STUDY ON ADAPTIVE CONTROL OF NONLINEAR DYNAMICAL SYSTEMS BASED ON QUASI-ARX MODELS

WANG, Lan

Graduate School of Information, Production and Systems
Waseda University

September, 2011

Abstract

Adaptive control is a very important field of system control and has attracted a lot of interest from researchers in recent years. Linear system theory is very developed and there exist many excellent adaptive control results for linear systems. On the other hand, most of real plants are nonlinear and linear approximative models can not do well in the accuracy problem of these plants. Therefore, many nonlinear black-box models (neural networks, wavelet networks, adaptive fuzzy systems, etc.) have been used to control of nonlinear systems. However, there are two problems for these nonlinear models: the controller designing and the stability of corresponding control system. The controllers based on these nonlinear models are more difficult to be obtained than based on the linear models. Stability and accuracy of the control system for nonlinear systems are difficult to be ensured in one method or one nonlinear model.

A quasi-linear black-box modeling scheme has been proposed with which the techniques based on well developed linear system theory could be extended to nonlinear systems. It constructs models consisting of two parts: a macro-part and a kernel-part. The macro-part is a user-friendly interface constructed using the specific knowledge and the characteristics of network structure; the efforts of this part are to introduce some properties favorable to certain applications, such as controller designing. In this thesis, AutoRegressive eXogenous (ARX) model structure is chosen as macro-part because of various useful linearity properties. This macro structure makes the proposed controller easily get and use like based on ARX model. The kernel-part is a nonlinear black-box model which is used to represent the complicated coefficients of macro-parts. In this thesis, neural networks, radial basis function networks, and neural fuzzy networks are chosen as the kernel-parts which improve the control accuracy. Obviously, the above modeling scheme can construct different macro-parts and kernel-parts with applying specific knowledge for different application interests. However, the stability is still a problem which must be solved if the controllers based on the quasi-linear black-box modeling scheme want to be used in the real world.

The motivation of this thesis is intended to research on adaptive control of nonlinear dynamical systems based on the quasi-ARX black box models. According to the quasi-ARX modeling scheme, several improved quasi-ARX black-box models are proposed for different nonlinear control requirement. The obtained quasi-ARX black-box model is considered to have two properties: the linear property and the nonlinear property. Based on the model characteristics, two controllers can be obtained: one linear controller and one nonlinear controller. The linear controller is used to ensure the control stability and the nonlinear controller is utilized to improve the control accuracy. A switching mechanism is proposed between the two controllers. In the premise of stability, the switching

mechanism will tend to choose the nonlinear controller for the accuracy. On the other hand, the switching mechanism will return to linear controller to ensure stability when the stability of control system is destroyed. Therefore, the stability and accuracy problems in adaptive control process are solved by one model following the quasi-ARX modeling scheme. Investigations are made to do system identification, control design for nonlinear systems and stability analysis of control system under the framework of linear control theory based on the new modeling scheme.

A quasi-ARX neural network (NN) following the quasi-linear black-box modeling scheme is constructed and its application for stability adaptive control of nonlinear systems is proposed. The obtained quasi-ARX NN model is divided into two parts: the linear part is used to ensure the nonlinear control stability, and the nonlinear part is utilized to improve the control accuracy. One linear controller is obtained based on the linear part and one nonlinear controller is given based on the quasi-ARX NN model. In order to combine both the stability and universal approximation capability, a 0/1 switching law is established in our proposed control system by a switching criterion function based on system input-output variables and prediction errors. An adaptive controller is designed for nonlinear dynamical systems based on the obtained quasi-ARX NN model and the proposed switching mechanism, and its stability is analyzed. It is obviously the stability of adaptive control system is proved in theory, and the accuracy of the proposed control method is higher than linear method through the simulations. Therefore, the proposed controller is friendly interface, stability, higher accuracy and adaptive.

Nevertheless, there are still some aspects needed to be improved in the above control method. One is that the 0/1 hard switching method is not very smooth; the second is the assumption of global boundedness also can be relaxed; the third is that the parameters of quasi-ARX NN model to be adjusted on-line are highly nonlinear, which deteriorates the adaptability of control system. Motivated by the above aspects, three improvements are given to the quasi-ARX model. Firstly, a fuzzy switching mechanism is constructed based on the system switching criterion function which is better than the 0/1 switching law. Secondly, a d-difference operator is used in the ARX-like expression of system to relax the assumption of global boundedness on higher-order nonlinear terms. At finally, Radial Basis Function Network (RBFN) is used to replace the NN in the quasi-ARX black-box model which is understandable in terms of parameters and is not a absolute black-box model, compared with NN. The simulation includes two parts: the fuzzy switching control results based on quasi-ARX NN model and the fuzzy switching control results based on quasi-ARX RBFN model and d-difference operator. The simulation results show that the proposed control model and method based on the three improvements have better control performance.

In real world, a lot of systems are MIMO with complicated coupling. Due to the difficulty of decoupling problem, most of the control techniques developed for SISO systems cannot be extended directly for MIMO systems. It is also a change for control system based on quasi-ARX black-box model. Therefore, a MIMO quasi-ARX black-box model is proposed in this thesis and improves the quasi-ARX model which can be used as the predictor of MIMO nonlinear systems. The adaptive multivariable PID controller with a decoupling compensator and a feed-forward compensator is presented for the control of nonlinear MIMO systems using the proposed MIMO quasi-ARX RBFN prediction model. The parameters of such controller are selected based on the generalized minimum control variance. In this chapter, the corresponding stability analysis is given. The proposed control method can satisfy accuracy, stability and decoupling requirements for MIMO nonlinear systems.

When NFN is used as kernel-part, variables and the order of the model increases, the complexity of input-output designing the NFN also increases. In order to resolve this problem in the identification process, a Nonlinear Principal Components Analysis (NPCA) network trained by Artificial Neural Network (ANN) is introduced in quasi-ARX Neuro-Fuzzy Network (NFN) model, instead of PCA network when the input variables of NFN are nonlinear correlation. Because the output of NPCA network is used as the input of quasi-ARX NFN model, then the number of input is reduced. The control method is given based on the improved quasi-ARX NFN model with NPCA. This method reduces the number of controller parameters and improves the control performance of the controller based on the quasi-ARX modeling.

Preface

The common theme of this thesis is studying on a quasi-linear models, especially corresponding controllers, their applications to adaptive control problems and their stability problem. The material is organized in six chapters. Most of the material has been published or considered to publish in journal papers and conference papers.

The material in Chapter 2 can be found in

- Lan Wang, Yu Cheng and Jinglu Hu, "Adaptive Control for Nonlinear Systems Based on Quasi-ARX Neural Network", *in Proc. of World Congress on Nature and Biologically Inspired Computing (NaBIC 2009)* , pp.1548–1551, Coimbatore, India, 2009.

- Lan Wang, Yu Cheng and Jinglu Hu, "Quasi-ARX neural network and its application to adaptive control of nonlinear systems", *in Proc. of 15th International Symposium on Artificial Life and Robotics (AROB 15th'10)*, pp.577–580, Bepu, Janpan, 2010.

The material in Chapter 2 has been extended into a journal paper

- Lan Wang, Yu Cheng and Jinglu Hu, "A Quasi-ARX Neural Network with Switching Mechanism to Adaptive Control of Nonlinear Systems", *SICE Journal of Control, Measurement, and System Integration*, Vol.3, No.4, pp.246–252, 2010.

The materials in Chapter 3 can be found in

- Lan Wang, Yu Cheng and Jinglu Hu, "Nonlinear Adaptive Control Using a Fuzzy Switching Mechanism Based on Improved Quasi-ARX Neural Network", *in Proc. of The 2010 International Joint Conference on Neural Networks (IJCNN2010)*, pp. 1–7, Barcelona, Spain, 2010.

- Lan Wang, Yu Cheng and Jinglu Hu, "Adaptive Switching Control Based on Quasi-ARX RBFN Model", *in Proc. of 2011 International Conference on Computers, Communications, Control and Automation (CCCA'2011)*, pp. 76–79, Hongkong, China, 2011.

which has been extended into a journal paper

- Lan Wang, Yu Cheng and Jinglu Hu, "Stabilizing Switching Adaptive Control for Nonlinear System Based on Quasi-ARX RBFN Model", *IEEJ Transactions on Electrical and Electronic Engineering (TEEE) (in press)*, Vol.7, No.4, 2012.

The material in Chapter 4 can be found in

- Lan Wang, Yu Cheng and Jinglu Hu, "Multivariable Self-Tuning Control for Nonlinear MIMO System Using Quasi-ARX RBFN Model", *in Proc. of The 30th Chinese Control Conference (CCC'2011)* pp. 3772–3776, Yantai, China, 2011.

which has been extended into a journal paper

- Lan Wang, Yu Cheng and Jinglu Hu, "A Quasi-ARX Model for Multivariable Decoupling Control of Nonlinear MIMO System", submitted to *Mathematical Problems in Engineering*, Accepted (Made On 2011-08-17).

The material in Chapter 5 has been presented in

- Lan Wang, Yu Cheng and Jinglu Hu, "An Improvement of Quasi-ARX Predictor to Control of Nonlinear Systems Using Nonlinear PCA Network", *in Proc. of ICROS-SICE International Joint Conference 2009*, pp. 5095–5099, Fukuoka, Japan, 2009.

- Lan Wang, Yu Cheng and Jinglu Hu, "Nonlinear Adaptive Control Using Support Vector Regression Based on Improved Quasi-ARX Model", *in Proc. of 2010 International Conference on Modeling, Simulation and Control (ICMSC'10),* pp.412–416, Cairo, Egypt, 2010.

Acknowledgements

In three years, at Waseda university, as a doctoral student, I have met many difficulties in study and life. However, at finally, I overcome them through my efforts and much support and encouragement from many people. At the moment, I would like to take the opportunity to thank the following people.

First, I would like to thank my supervisor, Professor Jinglu Hu whose support and encouragement helped me in every moment of the three years. Every bit of my achievement can't be got without his wholeheartedly teach. His serious attitude and rigorous academic style are wealths for my study and work, in the past, now and the future.

I am very grateful to Professor Kotaro Hirasawa and Professor Osamu Yoshie who proofread all the thesis. Many thanks are for their valuable comments.

During my three years study, I have got many constructive suggestions and warm-hearted care from my classmates in at Waseda University and Southeast University. I am very grateful to them who are Boyang Li, Jun Zhang, Yang Chen, Benhui Chen, Wenxiang Dou, Qiangwei Wang, Yu Cheng, Bo Zhou, Zhan Shi, Yuling Lin, Xiaoying Ma, Jing Zeng, Jinling Liang, Wenjun Xing, Fengli Ren, Wenwu Yu, Yonghu Sun, Ping Li, Wangli He, Gan Huang, et al and many other students and faculty, too numerous to name.

I would also like to express my thanks to my research group (Yu Cheng, Feng Zhang, Miao Yu, Jing Zeng). I have learned so much from all of you, from figuring out what research is, to choosing a research agenda, to learning how to present my work. Your constructive criticism and collaboration have been tremendous assets throughout my Ph.D.

Finally, I would like to dedicate this work to my parents, Zimin Wang and Feng Jin, my husband Yiping Dong, my sister Dan Wang and my brother Xu Wang. Without your unending support and encouragement, I would not have finished the degree. Thank you.

Kitakyushu, Japan Lan WANG
June 20, 2011

Table of Contents

List of Tables

List of Figures

Glossary

Some notations may have different meaning locally.

Notations

x^T	transpose
I	identity Matrix
$y(t), y^*(t), \hat{y}(t)$	output signal, reference output and prediction output at time t
$u(t)$	input signal at time t
$\boldsymbol{y}(t)$	output vector at time t
$\boldsymbol{u}(t)$	input vector at time t
$\varepsilon(t), \mathbf{e}(t)$	prediction error at time t
$e(t), v(t), \nu_t$	noise, system disturbance
$\varphi(t)$	regression vector
$\tilde{\varphi}(t)$	regression vector $\varphi(t)$ whose element $u(t-i)$ changed to $q^{-1}u(t-i)$
$J(t)$	switching criterion function
ξ	parameter vector including the parameters describing unmodeled dynamics
Θ_t	coefficient vector which is function of input-output variables
$\hat{\theta}$	estimate of θ
$\boldsymbol{p}_j$	scale and position parameter vector of the 'basis functions' in nonlinear nonparametric model
ω_{ij}	coordinate parameters of nonlinear nonparametric model
n	number of old output values in $\varphi(t)$
m	number of old input values in $\varphi(t)$
r	number of old input and output values in $\varphi(t)$, $r = n + m$
M	number of 'basis functions' in nonlinear nonparametric model, in particular for adaptive fuzzy systems, the number of rules
λ	weighting factor for control input in control law
W^1, W^2	the weight matrices of the first and second layers
B, B^1, B^2	the bias vector of hidden nodes
μ_t	a fuzzy switching law based on the criterion function
D	the bound of nonlinear term
$\mathcal{C}$	controller

Operators and Functions

$\|\cdot\|$	norm
Δ	difference operator
$\arg\min f(x)$	minimizing argument of $f(x)$
$\dim(\theta)$	dimension of the vector θ
q^{-1}	the backward shift operator, $q^{-1}f(t) = f(t-1)$
$\wedge$	minimum operator, $0.8 \wedge 0.3 = 0.3$
$\otimes$	Kronecker production
$\prod$	production
sum	summation
$\mu_{A_i^j}(x)$	fuzzy membership function of fuzzy set A_i^j
$\mathcal{N}_f(x)$	'basis function' in the nonlinear nonparametric model
$G(q^{-1})$	rational function in q^{-1} describing system dynamics
$\chi(t)$	0/1 switching function
$\Gamma(\cdot)$	the diagonal nonlinear operator with identical sigmoid elements σ (for example: $\sigma(x) = \frac{1-e^{-x}}{1+e^{-x}}$)
$\mathbf{M}(t)$	the criterion function of a minimum variance control
$\varsigma(\cdot)$	the nonlinear difference term of nonlinear system

Abbreviations

AP	Affinity Propagation
ARMAX	AutoregRressive Moving Average model structure with eXogenous inputs
ARX	AutoRegressive model structure with eXogenous input
BIBO	Bounded-Input Bounded-Output
GSR	Gain Scheduling Regulator
LS	Least Square
MA	Moving Average model structure
MIMO	Multiple-Input Multiple-Output
MRAC	Model Reference Adaptive Controller
MSE	Mean Square Error
NFN	Neuro-Fuzzy Network
NN	Neural Network
NNM	Nonlinear Nonparametric Model
NPCA	Nonlinear Principal Components Analysis
PEM	Prediction Error Method
PCA	Principal Components Analysis
PID	Proportion Integration Differentiation
RBFN	Radial Basis Function Network
RMSE	Root Mean Spare Error
SISO	Single-Input and Single-Output
STR	Self-Tuning Regulator
SVM	Support Vector Machine
SVR	Support Vector Regression
WN	Wavelet Network

Chapter 1

Introduction and Motivation

1.1 Systems

A system is an object in which variables of different kinds interact and produce observable signals, in loose terms[1]. Our interesting observable signals are output, and external signal which can be manipulated are called input. The systems can be divided into linear and nonlinear systems by the relation between input and output signals. Thanks to the simple frameworks and properties of linear systems, they have been found in much real application and researched in system identification control theory and signal processing[2, 3]. However, majority systems are nonlinear whose output is not directly proportional to their input. The study of nonlinear systems have attracted much attention from all fields of sciences and humanities. Because they have been everywhere in the real world, such as food-webs, ecosystems, metabolic pathways and also include systems which are founded and used by human, such as robot, aeronautical satellite, unpiloted avion, industrialized machine and electric arc furnace. A part of nonlinear systems can be considered of an approximation or combination of multiple linear systems[4, 5, 6]. Therefore, confronted with a kind of nonlinear systems problem, it is indeed a happy circumstance when a solution can be obtained by linearizing.

The systems also can be divided into single-variant and multi-variant systems by the input/output number of systems. A Single-Input and Single-Output (SISO) system is typically simpler than Multiple-Input Multiple-Output (MIMO) systems which is shown in Fig.1.1, where $u(t)$ is input variable and $y(t)$ is output variable. The theory research on SISO systems has been started since 1960s, and many significant results have been obtained[7, 8, 9, 10, 11]. Systems which have more than one input and more than one output are known as MIMO systems [12, 13]. Then, the vectors $\boldsymbol{u}(t)$ and $\boldsymbol{y}(t)$ are used to represent multiple inputs and multiple outputs with the desired number. As we know, MIMO systems usually have a complicated dynamical coupling behavior which are not

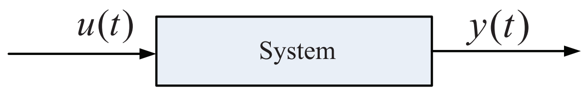

Figure 1.1: Schematic diagram of the SISO system.

several SISO systems side by side. Hence the traditional study on SISO systems can not directly to implement on complicated MIMO systems.

1.2 System Identification

System identification is the theory of how mathematical models for dynamical systems are constructed from observed data[14]. Prior knowledge or assumptions about the systems which generate the observed data guide the choice of model structure. It is general to distinguish under three levels of prior knowledge, which have been given as follows [15]

- White Box models: This means that a system is perfectly known; it is possible to construct the model entirely from prior knowledge and physical insight.

- Grey Box models: This means that some physical insight is available, but several parameters still need to be determined from observed data.

- Black Box models: This means that no physical insight is available or used, but the model structure is chosen from families which have good flexibility and have been “successful in the past”.

1.2.1 Black-Box Modeling

A black box model is chosen when little prior knowledge is available and is a standard flexible structure which can be used to approximate a lot of different systems. In order to describe the system exactly, some reasonable assumptions about system is made. One common assumption is that the unknown system is linear which is very useful for many problems but this is never true in real applications. Linear system theory is very well developed and there are many results which can be applied to the obtained linear models.

However, the linear assumption is strict for real world which has many nonlinear systems. In recent years, nonlinear modeling and identification have attracted much attention from control and system identification fields. Many nonlinear models have been proposed in the literatures: ‘classic’ models derived from Volterra series or Winner series [16, 17], and nonlinear black-box models based on the nonlinear nonparametric models (NNMs) such as Neural Networks (NNs)[18, 19, 20], Wavelet Networks (WNs) [21, 22], Neuro-fuzzy Networks (NFNs)[23, 24] and Radial Basis Function Networks (RBFNs) [25, 26]. We can see that the nonlinear black box models is very paid to the flexibility of the model structures. The structural linearity and simplicity, which are very important and useful features have been ignored. That is, in the literature, some authors have used a "linear model + NN" type hybrid scheme to identify and control nonlinear system [27, 28, 29]. However its linear structures and nonlinear structures are combined in a less effective and efficient way. Recently, a hybrid quasi-linear black-box modeling scheme is given by incorporating a group of certain NNMs into a linear structure[14]. The basic idea of such hybrid method is first to increase the overall model flexibility by using NNMs and then to restrict the flexibility in the higher order nonlinearity which can be to achieve the model simplicity [14].

It has been shown that a general nonlinear system can be expressed by a linear model whose coefficients consist of constant parameters and nonlinear terms. In this model, a group of NNMs are incorporated into the linear structure to represent the nonlinear terms. Since NNMs in the hybrid structure is only one nonlinear term of the coefficients, the requirement of each NNM is reduced and the flexibility of individual NNM also can be restricted to some extent. Therefore, some parameters of NNMs can be determined by using *a priori* knowledge. The efficient use of various *a priori* knowledge information will play an important role on the hybrid modeling. The model constructed in this way is named as quasi-linear black-box model shown in Fig.1.2, which has a linear structure, flexibility and simplicity [14].

The quasi-linear black-box model which consists two parts: a macro-model part and a kernel part was proposed in[24, 30]. ARX or ARMAX were used as the macro-model part which are a user-friendly interface constructed using already known knowledge and the characteristic of structure. The ordinary NN and NFN have been chosen as the kernel part which is used to parameterize the coefficients of macro-model, respectively. The identification results based on the quasi-linear black-box models for nonlinear systems have been got as in Refs.[31, 32, 33, 34].

1.3 Control technology and Control Theory

Control technology play an important role for the human progress during the 20th century. They bring much positive impact and scientific methodology to resolve many challenges in today's society. They also establish the theoretical basis to achieved automatization, and propose advanced

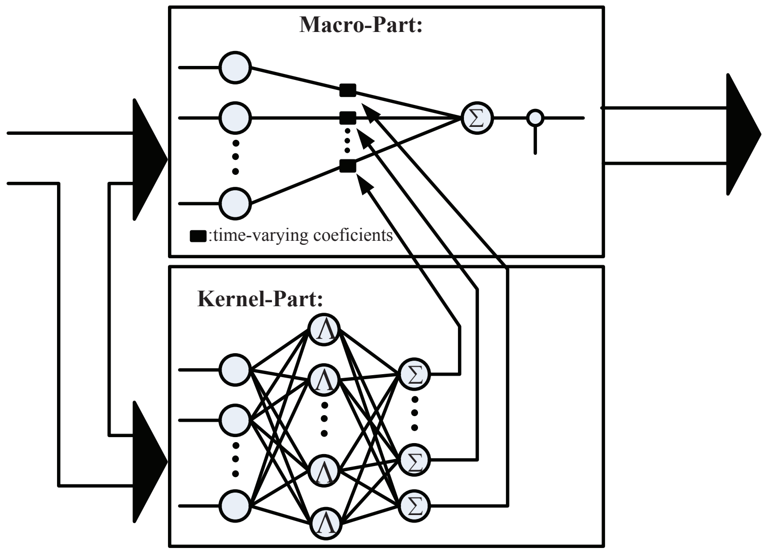

Figure 1.2: Hybrid quasi-linear black-box model.

control equipments and production technology for many industry fields. Especially, the widely used digital computer makes wider application field for control science and technology.

Control theory that deals with influencing the behavior of dynamical systems is an interdisciplinary subfield of science, which originated in mathematics and engineering, and evolved into use by the social sciences, like sociology, psychology, and criminology.

1.3.1 History and Development

The history and development of control theory has followed the control technical development to heel, and even is running far ahead of engineering practice in some fields. There four main phase for control under the different period as follows.

The first phase is Early Control. In this period, the development of control theory is based the invention and improvement of control technology. Early control systems of various types supported ancient civilizations, such as Clepsydra, Seismoscope, Jacquard loom and Speed Governor. J. Watt designed centrifugal governor to control the speed of an engine in 1788. Therefore, a more formal analysis just began with a dynamics analysis of the centrifugal governor which is conducted by the physicist J. C. Maxwell in 1868 [35]. Then, Maxwell's classmate E. J. Routh improved the analysis results of Maxwell to the general case of linear systems in 1875, which brought a flurry of interest in the field. In 1877, analyzed system stability using differential equations was analyzed by A. Hurwitz, resulting in what is known as the Routh-Hurwitz theorem. J. M. Gray designed the first

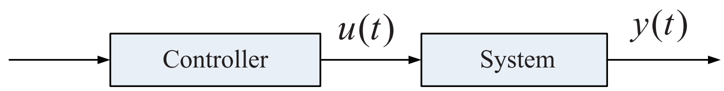

Figure 1.3: Open-loop control schematic diagram.

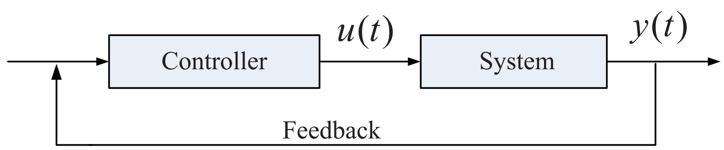

Figure 1.4: The structure of closed-loop control.

full automatic steamship in 1866. *The general problem of the stability of motion* was accomplished by A. M. Lyapunov as his doctoral thesis in 1892[36].

The second phase is The Pre-classical Period. Engineer N. Minorsky designed automatic steering systems for the US Navy, and published theoretical analysis of a Proportion Integration Differentiation (PID) controller in 1922. The open-loop control schematic diagram is given by Fig.1.3. The first widely practical version of the differential analyzer was constructed by H. L. Hazen and V. Bush at MIT, 1928–1931. The revolutionized Negative Feedback Amplifier was invented by electrical engineer H. S. Black in 1927[37]. The originator of cybernetics N. Wiener defined the notion of Feedback. The structure of closed-loop control is shown in Fig.1.4. Atmosphere pressure feedback control system was made by E. Sperry and C. Mason. The stability is the master problem, the differential equations with constant coefficients is mathematics tool and the control technology and control theory are developed synchronously in this period.

The third phase is Classical Control. The classical Frequency Response methods was developed by Nyquist and H. W. Bode[38]. In 1948, the book *Cybernetics* published by N. Wiener meant that the Cybernetics appeared. MIT radiation laboratory founded Nichols Chart Design method, and R. S. Philips introduced the effect of noise in servomechanisms. The Root Locus method was proposed by W. Evans in 1948. Thus, the classical control theory was finished which studied on signal-input linear system expressed by transfer function and based on the frequency method and Root Locus method. Many famous book were published in the period, such as E. D. Smith's *Automatic Control Engineer*, H. Bode's *Network Analysis and Feedback Amplifier* and X. Qian's *Engineering Cybernetics*. It was an important part of guidance systems, fire-control systems and electronics by World War II. The rapid development theory guides the industry developed at very fast speed in this process.

The forth phase is Modern Control. The world came in a peaceful development period. The control of nuclear reactor and aerospace is more complexity and requirement than the classical control object, this led to the development of multi-variable control systems[39]. Furthermore, since efficiency and optimality were paramount, Optimal Control method was proposed based on L. S. Pontryagin's Maximum Principle and R. Bellman's dynamic programming. R. E. Kalman introduced the state-space analysis systems, adaptive control system, controllability and so on, which is the theory foundation of modern control [40, 41]. The development of gigantic supercomputers offered the feasibility calculation. Although they could deal with the reactor and aerospace problem, those were limited to use in the generic industry because of the complexity and investment. Hence, many researcher still work on the frequency domain methods, in particular, N. H. Rosenbrock [42]. He transited multiple-variable system into several single-variable systems based on diagonal dominant. This method brought the revival of the frequency domain methods. In the 70s, the methods were appeared such as Sequence Return Difference method, Dyadic Expansions method and Characteristic Locus Design method, which were considered as modern frequency domain methods [43]. Their basal idea was to use the classical control method by transiting multiple-variable into several single-variable. In 1965, fuzzy set and fuzzy control was proposed by L. A. Zadeh [44]. And in 1967, K. J. Astrom proposed least squares identification which resolved linear system parameters identification problem. R. W. brockett used differential geometry to study nonlinear control in 1976 and A. Isidori published *Nonlinear Control Systems* in 1985. H_∞ robust control design was first given by G.Zames in 1981. Some theory such as nonlinear system control has been running far ahead of engineering practice.

1.3.2 Some Topics in control

Obviously, the stability of a general dynamical system is always main problem of the control theory research. The study of a general dynamical system described with Lyapunov stability criteria is just in theory. The overwhelming majority of obtained controller based on this study have never be used in practice. The bounded-input bounded-output (BIBO) stable for a linear system means that output will stay bounded for any bounded input. This theory has widely guided the controller design in real world. Therefore, stability for nonlinear systems that combines a notion similar to Lyapunov stability and BIBO stability have attracted much interest.

From the development of control, the linear control theory both as a branch of Engineering and as modern Applied Mathematics has been successfully established. Still, the vast majority of real systems is nonlinear. Although the nonlinear properties were dealt with by essentially patching together linear regimes, or linearize such classes of systems and applying linear techniques, in many cases it can be achieve the accuracy requirement of nonlinear system control. Some nonlinear control which directly use the NNM to design controller can never convenient usefulness for user.

In the real world, a lot of systems are MIMO with complicated coupling. Due to the difficulty of decoupling problem, most of the control techniques developed for SISO systems cannot be extended directly for MIMO systems by transiting multiple-variable into several single-variable. Then, multivariable decoupling control is also the topic popular research direction.

1.3.3 Adaptive Control

In the early 1950, it was found that ordinary constant-gain, linear feedback control could not work well in changed conditions. Therefore, adaptive control arisen for the requirement in connection with the design of autopilots for high performance aircraft [45]. Adaptive control is one control which involves modifying the control law to deal with the deed that the systems are slowly time-varying, disturbance or uncertain. In the 1960s, there were many contributions to control theory which were important for the development of adaptive control, such as state space, stability theory, stochastic control theory and dynamic programming. System identification and parameter estimation have also major developed. The stability of adaptive systems were correctly proved in the late 1970s and early 1980s, and it is possible to implement adaptive regulators simply and cheaply based on the rapid and revolutionary progress in microelectronics. Till now, a mass of development of the field is taking place, both on universities and industry [46, 47, 48, 13, 49, 50]. Adaptive control loops are widely used in aerospace, process control, ship steering, robotics and other industrial control systems. Therefore, it is no longer just an important theoretical subject of study, but is also providing solutions to real-world problems.

Types of adaptive control strategies mainly conclude Gain Scheduling Regulators (GSRs), Self-Tuning Regulators (STRs) and Model Reference Adaptive Controllers (MRAC, also know as an MRAS or Model Reference Adaptive System). GSR is a parameterized set of linear controllers which is one of the simplest and most intuitive forms of adaptive control. In operation the parameters are measured and the controller in action is scheduled according to the parameters.

STRs Control

The basic idea of STRs: it is assumed that the regulator parameters are adjusted all the time, in an adaptive system which is shown in Fig.1.5. $y^*(t)$ is the desired output. The main parameters estimation methods of STRs are gradient methods and least square method and control design methods

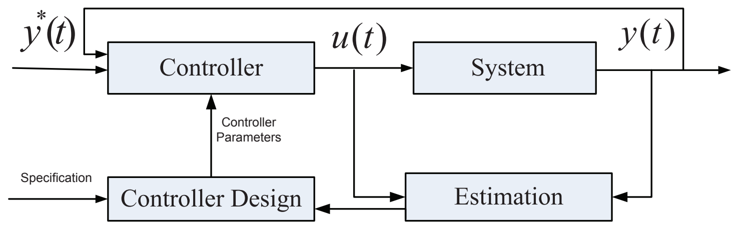

Figure 1.5: The self-tuning regulator principle.

are PID, pole-placement, LQG, predictive control, and so on. It was first given by Kalman in 1958. Until 1973, Åström and Wittenmark first proposed STRs in 1973 [51]. Before 1975, STRs controller are based minimum variance theory. The generalized self-tuning controller was developed by Clarke and Gawthrop [52] which resolved the main weakness of STRs. The pole-assignment STR algorithm based on the sub-optimal design was given which is better than above STRs except optimization by Edrounds in 1978. However, there are some problem when the systems have nonlinearity and serious uncertainty. Since 1980s, developed neural networks has shown potential ability to control the systems which are highly nonlinearity and serious uncertainty. Then, the research of STRs control based neural network has attracted much attention because of its approximate arbitrary, learning uncertain, highly robustness and parallel processing, and so on.

MRAC

It is one important category of feedback adaptive control as in Fig.1.6. The general idea of MRAC is to create a closed loop controller with parameters which can be updated to change the response of the system. Local parameter optimization method is the main idea to design the controllers from 1958-1966, which would lead to unstably. Therefore, Lyapunov stability theory was introduced in MRAC to resolve the stability problem by Butchart, Shachcloth, Park and Phillipson, from 1966 to 1972. But it is need that differentiation signals of all states or output. Augmented error signal method and Popov super stability theory have been used only based on input signal. However, it is difficult to that above methods need to direct get all system states. There two methods to use for the problem: direct method and indirect method. Since 1980, neural network model was introduced to MRAC.

There are many directions which adaptive control links with as in Fig 1.7. As we know, adaptive control are strong ties to nonlinear systems theory.

Adaptive controller offers certain advantages over conventional controller, When the systems to be controlled contain unknown parameters. Adaptive control theory has been developed into a

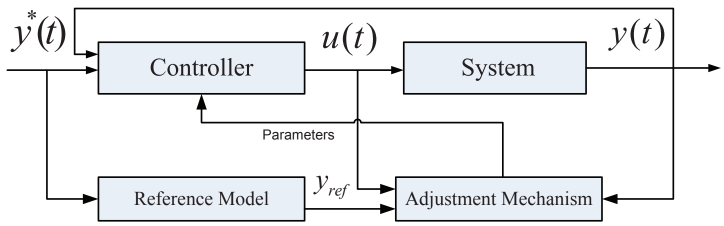

Figure 1.6: The schematic diagram of model reference adaptive control.

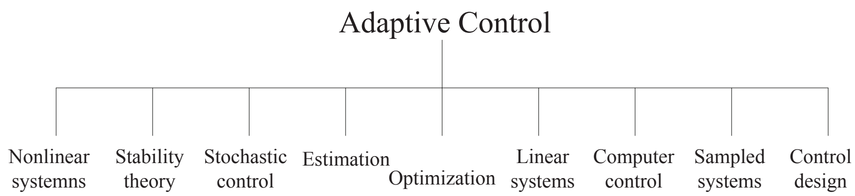

Figure 1.7: The directions link with adaptive control.

considerable mature stage based on linear models [2, 53]. However, it is difficult to control in the case of black-box type nonlinear systems. The difficulty is that a linear black-box model can not obtain enough accuracy, while a suitable nonlinear model is very difficult to find.

Hu *et al.*(1999) [30] proposes an adaptive predictor for general nonlinear systems based on the use of a class of NF models. The NF-based predictor can be interpreted as a linear predictor network consisting of a global linear predictor and several local linear predictors with interpolation. It has two distinctive features as well as good prediction ability: its parameters have explicit meaning useful for initial value setting in parameter adjustment; it may be transformed into a form linear for the variables synthesized in control system, which makes deriving a control law straightforward. Hu et al.(2004) [54] discusses quasi-ARX black-box model for the control of nonlinear systems. Contrast to a conventional method, the new method does not use NN directly as a nonlinear controller or nonlinear prediction model, but use it indirectly via an ARX-like macro-model. The ARX-like model incorporating NN is constructed in such a way that it has similar linear properties to linear ARX model. The nonlinear controller is then designed in a similar way as designing a controller based on a linear ARX model which is shown in Fig.1.8.

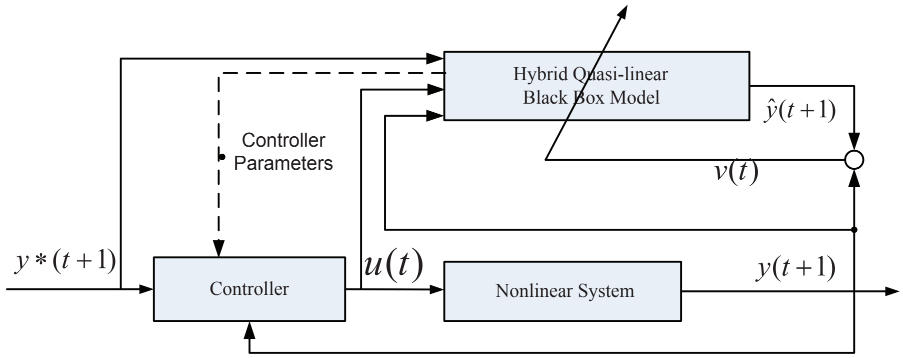

Figure 1.8: Controller based on the hybrid quasi-ARX black-box model.

1.4 Challenges

Stability and accuracy of control system are two important problems which have been resolved based on one model. When quasi-ARX black box models are used for nonlinear system control, several challenge must be faced:

- *Stability Problem*
  Stability problem must to be resolve if the control system want to be used in real world.

