Founded 1905

Adaptive Neural Network Control of Discrete-time Nonlinear Systems

JIN ZHANG

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2004

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor, Dr. Shuzhi Sam Ge, for all the time and efforts he had spent on me. Without his expertise in control engineering and patient edification, this thesis would not have been possible. His guidance greatly helped and spurred me, not only in my research work but also in many other aspects of my life. My thanks also go to my supervisor, Prof. Tong Heng Lee, for his kind suggestions and help in my PhD study. Extra special thanks go to the National University of Singapore, for allowing me to undertake the research for this degree.

Secondly, I really appreciate the kind and tremendous help from my previous supervisors, Prof. Xingren Wang, Prof. Shuling Dai and Prof. Qin Feng. When I was in the advanced simulation technology laboratory, Beijing University of Aeronautics and Astronautics, I learnt a lot from them.

I am also grateful to all other staff and students in the Control and Mechatronics Laboratory, Department of Electrical and Computer Engineering, National University of Singapore, who have made my working time pleasant and enjoyable. Especially, I would like to thank Mr. Guangyong Li, Dr. Jing Wang, Dr. Tao Zhang, Dr. Cong Wang, Dr. Youjing Cui, Dr. Zhuping Wang, Dr. Fan Hong, Mr. Feng Guan, Mr. Tok Meng Yong, Mr. Peng Xiao and Ms. Xin Liu for their kind help and instructive comments during my research process. Thank the staff, Mr. Tang Kok Zuea and Mr. Tan Chee Siong, who have made my working environment comfortable.

Finally, I really appreciate my parents, Mr. Sheng Zhang and Mrs. Qiufang Jiao, who brought me to this world, and taught me to know this world when I was a little child. I can feel their endless love no matter where I am and at anytime. To my brothers, Mr. Yu Zhang and Mr. Heng Zhang, my sister-in-law, Yuan Lin and my little nephew, Keming, I really enjoy the happy times being with them. At last, I would like to thank my family again, without their love, the life is meaningless to me.

Contents

Acknowledgements ii

Contents iii

Summary vii

List of Figures ix

List of Tables xii

1 Introduction 1

1.1 Adaptive Neural Network Control of Nonlinear Systems 2

1.1.1 Neural Networks 2

1.1.2 Adaptive NN Control of Continuous-time Systems 7

1.1.3 Adaptive NN Control of Discrete-time Systems 9

1.2 Objectives of the Thesis 13

1.3 Contributions of the Thesis 14

1.4 Organization of the Thesis 16

2 NN Control of Non-affine SISO Systems **17**

2.1 Introduction 17

2.2 Problem Formulation 18

2.3 Projection Algorithm 23

2.4 Controller Design 25

2.4.1 RBF NN Control 25

2.4.2 MNN Control 31

2.5 Numerical Simulation 42

2.5.1 RBF Control Simulation 42

2.5.2 MNN Control Simulation 43

2.6 Application to Practical CSTR Systems 44

2.6.1 Non-affine CSTR System 45

2.6.2 Affine CSTR System 50

2.7 Conclusion 55

3 NN Control of MIMO Systems with Triangular Form Inputs **60**

3.1 State Feedback Control 60

3.1.1 MIMO System Dynamics 62

3.1.2 Causality Analysis and System Transformation 65

3.1.3 Controller Design and Stability Analysis 71

3.1.4 Simulation 88

3.2 Output Feedback Control 90

3.2.1 MIMO System Dynamics 92

3.2.2 System Coordinate Transformation 93

3.2.3 Controller Design and Stability Analysis 109

3.2.4 Simulation 119

3.3 Conclusion 121

4 NN Control of NARMAX MIMO Systems 127

4.1 Affine MIMO NARMAX Systems 127

4.1.1 Introduction 127

4.1.2 System Dynamics and Stability Notions 128

4.1.3 Controller Design and Stability Analysis 132

4.1.4 Simulation 139

4.2 Non-affine MIMO NARMAX Systems 140

4.2.1 Introduction 140

4.2.2 MIMO System Dynamics 140

4.2.3 Stability Analysis 146

4.2.4 Simulation 151

4.3 Conclusion 152

5 Conclusions and Further Research 158

5.1 Conclusions 158

5.2 Further Research 160

A BIBO Stability and PE Condition 162

A.1 BIBO Stability 162

A.2 Persistent Exciting Condition 162

Bibliography **163**

Author's Publications **177**

Summary

In recent years, adaptive control for nonlinear systems has been studied by many researchers. State/output feedback, feedback linearization techniques, neural network (NN) control schemes and many other techniques have been studied. These elegant methods have been applied to different kinds of complex continuous-time nonlinear systems. However, for discrete-time nonlinear systems, especially for complex discrete-time nonlinear systems, those available schemes normally cannot be directly implemented. Therefore, effective control of complex discrete-time systems is a problem that needs to be further investigated.

The purpose of this thesis is to develop effective adaptive control schemes for complex nonlinear discrete-time systems using neural networks. Not only single-input single-output (SISO) discrete-time systems are studied in this thesis, but also multi-input multi-output (MIMO) discrete-time systems are studied in this thesis. Furthermore, besides affine discrete-time systems, for which feedback linearization technique can be implemented, non-affine discrete-time systems are also investigated in this thesis.

In general, the effective control schemes proposed in continuous-time domain cannot be directly implemented in discrete-time systems due to some technical difficulties, such as the lack of applicability of Lyapunov techniques and loss of linear parameterizability during the linearization process, and discrete-time adaptive control design is far more complex than continuous-time design, due primarily to the fact that discrete-time Lyapunov differences are quadratic in the state first difference, while for continuous-time systems the Lyapunov derivative is linear in the state derivative. In this thesis, effective adaptive neural network control schemes are developed for five different kinds of discrete-time nonlinear systems. They are SISO NARMAX

(Nonlinear Auto Regressive Moving Average with eXogenous inputs) systems, MIMO discrete-time systems with triangular form input and unknown disturbances in state space description, MIMO discrete-time systems with triangular form input and strict feedback form subsystems in state space description, MIMO NARMAX affine systems and MIMO NARMAX non-affine systems, which cover a wide class of nonlinear discrete-time systems. Noting the good approximation ability of neural networks, in this thesis, by using neural networks as the emulators of the explicit/implicit desired controls, stable adaptive controls are developed for those systems respectively. Single layer neural networks, including radial basis function (RBF) neural networks and high order neural networks (HONN), as well as multi-layer neural networks (MNN) are used. Lyapunov technique is used as the tool in system stability analysis. Backstepping design, state feedback and output feedback control schemes are implemented. Numerical simulations are also carried out to show the effectiveness of those proposed control schemes.

By using neural networks as the emulators of the desired controls and using Lyapunov method as the tool in system stability analysis, in this thesis, the five kinds of systems studied are proved to be semi-globally uniformly ultimately bounded (SGUUB). All the signals in the closed-loop systems are proved to be bounded. The discrete-time projection algorithm, the high order weight tuning algorithm proposed and the use of backstepping method in a nested manner are proved to be effective. Furthermore, the proposed control method for SISO system is applied to two kinds of practical chemical processes, continuous tank reactor systems (CSTR). The numerical simulation results show the effectiveness of the method.

In general, in this thesis, adaptive NN control schemes for different kinds of nonlinear discrete-time systems are investigated. Backstepping design, state feedback, output feedback control are investigated respectively. Neural networks are used to approximate the explicit/implicit desired controls. By using Lyapunov technique, the closed-loop systems are proved to be SGUUB. Numerical simulations are carried out for fictitious systems as well as practical processes.

List of Figures

2.1 Continuously Stirred Tank Reactor System . . . 46
2.2 Exothermic Reaction in a CSTR . . . 51
2.3 RBF Control - Tracking Performance . . . 56
2.4 RBF Control - Input Trajectory . . . 56
2.5 RBF Control - Weight Norm $\|\hat{W}\|_2$ . . . 56
2.6 MNN Control - Tracking Performance . . . 57
2.7 MNN Control - Input Trajectory . . . 57
2.8 MNN Control - Weight Norm $\|\hat{W}\|_2$ and $\|\hat{V}\|_F$ . . . 57
2.9 Non-affine CSTR - Tracking Performance . . . 58
2.10 Non-affine CSTR - Weight Norm $\|\hat{W}\|$ and $\|\hat{V}\|_F$ . . . 58
2.11 Non-affine CSTR - Control Trajectory . . . 58
2.12 Affine CSTR - Tracking Performance . . . 59
2.13 Affine CSTR - Weight Norm $\|\hat{W}\|$ and $\|\hat{V}\|_F$ . . . 59
2.14 Affine CSTR - Control Trajectory . . . 59

3.1 Example: y_1 and y_{d_1} . . . 72
3.2 Example: y_2 and y_{d_2} . . . 72

3.3 State Feedback Control - Control System Structure 73

3.4 Example: y_1 and y_{d_1} 108

3.5 Example: y_2 and y_{d_2} 108

3.6 Output Feedback Control - Control System Structure 109

3.7 State Feedback Control - Tracking Performance $y_1(k)$ and $y_{d_1}(k)$ . . . 123

3.8 State Feedback Control - Tracking Performance $y_2(k)$ and $y_{d_2}(k)$ . . . 123

3.9 State Feedback Control - Control Inputs $u_1(k)$ and $u_2(k)$ 123

3.10 State Feedback Control - Weight Norms $\|\hat{W}_{12}(k)\|$ and $\|\hat{W}_{22}(k)\|$ . . . 124

3.11 State Feedback Control - Error dynamics 124

3.12 Output Feedback Control - Tracking Performance $y_1(k)$ and $y_{d_1}(k)$ . . 125

3.13 Output Feedback Control - Tracking Performance $y_2(k)$ and $y_{d_2}(k)$ . . 125

3.14 Output Feedback Control - Control Inputs $u_1(k)$ and $u_2(k)$ 125

3.15 Output Feedback Control - Weight Norms $\|\hat{W}_1(k)\|$ and $\|\hat{W}_2(k)\|$ . . . 126

3.16 Output Feedback Control - Error dynamics 126

4.1 Affine NARMAX - Tracking Performance $y_1(k)$ and $y_{d_1}(k)$ 154

4.2 Affine NARMAX - Tracking Performance $y_2(k)$ and $y_{d_2}(k)$ 154

4.3 Affine NARMAX - Control Inputs $u_1(k)$ and $u_2(k)$ 154

4.4 Affine NARMAX - Weight Norm $\|\hat{W}(k)\|_F$ 155

4.5 Affine NARMAX - Error dynamics 155

4.6 Non-affine NARMAX - Tracking Performance $y_1(k)$ and $y_{d_1}(k)$ 156

4.7 Non-affine NARMAX - Tracking Performance $y_2(k)$ and $y_{d_2}(k)$ 156

4.8 Non-affine NARMAX - Control Inputs $u_1(k)$ and $u_2(k)$ 156

4.9 Non-affine NARMAX - Weight Norm $\|\hat{W}(k)\|_F$ 157

4.10 Non-affine NARMAX - Error dynamics 157

List of Tables

2.1 Nomenclature List (Non-affine CSTR System) 47

2.2 Nomenclature List (Affine CSTR System) 52

3.1 A Simple Example - System Variation 72

3.2 A Simple Example - System Variation 108

Chapter 1

Introduction

In recent years, adaptive control of nonlinear systems has received much attention and many significant advances have been made in this field. Due to the complexity of nonlinear systems, research on adaptive nonlinear control is still focusing on development of the fundamental methodologies. A great number of research articles, books, reporting inventions, control applications within the fields of adaptive, neural network control and fuzzy logic systems, have been published in various journals and conferences. Making a complete description for all aspects of adaptive control techniques is difficult due to the vast amount of literature. This thesis investigates adaptive control of nonlinear discrete-time systems using neural networks, effective neural network control schemes, corresponding weights update laws and closed-loop systems stability are investigated for several kinds of nonlinear SISO/MIMO, affine/non-affine discrete-time systems.

This chapter is organized as follows. Firstly, considering that neural networks are used as an effective tool in approximation based nonlinear control in this thesis, the definitions as well as the properties of neural networks are briefly reviewed in Section 1.1.1. Then, a brief introduction on adaptive control of continuous-time and discrete-time systems is given to provide an outline of the historical development and present status in these areas in Sections 1.1.2 and 1.1.3. Finally, the objectives, contributions and organization of this thesis are presented in Sections 1.2, 1.3 and 1.4 respectively.

1.1 Adaptive Neural Network Control of Nonlinear Systems

1.1.1 Neural Networks

Artificial neural networks (ANNs) are inspired by biological neural networks, which usually consist of a number of simple processing elements, call neurons, that are interconnected to each other. In most cases, one or more layers of neurons are connected to each other in a feedback or recurrent way. Since McCulloch and Pitts [1] introduced the idea of studying the computational abilities of networks composed of simple models of neurons in the 1940s, neural network techniques have undergone great development and have been successfully applied in many fields such as learning, pattern recognition, signal processing, modelling and system control. The approximation abilities of neural networks have been proven in many research works [2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. The major advantages of highly parallel structure, learning ability, nonlinear function approximation, fault tolerance and efficient analog VLSI implementation for real-time applications, greatly motivate the usage of neural networks in nonlinear system control and identification.

The early works of neural network applications for controller design were reported in [12, 13]. The popularization of backpropagation (BP) algorithm [14] in the late 1980s greatly boosted the development of neural control and many neural control approaches have been developed [15, 16, 17, 18, 19]. Most early works on neural control described creative ideas and demonstrated neural controllers through simulation or by particular experimental examples, but were short of analytical analysis on stability, robustness and convergence of the closed-loop neural control systems. The theoretical difficulty arose mainly from the nonlinearly parametrized networks used in the approximation. The analytical results obtained in [20, 21] showed that using multi-layer neural networks as function approximators guaranteed the stability and convergence results of the systems when the initial network weights chosen were sufficiently close to the ideal weights. This implies that for achieving a stable neural control system using the gradient learning algorithms such as BP, sufficient off-line training must be performed before neural network controllers are put into the systems.

Because their universal approximation abilities, parallel distributed processing abilities, learning, adaptation abilities, natural fault tolerance and feasibility for hardware implementation, neural networks are made one of the effective tools in approximation based control problems. Recently neural networks have been made particularly attractive and promising for applications to modelling and control of nonlinear systems. For neural network controller design of general nonlinear systems, several researchers have suggested to use neural networks as emulators of inverse systems. The main idea is that for a system with finite relative degree, the mapping between system input and system output is one-to-one, thus allowing the construction of a "left-inverse" of the nonlinear system using NN. Using the implicit function theory, the NN control methods proposed in [22, 21] have been used to emulate the "inverse controller" to achieve the desired control objectives. Based on this idea, an adaptive controller has been developed using high order neural networks with stable internal dynamics in [23] and applied in [24]. As an alternative, neural networks have been used to approximate the implicit desired feedback controller (IDFC) in [25]. A multi-layer neural network control method for SISO non-affine systems without zero dynamics was also proposed in that paper. In this thesis, we mainly investigate the implementation of neural networks as function approximators for the desired feedback control, which can realize exact tracking.

Except that neural networks can be used as function approximators to emulate the "inverse" control in nonlinear system research, there are many other areas, in which neural networks play an important role. For example, neural networks combined backstepping design are reported in [26, 27, 28, 29, 30, 31, 32], using neural networks to construct observers can be found in [33, 34], neural network control in robot manipulators are reported in [35, 36, 37, 38, 39, 40], neural identification of chemical processes by using dynamics neural networks can be found in [41, 42, 43], neural control for distillation column are reported in [44, 45], etc. It should be noted, similar to neural networks, fuzzy system is another kind of system, which has "intelligence" and has attracted many research interests. It can also be used as function approximators. Research works in fuzzy system can be found in [46, 47, 48].

In this thesis, HONN, RBF and MNN are used, which are three kinds of frequently used neural networks in nonlinear system control and identification [35, 49, 36, 50, 51, 52]. HONN and RBF networks can be considered as two-layer networks in which the hidden layer performs a fixed nonlinear transformation with no adjustable parameters, i.e., the input space is mapped on to a new space. The output layer then combines the outputs in the latter space linearly. Therefore they belong to a class of linearly parameterized networks. MNN, which are also called multi-layer perception in the literature, is a static feedforward network that consists of a number of layers, and each layer consists of a number of McCulloch-Pitts neurons [1]. Once the neurons have been selected, only the adjustable weights have to be determined to specify the networks completely. Since each node of any layer is connected to all the nodes of the following layer, it follows that a change in a single parameter at any one layer will generally affect all the outputs in the following layers. MNNs with one or more hidden layers are capable of approximating any continuous nonlinear function, which was obtained independently by [4, 2, 5]. This important character makes it one of the most widely used neural networks in system modelling and control.

Specifically, in this thesis, the following approximation representations of HONN, RBF and MNN are used:

<u>High Order Neural Networks:</u> Consider the following HONN [53]

$$\begin{aligned}
\phi(W, z) &= W^T S(z), \qquad W \in R^{l\times p} \quad \text{and} \quad S(z) \in R^l, \\
S(z) &= [s_1(z), s_2(z), ..., s_l(z)]^T, \\
s_i(z) &= \prod_{j\in I_i} [s(z_j)]^{d_j(i)}, \quad i = 1, 2, ..., l
\end{aligned}$$

where $z = [z_1, z_2, \cdots, z_q]^T \in \Omega_z \subset R^q$, positive integer l denotes the NN node number, and p is the dimension of function vector, $\{I_1, I_2, ..., I_l\}$ is a collection of l not-ordered subsets of $\{1, 2, ..., q\}$ and $d_j(i)$ are non-negative integers, W is an adjustable synaptic weight matrix, $s(z_j)$ is chosen as hyperbolic tangent function

$$s(z_j) = \frac{e^{z_j} - e^{-z_j}}{e^{z_j} + e^{-z_j}}$$

For a desired function $u^*(z)$, there exist ideal weights W^* such that the smooth function u^* can be approximated by an ideal NN on a compact set $\Omega_z \subset R^q$

$$u^* = W^{*T} S(z) + \epsilon_z \tag{1.1}$$

where ϵ_z is the bounded NN approximation error satisfying $|\epsilon_z| \leq \epsilon_0$ on the compact set, which can be reduced by increasing the number of the adjustable weights. The ideal weight matrix W^* is an "artificial" quantity required for analytical purpose, and is defined as that minimizes $|\epsilon_z|$ for all $z \in \Omega_z \subset R^q$ in a compact region, i.e.,

$$W^* \triangleq arg \min_{W \in R^{l \times m}} \left\{ \sup_{z \in \Omega_z} |u^* - W^T S(z)| \right\}, \quad \Omega_z \subset R^q \tag{1.2}$$

In general, the ideal NN weight matrix, W^*, is unknown though constant, its estimate, $\hat{W}$, should be used for controller design which will be discussed later.

Radial Basis Function Neural Networks: Considering the following RBF [35, 54] NN used to approximate a function $h(z) : R^q \to R$,

$$h_{nn}(z) = W^T S(z) \tag{1.3}$$

where the input vector $z \in \Omega_z \subset R^q$ where q is the neural network input dimension. Weight vector $W = [w_1, w_2, \cdots, w_l]^T \in R^l$, the NN node number $l > 1$, and $S(z) = [s_1(z), \cdots, s_l(z)]^T$, with $s_i(z)$ being chosen as the commonly used Gaussian functions, which is in the following form

$$s_i(z) = \exp\left[\frac{-(z-\mu_i)^T(z-\mu_i)}{\eta_i^2}\right], i = 1, 2, ..., l \tag{1.4}$$

where $\mu_i = [\mu_{i1}, \mu_{i2}, \cdots, \mu_{iq}]^T$ is the center of the receptive field and η_i is the width of the Gaussian function.

It has been proven that network (1.3) can approximate any continuous function over a compact set $\Omega_z \subset R^q$ to arbitrary accuracy as

$$h(z) = W^{*T} S(z) + \epsilon_z, \ \forall z \in \Omega_z \tag{1.5}$$

where W^* is ideal constant weights, and ϵ_z is the approximation error.

The ideal weight vector W^* is an "artificial" quantity required for analytical purposes. W^* is defined as the value of W that minimizes $|\epsilon_z|$ for all $z \in \Omega_z$ in a compact region, i.e.,

$$W^* \triangleq arg \min_{W \in R^l} \left\{ \sup |h(z) - W^T S(z)| \right\}, \quad z \in \Omega_z \tag{1.6}$$

It should be noted that, though HONN and RBF are used for analysis in this thesis, they may be replaced by any other linear approximators, such as spline functions [55] or fuzzy systems [56], which have the similar properties, while the stability and performance properties of the adaptive system are still valid.

Multi-layer Neural Networks: When linearity in the parameters holds, the rigorous results of adaptive control become applicable for the NN weight tuning, and eventually result in a stable closed-loop system. However, the same is not true for the multi-layer case, where the unknown parameters go through nonlinear activation functions. This structure not only offers a more general case than the previous one, allowing application to a much larger class of systems, but also avoids some limitations, such as defining a basis function set or choosing some centers and variations of radial basis type of activation functions. In [2, 5, 4], one of the important character of MNN, that MNN with one or more hidden layers is capable of approximating any continuous nonlinear function, was obtained independently.

In this thesis, the following MNN is used [50]. Define

$$\begin{aligned} \bar{Z} &= [\bar{z}_1, \bar{z}_2, \cdots, \bar{z}_{n+1}]^T = [z^T, 1]^T \in R^{n+1} \\ V &= [v_1, v_2, \cdots, v_l] \in R^{(n+1)\times l} \end{aligned}$$

with $v_i = [v_{i1}, v_{i2}, \cdots, v_{in+1}]^T,\ i = 1, 2, \cdots, l$. The term $\bar{z}_{n+1} = 1$ in input vector $\bar{z}$ allows one to include the threshold vector $[\theta_{v1}, \theta_{v2}, \cdots, \theta_{vl_1}]^T$ as the last column of V^T, so that V contains both the weights and thresholds of the first-to-second layer connections. Then the MNN can be expressed as

$$\begin{aligned} g_{nn}(W, V, Z) &= W^T S(V^T \bar{Z}) \\ S(V^T \bar{Z}) &= [s(v_1^T \bar{Z}), s(v_2^T \bar{Z}), ..., s(v_l^T \bar{Z}), 1]^T \\ W &= [w_1, w_2, \cdots, w_{l+1}]^T \in R^{l+1} \end{aligned} \tag{1.7}$$

where the last element in $S(V^T \bar{Z})$ incorporates the threshold θ_w as w_{l+1} of weight W. Any tuning of W and V then includes tuning of the thresholds as well [57]. Then in (1.7), the total number of the hidden-layer neurons is $l + 1$ and the number of input-layer neurons is $n + 1$. It is known that there are ideal constant W^* and V^* such that

$$\max_{Z\in\Omega_z} \left| g(Z) - g_{nn}(W^*, V^*, Z) \right| < \mu \le \bar{\mu}$$

with constant $\bar{\mu} > 0$ for all $Z \in \Omega_z$. The ideal weights W^* and V^* are defined as

$$(W^*, V^*) : \;=\; arg \min_{(W,V)} \left\{ \sup_{z \in \Omega_z} \left| W^T S(V^T \bar{Z}) - g(Z) \right| \right\} \tag{1.8}$$

In general, W^* and V^* are unknown and need to be estimated in function approximation. Let $\hat{W}$ and $\hat{V}$ be the estimates of W^* and V^*, respectively, and the weight estimation errors be $\tilde{W} = \hat{W} - W^*$ and $\tilde{V} = \hat{V} - V^*$. It can be seen that MNNs are nonlinearly parametrized function approximators, i.e., the hidden layer weight V^* appears in a nonlinear fashion.

1.1.2 Adaptive NN Control of Continuous-time Systems

Though the main objective of this thesis is to investigate adaptive neural network control for non-linear discrete-time systems, it is necessary to briefly review the achievements obtained in continuous-time domain, in which many classical and elegant methods have been developed, and are ready for discrete-time extension.

Research in adaptive control for continuous-time nonlinear systems have a long history of intense activities that involve rigorous problems for formulation, stability proof, robustness design, performance analysis and applications. The advances in stability theory and the progress of control theory in the 1960s improved the understanding of adaptive control and contributed to a strong interest in this field. By the early 1980's, several adaptive approaches have been proven to provide stable operation and asymptotic tracking. The adaptive control problem since then, was rigorously formulated and several leading researchers have laid the theoretical foundations for many basic adaptive schemes. In the mid 1980s, research of adaptive control mainly focused on the robustness problem in the presence of unmodeled dynamics and/or bounded disturbances. A number of redesigns and modifications were proposed and analyzed to improve the robustness of the adaptive controllers, e.g., by applying normalization techniques in controller design and modification of adaptation laws using projection method [58], dead zone modifications [59, 60], ϵ-modification [61] and σ-modification [62].

In last decades, in continuous-time domain, feedback linearization technique [63, 64, 65], backstepping design [66], neural network control and identification [35, 50] and tuning function design have attracted much attention. Many remarkable results in this area have been obtained [67, 68, 69, 70, 56, 47, 71, 72, 73, 74, 75]. In the following, some works for SISO and MIMO continuous-time systems are listed.

For SISO continuous-time nonlinear systems, the feasibility of applying neural networks for modelling unknown functions in dynamic systems has been demonstrated in several studies. It was shown that for stable and efficient on-line control using the BP learning algorithm, the identification of systems must be sufficiently accurate before control action is initiated [41, 21, 15]. Recently, several good NN control approaches have been proposed based on Lyapunov's stability theory [57, 76, 77, 78, 50]. One main advantage of these schemes is that the adaptive laws are derived based on the Lyapunov synthesis method and therefore guaranteed the stability of continuous-time systems without the requirement of off-line training. For strict-feedback nonlinear SISO system, adaptive control scheme is still an active topic in nonlinear system control area. Using the backstepping design procedures, a systematic approach of adaptive controller design was presented for a class of nonlinear systems transformable to a parametric strict-feedback canonical form, which guarantees the global and asymptotic stability of the closed-loop system [79, 66, 50]. Using the implicit function theory, the NN control methods proposed in [22, 21] have been used to emulate the "inverse controller" to achieve the desired control objectives. Based on this idea, an adaptive controller has been developed using high order neural networks with stable internal dynamics in [23] and applied in [24]. As an alternative, neural networks have been used to approximate the implicit desired feedback controller in [25]. Multi-layer neural network control method was also proposed for SISO non-affine systems without zero dynamics in that paper. Furthermore, previous works on nonlinear non-affine systems controller design [80] proposed a new control law for non-affine nonlinear system for a class of deterministic time-invariant discrete system which is free of the usual restrictions, such as minimum phase, known plant states etc. A general form of control structure of adaptive feedback linearization is $u = \hat{N}(x)/\hat{D}(x)$, where $\hat{D}(x)$ must be bounded away from zero to avoid the possible controller singularity problem [77]. The approach is only applicable to the class of systems whose dynamics are linear-in-the-parameters and satisfy the so-called matching conditions. The matching condition was relaxed to the extended matching condition in [81] and [82], and the extended matching barrier was broken in [83] by using adaptive backstepping design [84, 66, 50]. For single input multi outputs systems, some results can be found in [85, 86].

For MIMO continuous-time nonlinear systems, there are few results available, due primarily to the difficulty in handling the coupling matrix between different inputs. In [87], a stable neural network adaptive controller was developed for a class of nonlinear multi-variable systems, the control inputs are in triangular form and integral Lyapunov function was used to analyze the stability. In [88], a numerically robust approximate algorithms was given for input-output decoupling nonlinear MIMO systems. Several algorithms have been proposed in the literature for solving the problem of exact decoupling for nonlinear MIMO systems, see for examples [89, 90, 91, 92]. All these algorithms need the determination of the inverse, the so-called decoupling matrix. In [93], the problem of semi-global robust stabilization was investigated for a class of MIMO uncertain nonlinear system, which cannot be transformed into lower dimensional zero dynamics representation, via change of coordinates or state feedback. Both the partial state and dynamic output controllers were explicitly constructed via the design tools such as semi-global backstepping and high-gain observer. In [94], an adaptive fuzzy systems approach to state feedback input-output linearizing controller was outlined. The analysis was based on a general nonlinear MIMO system, with minimum phase zero dynamics and uncertainties satisfying the matching condition.

1.1.3 Adaptive NN Control of Discrete-time Systems

While fundamental physical models are almost always developed in continuous-time, computer based process control systems function in discrete-time: measurements are made and control actions are taken at discrete time instant, seconds, minute, hours, or days apart. In addition, the input output data available for model identification is generally only available at discrete time instant. It is usually easier to identify discrete-time models and use these as a basis to design discrete-time control systems for computer implementation. This observation motivates us to concentrate on discrete-time models, despite certain inherent differences between the behavior of discrete-time models and continuous-time models. In this section, the development in adaptive NN control of discrete-time nonlinear systems is briefly reviewed.

The design methodologies for both continuous-time systems and discrete-time systems are very different. Similar formulations in continuous-time and discrete-time domains may describe two totally different systems. Many properties in continuous-time domain may disappear in discrete-time domain, and vice versa. The same concepts in continuous-time and discrete-time domains may have different meanings. For example, the *relative degrees* defined for continuous-time systems [65] and discrete-time systems have totally different physical explanations [95]. As a consequence, results obtained in continuous-time domain may not be obtainable in discrete-time domain. Therefore, it is necessary to investigate them separately. Because the methods obtained in continuous-time systems cannot be directly applied to discrete-time systems due to some technical difficulties, such as lack of applicability of Lyapunov techniques [96], the loss of linear parameterizability during the linearization process. Furthermore, discrete-time adaptive control design is more complex than continuous-time design, due primarily to the fact that discrete-time Lyapunov differences are quadratic in the state first difference, while for continuous-time systems the Lyapunov derivative is linear in the state derivative. This has led to the traditional techniques where the parameter identification problem is decoupled from the control problem using so-called "certainty equivalence" assumptions. Some of the previous results in nonlinear discrete-time NN control are listed as follows.

For SISO discrete-time nonlinear systems, some good NN controllers have been obtained. In [20], a specific class of affine nonlinear systems was investigated. The plant under study was an unknown feedback-linearizable discrete-time system, represented by an input-output model. Single layer neural networks were used to model the unknown system and to generate the feedback control. Based on the error between plant output and reference signal, the neural network weights were updated, and local convergence result was given. In [97], direct control of a general nonlinear dynamical system with only weak assumptions about the order and relative degree of the plant was discussed based on implicit function theory. The neural network control method was firstly discussed for first order discrete-time nonlinear system, and then the control scheme was generalized to high order discrete-time nonlinear system. Recently, discrete-time systems transformable to the parametric-strict-feedback form and the parametric-pure-feedback form were studied in [98]. By using a time varying mapping, the noncausal problem was elegantly solved in the backstepping design procedures. The results therein were further extended to cases with time-varying parameters and nonparametric uncertainties in [99]. However, for strict-feedback nonlinear systems in a more general description form, the control construction still remains an open problem. In [21], input output based neural network control was studied for a class of nonlinear dynamical discrete-time systems. Further theoretical foundation and insights, which are essential for the design of neural network control based on inverse controller, were provided in [95], in which the relative degree of discrete-time systems was well explained. In [100], a direct adaptive NN control was presented for a class of discrete-time unknown nonlinear systems with general relative degree in the presence of bounded disturbances. The NN control scheme can be applied to the system without off-line training. In the study of nonlinear discrete-time control, one of the most popular representation is the NARMAX model [101]. As only input and output sequences appear in the NARMAX model, it is straightforward to use approximation based method to construct the "inverse" of the system to emulate the desired control input, which can then drive the system output to the desired trajectory. Studies on discrete-time NARMAX systems can be found in [102, 103, 104, 105, 106]. In [107], robust control was given for a class of "set-valued" discrete-time dynamical systems subject to persistent bounded noises. In [108], feedback limitations of linear sampled-data periodic digital control was investigated. In [99], by using the backstepping procedures with parameter projection update laws, robust adaptive control was designed for systems with the *priori* range of unknown time-varying parameters. In [109], a systematic design method was given for global stabilization and tracking of discrete-time output feedback nonlinear systems with unknown parameters. In [110], localization based switching adaptive control for time-varying discrete-time systems was investigated.

Compared with those results obtained for SISO discrete-time systems, fewer results can be found for MIMO discrete-time system. For MIMO nonlinear discrete-time systems, how to tune the NN weights is still an open problem, especially when there exists unknown strong inter connections between subsystems. In [111], the NN control was studied for a very special class of discrete-time MIMO nonlinear systems with relative degree of one and without any inter connections between subsystems. In [112], a new controller design method for non-affine nonlinear discrete-time system was presented. The control law is simple to implement and is based on a novel linearization of the input-output model. Extensive empirical studies have confirmed that the control law can be used to control a relative general class of highly nonlinear MIMO plants. In [113], stable NN-based adaptive control for a class of MIMO sampled-data nonlinear systems was studied. The control scheme is an integration of an NN approach and the variable structure method.

In general, for both continuous-time domain and discrete-time domain, especially for complex nonlinear systems, Lyapunov method plays an important role. The mainly differences in the design and analysis between continuous-time domain and discrete-time domain can be summarized as follows:

- In continuous-time domain, Lyapunov function is linear in the state derivative, however, in discrete-time domain, Lyapunov differences are quadratic in the state first difference;
- In continuous-time domain, there are many successful design methods that have been reported in previous literatures, such as backstepping method, feedback linearization techniques etc. However, for discrete-time domain, similar techniques cannot be directly implemented.

The new challenges in the control of nonlinear discrete-time systems can be summarized as follows:

- For complex discrete-time nonlinear systems, such as non-affine systems, MIMO systems, little results have been obtained;
- Though backstepping design has been proved to be successful in continuous-time domain, no similar design technique has been proposed for discrete-time systems due to the noncausal problem;

- For continuous-time systems, there are projection algorithms which restrict parameter estimation in a set, however, for discrete-time systems, no similar results have been obtained;

- For output feedback control of discrete-time nonlinear systems, further investigation should be carried out;

- For τ-step ahead discrete-time NARMAX models, usually one step ahead parameter update is not applicable. High order parameter update laws maybe effective in solving this kind of systems.

1.2 Objectives of the Thesis

In general, the objective of this thesis is to develop constructive and systematic neural adaptive control methods for discrete-time nonlinear systems.

The first objective of this thesis is to investigate direct adaptive NN control scheme for a class of discrete-time SISO non-affine nonlinear systems. Implicit function theorem is used to prove the existence and uniqueness of the implicit desired feedback control. Based on the input-output model, RBF neural networks and MNN are used to emulate the implicit desired feedback control respectively. For the MNN control, the proposed projection algorithms are used to guarantee the boundedness of the neural network weights. The closed-loop systems is proved to be SGUUB if the design parameters are suitably chosen under certain mild conditions.

The second objective is to investigate adaptive NN control scheme for nonlinear MIMO discrete-time systems with triangular form input. Firstly, a class of MIMO systems with each subsystem in strict feedback form is studied. The lengths of different subsystems may be different. Unknown bounded disturbances are also considered. Through coordinate transformation, the MIMO system is firstly transformed into Sequential Decrease Cascade Form (SDCF), which avoids the causality problem often met in discrete-time nonlinear system control. Then, by using backstepping design technique in a nested manner and using HONN as emulators of the desired virtual and practical controls, an effective neural network control scheme with corresponding weight update laws are developed. Noting that the developed state feedback scheme needs all the system states are available, subsequently, a relative simple NN control method is proposed for a class of similar systems by using output feedback, which is easier for practical implementation. Compared with the MIMO systems in state feedback control, in output feedback part, the lengths of each subsystems are required to be the same. Furthermore, disturbances are neglected due to the difficulty met in coordinate transformation. In the output feedback control part, firstly, the MIMO system is transformed into input-output representation with the triangular form input structure unchanged. By using HONNs as the emulators of the desired controls, an effective output feedback control scheme with corresponding weight update laws are developed by using backstepping design technique. The closed-loop system is proved to be SGUUB by using Lyapunov method. The output tracking errors are guaranteed to converge into a compact set whose size is adjustable, and all the other signals in the closed-loop system are proved to be bounded.

The third objective of this thesis is to investigate adaptive NN control schemes for MIMO NARMAX models. Two classes of MIMO NARMAX systems are studied. Firstly, direct adaptive neural network control is studied for a class MIMO nonlinear affine systems based on input-output discrete-time model with unknown interconnections between subsystems. By finding an orthogonal matrix to tune the NN weights, the closed-loop system is proven to be SGUUB. The control performance of the closed-loop system is guaranteed by suitably choosing the design parameters. Then adaptive NN control scheme is developed for a class of MIMO non-affine NARMAX systems, with triangular form inputs. By using implicit function theorem, the existence of the implicit desired feedback control is proved. Then HONNs are used as the emulators of the desired controls. The stability of the closed-loop system is proved by Lyapunov method.

1.3 Contributions of the Thesis

In this thesis, several neural network control schemes are investigated for different kinds of discrete-time nonlinear systems. They can be classified as follows:

T1: SISO non-affine nonlinear NARMAX systems;

T2: MIMO nonlinear systems in state space representation with unknown disturbances and different subsystem lengths (state feedback);

T3: MIMO nonlinear systems in state space representation with each subsystem in strict feedback form (output feedback);

T4: MIMO affine NARMAX systems with disturbances;

T5: MIMO non-affine NARMAX systems.

The contributions for each type of system have been summarized as follows:

T1: The main contributions are: (i) provide an effective neural network control method for non-affine nonlinear discrete-time systems which feedback linearization method is of no use; (ii) propose a different kind of neural network weight update law for discrete-time systems; (iii) propose a modified discrete-time projection algorithm compare to continuous-time projection algorithm used in [114]; and (iv) using multilayer neural networks to emulate the implicit desired feedback control of non-affine discrete-time systems, which is not only a challenging topic but also of academic interest.

T2: The main contributions are: (i) an effective neural network control scheme is proposed for a class of nonlinear MIMO system with triangular form inputs, for which feedback linearization cannot be applied; and (ii) by using neural networks as the emulators of the desired virtual controls and desired practical controls, and embedded using backstepping design, the closed-loop system is proved to be SGUUB in the presence of unknown bounded disturbances.

T3: The main contributions are: (i) an effective NN control scheme is developed for a class of complex nonlinear discrete-time non-affine MIMO systems in state space representation, for which, feedback linearization method cannot be implemented; (ii) only input and output sequences are used to construct the stable control, which is simple and easy to be implemented in practical applications; (iii) a system transformation technique is proposed, which can transform the system from state space description into input output representation, which extends our previous works in [115] from SISO systems to MIMO systems; and (iv) τ-step update laws are implemented, which is effective for this class of MIMO systems.

T4: The main contributions are: (i) an effective control scheme is proposed for a class of MIMO discrete-time systems with complex subsystem interconnections; (ii) in the presence of unknown bounded disturbances, SGUUB stability is guaranteed; (iii) different from previous one step parameter update law, τ-step update laws are essential to solve the problem of τ-step ahead predictor; and (iv) by finding an orthogonal matrix, $Q(k)$, to tune the NN weights, the technical difficulty in the prove procedure is elegantly solved.

T5: The main contributions of are: (i) an effective NN control scheme is developed for a class of non-affine nonlinear discrete-time MIMO systems with triangular form inputs; and (ii) the proposed method is very simple for practical implementation.

1.4 Organization of the Thesis

In Chapter 2, adaptive NN control is presented for a class of discrete-time SISO non-affine nonlinear systems. Then adaptive NN control scheme is investigated for MIMO discrete-time nonlinear systems in state space representation in Chapter 3. State feedback and output feedback control schemes are proposed for two kinds of MIMO systems respectively. In Chapter 4, MIMO NARMAX discrete-time nonlinear systems are studied. Firstly, direct adaptive neural network control is studied for a class of NARMAX MIMO affine nonlinear systems based on input-output discrete-time model with unknown interconnections between subsystems and disturbances. Then, inspired by the results obtained, a simple control scheme is proposed for a class of non-affine MIMO discrete-time nonlinear systems. Finally, conclusions and suggestions for further research are made in Chapter 5.

Chapter 2

NN Control of Non-affine SISO Systems

2.1 Introduction

For SISO nonlinear discrete-time systems, there has been many discussions. In [20], a specific class of nonlinear affine systems is investigated. The plant under study is an unknown feedback-linearizable discrete-time system, represented by an input-output model. Single layered neural networks are used to model the unknown system and generate the feedback control. Based on the output error between plant and model, the neural network weights are updated, and local convergence result is given. However, the developed method will lose its effect for non-affine nonlinear systems. In [97], direct control of a general nonlinear dynamical system with only weak assumptions about the order and relative degree of the plant is discussed based on implicit function theory. The neural network control method is firstly discussed for first order discrete-time nonlinear system, and then the control scheme is generalized to high order discrete-time nonlinear system without rigorous proof. In [95], the authors provided the theoretical foundation as well as insights that are essential for the efficient design of neural network controllers based on inverse control. Discrete NARMAX non-affine systems based on input-output models are discussed.

In this chapter, based on implicit function theorem, RBF neural networks and MNN neural networks are used respectively as the emulator to construct direct neural network controllers for a class of discrete-time non-affine nonlinear systems. The stability analysis method and the weight update laws are different from the literatures listed above. Because of the unbounded residual term of multi-layer neural network approximation, projection algorithms are used in this chapter to guarantee the MNN weights bounded in compact sets. The main idea of the projection algorithms [114, 116, 117] is that, firstly we assume the fictitious lower and upper bound for the unknown weight vector or matrix, then the projection mapping is that, when weight estimates are within the bound, we use the normal adaptive law, once weight estimates reach the fictitious bounds and tend to go out of the bound, they are projected into the prescribed bounds by the projection mapping. Then, all the MNN approximate weights are bounded and their error are bounded too.

This chapter is organized as follows. The NARMAX system dynamics is described in Section 2.2. The projection algorithms are proposed in Section 2.3. The direct neural network inverse adaptive control and stability analysis are discussed in Section 2.4 for RBF and MNN respectively. Simulation results are provided in Section 2.5 to show the effectiveness of the controllers and the adaptive laws for both RBF control and MNN control. Finally, the possible application of the proposed MNN control scheme in practical CSTR systems is investigated in Section 2.6.

2.2 Problem Formulation

In discrete-time systems, one of the most popular nonlinear representation is NARMAX model studied by Billings and Voon in [103]. Many systems can be represented by a NARMAX model known as τ-step ahead observer equation as follows [95]

$$\begin{aligned} y(k+\tau) &= f(y(k),\dots,y(k-n+1),u(k),\dots,u(k-n+1),d(k+\tau-1),\dots,d(k)) \\ &= f(\bar{y}_k,u(k),\bar{u}_{k-1},\bar{d}_{k+\tau-1}) \end{aligned} \tag{2.1}$$

where $\bar{y}_k=[y(k),\dots,y(k-n+1)]^T$, $\bar{u}_{k-1}=[u(k-1),\dots,u(k-n+1)]^T$ and $\bar{d}_{k+\tau-1}=[d(k+\tau-1),\dots,d(k)]^T$. This model relates an input sequence $\{u(k)\}$ to an output sequence $\{y(k)\}$ by nonlinear difference equation. Specifically, it is the relationship between the sequences $\{u(k)\}$ and $\{y(k)\}$ that is of primary importance, while the sequence $\{d(k)\}$ represents a "modelling error" in this relationship, arising from the combined effects of unmeasured process disturbances, neglected nonlinearities, etc. This model constitutes an extremely broad class, including many other classes of nonlinear discrete-time models as special cases.

Considering system (2.1), it is shown that for the future output of time instant $y(k+\tau)$, it is determined by the sequence of $y(k), \ldots, y(k-n+1)$ and $u(k), \ldots, u(k-n+1)$ and disturbance sequence $d(k+\tau-1), \ldots, d(k)$.

Assumption 2.1 *The unknown nonlinear function $f(\cdot)$ is continuous and differentiable.*

Assumption 2.2 *System output $y(k)$ can be measured and its initial values are assumed to remain in a compact set Ω_{y_0}.*

Assumption 2.3 *The disturbance $d(k)$ is bounded, $|d(k)| \leq d$, where d is a little unknown constant and the partial derivative $|\frac{\partial f}{\partial d(k)}| \leq g_2$, where g_2 is a positive constant.*

Assumption 2.4 *Assume that partial derivative $g_1 \geq |\frac{\partial f}{\partial u}| > \epsilon > 0$, where both ϵ and g_1 are positive constants.*

This assumption states that the partial derivative $\frac{\partial f}{\partial u}$ is either positive or negative. From now onwards, without loss of generality, we assume that $\frac{\partial f}{\partial u} > 0$.

Remark 2.1 *According to Assumption 2.4, the partial derivative $\frac{\partial f}{\partial u}$ can be viewed as the control gain of the normal system (2.1). Furthermore, $g_1 \geq |\frac{\partial f}{\partial u}| > \epsilon > 0$ means that the plant gain is bounded by a positive constant, which does not pose a strong restriction upon the class of systems. In the following design procedure, we only need the existence of Assumption 2.4. Positive constants g_1 and ϵ are not required to be a priori known.*

Assume that $y_m(k+\tau)$ is the system's desired output at time instant $k+\tau$. Under Assumption 2.4, adding and subtracting $y_m(k+\tau)$ to the right side of equation (2.1) and using Mean Value Theorem, we have

$$\begin{aligned} y(k+\tau) &= y_m(k+\tau) + f(\bar{y}_k, u(k), \bar{u}_{k-1}, \bar{d}_{k+\tau-1}) - y_m(k+\tau) \\ &= y_m(k+\tau) + f(\bar{y}_k, u(k), \bar{u}_{k-1}, 0) + \delta_f^T \bar{d}_{k+\tau-1} - y_m(k+\tau) \\ &= y_m(k+\tau) + f(\bar{y}_k, u(k), \bar{u}_{k-1}, 0) - y_m(k+\tau) + \delta_{d_k} \end{aligned} \tag{2.2}$$

where

$$\begin{aligned} \delta_f &= \Big[\frac{\partial f}{\partial d(k+\tau-1)}|_{d(k+\tau-1)=d_{\xi_{k+\tau-1}}}, \cdots, \frac{\partial f}{\partial d(k)}|_{d(k)=d_{\xi_k}}\Big]^T \\ d_\xi &= [d_{\xi_{k+\tau-1}}, \cdots, d_{\xi_k}]^T \\ \delta_{d_k} &= \delta_f^T \bar{d}_{k+\tau-1} \end{aligned}$$

and $d_\xi \in L(0, \bar{d}_{k+\tau-1})$ with $L(0, \bar{d}_{k+\tau-1})$ indicating a spatial line in τ dimension, which starts from $\mathbf{0} \in R^\tau$ and ends at $\bar{d}_{k+\tau-1}$.

Remark 2.2 *Noticing the disturbance items in equation (2.1), at time instant k, the sequence $d(k+\tau-1), \ldots, d(k+1)$ are the future unknown disturbances which cannot be controlled even if they are known. In the following sections, we can see by the developed direct NN control, the system tracking error can be kept in a bounded compact set even in the presence of these unknown future and current disturbances.*

Consider Assumption 2.3, we know that the disturbance item δ_{d_k} in equation (2.2) is bounded by

$$\begin{aligned} \delta_{d_k} &= \delta_f^T \bar{d}_{k+\tau-1} \\ &= \frac{\partial f}{\partial d(k+\tau-1)}|_{d(k+\tau-1)=d_{\xi_{k+\tau-1}}} d(k+\tau-1) + \cdots + \frac{\partial f}{\partial d(k)}|_{d(k)=d_{\xi_k}} d(k) \\ &\leq g_2 d + g_2 d + \ldots + g_2 d \\ &= \tau g_2 d \end{aligned} \tag{2.3}$$

Define the tracking error as $e(k) = y(k) - y_m(k)$, then the tracking error dynamic equation is given by

$$e(k+\tau) = -y_m(k+\tau) + f(\bar{y}_k, u(k), \bar{u}_{k-1}, 0) + \delta_{d_k} \tag{2.4}$$

In the ideal case, there is no disturbance ($\delta_{d_k} = 0$), we can show that if the control input $u^*(k)$ satisfying

$$f(\bar{y}_k, u^*(k), \bar{u}_{k-1}, 0) - y_m(k+\tau) = 0 \tag{2.5}$$

then the system's output tracking error will converge to 0.

Definition 2.1 *If there exists a controller $u^*(k)$ satisfy equation (2.5), then the controller will drive the system output to the desired output, control input $u^*(k)$ is called Implicit Desired Feedback Control (IDFC).*

It is obvious that if the input $u(k)$ equals the IDFC, then the error $e(k+\tau)$ will converge to a small value which is a function of disturbance. Furthermore, if there is no disturbance, the tracking error will be zero. Based on implicit function theorem, we have the following lemma to establish the existence of an implicit desired feedback control $u^*(k)$, which can bring the output of the system to the desired trajectory.

Lemma 2.1 *According to Assumption 2.1 and 2.4 if partial derivative $|\frac{\partial f}{\partial u(k)}| > \epsilon > 0$, then there exists a unique and continuous function $u^*(k) = \alpha^c(\bar{y}_k, \bar{u}_{k-1}, y_m(k+\tau))$, such that equation (2.5) holds [50].*

Because the IDFC input $u^*(k)$ is a continuous function on the compact set Ω_z, according to the neural network theory, there exists an integer l (the number of hidden neurons) and ideal constant weight matrices W^* and V^*, such that

$$u^*(k) = u^*(z) = W^{*T} S(V^{*T}\bar{z}) + \varepsilon_u(z), \ \forall z \in \Omega_z \tag{2.6}$$

where $\bar{z} = [z, 1]^T$. The following assumption is made for this function approximation.

Assumption 2.5 *On the compact set Ω_z, the ideal neural network weights W^*, V^* and the NN approximation error are bounded by*

$$\|W^*\| \le w_m, \qquad \|V^*\|_F \le v_m, \qquad |\varepsilon_u(z)| \le \varepsilon_l \tag{2.7}$$

with w_m, v_m and ε_l being positive constants.

For the MNN we used, sigmoid function $s(x) = \frac{1}{1+e^{-x}}$ are chosen as the activation function. The derivative of the sigmoid activation function $s(x) = \frac{1}{1+e^{-x}}$ with respect to x is

$$s'(x) = \frac{d[s(x)]}{dx} = \frac{e^{-x}}{(1+e^{-x})^2}$$

It is easy to check that

$$0 \leq s'(x) \leq 0.25 \quad \text{and} \quad |xs'(x)| \leq 0.2239 \quad \text{for all} \quad x \in R \tag{2.8}$$

Hence

$$\|\hat{S}'\|_F \leq \sum_{i=1}^{l} s'(\hat{v}_i^T \bar{z}) \leq 0.25l \quad \|\hat{S}'\hat{V}^T \bar{z}\| \leq \sum_{i=1}^{l} |\hat{v}_i^T \bar{z} s'(\hat{v}_i^T \bar{z})| \leq 0.2239l \tag{2.9}$$

here $\hat{S}' = diag\{s'(\hat{v}_1^T \bar{z}), \ldots, s'(\hat{v}_i^T \bar{z}), \ldots, s'(\hat{v}_l^T \bar{z})\}$ is a diagonal matrix and the Frobenius Norm $\|\cdot\|_F$ is defined as follows

$$\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}$$

with A is a matrix and a_{ij} is its element.

Using Taylor series expansion $S(V^{*T}\bar{z})$ about $\hat{V}^T \bar{z}$, noting abbreviation $\hat{S} = S(\hat{V}^T \bar{z})$ and $\tilde{V} = \hat{V} - V^*$, we have

$$S(V^{*T}\bar{z}) = \hat{S} - \hat{S}'\tilde{V}^T \bar{z} + O(\tilde{V}^T \bar{z})^2 \tag{2.10}$$

Using inequalities (2.9), we know that the high order term $O(\tilde{V}^T \bar{z})^2$ is bounded by

$$\begin{aligned} \|O(\tilde{V}^T \bar{z})^2\| &\leq \|\hat{S}'\tilde{V}^T \bar{z}\| + \|S(V^{*T}\bar{z}) - S(\hat{V}^T \bar{z})\| \\ &\leq \|\hat{S}'\hat{V}^T \bar{z}\| + \|\hat{S}'V^{*T}\bar{z}\| + \|S(V^{*T}\bar{z}) - S(\hat{V}^T \bar{z})\| \\ &\leq \|\hat{S}'\hat{V}^T \bar{z}\| + \|\hat{S}'\|_F \cdot \|V^*\|_F \cdot \|\bar{z}\| + \|S(V^{*T}\bar{z}) - S(\hat{V}^T \bar{z})\| \end{aligned}$$

Considering (2.9), $\|V^*\|_F \leq v_m$ and the fact that $\|S(V^{*T}\bar{z}) - S(\hat{V}^T \bar{z})\| \leq l$, we have

$$\|O(\tilde{V}^T \bar{z})^2\| \leq 1.2239l + 0.25v_m l\|\bar{z}\| \tag{2.11}$$

2.3 Projection Algorithm

In order to avoid the possible divergence of the online tuning of neural networks, discontinuous projections with fictitious bounds are used in the MNN weight adjusting law to make sure that all MNN weights are tuned within a prescribed range. By doing so, even in the presence of approximation error and non-repeatable nonlinearities such as disturbances, a controlled learning is achieved and the possible destabilizing effect of online tuning of MNN weights could be avoided.

Although the weights of the ideal MNN approximating unknown nonlinearities are unknown, they are constants and bounded by Assumption 2.5. Thus it is assumed that each element of W^* and V^* is bounded, i.e., $\rho_{w_i,\min} \leq w_i \leq \rho_{w_i,\max}$ for $i = 1, \ldots, l$ and $\rho_{v_{ij},\min} \leq v_{ij} \leq \rho_{v_{ij},\max}$ for $i = 1, \ldots, n, j = 1, \ldots, l$, where the lower and upper bounds $\rho_{w,\min}$, $\rho_{w,\max}$, $\rho_{v,\min}$, $\rho_{v,\max}$ maybe unknown. The number n stands for the input dimension of neural networks and the number l stands for the numbers of neurons used. It is natural to require that the estimates of the weights should be within the corresponding bounds. However, due to the fact that these bounds may not be known a prior, certain fictitious bounds have to be used [118].

In this chapter, we use the following projection mapping [118]. Let $\hat{\rho}_{\Theta_{ij},\min}$ and $\hat{\rho}_{\Theta_{ij},\max}$ be the fictitious lower and upper bound for Θ_{ij}, where Θ could be any of the unknown weight vector or matrix. Based on these fictitious lower and upper bounds, same as in [116] and [117] a discontinuous projection mapping $\mathrm{Proj}(*)$ can be defined as $\mathrm{Proj}_{\hat{\Theta}}(*) = \{\mathrm{Proj}_{\hat{\Theta}}(*_{ij})\}$ with its ijth element being

$$\mathrm{Proj}_{\hat{\Theta}}(*_{ij}) = \begin{cases} -*_{ij} & \text{if} \begin{cases} \hat{\Theta}_{ij} = \hat{\rho}_{\Theta_{ij},\max} & \text{and} \quad *_{ij} < 0 \\ \hat{\Theta}_{ij} = \hat{\rho}_{\Theta_{ij},\min} & \text{and} \quad *_{ij} > 0 \end{cases} \\ *_{ij} & \qquad\qquad \text{otherwise} \end{cases} \tag{2.12}$$

where $*$ denotes a vector or a matrix, then $*_{ij}$ denotes its element.

In this chapter, all parameter estimates will be updated by the projection type of adaptation laws given by

$$\hat{\Theta}(k + \tau) = \hat{\Theta}(k) - \mathrm{Proj}_{\hat{\Theta}}(\Gamma\eta) \tag{2.13}$$

where $\Gamma = \Gamma^T > 0$ is any diagonal positive-definite adaptation matrix with proper dimension, and η is any adaptation function. For simplicity, assume $\Gamma = \lambda I$ with λ being a positive constant. Similar to [114], we have the following lemma which indicates the nice properties of the above projection type of adaptation law.

Lemma 2.2 *Considering the projection algorithm (2.12) and parameter adaptation laws (2.13) used in this chapter, the following properties hold:*

1. *The parameter estimates are always within the known prescribed range, i.e.,* $\hat{\rho}_{\Theta_{ij},\min} \le \hat{\Theta}_{ij} \le \hat{\rho}_{\Theta_{ij},\max}$.
2. *In addition, if the true parameter* Θ *is actually within the prescribed range, noting* $\tilde{\Theta} = \hat{\Theta} - \Theta$, *then*

$$\begin{aligned}
\tilde{\Theta}^T(\Gamma^{-1} Proj_{\hat{\Theta}}(\Gamma\eta) - \eta) &\ge 0 \quad \text{if } \Theta \text{ is a vector.} \\
tr\{\tilde{\Theta}^T(\Gamma^{-1} Proj_{\hat{\Theta}}(\Gamma\eta) - \eta)\} &\ge 0 \quad \text{if } \Theta \text{ is a matrix.}
\end{aligned}$$

Proof. According to the projection algorithm (2.12) and adaptation law (2.13), it is obvious that the first property always holds. Now we prove the second property.

If Θ is a vector, consider the diagonal positive-definite adaptation matrix Γ, noticing that the possible effect of projection operator $\mathrm{Proj}_{\hat{\Theta}}(*_{ij})$ is to change the sign of $*_{ij}$, we have

$$\begin{aligned}
\tilde{\Theta}^T(\Gamma^{-1}\mathrm{Proj}_{\hat{\Theta}}(\Gamma\eta) - \eta) &= \tilde{\Theta}^T(\Gamma^{-1}\Gamma\mathrm{Proj}_{\hat{\Theta}}(\eta) - \eta) \\
&= \tilde{\Theta}^T(\mathrm{Proj}_{\hat{\Theta}}(\eta) - \eta) \\
&= \sum_{i=1}^{l} \tilde{\Theta}_i(\mathrm{Proj}_{\hat{\Theta}}(\eta_i) - \eta_i)
\end{aligned}$$

Then considering

$$\begin{aligned}
&\tilde{\Theta}_i(\mathrm{Proj}_{\hat{\Theta}}(\eta_i) - \eta_i) \\
&\qquad = (\hat{\Theta}_i - \Theta_i)(\mathrm{Proj}_{\hat{\Theta}}(\eta_i) - \eta_i)
\end{aligned}$$

$$
= \begin{cases} (\hat{\Theta}_i - \Theta_i)(-\eta_i - \eta_i) > 0 \quad \text{if} \begin{cases} \hat{\Theta}_i = \hat{\rho}_{\Theta_i,\max} \ \text{ means } \ (\hat{\Theta}_i - \Theta_i) > 0 \\ \qquad\qquad \text{and} \ \ \eta_i < 0 \\ \hat{\Theta}_i = \hat{\rho}_{\Theta_i,\min} \ \text{ means } \ (\hat{\Theta}_i - \Theta_i) < 0 \\ \qquad\qquad \text{and} \ \ \eta_i > 0 \end{cases} \\ \\ (\hat{\Theta}_i - \Theta_i)(\eta_i - \eta_i) = 0 \qquad\qquad \text{otherwise} \end{cases}
$$

we have $\tilde{\Theta}^T(\Gamma^{-1}\mathrm{Proj}_{\hat{\Theta}}(\Gamma\eta) - \eta) \geq 0$ holds.

If Θ is a matrix, following the same procedure, we have

$$
\mathrm{tr}\{\tilde{\Theta}^T(\Gamma^{-1}\mathrm{Proj}_{\hat{\Theta}}(\Gamma\eta) - \eta)\} \quad \geq \quad 0
$$

Its proof is omitted here for clarity. ■

2.4 Controller Design

2.4.1 RBF NN Control

Because the defined IDFC controller $u^*(k)$ is a continuous function in the compact set Ω_u, then according to the neural network theory, there exists an integer l (the number of hidden neurons) and ideal constant weight vector W^*, such that

$$
u^*(k) = u^*(z) = W^{*T}S(z) + \varepsilon_u(z), \ \forall z \in \Omega_z \tag{2.14}
$$

where $z = [\bar{y}_k, \bar{u}_{k-1}, y_m(k+\tau)]^T$, $\bar{y}_k$, $\bar{u}_{k-1}$ and $y_m(k+\tau)$ are defined in section 2.2.

Assumption 2.6 *On the compact set Ω_z, the ideal neural network weights W^* and the NN approximation error are bounded by*

$$
\|W^*\| \leq w_m, \qquad |\varepsilon_u(z)| \leq \varepsilon_l \tag{2.15}
$$

with w_m and ε_l being positive constants.

