

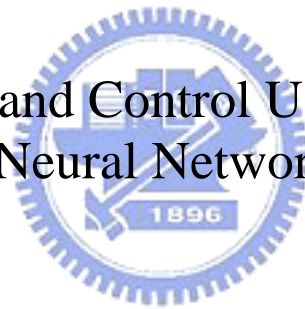
國立交通大學

電機與控制工程系

博士論文

使用靜態和動態類神經網路做系統鑑別和控制設計

System Identification and Control Using Static and Dynamic
Neural Networks



研究生：林炳榮

指導教授：王啟旭 教授

李祖添 教授

中華民國九十七年七月

使用靜態和動態類神經網路做系統鑑別和控制設計
System Identification and Control Using Static and Dynamic Neural Networks

研究生：林炳榮

Student : Ping-Zong Lin

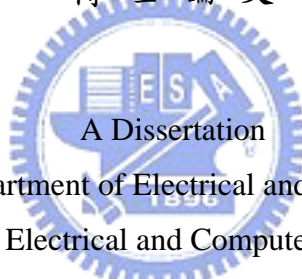
指導教授：王啟旭

Advisor(s) : Chi-Hsu Wang

李祖添

Tsu-Tian Lee

國立交通大學
電機與控制工程系
博士論文



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

Electrical and Control Engineering

July 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年七月

使用靜態和動態類神經網路做系統鑑別和控制設計

研究生：林炳榮

指導教授：王啟旭 博士

李祖添 博士

國立交通大學電機與控制工程系博士班

摘 要

針對非線性動態系統，本論文發展一個新的模糊類神經控制器和一個新的霍普菲爾動態類神經網路鑑別器。第一個設計是提出一個適應性自我建構的非對稱性模糊類神經網路控制器，此控制器是由一個自我建構的模糊類神經網路控制器和一個強健控制器組成。自我建構模糊類神經網路控制器具有架構和參數學習功能的自我建構模糊類神經網路，因此可用以模仿一個理想控制器。強健控制器是用來補償自我建構模糊類神經網路控制器和理想控制器之間的模仿誤差。提出的適應性自我建構非對稱性模糊類神經網路控制器應用到二階的混沌系統，模擬的結果顯示提出的控制器可以達到不錯的追跡效果。對於第二個設計，提出一個新的基於霍普菲爾的動態類神經網路，用以執行非線性動態系統的鑑別。應用 Lyapunov 方法調整神經網路的權重值。藉著似 Lyapunov 的穩定準則，執行穩定性的分析，且可以保證系統鑑別的誤差收斂性。最後，為了說明此方法的有效性，所提出的設計機構用以鑑別兩個非線性動態系統。模擬的結果顯示，使用 Lyapunov 方法訓練的動態類神經網路可以得到好的鑑別效果，且符合文中所推導的收斂作用。

System Identification and Control Using Static and Dynamic Neural Networks

Student : Ping-Zong Lin

Advisor(s): Dr. Chi-Hsu Wang

Dr. Tsu-Tian Lee

Department of Electrical and Control Engineering

National Chiao Tung University

ABSTRACT

In this dissertation, a novel fuzzy neural network control law and a new Hopfield-based dynamic neural network identifier is developed for nonlinear dynamic systems. For the first control design, an adaptive self-structuring asymmetric fuzzy neural-network control (ASAFNC) system which consists of a self-structuring fuzzy neural-network (SFNN) controller and a robust controller is proposed. The SFNN controller uses a SFNN with structure and parameter learning phases to mimic an ideal controller in a real-time environment. The robust controller is designed to compensate for the modeling error between the SFNN controller and the ideal controller. The proposed ASAFNC system is applied to a second-order chaotic dynamics system. The simulation results show that the proposed ASAFNC can achieve favorable tracking performance. For the second scheme, a new dynamic neural network based on the Hopfield neural network is proposed to perform the nonlinear system identification. The weighting factors of the proposed neural network are adjusted by the Lyapunov approach. Stability analysis is performed by the Lyapunov-like criterion to guarantee the error convergence during identification. Finally, in order to illustrate the effectiveness of this method, the proposed scheme is applied to identify two nonlinear systems. The simulation results demonstrate that the proposed dynamic neural network trained by the Lyapunov approach can obtain good identified performance which is consistent with the convergent analysis proposed in this dissertation.

Acknowledgement

首先，我非常感謝指導教授李祖添老師，在我就讀博士班期間，給予我課業和生活上的指導、教誨與幫助，老師也讓我和研究室的同學們學習承辦大型的國際研討會，讓我學到了很多東西，老師亦也提供許多機會，讓我可以參加國外研討會，學習到了許多寶貴的經驗與知識，儘管老師無法一直待在交大指導我，依舊不忘關心我的課業與生活。我也非常感謝另一位指導教授王啟旭老師，指導的時間雖然只有短短的一年多，卻提供我許多的機會，讓我磨練所欠缺的一些經驗，老師亦也在論文上提供許多寶貴的意見，花時間與我一對一的討論，解決研究上的瓶頸。對於老師們的栽培之恩，將永銘於心。

誠摯的感激口試委員：鄧清政教授、蘇順豐教授、王偉彥教授、以及呂藝光教授，在百忙之中撥冗，不辭遠道的蒞臨指導，提供寶貴的意見以及不同的思考方向與問題，使得論文內容能更加完整與正確。

在博士班的求學過程中，非常感謝王偉彥教授在我碩士班畢業後，亦願意繼續花時間給予我研究上的指導。也非常感謝許駿飛學長，願意投入大量的時間和我討論研究上的任何問題，分享經驗，參加國外研討會時，也給予我許多的意見與幫助；此外，對於研究室的保村學長、品程和欣翰，ITS的研究團隊，以及研究室已畢業和未畢業的學弟、學妹們給予感謝，大家一起在研究室裡度過讀書、研究、嬉鬧、出遊的日子。另外，也感謝汪汪社的社團指導老師和每位夥伴們，讓我學到很多課程以外的知識。

在此，我也非常感謝我摯愛的家人，你(妳)們的支持、照顧、容忍和深切的期盼是我完成博士學位的精神支柱，謹將這一份小小榮耀與你(妳)們同賀。

最後，感謝所有直接或是間接幫助過我的朋友們，有你(妳)們的協助，才能讓我完成這本博士論文，順利取得博士學位。

Table of Content

Abstract in Chinese	i
Abstract in English	ii
Acknowledgement	iii
Contents	iv
List of Figures	vi
List of Tables	viii
Nomenclature	ix
1. Introduction	1
1.1 Background and Motivation	1
1.2 Major Works	4
1.3 Dissertation Overview	5
2. Adaptive Self-structuring Asymmetric Fuzzy Neural-network Control Design	7
2.1 Problem Statement	7
2.2 Description of SFNN	9
2.3 Approximation of SFNN	17
2.4 ASAFNC Design	19
2.5 Boundary Analysis Using Projection Algorithm	21
2.6 Simulation Results	24
2.6.1 Comparison with AFNC	25
2.6.2 Simulation for ASAFNC	33
3. System Identification via Hopfield-based Dynamic Neural Network	40
3.1 Preliminary	40
3.1.1 Brief of HNN	41
3.1.2 Stability Analysis of Network	43
3.1.3 Problem Statement	45

3.2	Identification of Hopfield-based DNN.....	46
3.3	Robust Analysis.....	50
3.4	Simulation Results of Magnetic Levitation System.....	54
3.5	Simulation Results of Non-affine System.....	60
4.	Performance Comparison between SFNN and Hopfield DNN.....	66
4.1	Software Analysis with Implementation.....	66
4.2	Hardware Analysis with Implementation.....	67
4.3	Summary.....	68
5.	Conclusions with Future Works.....	69
	References	71
	Vita	76
	Publication List	77



List of Figures

Fig. 2-1.	The block diagram of ASAFNC system.....	8
Fig. 2-2.	The structure of SFNN.....	10
Fig. 2-3.	The rise and decay curves of the used frequency index.....	14
Fig. 2-4.	The flow chart of the ASAFNC system.....	16
Fig. 2-5.	Phase plane of uncontrolled chaotic dynamics system.....	25
Fig. 2-6.	Simulation results of AFNC using 3 symmetric membership functions.....	27
Fig. 2-7.	Simulation results of AFNC using 20 symmetric membership functions.....	29
Fig. 2-8.	Simulation results of AFNC using 3 asymmetric membership functions.....	31
Fig. 2-9.	Simulation results of AFNC using 20 asymmetric membership functions.....	33
Fig. 2-10.	Simulation results of ASAFNC for $q = 1.95$	35
Fig. 2-11.	Simulation results of ASAFNC for $q = 7.00$	36
Fig. 2-12.	Simulation results of ASAFNC for $q = 1.95$ with different trajectory.....	38
Fig. 2-13.	Simulation results of ASAFNC for $q = 7.00$ with different trajectory.....	39
Fig. 3-1.	Architectural graph of a Hopfield network with N neurons.....	41
Fig. 3-2.	A single neuron of Hopfield neural network.....	41
Fig. 3-3.	The hyperbolic tangent function with $a = 4$	42
Fig. 3-4.	The block diagram of identification architecture of the DNN based on a HNN.....	45
Fig. 3-5.	Architecture of the dynamic neural network based on a Hopfield neural network.....	47
Fig. 3-6.	Training data obtained from the magnetic levitation system.....	56
Fig. 3-7.	Behavior of identification system.....	57
Fig. 3-8.	The error of identification.....	58
Fig. 3-9.	The training conditions of weighting factors.....	60

Fig. 3-10.	Training data obtained from experimenting in advance.....	61
Fig. 3-11.	Simulation results of identification system. (a) is the response of state x ; (b) is the enlarging drawing of (a); and (c) shows the approximated errors.....	62
Fig. 3-12.	The training conditions of weighting factors.....	63
Fig. 3-13.	Simulation results of identification system. (a) is the response of state x ; (b) is the enlarging drawing of (a); and (c) shows the approximated errors.....	64
Fig. 3-14.	The training conditions of weighting factors.....	65



List of Tables

Table 4-1.	The comparison result between SFNN and Hopfield-based DNN for the software and hardware.....	68
------------	--	----



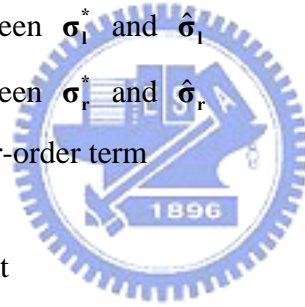
NOMENCLATURE

<Chapter 2>

x	System state
\mathbf{x}	State vector of the system
$f(\mathbf{x})$	System dynamic equation
u	Control effort
x_c	Command trajectory
e	Tracking error
k_i	Non-zero positive constants
u^*	Ideal control law
u_{ac}	Adaptive self-structuring asymmetric fuzzy neural-network control law
u_{sfnn}	SFNN controller
u_{rb}	Robust controller
s	Sliding surface
y_o	Output of the SFNN
N	Existing fuzzy rule number
w_k	Output action strength associated with the k -th rule
ϕ_k	Firing strength associated with the k -th rule
ζ_{ij}	Membership function
M	Total number of membership functions with respect to the respective input node
m_{ij}	Mean of the asymmetric Gaussian function in the j -th term of the i -th input linguistic variable x_i
σ_{ij}^l	Left-side variance of the asymmetric Gaussian function in the j -th term of the i -th input linguistic variable x_i
σ_{ij}^r	Right-side variance of the asymmetric Gaussian function in the j -th term of the i -th input linguistic variable x_i
ϖ	Small positive constant for the membership function
\mathbf{m}	Vector of mean of the asymmetric Gaussian function
σ_1	Vector of left-side variance of the asymmetric Gaussian function

σ_r	Vector of right-side variance of the asymmetric Gaussian function
\mathbf{w}	Vector of weighting factor
$\boldsymbol{\varphi}$	Vector of the firing strength
β_k	Degree measure of firing strength associated with the k -th rule
β_{\max}	Maximum degree measure
$N(t)$	Number of the existing fuzzy rules at the time t
G_{th}	Threshold for the growing method
m_i^{new}	Mean of the new membership function
$\sigma_i^{l,new}$	Left-side variance of the new membership function
$\sigma_i^{r,new}$	Right-side variance of the new membership function
w^{new}	Weight of the new membership function
P_{th}	Threshold for the pruning algorithm
I_r	Significant index of the r -th rule
τ_1	Designed constant for the pruning algorithm
τ_2	Designed constant for the pruning algorithm
I_{th}	Another threshold for the pruning algorithm
u_{sfm}^*	Optimal SFNN controller
Δ	Approximation error
$\boldsymbol{\varphi}^*$	Optimal vector of the firing strength
\mathbf{w}^*	Optimal vector of weighting factor
\mathbf{m}^*	Optimal vector of mean of the asymmetric Gaussian function
$\boldsymbol{\sigma}_l^*$	Optimal vector of left-side variance of the asymmetric Gaussian function
$\boldsymbol{\sigma}_r^*$	Optimal vector of right-side variance of the asymmetric Gaussian function
$\hat{\boldsymbol{\varphi}}$	Estimated vector of $\boldsymbol{\varphi}$
$\hat{\mathbf{w}}$	Estimated vector of \mathbf{w}
$\hat{\mathbf{m}}$	Estimated vector of \mathbf{m}
$\hat{\boldsymbol{\sigma}}_l$	Estimated vector of $\boldsymbol{\sigma}_l$
$\hat{\boldsymbol{\sigma}}_r$	Estimated vector of $\boldsymbol{\sigma}_r$
Ω_w	Compact set for \mathbf{w}

Ω_m	Compact set for \mathbf{m}
Ω_{σ_l}	Compact set for σ_l
Ω_{σ_r}	Compact set for σ_r
D_w	Positive constant
D_m	Positive constant
D_{σ_l}	Positive constant
D_{σ_r}	Positive constant
Δ^*	Upper bound for approximation error
\tilde{u}	Modeling error
$\tilde{\mathbf{w}}$	Difference between \mathbf{w}^* and $\hat{\mathbf{w}}$
$\tilde{\boldsymbol{\varphi}}$	Difference between $\boldsymbol{\varphi}^*$ and $\hat{\boldsymbol{\varphi}}$
$\tilde{\mathbf{m}}$	Difference between \mathbf{m}^* and $\hat{\mathbf{m}}$
$\tilde{\sigma}_l$	Difference between σ_l^* and $\hat{\sigma}_l$
$\tilde{\sigma}_r$	Difference between σ_r^* and $\hat{\sigma}_r$
\mathbf{h}	Vector of higher-order term
ε	Uncertain term
\mathbf{c}_0	Positive constant
\mathbf{c}_1	Positive constant
\mathbf{c}_2	Positive constant
\mathbf{c}_3	Positive constant
$\boldsymbol{\Theta}$	Vector of derived parameter
$\boldsymbol{\Gamma}$	Vector of derived parameter
η_w	Learning rate for w
η_m	Learning rate for m
η_{σ_l}	Learning rate for σ_l
η_{σ_r}	Learning rate for σ_r
δ	Attenuation constant
V	Lyapunov function
J_w	Parameter for the derivative of Lyapunov function



J_m	Parameter for the derivative of Lyapunov function
J_{σ_1}	Parameter for the derivative of Lyapunov function
J_{σ_r}	Parameter for the derivative of Lyapunov function
ω	Frequency for a second-order chaotic dynamics system
p	Real constants for a second-order chaotic dynamics system
p_1	Real constants for a second-order chaotic dynamics system
p_2	Real constants for a second-order chaotic dynamics system
q	Real constants for a second-order chaotic dynamics system

<Chapter 3>

$\varphi(\cdot)$	Nonlinear activation function
v_i	Voltage of the capacitance for the i th neural cell
z_i	Recurrent input and neural output for the i th neural cell
w_{ij}	Synaptic weighting factors
a_i	Gain parameter of neuron
C_i	Capacitance for the i th neural cell
R_i	Resistance for the i th neural cell
E	Energy function for the analysis of the Hopfield neural network
\mathbf{x}	System state vector
$F(\mathbf{x}, u)$	Unknown nonlinear function
u	Admissible control input
T	Time
$\hat{\mathbf{x}}$	State vector of the neural network
\mathbf{A}	Diagonal matrix of system state
\mathbf{B}	Diagonal matrix of nonlinear state feedback and system input
b_{ij}	Element of matrix \mathbf{B}
\mathbf{W}_φ	Matrix of synaptic weight for nonlinear state feedback
\mathbf{W}_u	Matrix of synaptic weight for input
Φ	Vector of the network feedback
q	Positive amplification

\mathbf{U}	Vector of the control force
\mathbf{W}_φ^*	Optimal matrix of \mathbf{W}_φ
\mathbf{W}_u^*	Optimal matrix of \mathbf{W}_u
$\Omega_{\mathbf{w}_\varphi}$	compact set for \mathbf{W}_φ
$\Omega_{\mathbf{w}_u}$	compact set for \mathbf{W}_u
Ω_x	compact set for \mathbf{x}
Ω_u	compact set for u
$D_{\mathbf{w}_\varphi}$	Positive constant
$D_{\mathbf{w}_u}$	Positive constant
\mathbf{e}	Approximation error
$\tilde{\mathbf{W}}_\varphi$	Difference between \mathbf{W}_φ^* and \mathbf{W}_φ
$\tilde{\mathbf{W}}_u$	Difference between \mathbf{W}_u^* and \mathbf{W}_u
η_φ	Learning rate
η_u	Learning rate
V	Lyapunov function
\mathbf{P}	Parameter of Lyapunov equation
\mathbf{Q}	Parameter of Lyapunov equation
s	Modeling error
γ	Constant
λ	Eigenvalue
$V_{\mathbf{w}_\varphi}$	Lyapunov function
$J_{\mathbf{w}_\varphi}$	Parameter for the derivative of Lyapunov function
$J_{\mathbf{w}_u}$	Parameter for the derivative of Lyapunov function
m	Mass of the ball
k	Viscous friction coefficient
g	Acceleration of gravity
$H(x, i)$	Force generated by the electromagnet
I	Electric current of the electromagnet system
x_1	Vertical gap between the ball and the magnet



x_2	Vertical velocity of the ball
u	Current in the coil of the electromagnet or control input
L_0	Nominal point inductance
a	Positive constant
k_1	State feedback gain for control law
k_2	State feedback gain for control law
$r(t)$	Reference position
$u_b(t)$	Model-based bias
A_j	Parameter of reference position
w_j	Parameter of reference position
r_0	Parameter of reference position
ζ	Disturbance



Chapter 1

Introduction

1.1 Background and Motivation

The development in the control area has been fueled by three major needs: the need to deal with increasingly complex systems, the need to accomplish increasingly demanding design requirements, and the need to attain these requirements with less precise advanced knowledge of the plant and its environment [1]. Hence, many researches are interested in some intelligent control design or intelligent systems to attain these needs.

In the past two decades, fuzzy systems have replaced conventional technologies in many scientific applications and engineering systems, especially in control systems. Fuzzy sets, introduced by Zadeh in 1965 [2] as a mathematical way to represent vagueness in linguistics, can be considered a generalization of classical set theory. Fuzzy sets are a generalization of conventional set theory and contain objects that belong imprecisely to the set. The degree of belonging is defined by the value of a membership function, which usually has values between 0 and 1. One of the biggest differences between crisp and fuzzy sets is that the former always have unique membership functions, whereas every fuzzy set has an infinite number of membership functions that may represent it. Fuzzy logic control (FLC) system, which induces human experience and human decision-making behavior, has been developed over 20 year. In the design of a FLC system, the sensory variables are converted into the fuzzy numbers by membership functions and they are matched with the preconditions of linguistic IF-THEN rules (fuzzy logic rules) and then the response of each rule is obtained through fuzzy computation. As a result, it will generally lead to fuzzy outputs. Finally, the fuzzy outputs are inverted into a crisp result to obtain the appropriate control signal. One major feature of fuzzy logic is its ability to express the amount of ambiguity in human thinking and subjectivity. In summary, the advantages of fuzzification include greater generality, higher expressive power, an enhanced ability to model real-world problems, and a methodology for exploiting the tolerance for imprecision. Hence, this algorithm provides a way of representing the uncertainties in a complex model. However, system designers must spend more time to ascertain how many rules are best [3] and fuzzy systems do not have

much learning capability [4].

The concept of neural network (NN) was first proposed by McCulloch and Pitts in 1943 [5]. NNs are a new generation of information processing systems that are deliberately constructed to make use of some of the organizational principles. They have a large number of highly interconnected processing elements (nodes) that usually operate in parallel and are configured in regular architectures. A NN has a massively parallel and distributed structure that is composed of many simple processing elements i.e., artificial neurons with nonlinear mapping functions. The neurons in a NN can communicate with each other through the links i.e., weights between the neurons [6]. The collective behavior of an NN is like a human brain to demonstrate the ability to learn, recall, and generalize from training patterns or data. NNs offer the salient characteristics and properties, such as nonlinear input-output mapping, generalization, adaptation, fault tolerance, and evidential response etc. Therefore, the NN has been applied to various areas [7-9]. However, because the internal layers of neural networks are always opaque to the user, the mapping rules in the network are not visible and are difficult to understand. The convergence of learning is usually very slow and not guaranteed [4].

Recently, the fuzzy neural network (FNN), which incorporates the advantages of fuzzy inference and neuro-learning, has been an interesting topic. Fuzzy logic and NNs are complementary technologies in the design of intelligent systems. The FNN possesses the merits of the low-level learning and computational power of NN, and the high-level human knowledge representation and thinking of fuzzy theory [4, 10]. Due to their learning ability, FNNs are increasingly receiving attention in solving the control problems [11-14]. Hence, the FNN will be a focus of our researches. Although the neuro-learning structure can tune membership functions and fuzzy rules automatically, the structure of the FNN should be determined in advance by trial-and-error. It is difficult to consider the balance between the rule number and the desired performance. As a result, if the number of fuzzy rules is chosen too large, the computation loading is heavy so that it is not suitable for practical applications. If the number of fuzzy rules is chosen too small, the control performance may be not good enough to achieve the desired performance.

To solve the problem of determining the structure in FNN approaches, much interest has been focused on the self-structuring fuzzy neural network (SFNN) approach [15-19]. The self-structuring approach demonstrates the properties of automatic generating rules for FNN without needing preliminary knowledge. In general, the mathematical description of the existing rules can be expressed as a set of clusters. As usually seen in other self-structuring

approaches, the new membership function is generated when a new input signal is too far from the current clusters, and an existing rule is deleted when the fuzzy rule is insignificant. SFNNs also have been adopted widely for the control of complex dynamic systems due to their good generalization capability, structural adaptation, and simple computation [20-25]. Some of them use the gradient descent method to derive the parameter learning algorithms; however, they can't guarantee the system stability [22, 23]. Some of them derive the parameter learning algorithms based on the Lyapunov function to guarantee system stability; however, the structure learning algorithm is too complex [20, 24, 25]. Some of them proposed a simple growing-and-pruning algorithm to online self-structure the FNN with symmetric membership functions; however, the bounds of parameters are not stated [21].

In addition, system identification also plays an important role in control field. It is an important task for control engineer to acquire system information so as to design a proper control law based on a good understanding of the plant under consideration and its environment. It has been clear that a mathematical description of a plant is often a prerequisite for system analysis and controller design in control system theory. System identification, whether online or offline, is an essential part of any control system design. The processes of system identification mainly consists of two steps: the first is to choose an appropriate identification model and the second is to adjust parameters of the selected model according to some derived adaptive laws so that the output of the selected model can approach the response of the real system under the same input [26]. Hence, the nonlinear system identification process has turned out to be one of central parts in various control researches.

Recent research results show that NN techniques seem to be very effective to identify a wide class of complex nonlinear systems when the complete model information can not be available [27-29]. NNs have been an interested focus because they have good learning, noise-tolerance, and generalization abilities to solve the nonlinear problem. According to the used types of NNs, they can be qualified as static (feed-forward) or as dynamic (recurrent) nets. The first one deals with the class of global optimization problems. The universal approximation property of static NNs makes them be a useful tool for modeling nonlinear systems. The designers try to adjust weights of such NNs to achieve favorable performance. The second approach, which converts the partial learning (training) focuses to an adequate feedback design, permits to avoid many problems related to global extremum search [30]. When outputs are directed back as inputs to the same or the preceding layer node, the network is a feedback network. Feedback networks that have closed loop are called recurrent networks. From a system theoretical point of view, multilayer networks represent static nonlinear maps

while recurrent networks are represented by nonlinear dynamic feedback systems [27]. However, an important viewpoint is that static NNs 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 [31, 32]. This drawback severely affects the applicability of static NNs to system identification, which is the central part in some control techniques for nonlinear systems. Dynamic neural networks (DNNs) can deal with this disadvantage since they have dynamic memory, which makes them more suitable for representing dynamic systems than static NNs. Hence, if the mathematical model of a considered process is incomplete or partially known, the DNN approach provides an effective instrument to research a wide spectrum of problems such as identification, state estimation, trajectories tracking, etc. [33]. Recurrently connected NNs, sometimes called Hopfield neural networks (HNN), which is a special kind of DNN, have been extensively studied in recent years. The HNN is first proposed by Hopfield J.J. in 1982 and 1984 [34, 35]. Because of the easy implementation of the HNN circuit, the characteristic of decreasing in energy by finite number of node-updating steps, and the dynamical behavior of the networks, the HNN has found many applications in different areas, such as optimization [36, 37], system identification [38, 39], and image processing [40, 41]. However, in [38, 39], the system identification via HNN involved a learning process which has no guarantee for convergence.

1.2 Major Works

In this dissertation, a SFNN in which the learning phase considers both the structure and parameter learning phases is proposed. The structure adaptation is described as follows. A new rule is generated when a new input signal is too far from the current clusters. To avoid the unrestricted growth of membership functions and fuzzy rules, we use an exponential function to calculate the significant indexes of each existing fuzzy rule. The exponential function can gradually increase or decrease the significant index values for each rule. If the fuzzy rule of SFNN is insignificant, it will be removed to reduce the computation load; and if the fuzzy rule of SFNN is significant, it will be retained. Thus, the SFNN can self-structure the fuzzy rules online to achieve an optimal network structure. Moreover, by accommodating the left-sided and right-sided spreads into a standard Gaussian membership functions, the asymmetric Gaussian membership functions can upgrade the learning capability and

flexibility of a NN [42].