- *Accuracy Problem*
  The controller should have better accuracy in the stable premise.

- *Complicated plants*
  In fact, the controlled systems is more complicated such as unboundedness, Multi-Input and Multi-Output (MIMO).

- *Adaptive Control*
  The off-line control can not do well in the changing conditions. The proposed control law is needed that adapts itself to such changing conditions.

- *Identification problem*
  Identification problems include the choosing of model structure and parameter estimation. It is necessary step before controlling.

1.5 Goals of the Thesis

For complicated dynamical, the linear model cannot attain expectable control results and the classical nonlinear model cannot be stability. The challenges of the tasks will require novel modifications of existing control models and methods. In this thesis, the improved adaptive stability controller based on quasi-linear black-box model are developed and applied for nonlinear systems control. More precisely, quasi-linear black-box model have linear part for stability of control and nonlinear part for control performance. In order to combine both the stability and universal approximation capability in our controller, a switching mechanism is introduced. The parameters of nonlinear part can be determined by *a prior* knowledge. The identification process is also improved.

The work presented here aims to assess the performances of the proposed control system. The thesis also shows how the proposed control method handle the aforementioned challenges.

1.6 Thesis Outlines and Main Contributions

This thesis presents our work that has been done over the last three years. It consists of six chapters. Chapter 1 gives a background and an outline for the whole thesis. Chapter 2 introduces an improved quasi-ARX NN model and discusses its application to adaptive switching control of nonlinear systems. Chapter 3 obtains a stabilizing fuzzy switching controller for nonlinear system based on a quasi-ARX RBFN model, a fuzzy switching function and a d-difference operator. Chapter 4 proposes a MIMO quasi-ARX model, and a multivariable decoupling PID controller for MIMO nonlinear systems based on the proposed model. Chapter 5 improves the quasi-ARX model based NPCA network which resolve the dimension problem in identification process. Finally, Chapter 6 gives a summary for the whole thesis. The flow of this thesis is depicted in Figure 1.9.

This thesis summarizes the research on quasi-linear black-box models, especially corresponding controllers, their applications to adaptive control problems and their stability problem.

Chapter 2 introduces an improved quasi-ARX NN and discusses its application to adaptive control of nonlinear systems. A switching mechanism is employed to improve the performance of the controller based on the quasi-ARX NN prediction model which has linear and nonlinear parts. An adaptive controller for a nonlinear system is established based on the proposed prediction model and the switching law, and some stability analysis of the control system is shown.

The proposed adaptive control system is distinctive to other control systems in the following issues:

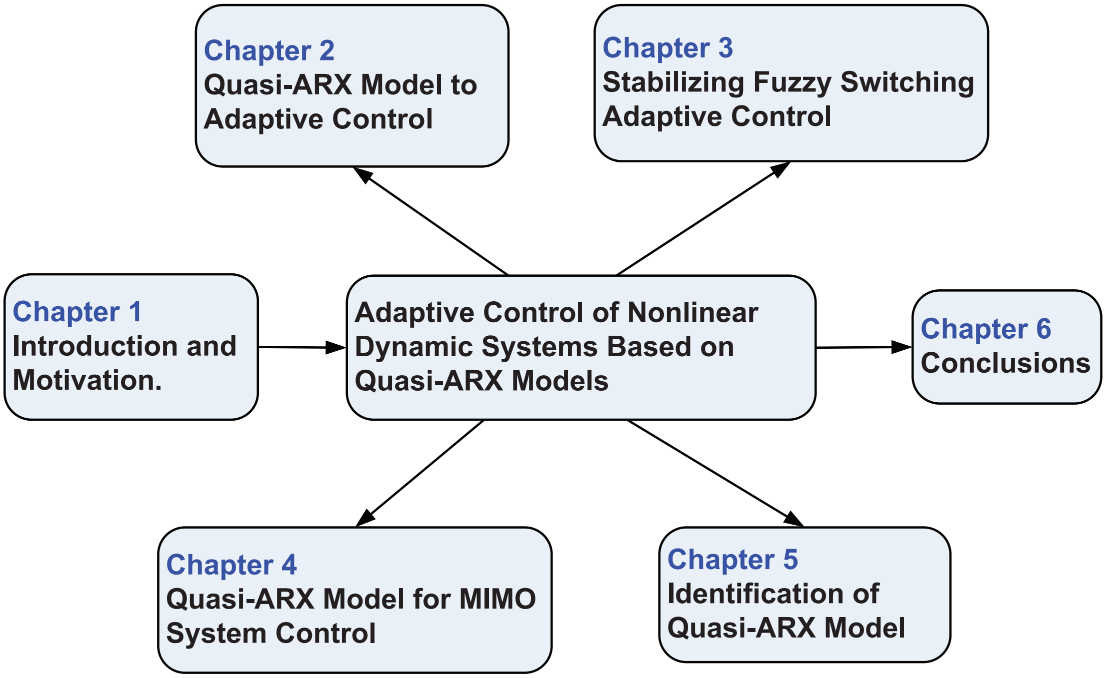

Figure 1.9: Flow diagram of this thesis.

- The proposed controller is linear for the variables synthesized in control systems
- The parameters of the proposed controller have explicit meanings
- The proposed control system is only one prediction model which combines a switching algorithm.

Chapter 3 explores a fuzzy switching adaptive control approach for nonlinear systems. The proposed fuzzy switching adaptive control law is composed of a quasi-ARX RBFN prediction model and a fuzzy switching mechanism. The quasi-ARX RBFN prediction model consists of two parts: the linear part used for a linear controller to assure boundedness of the input and output signals, and the RBFN nonlinear part used to improve the control accuracy. By using the fuzzy switching scheme between the linear and nonlinear controllers to replace the 0/1 switching, it can realize a better balance between stability and accuracy. Theory analysis and simulation results show the effectiveness of the proposed control method on stability, accuracy and robustness.

The contributions related to this fuzzy switching adaptive control are that:

- The proposed control system is linear for the variable synthesized, $u(t)$, including in the regression vectors $\psi(t)$ and $\Psi(t)$;
- The three predictors are obtained directly from only one identified quasi-ARX RBFN model, and all are linear for the control variable $u(t)$ to be synthesized in the control system;
- The nonlinear control system could have quick response since only linear parameters are adjusted on-line.
- The control system employs a fuzzy switching mechanism instead of a simple 0/1 switching.
- The control method of the previous control based the quasi-ARX model is off-line and doesn't give the stability analysis. The proposed control system is on-line and stability which is ensured by a fuzzy witching mechanism.

Chapter 4 introduces a MIMO quasi-ARX model and a multivariable decoupling PID controller for MIMO nonlinear systems based on the proposed model. The proposed MIMO quasi-ARX model improves the performance of ordinary quasi-ARX model. The proposed controller consists of a traditional PID controller with a decoupling compensator and a feed-forward compensator for the nonlinear dynamics from the MIMO quasi-ARX model. Then an adaptive control algorithm is presented using the MIMO quasi-ARX RBFN prediction model and some stability analysis of control system is shown.

The main contributions related to the MIMO quasi-ARX model and the nonlinear multivariable decoupling PID controller are that:

- The proposed method improve the quasi-ARX model to model the systems from SISO to MIMO which is more complex.
- The proposed method uses RBFNs as nonlinear models which are linear in parameters through fixing the nonlinear parameters by *a priori* knowledge. Incorporating the network models with this property, the quasi-ARX models become linear-in-parameters.
- The proposed adaptive control algorithm is a decoupling control algorithms which deals with coupling in nonlinear system based on linear methods and nonlinear networks.
- The proposed adaptive control algorithm based on the MIMO quasi-ARX RBFN prediction model is stability which is proved in this chapter.

Chapter 5 introduces a Nonlinear Principal Component Analysis (NPCA) to improve the identification of the quasi-ARX Neuro-fuzzy Networks (NFN) model. One part of the quasi-ARX model is the ordinary NFN to parameterize the coefficients which faces to a problem of high dimension. Because the controller shares the parameters with the quasi-ARX prediction model, then the complexity will lead to the huger parameters for controller designing. NPCA is used for this part to deal with this problem. The processes of modeling, parameter estimating and control are given in detail.

The main contributions related to this model are shown as follows:

- However, variables and the order of the model increases, the complexity of as the number of input-output designing the NFN also increases. A Principal Components Analysis (PCA) is introduced to reduce the dimension of the NFN.
- In fact, the input variables do not only depend on each other linearly. When nonlinear correlations between variables exist, a NPCA will describe the data with greater accuracy than PCA.

Chapter 6 concludes this work, summarizes the thesis and gives suggestions for further research.

Chapter 2

Adaptive Switching Control of Nonlinear Systems Based on Quasi-ARX Neural Network

2.1 Introduction

Adaptive control of complex nonlinear dynamical systems has attracted much attention and developed significantly during the last few decades. Many adaptive control methods have been proposed, and the corresponding stability and convergence have been proved [55, 56, 57, 10, 58, 59, 60, 61, 62, 63]. Neural networks have been used to identify and control nonlinear dynamical systems because of its ability to approximate arbitrary mapping to any desired accuracy [64, 65, 66, 67, 54, 22]. One of the successful examples is that neural networks are used directly to identify and control nonlinear systems [55, 66, 56, 48, 68].

However, from a user's point of view, there are three major criticisms on those neural network models. One is that their parameters do not have useful interpretations. The second is that they do not have a friendly interface for controller design and system analysis [24, 54, 69, 70]. The third one is that the result is local, i.e., the initial weights of a neural network have to be "close enough" to the true ones in order for the stability result to hold [71].

To solve these problems, a quasi-ARX neural network model has been proposed which embodied a macro-model part and a kernel part [54, 72]. The macro-model part is a user-friendly interface constructed using *a priori* knowledge [73] and the characteristic of network structure. In this chapter, we will limit our discussion to a quasi-ARX approach. The linear ARX model has a various useful linearity properties which will solve the former two problems. The kernel part is an ordinary neural network, which is used to parameterize the coefficients of macro-model and is different from a nonlinear ARX model based directly on neural networks. Because of the nonlinear characteristics, the quasi-ARX neural network can be used to identify and control nonlinear systems accurately. In our previous research, an off-line control scheme is given and the effectiveness of the quasi-ARX neural network is shown [54]. In the control system, the prediction model and controller share the same parameters as in linear cases. However, an adaptive controller has not been proposed for nonlinear systems control with the quasi-ARX model. What's more, the stability analysis is also lacked.

As we know, one of the successful approaches to solve the stability problem of neural network based control system is to use multiple models adaptive switched control [74, 75, 71, 60, 76, 77]. Therefore, those prediction and control systems have more than one model which adds the complexity of the control problem.

Motivated by the above discussion, an adaptive control law is proposed for nonlinear dynamical systems based on the characteristic of quasi-ARX neural network structure, and then the control system stability is proved. In this chapter, quasi-ARX neural network is divided into two parts: the linear part is used to ensure the nonlinear control stability, and the nonlinear part is utilized to improve the control accuracy. In order to combine both the stability and universal approximation capability in our controller, a switching law is established based on system input-output variables and prediction errors.

This chapter is organized as follows: In Section 2.2, the considered system is given. In Section 2.3, an improved quasi-ARX prediction model is introduced based on neural network and switching mechanism, then the parameters identification methods are given. Section 2.4 describes adaptive control using the improved quasi-ARX prediction model and analyzes the stability under the switching criterion function. Then, numerical simulations are carried out to show the effectiveness of the proposed model in Section 2.5. At last Section 2.6 gives some conclusions.

2.2 Problem Description

Consider a single-input-single-output (SISO) nonlinear time-invariant system whose input-output relation described by:

$$\begin{aligned} y(t) &= g(\varphi(t)) + v(t), \qquad (2.2.1)\\ \varphi(t) &= [y(t-1), ...y(t-n), u(t-d), ..., \\ &\qquad\qquad u(t-m-d+1)]^T \end{aligned}$$

where $y(t)$ denotes the output at time t ($t = 1, 2, ...$), $u(t)$ the input, d the known integer time delay, $\varphi(t)$ the regression vector, and n , m the system orders. $g(\cdot)$ is a smooth nonlinear function and $v(t)$ the system disturbance.

Now the following assumptions will be used:

Assumption 1: (i) $g(\cdot)$ is a continuous function, and at a small region around $\varphi(t) = 0$, it is C^∞ continuous;

(ii) there is a reasonable unknown controller which may be expressed by $u(t) = \tilde{\rho}(\tilde{\xi}(t))$, where $\tilde{\xi}(t) = [y(t) \; ... \; y(t-n) \; u(t-1) \; ... \; u(t-m) \; y^*(t+1) \; ... \; y^*(t+1-l)]^T$ ($y^*(t)$ denotes reference output);

(iii) the system has a globally uniformly asymptotically stable zero dynamics.

2.3 Quasi-ARX Neural Network

2.3.1 Regression Form Representation

A general nonlinear system described by (2.2.1) can be represented in a regression which has been shown in Ref.[24, 67].

Under **Assumption 1**(i), the unknown nonlinear function $g(\varphi(t))$ can be performed Taylor expansion in (2.2.1) on a small region around $\varphi(t) = 0$:

$$y(t) = g(0) + g'(0)\varphi(t) + \frac{1}{2}\varphi^T(t)g''(0)\varphi(t) + \; ... \; + v(t) \tag{2.3.1}$$

where the prime denotes differentiation with respect to $\varphi(t)$, then introducing the notations:

$$\begin{aligned} y_0 &= g(0) \\ \theta(\varphi(t)) &= \left(g'(0) + \frac{1}{2}\varphi^T(t)g''(0) + \cdots \right)^T \\ &= [a_{1,t} \; ... \; a_{n,t} \; b_{0,t} \; ... \; b_{m-1,t}]^T \end{aligned}$$

where the coefficients $a_{i,t} = a_i(\varphi(t))$ $(i = 1, .., n)$ and $b_{j,t} = b_j(\varphi(t))$ $(j = 0, ..., m-1)$ are nonlinear functions of $\varphi(t)$. A regression form of the system (2.2.1) is described by (2.3.2):

$$y(t) = y_0 + \varphi^T(t)\theta(\varphi(t)) + v(t). \tag{2.3.2}$$

However, $y(t)$ needs to be predicted using the input-output data available up to time $t - d$ in a prediction model. Considering this, we hope that the coefficients $a_{i,t}$ and $b_{j,t}$ are calculable using the input-output data up to time $t-d$. For this reason, replace iteratively $y(t-l),\ l=1,...,d-1$ in the expressions of $a_{i,t}$ and $b_{j,t}$ with their predictions:

$$y(t-l) \Rightarrow \hat{g}(\hat{\varphi}(t-l)),\ l=1,...,d-1 \tag{2.3.3}$$

where $\hat{g}(\cdot)$ is a predictor, $\hat{\varphi}(t-l)$ whose elements $y(t-k),\ l+1<k\leq d-1$ are replaced by their predictions, and define the new expressions of the coefficients by:

$$a_{i,t}=\tilde{a}_{i,t}=\tilde{a}_i(\phi(t-d)), \quad b_{i,t}=\tilde{b}_{i,t}=\tilde{b}_i(\phi(t-d))$$

where $\phi(t-d)=q^{-d}\phi(t)$ and $\phi(t)$ is a vector:

$$\phi(t)=[y(t)\,...\,y(t-n+1)\,u(t)\,...\,u(t-m-d+2)]^T. \tag{2.3.4}$$

And q^{-1} is a backward shift operator, e.g. $q^{-1}u(t)=u(t-1)$.

Now, two polynomials $A(q^{-1},\phi(t))$ and $B(q^{-1},\phi(t))$ based on the coefficients $a_{i,t}$ and $b_{j,t}$ is defined by:

$$\begin{aligned} A(q^{-1},\phi(t)) &= 1-a_{1,t}q^{-1}-\ldots-a_{n,t}q^{-n} \\ B(q^{-1},\phi(t)) &= b_{0,t}+\ldots+b_{m-1,t}q^{-m+1} \end{aligned}$$

A similar-linear ARX model is developed:

$$A(q^{-1},\phi(t))y(t)=y_0+B(q^{-1},\phi(t))q^{-d}u(t-1)+v(t). \tag{2.3.5}$$

For a system described by (2.3.5), a representation is given as in Ref.[54]:

$$y(t+d)=y_\phi+\alpha(q^{-1},\phi(t))y(t)+\beta(q^{-1},\phi(t))u(t) \tag{2.3.6}$$

where

$$\begin{aligned} y_\phi &= F(q^{-1},\phi(t))y_0, \\ \alpha(q^{-1},\phi(t)) &= G(q^{-1},\phi(t)))=\alpha_{0,t}+\alpha_{1,t}q^{-1}+...+\alpha_{n-1,t}q^{-n+1}; \\ \beta(q^{-1},\phi(t)) &= F(q^{-1},\phi(t))B(q^{-1},\phi(t))=\beta_{0,t}+\beta_{1,t}q^{-1}+...+\beta_{m+d-2,t}q^{-m-d+2}, \end{aligned}$$

and $G(q^{-1},\phi(t))$, $F(q^{-1},\phi(t))$ are unique polynomials satisfying:

$$F(q^{-1},\phi(t))A(q^{-1},\phi(t))=1-G(q^{-1},\phi(t))q^{-d}. \tag{2.3.7}$$

As we know, the linear ARX model is linear in the input variable $u(t)$, then an controller can be obtained easily and shares parameters from the model. However, the model (2.3.6) is a general one that is nonlinear in the variable $u(t)$, because the coefficients y_ϕ, $\alpha_{i,t}$ and $\beta_{j,t}$ are functions of $\phi(t)$ whose elements contain $u(t)$, where $i = 0, ..., n-1$ and $j = 0, ..., m+d-2$. To solve this problem, an *extra variable* $x(t)$ is introduced and replace the variable $u(t)$ in $\phi(t)$ with an unknown nonlinear function $\rho(\xi(t))$ where

$$\xi(t) \quad = \quad [y(t) \; ... \; y(t-n_1+1) \; x(t+d) \; ... \; x(t-n_3+d+1) \; u(t-1) \; ... \; u(t-n_2)]^T$$

including the extra variable $x(t+d)$ as an element. Under **Assumption 1**(ii), the function $\rho(\xi(t))$ is existent. Then we have a model expressed by:

$$y(t+d) = y_\xi + \alpha(q^{-1}, \xi(t))y(t) + \beta(q^{-1}, \xi(t))u(t) \tag{2.3.8}$$

where y_ξ is y_ϕ whose variable $u(t)$ is replaced by $\tilde{\rho}(\cdot)$.

As we know, the system model can be considered to have two parts. One part is linear on input and output variables and the model parameters is independent of $\xi(t)$. The other part is nonlinear on input and output variables which coefficients depend on $\xi(t)$. Define the new expressions of the coefficients by:

$$\alpha_{i,t} = \tilde{\alpha}_{i,t} = \tilde{\alpha}_{i,0} + \tilde{\alpha}_i(\xi(t)),$$

$$\beta_{j,t} = \tilde{\beta}_{j,t} = \tilde{\beta}_{i,0} + \tilde{\beta}_j(\xi(t)).$$

Moreover, we typically let $n_1 = n, \; n_2 = m+d-2, \; n_3 = 1$, which gets

$$\xi(t) = [y(t) \; ... \; y(t-n+1) \; x(t+d) \; u(t-1) \; ... \; u(t-d+2)].^T$$

As we know, in a control system, the extra variable $x(t+d)$ can be replaced with the reference signal $y^*(t+d)$. Introducing the following marks:

$$\Psi(t) = [1 \, y(t)...y(t-n+1) \, u(t)...u(t-m-d+2)]^T;$$

$$\Theta_\xi = [y_\xi \, \alpha_{0,t} \, ... \, \alpha_{n_y-1,t} \, \beta_{0,t} \, ... \, \beta_{n_u+d-2}]^T,$$

we get the improved ARX-like macro-model expression by:

$$y(t+d) = \Psi^T(t)\Theta_\xi. \tag{2.3.9}$$

The coefficients $\alpha_{i,t}$ ($i = 0, ..., n-1$) and $\beta_{j,t}$ ($j = 0, ..., m+d-2$) can be considered as a summation of two parts: the constant part α_i^l and β_j^l, and the nonlinear function part on $\Psi(t)$ which are denoted $\alpha_{i,t} - \alpha_i^l$ and $\beta_{j,t} - \beta_j^l$. Then, the expression of system in the predictor form (3.2.6) can be described by:

$$y(t+d) = \Psi^T(t)\theta + \Psi^T(t)\Theta_\xi^n, \tag{2.3.10}$$

where $\theta = [\alpha_0^l \ \dots \ \alpha_{n-1}^l \ \beta_0^l \ \dots \ \beta_{m+d-2}^l]$ and $\Theta_\xi^n = [(\alpha_{0,t} - \alpha_0^l) \ \dots \ (\alpha_{n-1,t} - \alpha_{n-1}^l) \ (\beta_{0,t} - \beta_0^l) \ \dots \ (\beta_{m+d-2,t} - \beta_{m+d-2}^l)]$. $\varsigma(\cdot) = \Psi^T(t)\Theta_\xi^n$

The following assumptions for the system are used as in Refs.[54, 71, 76]:

Assumption 2 (i) The linear part parameters θ lie in a compact region Σ; (ii) The nonlinear term $\varsigma(\cdot)$ is globally bounded, i.e. $\| \varsigma(\cdot) \| \leq D$ and the bound is known.

2.3.2 Quasi-ARX Neural Network

The elements of Θ_ξ are unknown nonlinear function of $\xi(t)$, which can be parameterized by neural-fuzzy networks and neural networks as in Refs.[24, 72]. In this chapter, a neural network is chosen which can deal with higher dimensional problems.

The quasi-ARX neural network model is expressed by the following equation after parameterizing Θ_ξ with an MIMO neural network:

$$y(t+d) = \Psi(t)^T \mathcal{N}(\xi(t), \Omega)) \tag{2.3.11}$$

where $\mathcal{N}(\cdot, \cdot, \cdot)$ is a generalized 3-layer neural network with n input nodes, M sigmoid hidden nodes and $n+1$ *linear* output nodes[1]. The 3-layer neural network can be expressed by:

$$\mathcal{N}(\xi(t), \Omega) = \theta + W^2 \Gamma(W^1 \xi(t) + B) \tag{2.3.12}$$

where $\Omega = \{W^1, W^2, B, \theta\}$ is the parameters set of the neural network, $W^1 \in \mathcal{R}^{M \times N}$, $W^2 \in \mathcal{R}^{(N+1) \times M}$ are the weight matrices of the first and second layers, $B \in \mathcal{R}^{M \times 1}$ is the bias vector of hidden nodes, $\theta \in \mathcal{R}^{(N+1) \times 1}$ is the bias vector of output nodes, and $\Gamma(\cdot)$ is the diagonal nonlinear operator with identical sigmoid elements σ (for example: $\sigma(x) = \frac{1-e^{-x}}{1+e^{-x}}$). $\xi(t)$ is the input variables of neural network which has been defined in the above section.

Then we can express the quasi-ARX neural network prediction model (2.3.9) in a form of:

$$y(t+d) = \Psi^T(t)\theta + \Psi^T(t) \cdot W^2 \Gamma(W^1 \xi(t) + B). \tag{2.3.13}$$

[1]The number of input node is $N = \dim(\xi(t)) = n + m$, the number of output node is equal to $\dim(\Psi(t)) = N + 1$

2.3.3 Model Parameter Identification

From (2.3.13) we can see that the model parameters can be divided into two classes: the linear part θ and the nonlinear part W^1, W^2, B. Different identification algorithms are used to estimate two parts.

The linear part parameter θ is updated as:

$$\hat{\theta}(t) = \hat{\theta}(t-d) + \frac{a(t)\Psi(t-d)e_1(t)}{1 + \Psi(t-d)^T\Psi(t-d)} \tag{2.3.14}$$

where $\hat{\theta}(t)$ is the estimate of θ at time instant t. And

$$a(t) = \begin{cases} 1 & \text{if} \quad |e_1(t)| > 2D \\ 0 & \text{otherwise} \end{cases} \tag{2.3.15}$$

where $e_1(t)$ is the linear part error and is defined as follows:

$$e_1(t) = y(t+d) - \Psi(t)^T\hat{\theta}(t). \tag{2.3.16}$$

The nonlinear part parameters are adjusted by BP algorithm. The adjusted error of this part is defined by:

$$e_2(t) \quad = \quad y(t+d) - \Psi(t)^T\hat{\theta}(t) - \Psi^T(t)\hat{W}^2(t)\Gamma(\hat{W}^1(t)\xi(t) + \hat{B}(t)) \tag{2.3.17}$$

where $\hat{\Theta}(t) \triangleq \{\hat{W}^1(t), \hat{W}^2(t), \hat{B}(t)\}$ are the estimates of W^1, W^2 and B at time instant t, respectively.

Similar to Ref.[71], no restriction is made on how the parameters $\hat{\Theta}(t)$ are updated except they always lie inside some pre-defined compact region $\hbar$:

$$\hat{\Theta}(t) \in \hbar \,\forall\, t. \tag{2.3.18}$$

2.3.4 Switching Criterion Function

Define the switching criterion function as follows:

$$\begin{aligned} J_i(t) = & \sum_{l=k}^{t} \frac{a_i(l)(\| e_i(l) \|^2 - 4D^2)}{2(1 + a_i(l)\Psi(l-k)^T\Psi(l-k))} \\ & + c * \sum_{l=t-N+1}^{t} (1 - a_i(l) \| e_i(l) \|^2), \quad i = 1, 2. \end{aligned} \tag{2.3.19}$$

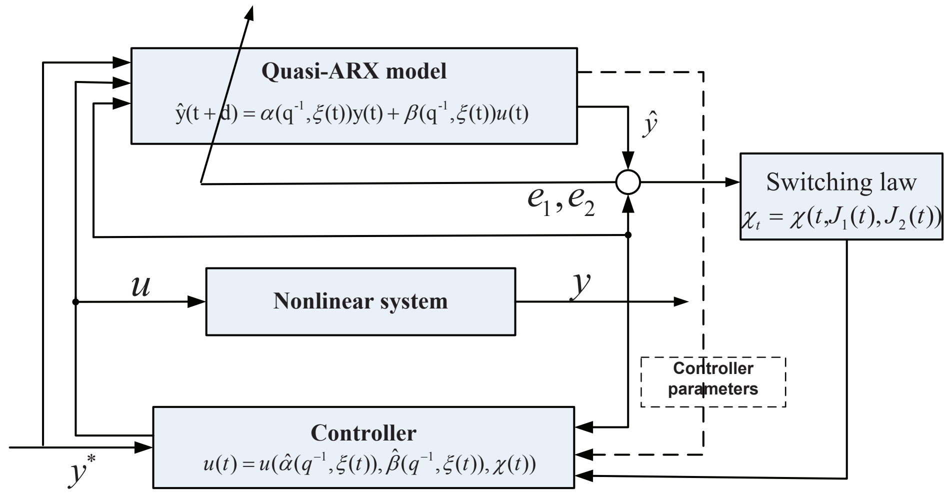

Figure 2.1: A switching control to nonlinear system based on quasi-ARX neural network.

where N is an integer, $c \geq 0$ is a predefined constant, and

$$a_i(t) = \begin{cases} 1 & \text{if} \quad |e_i(t)| > 2D \\ 0 & \text{otherwise}. \end{cases} \tag{2.3.20}$$

Now, give the expression of switching law χ_t based on the switching criterion function:

$$\chi_t = \begin{cases} 1 & \text{if} \quad J_1(t) > J_2(t) \\ 0 & \text{otherwise.} \end{cases} \tag{2.3.21}$$

By comparing $J_1(t)$ and $J_2(t)$, decides when the nonlinear part is abandoned. If $J_1(t) > J_2(t)$ the nonlinear part is added, else only use linear part to identify.

2.4 Controller Design and Its Stability

2.4.1 Controller Design

To control a given system, the controller design includes two steps: the first step for identifying the improved quasi-ARX prediction model; and the second step for deriving and implementing control law. We can obtained the identified improved quasi-ARX prediction model from above parts, expressed by:

$$\hat{y}(t+d) = \Psi^T(t)\hat{\Theta}(\xi(t), \chi_t) \tag{2.4.1}$$

where $\hat{\Theta}(\xi(t), \chi_t) = [\hat{y}_\xi \; \hat{\alpha}_{0,\xi,\chi} \ldots \hat{\alpha}_{n_y-1,\xi,\chi} \; \hat{\beta}_{0,\xi,\chi} \ldots \hat{\beta}_{n_u+d-2,\xi,\chi}]^T$, will be used for controller design. $\hat{\alpha}_{i,\xi,\chi} = \hat{\alpha}_i^l +, \chi_t \hat{\alpha^n}_{i,\xi}$, $\hat{\beta}_{i,\xi,\chi} = \hat{\beta^l}_i +, \chi_t \hat{\beta^n}_{i,\xi}$ and $[\hat{\alpha^n}_{i,\xi}, \hat{\beta^n}_{i,\xi}] = \hat{W}^2\Gamma(\hat{W}^1\xi(t) + \hat{B})$.

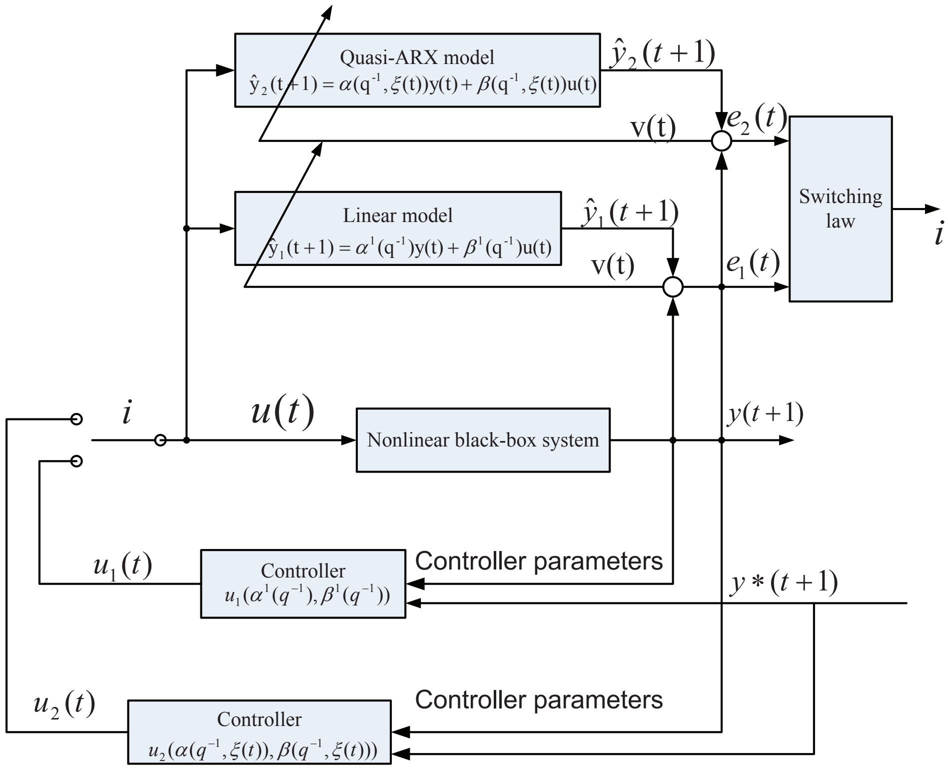

Figure 2.2: A switching control to nonlinear system between a quasi-ARX neural network and a linear model.

Consider a minimum variance control with the criterion function as follows:

$$\mathbf{M}(t+1) = \left[\frac{1}{2}(y(t+d) - y^*(t+d))^2 + \frac{\lambda}{2}u(t)^2\right] \tag{2.4.2}$$

where λ is weighting factor for the control input.

The controller can be obtained by solving:

$$\frac{\partial \mathbf{M}(t+1)}{\partial u(t)} = 0 \tag{2.4.3}$$

In the case where a conventional neural network is used as a prediction model, a controller can not be derived directly from an identified model because of the nonlinearities. However, the improved quasi-ARX neural network model is linear in the input variable $u(t)$. Therefore, a controller is derived from the proposed model:

$$\begin{aligned} u(t) = \frac{\hat{\beta}_{0,\xi,\chi}}{\hat{\beta}^2_{0,\xi,\chi} + \lambda}&((\hat{\beta}_{0,t} - \hat{\beta}(q^{-1}, \xi(t), \chi_t)q)u(t-1) \\ &+y^*(t+1) - \hat{\alpha}(q^{-1}, \xi(t), \chi_t)y(t) - \hat{y}_{\xi,\chi}). \end{aligned} \tag{2.4.4}$$

where the controller parameters $\hat{\alpha}_{i,\xi,\chi}$ and $\hat{\beta}_{j,\xi,\chi}$ come from the predictor and the switching law.

Figure 2.1 shows the adaptive switching controller based on the improved neural network prediction model for unknown nonlinear systems and Fig.2.2 gives a switching control to nonlinear system between linear model and quasi-ARX model. We can see that the identified model and controller share their parameters $\hat{\alpha}(t,\xi(t),\chi_t)$ and $\hat{\beta}(t,\xi(t),\chi_t)$. The switching law χ_t firstly is calculated from input and output signals and model errors, then is used in the controller.

The proposed controller has three distinctive features:

(1) it is linear for the variables synthesized in control systems;

(2) its parameters have explicit meanings;

(3) it is only one controller which combines a switching algorithm.

Give the stability analysis of the proposed nonlinear control system as follows:

Theorem: For the system (2.2.1) with adaptive controller (2.4.4), all the input and output signals in the closed-loop system are bounded. Moreover, the tracking error of the system can converge on zero when a properly neural network is determined.

Proof: Firstly, the model error $e(t)$ is defined by:

$$\begin{aligned} e(t) &= y(t+d) - \Psi(t)^T\hat{\theta}(t) - \chi_t\Psi^T(t)\cdot\hat{W}^2(t)\Gamma(\hat{W}^1(t)\xi(t)+\hat{B}(t)) \\ &= y^*(t+d) - y(t+d) \end{aligned} \tag{2.4.5}$$

Then subtracting θ_0 from both sides of (2.3.14), and gives:

$$\tilde{\theta}(t) = \tilde{\theta}(t-d) - \frac{a(t)\Psi(t-d)(\Psi(t-d)^T\tilde{\theta}(t-d)-\omega(t))}{1+\Psi(t-d)^T\Psi(t-d)} \tag{2.4.6}$$

where $\tilde{\theta}(t) = \hat{\theta}(t) - \theta_0$ and $\omega(t) = y(t+d) - \Psi(t)^T\hat{\theta}(t)$.

