, **30**, 1391-1413.

Bhat, N. and Mcavoy, T. J. (1990). Use neural nets for dynamic modeling and control of chemical process systems. *Computer and Chemical Engineering*, **14**, 573-583.

Billings, S. A. and Fakhouri, S. Y. (1978). Identification of a class of nonlinear systems using correlation analysis. *Proceedings of IEE*, **125**, 691-697.

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

te Braake, H. A. B., van Can, E. J. L., Scherpen, J. M. A., and Verbruggen, H. B. (1998). Control of nonlinear chemical processes using neural models and feedback linearization. *Computers and Chemical Engineering*, **22**, 1113-1127.

Braun, M. W., Rivera, D. E., and Stenman, A. (2001). A model-on-demand identification methodology for nonlinear process systems. *International Journal of Control*, **74**, 1708-1717.

Budman, H. M., Knapp T. (2001). Stability analysis of nonlinear processes using empirical state affine models and LMIs. *Journal of Process Control*, **11**, 375-386.

Calvet, J. P. and Arkun, Y. (1988). Feedforward and feedback linearization of nonlinear systems and its implementation using internal model control (IMC). *Industrial and Engineering Chemistry Research*, **27**, 1822-1831.

Chang, F., and Luus, R. (1971). A non-iterative method for identification using Hammerstein model. *IEEE Transactions on Automatic Control*, **16**, 464-468.

Chen, J. and Liao, C.M., (2002). Dynamic process fault monitoring based on neural network and PCA. *Journal of Process Control*, **12**, 277-289.

Chen, J. and Huang, T. (2004). Applying neural networks to on-line updated PID controllers for nonlinear process control. *Journal of Process Control*, **14**, 211-230.

Chen, S., Billings, S. A., and Grant, P. M. (1990). Non-linear system identification using neural networks. *International Journal of Control*, **51**, 1191-1214.

Chiang, H. L., Russell, E. L., and Braatz, R. D. (2001). Fault detection and diagnosis in industrial systems. Springer, New York.

Cybenko, G. (1989). Approximation by superpositions of sigmoidal function. *Mathematics of Control Signals and Systems*, **2**, 303-314.

Cybenko, G. (1996). Just-in-time learning and estimation. In S. Bittanti & G. Picci, Identification, adaptation, learning: the science of learning models from data, Springer, NY., 423-434.

Diaz, H. and Desrochers, A. (1988). Modeling of nonlinear discrete-time system from input/output data. *Automatica*, **24**, 629-641.

Dong, D. and McAvoy, T. J. (1996). Nonlinear principal component analysis-based principal curves and neural networks. *Computers and Chemical Engineering*, **20**, 65-78.

Doyle, F. J., Packard, A. K., and Morari, M. (1989). Robust controller design for a nonlinear CSTR. *Chemical Engineering Science*, **44**, 1929-1947.

Doyle, F. J., Ogunnaike, B**.** A., and Pearson, R. K. (1995). Nonlinear model-based control using second-order volterra models. *Automatica*, **31**, 697-714.

Economou, C. G., Morari, M., and Palsson, B. O. (1986). Internal model control. 5. Extention to nonlinear systems. *Ind. Eng. Chem. Process Des. Dev*. **25**, 403-411.

Frank, P. M. (1990). Fault diagnosis in dynamic systems using analytical and knowledge based redundancy. *Automatica*, **26**, 459-474.

Frank, P. M., Ding, S. X., and Koppen-Selioger, B. (2000). Current development in the theory of FDI. *4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Process*, Budapest, Hungary, Vol. 1, pp. 16-27.

Gao, F., Wang, F., and Li, M. (2000). An analytical predictive control law for a class of nonlinear process. *Industrial and Engineering Chemistry Research*, **39**, 2029-2034.

Ge, S. S., Hang, C. C., Lee, T. H., and Zhang, T. (2002). Stable adaptive neural network control. Kluwer Academic, Boston.

Garcia, C. E. and Morari, M. (1982). Internal model control – 1. a unifying review and some new results. *Ind. Eng. Chem. Process Des. & Dev*., **21**, 308-323.