Define $\hat{W}(k)$ as the actual neural network weight, then the practical control input is

$$
u(k) \quad = \quad \hat{W}^T(k)S(z) \tag{2.16}
$$

then noticing equation (2.14) the controller approximation error is

$$\begin{aligned} u(k) - u^*(k) &= \hat{W}^T(k)S(z) - [W^{*T}S(z) + \varepsilon_u(z)] \\ &= \tilde{W}^T(k)S(z) - \varepsilon_u(z) \end{aligned} \tag{2.17}$$

where $\tilde{W}(k) = \hat{W}(k) - W^*$ is the weight approximation error.

If we choose the weight update law as [119]

$$\hat{W}(k+\tau) = \hat{W}(k) - \Gamma[S(z(k))e(k+\tau) + \sigma\hat{W}(k)] \tag{2.18}$$

where $\Gamma = \Gamma^T > 0$ is a diagonal adaptation gain matrix, and $\sigma > 0$. This is the modified gradient algorithm and the last term of the right-hand side of equation (2.18) corresponds to σ-modification [62] introduced to improve the robustness in the presence of the RBF NN approximation error.

Noticing that $\tilde{W}(k) = \hat{W}(k) - W^*$, subtracting W^* to both sides of equation (2.18), then we have

$$\tilde{W}(k+\tau) = \tilde{W}(k) - \Gamma[S(z(k))e(k+\tau) + \sigma\tilde{W}(k) + \sigma W^*] \tag{2.19}$$

Substituting $u(k)$ in to the error equation (2.4) and noticing equation (2.17), then we have

$$\begin{aligned} e(k+\tau) &= -y_m(k+\tau) + f(\bar{y}_k, \hat{W}^T(k)S(z(k)), \bar{u}_{k-1}, 0) + \delta_{d_k} \\ &= -y_m(k+\tau) + f(\bar{y}_k, \hat{W}^T(k)S(z(k)), \bar{u}_{k-1}, 0) + \delta_{d_k} \\ &= -y_m(k+\tau) + f(\bar{y}_k, u^*(k) + \tilde{W}^T(k)S(z) - \varepsilon_u(z), \bar{u}_{k-1}, 0) + \delta_{d_k} \end{aligned} \tag{2.20}$$

Using the Mean Value Theorem, noting equation (2.5) and (2.17), then above equation becomes

$$\begin{aligned} e(k+\tau) &= -y_m(k+\tau) + f(\bar{y}_k, u^*(k), \bar{u}_{k-1}, 0) + \frac{\partial f}{\partial u}|_{u=\xi}[\tilde{W}^T(k)S(z) - \varepsilon_u(z)] + \delta_{d_k} \\ &= \frac{\partial f}{\partial u}|_{u=\xi}[\tilde{W}^T(k)S(z) - \varepsilon_u(z)] + \delta_{d_k} \\ &= [\tilde{W}^T(k)S(z) - \varepsilon_u(z)]f_u + \delta_{d_k} \end{aligned} \tag{2.21}$$

where

$$f_u = \frac{\partial f}{\partial u}|_{u=\xi}, \quad \xi \in [u^*(k), u(k)]$$

Remark 2.3 *We have assumed that f_u is bounded over the compact set Ω_u, then it is obvious that by increasing the neurons used, the neural approximation error term $\varepsilon_u(z)$ can be arbitrarily small. For the error item $\tilde{W}(k)S(z)$, if $\hat{W}(k)$ can get very close to W^*, noticing that every element of $S(z)$ is less than 1, we can derive that $\tilde{W}^T(k)S(z)$ can be made very small if the neural network approximation accuracy is sufficiently high. Therefore the error $e(k+\tau)$ will be bounded, the bound will depends on the neural approximation accuracy and the disturbance.*

Remark 2.4 *If the disturbance sequence $\{d(k)\}$ equal to 0, then tracking error will mainly depends on the neural network approximation accuracy. However, if there exists a small disturbance sequence $\{d(k)\}$, then the tracking error will depends on both the neural network approximate accuracy and the disturbance. We can see that the effects of disturbance can be eliminated by the developed direct NN control scheme as shown below.*

The stability results of the RBF neural networks controller are summarized by Theorem 2.1.

Theorem 2.1 *For the non-affine discrete-time system (2.1), neural network controller (2.16) and neural network weight update law (2.18). There exist compact sets Ω_y, Ω_w and positive constants l^*, σ^* and λ^* such that if*

(i) the initial parameter set $\Omega_{y_0} \in \Omega_y$, $\Omega_{w_0} \in \Omega_w$;

(ii) the neurons number $l > l^*$, σ-modification gain $\sigma < \sigma^*$ and adaptive gain $\lambda < \lambda^*$, with λ^* being the largest eigenvalue of Γ;

(iii) the initial future output sequence $y(k_0), \ldots, y(k_0+\tau-1)$ are kept in the compact set Ω_y;

then the output of system (2.1) will track the desired trajectory and the tracking error can be made arbitrary small by increasing the approximation accuracy of the neural network. The closed-loop system is semi-globally uniformly ultimately bounded (SGUUB).

Proof. Choose the Lyapunov function as follows

$$J(k) = \frac{1}{g_1}\sum_{j=0}^{\tau-1} e_y^2(k+j) + \sum_{j=0}^{\tau-1} \tilde{W}^T(k+j)\Gamma^{-1}\tilde{W}(k+j) \tag{2.22}$$

Considering (2.19), the first difference of (2.22) is given

$$\begin{aligned}
\triangle J(k) &= J(k+1) - J(k) \\
&= \frac{1}{g_1}[e^2(k+\tau) - e^2(k)] + \tilde{W}^T(k+\tau)\Gamma^{-1}\tilde{W}(k+\tau) - \tilde{W}^T(k)\Gamma^{-1}\tilde{W}(k) \\
&= \frac{1}{g_1}[e^2(k+\tau) - e^2(k)] - 2\tilde{W}^T(k)[S(z(k))e(k+\tau) + \sigma\hat{W}(k)] \\
&\quad +[S(z(k))e(k+\tau) + \sigma\hat{W}(k)]^T\Gamma^T[S(z(k))e(k+\tau) + \sigma\hat{W}(k)] \\
&= \frac{1}{g_1}[e^2(k+\tau) - e^2(k)] - 2\tilde{W}^T(k)S(z(k))e(k+\tau) - 2\sigma\tilde{W}^T(k)\hat{W}(k) \\
&\quad +S^T(z(k))\Gamma^T S(z(k))e^2(k+\tau) + 2\sigma\hat{W}^T(k)\Gamma^T S(z(k))e(k+\tau) \\
&\quad +\sigma^2\hat{W}^T(k)\Gamma^T\hat{W}(k)
\end{aligned}$$

Noticing equation (2.21), we have

$$\tilde{W}^T(k)S(z(k)) = \frac{e(k+\tau) - \delta_{d_k}}{f_u} + \varepsilon_u(z) \tag{2.23}$$

Furthermore, using the fact that

$$\begin{aligned}
2\sigma\tilde{W}^T(k)\hat{W}(k) &= \sigma(\| \tilde{W}(k) \|^2 + \| \hat{W}(k) \|^2 - \| W^* \|^2) \\
S^T(z(k))\Gamma^T S(z(k))e^2(k+\tau) &\le \lambda^* l e^2(k+\tau) \\
2\sigma\hat{W}(k)\Gamma^T S(z(k))e(k+\tau) &\le 2\sigma \| \hat{W}(k) \| \| \Gamma \|_F \| S(z(k)) \| \, |e(k+\tau)| \\
&\le 2\sigma \| \hat{W}(k) \| \sqrt{l}\lambda^*\sqrt{l}|e(k+\tau)| \\
&\le 2\sigma l\lambda^* \| \hat{W}(k) \| \, |e(k+\tau)| \\
&\le \sigma l\lambda^*[\| \hat{W}(k) \|^2 + e^2(k+\tau)] \\
\sigma^2\hat{W}^T(k)\Gamma^T\hat{W}(k) &\le \sigma^2\lambda^* \| \hat{W}(k) \|^2
\end{aligned}$$

where λ^* stands for the maximum eigenvalue of the matrix Γ, we obtain

$$\begin{aligned}
\triangle J(k) \le\ & \frac{1}{g_1}e^2(k+\tau) - \frac{1}{g_1}e^2(k) - 2\frac{e^2(k+\tau)}{f_u} - 2[\varepsilon_u(z) - \frac{\delta_{d_k}}{f_u}]e(k+\tau) \\
& -\sigma \| \tilde{W}(k) \|^2 - \sigma \| \hat{W}(k) \|^2 + \sigma \| W^* \|^2 + \lambda^* l e^2(k+\tau) \\
& +\sigma l\lambda^*[\| \hat{W}(k) \|^2 + e^2(k+\tau)] + \sigma^2\lambda^* \| \hat{W}(k) \|^2
\end{aligned}$$

that is

$$\begin{aligned}\triangle J(k) \;\le\; & [\frac{1}{g_1} - \frac{2}{f_u} + (1+\sigma)l\lambda^*]e^2(k+\tau) - 2[\varepsilon_u(z) - \frac{\delta_{d_k}}{f_u}]e(k+\tau) \\ & +\sigma(l\lambda^* + \sigma\lambda^* - 1) \parallel \hat{W}(k) \parallel^2 \\ & -\frac{1}{g_1}e^2(k) - \sigma \parallel \tilde{W}(k) \parallel^2 + \sigma \parallel W^* \parallel^2\end{aligned}$$

Noticing Assumption 2.4, from $0 < \epsilon < f_u < g_1$, we can derive that $-\frac{2}{f_u} < -\frac{2}{g_1}$. By further noticing equation (2.3), we obtain

$$\begin{aligned}-2[\varepsilon_u(z) - \frac{\delta_{d_k}}{f_u}]e(k+\tau) \;\le\; & \frac{1}{k_1}e^2(k+\tau) + k_1(\varepsilon_u(z) - \frac{\delta_{d_k}}{f_u})^2 \\ \le\; & \frac{1}{k_1}e^2(k+\tau) + k_1(\varepsilon_l + \frac{\tau g_2 d}{\epsilon})^2\end{aligned}$$

where k_1 is a positive number. Thus, we have

$$\begin{aligned}\triangle J(k) \;\le\; & [-\frac{1}{g_1} + (1+\sigma)l\lambda^* + \frac{1}{k_1}]e^2(k+\tau) + \sigma(l\lambda^* + \sigma\lambda^* - 1) \parallel \hat{W}(k) \parallel^2 \\ & -\frac{1}{g_1}e^2(k) - \sigma \parallel \tilde{W}(k) \parallel^2 + \sigma w_m^2 + k_1(\varepsilon_l + \frac{\tau g_2 d}{\epsilon})^2 \\ \le\; & [-\frac{1}{g_1} + (1+\sigma)l\lambda^* + \frac{1}{k_1}]e^2(k+\tau) + \sigma(l\lambda^* + \sigma\lambda^* - 1) \parallel \hat{W}(k) \parallel^2 \\ & -\frac{1}{g_1}[e^2(k) - \beta] - \sigma \parallel \tilde{W}(k) \parallel^2\end{aligned}$$

where positive number

$$\beta = g_1[\sigma w_m^2 + k_1(\varepsilon_l + \frac{\tau g_2 d}{\epsilon})^2] \tag{2.24}$$

By choosing the positive constants k_1, λ and σ satisfying the following inequalities

$$k_1 > g_1 \tag{2.25}$$

$$(1+\sigma)l\lambda \le \frac{1}{g_1} - \frac{1}{k_1} \tag{2.26}$$

$$(l+\sigma)\lambda \le 1 \tag{2.27}$$

we have $\triangle J(k) \le 0$ once $e^2(k) \ge \beta$. This states that for all $k \ge 0$, $J(k)$ is bounded because

$$J(k) = J(0) + \sum_{j=0}^{k} \triangle J(i) < \infty$$

Define compact set

$$\Omega_e \triangleq \{e \mid e^2 \leq \beta\}$$

then we can see that the tracking error $e(k)$ will converge to Ω_e if $e(k)$ is out of compact Ω_e. Therefore, for any a priori given (arbitrarily large) bounded set Ω and any a priori given (arbitrarily small) set Ω_0, which contains $(0,0)$ as an interior point, there exist a control u, such that every trajectory of the closed-loop system starting from Ω enters the set Ω_0 in a finite time and remains in it thereafter. That is to say, the whole closed-loop system is SGUUB. ■

Remark 2.5 *It is shown that a smaller β might be obtained by choosing a smaller σ or decreasing neural network approximation error ε_l which may lead to smaller tracking error. In general, smaller ε_l will need larger number of neurons which will lead to the need of more computational power. Positive constant k_1 is a intermediate positive variable. It is not a tuning parameter, but the tuning parameters must satisfy the inequalities (2.25) and (2.26) which contain k_1.*

Remark 2.6 *Consider the special character of discrete-time system, for RBF neural network, a new simulation receptive center selection method is used, which will greatly decrease the number of neurons, that is to say, to avoid the so-called the "curse of dimensionality" [54, 120] to some extent. The number of Radial Basis Function for RBF networks needed to approximate a given function is a critical factor in solving identification and control problem. Because such a number tends to increase exponentially with the dimension of the input space, the approximation approach becomes practically infeasible when the dimensionality of the input space is high. It is obvious that for the class of discrete-time system we are discussing, for a $n = 3$ order system, the input dimension of the neural networks controller will be 6, if we choose 4 receptive center points for every input, then there is a need of up to $4^6 = 4096$ neurons, which is a very large number. Considering the special character of discrete systems, we can use following method to reduce the number of neurons. For the discrete system, the state variables sequence $\{x(k)\}$, input sequence $\{u(k)\}$ and output sequence $\{y(k)\}$ are almost the same in one or two steps, then for every sequences above, we can use* *the same neurons. That is to say the input dimension of the neural networks controller will reduce to 4. Then the number of neurons will be $4^4 = 256$, which greatly improves the simulation performance, as can be seen in Section 2.5.*

Remark 2.7 *The neural networks update law consist of a modified gradient algorithm with standard σ-modification term [62]. These laws have been proven to be passive in [57]. No off-line training is required. No assumption on persistent excitation is required.*

Remark 2.8 *It can be seen from inequality (2.27), that the upper bound of the adaptation gain should decreases with an increase of the number of hidden-layer nodes, so that learning must slow down for guaranteed performance. The phenomenon of large NN requiring very slow learning rates has often been encountered in the practical NN literature [121, 14]. This major drawback can easily be overcome by modifying the update rule at each layer to obtain a projection algorithm [58]. By employing a projection algorithm, it is shown that the tuning rate can be made independent of the NN size. Modified tuning paradigms are finally proposed to make the NN robust so that the PE is not needed.*

2.4.2 MNN Control

For RBF neural networks, it is easy to use and the approximation error is bounded. But the selection of the receptive center is a big problem, you should have the preliminaries of the states varying range. Furthermore, when the states vary in a wide range, it is difficult to approximate the IDFC control with small number of neurons, however, too much neurons will tend to make the system unstable. Considering the universal approximate ability of MNN, in this section, we use MNN to approximate the IDFC control instead of RBF neural networks. The use of multi-layer neural networks in discrete-time nonlinear system control is not only challenging but also of academic interest.

At first, considering the multi-layer neural networks, neural weights adaptation laws and projection algorithms used in this chapter, we have the following lemma.

Assumption 2.7 *Considering the projection algorithms we used, on the compact set Ω_z, the estimates of neural network weights $\hat{W}$, $\hat{V}$ and the weight approximation error $\tilde{W}$, $\tilde{V}$ are bounded by*

$$\|\hat{W}\| \leq \hat{w}_m, \quad \|\hat{V}\|_F \leq \hat{v}_m, \quad \|\tilde{W}\| \leq \tilde{w}_m, \quad \|\tilde{V}\|_F \leq \tilde{v}_m \tag{2.28}$$

where $\tilde{W} = \hat{W} - W^*$, $\tilde{V} = \hat{V} - V^*$ and $\tilde{w}_m$, $\tilde{v}_m$, $\hat{w}_m$, $\hat{v}_m$ are positive constants.

In this chapter, we use the following adaptive function

$$\eta_w = \hat{S}(k)e(k+\tau) \tag{2.29}$$

$$\eta_v = (z_l \hat{W}^T(k)\hat{S}'(k))e(k+\tau) \tag{2.30}$$

where z_l is a constant vector which is dimension compatible with $\hat{V}(k)$. It is defined as $z_l = [\frac{1}{\sqrt{l}}, \ldots, \frac{1}{\sqrt{l}}]^T$ with $\|z\| = 1$. $\hat{S}'(k) = diag[\hat{s}'_1(k), \ldots, \hat{s}'_l(k)]$ is a diagonal matrix and $\hat{s}'_i(k) = s'(\hat{v}_i^T \bar{z}(k))$.

Define the multi-layer neural networks update law as follows

$$\hat{W}(k+\tau) = \hat{W}(k) - \mathrm{Proj}_{\hat{W}}[\Gamma_w \eta_w] \tag{2.31}$$

$$\hat{V}(k+\tau) = \hat{V}(k) - \mathrm{Proj}_{\hat{V}}[\Gamma_v \eta_v] \tag{2.32}$$

Subtract W^* and V^* to the both sides of the equation (2.31) and (2.32), we obtain

$$\begin{aligned} \tilde{W}(k+\tau) &= \tilde{W}(k) - \mathrm{Proj}_{\hat{W}}(\Gamma_w \eta_w) \\ &= \tilde{W}(k) - \mathrm{Proj}_{\hat{W}}[\Gamma_w \hat{S}(k)e(k+\tau)] \end{aligned} \tag{2.33}$$

$$\begin{aligned} \tilde{V}(k+\tau) &= \tilde{V}(k) - \mathrm{Proj}_{\hat{V}}(\Gamma_v \eta_v) \\ &= \tilde{V}(k) - \mathrm{Proj}_{\hat{V}}[\Gamma_v (z_l \hat{W}^T(k)\hat{S}'(k))e(k+\tau)] \end{aligned} \tag{2.34}$$

where $\Gamma_w = \Gamma_w^T = \lambda_w I$ and $\Gamma_v = \Gamma_v^T = \lambda_v I$.

Remark 2.9 *Noticing multi-layer neural network update laws (2.31)-(2.34), at time instant $k+\tau$, the MNN weight $\hat{W}(k+\tau)$ and $\hat{V}(k+\tau)$ are relevant to the tracking error $e(k+\tau)$, this seems to be non-causal. However, these parameters update at $k+\tau$ will be used only in $u(k+\tau)$, thus it is causal.*

Lemma 2.3 *Consider Lemma 2.2, we have the following inequalities*

$$\tilde{W}^T(\Gamma_w^{-1} Proj_{\hat{W}}(\Gamma_w \eta_w) - \eta_w) \geq 0 \quad (2.35)$$

$$tr\{\tilde{V}^T(\Gamma_v^{-1} Proj_{\hat{V}}(\Gamma_v \eta_v) - \eta_v)\} \geq 0 \quad (2.36)$$

Furthermore we have

$$Proj_{\hat{W}}^T \Gamma_w^{-1} Proj_{\hat{W}} - \eta_w^T \Gamma_w^T \eta_w = 0 \quad (2.37)$$

$$tr\{Proj_{\hat{V}}^T \Gamma_v^{-1} Proj_{\hat{V}}\} - tr\{\eta_v^T \Gamma_v^T \eta_v\} = 0 \quad (2.38)$$

where $\mathrm{Proj}_{\hat{W}} = \mathrm{Proj}_{\hat{W}}(\Gamma_w \eta_w)$ and $\mathrm{Proj}_{\hat{V}} = \mathrm{Proj}_{\hat{V}}(\Gamma_v \eta_v)$.

Proof. It is obvious that following Lemma 2.2, inequalities (2.35) and (2.36) hold.

Considering equation (2.37), because $\Gamma_w = \lambda_w I$, we have

$$\begin{aligned}
\mathrm{Proj}_{\hat{W}}^T \Gamma_w^{-1} \mathrm{Proj}_{\hat{W}} - \eta_w^T \Gamma_w^T \eta_w &= \mathrm{Proj}_{\hat{W}}^T(\eta_w) \Gamma_w^T \Gamma_w^{-1} \Gamma_w \mathrm{Proj}_{\hat{W}}(\eta_w) - \eta_w^T \Gamma_w^T \eta_w \\
&= \lambda_w \mathrm{Proj}_{\hat{W}}^T(\eta_w) \mathrm{Proj}_{\hat{W}}(\eta_w) - \lambda_w \eta_w^T \eta_w \\
&= \lambda_w [\mathrm{Proj}_{\hat{W}}^T(\eta_w) \mathrm{Proj}_{\hat{W}}(\eta_w) - \eta_w^T \eta_w] \\
&= 0
\end{aligned}$$

then equation (2.37) holds.

Considering equation (2.38), because $\Gamma_v = \lambda_v I$, we have

$$\begin{aligned}
&\mathrm{tr}\{\mathrm{Proj}_{\hat{V}}^T \Gamma_v^{-1} \mathrm{Proj}_{\hat{V}}\} - \mathrm{tr}\{\eta_v^T \Gamma_v^T \eta_v\} \\
&\quad = \mathrm{tr}\{\mathrm{Proj}_{\hat{V}}^T(\eta_v) \Gamma_v^T \Gamma_v^{-1} \Gamma_v \mathrm{Proj}_{\hat{V}}(\eta_v)\} - \mathrm{tr}\{\eta_v^T \Gamma_v^T \eta_v\} \\
&\quad = \lambda_v \mathrm{tr}\{\mathrm{Proj}_{\hat{V}}^T(\eta_v) \mathrm{Proj}_{\hat{V}}(\eta_v) - \eta_v^T \eta_v\} \\
&\quad = 0
\end{aligned}$$

then equation (2.38) holds. ■

Choose the practical control input as

$$u(k) = u_{nn}(k) \quad (2.39)$$

with

$$u_{nn}(k) = \hat{W}^T(k) S(\hat{V}^T(k) \bar{z}) \quad (2.40)$$

where $\bar{z} = [z^T, 1]^T$ with $z = [\bar{y}_k, \bar{u}_{k-1}, y_m(k+\tau)]^T$, $\bar{y}_k$, $\bar{u}_{k-1}$ and $y_m(k+\tau)$ are defined in Section 2.2.

Noticing equation (2.6), then we have

$$\begin{aligned} u(k) - u^*(k) &= \hat{W}^T(k)S(\hat{V}^T(k)\bar{z}) - W^{*T}(k)S(V^{*T}(k)\bar{z}) - \varepsilon_u(z) \\ &= \hat{W}^T(k)S(\hat{V}^T(k)\bar{z}) - W^{*T}(k)S(\hat{V}^T(k)\bar{z}) \\ &\quad + W^{*T}(k)S(\hat{V}^T(k)\bar{z}) - W^{*T}(k)S(V^{*T}(k)\bar{z}) - \varepsilon_u(z) \\ &= \tilde{W}^T(k)S(\hat{V}^T(k)\bar{z}) + W^{*T}(k)[S(\hat{V}^T(k)\bar{z}) - S(V^{*T}(k)\bar{z})] - \varepsilon_u(z) \\ &= \tilde{W}^T\hat{S} + W^{*T}(\hat{S} - S^*) - \varepsilon_u(z) \end{aligned}$$

Substitute $u(k)$ into the error equation (2.4), then we have

$$\begin{aligned} e(k+\tau) &= -y_m(k+\tau) + f(\bar{y}_k, \hat{W}^T(k)S(\hat{V}^T(k)\bar{z}), \bar{u}_{k-1}, 0) + \delta_{d_k} \\ &= -y_m(k+\tau) + f(\bar{y}_k, u^*(k) + \tilde{W}^T\hat{S} + W^{*T}(\hat{S} - S^*) - \varepsilon_u(z), \bar{u}_{k-1}, 0) \\ &\quad + \delta_{d_k} \end{aligned} \tag{2.41}$$

Using Mean Value Theorem, noticing equation (2.5), then the above equation becomes

$$\begin{aligned} e(k+\tau) &= -y_m(k+\tau) + f(\bar{y}_k, u^*(k), \bar{u}_{k-1}, 0) \\ &\quad + \frac{\partial f}{\partial u}|_{u=\xi}(\tilde{W}^T\hat{S} + W^{*T}(\hat{S} - S^*) - \varepsilon_u(z)) + \delta_{d_k} \\ &= [\tilde{W}^T\hat{S} + W^{*T}(\hat{S} - S^*) - \varepsilon_u(z)]f_u + \delta_{d_k} \end{aligned} \tag{2.42}$$

where

$$f_u = \frac{\partial f}{\partial u}|_{u=\xi} \quad \xi \in [u^*(k), u(k)]$$

Theorem 2.2 *For the non-affine discrete-time system (2.1), neural network controller (2.39) and neural network weight update laws (2.31) and (2.32). There exist compact sets Ω_y, Ω_w, Ω_v and positive constants l^*, λ_w^* and λ_v^* such that if*

(i) the initial parameter set $\Omega_{y_0} \in \Omega_y$, $\Omega_{w_0} \in \Omega_w$, $\Omega_{v_0} \in \Omega_v$;

(ii) the neural number $l > l^*$, adaptive gain $\lambda_w < \lambda_w^*$, with λ_w^* being the eigenvalue of Γ_w, $\lambda_v < \lambda_v^*$, with λ_v^* being the eigenvalue of Γ_v;

(iii) the initial future output sequence $y(k_0), \ldots, y(k_0+\tau-1)$ are kept in the compact set Ω_y, initial input sequence $u(k_0)$ are kept in the compact set Ω_u;

then the output of system (2.1) will track the desired trajectory and the tracking error is bounded. The closed-loop system is semi globally uniformly ultimately bounded (SGUUB).

Proof. Choose the Lyapunov function as follows

$$\begin{aligned} J(k) &= \frac{1}{g_1}\sum_{j=0}^{\tau-1} e^2(k+j) + \sum_{j=0}^{\tau-1} \tilde{W}^T(k+j)\Gamma_w^{-1}\tilde{W}(k+j) \\ &\quad + \sum_{j=0}^{\tau-1} tr\{\tilde{V}^T(k+\tau)\Gamma_v^{-1}\tilde{V}(k+\tau)\} \end{aligned} \tag{2.43}$$

The first difference of (2.43) is given

$$\begin{aligned} \triangle J(k) &= J(k+1) - J(k) \\ &= \frac{1}{g_1}[e^2(k+\tau) - e^2(k)] + \tilde{W}^T(k+\tau)\Gamma_w^{-1}\tilde{W}(k+\tau) - \tilde{W}^T(k)\Gamma_w^{-1}\tilde{W}(k) \\ &\quad + tr\{\tilde{V}^T(k+\tau)\Gamma_v^{-1}\tilde{V}(k+\tau) - \tilde{V}^T(k)\Gamma_v^{-1}\tilde{V}(k)\} \end{aligned}$$

Considering the neural network weight update laws (2.33) and (2.34), we have

$$\begin{aligned} \triangle J(k) &= \frac{1}{g_1}[e^2(k+\tau) - e^2(k)] \\ &\quad - \tilde{W}^T(k)\Gamma_w^{-1}\mathrm{Proj}_{\hat{W}} - \mathrm{Proj}_{\hat{W}}^T\Gamma_w^{-1}\tilde{W}(k) + \mathrm{Proj}_{\hat{W}}^T\Gamma_w^{-1}\mathrm{Proj}_{\hat{W}} \\ &\quad + tr\{-\tilde{V}^T(k)\Gamma_v^{-1}\mathrm{Proj}_{\hat{V}} - \mathrm{Proj}_{\hat{V}}^T\Gamma_v^{-1}\tilde{V}(k) + \mathrm{Proj}_{\hat{V}}^T\Gamma_v^{-1}\mathrm{Proj}_{\hat{V}}\} \end{aligned} \tag{2.44}$$

Considering the projection algorithms used, there are four possible Conditions:

1. All the elements of $\hat{W}(k)$ and $\hat{V}(k)$ are within the known prescribed fictitious bounds;

2. Only some elements of weight vector $\hat{W}(k)$ reach the fictitious bounds, projection algorithm (2.31) is applied;

3. Only some elements of weight matrix $\hat{V}(k)$ reach the fictitious bounds, projection algorithm (2.32) is applied;

4. Some elements of both $\hat{W}(k)$ and $\hat{V}(k)$ reach the fictitious bound, projection algorithms (2.31) and (2.32) are applied.

We will discuss them one by one in details below.

Condition 1. When all the elements of weight $\hat{W}(k)$ and $\hat{V}(k)$ are within the known prescribed bounds, equation (2.44) becomes

$$\begin{aligned}\triangle J(k) =& \frac{1}{g_1}\left[e^2(k+\tau)-e^2(k)\right] - \tilde{W}^T(k)\Gamma_w^{-1}\Gamma_w\eta_w - (\Gamma_w\eta_w)^T\Gamma_w^{-1}\tilde{W}(k)\\ &+(\Gamma_w\eta_w)^T\Gamma_w^{-1}\Gamma_w\eta_w + tr\{-\tilde{V}^T(k)\Gamma_v^{-1}\Gamma_v\eta_v - (\Gamma_v\eta_v)^T\Gamma_v^{-1}\tilde{V}(k)\\ &+(\Gamma_v\eta_v)^T\Gamma_v^{-1}\Gamma_v\eta_v\}\\ =& \frac{1}{g_1}[e^2(k+\tau)-e^2(k)] - 2\tilde{W}^T(k)\eta_w + \eta_w^T\Gamma_w^T\eta_w - 2tr\{\tilde{V}^T(k)\eta_v\} + tr\{\eta_v^T\Gamma_v^T\eta_v\}\end{aligned}$$

From equation (2.42), we obtain

$$\tilde{W}^T\hat{S} = \frac{e(k+\tau)-\delta_{d_k}}{f_u} - W^{*T}(\hat{S}-S^*) + \varepsilon_u(z)$$

Furthermore, considering the adaptive function (2.29) and (2.30), noticing that

$$tr\{\tilde{V}^T z_l \hat{W}^T\hat{S}'\} = \hat{W}^T\hat{S}'\tilde{V}^T z_l$$

and

$$tr\{(z_l\hat{W}^T\hat{S}')^T\Gamma_v^T(z_l\hat{W}^T\hat{S}')\} = \lambda_v\|z_l\hat{W}^T\hat{S}'\|_F^2$$

then

$$\begin{aligned}\triangle J(k) =& \left[\frac{1}{g_1}-\frac{2}{f_u}\right]e^2(k+\tau) - \frac{1}{g_1}e^2(k) + 2W^{*T}(\hat{S}-S^*)e(k+\tau)\\ &-2\left[\varepsilon_u(z)-\frac{\delta_{d_k}}{f_u}\right]e(k+\tau) + \hat{S}^T\Gamma_w^T\hat{S}e^2(k+\tau) - 2\hat{W}^T\hat{S}'\tilde{V}^T z_l e(k+\tau)\\ &+\lambda_v\|z_l\hat{W}^T\hat{S}'\|_F^2 e^2(k+\tau)\end{aligned}$$

Noticing Assumption 2.5 and the following inequalities

- $-\frac{2}{f_u} \leq -\frac{2}{g_1}$
  Noticing Assumption 2.4, we obtain the above inequality.

- $2W^{*T}(\hat{S} - S^*)e(k+\tau) \leq 4\|W^*\|\sqrt{l}|e(k+\tau)|$
  Because every element of $\hat{S}$ and S^* is less than 1, then $(\hat{S} - S^*) \leq \|\hat{S}\| + \|S^*\| = 2\sqrt{l}$. The above inequality holds.

- $-2\left[\varepsilon_u(z) - \frac{\delta_{d_k}}{f_u}\right] e(k+\tau) \leq 2\left[\varepsilon_l + \frac{\tau g_2 d}{\epsilon}\right]|e(k+\tau)|$
  Because $\varepsilon_u(z) \leq \varepsilon_l$, $\delta_{d_k} \leq \tau g_2 d$ and $\epsilon < |f_u|$, the above inequality holds.

- $\hat{S}^T \Gamma_w^T \hat{S} e^2(k+\tau) \leq \lambda_w l e^2(k+\tau)$
  Because $\Gamma_w = \lambda_w I$ is a positive diagonal matrix, $\hat{S}^T \Gamma_w^T \hat{S} = \lambda_w \hat{S}^T \hat{S}$. Furthermore, noticing every element of $\hat{S}$ is less than 1, thus the inner product of $\hat{S}$ must be less than its dimension l.

- $-2\hat{W}^T \hat{S}' \tilde{V}^T z_l e(k+\tau) \leq 0.5\hat{w}_m l \tilde{v}_m |e(k+\tau)|$
  Noticing Assumption 2.7, we have $\|\hat{W}\| \leq \hat{w}_m$ and $\|\tilde{V}\|_F \leq \tilde{v}_m$. By definition, $\|z_l\| = 1$. Furthermore, noticing equation (2.9), we have $\|\hat{S}'\|_F \leq 0.25l$. Thus, the above inequality holds.

- $\lambda_v \|z_l \hat{W}^T \hat{S}'\|_F^2 e^2(k+\tau) \leq 0.0625\lambda_v \hat{w}_m^2 l^2 e^2(k+\tau)$
  Noticing $\|z_l \hat{W}^T \hat{S}'\|_F \leq \|z_l\| \|\hat{W}\| \|\hat{S}'\|_F \leq 0.25 l \hat{w}_m$, the above inequality holds.

we have

$$\begin{aligned}\triangle J(k) \quad &\leq \quad -\left[\frac{1}{g_1} - \lambda_w l - 0.0625\lambda_v \hat{w}_m^2 l^2\right] e^2(k+\tau) - \frac{1}{g_1} e^2(k) \\ &\quad + \left[4\|W^*\|\sqrt{l} + 2\varepsilon_l + \frac{2\tau g_2 d}{\epsilon} + 0.5\hat{w}_m l \tilde{v}_m\right] |e(k+\tau)|\end{aligned}$$

If we choose the parameters to satisfy the following condition,

$$\frac{1}{g_1} - \lambda_w l - 0.0625\lambda_v \hat{w}_m^2 l^2 > 0 \tag{2.45}$$

and define positive constants

$$\begin{aligned}\alpha &= \frac{1}{g_1} - \lambda_w l - 0.0625\lambda_v \hat{w}_m^2 l^2 \\ \beta &= 4\|W^*\|\sqrt{l} + 2\varepsilon_l + 2\frac{\tau g_2 d}{\epsilon} + 0.5\hat{w}_m l \tilde{v}_m\end{aligned}$$

then

$$\begin{aligned}\triangle J(k) &\leq -\alpha|e(k+\tau)|^2 + \beta|e(k+\tau)| - \frac{1}{g_1}e^2(k) \\ &\leq -\alpha|e(k+\tau)|^2 + \beta|e(k+\tau)| \\ &= -\alpha|e(k+\tau)|\left(|e(k+\tau)| - \frac{\beta}{\alpha}\right)\end{aligned}$$

Define compact set

$$\Omega_e \triangleq \left\{e(k)\middle||e(k)| < \frac{\beta}{\alpha}\right\}$$

we can see that once $|e(k+\tau)|$ is out of the compact set Ω_e, $\triangle J(k) < 0$. That means $e(k+\tau)$ will converge to the compact set denoted by Ω_e.

Now it still remains to show that the weight estimates $\hat{W}(k)$ and $\hat{V}(k)$ are bounded. Considering the projection algorithms we used, it is obvious that $\hat{W}(k)$ and $\hat{V}(k)$ are bounded in compact sets.

Finally, for all $k \geq 0$, $J(k)$ is bounded because

$$\begin{aligned}J(k) &= \frac{1}{g_1}\sum_{j=0}^{\tau-1} e^2(k+j) + \sum_{j=0}^{\tau-1} \tilde{W}^T(k+j)\Gamma_w^{-1}\tilde{W}(k+j) \\ &\quad + \sum_{j=0}^{\tau-1} tr\{\tilde{V}^T(k+\tau)\Gamma_v^{-1}\tilde{V}(k+\tau)\}\end{aligned}$$

as $k \rightarrow +\infty$, we have

$$\begin{aligned}J(\infty) &= \frac{1}{g_1}\sum_{j=0}^{\tau-1} e^2(\infty+j) + \sum_{j=0}^{\tau-1} \tilde{W}^T(\infty+j)\Gamma_w^{-1}\tilde{W}(\infty+j) \\ &\quad + \sum_{j=0}^{\tau-1} tr\{\tilde{V}^T(\infty+\tau)\Gamma_v^{-1}\tilde{V}(\infty+\tau)\}\end{aligned}$$

Because we have proved that $e(k)$ is bounded, $\tilde{W}(k)$ and $\tilde{V}(k)$ are all bounded by the projection algorithms, we obtain $J(\infty) < \infty$, that is to say $J(k)$ is also bounded.

<u>Condition 2.</u> When only some elements of $\hat{W}(k)$ reach the fictitious bounds, equation (2.44) becomes

$$\triangle J(k) = \frac{1}{g_1}[e^2(k+\tau) - e^2(k)]$$

$$-\tilde{W}^T(k)\Gamma_w^{-1}\text{Proj}_{\hat{W}} - \text{Proj}_{\hat{W}}^T\Gamma_w^{-1}\tilde{W}(k) + \text{Proj}_{\hat{W}}^T\Gamma_w^{-1}\text{Proj}_{\hat{W}}$$
$$+tr\{-\tilde{V}^T(k)\Gamma_v^{-1}\Gamma_v\eta_v - (\Gamma_v\eta_v)^T\Gamma_v^{-1}\tilde{V}(k) + (\Gamma_v\eta_v)^T\Gamma_v^{-1}\Gamma_v\eta_v\}$$

Adding and subtracting $-\tilde{W}^T(k)\eta_w - \eta_w^T\tilde{W}(k) + \eta_w^T\Gamma_w^T\eta_w$ to the right side of the above equation, we obtain

$$\begin{aligned}\triangle J(k) &= \frac{1}{g_1}[e^2(k+\tau) - e^2(k)] - \tilde{W}^T(k)\eta_w - \eta_w^T\tilde{W}(k) + \eta_w^T\Gamma_w^T\eta_w \\ &\quad +tr\{-\tilde{V}^T(k)\Gamma_v^{-1}\Gamma_v\eta_v - (\Gamma_v\eta_v)^T\Gamma_v^{-1}\tilde{V}(k) + (\Gamma_v\eta_v)^T\Gamma_v^{-1}\Gamma_v\eta_v\} \\ &\quad -\tilde{W}^T(k)\Gamma_w^{-1}\text{Proj}_{\hat{W}} - \text{Proj}_{\hat{W}}^T\Gamma_w^{-1}\tilde{W}(k) + \text{Proj}_{\hat{W}}^T\Gamma_w^{-1}\text{Proj}_{\hat{W}} \\ &\quad +\tilde{W}^T(k)\eta_w + \eta_w^T\tilde{W}(k) - \eta_w^T\Gamma_w^T\eta_w \\ &= \frac{1}{g_1}[e^2(k+\tau) - e^2(k)] - \tilde{W}^T(k)\eta_w - \eta_w^T\tilde{W}(k) + \eta_w^T\Gamma_w^T\eta_w \\ &\quad +tr\{-\tilde{V}^T(k)\Gamma_v^{-1}\Gamma_v\eta_v - (\Gamma_v\eta_v)^T\Gamma_v^{-1}\tilde{V}(k) + (\Gamma_v\eta_v)^T\Gamma_v^{-1}\Gamma_v\eta_v\} \\ &\quad -\tilde{W}^T(k)(\Gamma_w^{-1}\text{Proj}_{\hat{W}} - \eta_w) - (\text{Proj}_{\hat{W}}^T\Gamma_w^{-1} - \eta_w^T)\tilde{W}(k) \\ &\quad +\text{Proj}_{\hat{W}}^T\Gamma_w^{-1}\text{Proj}_{\hat{W}} - \eta_w^T\Gamma_w^T\eta_w\end{aligned}$$

Noticing Lemma 2.3, using equations (2.35) and (2.37), we have

$$\begin{aligned}\triangle J(k) &\leq \frac{1}{g_1}[e^2(k+\tau) - e^2(k)] - \tilde{W}^T(k)\eta_w - \eta_w^T\tilde{W}(k) + \eta_w^T\Gamma_w^T\eta_w \\ &\quad +tr\{-\tilde{V}^T(k)\Gamma_v^{-1}\Gamma_v\eta_v - (\Gamma_v\eta_v)^T\Gamma_v^{-1}\tilde{V}(k) + (\Gamma_v\eta_v)^T\Gamma_v^{-1}\Gamma_v\eta_v\}\end{aligned}$$

which is the same as we discussed in Condition 1. Thus, we obtain the same stability results.

<u>Condition 3.</u> When only some elements of $\hat{V}(k)$ reach the fictitious bounds, equation (2.44) becomes

$$\begin{aligned}\triangle J(k) &= \frac{1}{g_1}[e^2(k+\tau) - e^2(k)] - \tilde{W}^T(k)\eta_w - \eta_w^T\tilde{W}(k) + \eta_w^T\Gamma_w^T\eta_w \\ &\quad -tr\{\tilde{V}^T(k)\Gamma_v^{-1}\text{Proj}_{\hat{V}} + \text{Proj}_{\hat{V}}^T\Gamma_v^{-1}\tilde{V}(k) - \text{Proj}_{\hat{V}}^T\Gamma_v^{-1}\text{Proj}_{\hat{V}}\}\end{aligned}$$

Adding and subtracting $tr\{-\tilde{V}^T(k)\Gamma_v^{-1}\Gamma_v\eta_v - (\Gamma_v\eta_v)^T\Gamma_v^{-1}\tilde{V}(k) + (\Gamma_v\eta_v)^T\Gamma_v^{-1}\Gamma_v\eta_v\}$ to the right side of the above equation, we obtain

$$\begin{aligned}\triangle J(k) &= \frac{1}{g_1}[e^2(k+\tau) - e^2(k)] - \tilde{W}^T(k)\eta_w - \eta_w^T\tilde{W}(k) + \eta_w^T\Gamma_w^T\eta_w \\ &\quad +tr\{-\tilde{V}^T(k)\Gamma_v^{-1}\Gamma_v\eta_v - (\Gamma_v\eta_v)^T\Gamma_v^{-1}\tilde{V}(k) + (\Gamma_v\eta_v)^T\Gamma_v^{-1}\Gamma_v\eta_v\} \\ &\quad -tr\{\tilde{V}^T(k)[\Gamma_v^{-1}\text{Proj}_{\hat{V}} - \eta_v]\} - tr\{[\text{Proj}_{\hat{V}}^T\Gamma_v^{-1} - \eta_v^T]\tilde{V}(k)\} \\ &\quad +tr\{\text{Proj}_{\hat{V}}^T\Gamma_v^{-1}\text{Proj}_{\hat{V}}\} - tr\{\eta_v^T\Gamma_v^T\eta_v\}\end{aligned}$$

Noticing Lemma 2.3, using equations (2.36) and (2.38), the above equation can be written as

$$\begin{aligned}\triangle J(k) \leq & \frac{1}{g_1}[e^2(k+\tau)-e^2(k)] - \tilde{W}^T(k)\eta_w - \eta_w^T\tilde{W}(k) + \eta_w^T\Gamma_w^T\eta_w \\ & +tr\{-\tilde{V}^T(k)\Gamma_v^{-1}\Gamma_v\eta_v - (\Gamma_v\eta_v)^T\Gamma_v^{-1}\tilde{V}(k) + (\Gamma_v\eta_v)^T\Gamma_v^{-1}\Gamma_v\eta_v\}\end{aligned}$$

which is the same as we discussed in Condition 1. Thus, we obtain the same stability results.

<u>Condition 4.</u> When there are elements of both $\hat{W}(k)$ and $\hat{V}(k)$ reach the fictitious bounds, equation (2.44) becomes

$$\begin{aligned}\triangle J(k) = & \frac{1}{g_1}[e^2(k+\tau)-e^2(k)] \\ & -\tilde{W}^T(k)\Gamma_w^{-1}\mathrm{Proj}_{\hat{W}} - \mathrm{Proj}_{\hat{W}}^T\Gamma_w^{-1}\tilde{W}(k) + \mathrm{Proj}_{\hat{W}}^T\Gamma_w^{-1}\mathrm{Proj}_{\hat{W}} \\ & -tr\{\tilde{V}^T(k)\Gamma_v^{-1}\mathrm{Proj}_{\hat{V}} + \mathrm{Proj}_{\hat{V}}^T\Gamma_v^{-1}\tilde{V}(k) - \mathrm{Proj}_{\hat{V}}^T\Gamma_v^{-1}\mathrm{Proj}_{\hat{V}}\}\end{aligned}$$

Adding and subtracting $tr\{-\tilde{V}^T(k)\Gamma_v^{-1}\Gamma_v\eta_v - (\Gamma_v\eta_v)^T\Gamma_v^{-1}\tilde{V}(k) + (\Gamma_v\eta_v)^T\Gamma_v^{-1}\Gamma_v\eta_v\}$ and $-\tilde{W}^T(k)\eta_w - \eta_w^T\tilde{W}(k) + \eta_w^T\Gamma_w^T\eta_w$ to the right side of the above equation, we obtain

$$\begin{aligned}\triangle J(k) = & \frac{1}{g_1}[e^2(k+\tau)-e^2(k)] - \tilde{W}^T(k)\eta_w - \eta_w^T\tilde{W}(k) + \eta_w^T\Gamma_w^T\eta_w \\ & +tr\{-\tilde{V}^T(k)\Gamma_v^{-1}\Gamma_v\eta_v - (\Gamma_v\eta_v)^T\Gamma_v^{-1}\tilde{V}(k) + (\Gamma_v\eta_v)^T\Gamma_v^{-1}\Gamma_v\eta_v\} \\ & -\tilde{W}^T(k)\Gamma_w^{-1}\mathrm{Proj}_{\hat{W}} - \mathrm{Proj}_{\hat{W}}^T\Gamma_w^{-1}\tilde{W}(k) + \mathrm{Proj}_{\hat{W}}^T\Gamma_w^{-1}\mathrm{Proj}_{\hat{W}} \\ & +\tilde{W}^T(k)\eta_w + \eta_w^T\tilde{W}(k) - \eta_w^T\Gamma_w^T\eta_w \\ & -tr\{\tilde{V}^T(k)[\Gamma_v^{-1}\mathrm{Proj}_{\hat{V}} - \eta_v]\} - tr\{[\mathrm{Proj}_{\hat{V}}^T\Gamma_v^{-1} - \eta_v^T]\tilde{V}(k)\} \\ & +tr\{\mathrm{Proj}_{\hat{V}}^T\Gamma_v^{-1}\mathrm{Proj}_{\hat{V}}\} - tr\{\eta_v^T\Gamma_v^T\eta_v\}\end{aligned}$$

$$\begin{aligned} = & \frac{1}{g_1}[e^2(k+\tau)-e^2(k)] - \tilde{W}^T(k)\eta_w - \eta_w^T\tilde{W}(k) + \eta_w^T\Gamma_w^T\eta_w \\ & +tr\{-\tilde{V}^T(k)\Gamma_v^{-1}\Gamma_v\eta_v - (\Gamma_v\eta_v)^T\Gamma_v^{-1}\tilde{V}(k) + (\Gamma_v\eta_v)^T\Gamma_v^{-1}\Gamma_v\eta_v\} \\ & -\tilde{W}^T(k)[\Gamma_w^{-1}\mathrm{Proj}_{\hat{W}} - \eta_w] - [\mathrm{Proj}_{\hat{W}}^T\Gamma_w^{-1} - \eta_w^T]\tilde{W}(k) \\ & +\mathrm{Proj}_{\hat{W}}^T\Gamma_w^{-1}\mathrm{Proj}_{\hat{W}} - \eta_w^T\Gamma_w^T\eta_w \\ & -tr\{\tilde{V}^T(k)[\Gamma_v^{-1}\mathrm{Proj}_{\hat{V}} - \eta_v]\} - tr\{[\mathrm{Proj}_{\hat{V}}^T\Gamma_v^{-1} - \eta_v^T]\tilde{V}(k)\} \\ & +tr\{\mathrm{Proj}_{\hat{V}}^T\Gamma_v^{-1}\mathrm{Proj}_{\hat{V}}\} - tr\{\eta_v^T\Gamma_v^T\eta_v\}\end{aligned}$$

Noticing Lemma 2.3, using equations (2.35)-(2.38), the above equation can be further written as

$$\begin{aligned}\triangle J(k) \;\leq\; & \frac{1}{g_1}[e^2(k+\tau)-e^2(k)] - \tilde{W}^T(k)\eta_w - \eta_w^T\tilde{W}(k) + \eta_w^T\Gamma_w^T\eta_w \\ & + tr\{-\tilde{V}^T(k)\Gamma_v^{-1}\Gamma_v\eta_v - (\Gamma_v\eta_v)^T\Gamma_v^{-1}\tilde{V}(k) + (\Gamma_v\eta_v)^T\Gamma_v^{-1}\Gamma_v\eta_v\}\end{aligned}$$

which is the same as we discussed in Condition 1. Thus, we obtain the same stability results.

We can see that Condition 2-4 can be transformed to Condition 1, in which we have proved that the tracking error to be bounded in a compact set, then Theorem 2.2 holds. Therefore, for any a priori given (arbitrarily large) bounded set Ω and any a priori given (arbitrarily small) set Ω_0, which contains $(0,0)$ as an interior point, there exist a control u, such that every trajectory of the closed-loop system starting from Ω enters the set Ω_0 in a finite time and remains in it thereafter. That is to say, the whole closed-loop system is SGUUB. ■

Remark 2.10 *It is shown that the error bound $\beta = k_1(\varepsilon_l + \frac{\tau g_2 d}{\epsilon})^2 + \beta_1 + \beta_2$ cannot be made arbitrarily small. This is simply because that in the process of proof, we use completion of square many times, which magnify the error terms. In fact, if the MNN weight can approximate the ideal weight sufficiently close and there are no disturbances exist, then there is only $k_1\varepsilon_l^2$ left in the expression of β. Then it is obvious that the tracking error can be made arbitrarily small by increasing the approximation accuracy.*

Remark 2.11 *The aim of using projection algorithms in this chapter is to guarantee the boundedness of the MNN weight. In simulation process, the fictitious upper bound $\hat{\rho}_{\Theta,\max}$ and lower bound $\hat{\rho}_{\Theta,\min}$ bound can be chosen sufficiently large at the very beginning, which can guarantee all practical weight vector or matrix element in $[\hat{\rho}_{\Theta,\min}, \hat{\rho}_{\Theta,\max}]$, and guarantee the convergence of the system.*

2.5 Numerical Simulation

Considering non-affine nonlinear discrete-time system described by the following difference equations

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ x_2(k+1) &= \frac{x_1(k)x_2(k)(x_1(k)+2.5)}{1+x_1^2(k)+x_2^2(k)} + u(k) + 0.1u^3(k) + d(k) \\ y(k) &= x_1(k) \end{aligned}$$

and the disturbance

$$d(k) = 0.1\cos(0.001k)$$

Its τ-steps-ahead model should be in the following form

$$y(k+\tau) = y(k+2) = f(y(k), y(k-1), u(k), u(k-1), d(k+1), d(k))$$

where $\tau = 2$.

The control gain $\frac{\partial f}{\partial u} = 1 + 0.3u^2(k) > 0$, considering $u(k) \in \Omega_u$, it is obvious that the control gain is $1 \leq \frac{\partial f}{\partial u} \leq g_1$ which satisfies the assumption. Because the disturbance is not known, then it cannot be used as the input of controller neural networks. The inputs of neural networks are $z = [\bar{y}_k, \bar{u}_{k-1}, y_m(k+\tau)]^T = [y(k), y(k-1), u(k-1), y_m(k+2)]^T$.

2.5.1 RBF Control Simulation

System initial conditions are chosen as follows, the neurons number $l = 5^4 = 625$ which is a large number, consider that $y(k)$ and $y(k-1)$ are almost the same in every step $k \to k+1$, then they can be approximated by using the same neurons. Thus the RBF neurons used in simulation can be reduced to $l = 5^3 = 125$. The center of the receptive field of RBF neural networks are chosen as follows, $y(k), y(k-1), y_m(k+2) \in \{-0.8, -0.4, 0, 0.4, 0.8\}$, $u(k-1) \in \{-1.0, -0.725, -0.450, -0.175, 0.1\}$. The width of the Gaussian function is initialized to 1.

The weight vector is initialized to 0, the initial states of the discrete-time system are set to 0. The adaptive law gain diagonal matrix is $\Gamma = 0.002I$ and the σ modification gain is $\sigma = 0.01$. Reference signal is $y_m(k) = 0.8\sin(\frac{\pi k}{400})$. The disturbance signal is $d(k) = 0.1\cos(0.001k)$.

The simulation results are presented in Figures 2.3-2.5. Figure 2.3 states that the system output following the reference model, the transient performance is bounded. Figure 2.4 shows the Implicit Desired Feedback Control trajectory. Figure 2.5 shows the RBF weight vector norm which is bounded.

2.5.2 MNN Control Simulation

For multi-layer neural networks controller, the design parameters are chosen as follows. The weight vector $\hat{W}$ and $\hat{V}$ are initialized to 0, the initial states of the discrete-time system are set to 0. The adaptive law gain diagonal matrix is $\Gamma_w = 0.05I$ and $\Gamma_v = 0.05I$. Reference signal is $y_m(k) = 0.8sin(\frac{\pi}{400}k)$. The disturbance signal is $d(k) = 0.1\cos(0.001k)$. The multi-layer neural networks neurons number is $l = 10$.

The simulation results are presented in Figures 2.6-2.8. Figure 2.6 states that the system output following the reference model, the transient performance is bounded. Figure 2.7 shows the Implicit Desired Feedback Control trajectory. Figure 2.8 shows the MNN weight vector norm and matrix norm which are all bounded.

Remark 2.12 *Noticing the simulation results, we can see that for the control trajectories of RBF and MNN controller, they are almost the same except their transient performance. This verifies the existence and uniqueness of the Implicit Desired Feedback Control.*

Remark 2.13 *Generally speaking, no matter increasing the number of the neurons used or choosing larger adaptation gain matrix will improve the performance. But this increase is limited to some extent. Noticing that for RBF controller, the adaptation gain is much smaller than that for the MNN controller, but the neuron number used for the former is much more than latter.*

2.6 Application to Practical CSTR Systems

Many industrial processes including distillation columns, exothermic chemical reactions, and PH neutralization can exhibit significant nonlinear behavior. If these processes are operated at a nominal steady state, the effects of the nonlinearities may not be severe and traditional control schemes based on local linearized models provide satisfactory control performance. However, if the systems are required to work over a wide range of conditions, conventional linear control approaches cannot handle the system nonlinearities. In recent years, many interesting results for chemical process control have been reported in the literature [122, 123, 124, 125]. Most of these feedback linearization strategies require exact mathematical models of the plant dynamics. However, it is generally difficult in practice to obtain an accurate model because of the inherent complexity of the chemical processes or the lack of informative process data. It is necessary to implement adaptive techniques or other robust control techniques. A number of applications of on-line adaptation in feedback controller design have been documented in the literature that demonstrated superior performance in the presence of unknown and time-varying process parameters [126, 127, 128, 129, 89, 64, 130, 131]. In practical industrial control applications, usually the information available are system outputs and inputs at discrete-time instants. Then it is necessary to investigated discrete-time input-output based control schemes which are different from the continuous time based control methods in the literatures.

Finally, noting the possible application of advance control technique in industrial processes, two CSTR systems are studied to show the effectiveness of the developed control method. The first CSTR system is in non-affine form for which the IDFC control cannot be expressed explicitly. The simulation results of the non-affine CSTR system shows the effectiveness of the developed MNN controller. The second CSTR system is in affine form, that means the IDFC control can be expressed explicitly. Thus for the affine CSTR system, both the ideal IDFC trajectory and direct MNN control trajectory can be obtained. The simulation results shows that the MNN control trajectory gradually approximate the ideal IDFC control. This states that the method of using MNN to emulate the IDFC control is effective.

The control objective is to make the concentration y track the set-point step change signal $y_d(t)$. In order to get a smooth reference signal, a linear reference model is used to shape the discontinuous reference signal for providing the desired signals y_d. The following reference model is to be implemented

$$\frac{y_d(s)}{r(s)} = \frac{\omega_n^2}{s^2 + 2\zeta_n\omega_n s + \omega_n^2}$$

where the natural frequency $\omega_n = 5.0 rad/min$ and the damping ratio $\zeta_n = 1.0$.

2.6.1 Non-affine CSTR System

Continuous System Description: Consider the CSTR system shown in Figure 2.1. The process dynamics are described by [123, 128, 50]

$$\begin{aligned}\dot{C}_a &= \frac{q}{V}(C_{a_0} - C_a) - a_0 C_a e^{-\frac{E}{RT_a}} \\ \dot{T}_a &= \frac{q}{V}(T_f - T_a) + a_1 C_a e^{-\frac{E}{RT_a}} + a_3 q_c[1 - e^{-\frac{a_2}{q_c}}](T_{cf} - T_a)\end{aligned} \tag{2.46}$$

where the variables C_a and T_a are the concentration and temperature of a tank, respectively; the coolant flow rate q_c is the control input and the parameters of the plant are defined in Table 2.1. Within the tank reactor, two chemicals are mixed and react to produce compound A at a concentration C_a with the temperature of the mixture being T_a. The reaction is both irreversible and exothermic. The control objective is to manipulate the coolant flow rate q_c to control the C_a at a desired value. It should be noticed that the above description of CSTR is different from those of conventional chemical reactor control systems [125]. In most applications, the coolant temperature is chosen as the manipulated variable and assumed to be constant through the cooling coil [123]. There are two major advantages of choosing the flow rate q_c as the manipulated control input. Firstly, the coolant temperature is allowed to vary along the length of the cooling coil [132]. If the cooling coil is long, which happens in many practical plants, the assumption of constant coolant temperature may cause significant bias in the CSTR model. Secondly, manipulation of the flow rate yields an easily implementable control scheme compared to the manipulation of the coolant temperature.

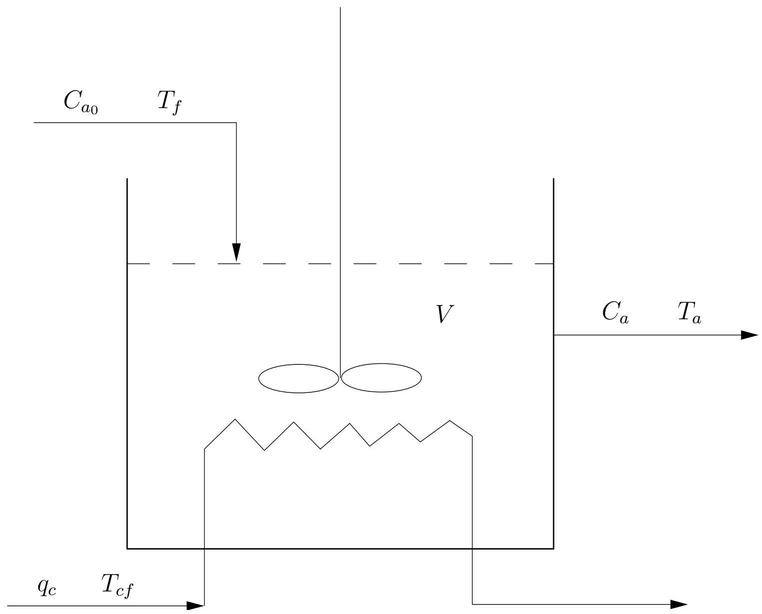

Figure 2.1: Continuously Stirred Tank Reactor System

The major challenge of this control problem is that the plant does not assume the customary control affine system structure because the control input q_c appears non-linearly. For the case where the system model is known exactly, the application of input-output linearization control for a large class of general nonlinear systems has been investigated in [123]. The application of neural network to uncertain nonlinear systems was studied in [133] and [24]. Control applications to CSTR systems were provided to illustrate the advantage of utilizing learning and adaptation. However, due to the high complexity of neural network system, a rigorous stability analysis was not provided for the closed-loop control system in [133]. The scheme presented in [24] requires the measurement of the time derivative of C_a which is difficult to estimate in practice. In [50], adaptive neural network control scheme is presented for this continuous CSTR system.