Therefore, one of purposes of this dissertation is to develop an adaptive self-structuring asymmetric fuzzy neural-network control (ASAFNC) system, which consists of a SFNN controller and a robust controller. The SFNN controller utilizes a SFNN to mimic an ideal controller, and the robust controller is designed to compensate for the modeling error between the SFNN controller and the ideal controller. The learning phase of SFNN includes the structure learning phase and the parameter learning phase. The structure learning phase consists of the growing and pruning algorithms of fuzzy rules to achieve an optimal network structure, and the parameter learning phase adjusts the interconnection weights of NN to achieve favorable approximation performance. All the parameters of ASAFNC are tuned online based on the Lyapunov stability to achieve favorable performance. Finally, the effectiveness of the proposed ASAFNC scheme is demonstrated by simulations. The simulation results show that not only favorable tracking performance can be achieved but also a concise network structure can be obtained by the proposed structure learning method.

In addition, for the system identification, the other purpose of this dissertation is to develop a new HNN identifier to perform nonlinear system identification which can guarantee the convergence subject to several constraints. The weights of the proposed scheme will be adjusted to minimize the identification error by Lyapunov's method in a real-time environment. The guarantee of convergence for the identification process with robustness analysis will be explored. Finally, the proposed scheme is applied to identify two nonlinear systems to illustrate its effectiveness. The simulation results demonstrate that the proposed Hopfield-based DNN trained by the Lyapunov approach can obtain good identified performance which is consistent with the convergent analysis discussed in the later chapter.

1.3 Dissertation Overview

The rest of this dissertation is organized as follows. Chapter 2 describes the design procedure of an adaptive self-structuring asymmetric fuzzy neural-network control for the static neural network. The training algorithms of parameters, including means and variances of membership functions and weights of the NN, are developed. The stability analysis and example illustrations are also provided in this chapter. For the DNN, the Hopfield-based DNN identifier is developed in Chapter 3. The training algorithm of weighting factors of the DNN is investigated. The stability analysis and example illustrations are also provided in this

chapter. The software and hardware of the implementation comparison between SFNN and Hopfield DNN is provided in Chapter 4. Finally, conclusions with future works are included in Chapter 5.



Chapter 2

Adaptive Self-structuring Asymmetric Fuzzy Neural-network Control Design

According to the used types of neural networks (NNs), they can be qualified as static (feed-forward) or as dynamic (recurrent) nets. In this chapter, the development of the static NN is priority to be discussed. The control design of fuzzy neural network (FNN) is explored first. The stability of the control system and examples will be also illustrated in this chapter.

2.1 Problem Statement

Consider the n th-order nonlinear dynamic system of the form

$$\dot{x}^{(n)} = f(\mathbf{x}) + u \quad (2-1)$$

where $\mathbf{x} = [x \ \dot{x} \ \dots \ x^{(n-1)}]^T$, which is assumed to be available for measurement, is the state vector of the system, $f(\mathbf{x})$ is the system dynamics equation, and u is the control effort. The control objective is to find a control law so that the state trajectory x can track a command trajectory x_c , and thus a tracking error is defined as

$$e = x_c - x. \quad (2-2)$$

If the system dynamics $f(\mathbf{x})$ in (2-1) is well known, there exists an ideal controller as [43]

$$u^* = -f(\mathbf{x}) + \dot{x}_c^{(n)} + k_n e^{(n-1)} + \dots + k_2 \dot{e} + k_1 e \quad (2-3)$$

where k_i , $i = 1, 2, \dots, n$ is non-zero positive constant. Substituting (2-3) into (2-1) yields

$$e^{(n)} + k_n e^{(n-1)} + \dots + k_2 \dot{e} + k_1 e = 0. \quad (2-4)$$

If k_i are chosen to correspond to the coefficients of a Hurwitz polynomial whose roots lie strictly in the open left half of the complex plane, then $\lim_{t \rightarrow \infty} e = 0$ can be inferred for any starting initial conditions. However, because the system dynamics $f(\mathbf{x})$ may be unknown or perturbed in practice, the ideal control law u^* in (2-3) cannot be implemented easily. To

solve the problem of the model-based control approach for real-time implementation, adaptive fuzzy neural-network control (AFNC) techniques have been developed to control these kinds of unknown nonlinear dynamic systems [11-14]. These techniques use a structure of FNN to estimate the plant or controller parameters in a real-time environment. If the FNN is applied to estimate the model of the plant, it is called an indirect AFNC, and if the FNN is applied to estimate the controller of the plant, it is called a direct AFNC [44].

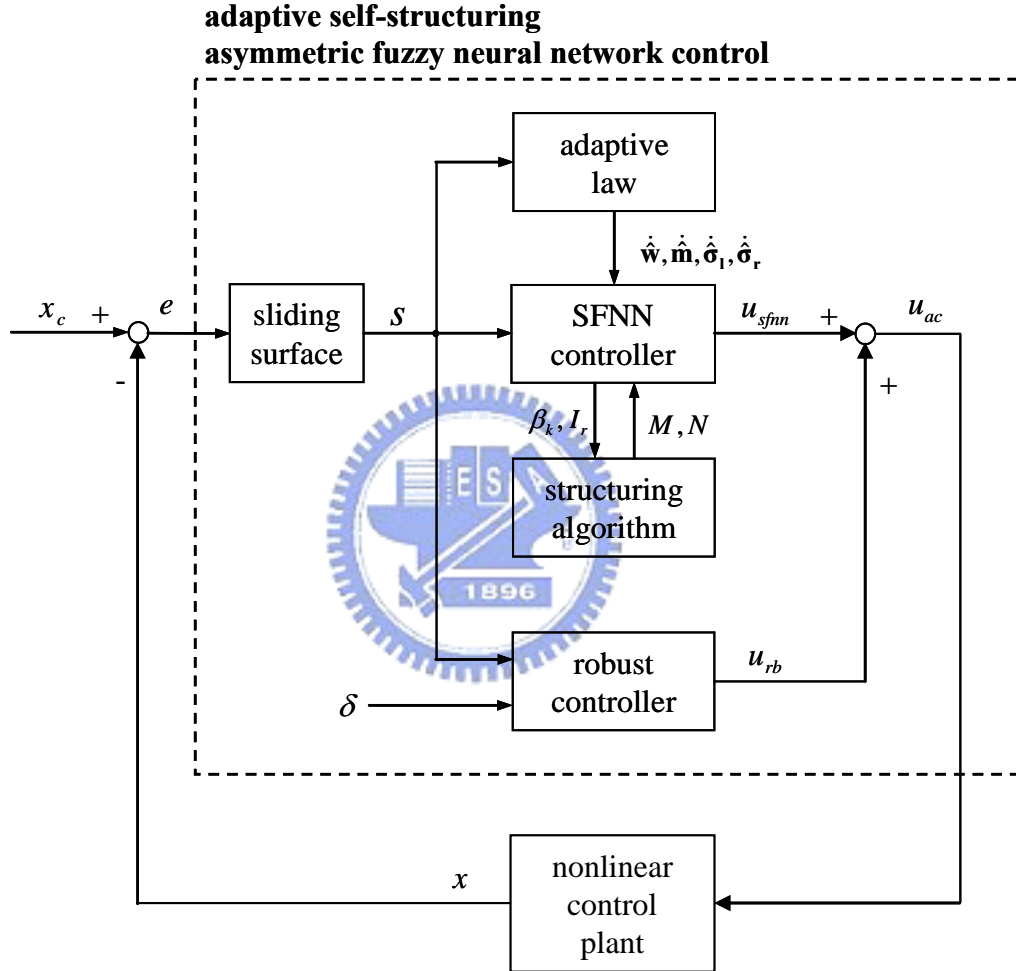


Fig. 2-1. The block diagram of ASAFNC system.

According to the design concept of the direct AFNC, we propose an adaptive self-structuring asymmetric fuzzy neural-network control (ASAFNC) system as shown in Fig. 2-1. The ASAFNC system is composed of a SFNN controller and a robust controller as

$$u_{ac} = u_{sfnn} + u_{rb} \quad (2-5)$$

The SFNN controller u_{sfnn} utilizes the SFNN with asymmetric Gaussian membership

functions to mimic the ideal controller in (2-3), and the robust controller u_{rb} is designed to compensate for the modeling error between the SFNN controller u_{sfnn} and the ideal controller u^* . For further analysis, first define a sliding surface as

$$s = e^{(n-1)} + k_n e^{(n-2)} + \dots + k_2 e + k_1 \int_0^t e \, d\tau. \quad (2-6)$$

Substituting (2-5) into (2-1) and using (2-3) and (2-6), yields

$$\dot{s} = u^* - u_{sfnn} - u_{rb}. \quad (2-7)$$

2.2 Description of SFNN

Fuzzy logic and NNs are complementary technologies in the design of intelligent systems. FNNs retain the basic properties and functions of NNs with some of their elements being fuzzified. In this approach, a network's domain knowledge becomes formalized in terms of fuzzy sets, later being applied to enhance the learning of the network and augment its interpretation capabilities. By incorporating fuzzy principles into a NN, more user flexibility is attained and the resultant network or system becomes more robust [4]. FNNs are generally a fuzzy inference system constructed from structure of NN. Learning algorithms are used to adjust the weightings of the fuzzy inference system.

Figure 2-2 shows the configuration of the proposed SFNN which is composed of the input, the membership, the rule, and the output layers. Layer 1 accepts the input variables. Nodes at layer 2 are term nodes which act as membership functions to represent the terms of the respective linguistic variables. The asymmetric Gaussian membership function constituted by a center, a left-side variance, and a right-side variance is considered. Nodes of layer 3 are regarded as fuzzy rules. The links before layer 3 represent the preconditions of rules and the links after layer 3 represent the consequences. Layer 4 is the output layer, where the node in this layer is the output of the NN. The interactions for those layers are given as follows.

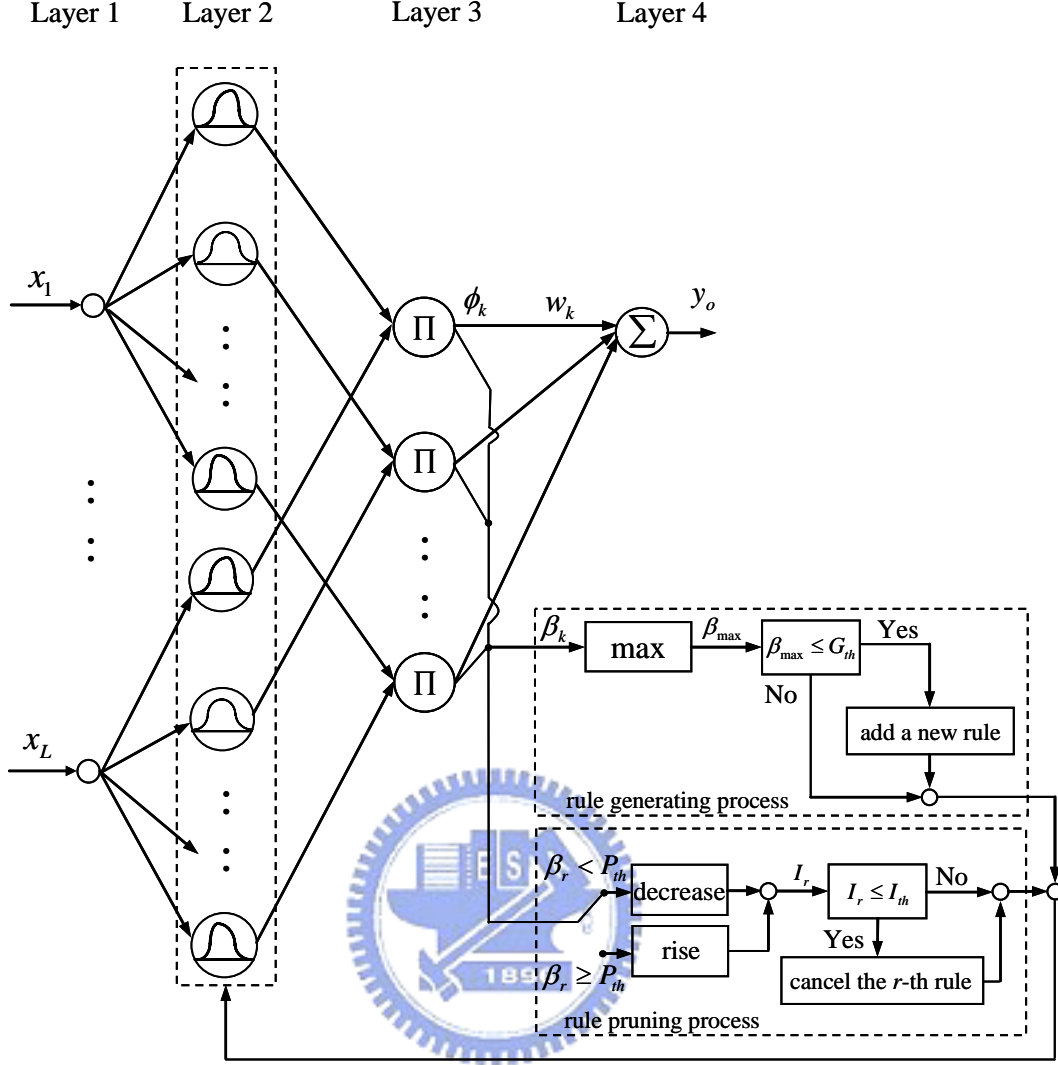


Fig. 2-2. The structure of SFNN.

Layer 1 - Input layer: For every node i in this layer, the net input and the net output are represented as

$${}^1net_i = {}^1x_i \quad (2-8)$$

$${}^1y_i = {}^1f_i({}^1net_i) = {}^1net_i, \quad i = 1, 2, \dots, L \quad (2-9)$$

where 1x_i represents the i th input to the node of layer 1 and L is the total number of input variables. They mean that output equals input in this layer. This layer of SFNN just executes the transmission work.

Layer 2 - Membership layer: In this layer, each node performs a membership function and acts as a unit of memory. The bell-shaped function is adopted as the membership function. For the i th input, the corresponding net input and output of the j th node can be expressed as

$${}^2net_{ij} = -\frac{({}^2x_i - {}^2m_{ij})^2}{({}^2\sigma_{ij})^2} \quad (2-10)$$

$${}^2y_{ij} = {}^2f_{ij}({}^2net_{ij}) = \exp({}^2net_{ij}), \quad j=1, 2, \dots, M \quad (2-11)$$

where ${}^2m_{ij}$ is the mean, ${}^2\sigma_{ij}$ is the variance and M is the total number of membership functions with respect to the respective input node. In this study, the input linguistic variable is the tracking error vector.

Layer 3 - Rule layer: Each node k in this layer is denoted by Π which multiplies the incoming signals and outputs the result of the product. For the k th rule node, the operation of the net input and output of this layer is presented as

$${}^3net_k = \prod {}^3w_{ij} {}^3x_{ij} \quad (2-12)$$

$${}^3y_k = {}^3f_k({}^3net_k) = {}^3net_k, \quad k=1, 2, \dots, N \quad (2-13)$$

where ${}^3x_{ij}$ represents the i, j th input to the k th node of layer 3, ${}^3w_{ij}$ between the membership and the rule layers are assumed as unity, and N is the total number of fuzzy rules.

Layer 4 - Output layer: The single node o in this layer is labeled as Σ , which computes the overall output as the summation of all incoming signals. It executes the sun-of-weighting defuzzification. The description of the net input and output is expressed as

$${}^4net_o = \sum_k {}^4w_k {}^4x_k \quad (2-14)$$

$${}^4y_o = {}^4f_o({}^4net_o) = {}^4net_o \quad (2-15)$$

where 4w_k is the output action strength of the output associated with the k th rule, 4x_k represents the k th input to the node of layer 4, and 4y_o is the output of SFNN.

In order to improve the learning capability and flexibility of a NN, asymmetric Gaussian membership functions are adopted, instead of ball-shaped functions described in layer 2. According to the above description, the output of the SFNN with N existing fuzzy rules can be represented simply as

$$y_o = \sum_{k=1}^N w_k \phi_k(\mathbf{x}) \quad (2-16)$$

in which w_k is the output action strength associated with the k -th rule and ϕ_k is the response of the firing weight for an input vector $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_L]^T$ and composed of asymmetric Gaussian membership functions defined as [42]

$$\zeta_{ij} = \begin{cases} \exp\left(-\frac{(x_i - m_{ij})^2}{(\sigma_{ij}^l)^2}\right), & \text{if } -\infty < x_i \leq m_{ij} \\ \exp\left(-\frac{(x_i - m_{ij})^2}{(\sigma_{ij}^r)^2}\right), & \text{if } m_{ij} \leq x_i < \infty \end{cases}, \quad j=1, 2, \dots, M \quad (2-17)$$

where M is the total number of membership functions with respect to the respective input node; m_{ij} , σ_{ij}^l , and σ_{ij}^r are the mean, left-side variance, and right-side variance of the asymmetric Gaussian function in the j -th term of the i -th input linguistic variable x_i , respectively. However, σ_{ij}^l and σ_{ij}^r may become zero in the training procedure, the membership function ζ_{ij} will not be defined. To avoid this problem, this dissertation considers a membership function form as [44]

$$\zeta_{ij} = \begin{cases} \exp\left(-\frac{(x_i - m_{ij})^2}{(\sigma_{ij}^l)^2 + \varpi}\right), & \text{if } -\infty < x_i \leq m_{ij} \\ \exp\left(-\frac{(x_i - m_{ij})^2}{(\sigma_{ij}^r)^2 + \varpi}\right), & \text{if } m_{ij} \leq x_i < \infty \end{cases}, \quad j=1, 2, \dots, M \quad (2-18)$$

where ϖ is a small positive constant. Then, the associated firing strength can be defined as

$$\phi_k = \prod_{j=1}^M \zeta_{jk} \quad (2-19)$$

To note easily, define vectors \mathbf{m} , $\boldsymbol{\sigma}_l$, and $\boldsymbol{\sigma}_r$ collecting all parameters of SFNN as

$$\mathbf{m} = [m_{11} \cdots m_{L1} \ m_{12} \cdots m_{L2} \ \cdots \ m_{1M} \cdots m_{LM}]^T \quad (2-20)$$

$$\boldsymbol{\sigma}_l = [\sigma_{11}^l \cdots \sigma_{L1}^l \ \sigma_{12}^l \cdots \sigma_{L2}^l \ \cdots \ \sigma_{1M}^l \cdots \sigma_{LM}^l]^T \quad (2-21)$$

$$\boldsymbol{\sigma}_r = [\sigma_{11}^r \cdots \sigma_{L1}^r \ \sigma_{12}^r \cdots \sigma_{L2}^r \ \cdots \ \sigma_{1M}^r \cdots \sigma_{LM}^r]^T \quad (2-22)$$

Thus, the output of the SFNN can be represented in a vector form as

$$y_o = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{m}, \boldsymbol{\sigma}_l, \boldsymbol{\sigma}_r) \quad (2-23)$$

where $\mathbf{w} = [w_1 \ w_2 \ \cdots \ w_N]^T$ and $\boldsymbol{\phi} = [\phi_1 \ \phi_2 \ \cdots \ \phi_N]^T$. For the FNN approaches, the structure of the FNN should be determined in advance by empiricism. However, it is difficult to consider the balance between the rule number and the desired performance. Therefore, the structure adaptation algorithm which contains the growing and pruning of membership functions and fuzzy rules is proposed in this dissertation. The descriptions are given as follows.

In the process of the growing of membership functions, the concept which decides

whether to add a new node (membership function) in layer 2 and the associated fuzzy rule in layer 3 will be introduced. The mathematical description of the existing rules can be expressed as a set of clusters. For constructing the initial fuzzy rules of the SFNN, the fuzzy clustering method is used to partition a set of data into a number of overlapping clusters based on the distance in a metric space between the data points and the cluster prototypes. Each cluster in the product space of the input-output data represents a rule. The firing strength of a rule for each incoming data x_i can be represented as the degree that the incoming data belong to the cluster [19]. If the value of firing strength is too small, it indicates that the input value is on the edge of range of the existing membership functions. Under this situation, the output will cause unsatisfactory performance. Therefore, a new membership function and a new fuzzy rule should be generated to improve the performance.

The firing strength from (2-19) is used as the degree measure

$$\beta_k = \phi_k, \quad k = 1, 2, \dots, N(t) \quad (2-24)$$

where $N(t)$ is the number of the existing fuzzy rules at the time t . Define the maximum degree β_{\max} as

$$\beta_{\max} = \max_{1 \leq k \leq N(t)} \beta_k. \quad (2-25)$$

If $\beta_{\max} \leq G_{th}$ is satisfied, where $G_{th} \in (0,1)$ is a pre-given threshold, the incoming data is far from the edge of range of the existing membership functions. Hence, a new membership function is generated. The mean and the variance of the new membership function and the weight are selected as follows

$$m_i^{new} = x_i, \quad (2-26)$$

$$\sigma_i^{l,new} = \sigma_i, \quad (2-27)$$

$$\sigma_i^{r,new} = \sigma_i, \quad (2-28)$$

$$w^{new} = 0 \quad (2-29)$$

where x_i is the new incoming data and σ_i is a pre-specified constant. If the unknown control system dynamics is too complex, we can choose the larger G_{th} so that many membership functions can be created.

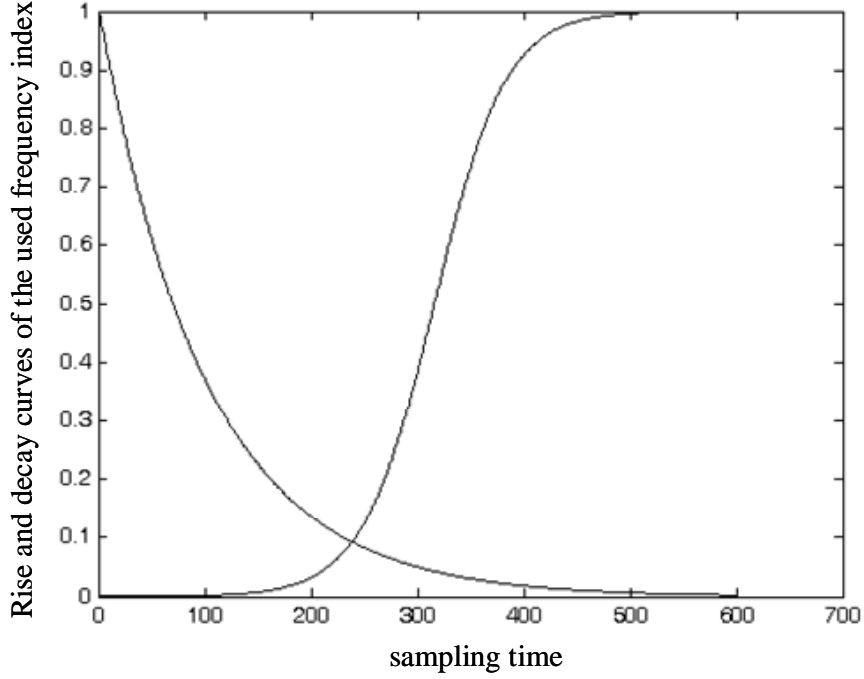


Fig. 2-3. The rise and decay curves of the used frequency index.

Next, to avoid the unrestricted growth of network structure and an overload computation, the pruning algorithm is developed to eliminate irrelevant fuzzy rules. In Ref. [21], a significance index is determined for the importance of the fuzzy rules. The elimination algorithm is derived from the observation that if the significance index fades when the firing weight is smaller than a special threshold value and if the significance index fixes when the firing weight is larger than a special threshold value [21]. In this dissertation, when the r -th firing strength β_r is smaller than the threshold value P_{th} , it indicates that the relationship becomes weak between the input and the r -th rule. Then, the significant index of r -th fuzzy rules will be decreased. When the r -th firing strength β_r is larger than the threshold value P_{th} , it indicates that the incoming inputs fall into the range of the r -th fuzzy rule. Thus, the significant index of r -th fuzzy rules should be raised. The rise and decay curves of the used frequency index show in Fig. 2-3. The significance index is determined for the r -th rules can be given as

$$I_r(t+1) = \begin{cases} I_r(t) \cdot \exp(-\tau_1), & \text{if } \beta_r < P_{th} \\ I_r(t) \cdot [2 - \exp(-\tau_2(1 - I_r(t)))] & \text{if } \beta_r \geq P_{th} \end{cases}, \quad r = 1, 2, \dots, N(t) \quad (2-30)$$

where I_r is the significant index of the r -th rule and its initial value is 1, P_{th} is the pruning

threshold value, and τ_1 and τ_2 are the designed constant. Exponential functions in (2-30) are used to rise or decrease the values of significant index in $[0, 1]$. If $I_r \leq I_{th}$ is satisfied, where I_{th} is another pre-given threshold, the r -th fuzzy rule will be deleted. For real-time implementation, if the computation load is the issue having highest priority, P_{th} should be chosen large, so that more fuzzy rules can be pruned. This operation will prevent the fuzzy rule, which may be less used but still significant, from being deleted in the training process. Hence, the computation load would be reduced.

In summary, the flow chart of the structure learning algorithm is shown in Fig. 2-4. The major contributions of the SFNN are: 1) SFNN can be operated directly without spending much time pre-determining membership functions and fuzzy rules; and 2) the computation load can be reduced simultaneously.