Harris, K. R. and Palazoglu, A. (1998). Studies on the analysis of nonlinear process via functional expansions-III: Controller design. *Chemical Engineering Science*, **53**, 4005-4022.

Henson, M. A. and Seborg, D.E. (1991a). Critique of exact linearization strategies for process control. *Journal of Process Control*, **1**, 122-139.

Henson, M. A. and Seborg, D. E. (1991b). An internal model control strategy for nonlinear systems. *AIChE Journal*, **37**, 1065-1078.

Henson, M. A. and Seborg, D. E. (1997). Nonlinear process control. Prentice Hall, NJ.

Hernandez, E. and Arkun, Y. (1992). Study of the control-relevant properties of backpropagation neural network models of nonlinear dynamical systems. *Computers and Chemical Engineering*, **16**, 227-240.

Hiden, H. G., Willis, M. J., Tham, M. T., and Montague, G. A. (1999). Non-linear principal components analysis using genetic programming. *Computers and Chemical Engineering*, **23**, 413-425.

Himmelblau, D. M. (1978). Fault detection and diagnosis in chemical and petrochemical processes. Elsevier, Amsterdam.

Hornik, K., Stinchombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Computation*, **2**, 359-366.

Hunt, K. J. and Sbarbaro, D. (1991). Neural networks for nonlinear internal model control. *IEE Proceedings-D*, **138**, 431-438.

Hur, S. M., Park, M. J., and Rhee, H. K. (2003). Design and application of model-on-demand predictive controller to a semibatch copolymerization reactor. *Industrial and Engineering Chemistry Research*, **42**, 847-859.

Isermann, R. and Belle, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, **5**, 709-719.

Jackson, J. E. and Mudholkar, G. S. (1979). Control procedures for residuals associated with principal component analysis. *Techometrics*, **21**, 341-349.

Jang, J. S. R. (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on System, Man, and Cybernetics*, **23**, 665-685.

Jang, J. S. R. and Sun, C. T. (1995). Neuro-fuzzy modeling and control. *Proceedings of the IEEE*, **83**, 378-406.

Jeon, G. J. and Lee, I. (1996). Neural network indirect adaptive control with fast learning algorithm. *Neurocomputing*, **13**, 185-199.

Jin, L., Nikiforuk, P. N., and Gupta, M. M. (1994). Adaptive control of discrete-time nonlinear systems using recurrent neural networks. *IEE Proceedings – Control Theory and Application*, **141**, 169-176.

Kalafatis, A. D., Wang, L., and Cluett, W. R. (1997). Identification of Wiener-type nonlinear systems in a noisy environment. *International Journal of Control*, **66**, 923-941.

Kano, M., Hasebe, S., Hashimoto, I., and Ohno, H. (2004). Evolution of multivariate statistical process control: application of independent component analysis and external analysis. *Computers and Chemical Engineering*, **28**, 1157-1166.

Knapp, T. D. and Budman, H. M. (2000). Robust control design of non-linear processes using empirical state affine models. *International Journal of Control*, **73**, 1525-1535.

Kohonen, T. (1995). Self-organization maps. Springer-Verlag, Berlin.

Kramer, M. A. (1992). Autoassociative neural networks. *Computers and Chemical Engineering*, **16**, 313-328.

Kresta, J. V., MacGregor, J. F., and Marlin, T. E. (1991). Multivariate statistical monitoring of process operating performance. *Canadian Journal of Chemical Engineering*, **69**, 34-47.

Krishnapura, V. G. and Jutan, A. (2000). A neural adaptive controller. *Chemical Engineering Science*, **55**, 3803-3812.

Ku, W., Storer, R. H., and Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, **30**, 179-196.

Lakshminararyanan, S., Shah, S. L., and Nandakumar, K., (1995). Identification of Hammerstein models using statistical tools. *Chemical Engineering Science*, **50**, 3599-3613.

Lewis, F. L., Yesildireck, A., and Liu, K. (1996). Multilayer neural-net robot controller with guaranteed tracking performance. *IEEE Transactions on Neural Networks*, **7**, 388-389.