Consider the following functional:

$$V(t) = \|\tilde{\theta}(t)\|^2. \tag{2.4.7}$$

Then, noting that $a(t) = 0$ or 1, and combined with (2.3.15) and (2.3.16), we can get as in Ref. [2]:

$$\begin{aligned} V(t) &= V(t-d) - \frac{2a(t)(e_1(t)-\omega(t))e_1(t)}{1+\Psi(t-d)^T\Psi(t-d)} + \frac{a(t)\Psi(t-d)^T\Psi(t-d)e_1(t)^2}{(1+\Psi(t-d)^T\Psi(t-d))^2} \\ &\leq V(t-d) + \frac{a(t)(2e_1(t)\omega(t))}{1+\Psi(t-d)^T\Psi(t-d)} - \frac{a(t)e_1(t)^2}{1+\Psi(t-d)^T\Psi(t-d)} \end{aligned} \tag{2.4.8}$$

From $2ab \leq \kappa a^2 + b^2/\kappa, \forall\kappa$, the following inequality holds:

$$\begin{aligned} V(t) \leq & V(t-d) + \frac{a(t)(e_1^2(t)/2 + 2\omega^2(t))}{1+\Psi(t-d)^T\Psi(t-d)} - \frac{a(t)e_1(t)^2}{1+\Psi(t-d)^T\Psi(t-d)} \\ \leq & V(t-d) + \frac{2a(t)D^2}{1+\Psi(t-d)^T\Psi(t-d)} - \frac{1}{2}\frac{a(t)e_1(t)^2}{1+\Psi(t-d)^T\Psi(t-d)}. \end{aligned} \tag{2.4.9}$$

In view of Eq.(2.4.9), $\{V(t)\}$ is a nonincreasing sequence bounded below by zero. Moreover,

$$\lim_{N\to\infty}\sum_{t=1}^{N}\frac{a(t)(e_1(t)^2-4D^2)}{2(1+\Psi(t-d)^T\Psi(t-d))} < \infty, \tag{2.4.10}$$

and

$$\lim_{N\to\infty}\frac{a(t)(e_1(t)^2-4D^2)}{2(1+\Psi(t-d)^T\Psi(t-d))} \to 0. \tag{2.4.11}$$

From the definition (2.3.16) of $e_1(t)$ and (2.4.1), we have:

$$\begin{aligned} e_1(t) = & \Delta y(t) - \psi^T(t-d)\hat{\theta}(t-d) = y(t) - y(t-d) + y(t-d) - y^*(t) \\ = & y(t) - y^*(t). \end{aligned} \tag{2.4.12}$$

Along with (2.4.12) and (iii) in **Assumptions 1**, there exist positive c_1 and c_2 as in [76] such that:

$$\| \varphi(t-d+1) \| \leq c_1 + c_2 \max_{0\leq\tau\leq t} \| e_1(\tau) \| \tag{2.4.13}$$

It can be seen that the boundedness of $e_1(t)$ determines the boundedness of the input and output signals. Now it is assumed that $e_1(t)$ is unbounded. Then through (2.3.20), there is $T > 0$, when $t > T$, $\|e_1(t)\| > 2D$ and $a_1(t) = 1$, and the numerator in Eq.(2.4.11) is a positive scalar sequence. Therefore, there is a monotony increasing sequence $\|e_1(t_n)\|$ such that $\lim_{t\to\infty}\|e_1(t_n)\|$ as in Ref.[76]. Since

$$\begin{aligned} & \frac{a(t_n)(e_1(t_n)^2-4D^2)}{2(1+\Psi(t_n-d)^T\Psi(t_n-d))} \geq \frac{a(t_n)(e_1(t_n)^2-4D^2)}{2(1+(\|\varphi(t_n-d+1)\|+\|\varphi(t_n-2d+2)\|)^2)} \\ & \geq \frac{a(t_n)(e_1(t_n)^2-4D^2)}{2(1+(2c_1+2c_2\max_{0\leq\tau\leq t_n}\| e_1(\tau) \|)^2)} = \frac{a(t_n)(e_1(t_n)^2-4D^2)}{2(1+(2c_1+2c_2\| e_1(t_n) \|)^2)}, \end{aligned}$$

then,

$$\lim_{t\to\infty}\frac{a(t_n)(e_1(t_n)^2-4D^2)}{2(1+\Psi(t_n-d)^T\Psi(t_n-d))} \geq \frac{1}{8c_6^2} > 0. \tag{2.4.14}$$

But it contradicts (2.4.11). Hence, the assumption is false and $e_1(t)$ is bounded.

By the definition (2.3.17) of $e_2(t)$, (2.4.1) and (iii) in **Assumptions 1**, there exist positive constants d_1, d_2 as in Ref.[76]:

$$\| \varphi(t-d+1) \| \leq d_1 + d_2 \max_{0\leq\tau\leq t} \| e_2(\tau) \| \tag{2.4.15}$$

Along with (iii) of Assumptions 1 similar to Ref.[71], $J_1(t)$ is always bounded by (2.3.19) and (2.4.10). $J_2(t)$ has two cases:

(i) Normal Case: $J_2(t)$ keeps to be small.

By the switching function (2.3.19), $\lim_{N\to\infty} \frac{a_2(t)(e_2(t)^2-4D^2)}{2(1+\psi(t-d)^T\psi(t-d))} \to 0$ holds on. With (2.4.15) and similar to the boundedness proof of $e_1(t)$, the error $e_2(t)$ is bounded. Since $e(t) = (1-\chi_t)e_1(t) + \chi_t e_2(t)$, therefore, $e(t)$ is bounded.

(ii) Abnormal Case: $J_2(t)$ becomes large gradually due to the overfitting of the quasi-ARX NN predictor.

Since $J_1(t)$ is bounded. So there exists a constant t_0 such that $\chi_t = 0,\ \forall t > t_0$. The model also has bounded error $e(t)$.

By (2.4.5) and (iii) in **Assumptions 1**, there also exist positive constants f_1, f_2 as in Ref.[76]:

$$\| \varphi(t-d+1) \| \leq f_1 + f_2 \max_{0\leq\tau\leq t} \| e(\tau) \| \tag{2.4.16}$$

From above inequalities and the boundedness of $e(t)$, the input and output of the closed-loop switching control system are bounded.

Then through the switching function (2.3.19) and switching law (2.3.21), it can be obtained that the system chooses the controller corresponding to the smaller model error as the control input of the system. Therefore, from the definitions of $e_1(t)$ and $e_2(t)$, the tracking error of the system is equivalent to the smaller model error.

The linear control system is always bounded. If a proper nonlinear model is chosen and the accurate parameters is adjusted, the nonlinear control error $e_2(t)$ can converge on zero. It also exists a constant T_0 which satisfies $\chi_t = 1,\ \forall t > T_0$. Then the tracking error of system $\lim_{t\to\infty} \|e(t)\| (= \lim_{t\to\infty} \|e_2(t)\|)$ can converge on zero.

2.5 Control Simulations

Example 1

Now consider a nonlinear SISO system:

$$\begin{aligned} y(t) = & \frac{\exp(-y^2(t-2)) * y(t-1)}{1 + u^2(t-3) + y^2(t-2)} + \frac{(0.5 * (u^2(t-2) + y^2(t-3))) * y(t-2)}{1 + u^2(t-2) + y^2(t-1)} \\ & + \frac{\sin(u(t-1) * y(t-3)) * y(t-3)}{1 + u^2(t-1) + y^2(t-3)} + \frac{\sin(u(t-1) * y(t-2)) * y(t-4)}{1 + u^2(t-2) + y^2(t-2)} \\ & + u(t-1) \end{aligned} \tag{2.5.1}$$

Case 1

The desired output in this example is a piecewise function.

$$y^*(t) = \begin{cases} 0.6y^*(t-1) + r(t-1) \\ \qquad t \in [1, 100] \cup [151, 200] \\ 0.7\text{sign}(0.4493y^*(t-1) + 0.57r(t-1)) \\ \qquad t \in [101, 150] \end{cases} \tag{2.5.2}$$

where $r(t) = 1.2 * \sin(2\pi t/25)$.

In this case, we will chose the switching control system between a linear model and a quasi-ARX model as show in Fig2.2. At the quasi-ARX model part, a neural network with one hidden layer and 20 hidden nodes as in Ref.[54] is used and other parameters satisfy $m = 4, n = 3, c = 1$ and $N = 2$. The quasi-ARX model can be trained off-line by the hierarchical training algorithm as in Ref.[54]. This model is used on-line as an identifier which nonlinear part is adjusted by BP algorithm and linear part by above section mentioned algorithm. The ARX model part, $m = 4, n = 3$. This model is adopted on-line as an identifier by above section mentioned algorithm.

Figure 2.3 gives the results of Example 1. In Fig.2.3(a), the dot line is the desired output, the solid line denotes the proposed method control output $y_1(t)$ and dashed line shows the linear control output $y_0(t)$. The Fig.2.3(b) gives the control input where solid and dashed lines denote the proposed method control and linear control input, respectively. The errors are shown in Fig.2.3(c). The switching sequence is presented which 1 is nonlinear model and 0 is linear model in Fig.2.3(d).

Case 2

The desired output in this example is a piecewise function:

$$y^*(t) = \begin{cases} 0.4493y^*(t-1) + 0.57r(t-1) & t \in [1, 100] \cup [151, 200] \\ 0.7\text{sign}(0.4493y^*(t-1) + 0.57r(t-1)) & t \in [101, 150] \end{cases} \tag{2.5.3}$$

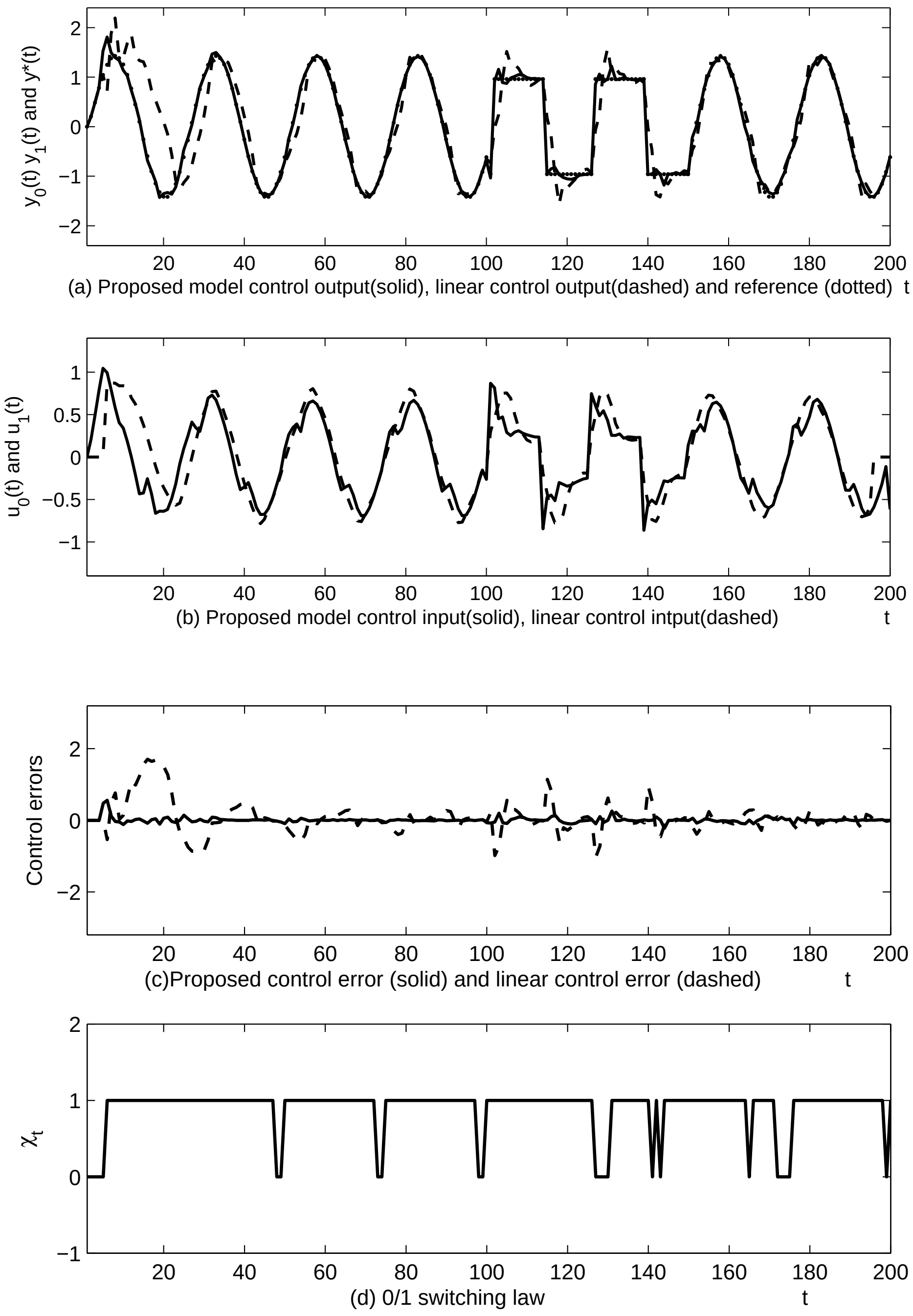

Figure 2.3: Switching control results of Example 1.

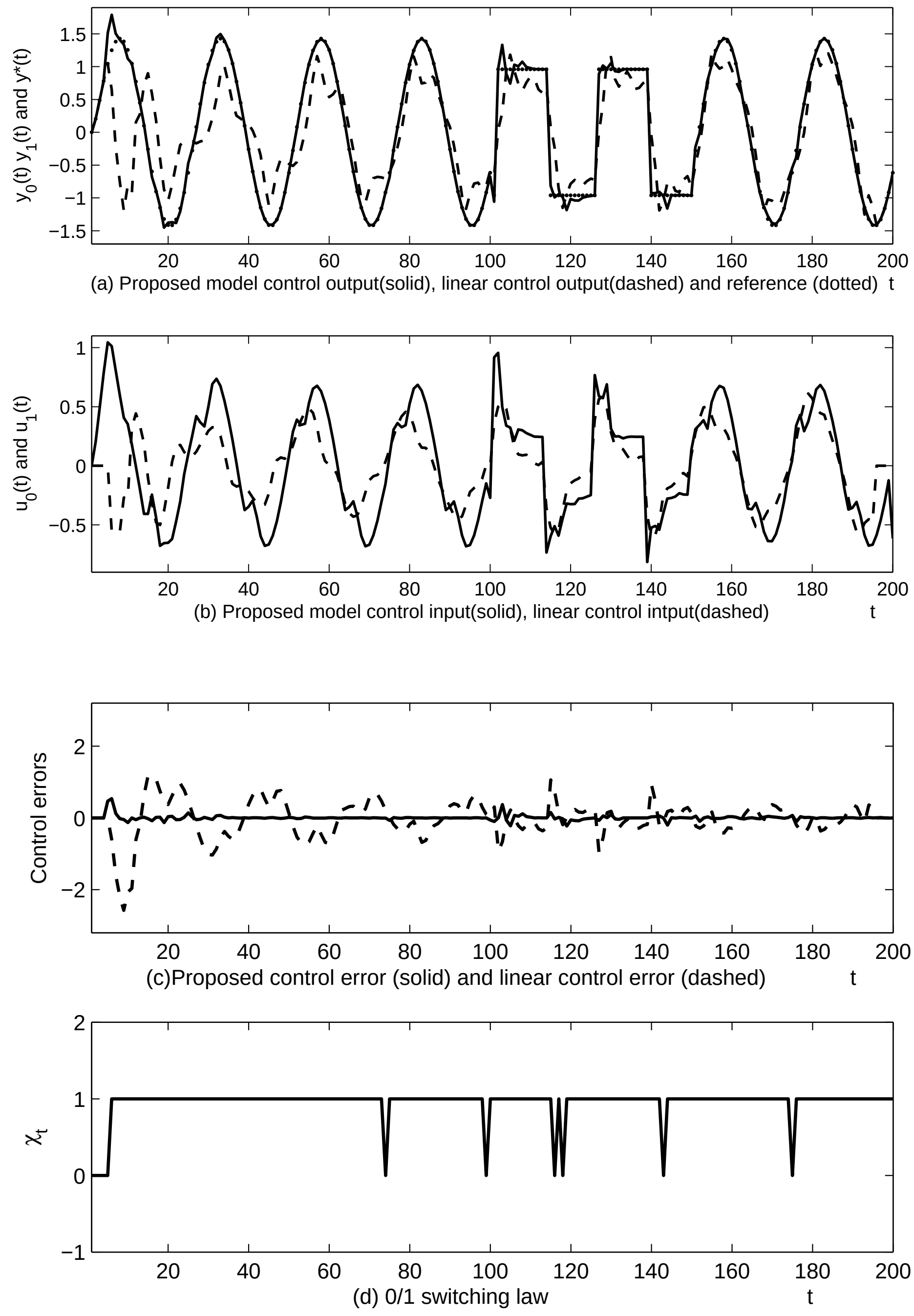

Figure 2.4: Control Results for Example 1.

where $r(t) = 1.2 * \sin(2\pi t/25)$.

To identify the system, we use the following quasi-ARX neural network model:

$$y_1(t+d) = \Psi^T(t)\theta + \Psi^T(t) \cdot W^2\Gamma(W^1\xi(t) + B). \tag{2.5.4}$$

In the nonlinear part, a neural network with one hidden layer and 20 hidden nodes is used and other parameters satisfy $m = 4, n = 3, d = 1$. The improved quasi-ARX model can be firstly trained off-line by the hierarchical training algorithm as in Ref.[54]. Figure 2.4 shows the performance when the adaptive controller (2.4.4) is used. The parameters of switching criterion function are chosen to be $c = 1.2$ and $N = 3$.

In Fig.2.4(a), the dot line is the desired output, the solid line denotes the proposed method control output $y_1(t)$ and dashed line shows the linear control output $y_0(t)$. Obviously, the control output with the proposed method is nearly consistent with the desired output at most of the time. The mean of linear control errors is -0.0364 and the variance is 0.2930. The mean of the proposed method control errors is 0.0035 and the variance is 0.0053. Therefore, our method is better than linear control. The Fig.2.4(b) gives the control input where solid line and dashed line denotes the proposed method control input $u_1(t)$ and linear control input $u_2(t)$, respectively. We can see that the input signals have small fluctuation. The errors are shown in Fig.2.4(c). The switching sequence is presented which 1 is model with nonlinear part and 0 is model without nonlinear part in Fig.2.4(d)). From the Fig.2.4(d), even though the model with nonlinear part can often control very well, it degrades sometimes and the model only with linear part has to work until the nonlinear part can recover. Therefore, the linear part will work all the time, but the neural network part will work under the switching sequence.

Example 2

The system is a nonlinear one governed by

$$y(t) = f[y(t-1), y(t-2), y(t-3), u(t-1), u(t-2)] \tag{2.5.5}$$

where

$$f[x_1, x_2, x_3, x_4, x_5] = \frac{x_1x_2x_3x_5(x_3-1) + x_4}{1 + x_2^2 + x_3^2}.$$

Case 1

$e(t) \in N(0, 0.001)$ is a white noise. The desired output in this example is a piecewise function.

$$y^*(t) = \begin{cases} 0.6y^*(t-1) + r(t-1) \\ \qquad\qquad t \in [1, 100] \cup [151, 200] \\ 0.7\text{sign}(0.4493y^*(t-1) + 0.57r(t-1)) \\ \qquad\qquad t \in [101, 150] \end{cases} \tag{2.5.6}$$

where $r(t) = 1.2 * \sin(2\pi t/25)$. The algorithm is similar with Example 1 whose parameters satisfy $m = 3, n = 2, c = 1.5$ and $N = 3$. and results is shown in Fig.2.5.

Case 2

The desired output in this example is a piecewise function.

$$y^*(t) = \begin{cases} 0.6y^*(t-1) + r(t-1) & t \in [1, 100] \cup [151, 200] \\ 0.7\text{sign}(0.4493y^*(t-1) + 0.57r(t-1)) & t \in [101, 150] \end{cases} \tag{2.5.7}$$

where $r(t) = \sin(2\pi t/25)$.

In the nonlinear part, a neural network with one hidden layer and 20 hidden nodes is used and other parameters satisfy $m = 3, n = 2, d = 1$. Figure 2.6 shows the performance when the adaptive controller (2.4.4) is used. The parameters of switching criterion function are chosen to be $c = 1.5$ and $N = 3$.

Figure 2.6 gives the results of Example 2 whose marks are same with Example 1. From the Fig.2.6(a), the linear control output signals have larger amplitude and far away from the desired output. However, the proposed control output is almost coincidence with the desired output. The similar conclusion also can be get from errors. The mean of linear control errors is -0.1011 and the variance is 0.0687. The mean of the proposed method control is -0.0090 and the variance is 0.0031. The Fig.2.6(d) shows that the switching mechanism is efficient.

2.6 Conclusion

In this chapter, a new framework for the nonlinear system adaptive control is established based on an improved quasi-ARX neural network which a switching algorithm is introduced. Different from some relative work which established more than two prediction models and made switching among so many corresponding controllers as in Ref. [71, 76], the proposed method is simpler and control-easier because of the compact and efficient structure of control system. Simulations have been given to show the effectiveness of the proposed method both on stability and accuracy.

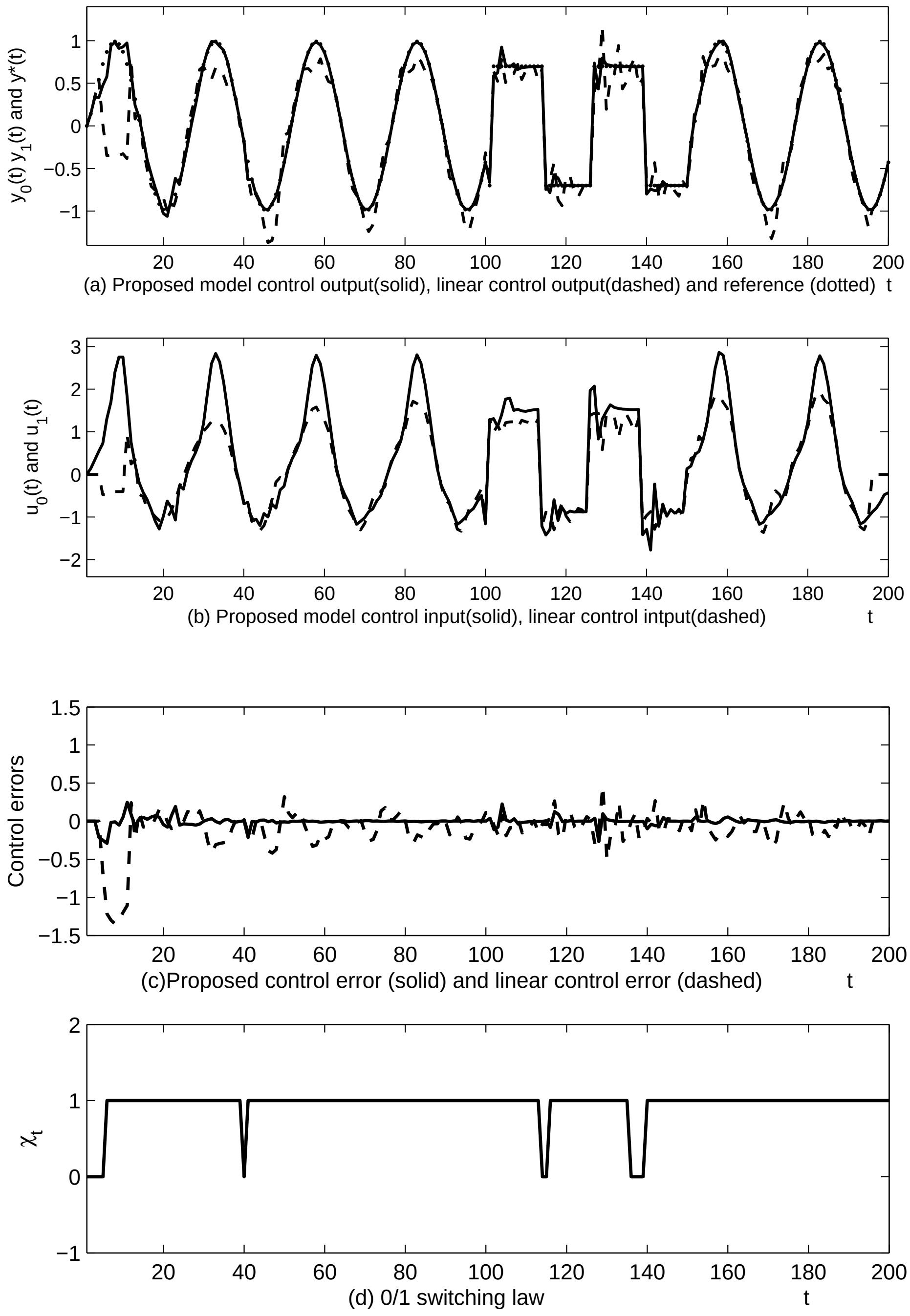

Figure 2.5: Switching control results of Example 2.

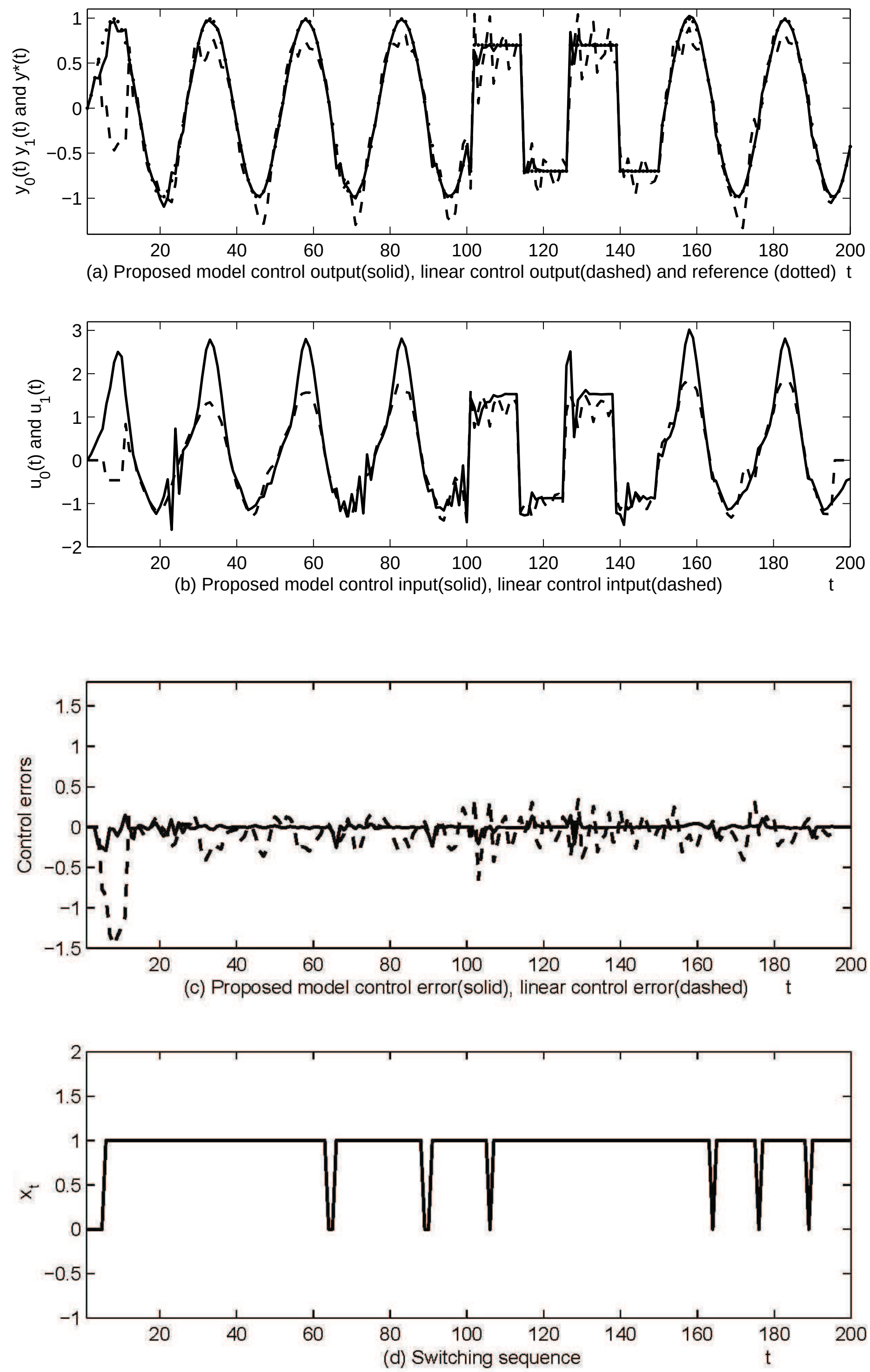

Figure 2.6: Control Results for Example 2.

Chapter 3

Adaptive Fuzzy Switching Control of Nonlinear Systems Based on Quasi-ARX RBFN Model

3.1 Introduction

In the past decades, there has been a lot of interests in the stabilizing adaptive control of dynamical systems [2, 78, 10]. Some adaptive control schemes for dynamical systems via linear control theory have been obtained as in Refs.[79, 60, 80]. However, the stabilizing adaptive control of dynamical systems is a difficult problem because the plants are always nonlinear in practical dynamical systems. Hence, the performance of linear control models can not satisfy requirement. For this reason, some nonlinear prediction models have been developed for nonlinear systems to overcome the difficulty in predictor and controller design for nonlinear systems. Until now, Neural Networks (NNs)[18, 19, 20], Wavelet Networks (WNs) [21, 22], Neuro-fuzzy Networks (NFNs)[23, 24] and Radial Basis Function Networks (RBFNs) [25, 26] have been directly used to identify and control nonlinear dynamical systems because of their abilities to approximate arbitrary mapping to any desired accuracy. However, it still exists difficulties in parameter identification, controller design, and stability guarantee, during using these control systems.

The multiple model system structure was firstly proposed in Ref.[71], which contains a linear model, a NN-based nonlinear model and a 0/1 switching mechanism. The system structure is utilized to ensure the stability of control system and to improve the control performance. And in Ref.[76], the assumption of global boundedness on higher-order nonlinear terms is relaxed by introducing a d-difference operator, and a rigorous analysis on the tracking error is presented. All these control methods have to identify at least two models. To simplify the identification for control, in our previous work a quasi-ARX NN model with a switching mechanism has been studied for nonlinear system adaptive control as in Refs.[81, 20], which is a combination of a linear part and a following 0/1 switching nonlinear part. It can satisfy the stability and the performance requirement by using only one model. Nevertheless, there are still some aspects needed to be improved in the control method based on quasi-ARX NN model. One is that the 0/1 hard switching method is not very smooth; the second is the assumption of global boundedness also can be relaxed; the third is that the parameters of quasi-ARX NN model to be adjusted on-line are highly nonlinear, which deteriorates the adaptability of control system.

In this chapter, a d-difference operator is used in the ARX-like expression of system to relax the assumption of global boundedness on higher-order nonlinear terms as in Ref.[76]. And a fuzzy switching mechanism is constructed based on the system switching criterion function. The corresponding switching controller is obtained, which is different with the 0/1 switching law between multiple models. The fuzzy switching mechanism has three situations: one is that the controller becomes a linear controller when the fuzzy switching function value equals to 0 and the nonlinear part is abandoned; another is that the fuzzy switching function value equals to 1 and the nonlinear part is fully used; the third is that the fuzzy switching function value belongs to (0,1), in which the control accuracy is improved with more emphasis on the nonlinear part, while the convergence speed is improved with less emphasis on the control accuracy. This fuzzy switching mechanism is also different with the normal fuzzy control as in [82, 83] because it is just used in the prediction model and depends on a switching criterion function.

As we know, the quasi-ARX model embodies an ARX-like macro model part and a kernel part [54, 72, 20]. The kernel part is an ordinary network model, such as NNs, WNs, NFNs and RBFNs. to parameterize the nonlinear coefficients of macro-model Some types of the ordinary network models, such as WNs, RBFNs, and NFNs, can be regarded as nonlinear models linear in parameters through fixing the nonlinear parameters by *a priori* knowledge[84, 85, 86, 87, 67]. Incorporating the network models with this characteristic, the quasi-ARX model becomes linear-in-parameters if those nonlinear parameters are determined off-line. During control process, only linear parameters are adjusted on-line which can reduce response time of adaptive control. RBFNs have been used for the nonlinear system control because of their simple topological structure and precision in nonlinear approximation [25, 26, 88, 89]. Compared with NN, RBFN is understandable in terms of parameters, then is introduced as the kernel part in the quasi-ARX model to replace the NN which has been used in Refs. [71, 54, 76, 20].

Motivated by the above discussions, a stabilizing switching control for nonlinear system is proposed based on the quasi-ARX RBFN model, the d-difference operator and the fuzzy switching mechanism. The parameters of quasi-ARX RBFN model are categorized into three types: the first type of parameters for the linear part of model, the second type of linear parameters for the nonlinear part of model and the third type of nonlinear parameters for the nonlinear part of model. The first two types of linear parameters are all adjusted by a recursive Least Square (LS) algorithm on-line, while the third type of nonlinear parameters is determined by applying an Affinity Propagation (AP) clustering method off-line [90].

The chapter is organized as follows: Section 3.2 describes the nonlinear system considered, and a d-different operator is used to obtain an ARX-like expression of system in d-different form. Section 3.3 introduces a quasi-ARX RBFN prediction model whose parameters are identified by AP clustering method and LS algorithms. Section 3.4 constructs a fuzzy switching adaptive control system based on the quasi-ARX RBFN predictors, and analyzes the stability of the control system. Section 3.5 carries out numerical simulations to show the effectiveness of the proposed control method. Finally, Section 3.6 presents the conclusions.

3.2 Problem Description

3.2.1 Systems

Consider a single-input-single-output (SISO) nonlinear time-invariant dynamical system with input-output relation as:

$$\begin{aligned} y(t+d) &= g(\varphi(t)), \\ \varphi(t) &= [y(t+d-1), ...y(t+d-n), u(t)\ , ...,\ u(t-m+1)]^T \end{aligned} \tag{3.2.1}$$

where $y(t)$ denotes the output at time t ($t = 1, 2, ...$), $u(t)$ the input, d the known integer time delay, $\varphi(t)$ the regression vector, and n , m the system orders. $g(\cdot)$ is a smooth nonlinear function, and at a small region around $\varphi(t) = 0$, it is C^∞ continuous. The origin is an equilibrium point, then $g(0) = 0$.

3.2.2 ARX-Like Expression

Under the continuous condition, the unknown nonlinear function $g(\varphi(t))$ can be performed Taylor expansion on a small region around $\varphi(t) = 0$:

$$y(t+d) = g'(0)\varphi(t) + \frac{1}{2}\varphi^T(t)g''(0)\varphi(t) + \ ... \tag{3.2.2}$$

where the prime denotes differentiation with respect to $\varphi(t)$. Then the following notations are introduced:

$$(g'(0)+\frac{1}{2}\varphi^T(t)g''(0)+\cdots)^T=[a_{1,t}\ ...\ a_{n,t}\ b_{0,t}\ ...\ b_{m-1,t}]^T$$

where $a_{i,t}=a_i(\varphi(t))$ $(i=1,..,n)$ and $b_{j,t}=b_j(\varphi(t))$ $(j=0,...,m-1)$ are nonlinear functions of $\varphi(t)$.

