國立交通大學

電機與控制工程系

博士論文

具自動建構特性之模糊與類神經網路控制架構於非線性動態系統之應用

Fuzzy and Neural Network Control Schemes with Automatic Structuring Process for Nonlinear Dynamic Systems

研究生：陳品程

指導教授：李祖添　教授

王啟旭　教授

中華民國九十七年十月

具自動建構特性之模糊與類神經網路控制架構於非線性動態系統之應用

Fuzzy and Neural Network Control Schemes with Automatic Structuring Process for Nonlinear Dynamic Systems

研究生：陳品程
指導教授：李祖添
王啟旭

Student : Pin-Cheng Chen
Advisor(s) : Tsu-Tian Lee
Chi-Hsu Wang

國立交通大學
電機與控制工程系
博士論文

A Dissertation
Submitted to Department of Electrical and Control Engineering
College of Electrical Engineering
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in

Electrical and Control Engineering

October 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年十月

具自動建構特性之模糊與類神經網路控制架構於非線性動態系統之應用

研究生：陳品程　　　　指導教授：李祖添 博士
王啟旭 博士

國立交通大學電機與控制工程系博士班

摘要

為解決非線性系統控制問題，本論文發展兩個嶄新的控制架構。首先針對非仿射的非線性動態系統，提出具自我建構特性的強健適應性模糊控制架構。此架構中的控制器包含一個自我建構的模糊控制器和一個強建控制器。自我建構的模糊控制器用來近似未知的系統非線性，並可以自動刪除及產生模糊規則以建立簡潔的模糊規則庫;強健控制器用來達成 L_2 追蹤表現，並抑制誤差至要求的範圍以使系統穩定。我們舉出四個例子顯示此控制架構不但能有良好的控制表現，也可大量減少運算量。其次針對仿射的非線性動態系統，提出使用以霍普菲爾為基礎的動態類神經網路的直接適應性控制架構。在此架構中，以霍普菲爾為基礎的動態類神經網路用來近似一個理想控制器；監督控制器則用來抑制近似誤差和外界干擾的影響。藉由 Lypunov 方法可堆導出適應法則，用以調整網路的權重值使得系統穩定。經由適當地選取參數，可將追蹤物差抑制到要求的範圍內。經由模擬證實了此架構的可行性及良好效果。僅含一個神經元的以霍普菲爾為基礎的動態類神經網路使得此架構易於以硬體實現，另外，我們亦探討由無自我回授神經元建構而成的以霍普菲爾為基礎的動態類神經網路。經比較發現，採用具有自我回授神經元的以霍普菲爾為基礎的動態類神經網路之控

制架構，其控制表現較佳。值得注意的是，本論文中所提出的自我建構模糊系統和固定架構的以霍普菲爾為基礎的動態類神經網路皆不需要專家的知識或是試誤過程來決定其架構，因此解決了模糊系統和類神經網路的架構問題。

Fuzzy and Neural Network Control Schemes with Automatic Structuring Process for Nonlinear Dynamic Systems

Student：Pin-Cheng Chen Advisor(s): Dr. Tsu-Tian Lee
Dr. Chi-Hsu Wang

Department of Electrical and Control Engineering
National Chiao Tung University

ABSTRACT

In this dissertation, two novel control schemes are proposed to solve the control problems of nonlinear systems. The first is a robust adaptive self-structuring fuzzy control (RASFC) scheme for nonaffine nonlinear systems, and the second is a direct adaptive control scheme using Hopfield-based dynamic neural network (DACHDNN) for affine nonlinear systems. The RASFC scheme is composed of a robust adaptive controller and a self-structuring fuzzy controller. The design of the self-structuring fuzzy controller design utilizes a novel self-structuring fuzzy system (SFS) to approximate the unknown plant nonlinearity, and the SFS can automatically grow and prune fuzzy rules to realize a compact fuzzy rule base. The robust adaptive controller is designed to achieve a L_2 tracking performance with a desired attenuation level to stabilize the closed-loop system. Four examples are presented to show that the proposed RASFC scheme can achieve favorable tracking performance and relieve heavy computational burden. In the DACHDNN, a Hopfield-based dynamic neural network is used to approximate the ideal controller, and a compensation controller is used to suppress the effect of approximation error and disturbance. The weightings of the Hopfield-based dynamic neural network are on-line tuned by the adaptive laws derived in the Lyapunov sense, so that the stability of the closed-loop system can be guaranteed. The tracking error can be attenuated to a desired level by adequately selecting some parameters. The case of Hopfield-based neural network without the self-feedback loop is also studied and shown to have inferior results than those of Hopfield neural network with the self-feedback loop. Simulation results illustrate the applicability of the proposed control scheme. The Hopfield-based dynamic neural network with a parsimonious structure has the best potential be realized in hardware. It should be emphasized

that the self-structuring property of the SFS and the fixed parsimonious structure of the DACHDNN eliminate the need for expert’s knowledge or error-trial process and thus provide perfect solutions to the structuring problems of fuzzy systems and neural networks, respectively.

Acknowledgement

經過了五年多的日子，終於能順利取得博士學位，心中百感交集，除了喜悅之外，更充滿了感謝。謝謝我的指導教授李祖添老師。除了在研究上的指導，老師自律嚴謹的為人處世、治學態度是我最好的典範。幾次在老師所承辦的大型國際學術會議中參與準備工作，讓我學習到做事的方法以及開闊的國際視野。雖然老師非常繁忙，對於我們的生活、畢業、未來出路方面，老師仍然相當關心並盡力給予我們幫助。謝謝我的另外一位指導教授王啟旭老師。王啟旭老師深具創意的思考為我的研究提供了莫大的啟發，與學生相處時親切爽朗的態度令人如沐春風，對於我的生活及身體健康的關懷，更使人備感溫暖。對於兩位老師的栽培之恩，我永生難忘。

感謝我的口試委員：徐保羅教授、蘇順豐教授、王偉彥教授、以及呂藝光教授，對於我的畢業論文提供了許多寶貴的意見及不同的思考方向，使得論文內容能更加完整與正確。其中尤其要感謝王偉彥老師，自從我從輔大碩士班畢業以後，您仍然不吝給予我許多指導及幫助，身為您的學生我深感幸運。

此外，還要感謝許駿飛學長，在您的指導下經過長時間的相互討論與分享，才能慢慢建立起我的研究理論基礎，沒有您的熱忱幫助，我不會這麼順利地取得博士學位。謝謝炳榮，七年來無論在研究或生活上，都合作無間，互相幫忙、彼此關懷；謝謝偉竹，總計八年的室友，後四年有炳榮的加入，三個人一起瘋癲笑鬧、一起快樂傷心過的青春，永難忘懷。謝謝我所深愛過的，那些快樂和悲傷都是我珍貴的寶物。謝謝彭昭暐學長、溫裕弘學長、保村學長、李宜勳學長、欣翰、文真、雅齡、逸幗、詠健、東璋、得裕、堃能、其他的學弟妹們，有你們在，研究生活更充實愉快。謝謝助理天鳳、鳳儀、佳明，謝謝在這段期間所有關心我的朋友和直接或間接給予我幫助的人們。

最後要謝謝我最愛的父母與家人，你們給予我無條件的愛，永遠支持我、給予我鼓勵，是我最安全的避風港。我願能以同樣深切的愛回報你們，並與你們分享今日我小小的成就。

Table of Contents

Abstract in Chinese i
Abstract in English iii
Acknowledgement v

Table of Content …………………………………………………………………. vi
List of Figures …………………………………………………………………. viii
List of Tables …………………………………………………………………. x

1. Introduction

1.1 Background and Motivation……………………………………………………….. 1
1.2 Major Works…………………………………………………………………….. 4
1.3 Dissertation Overview……………………………………………………………. 6

2. Robust Adaptive Self-structuring Fuzzy Control Design for Nonaffine Nonlinear Systems……………………………………………………………. 7
2.1 Problem Formulation…………………………………………………………… 7
2.2 Self-structuring Fuzzy System……………………………………………………... 9
2.2.1 Description of Fuzzy System……………………………………………. 9
2.2.2 Structure Learning Algorithm……………………………………………. 10
2.3 Design of RASFC……………………………………………………………….. 16
2.3.1 Fuzzy Approximation……………………………………………………. 16
2.3.2 Parameter Learning Algorithm…………………………………………... 20
2.4 Simulation Results……………………………………………………………….. 25

3. Direct Adaptive Control Design Using Hopfield-Based Dynamic Neural Network for Affine Nonlinear Systems……………………………………….. 40
3.1 Hopfield-Dynamic Neural Network……………………………………………... 40
3.1.1 Description of DNN Model……………………………………………… 40
3.1.2 Hopfield-based DNN Approximator…………………………………….. 42
3.2 Problem Formulation…………………………………………………………….. 43

3.3 Design of DACHDNN………………………………………………………... 44
3.4 Simulation Results………………………………………………………. 50
3.5 Performance analysis of Hopfield-based DNNs with and without the self-feedback loop…………………………………………………………….. 55

4. Conclusions And Future Works……………………………………………….. 60

References 62

Vita 67

Publication List 68

1896

List of Figures

Fig. 2-1(a) Improper fuzzy clustering of input variable *Xj*………………………... 11

Fig. 2-1(b) Newly created membership function……………………..……………. 11

Fig. 2-2 The flowchart of the self-structuring algorithm for the SFS…………... 15

Fig. 2-3(a) The recurrent fuzzy system……………………………………………. 17

Fig. 2-3(b) The static fuzzy system………………………………………………... 17

Fig. 2-4 The block diagram of RASFC for nonaffine nonlinear systems……… 24

Fig. 2-5 Approximation results in Example 2-1. (a) is approximation results of Condition 1a; (b) is approximation results of Condition 1b; (c) is approximation results of Condition 1c; (d) is the approximation error; (e) is the number of fuzzy rules……………………………………….. 26-27

Fig. 2-6 Approximation results in Example 2-2. (a) is approximation results of Condition 1a; (b) is approximation results of Condition 1b (c) is the approximation error; (d) is the number of fuzzy rules; (e) is the contribution and significance index of a certain rule………………….. 29-31

Fig. 2-7 Simulation results of Case 3a in Example 2-3. (a) is the response of state x_1; (b) is the response of state x_2; (c) is the control input; (d) is the number of fuzzy rules………………………………………………….. 32-33

Fig. 2-8 Simulation results of Case 3b in Example 2-4. (a) is the response of state x_1; (b) is the response of state x_2; (c) is the control input; (d) is the number of fuzzy rules………………………………………………….. 33-34

Fig. 2-9 Simulation results of Case 4a in Example 2-4. (a) is the response of state x_1; (b) is the response of state x_2; (c) is the control input; (d) is the number of fuzzy rules………………………………………………….. 36-37

Fig. 2-10 Simulation results of Case 4b in Example 2-4. (a) is the response of state x_1; (b) is the response of state x_2; (c) is the control input; (d) is the number of fuzzy rules…………...……………………………………... 37-38

Fig. 3-1 The structure of the dynamic neural network………………………… 41

Fig. 3-2 The Block diagram of the DACHDNN………………………………… 45

Fig. 3-3 The Electric circuit of Hopfield-based DNN containing only a single neuron……………………………………………………………………... 45

Fig. 3-4 The Phase plane of uncontrolled chaotic system……………………… 51

Fig. 3-5 Simulation results of Example 3-1. (a) is the response of state x_1; (b) is the response of state x_2; (c) is the control input; (d) show the trained weightings………………………………………………………… 52-53

Fig. 3-6 Simulation results of Example 3-2. (a) is the response of state x_1; (b) is the response of state x_2; (c) is the response of state x_3; (d) is the control input; (e) show the trained weightings………………………... 54-55

Fig. 3-7 Simulation results of Example 3-1 using Hopfield-based DNN without the feedback loop. (a) is the response of state x_1; (b) is the response of state x_2; (c) is the norm of tracking error…………………………….. 57

Fig. 3-8 Simulation results of Example 3-3 using Hopfield-based DNN without feedback. (a) is the response of state x_1; (b) is the response of state x_2; (c) is the response of state x_3 (d) is the norm of tracking error.. 58-59

List of Tables

Table 2-1. Three conditions in Example 2-1…………………………………….... 25
Table 2-2 Two conditions in Example 2-2……………………………………... 28
Table 2-3 Comparison between two cases in Example 2-3………………………. 31
Table 2-3 Comparison between two cases in Example 2-4………………………. 35

Chapter 1

Introduction

1.1 Background and Motivation

Recently, control system design for nonlinear systems has attracted a lot of research interests. Many remarkable results have been obtained, including feedback linearization [1], adaptive backstepping design [2], fuzzy logic control [3], neural network control [4], and fuzzy-neural control [5]. In general, nonlinear systems can be classified into two categories, affine nonlinear systems, i.e., systems characterized by inputs appearing linearly in the system state equation, and nonaffine nonlinear systems, where the control input appears in a nonlinear fashion [6]. Many systems encountered in engineering, by nature or by design, are affine systems, such as inverted pendulum systems [3], mass-spring-damper system [7-8], chua’s circuit [9-10], straight-arm robot [11], DC-to-DC converter [12], etc. On the other hand, nonaffine systems are quite common in the real world, such as Van de Pol oscillator [13-15], magnetic servo levitation systems [16], aircraft flight control systems [17], biochemical process [18], etc.

Fuzzy system (FS) which adopts human experience and human decision-making behavior has been widely recognized as a powerful tool in industrial control, commercial prediction, image processing applications, etc. [19-21]. To build a FS, there are two different phases to be carried out. The first is the structuring phase, which is used to construct the structure of FS, and the second is the parameter phase, which is used to determine the parameters of FS. Constructing the structure of FS is mainly to determine the optimal partition of fuzzy sets and the minimum number of fuzzy rules to achieve favorable performance. The adjustments of the parameters involve the tuning of the consequences of the fuzzy rules, the centers, widths, slopes of membership functions, etc. Traditionally, these two phases are performed by human experts or experienced operators. However, consulting experts may be difficult and expert knowledge is either unavailable or not helpful enough to achieve favorable performance. Having achieved many practical successes, fuzzy control (FC) using FS has still not been viewed as rigorous because it lacks a systematic design procedure to determine proper membership functions with fuzzy rules, and the way to guarantee the global stability. Adaptive fuzzy control (AFC) has been extensively studied to tackle this problem [21-26]. Adaptive fuzzy system can approximate the unknown system dynamics or ideal controller through learning in the Lyapunov sense, and thus the global stability can be guaranteed.

Although the control performances in [21-26] are acceptable, the structures of the FSs need to be predefined by a time-consuming trial-and-error process. Generally speaking, a more favorable performance requires more fuzzy rules, but this may lead to heavy computational burden. On the contrary, a FS with small fuzzy rule base may result in a poor approximation.

To solve the problem of structure determination, many researchers have focused their efforts on the self-structuring fuzzy system (SFS) and obtained some valuable results [27-31]. In [27], the structure learning phase aims at minimizing the number of rules generated and the number of fuzzy sets in the universe of discourse. A structure learning algorithm is proposed based on fuzzy similarity measure, and fuzzy rules can be created from the training data. In [28], the structure identification is accomplished automatically based only on Q-learning, which is the most important category of reinforcement learning algorithm. The basic fuzzy rules are used as starting points to reduce the number of iterations used to find an optimal fuzzy controller. In [29], the firing strength of a rule is used as the degree measure to judge whether or not to simultaneously generate a new membership function for every input variable (or equivalently, to generate a new rule.) Then, if the newly generated membership of the first input variable fails to pass the similarity checking, all new membership functions are abandoned. In [30], parameter and structure learning are performed sequentially for the proposed fuzzy neural network. That is, the fuzzy neural network is initially constructed to contain all possible fuzzy rules, and then the parameter training is performed. After the parameter training is completed, a pruning process is performed to delete redundant rules and thus leads to a concise fuzzy rule base. Note that the initially constructed rule base contains incompatible rules, i.e., the rules with the same antecedent but different consequents. The rule pruning strategy is that if the centroid of a set of incompatible rules is in the support of a consequent (an output fuzzy set), the corresponding fuzzy rule is remained and all other incompatible rules are pruned. In [31], the authors modified the fuzzy neural network proposed in [30] and proposed rule pruning scheme that always produces a rule set without incompatible rules.

However, although some achievements have been made in these works, there are still some problems need to be solved. In [27], the performance of the proposed neural fuzzy system is acceptable, but the back propagation learning algorithm cannot guarantee the global stability. In [28], during the training process, prior knowledge of fuzzy rules is needed to keep safe operation of the controlled system with fast convergence speed of parameters. In [29], the simplified similarity checking to reduce the complexity of the algorithm may weaken the power of the checking itself. In [30], because the connection weights of the network are unrestricted in sign, incompatible rules may be retained even rule pruning process is performed. This is contradictory to the basic design philosophy of fuzzy systems. Besides, the proposed sequential learning scheme is suitable for offline instead of online operation. In [31], although the fuzzy neural network in [30] is modified to guarantee a compatible rule base, the searching space for the connection weights is restricted to R^+. This may harm the capability of the proposed network to lower the value of residual square error. The common drawback in [27-31] is that the structuring learning phase conducts either rule generation or rule reduction, instead of both.

Recently, research interest has been increasing towards the usage of neural network (NN) for controlling a wide class of complex nonlinear systems under the restriction that complete model information is not available [32-36]. Due to their massive parallelism, fast adaptability, and inherent approximation capabilities, NN seems to be a feasible solution to the control problem of nonlinear systems. However, the structuring problem of NNs, which mainly refers to determining the number of the neurons, is an annoying problem. This choice faces a similar dilemma as the choice of fuzzy rule number in the FS design. Generally speaking, more favorable performance requires more neurons, but this may lead to a complicated network structure and heavy computational burden. On the contrary, an NN with too few neurons in the hidden layer(s) will make it hard for the network to recognize the relationships between the output and input parameters, and thus result in a poor approximation. In general, the number of neurons is chosen empirically and apparently not optimized.

Two major classes of NNs, static and dynamic NNs, have become enormously important in recent years. In static NNs, which are also called feedforward NNs, signals flows from the input units to the output units in a forward direction. In dynamic NNs, dynamic elements are involved in the structure of the NN, for example, in the form of feedback connections. Some static neural networks (SNNs), such as feedforward fuzzy neural network (FNN) or feedforward radius basis function networks (RBFN), are frequently used as powerful tools for modeling the ideal control input or nonlinear functions of systems. Some results are shown in [39-42]. Although feedforward FNNs and RBFNs have achieved much theoretical success, they leave some space for improvement. The complex structures of feedforward FNNs and

RBFNs make the practical implementation of the control schemes infeasible, and usually a large number of neurons are needed in the hidden layers of SNNs (in general more than the dimension of the controlled system). The other well-known disadvantage is that SNNs are quite sensitive to the major change that never learned in the training phase.

Despite the immense popularity of the usage of SNNs, some researchers adopt dynamic neural networks (DNNs) to solve the control problem of nonlinear systems. An important motivation is that a smaller DNN is possible to provide the functionality of a much larger SNN [43]. In addition, SNNs are unable to represent dynamic system mapping without the aid of tapped delay, which results in long computation time, high sensitivity to external noise, and a large number of neurons when high dimensional systems are considered [44]. This drawback severely affects the applicability of SNNs to system identification, which is the central part in some control techniques for nonlinear systems. On the other hand, since DNNs have dynamic memory, they have good performance on identification, state estimation, trajectory tracking, etc., even with the unmodeled dynamics. In [45-49], researchers first identify the nonlinear system according to the measured input and output, and then calculate the control low based on the NN model. The output of the nonlinear system is forced by the control law to track either a given trajectory or the output of a reference model. However, there are still some drawbacks. In [45], painful off-line identification is needed for the proposed approach, and the proposed control scheme deals with only singular perturbed systems. In [46], some strong assumptions are made, such as those ones related to the magnitude of the synaptic weightings and the stability of the closed-loop dynamics of the neural model. In [47], although both identification and tracking errors are bounded, it seems that the control performance is not satisfactory in the simulations. In [48], two DNNs are utilized in the iterative learning control system to approximation the nonlinear system and mimic the desired system output, respectively, thus increasing the complexity of the control scheme and computation loading. The work in [49] requires a prior knowledge of the strong relative degree of the controlled nonlinear system. Besides, an additional filter is needed to obtain the higher derivatives of the system output. These drawbacks impose the restriction on the applicability of the above works to practical implementation.

1.2 Major Works

To solve the structuring problem of FSs, this dissertation first proposes a novel SFS, which is used to approximate the unknown plant nonlinearity. The SFS considers both the growing and pruning of fuzzy rules. In fact, it is possible that some rules are less or never fired throughout the operation of FS. These redundant rules, which make no meaningful contributions to the system output, are insignificant and thus should be removed to ease computational load. Secondly, a robust adaptive self-structuring fuzzy control (RASFC) scheme is proposed for a SISO nonaffine nonlinear system. A robust adaptive controller is merged into the control law to achieve L_2 tracking performance with a desired attenuation level of tracking error. This L_2 tracking performance can provide a clear expression of tracking error in terms of the sum of lumped uncertainty and external disturbances, which has not been shown in previous works [50-51]. Moreover, all control parameters of the RASFC system are tuned on-line according to the adaptive laws derived in the Lyapunov sense to achieve favorable fuzzy approximation. Then, four examples are presented. For the purpose of interpreting the novel self-structuring algorithm, approximations of unknown nonlinear functions are performed in Examples 2-1 and 2-2 to illustrate the rule generation and pruning capabilities of the SFS. In Examples 2-3 and 2-4, tracking control for two nonaffine nonlinear systems is provided to verify the effectiveness of the proposed RASFC scheme. To highlight the power of the proposed SFS, an adaptive FS with fixed number of rules and an SFS which can only automatically grow rules are also adopted in the last two examples for comparison purpose. Simulation results show that the proposed RASFC can achieve favorable tracking performance with a compact fuzzy rule base profited from the self-structuring algorithm. Comparing with adaptive fuzzy system with fixed number of rules and SFS which can only grow rules, the proposed SFS with both rule growing and pruning capabilities can relieve computational load, yet still maintain the desired tracking accuracy.

To fix the drawbacks of the NN control designs mentioned in the preceding paragraphs, and at the same time, solve the inherent structuring problem of NNs, we then propose a direct adaptive control scheme using Hopfield-based dynamic neural networks (DACHDNN) for SISO nonlinear systems. Direct adaptive control is one of the important categories of adaptive control. In direct adaptive control, the parameters of the controller are directly adjusted to reduce some norm of the output error between the plant and the reference model. The Hopfield model was first proposed by Hopfield J.J. in 1982 and 1984 [52-53]. Because a Hopfiled circuit is quite easy to be realized and has the property of decreasing in energy by finite number of node-updating steps, it has many applications in different fields. The Hopfield-based DNN can be viewed as a special kind of DNNs. The control object is to force the system output to follow a given reference signal. The ideal controller is approximated by the internal state of a Hopfield-based DNN, and a compensation controller is used to compensate the effect caused by approximation error and the bounded external disturbance. The synaptic weightings of the Hopfiled-based DNN are on-line tuned by adaptive laws derived in the Lyapunov sense. The control law and adaptive laws provide semi-global stability for the closed-loop system with external disturbance. Furthermore, the tracking error can be attenuated to a desired level by adequately choosing parameters of the control law. The cases of Hopfield-based DNN without the self-feedback loop are also studied. We show that these cases have inferior results than those of Hopfield-based DNN with the self-feedback loop. The main contributions of the DACHDNN are summarized as follows. 1) The structure of the used Hopfield-based DNN is quite parsimonious. It contains only a single neuron, which is much less than those contained in SNNs or other DNNs for nonlinear system control. It is shown in the simulation that such a parsimonious structure of Hopfield-based DNN does not destroy the system performance. 2) The simple Hopfield circuit greatly improves the applicability of the whole control scheme for practical implementation. 3) No strong assumptions or prior knowledge of the controlled plant are needed in the development of DACHDNN.

1.3 Dissertation Overview

The rest of this dissertation is organized as follows. Chapter 2 describes the design procedure of the RASFC scheme for nonaffine nonlinear systems. The structure learning phase performed by the SFS is introduced. The adaptive laws to tune the parameters, including the means and variances of membership functions and the single consequents of the fuzzy rules and, are derived. The stability analysis and example are also provided in this chapter. The DACHDNN is developed in Chapter 3. The adaptive laws to tune the synaptic weightings are derived. The stability analysis and examples are also provided in this chapter. Finally, conclusions and future works are stated in Chapter 4.

Chapter 2

Robust Adaptive Self-structuring Fuzzy Control Design for Nonaffine Nonlinear Systems

Reviewing some literatures on nonaffine nonlinear system control, we find some problems left to be addressed. In [50], although the system stability is guaranteed in the Lyapunov sense, the un-measurable term in the adaptive law needs to be approximated. This will make the system stability questionable. Even the system stability can be guaranteed, the tracking error is only ultimately uniformly bounded. In [51], the tracking error is uniformly asymptotically stable, but the robust controller to compensate the external disturbance causes the chattering of control input. Although the authors in [50] suggested some remedies to reduce the chattering, the tracking error may not be UAS due to these remedies.

In this chapter, we aim at solving the control problem of SISO nonaffine nonlinear systems. An adaptive fuzzy control scheme is developed to achieve this goal, and the resulting structuring problem of fuzzy systems is also solved by a proposed self-structuring fuzzy system (SFS). The automatic rule pruning and growing functions of the SFS are discussed and separately illustrated in the Examples 2-1 and 2-2 to give more insights. Using the proposed SFS, we will show how a novel robust adaptive self-structuring fuzzy control (RASFC) scheme can remarkably reduce the computational burden without sacrificing the favorable control performance for SISO nonaffine nonlinear systems.

2.1 Problem Formulation

Consider a single-input and single-output (SISO) nonaffine nonlinear system

$$x^{(n)} = f(\mathbf{x}, u) + d \tag{2-1}$$

where $\mathbf{x} = [x\,\dot{x} \ldots x^{(n-1)}]^T$ is the measurable state vector of the system on a domain $\mathbf{\Omega}_{\mathbf{x}} \subset R^n$,

$f(\mathbf{x},u):\boldsymbol{\Omega}_{\mathbf{x}}\times R\rightarrow R$ is the smooth unknown nonlinear function, *u* is the control input, and *d* is the bounded external disturbance. Here the single output is *x*. It should be noted that *f*(**x**, *u*) is an implicit function with respect to *u*. Feedback linearization is performed by rewriting (2-1) as

$$x^{(n)}=zu+\Delta(\mathbf{x},u)+d \tag{2-2}$$

where *z* is a constant to be designed and $\Delta(\mathbf{x},u)=f(\mathbf{x},u)-zu$. Here we assume that $\frac{\partial f(\mathbf{x},u)}{\partial u}$ is nonzero for all $(\mathbf{x},u)\in\Omega_{\mathbf{x}}\times R$ with a known sign. Without losing generality, we further assume that [51, 54-55]

$$\frac{\partial f(\mathbf{x},u)}{\partial u}>0 \tag{2-3}$$

for all $f(\mathbf{x},u)\in\boldsymbol{\Omega}_{\mathbf{x}}\times R$. Note that for the nonaffine systems with property $\frac{\partial f(\mathbf{x},u)}{\partial u}<0$, the control scheme can be easily defined with minor modifications discussed in section 4. The control objective is to develop a control scheme for the nonaffine nonlinear system (2-1) so that the output trajectory *x* can track a given trajectory x_c closely. The tracking error is defined as

$$e=x_c-x \tag{2-4}$$

If the system dynamics and the external disturbance are well known, the ideal feedback controller can be determined as

$$u_{id}=\frac{1}{z}[u_{lc}-d-\Delta(\mathbf{x},u)] \tag{2-5}$$

where

$$u_{lc}=x_c^{(n)}+\mathbf{k}^T\mathbf{e} \tag{2-6}$$

with $\mathbf{e}=[e\,\dot{e}\ldots e^{(n-1)}]^T$ and $\mathbf{k}=[k_n\,k_{n-1}\ldots k_1]^T$. Applying (2-5) to (2-2) and using (2-4) yield the following error dynamics

$$e^{(n)}+k_1e^{(n-1)}+\cdots+k_ne=0 \tag{2-7}$$

If k_i, *i*=1, 2, …, *n* are chosen so that all roots of the polynomial $H(s)\underset{=}{\Delta}s^n+k_1s^{n-1}+\cdots+k_n$ lie strictly in the open left half of the complex plane, then $\lim_{t\rightarrow\infty}e(t)=0$ can be implied for any initial conditions. However, since $\Delta(\mathbf{x},u)$ and the external disturbance *d* may be unknown or perturbed, the ideal feedback controller u_{id} in (2-5) cannot be implemented. Thus, to achieve the control objective, an SFS is designed to estimate the system uncertainty $\Delta(\mathbf{x}, u)$ in (2-2).

