NONLINER CONTROLLER DESIGN FROM PLANT DATA

YASUKI KANSHA
(B. Eng., Kyoto University, Japan)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF CHEMICAL AND BIOMOLECULAR ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2007

ACKNOWLEDGEMENTS

I would like to thank my research supervisor, Dr. Min-Sen Chiu, for his constant support, invaluable guidance and suggestions throughout my research work at National University of Singapore. He showed me different ways to approach a research problem and the need to be persistent to accomplish any goal. My special thanks to Dr. Chiu for his invaluable time to read this manuscript.

I greatly appreciate the valuable advices and concerns I received from Dr. Gade Pandu Rangaiah, Dr. Lakshminarayanan Samavedham, and Dr Qing-Guo Wang. I would like to extend special thanks to Dr. Yoshihiro Hashimoto and Dr. Li Jia to give me invaluable suggestions to my research work. Special thanks and appreciation to my lab mates, Dr Cheng Cheng, Ye Myint Hlaing, Ankush Ganeshreddy Kalmukale, Martin Wijaya Hermanto, Bu Xu, Xin Yang, and Imma Nuella for actively participating discussion related to my research work and the help that they have rendered to me. I would also wish to thank technical and administrative staffs in the Chemical and Biomolecular Engineering Department 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 parents for their unconditional support, affection and encouragement, without which this research work would not have been possible. I also wish to thank my best partner, Ayano, for her understanding, continuous support and encouragement. Also I am greatly indebted to Dr. Iori Hashimoto for getting me interested in coming to Singapore.

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 controller design is one of the challenges in chemical industries. In this thesis, several data-based control strategies for nonlinear process control have been developed using the Just-in-Time Learning (JITL) modeling technique and Virtual Reference Feedback Tuning (VRFT) method, respectively. The main contributions of this thesis are as follows.

In the JITL modeling framework, which is capable of modeling the dynamic systems with a range of operating regimes, four adaptive control strategies are proposed, namely, a data-based linear quadratic regulator and integral compensator (LQI) design, an adaptive Internal Model Control (IMC) design, a self-tuning PID controller design, and a data-based Generalized Predictive Control (GPC) design. The traditional LQI controller design requires the availability of the state space model of the process, which is normally obtained from the first-principle model or closed-loop Kalman filter, which is either not available or too tedious to build in practice. To alleviate this drawback, a data-based LQI design method using JITL technique is developed. Next, by integrating the JITL into IMC design framework, an adaptive IMC design is developed. 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 updating algorithm derived based on the Lyapunov method to guarantee the convergence of JITL's predicted tracking error. In a similar setting for self-tuning PID controller design, a set of linear models obtained by the JITL provides the information required to adjust the parameters of PID controller by an updating algorithm derived by the Lyapunov method such that the JITL's predicted tracking error converges

asymptotically. Lastly, to extend the Generalized Predictive Control (GPC) design to nonlinear systems, a data-based GPC strategy based on the JITL is proposed. The local model obtained by the JITL at each sampling instant is used as the process model in GPC design where the optimal changes in the manipulated variable are determined by solving a quadratic optimization problem.

In the VRFT design framework, the design of feedback controller can be carried out directly based on the measured process input and output data without resorting to the identification of a process model. However, the existing results are restricted to the linear systems and their applications to nonlinear systems are limited. In this thesis, the relationship between the VRFT and the popular model-based design method, IMC design, is analyzed. Subsequently, to extend the VRFT design to nonlinear systems, two adaptive VRFT design methods are developed and their respective applications to adaptive PID controller design are discussed in detail.

Simulation results are presented to demonstrate that the proposed control strategies give better performances than their respective conventional counterparts.

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

$\mathbf{A}$, $\tilde{\mathbf{A}}$, $\mathbf{A}(k)$, $A(z^{-1})$	Coefficient matrix of state space model
A	Tuning parameter of reference closed-loop model
$\mathbf{B}$, $\tilde{\mathbf{B}}$, $\mathbf{B}(k)$, $B(z^{-1})$	Coefficient matrix of state space model
C	Controller
C_{A}, C_{Af}, C_{B}, C_{C}	Concentrations
C_{I}, $C_{\mathrm{I_{in}}}$	Initiator concentration
C_{m}, $C_{\mathrm{m_{in}}}$	Monomer concentration
$\mathbf{C}$	Coefficient vector of state space model
$\mathbf{C}_{\mathrm{u}}$, $\mathbf{c}_{\mathrm{f}}$	Matrix and vector for inequalities of input constraints
E_i, F_i	Coefficient matrix of Diophantine equation
e	Error between set-point and output
e_{m}	Error of reference model
e_r	Error between set-point and predicted output
F, F_{I}	Flow rate
$\mathbf{F}(k)$	Control law of LQI controller
f	Low-pass filter
f^*	Parameter of polymerization reaction
f_i	Past values of process input and output of GPC
H	Uncertainties of reference open-loop transfer function
J	Objective function
G	Process
G_i	Model parameter of GPC
G_{OL}	Reference open-loop transfer function
$\tilde{G}$	Process model
K_d, K_i, K_p	PID controller parameters
$\mathbf{K}(k)$	PID controller parameter vector
k_{I}, k_{f_m}, k_p, k_{T_c}, k_{T_d}	Parameters of polymerization reaction
k_0	Tuning parameter of reference transfer function

$k_1,\ k_2,\ k_3$	Kinetic parameters of van de Vusse reactor
$k_{\min},\ k_{\max}$	Number of minimum and maximum relevant data set
L	Level of reactor
l	Number of nearest neighbors
M_m	Molecular weight of monomer
M	Process model
$\mathbf{M}$	Diagonal matrix
N_{P}	Predictive horizon
N_{u}	Control horizon
n_d	Time-delay
P	Process
$\mathbf{P}_{l^*}$	Matrix of relevant data
$\mathbf{P}(k),\ \mathbf{P}_\infty$	Solution of the difference Riccati equation
Q	IMC controller
$\mathbf{Q}$	Weight matrix of quadratic function
q	Parameter of quadratic function
R	Scalar weight of quadratic function
r	Set-point
$\tilde{r}$	Virtual set-point
S	Sensitivity function
s_i	Similarity number
$T,\ T_i$	Reactor Temperatures
T	Reference closed-loop transfer function
u	Process input
$\tilde{u}$	Virtual input
V	Reactor volume
V	Lyapunov function
$\mathbf{W}_{l^*}$	Weight matrix of JITL
$w_1,\ w_2,\ w_3$	Parameters of self-tuning PID controller
$\mathbf{x}(k)$	State vector
$\mathbf{x}_i,\ \mathbf{x}_q$	database and query vector of JITL

$\bar{\mathbf{x}}(k)$	State vector including error
$y,\ \mathbf{y}(k)$	Process output
$y_l,\ y_{l^*}$	Relevant process output of JITL

Greek Symbols

$\alpha_1,\ \alpha_2,\ \beta_1, \beta_2$	Coefficients of ARX model
γ	Positive constant of Lyapunov function
Δt	Sampling time
$\eta,\ \eta_i$	Learning rate
κ	weight parameter of JITL
λ	IMC filter time constant
μ	Weight parameter
θ_i	Angle between $\Delta\mathbf{x}_i$ and $\Delta\mathbf{x}_q$
ϖ	Tuning parameter of reference model
χ	Design parameter
ρ	Vector of discrete time-transfer function
ζ	Update parameter
Ω	Matrix of learning rate
τ_i	Model parameter

Abbreviations

AIBN	Azo-bis-isobutyronitrile
ARX	Autoregressive exogenous
CSTR	Continuous stirred tank reactor
FA	Function approximator
GPC	Generalized predictive control
IFT	Iterative feedback tuning
IMC	Internal model control
JITL	Just-in-time learning
LQI	Linear quadratic regulator and integral compensator

MAE	Mean absolute error
MMA	Methyl methacrylate
MSE	Mean squared error
MPC	Model predictive control
NAMW	Number average molecular weight
NN	Neural network
PID	Proportional-integral-derivative
RLS	Recursive least square
VID^2	Virtual input direct design
VRD^2	Virtual reference direct design
VRFT	Virtual reference feedback tuning

Chapter 1

Introduction

1.1 Motivation

With the market competition getting more intense than before, growing demands for improving performance of process have stimulated engineers and researchers to develop more efficient and reliable techniques for process control. These techniques are useful not only for profits but also for safety, product specification and environment for chemical plants. Product quality and quantity must be accepted according to the customer demands for profits. Safety and environmental problems must be considered for the workers and residents in and around the plants. For these purposes, the study of process control has been becoming more important for the development of chemical industries.

In process industries, hundreds or thousands of variables, such as flow rates, temperatures, pressure, level and compositions, are routinely measured and automatically recorded to build the historical database, which can be utilized for the purpose of process control, optimization and monitoring. Despite that significant

benefits may be gained from the database, it is not a trivial task to extract useful information and knowledge from the database. Therefore, most chemical processes face the "data rich information poor" problem. Although an accurate process model is essential to design high performance controller, the construction of first-principles models is usually time-consuming and costly. Moreover, model-based controller design by incorporating these models would lead to complex controller structure. Thus, if one desires a simple controller, e.g. PID controller, a non-trivial controller reduction procedure needs to be performed. An alternative is the data-driven modeling methods which have been proposed for nonlinear system modeling (Pearson, 1999, Nelles, 2001) in the last two decades, for example artificial neural networks (ANN) and neuro-fuzzy model (Nelles, 2001). However, when dealing with large sets of data, these approaches are less attractive because of the difficulties in specifying model structure and the complexity of the associated optimization problems. To alleviate the aforementioned problems, Aha et al. (1991) developed instance-based learning algorithms for modeling nonlinear systems. This approach is inspired by the ideas from local modeling and machine learning techniques. Different variants of instance-based learning are also developed in the literature, e.g. locally weighted learning (Atkeson et al., 1997a, 1997b) and just-in-time learning (JITL) techniques (Bontempi et al., 1999, 2001). JITL has no standard learning phase because it merely stores the data in the database and the computation is not performed until the arrival of a query data. Furthermore, JITL constructs local approximation of the dynamic systems characterized by the current query data, and thus low-order model is usually employed in the JITL technique. Another advantage of the JITL is its inherent adaptive nature, which is achieved by storing the current measured data

into the database (Nelles, 2001). Therefore, this motivates the proposed research to develop data-based advanced control strategies using the JITL modeling technique in this thesis.

Another attractive data-based method for controller design is to design controller directly based on the measured process input and output data without resorting to the identification of a process model. For example, Spall and Cristion (1998) presented a stochastic design framework in which the controller is represented by a function approximator (FA), like a polynomial or a neural network, whose parameters are determined stochastically based on the process measurement, rather than a process model. Another direct design method is the iterative feedback tuning method developed by Hjalmarsson et al. (1994). However, this method requires considerable computational time to obtain a solution with a risk of being a local optimum in the proposed minimization problem, not to mention its dependence on the trial and error procedure for initialization. Furthermore, its computation needs unbiased estimates of some variables, which impose much more stringent requirements on the experiment. As a result, the experiment required for IFT is typically complicated. To overcome this problem, the virtual input direct design method (VID^2, Guardabassi and Savaresi, 1997; Savaresi and Guardabassi, 1998) was the first direct controller design method without any gradient calculation. Campi et al. (2000) improved and reorganized the idea of VID^2 and renamed the new method as the virtual reference feedback tuning (VRFT) method. Guardabassi and Savaresi (2000) also developed their new version called virtual reference direct design (VRD^2) which basically follows the same design principles as VRFT. The VRFT design and its variants share a common feature that controller parameters are obtained off-line

by solving a quadratic optimization problem based on a set of process input and output data. However, these methods are developed for linear systems and their applications to nonlinear systems are limited. Therefore, attempts will be made to extend the VRFT design to nonlinear systems as well in this thesis.

1.2 Contribution

In this thesis, data based methods for nonlinear process control are developed using the JITL modeling technique and VRFT design method, respectively. The main contributions of this thesis are as follows.

(1) Linear Quadratic Regulator and Integral Compensator (LQI) design using the JITL technique: Traditional LQI design requires the state space model of the process, which is normally obtained from the first-principle model or closed-loop Kalman filter, which is either not available or too tedious to build in practice. To overcome this problem, data-based LQI design using JITL technique will be investigated.

(2) Internal Model Controller (IMC) design using the JITL technique: IMC is a powerful controller design strategy for the open-loop stable dynamic systems. However, the performance of IMC controller will degrade or become unstable when it is applied to nonlinear processes with a range of operating conditions. In the proposed IMC design, controller parameters are updated not only based on the information provided by the JITL, but also its filter parameter is adjusted online by an updating algorithm derived based on the Lyapunov method to guarantee the convergence of the JITL's predicted tracking error.

(3) Self-tuning PID design using the JITL technique: The well-known PID controllers are still the most adopted controllers in the process industries. However, chemical processes often exhibit nonlinearities and contain high-order dynamics, all of which can deteriorate the performance of PID controllers. In the proposed design, a set of linear models obtained by the JITL provides the information required to adjust the parameters of PID controller by the updating algorithm derived based on the Lyapunov method such that the JITL's predicted tracking error converges asymptotically.

(4) Nonlinear Generalized Predictive Controller (GPC) design using the JITL technique: Model Predictive Controller (MPC) is now widely recognized as a powerful methodology to address industrially important control problems. However, most MPC techniques, like GPC, are based on linear models and thus not very well-suited for the control of nonlinear systems. In this thesis, the extension of GPC design to nonlinear system is attempted by using the JITL technique.

(5) Adaptive PID controller designs by the adaptive VRFT methods: VRFT design can be applied to determine the parameters of a PID controller by using a set of process input and output data without resorting to the identification of a process model. Although it is an attractive alternative to the popular model-based controller design methods, the existing results are restricted to the linear systems. In the proposed research, the connection between VRFT and IMC designs is firstly analyzed. Two adaptive VRFT design procedures, which are tailor-made for adaptive PID designs, are proposed.

1.3 Thesis Organization

This thesis is organized as follows. Chapter 2 comprises the literature review of nonlinear process control. In Chapter 3, a new optimal controller design using JITL technique for nonlinear process control is described. By incorporating the JITL into IMC and PID designs, an adaptive IMC controller and a self-tuning PID controller for nonlinear process control are developed in Chapters 4 and 5, respectively. In Chapter 6, a generalized predictive control based on the JITL technique is developed. To extend the existing VRFT design methods to nonlinear systems, two adaptive VRFT design methods are developed in Chapters 7 and 8, respectively. Finally, the general conclusions from the present work and suggestions for future work are given in Chapter 9.

Chapter 2

Literature Review

This chapter provides an overview of the current progress of several controller design methods. Furthermore, just-in-time learning (JITL) modeling technique and its application in various proposed data-based control strategies will be highlighted as well.

2.1 Just-in-Time Learning Modeling Technique

Process models are fundamentally important for process control because controller performance is dependent on the accuracy of process models. However, it is difficult to obtain an accurate first principles model for most of chemical processes because of the lack of complete physicochemical knowledge. To alleviate this drawback, several empirical and black-box models have been developed. For example, neural networks, fuzzy models, fuzzy neural networks, and local model networks have been investigated and developed in the literature. However, one fundamental limitation of these types of modeling approaches 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 model structure (e.g. number of hidden neurons in the neural network models and number of fuzzy rules in fuzzy models) and model parameters may need to be modified simultaneously. Evidently, this process is not only time-consuming but also it will interrupt the plant operation, if these models are used in controller design.

An alternative approach for nonlinear system modeling, just-in-time learning (JITL), is developed recently. The JITL is attractive not only because of its prediction capability for nonlinear processes but also its inherently adaptive nature. Aha et al. (1991) developed instance-based learning algorithms for modeling nonlinear systems. This approach is inspired by ideas from local modeling and machine learning techniques. Subsequent to Aha's work, different variants of instance-based learning are developed, such as locally weighted learning (Atkeson et al., 1997a, 1997b) and JITL (Bontempi et al., 1999, 2001). JITL has no standard learning phase because it merely stores the data in the database and the computation is not performed until a query data arrives. Furthermore, JITL constructs local approximation of the dynamic systems 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. In addition, JITL is inherent adaptive in nature, which is achieved by storing the current measured data into database (Nelles, 2001). To achieve better predictive performance of JITL algorithm, Cheng and Chiu (2004) recently proposed an enhanced JITL algorithm by using a new similarity measure that combines the conventional dis-

tance measure with the angular relationship. In the following, the JITL algorithm developed in Cheng and Chiu (2004), which is used in this thesis, is described.

There are three main steps in the JITL to compute model output corresponding to the query data: (i) relevant data samples in the database are searched to match the query data by some nearest neighborhood criterion; (ii) a low-order local model is built based on the relevant data; (iii) model output is calculated based on the local model and the current query data. When the next query data is available, a new local model will be built by repeating the aforementioned procedure.

As a simple low-order model is usually employed by the JITL, without the loss of generality, consider the following second-order ARX model:

$$\hat{y}(k) = \alpha_1^k y(k-1) + \alpha_2^k y(k-2) + \beta_1^k u(k-1) \tag{2.1}$$

where $\hat{y}(k)$ is the predicted output by the JITL at the k-th sampling time, $y(k-1)$ and $u(k-1)$ are the output and manipulated variables at the $(k-1)$-th sampling time, α_1^k, α_2^k and β_1^k are the model coefficients at the k-th sampling time.

Define regression vector for the ARX model given in Eq. (2.1) as

$$\mathbf{x}_k = \begin{bmatrix} y(k-1) & y(k-2) & u(k-1) \end{bmatrix} \tag{2.2}$$

To apply the JITL technique, its database is initially constructed by using process input and output data obtained around the nominal operating condition. Subsequently, this database can be updated during its on-line application, for example in the controller design, when modeling error between the process output and predicted output by the JITL is greater than the pre-specified threshold, by simply adding the current process data into the database. In those cases, the current process data is considered as new data that is not adequately represented by the present

database and is thus added to the database to improve its prediction accuracy for new operating region where the process data may not be available to construct the initial database for JITL.

Suppose that the present database of JITL consists of N process data $(y(i),\, \mathbf{x}_i)_{i=1\sim N}$, given a query data $\mathbf{x}_q$, the objective of JITL is to obtain the local ARX model of the nonlinear systems by focusing on the relevant region around the current operating condition. The first step is to select the relevant regression vectors from the database that resemble the query data. To do so, the following similarity measure, s_i, is considered.

$$s_i = \kappa\sqrt{e^{-\|\mathbf{x}_q-\mathbf{x}_i\|^2}} + (1-\kappa)\cos(\theta_i), \quad \text{if } \cos(\theta_i) \geq 0 \tag{2.3}$$

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

After all s_i are computed by Eq. (2.3), for each $l \in [k_{\min}\ k_{\max}]$, where $k_{\min}$ and $k_{\max}$ are the pre-specified minimum and maximum numbers of relevant data, the relevant data set $(\mathbf{y}_l,\, \Phi_l)$ is constructed by selecting the l most relevant data $(y_i,\, \mathbf{x}_i)$ corresponding to the largest s_i to the l-th largest s_i. The leave-one-out cross validation test (Myers, 1990) is then conducted and the validation error is calculated. Upon the completion of the above procedure, the optimal l, l^*, is determined by that giving the smallest validation error. Subsequently, the predicted output for query data is calculated as $\mathbf{x}_q^{\mathrm{T}}(\mathbf{P}_{l^*}^{\mathrm{T}}\mathbf{P}_{l^*})^{-1}\mathbf{P}_{l^*}^{\mathrm{T}}\mathbf{W}_{l^*}\mathbf{y}_{l^*}$, where $\mathbf{P}_{l^*}^{\mathrm{T}} = \mathbf{W}_{l^*}\Phi_{l^*}$ and $\mathbf{W}_{l^*}$ is a diagonal matrix with entries being the first l^* largest s_i.

2.2 Controller Designs for Nonlinear Processes

The majority of chemical and biochemical industries is inherently nonlinear, however most controller design techniques are based on linear control techniques to deal with such systems. 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 process input and output data. In addition, the theories for the stability analysis of linear control systems are quite well developed so that linear control techniques are widely accepted. However, owing to the nonlinear nature of most chemical processes, 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, five control strategies, i.e. linear quadratic regulator and Integral compensator (LQI) controller design as an example for optimal control, adaptive control, nonlinear internal model control, nonlinear model predictive control, and direct data-based control capable of providing the improved performance for nonlinear systems are reviewed.

2.2.1 LQI controller design method

The development of modern control concepts can be traced back to the work of Kalman in the early 1960's, who sought to determine when a linear control system can be said to be optimal (Kuo, 1980; Lewis and Syrmos, 1995; Ogata, 1997). Kalman studied state-space model design and optimal control strategy, which is the

well-known linear quadratic regulator (LQR) design based on the minimization of a quadratic objection function. Based on the LQR techniques, Fujii (1987) developed the inverse linear quadratic regulator (I-LQ). Ikeda and Suda (1988) modified Fujii's method by proposing LQI controller design that has an integrator compensator to eliminate steady-state offset within LQR frameworks. However, traditional LQI design depends on the state space process model of the process constructed from the first-principle model or closed-loop Kalman filter (Ebihara et al., 1988). Hashimoto et al. (2000) reported the application of LQI design for nonlinear system based on the discrete models obtained by the successive linearization of first-principle model, which is either not available or too tedious to build in practice. To alleviate the drawbacks of model-based LQI design methods, a data-based LQI design by using JITL technique is considered in this thesis and will be developed in Chapter 3.

2.2.2 Adaptive control

Research in adaptive control has a long and vigorous history. The development of adaptive control started in the 1950's with the aim of developing adaptive flight control systems. With the progress of control theories and computer technology, various adaptive control methodologies were proposed for process control in the last three decades. Åström (1983), Seborg et al. (1986), and Åström and Wittenmark (1995) gave detailed 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 of 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

2.1. 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 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 in process industry (Åström 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 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 control. The general strategy of this controller is to estimate model parameters on-line and then adjust the controller settings based on current parameter estimate (Åström, 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 settings are then updated. The whole control strategy can be divided into three steps: (i) information gathering of the present process behavior; (ii) control performance criterion optimization; and (iii) adjustment of the controller parameters. The first step implies the continuous determination of the current process condition based on measurable process input and output data and appropriate modeling approaches selected to identify the model parameters. Various types of model identification can be distinguished depending on the information gathered and the method of estimation. The last two steps calculate 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).

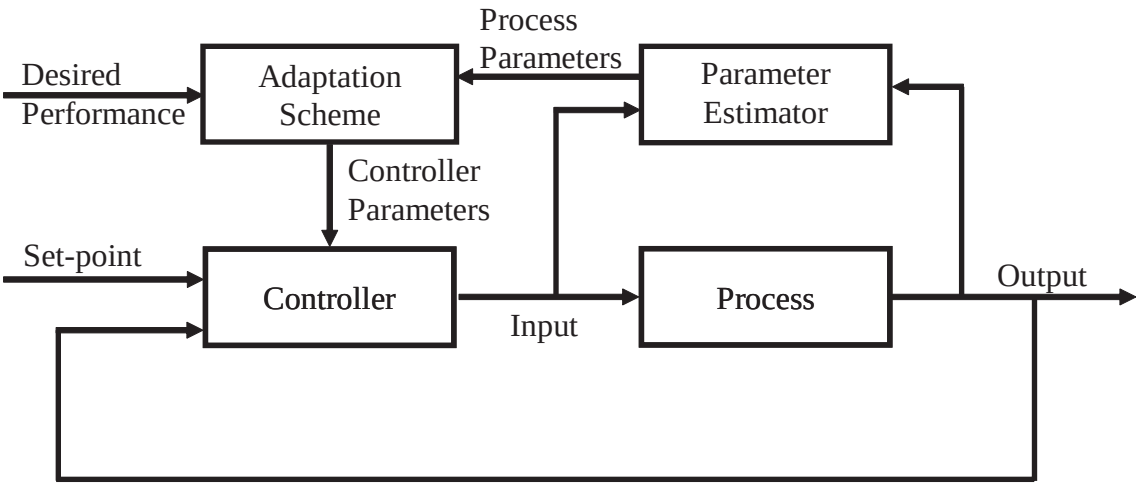

Figure 2.1: Block diagram of adaptive control scheme

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 (NN) 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. 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.

The PID controllers have received widespread use in the process industries pri-

marily because of its simple structure, ease of implementation, and robustness in operation. Due to these advantages, several adaptive PID controller designs have been developed in recent years. For example, Riverol and Napolitano (2000) proposed an adaptive PID controller whose parameters are adjusted on-line by a neural network, while Chen and Huang (2004) designed adaptive PID controller based on the instantaneous linearization of a neural network model. Altinten et al. (2004) applied the genetic algorithm to the optimal tuning of a PID controller on-line. Bisowarno et al. (2004) applied two adaptive PI control strategies for reactive distillation. Andrasik et al. (2004) made use of a hybrid model consisting of a neural network and a simplified first-principle model to design a neural PID-like controller. Yamamoto and Shah (2004) developed an adaptive PID controller using recursive least squares for on-line identification of multivariable system. Shahrokhi and Baghmisheh (2005) designed an adaptive IMC-PID controller based on the local models estimated by the recursive least squares method to control a fixed-bed reactor. Similar approaches for adjusting PID controller parameters on-line were investigated based on the multiple linearized models obtained by factorization algorithm and lazy learning identification method at each sampling instant (Ho et al., 1999; Alpbaz et al., 2006; Pan et al., 2007). In these works, basically, the parameters of the process model are updated with respect to the current process condition and then PID parameters are computed by the corresponding adaptation algorithm and implemented. However, these adaptation algorithms employed in the previous results are inadequate to address the convergence of the predicted tracking error. To overcome this problem, a self-tuning PID controller design based on a set of linear models obtained by the JITL and a self-tuning PID algorithm derived by the Lya-

punov method to guarantee the convergence of tracking error will be developed in Chapter 5.

2.2.3 Nonlinear internal model control

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 by using the IMC two-step design procedure. Moreover, plant and 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 state-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 and model mismatch. However, their

method relied on the availability of a nonlinear state-space model, which may be time-consuming and costly to obtain.

Another popular design method for implementing nonlinear IMC schemes is based on the neural networks. In the earlier methods given in Bhat and McAvoy (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 then used as the IMC model, while another NN was trained to learn the inverse dynamics of the process and was employed as the nonlinear IMC 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 and model mismatch. To ensure offset-free performance, 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 robustness filter with a single tuning parameter. In this control strategy, a numerical inversion of neural network process model was proposed instead of training neural networks on the process inverse. Aoyama et al. (1995) proposed a method using control-affine neural network models. Two neural networks were used in this approach: one for the model of the bias or drift term, and one for the model of the steady-state gain. As the process is approximated by a control-affine model, the inversion of process model is simply obtained by algebraically inverting the process model.

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, a promising approach has been proposed to yield a flexible nonlinear model inversion (Doyle et al., 1995; Harris and Palazoglu, 1998). This controller synthesis scheme based on partitioned model inverse retains the original spirit and characteristics of conventional (linear) IMC while extending its capabilities to nonlinear systems. When implemented as part of the control law, the nonlinear controller consists of a standard linear IMC controller augmented by an auxiliary loop of nonlinear "correction". The fact that only a linear inversion is required in the synthesis of this controller is the most attractive feature of this scheme. 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., 1996). 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 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. This limitation restricts the implementation of these models in practice (Xiong and Jutan, 2002).

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 the first experimental application of this control

strategy using NN as a nonlinear model and a linear ARX model. However, one fundamental limitation of these global approaches for modeling is that the on-line update of these models is not straightforward when the process dynamics are moved away from the nominal operating space. Evidently, this will interrupt the plant operation when these models are used in the controller design.

To alleviate the aforementioned problems, the JITL-based adaptive IMC design strategy will be investigated in Chapter 4. By taking advantage of simple models employed in JITL, the model inverse can be readily obtained for IMC design at each sampling instant. Therefore, the IMC control strategy can be extended to the nonlinear processes in a straightforward manner without scarifying the simplicity of the linear IMC design.

2.2.4 Constrained control

Virtually all practical control systems are subject to hard constraints on their manipulated inputs. Typically, these constraints arise due to the physical limitations inherent in the capacity of control actuators, e.g. bounds on the magnitude of valve opening. An important limitation imposed by the input constraints is that they can lead to degradation of the performance of closed-loop system and even loss of stability. While there are many nonlinear and robust controller design methods developed, they do not guarantee the control action to stay within the working range of control actuators because the presence of input constraints is not explicitly taken into account at the stage of controller design. The problems caused by input constraints have consequently motivated many recent studies on the dynamics and control of chemical processes subject to input constraints, e.g. the design of

"anti-windup" schemes in order to prevent excessive performance deterioration of an already designed controller when the input saturates (Kendi and Doyle, 1997, 1998; Kapoor and Daoutidis, 1999; Kothare et al., 1994)

Model predictive control (MPC) based on linear models, for example dynamic matrix control (Cutler and Ramaker, 1979), quadratic dynamic matrix control (Garcia and Morshedi, 1986), and generalized predictive control (Clarke et al., 1987), has gained wide-spread acceptance as an advanced control strategy in chemical process industries. This is primarily due to their ability to handle process constraints, time delays, and multivariable systems in a unified design framework. The general strategy of MPC algorithm is to utilize a model to predict the future output trajectory of the process and compute future control action by solving a minimization problem with suitable objective function that includes the difference between the predicted output trajectory and reference trajectory and a penalty term on the future control actions. Therefore, the effectiveness of MPC relies heavily on the availability of a reasonably accurate process model. As many chemical processes are highly nonlinear and may be operated in a range of operating points, it is clear that MPC algorithms based on linear process models can result in poor control performance. As a result, various variants of MPC techniques have been studied and extended to nonlinear systems (Bequette, 1991; Henson and Seborg, 1997; Henson, 1998; Lee, 2000; Mayne, 2000; Zheng, 2000). For example, Berber and Coskun (1996) studied nonlinear MPC (NMPC) on an industrial low density polyethylene reactor. Seki et al. (2001) implemented a two tier control algorithm. The first tier was formulated by successive linearization of a nonlinear first-principle process model. In the second tier, control actions were determined by solving a quadratic program

problem formulated by a linear model that is obtained by linearizing around the control trajectory. Piche et al. (2000) presented a neural network technique for developing a nonlinear dynamic model for NMPC design. They used step test data for building linear dynamic model and historical data for a nonlinear steady-state model. Nonlinear dynamic model was then formed by combining the aforementioned two models. Venkateswarlu and Gangiah (1997) utilized a recursive least squares (RLS) algorithm to update the local model in a nonlinear generalized predictive control strategy. However, the RLS algorithm can produce poor estimates of system parameters if the online process input and output data do not meet excitation conditions. Another popular nonlinear MPC technique by incorporating empirical models like neural networks (Saint-Donat et al., 1991; Pottmann and Seborg, 1997; Chu et al., 2003), fuzzy models (Kavsek-Biasizzo et al., 1997; Fischer et al., 1998; Abonyi et al., 2000; Mahfouf et al., 2000), fuzzy neural networks (Lu and Tsai, 2007), and local model networks (Prasad et al., 1998) have been investigated and developed in the literature. However, the use of neural network in nonlinear MPC design is computationally demanding due to the on-line optimization required to compute the control signals. For fuzzy models and local model networks, the problem of how to partition the operating regimes remains an ad-hoc procedure and therefore a prior knowledge of the processes, which may not be readily accessible in most practical cases, has to be exploited for the determination of the model structure. As discussed previously, another fundamental limitation of these modeling approaches is the difficulty to update these models on-line when the process dynamics are moved away from the nominal operating space.

To curtail the aforementioned problem encountered by the global models, a gen-

eralized predictive control (GPC) strategy based on the JITL technique is proposed in Chapter 6. The computational burden is reduced by modeling the nonlinear systems by a set of local models obtained on-line by the JITL. The current local model at each sampling instant is treated as the process model in the GPC design where the optimal changes in the manipulated variable are determined by solving a quadratic optimization problem formulated in the GPC design framework.

2.3 Direct Data-Based Controller Design Methods

Designing controllers directly based on a set of measured process input and output data, without resorting to the identification of a process model, is an attractive option for process control application. Such 'direct' data-based design techniques are conceptually more natural than model-based designs where the controller is designed on the basis of an estimated model of the process, because the former directly targets the final goal of tuning the parameters of a given class of controllers. However, despite the appeal of direct data-based design methods, very few genuine direct design techniques have been proposed in literature

Hjalmarsson et al. (1994) developed iterative feedback tuning (IFT) method with promising result for real application (1998). However, IFT may require considerable computational time to obtain a solution with a risk of being a local optimum in the proposed minimization problem, not to mention its dependence on the trial and error procedure for initialization. Furthermore, its computation needs unbiased estimates of some variables, which impose much more stringent requirements on the

experiment. As a result, the experiment required for IFT is typically complicated.

Spall and Cristion (1998) proposed a stochastic approach for adaptive control using a function approximator (FA) to calculate the action needed from the controller. FA can be a polynomial or an artificial neural network, whose parameters are updated repeatedly in accordance with the minimization of a cost function. However, since a plant model is not available, the gradient of this cost function has to be evaluated by simultaneous perturbation stochastic approximation instead of quadratic methods. Thus, the computational burden of this method is very high due to the iterations and the convergence of the trained parameters may not be guaranteed.

To alleviate the aforementioned drawbacks, Campi and Lecchini (2000, 2002) proposed the virtual reference feedback tuning method (VRFT). VRFT stems from the idea of virtual input direct design (VID^2) (Guardabassi and Savaresi, 1997; Savaresi and Guardabassi, 1998), but in a better-organized form. This methodology is simple and directly calculates the feedback controller parameters from the available process input and output data without the need of model identification. Under this tuning framework, only the specification of desired reference model is required. Nakamoto (2005) extended this controller design technique to multivariable systems and showed a chemical process application.

As the existing results on VRFT design are restricted to linear systems, to extend its application to nonlinear systems, two adaptive VRFT designs will be discussed and developed in Chapters 7 and 8, respectively.

Chapter 3

Data-Based LQI Controller Design Using the JITL Technique

3.1 Introduction

Process models are fundamentally important for process control because the performance of advanced control design methods is intimately dependent on the availability of accurate process models. However, most chemical processes are multivariable and nonlinear in nature and their dynamics can be time-varying, therefore first-principle models are often unavailable or too costly and time-consuming to build due to the lack of complete physicochemical knowledge of chemical processes. An alternative approach is to develop data-based methods to build a model from process data collected in industrial processes.

The development of modern control theory can be traced back to the work of Kalman (Lewis and Syrmos, 1995), who proposed Linear Quadratic Regulator (LQR) design based on the minimization of a quadratic objective function. Within

the LQR design framework, Ikeda and Suda (1988) proposed LQI controller design embedded with an integrator compensator to eliminate steady-state offset. However, traditional LQI design depends on the state space model of the process constructed from the first-principle model or closed-loop Kalman filter (Ebihara et al., 2000; Hashimoto et al., 2000), which is either not available or it is too tedious to build in practice. An alternative LQI design is based on an empirical ARX model with on-line parameter adaptation by the standard recursive procedure, where a forgetting factor is adopted to enable the recursive identification algorithm to handle both nonlinear and time-varying features of dynamic systems by giving more weight to the most recent process data. However, it was discussed by Bontempi et al. (1999) that recursive identification algorithm may suffer the problem of data interference, also known as stability-plasticity dilemma.

Motivated by the advantages of the JITL algorithm and to alleviate the drawbacks of model-based LQI design methods, data-based LQI design by using JITL technique is proposed in this chapter. Literature examples are used to illustrate the proposed design method and a comparison with its conventional counterparts is made.

3.2 Data-Based LQI Design

In the proposed data-based LQI design, the following second-order ARX model is employed in the JITL algorithm at each sampling time:

$$y(k) = \alpha_1 y(k-1) + \alpha_2 y(k-2) + \beta_1 u(k-1) + \beta_2 u(k-2) \tag{3.1}$$

where $u(k)$ is the manipulated variable at the k-th sampling time, $y(k)$ is the output variable at the k-th sampling time, α_i, β_j are the model coefficients $(i, j = 1, 2)$. It is evident that a first-order ARX model is obtained by setting $\alpha_2 = \beta_2 = 0$.

Equation (3.1) can be represented by the following state space equation.

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \\ y(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \tag{3.2}$$

where

$$\begin{aligned} \mathbf{x}(k) &= \begin{bmatrix} y(k) & y(k-1) & u(k-1) \end{bmatrix}^{\mathrm{T}} \\ \mathbf{A} &= \begin{bmatrix} \alpha_1 & \alpha_2 & \beta_2 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} \beta_1 & 0 & 1 \end{bmatrix}^{\mathrm{T}} \\ \mathbf{C} &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \end{aligned}$$

Equation (3.2) can be rewritten as

$$\Delta\mathbf{x}(k+1) = \mathbf{A}\Delta\mathbf{x}(k) + \mathbf{B}\Delta u(k) \tag{3.3}$$

$$\Delta y(k) = \mathbf{C}\Delta\mathbf{x}(k) \tag{3.4}$$

where

$$\Delta\mathbf{x}(k) = \mathbf{x}(k) - \mathbf{x}(k-1) \tag{3.5}$$

$$\Delta u(k) = u(k) - u(k-1) \tag{3.6}$$

$$\Delta y(k) = y(k) - y(k-1) \tag{3.7}$$

Subsequently, the following equation can be easily drawn:

$$\begin{aligned}\tilde{\mathbf{x}}(k+1) &= \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{CA} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}(k) \\ e(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ -\mathbf{CB} \end{bmatrix} \Delta u(k) \\ &= \tilde{\mathbf{A}}\tilde{\mathbf{x}}(k) + \tilde{\mathbf{B}}\Delta u(k) \end{aligned} \tag{3.8}$$

$$e(k) = \tilde{\mathbf{C}}\tilde{\mathbf{x}}(k) \tag{3.9}$$

where $\tilde{\mathbf{x}}(k) = \begin{bmatrix} \Delta\mathbf{x}(k) & e(k) \end{bmatrix}^{\mathrm{T}}$, $e(k) = r(k) - y(k)$, and $\tilde{\mathbf{C}} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$.

The objective of LQI design is to find the optimal input given by

$$\Delta u(k) = \mathbf{F}(k)\tilde{\mathbf{x}}(k) \tag{3.10}$$

such that the following quadratic function is minimized subject to the system described by Eq. (3.9).

$$J = \sum_{k=0}^{\infty} \{\tilde{\mathbf{x}}(k+1)^{\mathrm{T}}\mathbf{Q}\tilde{\mathbf{x}}(k+1) + \Delta u(k+1)R\Delta u(k+1)\} \tag{3.11}$$

where $\mathbf{Q}$ is a positive semidefinite symmetric weighting matrix and R is a scalar weight.

The optimal feedback gain matrix $\mathbf{F}(k)$ in Eq. (3.10) is calculated as

$$\mathbf{F}(k) = -\left(\tilde{\mathbf{B}}^{\mathrm{T}}\mathbf{P}\tilde{\mathbf{B}} + R\right)^{-1}\tilde{\mathbf{B}}^{\mathrm{T}}\mathbf{P}\tilde{\mathbf{A}} \tag{3.12}$$

where $\mathbf{P}$ is the solution of the following difference Riccati equation:

$$\mathbf{P} = \mathbf{Q} + \tilde{\mathbf{A}}^{\mathrm{T}}\mathbf{P}\tilde{\mathbf{A}} - \tilde{\mathbf{A}}^{\mathrm{T}}\mathbf{P}\tilde{\mathbf{B}}\left(R + \tilde{\mathbf{B}}^{\mathrm{T}}\mathbf{P}\tilde{\mathbf{B}}\right)^{-1}\tilde{\mathbf{B}}^{\mathrm{T}}\mathbf{P}\tilde{\mathbf{A}} \tag{3.13}$$

One remark on the design of $\mathbf{Q}$ is given as follows. Considering the following first-order plus time-delay reference model:

$$M(z^{-1}) = \frac{(1-\varpi)z^{-N-1}}{1-\varpi z^{-1}} \tag{3.14}$$

where α is the tuning parameter related to the speed of closed-loop response and N is the time-delay of the process.

In order to make LQI design include the poles of the reference model, the following formulation of $\mathbf{Q}$ is employed:

$$\mathbf{Q} = \left[\tilde{\mathbf{C}}\left(\mathbf{I} - \frac{1}{\varpi}\tilde{\mathbf{A}}\right)\right]^{\mathrm{T}}\left[\tilde{\mathbf{C}}\left(\mathbf{I} - \frac{1}{\varpi}\tilde{\mathbf{A}}\right)\right] \tag{3.15}$$

The implementation of the proposed data-based LQI design is summarized as follows:

Step 1 Obtain initial database for the JITL algorithm;

Step 2 Design ϖ and R;

Step 3 At each sampling instant, obtain the linear model given in Eq. (3.1) by using the most current process data and the JITL algorithm. The database is updated by the current process data if the absolute value of difference between the predicted output and the process output is larger than 5% of the process output.

Step 4 Calculate $\Delta u(k)$ by Eqs. (3.10) and (3.12) and go to Step 3.

3.3 Examples

Example 1 Consider the following Hammerstein process given by

$$y(k+1) = 0.8y(k) + 0.4\tanh(2u(k)) \tag{3.16}$$

To proceed with the proposed LQI design, a first-order ARX model is used. Therefore, the following state space equation is considered.

$$\begin{aligned} \Delta x(k+1) &= \alpha_1 \Delta y(k) + \beta_1 \Delta u(k) \\ \Delta y(k) &= \Delta x(k) \end{aligned} \tag{3.17}$$

where α_1 and β_1 are calculated at each sampling instant by the JITL algorithm with the following parameters: $\kappa = 0.99$, $k_{\min} = 6$ and $k_{\max} = 60$. The initial database for the JITL algorithm is built from process data generated by introducing uniformly random steps with distribution of [-0.05 0.05] in the process input $u(k)$.

Consequently, the following augmented state space equation can be formed.

$$\begin{aligned} \tilde{\mathbf{x}}(k+1) &= \begin{bmatrix} \alpha_1 & 0 \\ -\alpha_1 & 1 \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ e(k) \end{bmatrix} + \begin{bmatrix} \beta_1 \\ -\beta_1 \end{bmatrix} \Delta u \\ &= \tilde{\mathbf{A}} \tilde{\mathbf{x}}(k) + \tilde{\mathbf{B}} \Delta u(k) \end{aligned} \tag{3.18}$$

The quadratic function given in Eq. (3.11) is then minimized subject to the system described in Eq. (3.18). The optimal feedback gain $\mathbf{F}(k)$ is computed by Eqs. (3.12) and (3.13), and consequently $\Delta u(k)$ is obtained by Eq. (3.10).

For comparison purposes, the following analytical local model obtained by successive linearization of Eq. (3.16) is employed in the LQI design

$$\Delta y(k+1) \cong 0.8 \Delta y(k) + 0.4 \frac{3.2}{\cosh^2(2u(k-1))} \Delta u(k) \tag{3.19}$$

To evaluate the performance of these two LQI designs, the successive step changes in the set-point as depicted in Figure 3.1 are considered. The resulting performance indices J of two LQI designs are comparable as shown in Tables 3.1 and 3.2 for different values of tuning parameters R and ϖ. Likewise, the respective mean-squared errors (MSE) of the tracking performance are given in Tables 3.3 and 3.4.

It can be seen that the proposed data-based LQI design yields a slightly smaller tracking error than that achieved by LQI design using the analytical model given in Eq. (3.19). For illustration purposes, the closed-loop responses of two LQI designs with $R = 0.4$ and $\varpi = 5$ are compared in Figure 3.1.

To test the robustness of the proposed LQI design, both process inputs and outputs are corrupted by 4% Gaussian white noise. As shown in Figure 3.2, the proposed LQI design can yield reasonably good control performance in the presence of process noise. Furthermore, Figure 3.3 illustrates the disturbance rejection capability of the proposed design, where step disturbances of -0.2 and $+0.2$ are introduced at $t = 30$ and 60, respectively. It is clear that the proposed data-based LQI design achieves comparable control performance as that obtained by its counterpart based on the analytical model.

Lastly, the proposed LQI design is compared with LQI design based on a first-order ARX model with parameter adaptation by the recursive identification procedure. As can be seen from Figure 3.4, both designs exhibit comparable performance for smaller set-point changes, i.e. the first three successive set-point changes with magnitudes of 1 and -1. However, the proposed LQI design shows better control performance for large set-point changes, as evidenced by the last two consecutive set-point changes with magnitudes of 2 and -2.