However, we need to get $y(t+d)$ by using the input-output data up to time t in a model. The coefficients $a_{i,t}$, and $b_{i,t}$, need to be calculable using the input-output data up to time t. To do so, let us iteratively replace $y(t+l)$ in the expressions of $a_{i,t}$ and $b_{j,t}$ with functions:

$$y(t+l)\Rightarrow g(\tilde{\varphi}(t+l)),\ l=1,...,d-1 \tag{3.2.3}$$

where $\tilde{\varphi}(t+l)$ is $\varphi(t+l)$ whose elements $y(t+k),\ l+1<k\leq d-1$ are replaced by Equ.(3.2.3), and define the new expressions of the coefficients by:

$$a_{i,t}=\tilde{a}_{i,t}=\tilde{a}_i(\phi(t)),\quad b_{j,t}=\tilde{b}_{j,t}=\tilde{b}_j(\phi(t))$$

where $\phi(t)$ is a vector:

$$\phi(t)=[y(t)\,...\,y(t-n+1)\,u(t)\,...\,u(t-m-d+2)]^T. \tag{3.2.4}$$

Now, introduce two polynomials $A(q^{-1},\phi(t))$ and $B(q^{-1},\phi(t))$ based on the coefficients, defined by:

$$A(q^{-1},\phi(t))=1-a_{1,t}q^{-1}-\ldots-a_{n,t}q^{-n};$$
$$B(q^{-1},\phi(t))=b_{0,t}+\ldots+b_{m-1,t}q^{-m+1},$$

where q^{-1} is a backward shift operator, e.g. $q^{-1}u(t)=u(t-1)$. Then, the nonlinear system (3.2.1) can be equivalently represented as the following ARX-like expression:

$$A(q^{-1},\phi(t))y(t+d)=B(q^{-1},\phi(t))u(t). \tag{3.2.5}$$

By the Equ.(3.2.5), $y(t+d)$ satisfies the following equation as in Ref.[54]:

$$y(t+d)=\alpha(q^{-1},\phi(t))y(t)+\beta(q^{-1},\phi(t))u(t), \tag{3.2.6}$$

where

$$\alpha(q^{-1},\phi(t))=G(q^{-1},\phi(t)))=\alpha_{0,t}+\alpha_{1,t}q^{-1}+...+\alpha_{n-1,t}q^{-n+1}; \tag{3.2.7}$$
$$\beta(q^{-1},\phi(t))=F(q^{-1},\phi(t))B(q^{-1},\phi(t)),$$
$$=\beta_{0,t}+\beta_{1,t}q^{-1}+...+\beta_{m+d-2,t}q^{-m-d+2}, \tag{3.2.8}$$

and $G(q^{-1}, \phi(t))$, $F(q^{-1}, \phi(t))$ are unique polynomials satisfying:

$$F(q^{-1}, \phi(t))A(q^{-1}, \phi(t)) = 1 - G(q^{-1}, \phi(t))q^{-d}. \tag{3.2.9}$$

3.2.3 D-difference Expression

The coefficients $\alpha_{i,t}$ ($i = 0, ..., n-1$) and $\beta_{j,t}$ ($j = 0, ..., m+d-2$) can be considered as a summation of two parts: the constant part α_i^l and β_j^l, and the nonlinear function part on $\phi(t)$ which are denoted $\alpha_{i,t} - \alpha_i^l$ and $\beta_{j,t} - \beta_j^l$. Then, the expression of system in the predictor form (3.2.6) can be described by:

$$y(t+d) = \phi^T(t)\theta + \phi^T(t)\Theta_\phi^n, \tag{3.2.10}$$

where $\theta = [\alpha_0^l \ ... \ \alpha_{n-1}^l \ \beta_0^l \ ... \ \beta_{m+d-2}^l]$ and $\Theta_\phi^n = [(\alpha_{0,t} - \alpha_0^l) \ ... \ (\alpha_{n-1,t} - \alpha_{n-1}^l) \ (\beta_{0,t} - \beta_0^l) \ ... \ (\beta_{m+d-2,t} - \beta_{m+d-2}^l)]$.

Apply a d-difference operator, defined by $\Delta = 1 - q^d$, to (3.2.10). Then the following expression of system in d-difference form can be obtained:

$$\Delta y(t+d) = \psi^T(t)\theta + \varsigma(\Psi(t)), \tag{3.2.11}$$

where $\psi(t) = \Delta\phi(t)$. $\varsigma(\Psi(t)) = \Psi^T(t)\tilde{\theta}_\Psi^n = \Delta\phi^T(t)\Theta_\phi^n$ and $\Psi(t) = [y(t) \ ... \ y(t-d-n+1) \ u(t) \ ... \ u(t-m-2d+2)]^T$.

The following assumptions for the system are used as in Refs.[54, 71, 76]:

Assumption 1: (i) The system under consideration has a global representation (3.2.10); (ii) The linear part parameters θ lie in a compact region Σ; (iii) The system has a globally uniformly asymptotically stable zero dynamics; (iv) The nonlinear difference term $\varsigma(\cdot)$ is globally bounded, i.e. $\| \varsigma(\cdot) \| \leq D$ and the bound is known; (v) The system is controllable, in which a reasonable unknown controller may be expressed by $u(t) = \rho(\xi(t))$, where $\xi(t)$ is defined in Section (3.3.1).

3.3 Quasi-ARX RBFN Prediction Model

3.3.1 Quasi-ARX RBFN Model

As we know, a controller can be derived easily and can share parameters from the identified prediction model, when the prediction model is linear in the input variable $u(t)$. However, the Equ.(3.2.11) is a general one which is nonlinear in the variable $u(t)$, because the $\tilde{\theta}_\Psi^n$ are based on $\Psi(t)$ whose elements contain $u(t)$. To solve this problem, an *extra variable* $x(t)$[1] is introduced and an unknown

[1] Obviously, in a control system, the reference signal $y^*(t+d)$ can be used as the extra variable $x(t+d)$.

nonlinear function $\rho(\xi(t))$ is used to replace the variable $u(t)$ in $\tilde{\theta}^n_\Psi$, Under **Assumption 1**(v), the function $\rho(\xi(t))$ exists. Define:

$$\xi(t) = [y(t) \ ... \ y(t-n_1) \ x(t+d) \ ... \ x(t-n_3+d) \ u(t-1) \ ... \ u(t-n_2)]^T$$

including the extra variable $x(t+d)$ as an element. A typical choice for $n_1, \ n_2$, and n_3 in $\xi(t)$ is $n_1 = n+d-1$, $n_2 = m+2d-2$ and $n_3 = 0$. We can express the Equ.(3.2.11) by:

$$\Delta y(t+d) = \psi^T(t)\theta + \Psi^T(t)\theta^n_\xi, \tag{3.3.1}$$

where $\theta^n_\xi = \tilde{\theta}^n_\Psi$.

The elements of θ^n_ξ are unknown nonlinear function of $\Phi(t)$, which can be parameterized by NN or RBFN. In this chapter, the RNFN is used which has local property.

$$\theta^n_\xi = \sum_{j=1}^{M} \mathbf{w}_j R_j(\xi(t), \Omega_j), \tag{3.3.2}$$

where M is the number of RBFs, $\mathbf{w}_j = [\omega_{1j}, \omega_{2j}, ..., \omega_{Nj}]^T$ the coefficient vector, and $R_j(\xi(t), \Omega_j)$ the RBFs defined by:

$$R_j(\xi(t), \Omega_j) = e^{-\lambda_j \|\xi(t) - Z_j\|^2} \qquad j = 1, 2, ..., M, \tag{3.3.3}$$

where $\Omega_j = \{\lambda_j, Z_j\}$ is the parameters set of the RBFN; Z_j is the center vector of RBF and λ_j are the scaling parameters; $\| \bullet \|_2$ denotes the vector two-norm. Then we can express the quasi-ARX RBFN prediction model for (3.3.1) in a form of:

$$\Delta y(t+d) = \psi^T(t)\theta + \sum_{j=1}^{M} \Psi^T(t)\mathbf{w}_j R_j(\xi(t), \Omega_j). \tag{3.3.4}$$

Now, introducing the following notations:

$$\mathbf{W} = [\ \mathbf{w}_1 \ \mathbf{w}_2 \ ... \ \mathbf{w}_M \ ] = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1M} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ w_{N1} & w_{N2} & \cdots & w_{NM} \end{bmatrix}; \tag{3.3.5}$$

$$\mathcal{N}(\xi(t)) = \begin{bmatrix} e^{-\lambda_1 \|\xi(t) - Z_1\|^2} \\ \cdot \\ \cdot \\ e^{-\lambda_M \|\xi(t) - Z_M\|^2} \end{bmatrix}, \tag{3.3.6}$$

the quasi-ARX RBFN model is further expressed by

$$\Delta y(t+d)=\psi^T(t)\theta + \Psi^T(t)\mathbf{W}\mathcal{N}(\xi(t)) = \psi^T(t)\theta + \Xi(t)^T\Theta, \tag{3.3.7}$$

where $\Theta = [\ w_{11}\ ...\ w_{n1}\ ...\ w_{1M}\ ...\ w_{nM}\ ]^T$ and $\Xi(t) = \mathcal{N}(\xi(t)) \otimes \Psi(t)$.

Remark 1 Comparing with Ref.[76], in which the model described by its Equ.(16) is only an approximate one, the quasi-ARX RBFN prediction model described by Equ.(3.3.4) is an accurate model of the system in d-difference form (3.2.11).

3.3.2 Parameter Estimation

By (3.3.7), according to the parameter property, the model parameters are divided into three groups: the linear parameter θ of the linear part $\psi^T(t)\theta$, the linear parameter Θ and the nonlinear parameter Ω_j of the nonlinear part $\Psi^T(t)\mathbf{W}\mathcal{N}(\xi(t))$. The nonlinear parameters Ω_j are determined off-line. Let us denote the estimation of Ω_j by $\hat{\Omega}_j$. In order to determine the centers and widths of the RBFN, AP clustering method is employed. The center Z_j is the arithmetic mean value of all training data in each cluster. The width λ_j is ϱ times the largest distances between all training data in each cluster. The parameters θ and Θ are estimated by using on-line identification algorithms, respectively.

The linear parameter θ of linear part of model is updated as in Ref.[71]:

$$\hat{\theta}(t) = \hat{\theta}(t-d) + \frac{a(t)\psi(t-d)e_1(t)}{1+\psi(t-d)^T\psi(t-d)}, \tag{3.3.8}$$

where $\hat{\theta}(t)$ is the estimate of θ at time instant t, which also denotes the parameter of a linear model used to approximate the system in d-difference form. And

$$a(t) = \begin{cases} 1 \text{if} \quad |e_1(t)| > 2D \\ 0 \text{otherwise}, \end{cases} \tag{3.3.9}$$

where $e_1(t)$ denotes the error of the linear model, defined by

$$e_1(t) = \Delta y(t) - \psi(t-d)^T\hat{\theta}(t-d). \tag{3.3.10}$$

The linear parameter Θ of nonlinear part of the quasi-ARX model is updated by a LS algorithm:

$$\hat{\Theta}(t) = \hat{\Theta}(t-d) + \frac{P(t)\Xi(t-d)e_2(t)}{1+\Xi(t-d)^T P(t)\Xi(t-d)}, \tag{3.3.11}$$

where $\hat{\Theta}(t)$ is the estimate of Θ at time instant t. $\hat{\Theta}(0) = \Theta_0$ is assigned with an appropriate initial value. $e_2(t)$ is the error of quasi-ARX model, defined by

$$e_2(t) = \Delta y(t) - \psi(t-d)^T\hat{\theta}(t-d) - \Xi^T(t-d)\hat{\Theta}(t-d). \tag{3.3.12}$$

And

$$P(t) = \frac{P(t-d) - P^T(t-d)\Xi(t-d)^T\Xi(t-d)P(t-d)}{1 + \Xi(t-d)^T P(t)\Xi(t-d)}. \qquad (3.3.13)$$

Similar to Ref.[71], no restriction is made on how the parameters $\hat{\Theta}(t)$ are updated except they always lie inside some pre-defined compact region $\hbar$:

$$\hat{\Theta}(t) \in \hbar \, \forall \, t. \qquad (3.3.14)$$

3.4 Controller Design and Its Stability

3.4.1 Switching Criterion Function

Consider a similar switching criterion function as Ref.[71]:

$$\begin{aligned} J_i(t) = & \sum_{l=d}^{t} \frac{a_i(l)(\| e_i(l) \|^2 - 4D^2)}{2(1 + a_i(l)\psi(l-d)^T\psi(l-d))} \\ & + c * \sum_{l=t-N+1}^{t} (1 - a_i(l) \| e_i(l) \|^2), \quad i = 1, 2, \end{aligned} \qquad (3.4.1)$$

where N is an integer and $c \geq 0$ is a predefined constant. And,

$$a_i(t) = \begin{cases} 1 \text{ if } & |e_i(t)| > 2D \\ 0 \text{ otherwise}. \end{cases} \qquad (3.4.2)$$

It is obvious that $a1(t) = a(t)$.

In most switching control methods based on two or more prediction models [71, 76, 20], hard switching laws are used. That means that in those control systems, the linear and nonlinear predictors are alternately used. However, the jumping switch will decrease the precision and adaptability of the control system. Motivated by the accuracy requirement, we introduce a fuzzy switching law μ_t based on the criterion function $J_1(t)$ and $J_2(t)$:

$$\mu_t = \begin{cases} 1 & \text{if } \eta(t) > K \\ \eta(t) & \text{if } k \leq \eta(t) \leq K \\ 0 & \text{if } \eta(t) < k, \end{cases} \qquad (3.4.3)$$

where K and k are positive constants which satisfy $k \in (0, 0.5)$, $K \in (0.5, 1)$ and $\eta(t)$ is a function of $J_1(t)$ and $J_2(t)$ defined by

$$\eta(t) = \frac{J_1(t)}{J_1(t) + J_2(t) + \epsilon} \in [0, 1], \qquad (3.4.4)$$

where ϵ is a very small positive constant. When $\mu_t = 0$, $\hat{\Theta}(t) = \Theta_0$, which resets $\hat{\Theta}(t)$ to its initial value.

3.4.2 Adaptive Controller

Designing a controller for the nonlinear system (3.2.1) includes two steps: the first step to identify the quasi-ARX model; and the second step to derive and implement the control law. Based on the identified quasi-ARX model (3.3.4), we construct a prediction model expressed by:

$$\hat{y}(t+d)=(1-\mu_t)\hat{y}_l(t+d)+\mu_t\hat{y}_n(t+d) \tag{3.4.5}$$

where

$$\hat{y}_l(t+d)=\psi^T(t)\hat{\theta}(t)+y(t) \tag{3.4.6}$$

$$\hat{y}_n(t+d)=\psi^T(t)\hat{\theta}(t)+\sum_{j=1}^{M}\Psi^T(t)\hat{\mathbf{w}}_j(t)R_j(\xi(t),\hat{\Omega}_j)+y(t). \tag{3.4.7}$$

Consider a minimum variance control with the criterion function as follows:

$$\mathbf{M}(t+d)=\frac{1}{2}(y(t+d)-y^*(t+d))^2, \tag{3.4.8}$$

where $y^*(t)$ is a known bounded reference output. The optimal control law minimizing (3.4.8) is:

$$y(t+d)-y^*(t+d)=0. \tag{3.4.9}$$

Then corresponding to the predictors (3.4.5)-(3.4.7), we can obtain the following controllers:

$$\mathcal{C}:\psi^T(t)\hat{\theta}(t)+\mu_t\sum_{j=1}^{M}\Psi^T(t)\hat{\mathbf{w}}_j(t)R_j(\xi(t),\hat{\Omega}_j)=y^*(t+d)-y(t), \tag{3.4.10}$$

and two others $\mathcal{C}_l$ and $\mathcal{C}_n$ corresponding to the extreme cases of $\mu_t=0$ and $\mu_t=1$, respectively

$$\mathcal{C}_l:\psi^T(t)\hat{\theta}(t)=y^*(t+d)-y(t) \tag{3.4.11}$$

$$\mathcal{C}_n:\psi^T(t)\hat{\theta}(t)+\sum_{j=1}^{M}\Psi^T(t)\hat{\mathbf{w}}_j(t)R_j(\xi(t),\hat{\Omega}_j)=y^*(t+d)-y(t) \tag{3.4.12}$$

Figure 3.1 shows the proposed adaptive fuzzy switching control system based on the quasi-ARX RBFN for nonlinear systems. The control system has four distinctive features:

1) The control system (3.4.10) is linear for the variable synthesized, $u(t)$, including in the regression vectors $\psi(t)$ and $\Psi(t)$;

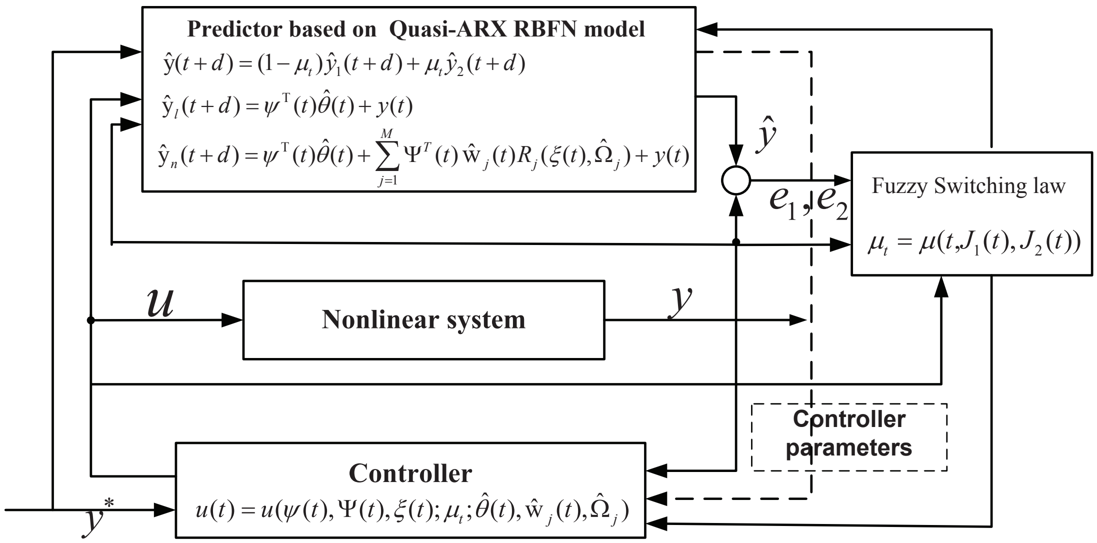

Figure 3.1: A nonlinear adaptive control system based on the quasi-ARX RBFN model and the fuzzy switching law.

2) The three predictors (3.4.5)-(3.4.7) are obtained directly from only one identified quasi-ARX model, and all are linear for the control variable $u(t)$ to be synthesized in the control system;

3) The nonlinear control system could have quick response since only linear parameters are adjusted on-line;

4) The control system employs a fuzzy switching mechanism instead of a simple 0/1 switching.

3.4.3 Stability Analysis

Give the stability analysis of the proposed nonlinear control system as follows:

Theorem: For the system (3.2.1) with adaptive fuzzy switching controller (3.4.10), all the input and output signals in the closed-loop system are bounded. Moreover, the tracking error of the system can converge on zero when a properly RBFN is determined.

Proof: Defining $\tilde{\theta}(t) = \hat{\theta}(t) - \theta$ and by the adaptation law (3.3.8), it follows that as described in Refs.[71, 76]:

$$\|\tilde{\theta}(t)\|^2 \leq \|\tilde{\theta}(t-d)\|^2 - \frac{a_1(t)(\|e_1(t)\|^2 - 4D^2)}{2(1+\psi^T(t))\psi(t)}.$$

Similar to Refs.[71, 76], under the condition (3.4.2), $\tilde{\theta}(t)$ is bounded. Moreover, we can get:

$$\lim_{N\to\infty}\sum_{t=1}^{N}\frac{a_1(t)(e_1(t)^2 - 4D^2)}{2(1+\psi(t-d)^T\psi(t-d))} < \infty, \tag{3.4.13}$$

and

$$\lim_{t\to\infty} \frac{a_1(t)(e_1(t)^2 - 4D^2)}{2(1+\psi(t-d)^T\psi(t-d))} \to 0. \tag{3.4.14}$$

From the definition (3.3.10) of $e_1(t)$ and (3.4.11), we have:

$$\begin{aligned} e_1(t) =& \Delta y(t) - \psi^T(t-d)\hat{\theta}(t-d) \\ =& y(t) - y(t-d) + y(t-d) - y^*(t) \\ =& y(t) - y^*(t). \end{aligned} \tag{3.4.15}$$

Along with (3.4.15) and (iii) in **Assumptions 1**, there exist positive c_1 and c_2 such that:

$$\| \varphi(t-d+1) \| \leq c_1 + c_2 \max_{0\leq\tau\leq t} \| e_1(\tau) \| \tag{3.4.16}$$

From Ref.[76], if $e_1(t)$ is unbounded, then it will introduce the contradiction of (3.4.14) through using (3.4.16). Therefore, we can get that $e_1(t)$ is bounded.

By the definition (3.3.12) of $e_2(t)$, (3.4.12) and (iii) in **Assumptions 1**, there exist positive constants d_1, d_2 as in Ref.[76]:

$$\| \varphi(t-d+1) \| \leq d_1 + d_2 \max_{0\leq\tau\leq t} \| e_2(\tau) \| \tag{3.4.17}$$

The error $e(t)$ is defined as follows:

$$\begin{aligned} e(t) =& \Delta y(t+d) - \psi^T\hat{\theta}(t) - \mu_t \sum_{j=1}^{M} \Psi^T(t)\hat{\mathbf{w}}_j R_j(\Phi(t), \hat{\Omega}_j) \\ =& y^*(t+d) - y(t+d). \end{aligned} \tag{3.4.18}$$

By (3.4.18) and (iii) in **Assumptions 1**, there also exist positive constants f_1, f_2 as in Ref.[76]:

$$\| \varphi(t-d+1) \| \leq f_1 + f_2 \max_{0\leq\tau\leq t} \| e(\tau) \| \tag{3.4.19}$$

We can easily find that the second term in (3.4.1) is always bounded by (3.4.2). Therefore, $J_1(t)$ is always bounded through employing (3.4.13). $J_2(t)$ has two cases:

(i) Normal Case: $J_2(t)$ keeps to be small.

By the switching function (3.4.1), $\lim_{N\to\infty} \frac{a_2(t)(e_2(t)^2-4D^2)}{2(1+\psi(t-d)^T\psi(t-d))} \to 0$ holds on. With (3.4.17) and similar to the boundedness proof of $e_1(t)$, the error $e_2(t)$ is bounded. Since $e(t) = (1 - \mu_t)e_1(t) + \mu_t e_2(t)$, therefore, $e(t)$ is bounded.

(ii) Abnormal Case: $J_2(t)$ becomes large gradually due to the overfitting of the quasi-ARX RBFN predictor.

Since $J_1(t)$ is bounded, from Equ.(3.4.3) and (3.4.4) there exists a constant t_k such that $\eta(t) < k, \mu_t = 0,\ \forall t > t_k$, so that the model error $e(t) = e_1(t)$. Therefore, $e(t)$ is also bounded. On the other hand, since $\hat{\Theta}(t)$ is reset to the initial value Θ_0 when $\mu_t = 0$, $e_2(t)$ becomes smaller again. $J_2(t)$ gradually returns to its Normal Case (i) by the switching criterion function (3.4.1).

From above inequality (3.4.19), since $e(t)$ is bounded, the input and output of the closed-loop switching control system are bounded.

As in Ref.[76], the error $e_i(t)$, $i = 1, 2$, satisfies $\lim_{t\to\infty} \parallel e_i(t) \parallel \leq 2D$. By the switching criterion function (3.4.1), the second term determines the fuzzy switching control system, that is to say, the tracking error of the system dependent on the model error only. For the model error, we have:

$$e_2(t){=}\Delta y(t+d) - \psi^T\hat{\theta}(t) - \sum_{j=1}^{M}\Psi^T(t)\hat{\mathbf{w}}_j R_j(\Phi(t),\hat{\Omega}_j). \tag{3.4.20}$$

The linear model is always bounded. If a proper nonlinear structure is chosen and the accurate parameters is adjusted, for a predefined arbitrary small positive constant ε, $\|e_2(t)\| < \varepsilon < \delta_K \lim_{t\to\infty} \|e_1(t)\|$ can hold on. It also exists a constant T_K satisfies $\eta(t) > K, \mu_t = 1,\ \forall t > T_K$. Then the tracking error of system $\lim_{t\to\infty} \|e(t)\| (= \lim_{t\to\infty} \|e_2(t)\|)$ can converge on zero.

Remark 2: In an abnormal case, $J_2(t)$ may become large. The condition of $\hat{\Theta}(t) \in \hbar\ \forall\ t$ in Equ.(3.3.14) prevents $e_2(t)$ and $J_2(t)$ to become unbounded suddenly. On the other hand, increasing $J_2(t)$ gradually leads to $\mu_t = 0$, then $\hat{\Theta}(t)$ is reset to its initial value Θ_0 in the switching mechanism. This makes $J_2(t)$ gradually return to it normal case.

3.5 Control Simulations

In this section, we will divide into two cases to discuss the control performance.

3.5.1 Case One

In this case, we will use two example to show the effectiveness of the proposed fuzzy switching based on NN.

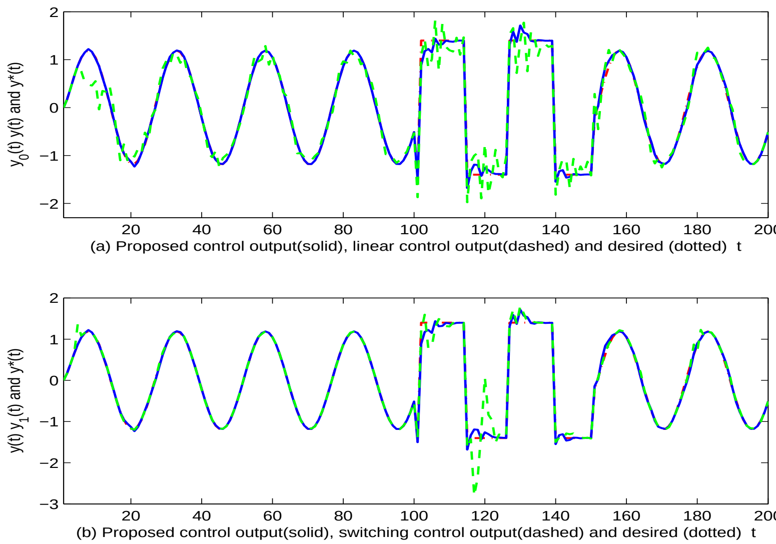

Figure 3.2: Control results for Example 1.

Example 1

Now consider a nonlinear SISO system:

$$\begin{aligned} y(t) = & \frac{\exp(-y^2(t-2)) * y(t-1)}{1+u^2(t-3)+y^2(t-2)} + \frac{(0.5*(u^2(t-2)+y^2(t-3)))*y(t-2)}{1+u^2(t-2)+y^2(t-1)} \\ & + \frac{\sin(u(t-1)*y(t-3))*y(t-3)}{1+u^2(t-1)+y^2(t-3)} + \frac{\sin(u(t-1)*y(t-2))*y(t-4)}{1+u^2(t-2)+y^2(t-2)} \\ & + u(t-1) \end{aligned} \tag{3.5.1}$$

The desired output in this example is a piecewise function:

$$y^*(t) = \begin{cases} 0.4493y^*(t-1)+0.57r(t-1) & t \in [1,100] \cup [151,200] \\ 1.4 * \mathrm{sign}(0.4493y^*(t-1)+0.57r(t-1)) & t \in [101,150] \end{cases} \tag{3.5.2}$$

where $r(t) = 1.2 * \sin(2\pi t/25)$.

To identify the system, we use the following improved quasi-ARX neural network model:

$$y(t+d) = \Psi^T(t)\theta + \Psi^T(t) \cdot W_2\Gamma(W_1\xi(t)+B). \tag{3.5.3}$$

In the nonlinear part, a NN with one hidden layer and 20 hidden nodes is used and other parameters are set as $m = 4, n = 3, d = 1$. The improved quasi-ARX model can be firstly trained off-line by the hierarchical training algorithm as in Ref.[54]. Figure 3.2-3.4 show the performance when

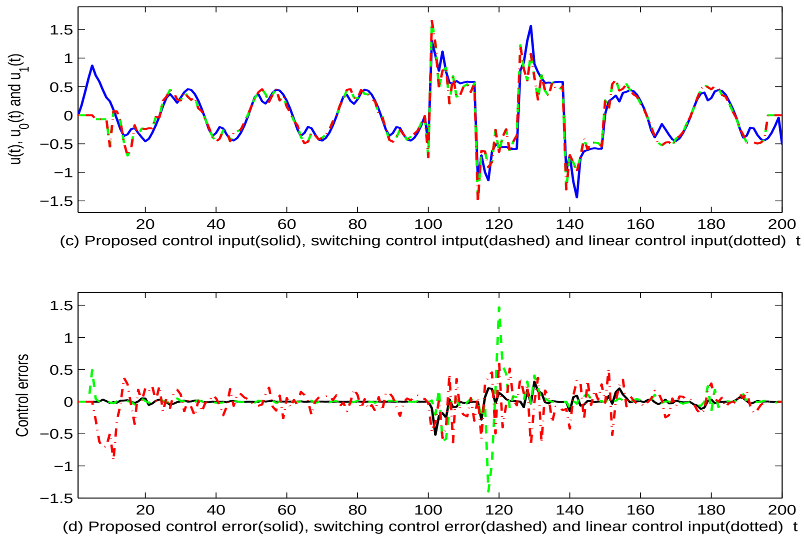

Figure 3.3: Control results for Example 1.

the proposed adaptive fuzzy switching controller is used. The parameters of switching criterion function and fuzzy membership function are chosen as $c = 1.5$, $N = 3$, $K = 0.9$ and $k = 0.1$.

Table 3.1: Comparison results of errors

	mean of errors	variance of errors
linear control	-0.0185	0.0551
switching control	0.0061	0.0365
proposed control	$0.3305 * 10^{-003}$	0.0051

In Fig.3.2(a), the red dot-solid line is the desired output, the blue solid line denotes the proposed method control output $y(t)$ and green dashed line shows the linear control output $y_0(t)$. Obviously, the control output with the proposed method is nearly consistent with the desired output at most of the time. Look at Fig.3.2(b), the red dot-solid line is the desired output and the blue solid line denotes the proposed method control output $y(t)$. The green dashed line shows the 0 or 1 switching control output $y_1(t)$. We can see that the proposed adaptive fuzzy switching controller can do better than the 0/1 switching control in two points: one is the convergence speed and the other is the adaptability. Figure 3.3(c) gives the control input where blue solid line, red dot-solid line and green

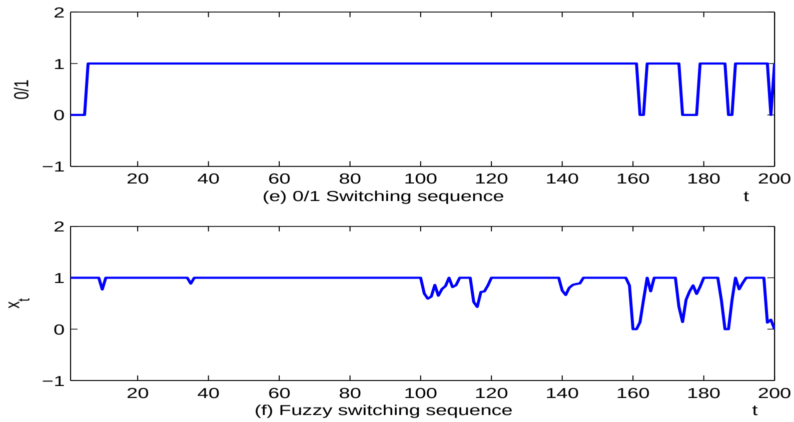

Figure 3.4: Control results for Example 1.

dashed line denotes the proposed method control input $u(t)$, the linear control input $u_0(t)$ and the switching control input $u_1(t)$, respectively. Obviously, the input signals have small fluctuation.

The errors are shown in Fig.3.3(d). Table 3.1 also gives the contrast of three methods errors. The error of the proposed control system is smaller than the other methods. The switching sequence is presented which 1 is model with nonlinear part and 0 is model without nonlinear part in Fig.3.4(e). In the Fig.3.4(f), the fuzzy switching function μ_t is shown. This figure can explain the reason of the advantage of proposed method in convergence speed and adaptive activity. The switching control use nonlinear part when $t \in [100, 150]$ and abandon nonlinear part when the value of the 0/1 switching law is 0. However, the fuzzy switching law establishes a proportion controller between linear and nonlinear control.

Example 2

The system is a nonlinear one governed by

$$y(t)=f[y(t-1), y(t-2), y(t-3), u(t-1), u(t-2)] \tag{3.5.4}$$

where

$$f[x_1, x_2, x_3, x_4, x_5] = \frac{x_1x_2x_3x_5(x_3-1)+x_4}{1+x_2^2+x_3^2}.$$

The desired output in this example is a piecewise function.

$$y^*(t) = \begin{cases} 0.6y^*(t-1)+r(t-1) & t \in [1, 100] \cup [151, 200] \\ 0.7\text{sign}(0.4493y^*(t-1)+0.57r(t-1)) & t \in [101, 150] \end{cases} \tag{3.5.5}$$

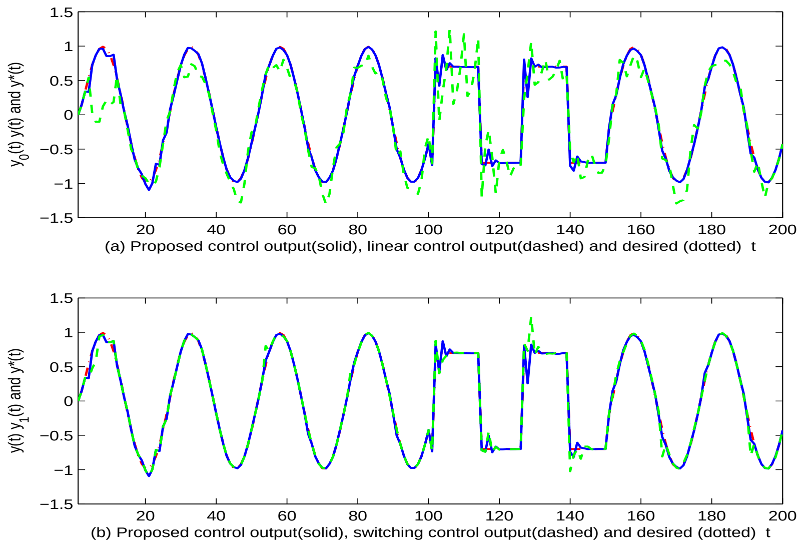

Figure 3.5: Control results for Example 2.

where $r(t) = \sin(2\pi t/25)$.

In the nonlinear part, a neural network with one hidden layer and 20 hidden nodes is used and other parameters satisfy $m = 3, n = 2, d = 1$. Figure 4.4 shows the performance when the adaptive controller is used. The parameters of switching criterion function and fuzzy switching function are chosen to be $c = 1.8$, $N = 3$, $K = 0.9$ and $k = 0.1$.