The state variables, the input and the output are defined as $x = [x_1, x_2]^T = [C_a, T_a]^T$, $u = q_c$, $y = C_a$. Using this notation, the CSTR plant (2.46) can be re-expressed as [50]

$$\begin{aligned}\dot{x}_1 &= 1 - x_1 - a_0 x_1 e^{-\frac{10^4}{x_2}} \\ \dot{x}_2 &= 350 - x_2 + a_1 x_1 e^{-\frac{10^4}{x_2}} + a_3 u (1 - e^{-\frac{a_2}{u}})(350 - x_2)\end{aligned}$$

Parameter	Description	Nominal value
q	process flow rate	$100l/min$
C_{a_0}	concentration of component A	$1mol/l$
T_f	feed temperature	$350k$
T_{cf}	inlet coolant temperature	$350k$
V	volume of tank	$100l$
h_a	heat transfer coefficient	$7 \times 10^5 J/min \cdot K$
a_0	preexponential factor	$7.2 \times 10^{10} min^{-1}$
$\frac{E}{R}$	activation energy	$1 \times 10^4 K$
$(-\triangle H)$	heat of reaction	$2 \times 10^5 cal/mol$
ρ_1, ρ_c	liquid densities	$1 \times 10^3 g/l$
C_p, C_{pc}	heat capacities	$1cal/g \cdot K$
$a_1 = \frac{(-\triangle H)a_0}{\rho_1 C_p} = 1.44 \times 10^{13}$	$a_2 = \frac{h_a}{\rho_c C_{pc}} = 6.987 \times 10^2$	$a_3 = \frac{\rho_c C_{pc}}{\rho_1 C_p V} = 0.01$

Table 2.1: Nomenclature List (Non-affine CSTR System)

$$y \;=\; x_1 \tag{2.47}$$

The control objective is to design a controller u such that the output y follows a desired signal y_d.

Given the parameters listed in Table 2.1 and the irreversible exothermic property of the chemical process, the operating condition of the CSTR system are restricted to

$$\Omega_x = \left\{ (x_1, x_2, u) \middle| 0.02 < x_1 < 0.8, 350 \leq x_2 < T_{max}, 0 \leq u \leq u_{max} \right\} \tag{2.48}$$

where the constants T_{max} and u_{max} are the maximum values of the coolant flow rate and the tank temperature, respectively.

Discretized System Model: Generally speaking, the exact discretization of nonlinear continuous dynamics is based on the Lie derivatives and leads to an infinite series representation [134]. In fact, the exact discretization of a continuous system must reproduce the continuous-time solution at the sampling instants, when the initial states are equal. It is shown in [135] that the exact discretization is given by the Lie-series. The exact discretization is an infinite power series in the input u and sampling time T that will converge if the input signal u is bounded and the sampling period T is sufficiently small. Various approximate discretization techniques use truncated versions of the exact series. In this chapter, first order Taylor expansion is used to approximate the derivative of x_1 and x_2, by discarding the high order error items, the dynamic properties of the continuous CSTR system (2.47) can be approximate by the following discrete-time system

$$\begin{aligned} x_1(k+1) &= x_1(k) + \dot{x}_1(k)T && (2.49)\\ x_2(k+1) &= x_2(k) + \dot{x}_2(k)T && (2.50)\\ y(k) &= x_1(k) && (2.51) \end{aligned}$$

where T is the sampling period and k represents kT, $x_1(k)$, $x_2(k)$ stand for corresponding state variables at sampling instants of continuous system (2.47) and

$$\begin{aligned} \dot{x}_1(k) &= 1 - x_1(k) - a_0 x_1(k) e^{-\frac{10^4}{x_2(k)}} && (2.52)\\ \dot{x}_2(k) &= 350 - x_2(k) + a_1 x_1(k) e^{-\frac{10^4}{x_2(k)}} + a_3 u(k)(1 - e^{-\frac{a_2}{u(k)}})(350 - x_2(k)) && (2.53) \end{aligned}$$

Substitute equation (2.52) into (2.49), we obtain

$$\begin{aligned} x_2(k) &= -\frac{10^4}{\ln \frac{-x_1(k+1)+(1-T)x_1(k)+T}{a_0 T x_1(k)}} = -\frac{10^4}{\ln \frac{-y(k+1)+(1-T)y(k)+T}{a_0 T y(k)}} \\ &\triangleq f_1(y(k+1), y(k)) && (2.54) \end{aligned}$$

Thus

$$x_2(k-1) = f_1(y(k), y(k-1)) \tag{2.55}$$

Furthermore, noticing $x_2(k+1) = x_2(k) + \dot{x}_2(k)T$ and equations (2.53) and (2.55), we obtain

$$\begin{aligned} x_2(k) &= x_2(k-1) + \dot{x}_2(k-1)T \\ &= f_1(y(k), y(k-1)) + \Big[350 - f_1(y(k), y(k-1)) + a_1 y(k-1) e^{-\frac{10^4}{f_1(y(k), y(k-1))}} \\ &\quad + a_3 u(k-1)(1 - e^{-\frac{a_2}{u(k-1)}})(350 - f_1(y(k), y(k-1)))\Big] T \\ &\triangleq f_2(y(k), y(k-1), u(k-1)) \end{aligned}$$

Noticing $x_1(k+1) = x_1(k) + \dot{x}_1(k)T$ and equation (2.52), we obtain

$$\begin{aligned} x_1(k+1) &= y(k) + \left[1 - y(k) - a_0 y(k) e^{-\frac{10^4}{f_2(y(k), y(k-1), u(k-1))}}\right] T \\ &\triangleq f_3(y(k), y(k-1), u(k-1)) && (2.56) \end{aligned}$$

From equation (2.53), we obtain

$$\begin{aligned} x_2(k+1) &= x_2(k) + \dot{x}_2(k)T \\ &= f_2(y(k), y(k-1), u(k-1)) + \Big[350 - f_2(y(k), y(k-1), u(k-1)) \\ &\quad + a_1 y(k) e^{-\frac{10^4}{f_2(y(k), y(k-1), u(k-1))}} \\ &\quad + a_3 u(k)(1 - e^{-\frac{a_2}{u(k)}})(350 - f_2(y(k), y(k-1), u(k-1)))\Big] T \\ &\triangleq f_4(y(k), y(k-1), u(k-1), u(k)) \end{aligned} \tag{2.57}$$

Finally, combining equations (2.56) and (2.57), we have

$$\begin{aligned} y(k+2) &= x_1(k+2) = x_1(k+1) + \dot{x}_1(k+1)T \\ &= f_3(y(k), y(k-1), u(k-1)) + \Big[1 - x_1(k+1) - a_0 x_1(k+1) e^{-\frac{10^4}{x_2(k+1)}}\Big] T \\ &= f_3(y(k), y(k-1), u(k-1)) + \Big[1 - f_3(y(k), y(k-1), u(k-1)) \\ &\quad - a_0 f_3(y(k), y(k-1), u(k-1)) e^{-\frac{10^4}{f_4(y(k), y(k-1), u(k-1), u(k))}}\Big] T \\ &\triangleq f_0(y(k), y(k-1), u(k), u(k-1)) \end{aligned}$$

If the sampling time T is sufficient small, this approximation is reasonable. Then the CSTR system can be transformed into the above τ-step ahead input-output model [95] (here $\tau = 2$). Consider the modelling error and disturbances, we have the following expression

$$\begin{aligned} y(k+2) &= f(y(k), y(k-1), u(k-1), d(k+1), d(k), u(k)) \\ &= f(\bar{y}_k, \bar{u}_{k-1}, \bar{d}_{k+\tau-1}, u(k)) \end{aligned} \tag{2.58}$$

where $\bar{y}_k = [y(k), y(k-1)]^T$, $\bar{u}_{k-1} = [u(k-1)]^T$ and $\bar{d}_{k+1} = [d(k+1), d(k)]^T$. The sequence $\{d(k)\}$ represents modelling error and disturbances.

Remark 2.14 *Usually the system description is in state space, in order to get the τ-step ahead input-output discrete-time model, iteration transform should be used. In the above procedure, we directly transform state space model to input-output τ-step ahead model. In Section 2.6.2, different transform method is used. Diffeomorphism is introduced to formulate the procedure.*

Remark 2.15 *For stable systems, the stability does not depend on the sampling interval, however, to ensure good closed-loop performance, the sampling interval should be small enough to capture adequately the dynamics of the process, yet large enough to permit the online computations necessary for implementation. Large sampling interval can result in ringing (excessive oscillations) between sample points. An example of this phenomenon is provided by Garcia and Morari using a linear system in [136].*

In fact, system (2.58) is just a special case of system (2.1), NARMAX model. Therefore, we can use the MNN control scheme to control this class of CSTR systems.

Numerical Simulation: For the non-affine system (2.46), it is verified in [50] that the control gain of the CSTR system is lower bounded by a positive constant. Therefore, the existence of the IDFC is guaranteed. The system initial value is $x(0) = [0.1, 440]^T$. Number of neurons used is $l = 40$. Neural network weights $\hat{W}(0) = 0$ and $\hat{V}(0) = 0$. $\Gamma_w = 0.1I$ and $\Gamma_v = 0.15I$.

Simulation results are shown in Figures 2.9-2.11. It can be seen from the simulation results, in Figure 2.9, the system output concentration follows the desired trajectory-step changes at the nominal operating point ($x_1 = 0.1 \pm 0.02$). Figure 2.10 shows that the MNN weight vectors norm are bounded. Figure 2.11 shows that the actual control input u varies around the nominal operating point.

2.6.2 Affine CSTR System

Continuous System Description: To show the existence of the IDFC controller, an affine CSTR system is studied in this section. As a special kind of non-affine system, for affine system, the IDFC control can be expressed explicitly. Consider an irreversible exothermic reaction A→B, carried out in a perfectly mixed CSTR as shown in Figure 2.2 [137].

Taking into account that the inlet flow rate F_0 is equal to the outlet flow rate F, then $\frac{dV}{dt} = 0$, the energy balances on the reactor and cooling jacket yield:

$$V\frac{dC_A}{dt} = F_0(C_{A_0} - C_A) - VkC_A$$

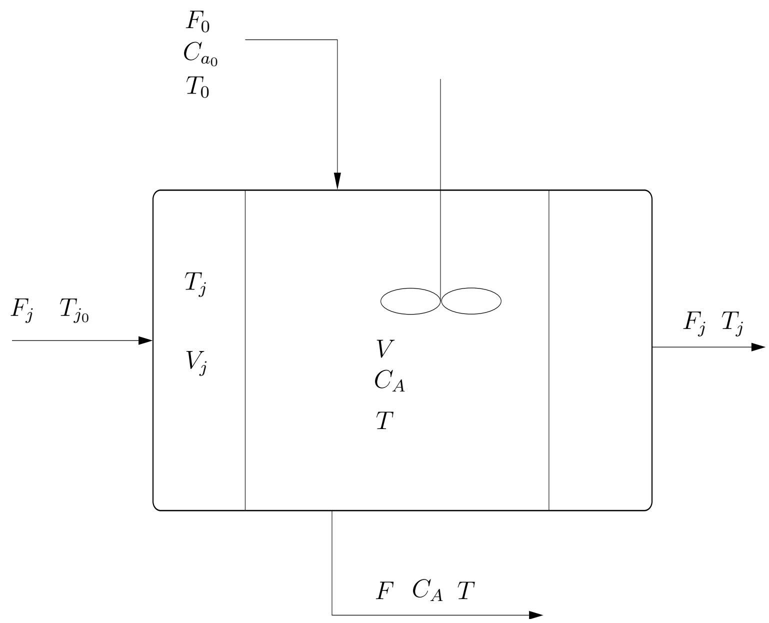

Figure 2.2: Exothermic Reaction in a CSTR

$$\rho C_p V \frac{dT}{dt} = \rho C_p F_0 (T_0 - T) - \lambda V k C_A - UA(T - T_j) \tag{2.59}$$

where the variables are detailed in Table 2.2. The dynamics of non-dimensional heat and mass balances are given by [138]:

$$\begin{aligned} \dot{x}_1 &= -x_1 + D_a(1 - x_1) e^{\frac{x_2}{1+\frac{x_2}{\gamma}}} \\ \dot{x}_2 &= -x_2 + BD_a(1 - x_1) e^{\frac{x_2}{1+\frac{x_2}{\gamma}}} - \beta(x_2 - u) + d \end{aligned} \tag{2.60}$$

The term d is added to represent an unmeasured load disturbance.

Discretized System Model: By using this affine CSTR system, the procedure of how to convert this system to discrete system is as follows. Firstly, using Lie derivative, define diffeomorphism

$$\begin{aligned} \xi_1 &= x_1 \\ \xi_2 &= -x_1 + D_a(1 - x_1) e^{\frac{x_2}{1+\frac{x_2}{\gamma}}} \end{aligned}$$

Thus $x_1 = \xi_1$, $\dot{x}_1 = \xi_2$ and

$$x_2 = \frac{\gamma \ln \frac{\xi_1 + \xi_2}{D_a(1-\xi_1)}}{\gamma - \ln \frac{\xi_1 + \xi_2}{D_a(1-\xi_1)}} \tag{2.61}$$

Parameter	Description
A	heat transfer surface
B	dimensionless heat of reaction $B = -\frac{\triangle H C_{A_0}\gamma}{C_p T_0}$
C_A	reactant concentration
C_{A_0}	feed concentration of reactant
C_p	heat capacity
D_a	Damkohler Number $D_a = \frac{V k_0 e^{-\gamma}}{Q_f}$
E	activation energy
$\triangle H$	heat of reaction
k_0	reaction rate constant
Q_f	mass feed flow rate
R	ideal gas constant
T_j	coolant temperature
T	feed temperature
T_0	nominal feed temperature
U	overall hear transfer coefficient
V	reactor volume
x_1	dimensionless concentration $x_1 = \frac{C_{A_0} - C_A}{C_{A_0}}$
x_2	dimensionless temperature $x_2 = \frac{(T-T_0)\gamma}{T_0}$
u	dimensionless coolant temperature $u = \frac{(T_j - T_0)\gamma}{T_0}$
β	dimensionless cooling rate $\beta = \frac{UA}{Q_f C_p}$
γ	dimensionless activation energy $\gamma = \frac{E}{RT_0}$

Table 2.2: Nomenclature List (Affine CSTR System)

Hence,

$$\dot{x}_2 = -\frac{\gamma \ln \frac{\xi_1+\xi_2}{D_a(1-\xi_1)}}{\gamma - \ln \frac{\xi_1+\xi_2}{D_a(1-\xi_1)}} + B(\xi_1+\xi_2) - \beta \frac{\gamma \ln \frac{\xi_1+\xi_2}{D_a(1-\xi_1)}}{\gamma - \ln \frac{\xi_1+\xi_2}{D_a(1-\xi_1)}} + \beta u \tag{2.62}$$

Since

$$\dot{\xi}_2 = -\xi_2 + D_a(1-\xi_1)e^{\frac{1}{\frac{1}{x_2}+\frac{1}{\gamma}}} \frac{1}{(1+\frac{x_2}{\gamma})^2}\dot{x}_2 - D_a\xi_2 e^{\frac{1}{\frac{1}{x_2}+\frac{1}{\gamma}}} \tag{2.63}$$

Substituting (2.61) and (2.62) into equation (2.63), the CSTR system can be transformed to

$$\dot{\xi}_1 = \xi_2$$

$$\begin{aligned}\dot{\xi}_2 &= f_1(\xi_1,\xi_2)+f_u(\xi_1,\xi_2)u\\ y &= \xi_1\end{aligned}$$

where

$$\begin{aligned} f_1(\xi_1,\xi_2) &= \frac{\xi_1+\xi_2}{\left[1+\frac{\ln\frac{\xi_1+\xi_2}{D_a(1-\xi_1)}}{\gamma-\ln\frac{\xi_1+\xi_2}{D_a(1-\xi_1)}}\right]^2}\left[-(1+\beta)\frac{\gamma\ln\frac{\xi_1+\xi_2}{D_a(1-\xi_1)}}{\gamma-\ln\frac{\xi_1+\xi_2}{D_a(1-\xi_1)}}+B(\xi_1+\xi_2)\right]\\ &\quad -\xi_2\frac{\xi_1+\xi_2}{1-\xi_1}-\xi_2\\ f_u(\xi_1,\xi_2) &= \frac{\xi_1+\xi_2}{\left[1+\frac{\ln\frac{\xi_1+\xi_2}{D_a(1-\xi_1)}}{\gamma-\ln\frac{\xi_1+\xi_2}{D_a(1-\xi_1)}}\right]^2}\beta \end{aligned}$$

Then by using sampling time T

$$\begin{aligned}\xi_1(k+1) &= \xi_1(k)+\dot{\xi}_1(k)T\\ &= \xi_1(k)+\xi_2(k)T && (2.64)\\ \xi_2(k+1) &= \xi_2(k)+\dot{\xi}_2(k)T\\ &= \xi_2(k)+f_1(\xi_1(k),\xi_2(k))T+f_u(\xi_1(k),\xi_2(k))u(k)T && (2.65)\\ y(k) &= \xi_1(k) && (2.66)\end{aligned}$$

Thus, noticing equations (2.64), (2.65) and (2.66), we have

$$\begin{aligned}\xi_1(k) &= y(k)\\ \xi_2(k) &= \xi_2(k-1)+f_1(\xi_1(k-1),\xi_2(k-1))T+f_u(\xi_1(k-1),\xi_2(k-1))u(k-1)T\\ &= \frac{y(k)-y(k-1)}{T}+f_1\left(y(k-1),\frac{y(k)-y(k-1)}{T}\right)T\\ &\quad +f_u\left(y(k-1),\frac{y(k)-y(k-1)}{T}\right)u(k-1)T\\ &\triangleq f_2(y(k),y(k-1),u(k-1))\end{aligned}$$

Therefore

$$\begin{aligned}y(k+2) &= \xi_1(k+1)+\xi_2(k+1)T\\ &= \xi_1(k)+\xi_2(k)T+\xi_2(k+1)T\\ &= y(k)+\xi_2(k)T+\xi_2(k)T+f_1(\xi_1(k),\xi_2(k))T^2+f_u(\xi_1(k),\xi_2(k))T^2u(k)\end{aligned}$$

$$
\begin{aligned}
&= y(k) + 2f_2(y(k), y(k-1), u(k-1))T \\
&\quad + f_1(y(k), f_2(y(k), y(k-1), u(k-1)))T^2 \\
&\quad + f_u(y(k), f_2(y(k), y(k-1), u(k-1)))T^2 u(k) \\
&= f_3(y(k), y(k-1), u(k-1)) \\
&\quad + f_u(y(k), f_2(y(k), y(k-1), u(k-1)))T^2 u(k)
\end{aligned}
$$

where

$$
\begin{aligned}
f_3(y(k), y(k-1), u(k-1)) &= y(k) + 2f_2(y(k), y(k-1), u(k-1))T \\
&\quad + f_1(y(k), f_2(y(k), y(k-1), u(k-1)))T^2
\end{aligned}
$$

Thus, for this affine CSTR system, the IDFC control can be expressed explicitly

$$
u^*(k) = \frac{y_d(k+\tau) - f_3(y(k), y(k-1), u(k-1))}{f_u(y(k), f_2(y(k), y(k-1), u(k-1)))T^2} \tag{2.67}
$$

The same as in Section 2.6.1, the MNN control scheme proposed is used to control this class of discretized affine CSTR System.

<u>Numerical Simulation:</u> Simulation parameters are $B = 21.5$, $\gamma = 28.5$, $D_a = 0.036$ and $\beta = 25.2$. The definition of these parameters are given in Table 2.2. The concentration of reactant A, x_1, is to be controlled by manipulating the temperature of the coolant, u.

For this affine CSTR system, it is easy to verify that the control gain is lower bounded by a positive constant. Therefore, the existence of the desired feedback control is guaranteed. System initial value is $x(0) = [0.4, 3.3]^T$. Number of neurons used is $l = 10$. Neural network weights $\hat{W}(0) = 0$ and $\hat{V}(0) = 0$. $\Gamma_w = 0.1I$ and $\Gamma_v = 0.1I$. The control objective is to manipulate the coolant temperature u to control the concentration x_1 tracking set-point step change. The nominal operating point of the CSTR was at $x_1 = 0.4126 \pm 0.02$, $x_2 = 3.28$ and $u = 3.04$.

Simulation results are shown in Figures 2.12-2.14. It can be seen from the simulation results, in Figure 2.12, the system output concentration follows the desired trajectory-step changes at the nominal operating point ($x_1 = 0.4126$). Figure 2.13 shows that the MNN weight vector norms are bounded. Figure 2.14 states that the actual control input u varies around the nominal operating point (u=3.04). Furthermore in Figure 2.14, the dash dotted line indicates the ideal implicit desired feedback control $u^*(k)$ in (2.67). It is obvious that the developed discrete-time MNN controller can emulate the ideal IDFC controller very accurately. This shows the effectiveness of the developed method.

Remark 2.16 *The procedure of how to convert a continuous system into a τ-step ahead discrete-time system shows in Section 2.6.2. It should be noticed that the sampling time should be small enough to guarantee the same dynamic property of the continuous and discrete system. The reason that we use diffeomorphism first is that, by using this conversion, we can easily get the relationship between the two states $\xi_1(k)$ and $\xi_2(k)$ (equation (2.64) $\xi_2(k) = \frac{\xi_1(k+1)-\xi_1(k)}{T} = \frac{y(k+1)-y(k)}{T}$), which will make the following process easier.*

Remark 2.17 *In practical applications, the high oscillation of the output is an undesirable behavior and should be reduced. By training the neural network weights, the high oscillation can be reduced. As clearly indicated in Figure 2.9, by using the neural network weights at the end of the 1st time run as the initial value of the 2nd time run, the oscillation peak was reduced, though was not completely eliminated.*

2.7 Conclusion

In this chapter, adaptive NN control scheme was investigated for a class of non-affine nonlinear discrete-time systems in NARMAX form. Based on implicit function theorem, RBF neural networks and MNNs were used respectively as the emulators to approximate the IDFC controller. All MNN weights were tuned online with no prior training needed. Discontinuous projections with fictitious bounds were used in the MNN weights tuning laws to guarantee that all MNN weights remain in a prescribed range. The stability of the closed-loop system was proved rigorously by using Lyapunov theorem and the simulation results show the effectiveness of the developed control method.

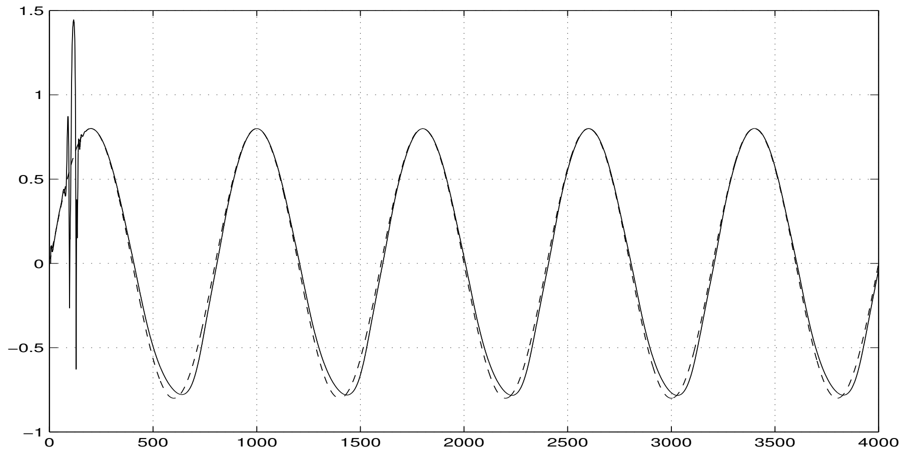

Figure 2.3: RBF Control - Tracking Performance

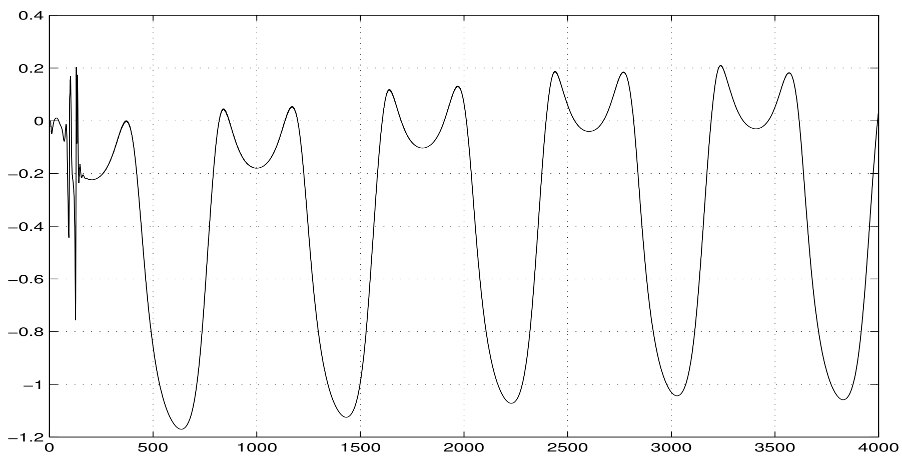

Figure 2.4: RBF Control - Input Trajectory

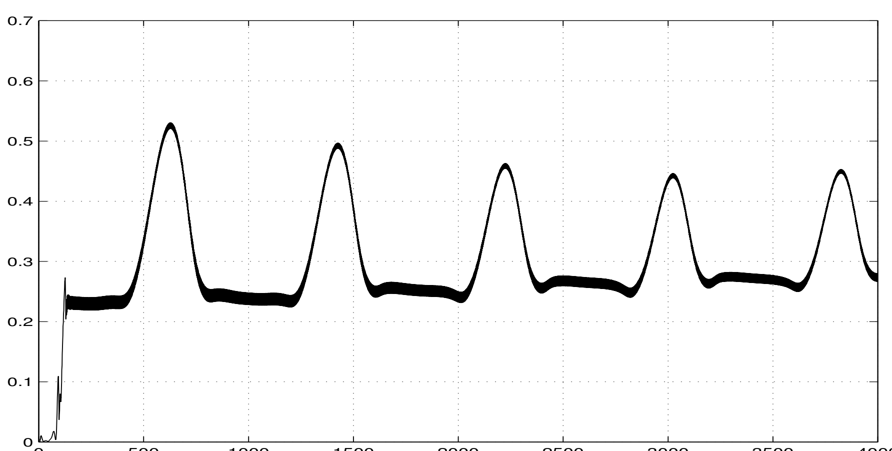

Figure 2.5: RBF Control - Weight Norm $\|\hat{W}\|_2$

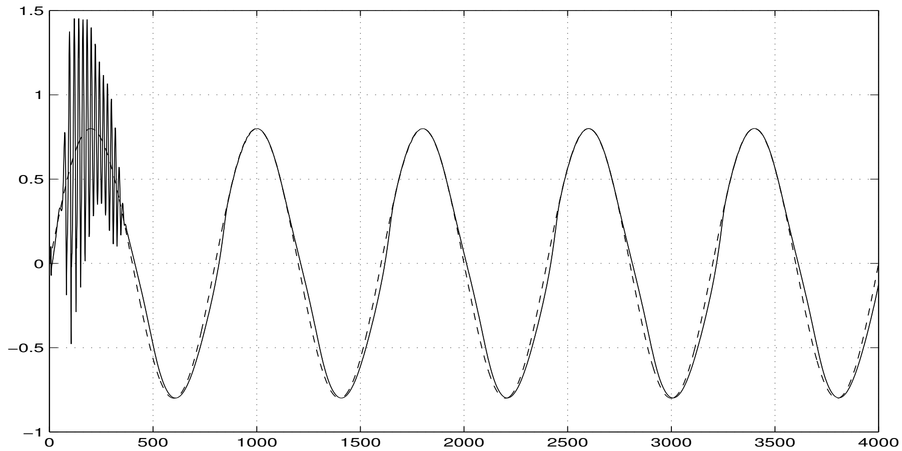

Figure 2.6: MNN Control - Tracking Performance

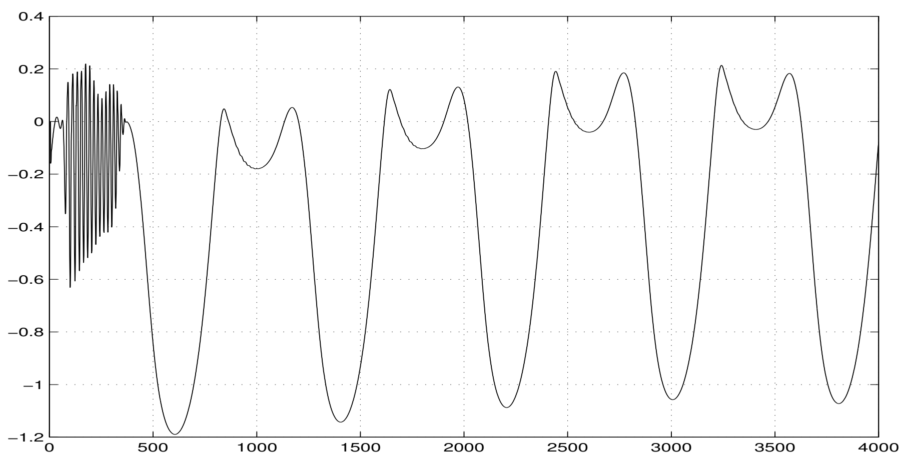

Figure 2.7: MNN Control - Input Trajectory

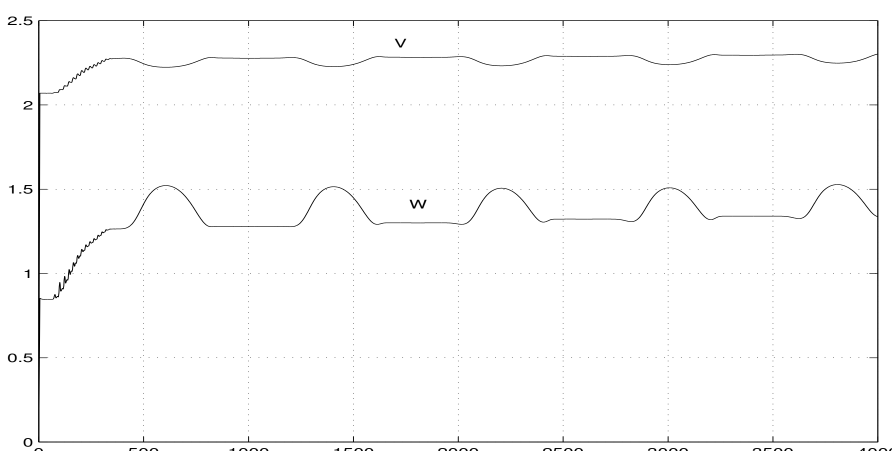

Figure 2.8: MNN Control - Weight Norm $\|\hat{W}\|_2$ and $\|\hat{V}\|_F$

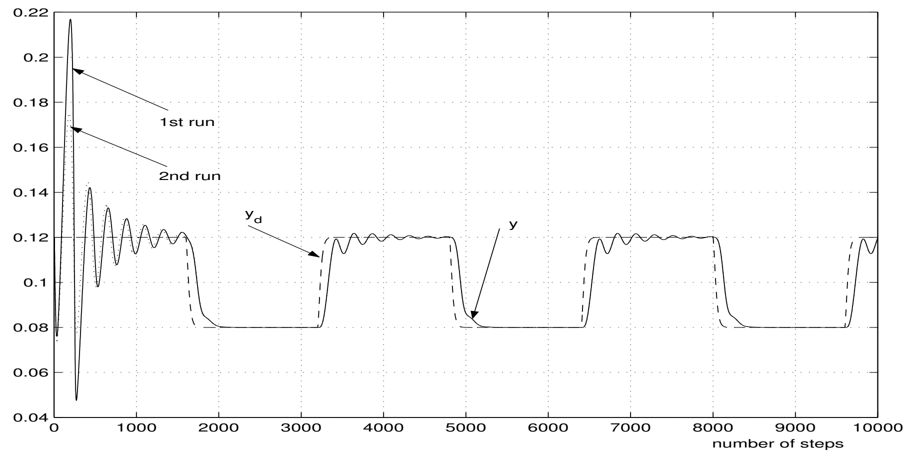

Figure 2.9: Non-affine CSTR - Tracking Performance

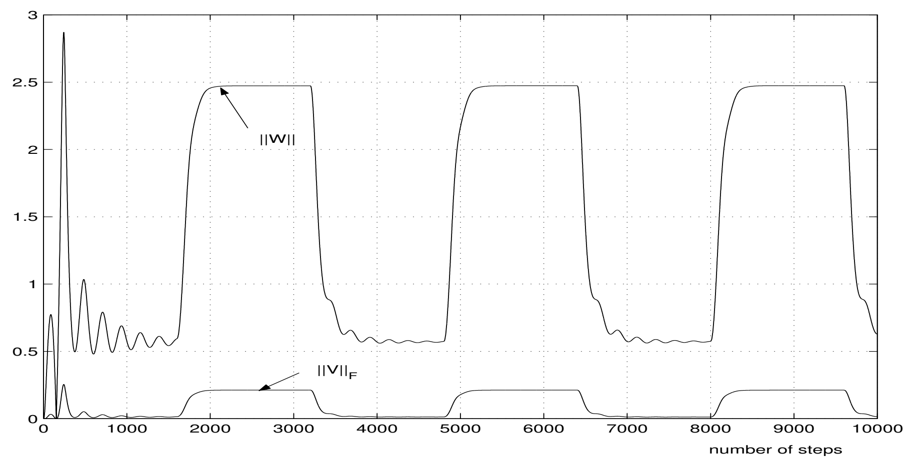

Figure 2.10: Non-affine CSTR - Weight Norm $\|\hat{W}\|$ and $\|\hat{V}\|_F$

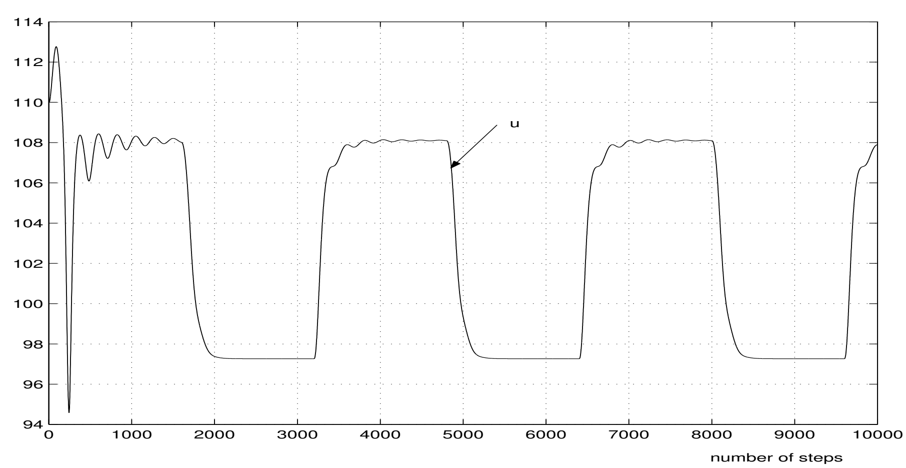

Figure 2.11: Non-affine CSTR - Control Trajectory

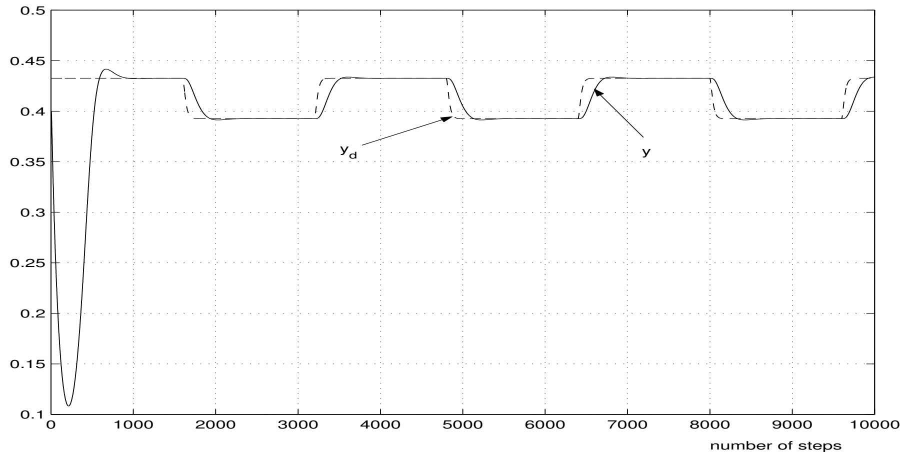

Figure 2.12: Affine CSTR - Tracking Performance

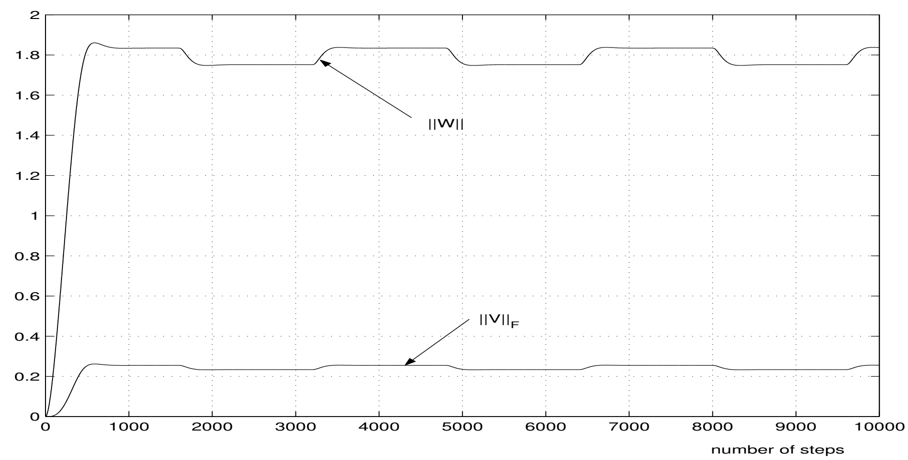

Figure 2.13: Affine CSTR - Weight Norm $\|\hat{W}\|$ and $\|\hat{V}\|_F$

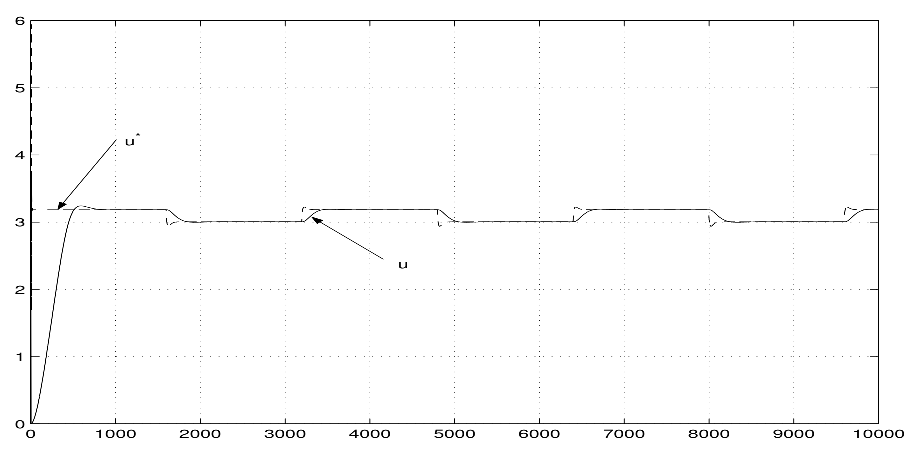

Figure 2.14: Affine CSTR - Control Trajectory

Chapter 3

NN Control of MIMO Systems with Triangular Form Inputs

In this chapter, adaptive NN control schemes are investigated for MIMO nonlinear discrete-time systems in state space representation. The chapter is organized as follows. Firstly, for a class of MIMO discrete-time nonlinear systems with triangular form inputs and disturbances, an effective state feedback control method is proposed in Section 3.1. Then, for a class of similar MIMO discrete-time systems without disturbances, an output feedback control scheme is investigated in Section 3.2. Conclusions are made in Section 3.3.

3.1 State Feedback Control

For nonlinear MIMO discrete-time systems, due to the couplings among subsystems, the various inputs and the various outputs, the control problem is more complex and few results are available in the literature relative to that in continuous time domain. Besides the difficulty of input coupling in continuous time MIMO system, non-causal problem [115] is another difficulty that is probably to be met when construct stable adaptive controllers for discrete-time systems. Furthermore, for neural network based MIMO nonlinear discrete-time system control, how to tune the NN weights is still a difficult problem, especially when there is unknown strong interconnections between subsystems. Due to these difficulties, researches on discrete-time nonlinear MIMO system control is not only challenging but also of academic interest. In [139] and [111], two layer neural networks and multi-layer neural networks were used respectively to construct stable controls for a special class of discrete-time nonlinear MIMO systems. Improved weight tuning algorithms were derived, which removes the need of persistent exciting (PE) condition for parameter convergence [8]. Though the methods proposed are effective, they are only applicable to a special class of discrete-time nonlinear MIMO systems, which can be represented in the form of $X(k+1) = F(X(k)) + GU(k)$, with G being a diagonal constant matrix. This is a very special class of discrete-time MIMO nonlinear systems without any interconnections between subsystems. Another effective neural network control scheme was developed for a class of discrete-time nonlinear MIMO systems based on input-output model in [140]. The MIMO system studied is in NARMAX model [101] and only past input and output data are used to construct stable NN control.

In this section, we are considering a class of more challenging discrete-time MIMO nonlinear system in state space description. Comparing with the systems studied in [139, 111], the control inputs of the system studied in this section are in triangular form that can only be represented as $X(k+1) = F(X(k), U(k))$ instead of $X(k+1) = F(X(k)) + G(X(k))U(k)$. Therefore, feedback linearization method is not applicable. In [115], an effective HONN control scheme for a class of strict feedback discrete-time nonlinear SISO system was proposed. Motivated by the design procedure in [115], we investigate a class of MIMO nonlinear discrete-time systems with unknown bounded disturbances here, which extend the results obtained in [115]. There are n subsystems in the MIMO system under study, with each subsystem in strict feedback form. States interconnections between different subsystems only appear in the last equations of each subsystems, where the corresponding controls also appear. By transforming the MIMO system into a Sequential Decrease Cascade Form, the non-causal problem is avoided.

The section is organized as follows. System dynamics and some stability notions are proposed in Section 3.1.1. The causality analysis and system transformation are proposed in Section 3.1.2. Controller design, neural network weight update law and stability analysis are studied in Section 3.1.3 via backstepping. Simulation results are given in Section 3.1.4 to show the effectiveness of the proposed control scheme.

3.1.1 MIMO System Dynamics

Considering the following n inputs n outputs discrete-time MIMO nonlinear systems

$$\Sigma : \begin{cases} \Sigma_1 : \begin{cases} x_{1,i_1}(k+1) = f_{1,i_1}(\bar{x}_{1,i_1}(k)) + g_{1,i_1}(\bar{x}_{1,i_1}(k))x_{1,i_1+1}(k) \\ \qquad 1 \le i_1 \le n_1 - 1 \\ x_{1,n_1}(k+1) = f_{1,n_1}(X(k)) + g_{1,n_1}(X(k))u_1(k) + d_1(k) \end{cases} \\ \vdots \\ \Sigma_j : \begin{cases} x_{j,i_j}(k+1) = f_{j,i_j}(\bar{x}_{j,i_j}(k)) + g_{j,i_j}(\bar{x}_{j,i_j}(k))x_{j,i_j+1}(k) \\ \qquad 1 \le i_j \le n_j - 1 \\ x_{j,n_j}(k+1) = f_{j,n_j}(X(k), \bar{u}_{j-1}(k)) + g_{j,n_j}(X(k))u_j(k) + d_j(k) \end{cases} \\ \vdots \\ \Sigma_n : \begin{cases} x_{n,i_n}(k+1) = f_{n,i_n}(\bar{x}_{n,i_n}(k)) + g_{n,i_n}(\bar{x}_{n,i_n}(k))x_{n,i_n+1}(k) \\ \qquad 1 \le i_n \le n_n - 1 \\ x_{n,n_n}(k+1) = f_{n,n_n}(X(k), \bar{u}_{n-1}(k)) + g_{n,n_n}(X(k))u_n(k) + d_n(k) \end{cases} \\ y_j(k) = x_{j,1}(k), \quad 1 \le j \le n \end{cases} \tag{3.1}$$

where

$$\begin{aligned} x_j(k) &= [x_{j,1}(k), x_{j,2}(k), \ldots, x_{j,n_j}(k)]^T \in R^{n_j} \\ X(k) &= [x_1^T(k), x_2^T(k), \ldots, x_n^T(k)]^T \\ u(k) &= [u_1(k), \ldots, u_n(k)]^T \in R^n \\ y(k) &= [y_1(k), \ldots, y_n(k)]^T \in R^n \end{aligned}$$

are the state variables, the inputs and outputs respectively, $d(k) = [d_1(k), \ldots, d_n(k)]^T$ is the bounded disturbance vector; $\bar{u}_{j-1}(k) = [u_1(k), \cdots, u_{j-1}(k)]$ $(j = 2, \ldots, n)$; $\bar{x}_{j,i_j}(k) = [x_{j,1}(k), \ldots, x_{j,i_j}(k)]^T \in R^{i_j}$ denotes the first i_j states of the j-th subsystem; $f_{j,i_j}(\cdot)$ and $g_{j,i_j}(\cdot)$ are smooth nonlinear functions; and j, i_j, and n_j are positive constants. It can be seen that each subsystem of (3.1) is in strict feedback form, which makes the use of backstepping design technique possible. Furthermore, noting that the control inputs of the whole system are in triangular form, then we may use backstepping in a nested manner to design stable controls for this class of systems.

Remark 3.1 *It should be noted that, different from the triangular form inputs MIMO nonlinear discrete-time system studied in [139, 111], whose inputs can be written into feedback linearizable form*

$$\begin{aligned} X(k+1) &= F(X(k)) + G(X(k))U(k) \\ U(k) &= [u_1(k), \ldots, u_n(k)]^T \end{aligned} \tag{3.2}$$

system (3.1) studied in this section cannot be written into the form of (3.2), due to the triangular form inputs. Instead, it is in the following form

$$\begin{aligned} X(k+1) &= F(X(k), \bar{U}_{n-1}(k)) + G(X(k))U(k) \\ U(k) &= [u_1(k), \ldots, u_n(k)]^T, \quad \bar{U}_{n-1}(k) = [u_1(k), \ldots, u_{n-1}(k)]^T \end{aligned} \tag{3.3}$$

It is obvious that feedback linearization method is not applicable for system (3.3). To construct stable controls for this class of system which is not feedback linearizable is more challenging.

In order to use backstepping design technique, it is required that the gains of each virtual control are not equal to zero. Therefore, the following assumption should be made.

Assumption 3.1 *The sign of $g_{j,i_j}(\cdot)$ ($j = 1, \ldots, n,\ i_j = 1, \ldots, n_j$), are known and there exist two constants $\underline{g}_{j,i_j}, \bar{g}_{j,i_j} > 0$ such that $\underline{g}_{j,i_j} \le |g_{j,i_j}(\cdot)| \le \bar{g}_{j,i_j}$, $\forall X(k) \in \Omega \subset R^{\sum_{i=1}^{n} n_i}$.*

Without losing generality, we shall assume that $g_{j,i_j}(\cdot)$ is positive in this section. The control objective is to design control input $u(k) = [u_1(k), \ldots, u_n(k)]^T$ to make the system output $y(k) = [y_1(k), \ldots, y_n(k)]^T$ follow a known and bounded trajectory $y_d(k) = [y_{d_1}(k), \ldots, y_{d_n}(k)]^T$. Thus, the following assumption should be made.

Assumption 3.2 *The desired trajectory $y_d(k) \in \Omega_y$, $\forall k > 0$ is smooth and known, where $\Omega_y \triangleq \{\chi \big| \chi = y(k)\}$.*

In [141] and [47], the definition of Uniform Ultimate Boundedness (UUB) for continuous time system has been given. A standard Lyapunov theorem extension proposed in [142] provided a method on how to judge the UUB stability. For completeness, it is cited here.

Theorem 3.1 *Let $V(x)$ be a Lyapunov function of a continuous time system that satisfies the following properties:*

$$\begin{aligned} \gamma_1(\|x\|) &\leq V(x) \leq \gamma_2(\|x\|) \\ \dot{V}(x) &\leq -\gamma_3(\|x\|) + \gamma_3(\eta) \end{aligned}$$

where η is a positive constant, $\gamma_1(\cdot)$ and $\gamma_2(\cdot)$ are continuous strictly increasing functions, and $\gamma_3(\cdot)$ is a continuous, nondecreasing function. Thus, if

$$\dot{V}(x) < 0, \quad \textit{for } \|x\| > \eta$$

then $x(t)$ is uniformly ultimately bounded. In addition, if $x(0) = 0$, $x(t)$ is uniformly bounded [142].

Similar to the definition of UUB for continuous time system, its counterpart in discrete-time system is as follows.

Definition 3.1 *The solution of (3.1) is SGUUB, if for any Ω, a compact subset of $R^{\sum_{i=1}^{n} n_i}$ and all $X(k_0) \in \Omega$, there exist an $\epsilon > 0$, and a number $N(\epsilon, X(k_0))$ such that $\|X(k)\| < \epsilon$ for all $k \geq k_0 + N$. In other words, the solution of (3.1) is said to be SGUUB if, for any a priori given (arbitrarily large) bounded set Ω and any a priori given (arbitrarily small) set Ω_0, which contains $(0, 0)$ as an interior point, there exist a control u, such that every trajectory of the closed-loop system starting from Ω enters the set Ω_0 in a finite time and remains in it thereafter [141].*

Lemma 3.1 *Let $V(x(k))$ be a Lyapunov function of a discrete-time system that satisfies the following properties:*

$$\begin{aligned} \gamma_1(\|x(k)\|) &\leq V(x(k)) \leq \gamma_2(\|x(k)\|) \\ V(x(k+1)) - V(x(k)) &= \Delta V(x(k)) \leq -\gamma_3(\|x(k)\|) + \gamma_3(\eta) \end{aligned} \tag{3.4}$$

where η is a positive constant, $\gamma_1(\cdot)$ and $\gamma_2(\cdot)$ are strictly increasing functions, and $\gamma_3(\cdot)$ is a continuous, non decreasing function. Thus, if

$$\Delta V(x(k)) < 0, \quad \textit{for } \|x(k)\| > \eta$$

then $x(k)$ is uniformly ultimately bounded on a compact set, i.e., there exists a time instant k_T, such that $\|x(k)\| < \eta$, $\forall\ k > k_T$.

Remark 3.2 *It should be noted that, the operator $\|\cdot\|$ in Lemma 3.1 can be any positive defined mono-increasing function or norm.*

3.1.2 Causality Analysis and System Transformation

In this section, the same as in [115], coordinate transform is used to avoid the non-causal problem, which often appears in discrete-time nonlinear system control. We have known that each subsystem of system (3.1) is in strict feedback form. It seems that backstepping technique can be used to construct stable control. However, different from that in continuous time systems, for discrete-time systems, the causality contradiction [115] is one of the major problems that we will encounter when we construct controls for strict-feedback nonlinear system through backstepping, as detailed in the following.

Consider the first subsystem in system (3.1)

$$\Sigma_1 : \left\{ \begin{array}{rcl} x_{1,1}(k+1) & = & f_{1,1}(\bar{x}_{1,1}(k)) + g_{1,1}(\bar{x}_{1,1}(k))x_{1,2}(k) \\ x_{1,2}(k+1) & = & f_{1,2}(\bar{x}_{1,2}(k)) + g_{1,2}(\bar{x}_{1,2}(k))x_{1,3}(k) \\ & \vdots & \\ x_{1,n_1-1}(k+1) & = & f_{1,n_1-1}(\bar{x}_{1,n_1-1}(k)) + g_{1,n_1-1}(\bar{x}_{1,n_1-1}(k))x_{1,n_1}(k) \\ x_{1,n_1}(k+1) & = & f_{1,n_1}(X(k)) + g_{1,n_1}(X(k))u_1(k) + d_1(k) \end{array} \right. \tag{3.5}$$

If we design the ideal fictitious control for the first equation in (3.5) as follows:

$$\alpha^*_{1,2}(k) = -\frac{1}{g_{1,1}(\bar{x}_{1,1}(k))}\left[f_{1,1}(\bar{x}_{1,1}(k)) - y_{d_1}(k+1)\right]$$

the first equation in (3.5) can be stabilized. Similarly, we can construct another ideal fictitious control

$$\alpha^*_{1,3}(k) = -\frac{1}{g_{1,2}(\bar{x}_{1,2}(k))}\left[f_{1,2}(\bar{x}_{1,2}(k)) - \alpha^*_{1,2}(k+1)\right] \tag{3.6}$$

to stabilize the second equation in (3.5). But unfortunately, $\alpha^*_{1,2}(k+1)$ in (3.6) is a fictitious control of the future. This means that the fictitious control $\alpha^*_{1,3}(k)$ is infeasible in practice. If we continue the process to construct the final desired control $u^*_1(k)$, we end up with a $u^*_1(k)$ that is infeasible due to unavailable future information.

However, the above problem can be avoided if we transform the system equation into a special form which is suitable for backstepping design. The basic idea is as follows. If we consider the original system description as a one-step ahead predictor, and then we can transform the one-step ahead predictor into an equivalent maximum n_1-step ahead predictor which can predict the future states, $x_{1,1}(k+n_1)$, $x_{1,2}(k+n_1-1)$, ..., $x_{1,n_1}(k+1)$, then the causality contradiction is avoided when controller is constructed based on the maximum n_1-step ahead predictor by backstepping. For the other $n-1$ subsystems, this transformation is also applicable. The transformation procedure for the j-th ($1 \leq j \leq n_j$) subsystem is detailed as follows.

Consider the i_j-th equation in j-th subsystem of system (3.1)

$$\begin{aligned} x_{j,i_j}(k+1) &= f_{j,i_j}(\bar{x}_{j,i_j}(k)) + g_{j,i_j}(\bar{x}_{j,i_j}(k))x_{j,i_j+1}(k) \\ & \quad 1 \leq j \leq n \text{ and } 1 \leq i_j \leq n_j - 1 \end{aligned}$$

It can be easily obtained that $x_{j,i_j}(k+1)$ is a function of $\bar{x}_{j,i_j+1}(k)$. For convenience of analysis, we define

$$x_{j,i_j}(k+1) \triangleq f^{n_j}_{j,i_j}(\bar{x}_{j,i_j+1}(k)) \tag{3.7}$$

with

$$f^{n_j}_{j,i_j}(\bar{x}_{j,i_j+1}(k)) = f_{j,i_j}(\bar{x}_{j,i_j}(k)) + g_{j,i_j}(\bar{x}_{j,i_j}(k))x_{j,i_j+1}(k)$$

Thus, we have

$$\bar{x}_{j,i_j}(k+1) = \begin{bmatrix} x_{j,1}(k+1) \\ \vdots \\ x_{j,i_j}(k+1) \end{bmatrix} = \begin{bmatrix} f^{n_j}_{j,1}(\bar{x}_{j,2}(k)) \\ \vdots \\ f^{n_j}_{j,i_j}(\bar{x}_{j,i_j+1}(k)) \end{bmatrix}, \ 1 \leq j \leq n, 1 \leq i_j \leq n_j - 1$$

It can be seen that $\bar{x}_{j,i_j}(k+1)$ is a function of $\bar{x}_{j,i_j+1}(k)$. Define function vector

$$\bar{x}_{j,i_j}(k+1) \triangleq F^{n_j}_{j,i_j}(\bar{x}_{j,i_j+1}(k)), \quad i_j = 1, \ldots, n_j - 1 \tag{3.8}$$

After one more step, the first $n_j - 1$ equations of each subsystem in (3.1) can be expressed as

$$\left\{ \begin{aligned} x_{j,i_j}(k+2) &= f_{j,i_j}(\bar{x}_{j,i_j}(k+1)) + g_{j,i_j}(\bar{x}_{j,i_j}(k+1))x_{j,i_j+1}(k+1) \\ & \quad i_j = 1, 2, \ldots, n_j - 2 \\ x_{j,n_j-1}(k+2) &= f_{j,n_j-1}(\bar{x}_{j,n_j-1}(k+1)) + g_{j,n_j-1}(\bar{x}_{j,n_j-1}(k+1))x_{j,n_j}(k+1) \\ & \quad 1 \leq j \leq n \end{aligned} \right. \tag{3.9}$$

Substituting (3.7) and (3.8) into equation (3.9), we can obtain

$$
\left\{
\begin{array}{rcl}
x_{j,i_j}(k+2) & = & f_{j,i_j}(F_{j,i_j}^{n_j}(\bar{x}_{j,i_j+1}(k))) + g_{j,i_j}(F_{j,i_j}^{n_j}(\bar{x}_{j,i_j+1}(k)))f_{j,i_j+1}^{n_j}(\bar{x}_{j,i_j+2}(k)) \\
 & \triangleq & f_{j,i_j}^{n_j-1}(\bar{x}_{j,i_j+2}(k)) \\
 & & \quad i_j = 1, 2, \ldots, n_j - 2 \\
x_{j,n_j-1}(k+2) & = & f_{j,n_j-1}(F_{j,n_j-1}^{n_j}(\bar{x}_{j,n_j}(k))) + g_{j,n_j-1}(F_{j,n_j-1}^{n_j}(\bar{x}_{j,n_j}(k)))x_{j,n_j}(k+1) \\
 & \triangleq & F_{j,n_j-1}(\bar{x}_{j,n_j}(k)) + G_{j,n_j-1}(\bar{x}_{j,n_j}(k))x_{j,n_j}(k+1)
\end{array}
\right.
\qquad (3.10)
$$
$$1 \le j \le n$$

where

$$
\begin{array}{rcl}
f_{j,i_j}^{n_j-1}(\bar{x}_{j,i_j+2}(k)) & = & f_{j,i_j}(F_{j,i_j}^{n_j}(\bar{x}_{j,i_j+1}(k))) + g_{j,i_j}(F_{j,i_j}^{n_j}(\bar{x}_{j,i_j+1}(k)))f_{j,i_j+1}^{n_j}(\bar{x}_{j,i_j+2}(k)) \\
F_{j,n_j-1}(\bar{x}_{j,n_j}(k)) & = & f_{j,n_j-1}(F_{j,n_j-1}^{n_j}(\bar{x}_{j,n_j}(k))) \\
G_{j,n_j-1}(\bar{x}_{j,n_j}(k)) & = & g_{j,n_j-1}(F_{j,n_j-1}^{n_j}(\bar{x}_{j,n_j}(k)))
\end{array}
$$

Following the same procedure, the first $(n_j - 2)$ equations in (3.10) of the j-th subsystem of system (3.1) can be described by

$$
\bar{x}_{j,i_j}(k+2) = \begin{bmatrix} x_{j,1}(k+2) \\ \vdots \\ x_{j,i_j}(k+2) \end{bmatrix} = \begin{bmatrix} f_{j,1}^{n_j-1}(\bar{x}_{j,3}(k)) \\ \vdots \\ f_{j,i_j}^{n_j-1}(\bar{x}_{j,i_j+2}(k)) \end{bmatrix}, \ 1 \le i_j \le n_j - 2
$$

which is a function of $\bar{x}_{j,i_j+2}(k)$ and is denoted as

$$
\bar{x}_{j,i_j}(k+2) = F_{j,i_j}^{n_j-1}(\bar{x}_{j,i_j+2}(k)), \quad i_j = 1, \ldots, n_j - 2
$$

Continue the above procedure recursively, after $(n_j - 2)$ steps, the first two equations in the j-th subsystem of (3.1) can be written as

$$
\left\{
\begin{array}{rcl}
x_{j,1}(k+n_j-1) & = & f_{j,1}^{2}(\bar{x}_{j,n_j}(k)) \\
x_{j,2}(k+n_j-1) & = & F_{j,2}(\bar{x}_{j,n_j}(k)) + G_{j,2}(\bar{x}_{j,n_j}(k))x_{j,3}(k+n_j-2)
\end{array}
\right.
\qquad (3.11)
$$

where

$$
\begin{array}{rcl}
f_{j,1}^{2}(\bar{x}_{j,n_j}(k)) & = & f_{j,1}(F_{j,1}^{3}(\bar{x}_{j,n_j-1}(k))) + g_{j,1}(F_{j,1}^{3}(\bar{x}_{j,n_j-1}(k)))f_{j,2}^{3}(\bar{x}_{j,n_j}(k)) \\
F_{j,2}(\bar{x}_{j,n_j}(k)) & = & f_{j,2}(F_{j,2}^{3}(\bar{x}_{j,n_j}(k))) \\
G_{j,2}(\bar{x}_{j,n_j}(k)) & = & g_{j,2}(F_{j,2}^{3}(\bar{x}_{j,n_j}(k)))
\end{array}
$$

After one more step, the first equations in the j-th subsystem of equation (3.1) becomes

$$x_{j,1}(k+n_j) = F_{j,1}(\bar{x}_{j,n_j}(k)) + G_{j,1}(\bar{x}_{j,n_j}(k))x_{j,2}(k+n_j-1) \tag{3.12}$$

where

$$\begin{aligned} F_{j,1}(\bar{x}_{j,n_j}(k)) &= f_{j,1}(f_{j,1}^2(\bar{x}_{j,n_j}(k))) \\ G_{j,1}(\bar{x}_{j,n_j}(k)) &= g_{j,1}(f_{j,1}^2(\bar{x}_{j,n_j}(k))) \end{aligned}$$

Since all the equations from (3.9) to (3.12) are derived from the original system, the j-th subsystem of original system (3.1) is equivalent to

$$\left\{\begin{aligned} x_{j,1}(k+n_j) &= F_{j,1}(\bar{x}_{j,n_j}(k)) + G_{j,1}(\bar{x}_{j,n_j}(k))x_{j,2}(k+n_j-1) \\ &\vdots \\ x_{j,n_j-1}(k+2) &= F_{j,n_j-1}(\bar{x}_{j,n_j}(k)) + G_{j,n_j-1}(\bar{x}_{j,n_j}(k))x_{j,n_j}(k+1) \\ x_{j,n_j}(k+1) &= f_{j,n_j}(X, \bar{u}_{j-1}(k)) + g_{j,n_j}(X)u_j(k) + d_j(k) \\ y_j(k) &= x_{j,1}(k) \end{aligned}\right. \tag{3.13}$$

Definition 3.2 *The form in (3.13) is defined as Sequential Decrease Cascade Form (SDCF).*

For convenience of analysis, define ($1 \leq j \leq n$ and $1 \leq i_j \leq n_j - 1$)

$$F_{j,i_j}(k) \triangleq F_{j,i_j}(\bar{x}_{j,n_j}(k)), \quad G_{j,i_j}(k) \triangleq G_{j,i_j}(\bar{x}_{j,n_j}(k))$$

and

$$f_{j,n_j}(k) \triangleq f_{j,n_j}(X, \bar{u}_{j-1}(k)), \quad g_{j,n_j}(k) \triangleq g_{j,n_j}(X)$$

then system (3.13) can be written as

$$\left\{\begin{aligned} x_{j,1}(k+n_j) &= F_{j,1}(k) + G_{j,1}(k)x_{j,2}(k+n_j-1) \\ &\vdots \\ x_{j,n_j-1}(k+2) &= F_{j,n_j-1}(k) + G_{j,n_j-1}(k)x_{j,n_j}(k+1) \\ x_{j,n_j}(k+1) &= f_{j,n_j}(k) + g_{j,n_j}(k)u_j(k) + d_j(k) \\ y_j(k) &= x_{j,1}(k) \end{aligned}\right. \tag{3.14}$$

Now, we can define the desired virtual controls and the ideal practical controls for each subsystem as follows

$$\left\{\begin{array}{rcl} \alpha^*_{j,2}(k) & \triangleq & x_{j,2}(k+n_j-1) = \frac{1}{G_{j,1}(k)}\left[y_{d_j}(k+n_j) - F_{j,1}(k)\right] \\ \alpha^*_{j,3}(k) & \triangleq & x_{j,3}(k+n_j-2) = \frac{1}{G_{j,2}(k)}\left[\alpha^*_{j,2}(k) - F_{j,2}(k)\right] \\ & \vdots & \\ \alpha^*_{j,n_j}(k) & \triangleq & x_{j,n_j}(k+1) = \frac{1}{G_{j,n_j-1}(k)}\left[\alpha^*_{j,n_j-1}(k) - F_{j,n_j-1}(k)\right] \\ u^*_j(k) & \triangleq & \frac{1}{g_{j,n_j}(k)}\left[\alpha^*_{j,n_j}(k) - f_{j,n_j}(k)\right] \\ y_j(k) & = & x_{j,1}(k) \end{array}\right. \tag{3.15}$$

which can stabilize the system in each step without the causality problem. (3.15) can be further written as

$$\left\{\begin{array}{rcl} \alpha^*_{j,2}(k) & \triangleq & \varphi_{j,1}(\bar{x}_{j,n_j}(k), y_{d_j}(k+n_j)) \\ \alpha^*_{j,3}(k) & \triangleq & \varphi_{j,2}(\bar{x}_{j,n_j}(k), \alpha^*_{j,2}(k)) \\ & \vdots & \\ \alpha^*_{j,n_j}(k) & \triangleq & \varphi_{j,n_j-1}(\bar{x}_{j,n_j}(k), \alpha^*_{j,n_j-1}(k)) \\ u^*_j(k) & \triangleq & \varphi_{j,n_j}(X, \bar{u}_{j-1}(k), \alpha^*_{j,n_j}(k)) \\ y_j(k) & = & x_{j,1}(k) \end{array}\right. \tag{3.16}$$

with the $\varphi_{j,1}(\cdot)$, ..., $\varphi_{j,n_j}(\cdot)$, $(1 \leq j \leq n)$ being nonlinear functions. It is obvious that the desired virtual controls $\alpha^*_{j,2}(k)$, ..., $\alpha^*_{j,n_j}(k)$ and the ideal control $u^*_j(k)$ are all applicable and will drive the output of the j-th subsystem to track $y_{d_j}(k+n_j)$ exactly provided that: (i) the exact system model is known; and (ii) the disturbance $d_j(k) = 0$. However, in practical applications, usually these two conditions cannot be satisfied. In the following, neural networks will be used to emulate the desired virtual controls as well as the desired practical controls when the exact system model is unknown. By using Lyapunov method, the closed-loop system is also shown to be SGUUB even in the presence of unknown bounded disturbances.