Table 3.1: Index J for data-based LQI design

		ϖ			
		4	5	10	50
	0.2	15.1	16.8	20.6	24.7
R	0.4	14.8	16.2	19.7	23.4
	0.8	15.4	17.3	20.2	23.3
	1.2	16.0	17.6	20.8	24.0

Table 3.2: Index J for LQI design based on analytical model

		ϖ			
		4	5	10	50
	0.2	13.9	15.5	19.0	22.0
R	0.4	14.8	16.1	19.3	22.2
	0.8	16.1	17.1	20.4	23.3
	1.2	16.3	18.1	21.4	24.3

Table 3.3: Tracking error of data-based LQI design

		ϖ			
		4	5	10	50
	0.2	0.154	0.155	0.159	0.167
R	0.4	0.149	0.148	0.149	0.152
	0.8	0.154	0.152	0.151	0.151
	1.2	0.158	0.156	0.155	0.152

Table 3.4: Tracking error of LQI design based on analytical model

		ϖ			
		4	5	10	50
	0.2	0.159	0.161	0.164	0.166
R	0.4	0.166	0.164	0.165	0.166
	0.8	0.177	0.172	0.171	0.171
	1.2	0.178	0.179	0.177	0.176

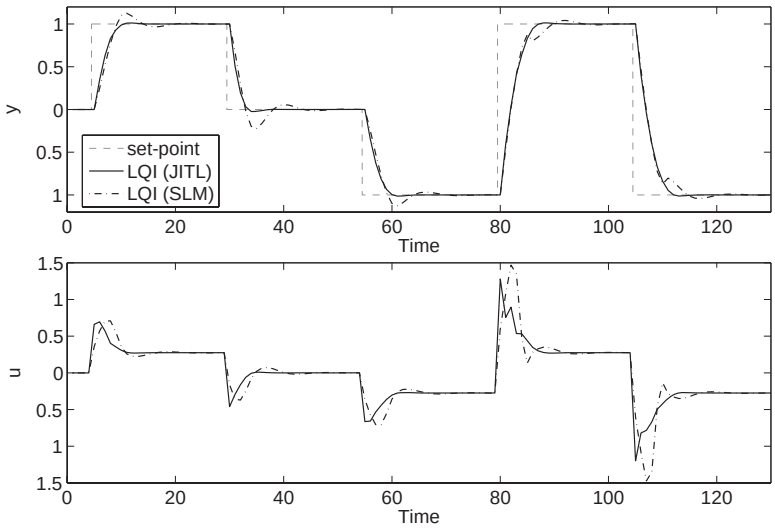

Figure 3.1: Servo performances of LQI designs based on JITL and successive linearization models (SLM)

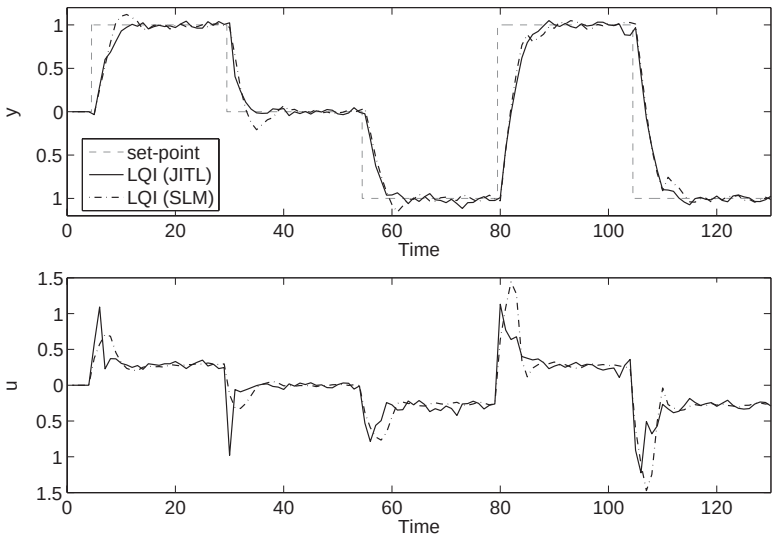

Figure 3.2: Servo performances of two LQI designs in the presence of noise

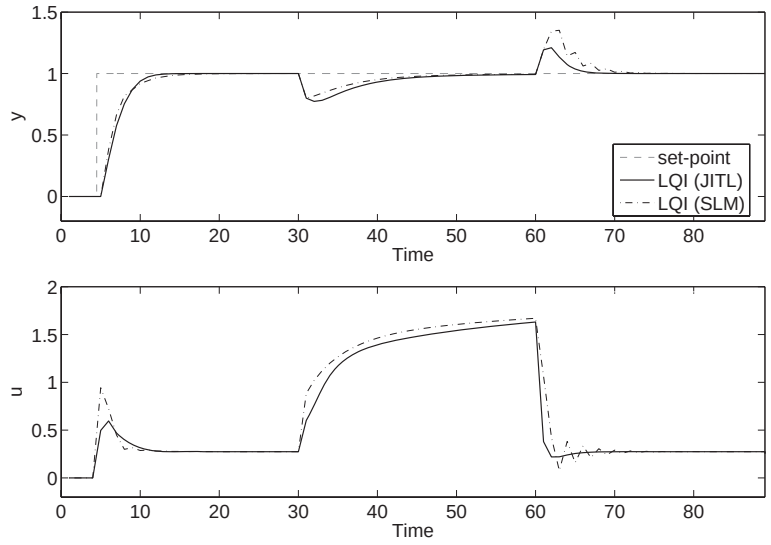

Figure 3.3: Disturbance rejection performances of LQI designs based on JITL and successive linearization models (SLM)

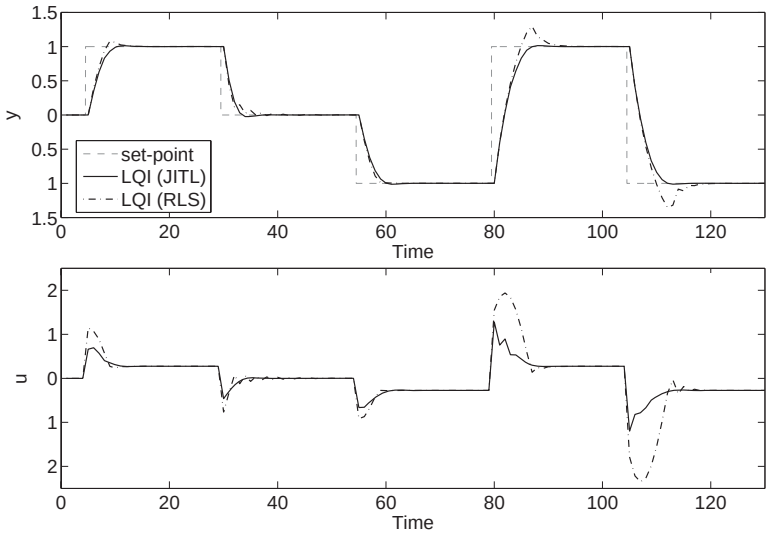

Figure 3.4: Servo performances of LQI designs based on JITL and recursive least square models (RLS)

Example 2 The second application considered is the Van de Vusse reaction, $A \rightarrow B \rightarrow C$ and $2A \rightarrow D$, carried out in an isothermal CSTR as described by the following equations (Doyle *et al.*, 1995):

$$\begin{aligned} \dot{C}_A &= -k_1 C_A - k_3 C_A^2 + \frac{F}{V}(C_{Af} - C_A) \\ \dot{C}_B &= k_1 C_A - k_2 C_B - \frac{F}{V} C_B \end{aligned} \tag{3.20}$$

where C_A, C_B, and C_C denote the concentrations of components A, B, and C, respectively. The model parameters and nominal operating condition used in the simulation are: $k_1 = 50$ h^{-1}, $k_2 = 100$ h^{-1}, $k_3 = 10$ L mol^{-1} h^{-1}, $C_{Af} = 10$ mol L^{-1}, $V = 1$ L, $F_0 = 34.3$ L h^{-1}, $C_A = 3.0$ mol L^{-1}, and $C_B = 1.12$ mol L^{-1}. The control problem is to regulate C_B by manipulating the inlet flow rate F.

To design the proposed data-based LQI controller, a first-order ARX model and parameters $\kappa = 0.9$, $k_{\min} = 12$, and $k_{\max} = 60$ are chosen for the JITL algorithm. The initial database is generated by introducing uniformly random steps with distribution of [30.3 38.3] in process input F. Again, for comparison purposes, LQI controller based on local models obtained by successive linearization of the first-principle model Eq. (3.20) as derived in Appendix A is designed. With weight $R = 2.80$ and $\varpi = 10$ chosen, the performances of these two controllers for setpoint change from 1.12 to 1.2 are compared in Figure 3.5. Although the LQI design based on the analytical models has slightly better performance, the simulation result shows that the proposed data-based LQI design, which does not require the availability of the first-principle model, is able to achieve comparable control performance as that achieved by its model-based counterpart.

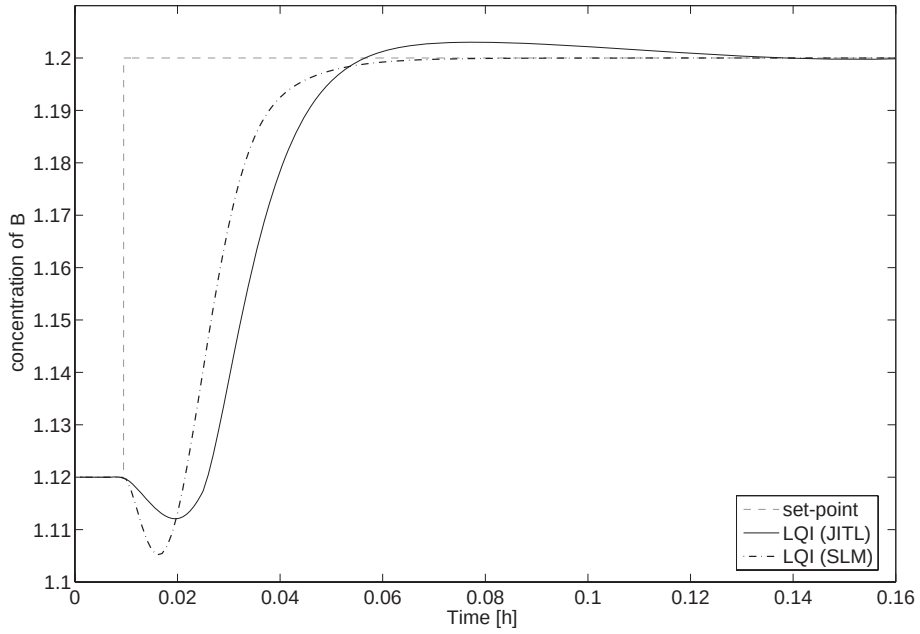

Figure 3.5: Performance comparison of two LQI designs

3.4 Conclusion

A data-based LQI design directly from plant data is proposed in this chapter. By incorporating the JITL technique, LQI design can be carried out without the need of a first-principle model or Kalman filter. Simulation results illustrate that the proposed LQI design achieves comparable control performance as that obtained by the LQI design based on the successive linearization of the nonlinear first-principle model.

Chapter 4

Internal Model Controller Design Using the JITL Technique

4.1 Introduction

Internal Model Control (IMC) is a powerful controller design strategy for the open-loop stable dynamic systems (Morari and Zafiriou, 1989). 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 non-linear processes with a range of operating conditions. To extend the IMC design to nonlinear processes, various nonlinear IMC schemes have been proposed 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) implemented a state-space linearization approach within

IMC framework to improve disturbance performance for nonlinear systems. A disadvantage of their 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 the plant/model mismatch. However, these IMC control strategies relied on the availability of a first-principle model, which is often unavailable or too time-consuming and costly to obtain due to the lack of complete physicochemical knowledge of chemical processes. An alternative approach is to develop black-box models from process data collected from the industrial processes. To this end, the ability of artificial neural networks to model almost any nonlinear function without a priori knowledge has led to the investigation of nonlinear IMC schemes using neural networks (NN). In the methods proposed by Bhat and Mcavoy (1990) and Hunt and Sbarbaro (1991), one NN was trained to represent the nonlinear dynamics of process, which was then used as the IMC model, while another NN was trained to learn the inverse dynamics of the process and was employed as the nonlinear IMC 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 the 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 obtain the inverse of process dynamics. To overcome these difficulties, Doyle et al. (1995) proposed a partitioned model-based IMC design based on the Volterra model that retains the original spirit and characteristics of conventional IMC while extending its capabilities to nonlinear systems. 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., 1996). Harris and Palazoglu (1998) proposed an alternative partitioned model-based 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. Shaw et al. (1997) used recurrent neural network within the partitioned model-based IMC scheme as an alternative for NN-based control application. Maksumov et al. (2002) investigated a similar control scheme consisting of a linear ARX model and a NN model. 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 network 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 incorporated into model based controller design.

To circumvent the aforementioned drawback, an adaptive IMC design based on the Just-in-Time Learning (JITL) technique is proposed in this chapter. In the proposed method, a set of linear models obtained on-line by the JITL is employed as the IMC model by which the IMC controller is designed. The JITL is considered not only because its prediction capability for nonlinear processes but also the low-order model employed in the JITL, which enables the construction of model-inverse in a straightforward manner. In addition, an updating algorithm for the IMC filter parameter is developed based on the Lyapunov method to guarantee the convergence of JITL's predicted tracking error. Literature examples are presented to illustrate the proposed control strategy and a comparison with its conventional counterparts is made.

4.2 JITL-Based Adaptive IMC Design

The block diagram of the IMC structure is shown in Figure 4.1, where G and $\tilde{G}$ denote the open-loop stable process and process model, respectively. The IMC controller, Q, can be designed by the following equation (Morari and Zafiriou, 1989):

$$Q = \tilde{G}_{-}^{-1} f \tag{4.1}$$

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 difficulty in the use of these models in the IMC framework arises in the design of IMC controller, which is based on the inverse

of minimum-phase part of the model $\tilde{G}$ that necessitates a reliable and efficient method to obtain this inversion (Maksumov et al., 2002). To address this problem, the JITL model is employed in the IMC framework so that the model inverse can be obtained readily. The proposed adaptive IMC scheme is depicted in Figure 4.2, where the process model $\tilde{G}$ is updated by the JITL algorithm on-line and controller Q is designed based on the inverse of the minimum-phase dynamics of process model $\tilde{G}$ augmented with a low-pass filter. In the proposed method, filter parameter is not fixed, instead it is adjusted on-line by an updating algorithm to be developed in the sequel. As such, the JITL is employed to update the parameters of both IMC model and controller.

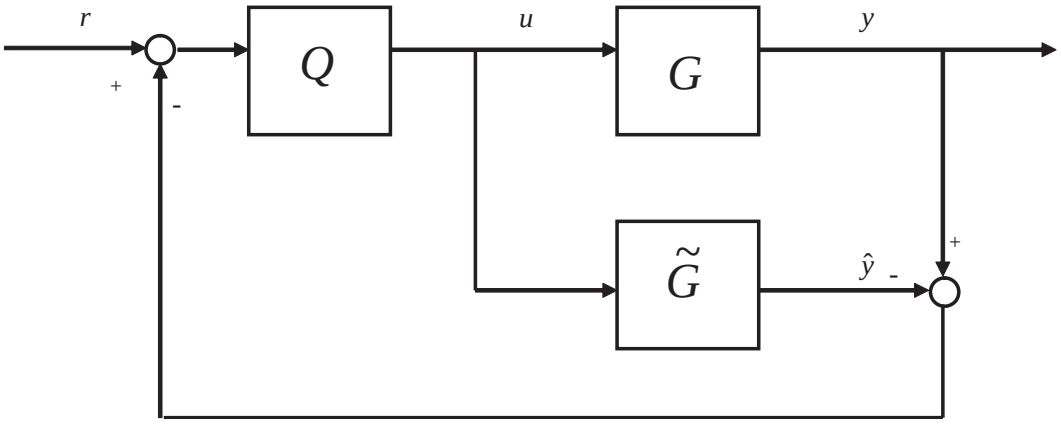

Figure 4.1: Block diagram of IMC structure

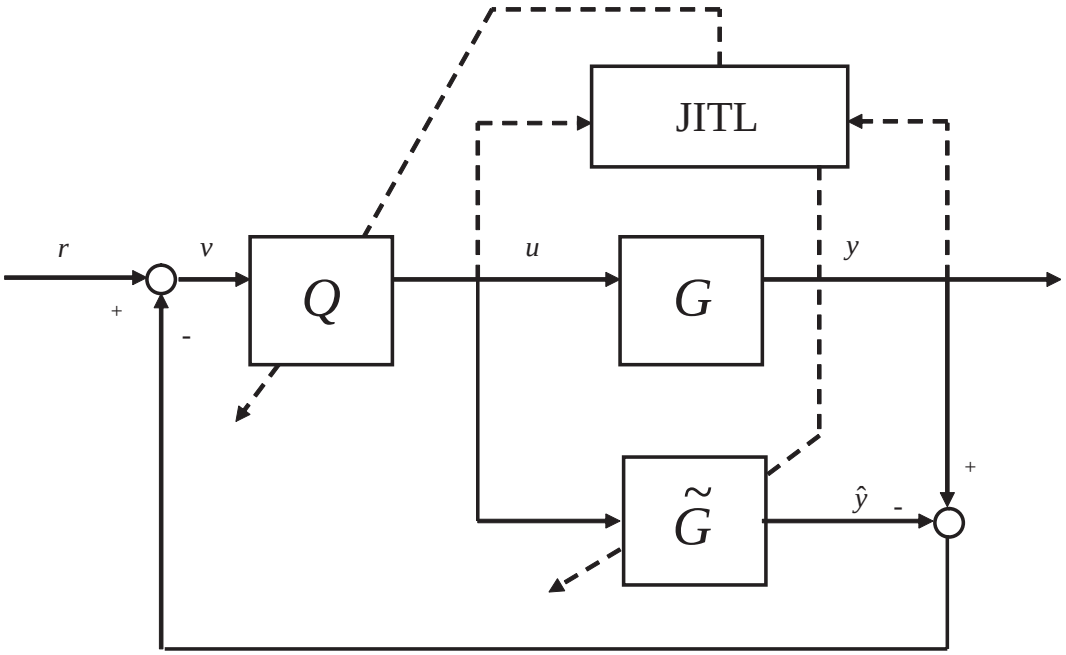

Figure 4.2: JITL-based adaptive IMC scheme

As discussed previously, a simple model structure, e.g. low-order ARX models, is usually employed by the JITL. Therefore, the following second-order ARX model is considered in the proposed controller design,

$$\hat{y}(k) = \alpha_1^k y(k-1) + \alpha_2^k y(k-2) + \beta_1^k u(k-1) \tag{4.2}$$

where $\hat{y}(k)$ is the predicted output by the JITL at the k-th sampling time, $y(k-1)$ and $u(k-1)$ are the output and manipulated variables at the $(k-1)$-th sampling time, α_1^k, α_2^k and β_1^k are the model coefficients at the k-th sampling time.

The transfer function model corresponding to Eq. (4.2) is given by:

$$\tilde{G}^k(z^{-1}) = \frac{\beta_1^k z^{-1}}{1 - \alpha_1^k z^{-1} - \alpha_2^k z^{-2}} \tag{4.3}$$

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

$$\tilde{Q}^k(z^{-1}) = \frac{1 - \alpha_1^k z^{-1} - \alpha_2^k z^{-2}}{\beta_1^k} \frac{1 - \lambda(k)}{1 - \lambda(k) z^{-1}} \tag{4.4}$$

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

The control law resulting from Eq. (4.4) is then given by

$$u(k) = \lambda(k)u(k-1) + \frac{1-\lambda(k)}{\beta_1^k}\left(v(k) - \alpha_1^k v(k-1) - \alpha_2^k v(k-2)\right) \tag{4.5}$$

where $v(k) \triangleq r(k) + \hat{y}(k) - y(k)$.

Because IMC filter parameter $\lambda(k)$ is constrained between 0 and 1, the following sigmoid function is employed to map the set [0 1] to $\Re$, which denotes the set of real number:

$$\lambda(k) = \frac{1}{1+e^{-\zeta(k)}} \tag{4.6}$$

where $\zeta(k) \in \Re$. In the sequel, an updating algorithm will be developed to adjust $\zeta(k)$ on-line, and subsequently the filter parameter $\lambda(k)$ can be easily calculated by Eq. (4.6).

In order to update the parameter $\zeta(k)$ so that the convergence of the JITL's predicted output to the desired set-point trajectory is guaranteed, Lyapunov function is chosen as follows:

$$V(k) = \gamma e_{\mathrm{r}}^2(k) \tag{4.7}$$

where $e_{\mathrm{r}}(k)$ is the predicted tracking error defined as $e_{\mathrm{r}}(k) = r(k) - \hat{y}(k)$ and γ is a positive constant.

Define

$$e_{\mathrm{r}}(k+1) = e_{\mathrm{r}}(k) + \Delta e_{\mathrm{r}}(k+1) \tag{4.8}$$

The increment of the Lyapunov function, $\Delta V(k)$, is obtained by

$$\begin{aligned}\Delta V(k) &= V(k+1) - V(k) \\ &= \gamma e_{\mathrm{r}}^2(k+1) - \gamma e_{\mathrm{r}}^2(k) \\ &= 2\gamma e_{\mathrm{r}}(k)\Delta e_{\mathrm{r}}(k+1) + \gamma \Delta e_{\mathrm{r}}^2(k+1) \end{aligned} \tag{4.9}$$

In Eq. (4.9), $\Delta e_{\mathrm{r}}(k+1)$ can be further expressed as

$$\begin{aligned}\Delta e_{\mathrm{r}}(k+1) &= \frac{\partial e_{\mathrm{r}}(k+1)}{\partial k} \\ &= \frac{\partial \left[r(k+1) - \hat{y}(k+1)\right]}{\partial u(k)} \frac{\partial u(k)}{\partial \lambda(k)} \frac{\partial \lambda(k)}{\partial \zeta(k)} \frac{\partial \zeta(k)}{\partial k} \\ &= -\frac{\partial \hat{y}(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial \lambda(k)} \frac{\partial \lambda(k)}{\partial \zeta(k)} \Delta\zeta(k) \end{aligned} \tag{4.10}$$

where

$$\begin{aligned}\frac{\partial \hat{y}(k+1)}{\partial u(k)} &= \beta_1^{k+1} \\ \frac{\partial u(k)}{\partial \lambda(k)} &= u(k-1) - \frac{1}{\beta_1^k}\left(v(k) - \alpha_1^k v(k-1) - \alpha_2^k v(k-2)\right) \\ \frac{\partial \lambda(k)}{\partial \zeta(k)} &= \lambda(k)\left(1 - \lambda(k)\right) \end{aligned} \tag{4.11}$$

Based on the on-going analysis, the following theorem provides the theoretical basis for the convergence property of the proposed updating algorithm for $\zeta(k)$.

Theorem 1. Let $0 < \eta(k) < 2$ and the parameter $\zeta(k)$ is updated by the following equation,

$$\zeta(k+1) = \zeta(k) + \frac{\eta(k)}{\beta_1^{k+1}} \frac{e_{\mathrm{r}}(k)}{\lambda(k)(1-\lambda(k))} \left[\frac{\partial u(k)}{\partial \lambda(k)}\right]^{-1} \tag{4.12}$$

then the tracking error $e_{\mathrm{r}}(k)$ is guaranteed to converge to zero asymptotically.

Proof: Based on Eqs. (4.10) and (4.12), Eq. (4.9) is further derived as:

$$
\begin{aligned}
&\Delta V(k) \\
&= -2\gamma e_{\mathrm{r}}(k)\frac{\partial \hat{y}(k+1)}{\partial u(k)}\frac{\partial u(k)}{\partial \lambda(k)}\frac{\partial \lambda(k)}{\partial \zeta(k)}\Delta\zeta(k) + \gamma\left(-\frac{\partial \hat{y}(k+1)}{\partial u(k)}\frac{\partial u(k)}{\partial \lambda(k)}\frac{\partial \lambda(k)}{\partial \zeta(k)}\Delta\zeta(k)\right)^2 \\
&= -2\gamma e_{\mathrm{r}}(k)\beta_1^{k+1}\frac{\partial u(k)}{\partial \lambda(k)}\lambda(k)\left(1-\lambda(k)\right)\frac{\eta(k)}{\beta_1^{k+1}}\frac{e_{\mathrm{r}}(k)}{\lambda(k)\left(1-\lambda(k)\right)}\left[\frac{\partial u(k)}{\partial \lambda(k)}\right]^{-1} \\
&\quad +\gamma\left(-\beta_1^{k+1}\frac{\partial u(k)}{\partial \lambda(k)}\lambda(k)\left(1-\lambda(k)\right)\frac{\eta(k)}{\beta_1^{k+1}}\frac{e_{\mathrm{r}}(k)}{\lambda(k)\left(1-\lambda(k)\right)}\left[\frac{\partial u(k)}{\partial \lambda(k)}\right]^{-1}\right)^2 \\
&= -2\gamma\eta(k)e_{\mathrm{r}}^2(k) + \gamma\eta^2(k)e_{\mathrm{r}}^2(k) \\
&= -\eta(k)(2-\eta(k))\gamma e_{\mathrm{r}}^2(k) \qquad (4.13)
\end{aligned}
$$

It is evident from Eq. (4.13) that $\Delta V(k)$ is always negative if $0 < \eta(k) < 2$ holds, meaning that the predicted tracking error $e_{\mathrm{r}}(k)$ is guaranteed to converge to zero by using the updating algorithm, Eq. (4.12), to design $\zeta(k+1)$. This completes the proof.

One remark about Theorem 1 is the determination of $\eta(k)$. Generally, a larger $\eta(k)$ in the specified range [0 2] would lead to faster convergence but might result in overshoot and oscillatory response, while a smaller $\eta(k)$ has the opposite effect. Owing to the lacking of systematic guidelines for the determination of $\eta(k)$, it is usually chosen experimentally for each problem. In the proposed design, the following rules are used to update the learning rate: (i) if the increment of $|e_{\mathrm{r}}(k)|$ is more than the threshold, the filter parameter remains unchanged and the learning rate is decreased by a factor l_{dec}, i.e. $\eta(k+1) = l_{dec}\eta(k)$; (ii) if the absolute value of the change of $|e_{\mathrm{r}}(k)|$ is within the threshold, only the filter parameter is updated; otherwise, (iii) the filter parameter is updated and the learning rate is increased by a factor l_{inc}, i.e. $\eta(k+1) = l_{inc}\eta(k)$.

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

Step 1 Given the initial database for the JITL, initialize the IMC filter parameter and its learning rate;

Step 2 Given the current process output $y(k)$, compute the manipulated variable $u(k)$ from Eq. (4.5);

Step 3 The JITL database is updated by the current process data if the absolute value of prediction error between the JITL's output and the current process output is larger than the specified threshold;

Step 4 Obtain ARX model for the next sampling instant by using the current process data and JITL algorithm, followed by updating $\eta(k)$ and $\zeta(k)$. Consequently, IMC filter parameter at the next sampling instant, $\lambda(k+1)$, is calculated by using Eq. (4.6)

Step 5 Set $k = k + 1$ and go to Step 2.

4.3 Examples

Example 1 Consider a continuous polymerization reaction that takes place in a jacketed CSTR (Doyle et al., 1995), where 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 (y =NAMW) by manipulating the flow rate of the

initiator ($u = F_I$). This process can be described by the following balance equations:

$$\begin{aligned}
\frac{dC_{\mathrm{m}}}{dt} &= -(k_{\mathrm{P}} + k_{\mathrm{fm}})C_{\mathrm{m}}P_0 + \frac{F(C_{\mathrm{m_{in}}} - C_{\mathrm{m}})}{V} \\
\frac{dC_{\mathrm{I}}}{dt} &= -k_{\mathrm{I}}C_{\mathrm{I}} + \frac{F_{\mathrm{I}}C_{\mathrm{I_{in}}} - FC_{\mathrm{I}}}{V} \\
\frac{dD_0}{dt} &= (0.5k_{\mathrm{T_C}} + k_{\mathrm{T_d}})P_0^2 + k_{\mathrm{fm}}C_{\mathrm{m}}P_0 - \frac{FD_0}{V} \\
\frac{dD_1}{dt} &= M_{\mathrm{m}}(k_{\mathrm{P}} + k_{\mathrm{fm}})C_{\mathrm{m}}P_0 - \frac{FD_1}{V}
\end{aligned} \tag{4.14}$$

where $P_0 = \left[\frac{2f^*k_{\mathrm{I}}C_{\mathrm{I}}}{k_{\mathrm{T_d}}+k_{\mathrm{T_c}}}\right]^{0.5}$ and $y = \frac{D_1}{D_0}$. The steady-state operation condition and model parameters are given in Tables 4.1 and 4.2.

To proceed with the JITL method, input and output data are generated by introducing uniformly random steps with distribution of [0.012 0.021] to the process input F_{I}. With sampling time of 0.03h, input and output data thus obtained (see Figure 4.3) are used to build the database. A second-order ARX model is used as the local model and the parameters chosen for JITL algorithm are as follows: $\kappa = 0.95$, $k_{\min} = 6$ and $k_{\max} = 60$. The initial IMC filter is $\lambda = 0.83$ with the initial learning rate $\eta = 1 \times 10^{-7}$. For the purpose of comparison, an adaptive IMC controller is designed based on a second-order ARX model with parameter adaptation by the recursive least-square (RLS) identification procedure and IMC filter parameter equal to $\lambda = 0.76$ (Shahrokhi and Baghmisheh, 2005). In addition, the following benchmark IMC controller employed in the previous work (Doyle et al., 1995) is also designed based on the linear model obtained at the nominal operating condition and a second-order IMC filter with filter time constant equal to 0.06:

$$Q(s) = \frac{-7.3s^4 - 298.7s^3 - 4589.6s^2 - 31334.6s - 80187.3}{s^4 + 55.1s^3 + 1122.1s^2 + 9983.6s + 32753.6} \tag{4.15}$$

Table 4.1: 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	=	0.016783	m^3/h
D_0	=	0.0019752	kmol/m^3	y	=	25000.5	kg/kmol

Table 4.2: Model parameters for polymerization reactor

k_{Tc}	=	1.3281×10^{10}	m^3/(kmol h)	F	=	1.00	m^3/h
k_{Td}	=	1.0930×10^{11}	m^3/(kmol h)	V	=	0.1	m^3
k_{I}	=	1.0225×10^{-1}	L/h	$C_{\mathrm{I_{in}}}$	=	8.0	kmol/m^3
k_{P}	=	2.4952×10^{6}	m^3/(kmol h)	M_{m}	=	100.12	kg/kmol
k_{fm}	=	2.4522×10^{3}	m^3/(kmol h)	$C_{\mathrm{m_{in}}}$	=	6.0	kmol/m^3
f^*	=	0.58					

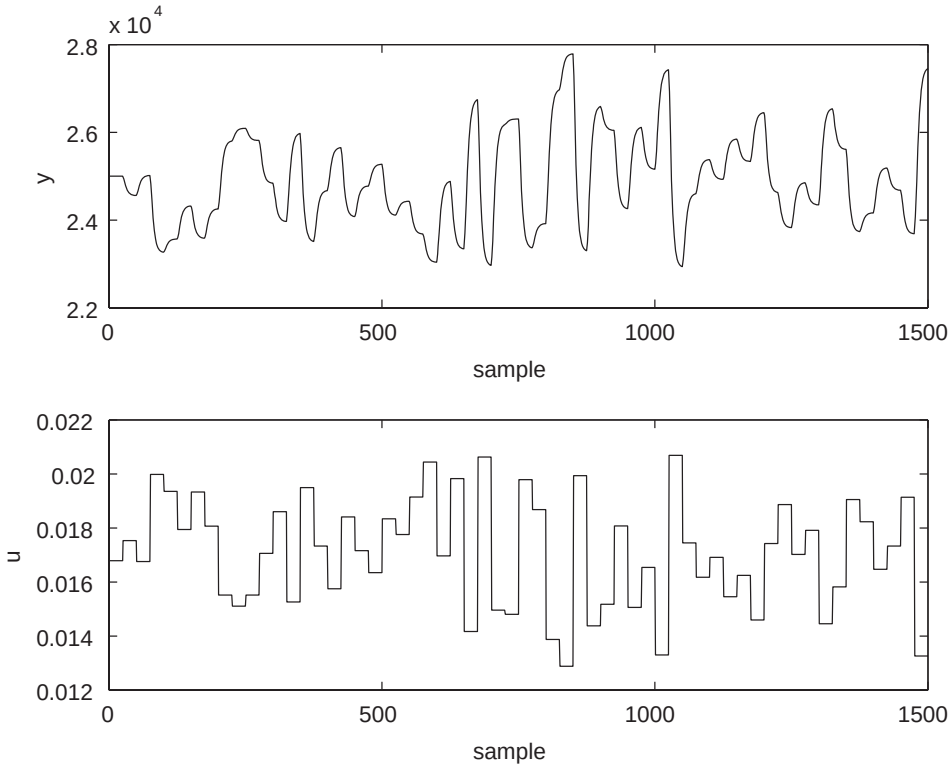

Figure 4.3: Input and output data used to construct the JITL's database

To evaluate the servo performances of three controllers, successive set-point changes between 25000.5 kg/kmol and 12500 kg/kmol as illustrated in Figure 4.4 are considered. Note that the symbol "$*$" in this figure denotes the sample instants at which JITL's database is updated. It is obvious that the proposed adaptive IMC design has better performance than those achieved by the other two IMC controllers, as evidenced by the reduction of the Mean Absolute Error (MAE) by 18.8% and 18.1%, relative to the linear IMC and RLS-based adaptive IMC controllers, respectively. Figure 4.5 shows the updating of the IMC filter parameter in the aforementioned closed-loop responses.

To compare the disturbance rejection capability of three IMC controllers, unmeasured $\pm 10\%$ step disturbances in the inlet initiator concentration $C_{\mathrm{I_{in}}}$ are considered. The resulting closed-loop responses at three different operating points are shown in Figures 4.6 and 4.7. As also can be seen from Table 4.3, the proposed IMC controller yields slightly better performance around the nominal operating condition compared to the other two IMC controllers. However, when the operating conditions are away from the nominal one, the proposed design gives marked improvement as evidenced by the respective MAE reductions in the ranges of 2% to 39% (RLS-IMC) and 33% to 64% (IMC). Next, to evaluate the robustness of the proposed control strategy, 10% modeling error in the kinetic parameter k_{I} and 20% error in the coefficients of the D_1 and M_{m} are assumed. It is evident from Figure 4.8 that the proposed controller still maintains better control performance over the linear IMC controller by achieving 22.7% reduction of MAE in the aforementioned set-point changes. In this case, RLS-based adaptive IMC controller yields steady-state off-set as shown in Figure 4.9. Lastly, 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 4.10, the proposed IMC controller can yield reasonably good control performance in the presence of process noise.

Table 4.3: Control performance comparison of three controllers

	Tracking error (MAE)		
	Adaptive IMC (JITL)	Adaptive IMC (RLS)	Linear IMC
Servo Response	799.59	976.49	984.27
Servo Response*	907.55	off-set	1174.70
+10% in $C_{I_{in}}$ at y=25000.5	45.45	45.94	46.42
+10% in $C_{I_{in}}$ at y=18750	43.72	69.50	90.16
+10% in $C_{I_{in}}$ at y=13500	94.42	96.90	141.15
-10% in $C_{I_{in}}$ at y=25000.5	52.35	55.12	56.88
-10% in $C_{I_{in}}$ at y=18750	39.27	64.34	110.81
-10% in $C_{I_{in}}$ at y=13500	85.40	97.31	171.93

* In the presence of modeling error

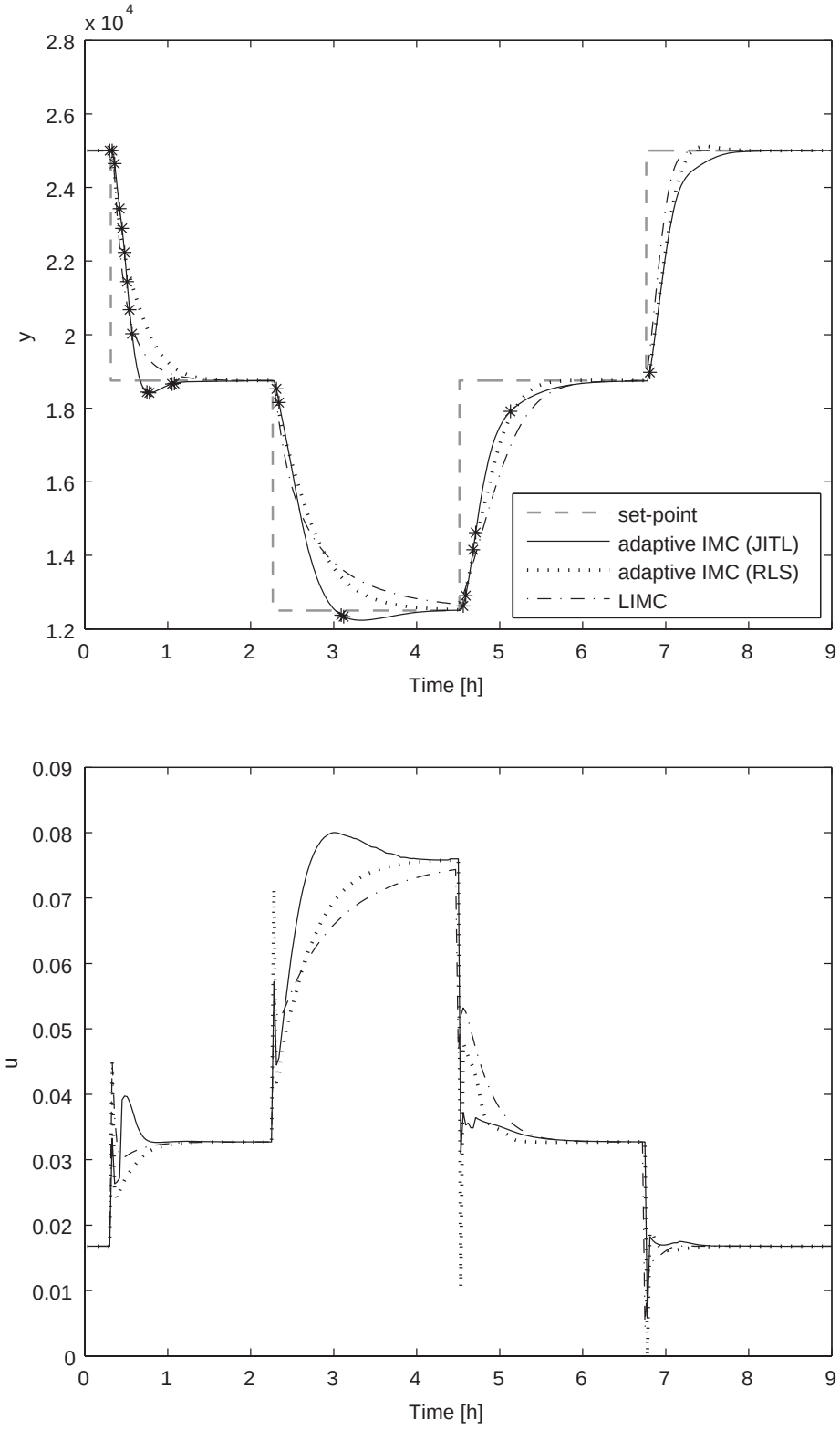

Figure 4.4: Servo responses of three IMC designs ($*$: database update)

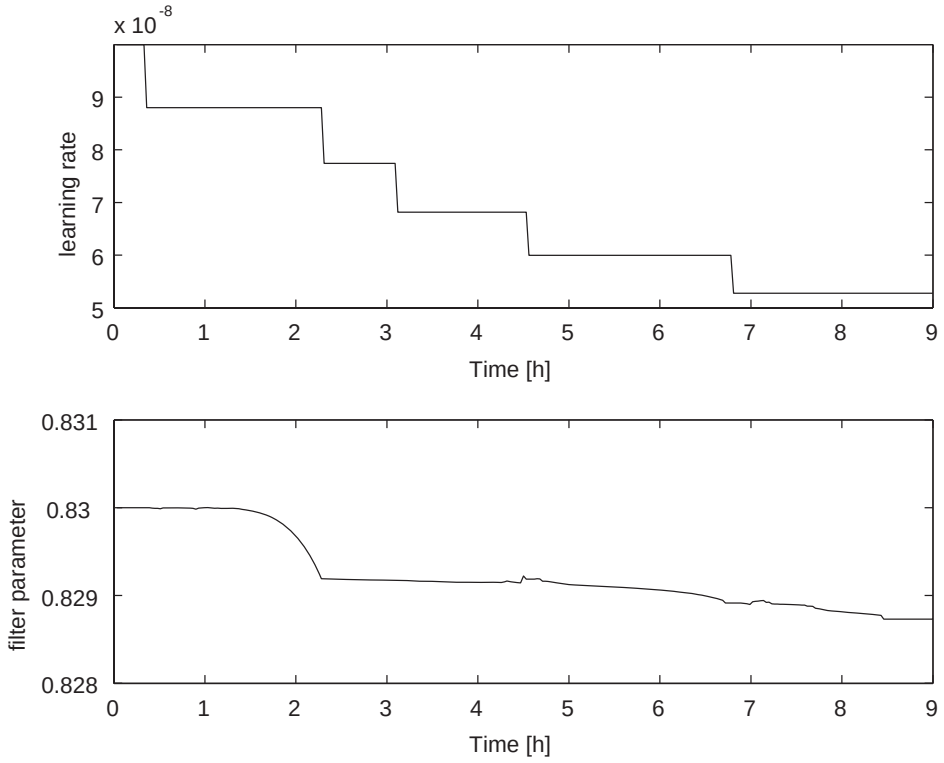

Figure 4.5: Updating of the IMC filter parameter and learning rate for servo response

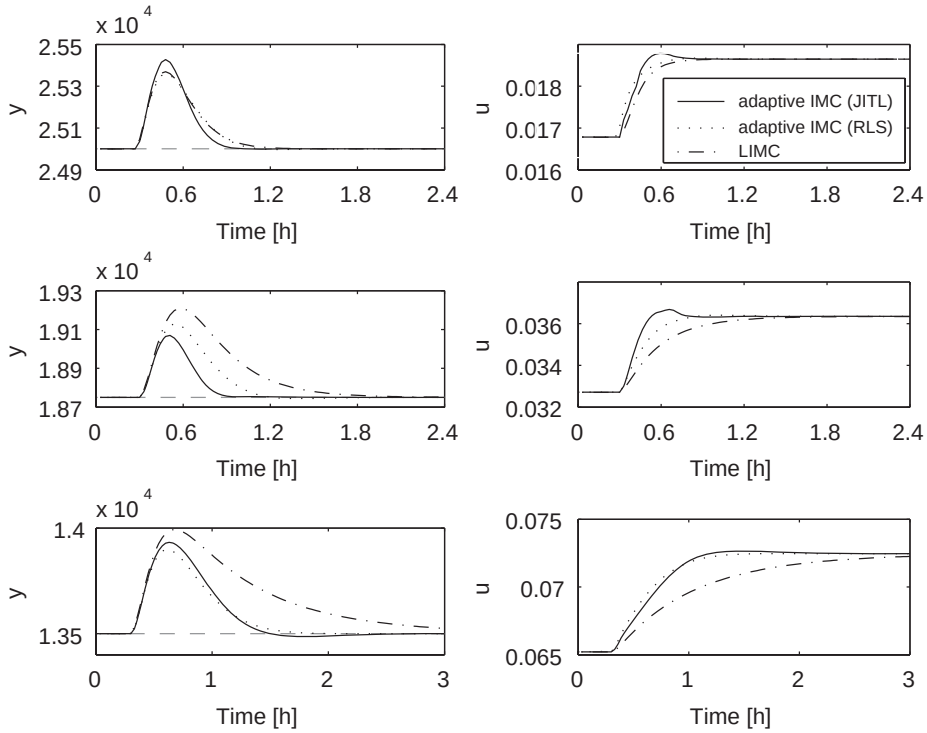

Figure 4.6: Closed-loop responses for -10% step change in $C_{\mathrm{I_{in}}}$

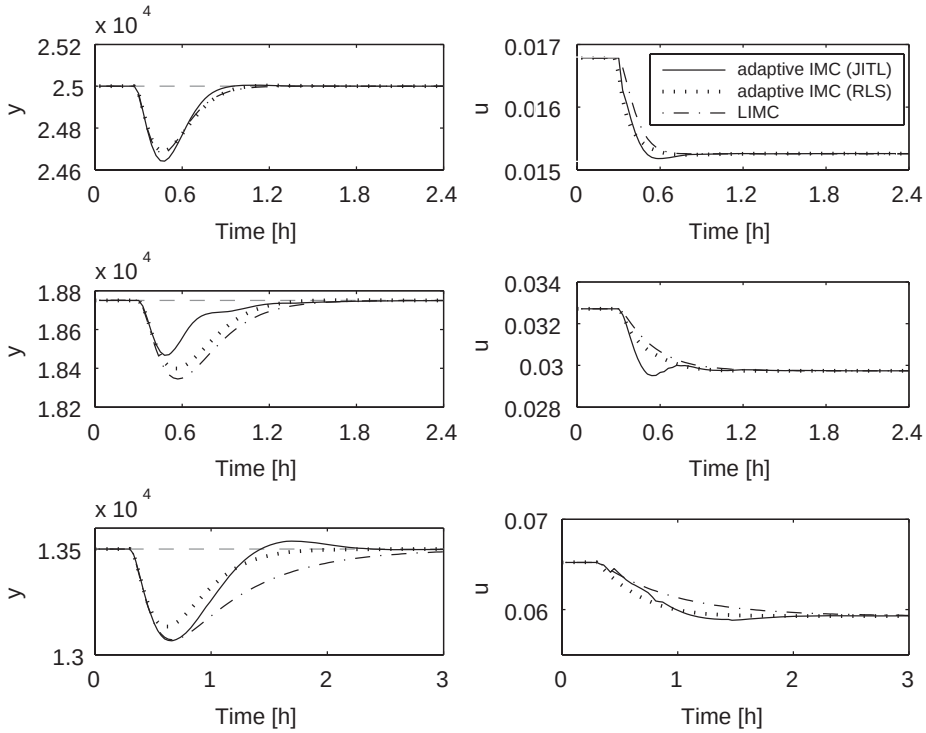

Figure 4.7: Closed-loop responses for +10% step change in $C_{\mathrm{I_{in}}}$

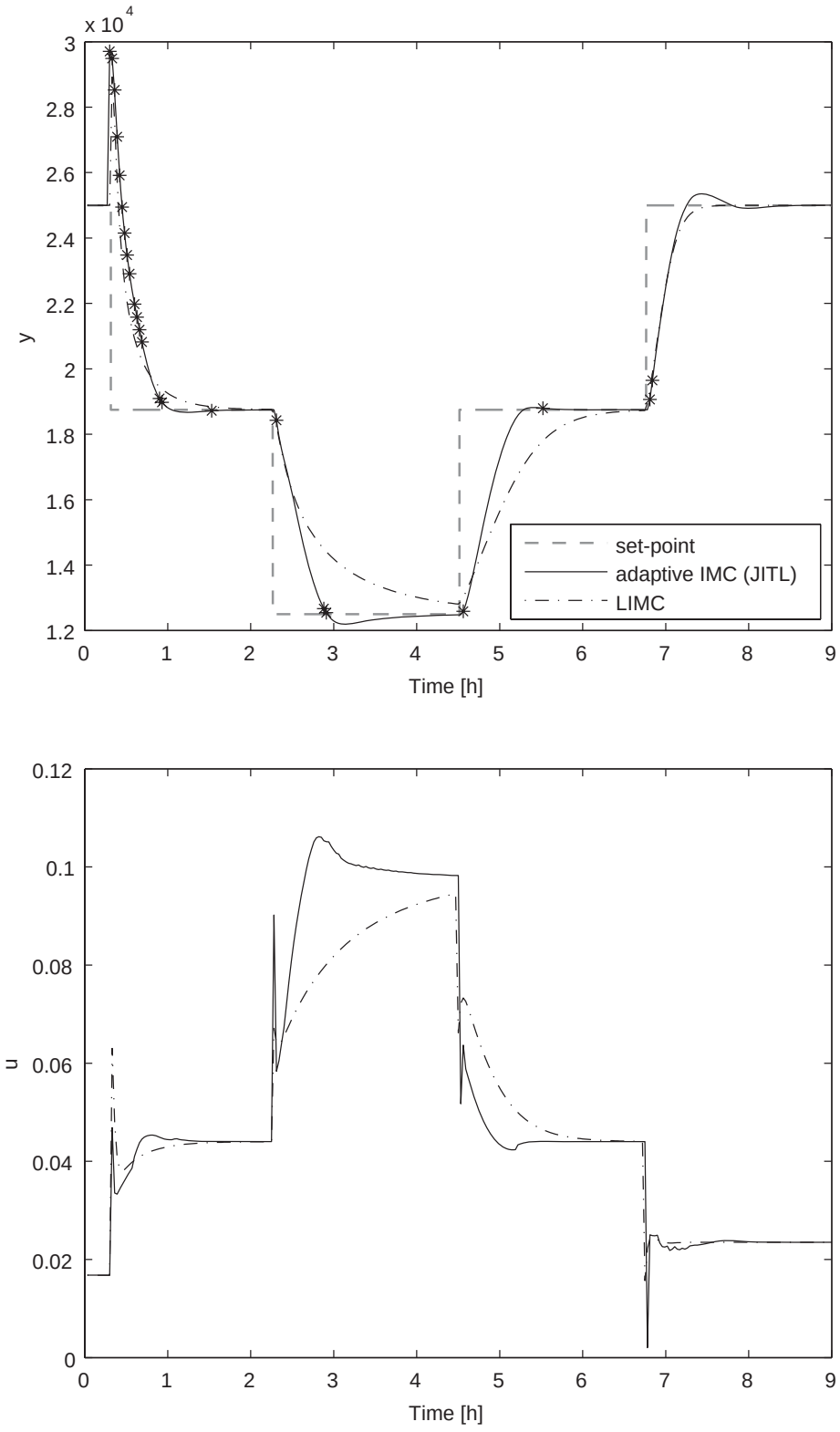

Figure 4.8: Servo responses of two IMC designs in the presence of modeling error

($*$: database update)