Li, Y., Powers, D. and Wen, P. (2000). Internal model control using recurrent neural networks for nonlinear dynamic systems. *Distributed Computer Control Systems 2000: A Proceedings Volume from the 16th IFAC Workshop*, Sydney, Australia, pp.7-12.

Lin, C. T. and Lee, C. S. G. (1991). Neural-network-based fuzzy logic control and decision system. *IEEE Transactions on Computers*, **40**, 1320-1336.

Linkens, D. A. and Nyongesa, H. O. (1996). Learning systems in intelligent control: an appraisal of fuzzy, neural and genetic algorithm control applications. *IEE Proceedings – Control Theory and Applications*, **143**, 367-386.

Loeblain, C. and Perkins, J. D. (1998). Economic analysis of different structures of on-line process optimization systems. *Computers and Chemical Engineering*, **22**, 1237-1269.

MacGregor, J. F. and Kourti, T. (1995). Statistical process control of multivariate processes. *Control Engineering Practice*, **3**, 403-414.

Maksumov, A., Mulder, D. J., Harris K. R., and Palazoglu, A. (2002). Experimental application of partitioned model-based control to PH neutralization. *Industrial and Engineering Chemistry Research*, **41**, 744-750.

Malan, S., Milanese, M., Regruto, D., and Taragna, M. (2004). Robust control from data via uncertainty model sets identification. *International Journal of Robust and Nonlinear Control*, **14**, 945-957.

Malthouse, E. C., Tamhane, A. C., and Mah, R. H. (1997). Nonlinear partial least squares. *Computers and Chemical Engineering*, **21**, 875-890.

Maner, B. R., Dolye, F. J. III, Ogunnaike, B. A., and Pearson, P. K. (1996). Nonlinear model predictive control of a simulated multivariable polymerization reactor using second-order Volterra models. *Automatica*, **32**, 1285-1301.

Mastorocostas, P. A. and Theocharis, J. B. (2002). A recurrent fuzzy-neural model for dynamic system identification. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, **32**, 176-190.

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

Myers, R. H. (1990). Classical and modern regression with applications. PWS-Kent Pub., Boston.

Nahas, E. P., Henson, M. A., and Seborg, D. E. (1992). Nonlinear internal model control strategy for neural network models. *Computers and Chemical Engineering*, **16**, 1039-1057.

Narendra, K. S. and Gallman P. G. (1966). An iterative method for the identification of nonlinear system using a Hammerstein model. *IEEE Transactions on Automatic Control*, **11**, 546-550.

Narendra, K. S. and Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, **1**, 4-27.

Negiz, A. and Cinar, A. (1997). Statistical monitoring of multivariable dynamic processes with state-space models. *AIChE Journal*, **43**, 2002-2019.

Nelles, O. (2001). Nonlinear system identification. Springer-Verlag, Berlin.

Nikravesh, M., Farell, A. E., and Stanford, T. G. (1997). Dynamic neural network control for nonlinear systems: optimal neural network structure and stability analysis. *Chemical Engineering Journal*, **68**, 41-50.

Nomikos, P. and MacGregor, J. F. (1994). Monitoring of batch processes using multi-way PCA. *AIChE Journal*, **40**, 1361-1375.

Nomikos, P. and MacGregor, J. F. (1995). Multi-way partial least squares in monitoring batch processes. *Chemometrics and Intelligent Laboratory Systems*, **30**, 97-108.

Ogunnaike, B. A. and Raymod, A. W. (1996). Industrial application of nonlinear control. *Proceedings of the 5th International Conference on Chemical Process Control*, Tahoe City, U.S.A., pp. 46-59.

Omatu, S., Khalid, M., and Yusof, R. (1996). Neuro-control and its application. Springer, London.

Packard, A. and Doyle, F. J. (1993). The complex structured singular value. *Automatica*, **29**, 71-109.

Patton, R. J., Chen, J., and Benkhedda, H. (2000). A study on neuro-fuzzy systems for fault diagnosis. *International Journal of Systems Science*, **31**, 1441-1448.

Pearson, R. K. (1995). Nonlinear input/output modeling. *Journal of Process Control*, **5**, 197-211.