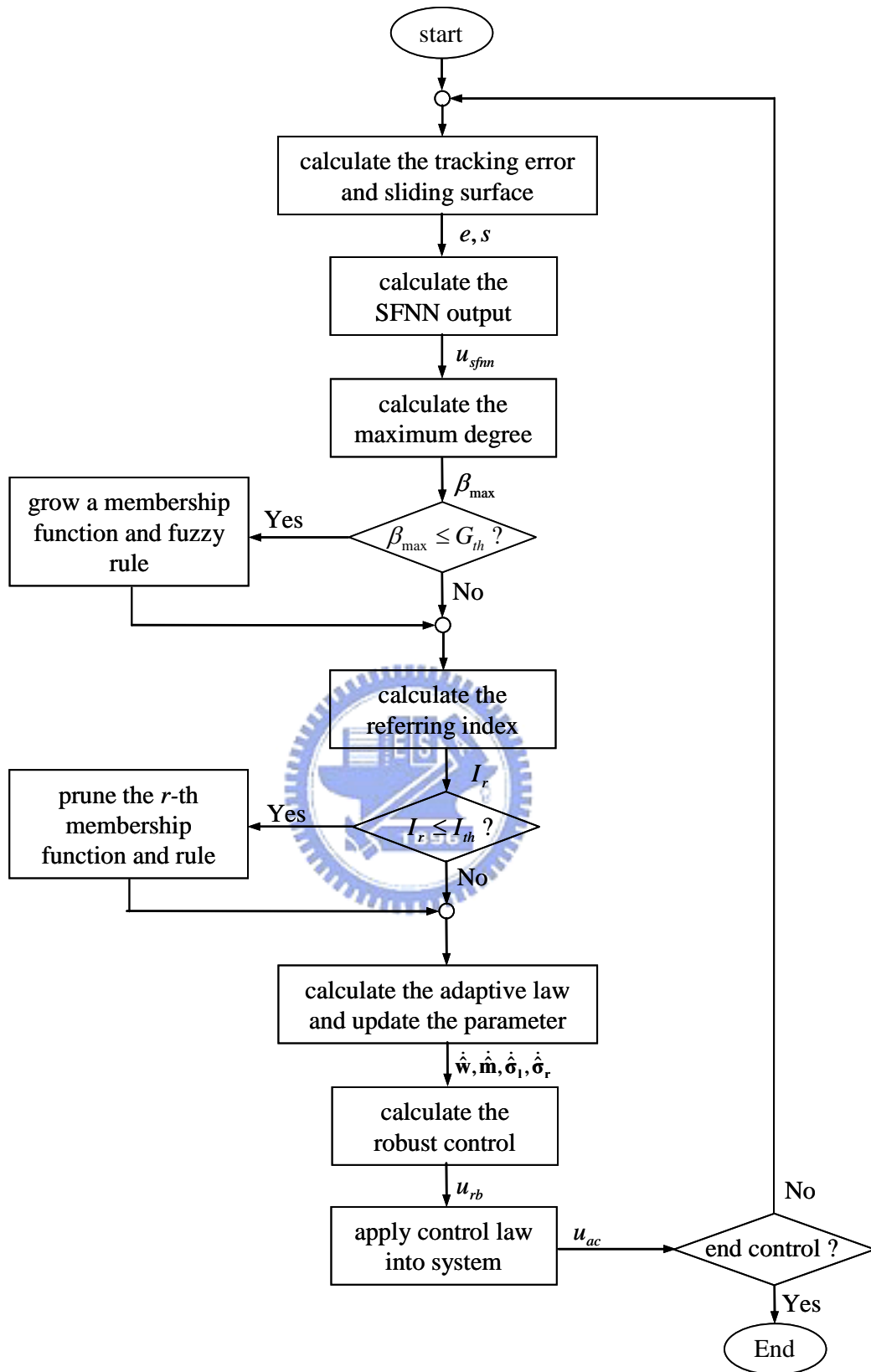


Fig. 2-4. The flow chart of the ASAFNC system.

2.3 Approximation of SFNN

An optimal SFNN controller can be designed to approximate the ideal controller (2-3) even under the structural change of neural network, such that [44, 45]

$$u^* = u_{sfnn}^* + \Delta = \mathbf{w}^{*T} \boldsymbol{\varphi}(\mathbf{x}, \mathbf{m}^*, \boldsymbol{\sigma}_1^*, \boldsymbol{\sigma}_r^*) + \Delta = \mathbf{w}^{*T} \boldsymbol{\varphi}^* + \Delta \quad (2-31)$$

where $\boldsymbol{\varphi}^* = \boldsymbol{\varphi}(\mathbf{x}, \mathbf{m}^*, \boldsymbol{\sigma}_1^*, \boldsymbol{\sigma}_r^*)$, Δ denotes the approximation error, and \mathbf{w}^* , \mathbf{m}^* , $\boldsymbol{\sigma}_1^*$, and $\boldsymbol{\sigma}_r^*$ are the optimal vectors. In fact, the optimal vectors that best approximate a given nonlinear function are difficult to be determined. Thus, an estimated SFNN controller is introduced as

$$u_{sfnn} = \hat{\mathbf{w}}^T \boldsymbol{\varphi}(\mathbf{x}, \hat{\mathbf{m}}, \hat{\boldsymbol{\sigma}}_1, \hat{\boldsymbol{\sigma}}_r) = \hat{\mathbf{w}}^T \hat{\boldsymbol{\varphi}} \quad (2-32)$$

where $\hat{\boldsymbol{\varphi}} = \boldsymbol{\varphi}(\mathbf{x}, \hat{\mathbf{m}}, \hat{\boldsymbol{\sigma}}_1, \hat{\boldsymbol{\sigma}}_r)$ and $\hat{\mathbf{w}}$, $\hat{\mathbf{m}}$, $\hat{\boldsymbol{\sigma}}_1$, and $\hat{\boldsymbol{\sigma}}_r$ are the estimated vectors of \mathbf{w} , \mathbf{m} , $\boldsymbol{\sigma}_1$, and $\boldsymbol{\sigma}_r$, respectively. Moreover, the optimal vectors can be further defined as [44]

$$(\mathbf{w}^*, \mathbf{m}^*, \boldsymbol{\sigma}_1^*, \boldsymbol{\sigma}_r^*) = \arg \min_{\hat{\mathbf{w}} \in \Omega_w, \hat{\mathbf{m}} \in \Omega_m, \hat{\boldsymbol{\sigma}}_1 \in \Omega_{\sigma_1}, \hat{\boldsymbol{\sigma}}_r \in \Omega_{\sigma_r}} \left[\sup_{\mathbf{x} \in \Omega_x \times R} |u_{sfnn}^*(\mathbf{x}) - u_{sfnn}(\mathbf{x}, \hat{\mathbf{m}}, \hat{\boldsymbol{\sigma}}_1, \hat{\boldsymbol{\sigma}}_r)| \right] \quad (2-33)$$

where

$$\Omega_w = \{ \hat{\mathbf{w}} : \|\hat{\mathbf{w}}\| \leq D_w \} \quad (2-34)$$

$$\Omega_m = \{ \hat{\mathbf{m}} : \|\hat{\mathbf{m}}\| \leq D_m \} \quad (2-35)$$

$$\Omega_{\sigma_1} = \{ \hat{\boldsymbol{\sigma}}_1 : \|\hat{\boldsymbol{\sigma}}_1\| \leq D_{\sigma_1} \} \quad (2-36)$$

$$\Omega_{\sigma_r} = \{ \hat{\boldsymbol{\sigma}}_r : \|\hat{\boldsymbol{\sigma}}_r\| \leq D_{\sigma_r} \} \quad (2-37)$$

where D_w , D_m , D_{σ_1} , and D_{σ_r} are positive constants specified by designers. There exists

Δ^* which is a finite positive constant such that the inequality $|\Delta| \leq \Delta^*$ can be held. Define a

modeling error, \tilde{u} , as

$$\tilde{u} = u^* - u_{sfnn} = \tilde{\mathbf{w}}^T \tilde{\boldsymbol{\varphi}} + \hat{\mathbf{w}}^T \tilde{\boldsymbol{\varphi}} + \tilde{\mathbf{w}}^T \hat{\boldsymbol{\varphi}} + \Delta \quad (2-38)$$

where $\tilde{\mathbf{w}} = \mathbf{w}^* - \hat{\mathbf{w}}$ and $\tilde{\boldsymbol{\varphi}} = \boldsymbol{\varphi}^* - \hat{\boldsymbol{\varphi}}$. In the following description, the linearization technique is employed to transform the nonlinear fuzzy function into a partially linear form so that the expansion $\tilde{\boldsymbol{\varphi}}$ can be expressed as [46]

$$\begin{aligned}
\tilde{\boldsymbol{\varphi}} = \begin{bmatrix} \tilde{\phi}_1 \\ \tilde{\phi}_2 \\ \vdots \\ \tilde{\phi}_N \end{bmatrix} &= \begin{bmatrix} \frac{\partial \phi_1}{\partial \mathbf{m}} \\ \frac{\partial \phi_2}{\partial \mathbf{m}} \\ \vdots \\ \frac{\partial \phi_N}{\partial \mathbf{m}} \end{bmatrix}_{\mathbf{m}=\hat{\mathbf{m}}} (\mathbf{m}^* - \hat{\mathbf{m}}) + \begin{bmatrix} \frac{\partial \phi_1}{\partial \boldsymbol{\sigma}_1} \\ \frac{\partial \phi_2}{\partial \boldsymbol{\sigma}_1} \\ \vdots \\ \frac{\partial \phi_N}{\partial \boldsymbol{\sigma}_1} \end{bmatrix}_{\boldsymbol{\sigma}_1=\hat{\boldsymbol{\sigma}}_1} (\boldsymbol{\sigma}_1^* - \hat{\boldsymbol{\sigma}}_1) + \begin{bmatrix} \frac{\partial \phi_1}{\partial \boldsymbol{\sigma}_r} \\ \frac{\partial \phi_2}{\partial \boldsymbol{\sigma}_r} \\ \vdots \\ \frac{\partial \phi_N}{\partial \boldsymbol{\sigma}_r} \end{bmatrix}_{\boldsymbol{\sigma}_r=\hat{\boldsymbol{\sigma}}_r} (\boldsymbol{\sigma}_r^* - \hat{\boldsymbol{\sigma}}_r) + \mathbf{h} \\
&= \boldsymbol{\varphi}_m^T \tilde{\mathbf{m}} + \boldsymbol{\varphi}_{\sigma_1}^T \tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{\varphi}_{\sigma_r}^T \tilde{\boldsymbol{\sigma}}_r + \mathbf{h}
\end{aligned} \tag{2-39}$$

where \mathbf{h} is a vector of higher-order term, $\tilde{\mathbf{m}} = \mathbf{m}^* - \hat{\mathbf{m}}$, $\tilde{\boldsymbol{\sigma}}_1 = \boldsymbol{\sigma}_1^* - \hat{\boldsymbol{\sigma}}_1$, and $\tilde{\boldsymbol{\sigma}}_r = \boldsymbol{\sigma}_r^* - \hat{\boldsymbol{\sigma}}_r$.

Substituting (2-39) into (2-38), (2-38) can be rewritten as

$$\begin{aligned}
\tilde{u} &= \tilde{\mathbf{w}}^T \tilde{\boldsymbol{\varphi}} + \hat{\mathbf{w}}^T (\boldsymbol{\varphi}_m^T \tilde{\mathbf{m}} + \boldsymbol{\varphi}_{\sigma_1}^T \tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{\varphi}_{\sigma_r}^T \tilde{\boldsymbol{\sigma}}_r + \mathbf{h}) + \tilde{\mathbf{w}}^T \hat{\boldsymbol{\varphi}} + \Delta \\
&= \tilde{\mathbf{w}}^T \hat{\boldsymbol{\varphi}} + \tilde{\mathbf{m}}^T \boldsymbol{\varphi}_m \hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}_1^T \boldsymbol{\varphi}_{\sigma_1} \hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}_r^T \boldsymbol{\varphi}_{\sigma_r} \hat{\mathbf{w}} + \varepsilon
\end{aligned} \tag{2-40}$$

where $\tilde{\mathbf{m}}^T \boldsymbol{\varphi}_m \hat{\mathbf{w}} = \hat{\mathbf{w}}^T \boldsymbol{\varphi}_m^T \tilde{\mathbf{m}}$, $\tilde{\boldsymbol{\sigma}}_1^T \boldsymbol{\varphi}_{\sigma_1} \hat{\mathbf{w}} = \hat{\mathbf{w}}^T \boldsymbol{\varphi}_{\sigma_1}^T \tilde{\boldsymbol{\sigma}}_1$, $\tilde{\boldsymbol{\sigma}}_r^T \boldsymbol{\varphi}_{\sigma_r} \hat{\mathbf{w}} = \hat{\mathbf{w}}^T \boldsymbol{\varphi}_{\sigma_r}^T \tilde{\boldsymbol{\sigma}}_r$, and the uncertain term $\varepsilon = \tilde{\mathbf{w}}^T \mathbf{h} + \tilde{\mathbf{w}}^T \tilde{\boldsymbol{\varphi}} + \Delta$. The higher-order term \mathbf{h} satisfies

$$\begin{aligned}
\|\mathbf{h}\| &= \left\| \tilde{\boldsymbol{\varphi}} - \boldsymbol{\varphi}_m^T \tilde{\mathbf{m}} - \boldsymbol{\varphi}_{\sigma_1}^T \tilde{\boldsymbol{\sigma}}_1 - \boldsymbol{\varphi}_{\sigma_r}^T \tilde{\boldsymbol{\sigma}}_r \right\| \\
&\leq \|\tilde{\boldsymbol{\varphi}}\| + \|\boldsymbol{\varphi}_m^T\| \|\tilde{\mathbf{m}}\| + \|\boldsymbol{\varphi}_{\sigma_1}^T\| \|\tilde{\boldsymbol{\sigma}}_1\| + \|\boldsymbol{\varphi}_{\sigma_r}^T\| \|\tilde{\boldsymbol{\sigma}}_r\| \\
&\leq \mathbf{c}_0 + \mathbf{c}_1 \|\tilde{\mathbf{m}}\| + \mathbf{c}_2 \|\tilde{\boldsymbol{\sigma}}_1\| + \mathbf{c}_3 \|\tilde{\boldsymbol{\sigma}}_r\|
\end{aligned} \tag{2-41}$$

where \mathbf{c}_0 , \mathbf{c}_1 , \mathbf{c}_2 , and \mathbf{c}_3 are positive constants satisfying $\|\tilde{\boldsymbol{\varphi}}\| \leq \mathbf{c}_0$, $\|\boldsymbol{\varphi}_m^T\| \leq \mathbf{c}_1$, $\|\boldsymbol{\varphi}_{\sigma_1}^T\| \leq \mathbf{c}_2$, $\|\boldsymbol{\varphi}_{\sigma_r}^T\| \leq \mathbf{c}_3$. The existence of \mathbf{c}_0 , \mathbf{c}_1 , \mathbf{c}_2 , and \mathbf{c}_3 is assured due to the fact that Gaussian function and its derivative are always bounded by constants. Moreover, $\tilde{\mathbf{w}}$, $\tilde{\mathbf{m}}$, $\tilde{\boldsymbol{\sigma}}_1$, and $\tilde{\boldsymbol{\sigma}}_r$ satisfy

$$\|\tilde{\mathbf{w}}\| = \|\mathbf{w}^* - \hat{\mathbf{w}}\| \leq \|\mathbf{w}^*\| + \|\hat{\mathbf{w}}\| \leq D_w + \|\hat{\mathbf{w}}\| \tag{2-42}$$

$$\|\tilde{\mathbf{m}}\| = \|\mathbf{m}^* - \hat{\mathbf{m}}\| \leq \|\mathbf{m}^*\| + \|\hat{\mathbf{m}}\| \leq D_m + \|\hat{\mathbf{m}}\| \tag{2-43}$$

$$\|\tilde{\boldsymbol{\sigma}}_1\| = \|\boldsymbol{\sigma}_1^* - \hat{\boldsymbol{\sigma}}_1\| \leq \|\boldsymbol{\sigma}_1^*\| + \|\hat{\boldsymbol{\sigma}}_1\| \leq D_{\sigma_1} + \|\hat{\boldsymbol{\sigma}}_1\| \tag{2-44}$$

$$\|\tilde{\boldsymbol{\sigma}}_r\| = \|\boldsymbol{\sigma}_r^* - \hat{\boldsymbol{\sigma}}_r\| \leq \|\boldsymbol{\sigma}_r^*\| + \|\hat{\boldsymbol{\sigma}}_r\| \leq D_{\sigma_r} + \|\hat{\boldsymbol{\sigma}}_r\|. \tag{2-45}$$

Next, the uncertain term ε is satisfied

$$\begin{aligned}
|\varepsilon| &= \left\| \tilde{\mathbf{w}}^T (\boldsymbol{\varphi}_m^T \tilde{\mathbf{m}} + \boldsymbol{\varphi}_{\sigma_1}^T \tilde{\boldsymbol{\sigma}}_1 + \boldsymbol{\varphi}_{\sigma_r}^T \tilde{\boldsymbol{\sigma}}_r + \mathbf{h}) + \hat{\mathbf{w}}^T \mathbf{h} + \Delta \right\| \\
&= \left\| \tilde{\mathbf{w}}^T \boldsymbol{\varphi}_m^T \tilde{\mathbf{m}} + \tilde{\mathbf{w}}^T \boldsymbol{\varphi}_{\sigma_1}^T \tilde{\boldsymbol{\sigma}}_1 + \tilde{\mathbf{w}}^T \boldsymbol{\varphi}_{\sigma_r}^T \tilde{\boldsymbol{\sigma}}_r + \mathbf{w}^{*T} \mathbf{h} + \Delta \right\|
\end{aligned}$$

$$\begin{aligned}
&\leq \mathbf{c}_1(D_w + \|\hat{\mathbf{w}}\|)(D_m + \|\hat{\mathbf{m}}\|) + \mathbf{c}_2(D_w + \|\hat{\mathbf{w}}\|)(D_{\sigma_1} + \|\hat{\boldsymbol{\sigma}}_1\|) + \mathbf{c}_3(D_w + \|\hat{\mathbf{w}}\|)(D_{\sigma_r} + \|\hat{\boldsymbol{\sigma}}_r\|) \\
&\quad + D_w[\mathbf{c}_0 + \mathbf{c}_1(D_m + \|\hat{\mathbf{m}}\|) + \mathbf{c}_2(D_{\sigma_1} + \|\hat{\boldsymbol{\sigma}}_1\|) + \mathbf{c}_3(D_{\sigma_r} + \|\hat{\boldsymbol{\sigma}}_r\|)] + \Delta^* \\
&= [\Theta_1, \Theta_2, \Theta_3, \Theta_4, \Theta_5, \Theta_6, \Theta_7, \Theta_8][1, \|\hat{\mathbf{w}}\|, \|\hat{\mathbf{m}}\|, \|\hat{\boldsymbol{\sigma}}_1\|, \|\hat{\boldsymbol{\sigma}}_r\|, \|\hat{\mathbf{m}}\|\|\hat{\mathbf{w}}\|, \|\hat{\boldsymbol{\sigma}}_1\|\|\hat{\mathbf{w}}\|, \|\hat{\boldsymbol{\sigma}}_r\|\|\hat{\mathbf{w}}\|]^T \\
&= \boldsymbol{\Theta}^T \boldsymbol{\Gamma} \tag{2-46}
\end{aligned}$$

where $\boldsymbol{\Theta} = [\Theta_1, \Theta_2, \Theta_3, \Theta_4, \Theta_5, \Theta_6, \Theta_7, \Theta_8]^T$, $\Theta_1 = (\mathbf{c}_0 + 2\mathbf{c}_1 D_m + 2\mathbf{c}_2 D_{\sigma_1} + 2\mathbf{c}_3 D_{\sigma_r})D_w + \Delta^*$, $\Theta_2 = \mathbf{c}_1 D_m + \mathbf{c}_2 D_{\sigma_1} + \mathbf{c}_3 D_{\sigma_r}$, $\Theta_3 = 2\mathbf{c}_1 D_w$, $\Theta_4 = 2\mathbf{c}_2 D_w$, $\Theta_5 = 2\mathbf{c}_3 D_w$, $\Theta_6 = \mathbf{c}_1$, $\Theta_7 = \mathbf{c}_2$, $\Theta_8 = \mathbf{c}_3$ and $\boldsymbol{\Gamma} = [1, \|\hat{\mathbf{w}}\|, \|\hat{\mathbf{m}}\|, \|\hat{\boldsymbol{\sigma}}_1\|, \|\hat{\boldsymbol{\sigma}}_r\|, \|\hat{\mathbf{m}}\|\|\hat{\mathbf{w}}\|, \|\hat{\boldsymbol{\sigma}}_1\|\|\hat{\mathbf{w}}\|, \|\hat{\boldsymbol{\sigma}}_r\|\|\hat{\mathbf{w}}\|]^T$. Since $\boldsymbol{\Theta}$ is a bounded vector, if $\boldsymbol{\Gamma}$ can be guaranteed to be bounded, the uncertain term ε is bounded. The analysis of boundness of $\boldsymbol{\Gamma}$ will be given in the later section.

2.4 ASAFNC Design

By using (2-40), (2-7) can be rewritten as

$$\dot{s} = \tilde{\mathbf{w}}^T \hat{\boldsymbol{\phi}} + \tilde{\mathbf{m}}^T \boldsymbol{\phi}_m \hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}_1^T \boldsymbol{\phi}_{\sigma_1} \hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}_r^T \boldsymbol{\phi}_{\sigma_r} \hat{\mathbf{w}} + \varepsilon - u_{rb}. \tag{2-47}$$

If ε exists, consider a specified L_2 tracking performance [46, 47]

$$\begin{aligned}
\int_0^T s^2(t)dt &\leq s^2(0) + \delta^2 \int_0^T \varepsilon^2(t)dt + \frac{1}{\eta_w} \tilde{\mathbf{w}}^T(0) \tilde{\mathbf{w}}(0) + \frac{1}{\eta_m} \tilde{\mathbf{m}}^T(0) \tilde{\mathbf{m}}(0) \\
&\quad + \frac{1}{\eta_{\sigma_1}} \tilde{\boldsymbol{\sigma}}_1^T(0) \tilde{\boldsymbol{\sigma}}_1(0) + \frac{1}{\eta_{\sigma_r}} \tilde{\boldsymbol{\sigma}}_r^T(0) \tilde{\boldsymbol{\sigma}}_r(0) \tag{2-48}
\end{aligned}$$

where η_w , η_m , η_{σ_1} , and η_{σ_r} are the positive-constant learning rates, and δ is a prescribed attenuation constant. If the system starts with initial conditions $s(0) = 0$, $\tilde{\mathbf{w}}(0) = 0$, $\tilde{\mathbf{m}}(0) = 0$, $\tilde{\boldsymbol{\sigma}}_1(0) = 0$, and $\tilde{\boldsymbol{\sigma}}_r(0) = 0$, the L_2 tracking performance in (2-48) can be rewritten as

$$\sup_{\varepsilon \in L_2[0, T]} \frac{\|s\|}{\|\varepsilon\|} \leq \delta \tag{2-49}$$

where $\|s\|^2 = \int_0^T s^2(t)dt$ and $\|\varepsilon\|^2 = \int_0^T \varepsilon^2(t)dt$. If $\delta = \infty$, this is the case of minimum error tracking control without disturbance attenuation. To determine the adaptive laws of the parameters of ASAFNC appropriately and guarantee the closed-loop system stability, the Lyapunov function candidate is defined as

$$V = \frac{1}{2}s^2 + \frac{\tilde{\mathbf{w}}^T \tilde{\mathbf{w}}}{2\eta_w} + \frac{\tilde{\mathbf{m}}^T \tilde{\mathbf{m}}}{2\eta_m} + \frac{\tilde{\boldsymbol{\sigma}}_1^T \tilde{\boldsymbol{\sigma}}_1}{2\eta_{\sigma_1}} + \frac{\tilde{\boldsymbol{\sigma}}_r^T \tilde{\boldsymbol{\sigma}}_r}{2\eta_{\sigma_r}}. \quad (2-50)$$

Differentiating (2-50) with respect to time and using (2-47) yield

$$\begin{aligned} \dot{V} &= s\dot{s} + \frac{\tilde{\mathbf{w}}^T \dot{\tilde{\mathbf{w}}}}{\eta_w} + \frac{\tilde{\mathbf{m}}^T \dot{\tilde{\mathbf{m}}}}{\eta_m} + \frac{\tilde{\boldsymbol{\sigma}}_1^T \dot{\tilde{\boldsymbol{\sigma}}}_1}{\eta_{\sigma_1}} + \frac{\tilde{\boldsymbol{\sigma}}_r^T \dot{\tilde{\boldsymbol{\sigma}}}_r}{\eta_{\sigma_r}} \\ &= s(\tilde{\mathbf{w}}^T \dot{\hat{\boldsymbol{\phi}}} + \tilde{\mathbf{m}}^T \boldsymbol{\varphi}_m \dot{\hat{\mathbf{w}}} + \tilde{\boldsymbol{\sigma}}_1^T \boldsymbol{\varphi}_{\sigma_1} \dot{\hat{\mathbf{w}}} + \tilde{\boldsymbol{\sigma}}_r^T \boldsymbol{\varphi}_{\sigma_r} \dot{\hat{\mathbf{w}}} + \varepsilon - u_{rb}) + \frac{\tilde{\mathbf{w}}^T \dot{\tilde{\mathbf{w}}}}{\eta_w} + \frac{\tilde{\mathbf{m}}^T \dot{\tilde{\mathbf{m}}}}{\eta_m} + \frac{\tilde{\boldsymbol{\sigma}}_1^T \dot{\tilde{\boldsymbol{\sigma}}}_1}{\eta_{\sigma_1}} + \frac{\tilde{\boldsymbol{\sigma}}_r^T \dot{\tilde{\boldsymbol{\sigma}}}_r}{\eta_{\sigma_r}} \\ &= \tilde{\mathbf{w}}^T (s\dot{\hat{\boldsymbol{\phi}}} + \frac{1}{\eta_w} \dot{\tilde{\mathbf{w}}}) + \tilde{\mathbf{m}}^T (s\boldsymbol{\varphi}_m \dot{\hat{\mathbf{w}}} + \frac{1}{\eta_m} \dot{\tilde{\mathbf{m}}}) + \tilde{\boldsymbol{\sigma}}_1^T (s\boldsymbol{\varphi}_{\sigma_1} \dot{\hat{\mathbf{w}}} + \frac{1}{\eta_{\sigma_1}} \dot{\tilde{\boldsymbol{\sigma}}}_1) \\ &\quad + \tilde{\boldsymbol{\sigma}}_r^T (s\boldsymbol{\varphi}_{\sigma_r} \dot{\hat{\mathbf{w}}} + \frac{1}{\eta_{\sigma_r}} \dot{\tilde{\boldsymbol{\sigma}}}_r) + s(\varepsilon - u_{rb}) \end{aligned} \quad (2-51)$$

Choose the adaptive laws as

$$\dot{\tilde{\mathbf{w}}} = -\dot{\hat{\mathbf{w}}} = -\eta_w s \hat{\boldsymbol{\phi}} \quad (2-52)$$

$$\dot{\tilde{\mathbf{m}}} = -\dot{\hat{\mathbf{m}}} = -\eta_m s \boldsymbol{\varphi}_m \hat{\mathbf{w}} \quad (2-53)$$

$$\dot{\tilde{\boldsymbol{\sigma}}}_1 = -\dot{\hat{\boldsymbol{\sigma}}}_1 = -\eta_{\sigma_1} s \boldsymbol{\varphi}_{\sigma_1} \hat{\mathbf{w}} \quad (2-54)$$

$$\dot{\tilde{\boldsymbol{\sigma}}}_r = -\dot{\hat{\boldsymbol{\sigma}}}_r = -\eta_{\sigma_r} s \boldsymbol{\varphi}_{\sigma_r} \hat{\mathbf{w}} \quad (2-55)$$

and the robust controller is designed as

$$u_{rb} = \frac{\delta^2 + 1}{2\delta^2} s. \quad (2-56)$$

Thus, equation (2-51) can be rewritten as

$$\begin{aligned} \dot{V} &= s(\varepsilon - \frac{\delta^2 + 1}{2\delta^2} s) \\ &= s\varepsilon - \frac{s^2}{2} - \frac{s^2}{2\delta^2} \\ &= -\frac{s^2}{2} - \frac{1}{2}(\frac{s}{\delta} - \varepsilon\delta)^2 + \frac{1}{2}\varepsilon^2\delta^2 \\ &\leq -\frac{1}{2}s^2 + \frac{1}{2}\varepsilon^2\delta^2. \end{aligned} \quad (2-57)$$

Assume $\varepsilon \in L_2[0, T]$, $\forall T \in [0, \infty)$. Integrating the above equation from $t=0$ to $t=T$ yields

$$V(T) - V(0) \leq -\frac{1}{2} \int_0^T s^2 dt + \frac{1}{2} \delta^2 \int_0^T \varepsilon^2 dt. \quad (2-58)$$

Since $V(t) \geq 0$, we can arrange (2-58) as follows

$$\frac{1}{2} \int_0^T s^2 dt \leq V(0) + \frac{1}{2} \delta^2 \int_0^T \varepsilon^2 dt \quad (2-59)$$

which is equivalent to inequality (2-48), i.e., L_2 tracking performance. Assume $\varepsilon \in L_2$, then the sliding surface s will converge to a certain small boundary. It is implied that the tracking error e will also converge to a certain small boundary [47].