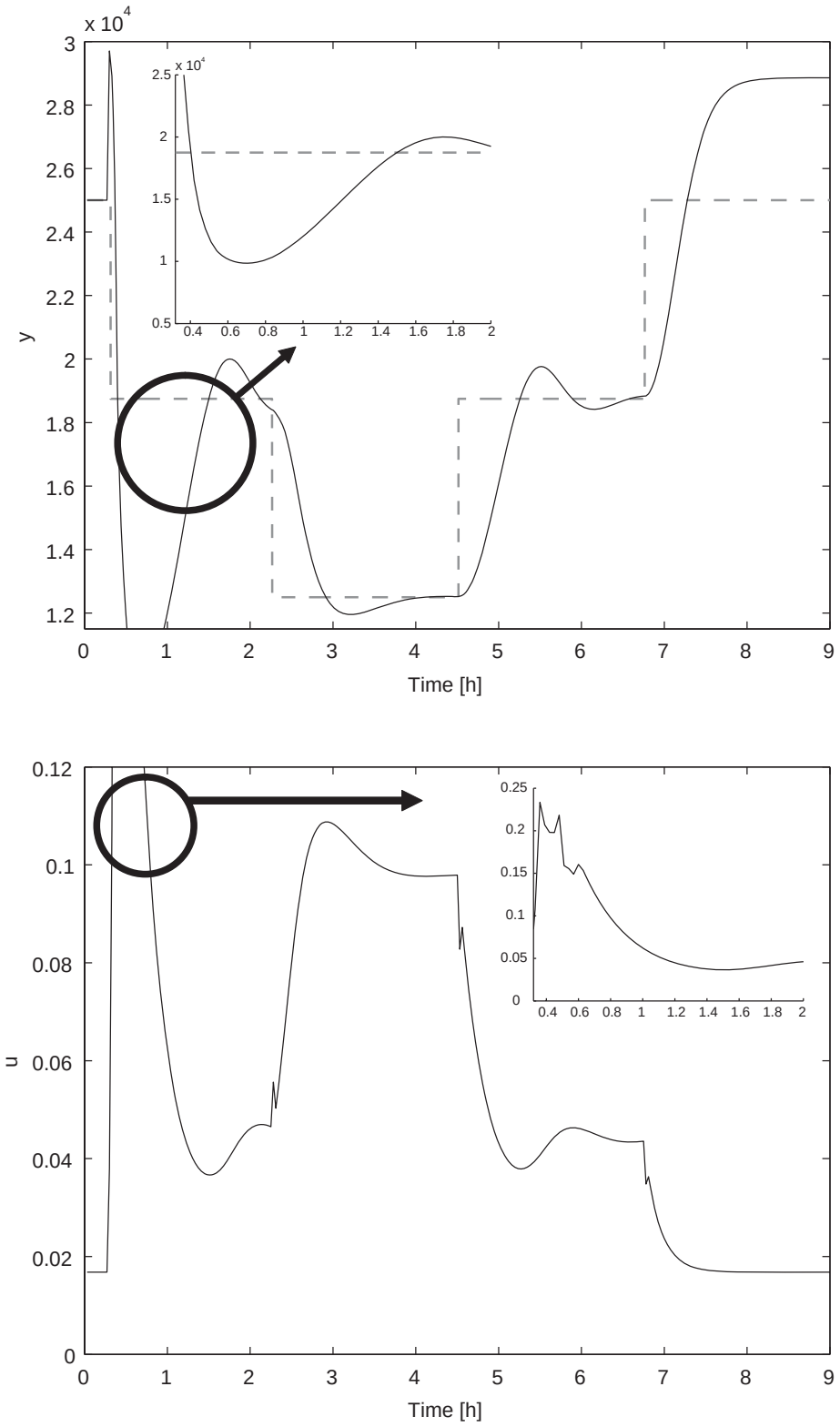

Figure 4.9: Servo responses of RLS-based IMC designs in the presence of modeling error

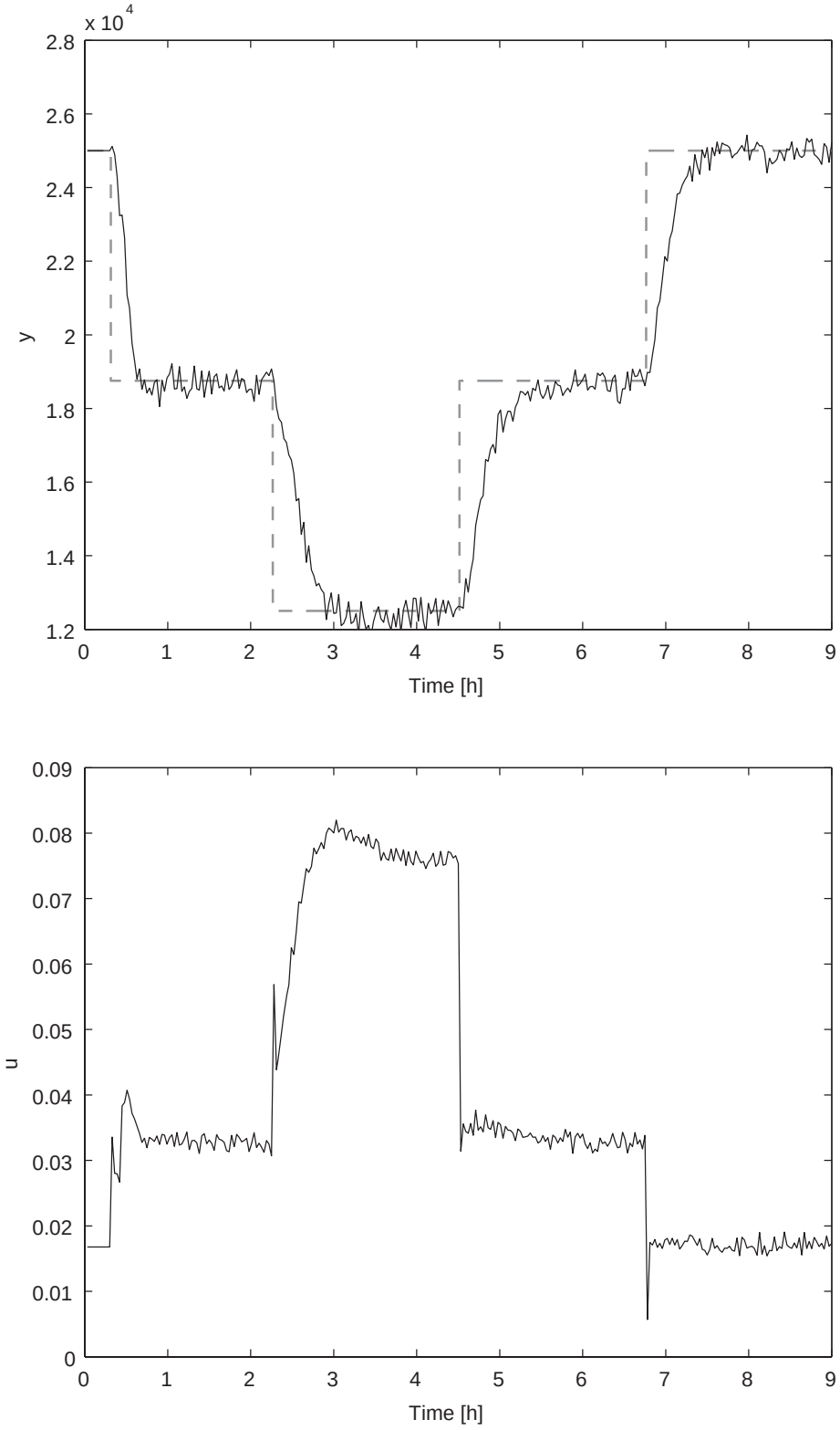

Figure 4.10: Servo response of the proposed IMC design in the presence of noise

Example 2 Consider the following first-order plus dead-time process (Tan et al., 1997):

$$\tau_1(y)\frac{dy}{dt} + y = \tau_2(y)u(t-\theta) \tag{4.16}$$

where y is the process output, u is the process input, θ is the dead-time of 0.5 unit time, and $\tau_1(y)$ and $\tau_2(y)$ are the model parameters dependent on y. The process has three operating regions as specified in Table 4.4, where the nonlinear region 2 is surrounded by two linear ones (regions 1 and 3).

Table 4.4: Process model for example 2

Process condition	Region 1	Region 2	Region 3
y	$y < 2$	$2 \leq y \leq 7$	$7 < y$
$\tau_1(y)$	10	$\sqrt{133.6 - 16.8y}$	4
$\tau_2(y)$	1	$\sqrt{4.8y - 8.6}$	5

To proceed with the JITL method, input and output data used to construct the JITL's database are generated by introducing uniformly random steps with distribution of [0 2] to the process input u. A first-order ARX model is used as the local model and the parameters chosen for JITL algorithm are as follows: $\kappa = 0.95$, $k_{\min} = 6$ and $k_{\max} = 60$. The initial IMC filter is $\lambda = 0.9$ with the initial learning rate $\eta = 1.3 \times 10^{-9}$. For the purpose of comparison, the gain-scheduling PI controller employed in the previous work (Tan et al., 1997) is designed. The parameters of this gain-scheduling PI controller are summarized in Table 4.5.

The servo performances of two controllers are compared for the successive set-point changes as illustrated in Figures 4.11 and 4.12. It is obvious that the pro-

posed adaptive IMC design has better performance than that achieved by the gain-scheduling PI controller because the latter exhibits sustained oscillation around the set-point equal to 3 and large overshoot for the set-point change from 3 to 5. Next, to evaluate the robustness of the proposed control strategy, it is assumed that the process parameter $\tau_2(y)$ is subject to 10% modeling error and the resulting servo responses of two controllers are compared in Figures 4.13 and 4.14. Evidently, the proposed controller still maintains superior control performance as compared with gain-scheduling PI controller. Lastly, to study the effect of process noise on the proposed design, both process input and output are corrupted by 3% Gaussian white noise. As shown in Figure 4.15, the proposed IMC controller can yield reasonably good control performance in the presence of process noise.

Table 4.5: Gain-scheduling PI controller

y	≤ 2.0	2.0	5	7
K_{C}	10.0	1.25	2.0	1.0
τ_{I}	10.0	3.5	2.15	2.0

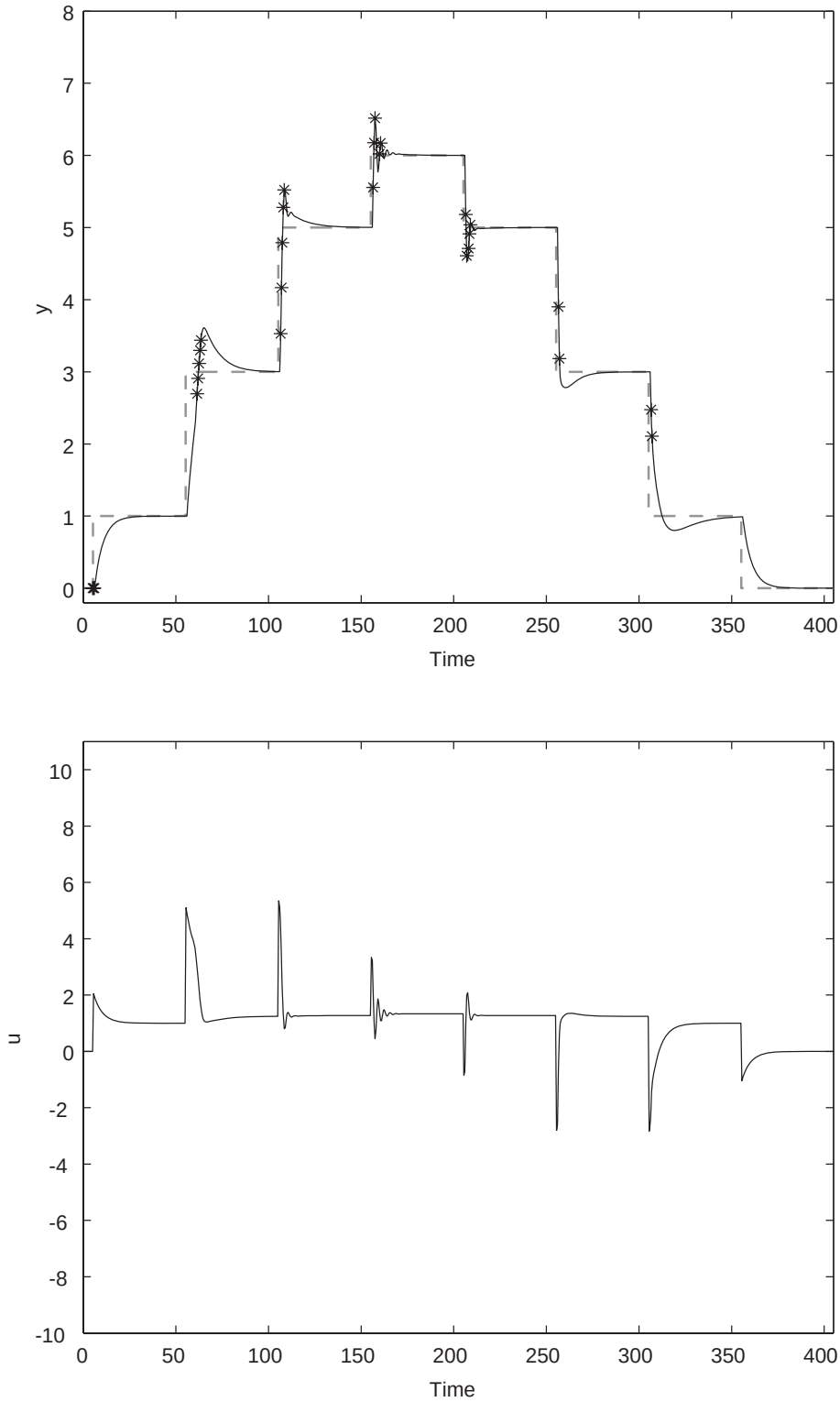

Figure 4.11: Servo response of the proposed IMC design ($*$: database update)

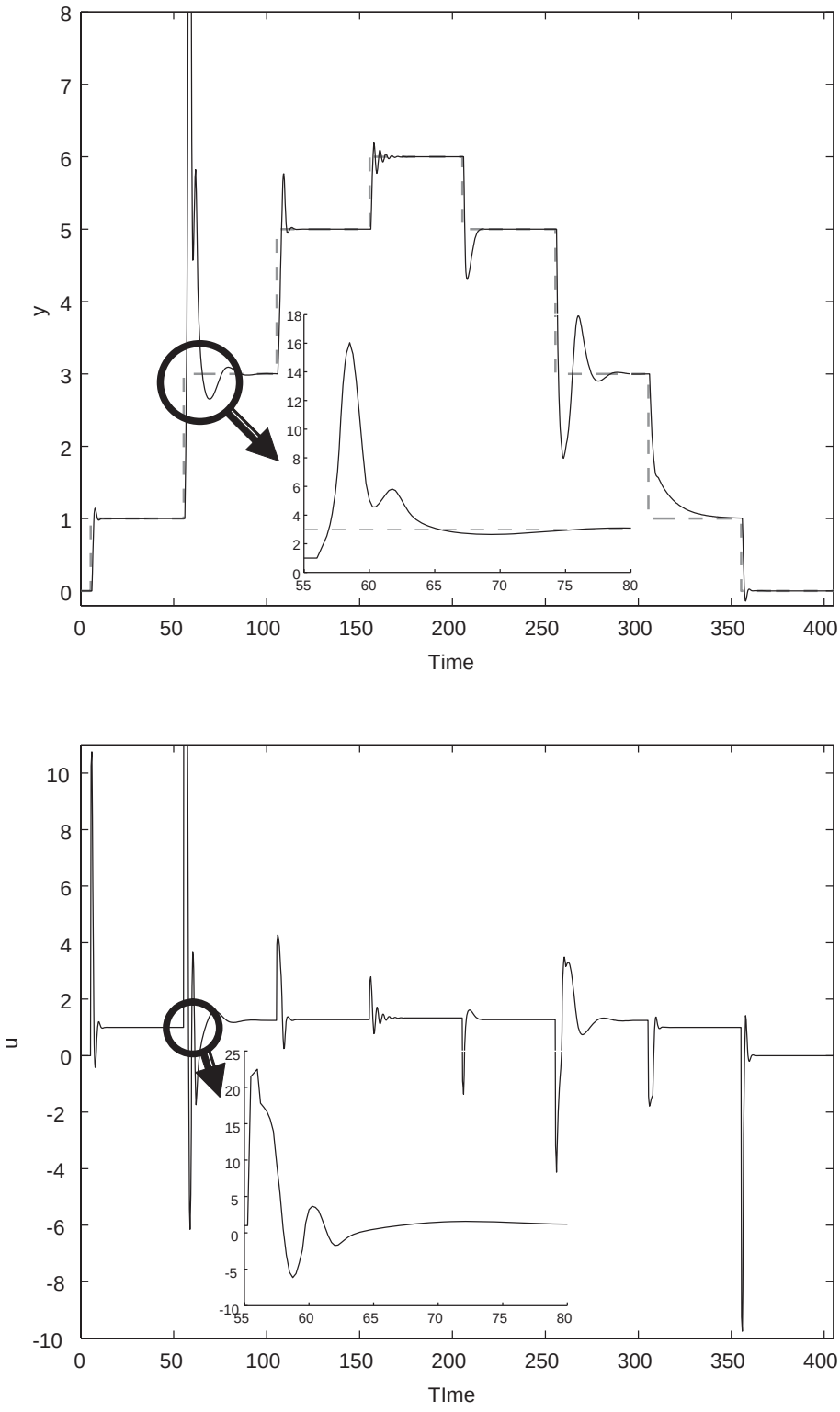

Figure 4.12: Servo response of the gain-scheduling PI controller

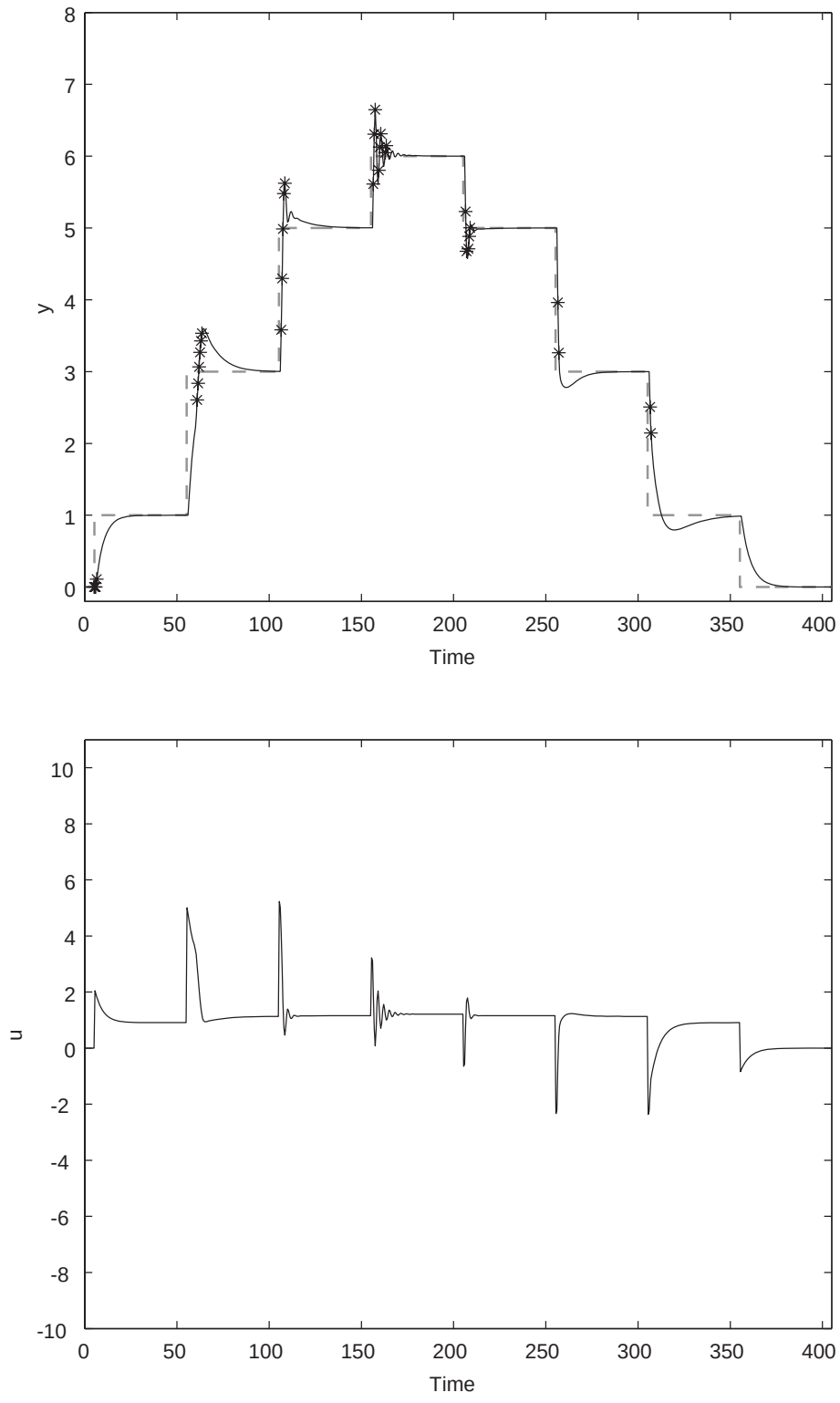

Figure 4.13: Servo response of the proposed IMC design under +10% modeling error in τ_2 ($*$: database update)

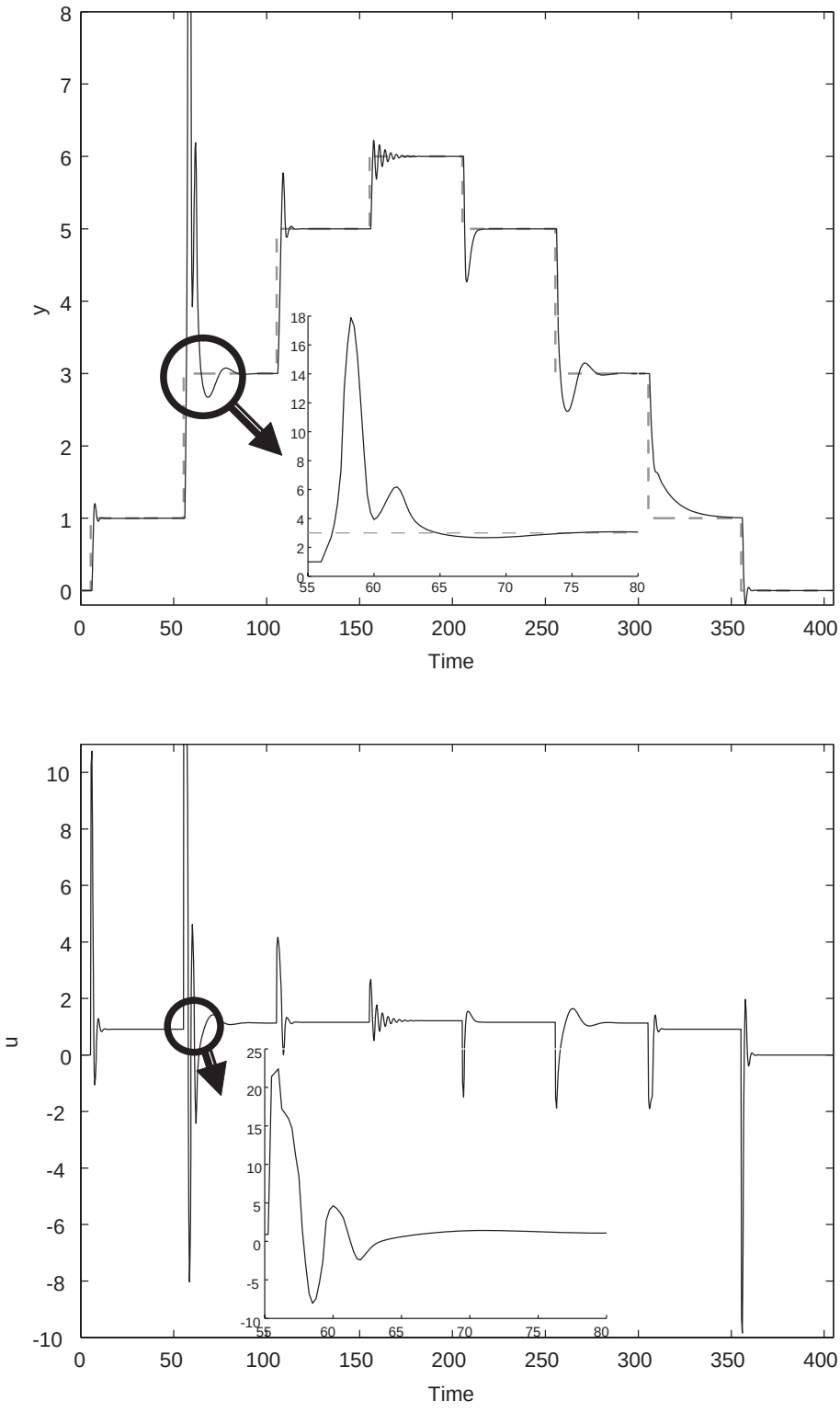

Figure 4.14: Servo response of the gain-scheduling PI controller under +10% modeling error in τ_2

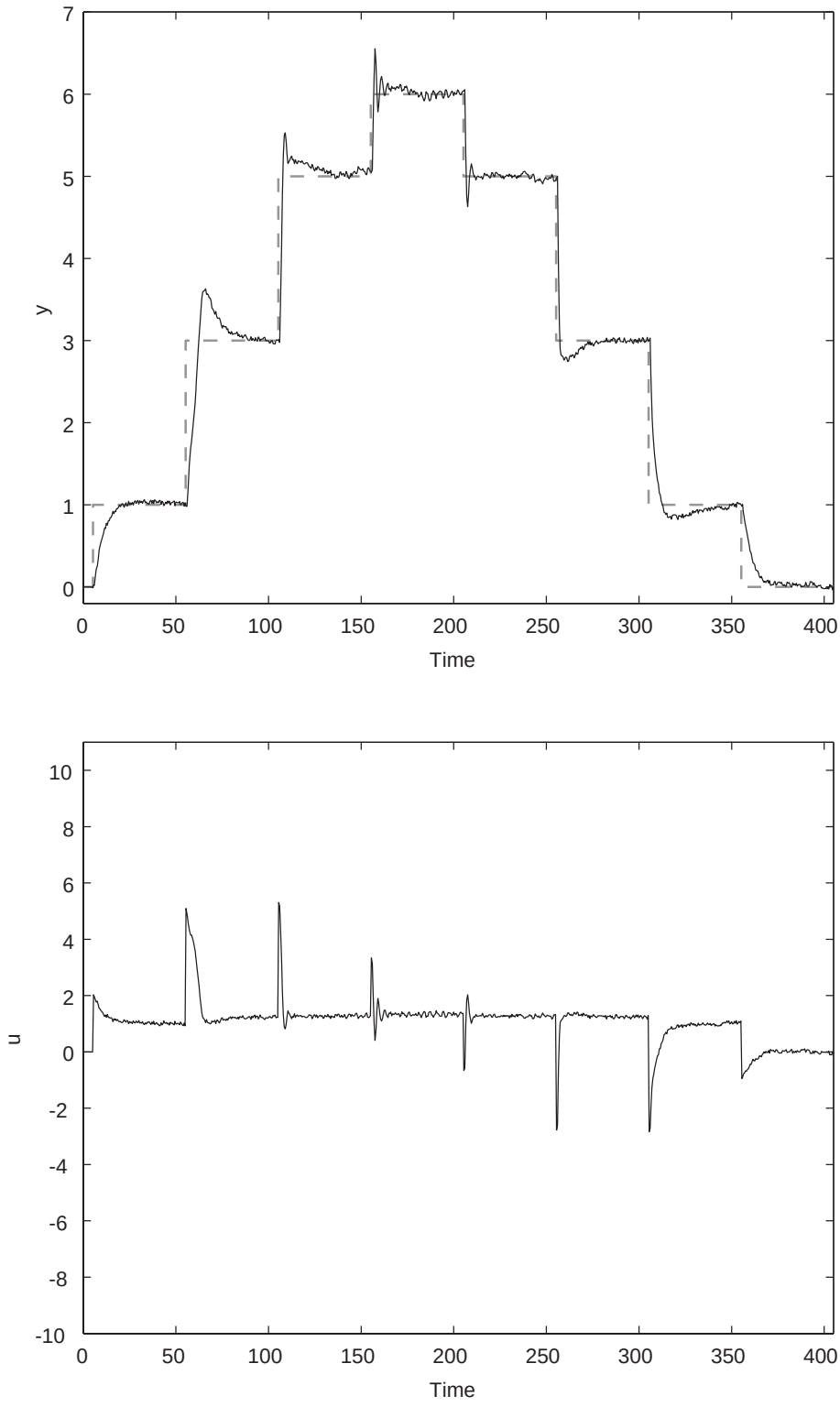

Figure 4.15: Servo response of the proposed IMC design in the presence of noise

4.4 Conclusion

By incorporating the JITL into IMC framework, an adaptive IMC design methodology is developed for nonlinear process control. The IMC controller parameters are updated not only based on a set of linear models identified on-line by the JITL, but also its filter parameter is adjusted on-line by an updating algorithm derived from the Lyapunov method. Compared with the previous nonlinear IMC controller design methods, it is straightforward for the proposed method to obtain the model inverse, and consequently the design of IMC controller can be carried out readily. Simulation results are presented to demonstrate the advantage of the proposed adaptive IMC design over its conventional counterpart.

Chapter 5

Self-Tuning PID Controllers Using the JITL Technique

5.1 Introduction

The PID controllers have received widespread use in the process industries primarily because of its simple structure, ease of implementation, and robustness in operation. Most of the tuning rules for PID controllers are based on a linear process model obtained either through a step test or by linearizing a nonlinear process model around the nominal operating condition. Although the use of a linear model makes the tuning of PID controller simple, performance of the conventional PID controller might degrade or even become unstable when the underlying process dynamics are nonlinear and time-varying in nature, which are the characteristics of many industrial chemical and biochemical processes. To improve the control performance, several schemes of incorporating nonlinear control techniques in the design of PID controller have been developed in recent years. For example, Riverol and Napolitano (2000)

proposed an adaptive PID controller whose parameters are adjusted on-line by a neural network, while Chen and Huang (2004) designed adaptive PID controller based on the instantaneous linearization of a neural network model. Altinten et al. (2004) applied the genetic algorithm to the optimal tuning of a PID controller on-line. Bisowarno et al. (2004) applied two adaptive PI control strategies for reactive distillation. Andrasik et al. (2004) made use of a hybrid model consisting of a neural network and a simplified first-principles model to design a neural PID-like controller. Yamamoto and Shah (2004) developed an adaptive PID controller using recursive least squares for on-line identification of multivariable system. Shahrokhi and Baghmisheh (2005) designed an adaptive IMC-PID controller based on the local models estimated by the recursive least squares method to control a fixed-bed reactor. Similar approaches for adjusting PID controller parameters on-line were investigated based on the multiple linearized models obtained by factorization algorithm and lazy learning identification method at each sampling instant (Ho et al., 1999; Alpbaz et al., 2006; Pan et al., 2007). In these works, basically, the parameters of the process model are updated with respect to the current process condition and then PID parameters are computed by the corresponding adaptation algorithm and implemented. However, these adaptation algorithms employed in the previous results are inadequate to address the convergence of the predicted tracking error. Toward this end, Chang et al. (2002) derived a stable adaptation mechanism in the continuous time domain by the Lyapunov approach such that the PID controller tracks a pre-specified feedback linearization control asymptotically.

Motivated by the above consideration, a self-tuning PID controller in the discrete time systems based on the Just-in-Time Learning (JITL) technique is proposed in

this chapter. The JITL is considered not only because its prediction capability for nonlinear processes can match that obtained by the neural network, but also its inherent adaptive nature. In the proposed method, a set of linear models obtained on-line by the JITL provides the information to adjust PID controller by a self-tuning algorithm derived by the Lyapunov method to guarantee the convergence of JITL's predicted tracking error. A literature example is presented to illustrate the proposed control strategy and a comparison with its conventional counterparts is made.

5.2 Self-Tuning PID Controller Design

The proposed self-tunign PID controller design is depicted in Figure 5.1, where the JITL technique is mainly used to identify the current process dynamics at each sampling instant. In the proposed self-tuning PID design, the following second-order ARX model is employed in the JITL algorithm:

$$\hat{y}(k) = \alpha_1^k y(k-1) + \alpha_2^k y(k-2) + \beta_1^k u(k-1) \tag{5.1}$$

where $\hat{y}(k)$ is the predicted output by the JITL at the k-th sampling time, $y(k-1)$ and $u(k-1)$ are the output and manipulated variables at the $(k-1)$-th sampling time, α_1^k, α_2^k and β_1^k are the model coefficients at the k-th sampling time.

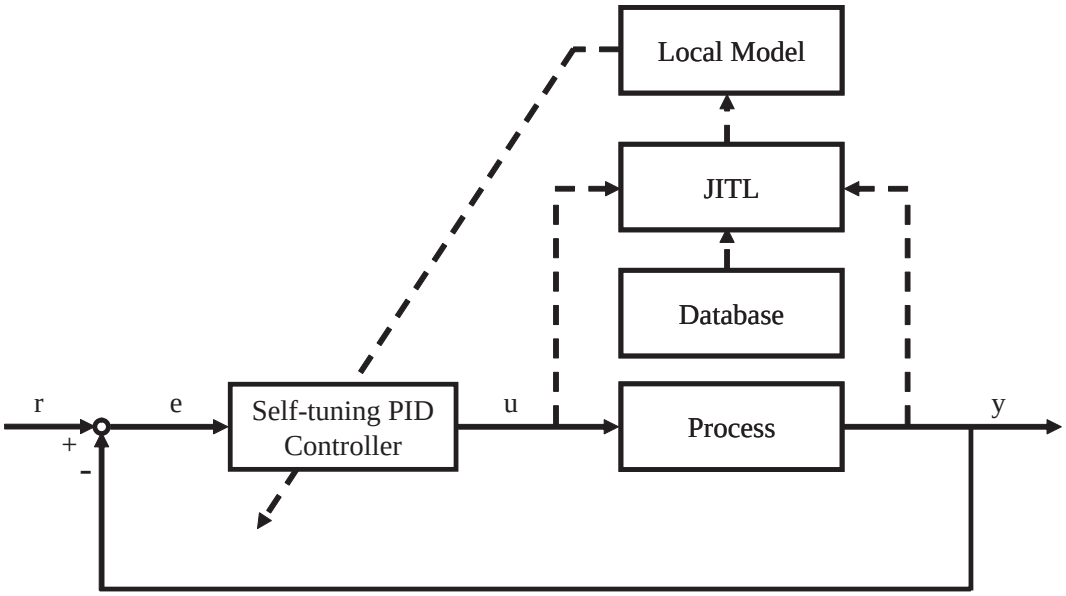

Figure 5.1: JITL based self-tuning PID control system

The control law of the proposed PID controller is given by:

$$u(k) = u(k-1) + w_1(k)e(k) + w_2(k)\Delta e(k) + w_3(k)\delta e(k) \tag{5.2}$$

where $w_1(k)$, $w_2(k)$, and $w_3(k)$ are the parameters of PID controller at the k-th sampling instant, $e(k)$ 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, and $\delta e(k) = \Delta e(k) - \Delta e(k-1)$.

Since the controller parameters w_i are constrained to be positive or negative, the following function is introduced to map the set of positive (or negative) number to the set of real number:

$$\zeta_i(k) = \begin{cases} \ln[w_i(k)], & \text{if} \quad w_i(k) \geq 0 \\ \ln[-w_i(k)], & \text{if} \quad w_i(k) < 0 \end{cases}, i = 1 \sim 3 \tag{5.3}$$

where $\zeta_i(k)$ is a real number. In the sequel, an updating algorithm will be developed to adjust $\zeta_i(k)$ online, and subsequently the PID parameters $w_i(k)$ can be easily

calculated by Eq. (5.3).

To facilitate the subsequent development, the following vectors related to the controller parameters are defined as:

$$w(k) = \begin{bmatrix} w_1(k) & w_2(k) & w_3(k) \end{bmatrix}^{\mathrm{T}} \tag{5.4}$$

$$\zeta(k) = \begin{bmatrix} \zeta_1(k) & \zeta_2(k) & \zeta_3(k) \end{bmatrix}^{\mathrm{T}} \tag{5.5}$$

In order to update the parameter $\zeta_i(k)$ at each sampling time so that the JITL's predicted output converges to the desired set-point trajectory, the following Lyapunov function is considered:

$$V(k) = \gamma e_{\mathrm{r}}^2(k) \tag{5.6}$$

where $e_{\mathrm{r}}(k)$ is the predicted tracking error defined as $e_{\mathrm{r}}(k) = r(k) - \hat{y}(k)$ and γ is a positive constant.

Define

$$e_{\mathrm{r}}(k+1) = e_{\mathrm{r}}(k) + \Delta e_{\mathrm{r}}(k+1) \tag{5.7}$$

The increment of the Lyapunov function, $\Delta V(k)$, is obtained by

$$\begin{aligned} \Delta V(k) &= V(k+1) - V(k) \\ &= \gamma e_{\mathrm{r}}^2(k+1) - \gamma e_{\mathrm{r}}^2(k) \\ &= 2\gamma e_{\mathrm{r}}(k)\Delta e_{\mathrm{r}}(k+1) + \gamma \Delta e_{\mathrm{r}}^2(k+1) \end{aligned} \tag{5.8}$$

In Eq. (5.8), $\Delta e_{\mathrm{r}}(k+1)$ can be further expressed as

$$\begin{aligned} \Delta e_{\mathrm{r}}(k+1) &= \frac{\partial e_{\mathrm{r}}(k+1)}{\partial k} \\ &= \frac{\partial \left[r(k+1) - \hat{y}(k+1) \right]}{\partial u(k)} \frac{\partial u(k)}{\partial w(k)} \frac{\partial w(k)}{\partial \zeta(k)} \frac{\partial \zeta(k)}{\partial k} \\ &= -\frac{\partial \hat{y}(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial w(k)} \frac{\partial w(k)}{\partial \zeta(k)} \Delta \zeta(k) \end{aligned} \tag{5.9}$$

where

$$\frac{\partial \hat{y}(k+1)}{\partial u(k)} = \beta_1^{k+1}$$
$$\frac{\partial u(k)}{\partial w(k)} = e_{\mathrm{u}}(k) = \begin{bmatrix} e(k) & \Delta e(k) & \delta e(k) \end{bmatrix}$$
$$\frac{\partial w(k)}{\partial \zeta(k)} = \begin{bmatrix} w_1(k) & 0 & 0 \\ 0 & w_2(k) & 0 \\ 0 & 0 & w_3(k) \end{bmatrix} \tag{5.10}$$

Based on the on-going analysis, the following theorem provides the theoretical basis for the convergence property of the proposed updating algorithm for $\zeta(k)$.

Theorem 1. Let $0 < \eta_i(k) < 2$ and the parameter vector $\zeta(k)$ is updated by the following equation,

$$\zeta(k+1) = \zeta(k) + \frac{1}{\beta_1^{k+1}} \Omega(k) \left[\frac{\partial w(k)}{\partial \zeta(k)}\right]^{-1} \frac{e_{\mathrm{u}}(k)^{\mathrm{T}} e_{\mathrm{r}}(k)}{e_{\mathrm{u}}(k) e_{\mathrm{u}}(k)^{\mathrm{T}}} \tag{5.11}$$

where

$$\Omega(k) = \begin{bmatrix} \eta_1(k) & 0 & 0 \\ 0 & \eta_2(k) & 0 \\ 0 & 0 & \eta_3(k) \end{bmatrix}$$

then the tracking error $e_{\mathrm{r}}(k)$ is guaranteed to converge to zero asymptotically.

Proof: Based on Eqs. (5.9) and (5.11), Eq. (5.8) is expressed as:

$$
\begin{aligned}
&\Delta V(k) \\
&= -2\gamma e_{\mathrm{r}}(k)\frac{\partial \hat{y}(k+1)}{\partial u(k)}\frac{\partial u(k)}{\partial w(k)}\frac{\partial w(k)}{\partial \zeta(k)}\Delta\zeta(k) + \gamma\left(-\frac{\partial \hat{y}(k+1)}{\partial u(k)}\frac{\partial u(k)}{\partial w(k)}\frac{\partial w(k)}{\partial \zeta(k)}\Delta\zeta(k)\right)^2 \\
&= -2\gamma e_{\mathrm{r}}(k)\beta_1^{k+1}e_{\mathrm{u}}(k)\frac{\partial w(k)}{\partial \zeta(k)}\frac{1}{\beta_1^{k+1}}\Omega(k)\left[\frac{\partial w(k)}{\partial \zeta(k)}\right]^{-1}\frac{e_{\mathrm{u}}(k)^{\mathrm{T}}e_{\mathrm{r}}(k)}{e_{\mathrm{u}}(k)e_{\mathrm{u}}(k)^{\mathrm{T}}} \\
&\quad +\gamma\left(\beta_1^{k+1}e_{\mathrm{u}}(k)\frac{\partial w(k)}{\partial \zeta(k)}\frac{1}{\beta_1^{k+1}}\Omega(k)\left[\frac{\partial w(k)}{\partial \zeta(k)}\right]^{-1}\frac{e_{\mathrm{u}}(k)^{\mathrm{T}}e_{\mathrm{r}}(k)}{e_{\mathrm{u}}(k)e_{\mathrm{u}}(k)^{\mathrm{T}}}\right)^2 \\
&= -2\gamma e_{\mathrm{r}}^2(k)e_{\mathrm{u}}(k)\Omega(k)\frac{e_{\mathrm{u}}(k)^{\mathrm{T}}}{e_{\mathrm{u}}(k)e_{\mathrm{u}}(k)^{\mathrm{T}}} + \gamma e_{\mathrm{r}}^2(k)\left(e_{\mathrm{u}}(k)\Omega(k)\frac{e_{\mathrm{u}}(k)^{\mathrm{T}}}{e_{\mathrm{u}}(k)e_{\mathrm{u}}(k)^{\mathrm{T}}}\right)^2 \\
&= -2\gamma e_{\mathrm{r}}^2(k)\frac{\eta_1 e^2(k)+\eta_2\Delta e^2(k)+\eta_3\delta e^2(k)}{e^2(k)+\Delta e^2(k)+\delta e^2(k)} \\
&\quad +\gamma e_{\mathrm{r}}^2(k)\left(\frac{\eta_1 e^2(k)+\eta_2\Delta e^2(k)+\eta_3\delta e^2(k)}{e^2(k)+\Delta e^2(k)+\delta e^2(k)}\right)^2 \\
&= \gamma e_{\mathrm{r}}^2(k)\frac{\eta_1 e^2(k)+\eta_2\Delta e^2(k)+\eta_3\delta e^2(k)}{e^2(k)+\Delta e^2(k)+\delta e^2(k)} \\
&\quad \times\left(\frac{(\eta_1-2)e^2(k)+(\eta_2-2)\Delta e^2(k)+(\eta_3-2)\delta e^2(k)}{e^2(k)+\Delta e^2(k)+\delta e^2(k)}\right)
\end{aligned}
\tag{5.12}
$$

It is evident from Eq. (5.12) that $\Delta V(k)$ is always negative if $0 < \eta_i(k) < 2$ holds, meaning that tracking error $e_{\mathrm{r}}(k)$ is guaranteed to converge to zero by using the updating algorithm, Eq. (5.11), to design $\zeta(k+1)$. This completes the proof.

One remark about Theorem 1 is the determination of $\eta(k)$. Generally, a larger $\eta(k)$ in the specified range [0 2] would lead to faster convergence but might result in overshoot and oscillatory response, while a smaller $\eta(k)$ has the opposite effect. Owing to the lacking of systematic guidelines for the determination of $\eta(k)$, it is usually chosen experimentally for each problem. In the proposed design, the following rules are used to update the learning rates: (i) if the increment of $|e_{\mathrm{r}}(k)|$ is more than the threshold, the PID parameters remain unchanged and learning rate is decreased by a factor $l_{dec,i}$, i.e. $\eta_i(k+1) = l_{dec,i}\eta_i(k)$; (ii) if the absolute value of the change of $|e_{\mathrm{r}}(k)|$ is within the threshold, only the PID parameters are updated;

otherwise, (iii) both PID parameters are updated and learning rate is adjusted by $\eta_i(k) = l_{inc,i}\eta_i(k-1)$.

The implementation of the proposed self-tuning PID controller design is summarized as follows:

Step 1 Given the initial database for the JITL, initialize the PID controller parameters and their respective learning rates;

Step 2 Given the current process output $y(k)$, compute the manipulated variable $u(k)$ from Eq. (5.2);

Step 3 The JITL's database is updated by the current process data if the absolute value of prediction error between the JITL's output and the current process output is larger than the specified threshold;

Step 4 Obtain ARX model for the next sampling instant by using the current process data and JITL algorithm, followed by updating $\eta_i(k)$ and $\zeta_i(k)$. Consequently, PID parameters at the next sampling instant, $w_i(k+1)$, are calculated by using Eq. (5.3)

Step 5 Set $k = k + 1$ and go to Step 2.

5.3 Examples

Example 1 The proposed self-tuning PID strategy is applied to a polymerization reactor example discussed earlier in Chapter 4. The model parameters and steady-state operation condition can be found in Tables 4.1 and 4.2. The proposed self-tuning PID controller design is based on the same database and parameters used for

the JITL algorithm mentioned in Chapter 4. In addition, the IMC and RLS-based adaptive PID controllers provided in Chapter 4 serve as the benchmark design for comparison purpose.