Pearson, R. K. (1999). Discrete-Time Dynamic Models. Oxford University Press, Oxford.

Piovoso, M. J., Kosanovich, K. A., and Yuk, J. P. (1992). Process data chemometrics. *IEEE Transactions on Instrumentation and Measurement*, **41**, 262-268.

Polycarpou, M. M. (1996). Stable adaptive neural control scheme for nonlinear systems. *IEEE Transactions on Neural Networks*, **3**, 837-863.

Porter, B. and Jones, A. H. (1992). Genetic tuning of digital PID controllers. *Electronics Letters*, **28**, 843-844.

Qin, S. J. and McAvoy, T. J. (1992). Nonlinear PLS modeling using neural network. *Computers and Chemical Engineering*, **16**, 379-391.

Raich, A. and Cinar, A. (1997). Diagnosis of process disturbances by statistical distance and angel measures. *Computers and Chemical Engineering*, **21**, 661-673.

Rhodes, C. and Morari, M (1997). Data-based control trajectory planning for nonlinear systems. *Physical Review E*, **56**, 2398-2406.

Rotem, Y., Wachs, A., and Lewin, D. R. (2000). Ethylene compressor monitoring using model-based PCA. *AIChE Journal*, **46**, 1825-1836.

Saludes, S. and Fuente, M. J. (1999). Neural network based fault detection and accommodation in a chemical reactor. *Proceedings of the 14th IFAC World Congress*, Beijing, China, Vol. P, pp. 169-174.

Seborg, D. E., Edgar, T. F., and Shah, S. L. (1986). Adaptive control strategies for process control: a survey. *AIChE Journal*, **32**, 881-913.

Shao, R., Jia, F., Martin, E. B., and Morris, A. J. (1999). Wavelets and nonlinear principal components analysis for process monitoring. *Control Engineering Practice*, **7**, 865-879.

Shaw, A. M., Dolye, F. J. III, and Schwaber, J. S. (1997). Dynamic neural network approach to nonlinear process modeling. *Computers and Chemical Engineering*, **21**, 371-385.

Singhal, A. and Seborg, D. E. (2002). Pattern matching in multivariate time series database using a moving-window approach. *Industrial and Engineering Chemistry Research*, **41**, 3822-3838.

Sontag, E. (1979). Realization theory of discrete-time nonlinear systems: Part I. The bounded case. *IEEE Transactions on Circuits and Systems*, **CAS-26**, 324-356.

Stephanopoulos, G. and Han, C. (1996). Intelligent systems in process engineering: a review. *Computers and Chemical Engineering*, **20**, 743-791.

Su, H. T. and McAvoy, T. J. (1992). Long-term predictions of chemical processes using recurrent neural networks: a parallel training approach. *Industrial and Engineering Chemistry Research*, **31**, 1338-1352.

Su, H. T. and McAvoy, T. J. (1997). Artificial neural networks for nonlinear process identification and control. In: Nonlinear Process Control (M. A. Henson and D. E. Seborg, Eds.), pp. 371-428, Prentice Hall, NJ.

Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, **15**, 116-132.

Toffner-Clausen, S. (1996). System identification and robust control. Springer-Verlag, London.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., and Kavuri, S. N. (2003a). A review of process fault detection and diagnosis Part I: Quantitative model-based methods. *Computers and Chemical Engineering*, **27**, 293-311.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., and Kavuri, S. N. (2003b). A review of process fault detection and diagnosis Part II: Qualitative models and search strategies. *Computers and Chemical Engineering*, **27**, 313-326.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., and Kavuri, S. N. (2003c). A review of process fault detection and diagnosis Part III: Process history based methods. *Computers and Chemical Engineering*, **27**, 293-311.

Wachs, A. and Lewin, D. R. (1998). Process monitoring using model-based PCA. *Proceedings of the 5th IFAC Symposium of Dynamics and Control of Process Systems*, Corfu, Greek, pp. 86-81.

Wigren, T. (1994). Convergence analysis of recursive identification algorithms based on the nonlinear Wiener model. *IEEE Transactions on Automatic Control*, **39**, 2191-2206.