Table 3.2: Comparison results of errors

	mean of errors	variance of errors
linear control	-0.0929	0.0610
switching control	-0.0051	0.0067
proposed control	-0.0044	0.0032

Figure 3.5-3.7 give the results of Example 2 whose marks are same with Example 1. From the Fig.3.5(a), the linear control output signals have larger amplitude and far away from the desired output. However, the proposed control output is almost coincidence with the desired output. It also can be found that the switching control results have some wobble at the last half time. The similar conclusion also can be gotten from errors. The table 3.2 shows the contrast of three methods. The

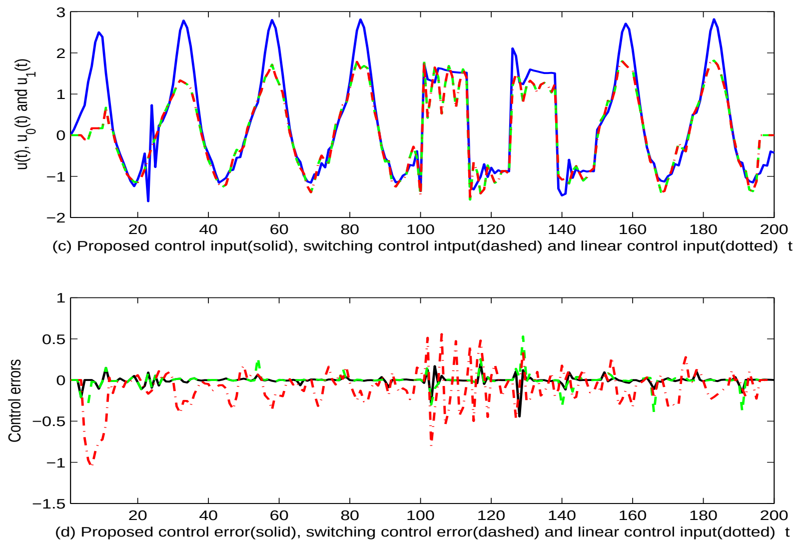

Figure 3.6: Control results for Example 2.

error of the proposed control system is smaller than the other methods. The Fig.3.7(f) shows that the fuzzy switching function is efficient.

3.5.2 Case Two

In this case, we will use two example to show the effectiveness of the proposed control method based on RBFN.

The system considered is a nonlinear one governed by

$$y(t)=g[y(t-1), y(t-2), y(t-3), u(t-1), u(t-2)] + v(t), \tag{3.5.6}$$

where $g(\cdot)$ is the nonlinear function with a disturbance:

$$g[x_1, x_2, x_3, x_4, x_5]=p_t\frac{x_1x_2x_3x_5(x_3-1)+x_4}{1+x_2^2+x_3^2} + q_t\ln(1+0.2x_4). \tag{3.5.7}$$

The two coefficients p_t and q_t of $g(\cdot)$ have a sudden change at $t = 101$, described by

$$p_t = \begin{cases} 1 & t \in [1, 100] \\ 0.99 & t \in [101, 200] \end{cases}$$

and

$$q_t = \begin{cases} 1 & t \in [1, 100] \\ 1.01 & t \in [101, 200]. \end{cases}$$

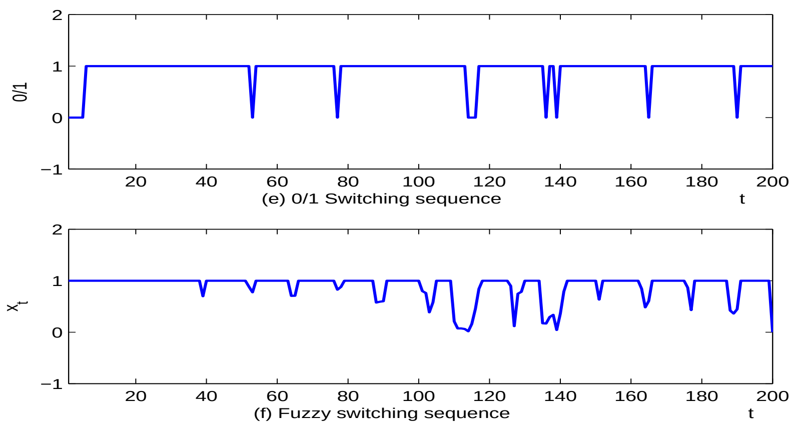

Figure 3.7: Control results for Example 2.

And $v(t)$ is the system disturbance described by $v(t) = (1+0.25q^{-1})\dot{e}(t)$ where $\dot{e}(t) \in N(0, 0.005)$ is a white noise. The desired output considered is a piecewise function, defined by

$$y^*(t) = \begin{cases} 0.6y^*(t-1) + \sin(2\pi(t-1)/25) & t \in [1, 100] \cup [151, 200] \\ 0.7\text{sign}(0.4493y^*(t-1) + 0.57\sin(2\pi(t-1)/25)) & t \in [101, 150]. \end{cases} \quad (3.5.8)$$

Note that the origin of the system is an equilibrium point, but the high-order nonlinear part is not bounded. A sudden change on the system and the system disturbance are introduced in order to show the robustness of the proposed control method.

When identifying the system, the quasi-ARX RBFN prediction model (3.3.4) is used, in which the number of RBF functions $M = 6$, the model orders $m = 3$ $n = 2$, the delay $d = 1$. And the bound of the nonlinear difference term of the system is set to $D = 0.05$.

1) Estimation of nonlinear parameter Ω_j

The nonlinear parameter vectors $\Omega_j = \{Z_j,\ \lambda_j\},\ j = 1, ..., M$ are first determined off-line. To do so, the system is excited by a random sequence with the amplitude between -1 and 1 as in Ref.[67] and 1000 input-output data set are recorded. Then an AP clustering algorithm is applied to the data set for partitioning the input space of $\xi(t) = [y(t)\ ...\ y(t-n)\ y^*(t+1)\ u(t-1)\ ...\ u(t-m)]^T$. After clustering, 6 clusters are generated automatically in the input space, so that $M = 6$. The parameter vector Z_j corresponds to the center of each cluster, while λ_j is calculated by multiplying a constant $\varrho = 0.2$ to the largest distance of the data in each cluster. The results of $\Omega_j = \{Z_j,\ \lambda_j\},\ j = 1, ..., 6$ are shown in Tab. 3.3. What should be mentioned is that the nonlinear parameters are fixed during the whole adaptive control procedure, even a sudden change occurs on the system.

2) Control without switching mechanism

Table 3.3: Estimates of parameters $\Omega_j,\ j = 1, .., 6$.

	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6
ξ_1	-0.7004	0.3719	-0.4277	-0.1317	0.4906	0.7850
ξ_2	-0.6574	-0.5189	0.5649	-0.5927	0.3156	0.7178
ξ_3	-0.7166	-0.5581	0.5334	-0.7247	0.1906	-0.1573
ξ_4	-0.7470	-0.3356	-0.3553	-0.7118	-0.4443	-0.3322
ξ_5	-0.0894	-0.3522	-0.2988	0.2550	-0.2004	-0.0555
ξ_6	-0.6441	0.4940	-0.7349	0.3924	0.5749	0.8512
ξ_7	-0.7733	-0.7251	0.7189	-0.8273	0.5151	0.8512
λ	0.0185	0.0276	0.0318	0.0233	0.0286	0.0176

Table 3.4: Comparison results of the errors

	mean of RMSEs [a]	mean of variances
fuzzy switching method	0.0147	0.047
0/1 switching method	0.0201	0.082
linear control method	0.0240	0.105

[a] Root mean spare errors (RMSEs) are calculated by RMSE(i) $= \frac{1}{T}\sqrt{\sum_{t=1}^{T}(y^i(t) - y^*(t))^2}$, where $T = 200,\ i = 1, .., 50$.

For comparison, the system is first controlled using a linear adaptive controller based on $\mathcal{C}_1$ (3.4.11). The control results are shown in Fig.3.8. Figure 3.8(a) shows the control output (solid) and the reference (dotted) and Fig. 3.8(b) shows the control error. We can see that the control result based on linear controller is not impressing and the performance needs to be improved. Then a nonlinear adaptive controller based on $\mathcal{C}_2$ (3.4.12) is applied to controlling the system. Although the control accuracy is improved, the control system converges only in 16 out of 100 trials Monte Carlo simulations. A stabilizing mechanism is required for the nonlinear adaptive control system.

3) Control with switching mechanisms

The adaptive control with a fuzzy switching mechanism described in Section 4 is applied to controlling the system. The parameters of switching criterion function and fuzzy membership function are chosen as $c = 1.5$, $N = 3$, $K = 0.9$ and $k = 0.1$. When $K = k = 0.5$ the fuzzy switching scheme reduces to a 0/1 switching scheme. Table 3.4 shows the average performance of a Monte Carlo simulation with 50 trials. The results of two switching methods and linear control method are proposed. We can see that the 0/1 switching control method gets smaller control error than the linear control method, and the fuzzy switching method improves the control performance further.

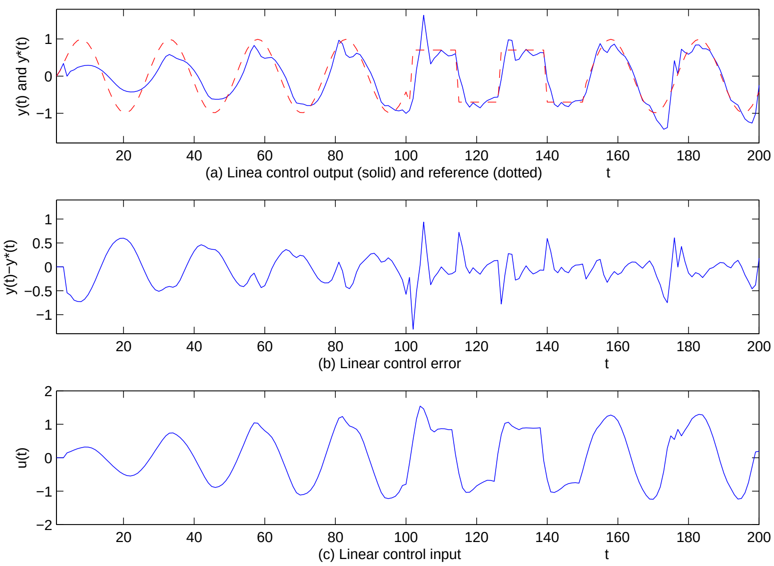

Figure 3.8: Linear control results.

Figure 3.9 shows the control results, in which the comparisons between the proposed adaptive fuzzy switching controller (3.4.10) and the 0/1 switching controller are shown. In Fig.3.9(a), the dotted line (red) is the desired output $y^*(t)$, the solid line (green) is the control output $y_1(t)$ of the proposed fuzzy switching method, the dashed line (blue) is the control output $y_2(t)$ of the 0/1 switching control method. Figure 3.9(b) gives the control errors where the solid line (green) and the dashed line (blue) denote the control errors $y_1(t) - y^*(t)$ of the proposed fuzzy switching method, the control errors $y_2(t) - y^*(t)$ of the 0/1 switching method, respectively. We can easily see that the proposed fuzzy switching method have approached a good result since $t = 10$ which is faster than the 0/1 switching method. The performance of the proposed fuzzy switching control method is better than the 0/1 switching method when $t \in [10, 100) \cup (110, 200]$, and the robustness of the proposed fuzzy switching control method is much better than the 0/1 switching method which have be illustrated since $t = 100$. Therefore, the proposed fuzzy switching method have a better control result than the contrastive control method. Figure 3.9(c) gives the control input where the solid line (green) and the dashed line (blue) denote the proposed fuzzy switching method control input $u_1(t)$, the 0/1 switching control input $u_2(t)$, respectively. Obviously, the input signals both of the proposed fuzzy switching control and switching control are smoother with very small fluctuation than linear control method.

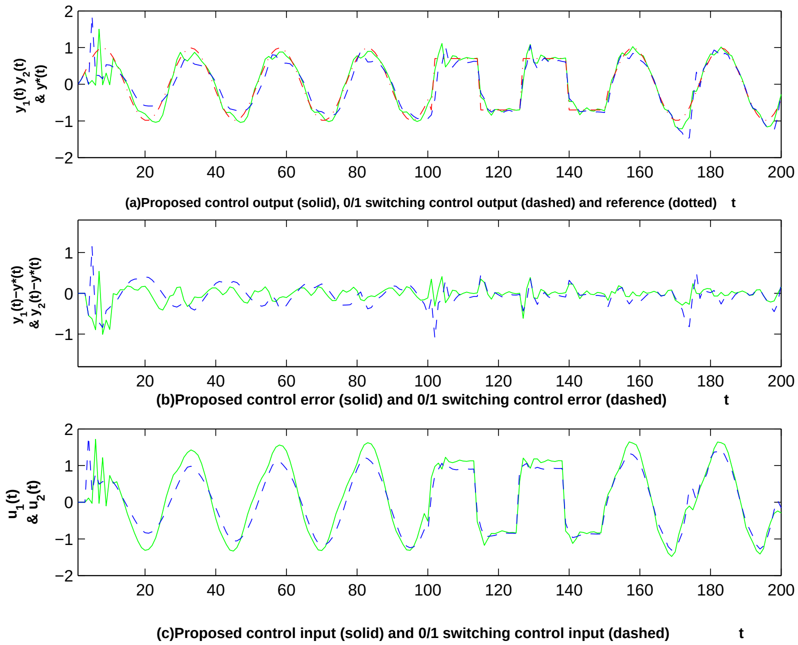

Figure 3.9: Control results for Example.

The 0/1 switching sequence are shown in Fig.3.10(a), in which 1 is the model with nonlinear part and 0 is the model without nonlinear part. In the Fig.3.10(b), the fuzzy switching function value μ_t is shown. The 0/1 switching control use nonlinear part since $t = 3$, while the nonlinear model may not be identified accurately, then it deteriorates the control convergence speed and adaptive activity. The switching control sequence changes between 0 and 1 frequently since the system have a disturbance. The switching control sequence equals to 0, and can not improve the control performance if the nonlinear model may be accurate . The proposed fuzzy control sequence $\mu_t \in [0, 1]$ when $t \in [10, 30] \cup [50, 60] \cup [120, 130] \cup [160, 180]$, improves convergence speed, performance and robustness of the control system.

Remark 3: The nonlinear parameters Z_j and λ_j $j = 1, ..., M$ of the RBFN part are determined by *a priori* and only the linear parameters $\mathbf{w}_j$ $j = 1, ..., M$ are adjusted during control process. The quasi-ARX RBFN prediction model used in the adaptive control is linear in the on-line adjustable parameters. Therefore, the proposed adaptive control system needs less response time of adaptive control and has more quick convergence speed than those using a nonlinear prediction model based on NN. The time od the proposed method is only about ten seconds and the control method based on NN is over five minutes with 200 steps.

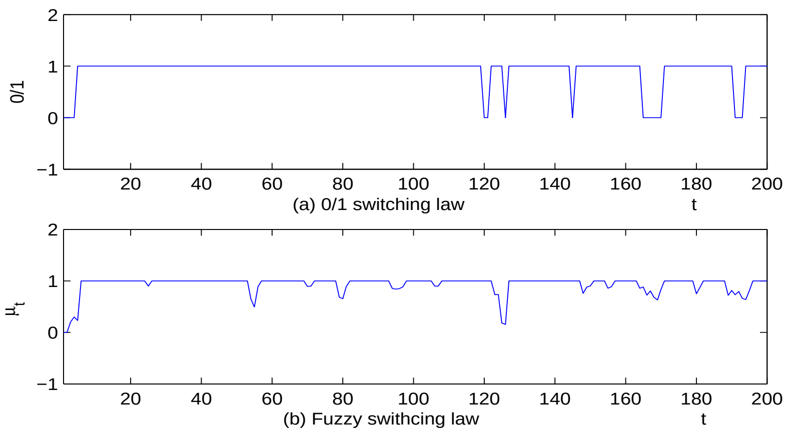

Figure 3.10: Switching sequences for Example.

3.6 Conclusions

In this chapter, a stabilizing switching controller for nonlinear system is designed based on a quasi-ARX RBFN model and a fuzzy switching function. Different with some relative works, in which more than two controllers are established and 0/1 switching mechanism is designed[71, 76], the proposed control method uses a smooth switching between a linear controller and a nonlinear controller both of which are derived from the same identified quasi-ARX RBFN prediction model. It can satisfy the stability, response and performance requirement with only one model used. A d-difference operator is used to relax the assumption of global boundedness on higher-order nonlinear terms as in Ref.[76], which improves our previous work of Refs.[20]. For parameterizing the coefficients of the macro-model, a RBFN is used in the kernel part to replace NN, thus nonlinear parameters of the proposed quasi-ARX RBFN prediction model can be determined by *a priori* knowledge, then the prediction model only remains linear parameters to be adjusted on-line. Simulations are given to show the effectiveness of the proposed method on control stability, accuracy, response and robustness.

Chapter 4

Multivariable Decoupling Control of Nonlinear MIMO Systems Based on MIMO Quasi-ARX Model

4.1 Introduction

Nonlinear system control has become a considerable topic in the field of control engineering. Many control results have been obtained for nonlinear Single-Input and Single-Output (SISO) systems based on the black box models, such as Neural Networks (NNs), Wavelet Networks (WNs), Neuro-Fuzzy Networks (NFNs) and Radial Basis Function Networks (RBFNs), because of their abilities to approximate arbitrary mapping to any desired accuracy[66, 20, 21, 22, 23, 24, 25]. These black box models have been directly used to identify and control nonlinear dynamical systems.

Due to the complexity of nonlinear Multi-Input and Multi-Output (MIMO) systems, most of the control techniques developed for SISO systems cannot be extended directly for MIMO systems. One of the main difficulties in MIMO nonlinear system control is coupling problem. As such, it is important to investigate the realization of decoupling control. Many adaptive decoupling control algorithms have been proposed to deal with coupling in nonlinear system based on linear methods and nonlinear networks [91, 92, 93, 94, 76]. Some decoupling control methods of them are difficult not only to achieve accurate requirement and stability, but also to be implemented in industrial applications. On the other hand, PID controller has been widely applied in controlling the SISO system because of its simple structure and relatively easy industrial application[95, 96]. However, PID controller can not be directly used for MIMO model. Lang, Gu & Chai[97] proposed a multivariable decoupling PID controller for MIMO linear systems based on the linear PID control and generalized minimum variance control law. What's more, Zhai & Chai[98] presented a multivariable PID control method using neural network to deal with nonlinear multivariable processes. In this control system, the nonlinear unmodeled part estimated by neural network is considered as a black box. The initial weights of neural network, local minima and overfitting are the problems which need to be resolved.

In our previous work, a quasi-AutoRegressive eXogenous (ARX) model with an ARX-like macro model part and a kernel part was proposed, and a controller was designed for SISO systems [72, 54, 20, 99]. The kernel part is an ordinary network model, but it is used to parameterize the nonlinear coefficients of macro model. As we know, RBFNs have played an important role in control engineering, especially in nonlinear system control because of their simple topological structure and precision in nonlinear approximation [88, 89]. Especially, RBFNs can be regarded as nonlinear models which are linear in parameters when fixing the nonlinear parameters by *a priori* knowledge[86, 87]. Incorporating the network models with this property, the quasi-ARX models become linear-in-parameters. Therefore, the RBFNs are chosen to replace the NNs as in [20].

The SISO model and control methods based on quasi-ARX model can not be directly applied to MIMO nonlinear systems. Motivated by the above discussions, an MIMO quasi-ARX model is first proposed for MIMO nonlinear systems and then a nonlinear multivariable decoupling PID controller is proposed based on the MIMO quasi-ARX model, which consists of a traditional PID controller with a decoupling compensator and a feed-forward compensator for the nonlinear dynamics based on the MIMO quasi-ARX model. Then an adaptive controller is presented using the MIMO quasi-ARX RBFN prediction model. The parameters of such controller is selected based on the generalized minimum control variance. In this paper, quasi-ARX RBFN model is divided into two parts: the linear part is used to guarantee the stability and decoupling, and the nonlinear part is used to improve the accuracy.

The chapter is organized as follows: Section 4.2 describes the nonlinear MIMO system considered, and then a hybrid system expression is obtained and an MIMO quasi-ARX RBFN model is proposed. In Section 4.3, a multivariable decoupling PID controller is got based on the proposed model and generalized minimum variance control law. Then an adaptive control algorithm is presented using the MIMO quasi-ARX RBFN prediction model and the corresponding parameter estimation methods are proposed in Section 4.4. Section 4.5 carries out numerical simulations to show the effectiveness of the proposed control method. Finally, Section 4.6 presents the conclusions.

4.2 An MIMO Quasi-ARX Model

4.2.1 Systems

Consider an MIMO nonlinear dynamical system with input-output relation as:

$$\begin{aligned} \boldsymbol{y}(t+d) &= \boldsymbol{f}(\varphi(t)), \\ \varphi(t) &= [\boldsymbol{y}(t+d-1)^T, ..., \boldsymbol{y}(t+d-n_y)^T, \boldsymbol{u}(t)^T, ..., \boldsymbol{u}(t-n_u+1)^T]^T \end{aligned} \tag{4.2.1}$$

where $\boldsymbol{y} = [y_1, ..., y_n]^T \in R^n$ and $\boldsymbol{u} = [u_1, ..., u_n]^T \in R^n$ are system input and output vectors, respectively, d the known integer time delay, $\varphi(t)$ the regression vector, and n_y , n_u the system orders. $\boldsymbol{f}(\cdot) = [f_1(\cdot), ..., f_n(\cdot)]^T$ is a vector-valued nonlinear function, and at a small region around $\varphi(t) = \mathbf{0}$ ($\mathbf{0} = [0, ..., 0]^T$), they are C^∞ continuous. The origin is an equilibrium point, then $\boldsymbol{f}(\mathbf{0}) = \mathbf{0}$. The system is controllable, in which a reasonable unknown controller may be expressed by $\boldsymbol{u}(t) = \rho(\xi(t))$, where $\xi(t)$ is defined in Section (4.2.4).

4.2.2 ARX-Like Expression

Under the continuous condition, the unknown nonlinear function $f_k(\varphi(t)), \ (i = 1, ..., n)$ can be performed Taylor expansion on a small region around $\varphi(t) = \mathbf{0}$:

$$y_k(t+d) = f_k'(0)\varphi(t) + \frac{1}{2}\varphi^T(t) f_k''(0)\varphi(t) + \ ... \tag{4.2.2}$$

where the prime denotes differentiation with respect to $\varphi(t)$. Then the following notations are introduced:

$$(f_k'(0) + \frac{1}{2}\varphi^T(t) f_k''(0) + \cdots)^T = [a_{1,t}^{1,k} \ ...a_{n_y,t}^{1,k} \ ... \ a_{n_y,t}^{n,k} \ b_{1,t}^{1,k} \ ...b_{n_u,t}^{1,k} \ ... \ b_{n_u,t}^{n,k}]^T$$

where $a_{i,t}^{l,k} = a_i^{l,k}(\varphi(t)) \ (i = 1, .., n_y)$ and $b_{j,t}^{l,k} = b_j^{l,k}(\varphi(t)) \ (j = 0, ..., n_u - 1)$ are nonlinear functions of $\varphi(t)$.

However, we need to get $\boldsymbol{y}(t+d)$ by using the input-output data up to time t in a model. The coefficients $a_{i,t}^{l,k}$ and $b_{j,t}^{l,k}$ need to be calculable using the input-output data up to time t. To do so, let us iteratively replace $y(t+l)$ in the expressions of $a_{i,t}^{l,k}$ and $b_{j,t}^{l,k}$ with functions:

$$\boldsymbol{y}(t+s) \Rightarrow \boldsymbol{g}(\tilde{\varphi}(t+s)), \ s = 1, ..., d-1 \tag{4.2.3}$$

where $\tilde{\varphi}(t+s)$ is $\varphi(t+s)$ whose elements $\boldsymbol{y}(t+m), \ s+1 < m \leq d-s$ are replaced by Equ.(4.2.3), and define the new expressions of the coefficients by:

$$a_{i,t}^{l,k} = \tilde{a}_{i,t}^{l,k} = \tilde{a}_i^{l,k}(\phi(t)), \quad b_{j,t}^{l,k} = \tilde{b}_{j,t}^{l,k} = \tilde{b}_j^{l,k}(\phi(t))$$

where $\phi(t)$ is a vector:

$$\phi(t) = [\boldsymbol{y}(t)^T \ldots \boldsymbol{y}(t-n_y+1)^T\, \boldsymbol{u}(t)^T \ldots \boldsymbol{u}(t-n_u-d+2)^T]^T. \tag{4.2.4}$$

Now, introduce two polynomial matrices $\boldsymbol{A}(q^{-1},\phi(t))$ and $\boldsymbol{B}(q^{-1},\phi(t))$ based on the coefficients, defined by:

$$\begin{aligned}\boldsymbol{A}(q^{-1},\phi(t)) &= \boldsymbol{I} - \boldsymbol{a}_{1,t}q^{-1} - \ldots - \boldsymbol{a}_{n_y,t}q^{-n_y};\\ \boldsymbol{B}(q^{-1},\phi(t)) &= \boldsymbol{b}_{0,t} + \ldots + \boldsymbol{b}_{n_u-1,t}q^{-n_u+1},\end{aligned}$$

where $\boldsymbol{a}_{i,t} = (a^{l,k}_{i,t})_{N\times N},\ i=1,..,n_y$ and $\boldsymbol{b}_{j,t} = (b^{l,k}_{j,t})_{N\times N},\ j=1,..,n_u$. Then, the nonlinear system (4.2.1) can be equivalently represented as the following ARX-like expression:

$$\boldsymbol{A}(q^{-1},\phi(t))\boldsymbol{y}(t+d) = \boldsymbol{B}(q^{-1},\phi(t))\boldsymbol{u}(t). \tag{4.2.5}$$

By the Equ.(4.2.5), let $\boldsymbol{y}(t+d)$ satisfies the following equation:

$$\boldsymbol{y}(t+d) = \mathcal{A}(q^{-1},\phi(t))\boldsymbol{y}(t) + \mathcal{B}(q^{-1},\phi(t))\boldsymbol{u}(t), \tag{4.2.6}$$

where

$$\mathcal{A}(q^{-1},\phi(t)) = A_{0,t} + A_{1,t}q^{-1} + \ldots + A_{n_y-1,t}q^{-n_y+1}, \tag{4.2.7}$$

$$\begin{aligned}\mathcal{B}(q^{-1},\phi(t)) &= \boldsymbol{F}(q^{-1},\phi(t))\boldsymbol{B}(q^{-1},\phi(t)),\\ &= B_{0,t} + B_{1,t}q^{-1} + \ldots + B_{n_u+d-2,t}q^{-n_u-d+2},\end{aligned} \tag{4.2.8}$$

$A_{i,t}(i=0,\ldots,n_y-1)$ and $B_{j,t}(j=0,\ldots,n_u+d-2)$ are coefficient matrices. And $\boldsymbol{G}(q^{-1},\phi(t))$, $\boldsymbol{F}(q^{-1},\phi(t))$ are unique polynomials satisfying:

$$\boldsymbol{F}(q^{-1},\phi(t))\boldsymbol{A}(q^{-1},\phi(t)) = \boldsymbol{I} - \mathcal{A}(q^{-1},\phi(t))q^{-d}. \tag{4.2.9}$$

4.2.3 Hybrid Expression

The coefficients matrices $A_{i,t}$ ($i=0,\ldots,n_y-1$) and $B_{j,t}$ ($j=0,\ldots,n_u+d-2$) can be considered as a summation of two parts: the constant part A^l_i and B^l_j, and the nonlinear function part on $\phi(t)$ which are denoted $A^n_{i,t}$ and $B^n_{i,t}$. Then, the expression of system in the predictor form (4.2.6) can be described by:

$$\boldsymbol{y}(t+d) = \mathcal{A}^l(q^{-1})\boldsymbol{y}(t) + \mathcal{B}^l(q^{-1})\boldsymbol{u}(t) + \mathcal{A}^n(q^{-1},\phi(t))\boldsymbol{y}(t) + \mathcal{B}^n(q^{-1},\phi(t))\boldsymbol{u}(t) \tag{4.2.10}$$

where

$$\mathcal{A}^l(q^{-1}) = A_0^l + A_1^l q^{-1} + ... + A_{n_y-1}^l q^{-n_y+1};$$
$$\mathcal{A}^n(q^{-1}, \phi(t)) = A_{0,t}^l + A_{1,t}^l q^{-1} + ... + A_{n_y-1,t}^l q^{-n_y+1};$$
$$\mathcal{B}^l(q^{-1}) = B_0^l + B_1^l q^{-1} + ... + B_{n_y-d+2}^l q^{-n_u+d-2};$$
$$\mathcal{B}^n(q^{-1}, \phi(t)) = B_{0,t}^l + B_{1,t}^l q^{-1} + ... + B_{n_y-d+2,t}^l q^{-n_u+d-2};$$

Similar with Ref.[98], the linear polynomial matrix $\mathcal{B}^l(q^{-1})$ can be expressed as $\mathcal{B}^l(q^{-1}) = \bar{\mathcal{B}}^l(q^{-1}) + \bar{\bar{\mathcal{B}}}^l(q^{-1})$ with $\bar{\mathcal{B}}^l(q^{-1})$ being diagonal and $\bar{\bar{\mathcal{B}}}^l(q^{-1})$ being a polynomial matrix with zero diagonal elements.

Then, the linear and nonlinear expression of system (4.2.10) can be obtained as:

$$\begin{aligned} \boldsymbol{y}(t+d) = & \mathcal{A}^l(q^{-1})\boldsymbol{y}(t) + \bar{\mathcal{B}}^l(q^{-1})\boldsymbol{u}(t) + \bar{\bar{\mathcal{B}}}^l(q^{-1})\boldsymbol{u}(t) \\ & + \mathcal{A}^n(q^{-1}, \phi(t))\boldsymbol{y}(t) + \mathcal{B}^n(q^{-1}, \phi(t))\boldsymbol{u}(t) \end{aligned} \tag{4.2.11}$$

4.2.4 Quasi-ARX RBFN Model

Now, we will propose an MIMO quasi-ARX RBFN model. However, the $\boldsymbol{v}(\phi(t))$ are based on $\Psi(t)$ whose elements contain $\boldsymbol{u}(t)$. To solve this problem, an *extra variable* $\boldsymbol{x}(t)$[1] is introduced and an unknown nonlinear function $\rho(\xi(t))$ is used to replace the variable $\boldsymbol{u}(t)$ in $\phi(t)$, Under **a**ssumption, the function $\rho(\xi(t))$ exists. Define:

$$\xi(t) = [\boldsymbol{y}(t)^T \; ... \; \boldsymbol{y}(t-n_1)^T \; \boldsymbol{x}(t+d)^T \; ...\boldsymbol{x}(t-n_3+d)^T \; \boldsymbol{u}(t-1)^T \; ... \; \boldsymbol{u}(t-n_2)^T]^T$$

including the extra variable $\boldsymbol{x}(t+d)$ as an element. A typical choice for $n_1,\ n_2$, and n_3 in $\xi(t)$ is $n_1 = n_y - 1$, $n_2 = n_u + d - 2$ and $n_3 = 0$. We can express the Equ.(4.2.11) by:

$$\boldsymbol{y}(t+d) = \psi^T(t)\Omega_0 + \xi^T(t)\theta_\xi^n, \tag{4.2.12}$$

where $\psi^T(t) = \varphi(t-d)$ and $\Omega_0 = [A_0^l, ..., A_{n_y-1}^l, B_0^l, ..., B_{n_y-d+2}^l]$. The elements of θ_ξ^n are unknown nonlinear function of $\xi(t)$, which can be parameterized by NN or RBFN. In this chapter, the RBFN is used which has local property:

$$\theta_\xi^n = \sum_{j=1}^{M} \Omega_j \mathcal{R}_j(\mathbf{p}_j, \xi(t)), \tag{4.2.13}$$

[1]Obviously, in a control system, the reference signal $\boldsymbol{y}^*(t+d)$ can be used as the extra variable $\boldsymbol{x}(t+d)$.

where M is the number of RBFs. $\Omega_j = [\Omega_{j,1}, ..., \Omega_{j,n}]$ is the coefficient matrix with $\Omega_{j,i} = [\omega_{j,i}^1, ..., \omega_{j,i}^N]^T$, $j = 1, ..., M$. And $\mathcal{R}_j(\xi(t), \Omega_j)$ the RBFs defined by:

$$\mathcal{R}_j(\mathbf{p}_j, \xi(t)) = e^{-\sigma_j \|\xi(t) - \mathbf{X}_j\|^2}, j = 1, 2, ..., M, \tag{4.2.14}$$

where $\mathbf{p}_j = \{\sigma_j, \mathbf{X}_j\}$ is the parameters set of the RBFN; X_j is the center vector of RBF and σ_j are the scaling parameters; $\| \bullet \|_2$ denotes the vector two-norm. Then we can express the quasi-ARX RBFN prediction model for (4.2.12) in a form of:

$$\boldsymbol{y}(t+d) = \psi^T(t)\Omega_0 + \xi^T(t) \sum_{j=1}^{M} \Omega_j \mathcal{R}_j(\mathbf{p}_j, \xi(t)), \tag{4.2.15}$$

4.3 Controller Design

4.3.1 Nonlinear Multivariable Decoupling PID Controller

Introduce the following performance index:

$$\mathbf{M}(t+d) = \|\boldsymbol{y}(t+d) - \boldsymbol{R}(q^{-1})\boldsymbol{y}^*(t+d) + \boldsymbol{S}(q^{-1})\boldsymbol{u}(t) + \boldsymbol{Q}(q^{-1})\boldsymbol{u}(t)\|, \tag{4.3.1}$$

where $\boldsymbol{R}$ and $\boldsymbol{S}$ are the diagonal weighting polynomial matrices, and $\boldsymbol{Q}$ is a weighting polynomial matrix with diagonal elements.

The optimal control law minimizing (4.3.1) is:

$$\boldsymbol{y}(t+d) - \boldsymbol{R}(q^{-1})\boldsymbol{y}^*(t+d) + \boldsymbol{S}(q^{-1})\boldsymbol{u}(t) + \boldsymbol{Q}(q^{-1})\boldsymbol{u}(t) = 0 \tag{4.3.2}$$

Substituting (4.2.11) into (4.3.2), the following equation is obtained:

$$\begin{aligned}(\bar{\mathcal{B}}^l(q^{-1}) + \boldsymbol{Q}(q^{-1}))\boldsymbol{u}(t) = \boldsymbol{R}(q^{-1})\boldsymbol{y}^*(t+d) - \mathcal{A}^l(q^{-1})\boldsymbol{y}(t) - (\bar{\bar{\mathcal{B}}}^l(q^{-1}) + \boldsymbol{S}(q^{-1}))\boldsymbol{u}(t) \\ - (\mathcal{B}^n(q^{-1}, \phi(t))\boldsymbol{u}(t) + \mathcal{A}^n(q^{-1}, \phi(t))\boldsymbol{y}(t)).\end{aligned} \tag{4.3.3}$$

where $\bar{\mathcal{B}}^l(q^{-1}) + \boldsymbol{Q}(q^{-1}) = \lambda^{-1}\bar{\boldsymbol{H}}(q^{-1})$, with $\lambda = \textbf{diag}\{\lambda_1, ..\lambda_n\}$ and $\bar{\boldsymbol{H}}(q^{-1}) = (1 - q^{-1}) \cdot \boldsymbol{I}$. By introducing $\boldsymbol{R}(q^{-1}) = \mathcal{A}^l(q^{-1})$ and $\bar{\mathcal{B}}^l(q^{-1})\boldsymbol{S}(q^{-1}) = \boldsymbol{Q}(q^{-1})\bar{\bar{\mathcal{B}}}^l(q^{-1})$. when $n_y - 1 \leq 2$, a nonlinear decoupling PID controller is obtained, similar to a traditional PID controller:

$$\bar{\boldsymbol{H}}(q^{-1})\boldsymbol{u}(t) = \lambda\mathcal{A}^l(q^{-1})\boldsymbol{e}(t) - \bar{\bar{\boldsymbol{H}}}(q^{-1})\boldsymbol{u}(t) - \boldsymbol{v}(\phi(t)). \tag{4.3.4}$$

where $\bar{\bar{\boldsymbol{H}}}(q^{-1}) = \lambda(\bar{\bar{\mathcal{B}}}^l(q^{-1}) + \boldsymbol{S}(q^{-1}))$ and $\boldsymbol{v}(\phi(t)) = \lambda(\mathcal{B}^n(q^{-1}, \phi(t))\boldsymbol{u}(t) + \mathcal{A}^n(q^{-1}, \phi(t))\boldsymbol{y}(t))$. $\boldsymbol{e}(t) = \boldsymbol{y}^*(t+d) - \boldsymbol{y}(t)$.