2.2 *Self-structuring Fuzzy System*

2.2.1 Description of Fuzzy System

FSs are attractive candidates for the systems that are structurally difficult to model due to inherent non-linearity and model complexities. Typically, a FS includes four well-known stages: a fuzzifier, a rule base, an inference engine, and a defuzzifier. The rule base is the collection of fuzzy rules which characterize the simple input-output relation of the system. Note that the self-structuring algorithm introduced in this section is applicable to multi-input and multi-output (MIMO) FS. However, without losing generality and to simplify the notation, a multi-input and single-output (MISO) FS is adopted to describe the algorithm. A MISO FS can be are expressed as [19]:

$$\text{Rule}_{i_1,i_2,\ldots,i_m}: \text{IF } X_1 \text{ is } F_1^{i_1} \text{ and } X_2 \text{ is } F_2^{i_2} \text{ and } \ldots \text{ and } X_m \text{ is } F_m^{i_m} \text{ THEN } y \text{ is } \alpha_{i_1,i_2,\ldots,i_m} \qquad \text{(2-8)}$$

where X_j, j=1, 2, …, m are input variables; y is output variable; $\alpha_{i_1,i_2,\cdots,i_m}$ is the crisp singleton consequent; $F_j^{i_j}$ is the fuzzy sets characterized by the fuzzy membership function $F_j^{i_j}(X_j)$, with $i_j \in \{1, 2, \ldots, N_j\}$ being the ordinal number of membership functions of X_j. Define a set $\mathbf{\Omega}$ which collects all possible fuzzy rules

$$\mathbf{\Omega} = \{\text{Rule}_{i_1,i_2,\ldots,i_m} \mid i_1 = 1, 2, \ldots, N_1;\ i_2 = 1, 2, \ldots, N_2; \ldots, i_m = 1, 2, \ldots, N_m\}. \qquad \text{(2-9)}$$

The output of the FS can be expressed as [19]:

$$y = \frac{\displaystyle\sum_{\text{Rule}_{i_1,i_2,\ldots,i_m} \in \mathbf{\Omega}_{sub}} \alpha_{i_1,i_2,\ldots,i_m} \left[\prod_{j=1}^{m} \mu_{F_j^{i_j}}(X_j)\right]}{\displaystyle\sum_{\text{Rule}_{i_1,i_2,\ldots,i_m} \in \mathbf{\Omega}_{sub}} \left[\prod_{j=1}^{m} \mu_{F_j^{i_j}}(X_j)\right]} \qquad \text{(2-10)}$$

where $\mathbf{\Omega}_{sub} \subseteq \mathbf{\Omega}$ is the rule base. From (2-10), the output of the FS can be represented as a linear combination of fuzzy basis functions defined as

$$\xi_{i_1,i_2,\cdots,i_m} = \frac{\displaystyle\prod_{j=1}^{m} \mu_{F_j^{i_j}}(X_j)}{\displaystyle\sum_{\text{Rule}_{i_1,i_2,\ldots,i_m} \in \mathbf{\Omega}_{sub}} \left[\prod_{j=1}^{m} \mu_{F_j^{i_j}}(X_j)\right]},\ i_j \in \{1, 2, \ldots, N_j\}, j=1, 2, \ldots, m. \qquad \text{(2-11)}$$

That is, (2-10) can be rewritten as

$$y = \boldsymbol{\alpha}^T \boldsymbol{\xi} \tag{2-12}$$

where $\boldsymbol{\alpha} \in R^{n\times 1}$ collects singleton consequents $\alpha_{i_1,i_2,\ldots,i_m}$ of all rules in $\boldsymbol{\Omega}_{sub}$, $\boldsymbol{\xi} \in R^{n\times 1}$ collects $\boldsymbol{\xi}_{i_1,i_2,\ldots,i_m}$ described in (2-11), and n is the number of the existing fuzzy rules. In this chapter, a Gaussian membership function is defined as

$$\mu_{F_j^{i_j}}(X_j, c_j^{i_j}, \sigma_j^{i_j}) = \exp\left\{ -\frac{[X_j - c_j^{i_j}]^2}{{\sigma_j^{i_j}}^2} \right\} \tag{2-13}$$

where $c_j^{i_j}$ and $\sigma_j^{i_j}$ are the mean and standard deviation of the Gaussian function, respectively.

2.2.2 Structure Learning Algorithm

The developed self-structuring algorithm consists of two parts: growing and pruning of fuzzy rules. Effective membership functions in the input spaces can be generated and ineffective fuzzy rules can be pruned automatically by the self-structuring algorithm, and thus a concise rule base can be obtained. In order to construct the fuzzy rule base, every input space $S(X_j)$ is partitioned into several overlapping clusters to construct the fuzzy sets of X_j. It can happen that for some incoming X_j, the degree of belongings to all its fuzzy sets are quite small, i.e., $F_j^{i_j}(X_j)$, $i_j = 1,2,\cdots,N_j$ are quite small, as depicted in Fig. 2-1(a). This means that the input space $S(X_j)$ is not properly clustered. Hence, the fundamental concept of the growing of fuzz rules is developed to adjust the inappropriate clustering. Initially, create one initial fuzzy rule with the given initial state as

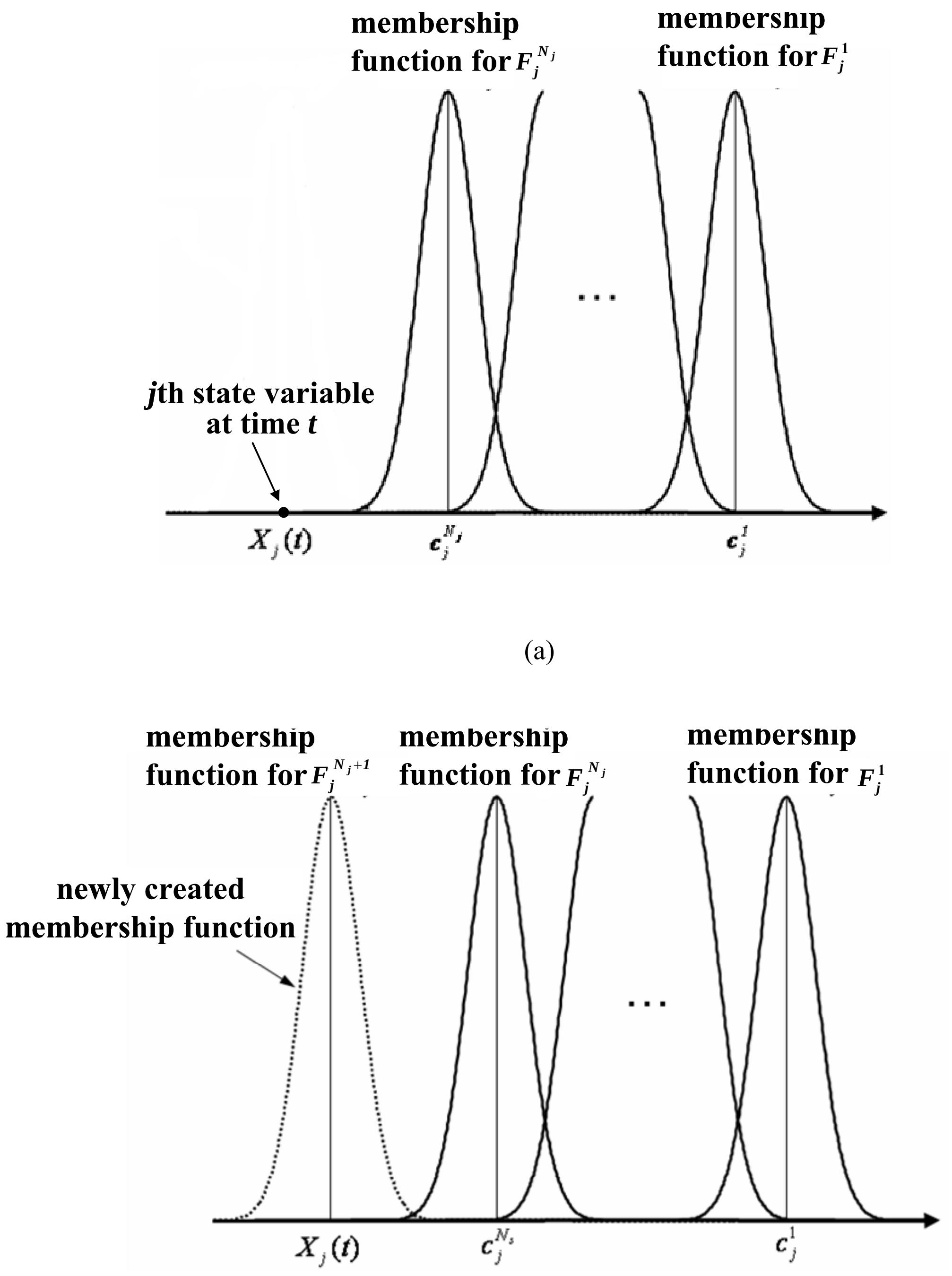
Rule$_{1,1,\cdots,1}$: IF X_1 is F_1^1 and X_2 is F_2^1 and… and X_m is F_m^1 THEN y is $\alpha_{1,1,\ldots,1}$ (2-14)

where the membership functions for F_j^1, j=1, 2, …, m, are defined with the initial input $X_j(0)$ as

$$\mu_{F_j^1}(X_j) = \exp\left\{ -\frac{[X_j - X_j(0)]^2}{{\sigma_j^1}^2} \right\}. \tag{2-15}$$

The SFS will start operating from this single rule. Define the growing criterion as

$$\mu_j^{\max} < \Theta_g, j=1, 2, \ldots, m \tag{2-16}$$


membership function for $F_j^{N_j}$
membership function for F_j^1
...
jth state variable at time t
$X_j(t)$
$c_j^{N_j}$
c_j^1
membership function for $F_j^{N_j+1}$
membership function for $F_j^{N_j}$
membership function for F_j^1
newly created membership function
...
$X_j(t)$
$c_j^{N_j}$
c_j^1

(a)

(b)

Fig. 2-1 (a) Improper fuzzy clustering of input variable X_j; (b) Newly created membership function

where $\mu_j^{\max} = \max_{i_j=1,2,\ldots,N_j} \mu_{F_j^{i_j}}(X_j)$ is the maximum membership function degree of X_j and $\Theta_g \in (0,1)$ is a given threshold. If at some time t_g, the growing criterion (2-16) is satisfied for a new incoming datum, $X_j(t_g)$, $1 \le j \le m$, a new membership function is created, whose initial mean and standard deviation are

$$c_j^{N_j+1} = X_j(t_g) \tag{2-17}$$

$$\sigma_j^{N_j+1} = q \tag{2-18}$$

where $q>0$ can be arbitrarily chosen, and it will be tuned by the adaptive law introduced in later section. The created membership function is shown in Fig. 2-1(b). For the case that one new membership function is created at some time, $N_1 \times \ldots \times N_{j-1} \times N_{j+1} \times \ldots \times N_m$ new fuzzy rules will be generated according to the new membership function as:

$\text{Rule}_{1,\ldots,N_j+1,\ldots,1}$: IF X_1 is $F_1^1 \ldots X_j$ is $F_j^{N_j+1} \ldots$ and X_m is F_m^1, THEN y is $\alpha_{1,\ldots,N_j+1,\ldots,1}$

$\text{Rule}_{2,\ldots,N_j+1,\ldots,1}$: IF X_1 is $F_1^2 \ldots X_j$ is $F_j^{N_j+1} \ldots$ and X_m is F_m^1, THEN y is $\alpha_{2,\ldots,N_j+1,\ldots,1}$

$\text{Rule}_{N_1,\ldots,N_j+1,\ldots,N_m}$: IF X_1 is $F_1^{N_1} \ldots X_j$ is $F_j^{N_j+1} \ldots$ and X_m is $F_m^{N_m}$, THEN y is $\alpha_{N_1,\ldots,N_j+1,\ldots,N_m}$

(2-19)

For example, consider a fuzzy system (m=2, N_1=1, and N_2=2) with the rule base:

$\text{Rule}_{1,1}$: IF X_1 is F_1^1 and X_2 is F_2^1 THEN y is $\alpha_{1,1}$

$\text{Rule}_{1,2}$: IF X_1 is F_1^1 and X_2 is F_2^2 THEN y is $\alpha_{1,2}$

Assume that the growing criterion for X_1 is satisfied at time t. Then, a new membership function

$$\mu_{F_1^2} = \exp\left\{-\frac{[X_1 - X_1(t)]^2}{(\sigma_1^2)^2}\right\} \tag{2-20}$$

is created, and two rules are grown according to the new membership function as

$\text{Rule}_{2,1}$: IF X_1 is F_1^2 and X_2 is F_2^1 THEN y is $\alpha_{2,1}$

$\text{Rule}_{2,2}$: IF $X1$ is F_1^2 and $X2$ is F_2^2 THEN y is $\alpha_{2,2}$ (2-21)

A self-structuring FS with only rule generation algorithm may suffer from the computational load or learning failure caused by an overly large rule base which includes both effective and redundant fuzzy rules. In the following, the strategy to prune redundant rules is developed to solve this problem. Recall that there are n existing fuzzy rules, and then express (2-12) as

$$y = \boldsymbol{\alpha}^T \boldsymbol{\xi} = [\alpha_k \ \boldsymbol{\alpha}_{rm}] \begin{bmatrix} \xi_k \\ \boldsymbol{\xi}_{rm} \end{bmatrix} \tag{2-22}$$

where $\alpha_k \in R$ and $\boldsymbol{\alpha}_{rm} \in R^{(n-1)\times 1}$ represent the singleton consequent and the fuzzy basis function of the kth fuzzy rule, respectively; $\boldsymbol{\alpha}_{rm} \in R^{(n-1)\times 1}$ and $\boldsymbol{\xi}_{rm} \in R^{(n-1)\times 1}$ represent the collections of the singleton consequents and the fuzzy basis functions of the rest of fuzzy rules, respectively. Thus, the contribution made by kth rule on the output y can be defined as follows:

$$C_k = \frac{|y_k|}{\sum_{k=1}^{n} |y_k|}, k=1, 2, \ldots, n \tag{2-23}$$

where $y_k = \alpha_k \xi_k$. Now, we are ready to introduce the significance index which can help us to decide whether or not to prune a fuzzy rule. Significance index is a measurement of the importance of every fuzzy rule. S_k, which represents the significance index of the kth fuzzy rule, is updated as follows:

$$S_k = \begin{cases} S_k^{rc}\tau, & \text{if } C_k < \beta \\ S_k^{rc}, & \text{if } C_k \geq \beta \end{cases}, k=1, 2, \ldots, n \tag{2-24}$$

where S_k^{rc} is the most recent S_k, $\tau \in (0,1)$ is a decay constant, and $\beta \in (0,1)$ is a given constant. All S_k, k=1, 2, …, n, are initialized from ones. According to (2-18), if the contribution C_k is equal or larger than β, S_k keeps invariant; if C_k is smaller than β, S_k will be attenuated. An invariant significance implies that the associated rule is still important and should be remained; a decaying significance index implies that the associated rule is becoming less and less important and thus should be pruned. The selection of τ will affect the rate of pruning the fuzzy rules. The smaller the τ is (or the larger the β is), the faster the significance index S_k decays, and thus the faster the ineffective fuzzy rules will be pruned. The pruning criterion of the kth fuzzy rule is defined as follows based on this knowledge

$$S_k < \Theta_p, k=1, 2, \ldots, n \tag{2-25}$$

where $\Theta_p \in (0,1)$ is a selected threshold. If the pruning criterion is satisfied for S_k, the associated kth rule is pruned.

Remark 2-1: It is a difficult task to determine the initial values of the singleton consequents of the newly generated fuzzy rules. Because an SFS is in general equipped with a parameter learning algorithm to automatically tune the parameters of the fuzzy rules, the initial values of the singleton consequents can simply set as zeros. However, from (2-10), we can see that this will cause abrupt variation of the fuzzy output y and may deteriorate the performance of the SFS for a short period. This phenomenon can be observed in Fig. 2-5(b). To fix this drawback, we maintain the approximation property of the SFS at the instant that new rules are generated. Assume that at some time t_g, an SFS has n fuzzy rules and the last h rules are just newly generated. Define y_p as the "pseudo fuzzy output" of the original n-h rules if h new rules were not generated at t_g. The initial consequents of those new rules are chosen so that $y(t_g) = y_p$.

Thus, we have

$$y(t_g) = \alpha_{new} \sum_{k=n-h+1}^{n} \xi_k + \sum_{k=1}^{n-h} \alpha_k \xi_k = y_p \tag{2-26}$$

where $\alpha_{n-h+1} = \alpha_{n-h+2} = \cdots = \alpha_n = \alpha_{new}$. From (2-26), we can easily obtain

$$\alpha_{new} = \frac{y_p - \sum_{k=1}^{n-h} \alpha_k \xi_k}{\sum_{k=n-h+1}^{n} \xi_k} \tag{2-27}$$

In this way, not only the bad effect caused by the abrupt variation can be mitigated, but also the future performance of the SFS can be improved by the *h* new rules.

Remark 2-2: While controlling, a membership function is possible to be pruned if all fuzzy rules associated with this membership function are pruned sequentially.

Remark 2-3: In the implementations of practical systems, if computational burden is the issue having highest priority, the threshold Θ_p can be chosen large enough so that more fuzzy rules are pruned. Hence, the computational burden will be substantially reduced at the expense of less favorable system performance.

Fig. 2-2 shows the flowchart to summarize the self-structuring algorithm for the SFS. The growing and pruning effects during the control period will be illustrated in later sections with excellent result.

1896

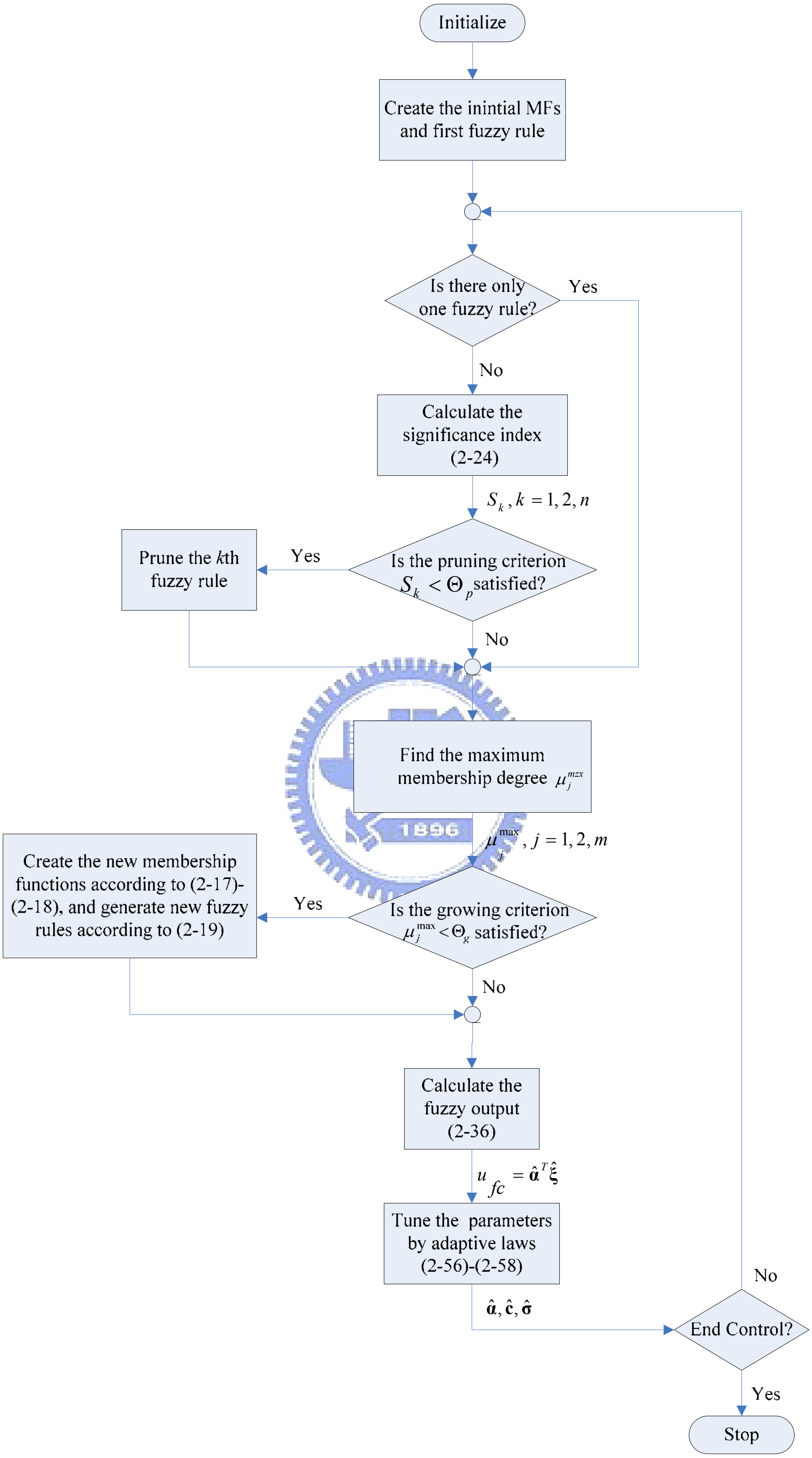

Initialize
Create the inintial MFs and first fuzzy rule
Is there only one fuzzy rule?
Yes
No
Calculate the significance index (2-24)
$S_k, k = 1, 2, n$
Is the pruning criterion $S_k < \Theta_p$ satisfied?
Yes
Prune the *k*th fuzzy rule
No
Find the maximum membership degree μ_j^{mzx}
$\mu_j^{\max}, j = 1, 2, m$
Is the growing criterion $\mu_j^{\max} < \Theta_g$ satisfied?
Yes
Create the new membership functions according to (2-17)-(2-18), and generate new fuzzy rules according to (2-19)
No
Calculate the fuzzy output (2-36)
$u_{fc} = \hat{\boldsymbol{\alpha}}^T \hat{\boldsymbol{\xi}}$
Tune the parameters by adaptive laws (2-56)-(2-58)
$\hat{\boldsymbol{\alpha}}, \hat{\mathbf{c}}, \hat{\boldsymbol{\sigma}}$
End Control?
No
Yes
Stop

Fig. 2-2 The flowchart of the self-structuring algorithm for the SFS

2.3 Design of RASFC

Now, we are ready for developing a robust adaptive self-structuring fuzzy controller (RASFC) for the unknown nonaffine nonlinear systems. In the RASFC, an SFS is used to estimate the system uncertainty $\Delta(\mathbf{x},u)$ in (2-2). The control law u in the RASFC system is designed as

$$u = \frac{1}{z}\left(u_{rac} - u_{fc}\right) \tag{2-28}$$

where u_{rac} is the robust adaptive controller to achieve a L_2 tracking performance with a desired attenuation level and u_{fc} is the self-structuring fuzzy controller to approximate unknown system dynamics $\Delta(\mathbf{x},u)$. Substituting (2-28) into (2-2) and using (2-4) yield

$$\begin{aligned} e^{(n)} &= x_c{}^{(n)} - \left[u_{rac} - u_{fc} + \Delta(\mathbf{x},u) + d\right] \\ &= x_c{}^{(n)} - u_{lc} - \left\{\left[\Delta(\mathbf{x},u) - u_{fc}\right] + (u_{rac} - u_{lc}) + d\right\} \\ &= -\mathbf{k}^T e - \left\{\left[\Delta(\mathbf{x},u) - u_{fc}\right] + (u_{rac} - u_{lc}) + d\right\} \end{aligned} \tag{2-29}$$

or

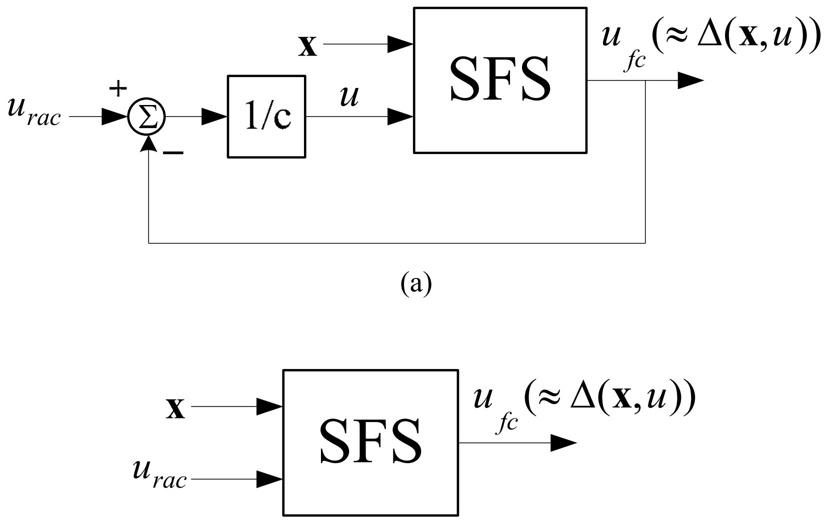
$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} - \mathbf{b}\left[\Delta(\mathbf{x},u) - u_{fc} + (u_{rac} - u_{lc}) + d\right] \tag{2-30}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ -k_n & -k_{n-1} & \cdots & \cdots & -k_1 \end{bmatrix} \text{ and } \mathbf{b} = [0\,0\ldots 1]^T$$

2.3.1 Fuzzy Approximation

The unknown nonlinear function $\Delta(\mathbf{x},u)$ is approximated by an SFS with inputs $\mathbf{x}$ and u. In this way, the output of the SFS u_{fc} should be directly fed back to produce u, which is one of the input of the SFS. This kind of fuzzy system is called a recurrent fuzzy system, as depicted in Fig. 2-3(a). However, a recurrent fuzzy system will lead to a fixed-point problem which must be solved at every time instant and thus imposes computational burden [51, 54-55]. Thus, the following Lemma 2-1 is stated to avoid this problem [51, 54-55].


x
+
u_{rac}
Σ
−
1/c
u
SFS
$u_{fc}(\approx \Delta(\mathbf{x},u))$
x
u_{rac}
SFS
$u_{fc}(\approx \Delta(\mathbf{x},u))$

(a)

(b)

Fig. 2-3 (a) The recurrent fuzzy system; (b) The static fuzzy system

Lemma 2-1: Let the constant *c* satisfies the condition

$$z > \frac{1}{2}\left(\frac{\partial f}{\partial u}\right) \tag{2-31}$$

Then, there exist a unique u_{fc}^* which is a function of $\mathbf{x}$ and u_{rac} so that $u_{fc}^*(\mathbf{x},u_{rac})$ satisfies

$$\psi(\mathbf{x},u_{rac},u_{fc}^*) \overset{\Delta}{=} \Delta(\mathbf{x},u_{rac},u_{fc}^*) - u_{fc}^*(\mathbf{x},u_{rac}) = 0 \tag{2-32}$$

for all $(\mathbf{x},u_{rac}) \in \mathbf{\Omega}_{\mathbf{x}} \times R$.

The Proof of Lemma 1 can be found in [51].

According to Lemma 2-1, the feedback path in Fig. 2-3(a) can be removed. Consequently, a static FS in Fig. 2-3(b) can be used to approximate $\Delta(\mathbf{x},u)$, and thus we do not need to solve the fixed-point problem at every time instant. For the nonaffine systems with the property $\frac{\partial f(\mathbf{x},u)}{\partial u} < 0$, Lemma 2-1 can be satisfied as well by simply modifying (2-31) as $z < \frac{1}{2}\left(\frac{\partial f}{\partial u}\right)$.