To compare the performances of three controllers, successive set-point changes between 25000.5 kg/kmol and 12500 kg/kmol are conducted. As can be seen from Figure 5.2, the proposed self-tuning PID controller has better performance than those achieved by the other two controllers, resulting in the respective reductions of the Mean Absolute Error (MAE) by 29.4% and 28.8% as compared with the linear IMC and RLS-based adaptive PID controllers. Note that the symbol "$*$" in Figure 5.2 denotes the sampling instants at which JITL's database is updated. Figure 5.3 shows the updating of PID parameters and learning rates of the proposed design in the aforementioned closed-loop responses.

Figures 5.4 and 5.5 compare the disturbance rejection capability of three controllers with respect to unmeasured $\pm 10\%$ step disturbances in the inlet initiator concentration $C_{\mathrm{I_{in}}}$. Again, the proposed self-tuning PID controller has superior control performance over the other two controllers. Table 5.1 summarizes the performance achieved by the proposed PID controller. Next, to evaluate the robustness of the proposed control strategy, it is assumed that there exist 10% modeling error in the kinetic parameter k_{I} and 20% error in the coefficients of the D_1 and M_{m}. As can be seen from Figure 5.6, the proposed controller outperforms the linear IMC controller, as also evidenced by the reduction of MAE by 28.5% for the aforementioned set-point changes. In this case, adaptive PID controller based on RLS model yields steady-state off-set as discussed in Chapter 4. Lastly, 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 5.7, the proposed self-tuning controller can yield reasonably good control performance in the presence of process noise.

Table 5.1: Control performance comparison of three controllers

	Tracking error (MAE)		
	self-tuning PID (JITL)	Adaptive PID (RLS)	Linear IMC
Servo Response	695.25	976.49	984.27
Servo Response*	840.12	off-set	1174.70
+10% in $C_{I_{in}}$ at y=25000.5	35.17	45.94	46.42
+10% in $C_{I_{in}}$ at y=18750	57.49	69.50	90.16
+10% in $C_{I_{in}}$ at y=13500	96.1	96.90	141.15
-10% in $C_{I_{in}}$ at y=25000.5	40.37	55.12	56.88
-10% in $C_{I_{in}}$ at y=18750	64.32	64.34	110.81
-10% in $C_{I_{in}}$ at y=13500	84.25	97.31	171.93

* In the presence of modeling error

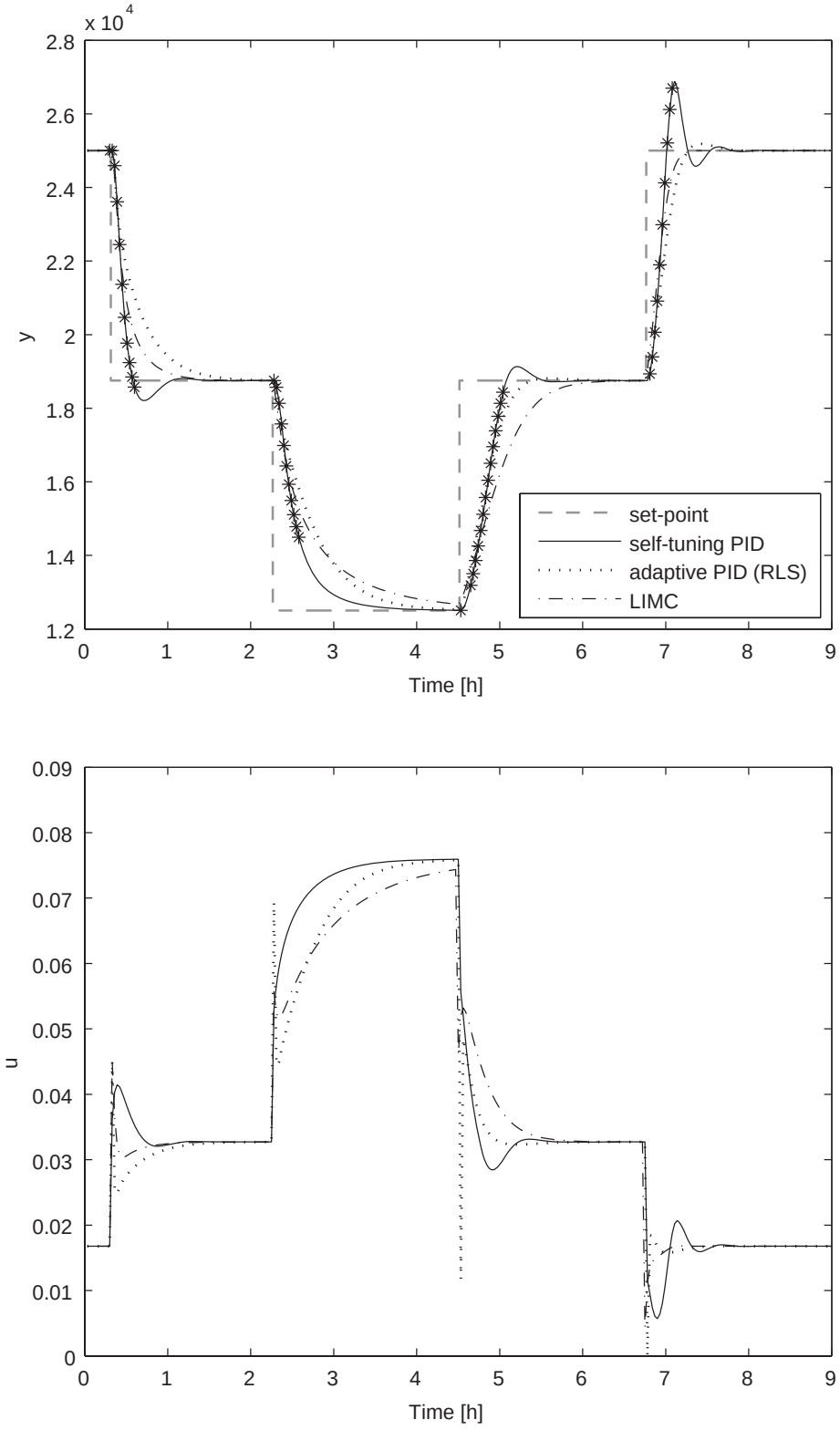

Figure 5.2: Servo responses of three controller designs ($*$: database update)

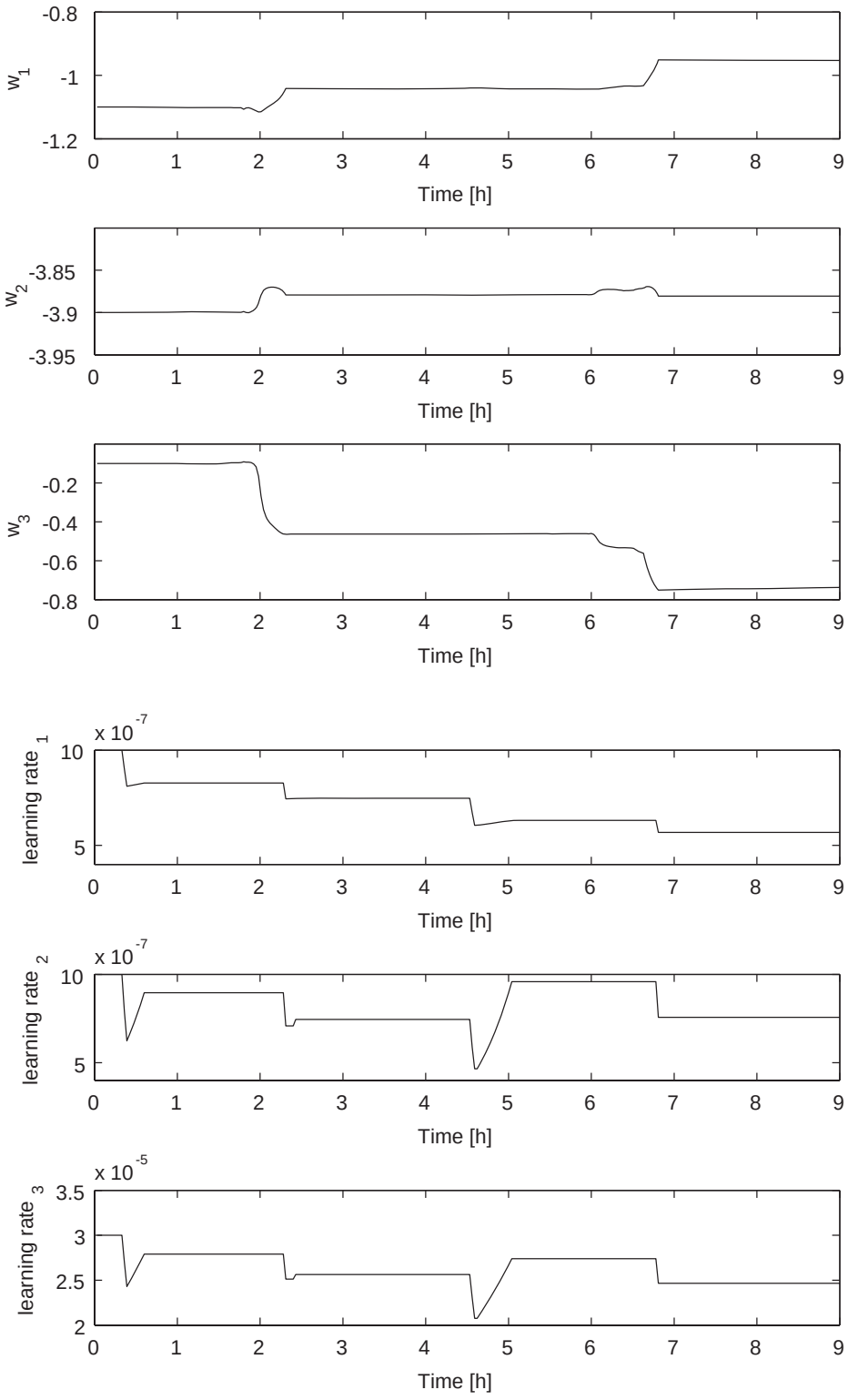

Figure 5.3: Updating of the PID parameters and learning rates for servo response

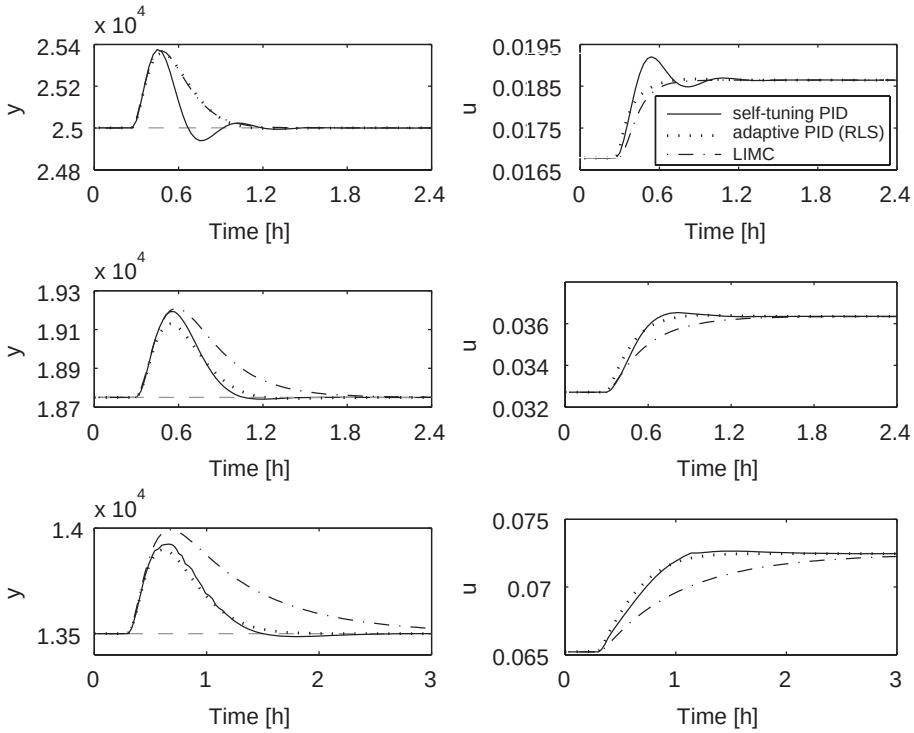

Figure 5.4: Closed-loop responses for -10% step change in $C_{\mathrm{I_{in}}}$

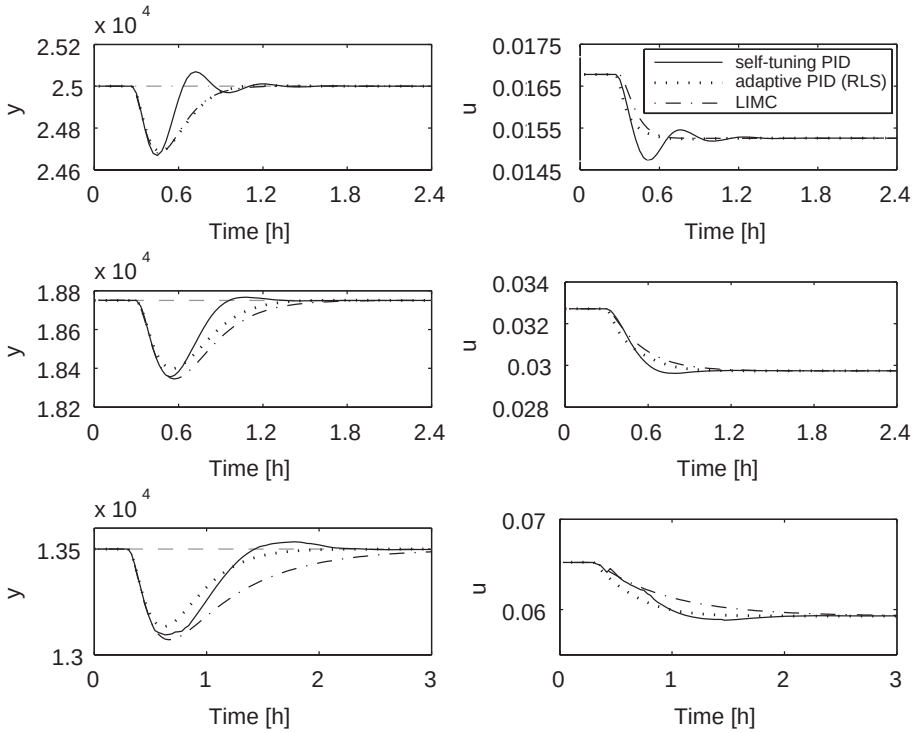

Figure 5.5: Closed-loop responses for +10% step change in $C_{\mathrm{I_{in}}}$

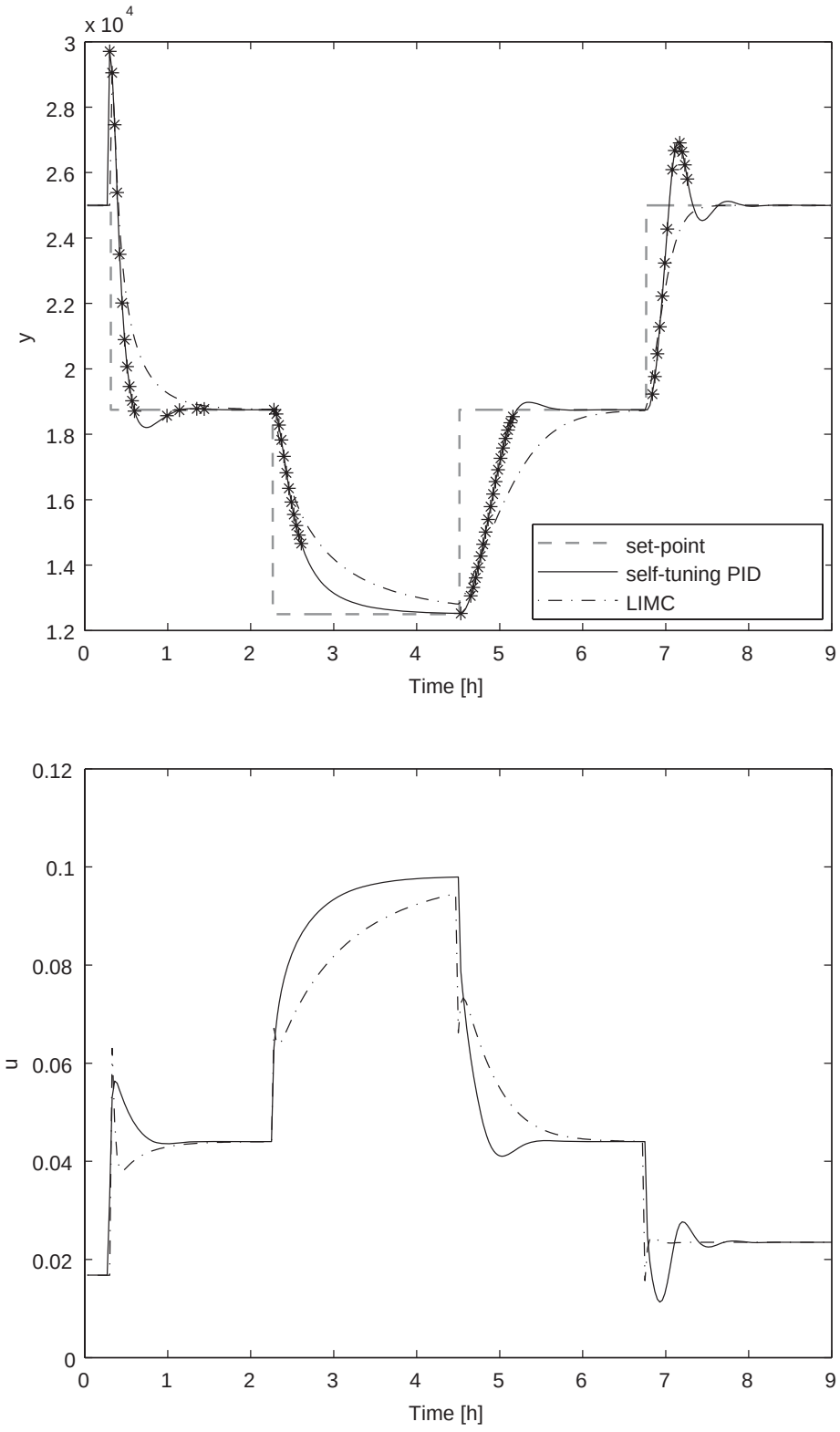

Figure 5.6: Servo responses of the self-tuning PID and IMC designs in the presence of modeling error ($*$: database update)

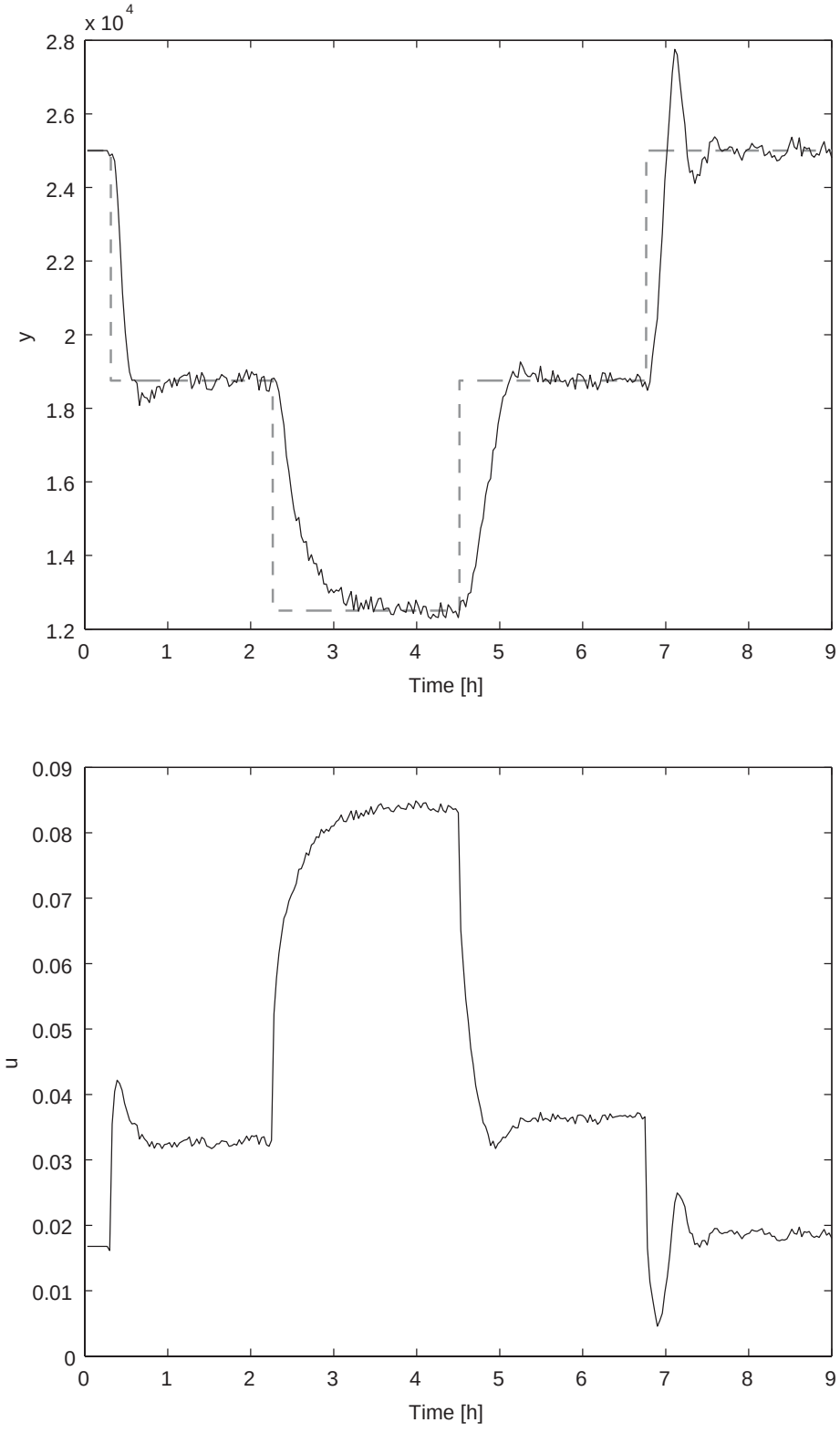

Figure 5.7: Servo response of the self-tuning PID controller in the presence of noise

Example 2 The proposed self-tuning PID strategy is applied to the first-order plus dead-time process discussed in Chapter 4. Like in Chpater 4, a first-order ARX model is used as the local model and the parameters chosen for JITL algorithm are as follows: $\kappa = 0.95$, $k_{\min} = 6$ and $k_{\max} = 60$. The initial PI controller is specified by $w_1 = 0.3$ and $w_2 = 1.6$ with initial learning rates given by $\eta_1 = 4 \times 10^{-11}$ and $\eta_2 = 1 \times 10^{-10}$. For the purpose of comparison, the gain-scheduling PI controller given in Table 4.5 is considered as benchmark design.

Figures 5.8 and 4.12 compare the servo performances of two controllers for the successive set-point changes between 0 and 6. It can be seen that the proposed self-tuning PI design has better performance than that achieved by the gain-scheduling PI controller because the latter exhibits sustained oscillation around the set-point at 3 and large overshoot for the set-point change from 3 to 5. Next, to evaluate the robustness of the proposed control strategy, 10% modeling error in the process parameter $\tau_2(y)$ is assumed and the resulting servo responses of two controllers are compared in Figures 5.9 and 4.14. It is evident that the proposed controller still maintains superior control performance in the aforementioned set-point changes. Lastly, to study the effect of process noise on the proposed design, both process input and output are corrupted by 3% Gaussian white noise. As shown in Figure 5.10, the proposed controller can yield reasonably good control performance in the presence of process noise.

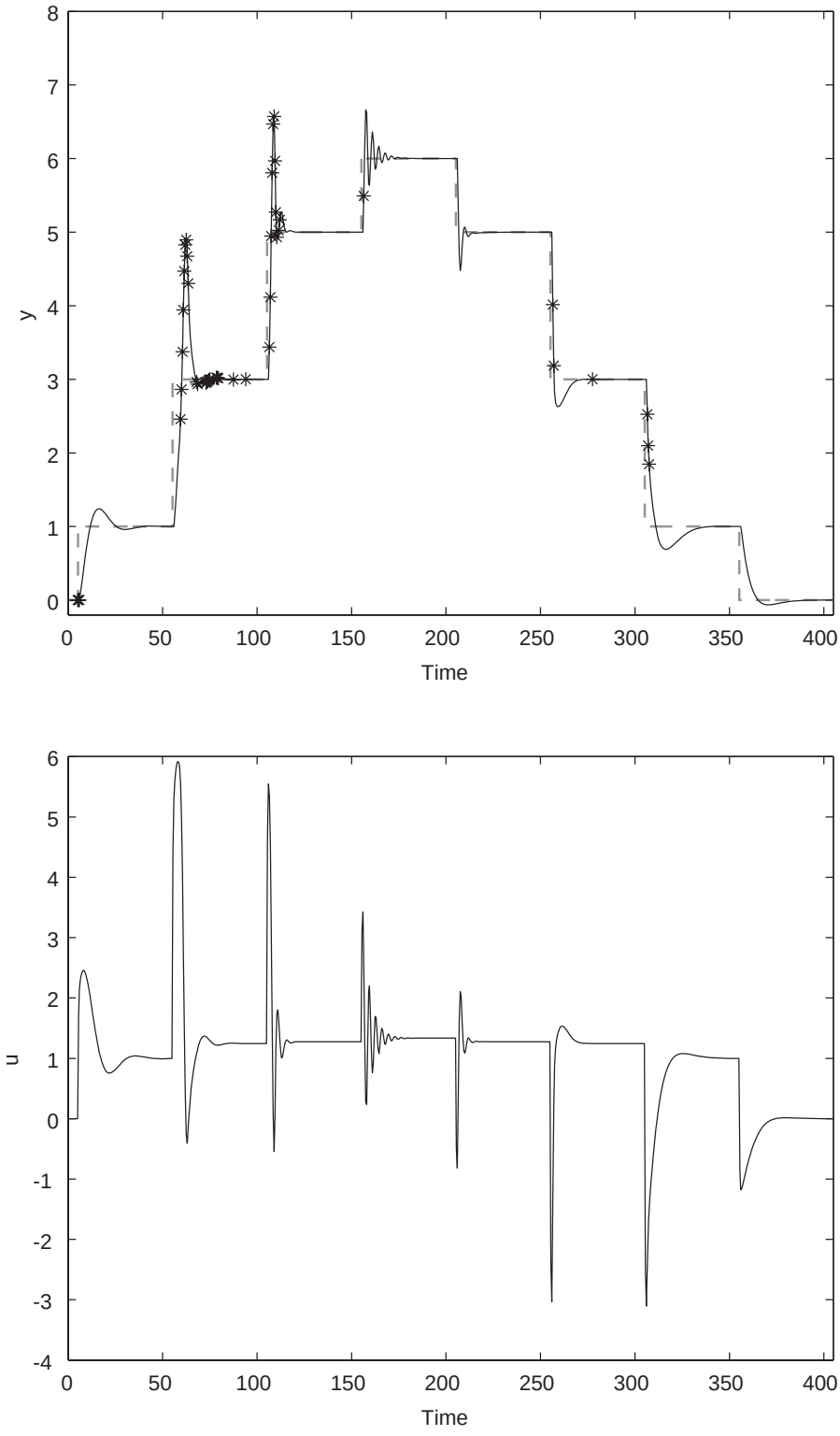

Figure 5.8: Servo response of the self-tuning PI design (∗: database update)

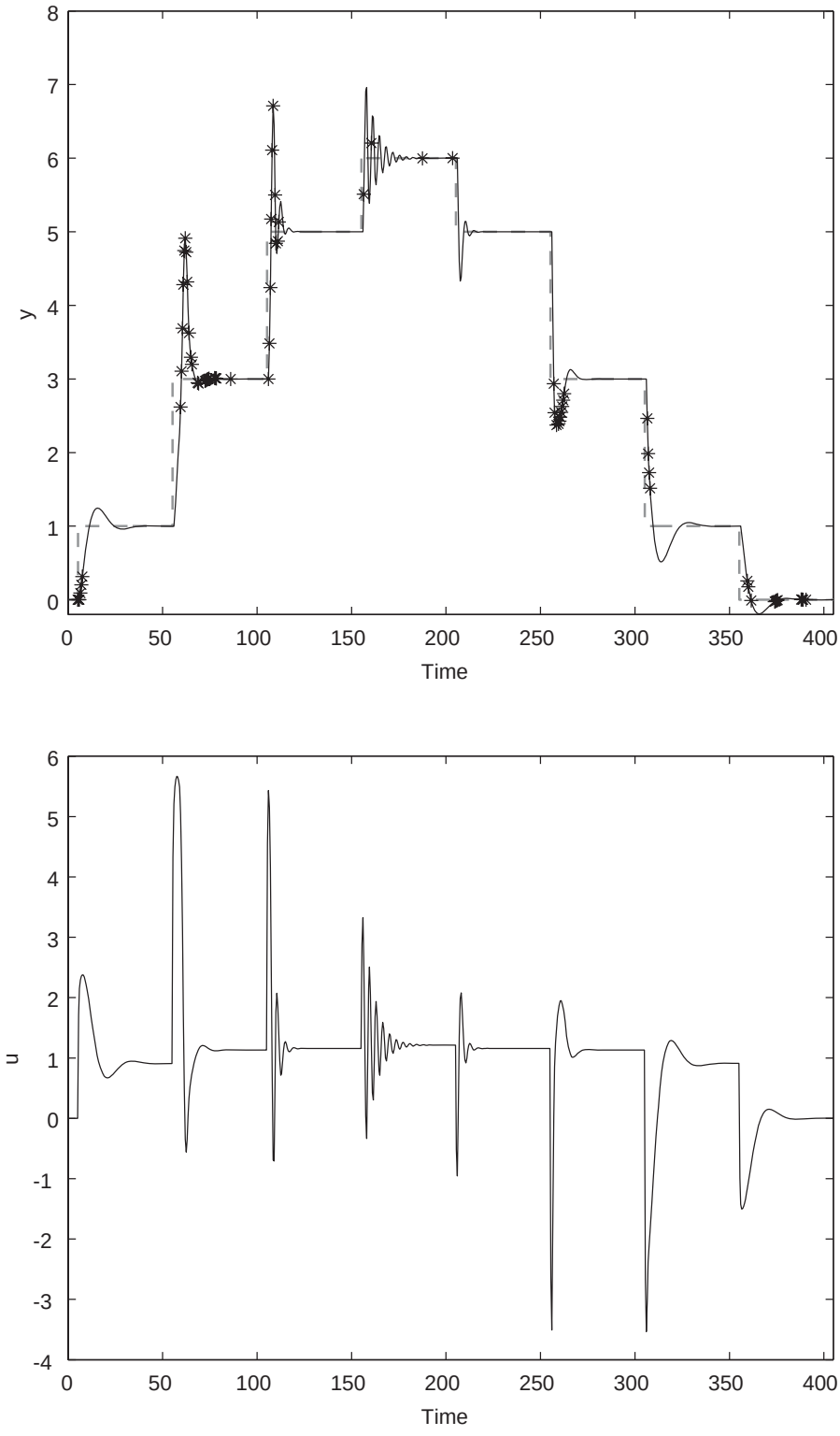

Figure 5.9: Servo response of the self-tuning PI controller under +10% modeling error in τ_2 ($*$: database update)

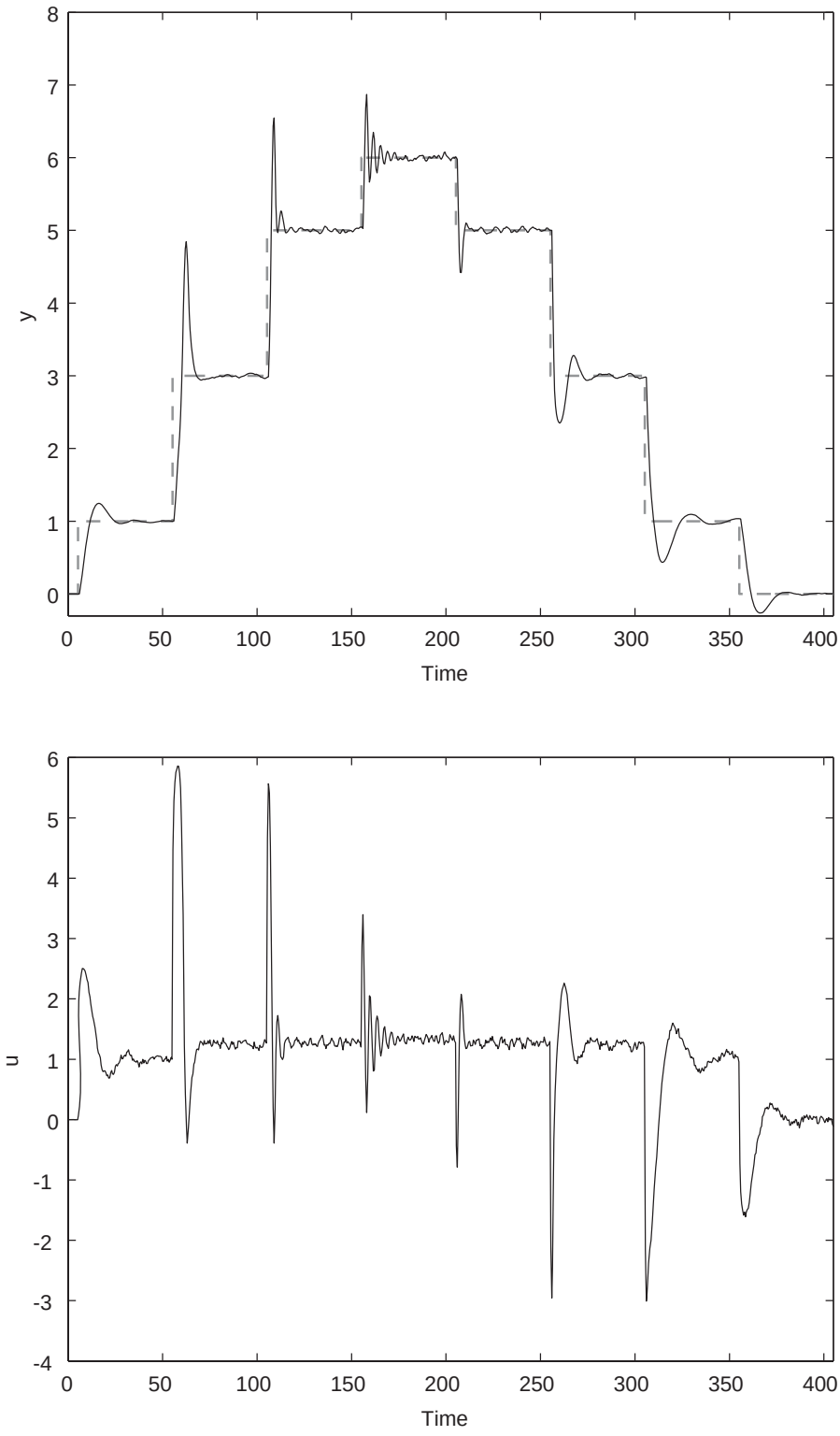

Figure 5.10: Servo response of the self-tuning PI controller in the presence of noise

5.4 Conclusion

By incorporating the JITL into PID framework, a self-tuning PID design methodology is developed for nonlinear process control in this chapter. PID controller parameters are updated not only based on the information provided by the JITL, but also an updating algorithm derived from the Lyapunov method. Compared with the previous neural network based PID controller designs, the proposed method is more straightforward for implementation. Simulation results are presented to demonstrate the advantage of the proposed adaptive PID controller design over its conventional counterpart.

Chapter 6

Generalized Predictive Control Using the JITL Technique

6.1 Introduction

Model predictive control (MPC) based on linear models, for example dynamic matrix control (Cutler and Ramaker, 1979), quadratic dynamic matrix control (Garcia and Morshedi, 1986), and generalized predictive control (Clarke et al. 1987), has gained wide-spread acceptance as an advanced control strategy in chemical process industries. This is primarily due to their ability to handle process constraints, time delay, and multivariable systems in a unified design framework. In MPC, a dynamic process model is first developed to predict the future process outputs in the prediction horizon, by which future control actions are computed by minimizing a pre-specified cost function. Therefore, the effectiveness of MPC relies heavily on the availability of a reasonably accurate process model. As many chemical processes are highly nonlinear and may be operated in a range of operating points, it is clear

that MPC algorithms based on linear process models can result in poor control performance. As a result, various variants of MPC techniques have been studied and extended to nonlinear systems. Although a number of nonlinear MPC designs have been developed based on the first-principle models, their applications have been limited due to the difficulty associated with the development of a reasonably accurate first-principles model and the extensive computational burden as inevitably required by solving the associated on-line nonlinear optimization problems. To reduce the computational burden arising from the use of nonlinear models in nonlinear MPC design, Venkateswarlu and Gangiah (1997) utilized a recursive least squares (RLS) algorithm to update the local model in a nonlinear generalized predictive control strategy. However, the RLS algorithm can produce poor estimates of system parameters if the online process input and output data do not meet excitation conditions. Another popular nonlinear MPC techniques by incorporating empirical models like neural networks (Saint-Donat et al., 1991; Pottmann and Seborg, 1997; Chu et al., 2003), fuzzy models (Kavsek-Biasizzo et al., 1997; Fischer et al., 1998; Abonyi et al., 2000; Mahfouf et al., 2000), fuzzy neural networks (Lu and Tsai, 2007), and local model networks (Prasad et al., 1998) have been investigated and developed in the literature. However, the use of neural network in nonlinear MPC design is computationally demanding due to the on-line optimization required to compute the control signals. For fuzzy models and local model networks, the problem of how to partition the operating regimes remains an ad-hoc procedure and therefore a prior knowledge of the processes, which may not be readily accessible in most practical cases, has to be exploited for the determination of the model structure. Another fundamental limitation of these types of modeling approaches 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 model structure (e.g. number of hidden neurons in the neural networks models and number of fuzzy rules in fuzzy models) and model parameters may need to be modified simultaneously. Evidently, this process is not only time-consuming but also it will interrupt the plant operation, if these models are used in controller design.

In this chapter, a generalized predictive control (GPC) strategy based on the Just-in-Time Learning (JITL) technique is proposed. The JITL is considered not only because of its prediction capability for nonlinear systems but also its inherent adaptive nature. This latter feature makes JITL an attractive alternative to be incorporated into the nonlinear MPC design so that the aforementioned problems encountered by the global models can be alleviated. Furthermore, the computational burden is reduced by modeling the nonlinear systems by a set of local models obtained on-line by the JITL. The current local model at each sampling instant is treated as the process model in GPC design where the optimal changes in the manipulated variable are determined by solving a quadratic optimization problem formulated in the GPC design framework. Literature examples are presented to illustrate the proposed control strategy and a comparison with its conventional counterparts is made.

6.2 JITL-Based Generalized Predictive Controller Design

Since the local models obtained by the JITL algorithm are ARX models, GPC design framework (Clarke et al., 1987) is considered for the development of the proposed nonlinear MPC design. In the proposed GPC design, the following second-order ARX model is employed in the JITL algorithm at each sampling time:

$$\hat{y}(k) = \alpha_1^k y(k-1) + \alpha_2^k y(k-2) + \beta_1^k u(k-n_d-1) \tag{6.1}$$

where n_d denotes the time delay.

Similar to the conventional GPC design, the aim of the proposed nonlinear GPC scheme is to find the optimal future changes in the manipulated variable that make the future process output track the reference trajectory as closely as possible in the presence of system constraints and disturbances. The above control performance requirement can be expressed by the following optimization problem:

$$\min_{\Delta u(k),\Delta u(k+1),\cdots,\Delta u(k+N_u-1)} \sum_{i=1}^{N_p} \left(r(k+i) - y(k+i/k)\right)^2 + \sum_{i=1}^{N_u} \mu_i \left(\Delta u(k+i-1)\right)^2 \tag{6.2}$$

subject to the following input constraints:

$$u_{\min} \leq u(k+i-1) \leq u_{\max}, \qquad i = 1, 2, \cdots, N_u \tag{6.3}$$

$$\Delta u_{\min} \leq \Delta u(k+i-1) \leq \Delta u_{\max}, \qquad i = 1, 2, \cdots, N_u \tag{6.4}$$

where $r(k+i)$ is the future set-point, $y(k+i/k)$ is the prediction of the future process output at the $(k+i)$-th sampling instant in the prediction horizon N_p, $\Delta u(k+i-1)$ is the future change in the manipulated variable at the $(k+i-1)$-th sampling instant

in the control horizon N_u, μ_i is a weighting factor, $u_{\min}$ and $u_{\max}$ are the lower and upper limits of manipulated variable, and $\Delta u_{\min}$ and $\Delta u_{\max}$ are the lower and upper limits for the increment of manipulated variable.

Equations (6.2) to (6.4) can be rewritten in the following vector form:

$$\min_{\mathbf{\Delta u}} (\mathbf{r}-\mathbf{y})^{\mathrm{T}}(\mathbf{r}-\mathbf{y}) + \Delta\mathbf{u}^{\mathrm{T}}\mathbf{M}\Delta\mathbf{u} \tag{6.5}$$

subject to the input constraints

$$\mathbf{C_u}\Delta\mathbf{u} \geq \mathbf{c_f} \tag{6.6}$$

where $\mathbf{r} = \begin{bmatrix} r(k+1) & \cdots & r(k+N_p) \end{bmatrix}^{\mathrm{T}}$, $\mathbf{y} = \begin{bmatrix} y(k+1/k) & \cdots & y(k+N_p/k) \end{bmatrix}^{\mathrm{T}}$, $\Delta\mathbf{u} = \begin{bmatrix} \Delta u(k) & \cdots & \Delta u(k+N_u-1) \end{bmatrix}^{\mathrm{T}}$, $\mathbf{M}$ is a diagonal matrix consisting of μ_i, and $\mathbf{C_u}$ and $\mathbf{c_f}$ are the corresponding matrix and vector accounting for the inequalities of input constraints.

In order to solve the aforementioned quadratic optimization problem, the objective function is requited to be formulated as a function of the future changes in the manipulated variable. To this end, consider the ARX local model expressed by the following CARMA (Controlled Auto - Regressive and Moving Average) model:

$$A(z^{-1})\Delta y(k) = B(z^{-1})\Delta u(k-1) \tag{6.7}$$

where z^{-1} denotes the backward shift operator and

$$A(z^{-1}) = 1 - \alpha_1^k z^{-1} - \alpha_2^k z^{-2} \tag{6.8}$$

$$B(z^{-1}) = \beta_1^k z^{-n_d} \tag{6.9}$$

To derive the ith-step ahead prediction of $y(k+1/k)$, consider a set of Diophantine equations:

$$1 = E_i(z^{-1})A(z^{-1})\Delta + z^{-i}F_i(z^{-1}), \qquad i = 1, 2, \cdots, N_p \tag{6.10}$$

where $E_i(z^{-1})$ and $F_i(z^{-1})$ are polynomials in z^{-i}. Multiplying Eq. (6.7) by $E_i(z^{-1})z^i$ yields:

$$E_i(z^{-1})A(z^{-1})\Delta y(k+i/k) = E_i(z^{-1})B(z^{-1})\Delta u(k+i-1) \tag{6.11}$$

Substituting Eq. (6.10) into above equation gets

$$y(k+i/k) = E_i(z^{-1})B(z^{-1})\Delta u(k+i-1) + F_i(z^{-1})y(k) \tag{6.12}$$

Subsequently, Eq. (6.12) is used to predict the future process outputs in the prediction horizon N_p:

$$\begin{aligned} y(k+1/k) &= G_1(z^{-1})\Delta u(k) + F_1(z^{-1})y(k) = \overbrace{g_1(z)\Delta u(k)}^{future} + f_1 \\ &\vdots \qquad \vdots \\ y(k+N_p/k) &= G_{N_p}(z^{-1})z^{Np-1}\Delta u(k) + F_{N_p}(z^{-1})y(k) = \overbrace{g_{N_p}(z)\Delta u(k)}^{future} + f_{N_p} \end{aligned} \tag{6.13}$$

where $G_i(z^{-1}) = E_i(z^{-1})B(z^{-1})$, $g_i(z)$ consists of those terms in $G_i(z^{-1})z^{i-1}$ with nonnegative exponent in z^i, and f_i is the remaining term calculated based on the past values of process output and input. Equation (6.13) can be written as

$$\mathbf{y} = \begin{bmatrix} y(k+1/k) & \cdots & y(k+N_p/k) \end{bmatrix}^{\mathrm{T}} = \mathbf{G}\Delta\mathbf{u} + \mathbf{h} \tag{6.14}$$

where $\mathbf{G}$ is a lower triangular matrix with elements consisting of the coefficients in $g_i(z)$ and $\mathbf{h} = \begin{bmatrix} f_1 & f_2 & \cdots & f_{N_p} \end{bmatrix}^{\mathrm{T}}$. By using Eq. (6.14), the objective function of Eq. (6.5) can be recast as a quadratic optimization problem as follows:

$$\min_{\Delta\mathbf{u}} (\mathbf{r} - \mathbf{h} - \mathbf{G}\Delta\mathbf{u})^{\mathrm{T}}(\mathbf{r} - \mathbf{h} - \mathbf{G}\Delta\mathbf{u}) + \Delta\mathbf{u}^{\mathrm{T}}\mathbf{M}\Delta\mathbf{u} \tag{6.15}$$

With aforementioned discussion, the implementation of the proposed GPC design is summarized as follows:

Step 1 Given the initial database for the JITL and the weighting matrix $\mathbf{M}$;

Step 2 Given the current process output $y(k)$, $\Delta\mathbf{u}$ is computed by solving the quadratic optimization problem of Eq. (6.15) subject to the constraints given in Eq. (6.6) and only $\Delta u(k)$ is implemented into the process;

Step 3 The JITL's database is updated by the current process data if the absolute value of prediction error between the JITL's output and the current process output is larger than the specified threshold;

Step 4 Obtain ARX model for the next sampling instant by using the current process data and JITL algorithm.

Step 5 Set $k = k + 1$ and go to Step 2.

6.3 Examples

Example 1 Consider the polymerization reactor studied in Chapters 4 and 5. The model parameters and steady-state operation condition can be found in Tables 4.1 and 4.2. In addition, the same database and parameters for the JITL used in Chapters 4 and 5 are employed.