Detailed design procedure will be described in Section 3.1.3. It should be noted that, different from the procedure in [115], in this section, embedded backstepping is used to construct the neural network controllers due to the complexity structure of the MIMO system. The procedure can be divided into two steps:

- Firstly, for each subsystem, by using backstepping design, the first $n_j - 1$ $(1 \leq$

$j < n$) equations can be stabilized if the corresponding virtual controls are properly chosen;

- Secondly, by considering the last equations of each subsystem, we can see that the MIMO system is in strict feedback form relative to the control inputs $u_1(k)$, ..., $u_n(k)$. Thus, by embedded using backstepping design, the stability of the whole closed-loop system can be guaranteed.

In the next, a simple example will be given to illustrate the detailed transformation procedure described above. Furthermore, the desired controls are also illustrated, which will be specifically discussed in Section 3.1.3.

Illustrative Example: To illustrate the transformation procedure, let us look at the following simple example ($n = \tau = 2$):

$$\Sigma : \begin{cases} \Sigma_1 : \begin{cases} x_{1,1}(k+1) = x_{1,1}(k) + x_{1,2}(k) \\ x_{1,2}(k+1) = x_{1,1}(k)x_{2,1}(k) + u_1(k) \end{cases} \\ \Sigma_2 : \begin{cases} x_{2,1}(k+1) = x_{2,1}(k) + x_{2,2}(k) \\ x_{2,2}(k+1) = x_{1,2}(k)x_{2,2}(k)u_1^2(k) + u_2(k) \end{cases} \\ y_j(k) = x_{j,1}(k), \quad 1 \leq j \leq 2 \end{cases} \tag{3.17}$$

It can be easily obtained that the SDCF form of system (3.17) is

$$\Sigma : \begin{cases} \Sigma_1 : \begin{cases} x_{1,1}(k+2) = [x_{1,1}(k) + x_{1,2}(k)] + x_{1,2}(k+1) \\ x_{1,2}(k+1) = x_{1,1}(k)x_{2,1}(k) + u_1(k) \end{cases} \\ \Sigma_2 : \begin{cases} x_{2,1}(k+2) = [x_{2,1}(k) + x_{2,2}(k)] + x_{2,2}(k+1) \\ x_{2,2}(k+1) = x_{1,2}(k)x_{2,2}(k)u_1^2(k) + u_2(k) \end{cases} \\ y_j(k) = x_{j,1}(k), \quad 1 \leq j \leq 2 \end{cases} \tag{3.18}$$

It can be further written as

$$\Sigma : \begin{cases} \Sigma_1 : \begin{cases} x_{1,1}(k+2) = F_{1,1}(k) + G_{1,1}(k)x_{1,2}(k+1) \\ x_{1,2}(k+1) = f_{1,2}(k) + g_{1,2}(k)u_1(k) \end{cases} \\ \Sigma_2 : \begin{cases} x_{2,1}(k+2) = F_{2,1}(k) + G_{2,1}(k)x_{2,2}(k+1) \\ x_{2,2}(k+1) = f_{2,2}(k) + g_{2,2}(k)u_2(k) \end{cases} \\ y_j(k) = x_{j,1}(k), \quad 1 \leq j \leq 2 \end{cases}$$

with

$$\begin{cases} F_{1,1}(k) &=& x_{1,1}(k)+x_{1,2}(k) \\ G_{1,1}(k) &=& 1 \end{cases} \qquad \begin{cases} f_{1,2}(k) &=& x_{1,1}(k)x_{2,1}(k) \\ g_{1,2}(k) &=& 1 \end{cases}$$

$$\begin{cases} F_{2,1}(k) &=& x_{2,1}(k)+x_{2,2}(k) \\ G_{2,1}(k) &=& 1 \end{cases} \qquad \begin{cases} f_{2,2}(k) &=& x_{1,2}(k)x_{2,2}(k)u_1^2(k) \\ g_{2,2}(k) &=& 1 \end{cases}$$

Assuming the desired trajectory is $y_d(k) = [y_{d_1}(k), y_{d_2}(k)]^T$, therefore, the desired virtual controls and ideal practical controls for system (3.18) can be defined as follows:

$$\begin{cases} \alpha^*_{1,2}(k) &\triangleq& x_{1,2}(k+1) = \frac{1}{G_{1,1}(k)}\left[y_{d_1}(k+2) - F_{1,1}(k)\right] \\ &=& y_{d_1}(k+2) - [x_{1,1}(k) + x_{1,2}(k)] \\ u^*_1(k) &\triangleq& \frac{1}{g_{1,2}(k)}\left[\alpha^*_{1,2}(k) - f_{1,2}(k)\right] \\ &=& \alpha^*_{1,2}(k) - x_{1,1}(k)x_{2,1}(k) \end{cases}$$

$$\begin{cases} \alpha^*_{2,2}(k) &\triangleq& x_{2,2}(k+1) = \frac{1}{G_{2,1}(k)}\left[y_{d_2}(k+2) - F_{2,1}(k)\right] \\ &=& y_{d_2}(k+2) - [x_{2,1}(k) + x_{2,2}(k)] \\ u^*_2(k) &\triangleq& \frac{1}{g_{2,2}(k)}\left[\alpha^*_{2,2}(k) - f_{2,2}(k)\right] \\ &=& \alpha^*_{2,2}(k) - x_{1,2}(k)x_{2,2}(k){u^*}^2_1(k) \end{cases}$$

Assume system initial conditions are: $\alpha^*_{1,2}(0) = u^*_1(0) = 0$, $\alpha^*_{2,2}(0) = u^*_2(0) = 0$, $y_1(0) = y_1(1) = y_1(2) = 0$ and $y_2(0) = y_2(1) = y_2(2) = 0$. The reference trajectory, $y_{d_1}(k)$ and $y_{d_2}(k)$, are shown in Table 3.1. Practical control action starts at time instant $k = 1$. Table 3.1, Figures 3.1 and 3.2 show the system variation from $k = 0$ to $k = 8$.

It can be seen that, for this example, the control action is started from $k = 1$. The exact tracking is achieved at $k = 3$, as what we expected. The exact tracking is achieved in $\tau = 2$ steps.

3.1.3 Controller Design and Stability Analysis

The closed-loop system structure is shown in Figure 3.3. For each subsystem of system (3.1), it can be transformed into the form of (3.14). Therefore, we can construct the controls via embedded using backstepping technique without causality contradiction.

k	0	1	2	3	4	5	6	7	8
$\alpha^*_{1,2}(k)$	$\underline{0}$	0.2	0.1	-0.2	-0.2	0.1	0.1	0.1	-0.3
$u^*_1(k)$	$\underline{0}$	0.2	0.1	-0.24	-0.2	0.12	0.09	0.1	-0.3
$y_1(k)$	$\underline{0}$	$\underline{0}$	$\underline{0}$	**0.2**	**0.3**	**0.1**	**-0.1**	**0**	**0.1**
$y_{d_1}(k)$	-0.2	-0.1	0.1	**0.2**	**0.3**	**0.1**	**-0.1**	**0**	**0.1**
$\alpha^*_{2,2}(k)$	$\underline{0}$	0.2	-0.2	-0.2	0.1	0.2	-0.1	-0.1	-0.1
$u^*_2(k)$	$\underline{0}$	0.2	-0.2004	-0.1988	0.0984	0.2003	-0.1002	-0.0999	-0.0991
$y_2(k)$	$\underline{0}$	$\underline{0}$	$\underline{0}$	**0.2**	**0**	**-0.2**	**-0.1**	**0.1**	**0**
$y_{d_2}(k)$	-0.1	0.1	0.3	**0.2**	**0**	**-0.2**	**-0.1**	**0.1**	**0**

§The numbers with underscores represent system initial conditions. The numbers in **bold** indicate that exact tracking is obtained.

Table 3.1: A Simple Example - System Variation

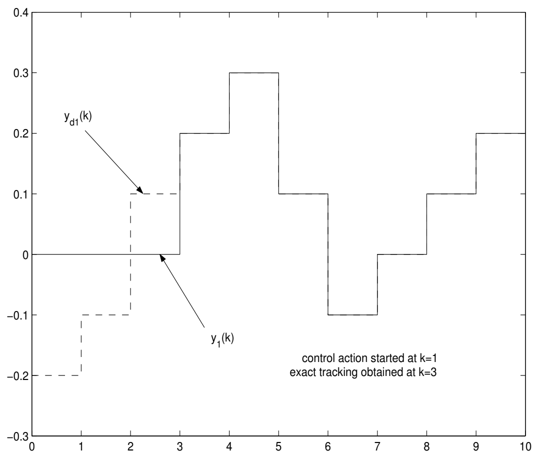

Figure 3.1: Example: y_1 and y_{d_1}

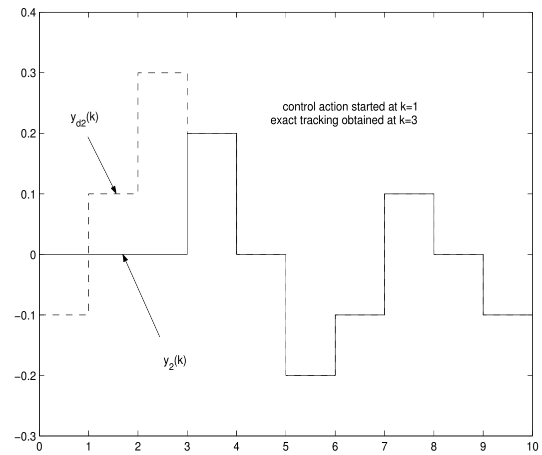

Figure 3.2: Example: y_2 and y_{d_2}

Choosing the practical virtual controls and practical controls as follows:

$$\begin{aligned} \alpha_{j,i_j}(k) &= \hat{W}^T_{j,i_j-1} S_{j,i_j-1}(z_{j,i_j-1}(k)), \quad i_j = 2, \ldots, n_j \\ u_j(k) &= \hat{W}^T_{j,n_j}(k) S_{j,n_j}(z_{j,n_j}(k)) \end{aligned} \tag{3.19}$$

with

$$\begin{aligned} z_{j,1}(k) &= [\bar{x}^T_{j,n_j}(k), y_{d_j}(k+n_j)]^T \\ z_{j,i_j}(k) &= [\bar{x}^T_{j,n_j}(k), \alpha_{j,i_j}(k)]^T, \quad i_j = 2, \ldots, n_j - 1 \\ z_{j,n_j}(k) &= [X, \bar{u}_{j-1}(k), \alpha_{j,n_j}(k)]^T \end{aligned}$$

where $\hat{W}_{j,i_j}$ denotes the estimation of ideal constant W^*_{j,i_j} weight ($1 \le j \le n$, $1 \le i_j \le n_j$), which will be specifically discussed in the proof of Theorem 3.2, and $S_{j,i_j}(\cdot)$ denotes the hyperbolic tangent function.

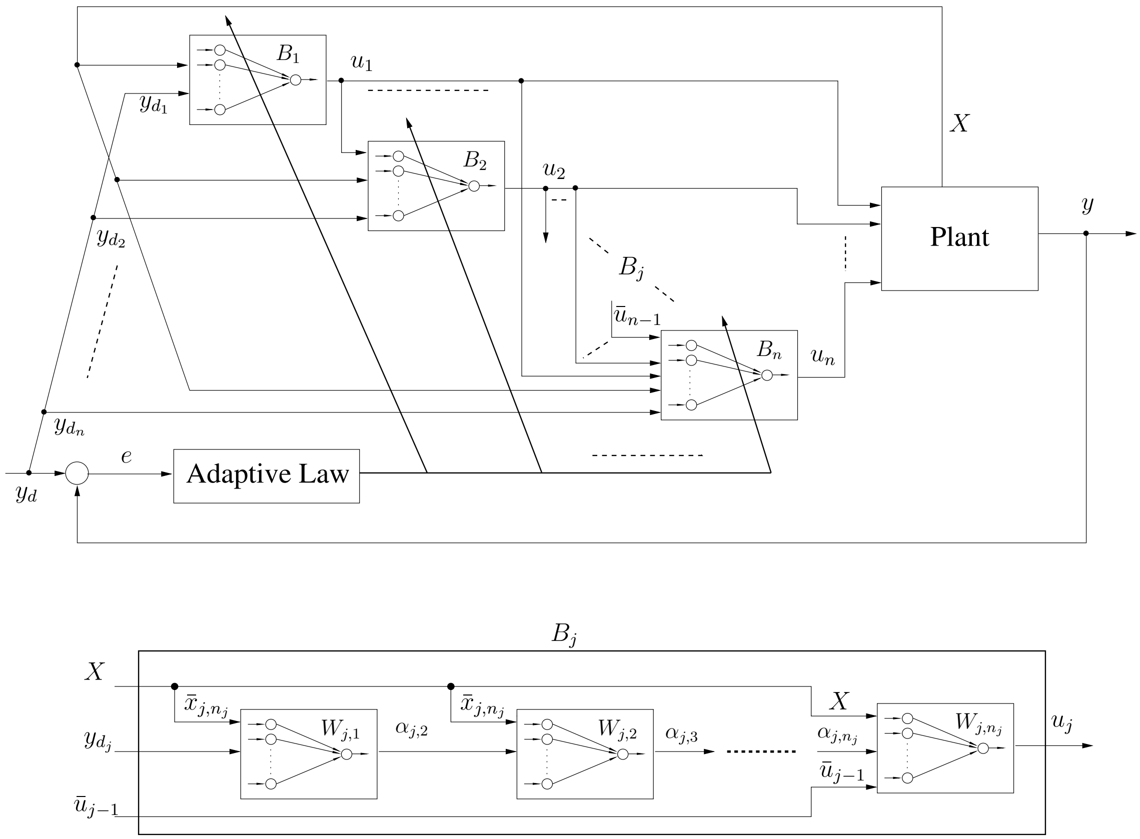

Figure 3.3: State Feedback Control - Control System Structure

The corresponding weights updating laws are chosen as

$$\begin{aligned} \hat{W}_{j,i_j}(k+1) &= \hat{W}_{j,i_j}(k_{i_j}) - \Gamma_{j,i_j}\Big[S(z_{j,i_j}(k_{i_j}))e_{j,i_j}(k+1) + \sigma_{j,i_j}\hat{W}_{j,i_j}(k_{i_j})\Big] \\ k_{i_j} &= k - n_j + i_j, \quad i_j = 1, 2, \ldots, n_j \end{aligned} \tag{3.20}$$

where $\Gamma_{j,i_j} = \gamma_{j,i_j} I > 0$ is diagonal adaptation gain matrix, $\gamma_{j,i_j} > 0$, $\sigma_{j,i_j} > 0$ are positive constants and $0 < \gamma_{j,i_j}\sigma_{j,i_j} < 1$. The error vector is defined as $e_j(k) = [e_{j,1}(k)$, $e_{j,2}(k)$, ... , $e_{j,i_j}(k)$, ..., $e_{j,n_j}(k)]^T$ with $e_{j,i_j}(k)$ denotes the error of each step defined as follows:

$$\begin{aligned} e_{j,1}(k) &= x_{j,1}(k) - y_{d_1}(k) \\ e_{j,2}(k) &= x_{j,2}(k) - \alpha_{j,2}(k - n_j + 1) \\ &\vdots \\ e_{j,n_j}(k) &= x_{j,n_j}(k) - \alpha_{j,n_j}(k-1) \end{aligned}$$

It should be noted that, in the neural network weights update, σ-modification [62] is used to improve the robustness of the proposed control scheme.

The stability of the closed-loop system is summarized in Theorem 3.2.

Theorem 3.2 *Consider the closed-loop nonlinear MIMO system consists of system (3.1), control (3.19) and adaptive law (3.20), it is semi-globally uniformly ultimately bounded, and has an equilibrium at* $[e_{1,1}(k), \ldots, e_{n,1}(k)]^T = 0$ *provided that the design parameters are properly chosen. This guarantees that all the signals include the states* $X(k)$*, the control input* $u(k)$ *and NN weight estimates* $\hat{W}_{j,i_j}(k)$ $(j = 1, \ldots, n,\ i_j = 1, \ldots, n_j)$*, are all bounded, subsequently,*

$$\lim_{k\to\infty} \|y(k) - y_d(k)\| \le \varepsilon$$

where ε *is a positive number.*

Proof: The prove procedure is as follows:

1. For the j-th $(1 \le j \le n)$ subsystem, use backstepping technique to proof its stability up to step $n_j - 1$, i.e., to guarantee the UUB stability for the first $n_j - 1$ equations;

2. For the last equations in each subsystem, noting that the practical control inputs are in strict feedback form, by embedded using backstepping design technique, the closed-loop system stability can be guaranteed.

At time instant k, assume that $\bar{x}_{j,n_j}(k) \in \Omega$, then we should prove that $\bar{x}_{j,n_j}(k+1) \in \Omega$ and $u_j(k)$ is bounded by backstepping. Before proceeding, let $k_i = k - n_j + i, i = 1, 2, \ldots, n_j - 1$ for the convenience of description.

<u>**Step 1:**</u> Considering the tracking error of the j-th subsystem $(1 \le j \le n)$, $e_{j,1}(k) = x_{j,1}(k) - y_{d_j}(k)$, noting the first equation in (3.14), we can obtain

$$\begin{aligned} e_{j,1}(k+n_j) &= x_{j,1}(k+n_j) - y_{d_j}(k+n_j) \\ &= F_{j,1}(k) + G_{j,1}(k) x_{j,2}(k+n_j-1) - y_{d_j}(k+n_j) \end{aligned} \tag{3.21}$$

Considering $x_{j,2}(k+n_j-1)$ as the fictitious control for (3.21), it is obviously that $e_{j,1}(k+n_j)=0$ if we let

$$\begin{aligned} x_{j,2}(k+n_j-1) &= \alpha^*_{j,2}(k) \\ &= -\frac{1}{G_{j,1}(k)}[F_{j,1}(k)-y_{d_j}(k+n_j)] \end{aligned} \tag{3.22}$$

Since $F_{j,1}(k)$ and $G_{j,1}(k)$ are unknown, they are not available for constructing the fictitious control $\alpha^*_{j,2}(k)$. However, $F_{j,1}(k)$ and $G_{j,1}(k)$ are function of system state $\bar{x}_{j,n_j}(k)$, therefore we can use high order neural networks (HONNs) to approximate $\alpha^*_{j,2}(k)$ as follows

$$\begin{aligned} \alpha^*_{j,2}(k) &= W^{*T}_{j,1}S_{j,1}(z_{j,1}(k))+\epsilon_{z_{j,1}}(z_{j,1}(k)) \\ z_{j,1}(k) &= [\bar{x}^T_{j,n_j}(k), y_{d_j}(k+n)]^T \in \Omega_{z_{j_1}} \subset R^{n_j+1} \end{aligned} \tag{3.23}$$

Letting $\hat{W}_{j,1}$ be the estimate of $W^*_{j,1}$, the practical virtual control, $\alpha_{j,2}(k)$, is chosen as follows

$$x_{j,2}(k+n_j-1)=\alpha_{j,2}(k)=\hat{W}^T_{j,1}(k)S_{j,1}(z_{j,1}(k)) \tag{3.24}$$

and the robust updating algorithm for NN weight is chosen as

$$\hat{W}_{j,1}(k+1)=\hat{W}_{j,1}(k_1)-\Gamma_{j,1}\Big[S_{j,1}(z_{j,1}(k_1))e_{j,1}(k+1)+\sigma_{j,1}\hat{W}_{j,1}(k_1)\Big] \tag{3.25}$$

Substituting fictitious control (3.24) into (3.21), the error equation (3.21) is re-written as

$$e_{j,1}(k+n_j)=F_{j,1}(k)-y_{d_j}(k+n_j)+G_{j,1}(k)\hat{W}^T_{j,1}(k)S_{j,1}(z_{j,1}(k)) \tag{3.26}$$

Adding and subtracting $G_{j,1}(k)\alpha^*_{j,2}(k)$ to the right side of (3.26) and noting (3.23), we have

$$\begin{aligned} e_{j,1}(k+n_j) = {} & F_{j,1}(k)-y_{d_1}(k+n)+G_{j,1}(k)[\hat{W}^T_{j,1}(k)S_{j,1}(z_{j,1}(k))-{W^*_{j,1}}^TS_{j,1}(z_{j,1}(k)) \\ & -\epsilon_{z_{j,1}}(z_{j,1}(k))]+G_{j,1}(k)\alpha^*_{j,2}(k), \quad \forall z_{j,1}(k)\in\Omega_{z_{j_1}} \end{aligned} \tag{3.27}$$

Substituting (3.22) into (3.27), we can obtain

$$e_{j,1}(k+n_j)=G_{j,1}(k)[\tilde{W}^T_{j,1}(k)S_{j,1}(z_{j,1}(k))-\epsilon_{z_{j,1}}] \tag{3.28}$$

Choose the Lyapunov function candidate

$$V_{j,1}(k) = \frac{1}{\bar{g}_{j,1}} e_{j,1}^2(k) + \sum_{p=0}^{n_j-1} \tilde{W}_{j,1}^T(k_1+p)\Gamma_1^{-1}\tilde{W}_{j,1}(k_1+p) \tag{3.29}$$

where $k_1 = k - n_j + 1$.

Noting the fact that $\tilde{W}_{j,1}^T(k_1)S_{j,1}(z_{j,1}(k_1)) = \frac{e_{j,1}(k+1)}{G_{j,1}(k_1)} + \epsilon_{z_{j,1}}$, the first difference of (3.29) along (3.25) and (3.28) is given by

$$\begin{aligned}
\Delta V_{j,1} &= \frac{1}{\bar{g}_{j,1}}[e_{j,1}^2(k+1) - e_{j,1}^2(k)] + \tilde{W}_{j,1}^T(k+1)\Gamma_{j,1}^{-1}\tilde{W}_{j,1}(k+1) - \tilde{W}_{j,1}^T(k_1)\Gamma_{j,1}^{-1}\tilde{W}_{j,1}(k_1) \\
&= \frac{1}{\bar{g}_{j,1}}[e_{j,1}^2(k+1) - e_{j,1}^2(k)] - 2\tilde{W}_{j,1}^T(k_1)\Big[S_{j,1}(z_{j,1}(k_1))e_{j,1}(k+1) + \sigma_{j,1}\hat{W}_{j,1}(k_1)\Big] \\
&\quad + \Big[S_{j,1}(z_{j,1}(k_1))e_{j,1}(k+1) + \sigma_{j,1}\hat{W}_{j,1}(k_1)\Big]^T \Gamma_{j,1}\Big[S_{j,1}(z_{j,1}(k_1))e_{j,1}(k+1) \\
&\quad + \sigma_{j,1}\hat{W}_{j,1}(k_1)\Big] \\
&= \frac{1}{\bar{g}_{j,1}}[e_{j,1}^2(k+1) - e_{j,1}^2(k)] - 2\tilde{W}_{j,1}^T(k_1)S_{j,1}(z_{j,1}(k_1))e_{j,1}(k+1) \\
&\quad - 2\sigma_{j,1}\tilde{W}_{j,1}^T(k_1)\hat{W}_{j,1}(k_1) + S_{j,1}^T(z_{j,1}(k_1))\Gamma_{j,1}S_{j,1}(z_{j,1}(k_1))e_{j,1}^2(k+1) \\
&\quad + 2\sigma_{j,1}\hat{W}_{j,1}^T(k_1)\Gamma_{j,1}S_{j,1}(z_{j,1}(k_1))e_{j,1}(k+1) + \sigma_{j,1}^2\hat{W}_{j,1}^T(k_1)\Gamma_{j,1}\hat{W}_{j,1}(k_1) \\
&\le -\frac{1}{\bar{g}_{j,1}}e_{j,1}^2(k+1) - \frac{1}{\bar{g}_{j,1}}e_{j,1}^2(k) - 2\epsilon_{z_{j,1}}e_{j,1}(k+1) - 2\sigma_{j,1}\tilde{W}_{j,1}^T(k_1)\hat{W}_{j,1}(k_1) \\
&\quad + S_{j,1}^T(z_{j,1}(k_1))\Gamma_{j,1}S_{j,1}(z_{j,1}(k_1))e_{j,1}^2(k+1) \\
&\quad + 2\sigma_{j,1}\hat{W}_{j,1}^T(k_1)\Gamma_{j,1}S_{j,1}(z_{j,1}(k_1))e_{j,1}(k+1) + \sigma_{j,1}^2\hat{W}_{j,1}^T(k_1)\Gamma_{j,1}\hat{W}_{j,1}(k_1)
\end{aligned}$$

Using the facts that

$$S_{j,1}^T(z_{j,1}(k_1))S_{j,1}(z_{j,1}(k_1)) < l_{j,1}$$

$$S_{j,1}^T(z_{j,1}(k_1))\Gamma_{j,1}S_{j,1}(z_{j,1}(k_1)) \le \bar{\gamma}_{j,1}S_{j,1}^T(z_{j,1}(k_1))S_{j,1}(z_{j,1}(k_1)) \le \bar{\gamma}_{j,1}l_{j,1}$$

$$2\epsilon_{z_{j,1}}e_{j,1}(k+1) \le \frac{\bar{\gamma}_{j,1}e_{j,1}^2(k+1)}{\bar{g}_{j,1}} + \frac{\bar{g}_{j,1}\epsilon_{z_{j,1}}^2}{\bar{\gamma}_{j,1}}$$

$$2\sigma_{j,1}\hat{W}_{j,1}^T(k_1)\Gamma_{j,1}S_{j,1}(z_{j,1}(k_1))e_{j,1}(k+1) \le \frac{\bar{\gamma}_{j,1}l_{j,1}e_{j,1}^2(k+1)}{\bar{g}_{j,1}} + \bar{g}_{j,1}\sigma_{j,1}^2\bar{\gamma}_{j,1}\|\hat{W}_{j,1}\|^2$$

$$2\tilde{W}_{j,1}^T(k_1)\hat{W}_{j,1}(k_1) = \|\tilde{W}_{j,1}(k_1)\|^2 + \|\hat{W}_{j,1}(k_1)\|^2 - \|W_{j,1}^*\|^2$$

we obtain

$$\Delta V_{j,1} \leq -\frac{\rho_{j,1}}{\bar{g}_{j,1}}e_{j,1}^2(k+1) - \frac{1}{\bar{g}_{j,1}}e_{j,1}^2(k) - \sigma_{j,1}(1-\sigma_{j,1}\bar{\gamma}_{j,1} - \bar{g}_{j,1}\sigma_{j,1}\bar{\gamma}_{j,1})\|\hat{W}_{j,1}(k_1)\|^2 + \beta_{j,1}$$

where

$$\rho_{j,1} = 1 - \bar{\gamma}_{j,1} - \bar{\gamma}_{j,1}l_{j,1} - \bar{g}_{j,1}\bar{\gamma}_{j,1}l_{j,1}, \qquad \beta_{j,1} = \frac{\bar{g}_{j,1}\epsilon_{z_{j,1}}^2}{\bar{\gamma}_{j,1}} + \sigma_{j,1}\|W_{j,1}^*\|^2$$

If we choose the design parameters as follows

$$\bar{\gamma}_{j,1} < \frac{1}{1 + l_{j,1} + \bar{g}_{j,1}l_{j,1}}, \qquad \sigma_{j,1} < \frac{1}{(1+\bar{g}_{j,1})\bar{\gamma}_{j,1}} \tag{3.30}$$

then $\Delta V_{j,1} \leq 0$ once the error $|e_{j,1}(k)|$ is larger than $\sqrt{\bar{g}_{j,1}\beta_{j,1}}$. This implies the boundedness of $V_{j,1}(k)$ for all $k \geq 0$, which leads to the boundedness of $e_{j,1}(k)$ because $V_{j,1}(k) = V_{j,1}(0) + \sum_{p=0}^{k} \Delta V_{j,1}(p) < \infty$. Furthermore, the tracking error $e_{j,1}(k)$ will asymptotically converge to the compact set denoted by $\Omega_{j,1} \subset R$, where $\Omega_{j,1} \triangleq \{\chi \big| |\chi| \leq \sqrt{\bar{g}_{j,1}\beta_{j,1}}\}$.

The adaptation dynamics (3.25) can be written as

$$\begin{aligned}\tilde{W}_{j,1}(k+1) &= (I - \Gamma_{j,1}\sigma_{j,1})\tilde{W}_{j,1}(k_1) - \Gamma_{j,1}[S_{j,1}(z_{j,1}(k_1))e_{j,1}(k+1) + \sigma_{j,1}W_{j,1}^*] \\ &= A_{j,1}(k)\tilde{W}_{j,1}(k_1) - \Gamma_{j,1}[S_{j,1}(z_{j,1}(k_1))e_{j,1}(k+1) + \sigma_{j,1}W_{j,1}^*]\end{aligned}$$

Because $\gamma_{j,1} > 0$, $\sigma_{j,1} > 0$ and $0 < \sigma_{j,1}\gamma_{j,1} < 1$, we know that the transition matrix of $A_{j,1}(k)$ always satisfies $\|\Phi(k_1, k_0)\| < 1$. Furthermore, noting $S_{j,1}(z_{j,1}(k_1))$, $e_{j,1}(k+1)$ and $\sigma_{j,1}W_{j,1}^*$ are all bounded, by applying Lemma A.1, $\tilde{W}_{j,1}(k)$ is bounded in a compact set denoted by $\Omega_{w_{j,1}}$, and hence the boundedness of $\hat{W}_{j,1}(k)$ is assured.

<u>**Step 2:**</u> As defined before, $e_{j,2}(k) = x_{j,2}(k) - \alpha_{j,2}(k_1)$. Its $(n_j - 1)$th difference is given by

$$\begin{aligned}e_{j,2}(k+n_j-1) &= x_{j,2}(k+n_j-1) - \alpha_{j,2}(k) \\ &= F_{j,2}(k) + G_{j,2}(k)x_{j,3}(k+n_j-2) - \alpha_{j,2}(k)\end{aligned} \tag{3.31}$$

Similarly, consider $x_{j,3}(k+n_j-2)$ as a fictitious control for (3.31). It is obviously that $e_{j,2}(k+n_j-1) = 0$ if we choose

$$x_{j,3}(k+n_j-2) = \alpha_{j,3}^*(k) = -\frac{1}{G_{j,2}(k)}[F_{j,2}(k) - \alpha_{j,2}(k)] \tag{3.32}$$

Accordingly, $\alpha^*_{j,3}(k)$ can be approximated by an ideal high-order neural network

$$\begin{aligned}\alpha^*_{j,3} &= W^{*T}_{j,2} S_{j,2}(z_{j,2}(k)) + \epsilon_{z_{j,2}}(z_{j,2}(k)) \\ z_{j,2}(k) &= [\bar{x}^T_{j,n_j}(k), \alpha_{j,2}(k)]^T \in \Omega_{z_{j,2}} \subset R^{n_j+1}\end{aligned} \tag{3.33}$$

Consider the direct adaptive fictitious controller as

$$x_{j,3}(k+n_j-1) = \alpha_{j,3}(k) = \hat{W}^T_{j,2}(k) S_{j,2}(z_{j,2}(k)) \tag{3.34}$$

and the robust updating algorithm for NN weights as

$$\hat{W}_{j,2}(k+1) = \hat{W}_{j,2}(k_2) - \Gamma_{j,2}\Big[S_{j,2}(z_{j,2}(k_2)) e_{j,2}(k+1) + \sigma_{j,2}\hat{W}_{j,2}(k_2)\Big] \tag{3.35}$$

Following the same procedure in *Step 1*, we obtain the second step error equation

$$e_{j,2}(k+n_j-1) = G_{j,2}(k)[\tilde{W}^T_{j,2}(k) S_{j,2}(z_{j,2}(k)) - \epsilon_{z_{j,2}}] \tag{3.36}$$

Choose the Lyapunov function candidate

$$V_{j,2}(k) = V_{j,1}(k) + \frac{1}{\bar{g}_{j,2}} e^2_{j,2}(k) + \sum_{p=0}^{n_j-2} \tilde{W}^T_{j,2}(k_2+j)\Gamma^{-1}_{j,2}\tilde{W}_{j,2}(k_2+j) \tag{3.37}$$

where $k_2 = k - n_j + 2$. The first difference of (3.37) along (3.35) and (3.36) is given by

$$\begin{aligned}\Delta V_{j,2} \le\ & -\frac{\rho_{j,1}}{\bar{g}_{j,1}} e^2_{j,2}(k+1) - \frac{1}{\bar{g}_{j,1}} e^2_{j,1}(k) - \frac{\rho_{j,2}}{\bar{g}_{j,2}} e^2_{j,2}(k+1) - \frac{1}{\bar{g}_{j,2}} e^2_{j,2}(k) \\ & + \beta_{j,2} - \sigma_{j,2}(1 - \sigma_{j,2}\bar{\gamma}_{j,2} - \bar{g}_{j,2}\sigma_{j,2}\bar{\gamma}_{j,2})\|\hat{W}_{j,2}(k_2)\|^2\end{aligned}$$

where $\rho_{j,1}$ is defined as in *Step 1*, and $\rho_{j,2} = 1 - \bar{\gamma}_{j,2} - \bar{\gamma}_{j,2} l_{j,2} - \bar{g}_{j,2}\bar{\gamma}_{j,2} l_{j,2}$, $\beta_{j,2} = \beta_{j,1} + \frac{\bar{g}_{j,2}\epsilon^2_{z_{j,2}}}{\bar{\gamma}_{j,2}} + \sigma_{j,2}\|W^*_{j,2}\|^2$.

If we choose the design parameters as follows

$$\bar{\gamma}_{j,2} < \frac{1}{1 + l_{j,2} + \bar{g}_{j,2} l_{j,2}}, \qquad \sigma_{j,2} < \frac{1}{(1+\bar{g}_{j,2})\bar{\gamma}_{j,2}} \tag{3.38}$$

then $\Delta V_{j,2} \le 0$ once $|e_{j,1}(k)| > \sqrt{\bar{g}_{j,1}\beta_{j,2}}$ or $|e_{j,2}(k)| > \sqrt{\bar{g}_{j,2}\beta_{j,2}}$.

As explained in *Step 1*, $V_{j,2}(k)$ is bounded for all $k \ge 0$, and the tracking errors $e_{j,1}(k)$ and $e_{j,2}(k)$ are also bounded and will asymptotically converge to the compact set denoted by $\Omega_{j,2} \subset R^2$, where $\Omega_{j,2} \triangleq \{\chi \big| \chi = [\chi_1, \chi_2]^T, |\chi_1| \le \sqrt{\bar{g}_{j,1}\beta_{j,2}},\ |\chi_2| \le \sqrt{\bar{g}_{j,2}\beta_{j,2}}\}$. The boundedness of $\hat{W}_{j,2}(k)$, or equivalently of $\tilde{W}_{j,2}(k)$ can be proved as in *Step 1*.

Step $i(2 < i < n_j)$**:** Following the same procedure as in *Step 2*, for $e_{j,i}(k) = x_{j,i}(k) - \alpha_{j,i}(k_{i-1})$, its $(n_j - i + 1)$th difference is

$$\begin{aligned} e_{j,i}(k + n_j - i + 1) &= x_{j,i}(k + n_j - i + 1) - \alpha_{j,i}(k) \\ &= F_{j,i}(k) + G_{j,i}(k)x_{j,i+1}(k + n_j - i) - \alpha_{j,i}(k) \end{aligned}$$

Similarly, we have the direct adaptive fictitious controller and the robust updating algorithm for NN weights as follows:

$$\begin{aligned} x_{j,i+1}(k + n_j - i) &= \alpha_{j,i+1}(k) = \hat{W}^T_{j,i}(k)S_{j,i}(z_{j,i}(k)) & (3.39) \\ \hat{W}_{j,i}(k+1) &= \hat{W}_{j,i}(k_i) - \Gamma_{j,i}\Big[S_{j,i}(z_{j,i}(k_i))e_{j,i}(k+1) + \sigma_{j,i}\hat{W}_{j,i}(k_i)\Big] \\ z_{j,i}(k) &= [\bar{x}^T_{j,n_j}(k), \alpha_{j,i}(k)]^T \in \Omega_{z_{j,i}} \subset R^{n_j+1} & (3.40) \end{aligned}$$

Accordingly, we obtain the ith error equation

$$e_{j,i}(k + n_j - i + 1) = G_{j,i}(k)[\tilde{W}^T_{j,i}(k)S_{j,i}(z_{j,i}(k)) - \epsilon_{z_{j,i}}] \tag{3.41}$$

Choose the Lyapunov function candidate

$$V_{j,i}(k) = V_{j,i-1}(k) + \frac{1}{\bar{g}_{j,i}}e^2_{j,i}(k) + \sum_{p=0}^{n_j-i}\tilde{W}^T_{j,i}(k_i + p)\Gamma^{-1}_{j,i}\tilde{W}_{j,i}(k_i + p) \tag{3.42}$$

where $k_i = k - n_j + i$. The first difference of (3.42) along (3.40) and (3.41) is given

$$\begin{aligned} \Delta V_{j,i} \le\ & -\sum_{p=1}^{i}\frac{\rho_{j,p}}{\bar{g}_{j,p}}e^2_{j,p}(k+1) - \sum_{p=1}^{i}\frac{1}{\bar{g}_{j,p}}e^2_{j,p}(k) + \beta_{j,i} \\ & -\sigma_{j,i}(1 - \sigma_{j,i}\bar{\gamma}_{j,i} - \bar{g}_{j,i}\sigma_{j,i}\bar{\gamma}_{j,i})\|\hat{W}_{j,i}(k_i)\|^2 \end{aligned}$$

where $\rho_{j,p}, p = 1, 2, \ldots, i-1$, are defined in previous $(i-1)$ steps, $\rho_{j,i} = 1 - \bar{\gamma}_{j,i} - \bar{\gamma}_{j,i}l_{j,i} - \bar{g}_{j,i}\bar{\gamma}_{j,i}l_{j,i}$ and $\beta_{j,i} = \beta_{j,i-1} + \frac{\bar{g}_{j,i}\epsilon^2_{z_{j,i}}}{\bar{\gamma}_{j,i}} + \sigma_{j,i}\|W^*_{j,i}\|^2$. If we choose the design parameters as follows

$$\bar{\gamma}_{j,i} < \frac{1}{1 + l_{j,i} + \bar{g}_{j,i}l_{j,i}}, \qquad \sigma_{j,i} < \frac{1}{(1 + \bar{g}_{j,i})\bar{\gamma}_{j,i}} \tag{3.43}$$

then $\Delta V_{j,i} \leq 0$ once any one of the i errors satisfies $|e_{j,p}(k)| > \sqrt{\bar{g}_{j,p}\beta_{j,i}}$, $p = 1, 2, \ldots, i$. This demonstrates that the tracking error $e_{j,1}(k)$, $e_{j,2}(k)$, ..., $e_{j,i}(k)$ are bounded for all $k \geq 0$, and will asymptotically converge to the compact set denoted by $\Omega_{j,i} \subset R^i$, where $\Omega_{j,i} \triangleq \{\chi \big| \chi = [\chi_1, \chi_2, \ldots, \chi_i]^T, \quad \chi_p \leq \sqrt{\bar{g}_{j,p}\beta_{j,i}} \quad p = 1, 2, \ldots, i\}$. The boundedness of $\hat{W}_{j,i}(k)$, or equivalently of $\tilde{W}_{j,i}(k)$ can be proved as in *Step 1*.

Step n_j: By now, we have shown that for the first $n_j - 1$ equations of each subsystem, by suitable chosen the virtual controls' design parameters, they can be stabilized by the virtual controls. Now, by carefully examining the last equations of all the subsystems, we can see that they are in strict feedback form relative to the practical control inputs, $u_1(k)$, $u_2(k)$, ... , $u_n(k)$. This motivates us to use the backstepping design technique again to guarantee the stability of the whole closed-loop system.

Sub-step 1: Considering the first subsystem of system (3.1), according to (3.14), it can be written as

$$\begin{cases} x_{1,1}(k+n_1) &= F_{1,1}(k) + G_{1,1}(k)x_{1,2}(k+n_1-1) \\ & \vdots \\ x_{1,n_1-1}(k+2) &= F_{1,n_1-1}(k) + G_{1,n_1-1}(k)x_{1,n_1}(k+1) \\ x_{1,n_1}(k+1) &= f_{1,n_1}(k) + g_{1,n_1}(k)u_1(k) + d_1(k) \\ y_1(k) &= x_{1,1}(k) \end{cases} \tag{3.44}$$

For the first $n_1 - 1$ equations of (3.44), we have shown their stability can be guaranteed by suitable chosen the virtual control design parameters. Now, let us consider the last equation. The error $e_{1,n_1}(k)$ can be written as $e_{1,n_1}(k) = x_{1,n_1}(k) - \alpha_{1,n_1}(k-1)$, its first difference is given by

$$\begin{aligned} e_{1,n_1}(k+1) &= x_{1,n_1}(k+1) - \alpha_{1,n_1}(k) \\ &= f_{1,n_1}(k) + g_{1,n_1}(k)u_1(k) + d_1(k) - \alpha_{1,n_1}(k) \end{aligned}$$

It is obviously that $e_{1,n_1}(k+1) = 0$ if we choose

$$u_1(k) = u_1^*(k) = -\frac{1}{g_{1,n_1}(k)}[f_{1,n_1}(k) - \alpha_{1,n_1}(k)]$$

and there are no disturbances, i.e. $d_1(k) = 0$. If $d_1(k) \neq 0$, we obtain $e_{1,n_1}(k+1) = d_1(k)$. Thus, exact tracking cannot be obtained though bounded due to the boundedness of the disturbances. Similarly, $u_1^*(k)$ can be approximated by an high-order neural network

$$\begin{aligned} u_1^*(k) &= W_{1,n_1}^{*T} S_{1,n_1}(z_{1,n_1}(k)) + \epsilon_{z_{1,n_1}}(z_{1,n_1}(k)) \\ z_{1,n_1}(k) &= [X, \alpha_{1,n_1}(k)]^T \in \Omega_{z_{1,n_1}} \subset R^{1+\sum_{i=1}^n n_i} \end{aligned} \tag{3.45}$$

Following the same procedure as in <u>*Step i or 2*</u>, we choose the direct adaptive controller and robust updating algorithm for NN weights as

$$u_1(k) = \hat{W}_{1,n_1}^T(k) S_{1,n_1}(z_{1,n_1}(k)) \tag{3.46}$$

$$\hat{W}_{1,n_1}(k+1) = \hat{W}_{1,n_1}(k) - \Gamma_{1,n_1}\Big[S_{1,n_1}(z_{1,n_1}(k)) e_{1,n_1}(k+1) + \sigma_{1,n_1}\hat{W}_{1,n_1}(k)\Big] \tag{3.47}$$

For the n_1-th step error equation

$$\begin{aligned} e_{1,n_1}(k+1) &= g_{1,n_1}(k)[\tilde{W}_{1,n_1}^T(k) S(z_{1,n_1}(k)) - \epsilon_{z_{1,n_1}}] + d_1(k) \\ &= g_{1,n_1}(k)\left[\tilde{W}_{1,n_1}^T(k) S(z_{1,n_1}(k)) - \epsilon_{z_{1,n_1}} + \frac{d_1(k)}{g_{1,n_1}(k)}\right] \\ &= g_{1,n_1}(k)\left[\tilde{W}_{1,n_1}^T(k) S(z_{1,n_1}(k)) - \epsilon'_{z_{1,n_1}}\right] \end{aligned} \tag{3.48}$$

with $\epsilon'_{z_{1,n_1}} = \epsilon_{z_{1,n_1}} - \frac{d_1(k)}{g_{1,n_1}(k)}$. It is obvious that $\epsilon'_{z_{1,n_1}}$ is bounded because of the boundedness of $\epsilon_{z_{1,n_1}}$, $d_1(k)$ and $g_{1,n_1}(k)$. Choosing the Lyapunov function candidate

$$V_{1,n_1}(k) = V_{1,n_1-1}(k) + \frac{1}{\bar{g}_{1,n_1}} e_{1,n_1}^2(k) + \tilde{W}_{1,n_1}^T(k)\Gamma_{1,n_1}^{-1}\tilde{W}_{1,n_1}(k) \tag{3.49}$$

The first difference of (3.49) along (3.47) and (3.48) is given

$$\begin{aligned} \Delta V_{1,n_1} \leq & -\sum_{p=1}^{n_1}\frac{\rho_{1,p}}{\bar{g}_{1,p}} e_{1,p}^2(k+1) - \sum_{p=1}^{n_1}\frac{1}{\bar{g}_{1,p}} e_{1,p}^2(k) \\ & +\beta_{1,n_1} - \sigma_{1,n_1}(1 - \sigma_{1,n_1}\bar{\gamma}_{1,n_1} - \bar{g}_{1,n_1}\sigma_{1,n_1}\bar{\gamma}_{1,n_1})\|\hat{W}_{1,n_1}(k)\|^2 \end{aligned} \tag{3.50}$$

where $\rho_{1,p}, p = 1, 2, \ldots, n_1 - 1$, are defined as previous $(n_1 - 1)$ steps, and $\rho_{1,n_1} = 1 - \bar{\gamma}_{1,n_1} - \bar{\gamma}_{1,n_1} l_{1,n_1} - \bar{g}_{1,n_1}\bar{\gamma}_{1,n_1} l_{1,n_1}$, $\beta_{1,n_1} = \beta_{1,n_1-1} + \frac{\bar{g}_{1,n_1}\epsilon'^2_{z_{1,n_1}}}{\bar{\gamma}_{1,n_1}} + \sigma_{1,n_1}\|W_{1,n_1}^*\|^2$.

If we choose the design parameters as follows

$$\bar{\gamma}_{1,n_1} < \frac{1}{1 + l_{1,n_1} + \bar{g}_{1,n_1} l_{1,n_1}}, \qquad \sigma_{1,n_1} < \frac{1}{(1 + \bar{g}_{1,n_1})\bar{\gamma}_{1,n_1}} \tag{3.51}$$

then $\Delta V_{1,n_1} \leq 0$ once any one of the n_1 errors satisfies $|e_{1,p}(k)| > \sqrt{\bar{g}_{1,p}\beta_{1,n_1}}$, $p = 1, 2, \ldots, n_1$. This demonstrates that the tracking error $e_{1,1}(k)$, $e_{1,2}(k)$, ..., $e_{1,n_1}(k)$ are bounded for all $k \geq 0$, and will asymptotically converge to the compact set denoted by Ω_{1,n_1}, where $\Omega_{1,n_1} \triangleq \{\chi \Big| \chi = [\chi_1, \chi_2, \ldots, \chi_n]^T, |\chi_j| \leq \sqrt{\bar{g}_{1,p}\beta_{1,n_1}} \;\; p = 1, 2, \ldots, n_1\}$ The boundedness of $\hat{W}_{1,n_1}(k)$, or equivalently of $\tilde{W}_{1,n_1}(k)$ can be proved as in *Step 1*.

Based on the procedure above, we can conclude that $\bar{x}_{1,n_1}(k+1) \in \Omega$ and $u_1(k)$ are bounded if $\bar{x}_{1,n_1}(k) \in \Omega$. Finally, if we initialize $\bar{x}_{1,n_1}(0) \in \Omega$, and choose the design parameters according to (3.30), (3.38), (3.43) and (3.51), we know here exists a k^*, such that all errors $e_{1,1}(k)$, $e_{1,2}(k)$, ..., $e_{1,n_1}(k)$ asymptotically converge to Ω_{1,n_1}. Furthermore, by applying Lemma A.1 and following the same procedure in *Step 1*, the boundedness of the weights $\hat{W}_{1,p}$ ($p = 1, 2, \ldots, n_1$) can be proved. Thus, the closed-loop system is SGUUB and $\bar{x}_{1,n_1}(k) \in \Omega$ will hold for all $k > 0$.

Sub-step 2: For $e_{2,n_2}(k) = x_{2,n_2}(k) - \alpha_{2,n_2}(k-1)$, its first difference is given by

$$\begin{aligned} e_{2,n_2}(k+1) &= x_{2,n_2}(k+1) - \alpha_{2,n_2}(k) \\ &= f_{2,n_2}(k) + g_{2,n_2}(k)u_2(k) + d_2(k) - \alpha_{2,n_2}(k) \end{aligned}$$

It is obviously that $e_{2,n_2}(k+1) = 0$ if we choose

$$u_2(k) = u_2^*(k) = -\frac{1}{g_{2,n_2}(k)}[f_{2,n_2}(k) - \alpha_{2,n_2}(k)]$$

and there are no disturbances, i.e. $d_2(k) = 0$. If $d_2(k) \neq 0$, we obtain $e_{2,n_2}(k+1) = d_2(k)$. Thus, exact tracking cannot be obtained though bounded due to the boundedness of the disturbances. Similarly, $u_2^*(k)$ can be approximated by an high-order neural network

$$\begin{aligned} u_2^*(k) &= W_{2,n_2}^{*T} S_{2,n_2}(z_{2,n_2}(k)) + \epsilon_{z_{2,n_2}}(z_{2,n_2}(k)) \\ z_{2,n_2}(k) &= [X, u_1(k), \alpha_{2,n_2}(k)]^T \in \Omega_{z_{2,n_2}} \subset R^{2+\sum_{i=1}^{n} n_i} \end{aligned}$$

Following the same procedure in *Sub-step 1*, in this step, we will design control $u_2(k)$ to stabilize the first two subsystems of system (3.1). Choosing the following Lyapunov candidate

$$V_{2,n_2}(k) = V_{1,n_1}(k) + V_{2,n_2-1}(k) + \frac{1}{\bar{g}_{2,n_2}}e_{2,n_2}^2(k) + \tilde{W}_{2,n_2}^T(k)\Gamma_{2,n_2}^{-1}\tilde{W}_{2,n_2}(k) \quad (3.52)$$

By following the same procedure in *Sub-step 1*, we can obtain (for clarity of presentation, detail procedure is omitted here) that the first difference of $V_{2,n_2}(k)$ is as follows

$$\begin{aligned}\Delta V_{2,n_2}(k) \leq & -\sum_{p=1}^{n_1}\frac{\rho_{1,p}}{\bar{g}_{1,p}}e_{1,p}^2(k+1)-\sum_{p=1}^{n_1}\frac{1}{\bar{g}_{1,p}}e_{1,p}^2(k) \\ & +\beta_{1,n_1}-\sigma_{1,n_1}(1-\sigma_{1,n_1}\bar{\gamma}_{1,n_1}-\bar{g}_{1,n_1}\sigma_{1,n_1}\bar{\gamma}_{1,n_1})\|\hat{W}_{1,n_1}(k)\|^2 \\ & -\sum_{p=1}^{n_2}\frac{\rho_{2,p}}{\bar{g}_{2,p}}e_{2,p}^2(k+1)-\sum_{p=1}^{n_2}\frac{1}{\bar{g}_{2,p}}e_{2,p}^2(k) \\ & +\beta_{2,n_2}-\sigma_{2,n_2}(1-\sigma_{2,n_2}\bar{\gamma}_{2,n_2}-\bar{g}_{2,n_2}\sigma_{2,n_2}\bar{\gamma}_{2,n_2})\|\hat{W}_{2,n_2}(k)\|^2 \quad (3.53)\end{aligned}$$

where $\rho_{2,n_2} = 1-\bar{\gamma}_{2,n_2}-\bar{\gamma}_{2,n_2}l_{2,n_2}-\bar{g}_{2,n_2}\bar{\gamma}_{2,n_2}l_{2,n_2}$ and $\beta_{2,n_2} = \beta_{2,n_2-1}+\frac{\bar{g}_{2,n_2}\epsilon'^2_{z_{2,n_2}}}{\bar{\gamma}_{2,n_2}}+\sigma_{2,n_2}\|W^*_{2,n_2}\|^2$. By noting (3.51) and choosing $\bar{\gamma}_{2,n_2}$ and σ_{2,n_2} as follows

$$\bar{\gamma}_{2,n_2} < \frac{1}{1+l_{2,n_2}+\bar{g}_{2,n_2}l_{2,n_2}}, \qquad \sigma_{2,n_2} < \frac{1}{(1+\bar{g}_{2,n_2})\bar{\gamma}_{2,n_2}} \quad (3.54)$$

we obtain

$$\Delta V_{2,n_2}(k) \leq -\sum_{p=1}^{n_1}\frac{1}{\bar{g}_{1,p}}e_{1,p}^2(k)-\sum_{p=1}^{n_2}\frac{1}{\bar{g}_{2,p}}e_{2,p}^2(k)+\beta_{1,n_1}+\beta_{2,n_2} \quad (3.55)$$

It is obvious that for the first two subsystems of system (3.1), $\Delta V_{2,n_2}(k) \leq 0$ once

$$e_{1,p}^2 > \bar{g}_{1,p}(\beta_{1,n_1}+\beta_{2,n_2}), \quad p = 1,\ldots,n_1$$

or

$$e_{2,p}^2 > \bar{g}_{2,p}(\beta_{1,n_1}+\beta_{2,n_2}), \quad p = 1,\ldots,n_2$$

It indicates that the errors $e_{1,p}^2$ $(p = 1,\ldots,n_1)$ and $e_{2,p}^2$ $(p = 1,\ldots,n_2)$ are all bounded in a compact set.

Sub-step j $(2 < j < n)$: For $e_{j,n_j}(k) = x_{j,n_j}(k)-\alpha_{j,n_j}(k-1)$, its first difference is given by

$$\begin{aligned}e_{j,n_j}(k+1) &= x_{j,n_j}(k+1)-\alpha_{j,n_j}(k) \\ &= f_{j,n_j}(k)+g_{j,n_j}(k)u_j(k)+d_j(k)-\alpha_{j,n_j}(k) \quad (3.56)\end{aligned}$$

It is obviously that $e_{j,n_j}(k+1) = 0$ if we choose

$$u_j(k) = u_j^*(k) = -\frac{1}{g_{j,n_j}(k)}[f_{j,n_j}(k)-\alpha_{j,n_j}(k)] \quad (3.57)$$

and there are no disturbances, i.e. $d_j(k) = 0$. If $d_j(k) \neq 0$, we obtain $e_{j,n_j}(k+1) = d_j(k)$. Thus, exact tracking cannot be obtained though bounded due to the boundedness of the disturbances. Similarly, $u_j^*(k)$ can be approximated by an high-order neural network

$$\begin{aligned} u_j^*(k) &= W_{j,n_j}^{*T} S_{j,n_j}(z_{j,n_j}(k)) + \epsilon_{z_{j,n_j}}(z_{j,n_j}(k)) \\ z_{j,n_j}(k) &= [X, \bar{u}_{j-1}(k), \alpha_{j,n_j}(k)]^T \in \Omega_{z_{j,n_j}} \subset R^{j+\sum_{i=1}^{n} n_i} \end{aligned} \tag{3.58}$$

Following the same procedure as in <u>*Sub-Step 1 or 2*</u>, we choose the direct adaptive controller and robust updating algorithm for NN weights as

$$u_j(k) = \hat{W}_{j,n_j}^T(k) S_{j,n_j}(z_{j,n_j}(k)) \tag{3.59}$$

$$\hat{W}_{j,n_j}(k+1) = \hat{W}_{j,n_j}(k) - \Gamma_{j,n_j}\left[S_{j,n_j}(z_{j,n_j}(k)) e_{j,n_j}(k+1) + \sigma_{j,n_j}\hat{W}_{j,n_j}(k)\right] \tag{3.60}$$

For the n_j-th step error equation

$$\begin{aligned} e_{j,n_j}(k+1) &= g_{j,n_j}(k)[\tilde{W}_{j,n_j}^T(k) S(z_{j,n_j}(k)) - \epsilon_{z_{j,n_j}}] + d_j(k) \\ &= g_{j,n_j}(k)\left[\tilde{W}_{j,n_j}^T(k) S(z_{j,n_j}(k)) - \epsilon_{z_{j,n_j}} + \frac{d_j(k)}{g_{j,n_j}(k)}\right] \\ &= g_{j,n_j}(k)\left[\tilde{W}_{j,n_j}^T(k) S(z_{j,n_j}(k)) - \epsilon'_{z_{j,n_j}}\right] \end{aligned} \tag{3.61}$$

with $\epsilon'_{z_{j,n_j}} = \epsilon_{z_{j,n_j}} - \frac{d_j(k)}{g_{j,n_j}(k)}$. It is obvious that $\epsilon'_{z_{j,n_j}}$ is bounded because of the boundedness of $\epsilon_{z_{j,n_j}}$, $d_j(k)$ and $g_{j,n_j}(k)$. Choosing the Lyapunov function candidate

$$V_{j,n_j}(k) = \sum_{p=1}^{j-1} V_{p,n_p}(k) + V_{j,n_j-1}(k) + \frac{1}{\bar{g}_{j,n_j}} e_{j,n_j}^2(k) + \tilde{W}_{j,n_j}^T(k)\Gamma_{j,n_j}^{-1}\tilde{W}_{j,n_j}(k) \tag{3.62}$$

It is obvious that $V_{j,n_j}(k)$ includes three parts. The first part, $\sum_{p=1}^{j-1} V_{p,n_p}(k)$ corresponds to the summation of the first $j-1$ subsystems' Lyapunov functions, the second part $V_{j,n_j-1}(k)$ corresponds to the first $n_j - 1$ equations of the j-th subsystem and $\frac{1}{\bar{g}_{j,n_j}} e_{j,n_j}^2(k) + \tilde{W}_{j,n_j}^T(k)\Gamma_{j,n_j}^{-1}\tilde{W}_{j,n_j}(k)$ corresponds to the last equation of the j-th subsystem.