2.5 Boundary Analysis Using Projection Algorithm

Although the stability of ASAFNC can be guaranteed, the parameters $\hat{\mathbf{w}}$, $\hat{\mathbf{m}}$, $\hat{\sigma}_1$, and $\hat{\sigma}_r$ cannot be guaranteed within a desired bound value by using the adaptive laws (2-52)-(2-55). According to the projection algorithm [44, 48, 49], the adaptive laws can be modified as follows. The adaptive law of weight is

$$\dot{\hat{\mathbf{w}}} = \begin{cases} \eta_w s \hat{\boldsymbol{\phi}}, & \text{if } \|\hat{\mathbf{w}}\| < D_w \text{ or } (\|\hat{\mathbf{w}}\| = D_w \text{ and } s \hat{\mathbf{w}}^T \hat{\boldsymbol{\phi}} \leq 0) \\ \mathbf{Pr}(\eta_w s \hat{\boldsymbol{\phi}}), & \text{if } (\|\hat{\mathbf{w}}\| = D_w \text{ and } s \hat{\mathbf{w}}^T \hat{\boldsymbol{\phi}} > 0) \end{cases} \quad (2-60)$$

where the projection operator is given as

$$\mathbf{Pr}(\eta_w s \hat{\boldsymbol{\phi}}) = \eta_w s \hat{\boldsymbol{\phi}} - \eta_w s \frac{\hat{\mathbf{w}}^T \hat{\boldsymbol{\phi}}}{\|\hat{\mathbf{w}}\|^2} \hat{\mathbf{w}}. \quad (2-61)$$

The adaptive law of mean of asymmetric membership function is

$$\dot{\hat{\mathbf{m}}} = \begin{cases} \eta_m s \boldsymbol{\phi}_m \hat{\mathbf{w}}, & \text{if } \|\hat{\mathbf{m}}\| < D_m \text{ or } (\|\hat{\mathbf{m}}\| = D_m \text{ and } s \hat{\mathbf{m}}^T \boldsymbol{\phi}_m \hat{\mathbf{w}} \leq 0) \\ \mathbf{Pr}(\eta_m s \boldsymbol{\phi}_m \hat{\mathbf{w}}), & \text{if } (\|\hat{\mathbf{m}}\| = D_m \text{ and } s \hat{\mathbf{m}}^T \boldsymbol{\phi}_m \hat{\mathbf{w}} > 0) \end{cases} \quad (2-62)$$

where the projection operator is given as

$$\mathbf{Pr}(\eta_m s \boldsymbol{\phi}_m \hat{\mathbf{w}}) = \eta_m s \boldsymbol{\phi}_m \hat{\mathbf{w}} - \eta_m s \frac{\hat{\mathbf{m}}^T \boldsymbol{\phi}_m \hat{\mathbf{w}}}{\|\hat{\mathbf{m}}\|^2} \hat{\mathbf{m}}. \quad (2-63)$$

The adaptive law of left-side variance of asymmetric membership function is

$$\dot{\hat{\sigma}}_1 = \begin{cases} \eta_{\sigma_1} s \boldsymbol{\phi}_{\sigma_1} \hat{\mathbf{w}}, & \text{if } \|\hat{\sigma}_1\| < D_{\sigma_1} \text{ or } (\|\hat{\sigma}_1\| = D_{\sigma_1} \text{ and } s \hat{\sigma}_1^T \boldsymbol{\phi}_{\sigma_1} \hat{\mathbf{w}} \leq 0) \\ \mathbf{Pr}(\eta_{\sigma_1} s \boldsymbol{\phi}_{\sigma_1} \hat{\mathbf{w}}), & \text{if } (\|\hat{\sigma}_1\| = D_{\sigma_1} \text{ and } s \hat{\sigma}_1^T \boldsymbol{\phi}_{\sigma_1} \hat{\mathbf{w}} > 0) \end{cases} \quad (2-64)$$

where the projection operator is given as

$$\mathbf{Pr}(\eta_{\sigma_1} s \boldsymbol{\phi}_{\sigma_1} \hat{\mathbf{w}}) = \eta_{\sigma_1} s \boldsymbol{\phi}_{\sigma_1} \hat{\mathbf{w}} - \eta_{\sigma_1} s \frac{\hat{\sigma}_1^T \boldsymbol{\phi}_{\sigma_1} \hat{\mathbf{w}}}{\|\hat{\sigma}_1\|^2} \hat{\sigma}_1. \quad (2-65)$$

The adaptive law of right-side variance of asymmetric membership function is

$$\dot{\hat{\sigma}}_r = \begin{cases} \eta_{\sigma_r} s \varphi_{\sigma_r} \hat{\mathbf{w}}, & \text{if } \|\hat{\sigma}_r\| < D_{\sigma_r} \text{ or } (\|\hat{\sigma}_r\| = D_{\sigma_r} \text{ and } s \hat{\sigma}_r^T \varphi_{\sigma_r} \hat{\mathbf{w}} \leq 0) \\ \mathbf{Pr}(\eta_{\sigma_r} s \varphi_{\sigma_r} \hat{\mathbf{w}}), & \text{if } (\|\hat{\sigma}_r\| = D_{\sigma_r} \text{ and } s \hat{\sigma}_r^T \varphi_{\sigma_r} \hat{\mathbf{w}} > 0) \end{cases} \quad (2-66)$$

where the projection operator is given as

$$\mathbf{Pr}(\eta_{\sigma_r} s \varphi_{\sigma_r} \hat{\mathbf{w}}) = \eta_{\sigma_r} s \varphi_{\sigma_r} \hat{\mathbf{w}} - \eta_{\sigma_r} s \frac{\hat{\sigma}_r^T \varphi_{\sigma_r} \hat{\mathbf{w}}}{\|\hat{\sigma}_r\|^2} \hat{\sigma}_r. \quad (2-67)$$

Then, let the initial values satisfy $\hat{\mathbf{w}}(0) \in \Omega_w$, $\hat{\mathbf{m}}(0) \in \Omega_m$, $\hat{\sigma}_1(0) \in \Omega_{\sigma_1}$, and $\hat{\sigma}_r(0) \in \Omega_{\sigma_r}$, the conditions $\hat{\mathbf{w}}(t) \in \Omega_w$, $\hat{\mathbf{m}}(t) \in \Omega_m$, $\hat{\sigma}_1(t) \in \Omega_{\sigma_1}$, and $\hat{\sigma}_r(t) \in \Omega_{\sigma_r}$ can be kept for all $t \geq 0$, i.e., $\|\hat{\mathbf{w}}\|$, $\|\hat{\mathbf{m}}\|$, $\|\hat{\sigma}_1\|$, and $\|\hat{\sigma}_r\|$ are all bounded.

Thus, the fact that the uncertain term ε is bounded can be guaranteed by the modified adaptive laws (2-60), (2-62), (2-64), and (2-66). The following description states that the analytic result of stability is the same as (2-59) by re-selecting the adaptive laws (2-60), (2-62), (2-64), and (2-66). First, define some useful variables as

$$J_w = \tilde{\mathbf{w}}^T (s \hat{\boldsymbol{\phi}} + \frac{1}{\eta_w} \dot{\tilde{\mathbf{w}}}) \quad (2-68)$$

$$J_m = \tilde{\mathbf{m}}^T (s \varphi_m \hat{\mathbf{w}} + \frac{1}{\eta_m} \dot{\tilde{\mathbf{m}}}) \quad (2-69)$$

$$J_{\sigma_1} = \tilde{\sigma}_1^T (s \varphi_{\sigma_1} \hat{\mathbf{w}} + \frac{1}{\eta_{\sigma_1}} \dot{\tilde{\sigma}}_1) \quad (2-70)$$

and

$$J_{\sigma_r} = \tilde{\sigma}_r^T (s \varphi_{\sigma_r} \hat{\mathbf{w}} + \frac{1}{\eta_{\sigma_r}} \dot{\tilde{\sigma}}_r). \quad (2-71)$$

Then, the derivative of Lyapunov function shown in (2-51) can be rewritten as

$$\dot{V} = J_w + J_m + J_{\sigma_1} + J_{\sigma_r} + s(\varepsilon - u_{rc}). \quad (2-72)$$

By using (2-60), $J_w = 0$ for $[\|\hat{\mathbf{w}}\| < D_w \text{ or } (\|\hat{\mathbf{w}}\| = D_w \text{ and } s \hat{\mathbf{w}}^T \hat{\boldsymbol{\phi}} \leq 0)]$ can be obtained.

For $(\|\hat{\mathbf{w}}\| = D_w \text{ and } s \hat{\mathbf{w}}^T \hat{\boldsymbol{\phi}} > 0)$,

$$J_w = s \frac{\tilde{\mathbf{w}}^T \hat{\mathbf{w}}}{\|\hat{\mathbf{w}}\|^2} \hat{\mathbf{w}}^T \hat{\boldsymbol{\phi}}. \quad (2-73)$$

can be obtained. Because \mathbf{w}^* belongs to the constraint set Ω_w , we have $\|\hat{\mathbf{w}}\| = D_w \geq \mathbf{w}^*$.

Using this fact, we obtain $\tilde{\mathbf{w}}^T \hat{\mathbf{w}} = \frac{1}{2} (\|\mathbf{w}^*\|^2 - \|\hat{\mathbf{w}}\|^2 - \|\tilde{\mathbf{w}}\|^2) \leq 0$. Thus, equation (2-73) can be

rewritten as

$$J_w = \frac{s}{2} \frac{(\|\mathbf{w}^*\|^2 - \|\hat{\mathbf{w}}\|^2 - \|\tilde{\mathbf{w}}\|^2)}{\|\hat{\mathbf{w}}\|^2} \hat{\mathbf{w}}^T \hat{\boldsymbol{\phi}} \leq 0. \quad (2-74)$$

Similarly, by using (2-62), $J_m = 0$ can be obtained for [$\|\hat{\mathbf{m}}\| < D_m$ or ($\|\hat{\mathbf{m}}\| = D_m$ and $s\hat{\mathbf{m}}^T \boldsymbol{\phi}_m \hat{\mathbf{w}} \leq 0$)]; and for ($\|\hat{\mathbf{m}}\| = D_m$ and $s\hat{\mathbf{m}}^T \boldsymbol{\phi}_m \hat{\mathbf{w}} > 0$), the inequality

$$J_m = \frac{s}{2} \frac{(\|\mathbf{m}^*\|^2 - \|\hat{\mathbf{m}}\|^2 - \|\tilde{\mathbf{m}}\|^2)}{\|\hat{\mathbf{m}}\|^2} \hat{\mathbf{m}}^T \boldsymbol{\phi}_m \hat{\mathbf{w}} \leq 0 \quad (2-75)$$

can be obtained. By using (2-64), we obtain $J_{\sigma_1} = 0$ for [$\|\hat{\boldsymbol{\sigma}}_1\| < D_{\sigma_1}$ or ($\|\hat{\boldsymbol{\sigma}}_1\| = D_{\sigma_1}$ and $s\hat{\boldsymbol{\sigma}}_1^T \boldsymbol{\phi}_{\sigma_1} \hat{\mathbf{w}} \leq 0$)]; and for ($\|\hat{\boldsymbol{\sigma}}_1\| = D_{\sigma_1}$ and $s\hat{\boldsymbol{\sigma}}_1^T \boldsymbol{\phi}_{\sigma_1} \hat{\mathbf{w}} > 0$), the inequality

$$J_{\sigma_1} = \frac{s}{2} \frac{(\|\boldsymbol{\sigma}_1^*\|^2 - \|\hat{\boldsymbol{\sigma}}_1\|^2 - \|\tilde{\boldsymbol{\sigma}}_1\|^2)}{\|\hat{\boldsymbol{\sigma}}_1\|^2} \hat{\boldsymbol{\sigma}}_1^T \boldsymbol{\phi}_{\sigma_1} \hat{\mathbf{w}} \leq 0 \quad (2-76)$$

can be obtained. By using (2-66), $J_{\sigma_r} = 0$ for [$\|\hat{\boldsymbol{\sigma}}_r\| < D_{\sigma_r}$ or ($\|\hat{\boldsymbol{\sigma}}_r\| = D_{\sigma_r}$ and $s\hat{\boldsymbol{\sigma}}_r^T \boldsymbol{\phi}_{\sigma_r} \hat{\mathbf{w}} \leq 0$)]; and for ($\|\hat{\boldsymbol{\sigma}}_r\| = D_{\sigma_r}$ and $s\hat{\boldsymbol{\sigma}}_r^T \boldsymbol{\phi}_{\sigma_r} \hat{\mathbf{w}} > 0$), the inequality

$$J_{\sigma_r} = \frac{s}{2} \frac{(\|\boldsymbol{\sigma}_r^*\|^2 - \|\hat{\boldsymbol{\sigma}}_r\|^2 - \|\tilde{\boldsymbol{\sigma}}_r\|^2)}{\|\hat{\boldsymbol{\sigma}}_r\|^2} \hat{\boldsymbol{\sigma}}_r^T \boldsymbol{\phi}_{\sigma_r} \hat{\mathbf{w}} \leq 0 \quad (2-77)$$

can be obtained. Hence, for any possible condition occurs in (2-60), (2-62), (2-64), and (2-66), the conditions $J_w \leq 0$, $J_m \leq 0$, $J_{\sigma_1} \leq 0$, and $J_{\sigma_r} \leq 0$ can be satisfied. Then, (2-72) can be reorganized as

$$\begin{aligned} \dot{V} &= J_w + J_m + J_{\sigma_1} + J_{\sigma_r} + s(\varepsilon - u_{rb}) \\ &\leq s(\varepsilon - u_{rb}). \end{aligned} \quad (2-78)$$

By substituting the robust controller (2-56), (2-78) can be rewritten as

$$\begin{aligned} \dot{V} &\leq s\left(\varepsilon - \frac{\delta^2 + 1}{2\delta^2} s\right) \\ &= -\frac{s^2}{2} - \frac{1}{2} \left(\frac{s}{\delta} - \varepsilon\delta\right)^2 + \frac{1}{2} \varepsilon^2 \delta^2 \\ &\leq -\frac{1}{2} s^2 + \frac{1}{2} \varepsilon^2 \delta^2. \end{aligned} \quad (2-79)$$

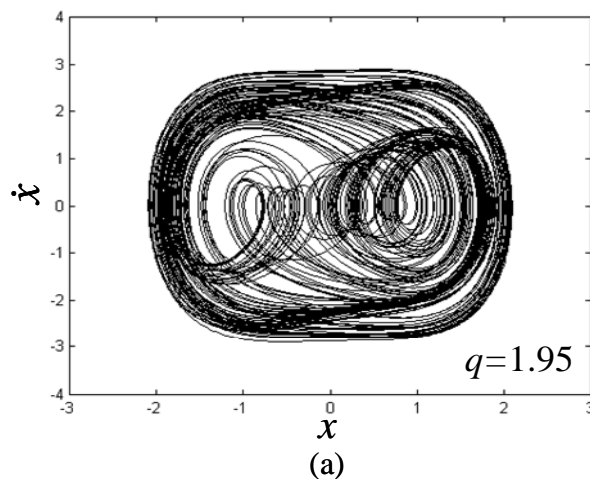
Using the same discussion in the *section 2.4*, the stability of the system with the projection algorithm can also be guaranteed.

2.6 Simulation Results

In this section, the proposed ASAFNC is applied to a second-order chaotic dynamics system to verify its effectiveness. This scheme emphasizes that the parameter and network structure of the SFNN can be tuned online by the proposed algorithm. Consider a second-order chaotic dynamics system such as the Duffing's equation describing a special nonlinear circuit or a pendulum moving in a viscous medium as follows [46]

$$\ddot{x} = f(\mathbf{x}) + u \quad (2-80)$$

where $f(\mathbf{x}) = -p\dot{x} - p_1x - p_2x^3 + q \cos(\omega t)$ is the system dynamics, t is the time variable, ω is the frequency, u is the control force, and p , p_1 , p_2 , and q are real constants. The solutions of (2-80) may exhibit periodic depending on the choice of these constants, i.e., it is almost periodic and chaotic behavior. The open-loop system behavior, i.e., $u = 0$, is simulated with $p=0.4$, $p_1 = -1.1$, $p_2 = 1.0$, and $\omega=1.8$ for observing the chaotic unpredictable behavior. The phase plane plots with an initial condition point $(0, 0)$ are shown in Figs. 2-5(a) and 2-5(b) for $q = 1.95$ and $q = 7.00$, respectively. The uncontrolled chaotic system has different trajectories for different values of q . To illustrate the effectiveness of the proposed design method, a comparison among a fix-structure AFNC using symmetric Gaussian membership functions [50], a fix-structure AFNC using asymmetric Gaussian membership functions [51], and the proposed ASAFNC is made.



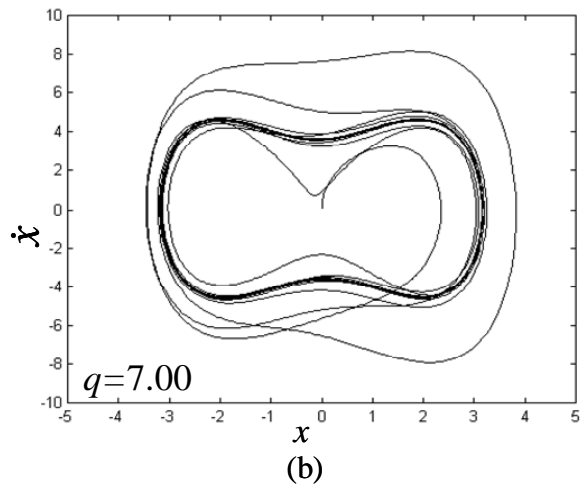
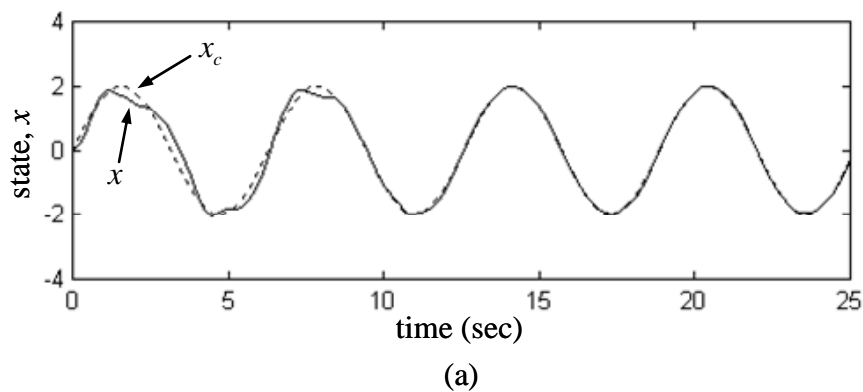
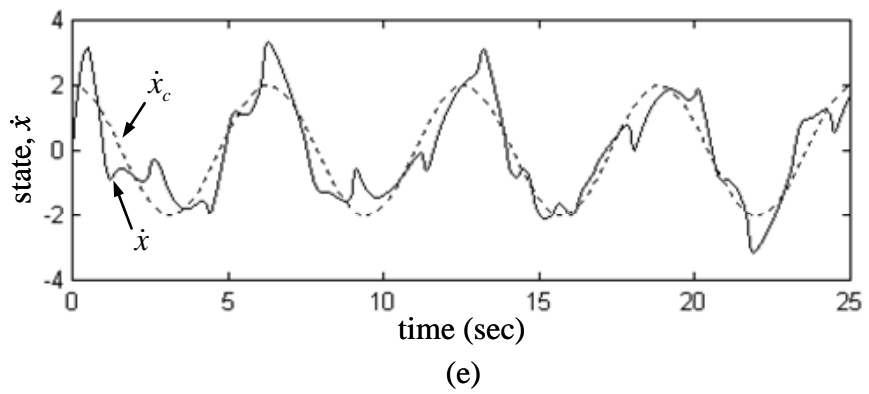
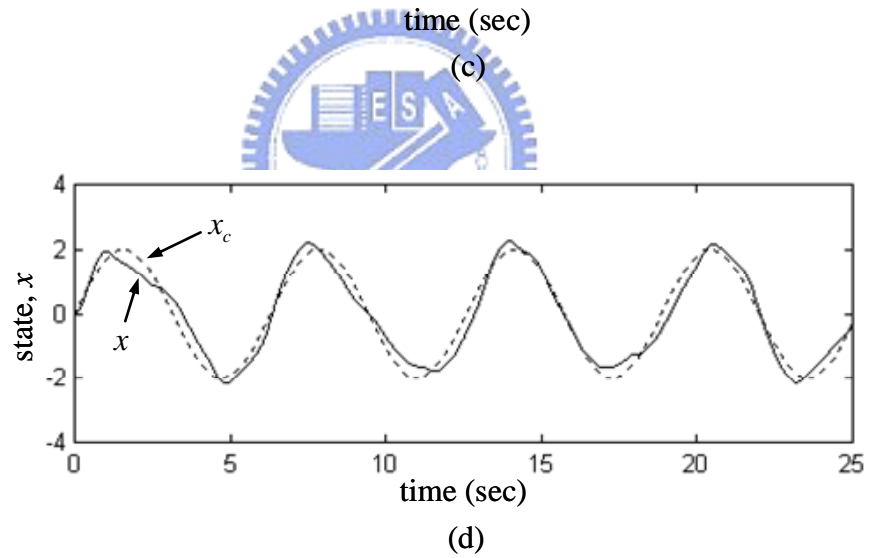
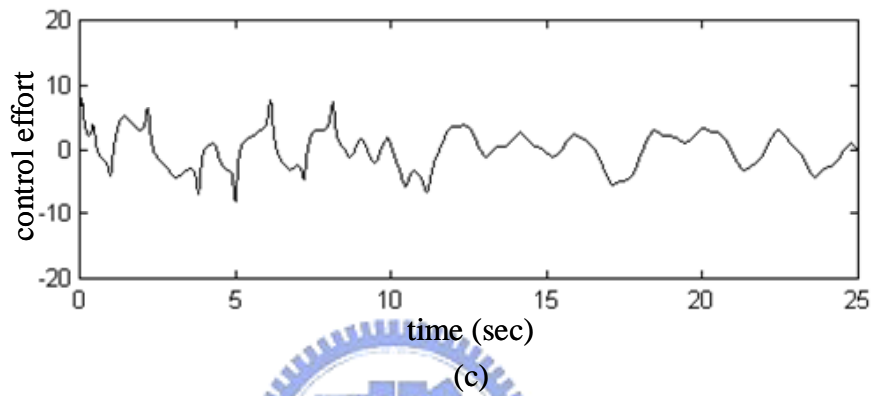
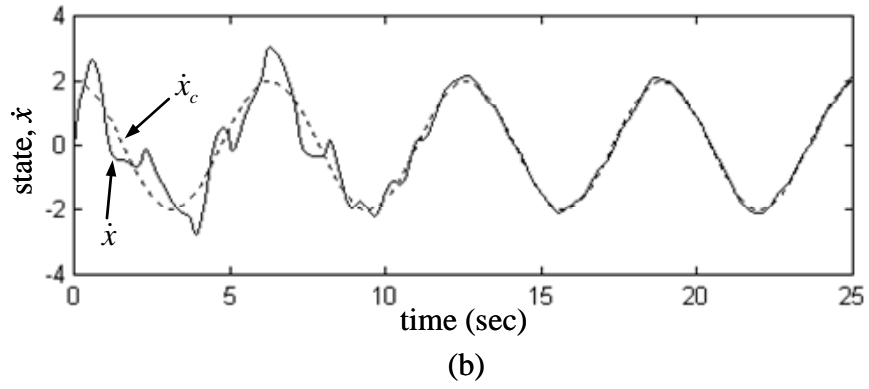


Fig. 2-5. Phase plane of uncontrolled chaotic dynamics system.

2.6.1 Comparison with AFNC

The simulation results of fix-structure AFNC using 3 symmetric membership functions are shown in Fig. 2-6. The tracking responses of state x are shown in Figs. 2-6(a) and 2-6(d); the tracking responses of state \dot{x} are shown in Figs. 2-6(b) and 2-6(e); and the associated control efforts are shown Figs. 2-6(c) and 2-6(f) for $q = 1.95$ and $q = 7.00$, respectively. The simulation results show that the tracking responses decline when membership functions are selected insufficiently.