To evaluate the servo performances of three controllers, successive set-point changes between 25000.5 kg/kmol and 12500 kg/kmol are considered. The input constraint is specified as $u \geq 0.007$ for three GPC designs. The parameters of three GPC designs are tuned to give their respective optimal tracking performances. For the proposed GPC design, the prediction horizon is $N_p = 7$, the control horizon $N_u = 1$, and the weighting matrix is $\mathbf{M} = 0.002\mathbf{I}$. In addition, $N_p = 6$, $N_u = 1$, and

$\mathbf{M} = 0\mathbf{I}$ are designed for linear GPC design that is based on the linear model given in Eq. (4.15), while $N_p = 10$, $N_u = 1$, and $\mathbf{M} = 0.002\mathbf{I}$ are used in the adaptive GPC design based on the RLS method. It is clear from Figure 6.1 that the proposed GPC design has faster response than those achieved by the other two GPC designs, resulting in the respective reductions of the Mean Absolute Error (MAE) by 11.8% and 14.3%.

Figures 6.2 and 6.3 compare the disturbance rejection capability of three controllers with respect to unmeasured ±10% step disturbances in the inlet initiator concentration $C_{\mathrm{I_{in}}}$. The proposed JITL-based GPC design gives better control performance over the other two GPC designs except at the nominal operating condition. Table 6.1 summarizes the control performances of three GPC designs. Next, to evaluate the robustness of the proposed control strategy, 10% modeling error in the kinetic parameter k_{I} and 20% error in the coefficients of the D_1 and M_{m} are assumed. As can be seen from Figure 6.4, the proposed controller still maintains better control performance over the other two controllers, leading to 2.6% and 17.8% reduction of MAEs compared with RLS-based GPC and linear GPC design, respectively. Lastly, 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 6.5, the proposed JITL-based GPC design can yield reasonably good control performance in the presence of process noise.

Table 6.1: Control performance comparison of three controllers

	Tracking error (MAE)		
	JITL-based GPC	RLS-based GPC	Linear GPC
Servo Response	419.10	475.66	488.18
Servo Response*	517.76	531.70	629.64
+10% in $C_{I_{in}}$ at y=25000.5	9.85	12.31	6.97
+10% in $C_{I_{in}}$ at y=18750	15.08	19.87	19.82
+10% in $C_{I_{in}}$ at y=13500	19.96	29.21	33.49
-10% in $C_{I_{in}}$ at y=25000.5	14.06	15.30	7.80
-10% in $C_{I_{in}}$ at y=18750	14.04	23.60	19.82
-10% in $C_{I_{in}}$ at y=13500	9.27	32.61	40.07

* In the presence of modeling error

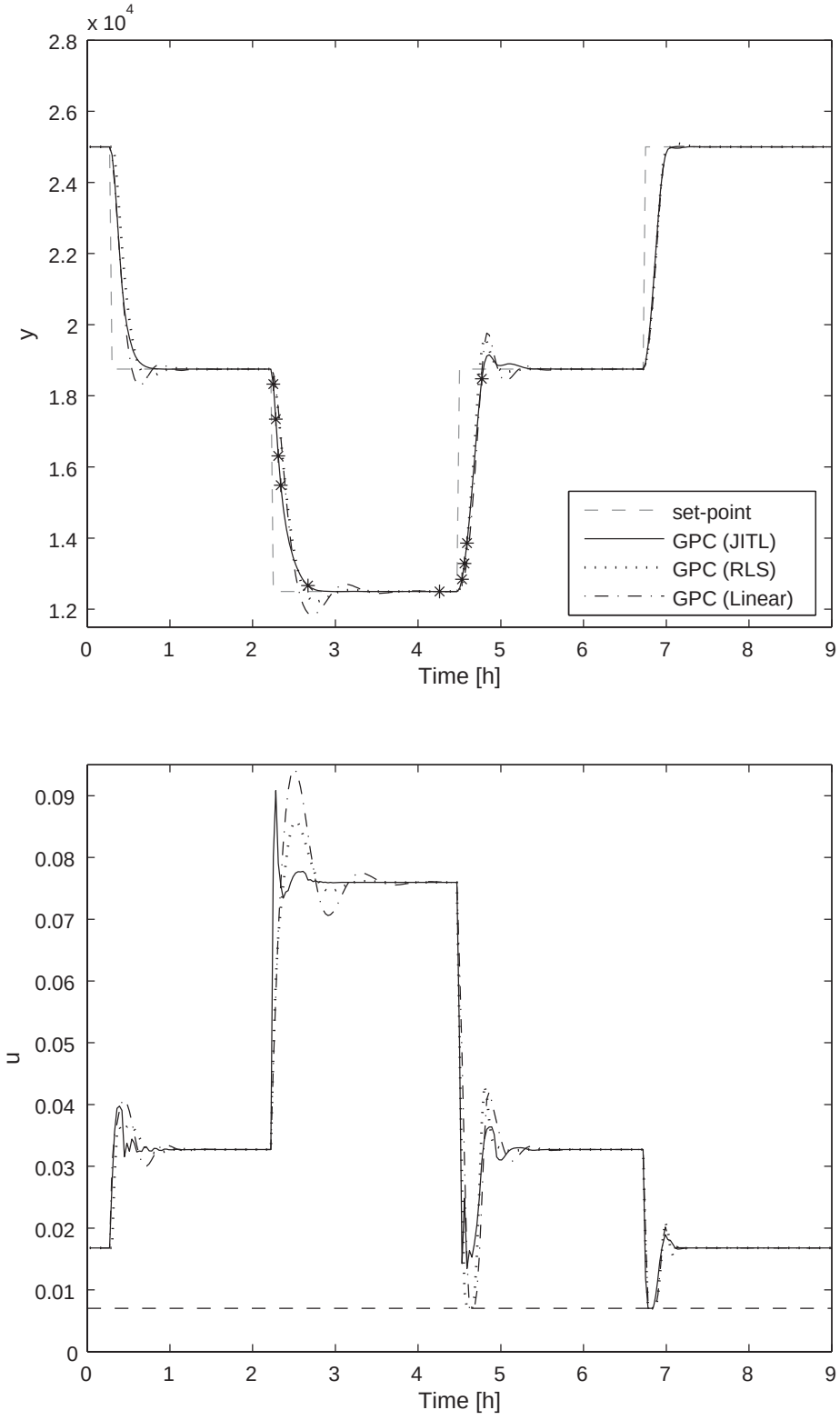

Figure 6.1: Servo responses of three GPC designs ($*$: database update)

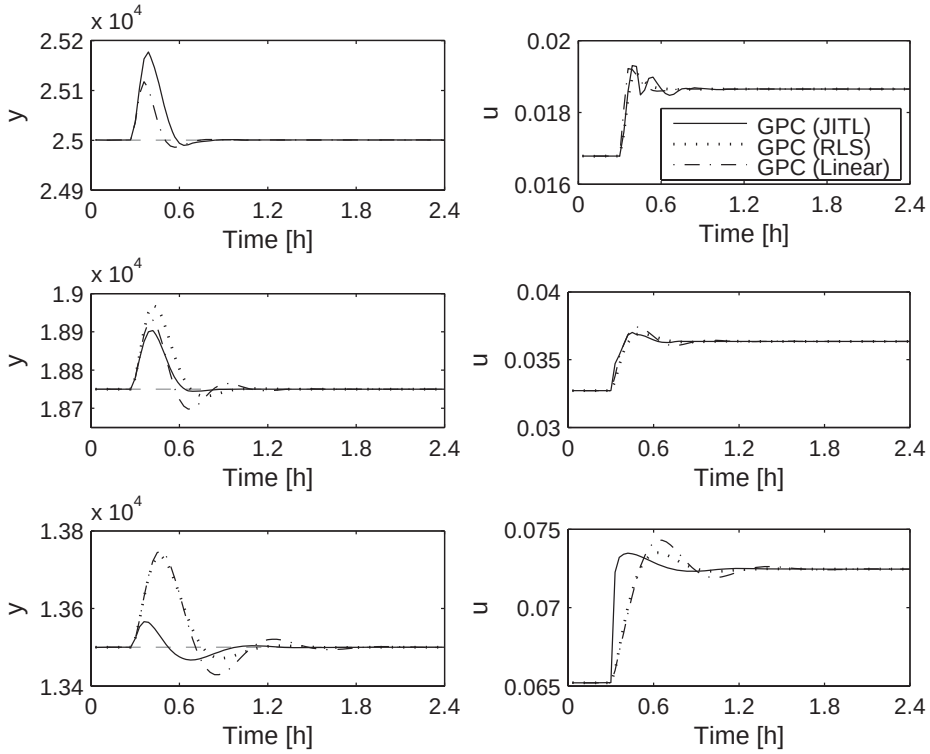

Figure 6.2: Closed-loop responses for -10% step change in $C_{\mathrm{I_{in}}}$

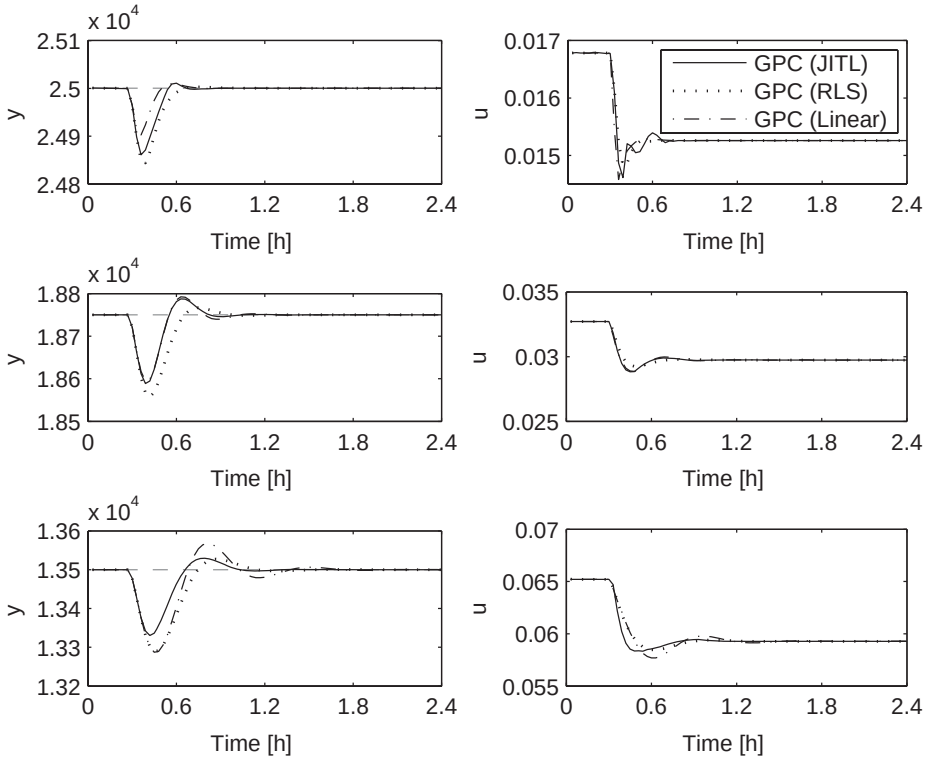

Figure 6.3: Closed-loop responses for +10% step change in $C_{\mathrm{I_{in}}}$

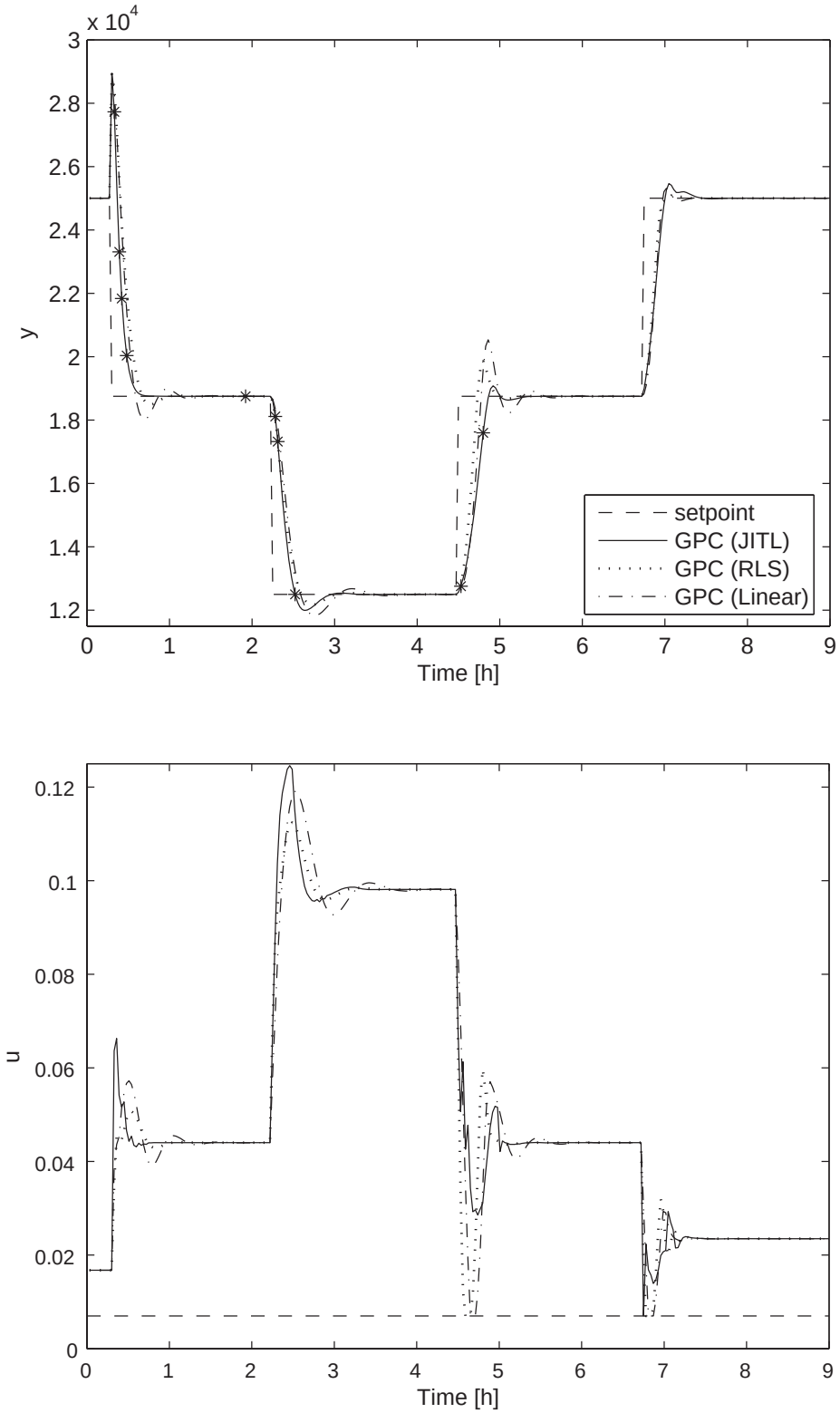

Figure 6.4: Servo responses of three GPC designs in the presence of modeling error

($*$: database update)

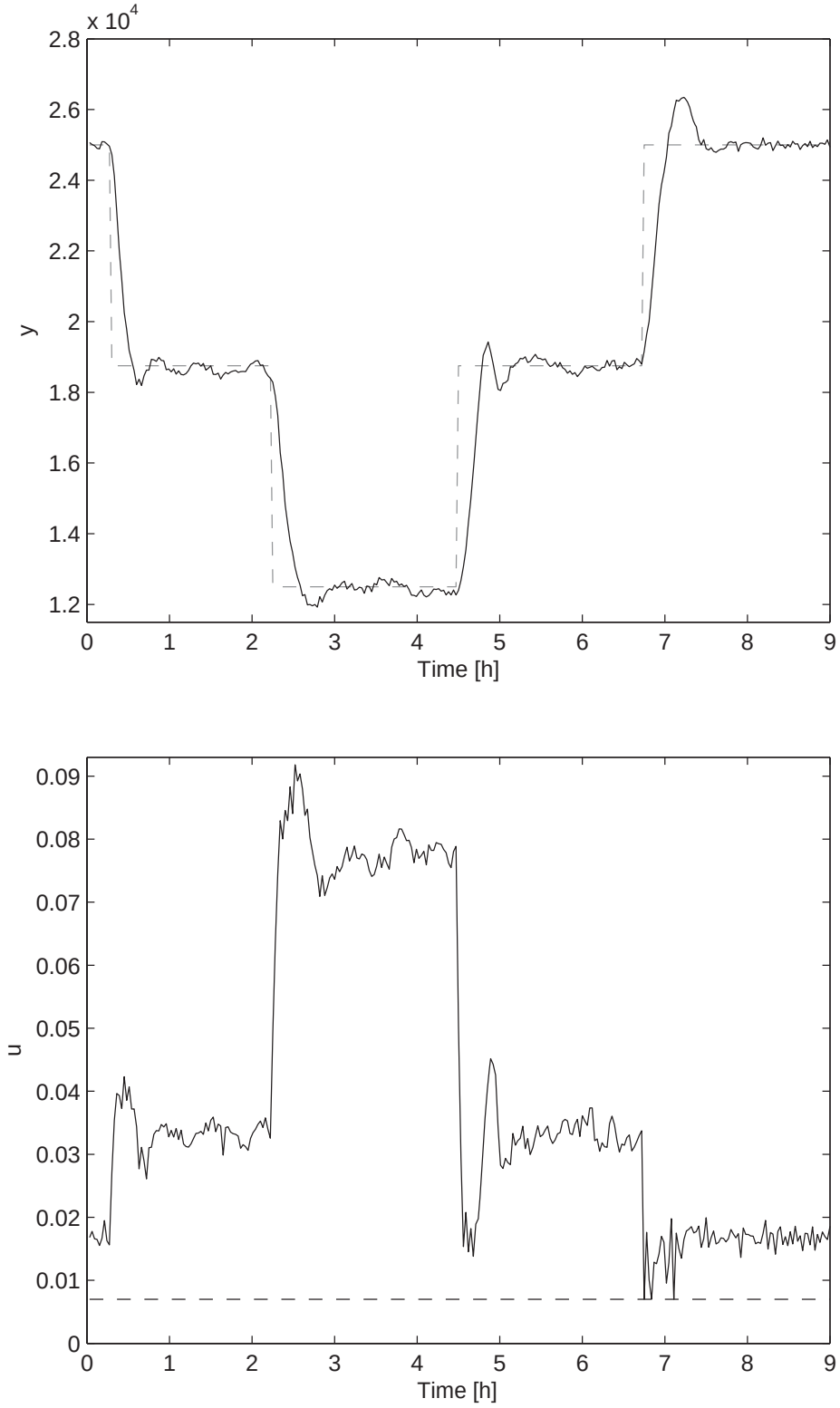

Figure 6.5: Servo response of the JITL-based GPC design in the presence of noise

Example 2 Consider the first-order plus dead-time process studied in Chapters 4 and 5. Three operating regions of this process are given in Table 4.4. In addition, the identical database and parameters for the JITL used in Chapters 4 and 5 are employed. For comparison purpose, an adaptive GPC design based on a first-order local model with parameter adaptation by the RLS identification procedure is designed. The following input constraints are considered in two GPC designs.

$$\begin{cases} u \leq 2.1 \quad \textit{while} \quad y \leq 2.5 \\ u \leq 1.5 \quad \textit{while} \quad y > 2.5 \end{cases} \tag{6.16}$$

Figures 6.6 and 6.7 compare the servo performances of two controllers for the successive set-point changes between 0 and 6. For the proposed GPC design, the prediction horizon is $N_p = 5$, the control horizon $N_u = 1$, and the weighting matrix is $\mathbf{M} = 3\mathbf{I}$, while $N_p = 5$, $N_u = 1$, and $\mathbf{M} = 4\mathbf{I}$ are used in the RLS-based GPC design. It is evident that the proposed GPC design has better overall performance than that achieved by the GPC controller based on the RLS method, especially the large overshoot for the set-point change from 1 to 3 displayed in the latter design. Next, to compare the disturbance rejection capability of two GPC designs, a step disturbance of 0.05 is introduced at the process output and the resulting control performances at different operating conditions are shown in Figures 6.8 and 6.9. Again, the proposed GPC design gives a marked improvement over that achieved by the RLS-based GPC design. Lastly, to evaluate the robustness of two GPC designs, 10% modeling error in the process parameter $\tau_2(y)$ is assumed. As can be seen from Figure 6.10, the proposed GPC design still maintains superior control performance in the aforementioned set-point changes.

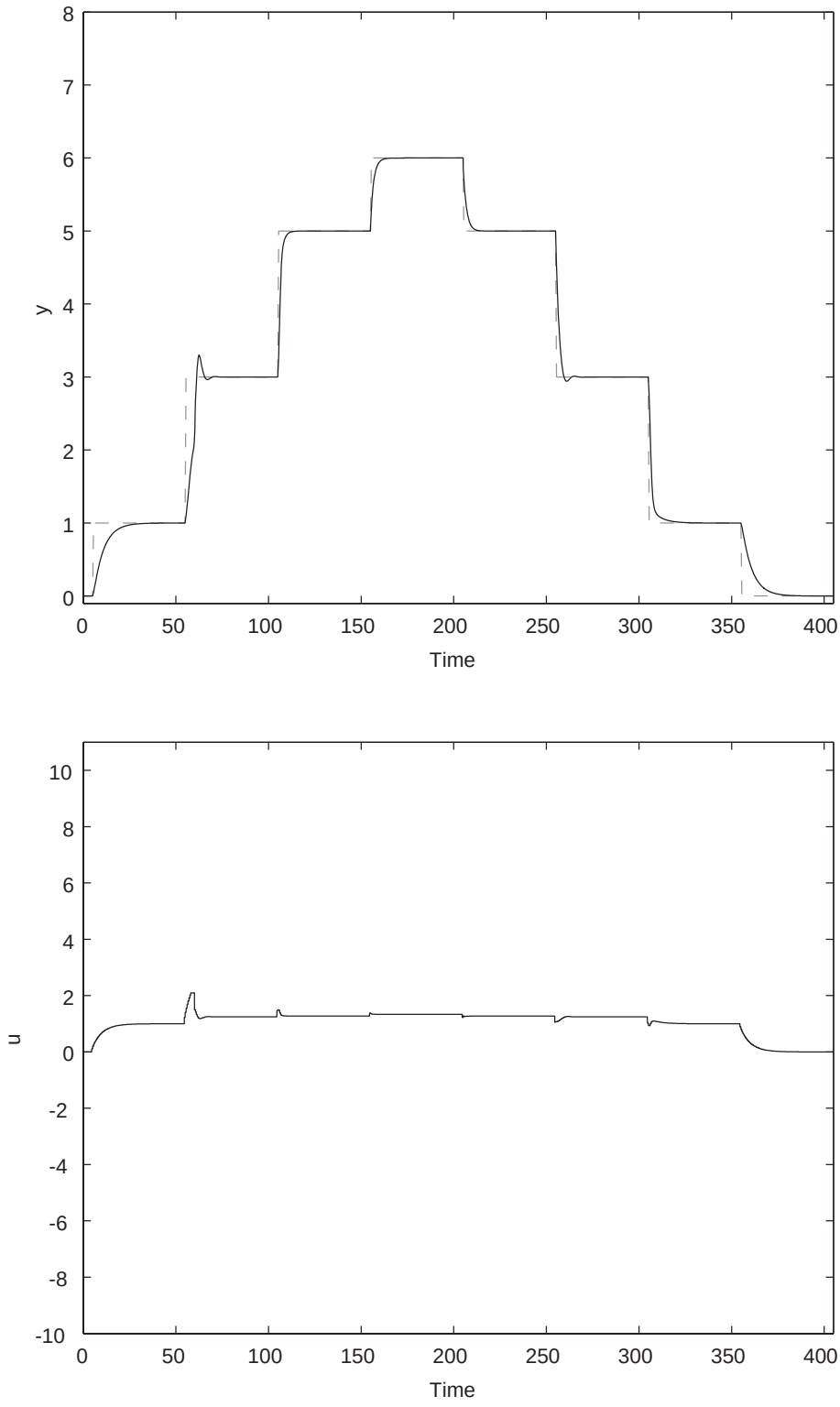

Figure 6.6: Servo response of the JITL-based GPC design

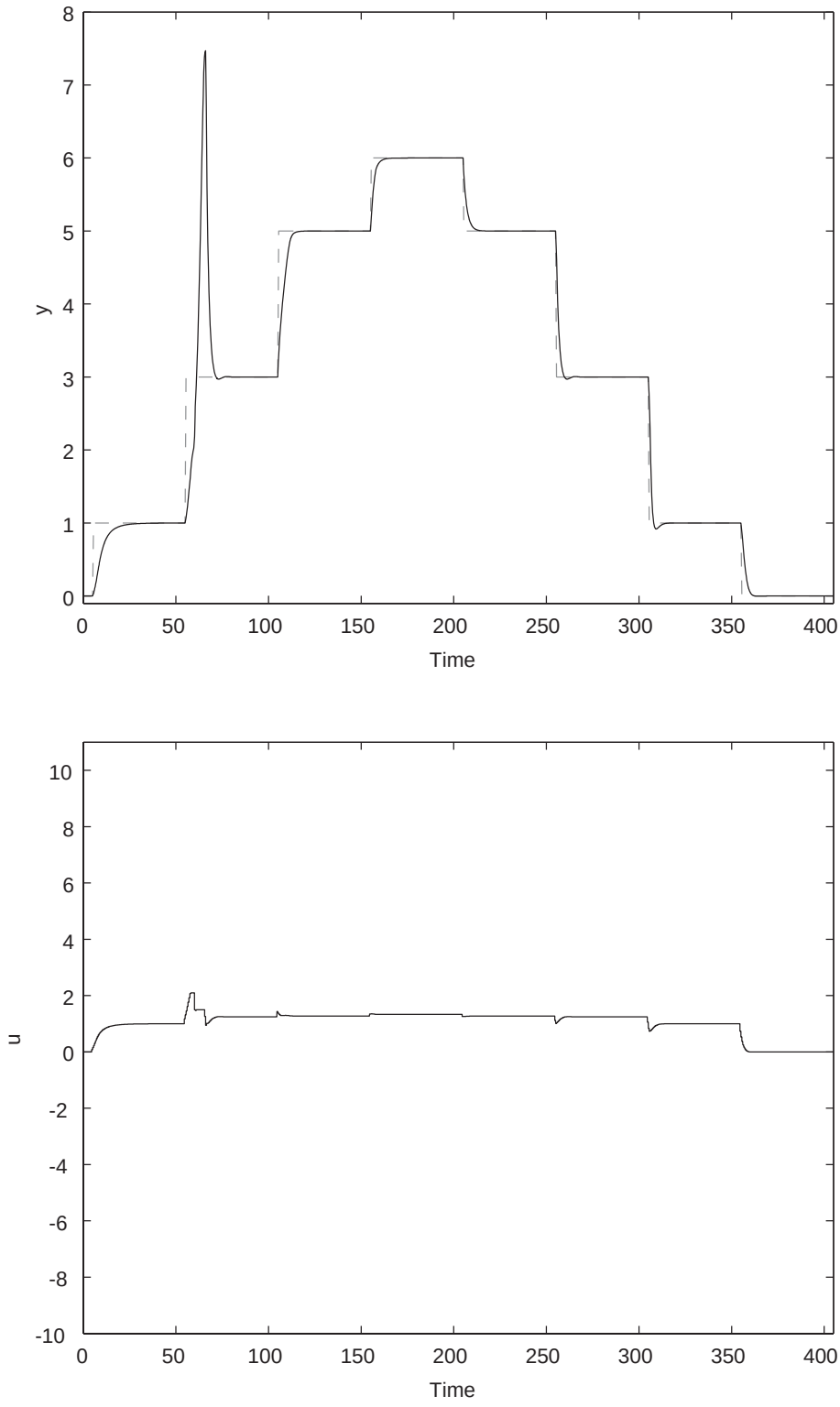

Figure 6.7: Servo response of the RLS-based GPC design

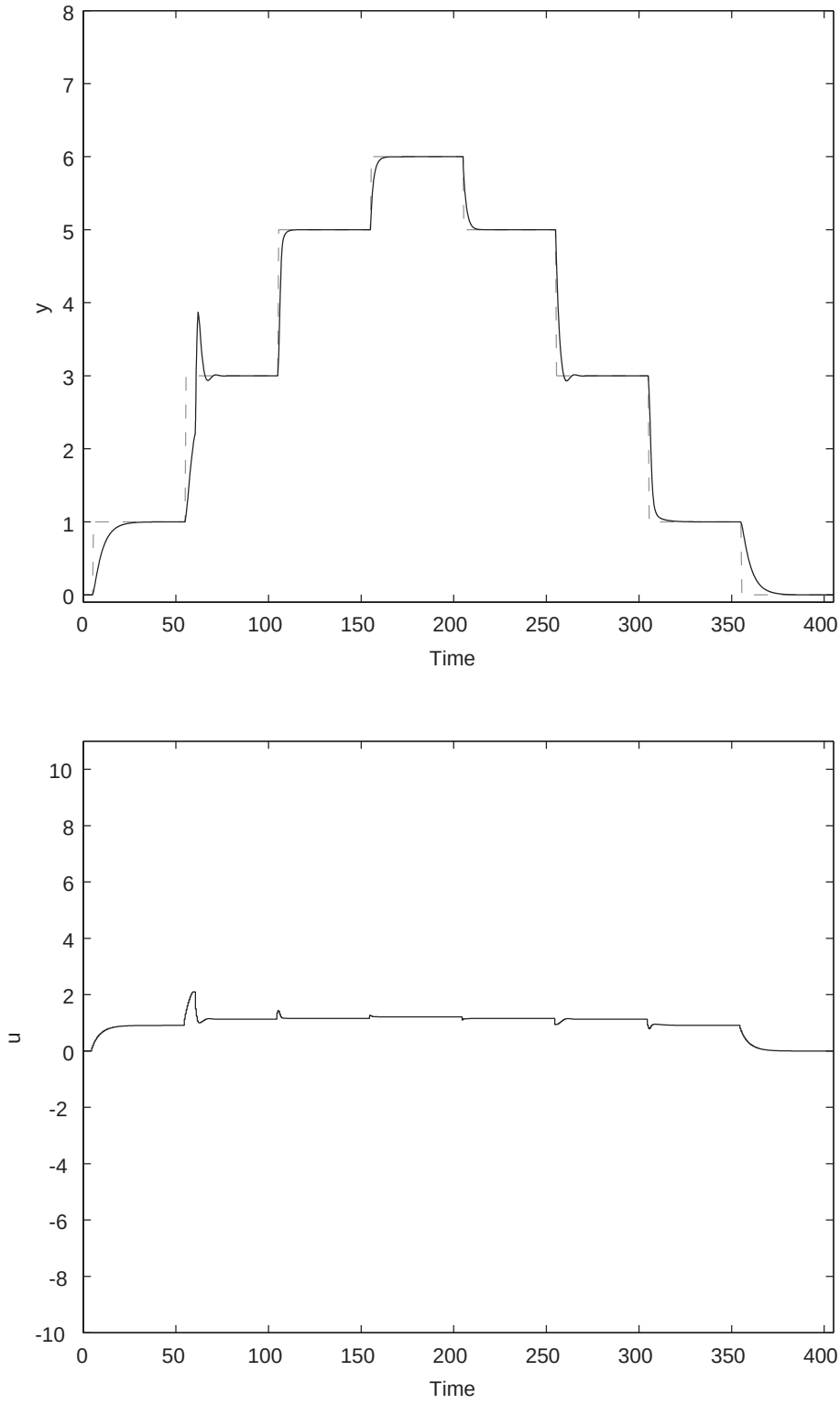

Figure 6.8: Servo response of the JITL-based GPC design under +10% modeling error

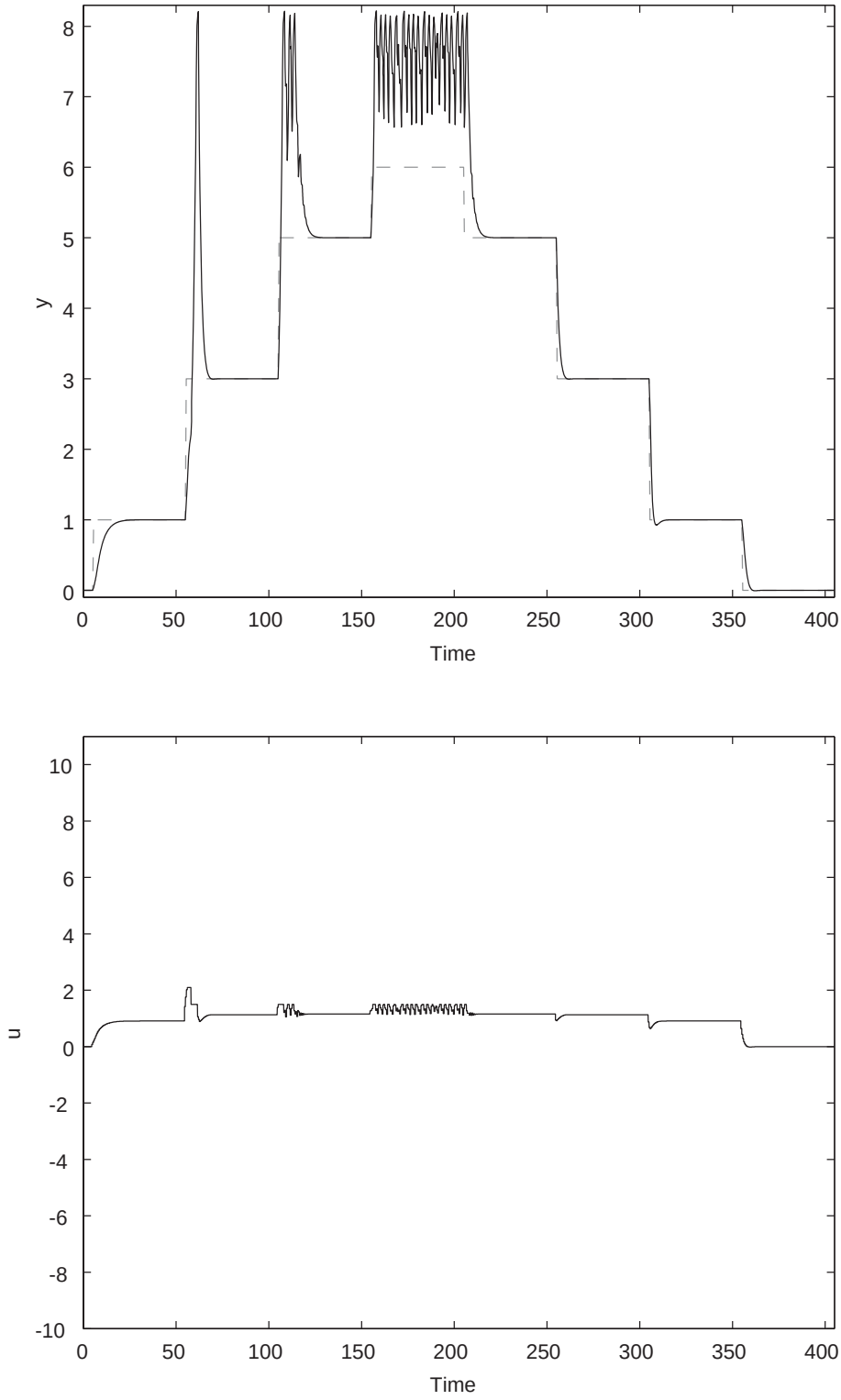

Figure 6.9: Servo response of the RLS-based GPC design under +10% modeling error

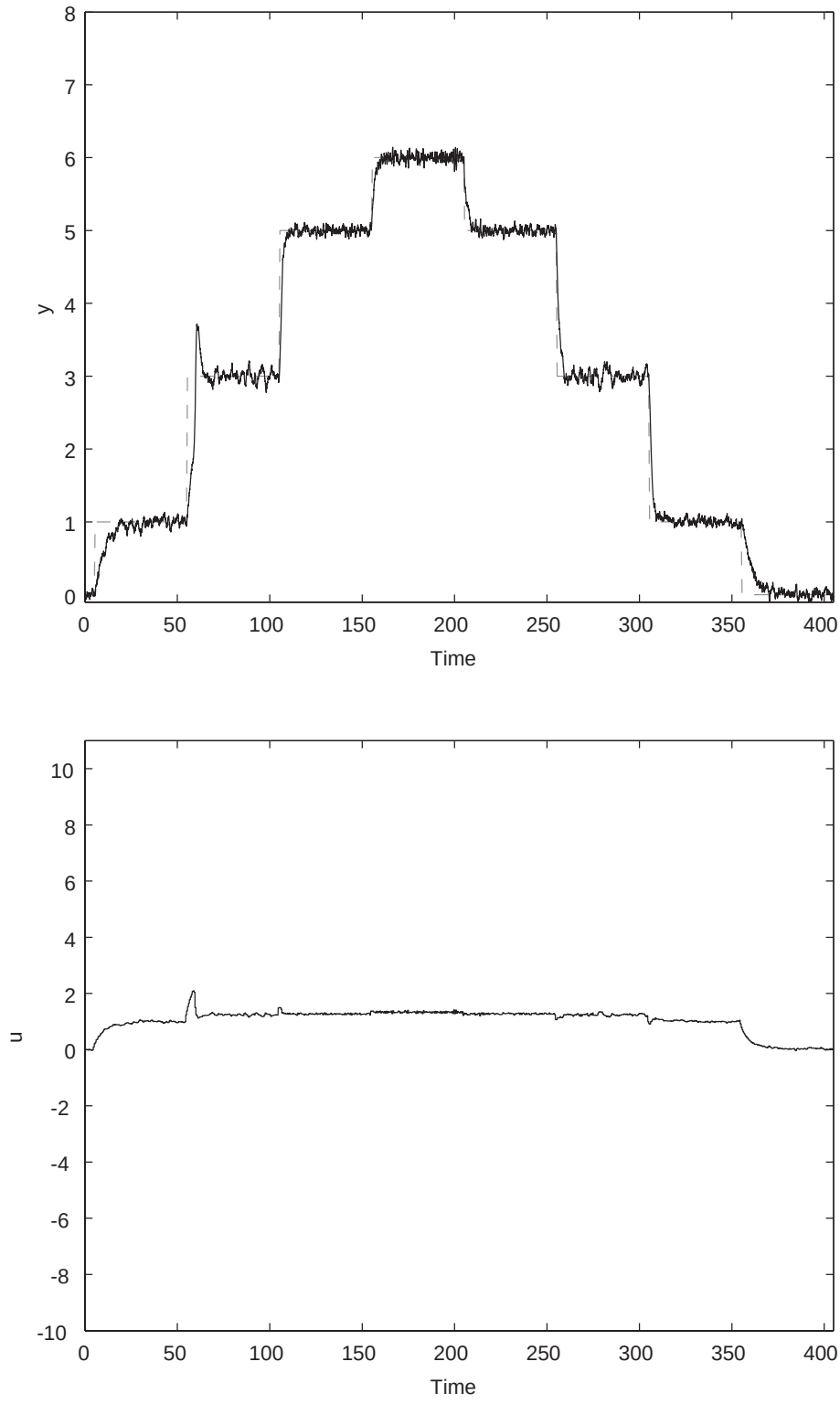

Figure 6.10: Servo response of the JITL-based GPC design in the presense of noise

6.4 Conclusion

A constrained GPC design based on multiple linear models is developed in this chapter. The set of local models obtained by the JITL modeling technique is used for modeling the nonlinear systems and serves as the process model in the GPC design. As a result, the computational burden associated with the conventional nonlinear model predictive control algorithms can be reduced. Simulation results show that the proposed GPC design is able to handle process uncertainties and nonlinearities as well as input constraints in a systematic manner. The comparative studies also reveal that the proposed GPC design exhibits better control performance to its counterparts designed based on a fixed linear model and the multiple linear models utilizing the recursive least squares method.

Chapter 7

Adaptive PID Controller Design Directly From Plant Data - Part I

7.1 Introduction

Model-based techniques have been the predominant controller design methods that have received much research interest in the past several decades. For example, based on the transfer function models like first-order-plus-dead-time model, various PID tuning formulas including ITAE performance index, direct synthesis design method, and Internal Model Control (IMC) design are well established in the literature. Generally, model-based controller design methods involve two-step procedure, where the first step is to identify a process model among the pre-specified model structures that gives reasonably good modeling accuracy, followed by the controller design based on the model thus obtained. However, these model-based design methods suffer the following drawbacks. First, those simplified transfer function models employed in controller design may not carry sufficient information for the process under con-

trol and thus the performance of the resulting controller will become poor if the discrepancy between the process and model is too large. Even when those models have acceptable modeling accuracy, a trial and error procedure is normally required to evaluate which model is best suited in controller design to give the best control performance.

To alleviate the aforementioned problems, the Virtual Reference Feedback Tuning (VRFT) method (Campi et al., 2000, 2002) is recently developed as a direct data-based method that determines the parameters of a controller by using a set of input and output data of a plant without resorting to the identification of a process model. However, this design framework is originally developed for linear systems and thus its application to nonlinear systems is restricted.

In this research, the connection between VRFT and IMC designs is first analyzed. Next, by suitable management of database for VRFT design, an adaptive PID controller design method for nonlinear processes is developed. In the proposed adaptive VRFT design, the off-line database employed in the conventional VRFT design is continuously updated by adding the current process data into the database. Furthermore, PID parameters are determined by the VRFT design at each sampling instant using the relevant dataset selected from the current database based on k-nearest neighborhood criterion. Simulation results are presented to illustrate the proposed design and a comparison with conventional VRFT design is made.

7.2 The VRFT Design Framework

The VRFT method approximately solves a model-reference problem in discrete time as depicted in Figure 7.1, where the reference model $T(z^{-1})$ describes the desired behavior of the closed-loop system consisting of a linear time-invariant process $P(z^{-1})$ and a parameterized controller $C(z^{-1};\theta)$ as shown in Figure 7.2. Let us assume that $P(z^{-1})$ is unknown and only a set of process input and output data, $\{u(k)\}_{k=1\sim N}$ and $\{y(k)\}_{k=1\sim N}$, have been collected from the experiment on the plant and that a reference model $T(z^{-1})$ has been chosen. The design goal is to solve θ, a vector consisting of the controller parameters, such that the feedback control system in Figure 7.1 behaves as closely as possible to the pre-specified reference model $T(z^{-1})$.

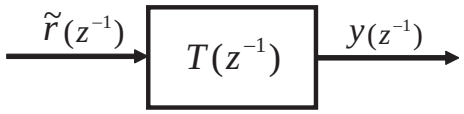

Figure 7.1: Reference model

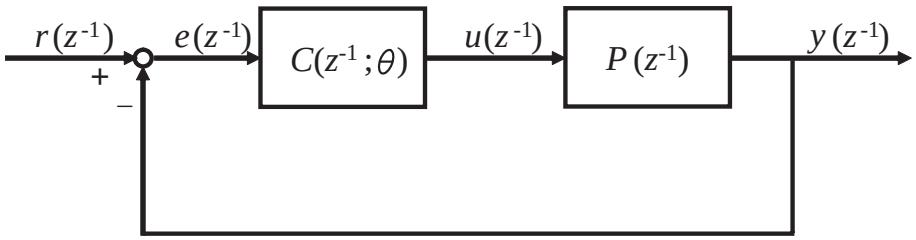

Figure 7.2: Feedback control system

Given the measured output signal $\{y(k)\}_{k=1\sim N}$, the corresponding reference signal $\{\tilde{r}(k)\}_{k=1\sim N}$ in Figure 7.1 is obtained by

$$\tilde{r}(z^{-1}) = T^{-1}(z^{-1})y(z^{-1}) \tag{7.1}$$

where $\tilde{r}(z^{-1})$ and $y(z^{-1})$ are the Z-transforms of discrete time signals $\{\tilde{r}(k)\}_{k=1\sim N}$ and $\{y(k)\}_{k=1\sim N}$, respectively. $\tilde{r}(z^{-1})$ is called 'virtual' reference signal because it does not exist in reality and in fact it was not used in the generation of $y(k)$. However, it plays a pivotal role in the VRFT framework in that the fundamental idea of the VRFT framework is to treat $\{y(k)\}_{k=1\sim N}$ as the desired output of the feedback system when the reference signal is specified by $\{\tilde{r}(k)\}_{k=1\sim N}$. As a consequence, given error signal $e(k) = \tilde{r}(k) - y(k)$, the controller output $\tilde{u}(k)$ is calculated as:

$$\tilde{u}(z^{-1}) = C(z^{-1};\theta)\{\tilde{r}(z^{-1}) - y(z^{-1})\} \tag{7.2}$$

where $\tilde{u}(z^{-1})$ is the Z-transforms of discrete time signal $\{\tilde{u}(k)\}_{k=1\sim N}$.

It is noted that, even though the process dynamics $P(z^{-1})$ is not known, when the process is fed by $u(k)$, i.e. the measured input signal, it generates $y(k)$, i.e. the corresponding measured output signal. Therefore, a good controller generates $u(k)$ when the error signal is given by $e(k)$. The idea is then to search for $C(z^{-1};\theta)$ whose output $\tilde{u}(k)$ matches $u(k)$ as closely as possible. Hence, the controller design task reduces to the following minimization problem:

$$J(\theta) = \min_{\theta} \frac{1}{N} \sum_{k=1}^{N} \{u(k) - \tilde{u}(k)\}^2 \tag{7.3}$$

If the controller is given by $C(z^{-1};\theta) = \rho^{\mathrm{T}}(z^{-1})\theta$ where $\rho(z^{-1})$ is a vector of discrete-time transfer function, it can be seen that Eq. (7.3) is quadratic in θ. Consequently, the controller parameter θ^* which minimizes Eq. (7.3) can be explicitly obtained by the classical least-square technique. As a result, the VRFT design framework effectively recasts the problem of designing a model-reference feedback controller into a standard system-identification problem. More detailed discussions on the VRFT can be found in Campi et al. (2000, 2002).

7.2.1 PID controller design by VRFT method

To illustrate the VRFT design in more detail, its application to PID design is discussed in this subsection. Consider a PID controller given by:

$$\begin{aligned} u(k) &= u(k-1) + K_P\{e(k) - e(k-1)\} + K_I e(k) \\ &\quad + K_D\{e(k) - 2e(k-1) + e(k-2)\} \end{aligned} \tag{7.4}$$

where $u(k)$ is the manipulated value at the k-th sampling instant, $e(k)$ is the error between process output and its set-point at the k-th sampling instant, and K_P, K_I and K_D are PID parameters.

In VRFT design framework, the reference model $T(z^{-1})$ is specified by the following first-order equation:

$$T(z^{-1}) = \frac{(1-A)z^{-1}}{1 - Az^{-1}} \tag{7.5}$$

where A is a tuning parameter related to the speed of response.