Define the vectors $\mathbf{c}$ and $\boldsymbol{\sigma}$ as

$$\mathbf{c} = [\mathbf{c}_1\ \mathbf{c}_2\ \cdots \mathbf{c}_m]^T \tag{2-33}$$

$$\boldsymbol{\sigma} = [\boldsymbol{\sigma}_1\ \boldsymbol{\sigma}_2\ \cdots \boldsymbol{\sigma}_m]^T \tag{2-34}$$

where $\mathbf{c}_j = [c_j^1 \cdots c_j^{N_j}]$ and $\boldsymbol{\sigma}_j = [\sigma_j^1 \cdots \sigma_j^{N_j}]$ collect the means and standard deviations of the

Gaussian membership functions of X_j, j=1, 2, …, m, respectively. Rewrite (2-12) in the vector form as

$$y=\boldsymbol{\alpha}^T\boldsymbol{\xi}(\mathbf{X},\mathbf{c},\boldsymbol{\sigma})=[\alpha_1\ \alpha_2\ \ldots\ \alpha_n]\begin{bmatrix}\xi_1\\ \xi_2\\ \vdots\\ \xi_n\end{bmatrix} \tag{2-35}$$

where $\mathbf{X}=[\mathbf{x}\ u_{rac}]^T$ is the input vector. The output of the SFS used to approximate $\Delta(\mathbf{x},u)$ is defined as

$$u_{fc}=\hat{\boldsymbol{\alpha}}^T\boldsymbol{\xi}(\mathbf{X},\hat{\mathbf{c}},\hat{\boldsymbol{\sigma}})=\hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\xi}} \tag{2-36}$$

where $\hat{\boldsymbol{\alpha}}$, $\hat{\mathbf{c}}$, and $\hat{\boldsymbol{\sigma}}$ are the estimation vectors of $\boldsymbol{\alpha}$, $\mathbf{c}$, and $\boldsymbol{\sigma}$, and $\hat{\boldsymbol{\xi}}=\boldsymbol{\xi}(\mathbf{X},\hat{\mathbf{c}},\hat{\boldsymbol{\sigma}})$. Define the optimal vectors $\boldsymbol{\alpha}^*$, $\mathbf{c}^*$, and $\boldsymbol{\sigma}^*$ as [3]:

$$(\boldsymbol{\alpha}^*,\mathbf{c}^*,\boldsymbol{\sigma}^*)=\underset{\hat{\boldsymbol{\alpha}}\in\boldsymbol{\Omega}_{\boldsymbol{\alpha}},\hat{\mathbf{c}}\in\boldsymbol{\Omega}_{\mathbf{c}},\hat{\boldsymbol{\sigma}}\in\boldsymbol{\Omega}_{\boldsymbol{\sigma}}}{\arg\min}\left[\sup_{\mathbf{X}\in\boldsymbol{\Omega}_{\mathbf{x}}\times R}\left|u_{fc}(\mathbf{X})-u_{fc}(\mathbf{X},\hat{\boldsymbol{\alpha}},\hat{\mathbf{c}},\hat{\boldsymbol{\sigma}})\right|\right] \tag{2-37}$$

where

$$\boldsymbol{\Omega}_{\boldsymbol{\alpha}}=\left\{\hat{\boldsymbol{\alpha}}:\|\hat{\boldsymbol{\alpha}}\|\le M_{\boldsymbol{\alpha}}\right\} \tag{2-38}$$

$$\boldsymbol{\Omega}_{\mathbf{c}}=\left\{\hat{\mathbf{c}}:\|\hat{\mathbf{c}}\|\le M_{\mathbf{c}}\right\} \tag{2-39}$$

$$\boldsymbol{\Omega}_{\boldsymbol{\sigma}}=\left\{\hat{\boldsymbol{\sigma}}:\|\hat{\boldsymbol{\sigma}}\|\le M_{\boldsymbol{\sigma}}\right\} \tag{2-40}$$

and $M_{\boldsymbol{\alpha}}$, $M_{\mathbf{c}}$, and $M_{\boldsymbol{\sigma}}$ are positive constants specified by designers. The unknown nonlinear function $\Delta(\mathbf{x},u)$ can be described as

$$\Delta=\boldsymbol{\alpha}^{*T}\boldsymbol{\xi}(\mathbf{X},\mathbf{c}^*,\boldsymbol{\sigma}^*)+\omega=\boldsymbol{\alpha}^{*T}\boldsymbol{\xi}^*+\omega \tag{2-41}$$

where $\boldsymbol{\xi}^*=\boldsymbol{\xi}(\mathbf{X},\mathbf{c}^*,\boldsymbol{\sigma}^*)$ and ω denotes the approximation error bounded by $|\omega|\le\bar{\omega}$, in which $\bar{\omega}$ is a finite positive constant. Then, modeling error $\tilde{u}$ can be expressed as

$$\tilde{u}=\Delta-u_{fc}=\tilde{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\xi}}+\hat{\boldsymbol{\alpha}}^T\tilde{\boldsymbol{\xi}}+\tilde{\boldsymbol{\alpha}}^T\tilde{\boldsymbol{\xi}}+\omega \tag{2-42}$$

where $\tilde{\boldsymbol{\alpha}}=\boldsymbol{\alpha}^*-\hat{\boldsymbol{\alpha}}$ and $\tilde{\boldsymbol{\xi}}=\boldsymbol{\xi}^*-\hat{\boldsymbol{\xi}}$. In the following, some preliminaries will be made for adaptive online-tuning of the parameters of fuzzy rules, and thus favorable approximation performance can be achieved in the presence of unexpected disturbances. To achieve this goal, the Taylor linearization technique is employed to transform the nonlinear fuzzy basis function into partially linear form as follows [25, 56]:

$$\tilde{\boldsymbol{\xi}} = \begin{bmatrix} \tilde{\xi}_1 \\ \tilde{\xi}_2 \\ \vdots \\ \tilde{\xi}_n \end{bmatrix} = \left.\begin{bmatrix} \dfrac{\xi_1}{\partial \mathbf{c}} \\ \dfrac{\xi_2}{\partial \mathbf{c}} \\ \vdots \\ \dfrac{\xi_n}{\partial \mathbf{c}} \end{bmatrix}\right|_{\mathbf{c}=\hat{\mathbf{c}}} (\mathbf{c}^* - \hat{\mathbf{c}}) + \left.\begin{bmatrix} \dfrac{\xi_1}{\partial \boldsymbol{\sigma}} \\ \dfrac{\xi_2}{\partial \boldsymbol{\sigma}} \\ \vdots \\ \dfrac{\xi_n}{\partial \boldsymbol{\sigma}} \end{bmatrix}\right|_{\boldsymbol{\sigma}=\hat{\boldsymbol{\sigma}}} (\boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}}) + \boldsymbol{o} \tag{2-43}$$

or

$$\tilde{\boldsymbol{\xi}} = \boldsymbol{\xi}_{\mathbf{c}}^{\boldsymbol{T}} \tilde{\mathbf{c}} + \boldsymbol{\xi}_{\boldsymbol{\sigma}}^{\boldsymbol{T}} \tilde{\boldsymbol{\sigma}} + \boldsymbol{o} \tag{2-44}$$

where $\boldsymbol{o}$ represents the higher order term, $\tilde{\mathbf{c}} = \mathbf{c}^* - \hat{\mathbf{c}}$, $\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}}$, and

$$\boldsymbol{\xi}_{\mathbf{c}} = \left.\begin{bmatrix} \dfrac{\partial \xi_1}{\partial \mathbf{c}} & \dfrac{\partial \xi_2}{\partial \mathbf{c}} & \cdots & \dfrac{\partial \xi_n}{\partial \mathbf{c}} \end{bmatrix}\right|_{\mathbf{c}=\hat{\mathbf{c}}} \tag{2-45}$$

$$\boldsymbol{\xi}_{\boldsymbol{\sigma}} = \left.\begin{bmatrix} \dfrac{\partial \xi_1}{\partial \boldsymbol{\sigma}} & \dfrac{\partial \xi_2}{\partial \boldsymbol{\sigma}} & \cdots & \dfrac{\partial \xi_n}{\partial \boldsymbol{\sigma}} \end{bmatrix}\right|_{\boldsymbol{\sigma}=\hat{\boldsymbol{\sigma}}} \tag{2-46}$$

Substituting (2-44) into (2-42) yields

$$\begin{aligned} \tilde{u} &= \tilde{\boldsymbol{\alpha}}^T \hat{\boldsymbol{\xi}} + \hat{\boldsymbol{\alpha}}^T \boldsymbol{\xi}_{\mathbf{c}}^T \tilde{\mathbf{c}} + \hat{\boldsymbol{\alpha}}^T \boldsymbol{\xi}_{\boldsymbol{\sigma}}^T \tilde{\boldsymbol{\sigma}} + \varepsilon \\ &= \tilde{\boldsymbol{\alpha}}^T \hat{\boldsymbol{\xi}} + \tilde{\mathbf{c}}^T \boldsymbol{\xi}_{\mathbf{c}} \hat{\boldsymbol{\alpha}} + \tilde{\boldsymbol{\sigma}}^T \boldsymbol{\xi}_{\boldsymbol{\sigma}} \hat{\boldsymbol{\alpha}} + \varepsilon \end{aligned} \tag{2-47}$$

where $\hat{\boldsymbol{\alpha}}^T \boldsymbol{\xi}_{\mathbf{c}}^T \tilde{\mathbf{c}} = \tilde{\mathbf{c}}^T \boldsymbol{\xi}_{\mathbf{c}} \hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\alpha}}^T \boldsymbol{\xi}_{\boldsymbol{\sigma}}^T \tilde{\boldsymbol{\sigma}} = \tilde{\boldsymbol{\sigma}}^T \boldsymbol{\xi}_{\boldsymbol{\sigma}} \hat{\boldsymbol{\alpha}}$ since they are scalars, and $\varepsilon = \tilde{\boldsymbol{\alpha}}^T \tilde{\boldsymbol{\xi}} + \hat{\boldsymbol{\alpha}}^T \boldsymbol{o} + \omega$ is the lumped uncertainty. The higher order term o satisfies

$$\begin{aligned} \|\boldsymbol{o}\| &= \tilde{\boldsymbol{\xi}} - \boldsymbol{\xi}_{\mathbf{c}}^{\boldsymbol{T}} \tilde{\mathbf{c}} + \boldsymbol{\xi}_{\boldsymbol{\sigma}}^{\boldsymbol{T}} \tilde{\boldsymbol{\sigma}} \\ &\le \|\tilde{\boldsymbol{\xi}}\| + \|\boldsymbol{\xi}_{\mathbf{c}}^T\| \|\tilde{\mathbf{c}}\| + \|\boldsymbol{\xi}_{\boldsymbol{\sigma}}^T\| \|\tilde{\boldsymbol{\sigma}}\| \\ &\le b_0 + b_1 \|\tilde{\mathbf{c}}\| + b_2 \|\tilde{\boldsymbol{\sigma}}\| \end{aligned} \tag{2-48}$$

where b_0, b_1, and b_2 are bounded positive constants satisfying $\|\tilde{\boldsymbol{\xi}}\| \le b_0$, $\|\boldsymbol{\xi}_{\mathbf{c}}^T\| \le b_1$, and $\|\boldsymbol{\xi}_{\boldsymbol{\sigma}}^T\| \le b_2$. It is reasonable that b_0, b_1, and b_2 exist because Gaussian function and its derivative are always bounded by constants. Moreover, $\tilde{\boldsymbol{\alpha}}$, $\tilde{\mathbf{c}}$, and $\tilde{\boldsymbol{\sigma}}$ satisfy

$$\|\tilde{\boldsymbol{\alpha}}\| = \|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}\| \le \|\boldsymbol{\alpha}^*\| + \|\hat{\boldsymbol{\alpha}}\| \le M_{\boldsymbol{\alpha}} + \|\hat{\boldsymbol{\alpha}}\| \tag{2-49}$$

$$\|\tilde{\mathbf{c}}\| = \|\mathbf{c}^* - \hat{\mathbf{c}}\| \le \|\mathbf{c}^*\| + \|\hat{\mathbf{c}}\| \le M_{\mathbf{c}} + \|\hat{\mathbf{c}}\| \tag{2-50}$$

$$\|\tilde{\boldsymbol{\sigma}}\| = \|\boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}}\| \le \|\boldsymbol{\sigma}^*\| + \|\hat{\boldsymbol{\sigma}}\| \le M_{\boldsymbol{\sigma}} + \|\hat{\boldsymbol{\sigma}}\| \tag{2-51}$$

Thus, the lumped uncertainty ε satisfies

$$|\varepsilon| = \left|\tilde{\boldsymbol{\alpha}}^T (\boldsymbol{\xi}_{\mathbf{c}}^{\boldsymbol{T}} \tilde{\mathbf{c}} + \boldsymbol{\xi}_{\boldsymbol{\sigma}}^{\boldsymbol{T}} \tilde{\boldsymbol{\sigma}} + \boldsymbol{o}) + \hat{\boldsymbol{\alpha}}^T \boldsymbol{o} + \omega\right|$$

$$
\begin{aligned}
&= \left| \tilde{\boldsymbol{\alpha}}^T \boldsymbol{\xi}_{\mathbf{c}}^{\boldsymbol{T}} \tilde{\mathbf{c}} + \tilde{\boldsymbol{\alpha}}^T \boldsymbol{\xi}_{\boldsymbol{\sigma}}^{\boldsymbol{T}} \tilde{\boldsymbol{\sigma}} + \boldsymbol{\alpha}^{*\boldsymbol{T}} \boldsymbol{o} + \omega \right| \\
&\le b_1 (M_{\boldsymbol{\alpha}} + \|\hat{\boldsymbol{\alpha}}\|)(M_{\mathbf{c}} + \|\hat{\mathbf{c}}\|) + b_2 (M_{\boldsymbol{\alpha}} + \|\hat{\boldsymbol{\alpha}}\|)(M_{\boldsymbol{\sigma}} + \|\hat{\boldsymbol{\sigma}}\|) \\
&\quad + M_{\boldsymbol{\alpha}}[b_0 + b_1 (M_{\mathbf{c}} + \|\hat{\mathbf{c}}\|) + b_2 (M_{\boldsymbol{\sigma}} + \|\hat{\boldsymbol{\sigma}}\|)] + \overline{\omega} \\
&= [\Lambda_1 \ \Lambda_2 \ \Lambda_3 \ \Lambda_4 \ \Lambda_5 \ \Lambda_6][1 \ \|\hat{\boldsymbol{\alpha}}\| \ \|\hat{\mathbf{c}}\| \ \|\hat{\boldsymbol{\sigma}}\| \ \|\hat{\boldsymbol{\alpha}}\|\|\hat{\mathbf{c}}\| \ \|\hat{\boldsymbol{\alpha}}\|\|\hat{\boldsymbol{\sigma}}\|]^T \\
&= \boldsymbol{\Lambda}^T \boldsymbol{\Gamma} \qquad (2\text{-}52)
\end{aligned}
$$

where $\boldsymbol{\Lambda} = [\Lambda_1 \ \Lambda_2 \ \Lambda_3 \ \Lambda_4 \ \Lambda_5 \ \Lambda_6]^T$, $\Lambda_1 = (b_0 + 2b_1 M_{\mathbf{c}} + 2b_2 M_{\boldsymbol{\sigma}})M_{\boldsymbol{\alpha}} + \overline{\omega}$, $\Lambda_2 = b_1 M_{\mathbf{c}} + b_2 M_{\boldsymbol{\sigma}}$, $\Lambda_3 = 2b_1 M_{\boldsymbol{\alpha}}$, $\Lambda_4 = 2b_2 M_{\boldsymbol{\alpha}}$, $\Lambda_5 = b_1$, $\Lambda_6 = b_2$, and $\boldsymbol{\Gamma} = [1 \ \|\hat{\boldsymbol{\alpha}}\| \ \|\hat{\mathbf{c}}\| \ \|\hat{\boldsymbol{\sigma}}\| \ \|\hat{\boldsymbol{\alpha}}\|\|\hat{\mathbf{c}}\| \ \|\hat{\boldsymbol{\alpha}}\|\|\hat{\boldsymbol{\sigma}}\|]^T$. Since $\boldsymbol{\Lambda}$ is a bounded vector, if $\boldsymbol{\Gamma}$ is guaranteed to be bounded, the lumped uncertainty term ε is thus bounded. We can guarantee the boundness of $\boldsymbol{\Gamma}$ by Lemma 2-2 given in the next subsection.

2.3.2 Parameter Learning Algorithm

Substituting (2-47) into (2-30) yields

$$
\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} - \mathbf{b}[\tilde{\boldsymbol{\alpha}}^T \hat{\boldsymbol{\xi}} + \tilde{\mathbf{c}}^T \boldsymbol{\xi}_{\mathbf{c}} \hat{\boldsymbol{\alpha}} + \tilde{\boldsymbol{\sigma}}^T \boldsymbol{\xi}_{\boldsymbol{\sigma}} \hat{\boldsymbol{\alpha}} + \varepsilon + d + (u_{rac} - u_{lc})]. \qquad (2\text{-}53)
$$

Lemma 2-2 [3]: Suppose that the adaptive laws are chosen as (2-56)-(2-58), where $\mathbf{Pr}(\cdot)$ is the projection operator, and the symmetric positive $\mathbf{P}$ satisfies the following Riccati-like equation

$$
\mathbf{A}^T \mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{Q} + \mathbf{P}\mathbf{b}(\frac{1}{\rho^2} - \frac{1}{\delta})\mathbf{b}^{\mathbf{T}} \mathbf{P} = 0 \qquad (2\text{-}54)
$$

where $\mathbf{Q}$ is a positive definite symmetric matrix and ρ is an attenuation level which satisfies $\frac{1}{\rho^2} - \frac{1}{\delta} \le 0$. If $\hat{\boldsymbol{\alpha}}(0) \in \boldsymbol{\Omega}_{\boldsymbol{\alpha}}$, $\hat{\mathbf{c}}(0) \in \boldsymbol{\Omega}_{\mathbf{c}}$, and $\hat{\boldsymbol{\sigma}}(0) \in \boldsymbol{\Omega}_{\boldsymbol{\sigma}}$, then $\hat{\boldsymbol{\alpha}}(t) \in \boldsymbol{\Omega}_{\boldsymbol{\alpha}}$, $\hat{\mathbf{c}}(t) \in \boldsymbol{\Omega}_{\mathbf{c}}$, and $\hat{\boldsymbol{\sigma}}(t) \in \boldsymbol{\Omega}_{\boldsymbol{\sigma}}$ for all $t \ge 0$ can be guaranteed.

According to Lemma 2-2, $\boldsymbol{\Gamma}$ in (2-52) is bounded, and hence the lumped uncertainty ε is bounded. The following theorem shows the properties of the developed control system.

Theorem 2-1: Suppose the assumption (2-3) holds. Consider a SISO nonaffine nonlinear system (2-1) with the control law (2-28), where the self-structuring fuzzy controller is given as

$$
u_{fc} = \hat{\boldsymbol{\alpha}}^T \boldsymbol{\xi}(\mathbf{X}, \hat{\mathbf{c}}, \hat{\boldsymbol{\sigma}}) \qquad (2\text{-}55)
$$

The adaptive laws are chosen as (2-56)-(2-58):

$$\dot{\hat{\boldsymbol{\alpha}}} = -\dot{\tilde{\boldsymbol{\alpha}}} = \begin{cases} -\eta_{\boldsymbol{\alpha}}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}\,, & \text{if } \|\hat{\boldsymbol{\alpha}}\| < M_{\boldsymbol{\alpha}} \text{ or } (\|\hat{\boldsymbol{\alpha}}\| = M_{\boldsymbol{\alpha}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\xi}} \ge 0) \\ \mathbf{Pr}(\eta_{\alpha}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}})\,, & \text{if } (\|\hat{\boldsymbol{\alpha}}\| = M_{\boldsymbol{\alpha}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\xi}} < 0) \end{cases} \tag{2-56}$$

where $\eta_{\boldsymbol{\alpha}}$ is the positive learning rate and $\mathbf{Pr}(\eta_{\alpha}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}) = -\eta_{\alpha}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}} + \eta_{\alpha}\mathbf{e}^T\mathbf{Pb}\dfrac{\hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\xi}}}{\|\hat{\boldsymbol{\alpha}}\|^2}\hat{\boldsymbol{\alpha}}$.

$$\dot{\hat{\mathbf{c}}} = -\dot{\tilde{\mathbf{c}}} = \begin{cases} -\eta_{c}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}_{\mathbf{c}}\hat{\boldsymbol{\alpha}}\,, & \text{if } \|\hat{\mathbf{c}}\| < M_{\mathbf{c}} \text{ or } (\|\hat{\mathbf{c}}\| = M_{\mathbf{c}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\mathbf{c}}^T\boldsymbol{\xi}_{\mathbf{c}}\hat{\boldsymbol{\alpha}} \ge 0) \\ \mathbf{Pr}(\eta_{c}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}_{\mathbf{c}}\hat{\boldsymbol{\alpha}})\,, & \text{if } (\|\hat{\mathbf{c}}\| = M_{\mathbf{c}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\mathbf{c}}^T\boldsymbol{\xi}_{\mathbf{c}}\hat{\boldsymbol{\alpha}} < 0) \end{cases} \tag{2-57}$$

where $\eta_{\mathbf{c}}$ is positive learning rate and $\mathbf{Pr}(\eta_{\mathbf{c}}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}_{\mathbf{c}}\hat{\boldsymbol{\alpha}}) = -\eta_{\mathbf{c}}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}_{\mathbf{c}}\hat{\boldsymbol{\alpha}} + \eta_{\mathbf{c}}\mathbf{e}^T\mathbf{Pb}\dfrac{\hat{\mathbf{c}}^T\boldsymbol{\xi}_{\mathbf{c}}\hat{\boldsymbol{\alpha}}}{\|\hat{\mathbf{c}}\|^2}\hat{\mathbf{c}}$.

$$\dot{\hat{\boldsymbol{\sigma}}} = -\dot{\tilde{\boldsymbol{\sigma}}} = \begin{cases} -\eta_{\boldsymbol{\sigma}}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}}\,, & \text{if } \|\hat{\boldsymbol{\sigma}}\| < M_{\boldsymbol{\sigma}} \text{ or } (\|\hat{\boldsymbol{\sigma}}\| = M_{\boldsymbol{\sigma}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\sigma}}^T\boldsymbol{\xi}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}} \ge 0) \\ \mathbf{Pr}(\eta_{\boldsymbol{\sigma}}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}})\,, & \text{if } (\|\hat{\boldsymbol{\sigma}}\| = M_{\boldsymbol{\sigma}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\sigma}}^T\boldsymbol{\xi}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}} < 0) \end{cases} \tag{2-58}$$

where $\eta_{\boldsymbol{\sigma}}$ is positive learning rate and $\mathbf{Pr}(\eta_{\boldsymbol{\sigma}}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}}) = -\eta_{\boldsymbol{\sigma}}\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}} + \eta_{\boldsymbol{\sigma}}\mathbf{e}^T\mathbf{Pb}\dfrac{\hat{\boldsymbol{\sigma}}^T\boldsymbol{\xi}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}}}{\|\boldsymbol{\sigma}\|^2}\hat{\boldsymbol{\sigma}}$

The robust adaptive controller is given as

$$u_{rac} = u_{lc} + \frac{1}{2\delta}\mathbf{b}^T\mathbf{Pe} \tag{2-59}$$

Note that since **A** is designed to be stable in (2-30) and **Q** in (2-54) is a positive definite symmetric matrix, therefore **P** must be a positive definite symmetric matrix. Then, the RASFC system can guarantee the global stability and robustness of the closed-loop system and achieve the following L_2 criterion [57-58]:

$$\frac{1}{2}\int_0^T \mathbf{e}^T\mathbf{Q}\mathbf{e}\,dt \le \frac{1}{2}\mathbf{e}(0)^T\mathbf{Pe}(0) + \frac{\tilde{\boldsymbol{\alpha}}^T(0)\tilde{\boldsymbol{\alpha}}(0)}{2\eta_\alpha} + \frac{\tilde{\mathbf{c}}^T(0)\tilde{\mathbf{c}}(0)}{2\eta_c} + \frac{\tilde{\boldsymbol{\sigma}}(0)^T\tilde{\boldsymbol{\sigma}}(0)}{2\eta_\sigma} + \frac{\rho^2}{2}\int_0^T(\varepsilon + d)^2\,dt \tag{2-60}$$

for $0 \le T < \infty$, where $\mathbf{e}(0)$, $\tilde{\boldsymbol{\alpha}}(0)$, $\tilde{\mathbf{c}}(0)$, and $\tilde{\boldsymbol{\sigma}}(0)$ are the initial values of $\mathbf{e}$, $\tilde{\boldsymbol{\alpha}}$, $\tilde{\mathbf{c}}$, and $\tilde{\boldsymbol{\sigma}}$, respectively.

Proof: Define the Lyapunov function candidate as

$$V = \frac{1}{2}\mathbf{e}^T\mathbf{Pe} + \frac{1}{2\eta_\alpha}\tilde{\boldsymbol{\alpha}}^T\tilde{\boldsymbol{\alpha}} + \frac{1}{2\eta_c}\tilde{\mathbf{c}}^T\tilde{\mathbf{c}} + \frac{1}{2\eta_\sigma}\tilde{\boldsymbol{\sigma}}^T\tilde{\boldsymbol{\sigma}}\,. \tag{2-61}$$

Differentiating (2-61) with respect to time and using (2-53) yield

$$\dot{V} = \frac{1}{2}\mathbf{e}^T\mathbf{P}\dot{\mathbf{e}} + \frac{1}{2}\dot{\mathbf{e}}^T\mathbf{Pe} + \frac{1}{\eta_\alpha}\tilde{\boldsymbol{\alpha}}^T\dot{\tilde{\boldsymbol{\alpha}}} + \frac{1}{\eta_c}\tilde{\mathbf{c}}^T\dot{\tilde{\mathbf{c}}} + \frac{1}{\eta_\sigma}\tilde{\boldsymbol{\sigma}}^T\dot{\tilde{\boldsymbol{\sigma}}}$$

$$= \frac{1}{2}\mathbf{e}^T(\mathbf{A}^T\mathbf{P}+\mathbf{PA})\mathbf{e} - \mathbf{e}^T\mathbf{P}\,\mathbf{b}[\tilde{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\xi}} + \tilde{\mathbf{c}}^T\boldsymbol{\xi}_{\mathbf{c}}\hat{\boldsymbol{\alpha}} + \tilde{\boldsymbol{\sigma}}^T\boldsymbol{\xi}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}} + \varepsilon + d + (u_{rac} - u_{lc})] + \frac{1}{\eta_\alpha}\tilde{\boldsymbol{\alpha}}^T\dot{\tilde{\boldsymbol{\alpha}}}$$

$$+ \frac{1}{\eta_c}\tilde{\mathbf{c}}^T\dot{\tilde{\mathbf{c}}} + \frac{1}{\eta_c}\tilde{\boldsymbol{\sigma}}^T\dot{\tilde{\boldsymbol{\sigma}}} \qquad (2\text{-}62)$$

Substituting (2-59) into (2-62), we obtain

$$\dot{V} = \frac{1}{2}\mathbf{e}^T(\mathbf{A}^T\mathbf{P}+\mathbf{PA}-\frac{1}{\delta}\mathbf{Pbb}^T\mathbf{P})\mathbf{e} - \mathbf{e}^T\mathbf{Pb}(\varepsilon+d) - G_{\boldsymbol{\alpha}} - G_{\mathbf{c}} - G_{\boldsymbol{\sigma}} \qquad (2\text{-}63)$$

where $G_{\boldsymbol{\alpha}} = \tilde{\boldsymbol{\alpha}}^T(\mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\xi}} - \frac{\dot{\tilde{\boldsymbol{\alpha}}}}{\eta_\alpha})$, $G_{\mathbf{c}} = \tilde{\mathbf{c}}^T(\mathbf{e}^T\mathbf{Pb}\boldsymbol{\xi}_c\hat{\boldsymbol{\alpha}} - \frac{\dot{\tilde{\mathbf{c}}}}{\eta_c})$, and $G_{\boldsymbol{\sigma}} = \tilde{\boldsymbol{\sigma}}^T(\mathbf{e}^T\mathbf{Pb}\boldsymbol{\xi}_\sigma\hat{\boldsymbol{\alpha}} - \frac{\dot{\tilde{\boldsymbol{\sigma}}}}{\eta_\sigma})$. By using (2-54), we can rewrite (2-63) as

$$\dot{V} = \frac{1}{2}\mathbf{e}^T(-\mathbf{Q}-\frac{1}{\rho^2}\mathbf{Pbb}^T\mathbf{P})\mathbf{e} - \mathbf{e}^T\mathbf{Pb}(\varepsilon+d) - G_{\boldsymbol{\alpha}} - G_{\mathbf{c}} - G_{\boldsymbol{\sigma}}$$

$$= -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} - \frac{1}{2}[\frac{1}{\rho}\mathbf{b}^T\mathbf{Pe} + \rho(\varepsilon+d)]^2 + \frac{1}{2}\rho^2(\varepsilon+d)^2 - G_{\boldsymbol{\alpha}} - G_{\mathbf{c}} - G_{\boldsymbol{\sigma}}\,. \qquad (2\text{-}64)$$

By using (2-56), we have $G_{\boldsymbol{\alpha}} = 0$ for $\left[\|\hat{\boldsymbol{\alpha}}\| \le M_{\boldsymbol{\alpha}} \text{ or } (\|\hat{\boldsymbol{\alpha}}\| = M_{\boldsymbol{\alpha}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\xi}} \ge 0)\right]$. For $\left[(\|\hat{\boldsymbol{\alpha}}\| = M_{\boldsymbol{\alpha}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\xi}} < 0)\right]$, we have

$$G_{\boldsymbol{\alpha}} = \eta_\alpha \mathbf{e}^T\mathbf{Pb}\frac{\tilde{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\alpha}}}{\|\hat{\boldsymbol{\alpha}}\|^2}\hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\xi}} \qquad (2\text{-}65)$$

Because $\boldsymbol{\alpha}^*$ belongs to the constraint set $\boldsymbol{\Omega}_\alpha$, we have $\|\hat{\boldsymbol{\alpha}}\| = M_{\boldsymbol{\alpha}} \ge \|\boldsymbol{\alpha}^*\|$. Using this fact, we obtain $\tilde{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\alpha}} = \frac{1}{2}(\|\boldsymbol{\alpha}^*\|^2 - \|\hat{\boldsymbol{\alpha}}\|^2 - \|\tilde{\boldsymbol{\alpha}}\|^2) \le 0$. Thus, (2-65) can be rewritten as

$$G_{\boldsymbol{\alpha}} = \frac{\eta_\alpha}{2}\mathbf{e}^T\mathbf{Pb}\frac{(\|\boldsymbol{\alpha}^*\|^2 - \|\hat{\boldsymbol{\alpha}}\|^2 - \|\tilde{\boldsymbol{\alpha}}\|^2)}{\|\hat{\boldsymbol{\alpha}}\|^2}\hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\xi}} \ge 0\,. \qquad (2\text{-}66)$$

Similarly, we have (2-67) and (2-68) by using (2-57) and (2-58) respectively.

$$G_{\mathbf{c}} = \begin{cases} 0 & \text{if } \|\hat{\mathbf{c}}\| < M_{\mathbf{c}} \text{ or } (\|\hat{\mathbf{c}}\| = M_{\mathbf{c}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\mathbf{c}}^T\boldsymbol{\xi}_{\mathbf{c}}\hat{\boldsymbol{\alpha}} \ge 0) \\ \frac{\eta_{\mathbf{c}}}{2}\mathbf{e}^T\mathbf{Pb}\frac{(\|\mathbf{c}^*\|^2 - \|\hat{\mathbf{c}}\|^2 - \|\tilde{\mathbf{c}}\|^2)}{\|\hat{\mathbf{c}}\|^2}\hat{\mathbf{c}}^T\boldsymbol{\xi}_{\mathbf{c}}\hat{\boldsymbol{\alpha}} \ge 0 & \text{if } (\|\hat{\mathbf{c}}\| = M_{\mathbf{c}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\mathbf{c}}^T\boldsymbol{\xi}_{\mathbf{c}}\hat{\boldsymbol{\alpha}} < 0) \end{cases}$$

(2-67)

$$G_{\boldsymbol{\sigma}} = \begin{cases} 0 & \text{if } \|\hat{\boldsymbol{\sigma}}\| < M_{\boldsymbol{\sigma}} \text{ or } (\|\hat{\boldsymbol{\sigma}}\| = M_{\boldsymbol{\sigma}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\sigma}}^T\boldsymbol{\xi}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}} \ge 0) \\ \frac{\eta_{\boldsymbol{\sigma}}}{2}\mathbf{e}^T\mathbf{Pb}\frac{(\|\boldsymbol{\sigma}^*\|^2 - \|\hat{\boldsymbol{\sigma}}\|^2 - \|\tilde{\boldsymbol{\sigma}}\|^2)}{\|\hat{\boldsymbol{\sigma}}\|^2}\hat{\boldsymbol{\sigma}}^T\boldsymbol{\xi}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}} \ge 0 & \text{if } (\|\hat{\boldsymbol{\sigma}}\| = M_{\boldsymbol{\sigma}} \text{ and } \mathbf{e}^T\mathbf{Pb}\hat{\boldsymbol{\sigma}}^T\boldsymbol{\xi}_{\boldsymbol{\sigma}}\hat{\boldsymbol{\alpha}} < 0) \end{cases}$$

(2-68)

Consequently, for any possible condition in (2-56)-(2-58), $G_{\boldsymbol{\alpha}} \geq 0$, $G_{\mathbf{c}} \geq 0$, and $G_{\boldsymbol{\sigma}} \geq 0$ are satisfied. Thus, we can rewrite (2-64) as

$$\dot{V} \leq -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} + \frac{1}{2}\rho^2(\varepsilon + d)^2 \tag{2-69}$$

Assume that there exists a finite constant γ so that [58]

$$\int_0^T (\varepsilon + d)^2 dt \leq \gamma \,, \quad \forall T \in [0, \infty) \tag{2-70}$$

i.e., $(\varepsilon + d) \in L_2[0, T]$, $\forall T \in [0, \infty)$. Integrating both sides of the inequality (2-69) yields

$$V(T) - V(0) \leq -\frac{1}{2}\int_0^T \mathbf{e}^T\mathbf{Q}\mathbf{e}dt + \frac{\rho^2}{2}\int_0^T (\varepsilon + d)^2 dt \,, 0 \leq T < \infty \,. \tag{2-71}$$

Since $V(T) \geq 0$, the following L_2 criterion can be obtained.

$$\frac{1}{2}\int_0^T \mathbf{e}^T\mathbf{Q}\mathbf{e}\, dt \leq V(0) + \frac{\rho^2}{2}\int_0^T (\varepsilon + d)^2 dt \,, \quad 0 \leq T < \infty \,. \tag{2-72}$$

Substituting (2-61) into (2-72), we have the L_2 criterion shown in (2-60). This completes the proof. **Q.E.D.**

From (2-72), we can see that because $V(0)$ is finite, the effect of lumped uncertainty and external disturbance on tracking error can be eliminated as small as possible by choosing an arbitrarily small attenuation level ρ.. In other words, a smaller ρ. results in smaller tracking error, which implies better tracking performance. The following Theorem 2-2 will present an explicit formulation of tracking error.