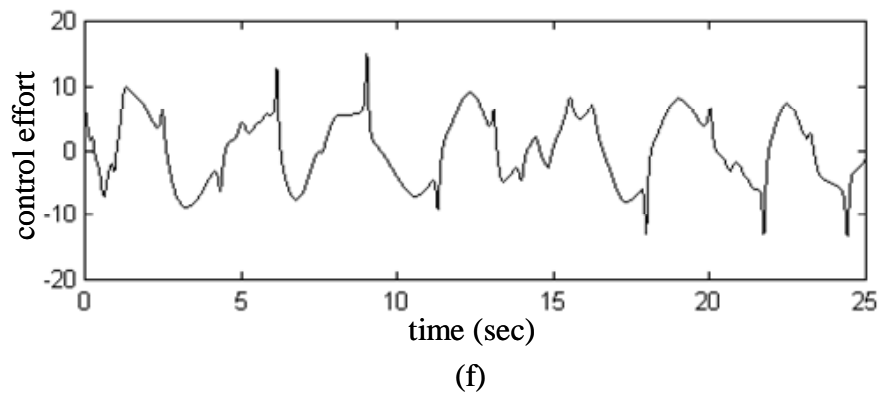
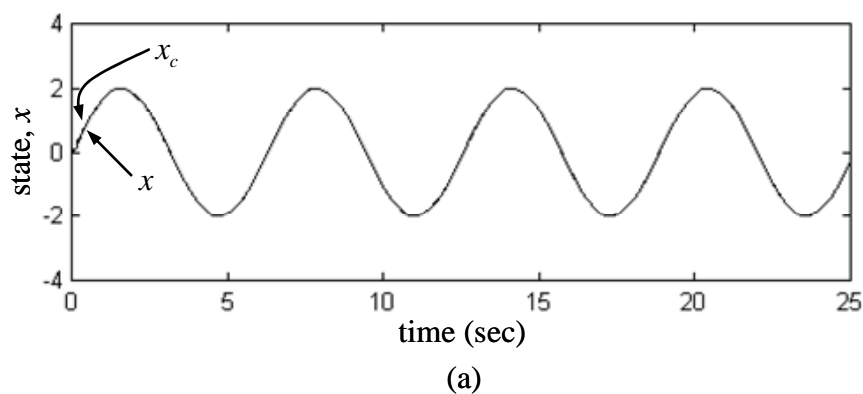
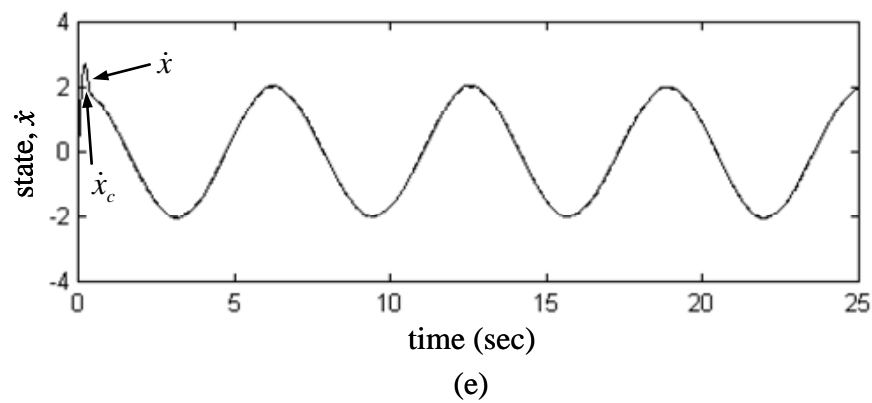
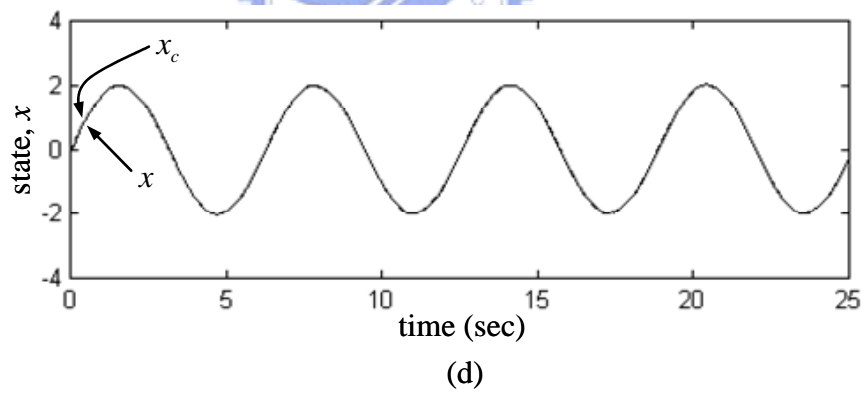
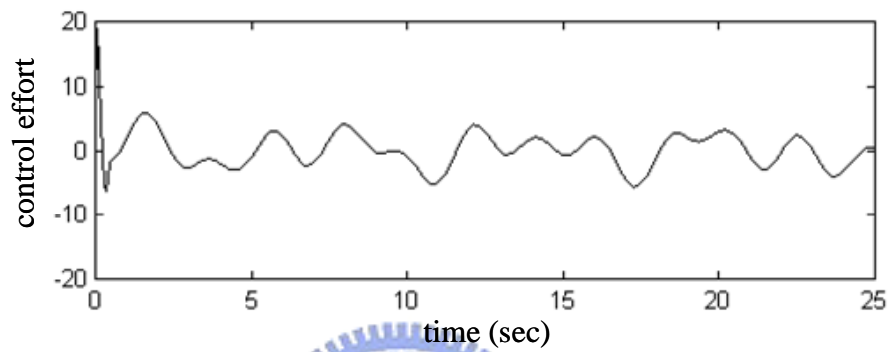
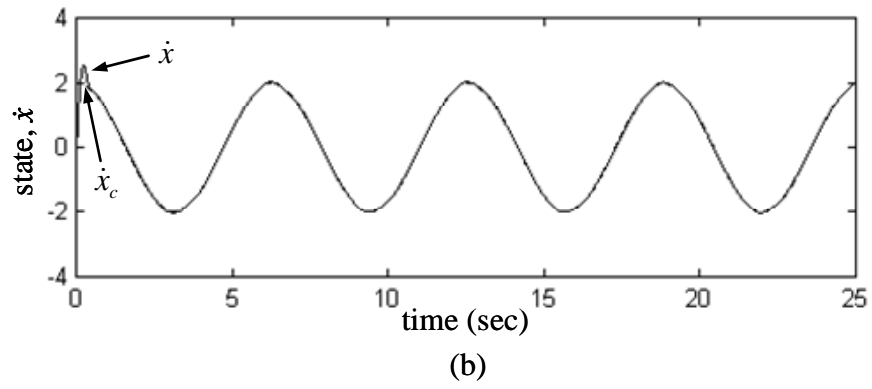


Fig. 2-6. Simulation results of AFNC using 3 symmetric membership functions.

Next, the simulation results of fix-structure AFNC using 20 symmetric membership functions are shown in Fig. 2-7. The tracking responses of state x are shown in Figs. 2-7(a) and 2-7(d); the tracking responses of state \dot{x} are shown in Figs. 2-7(b) and 2-7(e); and the associated control efforts are shown Figs. 2-7(c) and 2-7(f) for $q = 1.95$ and $q = 7.00$, respectively. The simulation results show that the favorable tracking performance can achieve; however, the computation load is heavy. These results demonstrate the fact that it is difficult to consider the balance between the rule number and the desired performance.





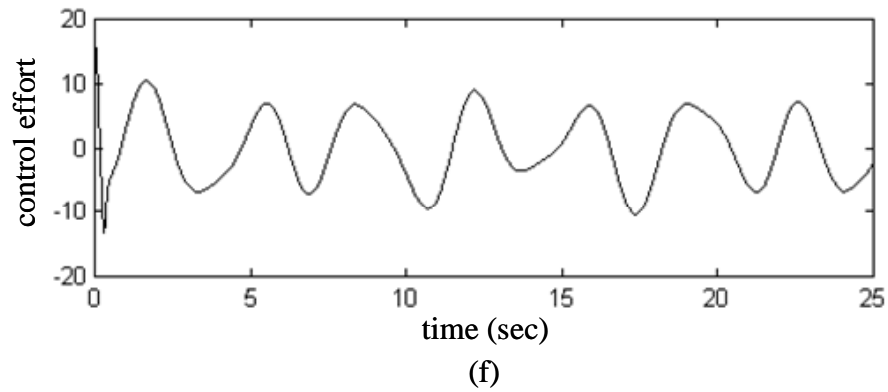
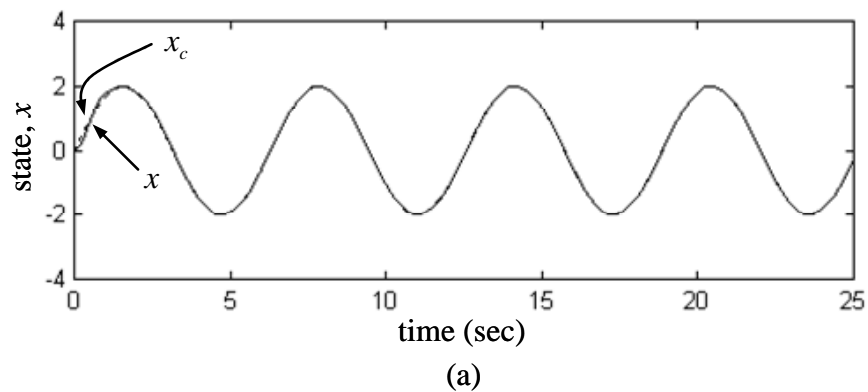
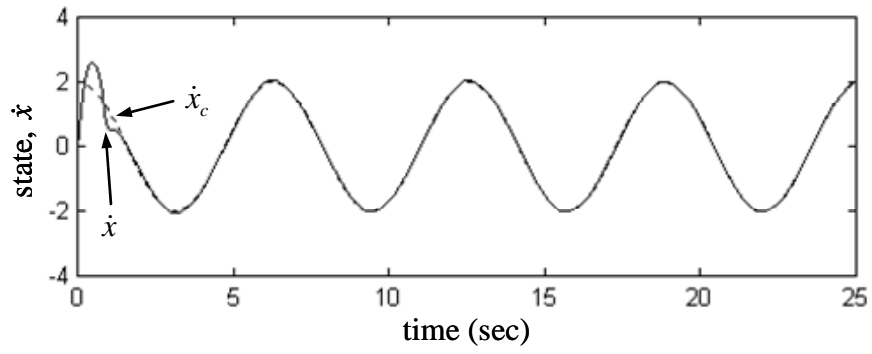


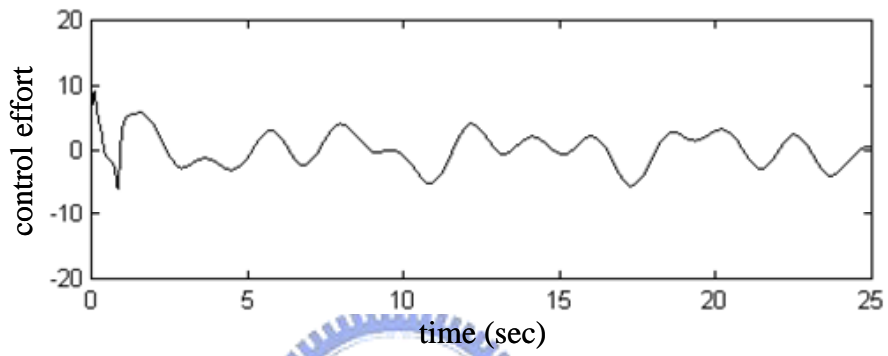
Fig. 2-7. Simulation results of AFNC using 20 symmetric membership functions.

To show that the learning capability of neural network can be upgraded as using the asymmetric Gaussian membership functions, the fix-structure AFNC using asymmetric Gaussian membership functions is applied to chaotic dynamics system again. The simulation results of fix-structure AFNC using 3 asymmetric membership functions are shown in Fig. 2-8. The tracking responses of state x are shown in Figs. 2-8(a) and 2-8(d); the tracking responses of state \dot{x} are shown in Figs. 2-8(b) and 2-8(e); and the associated control efforts are shown Figs. 2-8(c) and 2-8(f) for $q = 1.95$ and $q = 7.00$, respectively. The simulation results show that the favorable tracking performance can be achieved.

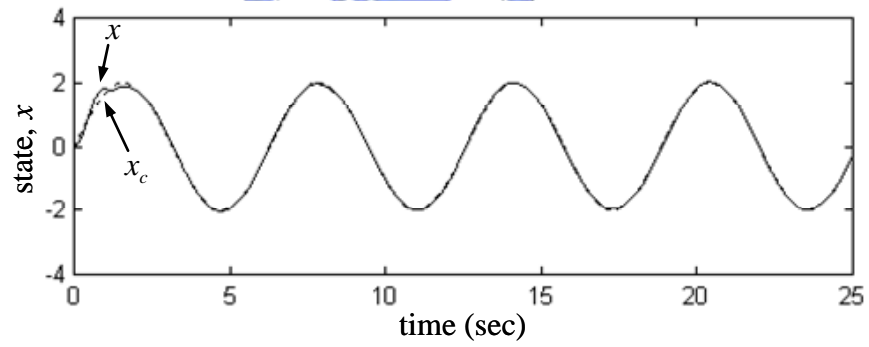




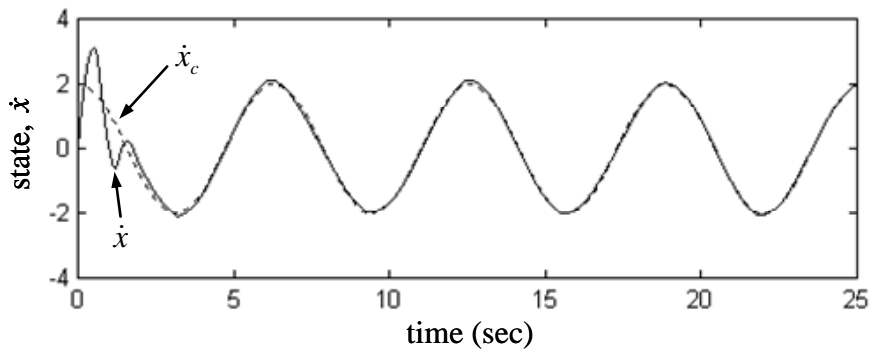
(b)



(c)



(d)



(e)

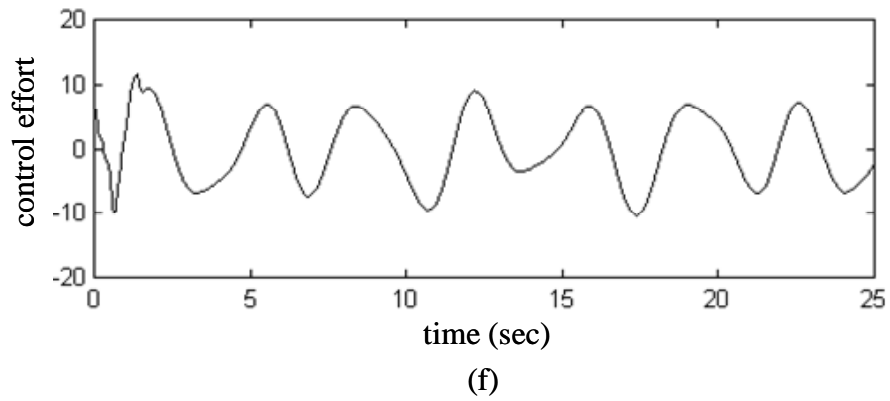
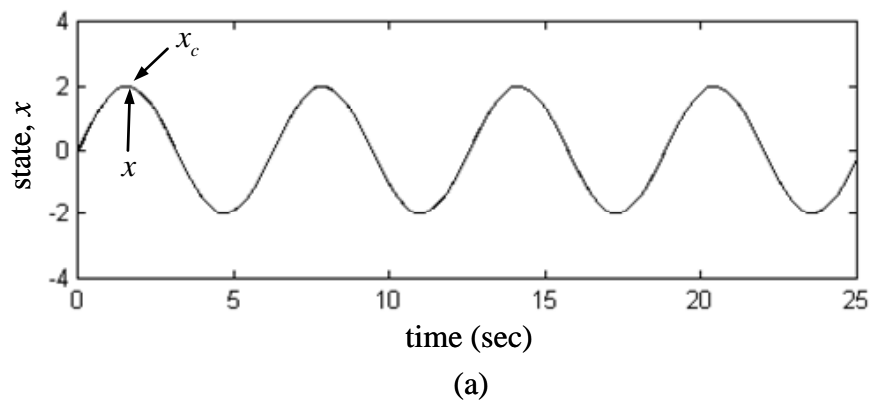
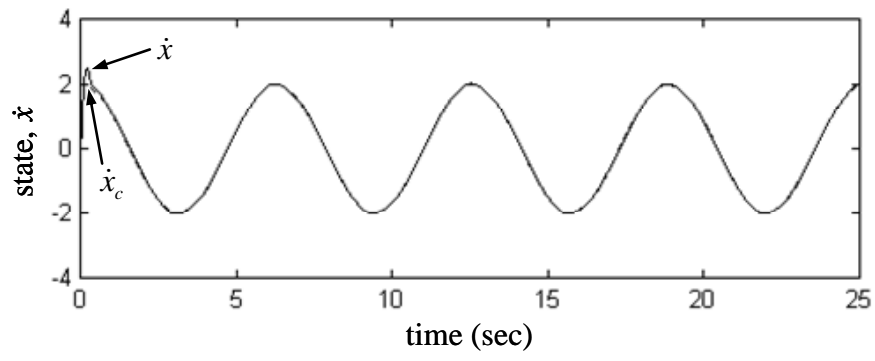


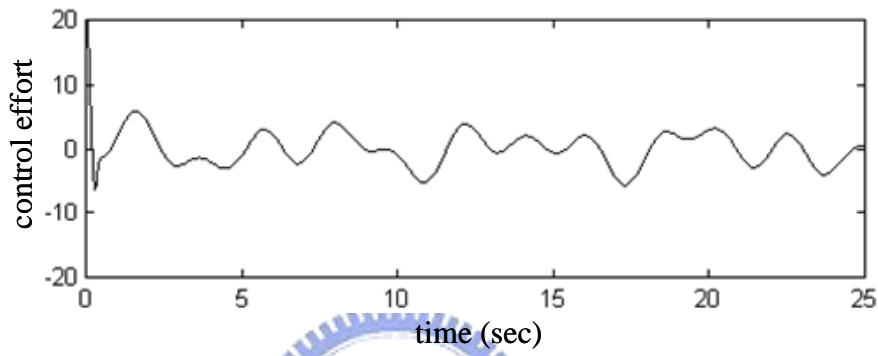
Fig. 2-8. Simulation results of AFNC using 3 asymmetric membership functions.

Next, the simulation results of fix-structure AFNC using 20 asymmetric membership functions are shown in Fig. 2-9. The tracking responses of state x are shown in Figs. 2-9(a) and 2-9(d); the tracking responses of state \dot{x} are shown in Figs. 2-9(b) and 2-9(e); and the associated control efforts are shown Figs. 2-9(c) and 2-9(f) for $q = 1.95$ and $q = 7.00$, respectively. The simulation results show that the favorable tracking performance can achieve; however, the computation load is heavy. Comparing with Figs. 2-6 and 2-8, and Figs. 2-7 and 2-9 shows that the adaptive fuzzy neural network with asymmetric membership functions performs better than the adaptive fuzzy neural network with symmetric membership functions. However, the structure of the FNN should still be determined by the empiricism.

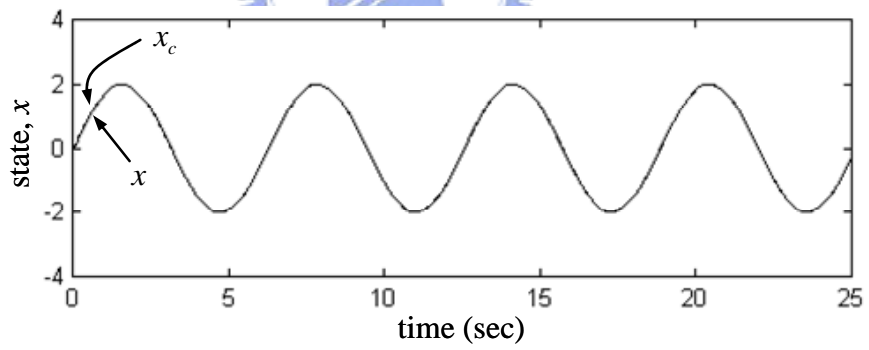




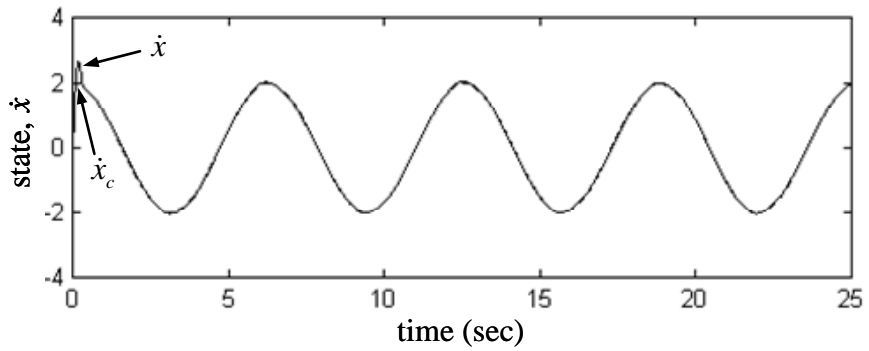
(b)



(c)



(d)



(e)

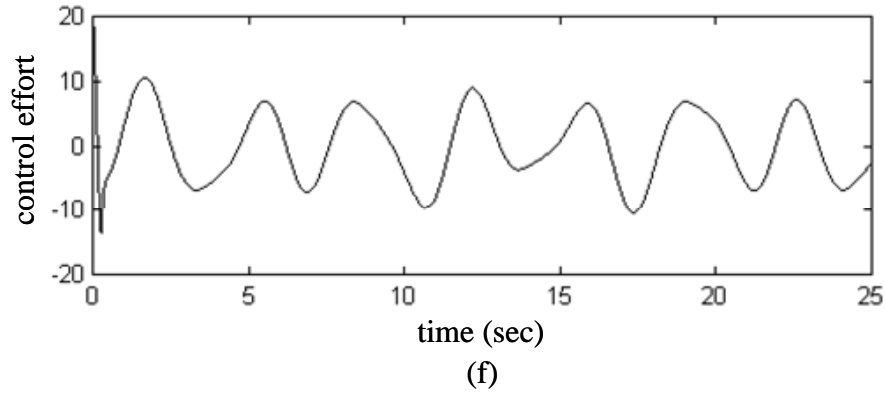
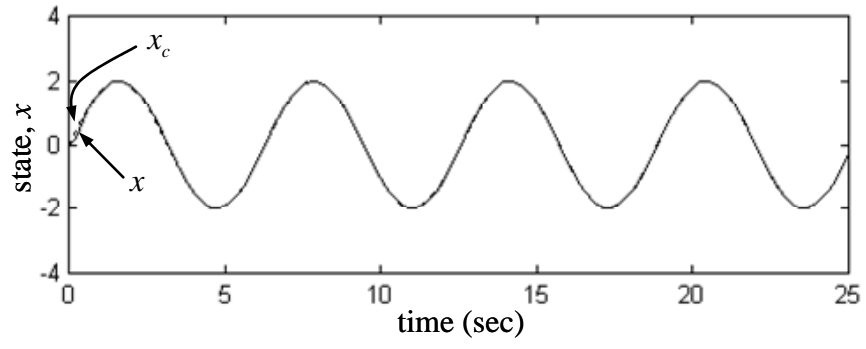


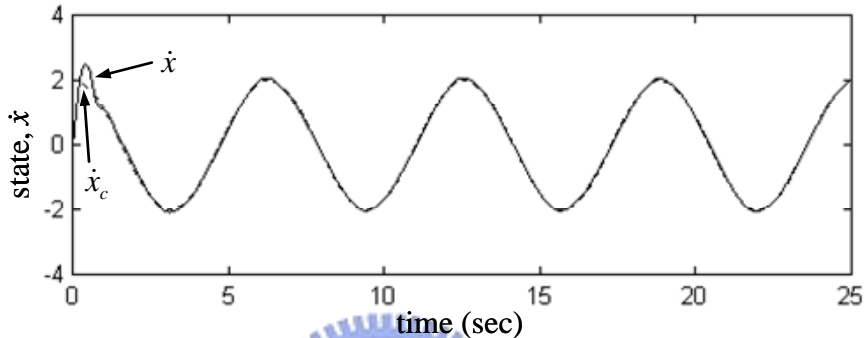
Fig. 2-9. Simulation results of AFNC using 20 asymmetric membership functions.

2.6.2 Simulation for ASAFNC

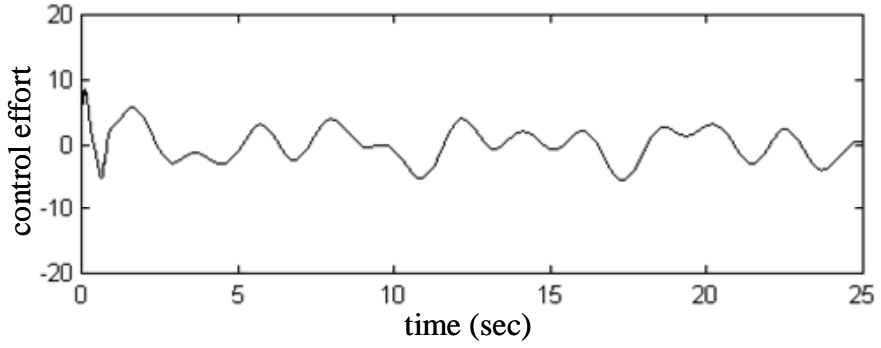
To solve the above problem, the proposed ASAFNC is applied to the chaotic dynamics system. The parameters of ASAFNC system are selected as $k_1 = 2$, $k_2 = 1$, $\eta_w = 80$, $\eta_m = \eta_{\sigma_i} = \eta_{\sigma_r} = 0.2$, $G_{th} = 0.5$, $I_{th} = 0.1$, $P_{th} = 0.1$, $\tau_1 = 0.01$, $\tau_2 = 0.05$, $\sigma_i = 0.6$, $\varpi = 0.1$, and $\delta = 0.6$. All the gains in the proposed control system are chosen to achieve the best transient control performance considering the stability and possible operating conditions. The parameters η_w , η_m , η_{σ_i} , and η_{σ_r} are the leaning rates of SFNN. If the leaning rates are chosen too small, the parameter convergence of SFNN will be easily achieved; however, this will result in slow learning speed. On the other hand, if the leaning rates are chosen too large, the learning speed will be fast; however, the SFNN system may become more unstable. The simulation results of ASAFNC for $q = 1.95$ and $q = 7.00$ are shown in Figs. 2-10 and 2-11, respectively. The tracking responses of state x are shown in Figs. 2-10(a) and 2-11(a); the tracking responses of state \dot{x} are shown in Figs. 2-10(b) and 2-11(b); the associated control efforts are shown Figs. 2-10(c) and 2-11(c); the number of fuzzy rules is shown in Figs. 2-10(d) and 2-11(d); and the final shapes of membership functions are shown in Figs. 2-10(e) and 2-11(e), respectively. These results state that the rule number and good tracking performance can be considered simultaneously in the simulation procedure.