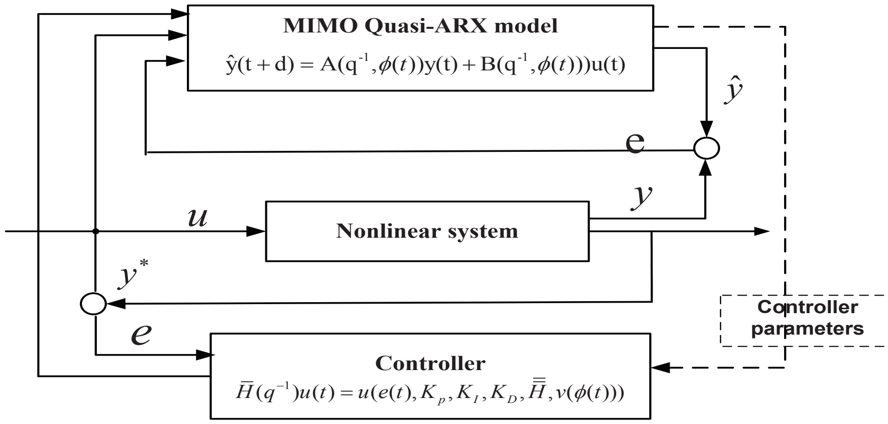

Figure 4.1: The multivariable decoupling PID control system based on MIMO quasi-ARX model.

The controller (4.3.4) is substituted into the system (4.3.2), the obtained closed-loop system which is show in Fig.4.1 will be stable, and the decoupling control effect and tracking errors can be eliminated.

A velocity-type form of the PID controller is given:

$$\begin{aligned}\bar{\boldsymbol{H}}(q^{-1})\boldsymbol{u}(t)=&\boldsymbol{K}_p(\boldsymbol{e}(t)-\boldsymbol{e}(t-1))+\boldsymbol{K}_I\boldsymbol{e}(t)+\boldsymbol{K}_D(\boldsymbol{e}(t)-2\boldsymbol{e}(t-1)+\boldsymbol{e}(t-2))\\&-\bar{\bar{\boldsymbol{H}}}(q^{-1})\boldsymbol{u}(t)-\boldsymbol{v}(\phi(t)).\end{aligned}\tag{4.3.5}$$

The gain can be selected as:

$$\begin{aligned}\boldsymbol{K}_p=&-\lambda(2A_2+A_1),\\\boldsymbol{K}_I=&\lambda(A_0+A_1+A_2),\\\boldsymbol{K}_D=&\lambda A_2.\end{aligned}\tag{4.3.6}$$

where when $n_y=1$, $A_1=A_2=0$, and when $n_y=2$, $A_2=0$.

4.3.2 Parameter Estimation

Determining $\mathbf{p}_j$ Using Knowledge Information

As mentioned earlier, we need the model is simplicity and flexibility simultaneously during the modeling. However, the uncertain parameters $\mathbf{p}_j$ increases the overall flexibility of model and then restricts the flexibility in the higher order nonlinearity. Then, the scale and position parameters $\mathbf{p}_j$ of the basis function in the RBFN is determined using knowledge information. It is assumed that the physical insight of the control plant is not available in a black-box modeling. Then, the prior knowledge information are mainly got from the obtained data and the errors. Some kinds of knowledge information can be used as follows:

- the information about the operating region of $\varphi(t)$ which can be got easily from the observed data.
- the information concerning the structure of the nonlinear part which can be obtained by using various linear models to identify the system.
- the information about the relations among the elements in $\xi(t)$. This information can be known when $\xi(t)$ is chosen.
- the information concerning the size of the prediction errors and their relations with the region of $\xi(t)$ which may be got during the estimation.

A: A strategy for Determining $\mathbf{p}_j$

Now propose a method to initialize $\mathbf{p}_j$ and the following strategy is not only suitable for RBFN, but also suitable for NFN and B-spline based models.

Denotes as follows: $\mathbf{p}_j = [\bar{x}_1^j \ \bar{x}_2^j \ ... \ \bar{x}_N^j, \sigma_j]^T \ \ (j = 1, ..., M)$. $N = \text{dim}(\xi(t))$ is the dimension of the inputs. $\xi = [x_i, i = 1, ..., N]$ and the inputs region is mostly located in $\boldsymbol{X}_{\min} \leq \xi \leq \boldsymbol{X}_{\max}$, $\boldsymbol{X}_{\min} = [x_{i \ \min}, i = 1, ..., N]$, $\boldsymbol{X}_{\max} = [x_{i \ \max}, i = 1, ..., N]$. The *nodes* are put into the input hyperplane as shown in Fig.4.2. If the number of nodes corresponding to x_i is denoted as n_i, the total number of the nodes will be $M = \Pi_{i=1}^N n_i$. Then, the parameters $\mathbf{p}_j$ are chosen so that the function $\mathcal{R}(\mathbf{p}_j, \xi(t))$ have appropriate shape and are put onto every nodes. A schematic diagram for determining $\mathbf{p}_j$ for RBFN with $N = 2$ and $M = 4 \times 3$ is shown in Fig.4.2.

B: Several Hints for Reducing M

The prior knowledge about the region $[\boldsymbol{X}_{\min}, \boldsymbol{X}_{\max}]$ is the least information for determining the scale and position parameters $\mathbf{p}_j$. However, when N is very large, the number M may be rather large. Therefore, more obtained information can reduce the number of nodes or improve the node assignment. The hints is given as follows:

- Hint A: If the system is linear in x_i, n_i can be equal to 1.
- Hint B: if no more information, we can assign n_1 and n_{n+1} corresponding to $y(t-1)$ and $u(t-1)$ with appropriate values, while set all other n_i to 1.

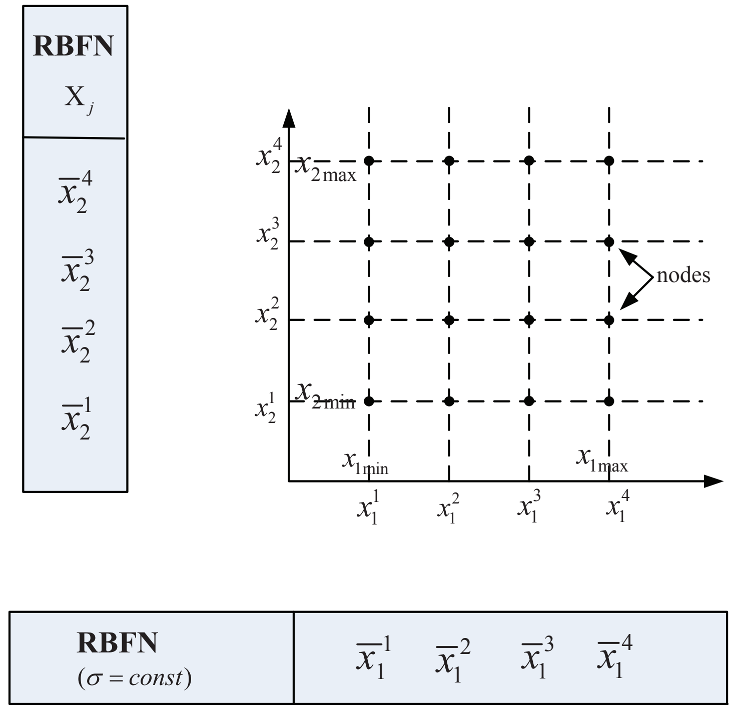

Figure 4.2: A schematic diagram for determining $\mathbf{p}_j$ for RBFN.

- Hint C: The nodes which can be replaced by employing interpolation of NNMS may be removed from the hyperplane.

Estimation of Parameter Vectors Ω_0

If the process is known, Ω_0 is obtained by using Talyor expansion at its equilibrium; otherwise, it can be replaced by its estimations $\hat{\Omega}_0$.

Estimation of Parameter Vectors Ω_j

Parameter vectors $\Omega_j,\ (j = 1, ..., M)$ can be estimated by simplified multivariable Least-Squares algorithm as in Ref.[2]. Now, introduce the notations:

$$\Omega = [\Omega_1^T, ..., \Omega_M^T]^T,\ \Phi(t) = [\xi(t)^T \otimes \Psi_{\mathcal{R}}^T(t)]^T, \tag{4.3.7}$$

where the symbol $\otimes$ denotes Kronecker production and $\Psi_{\mathcal{R}}^T(t) = [\mathcal{R}_j(\mathbf{p}_j, \xi(t)),\ j = 1, ..., M]$, the MIMO quasi-ARX model (4.2.10) can be expressed in a like-linear regression form:

$$\boldsymbol{y}(t+d) = \psi^T(t)\Omega_0 + \Phi^T(t)\Omega. \tag{4.3.8}$$

The parameter Ω is updated by a LS algorithm while fixing $\mathbf{p}_j$ and Ω_0:

$$\hat{\Omega}(t) = \hat{\Omega}(t-d) + \frac{P(t)\Phi(t-d)\boldsymbol{e}(t)}{1+\Phi(t-d)^T P(t)\Phi(t-d)}, \tag{4.3.9}$$

where $\hat{\Omega}(t)$ is the estimate of Ω at time instant t. $\mathbf{e}(t)$ is the error vector of MIMO quasi-ARX model, defined by

$$\mathbf{e}(t) = \boldsymbol{y}(t) - \psi^T(t)\Omega_0 - \Phi(t-d)^T\hat{\Omega}(t-d). \tag{4.3.10}$$

And

$$P(t) = \frac{P(t-d) - P^T(t-d)\Phi(t-d)^T\Phi(t-d)P(t-d)}{1+\Phi(t-d)^T P(t)\Phi(t-d)}. \tag{4.3.11}$$

4.4 Stability Analysis

There are some assumption made:

Assumption 1: (i) $y^*(t)$ ia a bounded deterministic sequence; (ii) $v(\phi(t))$ is globally bounded, $|v(\phi(t))| \le \Delta$, where the boundary Δ is known; (iii) The choices of λ and $\boldsymbol{S}(q^{-1})$ are such that det$\{\tilde{\boldsymbol{H}}(q^{-1})\boldsymbol{A}(q^{-1}) + q^{-d}\tilde{\boldsymbol{B}}(q^{-1})\lambda\mathcal{A}^l(q^{-1}))\boldsymbol{y}(t+d)\} \neq 0$.

Theorem For the MIMO nonlinear system (4.2.1) with the controller (4.3.5), together with the parameters of the controller selected by Sec.(4.2), all the signals in the closed-loop system described above can be bounded, and the tracking error can be made less than any specified constant δ over a compact set by properly choosing the structures and parameters of quasi-ARX RBFN model, that is $\lim_{t\to\infty} \|\boldsymbol{y}(t+d) - \boldsymbol{y}^*(t+d)\| \le \varepsilon$.

Proof. The nonlinear part estimation error vector can be described by:

$$\varepsilon(t) = \boldsymbol{v}(\phi(t+d)) - \xi^T(t+d)\sum_{j=1}^{M}\hat{\Omega}(t+d)_j\mathcal{R}_j(\mathbf{p}_j, \xi(t+d)). \tag{4.4.1}$$

We can see that, if the nonlinear decoupling PID controller (4.3.5) is used to the system (4.2.11), the following input-output dynamics are obtained as in Ref.[98]:

$$\begin{aligned}(\tilde{\boldsymbol{H}}(q^{-1})\boldsymbol{A}(q^{-1}) &+ q^{-d}\tilde{\boldsymbol{B}}(q^{-1})\lambda\mathcal{A}^l(q^{-1}))\boldsymbol{y}(t+d) \qquad (4.4.2)\\ &= \tilde{\boldsymbol{B}}(q^{-1})\lambda\mathcal{A}^l(q^{-1})\boldsymbol{y}^*(t+d) + \tilde{\boldsymbol{H}}(q^{-1})\boldsymbol{v}(\phi(t+d)) - \tilde{\boldsymbol{B}}(q^{-1})\hat{\boldsymbol{v}}(\phi(t+d)),\\ (\boldsymbol{A}(q^{-1})\boldsymbol{H}(q^{-1}) &+ q^{-d}\lambda\boldsymbol{A}(q^{-1})\mathcal{A}^l(q^{-1}))\boldsymbol{u}(t+d) \qquad (4.4.3)\\ &= \tilde{\boldsymbol{A}}(q^{-1})\lambda\mathcal{A}^l(q^{-1})\boldsymbol{y}^*(t+d) - q^{-d}\lambda\mathcal{A}^l(q^{-1})\boldsymbol{v}(\phi(t+d)) - \boldsymbol{A}(q^{-1})\hat{\boldsymbol{v}}(\phi(t+d)),\end{aligned}$$

Substitute (4.4.1) into (4.4.2) and (4.4.3), the equations are given as follows:

$$(\tilde{\boldsymbol{H}}(q^{-1})\boldsymbol{A}(q^{-1})+q^{-d}\tilde{\boldsymbol{B}}(q^{-1})\lambda G(q^{-1}))\boldsymbol{y}(t+d)=\tilde{\boldsymbol{B}}(q^{-1})\lambda G(q^{-1})\boldsymbol{y}^{*}(t+d) \tag{4.4.4}$$
$$+(\tilde{\boldsymbol{H}}(q^{-1})-\tilde{\boldsymbol{B}}(q^{-1}))\boldsymbol{v}(\phi(t+d))+\tilde{\boldsymbol{B}}(q^{-1})\varepsilon(t),$$
$$(\boldsymbol{A}(q^{-1})\boldsymbol{H}(q^{-1})+q^{-d}\lambda\boldsymbol{A}(q^{-1})\mathcal{A}^{l}(q^{-1}))\boldsymbol{u}(t+d)=\tilde{\boldsymbol{A}}(q^{-1})\lambda\mathcal{A}^{l}(q^{-1})\boldsymbol{y}^{*}(t+d) \tag{4.4.5}$$
$$-(q^{-d}\lambda\mathcal{A}^{l}(q^{-1})+\boldsymbol{A}(q^{-1}))\boldsymbol{v}(\phi(t+d))-\boldsymbol{A}(q^{-1})\varepsilon(t),$$

From (4.4.4), (4.4.5) and Assumption 1, there exist constants C_1,C_2,C_3,C_4 satisfying:

$$\|\boldsymbol{y}(t+d)\|\leq C_1+C_2\max_{0\leq\tau\leq t}\|\varepsilon(t)\|, \tag{4.4.6}$$
$$\|\boldsymbol{u}(t)\|\leq C_3+C_4\max_{0\leq\tau\leq t}\|\varepsilon(t)\|, \tag{4.4.7}$$

Because the universal approximations of the RBFNs, the estimation error $\varepsilon(t)$ can be achieved less than any constant ζ over a compact set by properly choosing their structures and parameters. It can be got that:

$$\|\varphi(t+d)\|\leq C_5+C_6\max_{0\leq\tau\leq t}\|\varepsilon(t)\|\leq C_7+C_8\zeta\leq C_9. \tag{4.4.8}$$

where C_5,C_6,C_7,C_8,C_9 constants.

Then, the boundness of all the signals in the closed-loop system is got.

The tracking error of the system is obtained as:

$$e(t)=\lim_{t\to\infty}\|\boldsymbol{y}(t+d)-\boldsymbol{y}^{*}(t+d)\|\leq C \tag{4.4.9}$$

where $C>0$ is a constant.

4.5 Numerical Simulations

In order to study the behavior of the proposed control method, some numerical simulations are described in this section.

4.5.1 Case One

The MIMO nonlinear system to be controlled is described by:

$$\begin{aligned}
y_1(t+1) =& 0.9y_1(t) - \frac{0.3y_1(t-1)}{1+y_2^2(t-1)} + 0.4\sin(u_1(t)) \\
& +0.7u_1(t-1) + 0.3u_2(t) - 0.5u_2(t-1) \\
& +v_1(t) \\
y_2(t+1) =& -0.4\sin(y_2^2(t)) - 0.1y_2(t-1) + u_2(t-1) \\
& -0.3sin(u_1(t)) + 0.2u_1(t-1) \\
& +0.8\sin(u_2(t)) + 0.5u_2^2(t-1) + v_2(t).
\end{aligned} \tag{4.5.1}$$

In this case, $v_1(t)$ and $v_2(t)$ are disturbance given by $v_1(t) = (1 + 0.25q^{-1})e(t)$ and $v_2(t) = (1 + 0.25q^{-1})e(t)$, where $e(t) \in N(0, 0.001)$ is a white noise. The desired output of system is given $y_1^*(t) = \text{sign}(\sin(2\pi t/50))$ and $y_2^*(t) = 0.7$.

The proposed control method in Section 3 and 4 is illustrated effective in the control stability and robustness. The order are chosen as $n_y = n_u = 2$ and time delay $d = 1$. The regression $\varphi(t) = [y_1(t-1)\ y_2(t-1)\ y_1(t-2)\ y_2(t-2)\ u_1(t-1)\ u_2(t-1)\ u_1(t-2)\ u_2(t-2)]^T$ and $\xi(t) = [y_1(t-1)\ y_2(t-1)\ y_1(t-2)\ y_2(t-2)\ y_1^*(t)\ y_1^*(t)\ y_2^*(t)\ u_1(t-2)\ u_2(t-2)]^T$. Based on the priori acknowledge, we choose $\boldsymbol{X}_{\max} = [2\,2\,2\,2\,4\,1\,4\,1]$ and $\boldsymbol{X}_{\min} = [-2\,-2\,-2\,-2\,-4\,-1\,-4\,-1]$. The parameters $\mathbf{p}_j$ can be determined by the proposed method in Section (4.3.2).

For comparison, under the same simulation conditions and with the same parameters value,the control output results by the typical PID controller is show by Fig.4.3, where the PID controller has neither the decoupling compensator nor the nonlinear part. The corresponding control inputs are given in Fig.4.4. Figure 4.5 and 4.6 show the proposed control results and the Tab.4.1 gives the comparison results of the errors. Obviously, the proposed controller has better control performance than the typical one.

Table 4.1: Comparison results of errors based on two control method

	mean of errors	variance of errors
$y_1(t)$ typical method	0.0317	0.1909
: proposed method	-0.0033	0.2161
$y_2(t)$ typical method	-0.0196	0.0377
: proposed method	-0.0190	0.0282

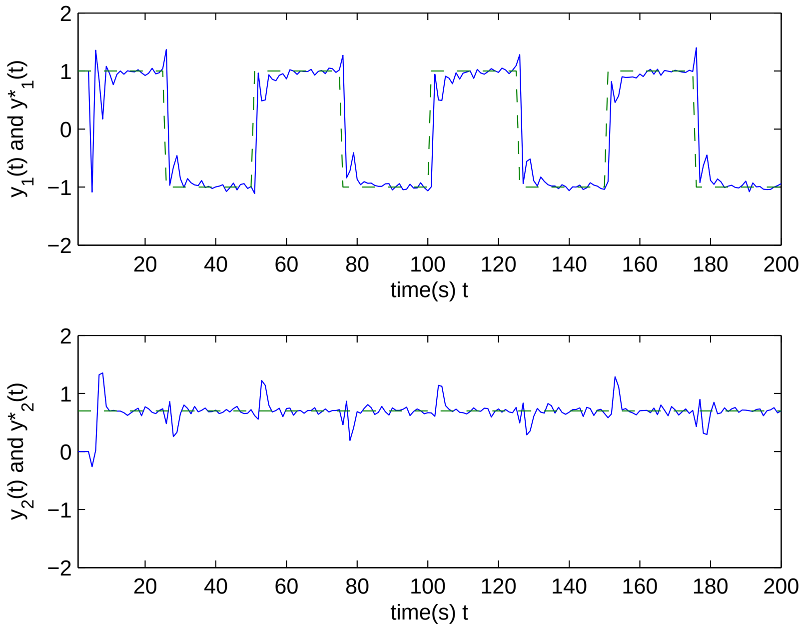

Figure 4.3: Control results of the typical PID control.

4.5.2 Case Two

The MIMO nonlinear system to be controlled is described by:

$$
\begin{aligned}
y_1(t+1) = & 0.9y_1(t) - \frac{0.3y_1(t-1)}{1+y_2^2(t-1)} + 0.4\sin(u_1(t)) \\
& +0.7u_1(t-1) + 0.3u_2(t) - 0.5u_2(t-1) \\
y_2(t+1) = & -0.4\sin(y_2^2(t)) - 0.1y_2(t-1) + u_2(t-1) \\
& -0.3\sin(u_1(t)) + 0.2u_1(t-1) \\
& +0.8\sin(u_2(t)) + 0.5u_2^2(t-1), t \in [0, 150). \qquad (4.5.2)
\end{aligned}
$$

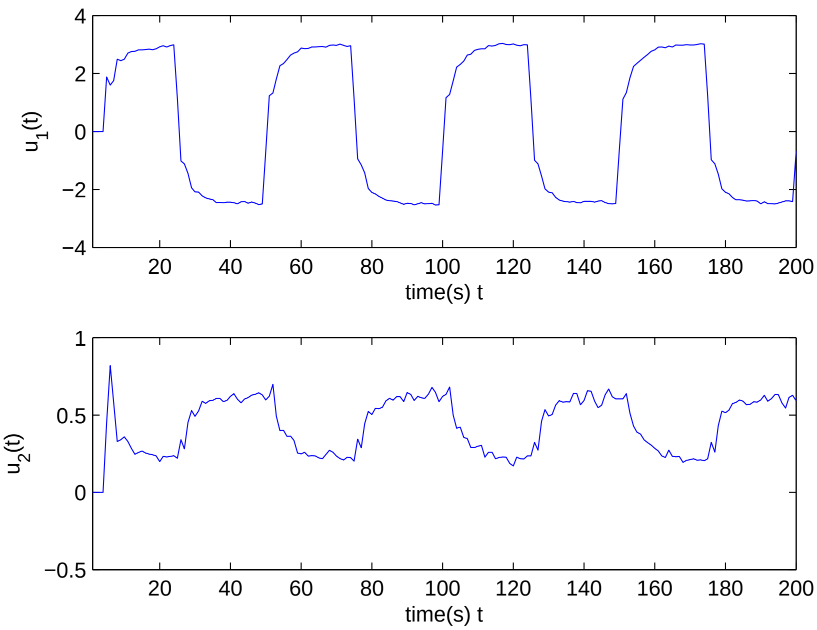

Figure 4.4: Corresponding control inputs of the PID control method.

$$
\begin{aligned}
y_1(t+1) &= 0.6y_1(t) - \frac{0.4y_1(t-1)}{1+y_2^2(t-1)} + 0.4\sin(u_1(t)) \\
&\quad +0.6u_1(t-1) + 0.4u_2(t) - 0.5u_2(t-1) \\
y_2(t+1) &= -0.5\sin(y_2^2(t)) - 0.1y_2(t-1) + u_2(t-1) \\
&\quad -0.3\sin(u_1(t)) + 0.3u_1(t-1) \\
&\quad +0.9\sin(u_2(t)) + 0.5u_2^2(t-1), t \in [150, \infty).
\end{aligned} \tag{4.5.3}
$$

In this example, a system disturbance appears when $t = 150$. The desired output of system is given $y_1^*(t) = \text{sign}(\sin(\pi t/50))$ and $y_2^*(t) = 0.7$.

In this example, the proposed control method in Section 3 and 4 is illustrated effective in the control stability and robustness. The order are chosen as $n_y = n_u = 2$ and time delay $d = 1$. The regression $\varphi(t) = [y_1(t-1)\ y_2(t-1)\ y_1(t-2)\ y_2(t-2)\ u_1(t-1)\ u_2(t-1)\ u_1(t-2)\ u_2(t-2)]^T$ and $\xi(t) = [y_1(t-1)\ y_2(t-1)\ y_1(t-2)\ y_2(t-2)\ y_1^*(t)\ y_1^*(t)\ y_2^*(t)\ u_1(t-2)\ u_2(t-2)]^T$. Based on the priori acknowledge, we choose $\boldsymbol{X}_{\max} = [2\ 2\ 2\ 2\ 4\ 1\ 4\ 1]$ and $\boldsymbol{X}_{\min} = [-2\ -2\ -2\ -2\ -4\ -1\ -4\ -1]$.

Under the same simulation conditions and with the same parameters value, the control output

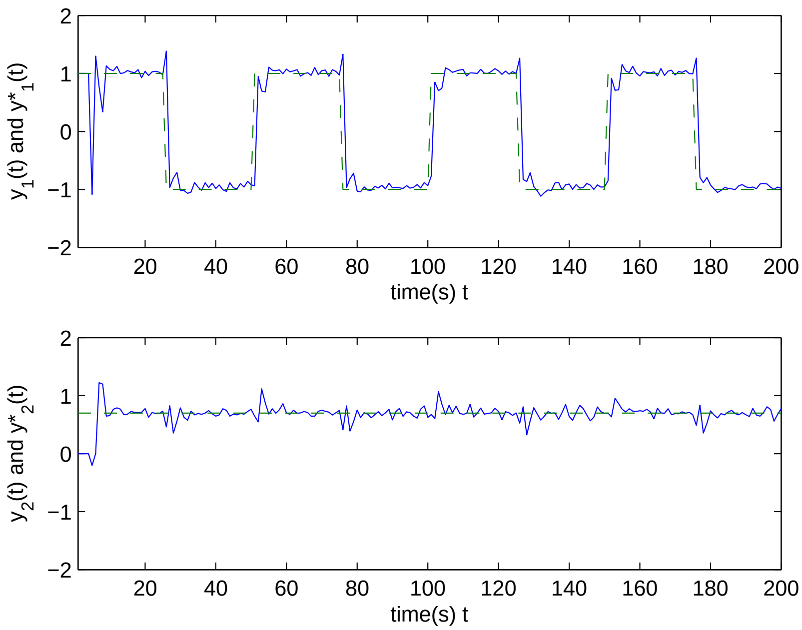

Figure 4.5: Control results of the proposed control method.

results by a typical PID controller is shown by Fig.4.7, where the PID controller has neither the decoupling compensator nor the nonlinear part, for comparison. In Fig.4.7, the dashed line is the desired output and the solid line denotes the proposed method control output $y_1(t)$ and $y_2(t)$. The corresponding control inputs $u_1(t)$ and $u_2(t)$ are given in Fig.4.8. The proposed method outputs and corresponding control inputs are shown in Fig.4.9 and 4.10. We can see that our proposed method is nearly consistent with the desired output at most of the time which is better than typical PID control method when $t \in [0, 150)$. Obviously, the control performance of our proposed method is much better than typical PID control method when the system has disturbance when $t = 150$. The input signals have small fluctuation as shown in Fig.4.10.

Tab.4.2 gives the comparison results of the errors. Obviously, the mean and variance of errors of the proposed method are smaller than the typical PID control method.

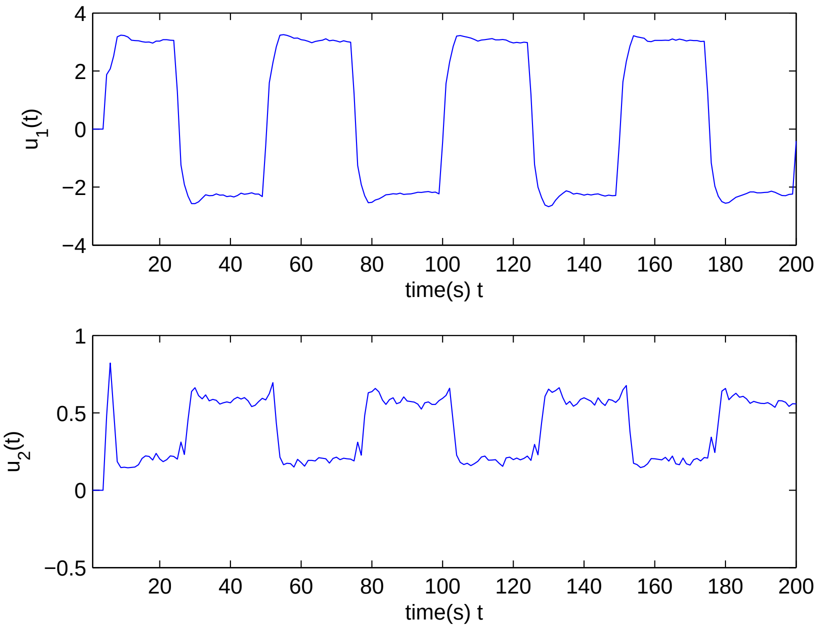

Figure 4.6: Corresponding control inputs of the proposed control method.

4.6 Conclusions

In this chapter, an MIMO quasi-ARX model is first introduced, and a nonlinear multivariable decoupling PID controller is proposed based on the proposed model for MIMO nonlinear systems. The proposed controller consists of a traditional PID controller with a decoupling compensator and a feed-forward compensator for the nonlinear dynamics from the MIMO quasi-ARX model. And an adaptive control system is presented using the MIMO quasi-ARX RBFN prediction model. The parameters of such controller is selected based on the generalized minimum control variance. The proposed control method has more simplicity structures and better control performance. The nonlinear part is not a black box whose parameters can be determined by *a priori* acknowledge. Simulations are given to show the effectiveness of the proposed method on control accuracy and robustness when a disturbance appears in the system.

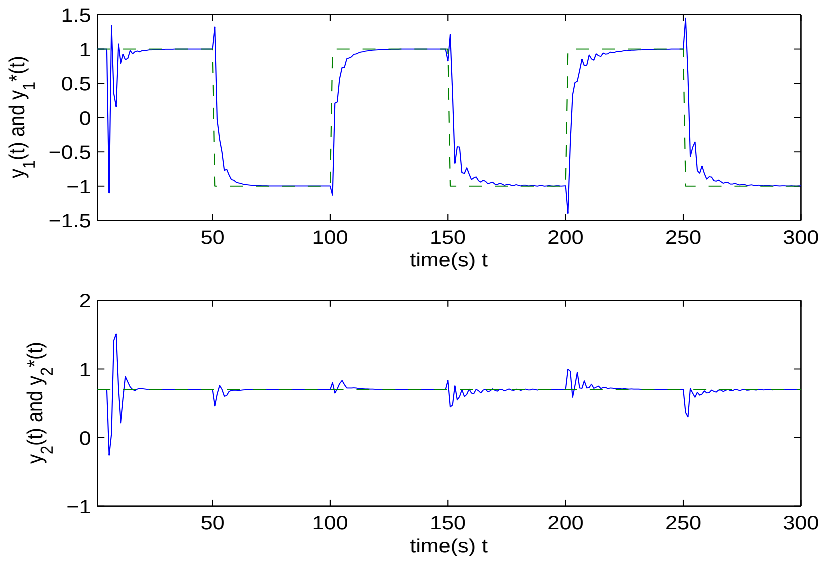

Figure 4.7: Control results of a typical PID control.

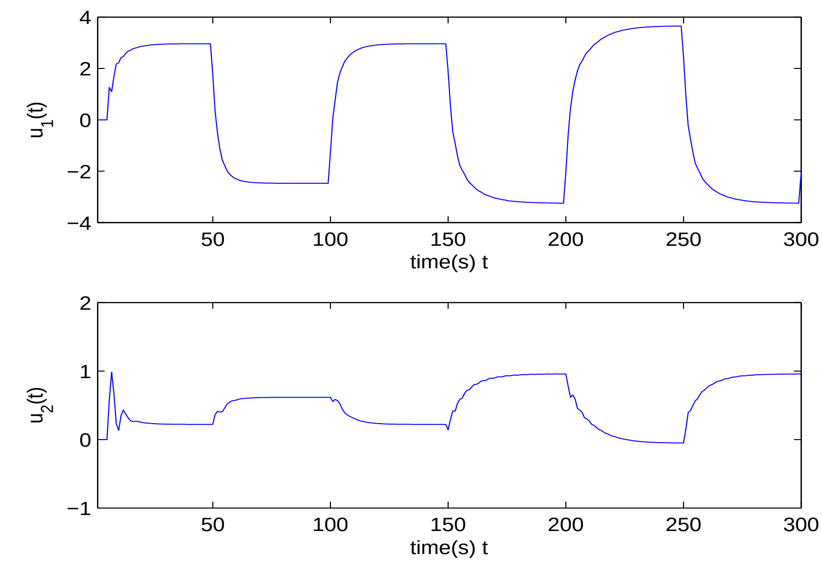

Figure 4.8: Corresponding control inputs of the PID control method.

Table 4.2: Comparison results of errors based on two control methods

	mean of errors	variance of errors
$y_1(t)$ typical method	0.066	0.1612
: proposed method	-0.0034	0.0108
$y_2(t)$ typical method	-0.0063	0.1256
: proposed method	-0.0029	0.0060

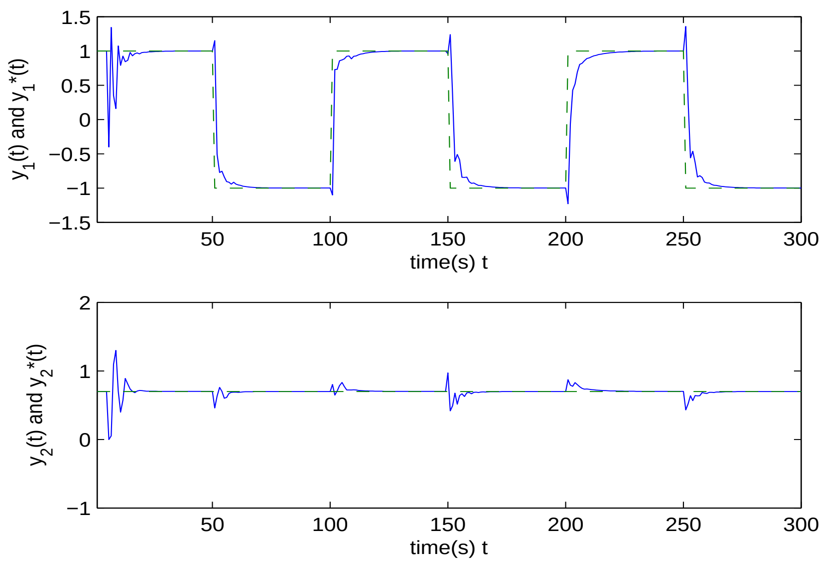

Figure 4.9: Control results of the proposed control method.