Theorem 2-2: The tracking error $\|\mathbf{e}\|$ can be expressed in terms of the sum of lumped uncertainty and external disturbance as

$$\|\mathbf{e}\| \leq \sqrt{\frac{2V(0) + \rho^2\gamma}{\lambda_{\min}(\mathbf{P})}} \tag{2-73}$$

Proof:

From (2-71), with the knowledge $\int_0^T \mathbf{e}^T\mathbf{Q}\mathbf{e}dt \geq 0$ and assumption (2-70), we have

$$2V(T) \leq 2V(0) + \rho^2\gamma \,, \quad 0 \leq T < \infty \,. \tag{2-74}$$

From (2-61), it is obvious that $\mathbf{e}^T\mathbf{P}\mathbf{e} \leq 2V$ for any V. Because $\mathbf{P}$ is a positive definite symmetric matrix, we have

$$\lambda_{\min}(\mathbf{P})\|\mathbf{e}\|^2 = \lambda_{\min}(\mathbf{P})\mathbf{e}^T\mathbf{e} \leq \mathbf{e}^T\mathbf{P}\mathbf{e} \tag{2-75}$$

where $\lambda_{\min}(\mathbf{P})$ is the minimum eigenvalue of $\mathbf{P}$. Thus, we obtain

$$\lambda_{\min}(\mathbf{P})\|\mathbf{e}\|^2 \le \mathbf{e}^T\mathbf{P}\mathbf{e} \le 2V(T) \le 2V(0) + \rho^2\gamma \tag{2-76}$$

from (2-74)-(2-75). Therefore (2-76) can be rearranged to yield the following important formula

$$\|\mathbf{e}\| \le \sqrt{\frac{2V(0)+\rho^2\gamma}{\lambda_{\min}(\mathbf{P})}} \tag{2-77}$$

which explicitly describe the tracking error $\|\mathbf{e}\|$ in terms of the sum of lumped uncertainty and external disturbance. **Q.E.D.**

If initial state $V(0)=0$, tracking error $\|\mathbf{e}\|$ can be made arbitrarily small by choosing adequate ρ. Unlike the results in [50-51], (2-77) is very crucial to show that the proposed RASFC will provide the closed-loop stability rigorously in the Lyapunov sense.

Remark 2-4: Affine systems can be viewed as a special kind of nonaffine systems [59]. Consider an SISO nonlinear affine system

$$x^{(n)} = F(\mathbf{x}) + G(\mathbf{x})u + d \tag{2-78}$$

where $\mathbf{x} = [x\ \dot{x} \ldots x^{(n-1)}]^T$ is the state vector of the system, $F(\mathbf{x})$ and $G(\mathbf{x})$ are unknown nonlinear mapping, u is the control input of the system, and d is a bounded external disturbance. By letting $f(\mathbf{x},u) = F(\mathbf{x}) + G(\mathbf{x})u$, we can easily find that the nonlinear affine system (2-78) can be viewed as a special case of nonaffine nonlinear system (2-1). Thus, the proposed RASFC scheme can be directly applied to such a nonlinear affine system when necessary assumptions hold. The overall RASFC can be shown in Fig. 2-4.

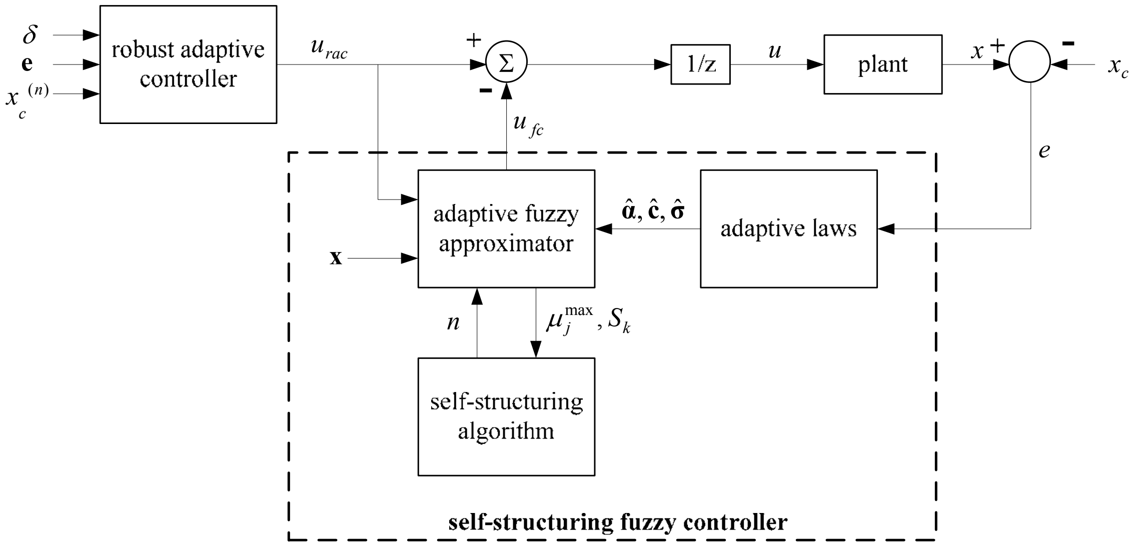

δ
$\mathbf{e}$
$x_c^{(n)}$
robust adaptive controller
u_{rac}
+
Σ
-
u_{fc}
1/z
u
plant
x
+
-
x_c
e
adaptive fuzzy approximator
$\hat{\boldsymbol{\alpha}}, \hat{\mathbf{c}}, \hat{\boldsymbol{\sigma}}$
adaptive laws
$\mathbf{x}$
n
$\mu_j^{\max}, S_k$
self-structuring algorithm
self-structuring fuzzy controller

Fig. 2-4 The block diagram of RASFC for nonaffine nonlinear systems

Table 2-1 Three conditions in Example 2-1

desired trajectory of tracking control: x_c=sin(1.5t)		
	number of rules	consequents of newly generated fuzzy rules
Condition 1a	fixed (4 rules)	
Condition 1b	t < 5: the same 4 rules in Condition 1a are used. $t \geqq 5$: rule growing is operated	initialized from zeros
Condition 1c	t < 5: the same 4 rules in Condition 1a are used $t \geqq 5$: rule growing is operated	initialized according to (2-27)

2.4 Simulation Results

In this section, the simulations are performed using MATLAB under Windows XP. Four examples are presented. Approximations of unknown nonlinear functions are shown in Examples 2-1 and 2-2 to reveal the growing and pruning capabilities of the proposed self-structuring algorithm, respectively. Examples 2-3 and 2-4 are used to examine the applicability and effectiveness of the proposed RASFC system for nonaffine nonlinear control problems. Two cases are performed in Examples 2-3 and 2-4 for comparison purpose. Case 3a and Case 4a show the effectiveness of the SFS with both rules growing and pruning capabilities. In Case 3b, an adaptive FS with fixed number of rules is adapted, and the parameters of the FS are also tuned by adaptive laws (2-56)-(2-58). In Case 4b, only the growing of fuzzy rules by SFS is considered. It can be easily shown that the following examples of nonaffine system control satisfy $\frac{\partial f(\mathbf{x},u)}{\partial u} > 0$. It should be emphasized that the development of the RASFC does not need to know the exact system dynamics of the controlled systems.

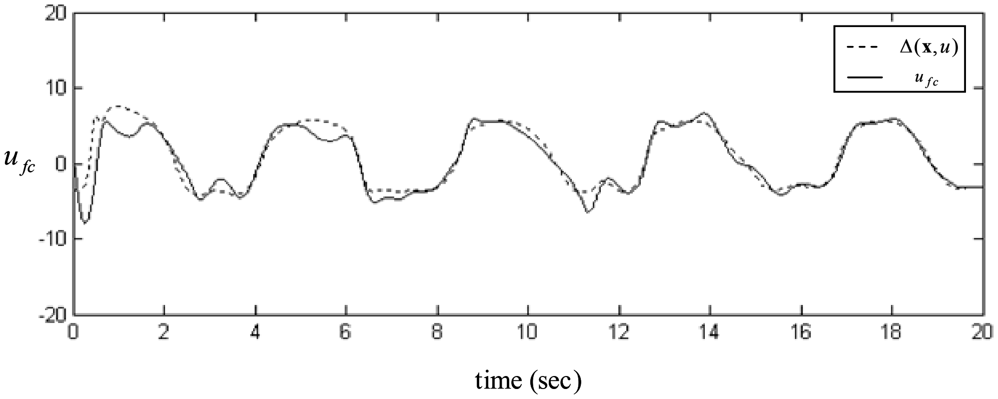
Example 2-1: Consider the following nonaffine nonlinear system [60]:

$$\dot{x}_1 = x_2$$

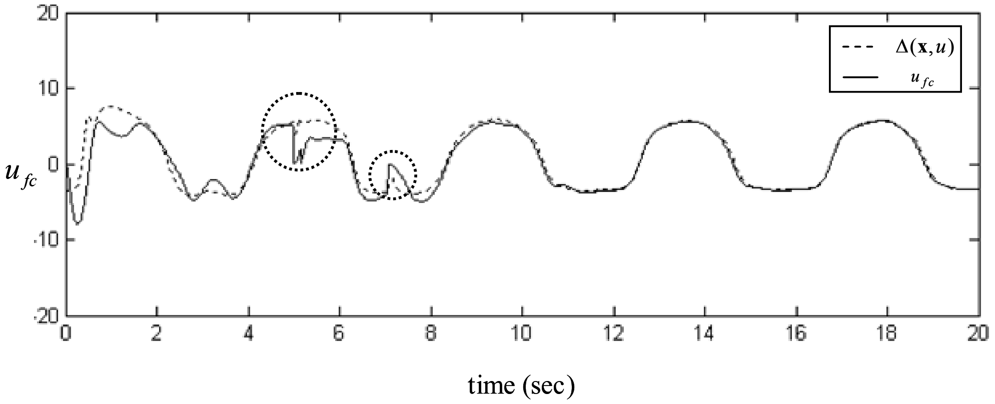
$$\dot{x}_2 = x_1{}^2 + 0.15u^3 + 0.1(1 + x_2{}^2)u + \sin(0.1u) \qquad (2\text{-}79)$$

In tracking control, the SFS is used to approximate an unknown function

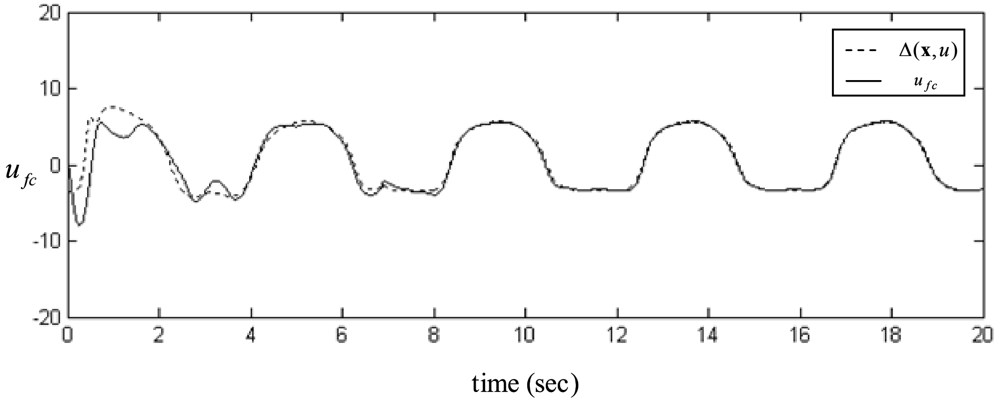
$\Delta(\mathbf{x},u) = x_1^{\ 2} + 0.15u^3 + 0.1(1+x_2^{\ 2})u + \sin(0.1u) - cu$. To illustrate the rule growing capability of the self-structuring algorithm, the approximation is performed under three conditions as shown in Table 2-1. Figures 2-5(a)-2-5(c) show the approximation results of Condition 1a, 1b and 1c, respectively, Fig. 2-5(d) shows the absolute value of the modeling error, $|\tilde{u}|$, and Fig. 2-5(e) shows the number of fuzzy rules. The approximation performances under Conditions 1a and 1b are better than that under Condition 1a after $t \geq 5$. In Fig. 2-5(b), the abrupt variations are marked by circles. These abrupt variations are obviously caused by the rule generation so that the approximation performance is affected for a short period. In Fig. 2-5(c), this phenomenon is mitigated by using (2-27) discussed in *Remark 2-1*. From Fig. 2-5(d), we can see the approximation performance under Condition 1c is the best among three conditions.



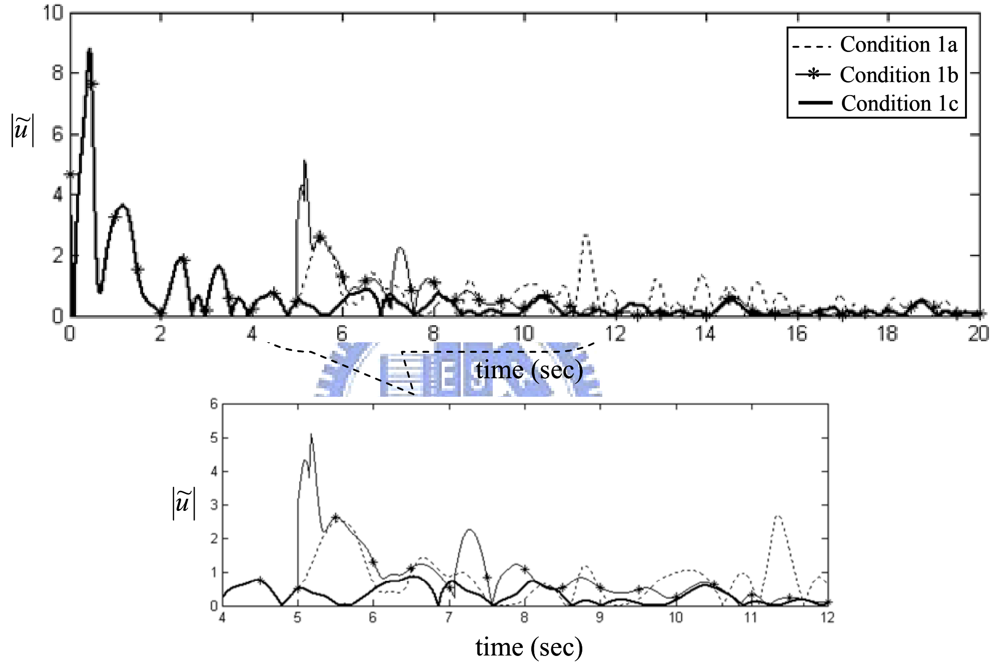
(a)



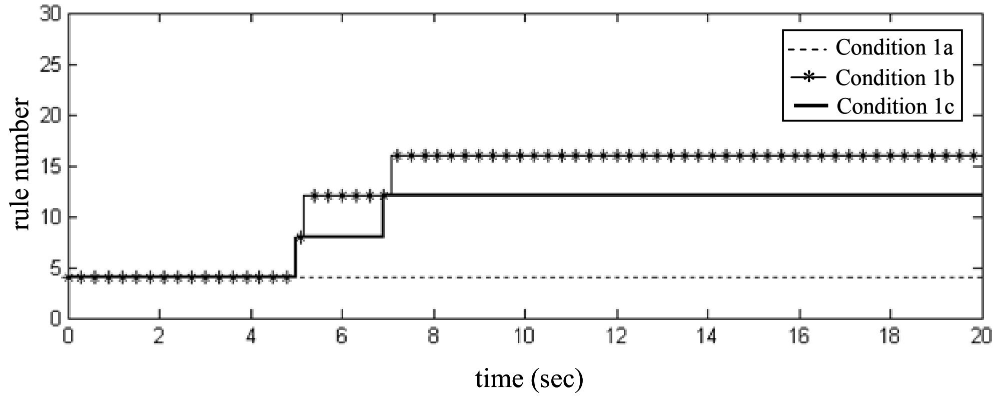
(b)



(c)



(d)



(e)

Fig. 2-5 Approximation results in Example 2-1

Table 2-2 Two conditions in Example 2-2

desired trajectory of tracking control: x_c=1.5sin(t)	
	rule number
Condition 2a	fixed (40 rules)
Condition 2b	t ≧0, rule pruning is operated

Example 2-2: A third-order Chua's chaotic circuit is a simple electronic system that consists of one linear resistor (R_c), two capacitors (C_1, C_2), one inductor (L), and one nonlinear resistor (η). It has been shown to own very rich nonlinear dynamics such as chaos and bifurcations. The dynamic equations of Chua's circuit are written as [9-10]

$$\begin{aligned} \dot{v}_{C_1} &= \frac{1}{C_1}\left(\frac{1}{R}(v_{C_2}-v_{C_1})-\eta(v_{C_2})\right) \\ \dot{v}_{C_2} &= \frac{1}{C_2}\left(\frac{1}{R}(v_{C_1}-v_{C_2})+i_L\right) \\ \dot{i}_L &= \frac{1}{L}(-v_{C_1}-R_0 i_L) \end{aligned} \tag{2-80}$$

where the voltages v_{C_1}, v_{C_2} and current i_L are state variables, R_0 is a constant, and η denotes the nonlinear resistor, which is a function of the voltage across the two terminals of C_1.Here, ϕ is defined as a cubic function as

$$\phi = \lambda_1 v_{C_1} + \lambda_2 v_{C_1}^3 \quad (\lambda_1 < 0, \lambda_2 > 0). \tag{2-81}$$

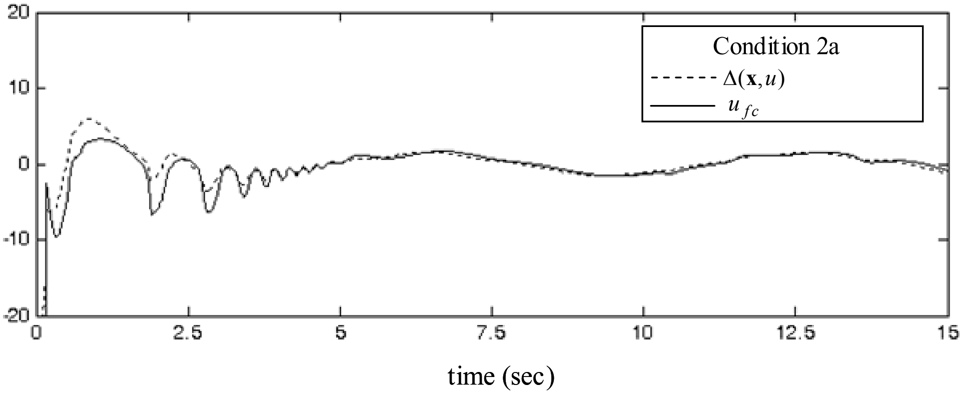
The state equations in (2-80) are not in the standard canonical form. Therefore, a linear transformation is needed to transform them into the form of (2-1). Then, the dynamic equations of transformed Chua's circuit can be rewritten as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= F + u \\ y &= x_1 \end{aligned} \tag{2-82}$$

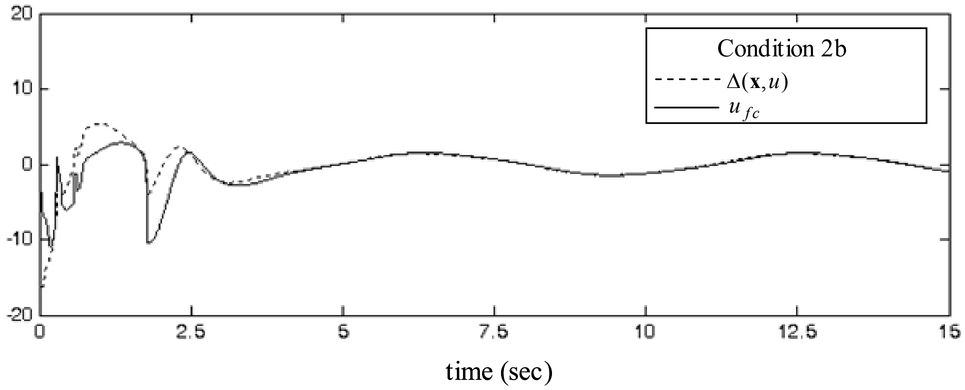
where $\mathbf{x} = [x_1 \ x_2 \ x_3]^T$ is the state vector of the system which is assumed to be available; the system dynamic function

$$F = \frac{14}{1805}x_1 - \frac{168}{9025}x_2 + \frac{1}{38}x_3 - \frac{2}{45}\left(\frac{28}{361}x_1 + \frac{7}{95}x_2 + x_3\right)^3 \tag{2-83}$$

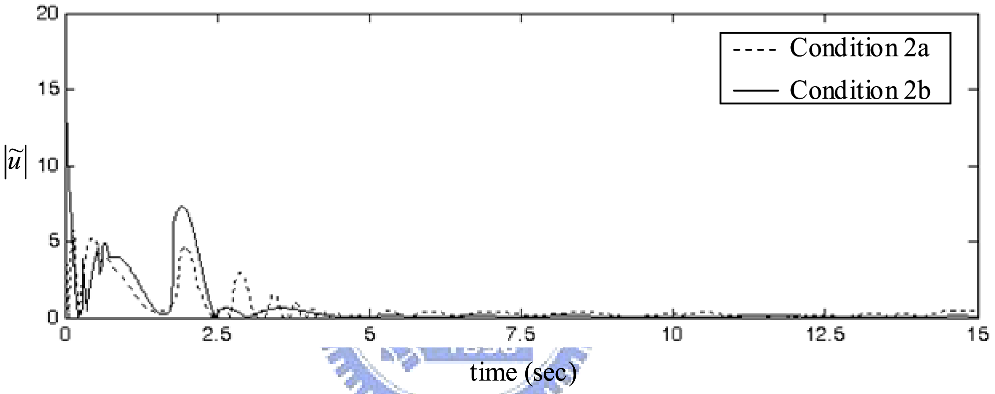
and u is the control input. The reference signal is $y_r(t) = 1.5\sin(t)$. In tracking control, the SFS is used to approximate an unknown function $\Delta(\mathbf{x}, u) = F + u - cu$. To illustrate the rule pruning of the self-structuring algorithm, the approximation is performed under two conditions as shown in Table 2-2. Figures 2-6(a)-2-6(b) show the approximation results. Figure 2-6(c) shows the approximation error *E*. Figure 2-6(d) shows the number of fuzzy rules. Taking the last pruned rule for example, we record the contribution and significance index of the rule pruned at *t=2*.28 in Fig. 2-6(e). Figures. 2-6(a)-2-6(c) show that the approximation performances of Conditions 2a and 2b are both quit well. However, the convergence speed of $|\tilde{u}|$ under Condition 2b is faster than that of Condition 2a. This shows that the parameter training of a large number of fuzzy rules slow down the convergence speed of approximation, and the pruned rules under Condition 2b are redundant and ineffective to the approximation performance. In Fig. 2-6(e), we show the contribution and significance index of a certain rule pruned at *t*=2.28. When the contribution calculated by (2-23) is smaller than a given constant β=0.005, the significance index (2-24) decays with decay constant τ=0.99. Once the significance index is smaller than the pruning threshold $\Theta_p = 0.005$ at *t*=2.28, this rule is insignificant thereafter and thus pruned to ease computational load.



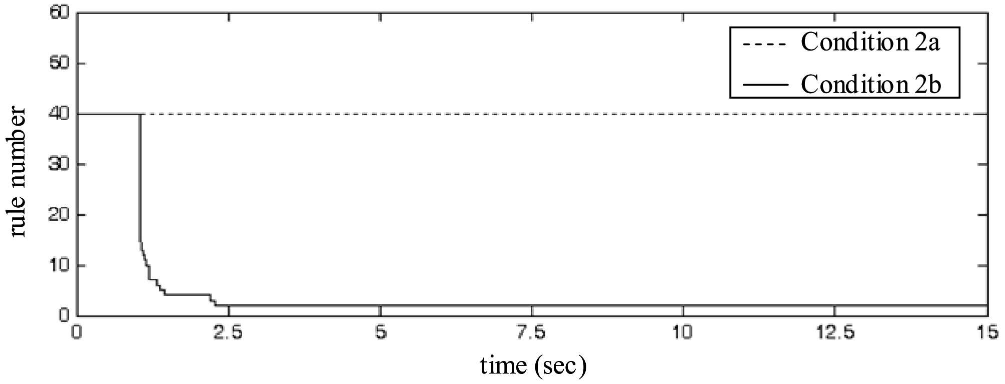
(a)


Condition 2b
$\Delta(\mathbf{x},u)$
u_{fc}
20
10
0
-10
-20
0
2.5
5
7.5
10
12.5
15
time (sec)

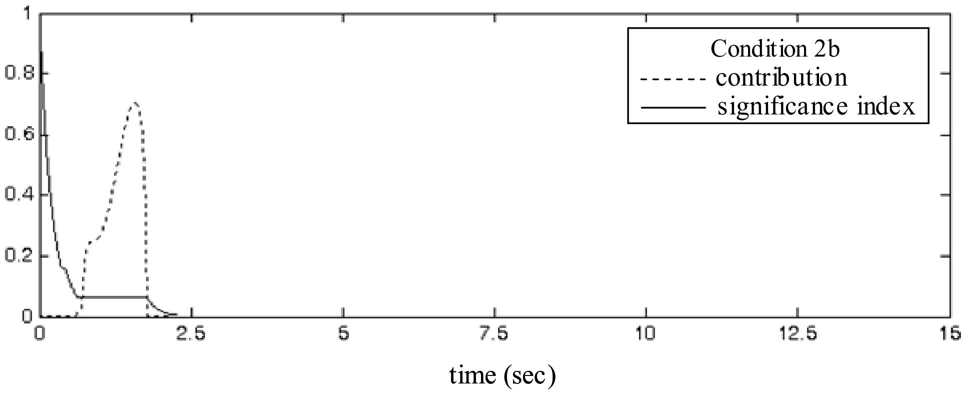
(b)


Condition 2a
Condition 2b
$|\tilde{u}|$
20
15
10
5
0
0
2.5
5
7.5
10
12.5
15
time (sec)

(c)


Condition 2a
Condition 2b
rule number
60
50
40
30
20
10
0
0
2.5
5
7.5
10
12.5
15
time (sec)

(d)



(e)

Fig. 2-6 Approximation results in Example 2-2

Example 2-3: Consider the following nonaffine nonlinear system [61]

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = 0.2(1+e^{x_1 x_2})[(2+\sin(x_2)](u+e^{u}-1)+d \qquad (2\text{-}84)$$

where d is a square wave with amplitude ± 3.0 and period 5 seconds. The desired trajectory is $x_d(t)=\sin(0.5t)+\cos(t)$. The initial sates are chosen as $\mathbf{x}(0)=[x_1(0)\ \ x_2(0)]=[0\ \ 0]^T$. The learning rates are selected as $\eta_{\alpha}=120$ and $\eta_{c}=\eta_{\sigma}=1$. The thresholds for growing and pruning criteria in Case 3a are selected as $\Theta_g=0.1$ and $\Theta_p=0.01$, respectively. These parameters are chosen through some trials to achieve favorable transient control performance. For a choice of **Q**=2**I**, $\mathbf{K}=[2\ \ \ 1]^T$, and $\rho^2=\delta$, we solve the Riccati-like equation shown in (2-62) and obtain the a positive definite symmetric matrix **P**:

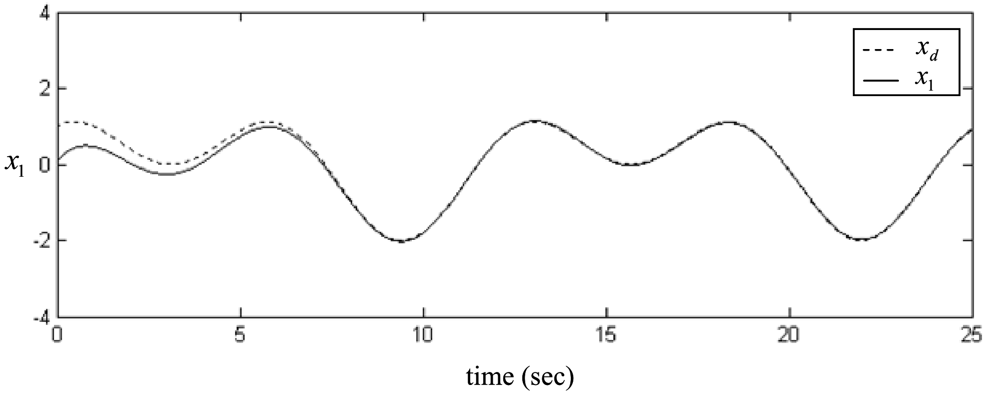
$$\mathbf{P}=\begin{bmatrix} 3.5 & 0.5 \\ 0.5 & 1.5 \end{bmatrix} \qquad (2\text{-}85)$$

The simulation results for Cases 3a and 3b are shown in Figs. 2-7 and 2-8, respectively. The tracking responses of state x_1 are shown in Figs. 2-7(a) and 2-8(a), the tracking responses of