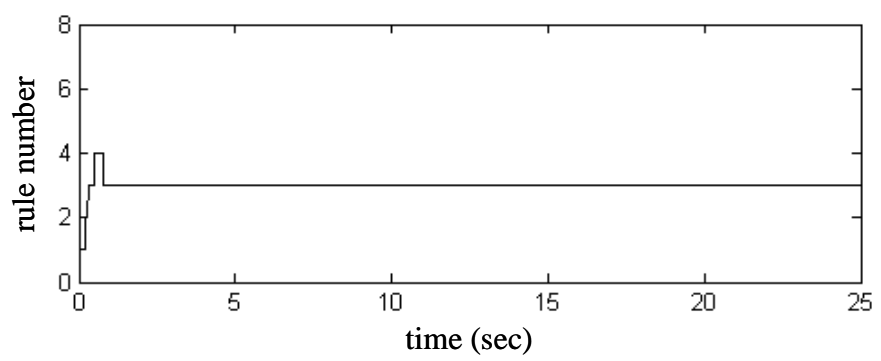
(a)



(b)



(c)



(d)

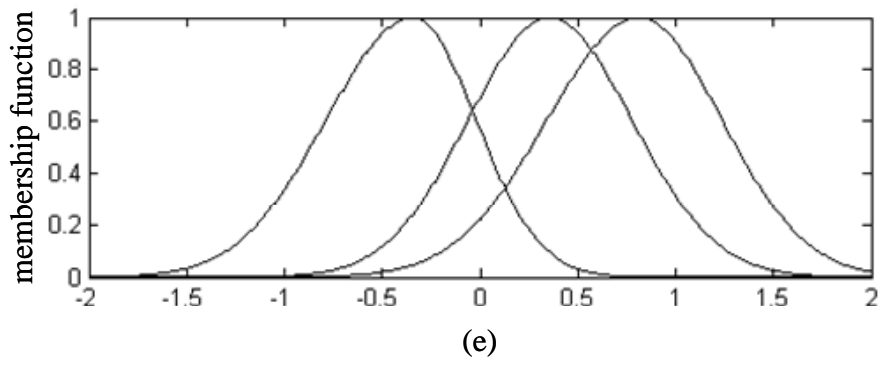
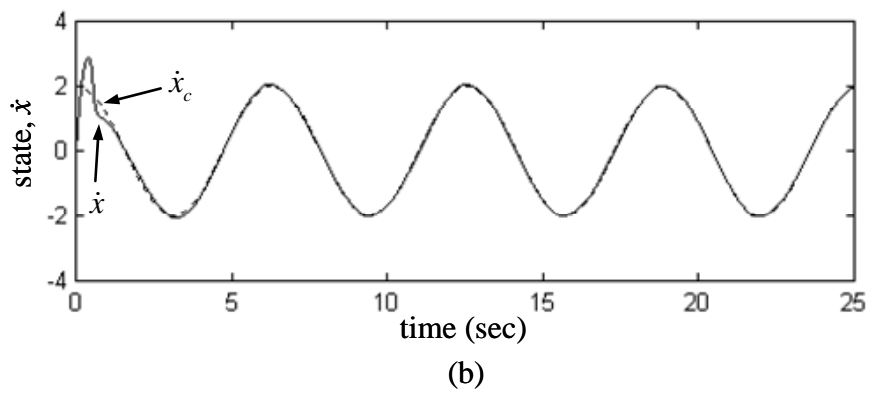
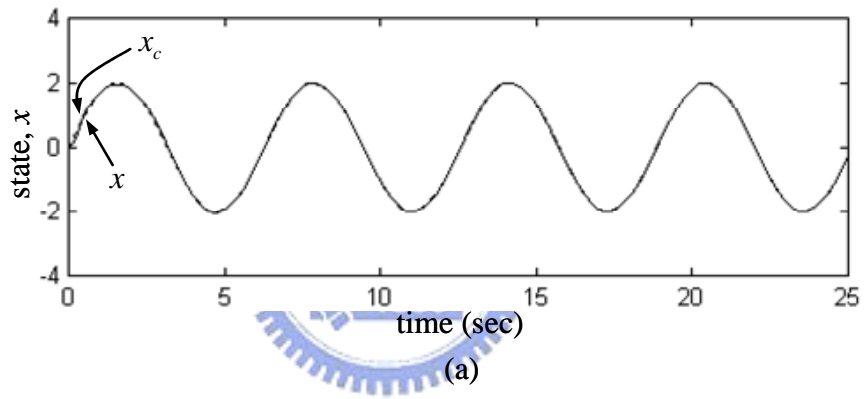
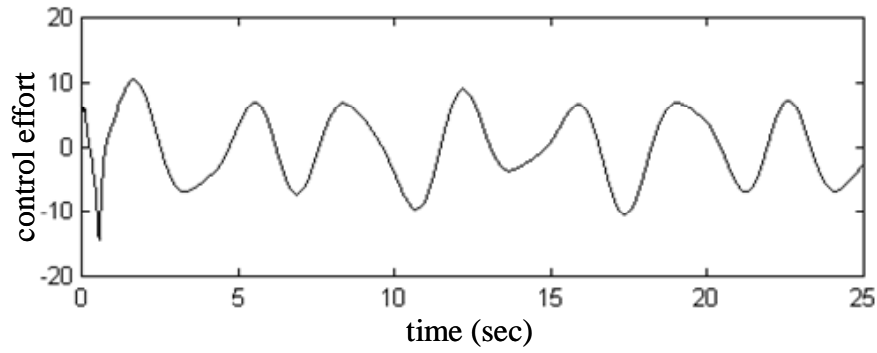
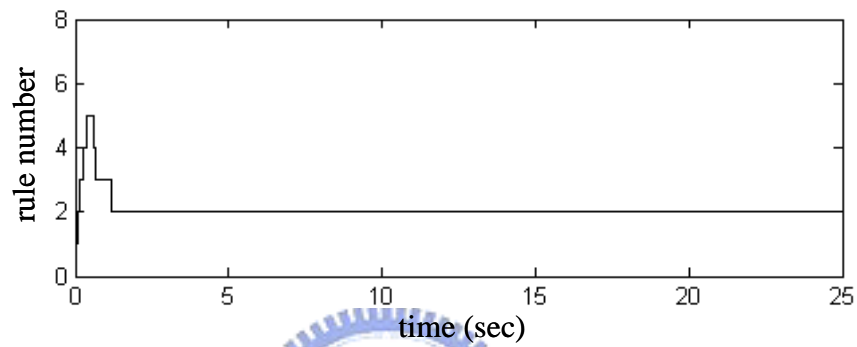


Fig. 2-10. Simulation results of ASAFNC for $q = 1.95$.

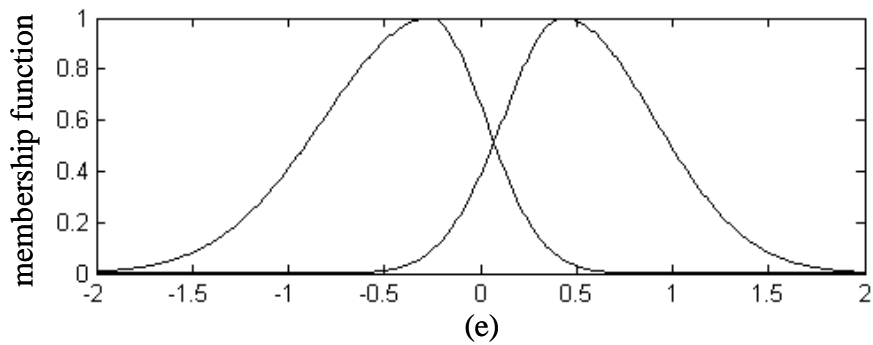




(c)



(d)

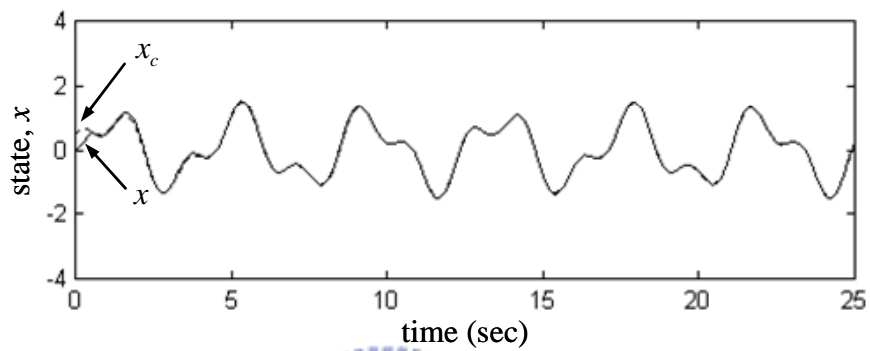


(e)

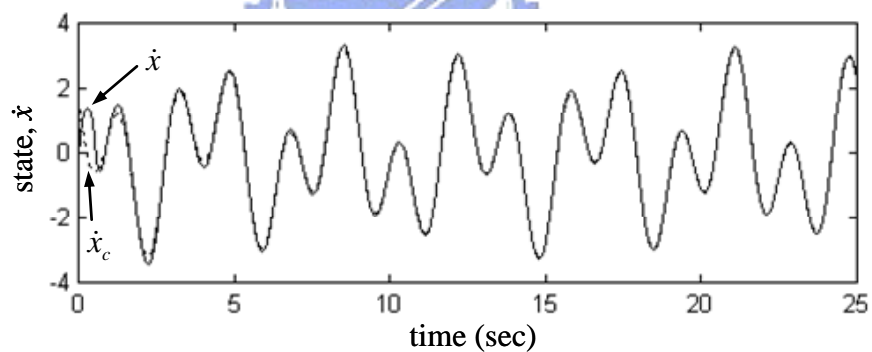
Fig. 2-11. Simulation results of ASAFNC for $q = 7.00$.

To demonstrate the control performance of the proposed ASAFNC system with different reference trajectories, the command $x_c(t) = \sin(1.5t) + 0.5\cos(3.5t)$ is examined here. The simulation results for $q = 1.95$ and $q = 7.00$ are shown in Figs. 2-12 and 2-13, respectively. The tracking responses of state x are shown in Figs. 2-12(a) and 2-13(a); the tracking responses of state \dot{x} are shown in Figs. 2-12(b) and 2-13(b); the associated control efforts

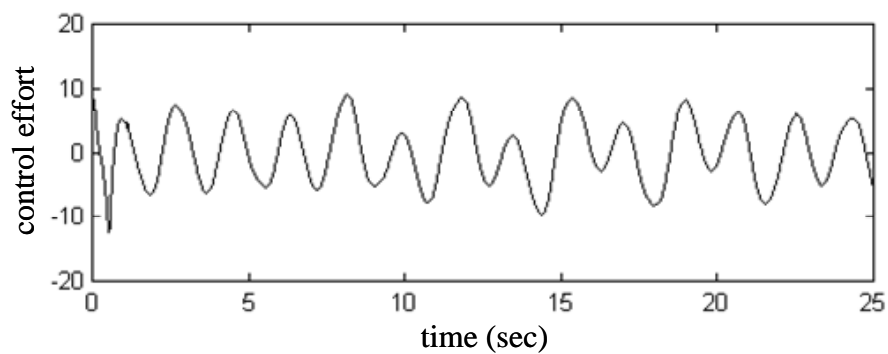
are shown Figs. 2-12(c) and 2-13(c); the number of fuzzy rules is shown in Figs. 2-12(d) and 2-13(d); and the final shapes of membership functions are shown in Figs. 2-12(e) and 2-13(e), respectively. The simulation results show that the proposed ASAFNC system, which includes SFNN with the asymmetric Gaussian membership function, can achieve satisfactory tracking responses in the presence of different reference trajectories. Moreover, a concise SFNN structure can be obtained by the proposed self-structuring mechanism and the online learning algorithms.



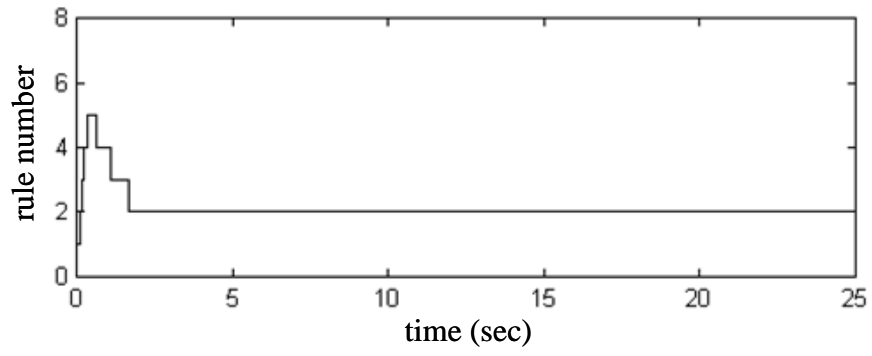
(a)



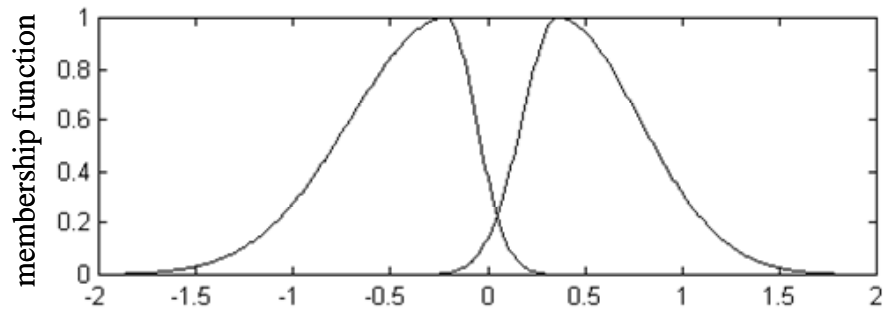
(b)



(c)

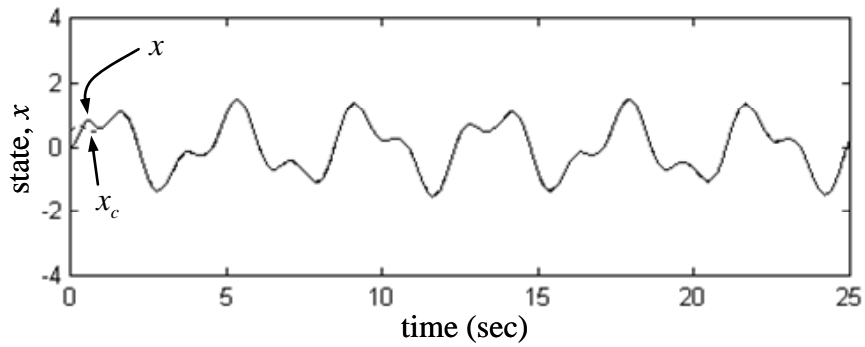


(d)

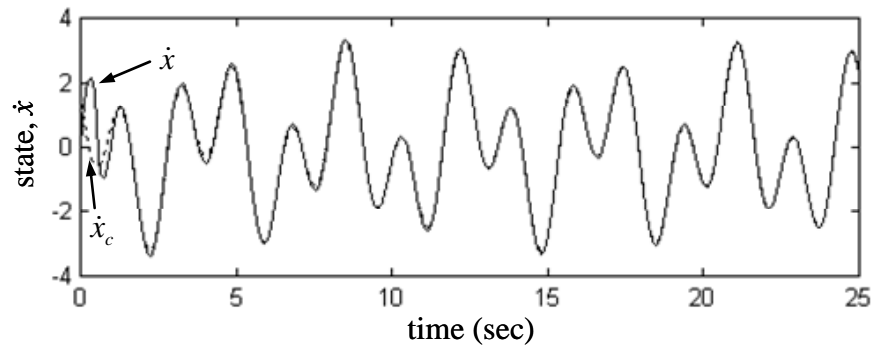


(e)

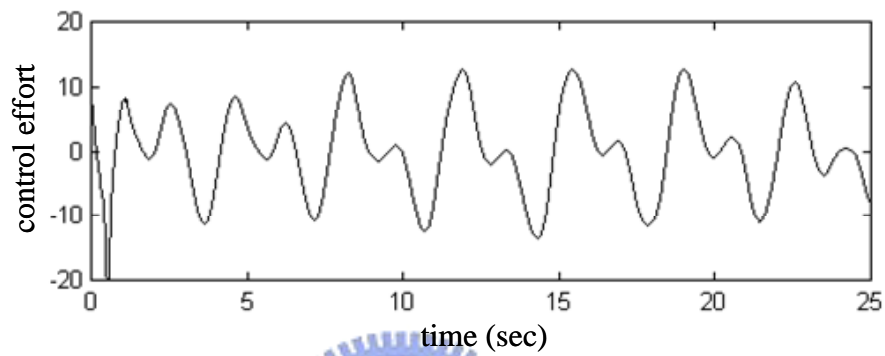
Fig. 2-12. Simulation results of ASAFNC for $q = 1.95$ with different trajectory.



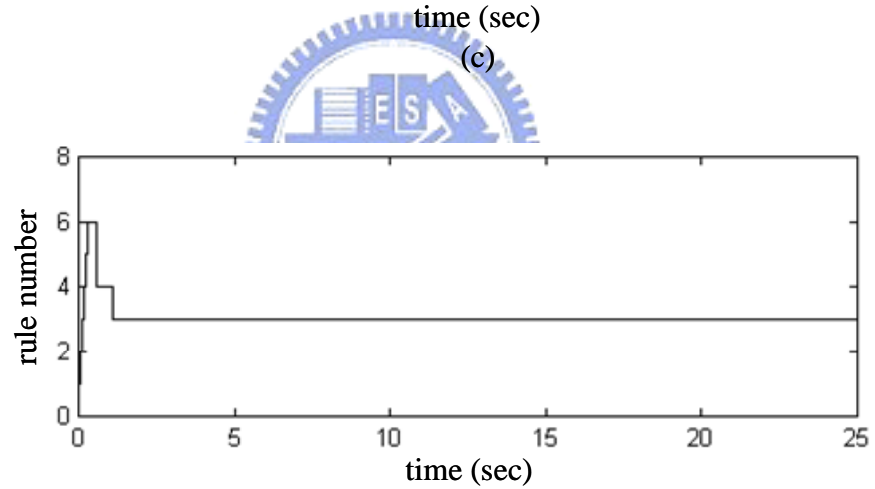
(a)



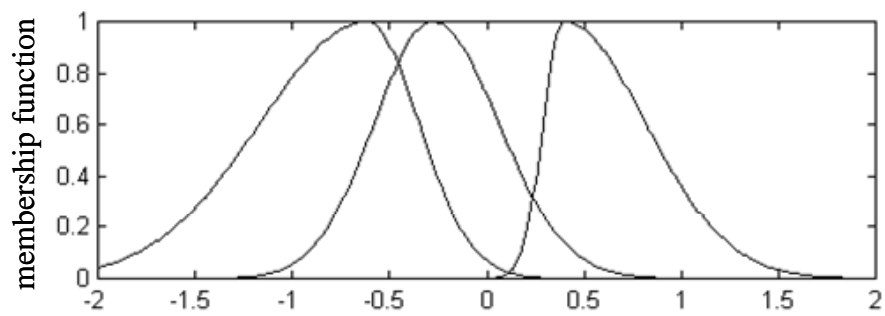
(b)



(c)



(d)



(e)

Fig. 2-13. Simulation results of ASAFNC for $q = 7.00$ with different trajectory.

Chapter 3

System Identification via Hopfield-based Dynamic Neural Network

A dynamic neural network (DNN) is a collection of dynamic neurons which are fully interconnected to a function of their own output. Its stability analysis has been intensively studied since the late 1980's. First, local asymptotic stability was proved [52, 53]. It was shown that there could exist multiple equilibria, which is useful for associative memory or pattern recognition. In this chapter, Hopfield-based DNN will be explored in system identification because of its inherent dynamic behavior.

3.1 Preliminary



Artificial neural networks can have three different types, which are based on their feedback link connection structures, i.e., recurrent (global feedback connections, e.g., Hopfield neural networks (HNNs) [34, 35, 54]), locally recurrent (local feedback connections, e.g., cellular neural networks [55, 56]), and non-recurrent (no feedback connection, e.g., perceptrons [57]). Feedback is like a two-edged sword, in that when it is applied improperly, it can produce harmful effects. In particular, the application of feedback can cause a system that is originally stable to become unstable. Our primary research in this chapter is in the stability of recurrent networks. We focus attention on recurrent networks that use global feedback. Figure 3-1 shows a kind of recurrent neural network, which consists of a set of neurons form a multiple-loop feedback system. The output of each neuron is fed back to each of all neurons as the input in the NN.

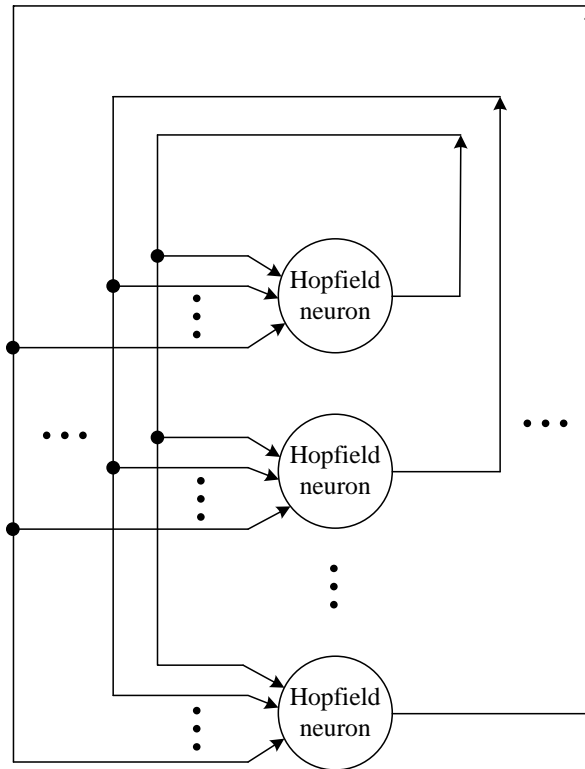
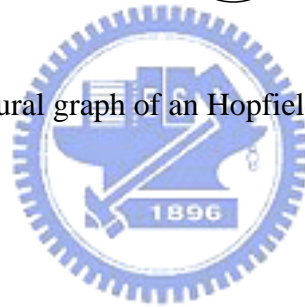


Fig. 3-1. Architectural graph of an Hopfield network with N neurons.



3.1.1 Brief of HNN

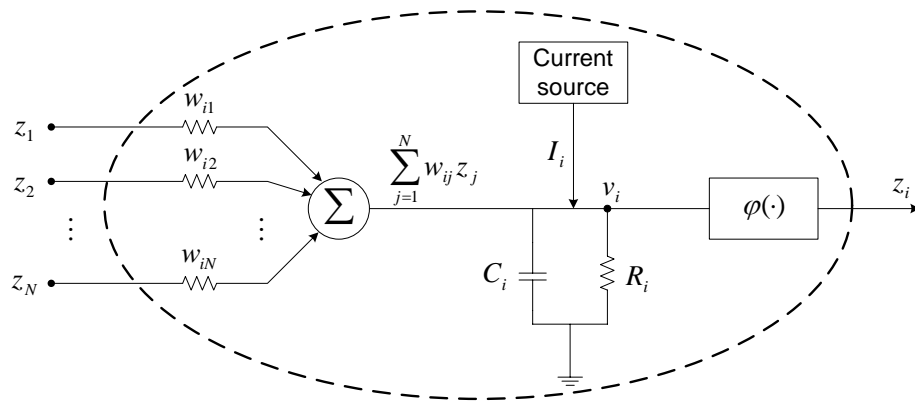


Fig. 3-2. A single neuron of the Hopfield neural network.

When each neuron in Fig. 3-1 is adopted an Hopfield neuron, this network in Fig. 3-1 is the so-called Hopfield neural network. The HNN is first proposed by Hopfield J. J. in 1982

[34]. It is actually a nonlinear closed-loop feedback system which will have dynamic responses in each of the output signals. Fig. 3-2 shows an electric circuit that implements one neuron of HNN. The circuit is based on an RC network connecting a nonlinear activation function $\varphi(\cdot)$ to confine v_i to yield the final output signal z_i . In Fig. 3-2, the inputs $z_j(t)$ ($j=1, \dots, N$) are fed back from the outputs $z_i(t)$ ($i=1, \dots, N$). The inputs $z_j(t)$ are represented by potentials; N is the number of inputs; and the synaptic weighting factors w_{ij} are represented by conductance. The summing junction is characterized by a low input resistance, unity current gain, and high output resistance; that is, it acts as a summing node for incoming currents. A current source I_i represents the externally applied bias in the model. The function $\varphi(\cdot)$ in the figure is a nonlinear sigmoid function. It limits the permissible amplitude range of the output signal to some finite value and is defined by hyperbolic tangent function:

$$z_i = \varphi(v_i) = \tanh\left(\frac{a_i v_i}{2}\right) = \frac{1 - \exp(-a_i v_i)}{1 + \exp(-a_i v_i)} \quad (3-1)$$

which has a slope of $a_i/2$ at the origin as shown by

$$\left. \frac{a_i}{2} = \frac{d\varphi_i}{dv_i} \right|_{v_i=0}$$

Hence, we can say that a_i is the gain parameter of neuron i . Figure 3-3 shows a plot of standard sigmoid nonlinearity $\varphi(v)$.

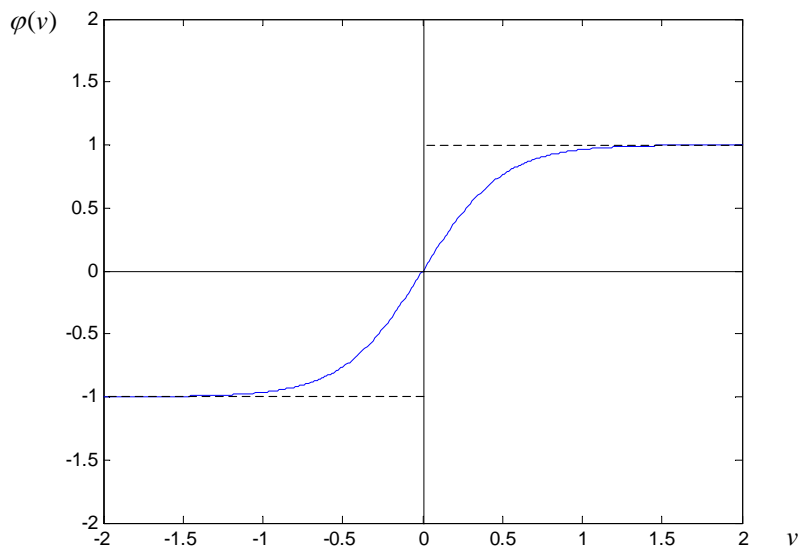


Fig. 3-3. The hyperbolic tangent function with $a = 4$.