To design a PID controller by the VRFT method, the virtual input $\tilde{u}(z^{-1})$ is calculated by Eqs. (7.1), (7.2), and (7.5) to obtain

$$\tilde{u}(z^{-1}) = \left[K_P + \frac{K_I}{1 - z^{-1}} + K_D\left(1 - z^{-1}\right)\right] \frac{1 - z^{-1}}{(1-A)z^{-1}} y(z^{-1}) \tag{7.6}$$

Equation (7.6) can be rewritten as

$$\tilde{u}(k) = \Psi(k)\mathbf{K} \tag{7.7}$$

where

$$\Psi(k) = \begin{bmatrix} \psi_P(k) & \psi_I(k) & \psi_D(k) \end{bmatrix} \tag{7.8}$$

$$\mathbf{K} = \begin{bmatrix} K_P & K_I & K_D \end{bmatrix}^{\mathrm{T}} \tag{7.9}$$

$$\psi_P(k) = \frac{1}{1-A}\{y(k+1) - y(k)\} \tag{7.10}$$

$$\psi_I(k) = \frac{1}{1-A}y(k+1) \tag{7.11}$$

$$\psi_D(k) = \frac{1}{1-A}\{y(k+1) - 2y(k) + y(k-1)\} \tag{7.12}$$

Equation (7.3) is then expressed by

$$\begin{aligned} J(\mathbf{K}) &= \min_{\mathbf{K}} \frac{1}{N}\sum_{k=1}^{N}\{u(k) - \Psi(k)\mathbf{K}\}^2 \\ &= \min_{\mathbf{K}} \frac{1}{N}\|\mathbf{u} - \Psi\mathbf{K}\|^2 \end{aligned} \tag{7.13}$$

subject to the sign constraints of PID controller parameters:

$$K_P, K_I, K_D \geq 0$$

or

$$K_P, K_I, K_D < 0 \tag{7.14}$$

where

$$\Psi = \begin{bmatrix} \psi_P(1) & \psi_I(1) & \psi_D(1) \\ \psi_P(2) & \psi_I(2) & \psi_D(2) \\ \vdots & \vdots & \vdots \\ \psi_P(N) & \psi_I(N) & \psi_D(N) \end{bmatrix} \tag{7.15}$$

$$\mathbf{u} = \begin{bmatrix} u(1) & \cdots & u(N) \end{bmatrix}^{\mathrm{T}} \tag{7.16}$$

Consequently, PID parameters are obtained by solving the constrained least square problem as stated above. It is evident that this solution not only depends on the database for VRFT design but also the design parameter A in the reference model. Typically, a smaller value of A would give more aggressive PID design and vice versa.

7.3 Connection Between VRFT and IMC Designs

In this section, the equivalence relationship between VRFT and IMC designs is established. First, in the IMC design, under the assumption of perfect model, i.e. $P(z^{-1}) = M(z^{-1})$ in Figure 7.3, the process model $M(z^{-1})$ can be factorized as

$$M(z^{-1}) \;=\; M_+(z^{-1})M_-(z^{-1}) \tag{7.17}$$

where $M_+(z^{-1})$ consists of time delay and zeros outside the unit circle.

Using a first-order filter, IMC controller Q is given by

$$Q(z^{-1}) = f(z^{-1})M_-^{-1}(z^{-1}) = \frac{1-A}{1-Az^{-1}}M_-^{-1}(z^{-1}) \tag{7.18}$$

Therefore, the resulting closed-loop transfer function is described by

$$\frac{y(z^{-1})}{r(z^{-1})} = \frac{1-A}{1-Az^{-1}}M_+(z^{-1}) \tag{7.19}$$

On the other hand, if the objective function $J(\theta)$ in VRFT design can be made sufficiently small and the database for VRFT design are rich enough to represent the process dynamics, the following equation can be obtained from Eqs. (7.2), (7.18) and (7.19) and the relation $P(z^{-1}) = \frac{y(z^{-1})}{u(z^{-1})}$:

$$C(z^{-1};\theta)\{(Q(z^{-1})P(z^{-1}))^{-1} - 1\}P(z^{-1}) = 1 \tag{7.20}$$

The feedback controller $C(z^{-1};\theta)$ is then obtained by

$$C(z^{-1};\theta) = \frac{Q(z^{-1})}{1 - Q(z^{-1})P(z^{-1})} \tag{7.21}$$

which is essentially identical to the feedback controller designed based on the IMC design method.

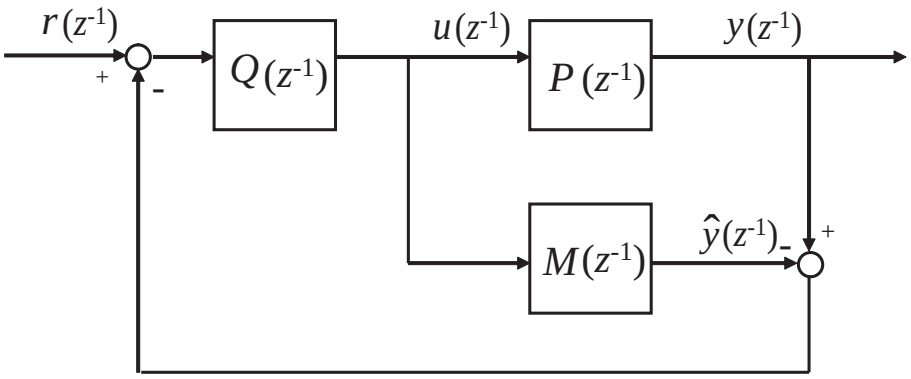

Figure 7.3: IMC control system

Example 1 Consider the following first-order linear system given by Seborg et al. (1989):

$$y(k) = 0.8187y(k-1) + 0.1813u(k-1) \tag{7.22}$$

To illustrate the equivalence between VRFT and IMC designs, without the loss of generality, the IMC filter time constant in the IMC design and the tuning parameter in VRFT design are chosen as $A = 0.3$. In the VRFT design, the number of process data considered is $N = 150$ and a PI controller is designed correspondingly. The servo performances of these two controllers are compared in Figure 7.4. As can be seen, the performances of these two controllers are almost indistinguishable, which supports the analysis given in this section. Next, to further investigate the effect of number of process data utilized in VRFT design on the resulting controller performance in the aforementioned set-point changes, Table 7.1 summarizes the average difference of tracking errors between IMC and VRFT designs obtained for various values of N and A. It is clear that VRFT design resembles closely to IMC design with minimum process data $N = 20$ for this example.

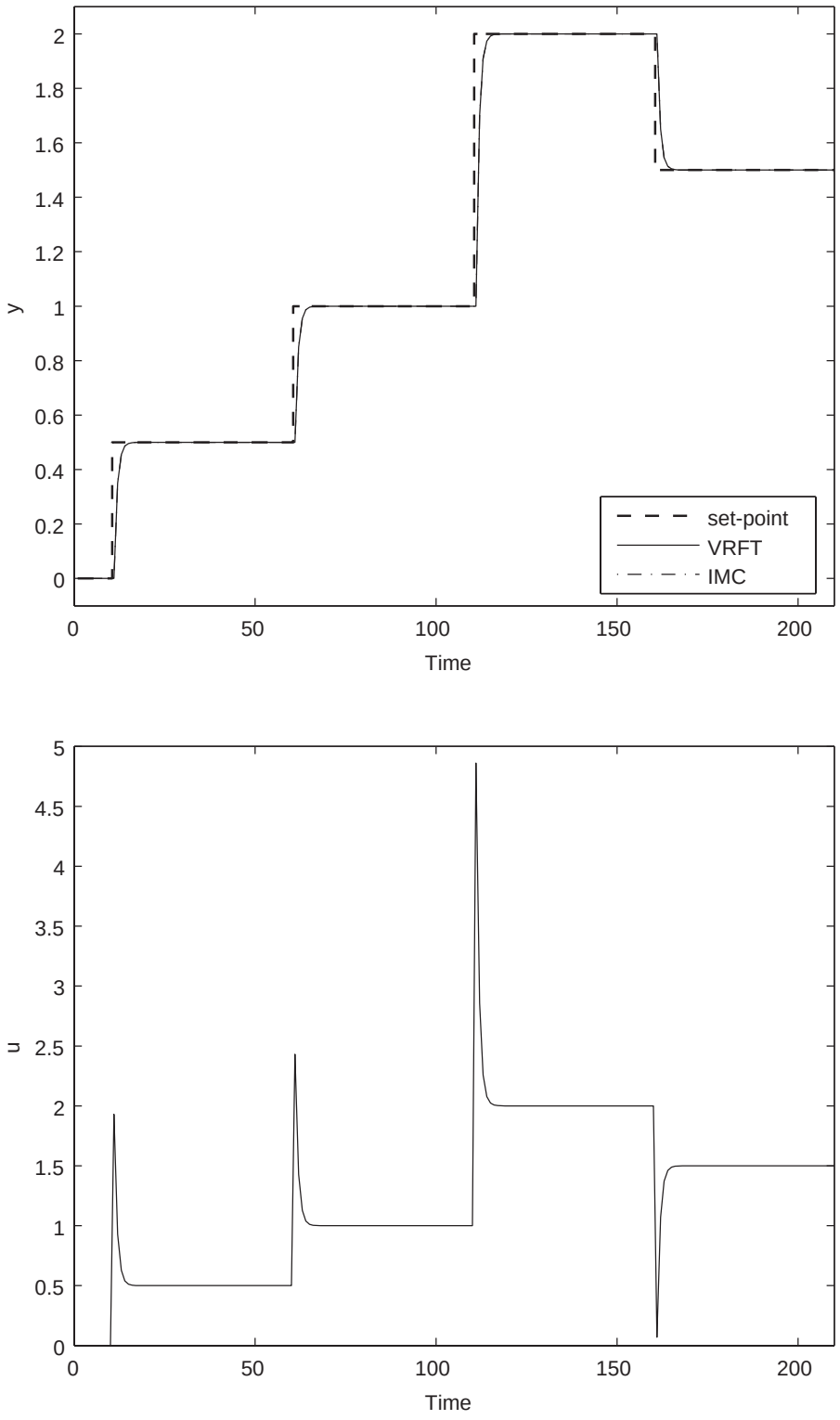

Figure 7.4: Comparison of servo performances between VRFT and IMC designs

Table 7.1: The difference of the tracking error between VRFT and IMC designs

N	20	60	100	140	180
MAE($\times 10^{-16}$)	0.982	1.011	1.032	1.033	1.040

7.4 Adaptive VRFT Design of PID Controller

In the conventional VRFT design, the database collected from an off-line open-loop experiment is utilized and as a result the resulting controller is expected to perform well in the vicinity of operating space close to the operating condition where this dataset is generated. To extend the VRFT design to nonlinear systems, one possible approach is to augment the original off-line database by adding the current process data at each sampling instant so that the expanded database can cover new operating space where its dynamics is not available in the construction of original database. This expanded database is subsequently used to obtain PID parameters by VRFT design at each sampling instant. In doing so, the relevant data in the expanded database that corresponds to the current process condition is first determined by using the k-nearest neighborhood criterion based on the following distance measure:

$$d_i = \|\bar{\mathbf{x}}(k-1) - \bar{\mathbf{x}}_i\| \tag{7.23}$$

where $\|\cdot\|$ denotes the Euclidean norm, $\bar{\mathbf{x}}_i = [y(i)\ u(i)]^{\mathrm{T}}$ is a pair of input and output data in the present dataset, and $\bar{\mathbf{x}}(k-1)$ is a vector with similar definition for the input and output data at the $(k-1)$-th sampling instant.

By using Eq. (7.23), those $\bar{\mathbf{x}}_i$ corresponding to the k smallest d_i are selected as the relevant data in the current database, by which the constrained least squares problem discussed in the subsection 7.2.1 is solved to calculate PID parameters for

the current sampling instant. This design procedure repeats at the next sampling instant when the database for VRFT design is further updated by the corresponding process data.

The following gives an outline of the computational algorithm for the proposed adaptive VRFT (AVRFT) design of PID controller:

Step 1 Input ($u(k)$) and output ($y(k)$) identification data which characterize the dynamics of nonlinear system are assumed to be available and the off-line database for VRFT design is constructed as $(\bar{\mathbf{x}}_i)_{i=1\sim N}$;

Step 2 The design parameter in reference model, A, and the number of nearest neighborhood are specified;

Step 3 At each sampling instant, based on the current database for VRFT design, the relevant data is selected according to Eq. (7.23), by which PID parameters are computed by solving the optimization problem, Eq. (7.13), subject to the constraint, Eq. (7.14), and the manipulated variable $u(k)$ is obtained by Eq. (7.4);

Step 4 The database for VRFT design is augmented by adding the current process data $y(k)$ and $u(k)$;

Step 5 Set $k = k + 1$ and go to Step 3.

In what follows, the performance of the AVRFT design is evaluated by using two literature examples provided in Chapter 4.

Example 2 Consider the polymerization reactor example studied in Chapters 4 to 6 with model parameters and steady-state operation condition given in Tables 4.1

and 4.2. To design PID controller by the proposed AVRFT and VRFT methods, the identical off-line process data in Chapter 4 are employed. Furthermore, the tuning parameters for the AVRFT are specified by $A = 0.78$ and $k = 350$, while $A = 0.72$ is chosen for the VRFT design.

To compare the performances of two designs, successive set-point changes between 25000.5 kg/kmol and 12500 kg/kmol are conducted. As can be seen from Figure 7.5, the proposed AVRFT design has better performance than that achieved by the conventional VRFT design, resulting in the reduction of Mean Absolute Error (MAE) by 7.1% as compared with the conventional VRFT design. Figure 7.6 shows the updating of PID parameters by the AVRFT design in the aforementioned closed-loop responses.

Figures 7.7 and 7.8 compare the disturbance rejection capability of two controllers with respect to unmeasured $\pm 10\%$ step disturbances in the inlet initiator concentration $C_{\mathrm{I_{in}}}$. Again, the proposed design has better control performance than that obtained by the VRFT design. Table 7.2 summarizes the performance improvement achieved by the AVRFT design as measured by the MAEs. Next, to evaluate the robustness of the proposed control strategy, it is assumed that there exist 10% modeling error in the kinetic parameter k_{I} and 20% error in the coefficients of the D_1 and M_{m}. As can be seen from Figure 7.9, the proposed design outperforms the VRFT design, as also evidenced by the reduction of MAE by 19.7% for the aforementioned set-point changes. Lastly, to investigate 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 VRFT design also contains the corrupted process data. As shown in Figure 7.10, the proposed design can yield

reasonably good control performance in the presence of process noise.

Table 7.2: Control performance comparison of two VRFT designs

	Tracking error (MAE)		% Decrease
	AVRFT	VRFT	in MAE
Servo Response	794.07	854.50	7.1
Servo Response*	883.49	1100.06	19.7
+10% in $C_{I_{in}}$ at y=25000.5	41.72	42.12	0.9
+10% in $C_{I_{in}}$ at y=18750	57.29	82.14	30.3
+10% in $C_{I_{in}}$ at y=13500	63.92	127.50	49.9
-10% in $C_{I_{in}}$ at y=25000.5	49.11	51.53	4.7
-10% in $C_{I_{in}}$ at y=18750	62.83	100.38	37.4
-10% in $C_{I_{in}}$ at y=13500	75.95	163.52	53.6

* In the presence of modeling error

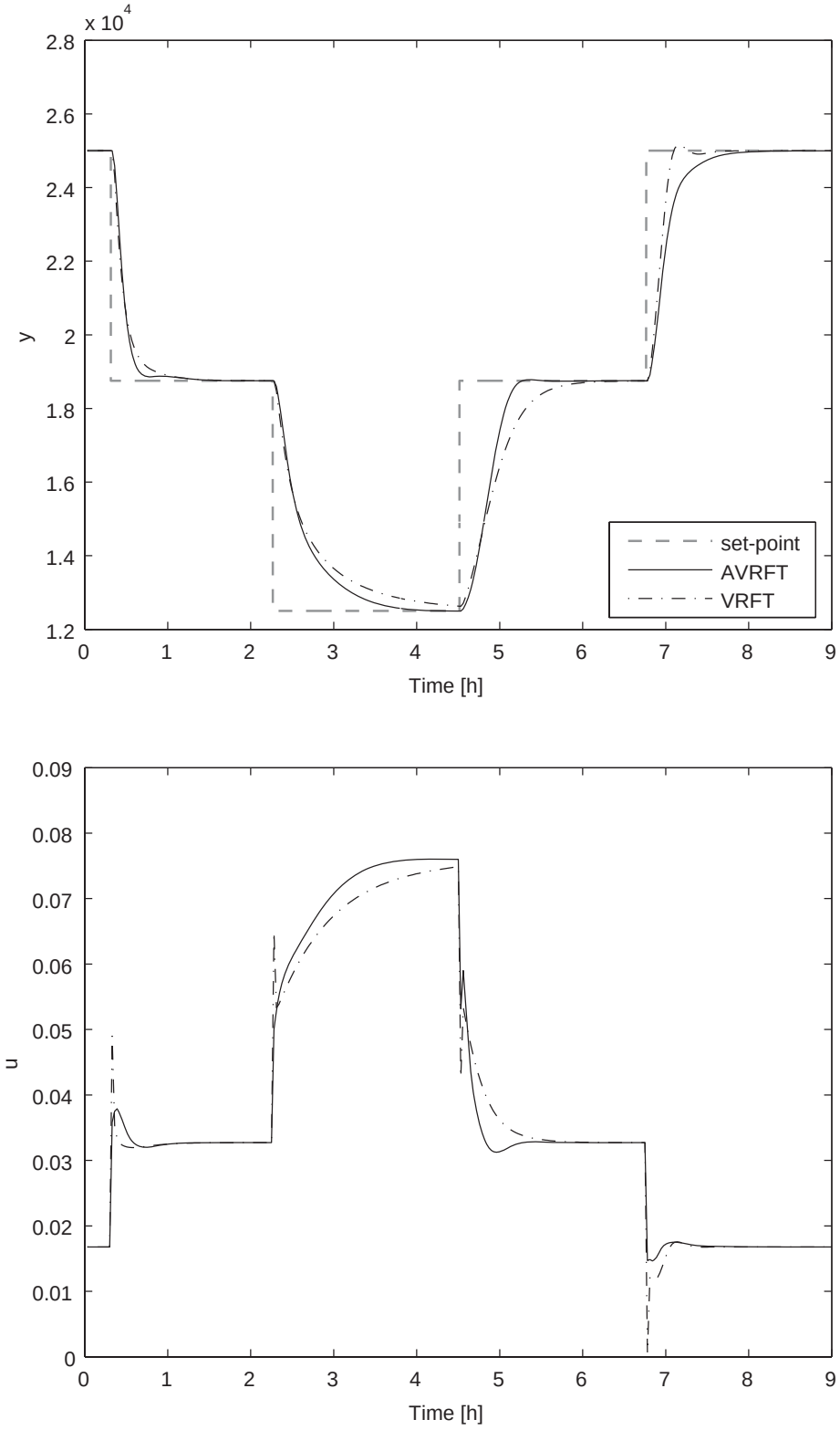

Figure 7.5: Servo responses of two VRFT designs

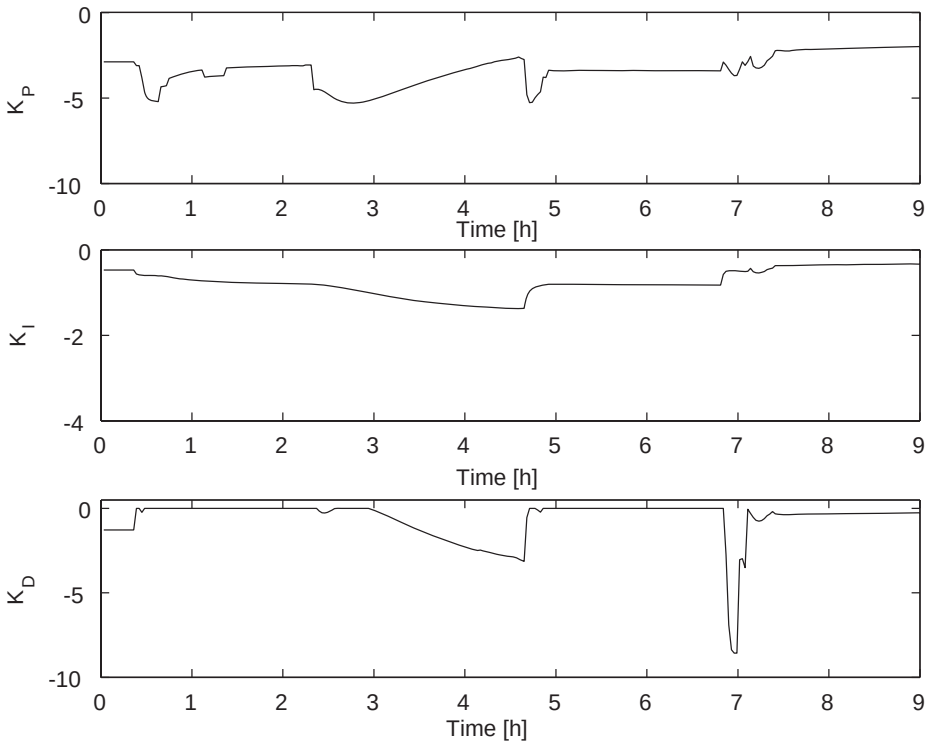

Figure 7.6: Updating of the PID parameters by the AVRFT design

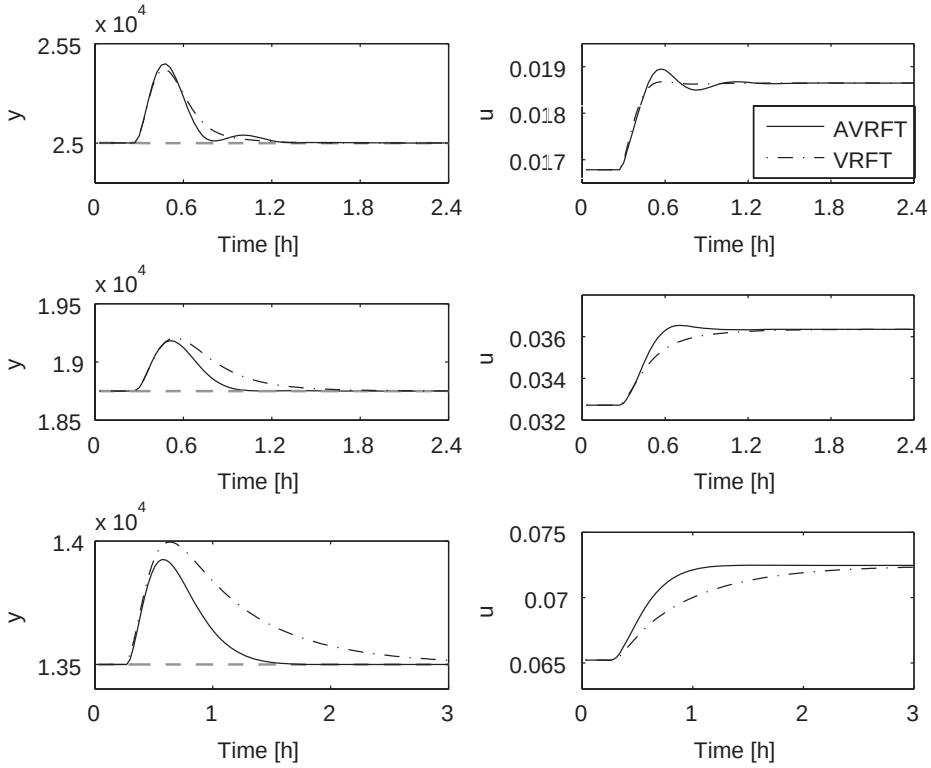

Figure 7.7: Closed-loop responses of two VRFT designs for -10% step change in $C_{\mathrm{I_{in}}}$

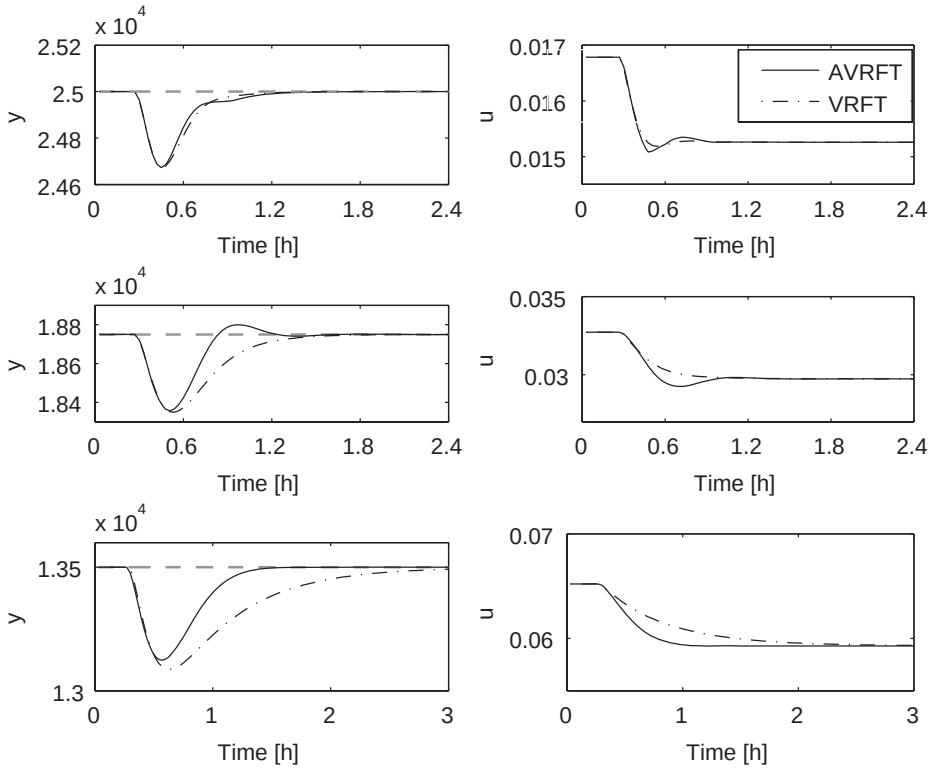

Figure 7.8: Closed-loop responses of two VRFT designs for +10% step change in $C_{\mathrm{I_{in}}}$

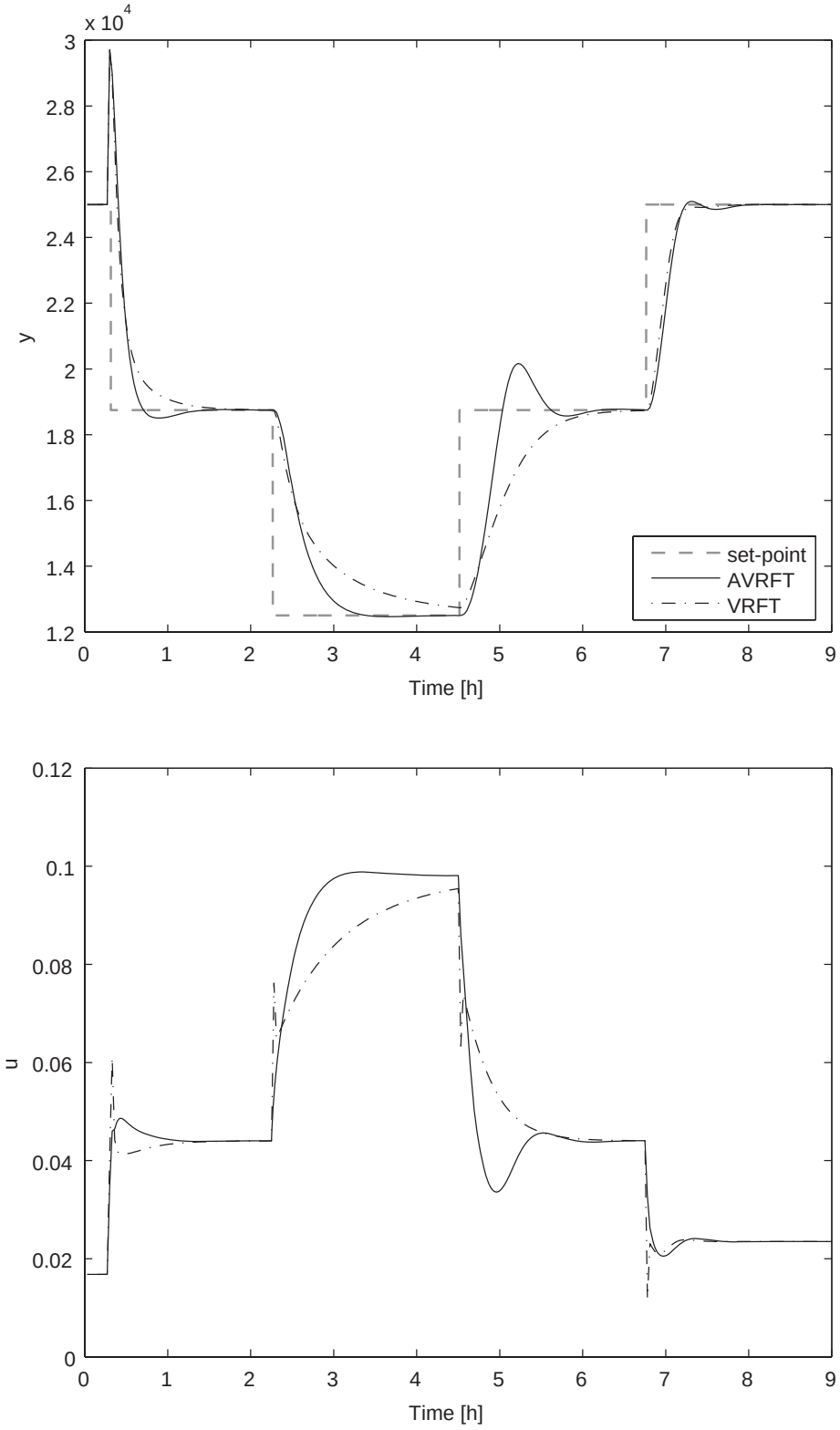

Figure 7.9: Servo responses of two VRFT designs in the presence of modeling error

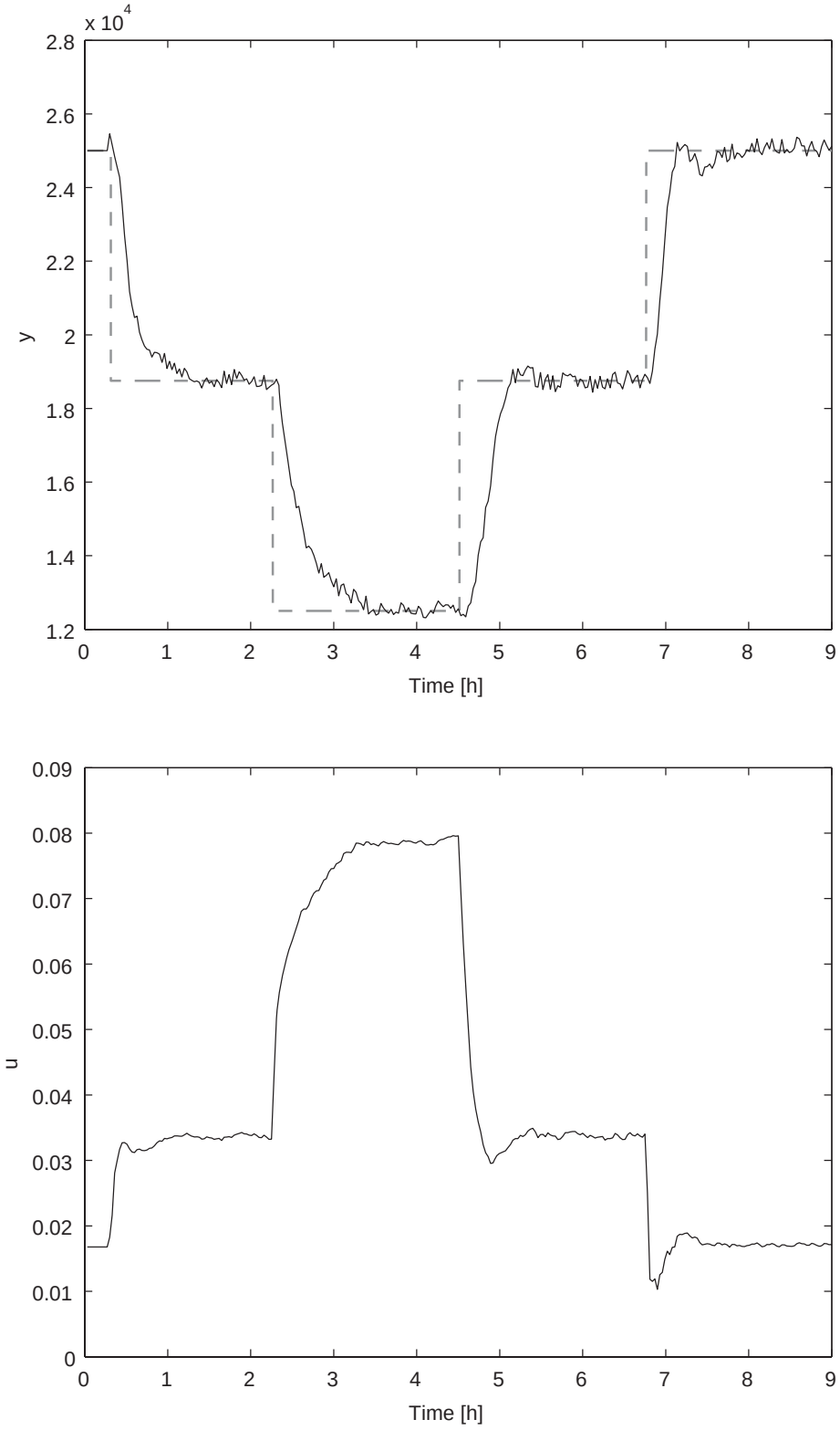

Figure 7.10: Servo response of the adaptive VRFT design in the presence of noise

Example 3 Consider the first-order plus dead-time process studied in Chapters 4 to 6. Two PID controllers are designed based on the VRFT and AVRFT methods. In the proposed AVRFT design, the tuning parameters are specified by $A = 0.88$ and $k = 630$, whereas $A = 0.89$ is selected for the VRFT design.

Figures 7.11 and 7.12 compare the servo performances of two PID controllers for the successive set-point changes between 0 and 6. It can be seen that the proposed AVRFT design has better performance than that achieved by the conventional VRFT design, resulting in the reduction of MAE by 6.6% as compared with the VRFT design. To evaluate the robustness of the proposed control strategy, 10% modeling error in the process parameter $\tau_2(y)$ is assumed and the resulting servo responses of two controllers are compared in Figures 7.13 and 7.14. It is evident that the proposed AVRFT design still maintains better control performance over the VRFT design by achieving 8.4% reduction of MAE in the aforementioned set-point changes. Lastly, to study the effect of process noise on the proposed design, both process input and output are corrupted by 3% Gaussian white noise. It can be seen from Figure 7.15 that the proposed design can yield reasonably good control performance in the presence of process noise.

7.5 Conclusion

In this chapter, the connection between the VRFT and IMC design methods is established and an extension of VRFT method to adaptive PID controller design is proposed. Specifically, under very mild assumptions, the equivalence between the VRFT and IMC designs is shown. Furthermore, by incorporating the current pro-

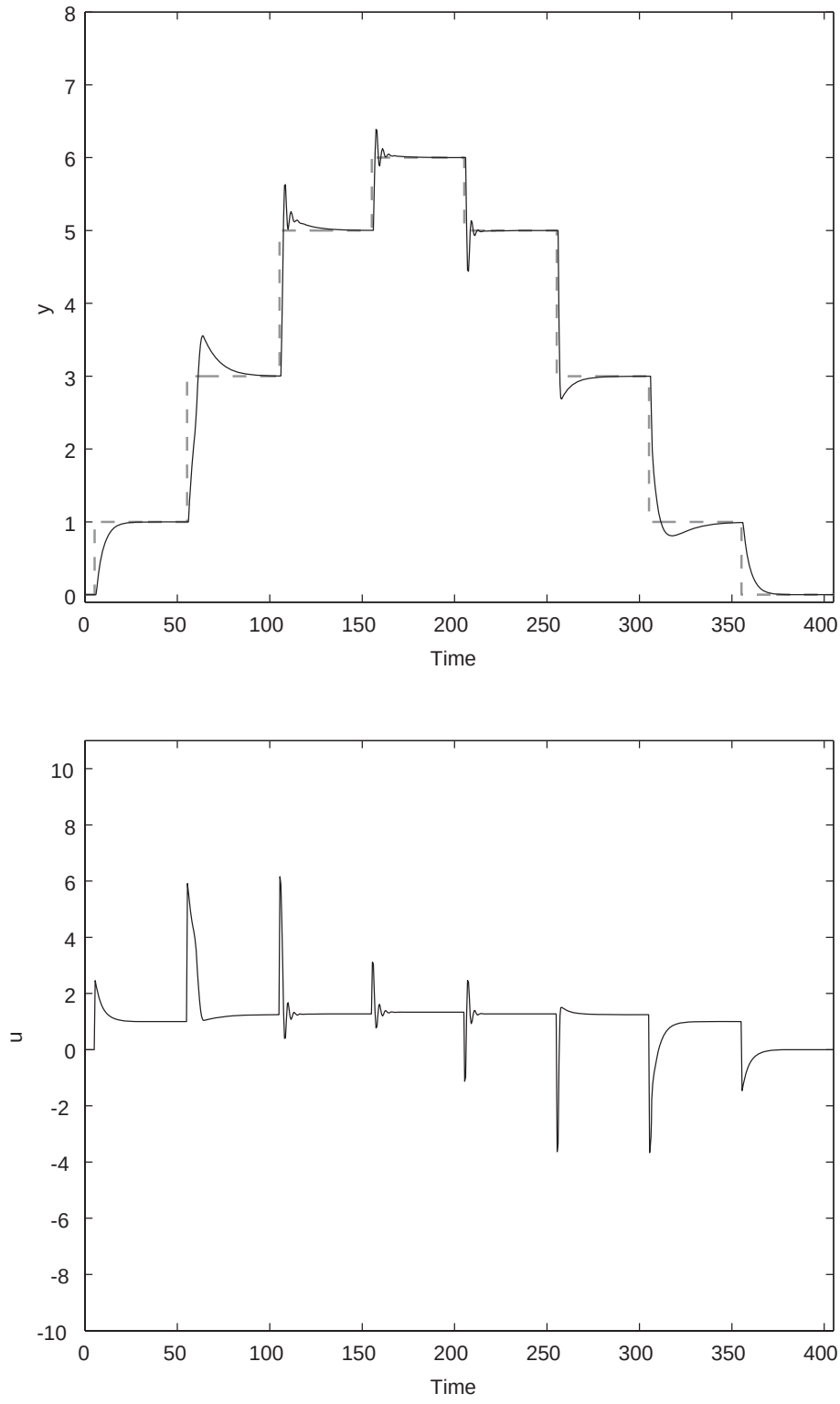

Figure 7.11: Servo response of the adaptive VRFT design

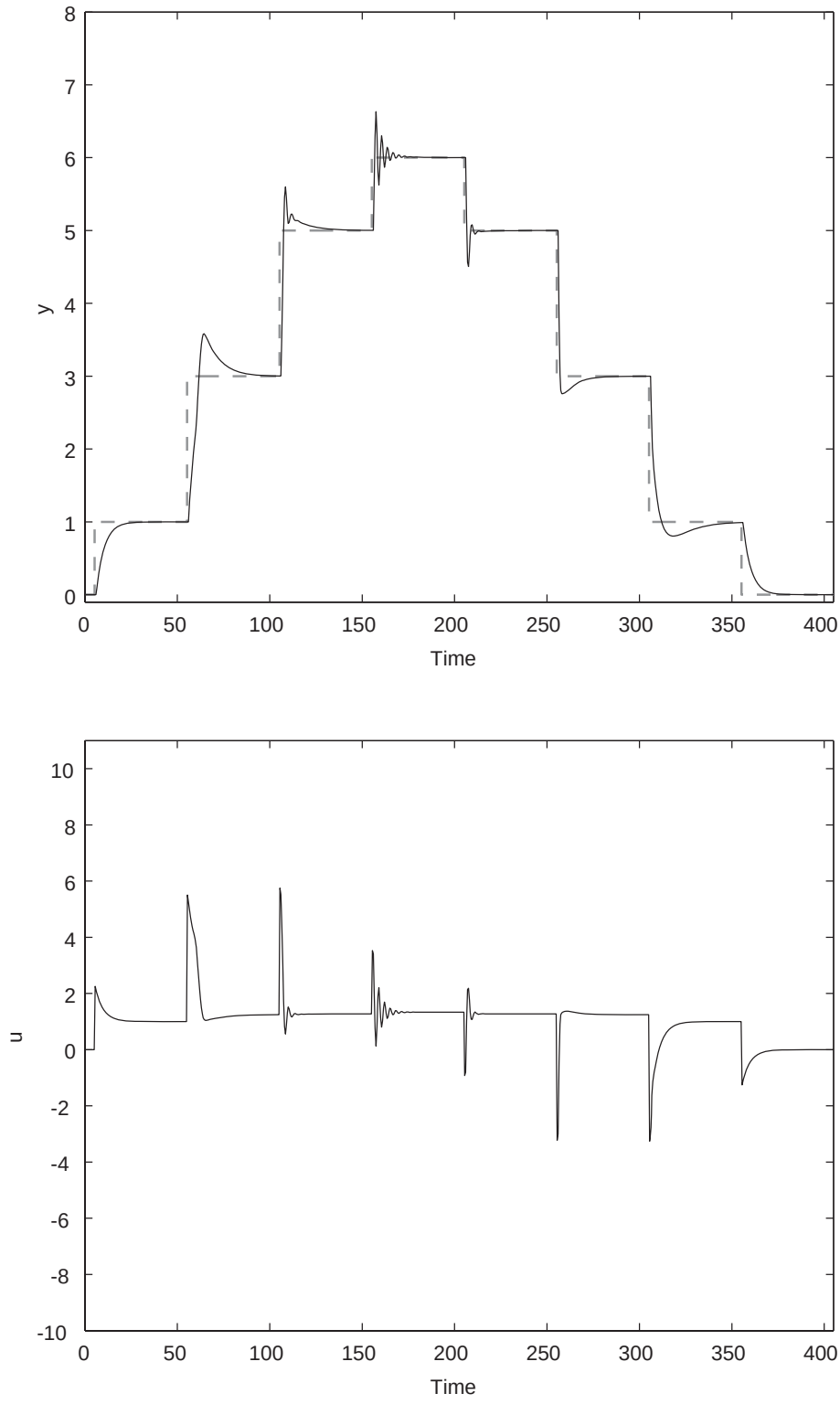

Figure 7.12: Servo response of the VRFT design

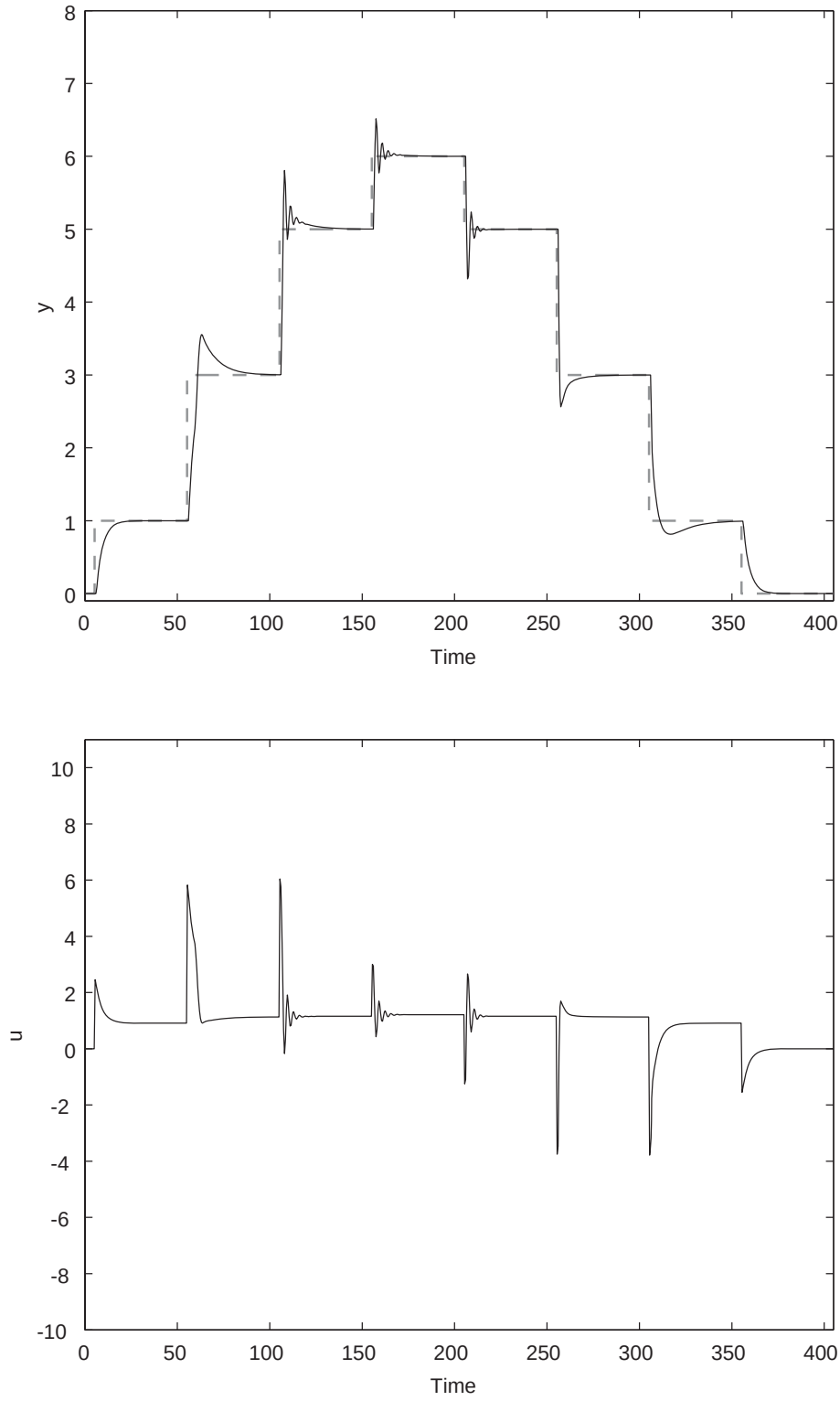

Figure 7.13: Servo response of the adaptive VRFT design under +10% modeling error