The first difference of (3.62) along (3.60) and (3.61) is given

$$\begin{aligned} \Delta V_{j,n_j} \leq\ & \sum_{p=1}^{j-1} \Delta V_{p,n_p}(k) - \sum_{p=1}^{n_j} \frac{\rho_{j,p}}{\bar{g}_{j,p}} e_{j,p}^2(k+1) - \sum_{p=1}^{n_j} \frac{1}{\bar{g}_{j,p}} e_{j,p}^2(k) \\ & + \beta_{j,n_j} - \sigma_{j,n_j}(1 - \sigma_{j,n_j}\bar{\gamma}_{j,n_j} - \bar{g}_{j,n_j}\sigma_{j,n_j}\bar{\gamma}_{j,n_j})\|\hat{W}_{j,n_j}(k)\|^2 \end{aligned} \tag{3.63}$$

where $\rho_{j,p}, p = 1, 2, \ldots, n_j - 1$, are defined as previous $(n_j - 1)$ steps, and $\rho_{j,n_j} = 1 - \bar{\gamma}_{j,n_j} - \bar{\gamma}_{j,n_j} l_{j,n_j} - \bar{g}_{j,n_j}\bar{\gamma}_{j,n_j} l_{j,n_j}$, $\beta_{j,n_j} = \beta_{j,n_j-1} + \frac{\bar{g}_{j,n_j}\epsilon'^2_{z_{j,n_j}}}{\bar{\gamma}_{j,n_j}} + \sigma_{j,n_j}\|W^*_{j,n_j}\|^2$.

Similar to the procedure in derivation of inequality (3.55), if we choose the design parameters as follows

$$\bar{\gamma}_{j,n_j} < \frac{1}{1 + l_{j,n_j} + \bar{g}_{j,n_j} l_{j,n_j}}, \qquad \sigma_{j,n_j} < \frac{1}{(1 + \bar{g}_{j,n_j})\bar{\gamma}_{j,n_j}} \tag{3.64}$$

then inequality (3.63) can be further written as

$$\Delta V_{j,n_j}(k) \;\; \leq \;\; -\sum_{p=1}^{n_1} \frac{1}{\bar{g}_{1,p}} e^2_{1,p}(k) - \ldots - \sum_{p=1}^{n_j} \frac{1}{\bar{g}_{j,p}} e^2_{j,p}(k) + \beta_{1,n_1} + \ldots + \beta_{j,n_j}$$

then $\Delta V_{j,n_j} \leq 0$ once any one of the errors

$$e^2_{q,p}(k) > \bar{g}_{q,p}\left(\beta_{1,n_1} + \ldots + \beta_{j,n_j}\right), \quad q = 1, \ldots, j \text{ and } p = 1, \ldots, n_q$$

This demonstrates that the errors $e_{q,p}$ ($q = 1, \ldots, j$, $p = 1, \ldots, n_q$) are bounded for all $k \geq 0$, and will asymptotically converge to the compact set denoted by Ω_{j,n_j}. The boundedness of $\hat{W}_{j,n_j}(k)$, or equivalently of $\tilde{W}_{j,n_j}(k)$ can be proved as in Step 1.

Based on the procedure above, we can conclude that $\bar{x}_{j,n_j}(k+1) \in \Omega$ and $u_j(k)$ are bounded if $\bar{x}_{j,n_j}(k) \in \Omega$. Finally, if we initialize $\bar{x}_{j,n_j}(0) \in \Omega$, and choose the design parameters according to (3.30), (3.38), (3.43) and (3.64), there exists a k^*, such that all errors asymptotically converge to Ω_{j,n_j}, and NN weight errors are all bounded. This implies that the closed-loop system is SGUUB. Then $\bar{x}_{j,n_j}(k) \in \Omega$, $\hat{W}_{j,p}$, $p = 1, 2, \ldots, n_j$ will hold for all $k > 0$.

Sub-step n: Finally, in this step, by combining the Lyapunov functions of each subsystem to give the whole system's Lyapunov function candidate, we can claim that the closed-loop system is SGUUB.

For $e_{n,n_n}(k) = x_{n,n_n}(k) - \alpha_{n,n_n}(k-1)$, its first difference is given by

$$\begin{aligned} e_{n,n_n}(k+1) &= x_{n,n_n}(k+1) - \alpha_{n,n_n}(k) \\ &= f_{n,n_n}(k) + g_{n,n_n}(k)u_n(k) + d_n(k) - \alpha_{n,n_n}(k) \end{aligned}$$

It is obviously that $e_{n,n_n}(k+1) = 0$ if we choose

$$u_n(k) = u^*_n(k) = -\frac{1}{g_{n,n_n}(k)}[f_{n,n_n}(k) - \alpha_{n,n_n}(k)] \tag{3.65}$$

and there are no disturbances, i.e. $d_n(k) = 0$. If $d_n(k) \neq 0$, we obtain $e_{n,n_n}(k+1) = d_n(k)$. Thus, exact tracking cannot be obtained though bounded due to the boundedness of the disturbances. Similarly, $u_n^*(k)$ can be approximated by an high-order neural network

$$\begin{aligned} u_n^*(k) &= W_{n,n_n}^{*T} S_{n,n_n}(z_{n,n_n}(k)) + \epsilon_{z_{n,n_n}}(z_{n,n_n}(k)) \\ z_{n,n_n}(k) &= [X, \bar{u}_{n-1}(k), \alpha_{n,n_n}(k)]^T \in \Omega_{z_{n,n_n}} \subset R^{n+\sum_{i=1}^{n} n_i} \end{aligned}$$

Choosing the direct adaptive controller and robust updating algorithm for NN weights as

$$\begin{aligned} u_n(k) &= \hat{W}_{n,n_n}^T(k) S_{n,n_n}(z_{n,n_n}(k)) \\ \hat{W}_{n,n_n}(k+1) &= \hat{W}_{n,n_n}(k) - \Gamma_{n,n_n}\Big[S_{n,n_n}(z_{n,n_n}(k)) e_{n,n_n}(k+1) + \sigma_{n,n_n}\hat{W}_{n,n_n}(k)\Big] \end{aligned}$$

Considering the following Lyapunov candidate

$$V_{n,n_n}(k) = \sum_{p=1}^{n-1} V_{p,n_p}(k) + V_{n,n_n-1}(k) + \frac{1}{\bar{g}_{n,n_n}} e_{n,n_n}^2(k) + \tilde{W}_{n,n_n}^T(k)\Gamma_{n,n_n}^{-1}\tilde{W}_{n,n_n}(k) \quad (3.66)$$

By following the same procedure in <u>*Sub-step j* $(2 < j < n)$</u>, if the design parameters are suitable chosen as

$$\bar{\gamma}_{n,n_n} < \frac{1}{1 + l_{n,n_n} + \bar{g}_{n,n_n} l_{n,n_n}}, \qquad \sigma_{n,n_n} < \frac{1}{(1+\bar{g}_{n,n_n})\bar{\gamma}_{n,n_n}} \quad (3.67)$$

we have

$$\Delta V_{n,n_n}(k) \;\leq\; -\sum_{p=1}^{n_1} \frac{1}{\bar{g}_{1,p}} e_{1,p}^2(k) - \ldots - \sum_{p=1}^{n_n} \frac{1}{\bar{g}_{n,p}} e_{n,p}^2(k) + \beta_{1,n_1} + \ldots + \beta_{n,n_n}$$

Define $\beta = \sum_{j=1}^{n} \beta_{j,n_j} = \beta_{1,n_1} + \cdots + \beta_{n,n_n}$, we obtain

$$\Delta V_{n,n_n}(k) \leq \sum_{j=1}^{n} \Big\{ -\sum_{i=1}^{n_j} \frac{1}{\bar{g}_{j,i}} e_{j,i}^2(k) \Big\} + \beta \quad (3.68)$$

then $\Delta V(k)_{n,n_n} \leq 0$ once any one of the $\sum_{j=1}^{n} n_j$ errors satisfies $|e_{j,i}(k)| > \sqrt{\bar{g}_{j,i}\beta}$, $j = 1, \ldots, n$ and $i = 1, \ldots, n_j$. This demonstrates that the tracking errors $e_{q,p}$ ($q = 1, \ldots, n$, $p = 1, \ldots, n_q$) are all bounded for all $k \geq 0$, and will asymptotically converge to the compact set denoted by Ω_{n,n_n}, where $\Omega_{n,n_n} \triangleq \{\chi \big| \chi = [\chi_{j,i}], j =$

$1,\dots,n, i=1,\dots,n_j, |\chi_{j,i}| \le \sqrt{\bar{g}_{j,i}\beta}\}$. Now, we can conclude that all the errors are bounded.

We have proved that all the errors $e_{q,p}$ ($q = 1,\dots,n$, $p = 1,\dots,n_q$) are bounded in a compact set, now we should prove that the neural network weights are also bounded. Considering the weights update law in equation (3.20), it can be rewritten as

$$\begin{aligned} \hat{W}_{j,i_j}(k+1) &= (I - \Gamma_{j,i_j}\sigma_{j,i_j})\hat{W}_{j,i_j}(k_{i_j}) - \Gamma_{j,i_j}S(z_{j,i_j}(k_{i_j}))e_{j,i_j}(k+1) \\ &\triangleq A_{j,i_j}\hat{W}_{j,i_j}(k_{i_j}) - \Gamma_{j,i_j}S(z_{j,i_j}(k_{i_j}))e_{j,i_j}(k+1) \\ k_{i_j} &= k - n_j + i_j, \quad i_j = 1,2,\dots,n_j \end{aligned} \tag{3.69}$$

where $A_{j,i_j} = I - \Gamma_{j,i_j}\sigma_{j,i_j}$. Because the eigenvalues of matrix A_{j,i_j} are all in the unit circle, it is easy to obtain that the eigenvalues of the transition matrix of system (3.69) are all in unit circle too. By using Lemma A.1, we concluded that the neural network weights are bounded.

In summary, the closed-loop nonlinear MIMO system consists of system (3.1), controller (3.19) and adaptive law (3.20) is semi-globally uniformly ultimately bounded, and has an equilibrium at $[e_{1,1}(k),\dots,e_{n,1}(k)]^T = 0$ provided that the design parameters are properly chosen. All the signals include the states $X(k)$, the control inputs $u_j(k)$ ($j = 1,\dots,n$), the tracking errors $e_{j,1}(k)$ ($j = 1,\dots,n$) and NN weight estimates $\hat{W}_{j,i_j}(k)$ ($j = 1,\dots,n$, $i_j = 1,\dots,n_j$), are all bounded.

Therefore, for any a priori given (arbitrarily large) bounded set Ω and any a priori given (arbitrarily small) set Ω_0, which contains $(0,0)$ as an interior point, there exist a control u, such that every trajectory of the closed-loop system starting from Ω enters the set Ω_0 in a finite time and remains in it thereafter. That is to say, the whole closed-loop system is SGUUB. ■

Remark 3.3 *Considering the parameter conditions in equation (3.64). It can be seen that faster learning rate (increasing $\bar{\gamma}_{j,n_j}$) requires the neurons number l_{j,n_j} to decrease. Thus, the approximation accuracy will be affected. In practical applications, how to choose the adaptation gain $\bar{\gamma}_{j,n_j}$ and the neurons number l_{j,n_j} is a problem that needs to be dealt with carefully.*

Remark 3.4 *In adaptive nonlinear system control, PE condition is important for parameter convergence and system robustness. However, it is usually very difficult to verify its existence in practical applications [62]. Noticing Appendix A.2, the definition of PE condition in discrete-time system, we can see that to check its existence is not an easy task. In this section, by combing a standard σ-modification term [62] in the weight update laws (3.20). The need of PE condition for weights update is removed.*

Remark 3.5 *In Theorem 3.2, by using the neural network emulator (3.19) and the weight update laws (3.20), through Lyapunov analysis, we can only obtain the boundedness of the closed-loop signals, include the states, the outputs and the neural network weights.*

3.1.4 Simulation

In order to illustrate the effectiveness of the proposed schemes, a simulation example is studied in this section. Considering the following MIMO discrete-time system with triangular form inputs

$$\begin{cases} x_{1,1}(k+1) &=& f_{1,1}(\bar{x}_{1,1}(k)) + g_{1,1}(\bar{x}_{1,1}(k))x_{1,2}(k) \\ x_{1,2}(k+1) &=& f_{1,2}(\bar{x}_{1,2}(k)) + g_{1,2}(\bar{x}_{1,2}(k))u_1(k) + d_1(k) \\ x_{2,1}(k+1) &=& f_{2,1}(\bar{x}_{2,1}(k)) + g_{2,1}(\bar{x}_{2,1}(k))x_{2,2}(k) \\ x_{2,2}(k+1) &=& f_{2,2}(\bar{x}_{2,2}(k), u_1(k)) + g_{2,2}(\bar{x}_{2,2}(k))u_2(k) + d_2(k) \\ y_1(k) &=& x_{1,1}(k) \\ y_2(k) &=& x_{2,1}(k) \end{cases}$$

where

$$\begin{cases} f_{1,1}(\bar{x}_{1,1}(k)) = \frac{x_{1,1}^2(k)}{1+x_{1,1}^2(k)}, & g_{1,1}(\bar{x}_{1,1}(k)) = 0.3 \\ f_{1,2}(\bar{x}_{1,2}(k)) = \frac{x_{1,1}^2(k)}{1+x_{1,2}^2(k)+x_{2,1}^2(k)+x_{2,2}^2(k)}, & g_{1,2}(\bar{x}_{1,2}(k)) = 1 \\ d_1(k) = 0.1\cos(0.05k)\cos(x_{1,1}(k)) \end{cases}$$

$$\begin{cases} f_{2,1}(\bar{x}_{2,1}(k)) = \frac{x_{2,1}^2(k)}{1+x_{2,1}^2(k)}, & g_{2,1}(\bar{x}_{2,1}(k)) = 0.2 \\ f_{2,2}(\bar{x}_{2,2}(k), u_1(k)) = \frac{x_{2,1}^2(k)}{1+x_{1,1}^2+x_{1,2}^2(k)+x_{2,2}^2(k)} u_1^2(k), & g_{2,2}(\bar{x}_{2,2}(k)) = 1 \\ d_2(k) = 0.1\cos(0.05k)\cos(x_{2,1}(k)) & \end{cases}$$

The control objective is to drive the output $y(k) = [y_1(k), y_2(k)]^T$ of the system to follow desired reference signals

$$y_{d_1}(k) = 0.5 + \frac{1}{4}\cos(\frac{\pi T k}{4}) + \frac{1}{4}\sin(\frac{\pi T k}{2})$$

$$y_{d_2}(k) = 0.5 + \frac{1}{4}\sin(\frac{\pi T k}{4}) + \frac{1}{4}\sin(\frac{\pi T k}{2})$$

with $T = 0.01$.

The initial condition for system states is $x_{1,1}(0) = 0$, $x_{1,2}(0) = 0$, $x_{2,1}(0) = 0$ and $x_{2,2}(0) = 0$. The neurons used are $l_{1,1} = 12$, $l_{1,2} = 20$, $l_{2,1} = 12$ and $l_{2,2} = 30$. All the elements of the neural network weights $\hat{W}_{1,1}(0)$, $\hat{W}_{1,2}(0)$, $\hat{W}_{2,1}(0)$ and $\hat{W}_{2,2}(0)$ are initialized to be random numbers between 0.00 and 0.01, and the active functions $S_{1,1}(0)$, $S_{1,2}(0)$, $S_{2,1}(0)$ and $S_{2,2}(0)$ are initialized to be random numbers between 0.00 and 0.02. The initial values of the virtual controls are $\alpha_{1,2}(0) = 0$ and $\alpha_{2,2}(0) = 0$. σ modification gains are $\sigma_{1,1} = \sigma_{1,2} = \sigma_{2,1} = \sigma_{2,2} = 0.01$, and adaptive gain matrices are $\Gamma_{1,1} = \Gamma_{1,2} = \Gamma_{2,1} = 0.025I$ and $\Gamma_{2,2} = 0.010I$.

For clarity, the formulas used in the simulation are listed here. The virtual controls and the practical controls for subsystem Σ_i $(i = 1, 2)$ are as follows:

$$\begin{aligned} \alpha_{i,2}(k) &= \hat{W}_{i,1}(k) S_{i,1}(z_{i,1}(k)), \quad z_{i,1}(k) = [x_{i,1}(k), x_{i,2}(k), y_{d_i}(k+2)]^T \\ u_i(k) &= \hat{W}_{i,2}(k) S_{i,2}(z_{i,2}(k)), \quad z_{i,2}(k) = [x_{1,1}(k), x_{1,2}(k), x_{2,1}(k), x_{2,2}(k), \alpha_{i,2}(k)]^T \end{aligned}$$

The errors' definitions for subsystem Σ_i $(i = 1, 2)$ are:

$$\Sigma_i : e_{i,1}(k) = y_i(k) - y_{d_i}(k), \quad e_{i,2}(k) = x_{i,2}(k) - \alpha_{i,2}(k-1)$$

The weights update law are as follows $(i = 1, 2)$:

$$\Sigma_i : \begin{cases} \hat{W}_{i,1}(k) = \hat{W}_{i,1}(k-2) - \Gamma_{i,1}[S_{i,1}(z_{i,1}(k-2)) e_{i,1}(k) + \sigma_{i,1} W_{i,1}(k-2)] \\ \hat{W}_{i,2}(k) = \hat{W}_{i,2}(k-1) - \Gamma_{i,2}[S_{i,2}(z_{i,2}(k-1)) e_{i,2}(k) + \sigma_{i,2} W_{i,2}(k-1)] \end{cases}$$

Simulation results are shown in Figure 3.7-Figure 3.10. Figure 3.7 and Figure 3.8 show the tracking performances of the first subsystem and the second subsystem respectively. It can be seen that, in the initial period of simulation, the tracking errors are large. Then, as the time increases, the practical outputs converge to the neighborhoods of the desired signals. The control input trajectories $u_1(k) = \hat{W}_{1,2}(k)S_{1,2}(z_{1,2}(k))$ and $u_2(k) = \hat{W}_{2,2}(k)S_{2,2}(z_{2,2}(k))$ are shown in Figure 3.9. Their corresponding neural network weights norms $\|\hat{W}_{1,2}(k)\|$ and $\|\hat{W}_{2,2}(k)\|$ are shown in Figure 3.10. From Figure 3.9 and 3.10, we can see that both the control inputs and their corresponding weights norms are all bounded. The dynamics of the tracking errors are shown in Figure 3.11. It can be seen that the tracking errors are also bounded.

3.2 Output Feedback Control

For MIMO discrete-time systems, some results can be found in [139, 111, 140, 113]. However, all of the works studied affine MIMO systems, i.e., the control inputs appear linearly, which makes feedback linearization method applicable. For non-affine discrete-time MIMO systems, due to the inputs are in non-affine form, feedback linearization method cannot be used. Therefore, how to find the "inverse" control, if there is, is a problem that needs to be investigated. In [112], a new method was proposed for a class of non-affine MIMO NARMAX systems. Firstly, SISO plants were studied, then the results were extended to MIMO cases. By first order Taylor linearization, neural networks were used to construct the inverse model for the linearized systems. Though the proposed method is effective in dealing with non-affine NARMAX models, there are some restrictions: (i) there is no input coupling in the system studied in [112], which avoided one of the major difficulties in MIMO nonlinear system control; and (ii) neural network identification should be carried out in advance in order to make the control implementable if the plant model is unknown. In Section 3.1, the first part of this chapter, state feedback control scheme was investigated for a class of discrete-time nonlinear MIMO system with triangular form inputs and bounded disturbances by using neural networks. Though the method proposed is effective, all the system states are needed in order to construct the stable control.

In this section, we are considering a class of MIMO nonlinear discrete-time systems with triangular form inputs [50]. Each subsystem of the MIMO system is in strict feedback form. Though the j-th input appears linearly in the j-th subsystem, the other $j-1$ inputs appear nonlinearly in the j-th subsystem, which leads to the whole system in non-affine form. Firstly, by coordinate transformation, the system studied is transformed from state space model into input output representation, with each subsystem is in τ-step (τ is the system delay) predictor form and the triangular form inputs remains unchanged. Then, backstepping design is implemented. Neural networks and input output sequences are used to construct the stable control. Comparing with the MIMO non-affine system studied in [112], we can see that: (i) there are complex inputs coupling; and (ii) neural network identification is not needed in this section.

This section is organized as follows. System dynamics as well as some stability notions are proposed in Section 3.2.1. The detailed transformation procedure is shown in Section 3.2.2. Controller design and stability analysis are discussed in Section 3.2.3. In Section 3.2.4, a simulation example is used to illustrate the effectiveness of the proposed scheme.

3.2.1 MIMO System Dynamics

Considering the following discrete-time MIMO system in state space representation

$$\Sigma : \begin{cases} \Sigma_1 : \begin{cases} x_{1,i_1}(k+1) = f_{1,i_1}(\bar{x}_{1,i_1}(k)) + g_{1,i_1}(\bar{x}_{1,i_1}(k))x_{1,i_1+1}(k) \\ \qquad 1 \le i_1 \le \tau - 1 \\ x_{1,\tau}(k+1) = f_{1,\tau}(X(k)) + g_{1,\tau}(X(k))u_1(k) \end{cases} \\ \quad \vdots \\ \Sigma_j : \begin{cases} x_{j,i_j}(k+1) = f_{j,i_j}(\bar{x}_{j,i_j}(k)) + g_{j,i_j}(\bar{x}_{j,i_j}(k))x_{j,i_j+1}(k) \\ \qquad 1 \le i_j \le \tau - 1 \\ x_{j,\tau}(k+1) = f_{j,\tau}(X(k), \bar{u}_{j-1}(k)) + g_{j,\tau}(X(k))u_j(k) \end{cases} \\ \quad \vdots \\ \Sigma_n : \begin{cases} x_{n,i_n}(k+1) = f_{n,i_n}(\bar{x}_{n,i_n}(k)) + g_{n,i_n}(\bar{x}_{n,i_n}(k))x_{n,i_n+1}(k) \\ \qquad 1 \le i_n \le \tau - 1 \\ x_{n,\tau}(k+1) = f_{n,\tau}(X(k), \bar{u}_{n-1}(k)) + g_{n,\tau}(X(k))u_n(k) \end{cases} \\ y_j(k) = x_{j,1}(k), \quad 1 \le j \le n \end{cases} \tag{3.70}$$

where $X(k) = [x_1^T(k), x_2^T(k), \ldots, x_n^T(k)]^T$ with $x_j(k) = [x_{j,1}(k), x_{j,2}(k), \ldots, x_{j,\tau}(k)]^T \in R^\tau$ (τ is the system delay), $u_j(k) \in R^n$ and $y_j(k) \in R^n$ are the state variables, the system inputs and outputs respectively; $\bar{u}_{j-1}(k) = [u_1(k), \cdots, u_{j-1}(k)]$ $(j = 2, \ldots, n)$; $\bar{x}_{j,i_j}(k) = [x_{j,1}(k), \ldots, x_{j,i_j}(k)]^T \in R^{i_j}$ denotes the first i_j states of the j-th subsystem; $f_{j,i_j}(\cdot)$ and $g_{j,i_j}(\cdot)$ are smooth nonlinear functions. Noting that the control inputs of the whole system are in triangular form, then backstepping design technique may be implemented to design stable controls for this class of systems. It is obvious that there are n subsystems in system (3.70), with the length of each subsystem is τ and system (3.70) has n inputs and n outputs.

Before proceed to the next section, the following assumptions are made.

Assumption 3.3 *The sign of $g_{j,i_j}(\cdot)$ ($j = 1, \ldots, n$ and $i_j = 1, \ldots, \tau$), are known and there exist two constants $\underline{g}_{j,i_j}, \bar{g}_{j,i_j} > 0$ such that $\underline{g}_{j,i_j} \le |g_{j,i_j}(\cdot)| \le \bar{g}_{j,i_j}$, $\forall X(k) \in \Omega \subset R^{n \times \tau}$.*

Without losing generality, we shall assume that $g_{j,i_j}(\cdot)$ is positive in this section.

The control objective is to design control input $u(k) = [u_1(k), \ldots, u_n(k)]^T$ to make the system output $y(k) = [y_1(k), \ldots, y_n(k)]^T$ follow a known and bounded trajectory $y_d(k) = [y_{d_1}(k), \ldots, y_{d_n}(k)]^T$. Thus, the following assumption should be made.

Assumption 3.4 *The desired trajectory $y_d(k) \in \Omega_y$, $\forall k > 0$ is smooth and known, where $\Omega_y \triangleq \{\chi \big| \chi = y(k)\}$.*

Remark 3.6 *Different from the triangular form inputs discrete-time MIMO non-linear system studied in [139] and [111], whose inputs can be written into feedback linearizable form*

$$\begin{aligned} \Xi(k+1) &= F(\Xi(k)) + G(\Xi(k))U(k) \\ U(k) &= [u_1(k), \ldots, u_n(k)]^T \end{aligned} \tag{3.71}$$

in this section, the triangular inputs cannot be written in the form of (3.71). Instead it is in the following form

$$\begin{aligned} \Xi(k+1) &= F(\Xi(k), U(k)) \\ U(k) &= [u_1(k), \ldots, u_n(k)]^T \end{aligned} \tag{3.72}$$

It is obvious that feedback linearization method is not applicable to system (3.72). To construct stable controls for this class of system which is not feedback linearizable is more challenging.

3.2.2 System Coordinate Transformation

In this section, the procedure of how to transform system (3.70) from state space description into input output description is illustrated. In general, the transformation procedure can be divided into two phases.

<u>Coordinate Transformation: Phase One</u> Considering the i-th $(1 \leq i \leq n)$ subsystem

of system (3.70), we have

$$\Sigma_i : \begin{cases} x_{i,1}(k+1) &=& f_{i,1}(\bar{x}_{i,1}(k)) + g_{i,1}(\bar{x}_{i,1}(k))x_{i,2}(k) \\ x_{i,2}(k+1) &=& f_{i,2}(\bar{x}_{i,2}(k)) + g_{i,2}(\bar{x}_{i,2}(k))x_{i,3}(k) \\ &\vdots& \\ x_{i,\tau-1}(k+1) &=& f_{i,\tau-1}(\bar{x}_{i,\tau-1}(k)) + g_{i,\tau-1}(\bar{x}_{i,\tau-1}(k))x_{i,\tau}(k) \\ x_{i,\tau}(k+1) &=& f_{i,\tau}(X(k), \bar{u}_{i-1}(k)) + g_{i,\tau}(X(k))u_i(k) \end{cases} \tag{3.73}$$

Define new coordinates ($1 \leq i \leq n$)

$$\xi_i = [\xi_{i,1}, \xi_{i,2}, \ldots, \xi_{i,\tau}]^T \tag{3.74}$$

with each element of ξ_i is defined as follows

$$\begin{cases} \xi_{i,1}(k) &=& x_{i,1}(k) \\ \xi_{i,2}(k) &=& x_{i,1}(k+1) \\ &\vdots& \\ \xi_{i,\tau}(k) &=& x_{i,1}(k+\tau-1) \end{cases} \tag{3.75}$$

Therefore, we know that the original system state $X = [x_1, x_2, \ldots, x_n]^T \in R^{n\times\tau}$ can be transformed into Ξ, with Ξ is defined as

$$\Xi = [\xi_1^T, \xi_2^T, \ldots, \xi_n^T]^T \in R^{n\times\tau} \tag{3.76}$$

Define this mapping as

$$T(X) : X \to \Xi \tag{3.77}$$

In order to guarantee that this transformation is valid, in the following, we should prove that the mapping is diffeomorphism [143, 65].

Considering (3.75), it can be easily obtained that

$$\begin{cases} \xi_{i,1}(k+1) &=& \xi_{i,2}(k) \\ \xi_{i,2}(k+1) &=& \xi_{i,3}(k) \\ &\vdots& \\ \xi_{i,\tau-1}(k+1) &=& \xi_{i,\tau}(k) \\ \xi_{i,\tau}(k+1) &=& x_{i,1}(k+\tau) \end{cases} \tag{3.78}$$

Considering the last equation in (3.78), because

$$
\left\{
\begin{array}{rcl}
x_{i,1}(k+1) & = & f_{i,1}(\bar{x}_{i,1}(k)) + g_{i,1}(\bar{x}_{i,1}(k))x_{i,2}(k) \\
 & \triangleq & p_{i,1}(\bar{x}_{i,1}(k)) + q_{i,1}(\bar{x}_{i,1}(k))x_{i,2}(k) \\
 & \triangleq & \alpha_{i,1}(\bar{x}_{i,2}(k)) \\
x_{i,1}(k+2) & = & f_{i,1}(\bar{x}_{i,1}(k+1)) + g_{i,1}(\bar{x}_{i,1}(k+1))x_{i,2}(k+1) \\
 & = & f_{i,1}(\alpha_{i,1}(\bar{x}_{i,2}(k))) + g_{i,1}(\alpha_{i,1}(\bar{x}_{i,2}(k))) \\
 & & \times\, [f_{i,2}(\bar{x}_{i,2}(k)) + g_{i,2}(\bar{x}_{i,2}(k))x_{i,3}(k)] \\
 & \triangleq & p_{i,2}(\bar{x}_{i,2}(k)) + q_{i,2}(\bar{x}_{i,2}(k))x_{i,3}(k) \\
 & \triangleq & \alpha_{i,2}(\bar{x}_{i,3}(k)) \\
x_{i,1}(k+3) & = & f_{i,1}(\alpha_{i,1}(\bar{x}_{i,2}(k+1))) + g_{i,1}(\alpha_{i,1}(\bar{x}_{i,2}(k+1))) \\
 & & \times\, [f_{i,2}(\bar{x}_{i,2}(k+1)) + g_{i,2}(\bar{x}_{i,2}(k+1))x_{i,3}(k+1)] \\
 & \triangleq & p_{i,3}(\bar{x}_{i,3}(k)) + q_{i,3}(\bar{x}_{i,3}(k))x_{i,4}(k) \\
 & \triangleq & \alpha_{i,3}(\bar{x}_{i,4}(k)) \\
 & \vdots & \\
x_{i,1}(k+\tau-1) & \triangleq & p_{i,\tau-1}(\bar{x}_{i,\tau-1}(k)) + q_{i,\tau-1}(\bar{x}_{i,\tau-1}(k))x_{i,\tau}(k) \\
 & \triangleq & \alpha_{i,\tau-1}(\bar{x}_{i,\tau}(k))
\end{array}
\right.
\tag{3.79}
$$

with $p_{i,j}(\cdot)$, $q_{i,j}(\cdot)$ and $\alpha_{i,j}(\cdot)$ $(j = 1, \ldots, \tau - 1)$ being nonlinear functions.

Remark 3.7 *Due to the boundedness of $g_{j,i_j}(\cdot)$ $(1 \le j \le n\ 1 \le i_j \le \tau)$ in Assumption 3.3, we know that $q_{i,j}(\cdot)$ is also bounded.*

Furthermore, due to the boundedness of $g_{i,i_j}(\cdot)$ in Assumption 3.3, we can see that $q_{i,j}(\cdot)$ $(j = 1, \ldots, \tau - 1)$ are also bounded.

Now considering $x_{i,1}(k + \tau - 1)$, we know that

$$x_{i,1}(k+\tau-1) = p_{i,\tau-1}(\bar{x}_{i,\tau-1}(k)) + q_{i,\tau-1}(\bar{x}_{i,\tau-1}(k))x_{i,\tau}(k) \tag{3.80}$$

with $p_{i,\tau-1}(\cdot)$ and $q_{i,\tau-1}(\cdot)$ are highly entangled nonlinear functions. Proceeding one more step and noting the last equation in (3.73), we have

$$
\begin{array}{rcl}
x_{i,1}(k+\tau) & = & p_{i,\tau-1}(\bar{x}_{i,\tau-1}(k+1)) + q_{i,\tau-1}(\bar{x}_{i,\tau-1}(k+1))x_{i,\tau}(k+1) \\
 & \triangleq & p_{i,\tau}(\bar{x}_{i,\tau}(k)) + q_{i,\tau}(\bar{x}_{i,\tau}(k))\,[f_{i,\tau}(X(k), \bar{u}_{i-1}(k)) + g_{i,\tau}(X(k))u_i(k)] \\
 & \triangleq & p_i(X(k), \bar{u}_{i-1}(k)) + q_i(X(k))u_i(k)
\end{array}
\tag{3.81}
$$

with

$$\begin{aligned} p_i(X(k), \bar{u}_{i-1}(k)) &\triangleq p_{i,\tau}(\bar{x}_{i,\tau}(k)) + q_{i,\tau}(\bar{x}_{i,\tau}(k)) f_{i,\tau}(X(k), \bar{u}_{i-1}(k)) \\ q_i(X(k)) &\triangleq q_{i,\tau}(\bar{x}_{i,\tau}(k)) g_{i,\tau}(X(k)) \end{aligned}$$

Remark 3.8 *Noting Assumption 3.3, Remark 3.7 and that we have assumed the positiveness of $g_{j,i_j}(\cdot)$, it can be easily obtained that $q_i(X(k)) = q_{i,\tau}(\bar{x}_{i,\tau}(k))g_{i,\tau}(X(k))$ is also bounded. Specifically, there are two positive constants, $\underline{q}_i$ and $\bar{q}_i$, such that, $\underline{q}_i \leq q_i(\cdot) \leq \bar{q}_i$ $(1 \leq i \leq n)$.*

Therefore, the original system (3.70) becomes $(1 \leq i \leq n)$

$$\Sigma_i : \left\{ \begin{aligned} \xi_{i,1}(k+1) &= \xi_{i,2}(k) \\ \xi_{i,2}(k+1) &= \xi_{i,3}(k) \\ &\vdots \\ \xi_{i,\tau-1}(k+1) &= \xi_{i,\tau}(k) \\ \xi_{i,\tau}(k+1) &= p_i(X(k), \bar{u}_{i-1}(k)) + q_i(X(k)) u_i(k) \end{aligned} \right. \tag{3.82}$$

provided that the coordinate transformation, $T(X)$, is diffeomorphism. In the next, we will show that the mapping $T(X)$ is diffeomorphism actually.

Considering equations (3.75) and (3.79), the mapping from

$$x_i(k) = [x_{i,1}(k), x_{i,2}(k), \ldots, x_{i,\tau}(k)]^T \Rightarrow \xi_i(k) = [\xi_{i,1}(k), \xi_{i,2}(k), \ldots (k), \xi_{i,\tau}(k)]^T$$

can be expressed as follows

$$\left\{ \begin{aligned} \xi_{i,1}(k) &= x_{i,1}(k) \\ \xi_{i,2}(k) &= x_{i,1}(k+1) = p_{i,1}(\bar{x}_{i,1}(k)) + q_{i,1}(\bar{x}_{i,1}(k)) x_{i,2}(k) \\ \xi_{i,3}(k) &= x_{i,1}(k+2) = p_{i,2}(\bar{x}_{i,2}(k)) + q_{i,2}(\bar{x}_{i,2}(k)) x_{i,3}(k) \\ &\vdots \\ \xi_{i,\tau-1}(k) &= x_{i,1}(k+\tau-2) = p_{i,\tau-2}(\bar{x}_{i,\tau-2}(k)) + q_{i,\tau-2}(\bar{x}_{i,\tau-2}(k)) x_{i,\tau-1}(k) \\ \xi_{i,\tau}(k) &= x_{i,1}(k+\tau-1) = p_{i,\tau-1}(\bar{x}_{i,\tau-1}(k)) + q_{i,\tau-1}(\bar{x}_{i,\tau-1}(k)) x_{i,\tau}(k) \end{aligned} \right. \tag{3.83}$$

From equation (3.83), it can be seen that the coordinate transformation is

- subsystem decoupled, i.e., the coordinate transformation from $x_i(k)$ to $\xi_i(k)$ are independent to each other for different subsystems;

- independent of the control input $u_i(k)$, i.e., the coordinate transformation has nothing to do with the control input.

Define this mapping as follows ($1 \leq i \leq n$)

$$T_i(x_i(k)) : x_i(k) \rightarrow \xi_i(k) \tag{3.84}$$

then we know that the whole system coordinate transformation from $X(k)$ to $\Xi(k)$ defined in (3.77) can be written as follows:

$$T(X(k)) = \begin{bmatrix} T_1(x_1(k)) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & T_2(x_2(k)) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & T_n(x_n(k)) \end{bmatrix} \tag{3.85}$$

If we can verify that the mapping $T_i(x_i(k))$ in equation (3.83) is diffeomorphism [143, 65], then owing to the independent property of $T_i(x_i(k))$ ($1 \leq i \leq n$), we know that the whole system coordinate transformation, $T(X(k))$ in (3.85) is also diffeomorphism.

Lemma 3.2 *Let U be an open subset of R^n and let $\varphi = (\varphi_1, \ldots, \varphi_n) : U \rightarrow R^n$ be a smooth map. If the* **Jacobian Matrix**

$$\frac{d\varphi}{dx} = \begin{bmatrix} \frac{\partial \varphi_1}{\partial x_1} & \cdots & \frac{\partial \varphi_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \varphi_n}{\partial x_1} & \cdots & \frac{\partial \varphi_n}{\partial x_n} \end{bmatrix}$$

is nonsingular at some point $p \in U$, or equivalently, $\mathrm{Rank}(\frac{d\varphi}{dx}) = n$ *at some point $p \in U$, then there exists a neighborhood $V \subset U$ of p such that $\varphi : V \rightarrow \varphi(V)$ is a diffeomorphism [143, 65, 115].*

By using Lemma 3.2, we will show that $T(X(k))$ in (3.85) is a diffeomorphism, as detailed in Lemma 3.3.

Lemma 3.3 *Considering the mapping $T(X(k)) : X(k) \rightarrow \Xi(k)$, defined as*

$$T(X(k)) = diag[T_1(x_1(k)), T_2(x_2(k)), \ldots, T_n(x_n(k))]$$

in (3.77) and (3.84), it is a diffeomorphism.

Proof: The proof that $T_i(x_i(k))$ is a diffeomorphism can be found in [115], for completeness, it is also detailed here.

Considering the i-th subsystem ($1 \leq i \leq n$), we have $\xi_i(k) = T_i(x_i(k))$, it is shown in Lemma 3.2 that once (i) the map $T_i(\cdot)$ is invertible; and (ii) $T_i(\cdot)$ and $T_i^{-1}(\cdot)$ are both continuously differentiable, the map $T_i(\cdot)$ is a diffeomorphism.

Because that

$$\begin{aligned} x_{i,1}(k) &= \xi_{i,1}(k) \\ x_{i,1}(k+1) &= \xi_{i,2}(k) \end{aligned}$$

and we know that

$$x_{i,2}(k) = \frac{x_{i,1}(k+1) - f_{i,1}(\bar{x}_{i,1}(k))}{g_{i,1}(\bar{x}_{i,1}(k))}$$

then we have

$$x_{i,2}(k) = \frac{\xi_{i,2}(k) - f_{i,1}(\bar{x}_{i,1}(k))}{g_{i,1}(\bar{x}_{i,1}(k))} = \frac{\xi_{i,2}(k) - f_{i,1}(\bar{\xi}_{i,1}(k))}{g_{i,1}(\bar{\xi}_{i,1}(k))}$$

Therefore, we can define that

$$x_{i,2}(k) \triangleq t_{i,2}(\xi_{i,1}(k), \xi_{i,2}(k)) \triangleq t_{i,2}(k) \tag{3.86}$$

with

$$t_{i,2}(k) = \frac{\xi_{i,2}(k) - f_{i,1}(\bar{\xi}_{i,1}(k))}{g_{i,1}(\bar{\xi}_{i,1}(k))}$$

It is clear that

$$\begin{aligned} x_{i,2}(k+1) &= t_{i,2}(k+1) \\ \frac{\partial t_{i,2}(k)}{\partial \xi_{i,2}(k)} &= \frac{1}{g_{i,1}(\bar{\xi}_{i,1}(k))} \\ \frac{\partial t_{i,2}(k+1)}{\partial \xi_{i,3}(k)} &= \partial \left[\frac{\xi_{i,2}(k+1) - f_{i,1}(\bar{\xi}_{i,1}(k+1))}{g_{i,1}(\bar{\xi}_{i,1}(k+1))} \right] / \partial \xi_{i,3}(k) = \frac{1}{g_{i,1}(\bar{\xi}_{i,2}(k))} \\ & \quad (\text{noting} \ \ \xi_{i,2}(k+1) = \xi_{i,3}(k)) \end{aligned}$$

Because we know that

$$x_{i,3}(k) = \frac{x_{i,2}(k+1) - f_{i,2}(\bar{x}_{i,2}(k))}{g_{i,2}(\bar{x}_{i,2}(k))}$$

Therefore, we can obtain

$$\begin{aligned} x_{i,3}(k) &= \frac{t_{i,2}(k+1) - f_{i,2}([\xi_{i,1}(k), t_{i,2}(k)]^T)}{g_{i,2}([\xi_{i,1}(k), t_{i,2}(k)]^T)} \\ &\triangleq t_{i,3}(k) \end{aligned}$$

It is obvious that

$$\begin{aligned} \frac{\partial t_{i,3}(k)}{\partial \xi_{i,3}(k)} &= \frac{\partial t_{i,2}(k+1)/\partial \xi_{i,3}(k)}{g_{i,2}([\xi_{i,1}(k), t_{i,2}(k)]^T)} \\ &= \frac{1}{g_{i,1}(\bar{\xi}_{i,2}(k)) \times g_{i,2}([\xi_{i,1}(k), t_{i,2}(k)]^T)} \end{aligned}$$

Continue this process recursively, finally, we can obtain

$$\begin{aligned} x_{i,\tau}(k) &= t_{i,\tau}(\bar{\xi}_{i,\tau}(k)) \triangleq t_{i,\tau}(k) \\ \frac{\partial t_{i,\tau}(k)}{\xi_{i,\tau}(k)} &= \frac{1}{g_{i,1}(\bar{\xi}_{i,2}(k)) g_{i,2}([\xi_{i,1}(k), t_{i,2}(k)]^T) \cdots g_{i,\tau}([\xi_{i,1}(k), t_{i,2}(k), \ldots, t_{i,\tau-1}(k)]^T)} \end{aligned}$$

Therefore, we can see that the inverse transformation, $T_i^{-1}(\xi_i(k))$, for the i-th subsystem can be denoted as

$$T_i^{-1}(\xi_i(k)) = \begin{bmatrix} x_{i,1}(k) \\ t_{i,2}(k) \\ \vdots \\ t_{i,\tau}(k) \end{bmatrix} = \begin{bmatrix} x_{i,1}(k) \\ x_{i,2}(k) \\ \vdots \\ x_{i,\tau}(k) \end{bmatrix}$$

and consequently we have

$$\frac{\partial T_i^{-1}(\xi_i(k))}{\partial \xi_i(k)} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \star & \frac{1}{g_{1,1}(\xi_{1,2}(k))} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \star & \star & \star & \star \end{bmatrix} \tag{3.87}$$

Similarly, for the other subsystems, this coordinate transformation still holds. Therefore, for the whole system, the inverse transformation from $\Xi(k)$ to $X(k)$ can be expressed as

$$T^{-1}(\Xi) = \begin{bmatrix} T_1^{-1}(\xi_1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & T_2^{-1}(\xi_2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & T_n^{-1}(\xi_n) \end{bmatrix} \tag{3.88}$$

Noting (3.87), it can be concluded that the Jacobian matrix of $T^{-1}(\Xi)$ is both nonsingular and differentiable. Therefore, we conclude that both the mapping $T(X(k))$ and its inverse, $T^{-1}(\Xi(k))$ are all nonsingular and differentiable. Therefore, we have the following equation

$$X(k) = T^{-1}(\Xi(k)) \tag{3.89}$$

and $T(X(k))$ is a diffeomorphism actually. This completes the proof. ∎

Therefore, considering (3.82), we know that the i-th subsystem of (3.70) is in the following form

$$\left\{\begin{array}{rcl} \xi_{i,1}(k+1) & = & \xi_{i,2}(k) \\ \xi_{i,2}(k+1) & = & \xi_{i,3}(k) \\ & \vdots & \\ \xi_{i,\tau-1}(k+1) & = & \xi_{i,\tau}(k) \\ \xi_{i,\tau}(k+1) & = & p_i(X(k),\bar{u}_{i-1}(k)) + q_i(X(k))u_i(k) \end{array}\right. \tag{3.90}$$

Noting (3.89), equation (3.90) can be written as

$$\left\{\begin{array}{rcl} \xi_{i,1}(k+1) & = & \xi_{i,2}(k) \\ \xi_{i,2}(k+1) & = & \xi_{i,3}(k) \\ & \vdots & \\ \xi_{i,\tau-1}(k+1) & = & \xi_{i,\tau}(k) \\ \xi_{i,\tau}(k+1) & = & f_i(\Xi(k),\bar{u}_{i-1}(k)) + g_i(\Xi(k))u_i(k) \end{array}\right. \tag{3.91}$$

with

$$\begin{array}{rcl} f_i(\Xi(k),\bar{u}_{i-1}(k)) & \triangleq & p_i(T^{-1}(\Xi(k)),\bar{u}_{i-1}(k)) \\ g_i(\Xi(k)) & \triangleq & q_i(T^{-1}(\Xi(k))) \end{array}$$

This completes the first phase of system coordinate transformation.

<u>Coordinate Transformation: Phase Two</u> Now, the original system (3.70) has been

transferred into the following form

$$\Sigma : \begin{cases} \Sigma_1 : \begin{cases} \xi_{1,i_1}(k+1) = \xi_{1,i_1+1}(k), \quad 1 \le i_1 \le \tau - 1 \\ \xi_{1,\tau}(k+1) = f_1(\Xi(k)) + g_1(\Xi(k))u_1(k) \end{cases} \\ \quad \vdots \\ \Sigma_j : \begin{cases} \xi_{j,i_j}(k+1) = \xi_{j,i_j+1}(k), \quad 1 \le i_j \le \tau - 1 \\ \xi_{j,\tau}(k+1) = f_j(\Xi(k), \bar{u}_{j-1}(k)) + g_j(\Xi(k))u_j(k) \end{cases} \\ \quad \vdots \\ \Sigma_n : \begin{cases} \xi_{n,i_n}(k+1) = \xi_{n,i_n+1}(k), \quad 1 \le i_n \le \tau - 1 \\ \xi_{n,\tau}(k+1) = f_n(\Xi(k), \bar{u}_{n-1}(k)) + g_n(\Xi(k))u_n(k) \end{cases} \\ y_j(k) = \xi_{j,1}(k), \quad 1 \le j \le n \end{cases} \tag{3.92}$$

with $f_j(\cdot)$ and $g_j(\cdot)$ $(1 \le j \le n)$ being smooth nonlinear functions. Noting Remark 3.8, we have the following assumption

Assumption 3.5 *There are two positive constants $\underline{g}_i$ and $\bar{g}_i > 0$, such that $\underline{g}_i \le g_i(\cdot) \le \bar{g}_i$ $(1 \le i \le n)$, $\forall \Xi(k) \in \Omega \subset R^{n \times \tau}$.*

Motivated by the design procedure in [115], coordinate transformation is used to transform system (3.92) from state space description to input output description. Considering the j-th $(1 \le j \le n)$ subsystem in system (3.92)

$$\Sigma_j : \begin{cases} \xi_{j,1}(k+1) &= \xi_{j,2}(k) \\ \xi_{j,2}(k+1) &= \xi_{j,3}(k) \\ &\vdots \\ \xi_{j,\tau-1}(k+1) &= \xi_{j,\tau}(k) \\ \xi_{j,\tau}(k+1) &= f_j(\Xi(k), \bar{u}_{j-1}(k)) + g_j(\Xi(k))u_j(k) \end{cases} \tag{3.93}$$

In order to develop the output feedback control scheme, define the following new variables

$$\begin{aligned} \underline{y}_1(k) &= [y_1(k-\tau+1), \cdots, y_1(k-1), y_1(k)]^T \\ \underline{y}_2(k) &= [y_2(k-\tau+1), \cdots, y_2(k-1), y_2(k)]^T \\ &\vdots \\ \underline{y}_n(k) &= [y_n(k-\tau+1), \cdots, y_n(k-1), y_n(k)]^T \end{aligned}$$

and

$$
\begin{aligned}
\underline{u}_1^{k-1}(k) &= [u_1(k-1), \cdots, u_1(k-\tau+1)]^T \\
\underline{u}_2^{k-1}(k) &= [u_2(k-1), \cdots, u_2(k-\tau+1)]^T \\
&\vdots \\
\underline{u}_n^{k-1}(k) &= [u_n(k-1), \cdots, u_n(k-\tau+1)]^T
\end{aligned}
$$

Furthermore, define

$$
\begin{aligned}
\underline{z}_1(k) &= [\underline{y}_1^T(k), \underline{u}_1^{k-1^T}(k)]^T \\
\underline{z}_2(k) &= [\underline{y}_2^T(k), \underline{u}_2^{k-1^T}(k)]^T \\
&\vdots \\
\underline{z}_n(k) &= [\underline{y}_n^T(k), \underline{u}_n^{k-1^T}(k)]^T \\
\underline{z}(k) &= [\underline{z}_1^T(k), \underline{z}_2^T(k), \cdots, \underline{z}_n^T(k)]^T \in R^{(2\tau-1)\times n}
\end{aligned}
$$

According to the definition of the new states, we know that

$$
\begin{aligned}
\underline{y}_1(k) &= [\xi_{1,1}(k-\tau+1), \cdots, \xi_{1,1}(k-1), \xi_{1,1}(k)]^T \\
\underline{y}_2(k) &= [\xi_{2,1}(k-\tau+1), \cdots, \xi_{2,1}(k-1), \xi_{2,1}(k)]^T \\
&\vdots \\
\underline{y}_n(k) &= [\xi_{n,1}(k-\tau+1), \cdots, \xi_{n,1}(k-1), \xi_{n,1}(k)]^T
\end{aligned}
$$

Noting (3.93), we obtain

$$
\begin{aligned}
y_j(k+1) &= \xi_{j,2}(k) = \xi_{j,3}(k-1) = \cdots = \xi_{j,\tau}(k-\tau+2) \\
&= f_j(\Xi(k-\tau+1), \bar{u}_{j-1}(k-\tau+1)) \\
&\quad + g_j(\Xi(k-\tau+1))u_j(k-\tau+1)
\end{aligned}
\tag{3.94}
$$

Noting that

$$
\begin{aligned}
\Xi(k-\tau+1) = \Big[&\xi_{1,1}(k-\tau+1), \ldots, \xi_{1,\tau}(k-\tau+1), \\
&\xi_{2,1}(k-\tau+1), \ldots, \xi_{2,\tau}(k-\tau+1), \\
&\ldots, \\
&\xi_{n,1}(k-\tau+1), \ldots, \xi_{n,\tau}(k-\tau+1)\Big]^T
\end{aligned}
$$

$$\begin{aligned} &= \Big[y_1(k-\tau+1), \ldots, y_1(k), \\ &\quad y_2(k-\tau+1), \ldots, y_2(k), \\ &\quad \ldots, \\ &\quad y_n(k-\tau+1), \ldots, y_n(k)\Big]^T \\ &= \Big[\underline{y}_1^T(k), \underline{y}_2^T(k), \ldots, \underline{y}_n^T(k)\Big]^T \end{aligned}$$

and define

$$Y(k) = \Big[\underline{y}_1^T(k), \underline{y}_2^T(k), \cdots, \underline{y}_n^T(k)\Big]^T$$

we have

$$\Xi(k-\tau+1) = Y(k) \tag{3.95}$$

Now, equation (3.94) becomes

$$y_j(k+1) = f_j(Y(k), \bar{u}_{j-1}(k-\tau+1)) + g_j(Y(k))u_j(k-\tau+1) \tag{3.96}$$

This means that $\xi_{j,2}(k)$ is a function of $Y(k)$, $\bar{u}_{j-1}(k-\tau+1)$ and $u_j(k-\tau+1)$. It should be noted that although the right hand side of (3.96) does not contain all the elements of $\underline{z}(k)$, for convenience of analysis, we can denote (3.96) as follows without any ambiguity:

$$\begin{aligned} y_j(k+1) &= \xi_{j,2}(k) = f_j(Y(k), \bar{u}_j(k-\tau+1)) + g_j(Y(k))u_j(k-\tau+1) \\ &\triangleq \psi_{1,2}(\underline{z}(k)) \end{aligned} \tag{3.97}$$

It is obvious that

$$\begin{aligned} y_1(k+1) &= \xi_{1,2}(k) \\ &= f_1(Y(k)) + g_1(Y(k))u_1(k-\tau+1) \\ &\triangleq \psi_{1,2}(\underline{z}(k)) \\ y_2(k+1) &= \xi_{2,2}(k) \\ &= f_2(Y(k), u_1(k-\tau+1)) + g_2(Y(k))u_2(k-\tau+1) \\ &\triangleq \psi_{2,2}(\underline{z}(k)) \\ &\vdots \end{aligned}$$

$$\begin{aligned} y_n(k+1) &= \xi_{n,2}(k) \\ &= f_n(Y(k), u_1(k-\tau+1), \ldots, u_{n-1}(k-\tau+1)) + g_n(Y(k))u_n(k-\tau+1) \\ &\triangleq \psi_{n,2}(\underline{z}(k)) \end{aligned}$$

Thus, we obtain

$$Y(k+1) = [\underline{y}_1^T(k+1), \cdots, \underline{y}_n^T(k+1)]^T \triangleq \Psi_1(\underline{z}(k)) \tag{3.98}$$

Similarly, noting equation (3.95) and (3.96), we can obtain

$$\begin{aligned} y_j(k+2) &= \xi_{j,3}(k) \\ &= f_j(\Xi(k-\tau+2), \bar{u}_j(k-\tau+2)) + g_j(\Xi(k-\tau+2))u_j(k-\tau+2) \\ &= f_j(Y(k+1), u_j(k-\tau+2)) + g_j(Y(k+1))u_j(k-\tau+2) \end{aligned} \tag{3.99}$$

Substituting (3.98) into (3.99), we obtain

$$y_j(k+2) = \xi_{j,3}(k) \triangleq \psi_{j,3}(\underline{z}(k)) \tag{3.100}$$

Therefore, we can obtain

$$\begin{aligned} y_1(k+2) &= \xi_{1,3}(k) \triangleq \psi_{1,3}(\underline{z}(k)) \\ y_2(k+2) &= \xi_{2,3}(k) \triangleq \psi_{2,3}(\underline{z}(k)) \\ &\vdots \\ y_n(k+2) &= \xi_{n,3}(k) \triangleq \psi_{n,3}(\underline{z}(k)) \end{aligned} \tag{3.101}$$

Noting equation (3.101) and so on, it can be easily obtained that

$$Y(k+2) = [\underline{y}_1^T(k+2), \cdots, \underline{y}_n^T(k+2)]^T \triangleq \Psi_2(\underline{z}(k)) \tag{3.102}$$

Repeat the above procedure recursively, we can prove that

$$y_j(k+\tau-1) = \xi_{j,\tau}(k) \triangleq \psi_{j,\tau}(\underline{z}(k)) \tag{3.103}$$

This implies that $\xi_{j,\tau}(k)$ is a function of $\underline{z}(k)$. Similarly, the following equations hold

$$\begin{aligned} y_1(k+\tau-1) &= \xi_{1,\tau}(k) \triangleq \psi_{1,\tau}(\underline{z}(k)) \\ y_2(k+\tau-1) &= \xi_{2,\tau}(k) \triangleq \psi_{2,\tau}(\underline{z}(k)) \\ &\vdots \\ y_n(k+\tau-1) &= \xi_{n,\tau}(k) \triangleq \psi_{n,\tau}(\underline{z}(k)) \end{aligned} \tag{3.104}$$

By noting equations (3.97), (3.100) and so on, we can conclude that

$$\begin{aligned} \xi_j(k) &= [\xi_{j,1}(k), \xi_{j,2}(k), \cdots, \xi_{j,\tau}(k)]^T = [y_j(k), \psi_{j,2}(\underline{z}(k)), \cdots, \psi_{j,\tau}(\underline{z}(k))]^T \\ &\triangleq \psi_j(\underline{z}(k)) \end{aligned}$$

Therefore, we have

$$\begin{aligned} \xi_1(k) &= [\xi_{1,1}(k), \xi_{1,2}(k), \cdots, \xi_{1,\tau}(k)]^T \triangleq \psi_1(\underline{z}(k)) \\ \xi_2(k) &= [\xi_{2,1}(k), \xi_{2,2}(k), \cdots, \xi_{2,\tau}(k)]^T \triangleq \psi_2(\underline{z}(k)) \\ &\vdots \\ \xi_n(k) &= [\xi_{n,1}(k), \xi_{n,2}(k), \cdots, \xi_{n,\tau}(k)]^T \triangleq \psi_n(\underline{z}(k)) \end{aligned}$$

Then, the system state $\Xi(k) = [\xi_1^T(k), \xi_2^T(k), \cdots, \xi_n^T(k)]^T$ is also depend on $\underline{z}(k)$, that means

$$\Xi(k) = \mathbf{\Psi}(\underline{z}(k)) \tag{3.105}$$

with $\mathbf{\Psi}(\cdot)$ being a vector nonlinear function. Up to this step, $\mathbf{\Psi}(\cdot)$ contains all the elements of $\underline{z}(k)$.

Noting equation (3.103) and the last equation in system (3.93), we have

$$y_j(k+\tau) = \xi_{j,\tau}(k+1) = f_j(\Xi(k), \bar{u}_{j-1}(k)) + g_j(\Xi(k))u_j(k) \tag{3.106}$$

Substituting (3.105) into (3.106), we have

$$y_j(k+\tau) = f_j(\mathbf{\Psi}(\underline{z}(k)), \bar{u}_{j-1}(k)) + g_j(\mathbf{\Psi}(\underline{z}(k)))u_j(k)$$

Now we can obtain the input output representation of system (3.92) as follows

$$\left\{ \begin{aligned} y_1(k+\tau) &= f_1(\mathbf{\Psi}(\underline{z}(k))) + g_1(\mathbf{\Psi}(\underline{z}(k)))u_1(k) \\ &\vdots \\ y_j(k+\tau) &= f_j(\mathbf{\Psi}(\underline{z}(k)), \bar{u}_{j-1}(k)) + g_j(\mathbf{\Psi}(\underline{z}(k)))u_j(k) \\ &\vdots \\ y_n(k+\tau) &= f_n(\mathbf{\Psi}(\underline{z}(k)), \bar{u}_{n-1}(k)) + g_n(\mathbf{\Psi}(\underline{z}(k)))u_n(k) \end{aligned} \right. \tag{3.107}$$

For the convenience of analysis, define

$$
\begin{aligned}
&f_1(k) = f_1(\mathbf{\Psi}(\underline{z}(k))), && g_1(k) = g_1(\mathbf{\Psi}(\underline{z}(k)))\\
&f_2(k, \bar{u}_1(k)) = f_2(\mathbf{\Psi}(\underline{z}(k)), \bar{u}_1(k)), && g_2(k) = g_2(\mathbf{\Psi}(\underline{z}(k)))\\
&\qquad\vdots && \qquad\vdots\\
&f_n(k, \bar{u}_{n-1}(k)) = f_n(\mathbf{\Psi}(\underline{z}(k)), \bar{u}_{n-1}(k)), && g_n(k) = g_n(\mathbf{\Psi}(\underline{z}(k)))
\end{aligned}
$$

Remark 3.9 *By now, we have successfully transformed the original MIMO system from state space representation (3.92) into input output representation (3.107), with the triangular form inputs structure unchanged. Considering the input output representation (3.107), it can be regarded as a τ-step ahead predictor model, in which, the current outputs are determined by system information of τ steps earlier. Thus, different from those traditional one step parameter update law [58] used for one-step ahead predictor, high order update laws should be used to deal with this τ-step predictor model, which will be discussed later.*

In the next, a simple example will be given to illustrate the detailed transformation procedure described above. Furthermore, the desired controls are also illustrated, which will be specifically discussed in Section 3.2.3.