Wilson, D. J. H. and Irwin, G. W. (2000). PLS modeling and fault detection on the Tennessee Eastman benchmark. *International Journal of System Science*, **31**, 1449-1457.

Wise, B. M. and Gallagher, N. B. (1996). The process chemometrics approach to process monitoring and fault detection. *Journal of Process Control*, **6**, 329-348.

Xu, L., Oja. E., and Suen, C. Y. (1992). Modified Hebbian learning for curve and surface fitting. *Neural Networks*, **5**, 441-457.

Yamamoto, T., Takao, K., and Hinamoto T. (2004). A design of memory-based PID controllers. *Proceedings of the 5th Asian Control Conference*, Melbourne, Australia, pp. 497-505.

Yeo, Y. and Kwon, T. (1999). A neural PID controller for the pH neutralization process. *Industrial and Engineering Chemistry Research*, **38**, 978-987.

Yoon, S. and MacGregor, J. F. (2000). Statistical and causal model-based approaches to fault detection and isolation. *AIChE Journal*, **46**, 1813-1824.

Yoon, S. and MacGregor, J. F. (2001). Fault diagnosis with multivariate statistical models part I: using steady state fault signatures. *Journal of Process Control*, **11**, 387-400.

You, Y. and Nikolaou, M. (1993). Dynamic process modeling with recurrent neural networks. *AIChE Journal*, **39**, 1654-1667.

Zhang, Q. (1997). Using wavelet network in nonparametric estimation. *IEEE Transactions on Neural Networks*, **8**, 227-236.

Zhang, J. and Morris, A. J. (1999). Recurrent neuro-fuzzy network for nonlinear process modeling. *IEEE Transactions on Neural Networks*, **10**, 313-326.

PUBLICATION AND PRESENTATION

Cheng, C. and M. S. Chiu. "Nonlinear Process Monitoring Using JITL-PCA", *Chemometrics and Intelligent Laboratory Systems*, **76**, 1-13 (2005).

Cheng, C. and M. S. Chiu. "A New Data-Based Methodology for Nonlinear Process Modeling", *Chemical Engineering Science*, **59** 2801-2810 (2004).

Zhuang, H., Cheng, C, and M. S. Chiu. "Empirical Modeling of a Pulse-Jet Fabric Filter: An Experimental Study", *Journal of the Chinese Institute of Chemical Engineers*, **35**, 17-22 (2004).

Cheng, C. and M. S. Chiu. "JITL Based Adaptive IMC Controller Design", Accepted by *Chemical Engineering Research & Design* (2007).

Cheng, C. and M. S. Chiu. "Adaptive Single-Neuron Controller Design For Nonlinear Process Control", *Submit to J. of Chem. Eng. of Japan* (2006).

Cheng, C. and M. S. Chiu. "Data-based Robust PID Controller Design", Submit to *Ind. Eng. Chem. Res.* (2006).

Cheng, C., Hashimoto, Y., and M. S. Chiu. "Adaptive Controller Design Using Just-in-Time Learning Algorithm", *IEEE Conference on Control Applications*, Taipei, Taiwan, 2-4 September (2004).

Cheng, C. and M. S. Chiu. "Data-Based PI Control Strategy of A Polymerization Reactor", *International Conference on Artificial Intelligence and Applications*, Benalmadena, Spain, 8-10 September (2003).

Cheng, C. and M. S. Chiu. "Nonlinear Process Modeling Based On Just-in-Time Learning and Angle Measure", *The 7th International Conference on Knowledge-Based Intelligent Information & Engineering Systems*, Cambridge, UK, 3-5 September (2003).

Cheng, C. and M. S. Chiu. "Model-based Process Fault Detection Using Lazy Learning", PSE Asia 2002, Taipei, Taiwan, 4-6 December (2002).

Cheng, C. and M. S. Chiu. “A Hierarchy Neural Network for Fault Detection and Diagnosis”, *Proc. of International Symposium of Advanced Control of Industrial Processes*, Kumamoto, Japan, 10-11 June, 429-433, (2002).