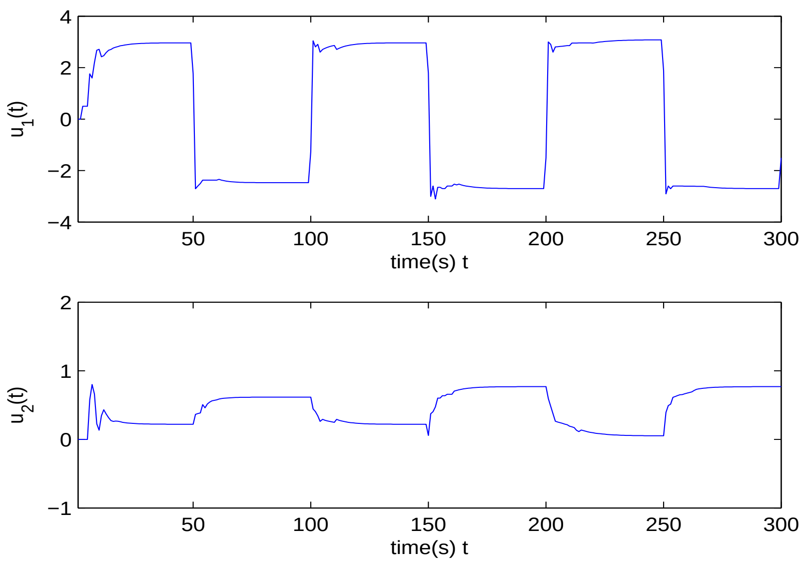

Figure 4.10: Corresponding control inputs of the proposed control method.

Chapter 5

An Identification Method for Quasi-ARX Model

5.1 Introduction

Today, nonlinear systems have received increasing attention from all fields of sciences and humanities [66, 100, 30, 54, 22, 63, 101], and have been everywhere in the real world, such as food-webs, ecosystems, metabolic pathways. They also include systems which are founded and used by human, such as aeronautical satellite, unpiloted avion, industrialized machine (electric arc furnace). How to accurately and handily control those complex systems has been the problem which we must face to. At the last few years, Neural Networks (NNs) and Neuro-Fuzzy Networks (NFNs) have been used to nonlinear modeling because they can learn any nonlinear mappings and got many good results [30]-[22]. Whereas, a nonlinear model based directly on NNs or NFNs are not handiness to be used for control and fault diagnosis.

To solve this problem, we have proposed a quasi-ARX modeling scheme which consists two parts: a macro-part and a kernel-part[30]. The macro-part is a user-friendly interface constructed using already known knowledge and the characteristic of network structure. Sometimes, linear model is chosen such as ARX model. The format of its coefficients can be easy got. The kernel-part is an ordinary NN or NFN, which is used to parameterize the coefficients of macro-model and is different from a nonlinear ARX model based directly on NNs or NFNs. When NFN is used in the kernel-part, the obtained quasi-ARX model is linear in the parameters to be estimated. This linearity is a very useful feature from the viewpoint of control.

However, variables and the order of the model increases, the complexity of as the number of input-output designing the NFN also increases. A linear Principal Components Analysis (PCA) is introduced to reduce the dimension of the NFN input on the assumption that the input variables of NFN is linear correlation [101]. In fact, the input variables do not only depend on each other linearly. When nonlinear correlations between variables exist, a Nonlinear Principal Components Analysis (NPCA) will describe the data with greater accuracy than PCA [102, 103, 104].

Motivated by the above discussion, a NPCA network trained by Artificial Neural Network (ANN) is used to reduce the dimension for the quasi-ARX modeling.

The rest of this chapter is organized as follows. In Section 5.2, the considered system is given and the quasi-ARX modeling is introduced. Section 5.3 provides a predictor. Section 5.4 introduces how to train NPCA and parameter adjustment. Then, numerical simulations are carried out to show the effectiveness of the proposed modeling approach in Section 5.5. At last Section 5.6 gives some conclusion.

5.2 Problem Description and Modeling

5.2.1 Problem Description

Consider a single-input-single-output (SISO) black-box nonlinear

$$\begin{aligned} y(t) &= g(\varphi(t)) + v(t), \\ \varphi(t) &= y(t-1), ...y(t-n), u(t-d), ..., \ u(t-m-d+1)]^T \end{aligned} \tag{5.2.1}$$

where $y(t)$ denotes the output at time t ($t = 1, 2, ...$), $u(t)$ the input, d the known integer time delay (For simply, let d=1 in this chapter. Other conditions can be got following same method.), $\varphi(t)$ the regression vector, and $v(t)$ the system disturbance. $g(\cdot)$ is a nonlinear function which satisfies the following assumes[30]:

- $g(\cdot)$ is a continuous function, and at $\varphi(t) = 0$ it is C^∞ continuous.
- the input-output of system, $u(t)$, $y(t)$, are bounded, where the bounds are known as *a priori*.
- the system is controllable, where a reasonable unknown controller may be expressed by $u(t) = \rho(\xi(t))$, where $\xi(t) = [y(t) \ ... \ y(t-n) \ u(t-1) \ ... \ u(t-m) \ y^*(t+1) \ ... \ y^*(t+1-l)]^T$ ($y^*(t)$ denotes reference output).

It needs to derive an explicit expression of $\rho(\cdot)$ to control the system (5.2.1). In this chapter, a minimum prediction error adaptive controller is got through minimizing the criterion function as follows:

$$\mathbf{M}(t+d) = \left[\frac{1}{2}(y(t+1) - y^*(t+1))^2 + \frac{\lambda}{2}u(t)^2\right] \tag{5.2.2}$$

where λ is a weighting factor for the control input.

The proposed controller has two distinctive features:

(1) it is linear for the variables synthesized in control systems;

(2) its parameters have explicit meanings.

5.2.2 Quasi-ARX Modeling

Through Taylor expansion of function $g(\cdot)$ around the region $\varphi(t) = 0$

$$y(t){=}g(0) + g'(0)\varphi(t) + \frac{1}{2}\varphi^T(t)g''(0)\varphi(t) + ...\ + v(t) \tag{5.2.3}$$

Let

$$\begin{aligned}\theta(\varphi(t)){=}&\left(g'(0) + \frac{1}{2}\varphi^T(t)g''(0) + \cdots\right)^T\\ {=}&[a_{1,t}\ ...\ a_{n,t}\ b_{0,t}\ ...\ b_{m-1,t}]^T\end{aligned}$$

where the coefficients $a_{i,t} = a_i(\varphi(t))$ and $b_{i,t} = b_i(\varphi(t))$ are nonlinear functions of $\varphi(t)$. g(0)= 0 is assumed for simplicity. We can get a regression form of the system (5.2.1) is described by (5.2.4) as in Ref.[30]:

$$y(t) = \varphi^T(t)\theta(\varphi(t)) + v(t) \tag{5.2.4}$$

A similar-linear ARX model (5.2.4) is developed as a macro-model:

$$A(q^{-1}, \varphi(t))y(t) = B(q^{-1}, \varphi(t))u(t-1) + v(t) \tag{5.2.5}$$

where q^{-1} is the backward shift operator, e.g. $q^{-1}u(t) = u(t-1)$.

$$\begin{aligned}A(q^{-1}, \varphi(t)){=}&1 - a_{1,t}q^{-1} - \ldots - a_{n,t}q^{-n}\\ B(q^{-1}, \varphi(t)){=}&b_{0,t} + \ldots + b_{m-1,t}q^{-m+1}.\end{aligned}$$

5.3 Prediction Based on Neurofuzzy and NPCA

When $d = 1$, an 1 step predictor is given Ref.[105]:

$$y(t+1) = \alpha(q^{-1}, \phi(t))y(t) + \beta(q^{-1}, \phi(t))u(t) \tag{5.3.1}$$

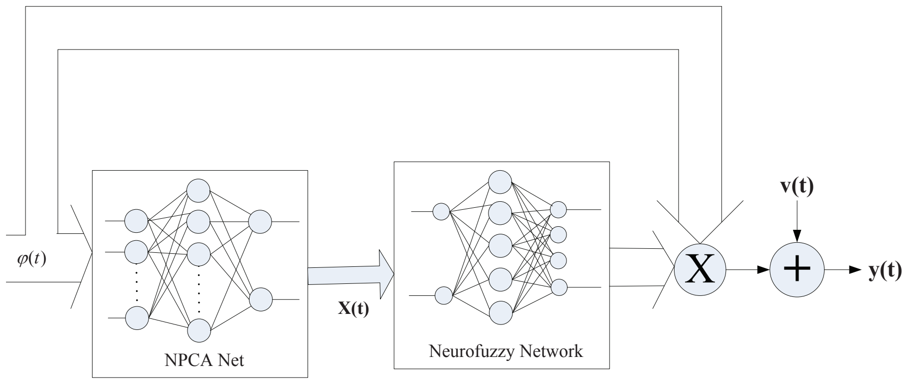

Figure 5.1: An image of the quasi-ARX model.

where

$$\phi(t) = [y(t), ...y(t-n+1), u(t), ...u(t-m+1)]^T$$

$$\alpha(q^{-1}, \phi(t)) = \alpha_{0,t} + \ldots + \alpha_{n-1,t}q^{-n-1}$$
$$\beta(q^{-1}, \phi(t)) = \beta_{0,t} + \ldots + \beta_{m-1,t}q^{-m+1}.$$

The predictor based on Neurofuzzy and NPCA networks is shown as Fig.5.1.

The system described by Eq.(5.2.1) are assumed to be bounded, so we can parameterize $\alpha_{i,t}, \beta_{i,t}$ by using a class of neurofuzzy models:

$$\alpha_{i,t} = \alpha_i + \sum_{j=1}^{M} \omega_{ij} \mathcal{N}_f(\boldsymbol{p}_j, \mathbf{x}(\phi(t)))$$
$$\beta_{k,t} = \beta_k + \sum_{j=1}^{M} \omega_{k+n,j} \mathcal{N}_f(\boldsymbol{p}_j, \mathbf{x}(\phi(t)))$$

where $\alpha_i, (i = 0, 1, ..., n-1)$, $\beta_k, (k = 0, 1, ..., m-1)$ and ω_{ij} are constant parameters. $\mathcal{N}_f(\cdot, \cdot)$ is the fuzzy "basis function" and $\boldsymbol{p}_j$ is its parameter vector. From Ref.[100], the fuzzy "basis function" $\mathcal{N}_f(\cdot, \cdot)$ is expressed explicitly by

$$\mathcal{N}_f(\boldsymbol{p}_j, \mathbf{x}(\phi(t))) = \frac{\sum_{j=1}^{M} \omega_{ij} (\wedge_{k=1}^{r} \mu_{A_k^j}(x_k(t)))}{\sum_{j=1}^{M} (\wedge_{k=1}^{r} \mu_{A_k^j}(x_k(t)))} \tag{5.3.2}$$

where $r = \dim(\mathbf{x}(t))$, and $\wedge$ is the minimum operator, M is the number of fuzzy rules, $x_k(t)$ are the elements of $\mathbf{x}(t)$, and $\mu_{A_k^j}(\cdot)$ is the membership function of fuzzy set A_k^j. The triangle function is used as membership function.

The input variables of NFN $\mathbf{x}(t)$ is supposed to be the vector $\phi(t)$. However, when the dimension of $\phi(t)$ is large, for a simple designing method the number of fuzzy rule may increase dramatically. To solve this problem, a NPCA network (5.3.3) is introduced to reduce the dimensionality instead of PCA network, because $\phi(t)$ is a regression one whose elements are highly nonlinear correlated. Express the NPCA network by:

$$\mathbf{x}(\phi(t)) = Q(\mathcal{W}, \phi(t)) \tag{5.3.3}$$

where $Q(\mathcal{W}, \phi(t)) = W^2 f(W^1\phi(t) + B^1) + B^2$. $\mathcal{W} = \{W^1, B^1, W^2, B^2\}$, $f(\cdot)$ is sigmoidal function (i.e.$f(x) = \frac{1}{1+e^{-x}}$).

Following from the equations (5.2.1)-(5.3.3) and defining :

$$\Omega_0 = (\alpha_i, \beta_k) \quad i = 0, ..., n-1; k = 0, ..., m-1$$

$$\Omega_j = (\omega_{i,j}, \omega_{k,j}) \quad i = 0, ..., n-1; k = 0, ..., m-1$$

We have a predictor expressed by:

$$y(t+1) = \phi^T(t)\Omega_0 + \sum_{j=1}^{M} \Omega_j \phi^T(t) \mathcal{N}_f(\mathbf{p}_j, Q(\mathcal{W}, \phi(t))) \tag{5.3.4}$$

5.4 Implementation Aspects

In this section, we discuss some issues concerning the implementation of the predictor to adaptive control of nonlinear systems.

5.4.1 Linearity for $u(t)$

In order to obtain a control law by differentiating the criterion function defined by (5.2.2)

$$\frac{\partial \mathbf{M}(t+1)}{\partial u(t)} = 0 \implies u(t)$$

the predictor must be linear with respect to $u(t)$. However, the predictor described by (5.3.4) is not the case because the coefficients $\alpha_{i,t}$ and $\beta_{i,t}$ are nonlinear functions of $\mathbf{x}(\varphi(t))$ that contains $u(t)$ as its element.

Now, we will use the method from Ref.[30]. Because we have assumed that the system 1 is controllable, where a reasonable unknown controller is $u(t) = \rho(\xi(t))$. we use this unknown $\rho(\cdot)$ to replace variable $u(t)$ in the coefficients $a_{i,t}$ and $b_{i,t}$

$$\begin{aligned}\alpha_{i,t} &= \alpha_i(\mathbf{x}(\varphi(t))) \simeq \alpha_i(\phi_\rho(t)) \triangleq \alpha_i(\xi(t)) \\ \beta_{i,t} &= \beta_i(\mathbf{x}(\varphi(t))) \simeq \beta_i(\phi_\rho(t)) \triangleq \beta_i(\xi(t))\end{aligned}$$

where $\phi_\rho(t)$ is $\phi(t)$ whose element $u(t)$ is replaced by $\rho(\xi(t))$, that is, $\phi_\rho(t) = [y(t) \ ... \ y(t-n+1) \ \rho(\xi(t)) \ u(t-1) \ ... \ u(t-m+1)]^T$. $\xi(t)$ has a form of

$$\xi(t) = [y(t) \ ... \ y(t-n+1) \ u(t-1) \ ... \ u(t-m+1) \ y^*(t+1)] \tag{5.4.1}$$

It follows that the predictor is expressed by

$$y(t+1) = \phi^T(t)\Omega_0 + \sum_{j=1}^{M} \Omega_j \phi^T(t) \mathcal{N}_f(\mathbf{p}_j, Q(\mathcal{W}, \xi(t)))(t)) \tag{5.4.2}$$

which is linear w.r.t $u(t)$.

Introduce the following notations

$$\begin{aligned}\Theta &= [\Omega_0^T, \Omega_1^T, ..., \Omega_M^T] \\ \Phi(t) &= [\phi^T(t), \ \phi^T(t)^T \otimes \xi_{\mathcal{N}_f}^T(t)]^T\end{aligned}$$

where the symbol $\otimes$ denotes Kronecker production, $\xi_{\mathcal{N}_f}^T(t) = [\mathcal{N}_f(\boldsymbol{p}_j, Q(\mathcal{W}, \xi(t)))(t), \ j = 1, ..., M]$. It follows that the predictor has a linear regression form expressed by

$$y(t+1) = \Phi^T(t)\Theta. \tag{5.4.3}$$

5.4.2 Parameter Adjustment

The predictor parameters must be adjusted on-line or off-line because they are unknown and can not be calculated from system parameters for the relation between system parameters and predictor parameters is unknown. Fortunately, many existing algorithm can be applied to our case without loss their properties.

Based on (5.3.4), parameters are divided into three parts: the $\mathcal{W}$ of NPCA network is the 1st part; $\boldsymbol{p}_j$ of the NFN network is the 2nd part; $\Omega_j (j = 0, ..., M)$ is the 3rd part. An algorithm consisting of three parts is used to the parameter adjustment.

(1) Adjusting Part 1

During control process, the parameters $\mathcal{W}$ are trained off-line firstly. An autoassociative network is designed to train NPCA network as Fig.5.2 as in Ref.[106]. $\xi(t)$ is the input and output layers. $\mathbf{x}(\xi(t))$ is the second hind layer. Weights W^1 W^2 W^3 W^4 and bias B^1 B^2 B^3 B^4 are updated by

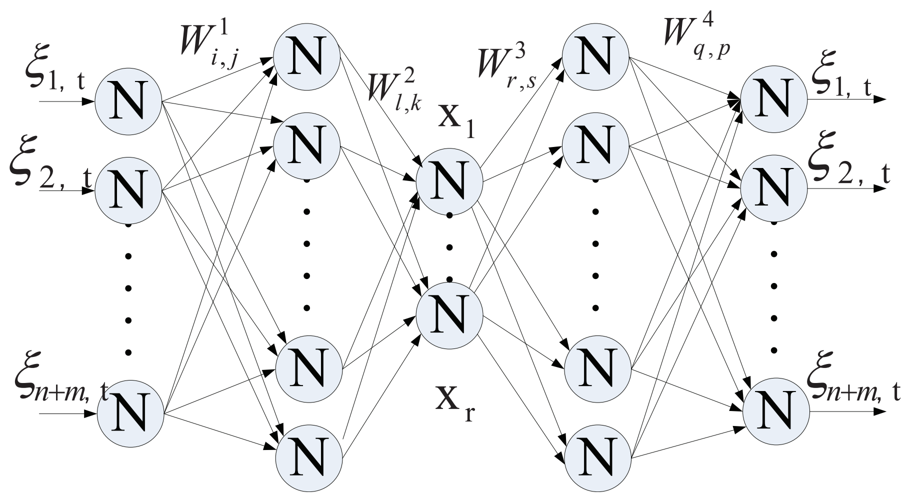

Figure 5.2: Network architecture for NPCA training with an autoassociative network.

using a BP algorithm which is same with Ref.[106]. After training, the input, first and second layers are used for the input of NFN. Then we can get the input of NFN network by (5.3.3).

(2) Adjusting Part 2

Now to initialize $\boldsymbol{p}$.

$$\boldsymbol{p}_j = [\bar{x}_1^j\ \bar{x}_2^j\ ...\ \bar{x}_r^j]^T \ \ (j = 1, ..., M)$$

It can be seen from Eq.(5.3.2) that $\boldsymbol{p}_j$ is a parameter vector associated with the partition of the operating region as in Sec.4.3.2. The similar simple strategy is used for determining the parameters $\boldsymbol{p}_j$. Only the least prior knowledge required for this method is the operating region of the input vector of multi-input and multi-output neurofuzzy model. We can get the information from the output of the trained NPCA network. Denotes: $\xi(t) = [x_1, x_2, ...x_r]^T$. $[X_{\min}, X_{\max}]$ which is the operating region need to be known. Then the neurofuzzy model can be built in a way shown in Fig.5.3, which shows the case where $r = 2$ and $M = 4 \times 4$. Obviously, the value $\boldsymbol{p}_j$ is easily determined based on a vector given by:

$$\bar{X} = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^{n_1} \\ x_2^1 & x_2^2 & \cdots & x_2^{n_2} \\ . & . & & . \\ x_r^1 & x_r^2 & \cdots & x_r^{n_r} \end{bmatrix}; \tag{5.4.4}$$

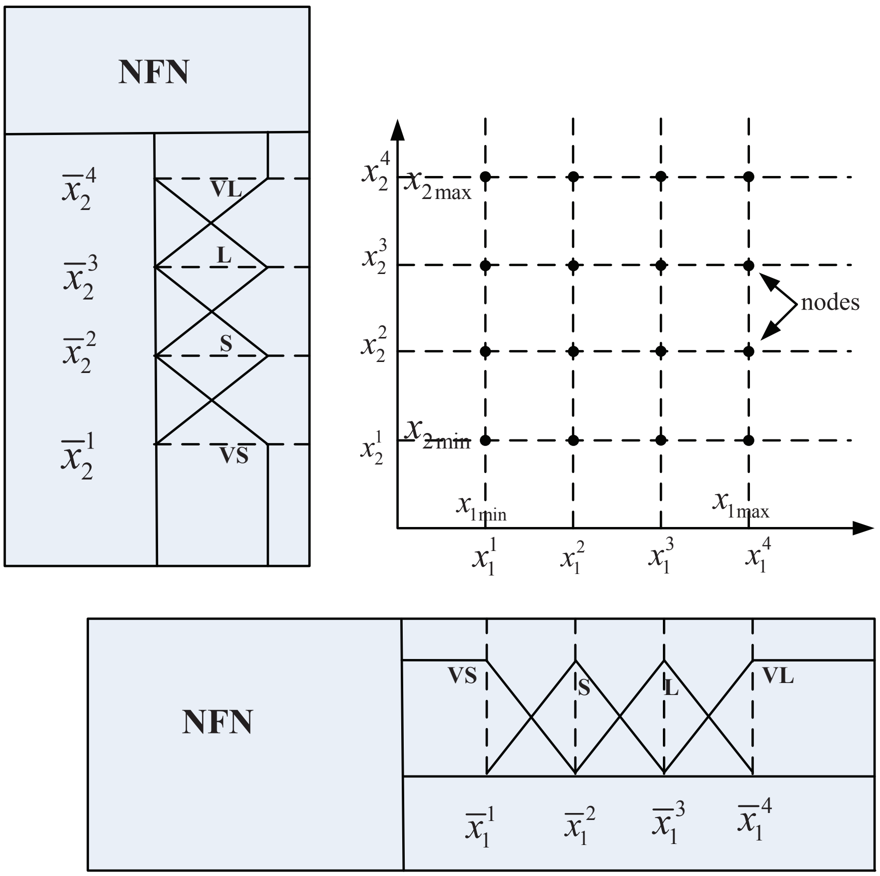

Figure 5.3: Network architecture for NPCA training with an autoassociative network.

The efficient use of prior knowledge information for determining the parameters $\boldsymbol{p}_j$ and the order M plays a key role in the quasi-ARMAX modeling [33], see Sec.4.3.2 for detail. The follows are some points:

- The least prior knowledge required for determining $\boldsymbol{p}_j$ is the information about operating region of $\mathbf{x}(t) = [x_i;\ i = 1, ..., r]^T$. That is, $[X_{\min}, X_{\max}]$ should be known for the modeling that the operating region is mostly located in $X_{\min} \leq x(t) \leq X_{\max}$.

- When NFNs described by (5.3.2) are used, the number of rules is $M = \prod_{i=1}^{r} n_i$ where the number of fuzzy sets for variable x_i is denotes as n_i. If $\dim(\tilde{\varphi}(t))$ is large, M will be rather large. Therefore, NPCA network is used to reduce the number of input ξ.

(3) Adjusting Part 3

In Adjusting part 2, Θ is adjusted while fixing $\boldsymbol{p}$ and W^1 B^1 W^2 B^2, which is performed by minimizing the following criterion function

$$V_N(\Theta) = \frac{1}{2}\sum_{t=1}^{N} \epsilon^2(t) + \frac{1}{2}C_\alpha \Theta^T(t)\Theta(t) \tag{5.4.5}$$

where $\epsilon(t) = y(t) - \Phi^T(t)\hat{\Theta}(t-1)$ is the prediction error and C_α is small positive value. If C_α is chosen so small that the second term of Equ.(5.4.5) does not affect the convergence property of adjusting algorithm, it is well known that the above minimization may be performed using many existing methods available for linear adaptive predictor [2].

5.5 Control Simulations

5.5.1 Deriving and Implementing Control Law

Consider a minimum variance control, we can obtained a control law with respect to $u(t)$:

$$\begin{aligned} u(t) = & \frac{\beta_{0,t}}{\beta_{0,t}^2 + \lambda}\{[\beta_{0,t} - B(q^{-1}, \mathbf{x}(t))q]u(t-1) \\ & -y_{\mathbf{x}}(t)) + y^*(t+1) - A(q^{-1}, \mathbf{x}(t))y(t)\} \end{aligned} \tag{5.5.1}$$

A robust adaptive algorithm with *dead zone* will be implemented which has been shown to be effective for dealing with prediction error due to unmodeled dynamics [2]. Through analysis in Sec.5.4.2, $\boldsymbol{p}$ and W^1 B^1 W^2 B^2 are fixed firstly, then the parameters of controller can be identified on-line. It can not implement it if directly introduces NN as in Ref.[54].

5.5.2 Numerical Simulations

In this part, two examples will be carried out to show the effectiveness of the proposed scheme.

Example 1

The unknown system to be controlled is given in which the linear part of system is described by

$$G(q^{-1}) = \frac{0.7q^{-1} - 0.68q^{-2}}{1 - 1.72q^{-1} + 0.74q^{-2}} \tag{5.5.2}$$

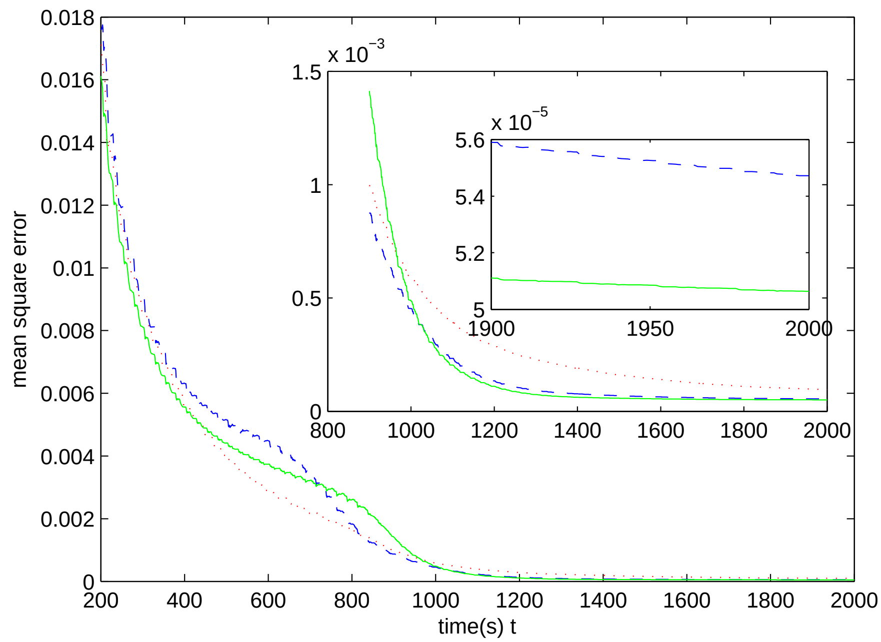

Figure 5.4: MSE between $y(t)$ and $y^*(t)$ calculated in a moving window for Example 1.

while the nonlinear element is a dead zone described by

$$z(t) = \begin{cases} u(t) - 1.75 & \text{if } u(t) > 2 \\ 0.0625 \times \text{sign}(u(t)) \times u^2(t) & \text{if} |u(t)| \leq 2 \\ u(t) + 1.75 & \text{if } u(t) < -2 \end{cases}$$

The desired output of system is

$$y^*(t) = -0.2y^*(t-1) + 0.63y^*(t-2) + r(t-1) + 0.8r(t-2) \tag{5.5.3}$$

where $r(t) = \sin(2\pi t/25) + \sin(2\pi t/10)$.

Estimation data are sampled when system is excited using random input sequence. Firstly, trains the autoassociative network using the algorithm described in Sec.5.4.1. We let $n = 3, m = 2, r = 2$, and a 5-6-2-6-5 autoassociative network is chosen as training net. 5-6-2 network trained parameters are used for NFN. We also train PCA network For comparison as in Ref.[101]. The other contrast is $n = 2, m = 2$, from Ref.[105] under some prior knowledge. All parameter vector $\boldsymbol{p}_j$ is fixed to its initial value.

The Mean Square Errors (MSE) of three adaptive controller is calculated respectively in a moving window:

$$\mathrm{MSE}(t) = \frac{1}{\mathcal{L}} \sum_{k=t-\mathcal{L}+1}^{t} (y(k) - y^*(k))^2 \tag{5.5.4}$$

where $\mathcal{L}$ was chosen to be 100. Figure 5.4 shows the convergence properties of MSE, in which solid green line is the result of the proposed predictor, dashed red line and dashed blue line are the results of Ref.[105] and Ref.[101] respectively and the least figure ignores the red line because it is too larger. It is clear that the proposed predictor has better performance than others.

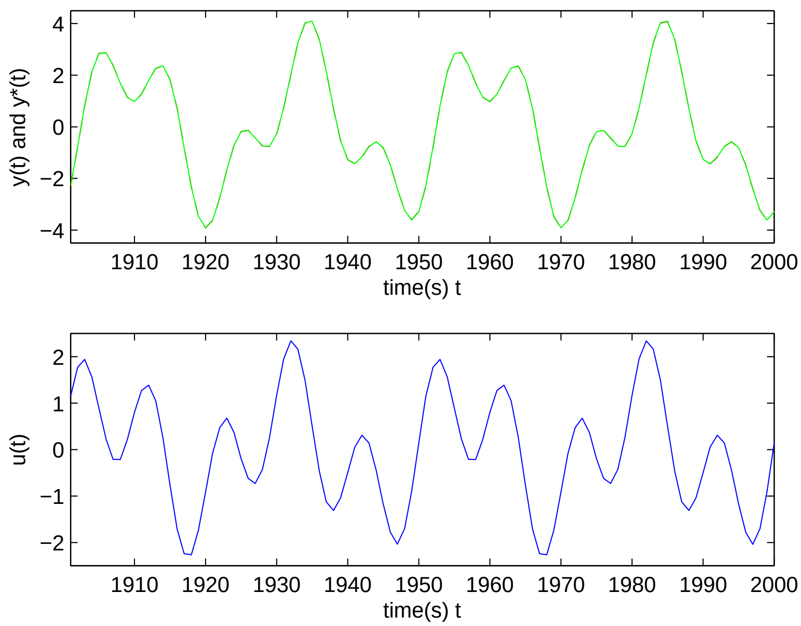

Figure 5.5: (*Upper diagram*) Controlled output $y(t)$ (solid red lines) and desired output $y^*(t)$ (dashed green lines); (*Lower diagram*) Control input $u(t)$.

Figure 5.5 shows the controlled system output, reference output and control signal. It is clear that the proposed nonlinear adaptive predictor can control this nonlinear very well. But Example 1 just show the well control ability of proposed method which don't need some prior knowledge. We will use it to reduce the dimension of the Example 2 control problem.

Example 2

$$
\begin{aligned}
y(t)=&\frac{\exp(-y^2(t-2))*y(t-1)}{1+u^2(t-3)+y^2(t-2)}+\frac{(0.5*(u^2(t-2)+y^2(t-3)))*y(t-2)}{1+u^2(t-2)+y^2(t-1)}\\
&+\frac{\sin(u(t-1)*y(t-3))*y(t-3)}{1+u^2(t-1)+y^2(t-3)}+\frac{\sin(u(t-1)*y(t-2))*y(t-4)}{1+u^2(t-2)+y^2(t-2)}\\
&+u(t-1)+v(t)
\end{aligned}
\tag{5.5.5}
$$

and the disturbance $v(t)$ is described by

$$
v(t)=(1+0.25q^{-1})e(t) \tag{5.5.6}
$$

where $e(t)\in N(0,0.001)$ is a white noise.

The desired output in this example is

$$
y^*(t)=0.6y^*(t-1)+r(t-1) \tag{5.5.7}
$$

where $r(t)=\sin(2\pi t/25)+\sin(2\pi t/10)$. Estimation data are sampled when system is excited using random input sequence. Firstly, trains the autoassociative network using the algorithm described in Sec.5.4.1. We let $n=4, m=3, r=2$, and a 7-6-2-6-7 autoassociative network is chosen as training net. 7-6-2 network trained parameters are used for NFN input.

Three kind predictors are used to compare with our proposed method. In all figure, the solid green line is the result of the proposed predictor and dashed red line is the result of compared method. Firstly, directly choose $(y(t-1),...y(t-4),u(t-1),..,u(t-3))$ as inputs of NFN, which involves more than 3^7 parameters and slow the on-line adaptive control speed. Fig.4.6 shows the convergence properties of MSE.

Secondly, directly choose $(y(t-1),u(t-1))$ as inputs of NFN which is shown in Fig.5.7.

Finally, two methods are compared with us under two conditions: with or without noise. PCA network has two output as the NFN input. The other is to choose $(y(t-1),y(t-2),u(t-1),u(t-2))$ as input which is same as Ref.[101]. Figure 5.8 and Fig.5.9 give the MSE without considering noise and with noise, respectively. Figure 5.10 shows the controlled system output, reference output and control signal. All results indicate that our method can reduce complexity and keep control precision.

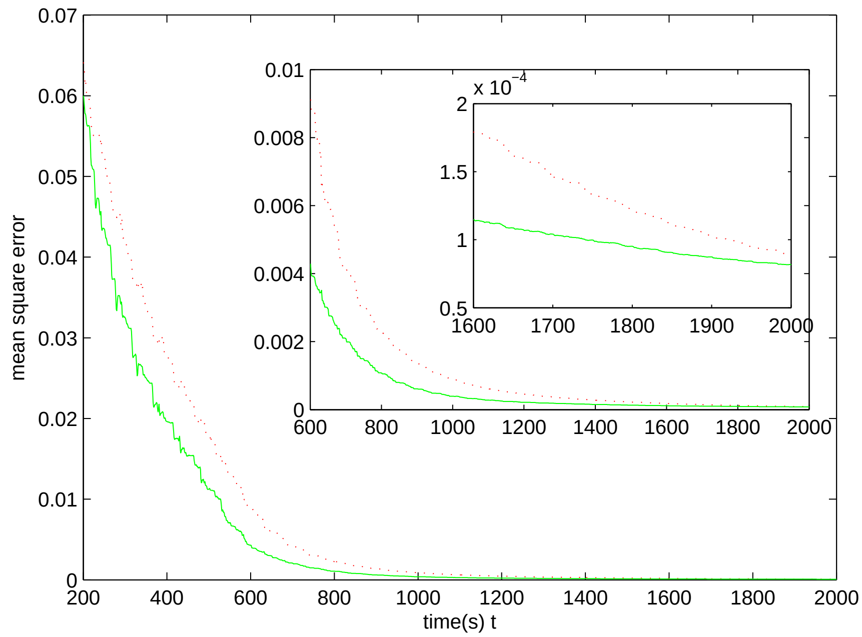

Figure 5.6: MSE between $y(t)$ and $y^*(t)$ calculated in a moving window for Example 2.

5.6 Conclusion

Quasi-ARX modeling scheme based on ARX model and NFN not only has accurate representation ability, but also has a structure similar to linear ARX model. Because it is linear in the parameters to be estimated. However, variables and the order of the model increases, the complexity of input-output designing the NFN also increases. A linear principal components analysis (PCA) is introduced to reduce the dimension of the NFN input on the assumption that the input variables of NFN is linear correlation. In fact, the input variables do not only depend on each other linearly. In this chapter, A NPCA network is used to reduce the dimension for the quasi-ARX NFN model. This method reduces the number of controller parameters and improves the control performance of the controller based on the quasi-ARX modeling. Numerical simulation results show that the performance of the quasi-ARX model has been improved by introducing the NPCA network.

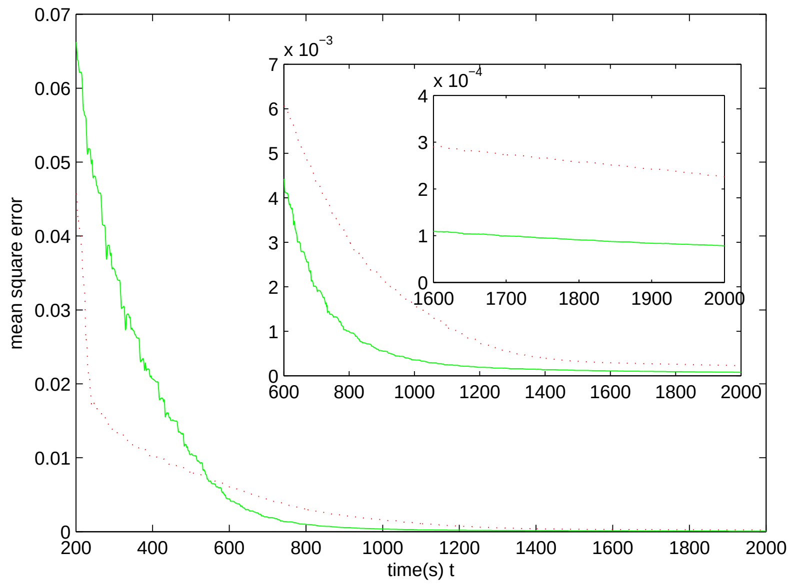

Figure 5.7: MSE between $y(t)$ and $y^*(t)$ calculated in a moving window for Example 2.