Illustrative Example: To illustrate the transformation procedure, let us look at the following simple example ($n = \tau = 2$):

$$
\Sigma : \begin{cases}
\Sigma_1 : \begin{cases} x_{1,1}(k+1) = x_{1,1}(k) + x_{1,2}(k) \\ x_{1,2}(k+1) = x_{1,1}(k)x_{2,1}(k) + u_1(k) \end{cases}\\
\Sigma_2 : \begin{cases} x_{2,1}(k+1) = x_{2,1}(k) + x_{2,2}(k) \\ x_{2,2}(k+1) = x_{1,2}(k)x_{2,2}(k)u_1^2(k) + u_2(k) \end{cases}\\
y_j(k) = x_{j,1}(k), \quad 1 \le j \le 2
\end{cases}
\tag{3.108}
$$

Define

$$
\begin{cases} \xi_{1,1}(k) = x_{1,1}(k) \\ \xi_{1,2}(k) = x_{1,1}(k+1) \end{cases}
\quad \text{and} \quad
\begin{cases} \xi_{2,1}(k) = x_{2,1}(k) \\ \xi_{2,2}(k) = x_{2,1}(k+1) \end{cases}
$$

The original system can be easily transformed into the follows:

$$\begin{cases} \Sigma_1 : \begin{cases} \xi_{1,1}(k+1) = \xi_{1,2}(k) \\ \xi_{1,2}(k+1) = \xi_{1,2}(k) + \xi_{1,1}(k)\xi_{2,1}(k) + u_1(k) \end{cases} \\ \Sigma_2 : \begin{cases} \xi_{2,1}(k+1) = \xi_{2,2}(k) \\ \xi_{2,2}(k+1) = \xi_{2,2}(k) + [\xi_{1,2}(k) - \xi_{1,1}(k)]\,[\xi_{2,2}(k) - \xi_{2,1}(k)]\,u_1^2(k) \\ \qquad\qquad\qquad +u_2(k) \end{cases} \\ y_j(k) = \xi_{j,1}(k), \quad 1 \le j \le 2 \end{cases}$$

Consequently, we can easily obtain that

$$\begin{cases} \xi_{1,1}(k+2) &= \xi_{1,2}(k) + \xi_{1,1}(k)\xi_{2,1}(k) + u_1(k) \\ \xi_{2,1}(k+2) &= \xi_{2,2}(k) + [\xi_{1,2}(k) - \xi_{1,1}(k)]\,[\xi_{2,2}(k) - \xi_{2,1}(k)]\,u_1^2(k) + u_2(k) \end{cases} \tag{3.109}$$

Noting that

$$\begin{cases} \xi_{1,1}(k) &= y_1(k) \\ \xi_{1,2}(k) &= y_1(k+1) \end{cases} \quad \text{and} \quad \begin{cases} \xi_{2,1}(k) &= y_2(k) \\ \xi_{2,2}(k) &= y_2(k+1) \end{cases}$$

Therefore, we can obtain that

$$\begin{aligned} \xi_{1,2}(k) &= \xi_{1,2}(k-1) + \xi_{1,1}(k-1)\xi_{2,1}(k-1) + u_1(k-1) \\ &= y_1(k) + y_1(k-1)y_2(k-1) + u_1(k-1) \\ &\triangleq f_a(k) \\ \xi_{2,2}(k) &= \xi_{2,2}(k-1) + [\xi_{1,2}(k-1) - \xi_{1,1}(k-1)]\,[\xi_{2,2}(k-1) - \xi_{2,1}(k-1)]\,u_1^2(k-1) \\ &\quad +u_2(k-1) \\ &= y_2(k) + [y_1(k) - y_1(k-1)]\,[y_2(k) - y_2(k-1)]\,u_1^2(k-1) + u_2(k-1) \\ &\triangleq f_b(k) \end{aligned}$$

Hence, equation (3.109) can be rewritten as

$$\begin{cases} y_1(k+2) &= f_a(k) + y_1(k)y_2(k) + u_1(k) \\ y_2(k+2) &= f_b(k) + [f_a(k) - y_1(k)]\,[f_b(k) - y_2(k)]\,u_1^2(k) + u_2(k) \end{cases} \tag{3.110}$$

Assuming the desired trajectory is $y_d(k) = [y_{d_1}(k), y_{d_2}(k)]^T$, therefore, we can get the desired control as

$$\begin{aligned} u_1^*(k) &= y_{d_1}(k+2) - y_1(k)y_2(k) - f_a(k) \\ u_2^*(k) &= y_{d_2}(k+2) - [f_a(k) - y_1(k)]\,[f_b(k) - y_2(k)]\,u_1^{*^2}(k) - f_b(k) \\ &= y_{d_2}(k+2) - [f_a(k) - y_1(k)]\,[f_b(k) - y_2(k)]\,[y_{d_1}(k) - y_1(k)y_2(k)]^2 - f_b(k) \end{aligned}$$

which can realize the exact tracking in 2 steps.

Assume system initial conditions are: $u_1^*(0) = 0$, $u_2^*(0) = 0$, $y_1(0) = y_1(1) = y_1(2) = 0$ and $y_2(0) = y_2(1) = y_2(2) = 0$. The reference trajectory, $y_{d_1}(k)$ and $y_{d_2}(k)$, are shown in Table 3.2. Practical control action starts at time instant $k = 1$. Table 3.2, Figures 3.4 and 3.5 show the system variation from $k = 0$ to $k = 8$.

k	0	1	2	3	4	5	6	7	8
$u_1^*(k)$	<u>0</u>	0.2	0.1	-0.24	-0.2	0.12	0.09	0.1	-0.3
$y_1(k)$	<u>0</u>	<u>0</u>	<u>0</u>	**0.2**	**0.3**	**0.1**	**-0.1**	**0**	**0.1**
$y_{d_1}(k)$	-0.2	-0.1	0.1	**0.2**	**0.3**	**0.1**	**-0.1**	**0**	**0.1**
$u_2^*(k)$	<u>0</u>	0.2	-0.2004	-0.1988	0.0984	0.2003	-0.1002	-0.0999	-0.0991
$y_2(k)$	<u>0</u>	<u>0</u>	<u>0</u>	**0.2**	**0**	**-0.2**	**-0.1**	**0.1**	**0**
$y_{d_2}(k)$	-0.1	0.1	0.3	**0.2**	**0**	**-0.2**	**-0.1**	**0.1**	**0**

§The numbers with <u>underscores</u> represent system initial conditions. The numbers in **bold** indicate that exact tracking is obtained.

Table 3.2: A Simple Example - System Variation

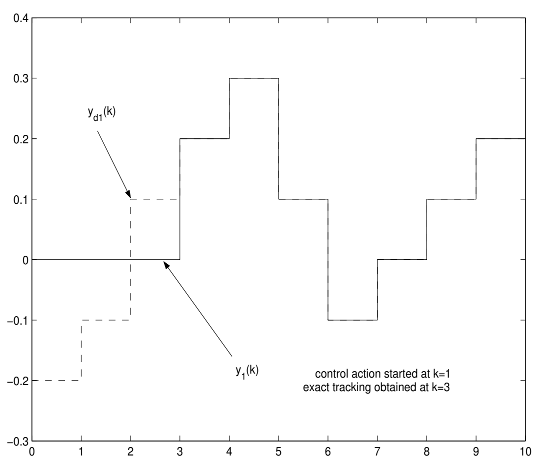

Figure 3.4: Example: y_1 and y_{d_1}

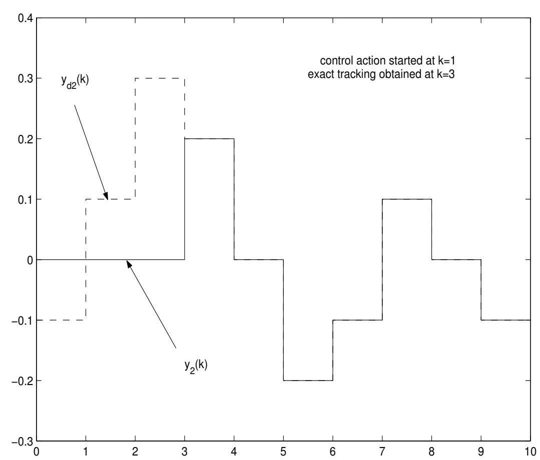

Figure 3.5: Example: y_2 and y_{d_2}

It can be seen that, for this example, the control action is started from $k = 1$. The exact tracking is achieved at $k = 3$, as what we expected. The exact tracking is achieved in $\tau = 2$ steps.

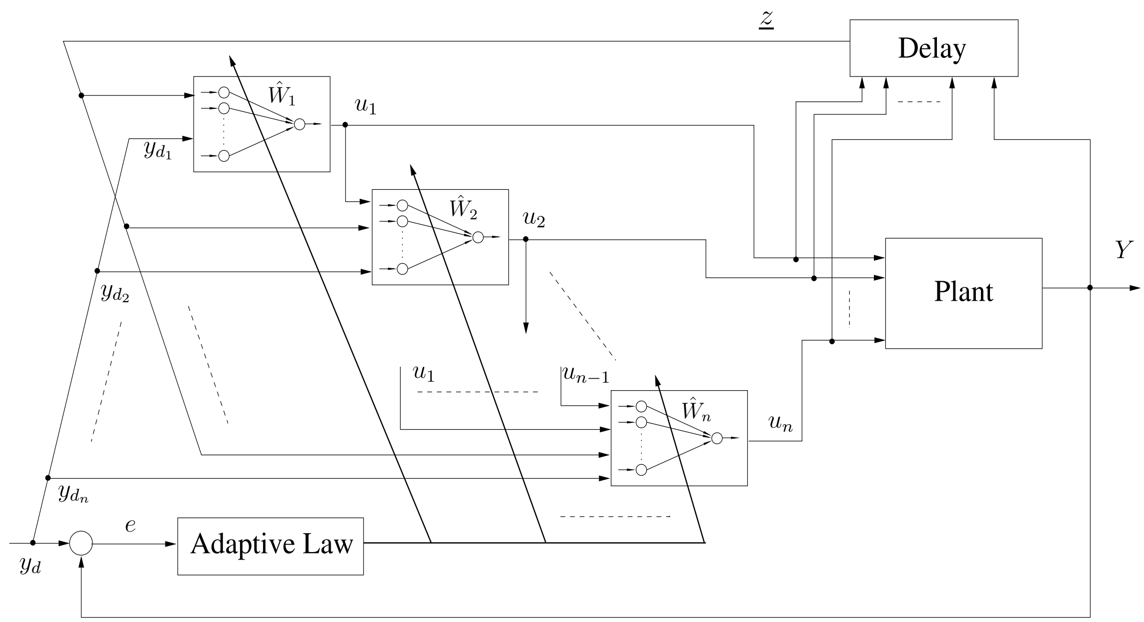

Figure 3.6: Output Feedback Control - Control System Structure

3.2.3 Controller Design and Stability Analysis

The closed-loop system structure is shown in Figure 3.6. Now, consider the input output representation (3.107) of system (3.92), we have illustrated that if (3.107) is stable, the stability of (3.92) will be guaranteed. In the next, we will develop stable adaptive NN controls and corresponding weight tuning laws for system (3.107), which will also stabilize system (3.92).

Define tracking error as $e(k) = [e_1(k), \ldots, e_n(k)]^T$, with

$$e_i(k) = y_i(k) - y_{d_i}(k), \quad i = 1, \ldots, n \tag{3.111}$$

then the error dynamics can be obtained

$$\left\{ \begin{array}{rcl} e_1(k+\tau) & = & f_1(k) + g_1(k)u_1(k) - y_{d_1}(k+\tau) \\ e_2(k+\tau) & = & f_2(k, \bar{u}_1(k)) + g_2(k)u_2(k) - y_{d_2}(k+\tau) \\ & \vdots & \\ e_n(k+\tau) & = & f_n(k, \bar{u}_{n-1}(k)) + g_n(k)u_n(k) - y_{d_n}(k+\tau) \end{array} \right. \tag{3.112}$$

Consider the first equation in error dynamics (3.112), if we choose the desired control

$u_1^*(k)$ as

$$u_1^*(k) = \frac{y_{d_1}(k+\tau) - f_1(k)}{g_1(k)} \tag{3.113}$$

we can obtain $e_1(k+\tau) = 0$. Therefore, the tracking error $e_1(k)$ will reach zero in τ steps. However, in practical applications, normally, exact system model cannot be obtained. Therefore, the desired control $u_1^*(k)$ is not applicable. Instead, we can use high order neural networks to approximate $u_1^*(k)$

$$\begin{aligned} u_1^*(k) &= W_1^{*T} S_1(z_1(k)) + \epsilon_{z_1}(z_1(k)) \qquad (3.114) \\ z_1(k) &= [\underline{z}(k), y_{d_1}(k+\tau)]^T \in \Omega_{z_1} \subset R^{1+(2\tau-1)\times n} \end{aligned}$$

Choose the practical adaptive control input $u_1(k)$ and robust updating algorithm for NN weights as

$$u_1(k) = \hat{W}_1^T(k) S_1(z_1(k)) \tag{3.115}$$

$$\hat{W}_1(k) = \hat{W}_1(k-\tau) - \Gamma_1 \left[S_1(z_1(k-\tau)) e_1(k) + \sigma_1 \hat{W}_1(k-\tau) \right] \tag{3.116}$$

where $\Gamma_1^T = \Gamma_1 > 0$ is the adaptation diagonal gain matrix and $\hat{W}_1(k)$ denotes the estimation of $W_1^*(k)$.

Once $u_1(k)$ is confirmed, the desired control $u_2^*(k)$ can be chosen as

$$u_2^*(k) = \frac{y_{d_2}(k+\tau) - f_2(k, \bar{u}_1(k))}{g_2(k)} \tag{3.117}$$

which will drive $e_2(k+\tau) = 0$. Similar, we know that there exists a high order neural network, such that

$$\begin{aligned} u_2^*(k) &= W_2^{*T} S_2(z_2(k)) + \epsilon_{z_2}(z_2(k)) \qquad (3.118) \\ z_2(k) &= [\underline{z}(k), \bar{u}_1(k), y_{d_2}(k+\tau)]^T \in \Omega_{z_2} \subset R^{2+(2\tau-1)\times n} \end{aligned}$$

Choose the direct adaptive control and corresponding neural weight update law as

$$u_2(k) = \hat{W}_2^T(k) S_2(z_2(k)) \tag{3.119}$$

$$\hat{W}_2(k) = \hat{W}_2(k-\tau) - \Gamma_2 \left[S_2(z_2(k-\tau)) e_2(k) + \sigma_2 \hat{W}_2(k-\tau) \right] \tag{3.120}$$

where $\Gamma_2^T = \Gamma_2 > 0$ is the adaptation diagonal gain matrix and $\hat{W}_2(k)$ denotes the estimation of $W_2^*(k)$.

Repeat the above procedure recursively, at step i, we know that the desired control $u_i^*(k)$ is

$$u_i^*(k) = \frac{y_{d_i}(k+\tau) - f_i(k, \bar{u}_{i-1}(k))}{g_i(k)} \tag{3.121}$$

Its HONN approximation is

$$\begin{aligned} u_i^*(k) &= W_i^{*T} S_i(z_i(k)) + \epsilon_{z_i}(z_i(k)) \\ z_i(k) &= [\underline{z}(k), \bar{u}_{i-1}(k), y_{d_i}(k+\tau)]^T \in \Omega_{z_i} \subset R^{i+(2\tau-1)\times n} \end{aligned} \tag{3.122}$$

Accordingly, the practical control input $u_i(k)$ and its NN weight update law are chosen as follows

$$u_i(k) = \hat{W}_i^T(k) S_i(z_i(k)) \tag{3.123}$$

$$\hat{W}_i(k) = \hat{W}_i(k-\tau) - \Gamma_i \left[S_i(z_i(k-\tau)) e_i(k) + \sigma_i \hat{W}_i(k-\tau) \right] \tag{3.124}$$

where $\Gamma_i^T = \Gamma_i > 0$ is the adaptation diagonal gain matrix and $\hat{W}_i(k)$ denotes the estimation of $W_i^*(k)$. In the final step, we know that the desired control $u_n^*(k)$ is

$$u_n^*(k) = \frac{y_{d_n}(k+\tau) - f_n(k, \bar{u}_{n-1}(k))}{g_n(k)} \tag{3.125}$$

Its HONN approximation is

$$\begin{aligned} u_n^*(k) &= W_n^{*T} S_n(z_n(k)) + \epsilon_{z_n}(z_n(k)) \\ z_n(k) &= [\underline{z}(k), \bar{u}_{n-1}(k), y_{d_n}(k+\tau)]^T \in \Omega_{z_n} \subset R^{2\tau n} \end{aligned} \tag{3.126}$$

Accordingly, the practical control input $u_n(k)$ and its NN weight update law are chosen as follows

$$u_n(k) = \hat{W}_n^T(k) S_n(z_n(k)) \tag{3.127}$$

$$\hat{W}_n(k) = \hat{W}_n(k-\tau) - \Gamma_n \left[S_n(z_n(k-\tau)) e_n(k) + \sigma_n \hat{W}_n(k-\tau) \right] \tag{3.128}$$

where $\Gamma_n^T = \Gamma_n > 0$ is the adaptation diagonal gain matrix and $\hat{W}_n(k)$ denotes the estimation of $W_n^*(k)$.

Summarize equations (3.115) and (3.116), (3.119) and (3.120), (3.123) and (3.124), (3.127) and (3.128), we propose the HONN controls and weight update laws for system

(3.107) as follows

$$
\begin{aligned}
u_i(k) &= \hat{W}_i^T(k) S_i(z_i(k)) && (3.129)\\
z_i(k) &= [\underline{z}(k), \bar{u}_{i-1}(k), y_{d_i}(k+\tau)]^T \in \Omega_{z_i} \subset R^{i+(2\tau-1)\times n} \\
\hat{W}_i(k) &= \hat{W}_i(k-\tau) - \Gamma_i \left[S_i(z_i(k-\tau)) e_i(k) + \sigma_i \hat{W}_i(k-\tau) \right] && (3.130)\\
& \quad i = 1, \ldots, n
\end{aligned}
$$

where $\Gamma_i = \mathrm{diag}[\gamma_{i_1}, \gamma_{i_2}, \ldots, \gamma_{i_n}] > 0$ is diagonal adaptation gain matrix with its every element, $0 < \gamma_{i_1}, \gamma_{i_2}, \ldots, \gamma_{i_n} < 1$ and $0 < \sigma_i < 1$ are positive constants. It should be noticed that, in the neural network weights update laws (3.130), σ-modification [62] is used to improve the robustness of the controller. For the ease of analysis, equation can also be written as

$$
\hat{W}_i(k+\tau) = \hat{W}_i(k) - \Gamma_i \left[S_i(z_i(k)) e_i(k+\tau) + \sigma_i \hat{W}_i(k) \right], \ i = 1, \ldots, n \quad (3.131)
$$

The stability of the closed-loop system is summarized in Theorem 3.3.

Theorem 3.3 *Consider the closed-loop nonlinear MIMO system consists of system (3.70), NN controls (3.129) and NN weight update laws (3.130), it is semi-globally uniformly ultimately bounded, and has an equilibrium at $[e_1(k), \ldots, e_n(k)]^T = 0$ provided that the design parameters are properly chosen. This guarantees that all the signals include the state vector $X(k)$, the control inputs $u_i(k)$ and NN weight estimates $\hat{W}_i(k)$, $i = 1, \ldots, n$ are all bounded, subsequently,*

$$
\lim_{k \to \infty} \|y(k) - y_d(k)\| \le \epsilon
$$

where ϵ is a small positive number.

Proof: The prove procedure is as follows:

1. In the first step, for the first subsystem, by choosing neural network controller $u_1(k)$, its stability is guaranteed by using Lyapunov analysis;

2. In the second step, once $u_1(k)$ is determined, by choosing $u_2(k)$, we prove the SGUUB stability for the first two subsystems Σ_1 and Σ_2;

3. Repeat this procedure recursively, in step i, choose $u_i(k)$ to stabilize subsystems Σ_1 to Σ_i;

4. Finally, in step n, choose $u_n(k)$ to guarantee the stability of all the subsystems.

Suppose that $Y(k-\tau), Y(k-\tau+1), \ldots, Y(k-1) \in \Omega$, $\forall k \geq 0$ and Ω denotes the compact set in which NN approximation (3.114), (3.118), (3.122) and (3.126) are valid. Now we prove that $Y(k) \in \Omega$ and $u(k)$ is bounded by backstepping.

Step 1: Noting that $e_1(k) = y_1(k) - y_{d_1}(k)$, its τth difference is given by

$$\begin{aligned} e_1(k+\tau) &= y_1(k+\tau) - y_{d_1}(k+\tau) \\ &= f_1(k) + g_1(k)u_1(k) - y_{d_1}(k+\tau) \end{aligned} \tag{3.132}$$

Adding and subtracting $g_1(k)u_1^*(k)$ on the right side of equation (3.132) and noting equation (3.113), we have

$$\begin{aligned} e_1(k+\tau) &= g_1(k)\left(u_1(k) - u_1^*(k)\right) \\ &= g_1(k)\left[\tilde{W}_1^T(k)S_1(z_1(k)) - \epsilon_{z_1}\right] \end{aligned}$$

with $\tilde{W}_1^T(k) = \hat{W}_1^T(k) - W_1^*(k)$ denotes the estimation error of the NN weight. Consequently, we obtain

$$\tilde{W}_1^T(k)S_1(z_1(k)) = \frac{e_1(k+\tau)}{g_1(k)} + \epsilon_{z_1} \tag{3.133}$$

Choose the following Lyapunov function candidate

$$V_1(k) = \frac{1}{\bar{g}_1}\sum_{j=0}^{\tau-1} e_1^2(k+j) + \sum_{j=0}^{\tau-1} \tilde{W}_1^T(k+j)\Gamma_1^{-1}\tilde{W}_1(k+j) \tag{3.134}$$

Its first difference is

$$\begin{aligned} \Delta V_1(k) &= V_1(k+1) - V_1(k) \\ &= \frac{1}{\bar{g}_1}e_1^2(k+\tau) - \frac{1}{\bar{g}_1}e_1^2(k) + \tilde{W}_1^T(k+\tau)\Gamma_1^{-1}\tilde{W}_1(k+\tau) - \tilde{W}_1^T(k)\Gamma_1^{-1}\tilde{W}_1(k) \end{aligned}$$

Noting the weight update algorithm (3.131) and equation (3.133), we have

$$\Delta V_1(k) = \frac{1}{\bar{g}_1}e_1^2(k+\tau) - \frac{1}{\bar{g}_1}e_1^2(k) - 2\tilde{W}_1^T(k)S_1(z_1(k))e_1(k+\tau) - 2\sigma_1\tilde{W}_1^T(k)\hat{W}_1(k)$$

$$\begin{aligned}
& +S_1^T(z_1(k))\Gamma_1^T S_1(z_1(k))e_1^2(k+\tau) + 2\sigma_1\hat{W}_1^T(k)\Gamma_1^T S_1(z_1(k))e_1(k+\tau) \\
& +\sigma_1^2\hat{W}_1^T(k)\Gamma_1^T\hat{W}_1(k) \\
= \; & \frac{1}{\bar{g}_1}e_1^2(k+\tau) - \frac{1}{\bar{g}_1}e_1^2(k) - 2\frac{1}{g_1(k)}e_1^2(k+\tau) - 2\epsilon_{z_1}e_1(k+\tau) \\
& -2\sigma_1\tilde{W}_1^T(k)\hat{W}_1(k) + S_1^T(z_1(k))\Gamma_1^T S_1(z_1(k))e_1^2(k+\tau) \\
& +2\sigma_1\hat{W}_1^T(k)\Gamma_1^T S_1(z_1(k))e_1(k+\tau) + \sigma_1^2\hat{W}_1^T(k)\Gamma_1^T\hat{W}_1(k) \\
\leq \; & -\frac{1}{\bar{g}_1}e_1^2(k+\tau) - \frac{1}{\bar{g}_1}e_1^2(k) - 2\epsilon_{z_1}e_1(k+\tau) - 2\sigma_1\tilde{W}_1^T(k)\hat{W}_1(k) \\
& S_1^T(z_1(k))\Gamma_1^T S_1(z_1(k))e_1^2(k+\tau) + 2\sigma_1\hat{W}_1^T(k)\Gamma_1^T S_1(z_1(k))e_1(k+\tau) \\
& +\sigma_1^2\hat{W}_1^T(k)\Gamma_1^T\hat{W}_1(k)
\end{aligned}$$

Using the following facts

$$S_1^T(z_1(k))S_1(z_1(k)) < l_1$$

$$S_1^T(z_1(k))\Gamma_1^T S_1(z_1(k)) \leq \bar{\gamma}_1 S_1^T(z_1(k))S_1(z_1(k)) \leq \bar{\gamma}_1 l_1$$

$$2\epsilon_{z_1}e_1(k+\tau) \leq \frac{\bar{\gamma}_1 e_1^2(k+\tau)}{\bar{g}_1} + \frac{\bar{g}_1\epsilon_{z_1}^2}{\bar{\gamma}_1}$$

$$2\sigma_1\hat{W}_1^T(k)\Gamma_1 S_1(z_1(k))e_1(k+\tau) \leq \frac{\bar{\gamma}_1 l_1 e_1^2(k+\tau)}{\bar{g}_1} + \bar{g}_1\sigma_1^2\bar{\gamma}_1\|\hat{W}_1\|^2$$

$$2\tilde{W}_1^T(k)\hat{W}_1(k) = \|\tilde{W}_1(k)\|^2 + \|\hat{W}_1(k)\|^2 - \|W_1^*\|^2$$

where l_1 denotes the neurons used and $\bar{\gamma}_1 = \max\{\gamma_{1_1}, \gamma_{1_2}, \ldots, \gamma_{1_n}\}$ denotes the biggest eigenvalue of Γ_1, we obtain

$$\begin{aligned}
\Delta V_1 \;\leq\; & -\frac{\rho_1}{\bar{g}_1}e_1^2(k+\tau) - \frac{1}{\bar{g}_1}e_1^2(k) + \beta_1 - \sigma_1(1-\sigma_1\bar{\gamma}_1 - \bar{g}_1\sigma_1\bar{\gamma}_1)\|\hat{W}_1(k)\|^2 \\
& -\sigma_1\|\tilde{W}_1(k)\|^2
\end{aligned}$$

where

$$\rho_1 = 1 - \bar{\gamma}_1 - \bar{\gamma}_1 l_1 - \bar{g}_1\bar{\gamma}_1 l_1, \qquad \beta_1 = \frac{\bar{g}_1\epsilon_{z_1}^2}{\bar{\gamma}_1} + \sigma_1\|W_1^*\|^2$$

If we choose the design parameters as follows

$$\bar{\gamma}_1 < \frac{1}{1 + l_1 + \bar{g}_1 l_1}, \qquad \sigma_1 < \frac{1}{(1+\bar{g}_1)\bar{\gamma}_1} \tag{3.135}$$

then we have

$$\Delta V_1 \leq -\frac{1}{\bar{g}_1}e_1^2(k) + \beta_1$$

Then $\Delta V_1 \leq 0$ once the error $|e_1(k)|$ is larger than $\sqrt{\bar{g}_1\beta_1}$. This implies the boundedness of $e_1(k)$, and we know that the tracking error $e_1(k)$ will asymptotically converge to a compact set denoted by $\Omega_1 \subset R$, where $\Omega_1 \triangleq \{\chi \big| \chi \leq \sqrt{\bar{g}_1\beta_1}\}$.

The adaptation dynamics (3.116) can be written as

$$\begin{aligned}\hat{W}_1(k+\tau) &= (I - \Gamma_1\sigma_1)\hat{W}_1(k) - \Gamma_1[S_1(z_1(k))e_1(k+\tau) + \sigma_1 W_1^*] \\ &= A_1(k)\hat{W}_1(k) - \Gamma_1[S_1(z_1(k))e_1(k+\tau) + \sigma_1 W_1^*]\end{aligned}$$

Because $0 < \gamma_{1_1}, \gamma_{1_2}, \ldots, \gamma_{1_n} < 1$ and $0 < \sigma_1 < 1$, we know that the transition matrix $\|\Phi(k_1, k_0)\|$ of $A_1(k)$ always satisfies $\|\Phi(k_1, k_0)\| < 1$. Furthermore, noting $S_1(z_1(k))$, $e_1(k+\tau)$ and $\sigma_1 W_1^*$ are all bounded, by applying Lemma A.1, $\hat{W}_1(k)$ is bounded in a compact set denoted by Ω_{w_1}, and hence the boundedness of $\hat{W}_1(k)$ is assured.

Step 2: As defined in equation (3.111), $e_2(k) = y_2(k) - y_{d_2}(k)$, its τth difference is given by

$$\begin{aligned}e_2(k+\tau) &= y_2(k+\tau) - y_{d_2}(k+\tau) \\ &= f_2(k, \bar{u}_1(k)) + g_2(k)u_2(k) - y_{d_2}(k+\tau)\end{aligned} \tag{3.136}$$

Adding and subtracting $g_2(k)u_2^*(k)$ on the right side of equation (3.136) and noting equation (3.117), we have

$$\begin{aligned}e_2(k+\tau) &= g_2(k)\left[u_2(k) - u_2^*(k)\right] \\ &= g_2(k)\left[\tilde{W}_2^T(k)S_2(z_2(k)) - \epsilon_{z_2}\right]\end{aligned}$$

with $\tilde{W}_2(k) = \hat{W}_2(k) - W_2^*(k)$ denotes the estimation error of the NN weight $W_2^*(k)$. Consequently, we obtain

$$\tilde{W}_2^T(k)S_2(z_2(k)) = \frac{e_2(k+\tau)}{g_2(k)} + \epsilon_{z_2} \tag{3.137}$$

Choose the following Lyapunov function candidate for subsystems Σ_1 and Σ_2

$$V_2(k) = V_1(k) + \frac{1}{\bar{g}_2}\sum_{j=0}^{\tau-1} e_2^2(k+j) + \sum_{j=0}^{\tau-1} \tilde{W}_2^T(k+j)\Gamma_2^{-1}\tilde{W}_2(k+j) \tag{3.138}$$

Its first difference is

$$\begin{aligned}\Delta V_2(k) &= \Delta V_1(k) + \frac{1}{\bar{g}_1}e_2^2(k+\tau) - \frac{1}{\bar{g}_1}e_2^2(k) \\ &\quad + \tilde{W}_2^T(k+\tau)\Gamma_2^{-1}\tilde{W}_2(k+\tau) - \tilde{W}_2^T(k)\Gamma_2^{-1}\tilde{W}_2(k)\end{aligned} \tag{3.139}$$

Noting the weight update algorithm (3.131) and equation (3.137), by following the similar procedure as in *Step 1*, we obtain

$$\begin{aligned}\Delta V_2(k) \;\; \leq \;\; & \Delta V_1(k) - \frac{\rho_2}{\bar{g}_2} e_2^2(k+\tau) - \frac{1}{\bar{g}_2} e_2^2(k) + \beta_2 - \sigma_2(1 - \sigma_2\bar{\gamma}_2 - \bar{g}_2\sigma_2\bar{\gamma}_2)\|\hat{W}_2(k)\|^2 \\ & - \sigma_2\|\tilde{W}_2(k)\|^2\end{aligned}$$

where

$$\rho_2 = 1 - \bar{\gamma}_2 - \bar{\gamma}_2 l_2 - \bar{g}_2\bar{\gamma}_2 l_2, \qquad \beta_2 = \frac{\bar{g}_2\epsilon_{z_2}^2}{\bar{\gamma}_2} + \sigma_2\|W_2^*\|^2$$

If we choose the design parameters as follows

$$\bar{\gamma}_2 < \frac{1}{1 + l_2 + \bar{g}_2 l_2}, \qquad \sigma_2 < \frac{1}{(1+\bar{g}_2)\bar{\gamma}_2} \tag{3.140}$$

then we have

$$\begin{aligned}\Delta V_2(k) \;\; & \leq \;\; \Delta V_1(k) - \frac{1}{\bar{g}_2} e_2^2(k) + \beta_2 \\ & \leq \;\; -\frac{1}{\bar{g}_1} e_1^2(k) + \beta_1 - \frac{1}{\bar{g}_2} e_2^2(k) + \beta_2\end{aligned}$$

Thus $\Delta V_2(k) \leq 0$ once the error $|e_i(k)|$ $(i = 1, 2)$ is larger than $\sqrt{\bar{g}_i(\beta_1 + \beta_2)}$. This implies the boundedness of $e_1(k)$ and $e_2(k)$. Furthermore, the tracking error $e_i(k)$ $(i = 1, 2)$ will asymptotically converge to the compact set denoted by $\Omega_2 \subset R$, where $\Omega_2 \triangleq \{\chi \Big| \chi \leq \max\Big\{\sqrt{\bar{g}_1(\beta_1+\beta_2)}, \sqrt{\bar{g}_2(\beta_1+\beta_2)}\Big\}\}$.

The adaptation dynamics (3.120) can be written as

$$\begin{aligned}\hat{W}_2(k+\tau) \;\; & = \;\; (I - \Gamma_2\sigma_2)\hat{W}_2(k) - \Gamma_2[S_2(z_2(k))e_2(k+\tau) + \sigma_2 W_2^*] \\ & = \;\; A_2(k)\hat{W}_2(k) - \Gamma_2[S_2(z_2(k))e_2(k+\tau) + \sigma_2 W_2^*]\end{aligned}$$

Because $0 < \gamma_{2_1}, \gamma_{2_2}, \ldots, \gamma_{2_n} < 1$ and $0 < \sigma_2 < 1$, we know that the transition matrix $\|\Phi(k_1, k_0)\|$ of $A_2(k)$ always satisfies $\|\Phi(k_1, k_0)\| < 1$. Furthermore, noting $S_2(z_2(k))$, $e_2(k+\tau)$ and $\sigma_2 W_2^*$ are all bounded, by applying Lemma A.1, $\hat{W}_2(k)$ is bounded in a compact set denoted by Ω_{w_2}, and hence the boundedness of $\hat{W}_2(k)$ is assured.

Step $i(1 < i < n)$: Following the same procedures in *Step 1* or *Step 2*, for $e_i(k) = y_i(k) - y_{d_i}(k)$, its τth difference is given by

$$\begin{aligned}e_i(k+\tau) \;\; & = \;\; y_i(k+\tau) - y_{d_i}(k+\tau) \\ & = \;\; f_i(k, \bar{u}_{i-1}(k)) + g_i(k)u_i(k) - y_{d_i}(k+\tau)\end{aligned} \tag{3.141}$$

Adding and subtracting $g_i(k)u_i^*(k)$ on the right side of equation (3.141) and noting equation (3.121), we have

$$\begin{aligned} e_i(k+\tau) &= g_i(k)\left(u_i(k)-u_i^*(k)\right) \\ &= g_i(k)\left[\tilde{W}_i^T(k)S_i(z_i(k))-\epsilon_{z_i}\right] \end{aligned}$$

with $\tilde{W}_i^T(k)=\hat{W}_i^T(k)-W_i^*(k)$ denotes the estimation error of the NN weight $W_i^*(k)$. Consequently, we obtain

$$\tilde{W}_i^T(k)S_i(z_i(k))=\frac{e_i(k+\tau)}{g_i(k)}+\epsilon_{z_i} \tag{3.142}$$

Similarly, choosing the following Lyapunov function candidate for subsystems Σ_1 to Σ_i

$$V_i(k)=\sum_{j=1}^{i-1}V_j(k)+\frac{1}{\bar{g}_i}\sum_{j=0}^{\tau-1}e_i^2(k+j)+\sum_{j=0}^{\tau-1}\tilde{W}_i^T(k+j)\Gamma_i^{-1}\tilde{W}_i(k+j) \tag{3.143}$$

By following the same procedure as in *step 1*, we have

$$\begin{aligned} \Delta V_i(k) \leq& \sum_{j=1}^{i-1}\Delta V_j(k)-\frac{\rho_i}{\bar{g}_i}e_i^2(k+\tau)-\frac{1}{\bar{g}_i}e_i^2(k)+\beta_i \\ &-\sigma_i(1-\sigma_i\bar{\gamma}_i-\bar{g}_i\sigma_i\bar{\gamma}_i)\|\hat{W}_i(k)\|^2-\sigma_i\|\tilde{W}_i(k)\|^2 \end{aligned}$$

where

$$\rho_i=1-\bar{\gamma}_i-\bar{\gamma}_il_i-\bar{g}_i\bar{\gamma}_il_i, \qquad \beta_i=\frac{\bar{g}_i\epsilon_{z_i}^2}{\bar{\gamma}_i}+\sigma_i\|W_i^*\|^2$$

If we choose the design parameters as follows

$$\bar{\gamma}_i<\frac{1}{1+l_i+\bar{g}_il_i}, \qquad \sigma_i<\frac{1}{(1+\bar{g}_i)\bar{\gamma}_i} \tag{3.144}$$

then we have

$$\Delta V_i(k) \leq \sum_{j=1}^{i}\left\{-\frac{1}{\bar{g}_j}e_j^2(k)\right\}+\sum_{j=1}^{i}\beta_j$$

Thus $\Delta V_i(k)\leq 0$ once $|e_j(k)|$ $(j=1,2,\ldots,i)$ is larger than $\sqrt{\bar{g}_j(\beta_1+\cdots+\beta_i)}$. This implies the boundedness of $e_j(k)$ $(j=1,2,\ldots,i)$. Furthermore, the tracking error

$e_j(k)$ $(j = 1, 2, \ldots, i)$ will asymptotically converge to the compact set denoted by $\Omega_i \subset R$, where

$$\Omega_i \triangleq \{\chi \Big| \chi \leq \max\left\{\sqrt{\bar{g}_1(\beta_1 + \cdots + \beta_i)}, \ldots, \sqrt{\bar{g}_i(\beta_1 + \cdots + \beta_i)}\right\}\}$$

By following the similar procedure as in *Step 1*, we know that $\hat{W}_i(k)$ is bounded in a compact set denoted by Ω_{w_i}, and hence the boundedness of $\hat{W}_i(k)$ is assured.

Step n: In the final step, following the same procedure as in *Step i*, we have the following Lyapunov function candidate (For clarity of presentation, details are omitted here)

$$V_n(k) = \sum_{j=1}^{n-1} V_j(k) + \frac{1}{\bar{g}_n}\sum_{j=0}^{\tau-1} e_n^2(k+j) + \sum_{j=0}^{\tau-1} \tilde{W}_n^T(k+j)\Gamma_n^{-1}\tilde{W}_n(k+j) \qquad (3.145)$$

Its first difference is

$$\begin{aligned} \Delta V_n(k) \;\; \leq \;\; & \sum_{j=1}^{n-1} \Delta V_j(k) - \frac{\rho_n}{\bar{g}_n} e_n^2(k+\tau) - \frac{1}{\bar{g}_n} e_n^2(k) + \beta_n - \\ & \sigma_n(1 - \sigma_n\bar{\gamma}_n - \bar{g}_n\sigma_n\bar{\gamma}_n)\|\hat{W}_n(k)\|^2 - \sigma_n\|\tilde{W}_n(k)\|^2 \end{aligned}$$

with

$$\rho_n = 1 - \bar{\gamma}_n - \bar{\gamma}_n l_n - \bar{g}_n\bar{\gamma}_n l_n, \qquad \beta_n = \frac{\bar{g}_n\epsilon_{z_n}^2}{\bar{\gamma}_n} + \sigma_n\|W_n^*\|^2$$

If we choose the design parameters as follows

$$\bar{\gamma}_n < \frac{1}{1 + l_n + \bar{g}_n l_n}, \qquad \sigma_n < \frac{1}{(1 + \bar{g}_n)\bar{\gamma}_n} \qquad (3.146)$$

then we have

$$\Delta V_n(k) \;\; \leq \;\; \sum_{j=1}^{n}\left\{-\frac{1}{\bar{g}_j} e_j^2(k)\right\} + \sum_{j=1}^{n}\beta_j$$

Thus $\Delta V_n(k) \leq 0$ once $e_j(k)$ $(j = 1, 2, \ldots, n)$ is larger than $\sqrt{\bar{g}_j(\beta_1 + \cdots + \beta_n)}$. This implies the boundedness of $e_j(k)$ $(j = 1, 2, \ldots, n)$. Furthermore, the tracking error $e_j(k)$ $(j = 1, 2, \ldots, n)$ will asymptotically converge to the compact set denoted by $\Omega_n \subset R$, where

$$\Omega_n \triangleq \left\{\chi \Big| \chi \leq \max\left\{\sqrt{\bar{g}_1(\beta_1 + \cdots + \beta_n)}, \ldots, \sqrt{\bar{g}_n(\beta_1 + \cdots + \beta_n)}\right\}\right\}$$

Following the procedures in previous steps, we know that $\hat{W}_n(k)$ is bounded in a compact set denoted by Ω_{w_n}, and hence the boundedness of $\hat{W}_n(k)$ is assured.

In summary, for the closed-loop nonlinear MIMO system consists of system (3.70), controller (3.129) and adaptive law (3.130), if the design parameters are chosen as

$$\bar{\gamma}_i < \frac{1}{1 + l_i + \bar{g}_i l_i}, \quad \sigma_i < \frac{1}{(1 + \bar{g}_i)\bar{\gamma}_i} \quad (i = 1, \ldots, n)$$

then the closed-loop system is semi-globally uniformly ultimately bounded, and has an equilibrium at $[e_1(k), \ldots, e_n(k)]^T = 0$. This guarantees that all the signals include the state vector $X(k)$, the control input $u(k)$ and NN weight estimates $\hat{W}_i(k)$, $i = 1, \ldots, n$ are all bounded. Subsequently,

$$\lim_{k \to \infty} \|y(k) - y_d(k)\| \leq \epsilon$$

where ϵ is a small positive number.

Therefore, for any a priori given (arbitrarily large) bounded set Ω and any a priori given (arbitrarily small) set Ω_0, which contains $(0, 0)$ as an interior point, there exist a control u, such that every trajectory of the closed-loop system starting from Ω enters the set Ω_0 in a finite time and remains in it thereafter. That is to say, the whole closed-loop system is SGUUB. ■

Remark 3.10 *In Theorem 3.3, by using neural network controls (3.129) and weights update laws (3.130), through Lyapunov analysis, we can only obtain the boundedness of the closed loop signals, include the states, the outputs and the neural network weights.*

3.2.4 Simulation

In order to illustrate the effectiveness of the proposed schemes, a simulation example is studied in this section. Considering the following MIMO discrete-time system with triangular form inputs

$$\begin{cases} x_{1,1}(k+1) &= f_{1,1}(\bar{x}_{1,1}(k)) + g_{1,1}(\bar{x}_{1,1}(k))x_{1,2}(k) \\ x_{1,2}(k+1) &= f_{1,2}(\bar{x}_{1,2}(k)) + g_{1,2}(\bar{x}_{1,2}(k))u_1(k) \\ x_{2,1}(k+1) &= f_{2,1}(\bar{x}_{2,1}(k)) + g_{2,1}(\bar{x}_{2,1}(k))x_{2,2}(k) \\ x_{2,2}(k+1) &= f_{2,2}(\bar{x}_{2,2}(k), u_1(k)) + g_{2,2}(\bar{x}_{2,2}(k))u_2(k) \\ y_1(k) &= x_{1,1}(k) \\ y_2(k) &= x_{2,1}(k) \end{cases}$$

where

$$\begin{cases} f_{1,1}(\bar{x}_{1,1}(k)) = \frac{x_{1,1}^2(k)}{1+x_{1,1}^2(k)}, & g_{1,1}(\bar{x}_{1,1}(k)) = 0.3 \\ f_{1,2}(\bar{x}_{1,2}(k)) = \frac{x_{1,1}^2(k)}{1+x_{1,2}^2(k)+x_{2,1}^2(k)+x_{2,2}^2(k)}, & g_{1,2}(\bar{x}_{1,2}(k)) = 1 \end{cases}$$

$$\begin{cases} f_{2,1}(\bar{x}_{2,1}(k)) = \frac{x_{2,1}^2(k)}{1+x_{2,1}^2(k)}, & g_{2,1}(\bar{x}_{2,1}(k)) = 0.2 \\ f_{2,2}(\bar{x}_{2,2}(k), u_1(k)) = \frac{x_{2,1}^2(k)}{1+x_{1,1}^2+x_{1,2}^2(k)+x_{2,2}^2(k)}u_1^2(k), & g_{2,2}(\bar{x}_{2,2}(k)) = 1 \end{cases}$$

The control objective is to drive the output $y(k) = [y_1(k), y_2(k)]^T$ of the system to follow desired reference signals

$$y_{d_1}(k) = 0.5 + \frac{1}{4}\cos(\frac{\pi T k}{4}) + \frac{1}{4}\sin(\frac{\pi T k}{2})$$

$$y_{d_2}(k) = 0.5 + \frac{1}{4}\sin(\frac{\pi T k}{4}) + \frac{1}{4}\sin(\frac{\pi T k}{2})$$

with $T = 0.01$.

The initial condition for system states is $x_{1,1}(0) = x_{1,1}(1) = 0.5$, $x_{1,2}(0) = x_{1,2}(1) = 0$, $x_{2,1}(0) = x_{2,1}(1) = 0.5$ and $x_{2,2}(0) = x_{2,2}(1) = 0$. The neurons used are $l_1 = 28$ and $l_2 = 36$. All the elements of the neural network weights $\hat{W}_1(0)$, $\hat{W}_1(1)$, $\hat{W}_2(0)$ and $\hat{W}_2(1)$ are initialized to be random numbers between 0.00 and 0.01, and the active functions $S_1(z_1(0))$, $S_1(z_1(1))$, $S_2(z_2(0))$ and $S_2(z_2(1))$ are initialized to be random numbers between 0.00 and 0.02. σ modification gains are $\sigma_1 = \sigma_2 = 0.01$, and adaptive gain matrices are $\Gamma_1 = \Gamma_2 = 0.015I$.

For clarity, the formulas used in the simulation are listed here. The practical controls are as follows:

$$\begin{aligned}
u_1(k) &= \hat{W}_1(k)S_1(z_1(k)) \\
z_1(k) &= [y_1(k-1), y_1(k), y_2(k-1), y_2(k), u_1(k-1), u_2(k-1), y_{d_1}(k+2)]^T \\
u_2(k) &= \hat{W}_2(k)S_2(z_2(k)) \\
z_2(k) &= [y_1(k-1), y_1(k), y_2(k-1), y_2(k), u_1(k-1), u_2(k-1), u_1(k), y_{d_2}(k+2)]^T
\end{aligned}$$

The errors' definitions are $(i = 1, 2)$:

$$\Sigma_i : e_i(k) = y_i(k) - y_{d_i}(k)$$

The weights update law are as follows $(i = 1, 2)$:

$$\hat{W}_i(k) = \hat{W}_i(k-2) - \Gamma_i[S_i(z_i(k-2))e_i(k) + \sigma_i W_i(k-2)]$$

Simulation results are shown in Figure 3.12-Figure 3.15. Figure 3.12 and Figure 3.13 show the tracking performances of the first sub-system and the second sub-system respectively. It can be seen that, in the initial period of simulation, the tracking errors are large. Then, as the time increases, the practical outputs converge to the neighborhoods of the desired signals. The control input trajectories $u_1(k) = \hat{W}_1(k)S_1(z_1(k))$ and $u_2(k) = \hat{W}_2(k)S_2(z_2(k))$ are shown in Figure 3.14. Their corresponding neural network weights norms $\|\hat{W}_1(k)\|$ and $\|\hat{W}_2(k)\|$ are shown in Figure 3.15. From Figure 3.14 and 3.15, we can see that both the control inputs and their corresponding weights norms are all bounded.

3.3 Conclusion

In this chapter, firstly, neural network control scheme was investigated for a class of MIMO nonlinear discrete-time systems with disturbances. In order to avoid the non-causal problem in backstepping design, the MIMO system under study was firstly transformed into SDFC form, which completely solved the non-causal problem. Then, HONNs were used to approximate the desired controls. By using backstepping design in a nested manner, the closed-loop system was proved to be SGUUB based on Lyapunov analysis. Secondly, a simple output feedback NN control scheme was developed for a class of similar MIMO nonlinear discrete-time systems without disturbances. By coordinate transformation, the system was firstly transformed into input output description. Then the input and output sequences were used to construct the effective neural network control. HONNs were used to approximate the desired controls. The closed-loop system was proved to be SGUUB based on Lyapunov analysis.

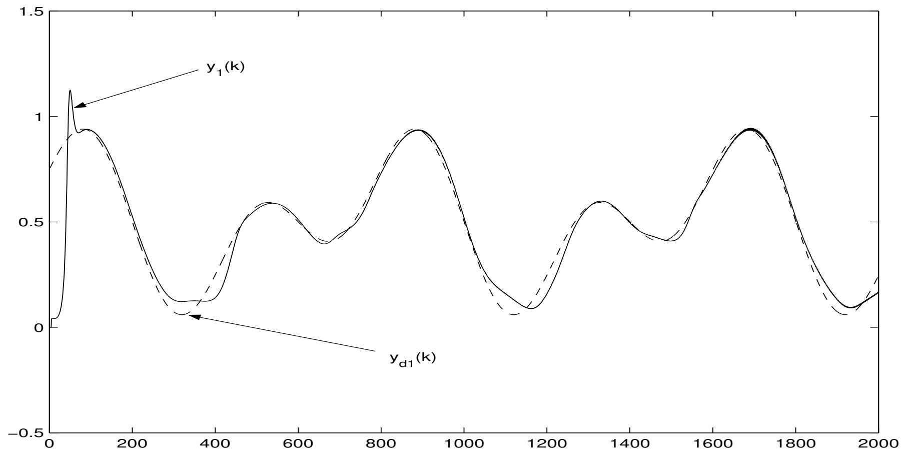

Figure 3.7: State Feedback Control - Tracking Performance $y_1(k)$ and $y_{d_1}(k)$

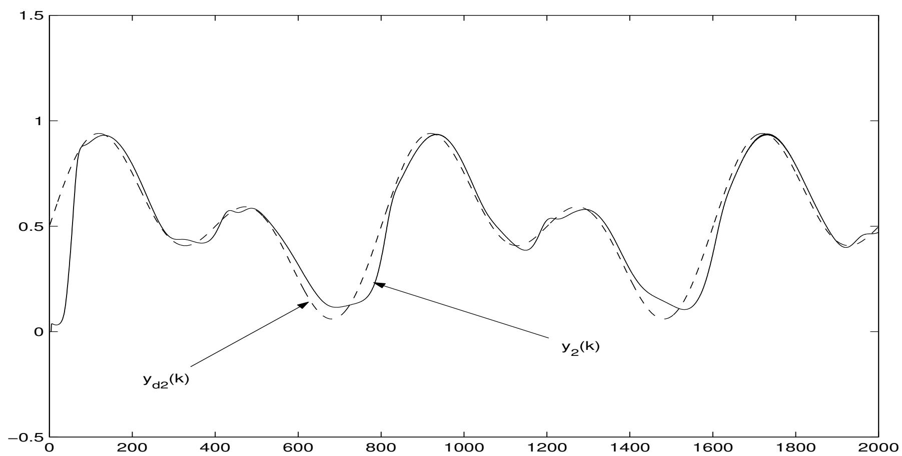

Figure 3.8: State Feedback Control - Tracking Performance $y_2(k)$ and $y_{d_2}(k)$

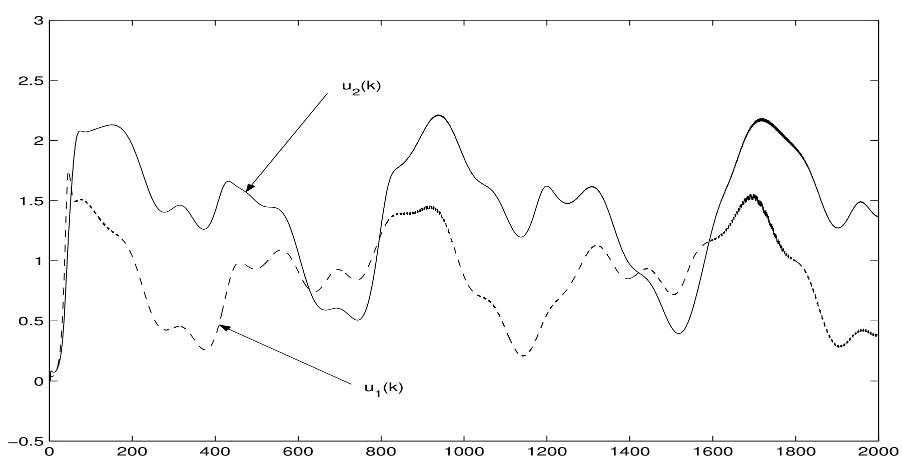

Figure 3.9: State Feedback Control - Control Inputs $u_1(k)$ and $u_2(k)$

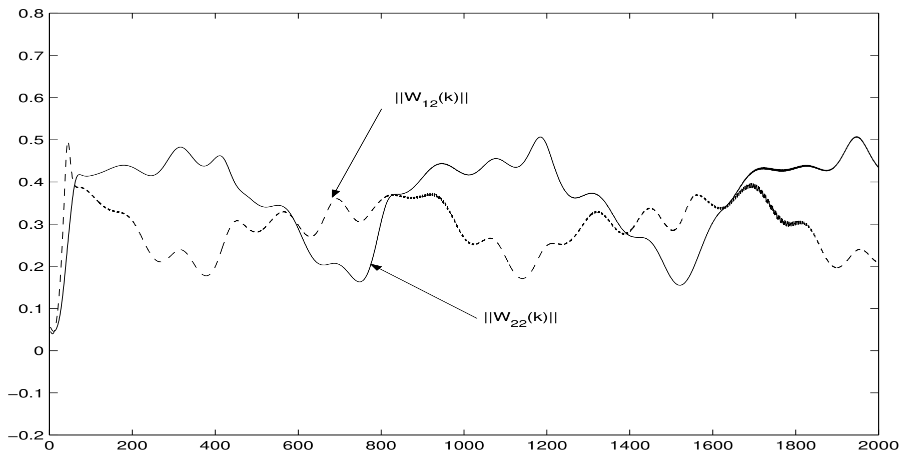

Figure 3.10: State Feedback Control - Weight Norms $\|\hat{W}_{12}(k)\|$ and $\|\hat{W}_{22}(k)\|$

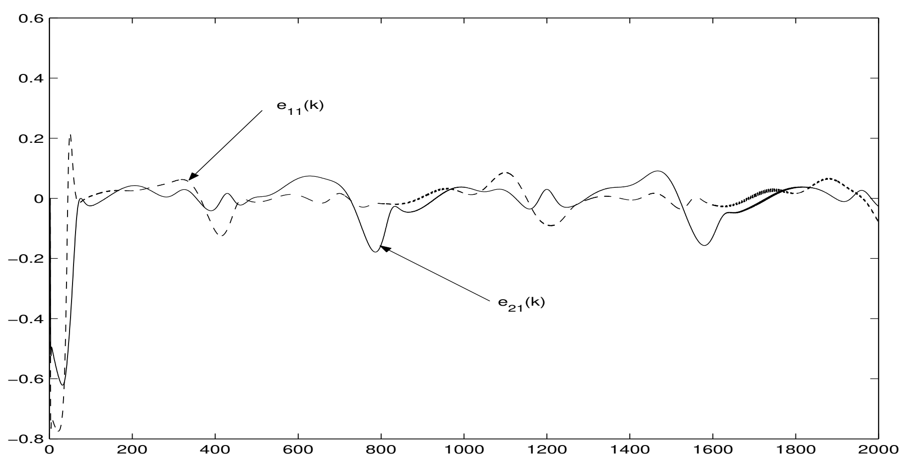

Figure 3.11: State Feedback Control - Error dynamics

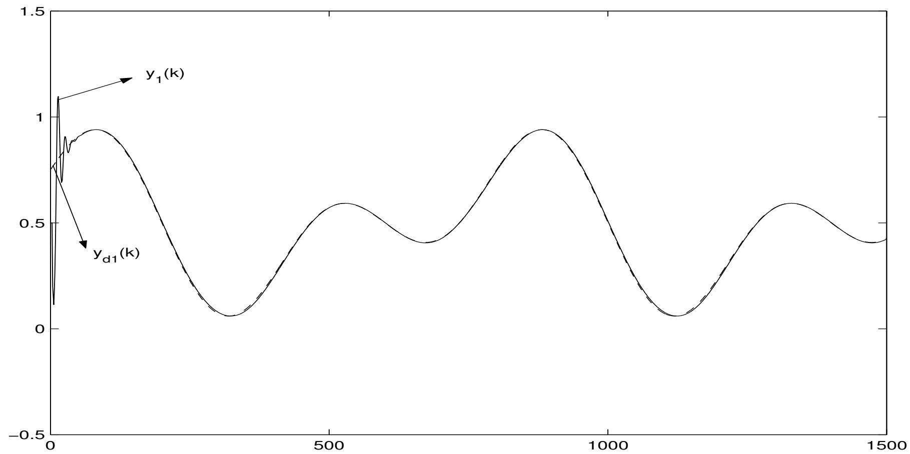

Figure 3.12: Output Feedback Control - Tracking Performance $y_1(k)$ and $y_{d_1}(k)$

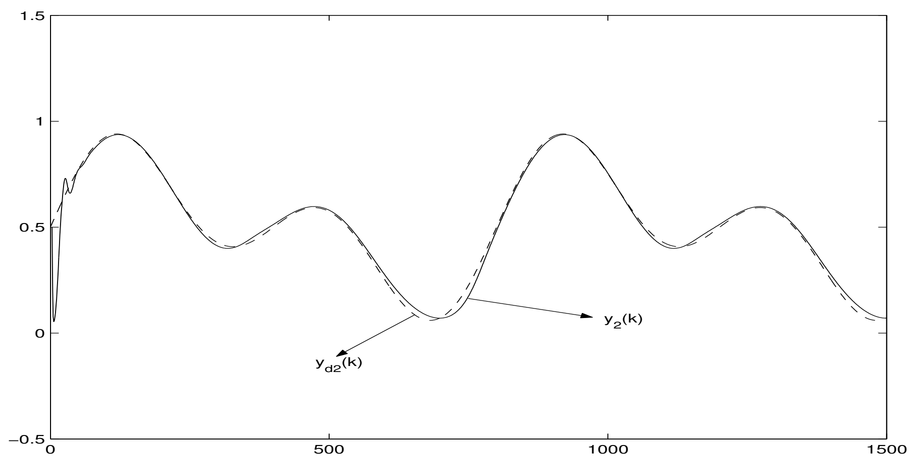

Figure 3.13: Output Feedback Control - Tracking Performance $y_2(k)$ and $y_{d_2}(k)$

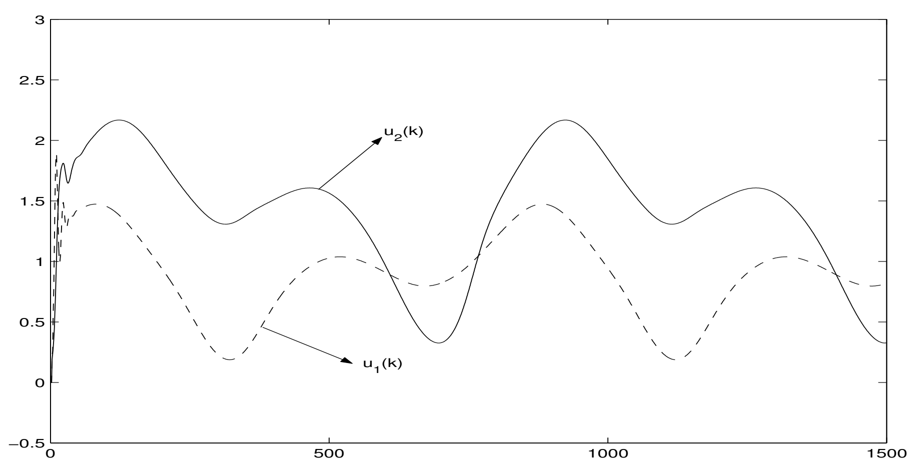

Figure 3.14: Output Feedback Control - Control Inputs $u_1(k)$ and $u_2(k)$

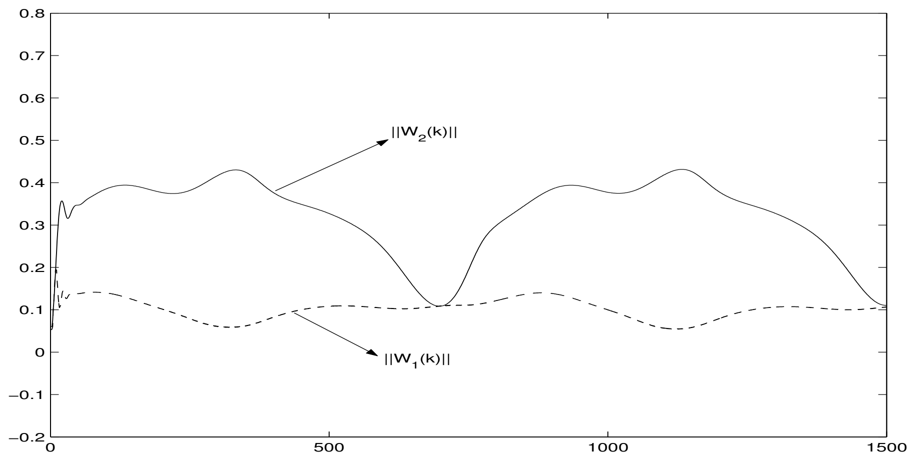

Figure 3.15: Output Feedback Control - Weight Norms $\|\hat{W}_1(k)\|$ and $\|\hat{W}_2(k)\|$

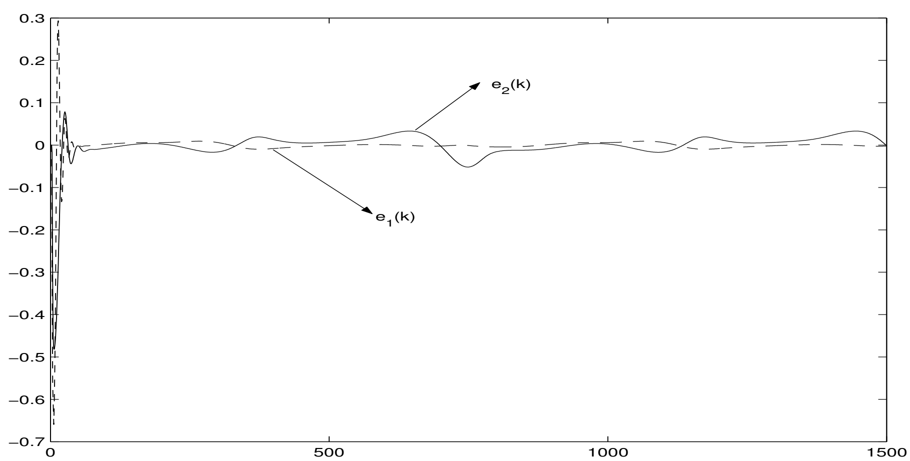

Figure 3.16: Output Feedback Control - Error dynamics

Chapter 4

NN Control of NARMAX MIMO Systems

In this chapter, adaptive NN control schemes are investigated for MIMO NARMAX systems. The chapter is organized as follows. Firstly, for a class of MIMO NARMAX systems in affine form, a simple and effective NN control scheme is proposed in Section 4.1. Subsequently, for a class of MIMO NARMAX non-affine systems, by using implicit function theory, another NN control scheme is developed in Section 4.2. Finally, conclusions are made in Section 4.3.