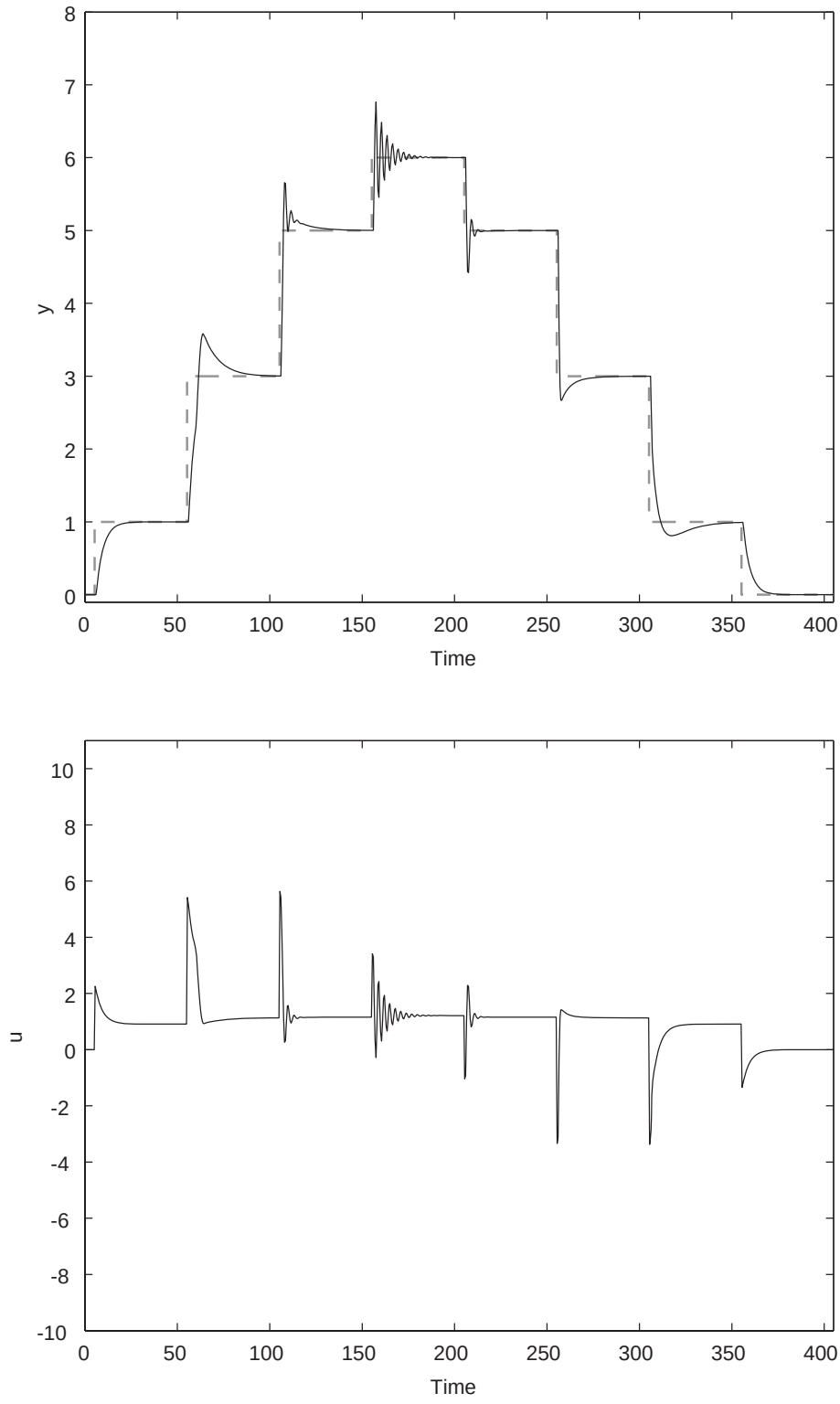

Figure 7.14: Servo response of the VRFT design under +10% modeling error

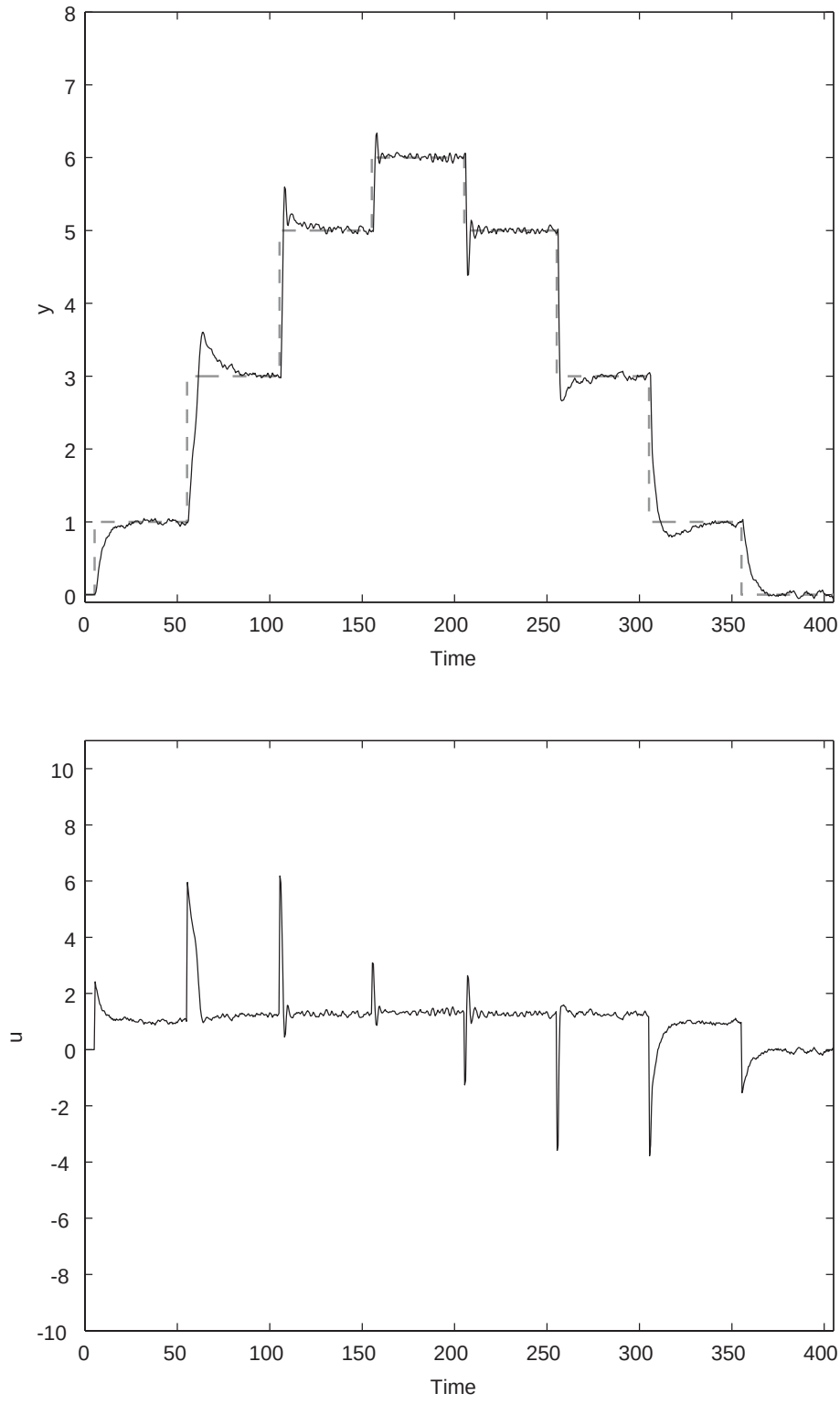

Figure 7.15: Servo response of the adaptive VRFT design in the presence of noise

cess data into the database for VRFT design, the relevant data of this expanded database corresponding to the current operating condition are selected using the k-nearest neighborhood criterion, by which PID parameters are calculated by the VRFT design at each sampling instant. Simulation results are presented to demonstrate the advantage of the proposed adaptive VRFT design over the conventional VRFT design in nonlinear process control.

Chapter 8

Adaptive PID Controller Design Directly From Plant Data - Part II

In Chapter 7, the main idea of the proposed adaptive VRFT design is to keep the database for VRFT design current with respect to the process dynamics by adding the current process data into the database. As such, the feedback controller thus obtained based on the VRFT design would utilize new process data, rather than those old process data obtained in an off-line experiment, which are more likely to be considered as the relevant data for VRFT design by using k-nearest neighborhood criterion. In this chapter, an attempt is made to update another important element in the VRFT design, namely the reference model, together with the update of database at each sampling instant. For this reason, the adaptive VRFT design developed in this chapter is termed as enhanced VRFT (EVRFT) design in order to distinguish it from AVRFT design developed in Chapter 7.

8.1 Enhanced VRFT Design

In the proposed method, the following reference model is considered:

$$T(z^{-1}) = \frac{G_{OL}(z^{-1})}{1 + G_{OL}(z^{-1})} \tag{8.1}$$

$$G_{OL}(z^{-1}) = \frac{k_0 z^{-(n_d+1)}}{1 - z^{-1}} H(z^{-1}) \tag{8.2}$$

where k_0 is the tuning parameter to be adjusted on-line by the proposed EVRFT design, n_d is the apparent time-delay and $H(z^{-1})$ is a transfer function in z^{-1}.

For a unit step change in the set-point, $r(z^{-1})$, the tracking error of the reference model, $e_{\mathrm{m}}(z^{-1})$, can be calculated as:

$$e_{\mathrm{m}}(z^{-1}) = \{1 - T(z^{-1})\} r(z^{-1}) = \frac{1}{1 - z^{-1} + k_0 z^{-(n_d+1)} H(z^{-1})} \tag{8.3}$$

It is clear from Eq. (8.3) that the minimum tracking error depends on k_0, n_d and $H(z^{-1})$. To keep $H(z^{-1})$ as simple as possible, $H(z^{-1})$ is specified as follows:

$$H(z^{-1}) = \frac{1}{1 + \chi z^{-1}} \tag{8.4}$$

where χ is a design parameter to be determined in the sequel.

With $H(z^{-1})$ specified by Eq. (8.4), the parameters k_0 and χ corresponding to the smallest tracking error can be obtained by solving the following optimization problem:

$$\min_{k_0, \chi} \sum_{i=1}^{\infty} |e_{\mathrm{m}}(i)| \tag{8.5}$$

where $e_{\mathrm{m}}(i)$ is the tracking error at the i-th sampling instant.

Table 8.1 summarizes the optimal values of k_0 and χ for various values of n_d

Table 8.1: The optimal values of k_0 and χ

n_d	0	1	2	3	4	5
k_0	1	1	0.43	0.38	0.25	0.23
χ	0	1	0.57	0.99	0.74	0.99

To apply the proposed EVRFT method to PID design, consider a PID controller given by:

$$\begin{aligned} u(k) &= u(k-1) + K_P\{e(k) - e(k-1)\} + K_I e(k) \\ &\quad + K_D\{e(k) - 2e(k-1) + e(k-2)\} \end{aligned} \tag{8.6}$$

where $u(k)$ is the manipulated value at the k-th sampling instant, $e(k)$ is the error between process output and its set-point at the k-th sampling instant, and K_P, K_I and K_D are PID parameters.

Combining Eqs. (8.1), (8.2) and (8.4), the virtual input $\tilde{u}(z^{-1})$ in the VRFT design can be calculated by Eq. (7.2) to obtain as

$$\tilde{u}(z^{-1}) = \left[K_P + \frac{K_I}{1-z^{-1}} + K_D\left(1-z^{-1}\right)\right] \frac{(1-z^{-1})(1+\chi z^{-1})}{k_0 z^{-(n_d+1)}} y(z^{-1}) \tag{8.7}$$

Equation (8.7) can be recast as

$$\tilde{u}(k) = \Psi(k)\mathbf{K} \tag{8.8}$$

where

$$\Psi(k) = \begin{bmatrix} \psi_P(k) & \psi_I(k) & \psi_D(k) \end{bmatrix} \tag{8.9}$$

$$\mathbf{K} = \begin{bmatrix} K_P & K_I & K_D \end{bmatrix}^{\mathrm{T}} \tag{8.10}$$

$$\begin{aligned}\psi_P(k) &= \frac{1}{k_0}\{y(k+n_d+1)+(\chi-1)y(k+n_d) \\ &\quad -\chi y(k+n_d-1)\} \qquad (8.11)\\ \psi_I(k) &= \frac{1}{k_0}\{y(k+n_d+1)+\chi y(k+n_d)\} \qquad (8.12)\\ \psi_D(k) &= \frac{1}{k_0}\{y(k+n_d+1)+(\chi-2)y(k+n_d) \\ &\quad -(2\chi-1)y(k+n_d-1)+\chi y(k+n_d-2)\} \qquad (8.13)\end{aligned}$$

Equation (7.3) is then expressed by

$$J_1(\mathbf{K}) = \min_{\mathbf{K}} \frac{1}{N}\|\mathbf{u}-\Psi\mathbf{K}\|^2 \qquad (8.14)$$

subject to the sign constraints of PID controller parameters:

$$K_P, K_I, K_D \geq 0$$

or

$$K_P, K_I, K_D < 0 \qquad (8.15)$$

where

$$\Psi = \begin{bmatrix} \psi_P(1) & \psi_I(1) & \psi_D(1) \\ \psi_P(2) & \psi_I(2) & \psi_D(2) \\ \vdots & \vdots & \vdots \\ \psi_P(N) & \psi_I(N) & \psi_D(N) \end{bmatrix} \qquad (8.16)$$

$$\mathbf{u} = \begin{bmatrix} u(1) & \cdots & u(N) \end{bmatrix}^{\mathrm{T}} \qquad (8.17)$$

Consequently, PID parameters are obtained by solving the constrained least square problems, Eqs. (8.14) and (8.15), at each sampling instant. For a specified value of n_d, the corresponding optimal parameters k_0 and χ in Table 8.1 are used to

calculate the initial PID parameters using the off-line database for VRFT design. Subsequently, k_0 will be adjusted to cope with process nonlinearity by an updating algorithm to be discussed in the next subsection. Based on this newly updated k_0 and database at each sampling instant, PID parameters are then obtained by solving the aforementioned optimization problem.

8.1.1 Updating algorithm for k_0

As discussed previously, in addition to the update of database for the VRFT design at each sampling instant as discussed in Chapter 7, the reference model employed in the VRFT design is also updated to enhance the capability of VRFT design to cope with the variation of process dynamics caused by process nonlinearity. Specifically, the parameter k_0 in the reference model will be updated at each sampling instant as it has direct effect on the PID parameters calculated by the VRFT design. It can be seen from Eq. (8.8) that the value of k_0 is proportional to the solution of PID parameters $\mathbf{K}(k)$, which can then be rewritten as:

$$\mathbf{K}(k) = k_0(k) \begin{bmatrix} k_P(k) & k_I(k) & k_D(k) \end{bmatrix}^{\mathrm{T}} \tag{8.18}$$

The objective of adjusting $k_0(k)$ is to minimize the following quadratic function:

$$J_2(k) = \frac{1}{2}\left[\{r(k) - y(k)\}^2 + \omega\{u(k) - u(k-1)\}^2\right] \tag{8.19}$$

where ω is a weight parameter.

By the steepest descent method, the updating algorithm is derived as:

$$k_0(k+1) = k_0(k) - \eta\frac{\partial J_2(k)}{\partial k_0(k)} \tag{8.20}$$

where η is the learning rate and

$$\begin{aligned}\frac{\partial J_2(k)}{\partial k_0(k)} &= \frac{\partial J_2(k)}{\partial u(k)}\frac{\partial u(k)}{\partial k_0(k)} \\ &= \left[\omega\{u(k)-u(k-1)\} - e(k)\frac{\partial y(k)}{\partial u(k)}\right]\cdot\Big\{k_P\Delta e(k) \\ &\quad +k_I e(k)+k_D\delta e(k)\Big\} \\ &\simeq \left[\omega\{u(k)-u(k-1)\} - e(k)\frac{y(k)-y(k-1)}{u(k)-u(k-1)}\right]\cdot \\ &\quad \Big\{k_P\Delta e(k)+k_I e(k)+k_D\delta e(k)\Big\} \qquad (8.21)\end{aligned}$$

$$\begin{aligned}\Delta e(k) &= e(k)-e(k-1) \\ \delta e(k) &= \Delta e(k)-\Delta e(k-1) \qquad (8.22)\end{aligned}$$

In the proposed design, the following rules are used to update the learning rate: (i) if the increment of J_2 is more than the threshold, k_0 remains unchanged and the learning rate is decreased by a factor l_{dec}, i.e. $\eta(k+1) = l_{dec}\eta(k)$; (ii) if the absolute value of the change of J_2 is within the threshold, only k_0 is updated; otherwise, (iii) k_0 is updated and the learning rate is increased by a factor l_{inc}, i.e. $\eta(k+1) = l_{inc}\eta(k)$.

The following gives an outline of the computational algorithm for the proposed EVRFT design of PID controller:

Step 1 Input $(u(k))$ and output $(y(k))$ identification data which characterize the dynamics of nonlinear system are assumed to be available and the off-line database for VRFT design is constructed as $(\bar{\mathbf{x}}_i)_{i=1\sim N}$;

Step 2 Given the apparent time-delay n_d, learning rate η, the weight ω and the number of nearest neighborhood, the corresponding parameters k_0 and χ are obtained from Table 8.1;

Step 3 At each sampling instant, based on the current database for VRFT design, the relevant data is selected according to Eq. (7.23), by which PID parameters are computed by solving the optimization problem, Eq. (8.14), subject to the constraint, Eq. (8.15), and the manipulated variable $u(k)$ is obtained by Eq. (8.6);

Step 4 The database for VRFT design is augmented by adding the current process data $y(k)$ and $u(k)$ and k_0 is updated by Eq. (8.20);

Step 5 Set $k = k + 1$ and go to Step 3.

8.2 Examples

Example 1 Consider the following nonlinear process given by Yamamoto and Hinamoto (2004):

(1) System 1

$$\begin{aligned} y(k) &= 0.6y(k-1) - 0.1y(k-2) + 1.2x(k-1) - 0.1x(k-2) \\ x(k) &= 1.5u(k) - 1.5u^2(k) + 0.5u^3(k) \end{aligned} \tag{8.23}$$

(2) System 2

$$\begin{aligned} y(k) &= 0.5y(k-1) - 0.05y(k-2) + 1.2x(k-1) - 0.1x(k-2) \\ x(k) &= 1.5u(k) - 1.5u^2(k) + 0.5u^3(k) \end{aligned} \tag{8.24}$$

To construct the initial database for VRFT design, input and output data are generated by introducing uniformly random step with distribution of [0 0.005] as

shown in Figure 8.1. The tuning parameters for the enhanced VRFT (EVRFT) design are specified by $n_d = 1$, $\omega = 0.1$, $\eta = 0.5$ and $k = 120$. For the purpose of comparison, adaptive VRFT (AVRFT) design developed in Chapter 7 is employed and its parameters are chosen as $A = 0.53$ and $k = 180$.

To evaluate two adaptive VRFT designs, successive set-point changes as illustrated in Figure 8.2 are conducted. It is clear that EVRFT design gives better performance than the AVRFT design. The tracking errors for these two designs in term of MAE are 0.0306 (EVRFT) and 0.0622 (AVRFT), respectively. Figure 8.3 shows the updating of PID parameters and k_0 in the EVRFT design for the aforementioned closed-loop responses.

Figure 8.4 compares the robustness of these two controllers by assuming that the process dynamics are changed from Eq. (8.23) to Eq. (8.24) at $k = 10$. As can be seen, EVRFT design still maintains superior control performance compared with AVRFT design, leading to 56.3% reduction of MAE. Lastly, to study the effect of process noise on the proposed design, both input and output are corrupted by 5% Gaussian white noise, which means that the initial database contains the corrupted process data. As shown in Figure 8.5, the proposed EVRFT design can yield reasonably good control performance in the presence of process noise.

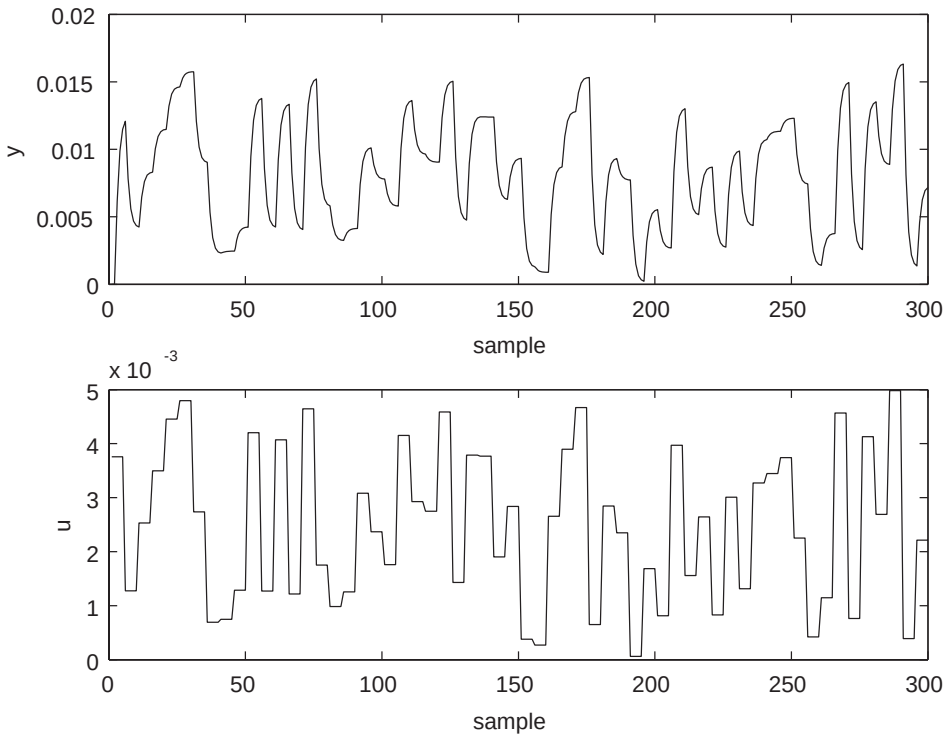

Figure 8.1: Input and output data used to construct the initial database for VRFT design

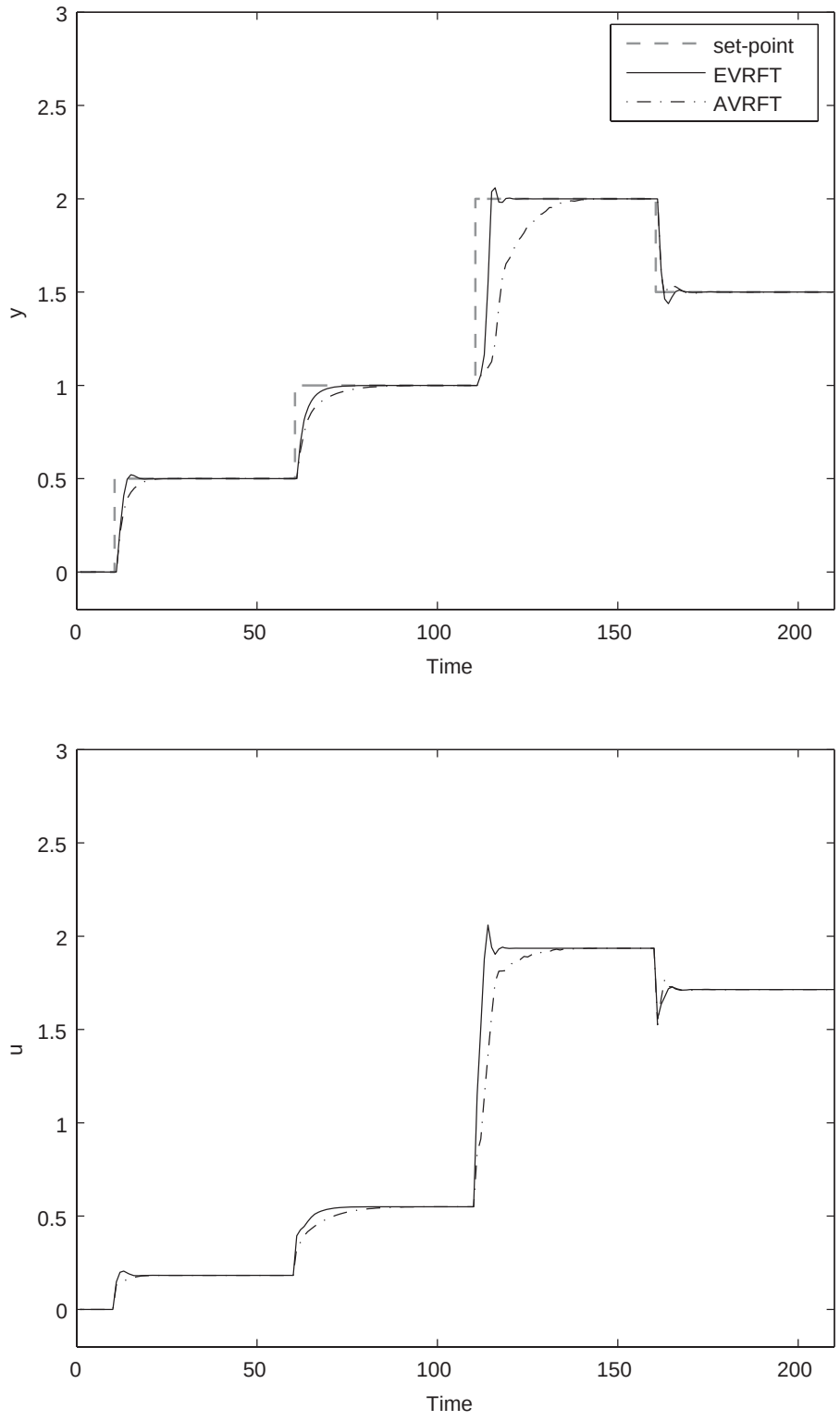

Figure 8.2: Servo responses of two VRFT designs

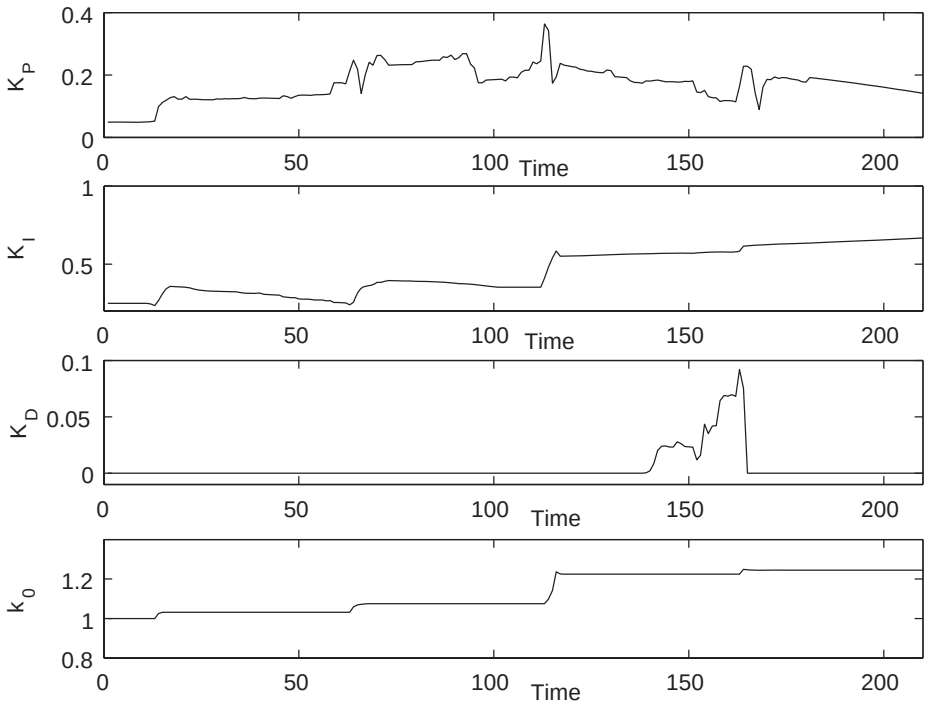

Figure 8.3: Updating of the PID parameters and k_0 by the EVRFT design

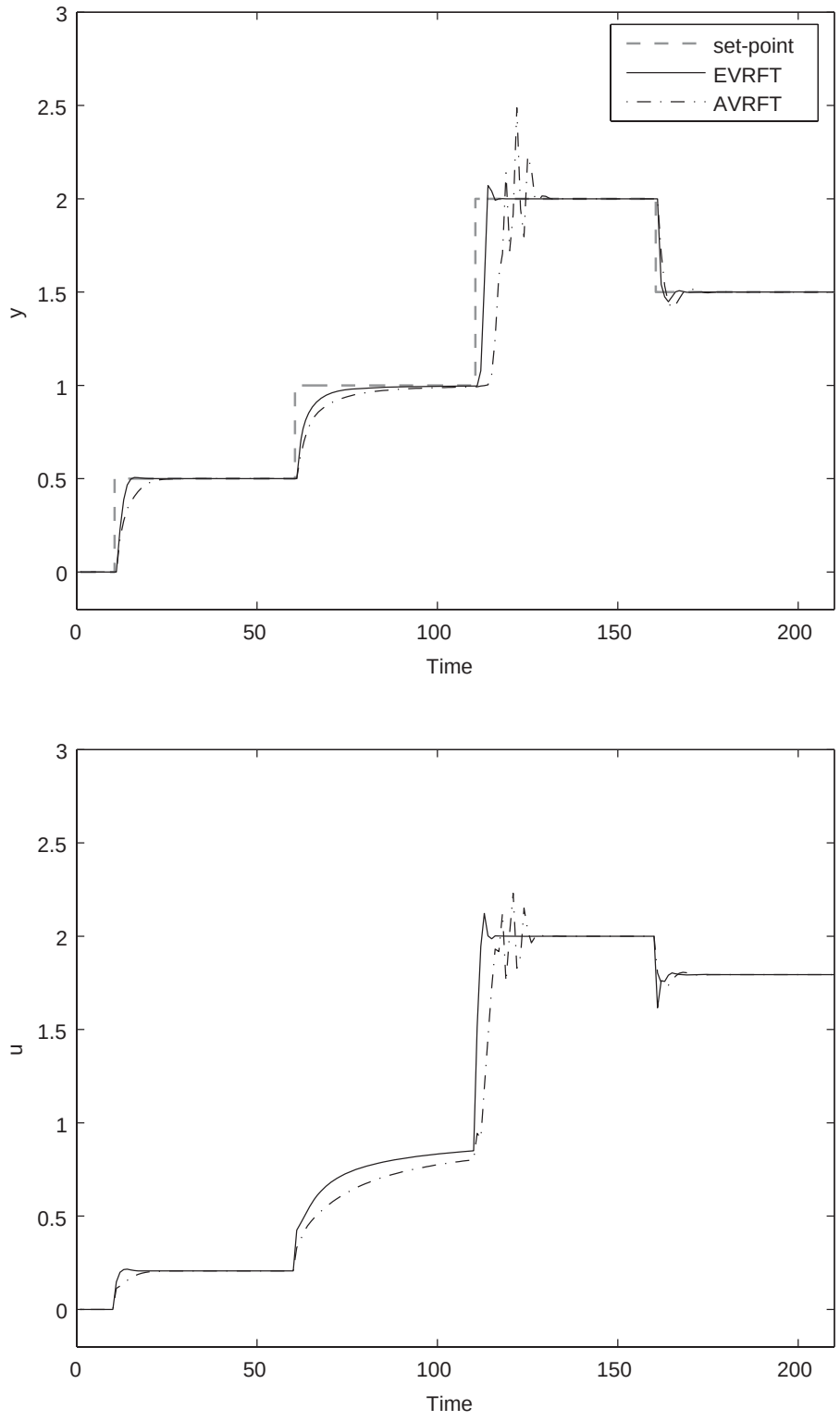

Figure 8.4: Servo responses of two VRFT designs in the presence of modeling error

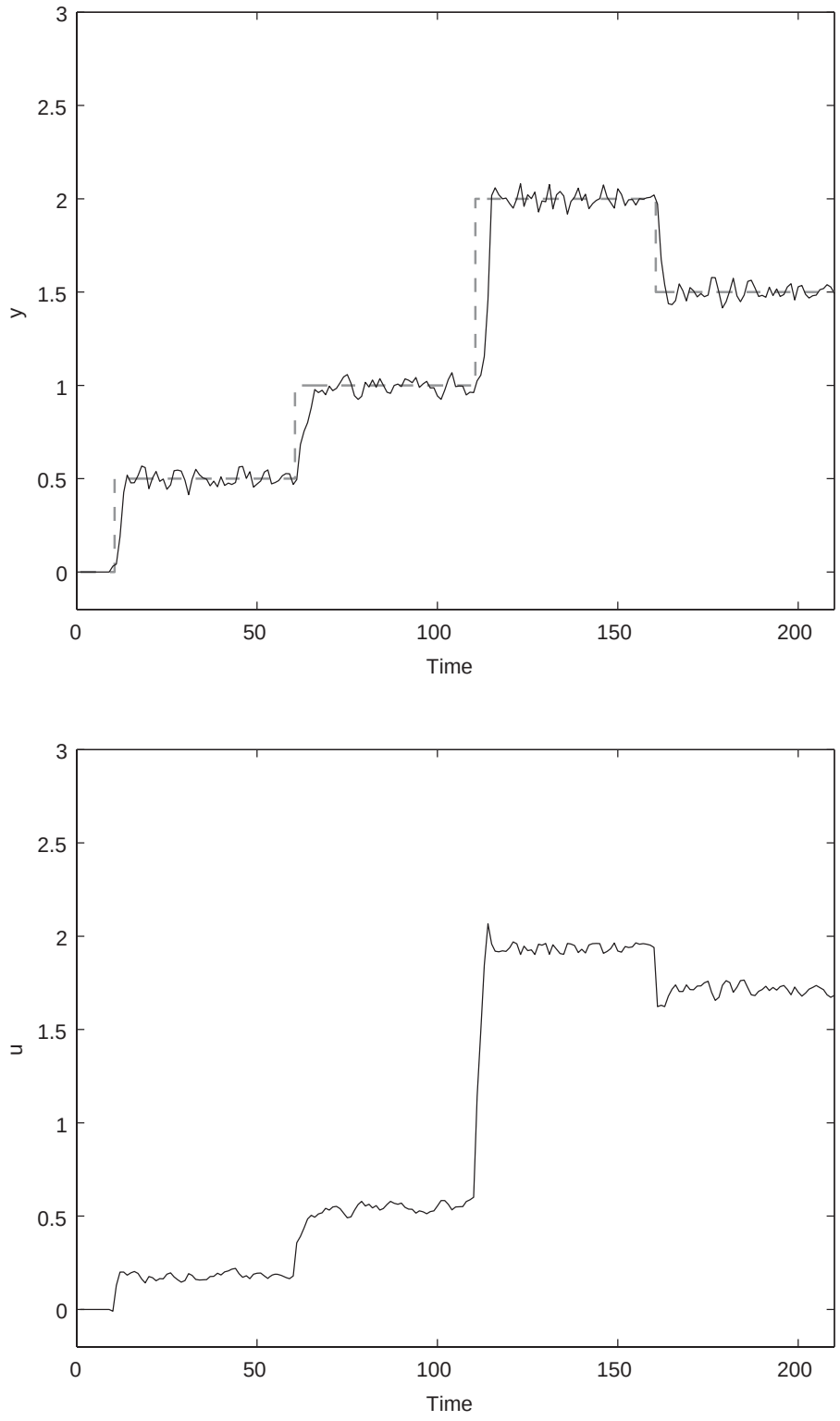

Figure 8.5: Servo response of the EVRFT design in the presence of noise

Example 2 Consider a continuous polymerization reactor studied in Chapters 4 to 7. The model parameters and steady-state operation condition are given in Tables 4.1 and 4.2. To design PID controller by the EVRFT and AVRFT methods, the identical off-line process data in Chapter 4 are employed. Furthermore, the tuning parameters for the EVRFT are specified by $n_d = 1$, $\omega = 1 \times 10^{-5}$, $\eta = 2 \times 10^{-4}$ and $k = 120$, while $A = 0.78$ and $k = 350$ are chosen for the AVRFT design.

To compare the performances of two controllers, successive set-point changes between 25000.5 kg/kmol and 12500 kg/kmol are conducted. As can be seen from Figure 8.6, EVRFT design has better performance than that achieved by the AVRFT design, resulting in the reduction of MAE by 31.7% as compared with the AVRFT design. Figure 8.7 shows the updating of PID parameters and k_0 by the EVRFT design in the aforementioned closed-loop responses.

Figures 8.8 and 8.9 compare the disturbance rejection capability of two controllers with respect to unmeasured $\pm 10\%$ step disturbances in the inlet initiator concentration $C_{\mathrm{I_{in}}}$. Again, EVRFT design has superior or similar control performance compared with the AVRFT design. Table 8.2 summarizes the performance improvement achieved by the proposed design as measured by the MAEs. Next, to evaluate the robustness of the proposed control strategy, it is assumed that there exist 10% modeling error in the kinetic parameter k_{I} and 20% error in the coefficients of the D_1 and M_{m}. As can be seen from Figure 8.10, EVRFT design outperforms AVRFT design by reducing MAE by 24.4% in the aforementioned set-point changes. Lastly, 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 VRFT design also contains the corrupted process data. As shown

in Figure 8.11, the proposed controller can yield reasonably good control performance in the presence of process noise.

To investigate the effect of time-delay on the EVRFT design, time-delay of 0.06 h (two sampling time) is assumed in the measurement of process output. The tuning parameters for two adaptive VRFT designs remain the same as those mentioned previously, except that $n_d = 3$ is chosen for the EVRFT design and $n_d = 2$ for AVRFT design. For the conventional VRFT design, $n_d = 2$ and $A = 0.6$ are chosen. It can be seen from Figure 8.12 that the EVRFT design outperforms the other two VRFT designs in the aforementioned set-point changes.

Table 8.2: Control performance comparison of two VRFT designs

	Tracking error (MAE)	
	EVRFT	AVRFT
Servo Response	542.74	794.07
Servo Response*	668.19	883.49
+10% in $C_{\text{I}_{\text{in}}}$ at y=25000.5	22.06	41.72
+10% in $C_{\text{I}_{\text{in}}}$ at y=18750	41.78	57.29
+10% in $C_{\text{I}_{\text{in}}}$ at y=13500	67.64	63.92
-10% in $C_{\text{I}_{\text{in}}}$ at y=25000.5	26.27	49.11
-10% in $C_{\text{I}_{\text{in}}}$ at y=18750	50.94	62.83
-10% in $C_{\text{I}_{\text{in}}}$ at y=13500	82.73	75.95

* In the presence of modeling error

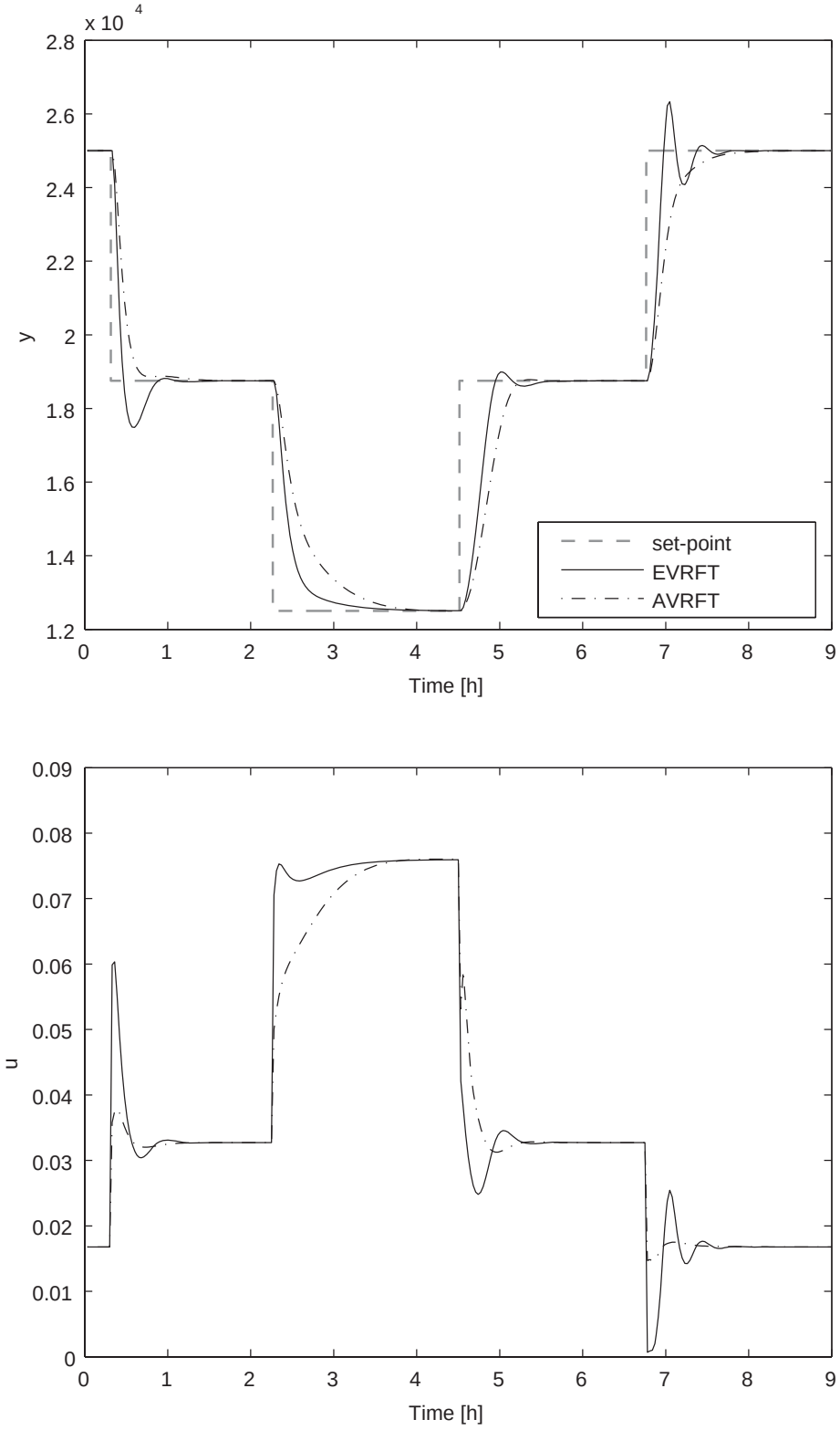

Figure 8.6: Servo responses of two VRFT controller designs

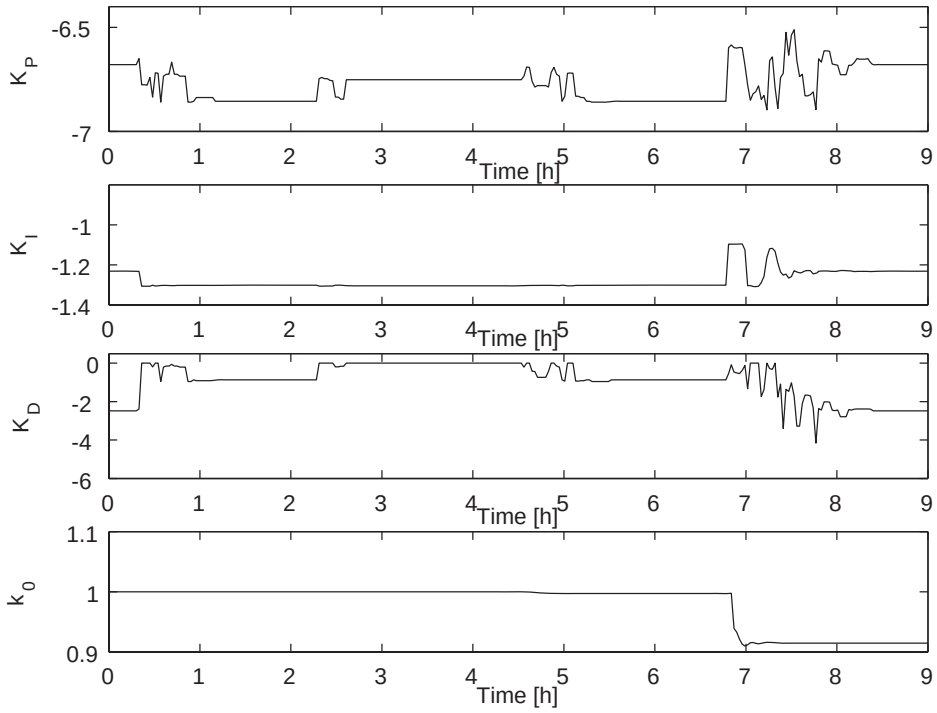

Figure 8.7: Updating of the PID parameters and k_0 for servo response

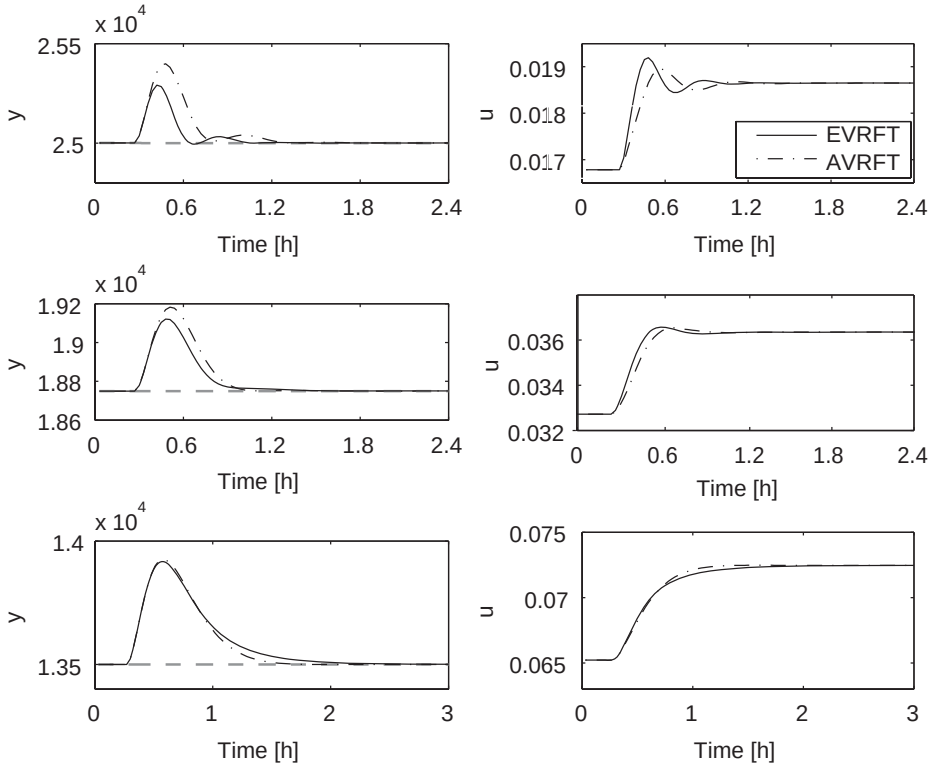

Figure 8.8: Closed-loop responses for -10% step change in $C_{\mathrm{I_{in}}}$

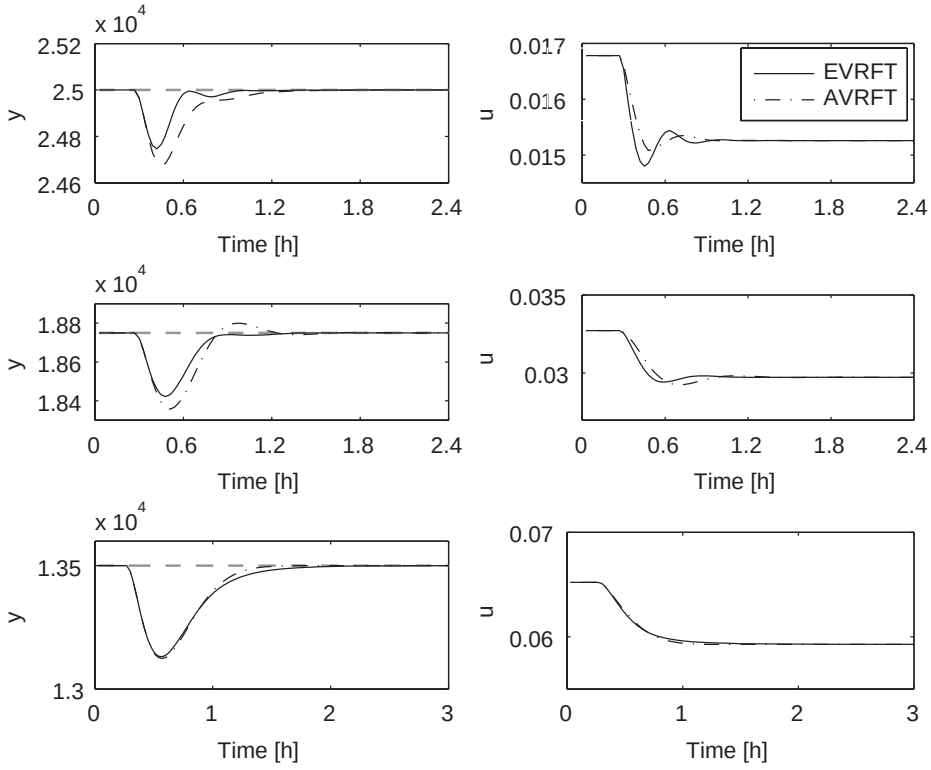

Figure 8.9: Closed-loop responses for +10% step change in $C_{\mathrm{I_{in}}}$

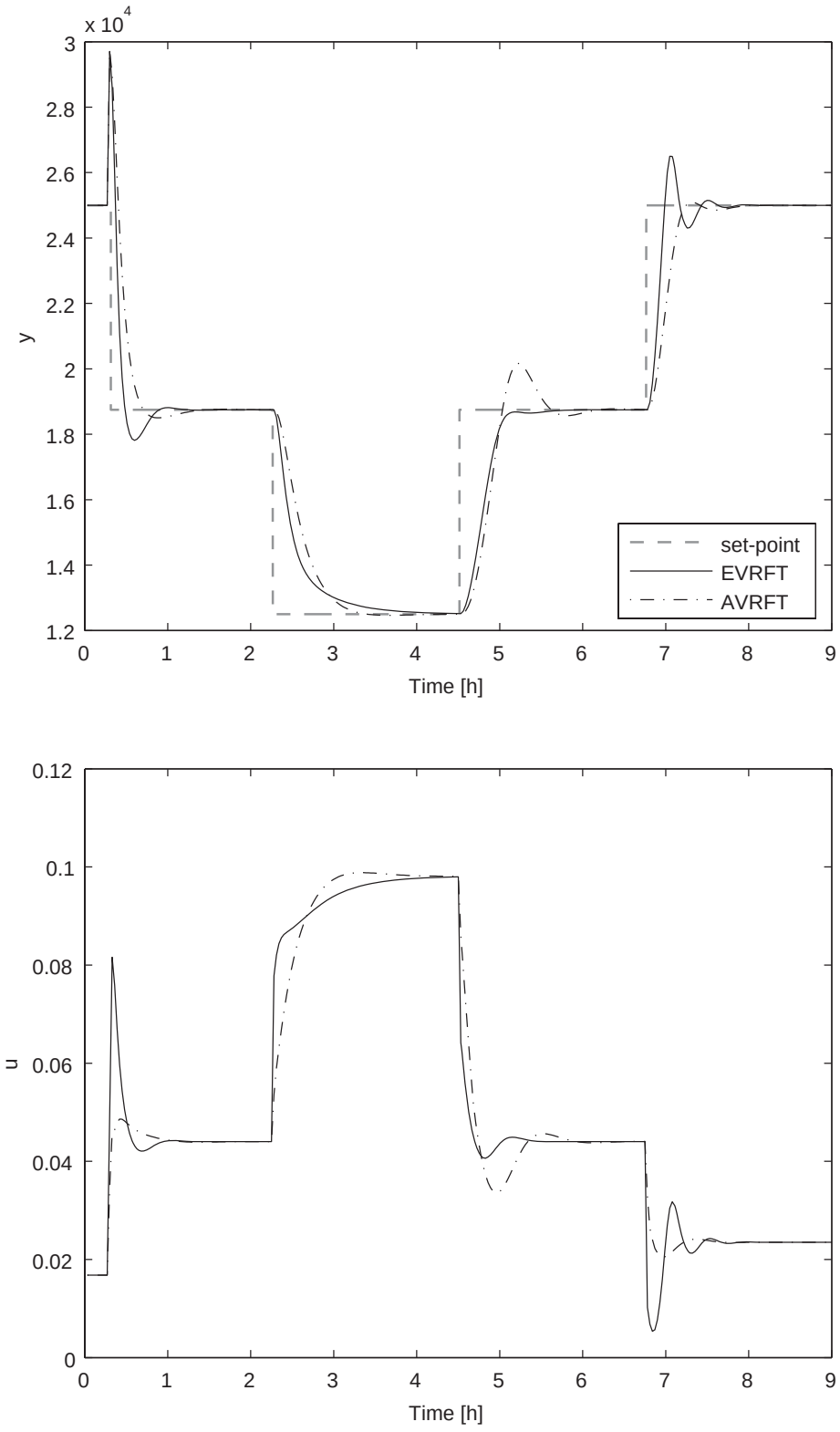

Figure 8.10: Servo responses of two VRFT designs in the presence of modeling error