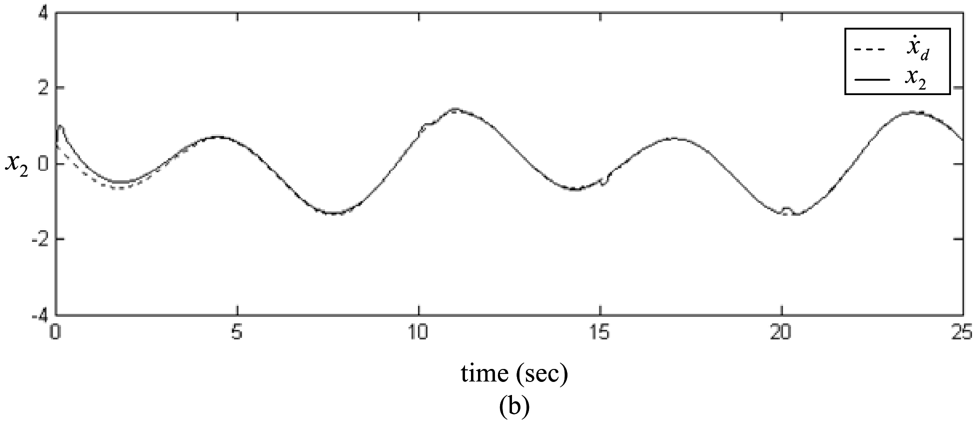
Table 2-3 Comparison between two cases in Example 2-3

1.25×10^4 iterations	Case 3a	Case 3b
maximum number of rules at any time instant	7	4 (fixed)
accumulated sum of rule number, N_a	34,577	60,000
total execution time, t_e (sec)	12.88	18.14

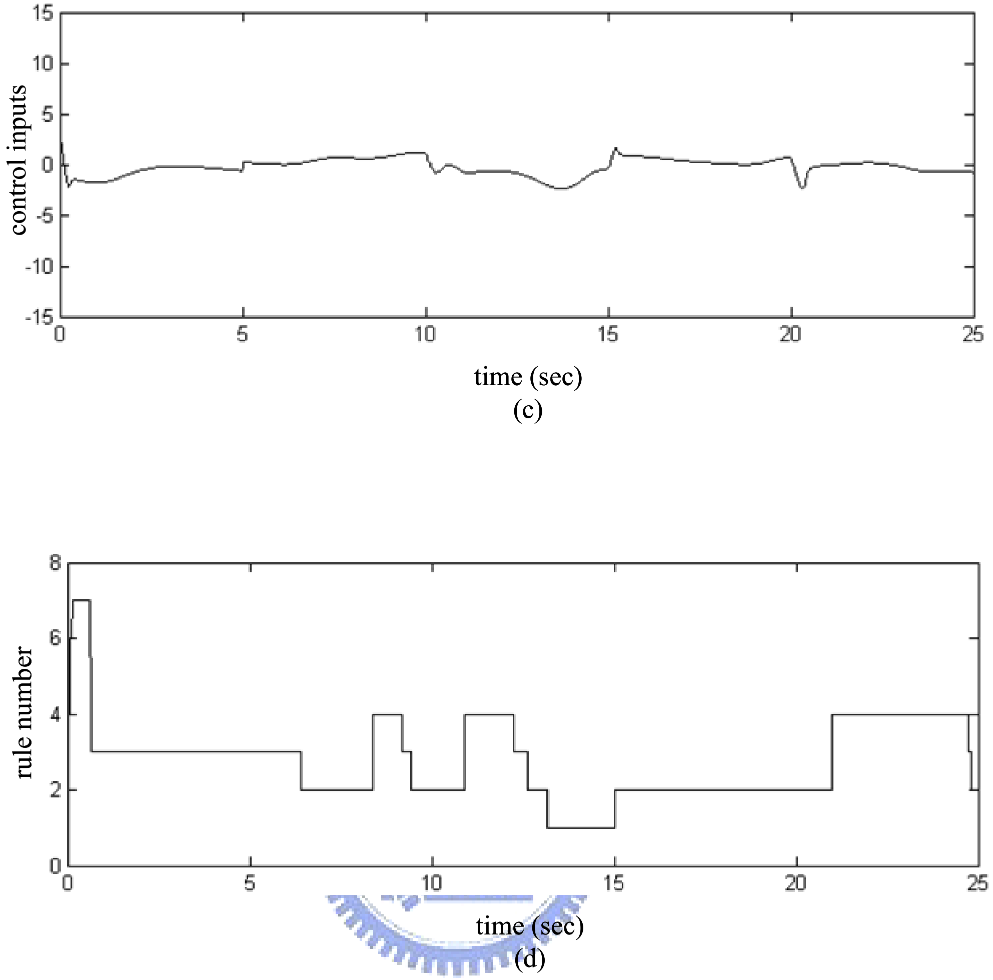
state x_2 are shown in Figs. 2-7(b) and 2-8(b), the associated control inputs are shown Figs. 2-7(c) and 2-8(c), and the numbers of fuzzy rules at every iteration are shown in Figs. 2-7(d) and 2-8(d). From Figs. 2-7(a)-2-7(b) and Figs. 2-8(a)-2-8(b), we can see that the tracking performance in Case 3a is better than that in Case 3b under the external disturbance. In Fig. 2-7(d) the maximum number of rules is 7; in Fig. 2-8(d), the number of rules is 4. Table 2-3 shows the comparison between the two cases, where N_a represents the accumulated sum of computed rules, and t_e denotes the total execution time during the simulation. The proposed self-structuring algorithm can relieve the heavy computational burden caused by 25,423 redundant rules (42.37% of the N_a in Case 3b), and the t_e in Case 3a is nearly one-half times faster than that in Case 3b.



(a)



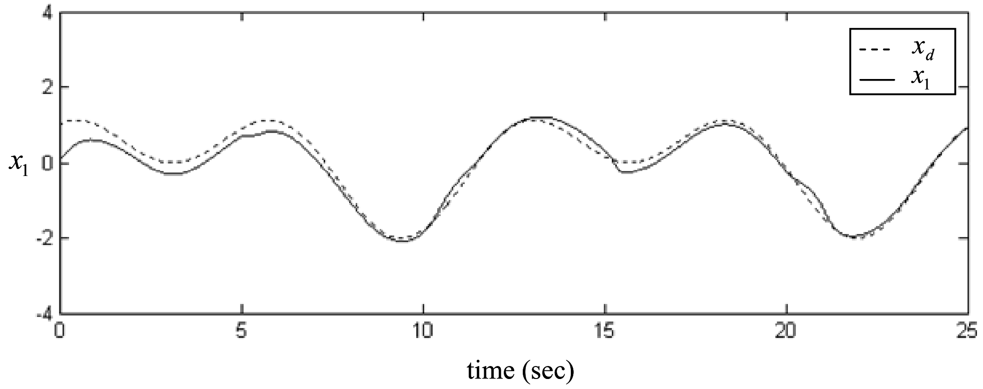
(b)



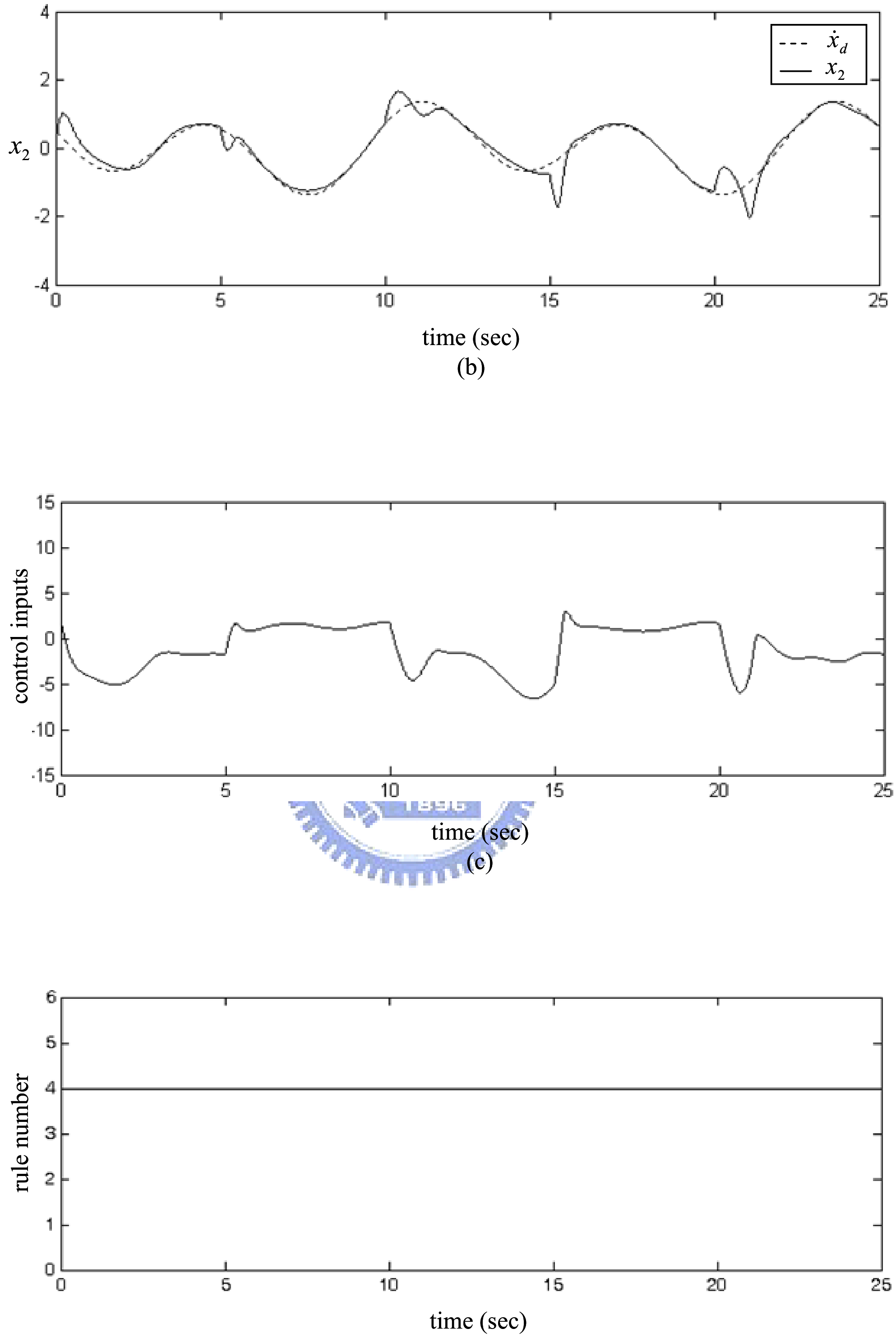
(c)

(d)

Fig. 2-7 Simulation results of Case 3a in Example 2-3



(a)



(b)

(c)

(d)

Fig. 2-8 Simulation results of Case 3b in Example 2-3

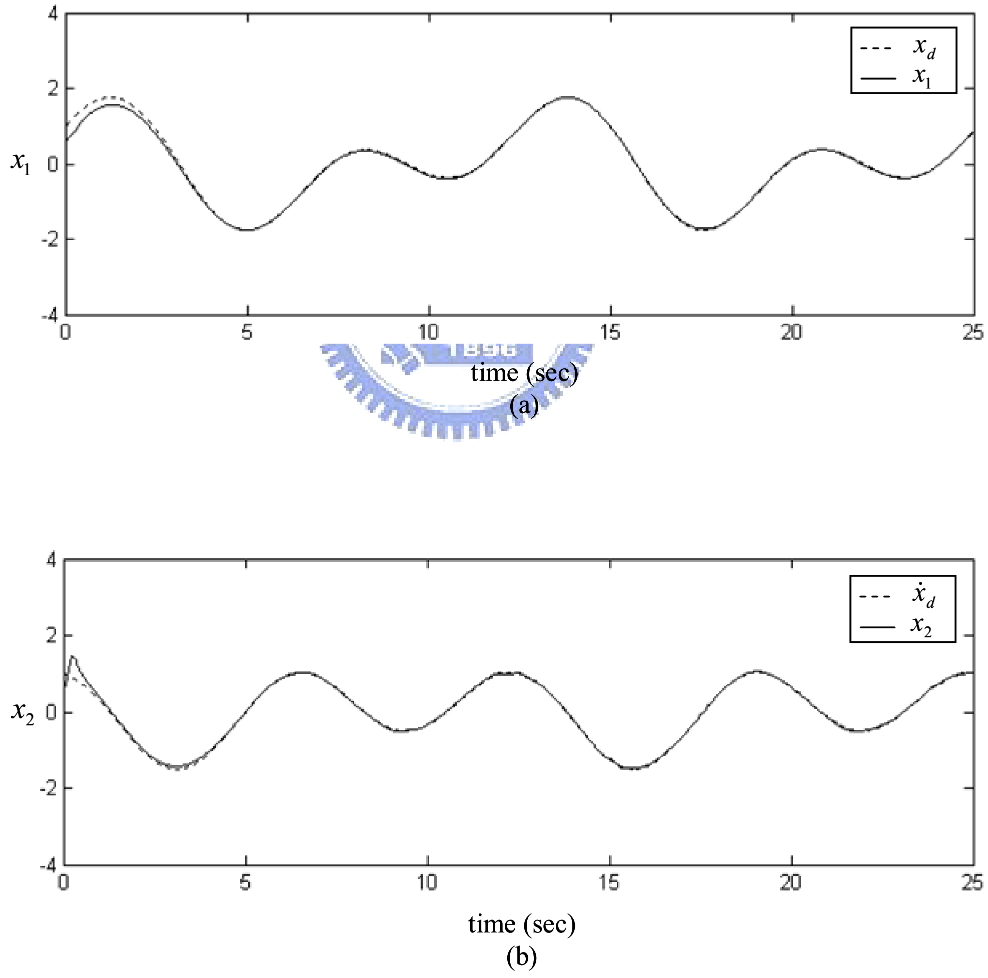
Table 2-4 Comparison between two cases in Example 2-4

1.25×10^4 iterations	Case 4a	Case 4b
maximum number of rules at any time instant	7	28
accumulated sum of computed fuzzy rules, N_a	39,973	227,650
total execution time, t_e (sec)	12.72	64.89

Example 2-4: The Van der Pol oscillator is the main model of self-oscillatory system with two dimensional phase space [13-15]. The oscillator and its extensions have been implemented in various types of electrical circuits. The nonaffine second-order Van der Pol oscillator with nonlinear damping is described as [62]

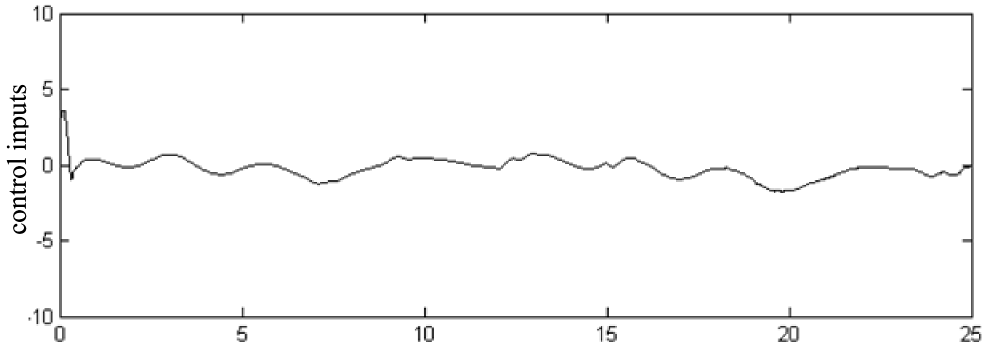
$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 + x_2 + u + (x_1^2 + x_2^2)(\frac{1+e^{-u}}{1-e^{-u}}) - x_1^2 x_2 + d \end{aligned} \qquad (2\text{-}86)$$

where d is a white noise with power 2 which occurs after $t\geq 15$ The desired trajectory is $x_d(t) = \sin(t) + \cos(0.5t)$, and the initial state is $\mathbf{x}(0) = [x_1(0)\ \ x_2(0)] = [0.6\ \ 0.5]^T$. All other parameter settings are chosen the same as those in Example 2-3. The simulation results for Cases 4a and 4b are shown in Figs. 2-9 and 2-10, respectively. The tracking responses of state x_1 are shown in Figs. 2-9(a) and 2-10(a), the tracking responses of state x_2 are shown in Figs. 2-9(b) and 2-10(b), the associated control inputs are shown Figs. 2-9(c) and 2-10(c), and the numbers of fuzzy rules at every iteration are shown in Figs. 2-9(d) and 2-10(d). From the simulation results, we can see that that the proposed RASFC scheme in Case 4a can achieve the same favorable tracking performance as that in Case 4b even an external disturbance suddenly occurs. In Fig. 2-9(d), rule growing plays the major role in SFS within $0 \leq t < 0.25$ and thus the rule number is increased from one to produce a suitable control effort to suppress the tracking error. For $t > 0.25$, to reduce tracking error, the pruning of unnecessary rules will be activated in SFS and thus the number of rules decreases gradually. After a large external disturbance occurs at $t \geq 15$, the rule number apparently increases to eliminate the effect caused by the disturbance. When tracking error is again suppressed to a small level, the rule pruning effect will be activated again. In Fig. 2-10(d), the number of rules increases very rapidly from the beginning to the end of control. Throughout the control process, the maximum number of rules is 7 in Case 4a and 28 in Case 4b. Table 2-4 shows the comparison between two cases. From Table 2-4, it is obvious that our proposed self-structuring algorithm can relieve the heavy computational burden caused by the 187,677 redundant rules (82.44 % of the N_a in Case 4b), and the t_e in Case 4a is over 5 times faster than that in Case 4b. It can be imagined that the relief of computational load caused by the redundant rules will become more and more remarkable as the control period continues.



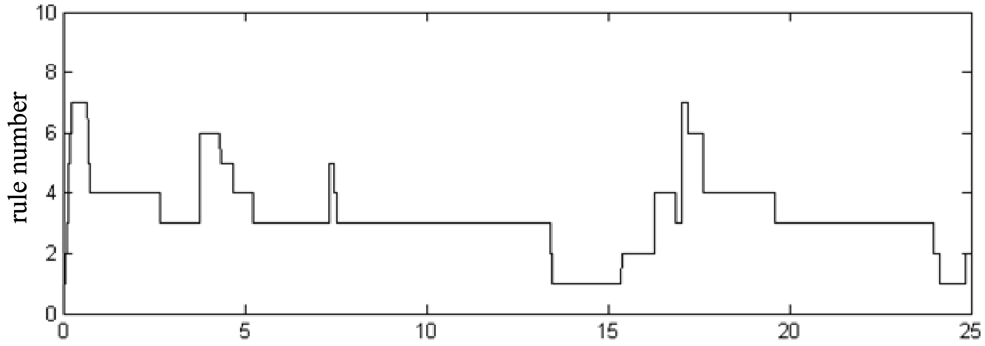
(a)

(b)



time (sec)

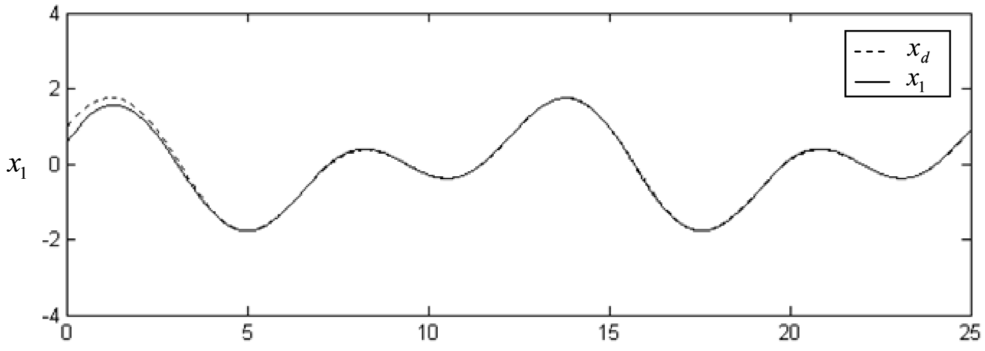
(c)



time (sec)

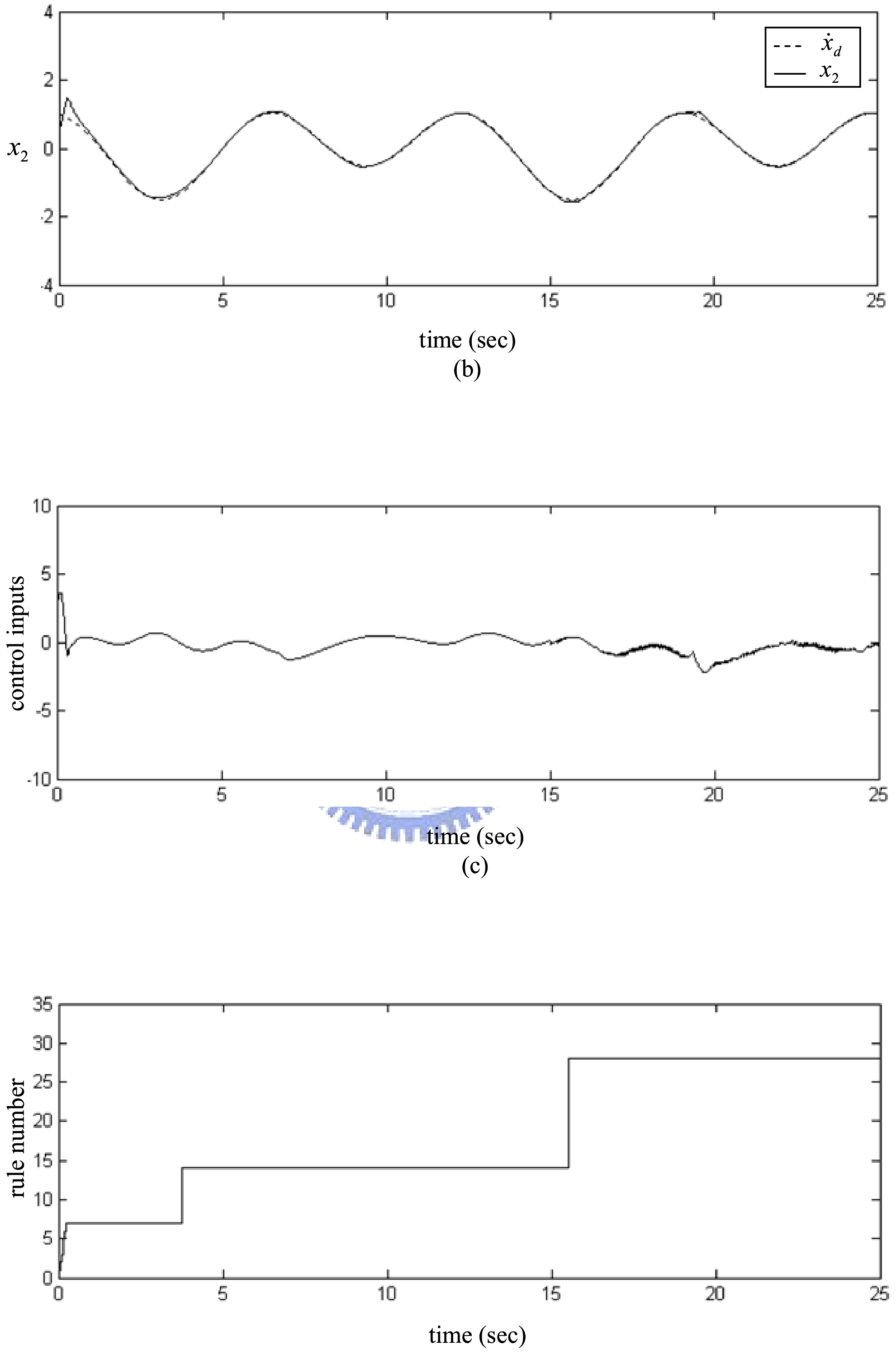
(d)

Fig. 2-9 Simulation results of Case 4a in Example 2-4



time (sec)

(a)



(b)

(c)

(d)

Fig. 2-10 Simulation results of Case 4b in Example 2-4

It is worth noting that in Examples 2-3 and 2-4, the tracking control is started with only one fuzzy rule, and thereafter a compact rule base is constructed automatically without human knowledge. In addition, the same parameter settings, including constants to be designed, learning rates, thresholds of growing and pruning, and the positive definite symmetric matrix **P**, are adopted in these two examples. These parameter settings are chosen for Example 2-3 to achieve favorable transient tracking performance, and they may be not equally suitable for Example 2-4. Nevertheless, as we can see, satisfactory tracking performance is still achieved in these two examples.

Chapter 3

Direct Adaptive Control Design Using Hopfield-Based Dynamic Neural Network for Affine Nonlinear Systems

A dynamic neural network (DNN) is a collection of dynamic neurons which are fully interconnected to a function of their own output. On the contrary, in a static neural network (SNN), the output is directly calculated from the input through feedforward interconnections. DNNs are proven to be more suitable for representing dynamic systems. In this chapter, we aim at solving the control problem of SISO affine nonlinear systems. A direct adaptive control scheme using a Hopfield-based DNN is developed to achieve this goal. Meantime, the structuring problem of NNs is solved by the proposed parsimonious structure of the Hopfield-based DNN, that is, only a single Hopfield neuron is needed to control any affine nonlinear system.

3.1 Hopfield-Based Dynamic Neural Network

3.1.1 Description of DNN Model

DNNs are made of recurrent and interconnected dynamic neurons which distinguish DNNs from feedforward neural works, where the output of one neuron is connected only to neurons in the next layer. Consider a DNN described by a nonlinear differential equation of the following form [47]

$$\dot{\boldsymbol{\chi}} = \mathbf{A}\boldsymbol{\chi} + \mathbf{B}\mathbf{W}\boldsymbol{\sigma}(\mathbf{V}_1\boldsymbol{\chi}) + \mathbf{B}\boldsymbol{\Psi}\boldsymbol{\varphi}(\mathbf{V}_2\boldsymbol{\chi})\boldsymbol{\gamma}(\overline{\mathbf{u}}) \tag{3-1}$$

where $\boldsymbol{\chi} = [\chi_1\ \chi_2 \cdots \chi_n]^T \in R^n$ is the state vector, $\overline{\mathbf{u}} = [\overline{u}_1\ \overline{\mathrm{u}}_2 \cdots \overline{u}_m]^T \in R^m$ is the input vector, $\boldsymbol{\sigma} : R^r \to R^k$, $\mathbf{A} \in R^{n\times n}$ is a Hurwitz matrix. $\mathbf{B} = \mathrm{diag}\{b_1, b_2, \cdots, b_n\} \in R^{n\times n}$, $\mathbf{W} \in R^{n\times k}$, $\mathbf{V}_1 \in R^{r\times n}$, $\boldsymbol{\Psi} \in R^{n\times l}$, $\mathbf{V}_2 \in R^{s\times n}$, $\boldsymbol{\varphi} : R^s \to R^{l\times n}$, and $\boldsymbol{\gamma} : R^m \to R^n$. Here $\boldsymbol{\chi}$ is

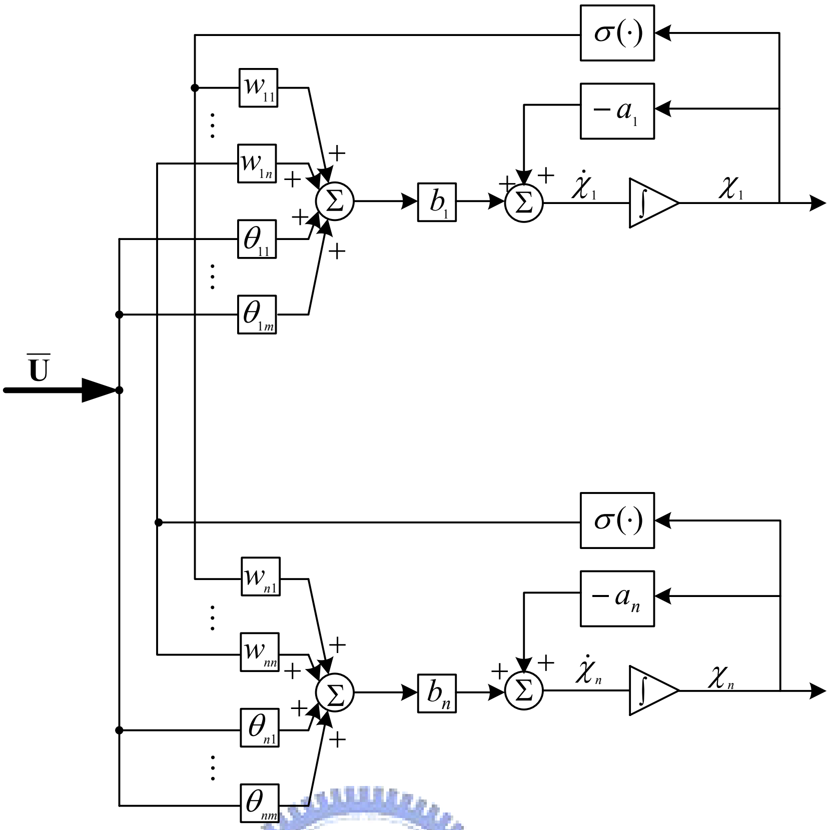


Fig. 3-1 The structure of the dynamic neural network

the state of the DNN, $\mathbf{W}$ and $\mathbf{\Psi}$ are the weight matrices describing output layer connections, $\mathbf{V}_1$ and $\mathbf{V}_2$ are the weight matrices describing the hidden layer connections, $\boldsymbol{\sigma}(\cdot)$ is a sigmoid vector function responsible for nonlinear state feedbacks, and $\boldsymbol{\gamma}(\cdot)$ is a differentiable input function. A DNN in (3-1) satisfying

$$r = s = n,\ \ \mathbf{V}_1 = \mathbf{V}_2 = I_{n\times n},\ \ \boldsymbol{\varphi}(\cdot) = I_{n\times n} \tag{3-2}$$

is a simplest DNN without any hidden layers. It can be expressed as

$$\dot{\boldsymbol{\chi}} = \mathbf{A}\boldsymbol{\chi} + \mathbf{BW}\boldsymbol{\sigma}(\boldsymbol{\chi}) + \mathbf{B\Psi}\boldsymbol{\gamma}(\bar{\mathbf{u}}) \tag{3-3}$$

Following the literatures [45, 47, 63], we choose $k = n$, $\mathbf{A} = diag\{-a_1\ \ -a_2\ \ \cdots\ \ -a_n\}$, where $a_i > 0$, i=1, 2, …, n, and $\boldsymbol{\gamma}(\bar{\mathbf{u}}) = [\bar{\mathbf{u}}\ \ \mathbf{0}]^T \in R^n$, where $n \ge m$ to simply our further analysis. Then, the expression in (3-3) can be modified as

$$\dot{\boldsymbol{\chi}} = \mathbf{A}\boldsymbol{\chi} + \mathbf{BW}\boldsymbol{\sigma}(\boldsymbol{\chi}) + \mathbf{B\Theta}\bar{\mathbf{u}} \tag{3-4}$$

where $\mathbf{\Theta} \in R^{n\times m}$ satisfying $\mathbf{\Psi} = [\mathbf{\Theta}\ \ \ \mathbf{0}]^T$. The structure of the DNN is shown in Fig. 3-1. The output of every neuron in Fig. 3-1 can be expressed as

$$\dot{\chi}_i = -a_i\chi_i + b_i W_i^T \boldsymbol{\sigma}(\boldsymbol{\chi}) + b_i \Theta_i^T \bar{\mathbf{u}},\quad i = 1, 2, \cdots, n \tag{3-5}$$

where $W_i^T = [w_{i1}\ w_{i2} \cdots w_{in}]$ and $\Theta_i^T = [\theta_{i1}\ \theta_{i2} \cdots \theta_{im}]$ are the ith rows of $\mathbf{W}$ and $\mathbf{\Theta}$, respectively. Solve the differential equation (3-5), we obtain

$$\chi_i = b_i\left(W_i^T \xi_{W,i} + \Theta_i^T \xi_{\Theta,i}\right) + e^{-a_i t}\chi_i^0 - e^{-a_i t} b_i\left(W_i^T \xi_{W,i}^0 + \Theta_i^T \xi_{\Theta,i}^0\right), \qquad i = 1, 2, \cdots, n \tag{3-6}$$

where χ_i^0 is the initial state of χ_i; $\xi_{W,i} \in R^n$ and $\xi_{\Theta,i} \in R^m$ are the solutions of

$$\dot{\xi}_{W,i} = -a_i \xi_{W,i} + \boldsymbol{\sigma}(\boldsymbol{\chi}) \tag{3-7}$$

and

$$\dot{\xi}_{\Theta,i} = -a_i \xi_{\Theta,i} + \bar{\mathbf{u}} \tag{3-8}$$

respectively; $\xi_{W,i}^0$ and $\xi_{\Theta,i}^0$ are initial states of $\xi_{W,i}$ and $\xi_{\Theta,i}$, respectively. Note that $e^{-a_i t}\chi_i^0$ and $e^{-a_i t} b_i\left(W_i^T \xi_{W,i}^0 + \Theta_i^T \xi_{\Theta,i}^0\right)$ in (3-6) will exponentially decay with time due to the fact $a_i > 0$.