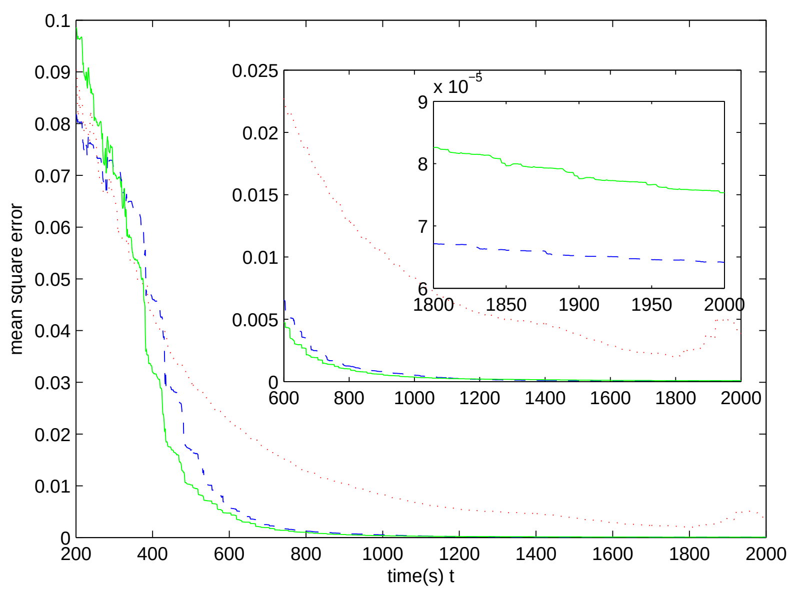

Figure 5.8: MSE between $y(t)$ and $y^*(t)$ calculated in a moving window without considering noise, in which dashed red line and dashed blue line are the results of Ref.3 and Ref.7 respectively.

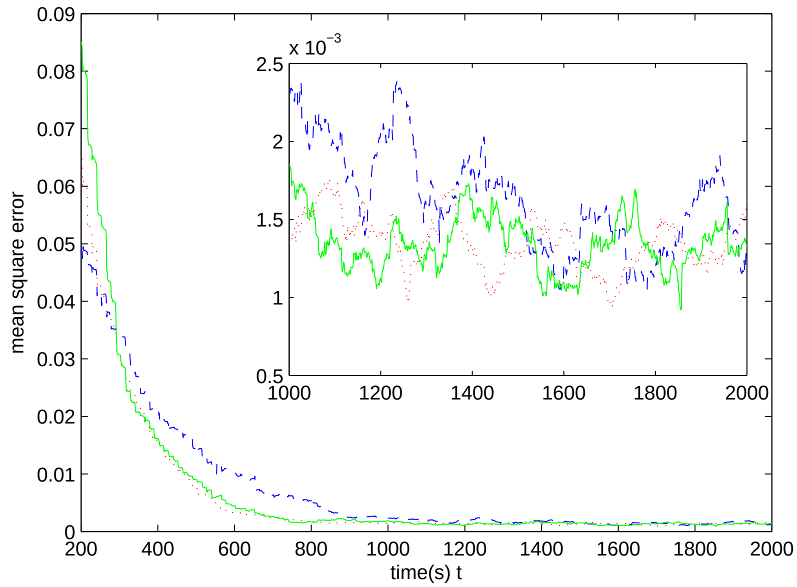

Figure 5.9: MSE between $y(t)$ and $y^*(t)$ calculated with noise, in which dashed red line and dashed blue line are the results of Ref.3 and Ref.7 respectively.

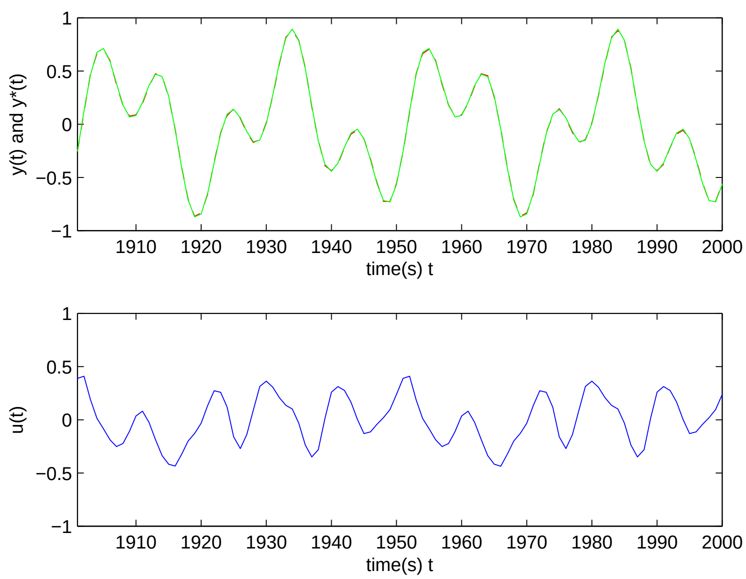

Figure 5.10: (*Upper diagram*) Controlled output $y(t)$ (solid red lines) and desired output $y^*(t)$ (dashed green lines); (*Lower diagram*) Control input $u(t)$.

Chapter 6

Conclusions

6.1 Summary

In this final chapter, a summary for whole thesis will be given.

Adaptive control have been studied as a classic research field since 1950s and adaptive control based on the linear system theory has got great achievements, especially, in many real-world applications. Recently, with the development of neural network, wavelet network, radial basis function network and some other nonlinear model, adaptive control have to face some new challenges. It is difficult to ensure the stability of these control system, although it can give a higher accuracy control performance. A quasi-linear black-box modeling scheme has been constructed based on the linear structure and the nonlinear model, so that the obtained nonlinear black-box models contain not only the linearity properties which are useful, but also have good flexibility which is used to deal with various nonlinear systems. In this thesis, quasi-ARX black box models are constructed and their applications for the nonlinear dynamical systems control scheme are studied. Investigations have made to identification, model analysis adaptive control design and stability analysis of nonlinear systems under the framework of linear system theory, on the basis of the improved model structure. The main work of the thesis has been described in Chapter 2, 3, 4, and 5.

In Chapter 2, quasi-ARX neural network is divided into two parts: the linear part is used to ensure the nonlinear control stability, and the nonlinear part is utilized to improve the control accuracy. In order to combine both the stability and universal approximation capability in our controller, a switching law is established based on system input-output variables and prediction errors. An adaptive control law is proposed for nonlinear dynamical systems and then the control system stability is proved. The proposed controller has three distinctive features:

(1) it is linear for the variables synthesized in control systems because of the linear structure.

(2) its parameters have explicit meanings which shares from the predictor.

(3) it is only one controller which combines a switching algorithm.

In Chapter 3, a stabilizing switching controller for nonlinear system is designed based on a quasi-ARX RBFN model and a fuzzy switching function. The proposed control method uses a smooth switching between a linear controller and a nonlinear controller both of which are derived from the same identified quasi-ARX RBFN prediction model. The effectiveness of the controller has been confirmed through numerical simulations. The work in this chapter has contributions as follows:

(1) The control system can satisfy the stability, response and performance requirement with only one model used.

(2) A d-difference operator is used to relax the assumption of global boundedness on higher-order nonlinear terms, which improves the work of Chapter 3.

(3) For parameterizing the coefficients of the macro-model, a RBFN is used in the kernel part to replace NN, thus nonlinear parameters of the proposed quasi-ARX RBFN prediction model can be determined by *a priori* knowledge.

(4) The prediction model only remains linear parameters to be adjusted on-line which reduces the number of on-line adjusted parameters.

In Chapter 4, an MIMO quasi-ARX model is first introduced, and a nonlinear multivariable decoupling PID controller is proposed based on the proposed model for MIMO nonlinear systems. a traditional PID controller with a decoupling compensator and a feed-forward compensator for the nonlinear dynamical from the MIMO quasi-ARX model consist the proposed controller. Then, an adaptive control system is constructed using the proposed MIMO quasi-ARX RBFN prediction model. Generalized minimum control variance are used to get the control law and the stability proof is also given. The proposed controller has the following distinctive feature:

(1) It has more simplicity structures and better control performance.

(2) It has better properties for controlling the system with disturbance (noise).

(3) Its nonlinear part is not a black box whose parameters can be determined by *a priori* acknowledge.

(4) It is a stability controller.

The work in this chapter also shows that

(1) with the improved model structure, the control algorithm based on well developed linear system theory could be extended to MIMO nonlinear systems.

(2) the linear structure of quasi-ARX model is used to resolve the decoupling problem and the nonlinear part improves the control performance.

Chapter 5 introduces a NPCA network to reduce the dimension for the quasi-ARX modeling. One part of the quasi-ARX model is the ordinary neurofuzzy network to parameterize the coefficients which faces to a problem of high dimension. A linear principal components analysis (PCA) has been introduced to reduce the dimension of the NFN input on the assumption that the input variables of NFN is linear correlation. In fact, the input variables do not only depend on each other linearly. When nonlinear correlations between variables exist, a nonlinear principal components analysis (NPCA) will describe the data with greater accuracy than PCA. This improves the performance of the quasi-ARX model. Numerical simulation results show that the performance of the quasi- ARX model has been improved by introducing the NPCA network.

6.2 Topics for Future Research

Although a lot of progress has been made, there are still many aspects that need further investigations.

- Other control methods as as PEM and GPC based on the quasi-linear model also can be used for MIMO system control. The corresponding stability and decoupling problem should be researched in the next step.

- Some parameters of the model is trained off-line to reduce the online feedback time. Therefore, we will improved the on-line algorithm to low the feedback time down in the next work.

- Although our control system can deal with some kinds of disturbances, it cannot do will when disturbance is larger. The robustness of the fuzzy switching adaptive control based on quasi-linear model is key problem for us.

- In this thesis, we use a NPCA network is used to reduce the dimension for the quasi-ARX modeling. However, a nonlinear network are introduced into a kernel part of quasi-ARX model. Although linear part can be sure stability, the condition between two parts should be made certain. That is our future work to be sure the proposed model stability.

- As we discuss in Chapter 5, variables and the order of the model increases, the complexity of as the number of input-output designing the NFN also increases. Motivated by the discussion, support vector regression can be used to deal with the complex calculations
- The control model can be used to resolve practical problem such as gene regulation network, missile control in future research.

Bibliography

[1] L. Ljung, System Identification: Theory for the User, Second Edition, Prentice-Hall PTR, Upper Saddle River, N.J., 1999.

[2] G.C. Goodwin and K.S. Sin, Adaptive Filtering Prediction and Control, Prentice-Hall, Inc., 1984.

[3] C.T. Chen, Linear System Theory and Design, Oxford University Press, Inc. New York, USA, 1998.

[4] A.J. Krener and A. Isidori, “Linearization by output injection and nonlinear observers,” Systems and Control Letters, vol.3, no.1, pp.47–52, 1983.

[5] D.G. Kelly, “Stability in contractive nonlinear neural networks,” IEEE Trans. on Biomedical Engineering, vol.37, no.3, pp.231–242, 1990.

[6] O. Nelles, Nonlinear System Identification, Springer, Berlin, 2001.

[7] G.C. Goodwin, J.C. Murdoch, and R.L. Payne, “Optimal test signal design for linear S.I.S.O. system identification,” International Journal of Control, vol.17, no.1, pp.45–55, 1973.

[8] W.L. Luyben, “Simple method for tuning SISO controllers in multivariable systems,” IMA Journal of Mathematical Control and Information, vol.25, no.3, pp.654–660, 1986.

[9] A.Abdulaziz and M.Farsi, “Non-linear system identification and control based on neural and self-tuning control,” INT J. of Adaptive Control and Signal Processing, vol.7, pp.297–307, 1993.

[10] H. Lee and M. Tomizuka, “Robust adaptive control using a universal approximator for SISO nonlinear systems,” IEEE Transactions on Fuzzy Syatems, vol.8, no.1, pp.95–106, 2000.

[11] A. Levant, “Universal single-inputcsingle-output (SISO) sliding-mode controllers with finite-time,” IEEE Transactions on Fuzzy Syatems, vol.46, no.9, pp.1447–1451, 2001.

[12] H.X. Li and S. Tong, “A hybrid adaptive fuzzy control for a class of nonlinear mimo systems,” IEEE Transactions on Fuzzy Systems, vol.11, no.1, pp.24–34, 2003.

[13] S.S. Ge, J. Zhang, and T.H. Lee, “Adaptive neural network control for a class of MIMO nonlinear systems with disturbances in discrete-time,” IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics, vol.34, no.4, pp.1630–1645, 2004.

[14] J. Hu, Research on Hybrid Black-Box Modeling for Nonlinear Systems and Its Applications, Ph.D. thesis, Kyushu Institute of Technology, Japan, 1997.

[15] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Deglon, P. Glorennec, H. Hjalmarsson, and A. Juditsky, “Nonlinear black-box modeling in system identification: a unified overview,” Automatica, vol.31, no.12, pp.1691–1724, 1995.

[16] S.A. Billings, “Identification of nonlinear systems - a survey,” Control Theory and Applications, IEE Proc. D, vol.127, no.6, pp.272–285, 1980.

[17] H. Chen, “Modeling and identification of parallel nonlinear systems: Structural classification and parameter estimation methods,” in Proc. of the IEEE, vol.83, no.1, pp.39–66, 1995.

[18] F. Chen and H. Khalil, “Adaptive control of nonlinear systems using neural networks,” Int. J. Control, vol.55, no.6, 1992.

[19] F. Chen and C. Liu, “Adaptively controlling nonlinear continuous-time systems using multilayer neural networks,” IEEE Trans. on Automatic Control, vol.39, no.6, pp.1306–1310, 1994.

[20] L. Wang, Y. Cheng, and J. Hu, “A quasi-ARX neural network with switching mechanism to adaptive control of nonlinear systems,” SICE Journal of Control, Measurement, and System Integration, vol.3, no.4, pp.246–252, 2010.

[21] Q. Zhang and A. Benveniste, “Wavelet networks,” IEEE Trans. on Neural Networks, vol.3, no.6, pp.889–898, 1992.

[22] S.A. Billings and H. Wei, “A new class of wavelet networks for nonlinear system identification,” IEEE Trans. on Neural Networks, vol.16, no.4, pp.862–874, 2005.

[23] J. Jang and C.T. Sun, “Neuro-fuzzy modeling and control,” in Proc. of the IEEE, vol.83, no.3, pp.378–406, 1995.

[24] J. Hu, K. Hirasawa, and K. Kumamaru, “A neurofuzzy-based adaptive predictor for control of nonlinear systems,” Trans. of the Society of Instrument and Control Engineers, vol.35, no.8, pp.1060–1068, 1999.

[25] S. Chen, S.A. Billings, and P.M. Grant, “Recursive hybrid algorithm for non-linear system identification using radial basis function networks,” International Journal of Control, vol.55, no.5, pp.1051–1070, 1992.

[26] J. Park and I. Sandberg, “Approximation and radial-basis-function networks,” Neural Computation, vol.5, no.2, pp.305–316, 1993.

[27] C. Jin, K. Wada, K. Hirasawa, J. Murata, and S. Sagara, “Identification of nonlinear continuous systems by using neural network compensator (in Japanese),” Trans. of IEE of Japan, vol.114-C, no.5, pp.595–602, 1994.

[28] J. Tsuji, H. Ohmori, and A. Sano, “Adaptive control incorporating neural network (in Japanese),” Trans. of the Society of Instrument and Control Engineers, vol.30, no.3, pp.295–302, 1994.

[29] Y. Hirashima, Y. Iiguni, and N. Adachi, “An adaptive control method using neural network (in Japanese),” Trans. of the Institute of Systems, Control and Information Engineers, vol.9, no.1, pp.281–286, 1994.

[30] J. Hu, K. Hirasawa, and K. Kumamaru, “Adaptive predictor for control of nonlinear systems based on neurofuzzy models,” Proc. of European Control Conference (ECC’99) (Karlsruhe), 8 1999.

[31] J. Hu and K. Kumamaru, “Identification of nonlinear systems based on adaptive fuzzy systems embedding quasi-ARMAX model,” in Proc. of the 34th SICE Annual Conference (international session), Sapporo, pp.1211–1216, 7 1995.

[32] J. Hu, K. Kumamaru, and K. Inoue, “A guaranteed nonlinear system identification using ARX networks,” in Proc. of the 27th ISCIE International Symposium on Stochastic Systems Theory and Its Applications (Beppu, Japan), pp.7–12, 11 1995.

[33] J. Hu, K. Kumamaru, and K. Inoue, “A hybrid quasi-ARMAX modeling scheme for identification and control of nonlinear systems,” in Proc. of the 35th IEEE Conference on Decision and Control (Kobe), pp.1413–1418, 12 1996.

[34] J. Hu, K. Hirasawa, and K. Kumamaru, “A hybrid quasi-ARMAX modeling and identification scheme for nonlinear stochastic systems,” Research Reports on Information Science and Electrical Engineering of Kyushu University, vol.2, no.2, pp.213–218, 1997.

[35] J. Maxwell, On Governors, Proceedings of the Royal Society of London, doi:10.1098/rspl.1867.0055. Retrieved 2008-04-14, 1867.

[36] P.C. Parks, “A. m. lyapunov’s stability theory-100 years on,” IMA Journal of Mathematical Control and Information, vol.9, pp.275–303, 1992.

[37] R. Kline, “Harold black and the negative-feedback amplifier,” IEEE Control Systems, vol.13, no.4, pp.83–85, 1993.

[38] H. Nyquist, “A. m. lyapunov’s stability theory-100 years on,” Bell System Technical Journal, vol.11, no.1, pp.126–147, 1932.

[39] E. Kappos, Classical Control Theory: A Course in the Linear Mathematics of Systems and Control, http://lanom.web.auth.gr/kappos/Research/ek29.pdf, 2002.

[40] R.E. Kalman, “Contribution to the theory of optimal control,” Boletin de la Sociedad Mathematica Mexicana, pp.102–119, 1960.

[41] R.E. Kalman, “On the general theory of control systems,” in Proc. 1st IFAC World Congress (Moscow, Russia), 1960.

[42] H. Wang, A. Nokbakhti, P. Cook, and D.H. Owens, “Professor howard h. rosenbrocks contribution to modern control,” Transactions of the Institute of Measurement and Control, vol.30, no.1, pp.5–15, 2008.

[43] K. Ogata, Modern Control Engineering (Fourth Edition), Prentice-Hall of India Private Limited, 2003.

[44] L.A. Zadeh, “Fuzzy sets,” Information and Control, vol.8, no.3, pp.338–353, 1965.

[45] K. Åström and B. Wittenmark, Adaptive Control (2nd edition), Addison Wesley, New York, 1995.

[46] G.C. Goodwin, P.J. Ramadge, and P.E. Caines, “Discrete-time multivariable aadaptive control,” IEEE Trans. on Automatic Control, vol.25, no.3, pp.449–456, 1980.

[47] H. Wang and S. Daley, “A fault detection method for unknown systems with unknown input and its application to hydraulic turbine monitoring,” INT J. of Control, vol.57, no.2, pp.247–260, 1993.

[48] X. Yu and L. Xu, “Automatic learning rate optimization by higher-order derivatives,” in Proc. IEEE International Conference on Neural Networks (Houston), pp.1072–1077, 1997.

[49] A. Alleyne and J.K. Hedrick, “Nonlinear adaptive control of active suspensions,” IEEE Trans. on Control Systems Technology, vol.1, no.1, pp.94–101, 1995.

[50] B. Gough, “Advanced adaptive control applications [in industry],” Pulp and Paper Industry Technical Conference, pp.122–132, 1995.

[51] K.J. Åström and B. Wittenmark, “On self tuning regulators,” Automatica, vol.9, pp.185–199, 1973.

[52] D. D. W. Clarke, M.A. and B. P. J. Gawthrop, “Self-tuning controller,” in Proc. The Institution of Electrical Engineers Control and Science, pp.929–934, 1975.

[53] Y. Landau, Adaptive Control, Marcel Dekker, Inc., New York, 1979.

[54] J. Hu and K. Hirasawa, “A method for applying neural networks to control of nonlinear systems,” in Neural Information Processing: Research and Development, ed. J. Rajapakse and L. Wang, pp.351–369, Springer, 5 2004.

[55] T.Y. Chai, “A self-tuning decoupling controller for a class of multivariable systems and global convergence analysis,” IEEE Trans. on Automatic Control, vol.33, no.8, pp.767–771, 1988.

[56] F.C. Chen and H.K. Khalil, “Adaptive control of a class of nonlinear discrete-time systems using neural networks,” IEEE Trans. on Automatic Control, vol.40, no.5, pp.791–801, 1995.

[57] J.Y. Choi and J.A. Farrell, “Nonlinear adaptive control using networks of piecewise linear approximators,” IEEE Trans. on Neural Networks, vol.11, no.2, pp.390–401, 2000.

[58] S.S. Ge, J. Zhang, and T.H. Lee, “Direct adaptive control for a class of mimo nonlinear systems using neural networks,” IEEE Trans. on Automatic Control, vol.49, no.11, pp.2001–2004, 2000.

[59] C. Wang, T.T.L. T. Lin, and H. Liu, “Adaptive hybrid intelligent control for uncertain nonlinear dynamical systems,” IEEE Trans. Syst. Man Cybern. B, vol.32, no.5, pp.583–597, 2002.

[60] L. Chen and K.S. Narendra, “Identification and control of a nonlinear discrete-time system based on its linearization: A unified framework,” IEEE Trans. Neural Network, vol.15, no.3, pp.663–673, 2004.

[61] S.S. Ge, J. Zhang, and T.H. Lee, “Adaptive mnn control for a class of non-affine narmax systems with disturbances,” Systems and Control Letters, vol.53, no.1, pp.1–12, 2004.

[62] Q.M. Zhu and L.Z. Guo, “Stable adaptive neurocontrol for nonlinear discrete-time systems,” IEEE Trans. on Neural Networks, vol.15, no.3, pp.653–662, 2004.

[63] P. Mhaskar, N.H. El-Farra, and P.D. Christofides, “Robust hybrid predictive control of nonlinear systems,” Automatica, vol.41, pp.209–217, 2005.

[64] K.Funahashi, “On the approximate realization of continuous mappings by neural networks,” Neural Networks, vol.2, no.3, pp.183–192, 1989.

[65] M. K.Hornik and H.White, “Multilayer feedforward networks are universal approximators,” Neural networks, vol.2, no.5, pp.359–366, 1989.

[66] K. Narendra and K. Parthasarathy, “Identification and control of dynamical systems using neural networks,” IEEE Trans. on Neural Networks, vol.1, no.1, pp.4–27, 1990.

[67] J. Hu, K. Kumamaru, and K. Hirasawa, “A quasi-ARMAX approach to modeling of nonlinear systems,” International Journal of Control, vol.74, no.18, pp.1754–1766, 12 2001.

[68] K.S. Narendra, O.A. Driollet, M. Feiler, and K. George, “Adaptive control using multiple models, switching and tuning,” International Journal of Adaptive Control and Signal Processing, vol.17, no.2, pp.87–102, 2003.

[69] J. Hu, K. Hirasawa, and K. Kumamaru, “A homotopy approach to improving PEM identification of ARMAX models,” Automatica, vol.37, no.9, pp.1323–1334, 2001.

[70] N.Baba, Y.Mogami, M.A.Kohzai, Y.Shiraishi, and Y. Yoshida, “A hybrid algorithm for finding the global minimum of error function of neural networks and its applications,” Neural Networks, vol.7, no.8, pp.1253–1265, 1994.

[71] L. Chen and K.S. Narendra, “Nonlinear adaptive control using neural networks and multiple models,” Automatica, vol.37, no.8, pp.1245–1255, 2001.

[72] J. Hu and K. Hirasawa, “Neural network based prediction model for control of nonlinear systems (in Japanese),” Trans. of the Society of Instrument and Control Engineers, vol.39, no.2, pp.166–175, 2003.

[73] R.A. Cozzio-Büeler, The Design of Neural Networks Using A Prior Knowledge, Ph.D. thesis, Informatik-Ing., ETH, Zurich, Swithzerland, 1995.

[74] M. Athans, D. Castanon, and K. Dunn, “The stochastic control of the f-8c aircraft using a multiple model adaptive control(mmac) method- part i: Equilibrium flight,” IEEE Trans Aytom Control, vol.22, no.5, pp.768–780, 1977.

[75] K. Narendra and S. Mukhopadhyay, “Adaptive control using neural networks and approximate models,” IEEE Trans. on Neural Networks, vol.8, no.3, pp.475–485, 1997.

[76] Y. Fu and T. Chai, “Nonlinear multivariable adaptive control using multiple models and neural networks,” Automatica, vol.43, no.6, pp.1101–1110, 2007.

[77] J.Y. Zhai, S.M. Fei, and X.H. Mo, “Multiple models switching control based on recurrent neural networks,” Neural Comput. and Applic., vol.17, no.4, pp.365–371, 2008.

[78] P.A. Ioannou and J. Sun, Linear System Theory and Design, Prentice-Hall, Inc. *Electronic copy at http://www-rcf.usc.edu/ ioannou/Robust_Adaptive_Control.htm*, 1996.

[79] S. Barnett, Polynominals and Linear Systems, Marcel Dekker, 1983.

[80] L. Baratchart and J.B. Pomet, “On local linearization of control systems,” Journal of Dynamical and Control Systems, vol.15, no.4, pp.47–536, 2009.

[81] L. Wang, Y. Cheng, and J. Hu, “Adaptive control for nonlinear systems based on quasi-ARX neural network,” in Proc. of World Congress on Nature and Biologically Inspired Computing(NaBIC2009) (Coimbatore), pp.1548–1551, 12 2009.

[82] S. Cao, N. Rees, and G. Feng, “Analysis and design for a class of complex control systems part I: Fuzzy modeling and identification,” Automatica, vol.33, no.6, pp.1017–1028, 1997.

[83] S. Cao, N. Rees, and G. Feng, “Analysis and design for a class of complex control systems part II: Fuzzy controller design,” Automatica, vol.33, no.6, pp.1029–1039, 1997.

[84] Y. Oussar and G. Dreyfus, “Initialization by selection for wavelet network training,” Neurocomputing, vol.34, no.1-4, pp.117–127, 2000.

[85] S. Chen, C. Cowan, and Grant, “Orthogonal least squares learning algorithm for radial basis function networks,” IEEE Trans. on Neural Networks, vol.2, no.2, pp.302–309, 1991.

[86] Y. Lu, N. Sundararajan, and P. Saratchandran, “A sequential learning scheme for function approximation using minimal radial basis function neural networks,” Neural Computation, vol.9, no.2, pp.461–478, 1997.

[87] C. Panchapakesan, M. Palaniswami, D. Ralph, and C. Manzie, “Effects of moving the centers in an RBF network,” IEEE Trans. on Neural Networks, vol.13, no.6, pp.1299–1307, 2002.

[88] Q. Zhu, S. Fei, T. Zhang, and T. Li, “Adaptive RBF neural-networks control for a class of time-delay nonlinear systems,” Neurocomputing, vol.71, no.16–18, pp.3617–3624, 2008.

[89] H. Peng, J. Wu, G. Inoussa, Q. Deng, and K. Nakano, “Nonlinear system modeling and predictive control using the RBF nets-based quasi-linear ARX model,” Control Engineering Practice, vol.17, no.1, pp.59–66, 2009.

[90] B.J. Frey and D. Dueck, “Clustering by passing messages between data points,” Science, vol.315, no.5814, pp.972–976, 2007.

[91] P.E. McDermott and D.A. Mellichamp, “A decoupling pole placement self-tuning controller for a class of multivariable processes,” Optimal Control Applications and Methods, vol.7, no.1, pp.55–79, 1986.

[92] R. Yusof, S. Omatu, and M. Khalid, “Self-tuning PID control: a multivariable derivation and application,” Automatica, vol.30, no.12, pp.1975–1981, 1994.

[93] S.S.S. Ge and C. Wang, “Adaptive neural control of uncertain MIMO nonlinear systems,” IEEE Trans. on Neural Networks, vol.15, no.3, pp.674–692, 2004.

[94] L.C. Hung and H.Y. Chung, “Decoupled control using neural network-based sliding-mode controller for nonlinear systems,” Expert Systems with Applications, vol.32, no.4, pp.1168–1182, 2007.

[95] P.J. Gawthrop, “Self-tuning PID controllers: Algorithms and implementation,” Automatic Control, IEEE Trans. on, vol.31, no.3, pp.201–209, 1986.

[96] W.D. Chang, R.C. Hwang, and J.G. Hsieh, "A self-tuning PID control for a class of nonlinear systems based on the lyapunov approach," Journal of Process Control, vol.12, no.2, pp.223–242, 2002.

[97] S. Lang, X. Gu, and T. Chai, "A multivariable generalized self-tuning controller with decoupling design," IEEE Trans. Automatic coltrol, vol.31, no.5, pp.475–477, 1994.

[98] L. Zhai and T. Chai, "Nonlinear decoupling PID control using neural nteworks and multiple models," Journal of Control Theory and Applications, vol.4, no.1, pp.62–69, 2006.

[99] L. Wang, Y. Cheng, and J. Hu, "Nonlinear adaptive control using a fuzzy switching mechanism based on improved quasi-ARX neural network," in Proc. of the 2010 International Joint Conference on Neural Networks (IJCNN), pp.1–7, 7 2010.

[100] L. Wang and J. Mendel, "Back-propagation fuzzy system as nonlinear dynamic system identifiers," in Proc. of IEEE Int. Conf. on Fuzzy Systems (San Diego, CA), pp.1409–1418, 1992.

[101] X. Shi, J. Hu, K. Kumamaru, and K. Hirasawa, "An improvement of quasi-arx modeling scheme for nonlinear systems using pca network," in Proc. of the 38th ISCIE International Symposium on Stochastic Systems Theory and Its Applications (SSS2006) (Nagano), 11 2006.

[102] E.Aarts and J.Korst, Simulated Annealing and Bosltzman Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing, John Wiley & Sons, 1989.

[103] M.A. KRAMER, "Nonlinear principal component analysis using autoassociative neural networks," AIChE Journal, vol.37, no.2, pp.233–243, 1991.

[104] F.M. Ham and I. Kostanic, Principles of Neurocomputing for Science and Engineering, McGraw-Hill, 2000.

[105] J. Hu, K. Hirasawa, C. Jin, and T. Matsuoka, "Control of nonlinear systems based on a probabilistic learning network," in Proc. of the 14th IFAC World Congress (IFAC'99) (Beijing), pp.447–452, 7 1999.

[106] M. Scholz, M. Fraunholz, and J. Selbig, Nonlinear principal component analysis: neural network models and applications, Springer Berlin Heidelberg, 2007.

Publication List

Journal Paper

1. Lan Wang, Yu Cheng and Jinglu Hu, “Stabilizing Switching Adaptive Control for Nonlinear System Based on Quasi-ARX RBFN Model”, *IEEJ Transactions on Electrical and Electronic Engineering (TEEE), Vol.7, No.4,* in press, 2012.

2. Lan Wang, Yu Cheng and Jinglu Hu, “A Quasi-ARX Neural Network with Switching Mechanism to Adaptive Control of Nonlinear Systems”, *SICE Journal of Control, Measurement, and System Integration (SICE JCMSI), Vol.3, No.4,* pp.246–252, 2010.

3. Lan Wang, Yu Cheng and Jinglu Hu, “A Quasi-ARX Model for Multivariable Decoupling Control of Nonlinear MIMO System”, *Mathematical Problems in Engineering,* in press, 2012.

4. Lan Wang and Jinde Cao, “Global Robust Point Dissipativity of Interval Neural Networks with Mixed Time-varying Delays”, *Nonlinear Dynamics, Vol.55, No.1-2*, pp. 169–178, 2009.

5. Yu Cheng, Lan Wang and Jinglu Hu, “A Two-step Scheme for Polynomial NARX Model Identification Based on MOEA with Pre-screening Process”, *IEEJ Transactions on Electrical and Electronic Engineering (TEEE), Vol.6, No.3,* pp.253–259, 2011.

6. Yu Cheng, Lan Wang and Jinglu Hu, “Quasi-ARX Wavelet Network for SVR Based Nonlinear System Identification”, *Nonlinear Theory and its Applications (NOLTA), IEICE, Vol.2, No.2,* pp.165–179, 2011.

7. Guanjun Wang, Jinde Cao and Lan Wang, “Global Dissipativity of Stochastic Neural Networks with Time Delay”, *Journal of the Franklin Institute, Vol.346, No.8*, pp. 794–807, 2009.

Conference Paper with Review

8. Lan Wang, Yu Cheng and Jinglu Hu, “Multivariable Self-Tuning Control for Nonlinear MIMO System Using Quasi-ARX RBFN Model”, in *Proc. of The 30th Chinese Control Conference (CCC’2011)*, pp. 3772–3776, Yantai, China, 2011.

9. Lan Wang, Yu Cheng and Jinglu Hu, “Adaptive Switching Control Based on Quasi-ARX RBFN Model”, in *Proc. of 2011 International Conference on Computers, Communications, Control and Automation (CCCA’2011)*, pp. 76–79, Hongkong, China, 2011.

10. Lan Wang, Yu Cheng and Jinglu Hu, “Nonlinear Adaptive Control Using Support Vector Regression Based on Improved Quasi-ARX Model”, in *Proc. of 010 International Conference on Modeling, Simulation and Control (ICMSC’10)*, pp.412–416, Cairo, Egypt, 2010.

11. Lan Wang, Yu Cheng and Jinglu Hu, “Nonlinear Adaptive Control Using a Fuzzy Switching Mechanism Based on Improved Quasi-ARX Neural Network”, in *Proc. of IEEE World Congress on Computational Intelligence 2010 (WCCI 2010) – The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, Barcelona, Spain, 2010.

12. Lan Wang, Yu Cheng and Jinglu Hu, “Quasi-ARX Neural Network and Its Application to Adaptive Control of Nonlinear Systems”, in *Proc. of The Fifteenth International Symposium on Artificial Life and Robotics(AROB 15th 10)*, pp. 577–580, Oita, Japan, 2010.

13. Lan Wang, Yu Cheng and Jinglu Hu, “Adaptive Control for Nonlinear Systems Based on Quasi-ARX Neural Network”, in *Proc. of World Congress on Nature and Biologically Inspired Computing (NaBIC 2009)*, pp.1548–1551, Coimbatore, India, 2009.

14. Lan Wang, Yu Cheng and Jinglu Hu, “An Improvement of Quasi-ARX Predictor to Control of Nonlinear Systems Using Nonlinear PCA Network”, in *Proc. of ICROS-SICE International Joint Conference 2009*, pp. 5095–5099, Fukuoka, Japan, 2009.