4.1 Affine MIMO NARMAX Systems

4.1.1 Introduction

In this section, using HONNs, adaptive controller design is investigated for a class of affine discrete-time MIMO nonlinear systems with unknown interconnections between subsystems. The controller can be applied directly to the system without the requirement of off-line training if the node number of the neural networks is sufficient large. By finding an orthogonal matrix to tune the NN weight matrix, the overall system is proved to be SGUUB, and the tracking error converges to a small neighborhood of the origin.

This section is organized as follows. Section 4.1.2 describes the nonlinear systems under study and the control objective, as well as some stability notions. An ideal control is also presented if there are no uncertainties. A direct NN controller is proposed in Section 4.1.3, which guarantees the stability of the closed-loop system and the boundedness of all signals in the closed-loop system.

4.1.2 System Dynamics and Stability Notions

In discrete-time formulations, one of the most popular nonlinear representations is the NARMAX model [101]. Many $p \times p$ multi-inputs and multi-outputs processes can be represented by a NARMAX model known as τ-step ahead observer equation as follows

$$\begin{aligned} y(k+\tau) &= F_\tau(Y(k), U_{k-1}(k), D_{k-1}(k), \bar{d}(k)) + G_\tau(Y(k), U_{k-1}(k))u(k) \\ &\quad + d(k+\tau-1) \end{aligned} \tag{4.1}$$

where τ is system delay, $y(k) = [y_1(k), \ldots, y_p(k)]^T$ and $u(k) = [u_1(k), \ldots, u_p(k)]^T$ are system output and input respectively, $d(k) = [d_1(k), d_2(k), \ldots, d_p(k)]^T$ denotes the external unmeasured disturbance vector bounded by a known constant $d_0 > 0$, i.e., $\|d(k)\| \le d_0$, $Y(k)$ is a vector containing current and past outputs, $U_{k-1}(k)$ is a vector containing only past inputs, and $D_{k-1}(k)$ is a vector containing the past disturbances, $F_\tau(*)$ is a nonlinear function vector, and $G_\tau(*)$ is a nonlinear function matrix. In particular, they are defined as

$$\begin{aligned} \bar{d}(k) &= [d(k+\tau-2), \ldots, d(k)]^T, \quad \text{if} \quad \tau \ge 2 \\ U_{k-1}(k) &= [u_1(k-1), \ldots, u_1(k-m_1), u_2(k-1), \ldots, u_2(k-m_2), \ldots, \\ &\quad u_p(k-1), \ldots, u_p(k-m_p)]^T \\ Y(k) &= [y_1(k), \ldots, y_1(k-n_1+1), y_2(k), \ldots, y_2(k-n_2+1), \ldots, \\ &\quad y_p(k), \ldots, y_p(k-n_p+1)]^T \\ D_{k-1}(k) &= [d_1(k-1), \ldots, d_1(k-t_1+1), d_2(k-1), \ldots, d_2(k-t_2+1), \ldots, \\ &\quad d_p(k-1), \ldots, d_p(k-t_p+1)]^T \\ F_\tau(k) &= [f_{\tau_i}(Y(k), U_{k-1}(k), D_{k-1}(k), \bar{d}(k))] \in R^p \\ G_\tau(k) &= [g_{\tau_{ij}}(Y(k), U_{k-1}(k))] \in R^{p \times p} \end{aligned}$$

with n_i denotes the length of the i-th subsystem's outputs, and m_i is the length of the i-th subsystem's inputs, which satisfies $m_i < n_i$, $i = 1, \ldots, p$; t_i is the length of the i-th disturbance, $i = 1, \ldots, p$; $f_{\tau_i}(Y(k), U_{k-1}(k), D_{k-1}(k), \bar{d}(k))$ and $g_{\tau_{ij}}(Y(k), U_{k-1}(k))$, $i, j = 1, \ldots, p$, are smooth nonlinear functions.

Assumption 4.1 *$F_\tau(Y(k), U_{k-1}(k), D_{k-1}(k), \bar{d}(k))$ is locally Lipschitz in $\bar{d}(k)$ and $D_{k-1}(k)$ at $(0,0)$, i.e., there are Lipschitz constants L_1 and L_2 such that*

$$\begin{aligned} &\|F_\tau(Y(k), U_{k-1}(k), D_{k-1}(k), \bar{d}(k)) - F_\tau(Y(k), U_{k-1}(k), 0, 0)\| \\ &\qquad\qquad\qquad \leq L_1 \|D_{k-1}(k)\| + L_2 \|\bar{d}(k)\| \end{aligned}$$

with L_1 and L_2 being positive constants.

Suppose the objective is to design control $u(k)$ to drive the system output $y(k)$ follow a known and bounded trajectory $y_d(k) = [y_{d_1}(k), y_{d_2}(k), \ldots, y_{d_p}(k)]^T$.

Definition 4.1 *The future outputs, $y(k+i)$, $i > 0$, of discrete-time system (4.1) are said to be semi-determined future outputs, if the future outputs are independent of the current control $u(k)$.*

From Definition 4.1, it is clear that future outputs $y(k+1)$,. . ., $y(k+\tau-1)$ in (4.1) are all *semi-determined future outputs* as they are independent of the current control $u(k)$ though they are influenced by the unknown external disturbances of the past and the future. As the external disturbances are unknown, their effects could not be cancelled through control action. Thus, we are interested in designing robust control for (4.1) by using the results for the ideal case when the unknown disturbances are isolated.

Assumption 4.2 *The desired trajectory $y_d(k) \in \Omega_{yd} \subset R^p$, $\forall k > 0$ is smooth and known, where Ω_{yd} is a small subset of Ω_y and $\Omega_y \triangleq \{\chi(k) | \chi(k) = y(k)\} \subset R^p$.*

Define error vector $e(k) = y(k) - y_d(k) = [e_1(k), e_2(k), \ldots, e_p(k)]^T$. Noting equation (4.1), the error equation of $e(k)$ can then be written as

$$e(k+\tau) = F_\tau(k) - y_d(k+\tau) + G_\tau(k)u(k) + d(k+\tau-1) \tag{4.2}$$

Definition 4.2 *The solution of (4.2) is semi-globally uniformly ultimately bounded (SGUUB), if for any Ω_y and Ω_u, compact subsets of R^p and all $y(k_0 - i) \in \Omega_y$, $i = 0, \ldots, \max\{n_1, \ldots, n_p\} - 1$, $u(k_0 - j) \in \Omega_u, j = 1, \ldots, \tau + \max\{m_1, \ldots, m_p\}$, and all semi determined future outputs are in Ω_y, there exist an $\epsilon > 0$, and a number N such that $\|e(k)\| < \epsilon$ for all $k \geq k_0 + N$.*

Error dynamics (4.2) can be written as

$$\begin{aligned} e(k+\tau) &= F(Y(k), U_{k-1}(k)) - y_d(k+\tau) + G_\tau(Y(k), U_{k-1}(k))u(k) \\ &\quad + \Delta F(k) + d(k+\tau-1) \end{aligned} \tag{4.3}$$

where

$$F(Y(k), U_{k-1}(k)) = F_\tau(Y(k), U_{k-1}(k), 0, 0)$$

$$\Delta F(k) = F_\tau(Y(k), U_{k-1}(k), D_{k-1}, \bar{d}(k)) - F(Y(k), U_{k-1}(k))$$

We can see that $\Delta F(k)$ is generated by the external disturbances. By Assumption 4.1 and the boundedness of disturbances, we can conclude that $\Delta F(k) \leq L_1\|D_{k-1}(k)\| + L_2\|\bar{d}(k)\|$ is bounded.

If $F(Y(k), U_{k-1}(k))$ and $G_\tau(Y(k), U_{k-1}(k))$ are known and $G_\tau^{-1}(Y(k), U_{k-1}(k))$ exists, then we can choose the ideal control input as

$$u^*(k) = G_\tau^{-1}(Y(k), U_{k-1}(k))[y_d(k+\tau) - F(Y(k), U_{k-1}(k))] \tag{4.4}$$

Thus, we have the closed-loop error equation

$$\|e(k+\tau)\| = \|\Delta F(k) + d(k+\tau-1)\| \leq L_1\|D_{k-1}(k)\| + L_2\|\bar{d}(k)\| + \|d(k+\tau-1)\|$$

If there are no disturbances, i.e., $D_{k-1}(k) = 0$ and $d(k+\tau-1) = 0$, we have $e(k+\tau) = 0$, which is achieved in τ steps. Under this condition, the desired control, $u^*(k)$, is the so-called τ-step deadbeat control, or *exact tracking* control, which is well defined and has been proven to be unique in [95]. In practice, $u^*(k)$ is not realizable as $F(k)$ and $G_\tau(k)$ are unknown. In the following, adaptive neural networks shall be used to approximate the unknown desired control $u^*(k)$, which is introduced for analytical purpose only. Note that saturation control is out of the scope of the technical notes, and further research will be carried out for other types of ideal controls rather than deadbeat control.

Remark 4.1 *It is obvious that if there is no disturbances in the system, i.e., $d(k) = 0$ and $D_{k-1}(k) = 0$, the tracking error $e(k+\tau) = 0$. If $D_{k-1}(k) \neq 0$ and $d(k) \neq 0$, the error equation is $e(k+\tau) = \Delta F(k) + d(k+\tau-1)$, thus, exact tracking cannot be obtained though bounded due to Assumption 4.1. Instead, we propose SGUUB stability of the system in the presence of the unknown bounded disturbances.*

Assumption 4.3 *The desired control $u^*(k)$ is within the compact set $\Omega_{u^*} \subset \Omega_u$, $\forall y(k-i) \in \Omega_y \subset R^p, i = 0, \ldots, \max\{n_1, \ldots, n_p\} - 1$ and $\forall u(k-j) \in \Omega_u \subset R^p, j = 1, \ldots, \tau + \max\{m_1, \ldots, m_p\}$.*

The desired trajectory is assumed to be chosen such that the system can achieve since it is meaningless to ask the system to track an unrealistic trajectory. Assumption 4.3 is only introduced for mathematical rigor (stating that the desired control u^* is within the capability of the control system) as the boundedness of the actual control $u(k)$ is establish via Lyapunov analysis later.

By examining expression (4.4), the desired control input $u^*(k)$ is a function of $Y(k)$, $U_{k-1}(k)$ and $y_d(k+\tau)$. Thus, there exist ideal weights W^* such that the smooth function vector $u^*(k)$ can be approximated by an ideal NN on a compact set $\Omega_z \subset R^q$

$$u^*(k) = W^{*T} S(\bar{z}(k)) + \epsilon_z \tag{4.5}$$

where

$$\begin{aligned} \bar{z}(k) &= \begin{bmatrix} Y(k) \\ U_{k-1}(k) \\ y_d(k+\tau) \end{bmatrix} \in \Omega_z \subset R^q, \qquad q = \textstyle\sum_{i=1}^{p}(n_i + m_i + 1) \\ \epsilon_z &= [\epsilon_{z_1}, \ldots, \epsilon_{z_p}]^T \end{aligned}$$

and ϵ_z is the bounded NN approximation error vector satisfying $\|\epsilon_z\| \le \epsilon_0$ on the compact set, which can be reduced by increasing the number of the adjustable weights. The ideal weight matrix W^* is an "artificial" quantity required for analytical purpose, and is defined as that minimizes $\|\epsilon_z\|$ for all $\bar{z} \in \Omega_z \subset R^q$ in a compact region, i.e.,

$$W^* \triangleq arg \min_{W \in \Omega_w} \left\{ \sup_{z \in \Omega_z} |u^*(k) - W^T S(\bar{z}(k))| \right\} \tag{4.6}$$

$$\Omega_z \subset R^q \;\text{ and compact set }\; \Omega_w \subset R^{l \times p}$$

In general, the ideal NN weight matrix, W^*, is unknown though constant, its estimate, $\hat{W}$, should be used for controller design which will be discussed in Section 4.1.3.

Though HONN is used for analysis, other linear-in-parameters function approximators such as polynomials, splines, fuzzy systems and wavelet networks, among others, can also be used to construct the controller without any difficulty.

4.1.3 Controller Design and Stability Analysis

In this section, we present the robust adaptive NN controller for (4.1) under some mild conditions.

Assumption 4.4 *For system (4.1), assume $G_\tau(k)$ is a full rank matrix, and there exists an orthogonal matrix $Q(k) \in R^{p\times p}$, such that the eigenvalues of $Q(k)G_\tau^{-1}(k)$ are upper and lower bounded by $0 < \frac{b}{(1-\sigma\gamma)} \leq \lambda\{Q(k)G_\tau^{-1}(k)\} \leq a$, where a and b are constant numbers, $\sigma > 0$, $\gamma > 0$ and $0 < \sigma\gamma < 1$ (γ is the adaptation gain and σ is a positive constant indicates the leakage term of σ-modification used in weight update and $\lambda\{M\}$ denotes the eigenvalue of M).*

Remark 4.2 *If $G_\tau(k)$ is totally unknown, there is no valid method to construct such a $Q(k)$. However, if we known some properties of $G_\tau(k)$, then we may select such a $Q(k)$ that satisfies the requirement. For example, if all the eigenvalues of $G_\tau(k)$ are larger than zero, then we can select identity matrix $Q(k) = I$; if all the eigenvalues of $G_\tau(k)$ are less than zero, then we can choose $Q(k) = -I$. In practice, there are some physical systems possessing such a nice property, which include rigid robotic arms, and flexible joint robots, where the input matrix $G_\tau = M^{-1}(q)$, $0 < \alpha_1 I \leq M(q) \leq \alpha_2 I$ with q denotes the coordinates, M denotes the inertia matrix and constants α_1 and $\alpha_2 > 0$.*

Once we find such an orthogonal matrix $Q(k)$, we are ready to present the direct adaptive controller and the weights updating law as

$$\begin{aligned} u(k) &= \hat{W}^T(k)S(\bar{z}(k)) \qquad (4.7)\\ \hat{W}(k+1) &= \hat{W}(k-\tau+1) \end{aligned}$$

$$-\Gamma[S(\bar{z}(k-\tau+1))e^T(k+1)Q(k-\tau+1)+\sigma\hat{W}(k-\tau+1)] \quad (4.8)$$

where $\Gamma = \gamma I$ is a diagonal matrix with $\gamma > 0$, σ is a positive constant number, $\hat{W}(k) \in R^{p\times l}$ and $S(\bar{z}(k)) \in R^l$. The σ-modification is used here to eliminate the need of persistent exciting (PE) condition for parameter convergence. In comparison with the standard parameter adaptation algorithms, it should be noted that parameter adaptation algorithm (4.8) is of τ steps ahead in order to solve the control problem of general τ order nonlinear systems. In fact, the current estimate, $\hat{W}(k)$, is deviated from the estimate, $\hat{W}(k-\tau)$, of τ steps earlier rather than that of the previous step.

Substituting controller (4.7) into (4.3), the error equation (4.3) can be re-written as

$$\begin{aligned} e(k+\tau) &= F(Y(k),U_{k-1}) - y_d(k+\tau) + G_\tau(k)\hat{W}^T(k)S(\bar{z}(k)) \\ &\quad +\Delta F(k) + d(k+\tau-1) \end{aligned} \quad (4.9)$$

Adding and subtracting $G_\tau(k)u^*(\bar{z}(k))$ on the right side of (4.9) and noting (4.5), we have

$$\begin{aligned} e(k+\tau) &= F(Y(k),U_{k-1}) - y_d(k+\tau) + G_\tau(k)u^*(k) \\ &\quad +G_\tau(k)[\hat{W}^T(k)S(\bar{z}(k)) - W^{*T}S(\bar{z}(k)) - \epsilon_z] + \Delta F(k) + d(k+\tau-1) \end{aligned}$$

Substituting (4.4) into (4.10) leads to

$$e(k+\tau) = G_\tau(k)[\tilde{W}^T(k)S(\bar{z}(k)) - \epsilon_z] + D(k) \quad (4.10)$$

where $\tilde{W}(k) = \hat{W}(k) - W^*$ and $D(k) = \Delta F(k) + d(k+\tau-1)$. Since $\Delta F(k)$, $d_{k+\tau-1}$ are due to the existence of external disturbances and they are bounded, we can consider that $D(k)$ is bounded by a positive constant D_0, i.e. $\|D(k)\| < D_0$.

Control input (4.7) can be rewritten as

$$u(k) = (\tilde{W}(k) + W^*)^T S(\bar{z}(k)) = u^*(k) + \tilde{W}^T(k)S(\bar{z}(k)) - \epsilon_z$$

Due to $u^*(k) \in \Omega_{u^*}$ and Ω_{u^*} is a subset of Ω_u under Assumption 4.3, there must exist a nonzero compact set $\Omega_w \subset R^{l\times p}$ such that any $\tilde{W}(k) \in \Omega_w$ guarantees $u(k) \in \Omega_u$. Since Ω_{yd} is a small subset of Ω_y under Assumption 4.2, there must exist a large enough compact set $\Omega_e \subset R^p$, such that for any $e(k) \in \Omega_e$ guarantees that $y(k) \in \Omega_y$.

Theorem 4.1 *Consider the closed-loop system consisting of system (4.2), controller (4.7) and adaptation law (4.8). There exist compact sets* $\Omega_{y0} \subset \Omega_y$, $\Omega_{w0} \subset \Omega_w$ *and positive constants* l^*, γ^* *and* σ^* *such that if*

(i) Assumptions 4.2-4.4 being satisfied, the condition at time instant k_0 *is initialized as*

$$\begin{aligned} y(k_0 - j) &\in \Omega_{y0}, \quad j = 0, \ldots, \max\{n_1, \ldots, n_p\} - 1 \\ u(k_0 - j) &\in \Omega_u, \quad j = 1, \ldots, \tau + \max\{m_1, \ldots, m_p\} \\ \tilde{W}(k_0 - j) &\in \Omega_{w0}, \quad j = 0, \ldots, \tau - 1 \end{aligned}$$

(ii) the semi determined future outputs at time instant k_0, $y(k_0+1), \ldots, y(k_0+\tau-1)$ *are all in compact set* Ω_y, *and*

(iii) the design parameters are suitably chosen such that $l > l^*$, $\sigma < \sigma^*$ *and* $\gamma < \gamma^*$ *with* γ *being the eigenvalue of* Γ,

then, the closed-loop system is SGUUB.

Proof: We have illustrated that there exists an ideal control $u^*(k)$ which guarantees that $e(k + \tau) = 0$ if there is no unknown disturbance. Since all the assumptions are only valid in compact set Ω_y and Ω_u, we must prove that the system outputs and inputs will remain in these compact sets all the time indeed. At time instant k, suppose that all past inputs are in Ω_u, current output and all past outputs are in Ω_y, the *semi determined future outputs*, $y(k + 1), \ldots, y(k + \tau - 1)$, are all in Ω_y, all past NN weight errors are in Ω_w, we will prove that all these conditions still hold after time instant k and the tracking error converges into a small neighborhood of zero.

Choose the Lyapunov function candidate as

$$J(k) = b \sum_{j=0}^{\tau-1} tr\{e(k + j)e^T(k + j)\} + \sum_{j=0}^{\tau-1} tr\{\tilde{W}^T(k + j)\Gamma^{-1}\tilde{W}(k + j)\} \qquad (4.11)$$

where b is a positive constant, defined in Assumption 4.4. Apparently, the Lyapunov function candidate $J(k)$ contains the states of the error dynamics of the systems (4.10), and the parameter adaptation (4.8). Note that the future variables, $e(k + 1), \ldots, e(k + \tau - 1)$ and $\tilde{W}(k + 1), \ldots \tilde{W}(k + \tau - 1)$, are all semi-determined at time instant k as they are independent of current control $u(k)$. We have shown that $y(k+\tau-1), \ldots, y(k+1)$ are all independent of $u(k)$, so are $e(k+\tau-1), \ldots, e(k+1)$. For the same reason, it can be shown that $\tilde{W}(k+\tau-1), \ldots, \tilde{W}(k+1)$ are all determined at time instant k. For example,

$$\tilde{W}(k+\tau-1) = \tilde{W}(k-1) - \Gamma\left[S(\bar{z}(k-1))e^T(k+\tau-1)Q(k-1) + \sigma\hat{W}(k-1)\right]$$

is uniquely determined since (i) $e^T(k+\tau-1)$ is semi-determined, and (ii) all other signals are well defined at time instant k.

The first difference of (4.11) along (4.7), (4.8) and (4.10) is given by

$$\begin{aligned}
\Delta J(k) &= be^T(k+\tau)e(k+\tau) - be^T(k)e(k) \\
&\quad + tr\{\tilde{W}^T(k+\tau)\Gamma^{-1}\tilde{W}(k+\tau)\} - tr\{\tilde{W}^T(k)\Gamma^{-1}\tilde{W}(k)\} \\
&= be^T(k+\tau)e(k+\tau) - be^T(k)e(k) - 2\sigma tr\{\tilde{W}^T(k)\hat{W}(k)\} \\
&\quad + \sigma^2 tr\{\hat{W}^T(k)\Gamma\hat{W}(k)\} - 2tr\{\tilde{W}^T(k)S(\bar{z}(k))e^T(k+\tau)Q(k)\} \\
&\quad + 2\sigma tr\{\hat{W}^T(k)\Gamma S(\bar{z}(k))e^T(k+\tau)Q(k)\} \\
&\quad + tr\{Q^T(k)e(k+\tau)S^T(\bar{z}(k))\Gamma S(\bar{z}(k))e^T(k+\tau)Q(k)\}
\end{aligned}$$

Noting that

$$-2\sigma tr\{\tilde{W}^T(k)\hat{W}(k)\} = -\sigma\|\tilde{W}\|_F^2 - \sigma\|\hat{W}\|_F^2 + \sigma\|W^*\|_F^2$$

$$\sigma^2 tr\{\hat{W}^T(k)\Gamma\hat{W}(k)\} = \sigma^2\gamma\|\hat{W}\|_F^2$$

$$-2tr\{\tilde{W}^T(k)S(\bar{z}(k))e^T(k+\tau)Q(k)\} = -2e^T(k+\tau)Q(k)\tilde{W}^T(k)S(\bar{z}(k))$$

$$2\sigma tr\{\hat{W}^T(k)\Gamma S(\bar{z}(k))e^T(k+\tau)Q(k)\} = 2\sigma\gamma e^T(k+\tau)Q(k)\hat{W}^T(k)S(\bar{z}(k))$$

$$tr\{Q^T(k)e(k+\tau)S^T(\bar{z}(k))\Gamma S(\bar{z}(k))e^T(k+\tau)Q(k)\} = S^T(\bar{z}(k))\Gamma S(\bar{z}(k))e^T(k+\tau)e(k+\tau)$$

$$Q(k)Q^T(k) = Q^T(k)Q(k) = I$$

We can obtain

$$\begin{aligned}
\Delta J(k) = {} & be^T(k+\tau)e(k+\tau) - be^T(k)e(k) - \sigma\|\tilde{W}\|_F^2 - \sigma(1-\sigma\gamma)\|\hat{W}\|_F^2 + \sigma\|W^*\|_F^2 \\
& - 2e^T(k+\tau)Q(k)\tilde{W}^T(k)S(\bar{z}(k)) + 2\sigma\gamma e^T(k+\tau)Q(k)\hat{W}^T(k)S(\bar{z}(k)) \\
& + S^T(\bar{z}(k))\Gamma S(\bar{z}(k))e^T(k+\tau)e(k+\tau)
\end{aligned}$$

Noticing equation (4.10), we can obtain

$$\tilde{W}^T S(\bar{z}(k)) = G_\tau^{-1}(k)[e(k+\tau) - D(k)] + \epsilon_z$$

$$\hat{W}^T S(\bar{z}(k)) = G_\tau^{-1}(k)[e(k+\tau) - D(k)] + \epsilon_z + W^{*T} S(\bar{z}(k))$$

Thus

$$\begin{aligned}\Delta J(k) &= be^T(k+\tau)e(k+\tau) - be^T(k)e(k) - \sigma\|\tilde{W}\|_F^2 - \sigma(1-\sigma\gamma)\|\hat{W}\|_F^2 \\ &\quad +\sigma\|W^*\|_F^2 - 2e^T(k+\tau)Q(k)G_\tau^{-1}(k)e(k+\tau) + 2e^T(k+\tau)\beta(k) \\ &\quad +2\sigma\gamma e^T(k+\tau)Q(k)G_\tau^{-1}(k)e(k+\tau) + 2\sigma\gamma e^T(k+\tau)\alpha(k) \\ &\quad +S^T(\bar{z}(k))\Gamma S(\bar{z}(k))e^T(k+\tau)e(k+\tau)\end{aligned}$$

where

$$\begin{aligned}\alpha(k) &\triangleq Q(k)[-G_\tau^{-1}(k)D(k) + \epsilon_z + W^{*T}S(\bar{z}(k))] \\ \beta(k) &\triangleq Q(k)[-G_\tau^{-1}(k)D(k) + \epsilon_z]\end{aligned}$$

Since ϵ_z, $D(k)$ and $W^{*T}S(\bar{z}(k))$ are all bounded, it is reasonable to assume that both $\alpha(k)$ and $\beta(k)$ are bounded. For convenience of analysis, let $\alpha_i(k) \leq \alpha_{0i}$ and $\beta_i(k) \leq \beta_{0i}$, where α_{0i} and β_{0i} denote the i-th elements of constant vectors α_0 and β_0 respectively, which are only introduced to establish the stability results rather than for controller design.

Remark 4.3 *From Assumption 4.4, we know that $Q(k)G_\tau^{-1}(k)$ has p linearly independent eigenvectors, and can be written in the form $Q(k)G_\tau^{-1}(k) = T(k)\Lambda(k)T^{-1}(k)$, where $\Lambda(k)$ is a diagonal matrix with the eigenvalues of $Q(k)G_\tau^{-1}(k)$ as its entries and $T(k)$ is the corresponding invertible matrix consists of the eigenvectors. The technical benefit due to the existence of matrix $Q(k)$, subsequently, the existence of matrix $T(k)$, is also apparent in merging the three items, $\frac{2(1-\sigma\gamma)}{b}Q(k)G_\tau^{-1}(k)$, I and $\gamma\frac{1+\sigma+l}{b}I$ in (4.12), to continue the meaningful stability deduction as shown below.*

Combining with the following facts

$$S^T(\bar{z}(k))\Gamma S(\bar{z}(k)) = \gamma S^T(\bar{z}(k))S(\bar{z}(k)) \quad \text{and} \quad S^T(\bar{z}(k))S(\bar{z}(k)) < l$$

$$2e^T(k+\tau)\beta(k) \leq \gamma e^T(k+\tau)e(k+\tau) + \frac{1}{\gamma}\beta_0^T\beta_0$$

$$2\sigma\gamma e^T(k+\tau)\alpha(k) \le \sigma\gamma e^T(k+\tau)e(k+\tau) + \sigma\gamma\alpha_0^T\alpha_0$$

we further obtain

$$\begin{aligned}\Delta J(k) &\le -be^T(k+\tau)\{\frac{2(1-\sigma\gamma)}{b}Q(k)G_\tau^{-1}(k) - I - \gamma\frac{1+\sigma+l}{b}I\}e(k+\tau) \quad (4.12)\\ &\quad -be^T(k)e(k) - \sigma\|\tilde{W}\|_F^2 - \sigma(1-\sigma\gamma)\|\hat{W}\|_F^2 + C_0\\ &\le -be^T(k+\tau)T(k)\{\frac{2(1-\sigma\gamma)}{b}\Lambda(k) - I - \gamma\frac{1+\sigma+l}{b}I\}T^{-1}(k)e(k+\tau)\\ &\quad -be^T(k)e(k) + C_0\end{aligned}$$

with $C_0 = \sigma\|W^*\|_F^2 + \sigma\gamma\alpha_0^T\alpha_0 + \frac{1}{\gamma}\beta_0^T\beta_0$ being a positive constant. From Assumption 4, we know that

$$\frac{1-\sigma\gamma}{b}\Lambda(k) > I \quad \text{and} \quad 0 < \sigma\gamma < 1 \quad (\sigma > 0 \quad \text{and} \quad \gamma > 0)$$

we have

$$\begin{aligned}\Delta J(k) &\le -be^T(k+\tau)T(k)\{I - \gamma\frac{1+\sigma+l}{b}I\}T^{-1}(k)e(k+\tau) - be^T(k)e(k) + C_0\\ &\le -\{b - \gamma(1+\sigma+l)\}e^T(k+\tau)e(k+\tau) - be^T(k)e(k) + C_0\end{aligned}$$

If we choose the design parameters as follows

$$\gamma < \frac{b}{1+\sigma+l} \tag{4.13}$$

then $\Delta J(k) \le 0$ once any of the tracking errors $|e_i(k)|, i = 1, \ldots, p$ is larger than $\sqrt{\frac{C_0}{b}}$. Furthermore, the tracking error $e(k)$ will converge to the compact set denoted by

$$\Omega_{e_0} \triangleq \{e(k) \Big| \; |e_i(k)| \le \sqrt{\frac{C_0}{b}}, i = 1, 2, \ldots, p\} \tag{4.14}$$

Due to negativeness of $\Delta J(k)$, we can conclude that $e(k+\tau)$ must converges to the compact set Ω_{e_0} if $e(k)$ outside of Ω_{e_0} and all other conditions hold. Thus $y(k+\tau) \in \Omega_y$ will still hold if $\Omega_{e_0} \subset \Omega_e$.

By subtracting W^* to both sides of weights updating equation (4.8), it can be rewritten as

$$\tilde{W}(k+1) = (1-\sigma\gamma)\tilde{W}(k-\tau+1) - \sigma\gamma W^* - \Gamma S(\bar{z}(k-\tau+1))e^T(k+1)Q(k-\tau+1)$$

Since $e(k+1)$ converges to the small compact set Ω_{e_0} and all the elements of $S(\bar{z}(k))$ are less than 1, $Q(k-\tau+1)$ and W^* are also bounded, thus, noting Lemma A.1 and $0 < 1-\sigma\gamma < 1$, $\tilde{W}(k)$ will be bounded in a compact set denoted by Ω_{we} by recursive computation if its initial value $\tilde{W}(k_0)$ is bounded. It is obvious that we can initialize $\tilde{W}(k_0)$ to be in the compact set $\Omega_{w_0} \subset \Omega_w$. Hence, according to $\hat{W}(k) = \tilde{W}(k) + W^*$, we conclude $\hat{W}(k)$ is bounded without the need of PE condition. Thus $u(k) \in \Omega_u$ will still hold if $\Omega_{we} \subset \Omega_w$.

Finally, if we initialize system at time instant k_0 as follows

$$\begin{aligned} y(k_0 - j) &\in \Omega_{y_0}, j = 0, \ldots, \max\{n_1, \ldots, n_p\} - 1, \\ u(k_0 - j) &\in \Omega_u, j = 1, \ldots, \max\{m_1, \ldots, m_p\} + \tau, \\ \tilde{W}(k_0 - j) &\in \Omega_{w_0}, j = 0, \ldots, \tau - 1, \end{aligned}$$

and we choose suitable parameters γ, l and σ according to (4.13), there exists a constant $k^* > k_0 + \tau$ such that tracking error converge to Ω_{e_0}, and NN weight error converges to Ω_{we} for all $k > k^*$. This implies the closed-loop system is SGUUB. Then $y(k) \in \Omega_y$, and $u(k) \in \Omega_u$ will hold for all $k > k_0$.

Therefore, for any a priori given (arbitrarily large) bounded set Ω and any a priori given (arbitrarily small) set Ω_0, which contains $(0, 0)$ as an interior point, there exist a control u, such that every trajectory of the closed-loop system starting from Ω enters the set Ω_0 in a finite time and remains in it thereafter. That is to say, the whole closed-loop system is SGUUB. ■

Remark 4.4 *It should be noted that the size of Ω_{e_0} indicates the possible maximum bound that the tracking error can reach. Considering Ω_{e_0} defined in (4.14), we can see that the size of Ω_{e_0} cannot be made arbitrarily small and it cannot be known a priori also. Noting that $C_0 = \sigma\|W^*\|_F^2 + \sigma\gamma\alpha_0^T\alpha_0 + \frac{1}{\gamma}\beta_0^T\beta_0$, by choosing sufficient small σ, we can see that C_0 (Ω_{e_0}) is approximately proportional to $\frac{1}{\gamma}$ provided that b is fixed. Furthermore, noting (4.13), we know γ is of order $1/l$. Therefore, the larger the approximator size, the larger the error peak maybe be expected, as C_0 grows in proportion to l.*

Remark 4.5 *Note that the size of Ω_w is not predetermined, and it is introduced for analytical purpose because neural network approximation is only valid on a compact set. In fact, Ω_w can be made arbitrary large to guarantee $\hat{W}(k) \in \Omega_w$, even in the transient period, as we have proved that $\hat{W}(k)$ is bounded. In practical implementation, we can initialize $\hat{W}(0) = 0$ (thus $u(0) = 0$, which must be within Ω_u), and the corresponding parameter estimation error, $\tilde{W}(0) = \hat{W}(0) - W^* = -W^*$, is obviously bounded, and within the compact set Ω_w as it can be made arbitrarily large. As the control system needs to be initialized for the first τ steps, they could be simply set to be 0. For better performance, especially the transient performance, off-line training could be used to initialize the controller [20].*

4.1.4 Simulation

Consider the following discrete-time MIMO system

$$\begin{aligned}
x_1(k+1) &= x_2(k) \\
x_2(k+1) &= \frac{x_2(k)x_4(k)}{1+x_1^2(k)+x_3^2(k)} + \frac{u_1(k)}{1+x_1^2(k)+x_3^2(k)} - \frac{0.5u_2(k)}{1+x_1^2(k)+x_3^2(k)} \\
x_3(k+1) &= x_4(k) \\
x_4(k+1) &= \frac{x_4(k)}{1+x_1^2(k)+x_3^2(k)} + \frac{0.5u_1(k)}{1+x_1^2(k)+x_3^2(k)} - \frac{u_2(k)}{1+x_1^2(k)+x_3^2(k)} + d(k) \\
y_1(k) &= x_2(k) \\
y_2(k) &= x_1^2(k) + x_4(k)
\end{aligned}$$

The control objective is to control the system outputs $y_1(k)$ and $y_2(k)$ tracking the reference trajectories $y_{d_1}(k) = 0.25\sin(\frac{k\pi}{200}) + 0.25\sin(\frac{k\pi}{100})$, $y_{d_2}(k) = 1 - 0.25\sin(\frac{k\pi}{300})$ and disturbance $d(k) = 0.05\cos(0.05k)$ respectively.

Simulation parameters are chosen as follows: neural number $l_1 = l_2 = 142$, orthogonal matrix $Q(k) = [1, 0; 0, -1]$, system initial states and neural network weights are initialized to zero, $\sigma = 0.01$ and adaptation gain matrix $\Gamma = 0.025I_{142\times142}$,

Simulation results are shown in Figure 4.1-Figure 4.5. Figure 4.1 and Figure 4.2 show the tracking performances of the first sub-system and the second sub-system respectively. The control input trajectories $u_1(k)$ and $u_2(k)$ are shown in Figure 4.3.

The weight matrix norm $\|\hat{W}(k)\|_F$ is shown in Figure 4.4. Tracking errors are shown in Figure 4.5.

4.2 Non-affine MIMO NARMAX Systems

4.2.1 Introduction

In Section 4.1, neural network control scheme was investigated for a class of MIMO NARMAX discrete-time systems. The τ-step weight update laws was proved to be effective in handling the τ-step predictor model in the presence of unknown bounded disturbances. However, the system studied is in affine form and an orthogonal matrix should be found in order to update the NN weights. In this section, the system studied is in non-affine MIMO NARMAX form. For the $n \times n$ MIMO systems, the inputs of the system are in triangular form. Due to this property and by implicit function theorem [50], we can firstly define the IDFC control in a nested manner, then using neural networks to emulate those IDFC.

This section is organized as follows. System dynamics as well as some stability notions are proposed in Section 4.2.2. In Section 4.2.4, a simulation example is used to illustrate the effectiveness of the proposed scheme.

4.2.2 MIMO System Dynamics

Considering the following n inputs n outputs non-affine nonlinear NARMAX MIMO systems with triangular form inputs

$$\left\{\begin{array}{lcl} y_1(k+\tau) & = & f_1(Y(k), U_{k-1}(k), u_1(k)) \\ & \vdots & \\ y_j(k+\tau) & = & f_j(Y(k), U_{k-1}(k), u_1(k), \ldots, u_j(k)) \\ & \vdots & \\ y_n(k+\tau) & = & f_n(Y(k), U_{k-1}(k), u_1(k), \ldots, u_j(k), \ldots, u_n(k)) \end{array}\right. \tag{4.15}$$

where τ is the system delay; $y(k) = [y_1(k), \ldots, y_n(k)]^T$ and $u(k) = [u_1(k), \ldots, u_n(k)]^T$ are system outputs and inputs, respectively; $Y(k)$ is a vector containing current and past outputs, $U_{k-1}(k)$ is a vector containing only past inputs. In particular, they are defined as

$$\begin{aligned} U_{k-1}(k) &= [u_1(k-1),\dots,u_1(k-m_1),u_2(k-1),\dots,u_2(k-m_2),\dots,\\ &\quad u_n(k-1),\dots,u_n(k-m_n)]^T \\ Y(k) &= [y_1(k),\dots,y_1(k-n_1+1),y_2(k),\dots,y_2(k-n_2+1),\dots,\\ &\quad y_n(k),\dots,y_n(k-n_n+1)]^T \end{aligned}$$

with n_i denotes the length of the i-th subsystem's outputs, and m_i is the length of the i-th subsystem's inputs, which satisfies $m_i < n_i$, $i = 1,\dots,n$. $f_j(\cdot)$ are nonlinear functions; $\bar{u}_{j-1}(k) = [u_1(k),\dots,u_{j-1}(k))]^T$.

The control objective is to design control input $u(k)$ for system (4.15) to drive the system output $y(k)$ follow a known and bounded trajectory

$$y_d(k) = [y_{d_1}(k), y_{d_2}(k),\dots,y_{d_n}(k)]^T \in R^n$$

Assumption 4.5 *The desired trajectory $y_d(k) \in \Omega_{yd} \subset R^p$, $\forall k > 0$ is smooth and known, where Ω_{yd} is a small subset of Ω_y and $\Omega_y \triangleq \{\chi(k)|\chi(k) = y(k)\} \subset R^p$.*

Assumption 4.6 *There are positive constants $\underline{d}_i$ and $\bar{d}_i$ $(i = 1,\dots,n)$, such that $0 < \underline{d}_i \le |\frac{\partial f_i(Y(k),U_{k-1}(k),u_1(k),\dots,u_i(k))}{\partial u_i}| \le \bar{d}_i$.*

Remark 4.6 *The partial derivative $\frac{\partial f_i(\cdot)}{\partial u_i(k)}$ can be considered as the controller gain of the i-th input for the i-th subsystem. Assumption 4.6 indicates that this control gain is either positive or negative, and is also upper and lower bounded. The sign does not need to be known a priori.*

Assumption 4.7 *The nonlinear functions $f_i(\cdot)$ $(i = 1,\dots,n)$ are differentiable.*

In the following, Lemma 4.1 (Mean Value Theorem for multi variables), Lemma 4.2 (Implicit Function Theorem) and Lemma A.1 (Bounded Input Bounded Output) are given, which will be used later.

Lemma 4.1 *Let $f : R^n \rightarrow R$ be differentiable at every point in an open set containing the line segment L joining two vectors $\bar{a}$ and $\bar{b}$ in R^n, then there is a vector $\bar{\xi}$ on L such that*

$$f(\bar{b}) - f(\bar{a}) = \nabla f(\bar{\xi}) \cdot (\bar{b} - \bar{a})$$

with $\nabla f(\cdot)$ denotes the gradient of $f(\cdot)$ [144].

Lemma 4.2 *Assume that $f(x, y) : R^n \times R \rightarrow R$ is continuously differentiable $\forall (x, y) \in R^n \times R$, and there is a positive constant d such that $\partial f(x, y)/\partial y(x, y) > d > 0$, $\forall (x, y) \in R^n \times R$. Then there exists a continuous (smooth) function $y^* = g(x)$ such that $f(x, y^*) = 0$. For the case $\partial f(x, y)/\partial y(x, y) < -d < 0$, $\forall (x, y) \in R^n \times R$. The result still holds [50].*

Define the tracking error $e(k) = y(k) - y_d(k)$ as

$$\begin{aligned} e(k) &= [e_1(k), \ldots, e_n(k)]^T \\ &= [y_1(k) - y_{d_1}(k), \ldots, y_n(k) - y_{d_n}(k)]^T \end{aligned} \tag{4.16}$$

Considering the first equation in (4.15), subtracting $y_{d_1}(k+\tau)$ on both sides, we have

$$e_1(k + \tau) = f_1(Y(k), U_{k-1}(k), u_1(k)) - y_{d_1}(k + \tau)$$

Noting Assumption 4.6, we can obtain

$$\left|\frac{\partial\left[f_1(Y(k), U_{k-1}(k), u_1(k)) - y_{d_1}(k + \tau)\right]}{\partial u_1}\right| > \underline{d}_1 > 0$$

by Lemma 4.2, we know that there is

$$\begin{aligned} u_1^*(k) &\triangleq \alpha_1'(Y(k), U_{k-1}(k), y_{d_1}(k + \tau)) \\ &\triangleq \alpha_1(Y(k), U_{k-1}(k), y_d(k + \tau)) \end{aligned} \tag{4.17}$$

such that

$$e_1(k + \tau) = f_1(Y(k), U_{k-1}(k), u_1^*(k)) - y_{d_1}(k + \tau) = 0$$

Remark 4.7 *It should be noted that though $u_1^*(k)$ only depends on $Y(k)$, $U_{k-1}(k)$ and $y_{d_1}(k+\tau)$, for the ease of analysis, we regard it as a function of $Y(k)$, $U_{k-1}(k)$ and $y_d(k+\tau)$.*

Considering the second equation in (4.15), we have

$$e_2(k+\tau) = f_2(Y(k), U_{k-1}(k), u_1(k), u_2(k)) - y_{d_2}(k+\tau)$$

Let $u_1(k) = u_1^*(k)$, noting Assumption 4.7 and by Lemma 4.2, we know that there is a ideal control

$$\begin{aligned} u_2^*(k) &\triangleq \alpha_2'(Y(k), U_{k-1}(k), y_{d_1}(k+\tau), y_{d_2}(k+\tau)) \\ &\triangleq \alpha_2(Y(k), U_{k-1}(k), y_{d_1}(k+\tau), y_d(k+\tau)) \end{aligned} \tag{4.18}$$

such that

$$e_2(k+\tau) = f_2(Y(k), U_{k-1}(k), u_1^*(k), u_2^*(k)) - y_{d_2}(k+\tau) = 0$$

Similarly, we know that there are ideal controls

$$\begin{cases} u_3^*(k) & \triangleq & \alpha_3(Y(k), U_{k-1}(k), y_d(k+\tau)) \\ & \cdots & \\ u_j^*(k) & \triangleq & \alpha_j(Y(k), U_{k-1}(k), y_d(k+\tau)) \\ & \cdots & \\ u_n^*(k) & \triangleq & \alpha_n(Y(k), U_{k-1}(k), y_d(k+\tau)) \end{cases}$$

such that

$$\begin{cases} e_3(k+\tau) & = & f_3(Y(k), U_{k-1}(k), u_1^*(k), \ldots, u_3^*(k)) - y_{d_3}(k+\tau) = 0 \\ & \vdots & \\ e_j(k+\tau) & = & f_j(Y(k), U_{k-1}(k), u_1^*(k), \ldots, u_j^*(k)) - y_{d_j}(k+\tau) = 0 \\ & \vdots & \\ e_n(k+\tau) & = & f_n(Y(k), U_{k-1}(k), u_1^*(k), \ldots, u_n^*(k)) - y_{d_n}(k+\tau) = 0 \end{cases}$$

Definition 4.3 *The ideal controls $u_1^*(k)$, $u_2^*(k)$, ..., $u_n^*(k)$, which can realize exact tracking in τ steps and cannot be explicitly spelt out, are called implicit desired feedback control (IDFC).*

Summarizing equations (4.17), (4.18) and (4.19), we can see that the i-th IDFC, $u_i^*(k)$, can be expressed as follows

$$\begin{aligned} u_i^*(k) &= \alpha_i(z(k)), \qquad i = 1, 2, \ldots, n \\ z(k) &\triangleq [Y^T(k), U_{k-1}^T(k), y_d^T(k+\tau)]^T \in R^{\sum_{j=1}^{n}(m_j+n_j)+n} \end{aligned} \tag{4.19}$$

Its vector form is as follows

$$u^*(k) = \begin{bmatrix} \alpha_1(z(k)) \\ \alpha_2(z(k)) \\ \vdots \\ \alpha_n(z(k)) \end{bmatrix} \in R^{n\times 1} \tag{4.20}$$

It can be seen that system (4.15) is in non-affine form. For the convenience of analysis, denote system (4.15) in the following vector form

$$y(k+\tau) = F(Y(k), U_{k-1}(k), u(k)) \tag{4.21}$$

with nonlinear vector function $F(\cdot) \in R^{n\times 1}$ is defined as

$$F(Y(k), U_{k-1}(k), u(k)) = \begin{bmatrix} f_1(Y(k), U_{k-1}(k), u_1(k)) \\ \vdots \\ f_j(Y(k), U_{k-1}(k), u_1(k), \ldots, u_j(k)) \\ \vdots \\ f_n(Y(k), U_{k-1}(k), u_1(k), \ldots, u_j(k), \ldots, u_n(k)) \end{bmatrix}$$

Therefore, we have

$$e(k+\tau) = F(Y(k), U_{k-1}(k), u(k)) - y_d(k+\tau) \tag{4.22}$$

Adding and subtracting $F(Y(k), U_{k-1}(k), u^*(k))$ to the right side of equation (4.22), we have

$$\begin{aligned} e(k+\tau) &= F(Y(k), U_{k-1}(k), u(k)) - y_d(k+\tau) + F(Y(k), U_{k-1}(k), u^*(k)) \\ &\quad -F(Y(k), U_{k-1}(k), u^*(k)) \\ &= F(Y(k), U_{k-1}(k), u(k)) - F(Y(k), U_{k-1}(k), u^*(k)) \end{aligned} \tag{4.23}$$

Considering the i-th $1 \leq i \leq n$ equation in the error dynamics (4.23), we have

$$\begin{aligned} e_i(k+\tau) &= f_i(Y(k), U_{k-1}(k), u_1(k), \ldots, u_i(k)) \\ &\quad -f_i(Y(k), U_{k-1}(k), u_1^*(k), \ldots, u_i^*(k)) \end{aligned} \tag{4.24}$$

By noting Lemma 4.1, the Mean Value Theorem for multi variables, equation (4.24) can be written as

$$e_i(k+\tau) = \nabla f_i(Y(k), U_{k-1}(k), \bar{u}_{\xi_i})\,[\bar{u}_i(k) - \bar{u}_i^*(k)] \tag{4.25}$$

with

$$\begin{aligned} \bar{u}_{\xi_i} &= [u_{\xi_{i1}}, u_{\xi_{i2}}, \ldots, u_{\xi_{ii}}]^T \\ \nabla f_i(Y(k), U_{k-1}(k), u_{\xi_i}) &= [\frac{\partial f_i}{\partial u_1}, \frac{\partial f_i}{\partial u_2}, \ldots, \frac{\partial f_i}{\partial u_i}]^T \in R^{i\times 1} \end{aligned}$$

with $\bar{u}_{\xi_i} \in [\bar{u}_i^*(k), \bar{u}_i(k)]$.

Then equation (4.23) can be written as

$$e(k+\tau) = \nabla F(k) \cdot [u(k) - u^*(k)] \tag{4.26}$$

with

$$\nabla F(k) \triangleq \begin{bmatrix} \frac{\partial f_1}{\partial u_1}|_{u_1(k)=u_{\xi_{11}}} & 0 & 0 & \cdots & 0 \\ \frac{\partial f_2}{\partial u_1}|_{u_1(k)=u_{\xi_{21}}} & \frac{\partial f_2}{\partial u_2}|_{u_2(k)=u_{\xi_{22}}} & 0 & \cdots & 0 \\ \frac{\partial f_3}{\partial u_1}|_{u_1(k)=u_{\xi_{31}}} & \frac{\partial f_3}{\partial u_2}|_{u_2(k)=u_{\xi_{32}}} & \frac{\partial f_3}{\partial u_3}|_{u_3(k)=u_{\xi_{33}}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ \frac{\partial f_n}{\partial u_1}|_{u_1(k)=u_{\xi_{n1}}} & \frac{\partial f_n}{\partial u_2}|_{u_2(k)=u_{\xi_{n2}}} & \frac{\partial f_n}{\partial u_3}|_{u_3(k)=u_{\xi_{n3}}} & \cdots & \frac{\partial f_n}{\partial u_n}|_{u_n(k)=u_{\xi_{nn}}} \end{bmatrix} \tag{4.27}$$

and $\nabla F(k) \in R^{n\times n}$. For the ease of analysis, define

$$G(k) \triangleq \nabla F(k) \tag{4.28}$$

Therefore, we have

$$e(k+\tau) = G(k)\,[u(k) - u^*(k)] \tag{4.29}$$

It can be easily obtained that the matrix $G(k)$ possess the following properties:

1. $G(k)$ is full rank and $|G(k)| = \Pi_{i=1}^{n}\left(\frac{\partial f_i}{\partial u_i}|_{u_i(k)=u_{\xi_{ii}}}\right)$;

2. $G(k)$ is upper and lower bounded, i.e, there are two constants $a = \Pi_{i=1}^{n}\underline{d}_i$ and $b = \Pi_{i=1}^{n}\bar{d}_i$, such that $aI \leq G(k) \leq bI$ $(a, b > 0)$ or $bI \leq G(k) \leq aI$ $(a, b < 0)$.

It can be seen that the matrix $G(k)$ is either positive or negative, which depends on the signs of its diagonal elements. In the following, without losing of generality, we assume that $G(k)$ is positive, i.e, $aI \leq G(k) \leq bI$ $(a, b > 0)$. Therefore, we can obtain

$$\frac{1}{b}I \leq G^{-1}(k) \leq \frac{1}{a}I, \qquad a, b > 0 \tag{4.30}$$

4.2.3 Stability Analysis

Considering the implicit desired feedback controls (IDFCs) defined in equation (4.20), they are continuous nonlinear functions. Therefore, there are ideal weights W^* such that the smooth function vector $u^*(k)$ can be approximated by an ideal NN on a compact set $\Omega_z \subset R^q$

$$u^*(k) = W^{*T}S(z(k)) + \epsilon_z \tag{4.31}$$

where $z(k)$ has been defined in equation (4.19) as follows

$$\begin{aligned} z(k) &= \begin{bmatrix} Y(k) \\ U_{k-1}(k) \\ y_d(k+\tau) \end{bmatrix} \in \Omega_z \subset R^q, \qquad q = \textstyle\sum_{i=1}^{n}(n_i + m_i) + n \\ \epsilon_z &= [\epsilon_{z_1}, \ldots, \epsilon_{z_n}]^T \end{aligned}$$

and ϵ_z is the bounded NN approximation error vector satisfying $\|\epsilon_z\| \leq \epsilon_0$ (ϵ_0 is a constant vector) on the compact set, which can be reduced by increasing the number of the adjustable weights. The ideal weight matrix W^* is required for analytical purpose only, and is defined as that minimizes $\|\epsilon_z\|$ for all $z(k) \in \Omega_z \subset R^q$ in a compact region, i.e.,

$$W^* \triangleq arg \min_{W \in \Omega_w} \left\{ \sup_{z \in \Omega_z} |u^*(k) - W^T S(\bar{z}(k))| \right\} \tag{4.32}$$
$$\Omega_z \subset R^q \;\; \text{and compact set} \;\; \Omega_w \subset R^{l \times p}$$

In general, the ideal NN weight matrix, W^*, is unknown though constant, its estimate, $\hat{W}$, should be used for controller design which will be discussed in the following.

Choosing the practical neural network controls and corresponding weight update laws as follows

$$\begin{aligned} u(k) &= \hat{W}^T(k)S(z(k)) && (4.33)\\ \hat{W}(k+1) &= \hat{W}(k-\tau+1) \\ &\quad -\Gamma[S(z(k-\tau+1))e^T(k+1)+\sigma\hat{W}(k-\tau+1)] && (4.34)\end{aligned}$$

where $\Gamma = \gamma I$ is a diagonal matrix with $\gamma > 0$, σ is a positive constant number, $\hat{W}(k) \in R^{p\times l}$ and $S(z(k)) \in R^l$. For the ease of analysis, we rewrite equation (4.34) as follows

$$\hat{W}(k+\tau) = \hat{W}(k) - \Gamma\left[S(z(k))e^T(k+\tau)+\sigma\hat{W}(k)\right] \qquad (4.35)$$

Noting equation (4.29), we can obtain that

$$\begin{aligned} e(k+\tau) &= G(k)[\hat{W}^T(k)S(z(k)) - {W^*}^T(k)S(z(k)) - \epsilon_z] \\ &= G(k)\tilde{W}^T(k)S(z(k)) - G(k)\epsilon_z && (4.36)\end{aligned}$$

Thus, we can obtain

$$\tilde{W}^T(k)S(z(k)) = G^{-1}(k)e(k+\tau)+\epsilon_z \qquad (4.37)$$

Theorem 4.2 *Consider the closed-loop system consisting of system (4.15), controller (4.33) and adaptation law (4.34). There exist compact sets $\Omega_{y0} \subset \Omega_y$, $\Omega_{w_0} \subset \Omega_w$ and positive constants l^*, γ^* and σ^* such that if*

1. *Assumptions 4.6, 4.7 and 4.5 being satisfied, the condition at time instant k_0 is initialized as*

$$\begin{aligned} y(k_0-j) &\in \Omega_{y_0}, \quad j=0,\ldots,\max\{n_1,\ldots,n_n\}-1 \\ u(k_0-j) &\in \Omega_u, \quad j=1,\ldots,\tau+\max\{m_1,\ldots,m_n\} \\ \tilde{W}(k_0-j) &\in \Omega_{w_0}, \quad j=0,\ldots,\tau-1\end{aligned}$$

2. *the semi determined future outputs at time instant k_0, $y(k_0+1), \ldots, y(k_0+\tau-1)$ are all in compact set Ω_y, and*

3. *the design parameters are suitably chosen such that $l > l^*$, $\sigma < \sigma^*$ and $\gamma < \gamma^*$ with γ being the eigenvalue of Γ,*

then, the closed-loop system is SGUUB.

Proof: Choose the Lyapunov function candidate as

$$J(k) = \frac{1}{b}\sum_{j=0}^{\tau-1} tr\{e(k+j)e^T(k+j)\} + \sum_{j=0}^{\tau-1} tr\{\tilde{W}^T(k+j)\Gamma^{-1}\tilde{W}(k+j)\} \tag{4.38}$$

where b is the positive constant, which denotes the upper bound of the matrix $G(k)$. Apparently, the Lyapunov function candidate $J(k)$ contains the states of the error dynamics of the systems, and the parameter adaptation. Note that the future variables, $e(k+1), \ldots, e(k+\tau-1)$ and $\tilde{W}(k+1), \ldots \tilde{W}(k+\tau-1)$, are all semi-determined at time instant k as they are independent of current control $u(k)$. We have shown that $y(k+\tau-1), \ldots, y(k+1)$ are all independent of $u(k)$, so are $e(k+\tau-1), \ldots, e(k+1)$. For the same reason, it can be shown that $\tilde{W}(k+\tau-1), \ldots, \tilde{W}(k+1)$ are all determined at time instant k. For example,

$$\tilde{W}(k+\tau-1) = \tilde{W}(k-1) - \Gamma\left[S(z(k-1))e^T(k+\tau-1) + \sigma\hat{W}(k-1)\right]$$

is uniquely determined since (i) $e^T(k+\tau-1)$ is semi-determined, and (ii) all other signals are well defined at time instant k.