3.1.2 Hopfield-based DNN Approximator

A DNN approximator for continuous functions can be defined as

$$\chi_i = b_i\left(\hat{W}_i^T \xi_{W,i} + \hat{\Theta}_i^T \xi_{\Theta,i}\right) + e^{-a_i t}\chi_i^0 - e^{-a_i t} b_i\left(\hat{W}_i^T \xi_{W,i}^0 + \hat{\Theta}_i^T \xi_{\Theta,i}^0\right), \qquad i = 1, 2, \cdots, n \tag{3-9}$$

where $\hat{W}_i$ and $\hat{\Theta}_i$ are the estimations of W_i and Θ_i, respectively. Define optimal vectors W_i^* and Θ_i^* as

$$\left(W_i^*, \Theta_i^*\right) = \arg \min_{\hat{W}_i \in \mathbf{\Omega}_{W_i},\, \hat{\Theta}_i \in \mathbf{\Omega}_{\Theta_i}} \left\{ \sup_{\chi \in D_\chi,\, \bar{\mathbf{u}} \in D_{\bar{U}}} \left| \Phi_i - \left[b_i\left(\hat{W}_i^T \xi_{W,i} + \hat{\Theta}_i^T \xi_{\Theta,i}\right) + e^{-a_i t}\chi_i^0 - e^{-a_i t} b_i\left(\hat{W}_i^T \xi_{W,i}^0 + \hat{\Theta}_i^T \xi_{\Theta,i}^0\right)\right]\right| \right\} \tag{3-10}$$

where $D_\chi \subset R^N$ and $D_{\bar{U}} \subset R^m$ are compact sets; $\mathbf{\Omega}_{W_i} = \left\{\hat{W}_i : \left\|\hat{W}_i\right\| \le M_{W_i}\right\}$ and $\mathbf{\Omega}_{\Theta_i} = \left\{\hat{\Theta}_i : \left\|\hat{\Theta}_i\right\| \le M_{\Theta_i}\right\}$ are constraint sets for $\hat{W}_i$ and $\hat{\Theta}_i$. Then, a continuous vector function $\mathbf{\Phi} = [\Phi_1\ \Phi_2 \cdots \Phi_n]^T \in R^n$ can be expressed as

$$\Phi_i = b_i\left(W_i^{*T} \xi_{W,i} + \Theta_i^{*T} \xi_{\Theta,i}\right) + e^{-a_i t}\chi_i^0 - e^{-a_i t} b_i\left(W_i^{*T} \xi_{W,i}^0 + \Theta_i^{*T} \xi_{\Theta,i}^0\right) + \Delta_i, \quad i = 1, 2, \cdots, n \tag{3-11}$$

where Δ_i is the approximation error. Note that the optimal vectors W_i^* and Θ_i^* are difficult to be determined and might not be unique. The modeling error $\tilde{\chi}_i$ is defined as

$$
\begin{aligned}
\widetilde{\chi}_i &= \Phi_i - \chi_i \\
&= \left\lfloor b_i \left(W_i^{T^*} \boldsymbol{\xi}_{W,i}^{*} + \Theta_i^{T^*} \boldsymbol{\xi}_{\Theta,i} \right) + e^{-a_i t} \chi_i^0 - e^{-a_i t} b_i \left(W_i^{T^*} \boldsymbol{\xi}_{W,i}^{*,0} + \Theta_i^{*^T} \boldsymbol{\xi}_{\Theta,i}^{0} \right) + \Delta_i \right\rfloor \\
&\quad - \left\lfloor b_i \left(\hat{W}_i^T \hat{\boldsymbol{\xi}}_{W,i} + \hat{\Theta}_i^T \boldsymbol{\xi}_{\Theta,i} \right) + e^{-a_i t} \chi_i^0 - e^{-a_i t} b_i \left(\hat{W}_i^T \hat{\boldsymbol{\xi}}_{W,i}^{0} + \hat{\Theta}_i^T \boldsymbol{\xi}_{\Theta,i}^{0} \right) \right\rfloor \\
&= b_i \left(\widetilde{W}_i^T \boldsymbol{\xi}_{W,i} + \widetilde{\Theta}_i^T \boldsymbol{\xi}_{\Theta,i} \right) - e^{-a_i t} b_i \left(\widetilde{W}_i^T \boldsymbol{\xi}_{W,i}^{0} + \widetilde{\Theta}_i^T \boldsymbol{\xi}_{\Theta,i}^{0} \right) + \Delta_i \qquad i = 1, 2, \cdots, n
\end{aligned}
\tag{3-12}
$$

where $\widetilde{W}_i = W_i^* - \hat{W}_i$, and $\widetilde{\Theta}_i = \Theta_i^* - \hat{\Theta}_i$.

In this paper, a Hopfield-based dynamic neural network is adopted as the approximator. It is known as a special case of DNN with $a_i = 1/(R_i C_i)$ and $b_i = 1/C_i$, where $R_i > 0$ and $C_i > 0$ representing the resistance and capacitance at the *i*th neuron, respectively [25],[29]. The sigmoid function $\boldsymbol{\sigma}(\boldsymbol{\chi}) = [\sigma(\chi_1)\, \sigma(\chi_2) \cdots \sigma(\chi_n)]^T$ is defined by a hyperbolic tangent function as

$$\sigma(\chi_i) = \tanh(\kappa_i \chi_i), \qquad i = 1, 2, \cdots, n \tag{3-13}$$

where κ_i is the slope of $\tanh(\cdot)$ at the origin. It is known that tangent function is bounded by $-1 < \tanh(\cdot) < 1$.

3.2 Problem Formulation

Let $S \subset R^n$ be an open set, $D_S \subset S$ be and compact set. Consider the nth-order nonlinear dynamic system of the form

$$
\begin{aligned}
x^{(n)} &= f(\mathbf{x}) + gu + d \\
y &= x
\end{aligned}
\tag{3-14}
$$

where $\mathbf{x} = [x, \dot{x}, \cdots, x^{(n-1)}]^T$ is the state vector., $f : D_s \to R$ is a uncertain, continuous functions, g is an unknown constant, $u \in R$ is continuous control input of the system, $y \in R$ is the output of the system, and $d \in R$ is a bounded external disturbance. We consider only the nonlinear systems which can be represented in (3-14). In order for (3-14) to be controllable, it is required that $g \neq 0$. Without losing generality, we assume that $0 < g < \infty$. The control objective is to force the system output y to follow a given bounded reference signal $y_r \in C^h$, $h \geq n$. The reference signal vector $\mathbf{y}_r$ and the error vector $\mathbf{e}$ are defined as

$$\mathbf{e} = [e, \dot{e}, \cdots, e^{(n-1)}]^T \in R^n \tag{3-15}$$

with $e = y_r - x = y_r - y$.

If the functions $f(\mathbf{x})$ and g are known and the system is free of external disturbance, the ideal controller can be designed as

$$u_{id} = \frac{1}{g}\left[-f(\mathbf{x}) + y_r^{(n)} + \mathbf{k}_c^T\mathbf{e}\right] \tag{3-16}$$

where $\mathbf{k}_c = \left[k_n \; k_{n-1} \cdots k_1\right]^T$. Applying (3-16) to (3-14), we have the following error dynamics system

$$e^{(n)} + k_1 e^{(n-1)} + \cdots + k_n e = 0. \tag{3-17}$$

If k_i, i=1, 2, …, n are chosen so that all roots of the polynomial $H(s) \triangleq s^n + k_1 s^{n-1} + \cdots + k_n$ lie strictly in the open left half of the complex plane, then $\lim_{t\to\infty} e(t) = 0$ can be implied for any initial conditions. However, since the system dynamics may be unknown or perturbed, the ideal feedback controller u_{id} in (3-16) cannot be implemented.

3.3 Design of DACHDNN

To solve this problem, a new direct adaptive control scheme using Hopfield neural networks for SISO nonlinear systems is proposed. In the DACHDNN, a Hopfield-based DNN is used to estimate the ideal controller u_{id}. The direct adaptive Hopfield-based DNN controller takes the following form

$$u_d = u_{HDNN} + u_s \tag{3-18}$$

where u_{HDNN} is the Hopfield-based DNN controller used to approximate the ideal controller u_{id} in (3-16); u_s is the compensation controller employed to compensate the effects of external disturbance and the approximation error introduced by the Hopfield-based DNN approximation (described later). The overall DACHDNN is shown in Fig. 3-2, wherein the adaptive laws are described later. Substituting (3-18) into (3-14) and using (3-16) yield

$$\begin{aligned}\dot{\mathbf{e}} &= \mathbf{A}_c\mathbf{e} + g\mathbf{B}_c(u_{ideal} - u_{HDNN} - u_s) - \mathbf{B}_c d \\ &= \mathbf{A}_c\mathbf{e} + g\mathbf{B}_c(\tilde{u} - u_s) - \mathbf{B}_c d \end{aligned} \tag{3-19}$$

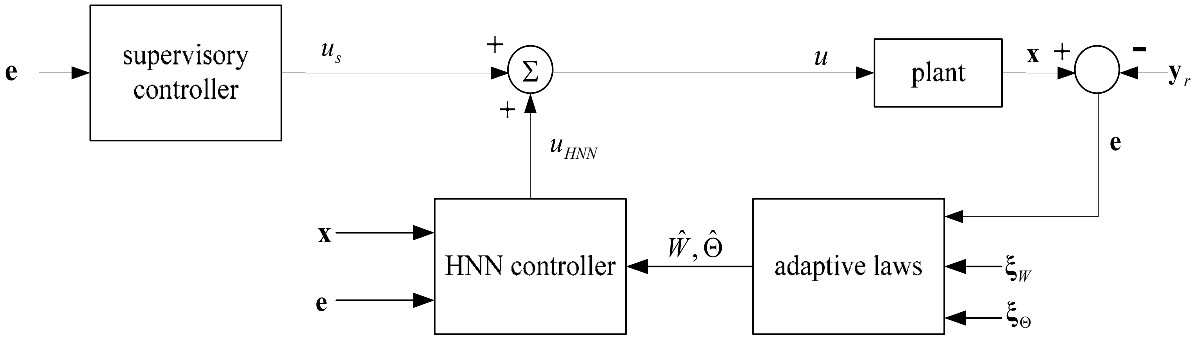


Fig. 3-2 The Block diagram of the DACHDNN

where $\mathbf{A}_c = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ -k_n & -k_{n-1} & \cdots & \cdots & -k_1 \end{bmatrix} \in R^{n\times n}$, $\mathbf{B}_c = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \in R^n$, and $\tilde{u} = u_{ideal} - u_{HDNN}$.

Note that the ideal controller u_{id} is a scalar, and thus the Hopfield-based DNN used to approximate u_{id} contains only a single neuron. The output of such a Hopfield-based DNN can be express as

$$u_{HDNN} = \frac{1}{C}\left(\hat{W}\xi_W + \hat{\Theta}^T\xi_\Theta\right) + e^{-\frac{1}{RC}t}u_{HDNN}^0 - e^{-\frac{1}{RC}t}\frac{1}{C}\left(\hat{W}\xi_W^0 + \hat{\Theta}^T\xi_\Theta^0\right) \tag{3-20}$$

where u_{HDNN}^0 is the initial value of u_{HDNN}. Note that $\hat{W}$ and ξ_W are scalars, and the input signal of the Hopfield-based DNN is $\bar{u} = [e \ \dot{e}]^T$. Fig. 3-3 shows the electric circuit of the Hopfield-based DNN containing only a single neuron.

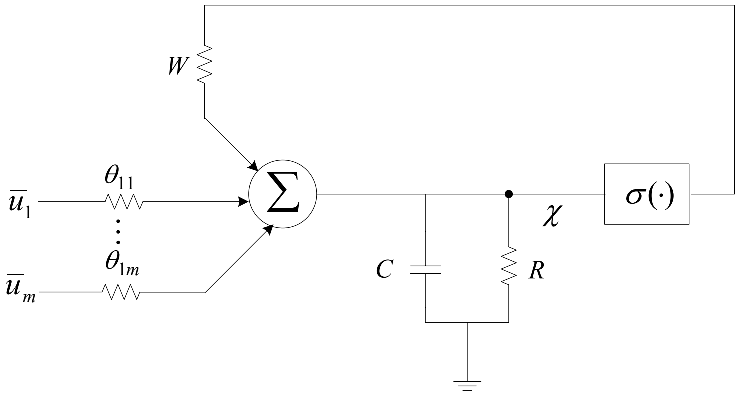


Fig. 3-3 The electric circuit of the Hopfield-based DNN containing only a single neuron

Substituting (3-20) into (3-19) yields

$$\dot{\mathbf{e}} = \mathbf{A}_c\mathbf{e} + g\mathbf{B}_c\left\{\frac{1}{C}\tilde{W}\left(\xi_W - e^{-\frac{1}{RC}t}\xi_W^0\right) + \frac{1}{C}\tilde{\Theta}^T\left(\xi_\Theta - e^{-\frac{1}{RC}t}\xi_\Theta^0\right) + \Delta - u_s\right\} - \mathbf{B}_c d \qquad (3\text{-}21)$$

where Δ is the approximation error. In order to derive the one of the main theorems in this chapter, the following assumption and lemma is required.

Assumption: Let $\varepsilon = \Delta - \frac{1}{g}d$. Assume that there exists a finite constant μ so that

$$\int_0^t \varepsilon^2 d\tau \le \mu,\ \ 0 \le t < \infty. \qquad (3\text{-}22)$$

Lemma: Choose $\hat{W}^0 \in \mathbf{\Omega}_W$ and $\hat{\Theta}^0 \in \mathbf{\Omega}_\Theta$, where $\hat{W}^0$ and $\hat{\Theta}^0$ are the initial values of W and Θ, respectively. If the adaptive laws are designed as

$$\dot{\hat{W}} = -\dot{\tilde{W}} = \begin{cases} \dfrac{\beta_W}{C}\mathbf{e}^T\mathbf{PB}_c\left(\xi_W - e^{-\frac{1}{RC}t}\xi_W^0\right) & \text{if } \left(\left|\hat{W}\right| < M_W\right) \text{or} \left(\left|\hat{W}\right| = M_W \text{ and } \mathbf{e}^T\mathbf{PB}_c\hat{W}\left(\xi_W - e^{-\frac{1}{RC}t}\xi_W^0\right) \ge 0\right) \\ \mathbf{Pr}\left[\dfrac{\beta_W}{C}\mathbf{e}^T\mathbf{PB}_c\left(\xi_W - e^{-\frac{1}{RC}t}\xi_W^0\right)\right] & \text{if } \left(\left|\hat{W}\right| = M_W \text{ and } \mathbf{e}^T\mathbf{PB}_c\hat{W}\left(\xi_W - e^{-\frac{1}{RC}t}\xi_W^0\right) < 0\right) \end{cases} \qquad (3\text{-}23)$$

$$\dot{\hat{\Theta}} = -\dot{\tilde{\Theta}} = \begin{cases} \dfrac{\beta_\Theta}{C}\mathbf{e}^T\mathbf{PB}_c\left(\xi_\Theta - e^{-\frac{1}{RC}t}\xi_\Theta^0\right) & \text{if } \left(\left\|\hat{\Theta}\right\| < M_\Theta\right) \text{or} \left(\left\|\hat{\Theta}\right\| = M_\Theta \text{ and } \mathbf{e}^T\mathbf{PB}_c\hat{\Theta}\left(\xi_\Theta - e^{-\frac{1}{RC}t}\xi_\Theta^0\right) \ge 0\right) \\ \mathbf{Pr}\left[\dfrac{\beta_\Theta}{C}\mathbf{e}^T\mathbf{PB}_c\left(\xi_\Theta - e^{-\frac{1}{RC}t}\xi_\Theta^0\right)\right] & \text{if } \left(\left\|\hat{\Theta}\right\| = M_\Theta \text{ and } \mathbf{e}^T\mathbf{PB}_c\hat{\Theta}\left(\xi_\Theta - e^{-\frac{1}{RC}t}\xi_\Theta^0\right) < 0\right) \end{cases} \qquad (3\text{-}24)$$

where β_W and β_Θ are positive learning rates; the symmetric positive definite matrix $\mathbf{P}$ satisfies the following Riccati-like equation

$$\mathbf{A}_c{}^T\mathbf{P} + \mathbf{PA}_c + \mathbf{Q} + \mathbf{PB}_c\left(\frac{1}{\rho^2} - \frac{1}{\delta}\right)\mathbf{B}_c^T\mathbf{P} = 0 \qquad (3\text{-}25)$$

where $\mathbf{Q}$ is a symmetric positive matrix and $\frac{1}{\rho^2} - \frac{1}{\delta} \le 0$; the projection operators $\mathbf{Pr}[*]$ are defined as

$$\mathbf{P}_{\mathbf{r}}\left[\frac{\beta_W}{C}\mathbf{e}^T\mathbf{PB}_c\left(\xi_W - e^{-\frac{1}{RC}t}\xi_W^0\right)\right] = \frac{\beta_W}{C}\left[\mathbf{e}^T\mathbf{PB}_c\left(\xi_W - e^{-\frac{1}{RC}t}\xi_W^0\right) + \mathbf{e}^T\mathbf{PB}_c\frac{\hat{W}\left(\xi_W - e^{-\frac{1}{RC}t}\xi_W^0\right)}{\left|\hat{W}\right|^2}\hat{W}\right] \qquad (3\text{-}26)$$

and

$$\mathbf{P}_{\mathbf{r}}\left[\frac{\beta_\Theta}{C}\mathbf{e}^T\mathbf{PB}_c\left(\boldsymbol{\xi}_\Theta - e^{-\frac{1}{RC}t}\boldsymbol{\xi}_\Theta^0\right)\right] = \frac{\beta_\Theta}{C}\left[\mathbf{e}^T\mathbf{PB}_c\left(\boldsymbol{\xi}_\Theta - e^{-\frac{1}{RC}t}\boldsymbol{\xi}_\Theta^0\right) + \mathbf{e}^T\mathbf{PB}_c\,\frac{\hat{\Theta}^T\left(\boldsymbol{\xi}_\Theta - e^{-\frac{1}{RC}t}\boldsymbol{\xi}_\Theta^0\right)}{\left\|\hat{\Theta}\right\|^2}\hat{\Theta}\right] \tag{3-27}$$

then $\hat{W}$ and $\hat{\Theta}$ are bounded by $\left|\hat{W}\right| \le M_W$ and $\left\|\hat{\Theta}\right\| \le M_\Theta$ for all $t \ge 0$ [3, 61].

Following the preceding consideration, we have the following theorem.

Theorem 3-1: Suppose the *Assumption* (3-22) holds. Consider the plant (3-14) with the control law (3-18). The Hopfield-based DNN controller u_{HDNN} is given by (3-20) with the adaptive laws (3-23) and (3-24). The compensation controller u_s is given as

$$u_s = \frac{1}{2\delta g_L}\mathbf{B}_c^T\mathbf{Pe} \tag{3-28}$$

where $g_L > 0$ is a known constant satisfying $g_L < g(\mathbf{x}) < \infty$. Then, the overall control scheme guarantees the following properties:

i) $$\frac{1}{2}\int_0^t \mathbf{e}^T\mathbf{Qe}\,d\tau \le \frac{1}{2}\mathbf{e}_0{}^T\mathbf{Pe}_0 + \frac{\tilde{W}_0\dot{\tilde{W}}_0}{2\beta_W} + \frac{\tilde{\Theta}_0^T\dot{\tilde{\Theta}}_0}{2\beta_\Theta}\Theta + \frac{g^2\rho^2}{2}\int_0^t \varepsilon^2\,d\tau \tag{3-29}$$

for $0 \le t < \infty$, where $\mathbf{e}_0$, $\tilde{W}_0$, and $\tilde{\Theta}_0$ are the initial values of $\mathbf{e}$, $\tilde{W}$, and $\tilde{\Theta}$, respectively.

ii) The tracking error $\|\mathbf{e}\|$ can be expressed in terms of the lumped uncertainty as

$$\|\mathbf{e}\| \le \sqrt{\frac{2V^0 + g^2\rho^2\mu}{\lambda_{\min}(\mathbf{P})}} \tag{3-30}$$

where V^0 is the initial value of a Lyapunov function candidate defined later and $\lambda_{\min}(\mathbf{P})$ is the minimum eigenvalue of $\mathbf{P}$.

Proof:

i) Define the Lyapunov function candidate as

$$V = \frac{1}{2}\mathbf{e}^T\mathbf{Pe} + \frac{1}{2\eta_W}\tilde{W}^2 + \frac{1}{2\eta_\Theta}\tilde{\Theta}^T\tilde{\Theta} \tag{3-31}$$

where $\eta_W = \frac{\beta_W}{g}$ and $\eta_\Theta = \frac{\beta_\Theta}{g}$. Differentiating (3-31) with respect to time and using (3-21) yield

$$\dot{V} = \frac{1}{2}\mathbf{e}^T\mathbf{P}\dot{\mathbf{e}} + \frac{1}{2}\dot{\mathbf{e}}^T\mathbf{P}\mathbf{e} + \frac{1}{\eta_W}\tilde{W}\dot{\tilde{W}} + \frac{1}{\eta_\Theta}\tilde{\Theta}^T\tilde{\Theta}$$

$$= \frac{1}{2}\mathbf{e}^T(\mathbf{A}_{c_c}^T\mathbf{P} + \mathbf{P}\mathbf{A}_c)\mathbf{e} + g\mathbf{e}^T\mathbf{P}\,\mathbf{B}_c\left[\frac{1}{C}\tilde{W}\left(\boldsymbol{\xi}_W - e^{-\frac{1}{RC}t}\boldsymbol{\xi}_W^0\right) + \frac{1}{C}\tilde{\Theta}^T\left(\boldsymbol{\xi}_\Theta - e^{-\frac{1}{RC}t}\boldsymbol{\xi}_\Theta^0\right) + \Delta - u_s\right]$$

$$-\mathbf{e}^T\mathbf{P}\mathbf{B}_c d + \frac{1}{\eta_W}\tilde{W}\dot{\tilde{W}} + \frac{1}{\eta_\Theta}\tilde{\Theta}^T\dot{\tilde{\Theta}}$$

$$= \frac{1}{2}\mathbf{e}^T(\mathbf{A}_{c_c}^T\mathbf{P} + \mathbf{P}\mathbf{A}_c)\mathbf{e} - g\mathbf{e}^T\mathbf{P}\,\mathbf{B}_c u_s + g\mathbf{e}^T\mathbf{P}\,\mathbf{B}_c\left(\Delta - \frac{1}{g}d\right) + V_W + V_\Theta \tag{3-32}$$

where

$$V_W = g\tilde{W}\left[\frac{1}{C}\mathbf{e}^T\mathbf{P}\,\mathbf{B}_c\left(\boldsymbol{\xi}_W - e^{-\frac{1}{RC}t}\boldsymbol{\xi}_W^0\right) + \frac{1}{\beta_W}\dot{\tilde{W}}\right] \tag{3-33}$$

and

$$V_\Theta = g\tilde{\Theta}^T\left[\frac{1}{C}\mathbf{e}^T\mathbf{P}\,\mathbf{B}_c\left(\boldsymbol{\xi}_\Theta - e^{-\frac{1}{RC}t}\boldsymbol{\xi}_\Theta^0\right) + \frac{1}{\beta_\Theta}\dot{\tilde{\Theta}}\right]. \tag{3-34}$$

Substituting (3-28) into (3-32), we have

$$\dot{V} = \frac{1}{2}\mathbf{e}^T(\mathbf{A}_{c_c}^T\mathbf{P} + \mathbf{P}\mathbf{A}_c)\mathbf{e} - \frac{1}{2\delta}\frac{g}{g_L}(\mathbf{e}^T\mathbf{P}\,\mathbf{B}_c)(\mathbf{B}_c\mathbf{P}\mathbf{e}) + g\mathbf{e}^T\mathbf{P}\,\mathbf{B}_c\varepsilon + V_W + V_\Theta\,. \tag{3-35}$$

Due to the facts $\delta > 0$ and $g / g_L \geq 1$, we can rewrite (3-35) as

$$\dot{V} \leq \frac{1}{2}\mathbf{e}^T(\mathbf{A}_{c_c}^T\mathbf{P} + \mathbf{P}\mathbf{A}_c)\mathbf{e} - \frac{1}{2\delta}(\mathbf{e}^T\mathbf{P}\,\mathbf{B}_c)(\mathbf{B}_c\mathbf{P}\mathbf{e}) + g\mathbf{e}^T\mathbf{P}\,\mathbf{B}_c\varepsilon + V_W + V_\Theta$$

$$= \frac{1}{2}\mathbf{e}^T(\mathbf{A}_{c_c}^T\mathbf{P} + \mathbf{P}\mathbf{A}_c - \frac{1}{\delta}\mathbf{P}\mathbf{B}_c\mathbf{B}_c{}^T\mathbf{P})\mathbf{e} + g\mathbf{e}^T\mathbf{P}\,\mathbf{B}_c\varepsilon + V_W + V_\Theta \tag{3-36}$$

By using the Riccati-like equation (3-25), (3-36) can be rewritten as

$$\dot{V} = \frac{1}{2}\mathbf{e}^T(-\mathbf{Q} - \frac{1}{\rho^2}\mathbf{P}\mathbf{B}_c\mathbf{B}_c{}^T\mathbf{P})\mathbf{e} + g\mathbf{e}^T\mathbf{P}\,\mathbf{B}_c\varepsilon + V_W + V_\Theta$$

$$= -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} - \frac{1}{2}\left[\frac{1}{\rho}\mathbf{B}_c^T\mathbf{P}\mathbf{e} - g\rho\varepsilon\right]^2 + \frac{1}{2}g^2\rho^2\varepsilon^2 + V_W + V_\Theta \tag{3-37}$$

Using (3-23), we have

$$V_W = \begin{cases} 0 & \text{if}\left(\left|\hat{W}\right| < M_W\right)\text{or}\left(\left|\hat{W}\right| = M_W \text{ and } \mathbf{e}^T\mathbf{P}\mathbf{B}_c\hat{W}\left(\boldsymbol{\xi}_W - e^{-\frac{1}{RC}t}\boldsymbol{\xi}_W^0\right) \geq 0\right) \\ -\dfrac{g}{C}\mathbf{e}^T\mathbf{P}\mathbf{B}_c\dfrac{\hat{W}^T\left(\boldsymbol{\xi}_W - e^{-\frac{1}{RC}t}\boldsymbol{\xi}_W^0\right)}{\left\|\hat{W}\right\|^2}\tilde{W}\hat{W} & \text{if}\left(\left|\hat{W}\right| = M_W \text{ and } \mathbf{e}^T\mathbf{P}\mathbf{B}_c\hat{W}\left(\boldsymbol{\xi}_W - e^{-\frac{1}{RC}t}\boldsymbol{\xi}_W^0\right) < 0\right) \end{cases}$$

(3-38)

For the condition $\left(\left|\hat{W}\right| = M_W \text{ and } \mathbf{e}^T\mathbf{PB}_c\hat{W}\left(\xi_W - e^{-\frac{1}{RC}t}\xi_W^0\right) < 0\right)$, we have $\left|\hat{W}\right| = M_W \ge \left|W^*\right|$ because W^* belongs to the constraint set $\mathbf{\Omega}_W$. Using this fact, we obtain $\tilde{W}\hat{W} = \frac{1}{2}(W^{*2} - \hat{W}^2 - \tilde{W}^2) \le 0$. Thus, the second line of (3-38) can be rewritten as

$$V_W = -\frac{g}{2C}\mathbf{e}^T\mathbf{Pb}\frac{\hat{W}^T\left(\xi_W - e^{-\frac{1}{RC}t}\xi_W^0\right)}{\left\|\hat{W}\right\|^2}(W^{*2} - \hat{W}^2 - \tilde{W}^2) \le 0. \tag{3-39}$$