By using the Kirchoff's law which states that the total current flowing toward a junction is equal to the total current flowing away from that junction, the following dynamic node equation can be obtained:

$$C_i \frac{dv_i(t)}{dt} + \frac{v_i(t)}{R_i} = \sum_{j=1}^N w_{ij} z_j(t) + I_i, \quad i = 1, \dots, N. \quad (3-2)$$

Because the input $z_j(t)$ is the feedback of the output of the nonlinear sigmoid function $\varphi(\cdot)$, equation (3-2) becomes:

$$C_i \frac{dv_i(t)}{dt} + \frac{v_i(t)}{R_i} = \sum_{j=1}^N w_{ij} \varphi(v_j(t)) + I_i, \quad i = 1, \dots, N. \quad (3-3)$$

Equation (3-3) completely describes the time evolution of the system. A characteristic feature of (3-3) is that the signal $\varphi(v_j(t))$ applied to neuron i by adjoining neuron j is a slowly varying function of time t . If each node is given an initial value $v_j(0)$, the value $v_j(t)$ and the nonlinear activation function output $z_j(t) = \varphi(v_j(t))$ at time t can be known by solving the differential equation in (3-3). In addition, the stability analysis of the HNN plays a major role in the applicability of HNN to engineering fields. The stability analysis of the HNN can be discussed via the energy (or Lyapunov) function of the HNN, which will be introduced in the next subsection.

3.1.2 Stability Analysis of Network

The energy (Lyapunov) function [29, 58] of the HNN can be defined by

$$E = -\frac{1}{2} \sum_{j=1}^N \sum_{i=1}^N w_{ij} z_j z_i + \sum_{i=1}^N \frac{1}{R_i} \int_0^{z_i} \varphi^{-1}(z) dz - \sum_{i=1}^N I_i z_i. \quad (3-4)$$

Differentiating the energy function E with respect to time and using the inverse relation,

$$v_i = \varphi^{-1}(z_i) = \frac{1}{a_i} \log \left(\frac{1 - z_i}{1 + z_i} \right), \text{ based on (3-1) yield}$$

$$\frac{dE}{dt} = -\sum_{i=1}^N \left(\sum_{j=1}^N w_{ij} z_j - \frac{v_i}{R_i} + I_i \right) \frac{dz_i}{dt}. \quad (3-5)$$

The quantity inside the parentheses on the right-side of (3-5) equals actually to $C_i \frac{dv_i}{dt}$ based

on (3-2). Thus, equation (3-5) can be simplified as

$$\frac{dE}{dt} = -\sum_{i=1}^N C_i \left(\frac{dv_i}{dt} \right) \frac{dz_i}{dt}. \quad (3-6)$$

According to (3-1), the relation, $v_i = \varphi^{-1}(z_i)$, can be obtained. Then, the above (3-6) can be rearranged as

$$\frac{dE}{dt} = -\sum_{i=1}^N C_i \left(\frac{d\varphi^{-1}(z_i)}{dt} \right) \frac{dz_i}{dt}. \quad (3-7)$$

By using the chain rule, equation (3-7) can be further derived as:

$$\frac{dE}{dt} = -\sum_{i=1}^N C_i \left(\frac{dz_i}{dt} \right)^2 \left(\frac{d\varphi^{-1}(z_i)}{dz_i} \right). \quad (3-8)$$

Because the inverse output-input relation $\varphi^{-1}(z_i)$ is a monotonically increasing function of the output z_i . It follows therefore that

$$\frac{d\varphi^{-1}(z_i)}{dz_i} > 0, \quad (3-9)$$

for all z_i . Moreover, the following inequality is also true that

$$\left(\frac{dz_i}{dt} \right)^2 \geq 0, \quad (3-10)$$

for all z_i . Hence, according to (3-9) and (3-10), the result can be obtained

$$\frac{dE}{dt} = -\sum_{i=1}^N C_i \left(\frac{dz_i}{dt} \right)^2 \left(\frac{d\varphi^{-1}(z_i)}{dz_i} \right) \leq 0. \quad (3-11)$$

Equation (3-11) states that if the nonlinear activation function is defined as the hyperbolic tangent function in (3-1), the set of nonlinear differential equations defined in (3-3), which represents the dynamic equations of the HNN, is asymptotically stable. Owing to the above introduction of the dynamic model of the HNN and the stability analysis of the network, many researchers are interested in the HNN applications. Thus, in this chapter, an HNN will be adopted as a basic cell of the nonlinear system identifier.

3.1.3 Problem Statement

Consider a continuous time nonlinear dynamic system of the form

$$\dot{\mathbf{x}} = F(\mathbf{x}, u) \quad (3-12)$$

where $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_n]^T$ is the system state vector which is assumed to be available for measurement, $F(\mathbf{x}, u)$ is a nonlinear function, which describes the dynamics of this system, and satisfies a local Lipschits condition such that (3-12) has a unique solution in the sense of Caratheodory, and u is the admissible control input. Also, equation (3-12) can represent either affine or non-affine systems. In addition, we assume that $F(\mathbf{x}, u)$ is bounded. If the admissible control input is given, then for any finite initial condition, the state trajectories are uniformly bounded for any finite $T > 0$. Hence, $|\mathbf{x}(T)| < \infty$. In practice, however, the system information can not be acquired effectively, the works about the system analysis and the controller design can not be proceeded. Thus, the purpose of this chapter is to design an identifier to perform the nonlinear unknown system identification by the dynamic neural network (DNN) based on an HNN with the Lyapunov's training algorithm. A block diagram of identification architecture of the DNN based on an HNN is shown in Fig. 3-4.

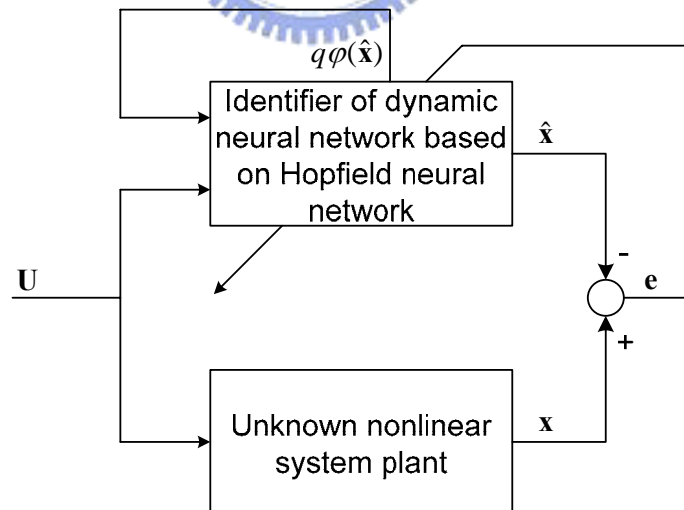


Fig. 3-4. The block diagram of identification architecture of the DNN based on an HNN.

3.2 Identification of Hopfield-based DNN

In order to identify the nonlinear dynamic system in (3-12), a dynamic neural network based on the Hopfield neural network is developed. This proposed DNN based on the HNN is single layer, fully interconnected, recurrent nets, containing connections of sigmoid functions in its neurons. The proposed scheme in the input and output parts is changed slightly the structure from the original HNN structure to achieve better identified effect. According to (3-3), the mathematical formula of the proposed Hopfield-based DNN with zero bias can be expressed as follows

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{W}_\phi\Phi + \mathbf{B}\mathbf{W}_u\mathbf{U} \quad (3-13)$$

where $\hat{\mathbf{x}} = [\hat{x}_1 \ \cdots \ \hat{x}_n]^T = [v_1 \ \cdots \ v_n]^T$ is the state vector of the neural network; $\mathbf{A} = \text{diag}[-1/R_1C_1 \ \cdots \ -1/R_nC_n]$ and $\mathbf{B} = \text{diag}[1/C_1 \ \cdots \ 1/C_n]$ are diagonal matrices; \mathbf{W}_ϕ is an $n \times n$ matrix of synaptic weights for nonlinear state feedback; \mathbf{W}_u is an $n \times n$ matrix of synaptic weights for a single input, of the forms $\mathbf{W}_u = \text{diag}[w_{1(n+1)} \ \cdots \ w_{n(n+1)}]$, or $\mathbf{W}_u \in \mathfrak{R}^{n \times m}$ in which every row is defined as $\mathbf{W}_{u,i} = [w_{i(n+1)} \ \cdots \ w_{i(n+m)}]$ for multi-inputs; $\Phi = [q\varphi(\hat{x}_1) \ \cdots \ q\varphi(\hat{x}_n)]^T$ is the vector of the network feedback and q is a positive constant; and $\mathbf{U} = [u_1 \ \cdots \ u_m]^T$ is the vector of the control force. $\varphi(\cdot)$ is a nonlinear mapping and frequently defined via a so-called sigmoid function, which may, for example, be defined by hyperbolic tangent function as well as (3-1). Fig. 3-5 shows the designed DNN using HNN.

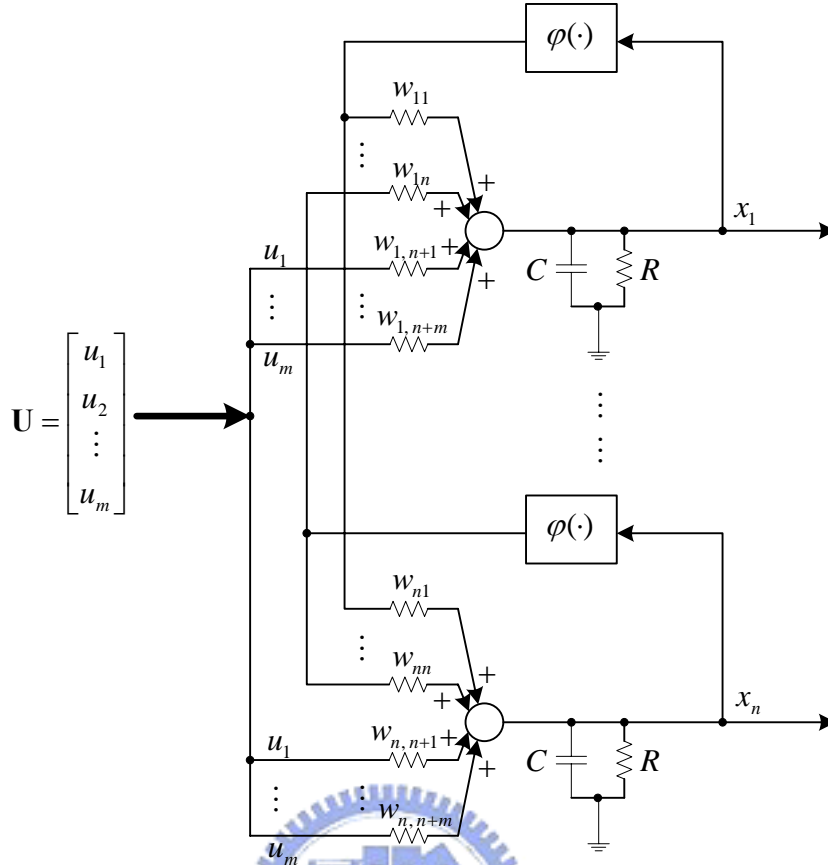


Fig. 3-5. Architecture of the dynamic neural network based on an Hopfield neural network.

From above discussion, a deduction that for a given nonlinear system, there exists an approximate Hopfield-based DNN model can be obtained. Let us first assume that an exact model of the plant is available (i.e., there is no model error). That is, there exist optimal matrices of weighting factor, \mathbf{W}_φ^* and \mathbf{W}_u^* , such that the nonlinear dynamic system (3-12) is completely described by a Hopfield-based DNN of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{W}_\varphi^* \Phi + \mathbf{B}\mathbf{W}_u^* \mathbf{U} \quad (3-14)$$

where all parameters or matrices are as defined earlier. Moreover, the optimal matrices can be further defined as [44, 59]

$$(\mathbf{W}_\varphi^*, \mathbf{W}_u^*) = \arg \min_{\mathbf{W}_\varphi \in \Omega_{\mathbf{W}_\varphi}, \mathbf{W}_u \in \Omega_{\mathbf{W}_u}} \left[\sup_{\mathbf{x} \in \Omega_{\mathbf{x}}, \hat{\mathbf{x}} \in \Omega_{\hat{\mathbf{x}}}, \mathbf{u} \in \Omega_{\mathbf{u}}} |F(\mathbf{x}, \mathbf{u}) - (\mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{W}_\varphi \Phi + \mathbf{B}\mathbf{W}_u \mathbf{U})| \right] \quad (3-15)$$

where

$$\Omega_{\mathbf{W}_\varphi} = \left\{ \mathbf{W}_\varphi : \text{trace}(\mathbf{W}_\varphi^T \mathbf{W}_\varphi) \leq D_{\mathbf{W}_\varphi} \right\} \quad (3-16)$$

and

$$\Omega_{\mathbf{w}_u} = \{ \mathbf{W}_u : \text{trace}(\mathbf{W}_u^T \mathbf{W}_u) \leq D_{\mathbf{w}_u} \}. \quad (3-17)$$

$D_{\mathbf{w}_\varphi}$ and $D_{\mathbf{w}_u}$ are positive constants specified by designers, and $\Omega_{\mathbf{x}}$, $\Omega_{\hat{\mathbf{x}}}$, and Ω_u are compact sets. However, in fact, the optimal weighting factors are difficult to be determined. Thus, the adaptive laws for weighting training have to be appropriately designed to guarantee the identification performance of the Hopfield-based DNN identifier. Define the approximation error between states of the identified DNN and states of the real system as

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}. \quad (3-18)$$

Thus, the derivative of \mathbf{e} with respect to time can be obtained by (3-13) and (3-14)

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} + \mathbf{B}\tilde{\mathbf{W}}_\varphi\Phi + \mathbf{B}\tilde{\mathbf{W}}_u\mathbf{U} \quad (3-19)$$

where $\tilde{\mathbf{W}}_\varphi = \mathbf{W}_\varphi^* - \mathbf{W}_\varphi$ and $\tilde{\mathbf{W}}_u = \mathbf{W}_u^* - \mathbf{W}_u$. The next discussion is to find weight adaptive laws that guarantee to minimize the identified error and the convergence for the identification process. Thus, the identification problem can be stated as optimization and stability problems. The weight adaptive laws will be obtained by proving the stability of the identified system. Now, the following theorem is given first to discuss the system stability and determine the adaptive laws of weighting factors based on no existence of the modeling error. The theorem states the main result concerning the convergence of the proposed identification scheme.

Theorem 1: A nonlinear dynamic system is considered in (3-12), and we assume it can be modeled exactly by (3-14). The identified system is designed as (3-13). If the adaptive laws of weighting factors are selected as

$$\dot{w}_{\varphi, ij} = \eta_\varphi q \varphi(v_j) b_{ii} p_{ii} e_i \quad (3-20)$$

and

$$\dot{w}_{u, ij} = \eta_u u b_{ii} p_{ii} e_i \quad (3-21)$$

or

$$\dot{w}_{u, i(n+j)} = \eta_u u_j b_{ii} p_{ii} e_i \quad (3-22)$$

where η_φ and η_u are learning rates and positive constants, the stability of the overall identification scheme is guaranteed. The adaptive law in (3-21) is for a single input and (3-22) is for multi-inputs.

Proof: Consider the Lyapunov candidate function as

$$V = \frac{1}{2} \mathbf{e}^T \mathbf{P} \mathbf{e} + \frac{1}{2\eta_\phi} \text{trace}(\tilde{\mathbf{W}}_\phi^T \tilde{\mathbf{W}}_\phi) + \frac{1}{2\eta_u} \text{trace}(\tilde{\mathbf{W}}_u^T \tilde{\mathbf{W}}_u) \quad (3-23)$$

where $\mathbf{P} > 0$ is chosen to satisfy the Lyapunov equation, $\mathbf{P}\mathbf{A} + \mathbf{A}^T \mathbf{P} = -\mathbf{Q}$. Taking the derivative of V with respect to time and using (3-19) yields

$$\begin{aligned} \dot{V} &= \frac{1}{2} (\dot{\mathbf{e}}^T \mathbf{P} \mathbf{e} + \mathbf{e}^T \mathbf{P} \dot{\mathbf{e}}) + \frac{1}{\eta_\phi} \text{trace}(\dot{\tilde{\mathbf{W}}}_\phi^T \tilde{\mathbf{W}}_\phi) + \frac{1}{\eta_u} \text{trace}(\dot{\tilde{\mathbf{W}}}_u^T \tilde{\mathbf{W}}_u) \\ &= \frac{1}{2} [(\mathbf{A}\mathbf{e} + \mathbf{B}\tilde{\mathbf{W}}_\phi \Phi + \mathbf{B}\tilde{\mathbf{W}}_u \mathbf{U})^T \mathbf{P} \mathbf{e} + \mathbf{e}^T \mathbf{P} (\mathbf{A}\mathbf{e} + \mathbf{B}\tilde{\mathbf{W}}_\phi \Phi + \mathbf{B}\tilde{\mathbf{W}}_u \mathbf{U})] \\ &\quad + \frac{1}{\eta_\phi} \text{trace}(\dot{\tilde{\mathbf{W}}}_\phi^T \tilde{\mathbf{W}}_\phi) + \frac{1}{\eta_u} \text{trace}(\dot{\tilde{\mathbf{W}}}_u^T \tilde{\mathbf{W}}_u) \\ &= \frac{1}{2} [\mathbf{e}^T (\mathbf{A}^T \mathbf{P} + \mathbf{P}\mathbf{A}) \mathbf{e} + \Phi^T \tilde{\mathbf{W}}_\phi^T \mathbf{B}^T \mathbf{P} \mathbf{e} + \mathbf{U}^T \tilde{\mathbf{W}}_u^T \mathbf{B}^T \mathbf{P} \mathbf{e} + \mathbf{e}^T \mathbf{P} \mathbf{B} \tilde{\mathbf{W}}_\phi \Phi + \mathbf{e}^T \mathbf{P} \mathbf{B} \tilde{\mathbf{W}}_u \mathbf{U}] \\ &\quad + \frac{1}{\eta_\phi} \text{trace}(\dot{\tilde{\mathbf{W}}}_\phi^T \tilde{\mathbf{W}}_\phi) + \frac{1}{\eta_u} \text{trace}(\dot{\tilde{\mathbf{W}}}_u^T \tilde{\mathbf{W}}_u) \end{aligned}$$

Using the Lyapunov equation, i.e., $\mathbf{P}\mathbf{A} + \mathbf{A}^T \mathbf{P} = -\mathbf{Q}$, obtains

$$\begin{aligned} \dot{V} &= -\frac{1}{2} \mathbf{e}^T \mathbf{Q} \mathbf{e} + \frac{1}{2} [\Phi^T \tilde{\mathbf{W}}_\phi^T \mathbf{B}^T \mathbf{P} \mathbf{e} + \mathbf{U}^T \tilde{\mathbf{W}}_u^T \mathbf{B}^T \mathbf{P} \mathbf{e} + \mathbf{e}^T \mathbf{P} \mathbf{B} \tilde{\mathbf{W}}_\phi \Phi + \mathbf{e}^T \mathbf{P} \mathbf{B} \tilde{\mathbf{W}}_u \mathbf{U}] \\ &\quad + \frac{1}{\eta_\phi} \text{trace}(\dot{\tilde{\mathbf{W}}}_\phi^T \tilde{\mathbf{W}}_\phi) + \frac{1}{\eta_u} \text{trace}(\dot{\tilde{\mathbf{W}}}_u^T \tilde{\mathbf{W}}_u) \end{aligned}$$

Because $\Phi^T \tilde{\mathbf{W}}_\phi^T \mathbf{B}^T \mathbf{P} \mathbf{e}$, $\mathbf{U}^T \tilde{\mathbf{W}}_u^T \mathbf{B}^T \mathbf{P} \mathbf{e}$, $\mathbf{e}^T \mathbf{P} \mathbf{B} \tilde{\mathbf{W}}_\phi \Phi$, and $\mathbf{e}^T \mathbf{P} \mathbf{B} \tilde{\mathbf{W}}_u \mathbf{U}$ are all scalars, the following relationship between them can be obtained as

$$\Phi^T \tilde{\mathbf{W}}_\phi^T \mathbf{B}^T \mathbf{P} \mathbf{e} = (\Phi^T \tilde{\mathbf{W}}_\phi^T \mathbf{B}^T \mathbf{P} \mathbf{e})^T = \mathbf{e}^T \mathbf{P} \mathbf{B} \tilde{\mathbf{W}}_\phi \Phi$$

and

$$\mathbf{U}^T \tilde{\mathbf{W}}_u^T \mathbf{B}^T \mathbf{P} \mathbf{e} = (\mathbf{U}^T \tilde{\mathbf{W}}_u^T \mathbf{B}^T \mathbf{P} \mathbf{e})^T = \mathbf{e}^T \mathbf{P} \mathbf{B} \tilde{\mathbf{W}}_u \mathbf{U}.$$

Hence, the derivative of V with respect to time can arrange as

$$\dot{V} = -\frac{1}{2} \mathbf{e}^T \mathbf{Q} \mathbf{e} + \Phi^T \tilde{\mathbf{W}}_\phi^T \mathbf{B}^T \mathbf{P} \mathbf{e} + \mathbf{U}^T \tilde{\mathbf{W}}_u^T \mathbf{B}^T \mathbf{P} \mathbf{e} + \frac{1}{\eta_\phi} \text{trace}(\dot{\tilde{\mathbf{W}}}_\phi^T \tilde{\mathbf{W}}_\phi) + \frac{1}{\eta_u} \text{trace}(\dot{\tilde{\mathbf{W}}}_u^T \tilde{\mathbf{W}}_u). \quad (3-24)$$

We select

$$\frac{1}{\eta_\phi} \text{trace}(\dot{\tilde{\mathbf{W}}}_\phi^T \tilde{\mathbf{W}}_\phi) = -\Phi^T \tilde{\mathbf{W}}_\phi^T \mathbf{B}^T \mathbf{P} \mathbf{e} \quad (3-25)$$

and

$$\frac{1}{\eta_u} \text{trace}(\dot{\tilde{\mathbf{W}}}_u^T \tilde{\mathbf{W}}_u) = -\mathbf{U}^T \tilde{\mathbf{W}}_u^T \mathbf{B}^T \mathbf{P} \mathbf{e}. \quad (3-26)$$

Substituting (3-25) and (3-26) into (3-24), (3-24) can arrange as

$$\dot{V} = -\frac{1}{2} \mathbf{e}^T \mathbf{Q} \mathbf{e} \leq 0. \quad (3-27)$$

Therefore, the stability of the overall identification scheme is guaranteed based on the above result and Lyapunov stability theorem. The adaptive laws of weighting factors in an element form can be obtained based on (3-25) and (3-26) as

$$\dot{w}_{\phi, ij} = \eta_{\phi} q \varphi(v_j) b_{ii} p_{ii} e_i$$

and

$$\dot{w}_{u, ij} = \eta_u u b_{ii} p_{ii} e_i$$

or

$$\dot{w}_{u, i(n+j)} = \eta_u u_j b_{ii} p_{ii} e_i$$

where the second adaptive law is for a single input and the last is for multi-inputs. ■

This stability theorem is guaranteed under the optimal identification model with no modeling error. The next section, we will discuss that if the modeling error does not equal zero, whether the system stability is still guaranteed or not.



3.3 Robust Analysis

Let us consider the same identification scheme in (3-13). If the modeling error s does not equal to zero but small, the stability proof in the previous section may not be guaranteed, i.e., $\dot{V} > 0$. Assume there exist a constant $\gamma < \infty$ so that

$$\int_0^t s^2 d\tau < \gamma, \quad 0 \leq t < \infty.$$

The exact identification model (3-14) needs to correct to consist with the fact and represents as follows:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{W}_{\phi}^* \Phi + \mathbf{B}\mathbf{W}_u^* \mathbf{U} + s \quad (3-28)$$

where all parameters or matrices are as defined earlier. The approximation error is still defined as $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$, and the derivative of \mathbf{e} with respect to time is corrected as

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} + \mathbf{B}\tilde{\mathbf{W}}_{\phi} \Phi + \mathbf{B}\tilde{\mathbf{W}}_u \mathbf{U} + s. \quad (3-29)$$

In addition, the projection algorithm [44, 48, 49] is considered to modify **Theorem 1** and **Theorem 2** is given as follows.

Theorem 2: Consider a nonlinear dynamic system represented by (3-12), and we assume it can be described exactly by (3-28) with the modeling error s . The identified system is defined as (3-13). Let the weighting factors be adjusted by the following adaptive laws:

$$\dot{w}_{\phi, ij} = \begin{cases} \eta_{\phi} q\phi(v_j) b_{ii} p_{ii} e_i, & \text{if } \text{tr}(\mathbf{W}_{\phi}^T \mathbf{W}_{\phi}) < D_{w_{\phi}} \text{ or } (\text{tr}(\mathbf{W}_{\phi}^T \mathbf{W}_{\phi}) = D_{w_{\phi}} \text{ and } q\phi(v_j) w_{\phi, ij} b_{ii} p_{ii} e_i \leq 0) \\ \mathbf{Pr}(\eta_{\phi} q\phi(v_j) b_{ii} p_{ii} e_i), & \text{if } (\text{tr}(\mathbf{W}_{\phi}^T \mathbf{W}_{\phi}) = D_{w_{\phi}} \text{ and } q\phi(v_j) w_{\phi, ij} b_{ii} p_{ii} e_i > 0) \end{cases} \quad (3-30)$$

where the symbol “tr” means “trace” and η_{ϕ} is learning rate and positive constant, and the projection operator is given as

$$\mathbf{Pr}(\eta_{\phi} q\phi(v_j) b_{ii} p_{ii} e_i) = \eta_{\phi} q\phi(v_j) b_{ii} p_{ii} e_i - \eta_{\phi} \frac{q\phi(v_j) w_{\phi, ij} b_{ii} p_{ii} e_i}{|w_{\phi, ij}|^2} w_{\phi, ij},$$

and

$$\dot{w}_{u, ij} = \begin{cases} \eta_u u b_{ii} p_{ii} e_i, & \text{if } \text{tr}(\mathbf{W}_u^T \mathbf{W}_u) < D_{w_u} \text{ or } (\text{tr}(\mathbf{W}_u^T \mathbf{W}_u) = D_{w_u} \text{ and } u w_{u, ij} b_{ii} p_{ii} e_i \leq 0) \\ \mathbf{Pr}(\eta_u u b_{ii} p_{ii} e_i), & \text{if } (\text{tr}(\mathbf{W}_u^T \mathbf{W}_u) = D_{w_u} \text{ and } u w_{u, ij} b_{ii} p_{ii} e_i > 0) \end{cases} \quad (3-31)$$

where η_u is learning rate and positive constant, and the projection operator is given as

$$\mathbf{Pr}(\eta_u u b_{ii} p_{ii} e_i) = \eta_u u b_{ii} p_{ii} e_i - \eta_u \frac{u w_{u, ij} b_{ii} p_{ii} e_i}{|w_{u, ij}|^2} w_{u, ij}.$$

Then, the following properties can be held:

1. $\text{trace}(\mathbf{W}_{\phi}^T \mathbf{W}_{\phi}) \leq D_{w_{\phi}}$ and $\text{trace}(\mathbf{W}_u^T \mathbf{W}_u) \leq D_{w_u}$
2. $\int_0^T \|\mathbf{e}\|^2 dt \leq \frac{2}{\lambda_{\min}(\mathbf{Q}) - 1} V(0) + \frac{1}{\lambda_{\min}(\mathbf{Q}) - 1} \|\mathbf{P}\|^2 \int_0^T \|s\|^2 dt$ where $\lambda_{\min}(\mathbf{Q})$ is the minimum eigenvalue of \mathbf{Q} .