The first difference of (4.38) along (4.35) is given by

$$\begin{aligned}
\Delta J(k) \;=\;& \frac{1}{b}e^T(k+\tau)e(k+\tau) - \frac{1}{b}e^T(k)e(k) \\
& + tr\{\tilde{W}^T(k+\tau)\Gamma^{-1}\tilde{W}(k+\tau)\} - tr\{\tilde{W}^T(k)\Gamma^{-1}\tilde{W}(k)\} \\
\;=\;& \frac{1}{b}e^T(k+\tau)e(k+\tau) - \frac{1}{b}e^T(k)e(k) - 2\sigma tr\{\tilde{W}^T(k)\hat{W}(k)\} \\
& +\sigma^2 tr\{\hat{W}^T(k)\Gamma\hat{W}(k)\} - 2tr\{\tilde{W}^T(k)S(z(k))e^T(k+\tau)\} \\
& +2\sigma tr\{\hat{W}^T(k)\Gamma S(z(k))e^T(k+\tau)\} \\
& +tr\{e(k+\tau)S^T(z(k))\Gamma S(z(k))e^T(k+\tau)\}
\end{aligned}$$

Noting that

$$-2\sigma tr\{\tilde{W}^T(k)\hat{W}(k)\} = -\sigma\|\tilde{W}\|_F^2 - \sigma\|\hat{W}\|_F^2 + \sigma\|W^*\|_F^2$$

$$\sigma^2 tr\{\hat{W}^T(k)\Gamma\hat{W}(k)\} = \sigma^2\gamma\|\hat{W}\|_F^2$$

$$-2tr\{\tilde{W}^T(k)S(z(k))e^T(k+\tau)\} = -2e^T(k+\tau)\tilde{W}^T(k)S(z(k))$$

$$2\sigma tr\{\hat{W}^T(k)\Gamma S(z(k))e^T(k+\tau)\} = 2\sigma\gamma e^T(k+\tau)\hat{W}^T(k)S(z(k))$$

$$tr\{e(k+\tau)S^T(z(k))\Gamma S(z(k))e^T(k+\tau)\} = S^T(z(k))\Gamma S(z(k))e^T(k+\tau)e(k+\tau)$$

We can obtain

$$\begin{aligned}\Delta J(k) \;=\;& \frac{1}{b}e^T(k+\tau)e(k+\tau) - \frac{1}{b}e^T(k)e(k) - \sigma\|\tilde{W}\|_F^2 - \sigma(1-\sigma\gamma)\|\hat{W}\|_F^2 \\ &+\sigma\|W^*\|_F^2 - 2e^T(k+\tau)\tilde{W}^T(k)S(z(k)) + 2\sigma\gamma e^T(k+\tau)\hat{W}^T(k)S(z(k)) \\ &+S^T(z(k))\Gamma S(z(k))e^T(k+\tau)e(k+\tau)\end{aligned}$$

Noting equation (4.37) and $\hat{W}(k) = \tilde{W}(k) + W^*$, we can obtain

$$\begin{aligned}\Delta J(k) \;=\;& \frac{1}{b}e^T(k+\tau)e(k+\tau) - \frac{1}{b}e^T(k)e(k) - \sigma\|\tilde{W}\|_F^2 - \sigma(1-\sigma\gamma)\|\hat{W}\|_F^2 \\ &+\sigma\|W^*\|_F^2 - 2e^T(k+\tau)G^{-1}(k)e(k+\tau) - 2e^T(k+\tau)\epsilon_z \\ &+2\sigma\gamma e^T(k+\tau)G^{-1}(k)e(k+\tau) + 2\sigma\gamma e^T(k+\tau)\alpha(k) \\ &+S^T(z(k))\Gamma S(z(k))e^T(k+\tau)e(k+\tau)\end{aligned}$$

where $\alpha(k) = W^{*T}S(z(k)) + \epsilon_z$. Since ϵ_z and $W^{*T}S(z(k))$ are all bounded, it is reasonable to assume that $\alpha(k)$ is bounded. For convenience of analysis, let $\alpha_i(k) \le \alpha_{0i}$, where α_{0i} denotes the i-th element of constant vector α_0, which is only introduced to establish the stability results rather than for controller design.

Combining with the following facts

$$S^T(z(k))\Gamma S(z(k)) = \gamma S^T(z(k))S(z(k)) \quad \text{and} \quad S^T(z(k))S(z(k)) < l$$

$$2e^T(k+\tau)\epsilon_z \le \gamma e^T(k+\tau)e(k+\tau) + \frac{1}{\gamma}\epsilon_0^T\epsilon_0$$

$$2\sigma\gamma e^T(k+\tau)\alpha(k) \le \sigma\gamma e^T(k+\tau)e(k+\tau) + \sigma\gamma\alpha_0^T\alpha_0$$

we further obtain

$$\begin{aligned}\Delta J(k) &\leq -\frac{1}{b}e^T(k+\tau)\{2b(1-\sigma\gamma)G^{-1}(k) - I - \gamma b(1+\sigma+l)I\}e(k+\tau)\\ &\quad -\frac{1}{b}e^T(k)e(k) - \sigma\|\tilde{W}\|_F^2 - \sigma(1-\sigma\gamma)\|\hat{W}\|_F^2 + C_0\\ &\leq -\frac{1}{b}e^T(k+\tau)\{2(1-\sigma\gamma)I - I - \gamma b(1+\sigma+l)I\}e(k+\tau)\\ &\quad -\frac{1}{b}e^T(k)e(k) + C_0\end{aligned}$$

with $C_0 = \sigma\|W^*\|_F^2 + \sigma\gamma\alpha_0^T\alpha_0 + \frac{1}{\gamma}\epsilon_0^T\epsilon_0$ being a positive constant. We have

$$\Delta J(k) \leq -\frac{1}{b}e^T(k+\tau)\{1-2\sigma\gamma-\gamma b(1+\sigma+l)\}e(k+\tau) - be^T(k)e(k) + C_0$$

If we choose the design parameters as follows

$$\frac{1}{\gamma} > 2\sigma + b(1+\sigma+l) \tag{4.39}$$

then we can obtain

$$\Delta J(k) \leq -be^T(k)e(k) + C_0$$

then $\Delta J(k) \leq 0$ once any of the tracking errors $|e_i(k)|, i = 1, \ldots, p$ is larger than $\sqrt{\frac{C_0}{b}}$. Furthermore, the tracking error $e(k)$ will converge to the compact set denoted by

$$\Omega_{e_0} \triangleq \{e(k)\Big|\ |e_i(k)| \leq \sqrt{\frac{C_0}{b}}, i = 1, 2, \ldots, p\} \tag{4.40}$$

Due to negativeness of $\Delta J(k)$, we can conclude that $e(k+\tau)$ must converges to the compact set Ω_{e_0} if $e(k)$ outside of Ω_{e_0} and all other conditions hold. Thus $y(k+\tau) \in \Omega_y$ will still hold if $\Omega_{e_0} \subset \Omega_e$.

By subtracting W^* to both sides of weights updating equation (4.34), it can be rewritten as

$$\tilde{W}(k+1) = (1-\sigma\gamma)\tilde{W}(k-\tau+1) - \sigma\gamma W^* - \Gamma S(z(k-\tau+1))e^T(k+1)Q(k-\tau+1)$$

Since $e(k+1)$ converges to the small compact set Ω_{e_0} and all the elements of $S(z(k))$ are less than 1, $Q(k-\tau+1)$ and W^* are also bounded, thus, noting Lemma A.1 and

$0 < 1 - \sigma\gamma < 1$, $\tilde{W}(k)$ will be bounded in a compact set denoted by Ω_{we} by recursive computation if its initial value $\tilde{W}(k_0)$ is bounded. It is obvious that we can initialize $\tilde{W}(k_0)$ to be in the compact set $\Omega_{w_0} \subset \Omega_w$. Hence, according to $\hat{W}(k) = \tilde{W}(k) + W^*$, we conclude $\hat{W}(k)$ is bounded without the need of PE condition. Thus $u(k) \in \Omega_u$ will still hold if $\Omega_{we} \subset \Omega_w$.

Finally, if we initialize system at time instant k_0 as follows

$$\begin{aligned}
y(k_0 - j) &\in \Omega_{y_0}, j = 0, \ldots, \max\{n_1, \ldots, n_n\} - 1, \\
u(k_0 - j) &\in \Omega_u, j = 1, \ldots, \max\{m_1, \ldots, m_n\} + \tau, \\
\tilde{W}(k_0 - j) &\in \Omega_{w_0}, j = 0, \ldots, \tau - 1,
\end{aligned}$$

and we choose suitable parameters γ, l and σ according to (4.39), there exists a constant $k^* > k_0 + \tau$ such that tracking error converge to Ω_{e_0}, and NN weight error converges to Ω_{we} for all $k > k^*$. This implies the closed-loop system is SGUUB. Then $y(k) \in \Omega_y$, and $u(k) \in \Omega_u$ will hold for all $k > k_0$.

Therefore, for any a priori given (arbitrarily large) bounded set Ω and any a priori given (arbitrarily small) set Ω_0, which contains $(0, 0)$ as an interior point, there exist a control u, such that every trajectory of the closed-loop system starting from Ω enters the set Ω_0 in a finite time and remains in it thereafter. That is to say, the whole closed-loop system is SGUUB. ■

4.2.4 Simulation

Considering the following discrete-time non-affine MIMO system with triangular form inputs

$$\begin{cases}
y_1(k+2) = \frac{y_1(k-1) + y_2(k) u_2(k-1)}{1 + y_2^2(k)} + \sin(u_1(k)) + 2u_1(k) \\
y_2(k+2) = \frac{u_1(k) + y_1(k) u_1(k-1)}{1 + u_1^2(k) + y_2^2(k-1)} + \sin(u_2(k)) + 2u_2(k)
\end{cases}$$

we can see that the system delay $\tau = 2$ and the order of the system is $n = 2$. The control objective is to drive the output $y(k) = [y_1(k), y_2(k)]^T$ of the system to follow desired reference signals

$$y_{d_1}(k) = 0.5 + \frac{1}{4}\cos(\frac{\pi T k}{4}) + \frac{1}{4}\sin(\frac{\pi T k}{2})$$

$$y_{d_2}(k) = 0.5 + \frac{1}{4}\sin(\frac{\pi Tk}{4}) + \frac{1}{4}\sin(\frac{\pi Tk}{2})$$

with $T = 0.01$.

System initial conditions are as follows, $y_1(0) = y1(1) = 0.0$ and $y_2(0) = y_2(1) = 0$. The neurons used are $l = 36$. All the elements of the neural network weights $\hat{W}(0)$ and $\hat{W}(1)$ are initialized to zero, and the active functions $S(z(0))$ and $S(z(1))$ are initialized to be zero. σ modification gain is $\sigma = 0.01$, and adaptive gain matrix is $\Gamma = 0.015I$.

For clarity, the formulas used in the simulation are listed here. The practical controls are as follows:

$$\begin{cases} u(k) &= \hat{W}^T(k)S(z(k)), \qquad W(k) \in R^{l\times n} \text{ and } S(\cdot) \in R^{l\times 1} \\ Y(k) &= [y_1(k), y_1(k-1), y_2(k), y_2(k-1)]^T \in R^4 \\ U_{k-1}(k) &= [u_1(k-1), u_2(k-1)]^T \in R^2 \\ z(k) &= [Y^T(k), U_{k-1}^T(k), y_{d_1}(k+2), y_{d_2}(k+2)]^T \in R^8 \end{cases}$$

The errors' definitions are $(i = 1, 2)$:

$$\Sigma_i : e_i(k) \;=\; y_i(k) - y_{d_i}(k)$$

The weights update law are as follows $(i = 1, 2)$:

$$\hat{W}(k) = \hat{W}(k-2) - \Gamma[S(z(k-2))e(k) + \sigma W(k-2)]$$

Simulation results are shown in Figure 4.6-Figure 4.10. Figure 4.6 and Figure 4.7 show the tracking performances of the first sub-system and the second sub-system respectively. It can be seen that, in the initial period of simulation, the tracking errors are large. Then, as the time increases, the practical outputs converge to the neighborhoods of the desired signals. The control input trajectories $u_1(k)$ and $u_2(k)$ are shown in Figure 4.8. The weight matrix norm $\|\hat{W}(k)\|_F$ is shown in Figure 4.9. Tracking errors are shown in Figure 4.10.

4.3 Conclusion

In this chapter, firstly, for a class of nonlinear discrete-time MIMO systems with unknown interconnections between subsystems, adaptive direct NN control scheme was presented using neural networks. By finding an orthogonal matrix to update the NN weight matrix, it was shown that for appropriately chosen controller parameters, stability of the closed-loop adaptive system can be guaranteed.

Secondly, a simple neural network control scheme was developed for a class of discrete-time nonlinear non-affine MIMO systems. The inputs of the MIMO system are in triangular form. By implicit function theorem, firstly, the existence of the IDFC was shown. Then HONNs were used as the emulators of the IDFCs. Only input and output sequences were used to construct the effective neural network control, which is simple to be implemented in practical applications. Finally, the closed-loop system was proved to be SGUUB based on Lyapunov analysis.

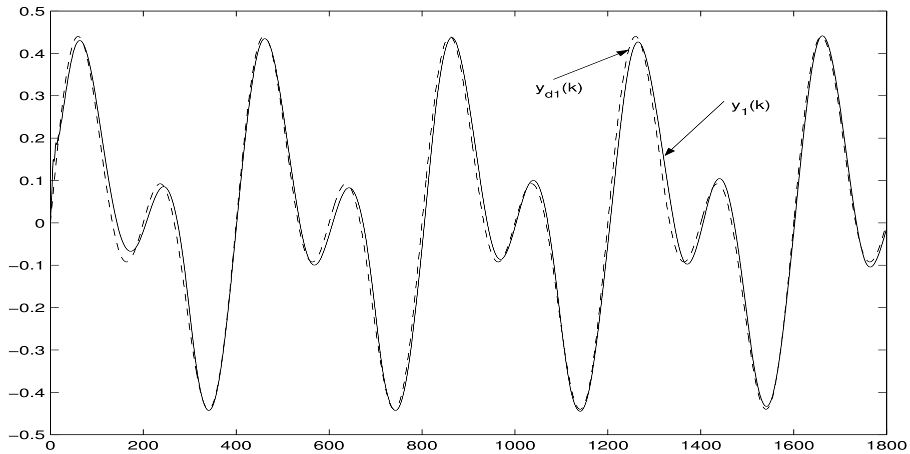

Figure 4.1: Affine NARMAX - Tracking Performance $y_1(k)$ and $y_{d_1}(k)$

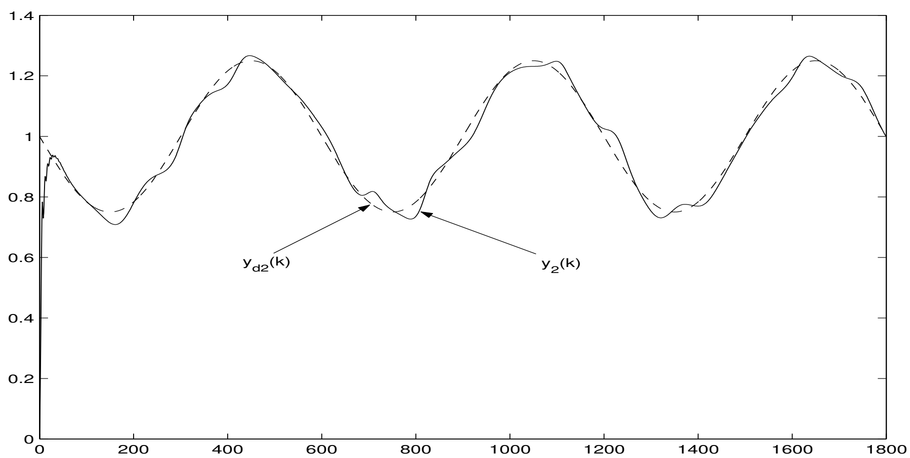

Figure 4.2: Affine NARMAX - Tracking Performance $y_2(k)$ and $y_{d_2}(k)$

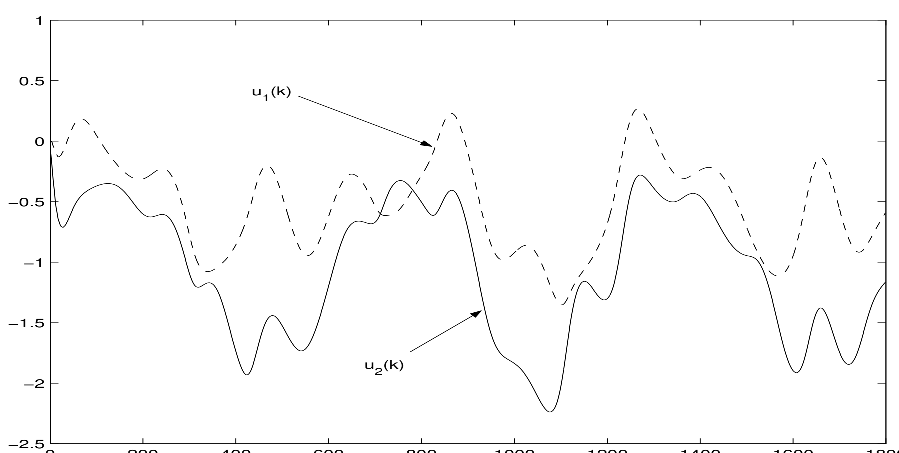

Figure 4.3: Affine NARMAX - Control Inputs $u_1(k)$ and $u_2(k)$

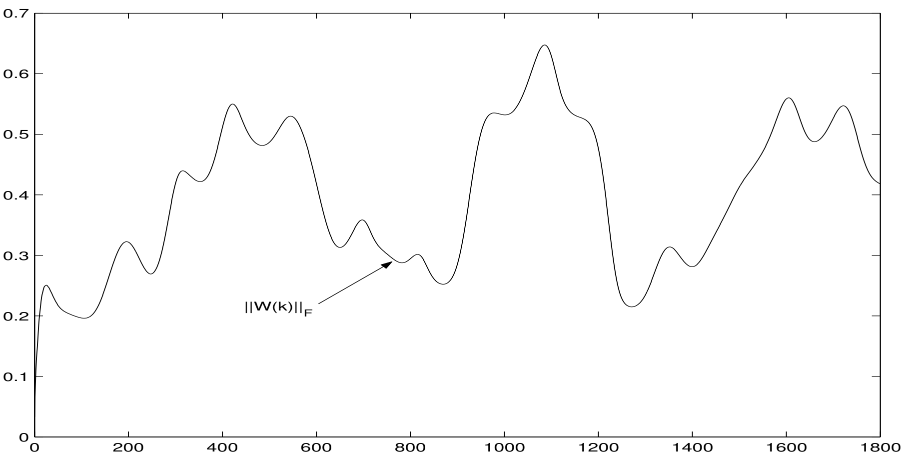

Figure 4.4: Affine NARMAX - Weight Norm $\|\hat{W}(k)\|_F$

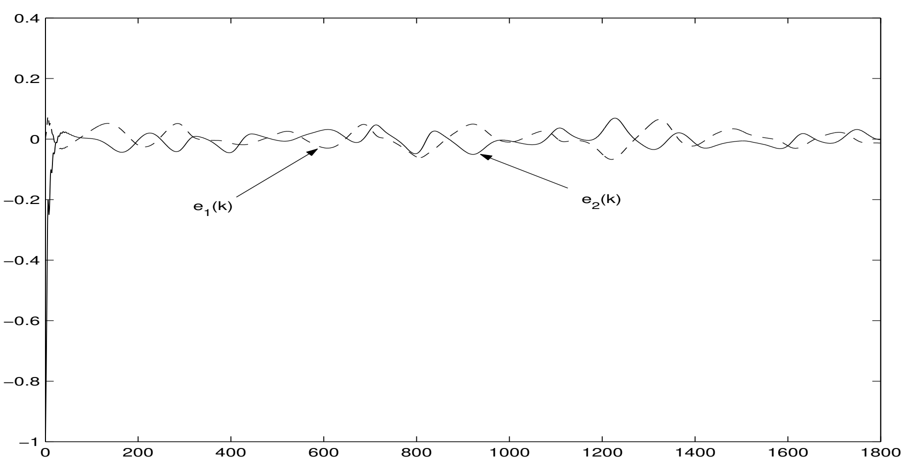

Figure 4.5: Affine NARMAX - Error dynamics

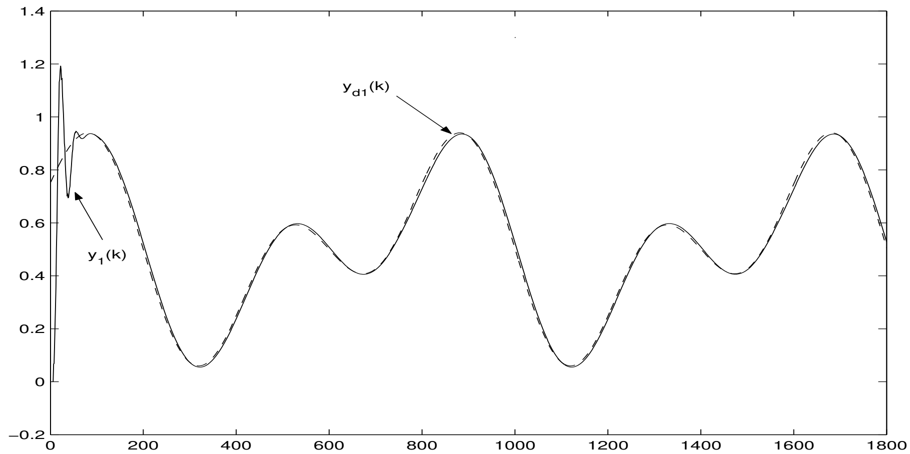

Figure 4.6: Non-affine NARMAX - Tracking Performance $y_1(k)$ and $y_{d_1}(k)$

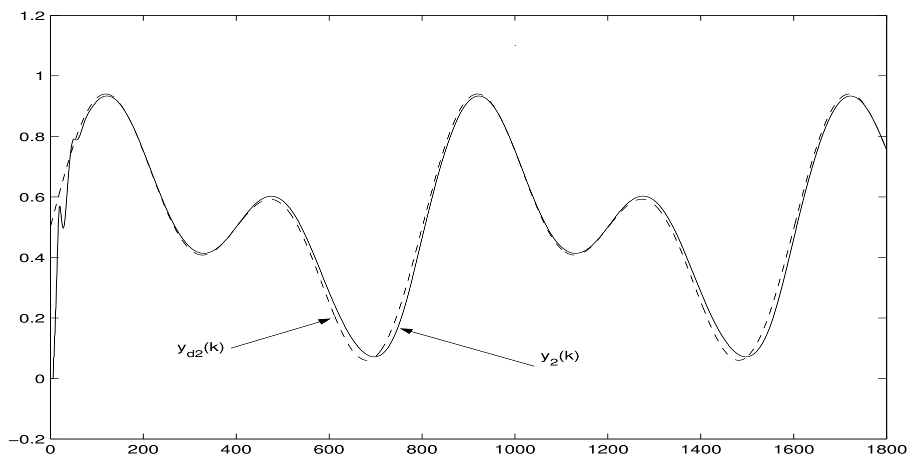

Figure 4.7: Non-affine NARMAX - Tracking Performance $y_2(k)$ and $y_{d_2}(k)$

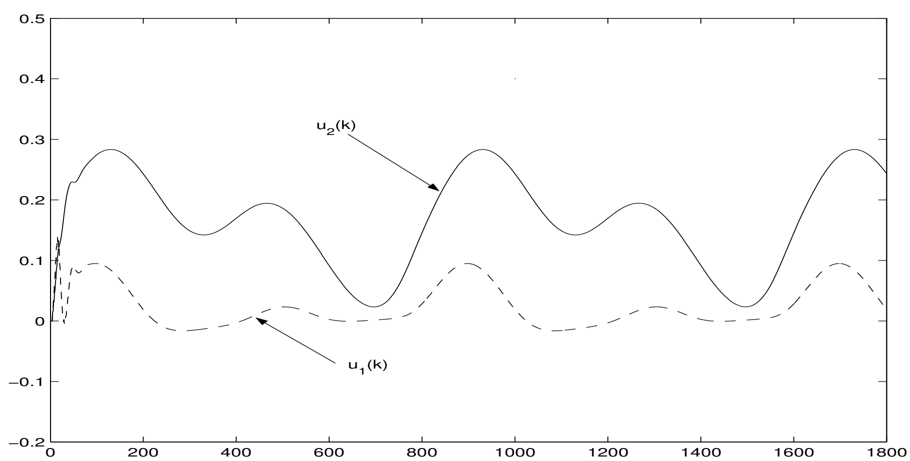

Figure 4.8: Non-affine NARMAX - Control Inputs $u_1(k)$ and $u_2(k)$

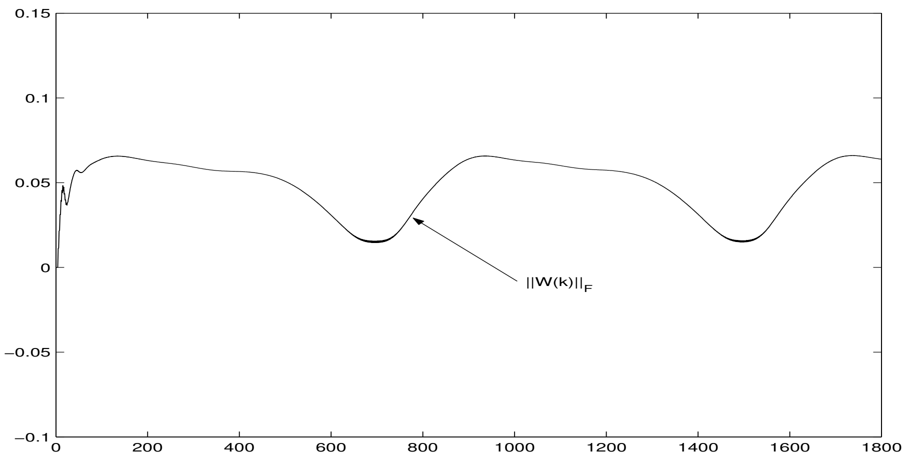

Figure 4.9: Non-affine NARMAX - Weight Norm $\|\hat{W}(k)\|_F$

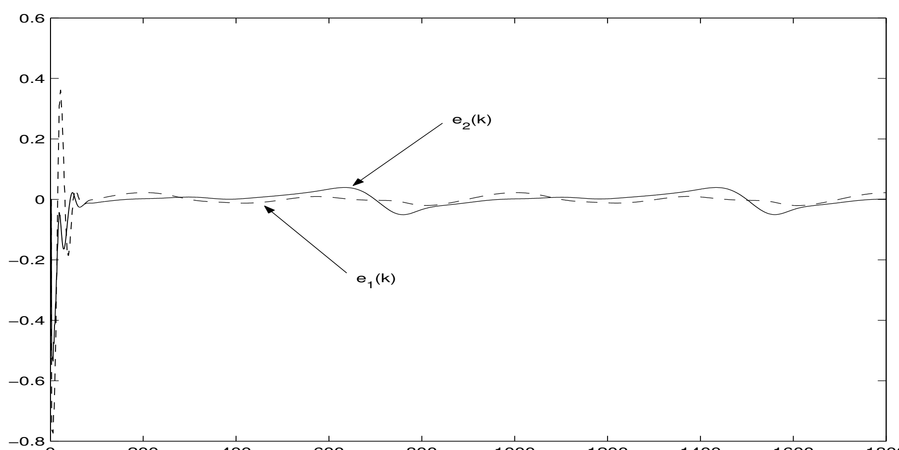

Figure 4.10: Non-affine NARMAX - Error dynamics

Chapter 5

Conclusions and Further Research

5.1 Conclusions

In this thesis, NN control schemes were investigated for five kinds of nonlinear discrete-time systems, HONN, RBF and MNN were used as the function approximators respectively. SGUUB stability was proposed for each kind of system. Specifically, in each chapter, the studied problem is as follows:

In Chapter 2, adaptive NN control scheme for a class of non-affine nonlinear SISO discrete-time systems was investigated. Based on the implicit function theorem, RBF neural networks and MNNs were used respectively as the emulators to approximate the IDFC controller. Projection algorithm was used to guarantee the boundedness of the multi-layer neural network weights. In order to guarantee all multi-layer neural networks tuned within a prescribed range, a newly proposed discontinuous projections with fictitious bounds were used in the MNN weights updating laws. Therefore, a controlled learning may be achieved and the possible destabilizing effect of online tuning of MNN weights can be avoided. The stability of the closed-loop system is proved rigorously by using Lyapunov technique.

Considering the lack of NN control schemes for MIMO nonlinear discrete-time systems, in Chapter 3, state feedback control scheme was investigated for a class of non-affine nonlinear discrete-time MIMO systems with triangular form inputs and bounded disturbances. Because each subsystem of the system studied is in strict feedback form, backstepping design technique was implemented. In order to avoid the non-causal problem in backstepping design, the MIMO system under study was firstly transformed into sequential decrease cascade form, for which, the non-causal problem can be completely removed. Then, HONNs were used to approximate the desired virtual and practical controls. By using backstepping design in a nested manner, the closed-loop system was proved to be SGUUB based on Lyapunov analysis.

Consequently, in the second part of Chapter 3, a simple output feedback control scheme was proposed for a class of MIMO non-affine nonlinear systems with triangular form input, which is similar to the class of systems studied in the first part. However, compared with the first class system studied, two simplifications were introduced due to the need of system coordinate transformation. Firstly, the lengths of different subsystems are required to be equal. Secondly, there are no bounded disturbances's interference. By coordinate transformation, the system was firstly transformed into input output description. Then, the input and output sequences were used to construct the effective neural network control by backstepping technique. HONNs were used to approximate the desired controls. The closed-loop system was proved to be SGUUB based on Lyapunov analysis.

The systems studied in Chapter 3 are all in state space description. However, in the research of discrete-time systems, NARMAX models are also a class of often used discrete-time system representation, for which, only future/current/past input and output sequences appear in the system description. In Chapter 4, two kinds of MIMO NARMAX systems were studied. Firstly, a class of MIMO NARMAX systems in affine form with unknown interconnections between subsystems and bounded disturbances was investigated. By finding an orthogonal matrix to update the NN weight matrix, it was shown that for appropriately chosen controller parameters, SGUUB stability of the closed-loop adaptive system can be guaranteed. Secondly, a simple NN control scheme was developed for a class of non-affine MIMO NARMAX systems. By implicit function theorem, firstly, the existence of the IDFC was shown. Then HONNs were used as the emulators of the IDFCs. Only input and output sequences were used to construct the effective neural network control, which is simple to be implemented in practical applications. Finally, the closed-loop system was proved to be SGUUB based on Lyapunov analysis.

5.2 Further Research

In this section, some research topics are proposed for further investigation:

- *Extension to output feedback control in the presence of unknown bounded disturbances and different lengths of each subsystems.*

  In Chapter 3, we investigated state feedback control scheme for a class of MIMO systems in state space representation with each subsystem is in strict feedback form, and the lengths of different subsystems are different. Then, output control scheme, which is easier for practical implementation, was investigated in the same Chapter. However, due to the transformation difficulty, the lengths of different subsystem are all the same and disturbances were not considered. Thus, it is meaningful to further investigate output feedback control schemes for the first class of system studied in Chapter 3.

- *NN control for general non-affine MIMO NARMAX model.*

  In Chapter 4, we investigated two kinds of MIMO NARMAX systems. Though the second class of MIMO systems studied is in non-affine form, it is a special class of non-affine MIMO systems due to the triangular form control inputs. Therefore, it is meaningful to further investigate NN control schemes for MIMO NARMAX systems in general form, i.e., the control gain matrix is in general form instead of in triangular form. For this class of MIMO systems, if the approximation based control schemes are to be implemented, the existence of the desired feedback controls should be guaranteed. For SISO non-affine systems studied in Chapter 2, by implicit function theorem, we know there is an IDFC, which can realize the exact tracking. However, for MIMO cases, the existence of the matrix form implicit function theorem is not clear yet. Therefore, the approximation based NN control for general MIMO non-affine system is a problem which needs to be further investigated. The major difficulty of that problem is how to guarantee/find the implicit desired feedback controls and develop corresponding weight tuning laws.

- *The implementation of MNN in the control schemes proposed.*

  Except RBF and MNN was used as function approximator in Chapter 2, HONNs were used as function approximators in the other chapters. However, HONN, the same as RBF NN, is a kind of so-called linear in the parameter (LIP) networks [50]. Noting the universal approximation ability of MNN, the use of MNN in those schemes is not only challenging but also of academic interest.

Appendix A

BIBO Stability and PE Condition

A.1 BIBO Stability

Consider the linear time varying discrete-time system given by

$$x(k+1) = A(k)x(k) + Bu(k), \qquad y(k) = Cx(k) \tag{1.1}$$

where $A(k), B$ *and* C *are appropriately dimensional matrices with* B *and* C *are constant matrices. Let* $\Phi(k_1, k_0)$ *be the state-transition matrix corresponding to* $A(k)$ *for system (1.1), i.e.* $\Phi(k_1, k_0) = \prod_{k=k_0}^{k_1-1} A(k)$*. If* $\|\Phi(k_1, k_0)\| < 1, \forall k_1 > k_0 \geq 0$*, then system (1.1) is (i) globally exponentially stable for the unforced system (i.e.* $u(k) = 0$*); and (ii) bounded input bounded output (BIBO) stable [102].*

A.2 Persistent Exciting Condition

The sequence $S(k)$ *is said to be persistent exciting if there is* $\bar{\lambda} > 0$ *and integer* $L > 0$ *such that*

$$\lambda_{\min}\Big[\sum_{k=k_0}^{k_0+L-1} S(k)S^T(k)\Big] \geq \bar{\lambda}, \quad \forall k_0 \geq 0 \tag{1.2}$$

where $\lambda_{\min}(M)$ *denotes the smallest eigenvalue of* M *[8].*

Bibliography

[1] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. biophys*, vol. 5, pp. 115–133, 1943.

[2] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, no. 4, pp. 303–312, 1989.

[3] T. Khanna, *Foundations of Neural Networks.* Addison Wesley, Reading, MA, 1990.

[4] K. I. Funahashi, "On the approximate qealization of continuous mappings by neoral networks," *Neural Networks*, vol. 2, pp. 183–192, 1989.

[5] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximator," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[6] A. Juditsky, Q. Zhang, B. Delyon, P. Y. Golorennec, and A. Benveniste, "Wavelets in identification: Wavelets, splines, neurones, fuzzies, how good for identification?," *Report*, vol. IRISA, Intern Publication no. 849, 1994.

[7] V. Cherkassky, D. Ghering, and F. Mulier, "Comparison of adaptive methods for function estimation from samples," *IEEE Trans. on Neural Networks*, vol. 7, no. 4, 1991.

[8] N. Sadegh, "A perception network for functional identification and control of nonlinear systems," *IEEE Trans. Neural Networks*, vol. 4, no. 6, pp. 982–988, 1993.

[9] A. R. Barron, "Universal approximation bounds for sugerpostion for a sigmoidal function," *IEEE Transactions on Information Theory*, vol. 39, pp. 430–445, 1993.

[10] A. R. Barron, "When do neural nets avoid the curse of dimensionality?," *NATO advanced study institute - From statistics to neural networks*, June 21-July 2,1993.

[11] A. S. Poznyak, W. Yu, E. N. Schanchez, and J. P. Perez, "Nonlinear adaptive trajectory tracking using dynamic networks," *IEEE Transactions on Neural Networks*, vol. 10, no. 6, pp. 1209–1211, 1999.

[12] B. Widrow and F. W. Smith, "Pattern-recognizing control systems," *Proc. of Computer and Information Sciences*, 1964.

[13] J. S. Albus, "A new approach to manipulator control: the cerebellar model articulation controller (cmac)," *Journal of Dynamic Systems*, pp. 220–227, 1975.

[14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," *Parallel Distributed Processing*, vol. 1, pp. 378–362, 1986.

[15] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamic systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.

[16] K. S. Narendra and S. Mukhopadhyay, "Adaptive control of nonlinear multivariable system using neural networks," *Neural networks*, vol. 7, no. 5, pp. 737–752, 1994.

[17] D. Psaltis, A. Sideris, and A. Yamamura, "A multilayered neural network controller," *IEEE Control Sys. Magazine*, vol. 8, pp. 17–21, 1998.

[18] L. Jin, P. N. Nikiforuk, and M. M. Gupta, "Direct adaptive output tracking control using multilayered neural networks," *IEE Proc. D*, vol. 140, no. 6, pp. 393–398, 1993.

[19] M. Polycarpou and P. Ioannou, "Learning and convergence analysis of neural-type structured networks," *IEEE Transactions on Neural Networks*, vol. 3, no. 1, pp. 39–50, 1992.

[20] F. C. Chen and H. K. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks," *IEEE Transactions on Automatic Control*, vol. 72, no. 7, pp. 791–807, 1995.

[21] A. U. Levin and K. S. Narendra, "Control of nonlinear dynamical systems using neural networks-part ii: observability, identification, and control," *IEEE Transactions on Neural Networks*, vol. 7, no. 1, pp. 30–42, 1996.

[22] C. J. Goh, "Model reference control of non-linear systems via implicit function emulation," *Int. J. Control*, vol. 60, pp. 91–115, 1994.

[23] T. Zhang, S. S. Ge, and C. C. Hang, "Neural-based direct adaptive control for a class of general nonlinear systems," *International Journal of Systems Science*, vol. 28, no. 10, pp. 1011–1020, 1997.

[24] S. S. Ge, C. C. Hang, and T. Zhang, "Nonlinear adaptive control using neural network and its application to CSTR systems," *Journal of Process Control*, vol. 9, pp. 313–323, 1998.

[25] S. S. Ge, C. C. Hang, and T. Zhang, "Adaptive neural network control of non-linear systems by state and output feedback," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 29, no. 6, pp. 818–828, December 1999.

[26] S. S. Ge and C. Wang, "Adaptive control of uncertain chua's circuits," *IEEE Transactions on Circuits and Systems, Part I Fundamental Theory and Applications*, vol. 47, no. 9, 2000.

[27] S. Ge, C. Wang, and T. Lee, "Adaptive backstepping control of a class of chaotic systems," *International Journal of Bifurcation and Chaos*, vol. 10, no. 5, pp. 1149–1156, 2000.

[28] S. Ge and C. Wang, "Adaptive control of uncertain chua's circuits," *IEEE Transactions on Circuits and Systems, Part I Fundamental Theory and Applications*, vol. 47, no. 9, pp. 1397–1402, Sep, 2000.

[29] C. Wang and S. S. Ge, "Adaptive synchronization of uncertain chaotic systems via backstepping design," *Chaos, Solitons and Fractals*, vol. 12, pp. 1199–1206, 2001.

[30] C. Wang and S. S. Ge, "Adaptive backstepping control of uncertain lorenz system," *International Journal of Bifurcation and Chaos*, vol. 11, no. 4, pp. 1115–1119, 2001.

[31] C. Wang and S. S. Ge, "Synchronization of two uncertain chaotic systems via adaptive backstepping," *International Journal of bifurcation and Chaos*, vol. 11, no. 6, pp. 1743–1751, 2001.

[32] C. Wang and S. S. Ge, "Uncertain chaotic system control via adaptive neural design," *International Journal of bifurcation and Chaos*, vol. 12, no. 5, pp. 1097–1109, 2002.

[33] P. E. Moraal and J. W. Grizzle, "Observer design for nonlinear systems with discrete-time measurements," *IEEE Transactions on Automatic Control*, vol. 40, no. 3, pp. 395–404, March, 1995.

[34] A. M. Dabroom and H. K. Khalil, "Output feedback sampled-data control of nonlinear systems using high-gain observers," *IEEE Transactions on Automatic Control*, vol. 46, no. 11, 2001.

[35] F. L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. London : Taylor & Francis, 1999.

[36] S. S. Ge, T. H. Lee, and C. J. Harris, *Adaptive Neural Network Control of Robotic Manipulators*. London: World Scientific, 1998.

[37] F. C. Sun and Z. Q. Sun, "Stable sampled-data adaptive control of robot arms using neural networks," *Journal of intelligent and robotic systems*, vol. 20, pp. 131–155, 1997.

[38] F. Sun, H. Li, and L. Li, "Robot discrete adaptive control based on dynamic inversion using dynamical neural networks," *Automatica*, vol. 38, pp. 1977–1983, 2002.

[39] A. Vemuri and M. Polycarpou, "Neural network based robust fault diagnosis in robotic systems," *IEEE Transactions on Neural Networks*, vol. 8, no. 6, pp. 1410–1420, 1997.

[40] A. Vemuri, M. Polycarpou, and S. Diakourtis, "Neural network based fault detection and accommodation in robotic manipulators," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 342–348, 1998.

[41] K. J. Hunt, D. Sbarbaro, R. Zbikowski, and P. J. Gawthrop, "Neural networks for control system- A survey," *Automatica*, vol. 28, no. 3, pp. 1083–1112, 1992.

[42] K. Najim, *Process Modeling and Control in Chemical Engineering.* Marcel Dekker: New York, 1989.

[43] G. Lera, "A state-space-based recurrent neural network for dynamic system identification," *Journal of Systems Engineering*, vol. 6, pp. 186–193, 1996.

[44] G. K. Kelmans, A. S. Poznyak, and A. V. Chernitser, "Local optimization algoritms in asymptotic control of nonlinear dynamic plants," *Automation and Remote Control*, vol. 38, pp. 1639–1652, 1977.

[45] A. M. Shaw and F. J. Doyle, "Multivariable nonlinear control applications for a high purity distillation column using a recurrent dynamic neuron model," *Journal of Process Control*, vol. 7, pp. 255–268, 1997.

[46] R. Ordonez, J. Zumberge, J. T. Spooner, and K. M. Passino, "Adaptive fuzzy control: Experiments and comparative analyses," *IEEE Transactions on Fuzzy Systems*, vol. 2, no. 5, pp. 147–187, 1997.

[47] J. T. Spooner, M. Maggiore, R. Ordonez, and K. M. Passino, *Stable Adaptive Control and Estimation for Nonlinear Systems – Neural and Fuzzy Approximator Techniques.* New Yori: Wiley, 2002.

[48] M. Brown and C. Harris, *Neurofuzzy Adaptive Modelling and Control.* London: Pretice-Hale International, 1994.

[49] S. Jagannathan and F. L. Lewis, "Identification of nonlinear dynamical systems using multilayered neural networks," *Automatica*, vol. 32, pp. 1707–1712, 1996.

[50] S. S. Ge, C. C. Hang, T. H. Lee, and T. Zhang, *Stable Adaptive Neural Network Control*. Boston, MA: Kluwer Academic, 2001.

[51] T. Parisini and R. Zoppoli, "Nonlinear modelling of complex large–scale plants using neural networks and stochastic approximation," *IEEE Transactions on Systems, Man and Cybernetics–Part A: Systems and Humans*, vol. 27, no. 6, pp. 750–757, 1997.

[52] M. Baglietto, T. Parisini, and R. Zoppoli, "Nonlinear modelling of complex large–scale plants using neural networks and stochastic approximation," *IEEE Trans. on Neural Networks*, vol. 12, no. 3, pp. 485–582, 2001.

[53] E. B. Kosmatopoulos, M. M. Polycarpou, M. A. Christodoulou, and P. A. Ioannou, "High-order neural network structures for identification of dynamical systems," *IEEE Trans. Neural Networks*, vol. 6, no. 2, pp. 422–431, 1995.

[54] R. M. Sanner and J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Trans. Neural Networks*, vol. 3, no. 6, pp. 837–863, 1992.

[55] G. Nurnberger, *Approximation by Spline Functions*. New York: Springer-Verlag, 1989.

[56] J. T. Spooner and K. M. Passino, "Stable adaptive control using fuzzy systems and neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 3, pp. 339–359, 1996.

[57] F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Transactions on Neural Networks*, vol. 7, no. 2, pp. 388–398, 1996.

[58] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and control*. Englewood Cliffs NJ: Prentice-Hall, 1984.

[59] B. B. Petersen and K. S. Narendra, "Bounded error adaptive control," *IEEE Transactions on Automatic Control*, vol. 27, pp. 1161–1168, 1982.

[60] C. Samson, "Stability analysis of adaptively controlled system subject to bounded disturbances," *Autopmatica*, vol. 19, pp. 81–86, 1983.

[61] K. S. Narendra and A. M. Annaswamy, "A new adaptive law for robust adaptation without persistent excitation," *IEEE Transactions on Automatic Control*, vol. 32, no. 2, pp. 134–145, 1987.

[62] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[63] K. Nam and A. Arapostathis, "A model-reference adaptive control scheme for pure-feedback nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 33, pp. 803–811, 1988.

[64] S. S. Sastry and A. Isidori, "Adaptive control of linearizable systems," *IEEE Transactions on Automatic Control*, vol. 34, no. 31, pp. 1123–1231, 1989.

[65] H. K. Khalil, *Nonlinear systems*. New Jersey, USA: Prentice Hall, 1996.

[66] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, *Nonlinear and Adaptive Control Design*. New York: Wiley and Sons, 1995.

[67] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive System*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[68] K. S. Narendra and S. Mukhopadhyay, "Adaptive control using neural networks and approximate models," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 475–485, 1997.

[69] T. Parisini and R. Zoppoli, "Neural networks for feedback feedforward nonlinear control systems," *IEEE Transactions on Neural Networks*, vol. 5, no. 3, pp. 436–449, 1994.

[70] J. T. Spooner and K. Q. Passino, "Adaptive control of a class of decentralized nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 41, no. 2, pp. 280–284, 1996.

[71] T. Parisini and R. Zoppoli, "Neural networks for nonlinear state estimation," *International Journal of Robust and Nonlinear Control*, vol. 4, no. 2, pp. 231–248, 1994.

[72] H. K. Khalil, “Adaptive output feedback control of nonlinear system represented by input-output models,” *IEEE Trans. Automatic Contr.*, vol. 41, no. 2, pp. 177–188, 1996.

[73] T. Parisini and R. Zoppoli, “Neural approximations for multistage optimal control of nonlinear stochastic systems,” *IEEE Transactions on Automatic Control*, vol. 41, no. 6, pp. 889–895, 1996.

[74] J. H. Braslavsky, R. H. Middleton, and J. S. Freudenberg, “Cheap control performance of a class of non-right-invertible nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 47, no. 8, pp. 1314–1319, 2002.

[75] M. M. Polycarpou and P. A. Ioannou, “A robust adaptive nonlinear control design,” *Automatica*, vol. 32, no. 3, pp. 423–427, 1996.

[76] M. M. Polycarpou, “Stable adaptive neural control scheme for nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 447–451, 1996.

[77] A. Yesidirek and F. L. Lewis, “Feedback linearization using neural networks,” *Automatica*, vol. 31, no. 11, pp. 1659–1664, 1995.

[78] T. Zhang, S. S. Ge, and C. C. Hang, “Design and performanue analysis of a direct adaptive controller for nonlinear systems,” *Automatica*, vol. 35, no. 11, pp. 1809–1817, 1999.

[79] I. Kanellakopoulos, P. V. Kokotovic, and A. S. Morse, “Systematic design of adaptive controller for feedback linearizable systems,” *IEEE Trans. Automat. Control*, no. 11, pp. 1241–1253, 1991.

[80] O. Adetona, E. Garcia, and L. H. Keel, “Stable adaptive control of unknown nonlinear dynamic systems using neural networks,” *Proceedings of the American Control Conference*, June 1999.

[81] I. Kanellakopoulos, P. V. Kokotovic, and R. Marino, “An extended direct scheme for robust adaptive nonlinear control,” *Automatica*, vol. 27, pp. 247–255, 1991.

[82] G. Campion and G. Bastin, "Indirect adaptive state feedback control of linearly parametrized nonlinear systems," *Int. J. Adaptive Control Signal Procese*, vol. 6, pp. 345–358, 1990.

[83] I. Kanellakopoulos, P. V. Kokotovic, and A. S. Morse, "Systematic design of adaptive controller for feedback linearizable systems," *IEEE Trans. Automat. Control*, vol. 36, no. 11, pp. 1241–1253, 1991.

[84] I. Kanellakopoulos, *Adaptive control of nonlinear systems.* PhD thesis, University of Illinois, Urbana, 1991.

[85] J. S. Freudenberg and R. H. Middleton, "Properties of single input, two output feedback systems," *International Journal of Control*, vol. 72, no. 16, pp. 1446–1465, 1999.

[86] G. Chen, J. Chen, and R. H. Middleton, "Optimal tracking performance for simo systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 10, pp. 1770–1775, 2002.

[87] S. S. Ge, C. C. Hang, and T. Zhang, "Stable adaptive control for nonlinear multivariable systems with a triangular control structure," *IEEE Transactions on Automatic Control*, vol. 45, no. 6, pp. 1221–1225, June, 2000.

[88] D. N. Godbole and S. S. Sastry, "Approximate decoupling and asymptotic tracking for MIMO systems," *Proceedings of the 32nd Conference on Decision and Control*, pp. 2754–2759, December., 1993.

[89] A. Isidori, *Nonlinear Control System.* Springer-Verlag, Berlin, 2nd edition 1989, 3rd edition, 1995.

[90] H. Nijmeijer and J. M. Schumacher, "The regular local noninteraction control problem for nonlinear control systems," *SIMA Journal of Control and Optimization*, vol. 24, pp. 1232–1245, 1986.

[91] H. Nijmeijer and A. J. van der Schaft, *Nonlinear Dynamical Control Systems.* New York: Springer-Verlag, 1990.

[92] H. Nijmeijer and W. Respondek, "Decoupling via dynamic compensation for nonlinear control systems," *IEEE Control and Decision Conference*, pp. 192–197, 1986.

[93] W. Lin and C. J. Qian, "Semi-global robust stabilization of nonlinear systems by partial state and output feedback," *Proceedings of the 37th IEEE Conference on Decision and Control*, pp. 3105–3110, December, 1998.

[94] A. Trebi-Ollennu and B. A. White, "Robust output tracking for MIMO nonlinear systems: An adaptive fuzzy systems approach," *IEE Proc. Control Theory Appl.*, vol. 144, pp. 537–544, November. 1997.

[95] J. B. D. Cabrera and K. S. Narendra, "Issues in the application of neural networks for tracking based on inverse control," *IEEE Transaction on Automatic Control*, vol. 44, no. 11, pp. 2007–2027, 1999.

[96] I. Kanellakopoulos, "A discrete-time adaptive nonlinear system," *IEEE Trans. Automat. Control*, vol. 39, no. 11, pp. 2362–2365, 1994.

[97] C. J. Goh and T. H. Lee, "Direct adaptive control of nonlinear systems via implicit function emulation," *Control-Theory and Advance Technology*, vol. 10, no. 3, pp. 539–552, 1994.

[98] P. C. Yeh and P. V. Kokotovic, "Adaptive control of a class of nonlinear discrete-time systems," *International Journal of Control*, vol. 62, no. 2, pp. 303–324, 1995.

[99] Y. Zhang, C. Y. Wen, and Y. C. Soh, "Discrete-time robust backstepping adaptive control for nonlinear time-varying systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 9, pp. 1749–1755, Septmber, 2000.

[100] S. S. Ge, G. Y. Li, and T. H. Lee, "Adaptive NN control for a class of nonlinear discrete-time systems," *Proceedings of the IEEE International Symposium on Intelligent Control*, pp. 97–102, 5-7, September, 2001.

[101] I. J. Leontaritis and S. A. Billings, "Input-output parametric models for nonlinear systems," *International Journal of Control*, vol. 41, no. 2, pp. 303–344, 1985.

[102] S. S. Ge, T. H. Lee, G. Y. Li, and J. Zhang, "Adaptive NN control for a class of discrete-time nonlinear systems," *International Journal of Control*, vol. 76, no. 4, pp. 334–354, 2003.

[103] S. A. Billings and W. S. F. Voon, "A prediction-error and stepwise regression algorithm for nonlinear systems," *Internatonal Journal of Control*, vol. 44, pp. 803–822, 1986.

[104] S. Chen, S. A. Billings, C. F. N. Cowan, and P. M. Grant, "Practical identification of NARMAX models using radial basis functions," *International Journal of Control*, vol. 52, pp. 1327–1350, 1990.

[105] S. Chen and S. A. Billings, "Representations of non-linear systems: the narmax model," *International Journal of Control*, vol. 49, no. 3, pp. 1013–1032, 1989.

[106] M. Fliess, "Reversible linear and nonlinear discrete-time dynamics," *IEEE Transactions on Automatic Control*, vol. 37, no. 8, pp. 1144–1173, 1992.

[107] J. S. Baras and N. S. Patel, "Robust control of set-valued discrete-time dynamical systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 1, pp. 61–75, 1998.

[108] D. S. Ortiz, J. S. Freudenberg, and R. H. Middleton, "Feedback limitations of linear sampled-data periodic digital control," *The International Journal of Robust and Nonlinear Control*, vol. 10, no. 9, pp. 729–745, 2000.

[109] J. X. Zhao and I. Kanellakopoulos, "Active identification for discrete-time nonlinear control-part i: Output-feedback systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 2, pp. 210–224, February, 2002.

[110] P. V. Zhivoglyadov, R. H. Middleton, and M. Fu, "Localization based switching adaptive control for time-varying discrete-time systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 4, pp. 752–755, 2000.

[111] S. Jagannathan and F. L. Lewis, "Multilayer discrete-time neural-net controller with guaranteed performance," *IEEE Trans. Neural Network*, vol. 7, no. 1, pp. 107–130, 1996.

[112] O. Adetona, E. Garcia, and L. H. Keel, "A new method for the control of discrete nonlinear dynamic systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 11, no. 1, pp. 102–112, January 2000.

[113] F. Sun, Z. Sun, and P. Y. Woo, "Stable neural-network-based adaptive control for sampled-data nonlinear systems," *IEEE Trans. Neural networks*, vol. 9, no. 5, pp. 956–968, 1998.

[114] J. Q. Gong and B. Yao, "Neural network adaptive robust control of nonlinear systems in semi-strict feedback form," *Automatica*, pp. 1149–1160, 2001.

[115] S. S. Ge, G. Y. Li, and T. H. Lee, "Adaptive NN control for a class of strict-feedback discrete-time nonlinear systems," *Automatica*, vol. 39, no. 5, pp. 807–819, May, 2003.

[116] G. C. Goodwin and D. Q. Mayne, "A parameter estimation perspective of continuous time model reference adaptive control," *Automatica*, vol. 23, no. 1, pp. 57–70, 1987.

[117] S. O. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence, and Robustness*. Englewood Cliff, NJ: Prentice-Hall, 1989.

[118] J. Q. Gong and B. Yao, "Adaptive robust control without knowing bounds of parameter variations," *Proceedings of the 38th IEEE conference on decision and control*, pp. 3334–3339, Dec 7-10, 1999.

[119] S. S. Ge, G. Y. Li, and T. H. Lee, "Adaptive NN control for a class of discrete-time nonlinear systems," *submitted to International Journal of Control*, 2001.

[120] R. Ortega, "Some remarks on adaptive neuro-fuzzy systems," *Int. J. of Adaptive contr. and Signal Processing*, vol. 10, pp. 79–83, 1992.

[121] F. C. Chen and H. K. Khalil, "Adaptive control of nonlinear systems using neural networks," *International Journal of Control*, vol. 45, no. 6, pp. 1299–1317, 1992.

[122] M. A. Henson and D. E. Seborg, *Nonlinear Process Control.* Upper Saddle River: NJ: Prentice-Hall, 1997.

[123] M. A. Henson and D. E. Seborg, "Input-output linearization of general nonlinear processes," *AICHE J*, vol. 36, no. 11, pp. 1753–1757, 1990.

[124] P. L. Lee and G. R. Sullivan, "Generic model control (gmc)," *Comput. and Chem Engng.*, vol. 12, p. 573, 1988.

[125] C. Kravaris and C. B. Chung, "Nonlinear state feedback synthesis by global input-output linearization," *AICHE Journal*, vol. 33, pp. 592–603, 1987.

[126] G. Bastin, "Adaptive non-linear control of a fed-bath stirred tank reactors," *Internationl Journal of Adaptive Control and Signal Processing*, vol. 6, pp. 273–284, 1992.

[127] K. Y. Rani and K. Gangiah, "Adaptive generic model control," *AICHE J.*, vol. 37, pp. 1634–1644, 1991.

[128] J. D. Morningred, B. E. Paden, D. E. Seborg, and D. A. Mellichamp, "An adaptive nonlinear predictive controller," *Proc. American Control Conference*, vol. 2, pp. 1614–1619, 1990.

[129] M. A. Henson and D. E. Seborg, "Adaptive nonlinear control of a ph neutralization process," *IEEE Trans, Control Systems Technology*, vol. 2, pp. 169–182, 1994.

[130] A. R. Teel, R. R. Kadiyala, P. V. Kokotovic, and S. S. Sastry, "Indirect techniques for adaptive input-output linearization of non-linear systems," *International Journal of Control*, vol. 53, pp. 193–222, 1991.

[131] R. Marino and P. Tomei, *Nonlinear Adaptive Design : Geometric, Adaptive, and Robust*. London: Printice Hall International (UK) Limited, 1995.

[132] R. Aris, *Elementary Chemical Analysis*. Englewood Cliffs: NJ: Prentice-Hall, 1969.

[133] G. Lightbody and G. W. Irwin, "Direct neural model reference adaptive control," *IEE Proc-Control Theory Appl*, vol. 142, no. 1, 1995.

[134] J. C. Kallkuhl and K. J. Hunt, *Discrete-time neural model structures for continuous-time nonlinear systems.* Neural Adaptive Control Technology, ed. R. Zbikowski and K. J. Hunt, Chapter1, Would Scientific, Singapore, 1996.

[135] S. Monaco and D. Normand-Cyrot, "On the sampling of a linear analytic control system," *Proceedings of the 24th Conference on Decision and Control*, pp. 1457–1463, Decemeber, 1985.

[136] C. E. Garcia and M. Morari, "Internal model control. 1. a unifying review and some new results," *Ind. eng. Chem. Proc. Des. Dev.*, vol. 21, pp. 308–323, 1982.

[137] M. Shacham, N. Brauner, and M. Cutlip, "Exothermic CSTRs: just how stable are the multiple steady states?," *Chemical Engineering Education*, Winter, 1994.

[138] T. D. Knapp, H. M. Budman, and G. Broderick, "Adaptive control of a CSTR with a neural network model," *Journal of Process Control*, vol. 11, pp. 53–68, 2001.

[139] S. Jagannathan and F. L. Lewis, "Discrete-time neural net controller for a class of nonlinear dynamical systems," *IEEE Transaction on Automatic Control*, vol. 41, no. 11, pp. 1693–1699, 1996.

[140] S. S. Ge, G. Y. Li, J. Zhang, and T. H. Lee, "Direct adaptive control for a class of MIMO nonlinear systems using neural networks," *submitted to IEEE Transactions on Automatic Control*, 2003.

[141] Z. Lin and A. Saberi, "Robust semi-global stabilization of minimum-phase input-output linearizable systems via partial state and output feedback," *IEEE Trans. Automat.Control*, vol. 40, no. 6, pp. 1029–1041, 1995.

[142] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, *Control of Robot Manipulators.* Macmillan, New York, 1993.

[143] R. Marino and P. Tomei, *Nonlinear Control Design.* London: Printice Hall, 1995.

[144] T. M. Apostol, *Mathematical Analysis.* Addison-Wesley, 1974.

Author's Publications

1. Adaptive NN control for a class of discrete-time non-linear systems, S. S. Ge, T. H. Lee, G. Y. Li and J. Zhang, International Journal of Control (Taylor and Francis Ltd), Vol. 76, No. 4, pp. 334-354, 2003.

2. State feedback NN control of a class of discrete MIMO nonlinear systems with disturbances, S. S. Ge, J. Zhang and T. H. Lee, accepted by IEEE Transactions on Systems, Man and Cybernetics, Part B, 2004.

3. Adaptive MNN control for a class of non-affine NARMAX systems with disturbances, S. S. Ge, J. Zhang and T. H. Lee, accepted by Systems and Control Letters, 2004.

4. Direct adaptive control for a class of MIMO nonlinear systems using neural networks, S. S. Ge, G. Y. Li, J. Zhang and T. H. Lee, accepted by IEEE Transactions on Automatic Control, 2004.

5. Output feedback control of a class of discrete MIMO nonlinear systems with triangular form inputs, J. Zhang, S. S. Ge and T. H. Lee, *submitted to* IEEE Transactions on Neural Networks, 2003.

6. Direct RBF neural network control of a class of discrete-time non-affine nonlinear systems, J. Zhang, S. S. Ge and T. H. Lee, Proceedings of the American Control Conference, 2002, Vol. 1, pp. 424-429, 8-10 May, 2002, Anchorage, Alaska, USA.

7. Direct MNN control of a class of discrete-time non-affine nonlinear systems, S. S. Ge, J. Zhang and T. H. Lee, the 15th IFAC World Congress on Automatic Control, 21-26 July, 2002, Barcelona, Spain.

8. State feedback NN control of non-affine nonlinear system with zero dynamics, S. S. Ge and J. Zhang, Proceedings of the American Control Conference, 4-6 June, 2003, Denver, Colorado USA.

9. Direct MNN control of continuous stirred tank reactor based on input-output model, S. S. Ge, J. Zhang and T. H. Lee, Proceedings of the Society of Instrument and Control Engineers (SICE), pp. 3066-3071, 5C7 August, 2002, Osaka, Japan.

10. Output NN control for a class of discrete-time nonlinear systems with disturbances, J. Zhang, S. S. Ge and T. H. Lee, Proceedings of IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, 8-13 October, 2003, Changsha, Hunan, China.

11. Direct adaptive control for a class of MIMO nonlinear systems using neural networks, S. S. Ge, J. Zhang and T. H. Lee, Proceedings of the International Conference on Humanoid, Nanotechnology, Information technology, Communication and control, Environment, and Management (HNICEM), 29 March-1 April, 2003, Manila, Philippines.