Similarly, we obtain

$$V_\Theta = \begin{cases} 0 & \text{if } \left(\left\|\hat{\Theta}\right\| < M_\Theta\right) \text{or} \left(\left\|\hat{\Theta}\right\| = M_\Theta \text{ and } \mathbf{e}^T\mathbf{PB}_c\hat{\Theta}^T\left(\xi_\Theta - e^{-\frac{1}{RC}t}\xi_\Theta^0\right) \ge 0\right) \\ -\frac{g}{C}\mathbf{e}^T\mathbf{PB}_c\frac{\hat{\Theta}^T\left(\xi_\Theta - e^{-\frac{1}{RC}t}\xi_\Theta^0\right)}{\left\|\hat{\Theta}\right\|^2}\tilde{\Theta}^T\hat{\Theta} & \text{if } \left(\left\|\hat{\Theta}\right\| = M_\Theta \text{ and } \mathbf{e}^T\mathbf{PB}_c\hat{\Theta}^T\left(\xi_\Theta - e^{-\frac{1}{RC}t}\xi_\Theta^0\right) < 0\right) \end{cases}$$

(3-40)

and the second line of (3-40) can be rewritten as

$$V_\Theta = -\frac{g}{2C}\mathbf{e}^T\mathbf{Pb}\frac{\hat{\Theta}^T\left(\xi_\Theta - e^{-\frac{1}{RC}t}\xi_\Theta^0\right)}{\left\|\hat{\Theta}\right\|^2}\left(\left\|\Theta^*\right\|^2 - \left\|\hat{\Theta}\right\|^2 - \left\|\tilde{\Theta}\right\|^2\right) \le 0 \tag{3-41}$$

Using the knowledge that $V_W \le 0$ and $V_\Theta \le 0$, we can further rewrite (3-37) as

$$\dot{V} \le -\frac{1}{2}\mathbf{e}^T\mathbf{Qe} + \frac{1}{2}g^2\rho^2\varepsilon^2 \tag{3-42}$$

Integrating both sides of the inequality (3-43) yields

$$V(t) - V(0) \le -\frac{1}{2}\int_0^t \mathbf{e}^T\mathbf{Qe}\,d\tau + \frac{g^2\rho^2}{2}\int_0^t \varepsilon^2\,dt \tag{3-43}$$

for $0 \le t < \infty$. Since $V(t) \ge 0$, we obtain

$$\frac{1}{2}\int_0^t \mathbf{e}^T\mathbf{Qe}\,d\tau \le V(0) + \frac{g^2\rho^2}{2}\int_0^T \varepsilon^2\,dt. \tag{3-44}$$

Substituting (3-31) into (3-44), we can prove (3-29).

ii) From (3-44) and since $\int_0^t \mathbf{e}^T\mathbf{Qe}\,dt \ge 0$, we have

$$2V(t) \le 2V(0) + g^2\rho^2\mu, \qquad 0 \le t < \infty \tag{3-45}$$

From (3-31), it is obvious that $\mathbf{e}^T\mathbf{P}\mathbf{e} \le 2V$, for any V. Because $\mathbf{P}$ is a positive definite symmetric matrix, we have

$$\lambda_{\min}(\mathbf{P})\|\mathbf{e}\|^2 = \lambda_{\min}(\mathbf{P})\mathbf{e}^T\mathbf{e} \le \mathbf{e}^T\mathbf{P}\mathbf{e} \tag{3-46}$$

Thus, we obtain

$$\lambda_{\min}(\mathbf{P})\|\mathbf{e}\|^2 \le \mathbf{e}^T\mathbf{P}\mathbf{e} \le 2V(t) \le 2V(0) + g^2\rho^2\mu \tag{3-47}$$

from (3-45)-(3-46). Therefore, from (3-47), we can easily obtain (3-30), which explicitly describe the bound of tracking error $\|\mathbf{e}\|$. If initial state $V(0)=0$, tracking error $\|\mathbf{e}\|$ can be made arbitrarily small by choosing adequate ρ. Equation (3-30) is very crucial to show that the proposed DACHDNN will provide the closed-loop stability rigorously in the Lyapunov sense under the Assumption (3-22). **Q.E.D.**

Remark: Equation (3-30) shows the relations among $\|\mathbf{e}\|$, ρ, and $\lambda_{\min}(\mathbf{P})$. For more insight of (3-30), we first choose $\rho^2 = \delta$ in (3-25) to simplify the analysis. Thus, from (3-25), we can see that $\lambda_{\min}(\mathbf{P})$ is fully affected by the choice of $\lambda_{\min}(\mathbf{Q})$ in the way that a larger $\lambda_{\min}(\mathbf{Q})$ leads to a larger $\lambda_{\min}(\mathbf{P})$, and vice versa. Now, one can easily observe form (3-30) that the norm of tracking error can be attenuated to any desired small level by choosing ρ and $\lambda_{\min}(\mathbf{Q})$ as small as possible. However, this may lead to a large control signal which is usually undesirable in practical systems.

3. 4 Simulation Results

In this section, two examples are presented to illustrate the effectiveness of the proposed DACHDNN. It should be emphasized that the development of the DACHDNN does not need to know the exact dynamics of the controlled system.

Example 3-1: Chaotic dynamic systems are known for their complex, unpredictable behavior and extreme sensitivity to initial conditions as well as parameter variations. Consider a second-order chaotic dynamic system, the well known Duffing's equation, which describes a special nonlinear circuit or a pendulum moving in a viscous medium under control [65]:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -p\dot{x} - p_1 x - p_2 x^3 + q\cos(wt) + u \\ y &= x_1 \end{aligned} \tag{3-48}$$

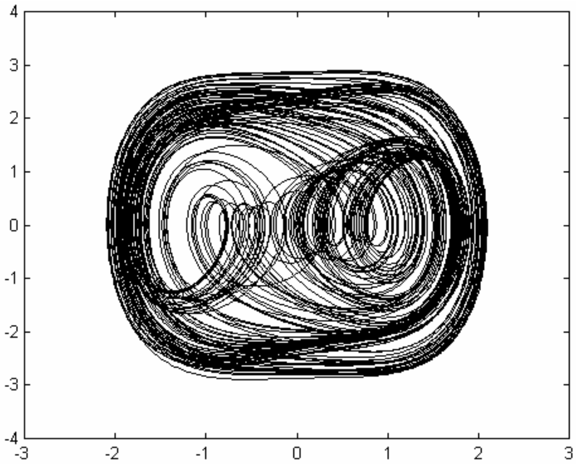
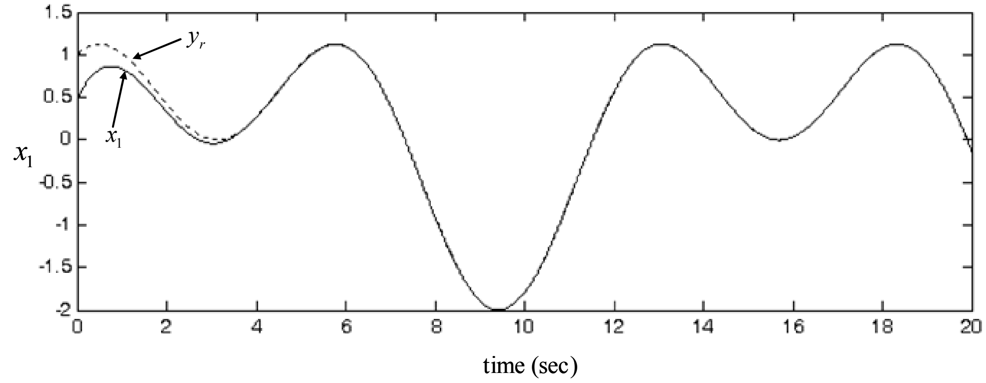
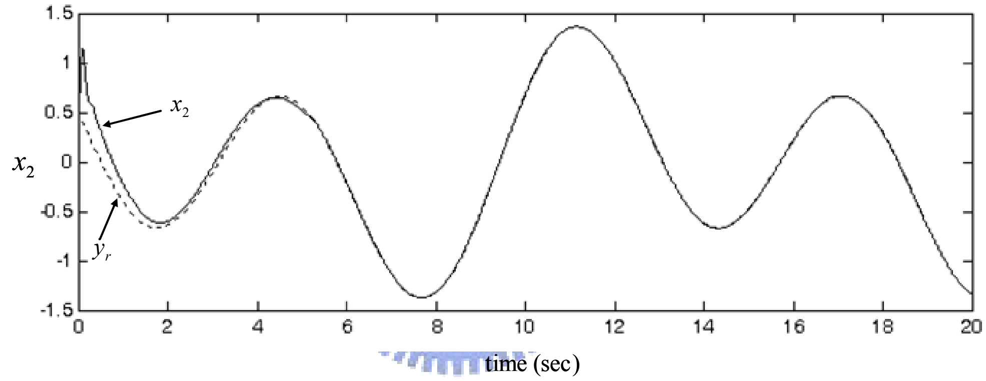


Fig. 3-4 The Phase plane of uncontrolled chaotic system

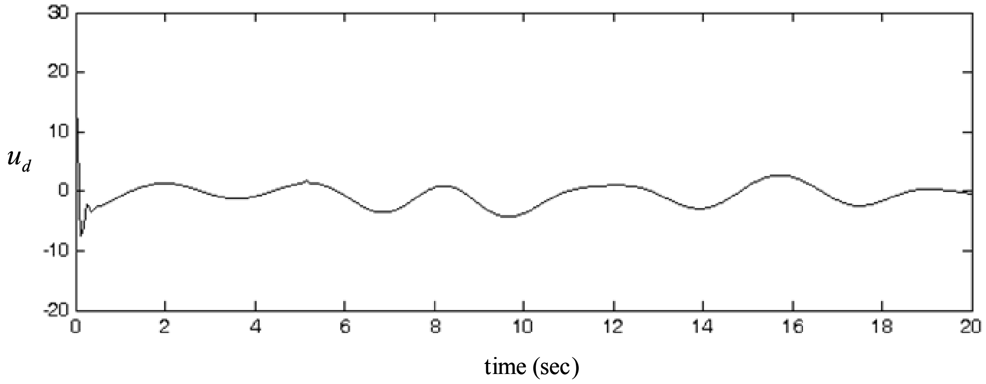
where p, p_1, p_2, q and w are real constants. Depending on the choices of these constants, the solutions of system (3-49) may display complex phenomena, including various periodic orbits behaviors and some chaotic behaviors [66]. Fig. 3-4 shows the complex open-loop system behaviors simulated with $u = 0$, $p = 0.4$, $p_1 = -1.1$, $p_2 = 1.0$, $w = 1.8$, $q = 1.95$, and $[x_1 \ x_2]^T = [0 \ 0]^T$. Assume the system is free of external disturbance in this example. The reference signal is $y_r(t) = \sin(0.5t) + \cos(t)$.Some initial parameter settings of DACHDNN are chosen as $\left[x_1^{\ 0} \ x_2^{\ 0}\right]^T = [0.5 \ 0]^T$, $u_{HNN,0} = 0$, $\xi_W^0 = 0$, $\xi_\Theta^0 = [0 \ 0\ ]^T$,$\hat{W}^0 = 0$, and $\hat{\Theta}^0 = [1 \ 1\ ]^T$. These initial settings are chosen through some trials to achieve favorable transient control performance. The learning rates of weights adaption are selected as $\beta_W = \beta_\Theta = 7.5$; the slope of $\tanh(\cdot)$ at the origin are selected as $\kappa = 1$; $g_L = 0.1$ and $\delta = 0.5$ for the compensation controller. The resistance and capacitance are chosen as $R = 5\Omega$ and $C = 0.005F$. Solving the Riccati-like equation (3-25) for a choice of $\mathbf{Q} = 10\mathbf{I}$, $\mathbf{k}_c = [2 \ 1]^T$, we have $\mathbf{P} = \begin{bmatrix} 15 & 5 \\ 5 & 5 \end{bmatrix}$. The simulation results for are shown in Figs. 3-5, where the tracking responses of state x_1 and x_2 are shown in Figs. 3-5(a) and 3-5(b), respectively, the associated control inputs are shown Fig. 3-5(c), and the trained weightings are shown in Fig. 3-5(d). From the simulation results, we can see that the proposed DACHDNN can achieve favorable tracking performances without external disturbance.


1.5
1
0.5
0
-0.5
-1
-1.5
-2
x_1
y_r
x_1
0
2
4
6
8
10
12
14
16
18
20
time (sec)

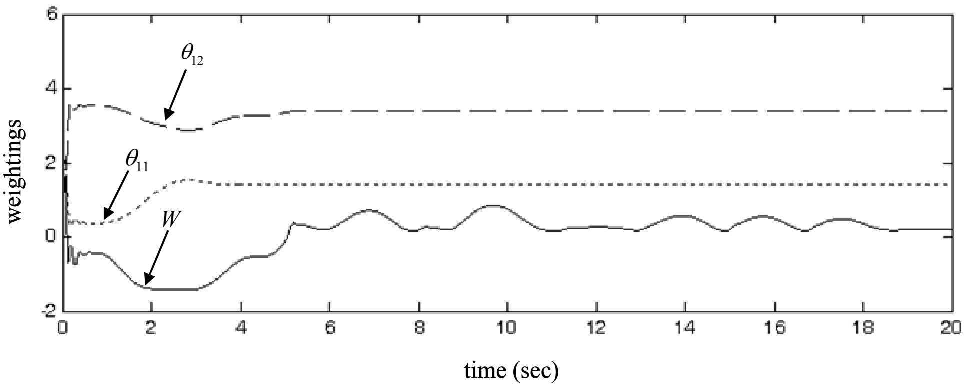
(a)


1.5
1
0.5
0
-0.5
-1
-1.5
x_2
x_2
y_r
0
2
4
6
8
10
12
14
16
18
20
time (sec)

(b)


30
20
10
0
-10
-20
u_d
0
2
4
6
8
10
12
14
16
18
20
time (sec)

(c)



(d)

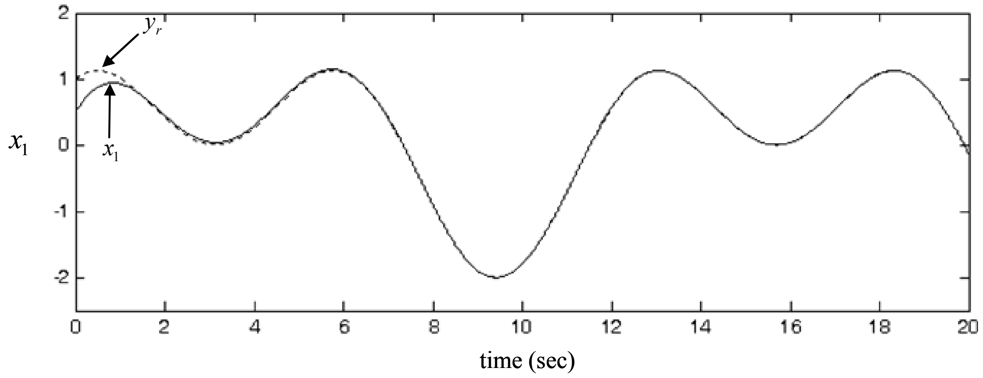
Fig. 3-5 Simulation results of Example 3-1

Example 3-2:

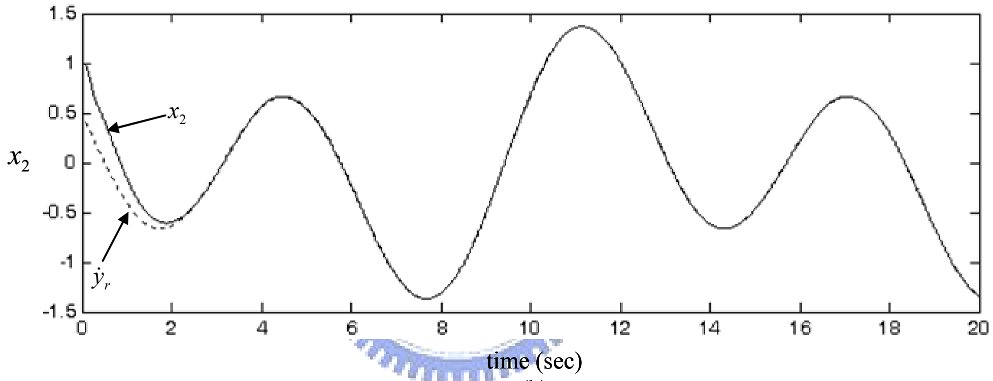
Consider the following nonlinear dynamic system described as [58, 67]

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= x_3\\ \dot{x}_3 &= x_1x_2 + u + d\\ y &= x_1\end{aligned} \tag{3-49}$$

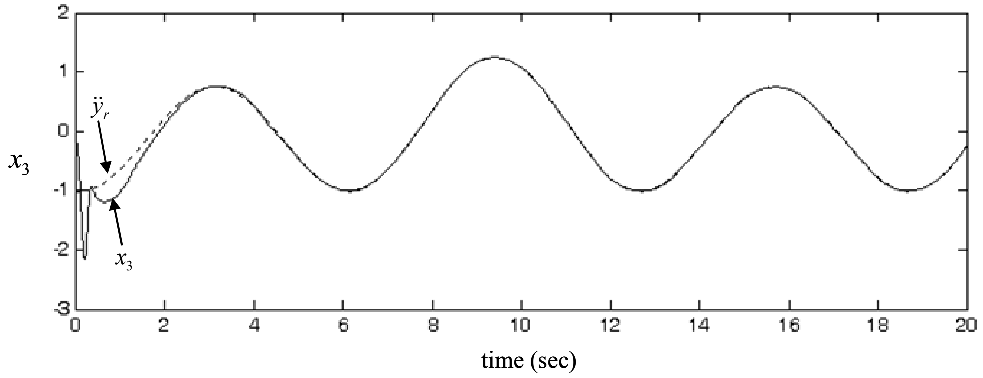
where $d = 0.5\sin(t)$ is the external bounded disturbance which occurs at $t \geq 10$. The reference signal is $y_r(t) = \cos(t) + \sin(0.5t)$. Some initial parameter settings of DACHDNN are chosen as $[x_1(0)\ x_2(0)\ x_3(0)]^T = [0.5\ 1\ 0]^T$, $u_{HDNN}^0 = 0$, $\xi_W^0 = 0$, $\xi_\Theta^0 = [0\ 0\ 0]^T$, $\hat{W}^0 = 0$, and $\hat{\Theta}_0 = [0\ 0\ 0]^T$. These initial settings are chosen through some trials to achieve favorable transient control performance. Other parameter settings are $\beta_W = \beta_\Theta = 25$; $\kappa = 1$; $g_L = 0.1$, $\delta = 0.5$, $R = 5\Omega$, and $C = 0.01F$. Solving the Riccati-like equation (3-25) for or a choice of $\mathbf{Q} = 3\mathbf{I}$, $\mathbf{k}_c = [3\ 7\ 5]^T$, we have $\mathbf{P} = \begin{bmatrix} 5.7969 & 3.8594 & 0.5 \\ 3.8594 & 6.5 & 0.7656 \\ 0.5 & 0.7656 & 0.4531 \end{bmatrix}$. The simulation results for are shown in Figs. 3-6, where the tracking responses of state x_1, x_2 and x_3 are shown in Figs. 3-6(a), 3-6(b), and 3-6(c), respectively, the associated control inputs are shown in Fig. 3-6(d), and the trained weightings are shown in Fig. 3-6(e). From Fig. 3-6(a), we can observe that the output of the system well tracks the reference signal throughout the whole control process, even with the external disturbance occurring in the middle time ($t \geq 10$). This fact shows the strong disturbance-tolerance ability of the proposed system.


y_r
x_1
x_1
2
1
0
-1
-2
0
2
4
6
8
10
12
14
16
18
20
time (sec)

(a)


x_2
x_2
$\dot{y}_r$
1.5
1
0.5
0
-0.5
-1
-1.5
0
2
4
6
8
10
12
14
16
18
20
time (sec)

(b)


x_3
$\ddot{y}_r$
x_3
2
1
0
-1
-2
-3
0
2
4
6
8
10
12
14
16
18
20
time (sec)

(c)

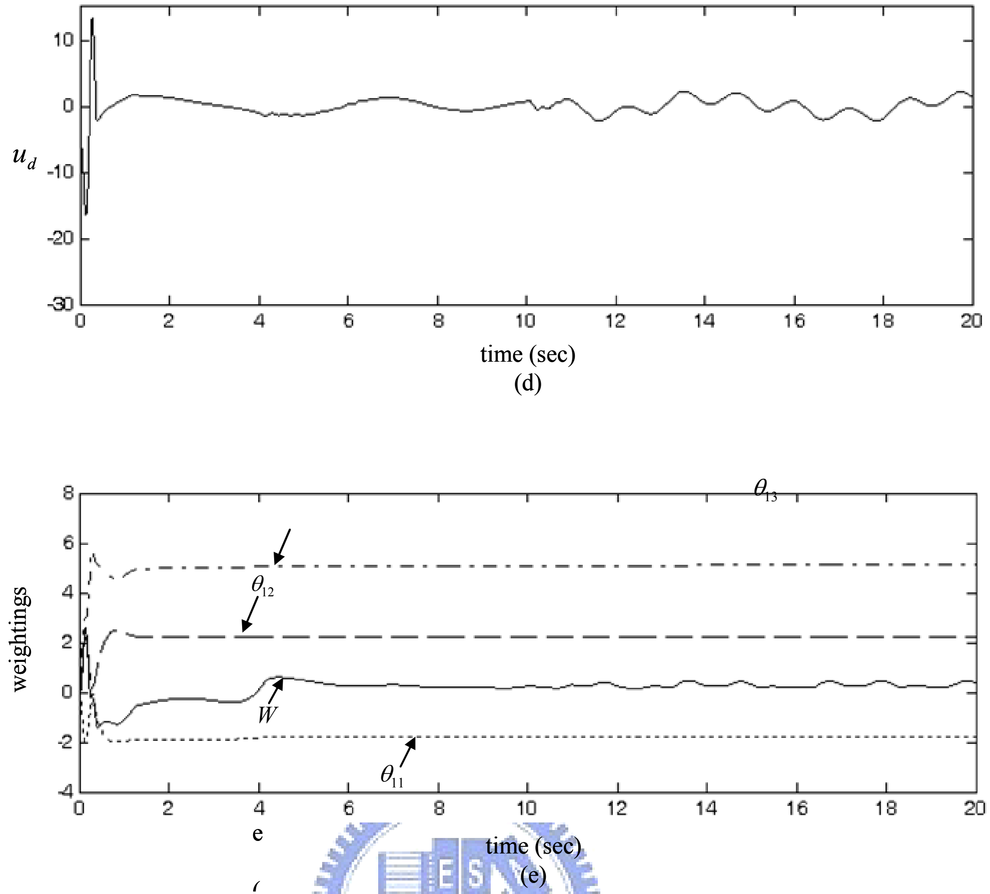


Fig. 3-6 Simulation results of Example 3-2

3.5 Performance analysis of Hopfield-based DNNs with and without the self-feedback loop

The performance of Hopfield-based DNNs with and without the self-feedback loop will be compared in this section. Hopfield networks are sometimes composed of neurons without self-feedback loops in some applications, such as pattern recognition [68]. This is to minimize the number of potential stable states so as to increase the recognition rate [68]. However, is it true that a Hopfield-based DNN composed of neurons without self-feedback loops performs better in the control problem of SISO affine nonlinear systems? We will try to answer this question by the following discussions and simulation results.

Because the proposed Hopfield-based DNN contains only a single neuron for SISO affine nonlinear systems, we can simply set $W = 0$ (and hence $W^* = \hat{W} = \tilde{W} = 0$) when a neuron without self-feedback loop is used. Thus, repeating the discussions with $W = 0$ in sections 3.1 and 3.3, we have the following theorem:

Theorem 3-2: Suppose the required assumption holds. Consider the plant (3-14) with the control law (3-18), where the Hopfield-based DNN controller u_{HDNN} is given as

$$u_{HDNN} = \frac{1}{C}\hat{\Theta}^T\xi_\Theta + e^{-\frac{1}{RC}t}u_{HDNN}^0 - e^{-\frac{1}{RC}t}\frac{1}{C}\hat{\Theta}^T\xi_\Theta^0 . \tag{3-50}$$

with the adaptive law (3-24). The compensation controller u_s is given as (3-28). Then, the overall control schemes guarantees that

i) $$\frac{1}{2}\int_0^t \mathbf{e}^T\mathbf{Q}\mathbf{e}\,d\tau \le \frac{1}{2}\mathbf{e}_0{}^T\mathbf{P}\mathbf{e}_0 + \frac{\tilde{\Theta}_0^T\dot{\tilde{\Theta}}_0}{2\beta_\Theta}\Theta + \frac{g^2\rho^2}{2}\int_0^t \varepsilon^2 d\tau \tag{3-51}$$

for $0 \le t < \infty$.

ii) The tracking error $\|\mathbf{e}\|$ can be expressed in terms of the lumped uncertainty as

$$\|\mathbf{e}\| \le \sqrt{\frac{2V^0 + g^2\rho^2\mu}{\lambda_{\min}(\mathbf{P})}} . \tag{3-30}$$

Proof: **Theorem 3-2** can be easily proven by following the proof of **Theorem 3-1** under the premise that $W = W^* = \hat{W} = \tilde{W} = 0$. **Q.E.D.**

From **Theorem 3-2**, we ascertain that the convergence performance of the Hopfield-based DNN without the self-feedback loop can still be guaranteed.

Next, simulations for the Hopfield-based DNN without the self-feedback loop are performed. For Example 3-1, the tracking responses of state x_1 and x_2 are shown in Figs. 3-7(a) and 3-7(b), respectively. The norms of error vectors, $\|\mathbf{e}\|$, for the cases of Hopfield-based DNN with and without the self-feedback loop are shown in Fig. 3-8(c). For example 3-2, the tracking responses of state x_1, x_2, and x_3 are shown in Figs. 3-8(a), 3-8(b), and 3-8(c), respectively; Fig. 3-8(d) shows the $\|\mathbf{e}\|$ for both cases. From the simulation results, we can see that as we expect, a Hopfield-based DNN without the self-feedback loop can also result in acceptable tracking performance. However, form Figs. 3-8(c) and 3-8(d), it can be easily observed that a Hopfield-based DNN with the self-feedback loop perform better in the tracking control problem of SISO nonlinear systems. This fact is totally opposite to the knowledge that a Hopfield network without the self-feedback loop can be used to increase recognition rate in pattern recognition. [68].