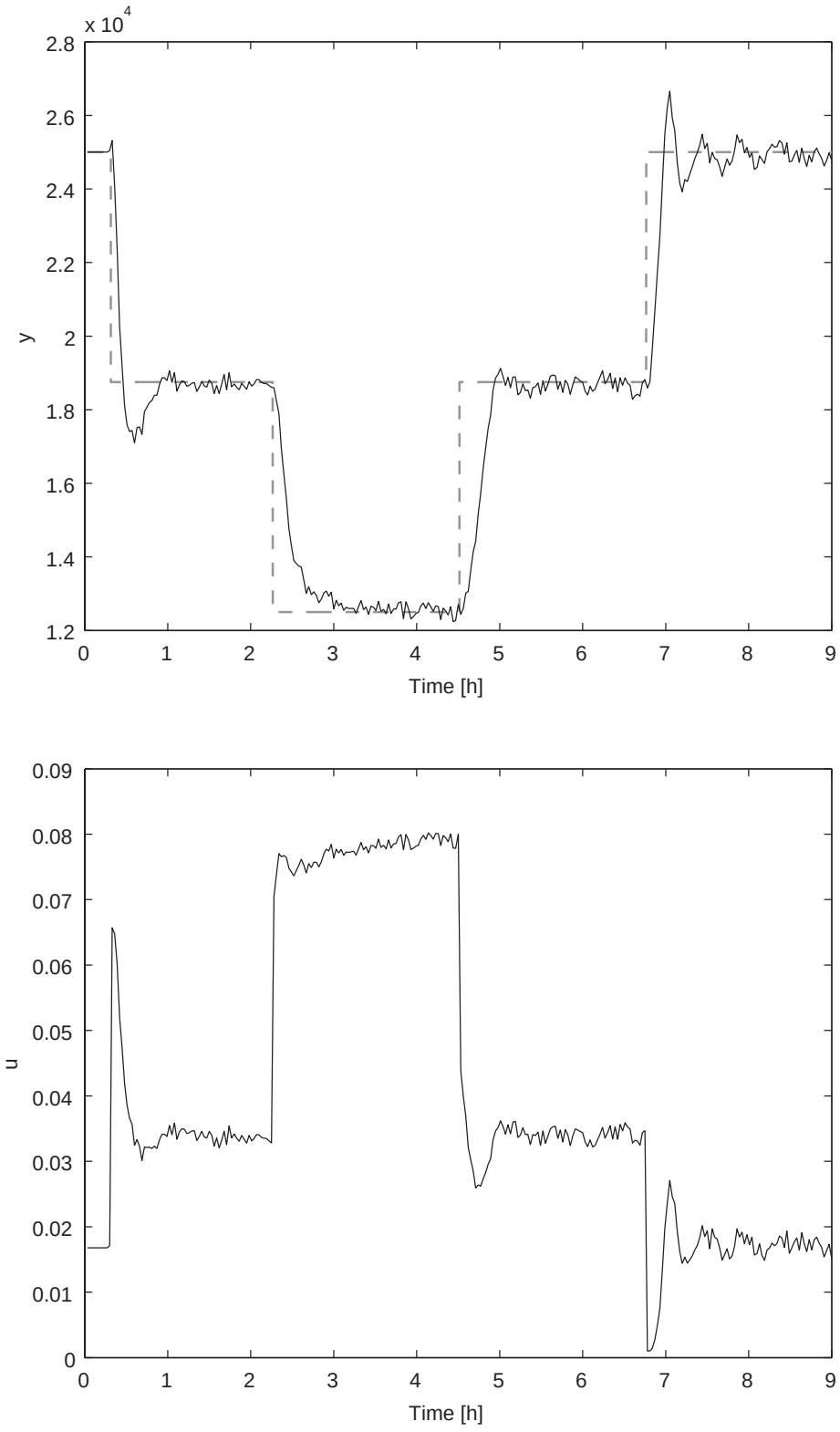

Figure 8.11: Servo response of the EVRFT designs in the presence of noise

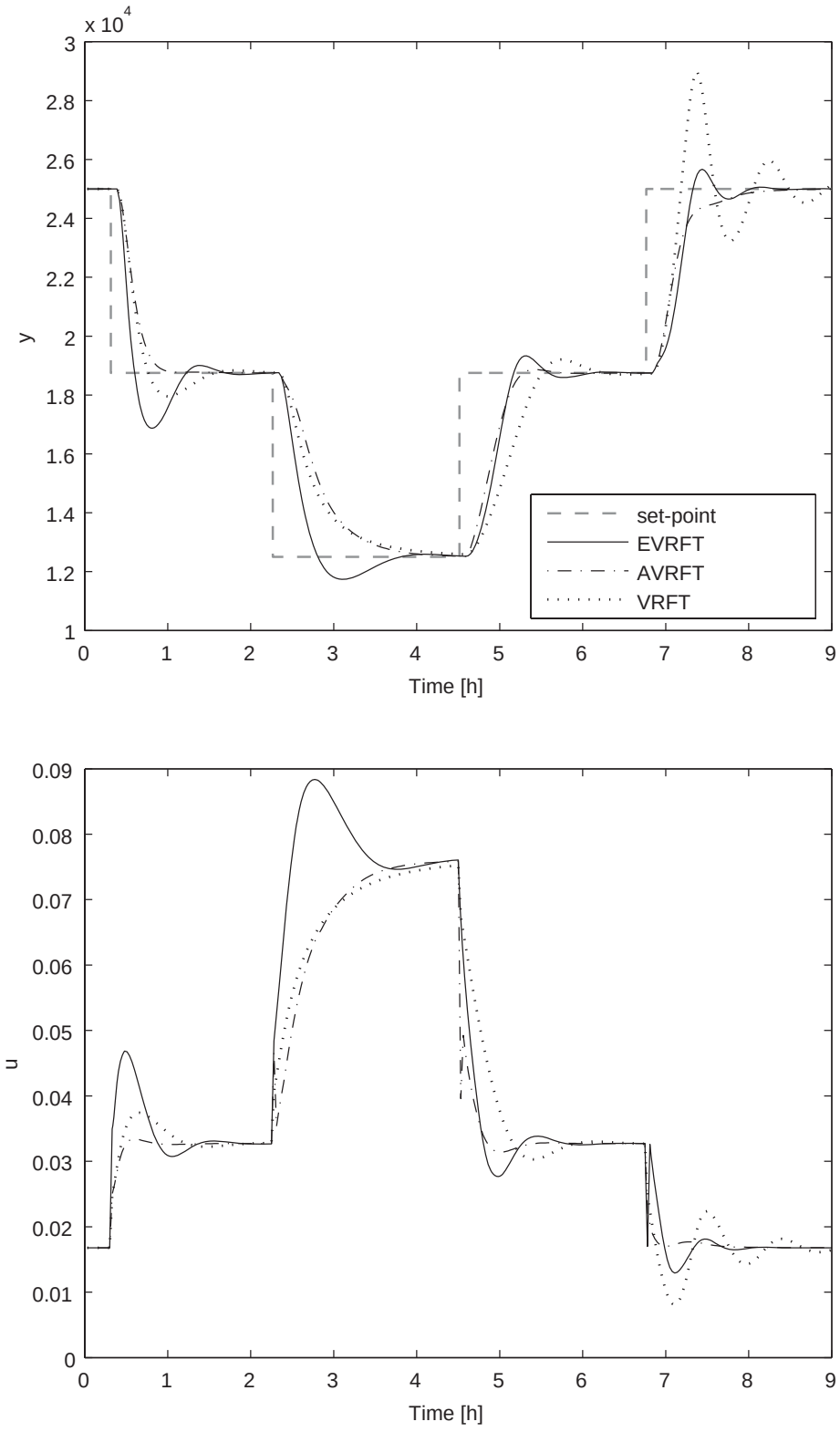

Figure 8.12: Servo responses of three VRFT designs in the presence of time-delay

Example 3 Consider the first-order plus dead-time process studied in Chapters 4 and 7. Two PID controllers are designed based on the EVRFT and AVRFT methods. In the EVRFT design, the tuning parameters are specified by $n_d = 2$, $\omega = 0.0032$, $\eta = 0.003$ and $k = 190$, whereas $A = 0.88$ and $k = 630$ are selected for the AVRFT design as discussed in Chapter 7.

Figure 8.13 shows the servo performance of the proposed controller for the successive set-point changes between 0 and 6. Compared with figures in Chapter 7, it can be seen that EVRFT design has better performance than that achieved by the AVRFT design, resulting in the reduction of MAE by 19.2%. Next, to evaluate the robustness of the EVRFT design strategy, 10% modeling error in the process parameter $\tau_2(y)$ is assumed and the resulting servo response is shown in Figure 8.14. In this case, EVRFT design gives slightly better control performance over the AVRFT design by achieving 3.6% reduction of MAE in the aforementioned set-point changes. Lastly, to study the effect of process noise on the proposed design, both process input and output are corrupted by 3% Gaussian white noise. As shown in Figure 8.15, EVRFT design can yield reasonably good control performance in the presence of process noise.

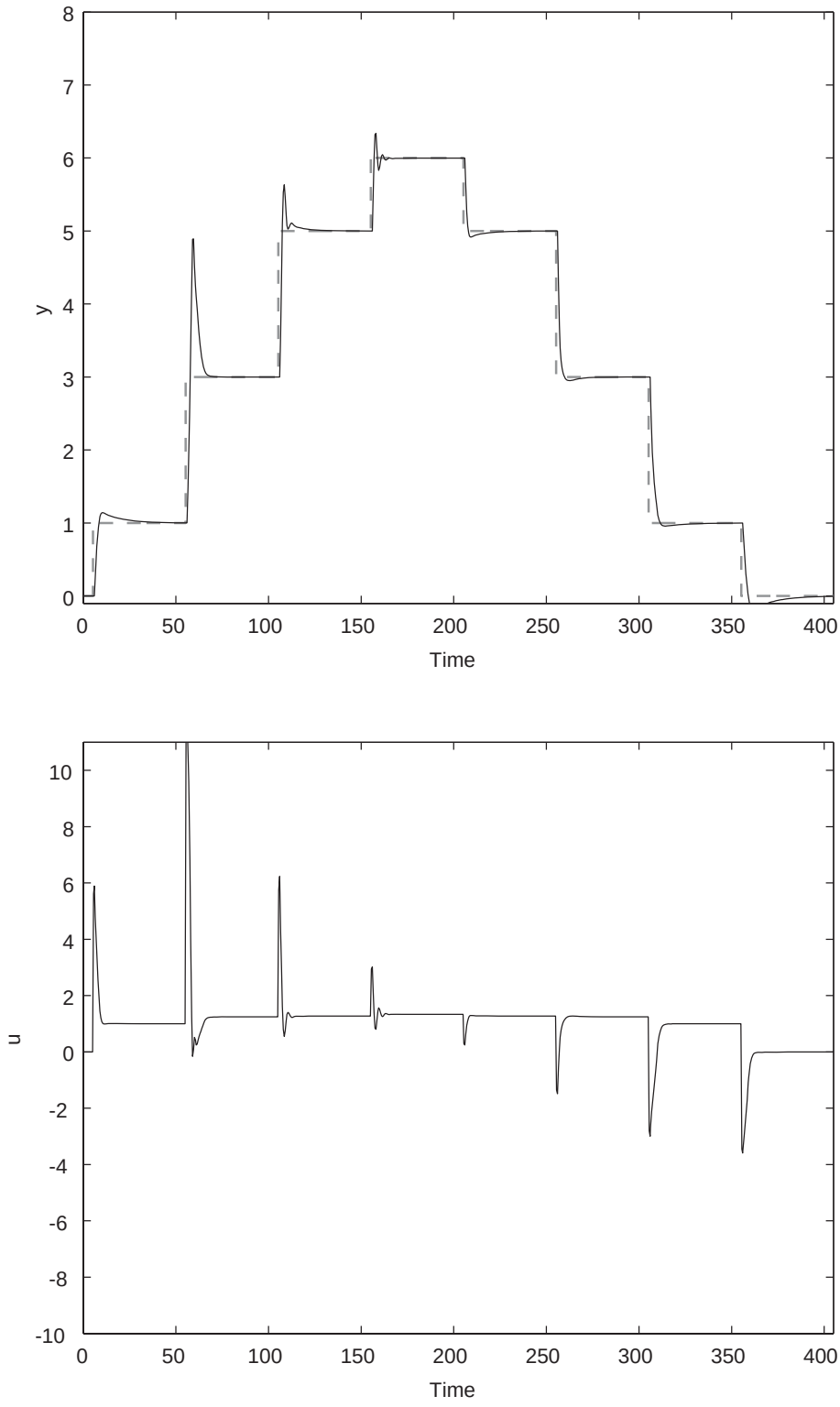

Figure 8.13: Servo response of the EVRFT design

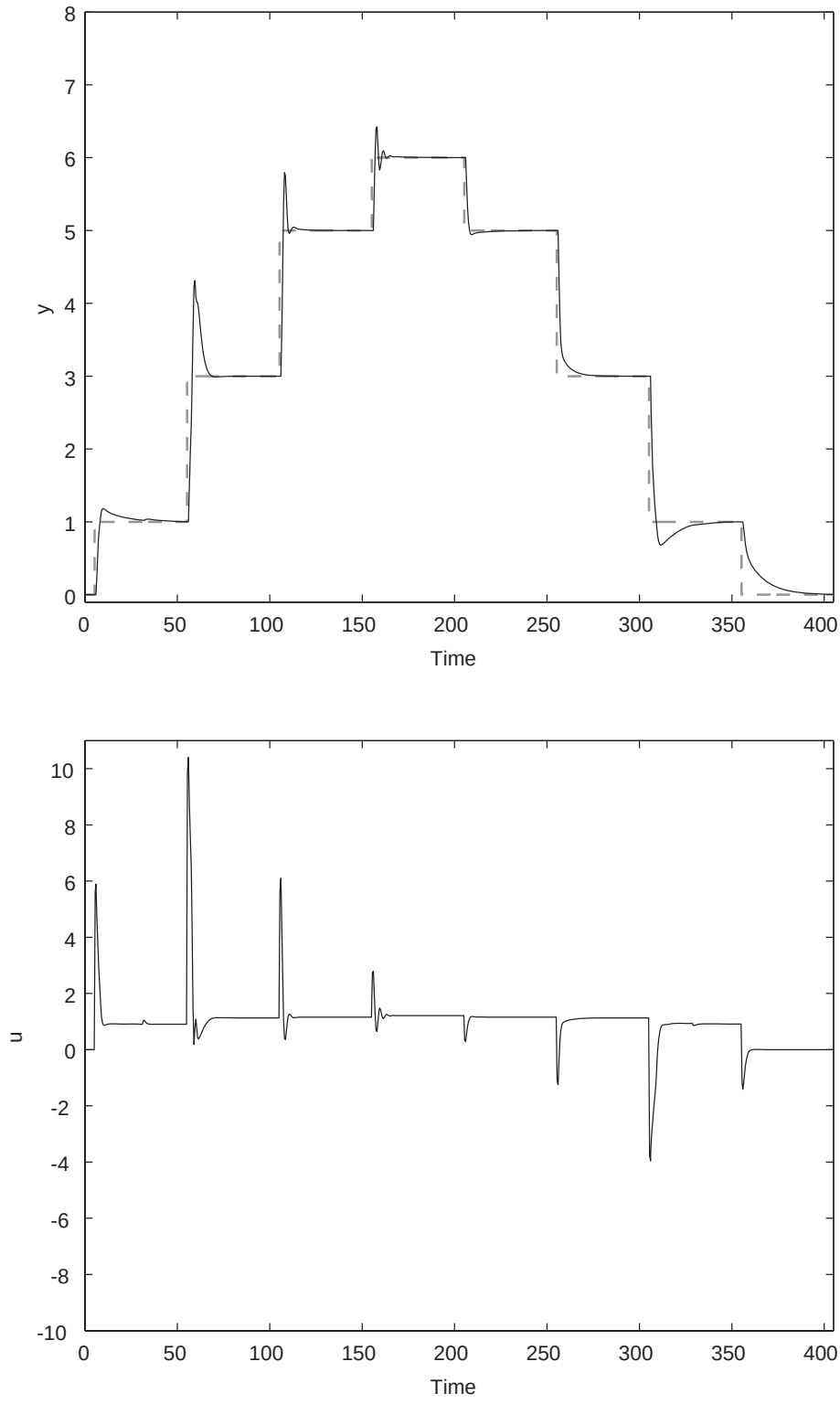

Figure 8.14: Servo response of the EVRFT design under +10% modeling error

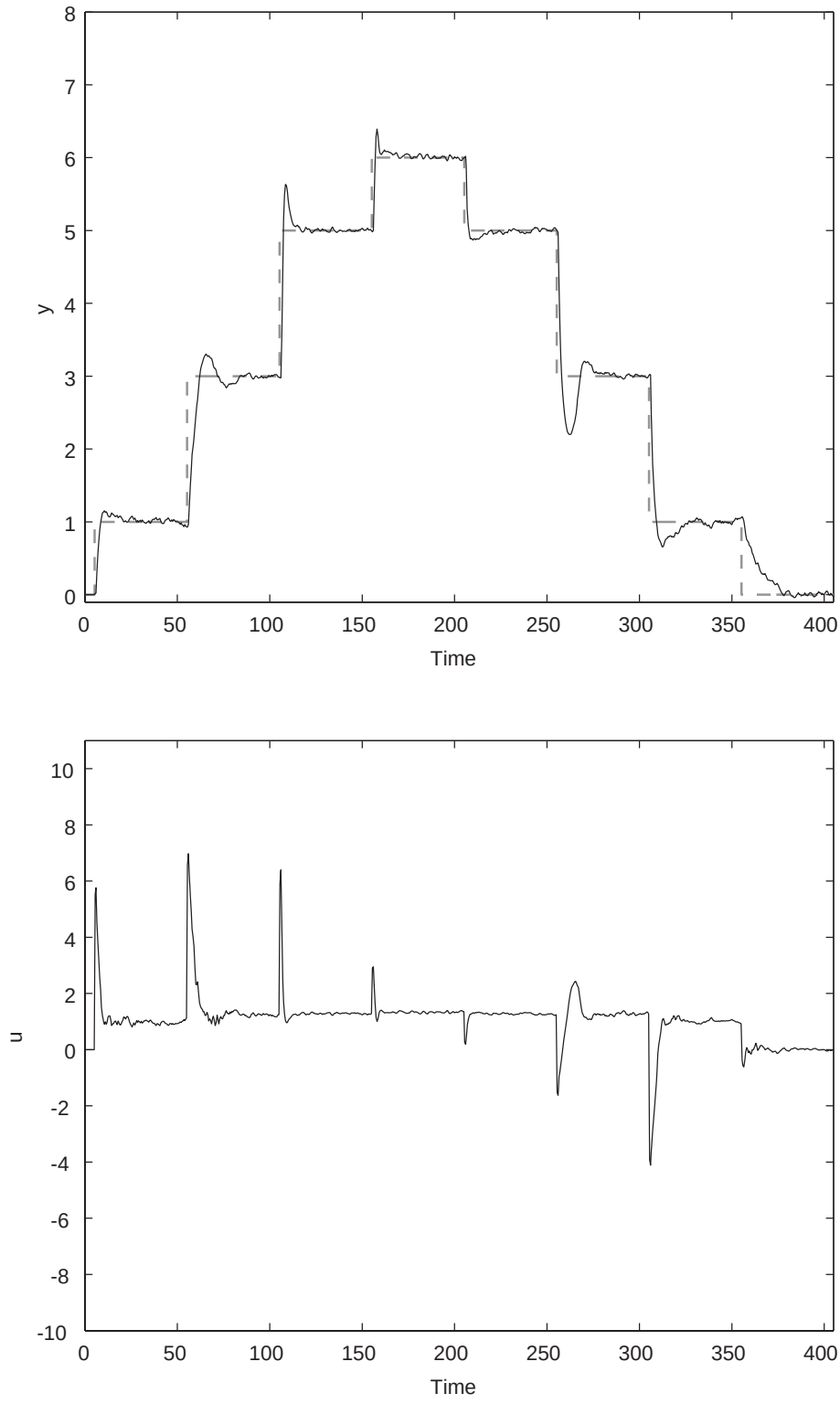

Figure 8.15: Servo response of the EVRFT design in the presence of noise

8.3 Conclusion

Based on the adaptive VRFT design provided in Chapter 7, an enhanced VRFT design method for nonlinear process control is developed in this chapter. In addition to the update of database for VRFT design, the reference model is updated as well by an updating algorithm at each sampling instant. Simulation results demonstrate the advantage of the enhanced VRFT design over the adaptive VRFT design developed in Chapter 7 and the conventional VRFT design for that matter.

Chapter 9

Conclusions and Further Work

9.1 Conclusions

In this thesis, several control strategies for nonlinear systems are developed. By integrating the just-in-time learning (JITL) technique into the controller design, four data-based control strategies are developed, meaning that data-based LQI controller design, adaptive IMC controller, self-tuning PID controller, and nonlinear GPC design. These controllers take use of the information provided by the JITL to realize online tuning of control parameters or calculation of the manipulated variable for nonlinear process control.

Firstly, to lessen the modeling requirement in the LQI design, which typically relies upon the availability of the first-principles model or Kalman filter, the proposed data-based LQI design offers an attractive design alternative because simulation results reveal that it can achieve comparable control performance as that obtained by the traditional LQI design, which is based on a set of analytical models derived from the successive linearization of the first-principles process model.

Secondly, by incorporating the JITL into the IMC design, an adaptive IMC design methodology is developed for nonlinear systems. The IMC controller parameters are updated not only based on the information provided by the JITL, but also IMC filter parameter is adjusted by an updating algorithm derived by Lyapunov method. Compared with the previous nonlinear IMC controller design methods, it is straightforward to obtain the model inversion in the proposed design.

Thirdly, a self-tuning PID design utilizing JITL model technique is developed for nonlinear process control. PID parameters are updated based on the information provided by the JITL and an updating algorithm derived from the Lyapunov method. Compared with the previous nonlinear PID controller design methods, such as neural networks based PID designs, the proposed method is more amenable for implementation.

Fourthly, a constrained GPC design based on multiple models is developed. The set of local models obtained by the JITL modeling technique is used for modeling the nonlinear systems and serves as the process model in the GPC design. As a consequence, the computational burden associated with the conventional nonlinear model predictive control algorithms can be reduced.

Lastly, two adaptive PID controllers under the VRFT framework are designed for nonlinear processes. In addition, the equivalence between the VRFT and IMC designs is established. In the first adaptive VRFT design of PID controller, the k-nearest neighborhood criterion is used to determine the relevant data in the expanded database for the VRFT design, by which PID parameters are calculated according to the design equation developed under the VRFT framework. Another adaptive VRFT design of PID controller is based on not only the update of database

for VRFT design, but also the update of reference model. In doing so, a new reference model is proposed and its associated updating equation is developed.

Simulation results are presented to illustrate the improved control performance obtained by the proposed controller designs, except the data-based LQI design, over their respective conventional counterparts. By using the polymerization reactor as an example, the control performances of five proposed controller designs are compared in Table 9.1. Furthermore, looking back to the simulation results provided in the previous chapters, the proposed nonlinear GPC design (as expected) is the only viable design approach capable of taking into account the input constraint, $F_I \geq 0.007$, while achieving the best control performances in terms of the smallest MAEs among the five proposed design methods. When the above input constraint is replaced by a soft one, i.e. $F_I \geq 0$, adaptive PID controller obtained by the EVRFT design gives the second best control performance, followed by the adaptive PID controller based on the JITL technique. Both adaptive PID controller resulting from AVRFT design and adaptive IMC controller give the worst control performance for this example. The better control performance attained by the APID (or EVRFT) design over the AIMC (or AVRFT) design maybe can be explained as a result of more tuning parameters (or more complicated control scheme) in the former design method.

Table 9.1: Comparison of five proposed controller designs

		Tracking error (MAE)		
		JITL-based GPC	EVRFT	Self-tuning PID
Servo Response		419.10	542.74	695.25
Servo Response*		517.76	668.19	840.12
+10% in $C_{I_{in}}$ at	y=25000.5	9.85	22.06	35.17
+10% in $C_{I_{in}}$ at	y=18750	15.08	41.78	57.49
+10% in $C_{I_{in}}$ at	y=13500	19.96	67.64	96.1
-10% in $C_{I_{in}}$ at	y=25000.5	14.06	26.27	40.37
-10% in $C_{I_{in}}$ at	y=18750	14.04	50.94	64.32
-10% in $C_{I_{in}}$ at	y=13500	9.27	82.73	84.25

		Adaptive IMC	AVRFT
Servo Response		799.59	794.07
Servo Response*		907.55	883.49
+10% in $C_{I_{in}}$ at	y=25000.5	45.45	41.72
+10% in $C_{I_{in}}$ at	y=18750	43.72	57.29
+10% in $C_{I_{in}}$ at	y=13500	94.42	63.92
-10% in $C_{I_{in}}$ at	y=25000.5	52.35	49.11
-10% in $C_{I_{in}}$ at	y=18750	39.27	62.83
-10% in $C_{I_{in}}$ at	y=13500	85.40	75.95

* In the presence of modeling error

9.2 Suggestions for Further Work

There are few open questions that warrant further investigation, which are summarized in below.

The data-based control strategies developed in this thesis are only applicable to the single-input single-output systems. Therefore, it is of practical importance to extend the proposed design methods to the multivariable systems, which are often encountered industrial control practices.

Another open problem is fault tolerant controller design by integrating the tasks of process monitoring and controller design into an unified framework. Specifically, when a process fault is identified and diagnosed in a control system, how can such monitoring information be utilized to redesign the controller in order to maintain acceptable control performance? Given the complexity of chemical processes, this problem is indeed a challenging one worthwhile further investigation.

Appendix A

Analytical Linear Model for Example 2 in Chapter 3

The analytical local model for Eq. (3.20) is derived as follows:

$$\begin{bmatrix} C_{\mathrm{A}}(k+1) \\ C_{\mathrm{B}}(k+1) \end{bmatrix} = \mathbf{A}(k) \begin{bmatrix} C_{\mathrm{A}}(k) \\ C_{\mathrm{B}}(k) \end{bmatrix} + \mathbf{B}(k)F(k) \tag{A.1}$$

$$y(k) = \mathbf{C}(k) \begin{bmatrix} C_{\mathrm{A}}(k) \\ C_{\mathrm{B}}(k) \end{bmatrix} \tag{A.2}$$

where

$$\mathbf{A}(k) = \begin{bmatrix} e^{A_1(k)\Delta t} & 0 \\ \frac{50}{A_2(k)-A_1(k)}(e^{A_2(k)\Delta t} - e^{A_1(k)\Delta t}) & e^{A_2(k)\Delta t} \end{bmatrix}$$

$$\mathbf{B}(k) = \begin{bmatrix} \frac{B_1(k)}{A_1(k)}(e^{A_1(k)\Delta t} - 1) \\ \frac{50B_1(k)-A_1(k)B_2(k)}{A_1(k)A_2(k)} + \frac{50B_1(k)}{A_1(k)(A_1(k)-A_2(k))}e^{A_1(k)\Delta t} + \frac{50B_1(k)+(A_2(k)-A_1(k))B_2(k)}{A_2(k)(A_2(k)-A_1(k))}e^{A_2(k)\Delta t} \end{bmatrix}$$

$$\mathbf{C}(k) = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

and $A_1(k) = -50 - 20C_A(k-1) - F(k-1)$, $A_2(k) = -100 - F(k-1)$, $B_1(k) = 10 - C_A(k-1)$, $B_2(k) = -C_B(k-1)$, and Δt is the sampling time.

References

Abonyi, J., Bodizs, A., Nagy, L., and Szeifert, F. (2000). Hybrid fuzzy convolution model and its application in predictive control. *Chemical Engineering Research and Design*, **78**, 597-604.

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

Alpbaz, M., Hapoglu, H., Ozkan, G., and Altuntas, S. (2006). Application of self-tuning PID control to a reactor of limestone slurry titrated with sulfuric acid. *Chemical Engineering Journal*, **116**, 19-24.

Altinten, A., Erdogan, S., Alioglu, F., Hapoglu, H., and Alpbaz, M. (2004). Application of adaptive PID control with genetic algorithm to a polymerization reactor. *Chemical Engineering Communications*, **191**, 1158-1172.

Andrasik, A., Meszaros, A., and de Azevedo, S. E. (2004). On-line tuning of a neural PID controller based on plant hybrid modeling. *Computers and Chemical Engineering*, **28**, 1499-1509.

Aoyama, A., Doyle, F. J. III, and Venkatasubramanian, V. (1995). Control-affine fuzzy neural network approach for nonlinear process control. *Journal of Process Control*, **5**, 375-386.

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

Åström, K. J., and Wittenmark, B. (1995). Adaptive control, 2^{nd} Edition. Addison-Welsey.

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

Atkeson, C. G., Moore, A. W., and Schaal, S. (1997b). Locally weighted learning for control. *Artificial Intelligence Review*, **11**, 75-113.

Berber, R., and Coskun, S. (1996). Dynamic simulation and quadratic dynamic matrix control of an industrial low density polyethylene reactor. *Computers and Chemical Engineering*, **20**, S799-S804.

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

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

Bisowarno, B. H., Tian, Y. C., and Tade, M. O. (2004). Adaptive control of an ETBE reactive distillation column. *Journal of Chemical Engineering of Japan*, **37**, 210-216.

Bontempi, G., Birattari, M., and Bersini, H. (1999). Lazy learning for local modeling and control design. *International Journal of Control*, **72**, 643-658

Bontempi, G., Bersini, H., and Birattari, M. (2001). The local paradigm for mod-

eling 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.

Campi, M. C., Lecchini, A., and Savaresi, S. M. (2000). Virtual reference feedback tuning (VRFT): a new direct approach to the design of feedback controllers. *Proceedings of the 39th IEEE Conference on Decision and Control*, 623-629.

Campi, M. C., Lecchini, A., and Savaresi, S. M. (2002). Virtual reference feedback tuning: a direct method for the design of feedback controllers. *Automatica*, **38**, 1337-1346.

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, W. D., Hwang, R. C., and Hsieh, J. G. (2002). A self-tuning PID control for a class of nonlinear systems based on the Lyapunov approach. *Journal of Process Control*, **12**, 233-242.

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

Cheng, C., and Chiu, M. S. (2004). A new data-based methodology for nonlinear process modeling. *Chemical Engineering Science*, **59**, 2801-2810.

Chu, J. Z., Tsai, W. Y., Jang, S. S., Shie, S. S., Ling, P. H., and Jiang, S. J. (2003). Multistep model predictive control based on artificial neural networks. *Industrial Engineering Chemistry Research*, **42**, 5215-5228.

Clarke, D. W., Mohtadi, C., and Tuffs, P. S. (1987). Generalized predictive control - part I. the basic algorithm. *Automatica*, **23**, 137-148.

Cutler, C. R., and Ramaker, B. L. (1979). Dynamic matrix control - a computer control algorithm. AIChE National Meeting, Houston.

Doyle, F. J. III, 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. extension to nonlinear-systems. *Ind. Eng. Chem. Process Des. Dev.*, **25**, 403-411.

Ebihara, Y., Hagiwara, T., and Araki, M. (2000). Sequential tuning methods of LQ/LQI controllers for multivariable systems and their application to hot strip mills. *International Journal of Control*, **73**, 1392-1404.

Fischer, M., Nelles, O., and Isermann, R. (1998). Predictive control based on local linear fuzzy models. *International Journal of Systems Science*, **29**, 679-697.

Fujii, T. (1987). A new approach to the LQ design from the viewpoint of the inverse regulator problem. *IEEE Transactions on Automatic Control*, **32**, 995-1004.

Garcia, C. E., and Morari, M. (1982). Internal model control. 1. a unifying review and some new results. *Industrial and Engineering Chemistry Process Design and Development*, **21**, 308-323.

Garcia, C. E., and Morshedi, B. L. (1986). Quadratic programming solution of dynamic matrix control (QDMC). *Chemical Engineering Communications*, **46**, 73-87.

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

Guardabassi, G. O., and Savaresi, S. M. (1997). Approximate feedback linearization of discrete-time non-linear systems using virtual input direct design. *Systems and Control Letters*, **32**, 63-74.

Guardabassi, G. O., and Savaresi, S. M. (2000). Virtual reference direct design method: an off-line approach to data-based control system design. *IEEE Transactions on Automatic Control*, **45**, 954-959.

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

Hashimoto, Y., Matsumoto, K., Matsuo, T., Hamaguchi, T., Yoneya, A., and Togari, Y. (2000). Anti-saturation discrete-time nonlinear controllers and observers. *Computers and Chemical Engineering*, **24**, 815-821.

Henson, M. A. (1998). Nonlinear model predictive control: current status and future directions. *Computers and Chemical Engineering*, **23**, 187-202.

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.

Hjalmarsson, H., Gunnarsson, S., and Gevers, M. (1994). A convergent iterative restricted complexity control design scheme. *Proceedings of the 33rd IEEE Conference on Decision and Control*, 1735-1740.

Hjalmarsson, H., Gevers, M., Gunnarsson, S., and Lequin, O. (1998). Iterative feedback tuning: theory and applications. *IEEE Control Systems Magazine*,**18**, 26-41.

Ho, H. L., Rad, A. B., Chan, C. C., and Wong, Y. K. (1999). Comparative studies of three adaptive controllers. *ISA Transactions*, **38**, 43-53.

Hunt, K. J., and Sbarbaro, D. (1991). Neural networks for nonlinear internal model

control. *IEE Proceedings Part D-Control Theory and Applications*, **138**, 431-438.

Ikeda, M., and Suda, N. (1988) Synthesis of optimal servo systems. *Transaction of the Society of Instrument and Control Engineers*, **24**, 40-46.

Jin, L., Nikiforuk, P., 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.

Kapoor, N., and Daoutidis, P. (1999). An observer-based anti-windup scheme of non-linear systems with input constraints. *International Journal of Control*, **72**, 18-29.

Kavsek-Biasizzo, K., Skrjac, I., and Matko, D. (1997). Fuzzy predictive control of highly nonlinear pH process. *Computers and Chimical Engineering*, **21**, S613-S618.

Kendi, T. A., and Doyle, F. J. III (1997). An anti-windup scheme for multivariable nonlinear systems. *Journal of Process Control*, **7**, 329-343.

Kothare, M. V., Campo, P. J., Morari, M. and Nett, C. N. (1994). A unified framework for the study of anti-windup designs. *Automatica*, **30**, 1869-1883.

Kuo, B. C. (1980). Digital control systems. Holt, Rinehart and Winston, New York.

Lee, J. H. (2000). Modeling and identification for nonlinear model predictive control: requirements, current status and future research needs. *Nonlinear Model Predictive Control (Progress in Systems and Control Theory)*, **26**, 270-293.

Lewis, F. L. and Syrmos, V. L. (1995) Optimal control, John Wiley, New York.

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.

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

Lu, C. H., and Tsai, C. C. (2007). Generalized predictive control using recurrent fuzzy neural networks for industrial processes. *Journal of Process Control*, **17**, 83-92.

Mahfouf, M., Linkens, D. A., and Abbod, M. F. (2000). Adaptive fuzzy TSK model-based predictive control using a CARIMA model structure. *Chemical Engineering Research and Design*, **78**, 590-596.

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.

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

Mayne, D. (2000). Nonlinear model predictive control: challenges and oppotunities. *Nonlinear Model Predictive Control (Progress in Systems and Control Theory)*, **26**,

23-44.

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.

Nakamoto, M. (2005). An application of the virtual reference feedback tuning method for a multivariable process control. *Transactions of the Society of Instrument and Control Engineers*, **41**.

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

Nelles, O. (2001). Nonlinear system identification : from classical approaches to neural networks and fuzzy models. Springer, Berlin.

Ogata, K. (1997). Modern control engineering. Pretice Hall, NJ.

Pan, T., Li, S., and Cai, W. J. (2007). Lazy learning-based online identification and adaptive PID control: a case study for CSTR process. *Industrial and Engineering Chemistry Research*, **46**, 472-480.

Pearson, R. K. (1999). Discrete-time dynamic models. Oxford University Press, Oxford.

Piche, S. B., Sayyar-Rodsari, B., Johnson, D., and Gerules, M. (2000). Nonlinear model predictive control using neural networks. *IEEE Control Systems Magazine*, **20**, 53-62.

Prasad, G., Swidenbank, E., and Hogg, B. W. (1998). A local model networks based multivariable long-range predictive control strategy for thermal power plants. *Automatica*, **34**, 1185-1204.

Pottmann, M., and Seborg, D. E. (1997). A nonlinear predictive control strategy based on radial basis function models. *Computers and Chemical Engineering*, **21**, 965-980.

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

Riverol, C., and Napolitano, V. (2000). Use of neural networks as a tuning method for an adaptive PID application in a heat exchanger. *Chemical Engineering Research and Design*, **78**, 1115-1119.

Saint-Donat, J., Bhat, N., and McAvoy, T. J. (1991). Neural net based model predictive control. *International Journal of Control*, **54**, 1453-1468.

Savaresi, S. M., and Guardabassi, G. O. (1998). Approximate I/O feedback linearization of discrete-time non-linear systems via virtual input direct design. *Auto-*

matica, **34**, 715-722.

Seborg, D. E., Edgar, T. F., and Mellichamp, D. A. (1989). Process dynamics and control. John Wiley, New York.

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

Seki, H., Ogawa, M., Ooyama, S., Akamatsu, K., Oshima, M., and Yang, W. (2001). Industrial application of a nonlinear model predictive control to polymerization reactors. *Control Engineering Practice*, **9**, 819-828.

Shahrokhi, M., and Baghmisheh, G. R. (2005). Modeling, simulation and control of a methanol synthesis fixed-bed reactor. *Chemical Engineering Science*, **60**, 4275-4286.

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

Spall, J. C., and Cristion, J. A. (1998). Model-free control of nonlinear stochastic systems with discrete-time measurements. *IEEE Transactions on Automatic Control*, **43**, 1198-1210.

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

Tan, S., Hang, C. C., and Chai, J. S. (1997). Gain scheduling: from conventional

to neuro-fuzzy. *Automatica*, **33**, 411-419.

Venkateswarlu, C. H., and Gangiah, K. (1997). Constrained generalized predictive control of unstable nonlinear processes. *Chemical Engineering Research and Design*, **75**, 371-376.

Xiong, Q., and Jutan, A. (2002). Grey-box modeling and control of chemical processes. *Chemical Engineering Science*, **57**, 1027-1039.

Yamamoto, T., and Hinamoto. T. (2004). A design of memory-based PID controllers. *Proceedings of the 5^{t}h Asian Control Conference*, 497-505.

Yamamoto, T., and Shah, S. L. (2004). Design and experimental evaluation of a multivariable self-tuning PID controller. *IEE Proceeding - Control Theory and Applications*, **151**, 645-652.

Zheng, A. (2000). Some prectical issues and possible solutions for nonlinear model predictive control. *Nonlinear Model Predictive Control (Progress in Systems and Control Theory)*, **26**, 129-143.

Publications and Presentations

Kansha, Y., Hashimoto, Y., and Chiu, M. S. (2006). Data-based LQI controller design from plant data. *Journal of Chemical Engineering of Japan*, **39**, 746-752.

Kansha, Y., Jia, L., and Chiu, M. S. (2008). Self-tuning PID controllers based on the Lyapunov Approach. *Chemical Engineering Science*, accepted.

Kansha, Y., Hashimoto, Y., and Chiu, M. S. (2008). New results on VRFT design of PID controller. *Chemical Engineering Research and Design*, accepted.

Kansha, Y., Jia, L., and Chiu, M. S. (2007). Internal model controller design using linear multiple models. *Industrial and Engineering Chemistry Research*, revision submitted.

Kansha, Y., and Chiu, M. S. (2007). Generalized predictive control based on multiple models. *Journal of Process Control*, revision submitted.

Kansha, Y., Hashimoto, Y., and Chiu, M. S. (2005). LQI controller design from plant data. *PSE ASIA, The* 3^{rd} *International Symposium on Design, Operation and Control of Chemical Engineering*, Seoul, Korea, 18-19 August.

Kansha, Y., Hashimoto, Y., and Chiu, M. S. (2005). Adaptive controller design by virtual reference feedback tuning. *AdCONIP'05, The* 2^{nd} *International Symposium on Advanced Control of Industrial Processes*, Seoul, Korea, 22-23 August.

Kansha, Y. (2005). An integrated method for internal model controller design. *The*

2^{nd} *Annual Graduate Student Symposium, Department of Chemical and Biomolecular Engineering, National University of Singapore*, Singapore, 6 October.

Kansha, Y., Jia, L., and Chiu, M. S. (2007). Adaptive IMC design using multiple models. *The* 6^{th} *IEEE International Conference on Control and Automation*, Guangzhou, China, 31 May -1 June.