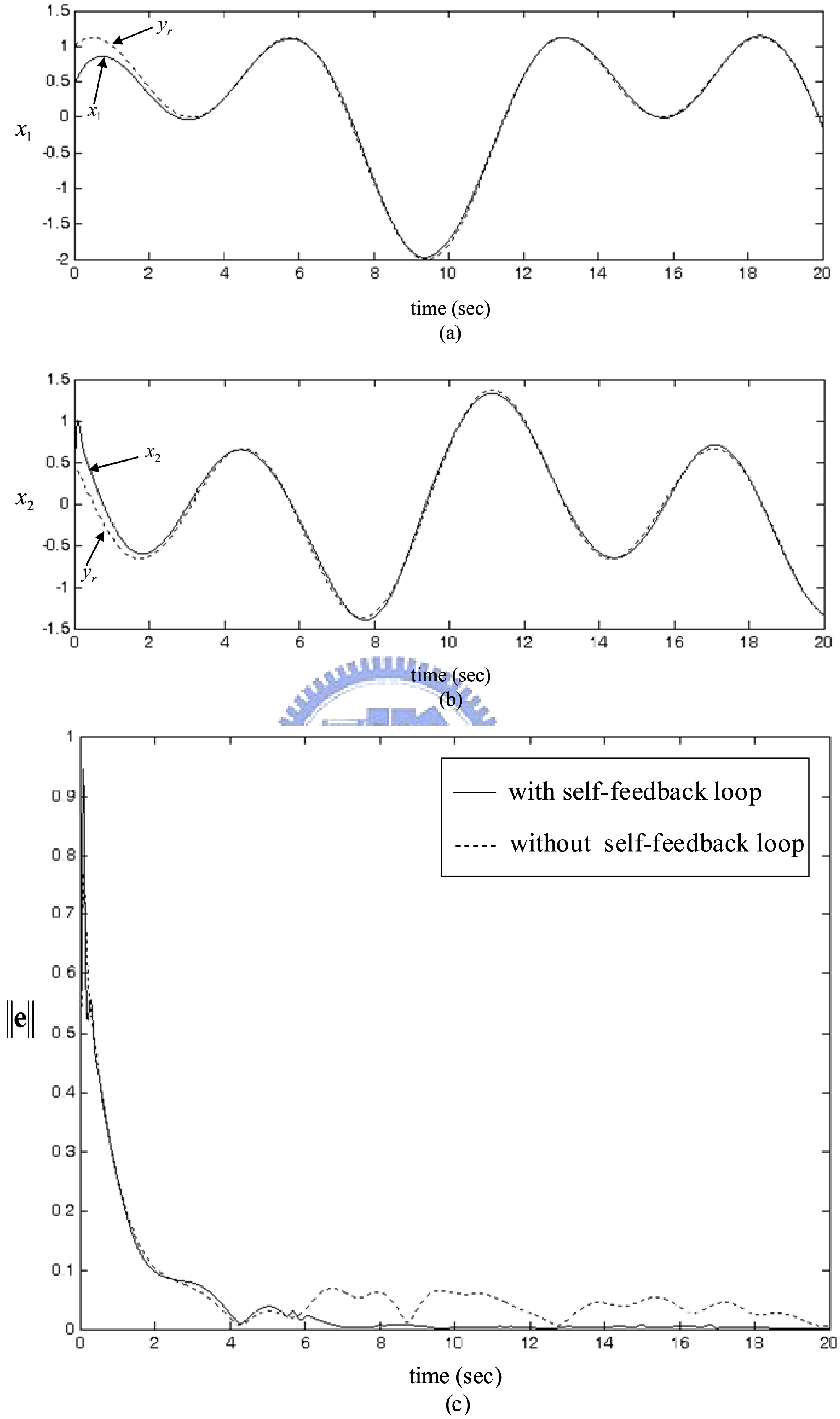
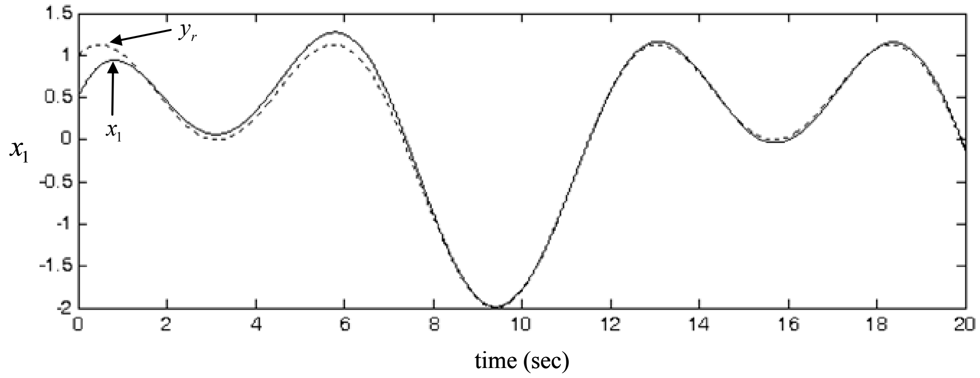
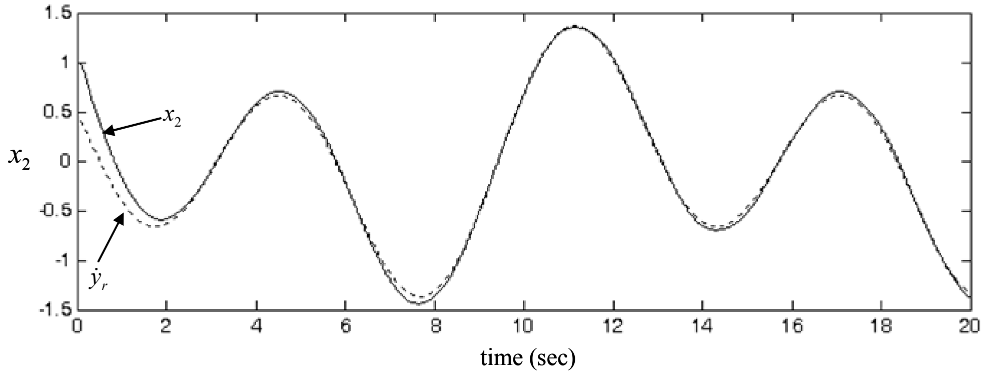


Fig. 3-7 Simulation results of Example 3-1 using Hopfield-based DNN without the feedback loop



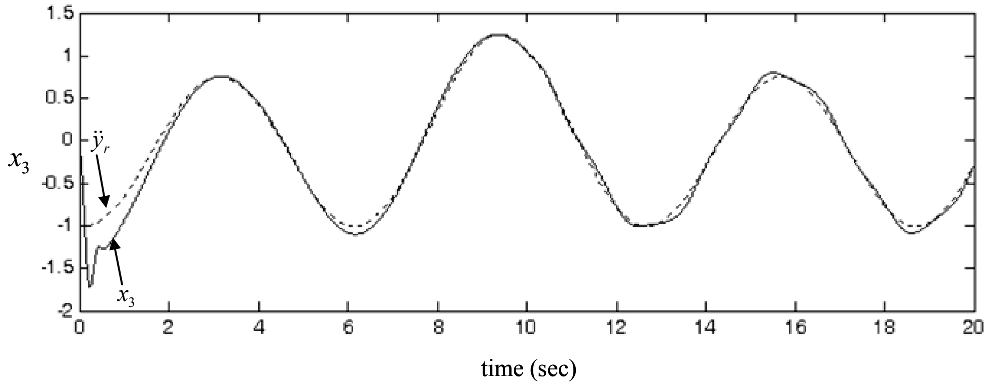
y_r
x_1
x_1
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
2
4
6
8
10
12
14
16
18
20
time (sec)

(a)



x_2
$\dot{y}_r$
x_2
1.5
1
0.5
0
-0.5
-1
-1.5
0
2
4
6
8
10
12
14
16
18
20
time (sec)

(b)



$\ddot{y}_r$
x_3
x_3
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
2
4
6
8
10
12
14
16
18
20
time (sec)

(c)

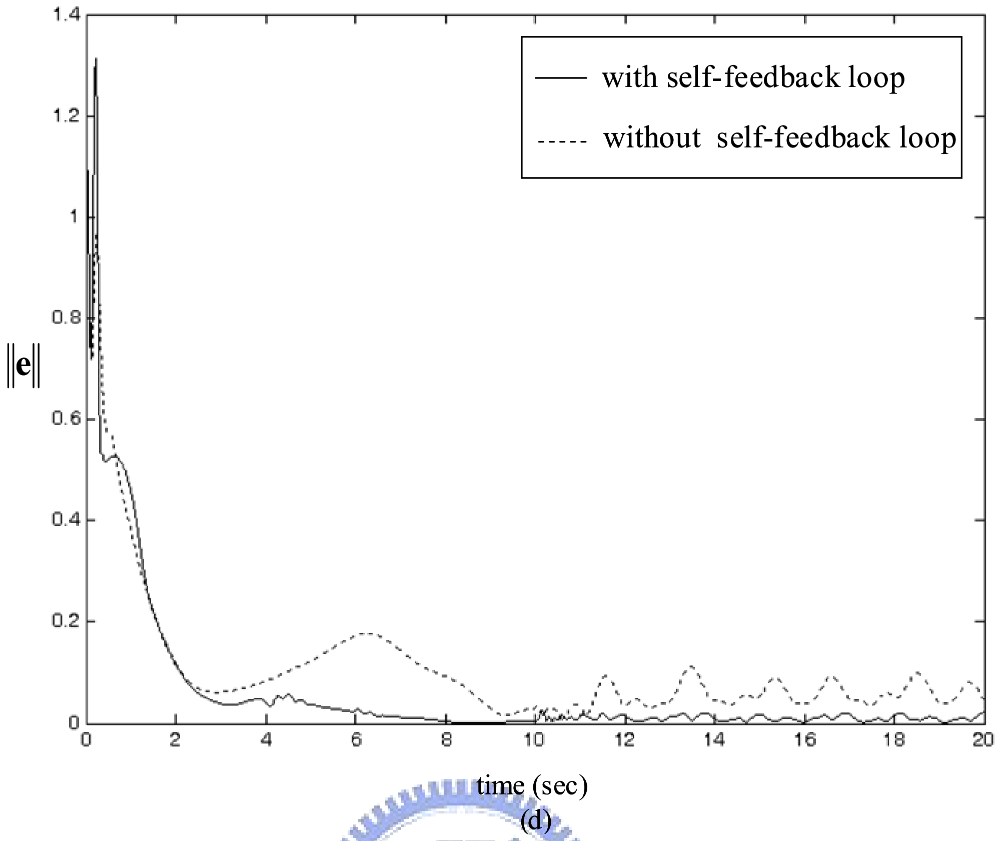


Fig. 3-8 Simulation results of Example 3-2 using Hopfield-based DNN without the feedback loop

Chapter 4

Conclusions and Future Works

For decades, many researchers and designers, from such broad areas as aircraft and spacecraft control, robotics, process control, and biomedical engineering, have shown an active interest in the control problem of nonlinear systems. Among these research efforts, adaptive fuzzy control and adaptive NN control have been shown to be powerful and effective methodologies for nonlinear control. However, in the control design, the structure determination is a difficult task for both FSs and NNs. More specifically, choosing the number of fuzzy rules, inherently involving fuzzy partitioning of input and output spaces, can greatly affect the approximation capability of fuzzy systems; similarly, the number of neurons can be a decisive factor to the performance of NNs.

In Chapter 2, the proposed self-structuring fuzzy system (SFS) can construct a compact fuzzy rule base by automatic rule generation and pruning. The problems of determining the fuzzy partitions of input spaces and the number of fuzzy rules are solved simultaneously. The provided systematic method can cope with the tradeoff between the approximation accuracy and computational load of FS. New rules are generated according to the newly added membership functions to adjust the improper fuzzy clustering of the input spaces. Insignificant rules with negligible contribution toward the output of FS will be removed after a short period. Further, a robust adaptive self-structuring fuzzy control (RASFC) scheme for the uncertain or ill-defined nonlinear nonaffine systems is proposed. Some adaptive laws for on-line tuning the parameters of fuzzy rules are derived in the Lyapunov sense to realize favorable fuzzy approximation. As shown in Chapter 2, the RASFC can achieve a L_2 tracking performance with arbitrarily attenuation level. This L_2 tracking performance can provide a clear expression of tracking error in terms of the sum of lumped uncertainty and external disturbance, which has not been shown in previous works. Several examples are illustrated to show that the RASFC can achieve favorable tracking performance in the presence of external disturbance, yet heavy computational burden is relieved.

In Chapter 3, we propose a direct adaptive control scheme using Hopfield-based dynamic neural networks for SISO nonlinear systems. A simple Hopfield-based DNN is used to approximate the ideal controller and the synaptic weights Hopfield-based DNN are on-line tuned by adaptive laws. A compensation controller is merged into control law to suppress the effect of modeling error and external disturbance. By Lyapunov stability analysis, we prove that the closed-loop system is stable, and the tracking error can be attenuated to a desired level. Note that no strong assumptions and prior knowledge of the controlled plant are needed in the development of DACHDNN. Simulation results demonstrate the effectiveness and robustness of the proposed DACHDNN in the presence of external disturbance. The case of Hopfield-based neural network without the self-feedback loop is also studied. We show that this case has inferior results than those of Hopfield neural network with the self-feedback loop. The most important is, for SISO affine nonlinear systems, we propose an adaptive control scheme which results in a Hopfield-based DNN containing only one neuron but still maintain good tracking performance. The parsimonious structure of the Hopfield-based DNN solve the structuring problem of NNs, and the simple Hopfield circuit makes the DACHDNN much easier to implement and more reliable in practical purposes.

Although we have basically solved the control problem of nonlinear systems by the fuzzy and NN control schemes with automatic structuring processes, some underlying details need to be examined to make the solutions more perfect and practical. The first is the universal approximation property of the SFS. It has been proven by many researchers that fuzzy systems can approximate any nonlinear function to any desired accuracy because of the universal approximation theorem. However, the validness of the universal approximation property for a fuzzy system with variable number of rules, such as the proposed SFS, is still left to be explored. Although the research results in our work and many other literatures have provided strong collateral evidences, a direct and rigorous proof of the universal approximation theorem for fuzzy systems with variable structure is indispensable. This will be one of our future works. The implementation of the proposed RASFC scheme in a real hardware platform is also a problem. Although the concepts of rule pruning and growing are quite intuitive and simple; however, it is not an easy task to realize them in hardware.

On the other hand, in Chapter 3, the parsimonious structure makes the proposed DACHDNN scheme has the best chance to be realized in hardware for real world applications. However, the proposed DACHDNN scheme Chapter 3 is now only applicable to SISO nonlinear systems. In the future, we will work on extending the research results to the MIMO nonlinear systems.

Reference

[1] A. Isidori, *Nonlinear Control System* (2nd ed.), Berlin, Germany: Springer-Verlag, 1989.

[2] M. Krstic, I. Kanellakopoulos, and P.V. Kokotovic, *Nonlinear and Adaptive Control Design*, New York: Wiley, 1995.

[3] L. X. Wang, *Adaptive Fuzzy Systems and Control - Design and Stability Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey, 1994.

[4] F. L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*, London, U.K.: Taylor and Francis, 1999.

[5] Y. G. Leu, T. T. Lee, and W. Y. Wang , "Observer-Based Adaptive Fuzzy-Neural Control for Unknown Nonlinear Dynamical System," *IEEE Trans Syst Man Cybern B* Vol. 29, No. 5, pp. 583-591, 1999.

[6] S. S. Ge, C. Wang, "Adaptive NN control of uncertain nonlinear pure-feedback systems," *Automatica*, Vol. 38, pp. 671–682, 2002.

[7] K. Tanaka, T. Ikeda, and H. O. Wang, "Robust stabilization of a class of uncertain nonlinear systems via fuzzy control: quadratic stability, H^{∞} control theory, and linear matrix inequalities," *IEEE Trans. Fuzzy Syst.*, Vol. 4, pp. 1–13, Feb. 1996.

[8] H. K. Lam, F. H. F. Leung, and P. K. S. Tam, "Nonlinear State Feedback Controller for Nonlinear Systems: Stability Analysis and Design Based on Fuzzy Plant Model," *IEEE Trans. Fuzzy Syst.*, Vol.9, pp. 657-661, 2001.

[9] C. H. Wang, T. C. Lin, T. T. Lee, and H. L. Liu, "Adaptive hybrid intelligent control for uncertain nonlinear dynamical systems," *IEEE Trans Syst Man Cybern B*, Vol. 32, No. 5, pp. 583-597, 2002.

[10] C. F. Hsu, G. M. Chen, and T. T. Lee, "Robust intelligent tracking control with PID-type learning algorithm," *Neurocomputing*, Vol. 71, issue 1-3, pp. 234-243, 2007.

[11] Y. G. Leu, W. Y. Wang, and T. T. Lee, "Robust adaptive fuzzy-neural controllers for uncertain nonlinear systems," *IEEE Trans.* Robotics and Automation, Vol. 15, pp. 805-817, 1999.

[12] Chun-Fei Hsu, Chih-Min Lin, and Kuo-Hsiang Cheng, "Supervisory intelligent control system design for forward DC-DC converters," *IEE Proc. Electric Power Applications*, Vol. 153, No. 5, pp. 691-701, 2006.

[13] G. M. Mahmoud and A. A. Farghaly, "Chaos control of chaotic limit cycles of a real

complex van der Pol oscillators," *Chaos Solitons Fractals*, Vol. 21, pp. 915-924, 2004.

[14] A. L. Pourhiet, M. Correge, and D. Caruana, "Control of self-oscillating systems," *IEE Proc. Control Theory*, Vol. 150, No. 6, pp. 599-610, 2004.

[15] H. H. Wang and M. Krstic, "Extremum seeking for limit cycle minimization," *IEEE Trans Automatic control*, Vol. 45, No. 12, pp. 2432-2437, 2000.

[16] H. M. Gutierrez and P. I. Ro, "Magnetic servo levitation by sliding-mode of nonaffine systems control with algebraic input invertibility", *IEEE Trans Industrial Electronics*, Vol. 52, No. 5, pp. 1449-1455, 2005.

[17] L. R. Hunt and, G. Meyer, "Stable inversion for nonlinear systems," *Automatica*, Vol. 33, pp. 1549–1554, 1997.

[18] M. Krstic, I. Kanellakopoulos, and P.V. Kokotovic, "Adaptive nonlinear control without overparameterization," *Systems and Control Letters*, Vol. 19, pp. 177–185, 1992.

[19] T. Terano, K. Asai, and M. Sugeno, *Fuzzy Systems Theory and Its Applications*, Academic Press, Boston, 1992.

[20] A. M. Gil-Lafuente, *Fuzzy logic in financial analysis*, Springer, Berlin Heidelberg New York, 2005.

[21] J. L. Castro, "Fuzzy logic controllers are universal approximators," *IEEE Trans. Syst. Man Cybern.*, Vol. 25, No. 4, pp. 629-635, 1995.

[22] C. M. Lin and C. F. Hsu, "Guidance law design by adaptive fuzzy sliding-mode control," *Guidance, Control, and Dynamics*, Vol. 25, No. 2, pp. 248-256, 2002.

[23] H. X. Li and S. C. Tong "A Hybrid Adaptive Fuzzy Control for A Class of Nonlinear MIMO Systems," *IEEE Trans. Fuzzy Systems*, Vol. 11, No. 1, pp. 24-34, 2003.

[24] S. Labiod, M. S. Boucherit, and T. M. Guerra, "Adaptive fuzzy control of a class of MIMO nonlinear systems," *Fuzzy Sets Syst.*, Vol. 151, issue 1, pp. 59-77, 2005.

[25] C. F. Hsu and C. M. Lin, "Fuzzy-identification-based adaptive controller design via backstepping approach," *Fuzzy Sets Syst.*, Vol. 151, issue 1, pp. 43-57, 2005.

[26] A. Chatterjee and K. Watanabe, "An adaptive fuzzy control strategy for motion control of robot manipulators," *Soft Comput.* Vol. 9, pp.185-193, 2005.

[27] C. J. Lin and C. H. Chen, "A self-Constructing compensatory neural fuzzy system and its applications," *Mathematical and Computer Modelling*, Vol. 42, pp. 339-351, 2005.

[28] J. E. Meng and D. Chang, "Online tuning of fuzzy inference systems using dynamic fuzzy Q-learning," *IEEE Trans. Syst. Man Cybern. B*, Vol. 34, No. 3, pp. 1478-1489, 2004.

[29] F. J. Lin and C. H. Lin, "A permanent-magnet synchronous motor servo drive using

self-constructing fuzzy neural network controller,” *IEEE Trans. Energy Conversion*, Vol. 19, No. 1, pp. 66-72, 2004.

[30] J. J. Shann and H. C. Fu, “A fuzzy neural network for rule acquiring on fuzzy control systems,” *Fuzzy Sets Syst.*, Vol. 71, issue 3, pp. 345-357, 1995.

[31] N. R. Pal and T. Pal, “On rule pruning using fuzzy neural networks,” *Fuzzy Sets Syst.*, Vol. 106, issue 3, pp. 335-347, 1999.

[32] S. Haykin, *Nerual Neworks-A Comprehensive Foundatio*, New York: Macmillan, 1994.

[33] K. S. Narendra and K. Parthasarathy, “Identification and control for dynamic systems using neural networks, “ *IEEE Trans. Neural Networks*, Vol. 1, pp. 4-27, 1993.

[34] K. J. Hunt, D. Sbarbaro, and P. J. Gawthrop, “Neural networks for control systems—A survey,” *Automatica*, Vol. 28, pp. 1083-1112, 1992.

[35] C. Y. Lee and J. J. Lee, “Adaptive control for uncertain nonlinear systems based on multiple neural networks,” *IEEE Tran. Syst., Man, Cybern. B*, Vol. 34, pp. 325-333, 2004.

[36] T. W. S. Chow and Y. Fang, “A recurrent neural-network-based real-time learning control strategy applying to nonlinear systems with unknown dynamics,” *IEEE Trans. Ind. Electronics*, Vol. 45, pp. 151-161, 1998.

[37] K. Homik, M. Stinchcombe, “Multilayer feedforward networks are universal approximators, “ *Neural Network*, Vol. 2, pp. 359-366, 1989.

[38] K. Funahashi and Y. Nakamura, “Approximation of dynamical systems by continuous time recurrent neural networks,” *Neural Networks*, Vol. 6, pp. 801-806, 1993.

[39] B. S. Chen, C. H. Lee, and Y. C. Chang, “H^∞ Tracking Design of Uncertain Nonlinear SISO Systems: Adaptive Fuzzy Approach,” *IEEE Trans. Fuzzy Syst.*, Vol. 4, pp. 32-43, 1996.

[40] Y. Gao and M. J. Er, “Online adaptive fuzzy neural identification and control of a class of MIMO nonlinear systems,” *IEEE Trans. Fuzzy Syst.*, Vol. 11, pp. 462-477, 2003.

[41] S. Seshagiri and K. H. Kahalil, “Output feedback control of nonlinear systems using RBF neural networks,” *IEEE Trans. Neural Networks*, Vol. 11, pp. 69-79, 2000.

[42] Y. Li, S. Qiang, X. Zhang, and O. Kaynak, “Robust and adaptive backstepping control for nonlinear systems using RBF neural networks,” *IEEE Trans. Neural Networks*, Vol. 15, pp. 693-701, 2004.

[43] C. T. Lin and C. S. George Lee, *Neural fuzzy systems : a neuro-fuzzy synergism to intelligent systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1996.

[44] D. T. Pham and X. Liu, “Dynamic system identification using partially recurrent neural

networks," *J. Syst. Eng.*, Vol. 2, pp. 90-97, 1992.

[45] G. A. Rovithakis, and M. A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks," *IEEE Tran. Syst., Man, Cybern.*, Vol. 24, pp. 400-412, 1994.

[46] M. A. Brdys and G. J. Kulawski, "Dynamic Neural Controllers for Induction Motor," *IEEE Trans. Neural Networks*, Vol. 10, pp. 340-355, 1999.

[47] A. S. Poznyak, W. Yu, D. N. Sanchez, and J. P. Perez, "Nonlinear adaptive trajectory tracking using dynamic neural networks," *IEEE Trans. Neural Networks*, Vol. 10, pp. 1402-1411, 1999.

[48] T. W. S. Chow, X. D. Li, Y. Fang, "A real-time learning control approach for nonlinear continuous-time system using recurrent neural networks," *IEEE Trans. Ind. Electronics.*, Vol. 47, pp. 478-486, 2000.

[49] X. M. Ren, A. B. Rad, P. T. Chan, and W. L. Lo, "Identification and control of continuous-time nonlinear systems via dynamic neural networks," *IEEE Trans. Ind. Electronics.*, Vol. 50, pp. 478-486, 2003.

[50] S. Labiod and T. M. Guerra, "Adaptive fuzzy control of a class of SISO nonaffine nonlinear systems," *Fuzzy Sets Syst.*, Vol. 158, pp. 1126-1137, 2007.

[51] J. H. Park and S. H. Kim, "Direct adaptive self-structuring fuzzy controller for nonaffine nonlinear system," *IEE Proc. Control Theory*, Vol. 153, pp. 429-445, 2005.

[52] J. J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proc. National Academy of sciences, USA*, Vol. 79, pp. 2554-2558, 1982.

[53] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. National Academy of sciences, USA*, Vol. 81, pp. 3088-3092, 1984.

[54] N. Hovakimyan, F. Nardi, A. Calise, and N. Kim, "Adaptive Output Feedback Control of Uncertain Nonlinear Systems Using Single-Hidden-Layer Neural Networks," *IEEE Trans. Neural Networks* , Vol. 13, No. 6, pp. 1420-1431, 2002.

[55] A. Calise, N. Hovakimyan, and M. Idan, "Adaptive output feedback control of nonlinear systems using neural networks," *Automatica*, Vol. 37, issue 8, pp. 1201-1211, 2001.

[56] H. Han, C. Y. Su, and Y. Stepanenko, "Adaptive control of a class of nonlinear systems with nonlinearly parameterized fuzzy approximators," *IEEE Trans. Fuzzy Syst.*, Vol. 9, pp.315-323, 2001.

[57] C. F. Hsu, C. M. Lin, and T. Y. Chen, "Neural-network-identification-based adaptive

control of wing rock motion," *IEE Proc. Control Theory and Applications*, Vol. 152, No. 1, pp. 65-71, 2005.

[58] W. Y. Wang, M. L. Chan, C. C. Hsu, and T. T. Lee, "H_∞ tracking-based sliding mode control for uncertain nonlinear systems via an adaptive fuzzy-neural approach," *IEEE Trans. Syst Man Cybern B*, Vol. 32, No. 4, pp. 483-492, 2002.

[59] S. S. Ge and J. Zhang, "Neural-network of nonaffine nonlinear systems with zero dynamics by state and output feedback," *IEEE Trans. Neural Networks*, Vol. 14, No. 14, pp. 900-918, 2003.

[60] S. S. Ge, C. C. Hang, and T. Zhang, "Adaptive neural network control of nonlinear systems by state and output feedback," *IEEE Trans. Syst Man Cybern*, Vol. 129, No. 6, pp. 818-828, 1999.

[61] Y. G. Leu, W. Y. Wang, T. T. Lee, "Observer-Based Direct Adaptive Fuzzy-Neural Control for Nonaffine Nonlinear Systems,", *IEEE Trans. Neural Networks*, Vol. 16, No. 4, 853-861, 2005.

[62] B. Karimi, M. B. Menhaj, and I. Saboori, "Robust adaptive control of nonaffine nonlinear systems using radial basis function neural networks," *IEEE Proc. Conf Industrial Electronics*, pp. 495-500, 2006.

[63] A. S. Poznyak, W. Yu, E. N. Sanchez, and J. P. Perez, "Stability analysis of dynamic neural control, " *Expert Syst. Applicat.*, Vol. 14, pp. 227-236, 1998.

[64] 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, pp. 422-431, 1995.

[65] A. Loria, E. Panteley, and H. Nijmeijer, "Control of the chaotic Duffing equation with uncertainty in all parameter," *IEEE Trans. Circuits Syst.*, I, Vol. 45, pp. 1252-1255, 1998.

[66] G. Chen, X. Dong, "On Feedback Control of Chaotic Continuous-Time Systems," *IEEE Trans. Circuits Syst. I: Fundamental Theory and Applications,* Vol. 40, pp. 591-601, 1993.

[67] W. Perruquetti, T. Floquet, and P. Borne, "A note on sliding observer and controller for generalized canonical forms," *Proc. 37th IEEE Conf. Decision Control*, pp. 1920–1925, 1998.

[68] D. L. Prados, "Capacity of a neural network," *IEE Electronics Letters*, Vol. 24, No. 8, pp. 454-455, 1988.

Vita

Name: Pin-Cheng Chen

Personal Details:

Place of Birth: Kaohsiung, Taiwan, R.O.C.

Day of Birth: Sept. 20 1975

Gender: Male

Education background:

Degree	Date	School
B.S. E.E.	1994/9 - 1999/6	National Chiao Tung University, Department of Electrical and Control Engineering
M.S. E.E.	2001/9 - 2003/6	Fu-Jen Catholic University Department of Electronic Engineering
Ph.D. E.C.E.	2003/9 - 2008/10	National Chiao Tung University, Department of Electrical and Control Engineering

Advisor(s):

M.S. -

Professor Wei-Yen Wang, National Taiwan Normal University

Ph.D. -

Professor Tsu-Tian Lee, National Taipei University of Technology

Professor Chi-Hsu Wang, National Chiao Tung University

Publication List

Accepted Journal Papers:

[1] Pin-Cheng Chen (Prev. as Guan-Ming Chen), Wei-Yen Wang, Tsu-Tian Lee, and CW Tao, “Observer-Based Direct Adaptive Fuzzy-Neural Control for. Anti-lock Braking Systems,” *International Journal of Fuzzy System*, vol. 8, no. 4, pp. 208-218, 2006.

[2] Chun-Fei Hsu, Pin-Cheng Chen (Prev. as Guan-Ming Chen), Tsu-Tian Lee, “ Robust intelligent tracking control with PID-type learning algorithm,” *Neurocomputing*, vol. 71, issue 1-3, pp. 234-243, 2007.

[3] Pin-Cheng Chen (Prev. as Guan-Ming Chen), Ping-Zong Lin, Wei-Yen Wang, Tsu-Tian Lee, and Chi-Hsu Wan, **“**Image-based fuzzy control system**”** *IEEE Electronics Letters*, vol. 44, issue 7, pp. 461-462, 2008.

[4] Pin-Cheng Chen, Chun-Fei Hsu, Chi-Hsu Wang, and Tsu-Tian Lee, “Fuzzy-identification-based adaptive backstepping control using a self-organizing fuzzy system Robust Sliding Mode-Like Fuzzy Logic Control of Anti-Lock Braking System,” Accepted by *Soft Computing*, 2008.

Submitted Journal Papers:

[1] Pin-Cheng Chen, Chun-Fei Hsu, Chi-Hsu Wang, and Tsu-Tian Lee, “Design of Robust Adaptive Self-structuring Fuzzy Controller for Nonlinear Nonaffine Systems,” submitted to International Journal of System Science, Sept. 2007.

International Conference Papers:

[1] Wei-Yen Wang, Kou-Cheng Hsu, C. W. Tao, and Pin-Cheng Chen (Prev. as Guan-Ming Chen), “Stable Anti-Lock Brake System Using Output-Feedback Direct Adaptive Fuzzy-Neural Control” *Proc. IEEE Conf. System, Man, and Cybernetics*, vol. 4, pp. 3675-3680, 2003.

[2] Pin-Cheng Chen (Prev. as Guan-Ming Chen), Chun-Fei Hsu, and Tsu-Tian Lee, “Adaptive Fuzzy Control with PI Learning Algorithm for Induction Servomotor Systems,” *IEEE Conf. Fuzzy Systems*, pp. 530-535, 2005

[3] T. T. Lee, P. C. Chen (Prev. as G. M. Chen), C. F. Hsu and C. M. Lin, “PI-type adaptation algorithm of adaptive fuzzy control design for chaotic circuit systems,” *11th World Congress of International Fuzzy Systems Association*, pp. 876-881, 2005.

[4] Ping-Zong Lin, Wei-Yen Wang, Tsu-Tian Lee, Pin-Cheng Chen (Prev. as Guan-Ming

Chen), "On-line genetic fuzzy-neural sliding mode controller design," *IEEE Conf. Systems, Man and Cybernetics*, vol. 1, pp. 245-250, 2005.

[5] Bore-Kuen Lee, Chun-Fei Hsu, Pin-Cheng Chen (Prev. as Guan-Ming Chen), Tsu-Tian Lee, "Adaptive Fuzzy Control Using PID-Type Learning Algorithm," *International MultiConference of Engineers and Computer Scientists*, pp. 1609-1613, 2007.

[6] Pin-Cheng Chen (Prev. as Guan-Ming Chen), Ping-Zong Lin, Wei-Yen Wang, Tsu-Tian Lee, and Chi-Hsu Wang, "Fuzzy control using intuitive image analysis**,"** *IEEE Conf. Human System Interactions***,** pp. 394-399. **(Best Paper Award)**

國內研討會論文:

[1] Pin-Cheng Chen (Prev. as Guan-Ming Chen), Chun-Fei Hsu, Chi-Hsu Wang, Tsu-Tian Lee, "Design of an Adaptive Fuzzy Backstepping Controller with Bound Estimation Algorithm," *National Symposium on System Science and Engineering*, 2007（2007年中華民國系統科學與工程會議）