Proof:

1. In order to prove the fact that the condition $\text{trace}(\mathbf{W}_{\phi}^T \mathbf{W}_{\phi}) \leq D_{w_{\phi}}$ can be held, let

$$V_{w_{\phi}} = \frac{1}{2} \text{trace}(\mathbf{W}_{\phi}^T \mathbf{W}_{\phi}). \quad (3-32)$$

If the first line of (3-30) is true, we have either $\text{trace}(\mathbf{W}_{\phi}^T \mathbf{W}_{\phi}) < D_{w_{\phi}}$ or

$$\dot{V}_{w_{\phi}} = \eta_{\phi} q\phi(v_j) w_{\phi, ij} b_{ii} p_{ii} e_i \leq 0 \quad (3-33)$$

when $\text{trace}(\mathbf{W}_{\phi}^T \mathbf{W}_{\phi}) = D_{w_{\phi}}$. That is, the condition $\text{trace}(\mathbf{W}_{\phi}^T \mathbf{W}_{\phi}) \leq D_{w_{\phi}}$ can always be satisfied if the first line of (3-30) is true. If the second line of (3-30) is true, we have

$trace(\mathbf{W}_\varphi^T \mathbf{W}_\varphi) = D_{w_\varphi}$ and

$$\dot{V}_{w_\varphi} = \eta_\varphi q \varphi(v_j) b_{ii} p_{ii} e_i w_{\varphi, ij} - \eta_\varphi \frac{q \varphi(v_j) w_{\varphi, ij} b_{ii} p_{ii} e_i}{|w_{\varphi, ij}|^2} |w_{\varphi, ij}|^2 = 0. \quad (3-34)$$

That is, $trace(\mathbf{W}_\varphi^T \mathbf{W}_\varphi) \leq D_{w_\varphi}$. Therefore, the condition $trace(\mathbf{W}_\varphi^T \mathbf{W}_\varphi) \leq D_{w_\varphi}, \forall t \geq 0$ can be held. By using the same discussion, the condition $trace(\mathbf{W}_u^T \mathbf{W}_u) \leq D_{w_u}, \forall t \geq 0$ can also be derived.

2. Consider the same Lyapunov candidate function in (3-23), and taking the derivative of V with respect to time and using (3-29) yields

$$\begin{aligned} \dot{V} &= \frac{1}{2} [\mathbf{e}^T \mathbf{A}^T \mathbf{P} \mathbf{e} + \Phi^T \tilde{\mathbf{W}}_\varphi^T \mathbf{B}^T \mathbf{P} \mathbf{e} + \mathbf{U}^T \tilde{\mathbf{W}}_u^T \mathbf{B}^T \mathbf{P} \mathbf{e} + s^T \mathbf{P} \mathbf{e} + \mathbf{e}^T \mathbf{P} \mathbf{A} \mathbf{e} + \mathbf{e}^T \mathbf{P} \mathbf{B} \tilde{\mathbf{W}}_\varphi \Phi \\ &\quad + \mathbf{e}^T \mathbf{P} \mathbf{B} \tilde{\mathbf{W}}_u \mathbf{U} + \mathbf{e}^T \mathbf{P} \mathbf{s}] + \frac{1}{\eta_\varphi} trace(\dot{\tilde{\mathbf{W}}}_\varphi^T \tilde{\mathbf{W}}_\varphi) + \frac{1}{\eta_u} trace(\dot{\tilde{\mathbf{W}}}_u^T \tilde{\mathbf{W}}_u) \\ &= -\frac{1}{2} \mathbf{e}^T \mathbf{Q} \mathbf{e} + \frac{1}{2} [\Phi^T \tilde{\mathbf{W}}_\varphi^T \mathbf{B}^T \mathbf{P} \mathbf{e} + \mathbf{U}^T \tilde{\mathbf{W}}_u^T \mathbf{B}^T \mathbf{P} \mathbf{e} + \mathbf{e}^T \mathbf{P} \mathbf{B} \tilde{\mathbf{W}}_\varphi \Phi + \mathbf{e}^T \mathbf{P} \mathbf{B} \tilde{\mathbf{W}}_u \mathbf{U} \\ &\quad + s^T \mathbf{P} \mathbf{e} + \mathbf{e}^T \mathbf{P} \mathbf{s}] + \frac{1}{\eta_\varphi} trace(\dot{\tilde{\mathbf{W}}}_\varphi^T \tilde{\mathbf{W}}_\varphi) + \frac{1}{\eta_u} trace(\dot{\tilde{\mathbf{W}}}_u^T \tilde{\mathbf{W}}_u) \end{aligned}$$

Because $\Phi^T \tilde{\mathbf{W}}_\varphi^T \mathbf{B}^T \mathbf{P} \mathbf{e}$, $\mathbf{U}^T \tilde{\mathbf{W}}_u^T \mathbf{B}^T \mathbf{P} \mathbf{e}$, $\mathbf{e}^T \mathbf{P} \mathbf{B} \tilde{\mathbf{W}}_\varphi \Phi$, $\mathbf{e}^T \mathbf{P} \mathbf{B} \tilde{\mathbf{W}}_u \mathbf{U}$ and $s^T \mathbf{P} \mathbf{e}$ are all scales, the following relationship between them can be obtained as

$$\begin{aligned} \Phi^T \tilde{\mathbf{W}}_\varphi^T \mathbf{B}^T \mathbf{P} \mathbf{e} &= (\Phi^T \tilde{\mathbf{W}}_\varphi^T \mathbf{B}^T \mathbf{P} \mathbf{e})^T = \mathbf{e}^T \mathbf{P} \mathbf{B} \tilde{\mathbf{W}}_\varphi \Phi \\ \mathbf{U}^T \tilde{\mathbf{W}}_u^T \mathbf{B}^T \mathbf{P} \mathbf{e} &= (\mathbf{U}^T \tilde{\mathbf{W}}_u^T \mathbf{B}^T \mathbf{P} \mathbf{e})^T = \mathbf{e}^T \mathbf{P} \mathbf{B} \tilde{\mathbf{W}}_u \mathbf{U} \\ s^T \mathbf{P} \mathbf{e} &= (s^T \mathbf{P} \mathbf{e})^T = \mathbf{e}^T \mathbf{P} \mathbf{s}. \end{aligned}$$

Hence, the derivative of V with respect to time can arrange as

$$\dot{V} = -\frac{1}{2} \mathbf{e}^T \mathbf{Q} \mathbf{e} + \Phi^T \tilde{\mathbf{W}}_\varphi^T \mathbf{B}^T \mathbf{P} \mathbf{e} + \mathbf{U}^T \tilde{\mathbf{W}}_u^T \mathbf{B}^T \mathbf{P} \mathbf{e} + \mathbf{e}^T \mathbf{P} \mathbf{s} + \frac{1}{\eta_\varphi} trace(\dot{\tilde{\mathbf{W}}}_\varphi^T \tilde{\mathbf{W}}_\varphi) + \frac{1}{\eta_u} trace(\dot{\tilde{\mathbf{W}}}_u^T \tilde{\mathbf{W}}_u). \quad (3-35)$$

Define some useful variables as

$$J_{w_\varphi} = \frac{1}{\eta_\varphi} trace(\dot{\tilde{\mathbf{W}}}_\varphi^T \tilde{\mathbf{W}}_\varphi) + \Phi^T \tilde{\mathbf{W}}_\varphi^T \mathbf{B}^T \mathbf{P} \mathbf{e} \quad (3-36)$$

and

$$J_{w_u} = \frac{1}{\eta_u} trace(\dot{\tilde{\mathbf{W}}}_u^T \tilde{\mathbf{W}}_u) + \mathbf{U}^T \tilde{\mathbf{W}}_u^T \mathbf{B}^T \mathbf{P} \mathbf{e}. \quad (3-37)$$

Thus, the derivative of V with respect to time shown in (3-35) can be rewritten as

$$\dot{V} = -\frac{1}{2} \mathbf{e}^T \mathbf{Q} \mathbf{e} + J_{\mathbf{w}_\varphi} + J_{\mathbf{w}_u} + \mathbf{e}^T \mathbf{P} \mathbf{s}. \quad (3-38)$$

By using (3-30), $J_{\mathbf{w}_\varphi} = 0$ for $[\text{trace}(\mathbf{W}_\varphi^T \mathbf{W}_\varphi) < D_{\mathbf{w}_\varphi}$ or $(\text{trace}(\mathbf{W}_\varphi^T \mathbf{W}_\varphi) = D_{\mathbf{w}_\varphi}$, and $q\varphi(v_j)w_{\varphi,ij}b_{ii}p_{ii}e_i \leq 0)$] can be obtained. For $[\text{trace}(\mathbf{W}_\varphi^T \mathbf{W}_\varphi) = D_{\mathbf{w}_\varphi}$ and $q\varphi(v_j)w_{\varphi,ij}b_{ii}p_{ii}e_i > 0]$,

$$J_{\mathbf{w}_\varphi} = \sum_{i,j} q\varphi(v_j) \frac{\tilde{w}_{\varphi,ij} w_{\varphi,ij}}{|w_{\varphi,ij}|^2} b_{ii} p_{ii} e_i w_{\varphi,ij}. \quad (3-39)$$

can be obtained. Because \mathbf{W}_φ^* belongs to the constraint set $\Omega_{\mathbf{w}_\varphi}$, the inequality $|w_{\varphi,ij}| = D_{\mathbf{w}_\varphi} \geq w_{\varphi,ij}^*$ can be obtained. By using the previous inequality,

$\tilde{w}_{\varphi,ij} w_{\varphi,ij} = \frac{1}{2} (|w_{\varphi,ij}^*|^2 - |w_{\varphi,ij}|^2 - |\tilde{w}_{\varphi,ij}|^2) \leq 0$ can be obtained. Thus, equation (3-39) can be rewritten as

$$J_{\mathbf{w}_\varphi} = \sum_{i,j} \frac{q\varphi(v_j)}{2} \frac{(|w_{\varphi,ij}^*|^2 - |w_{\varphi,ij}|^2 - |\tilde{w}_{\varphi,ij}|^2)}{|w_{\varphi,ij}|^2} b_{ii} p_{ii} e_i w_{\varphi,ij} \leq 0. \quad (3-40)$$

Similarly, by using (3-31), $J_{\mathbf{w}_u} = 0$ for $[\text{trace}(\mathbf{W}_u^T \mathbf{W}_u) < D_{\mathbf{w}_u}$ or $(\text{trace}(\mathbf{W}_u^T \mathbf{W}_u) = D_{\mathbf{w}_u}$, and $uw_{u,ij}b_{ii}p_{ii}e_i \leq 0)$] can be obtained. For $[\text{tr}(\mathbf{W}_u^T \mathbf{W}_u) = D_{\mathbf{w}_u}$ and $uw_{u,ij}b_{ii}p_{ii}e_i > 0]$,

$$J_{\mathbf{w}_u} = \sum_{i,j} u \frac{\tilde{w}_{u,ij} w_{u,ij}}{|w_{u,ij}|^2} b_{ii} p_{ii} e_i w_{u,ij} \quad (3-41)$$

can be obtained and the following inequality

$$J_{\mathbf{w}_u} = \sum_{i,j} \frac{u}{2} \frac{(|w_{u,ij}^*|^2 - |w_{u,ij}|^2 - |\tilde{w}_{u,ij}|^2)}{|w_{u,ij}|^2} b_{ii} p_{ii} e_i w_{u,ij} \leq 0 \quad (3-42)$$

can be derived. Hence, for any possible condition occurs in (3-30) and (3-31), the conditions $J_{\mathbf{w}_\varphi} \leq 0$ and $J_{\mathbf{w}_u} \leq 0$ can be satisfied. Then, equation (3-38) can be reorganized as

$$\begin{aligned} \dot{V} &= -\frac{1}{2} \mathbf{e}^T \mathbf{Q} \mathbf{e} + J_{\mathbf{w}_\varphi} + J_{\mathbf{w}_u} + \mathbf{e}^T \mathbf{P} \mathbf{s} \\ &\leq -\frac{1}{2} \mathbf{e}^T \mathbf{Q} \mathbf{e} + \mathbf{e}^T \mathbf{P} \mathbf{s} \\ &\leq -\frac{1}{2} \lambda_{\min}(\mathbf{Q}) \|\mathbf{e}\|^2 + \mathbf{e}^T \mathbf{P} \mathbf{s} \end{aligned}$$

$$\begin{aligned}
&= -\frac{1}{2}(\lambda_{\min}(\mathbf{Q})-1)\|\mathbf{e}\|^2 - \frac{1}{2}(\|\mathbf{e}\|^2 + 2\mathbf{e}^T \mathbf{P}s + \|\mathbf{P}s\|^2) + \frac{1}{2}\|\mathbf{P}s\|^2 \\
&\leq -\frac{1}{2}(\lambda_{\min}(\mathbf{Q})-1)\|\mathbf{e}\|^2 + \frac{1}{2}\|\mathbf{P}s\|^2
\end{aligned} \tag{3-43}$$

Integrating both sides of (3-43) from $t=0$ to $t=T$ ($0 < T < \infty$) and choosing that $\lambda_{\min}(\mathbf{Q}) > 1$ yield

$$\frac{1}{2}(\lambda_{\min}(\mathbf{Q})-1)\int_0^T \|\mathbf{e}\|^2 dt \leq -[V(t)-V(0)] + \frac{1}{2}\|\mathbf{P}\|^2 \int_0^T \|s\|^2 dt \tag{3-44}$$

Since $V(t) \geq 0$, equation (3-44) can be arranged as

$$\int_0^T \|\mathbf{e}\|^2 dt \leq \frac{2}{\lambda_{\min}(\mathbf{Q})-1}V(0) + \frac{1}{\lambda_{\min}(\mathbf{Q})-1}\|\mathbf{P}\|^2 \int_0^T \|s\|^2 dt \tag{3-45}$$

which is equivalent to the second property. Thus, the approximation error e will converge to a certain small boundary. ■

Remark: To simplify the discussion, without losing generality, the adaptive law (3-31) derived in **Theorem 2** is only for a single input. The adaptive law for multi-inputs can also be derived by using the similar approach in the proof of **Theorem 2**.

In the next two sections, the proposed DNN based on the HNN will be applied to identify nonlinear dynamic systems. Two examples are employed to illustrate the effectiveness of the proposed scheme.

3.4 Simulation Results of Magnetic Levitation System

In this section, a model of the magnetic levitation system [53, 60], which is nonlinear, unstable, and non-affine, is considered first. This system has the basic ingredients of systems constructed to levitate mass, used in gyroscopes, accelerometers, and fast trains. The equation of motion of the ball is

$$m\ddot{x} = -k\dot{x} + mg + H(x, i) \tag{3-46}$$

where m is the mass of the ball, x is the vertical (downward) position of the ball, k is a viscous friction coefficient, g is the acceleration of gravity, $H(x, i)$ is the force generated by the electromagnet, and i is its electric current. According to Reference [53], $H(x, i)$ can

be obtained and (3-46) can be represented as the standard canonical form as follows:

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= g - \frac{k}{m}x_2 - \frac{L_0 a u^2}{2m(a+x_1)^2}\end{aligned}\quad (3-47)$$

where x_1 (m) is the vertical gap between the ball and the magnet, x_2 (m/second) is the vertical velocity of the ball, u (A) is the current in the coil of the electromagnet, $k = 0.001$ N/m/s, $m = 0.1$ kg, $L_0 = 0.02$ H is the nominal point inductance, and $a = 0.05$ m is positive constant. In order to obtain suitable data for identification, the control force is selected as

$$u(t) = -k_1(x_1(t) - r(t)) - k_2x_2 + u_b(t) \quad (3-48)$$

where $k_1 = -50.9568$ and $k_2 = -2.5640$ are the state feedback gains, $r(t)$ is a reference position and $u_b(t)$ is a model-based bias given by

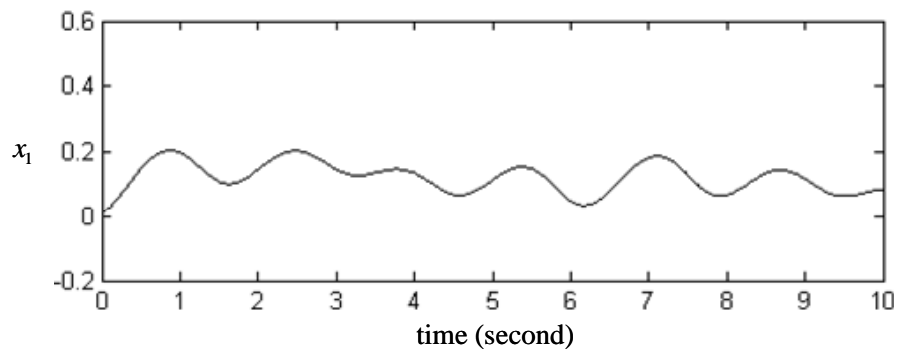
$$u_b(t) = (a + r(t)) \sqrt{\frac{2mg}{aL_0}}. \quad (3-49)$$

This bias is introduced to make the point $x = [r \ 0]^T$ an equilibrium point. Moreover, the reference is chosen to be a sum of five sinusoidals with different frequencies plus an offset

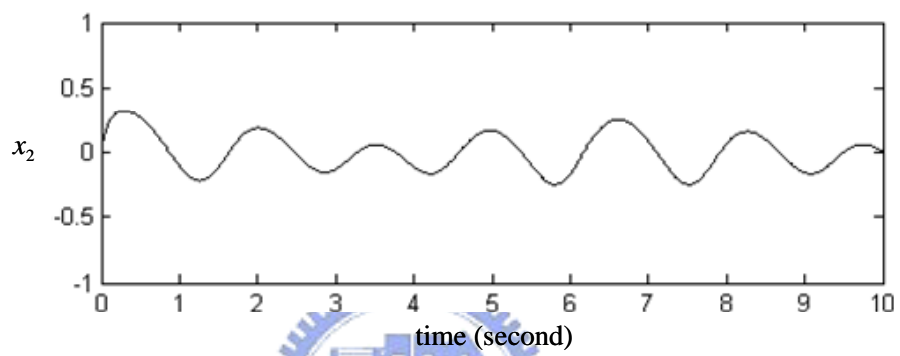
$$r(t) = \sum_{j=1}^5 A_j \sin(w_j t) + r_0 \quad (3-50)$$

where $A_j = 0.04$ and $j = 1, \dots, 5$, $w_1 = 0.5$, $w_2 = 1$, $w_3 = 3$, $w_4 = 4$, $w_5 = 4$, and $r_0 = 0.13$. Fig. 3-6 shows the training data obtained from the closed-loop identification experiment in advance.

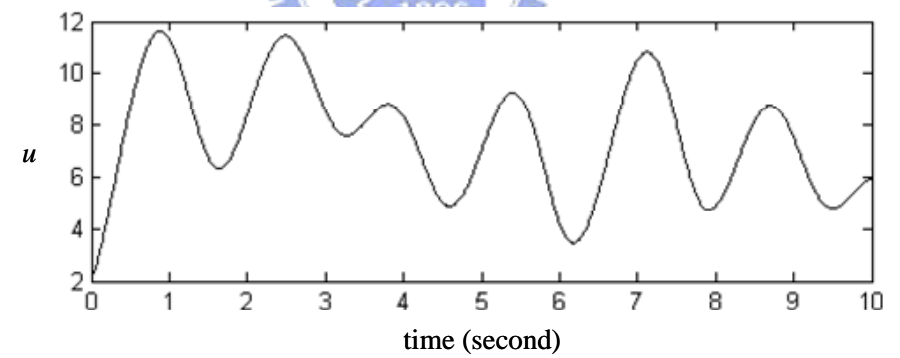
The parameters of the DNN based on the HNN are selected as $R = 100 \ \Omega$, $C = 0.01$ F, $a_i = 1$, $q = 1$, and $\mathbf{Q} = \text{diag}[2 \ 2]$. The initial voltages of two Hopfield neurons are both zero volt. The learning rates of weighting factors are selected as $\eta_\varphi = 0.05$ and $\eta_u = 0.01$. The initial weight values are all set as 0.1. The following simulation results show the identification ability of the proposed network. The simulation results are shown in Figs. 3-7, 3-8, and 3-9. The command trajectories x_{c1} and x_{c2} , the identified vertical position of the ball \hat{x}_1 , and the identified vertical velocity of the ball \hat{x}_2 are shown in Figs. 3-7(a) and 3-7(b). Figs. 3-7(c) and 3-7(d) are the enlarging drawing of y-axis of Figs. 3-7(a) and 3-7(b), respectively. Figs. 3-8(a) and 3-8(b) show the approximated errors. Fig. 3-9 shows the training process of weighting factors.



(a)



(b)



(c)

Fig. 3-6. Training data obtained from the magnetic levitation system.

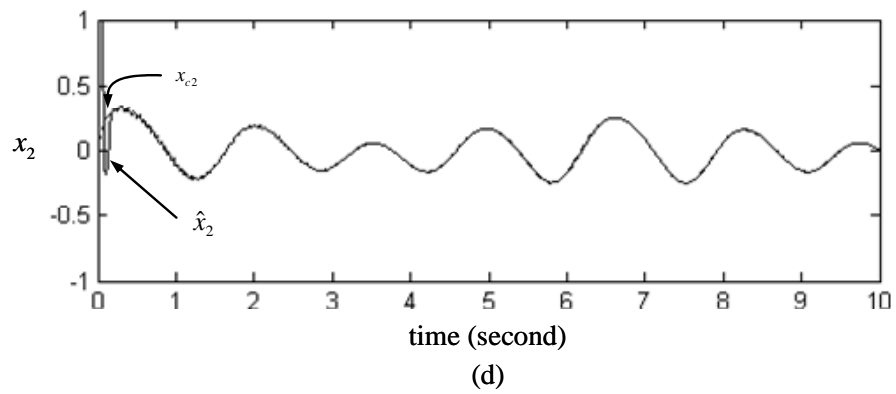
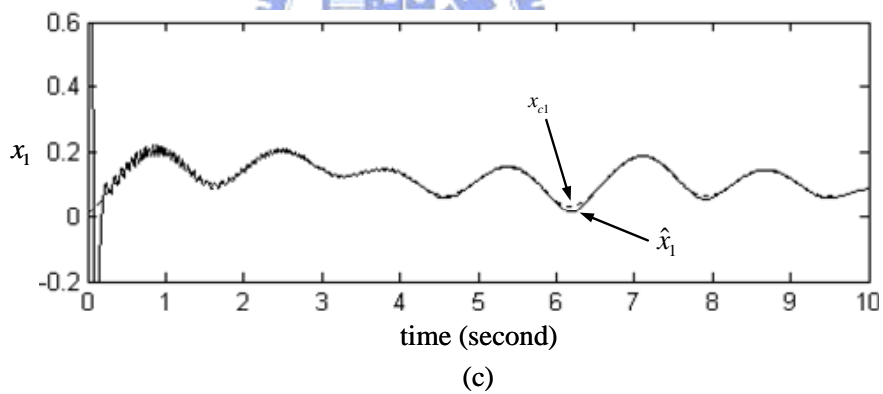
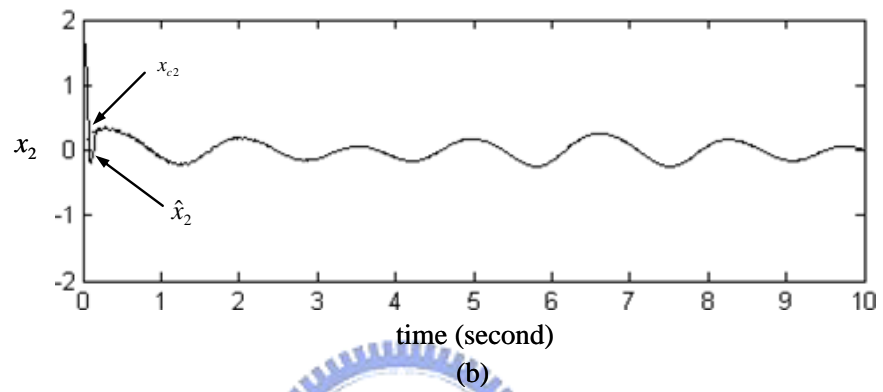
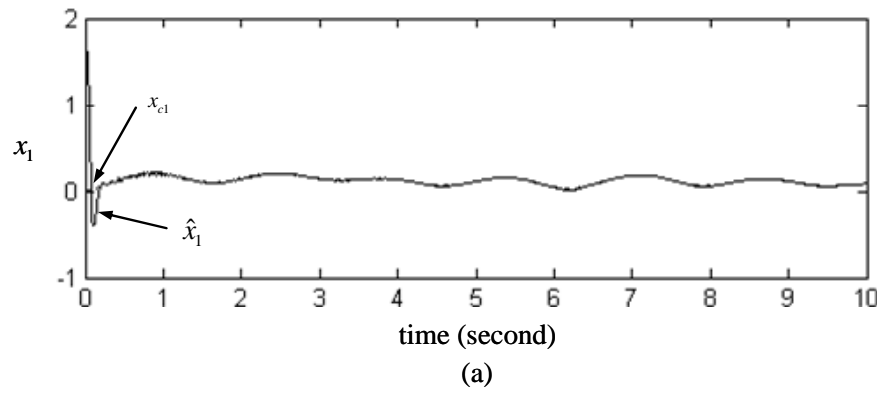
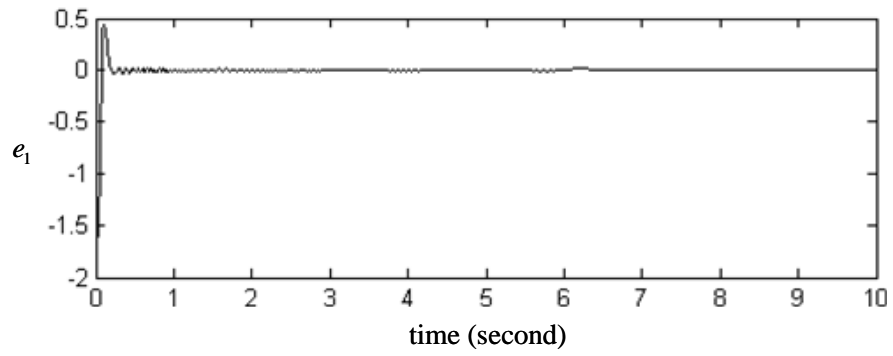
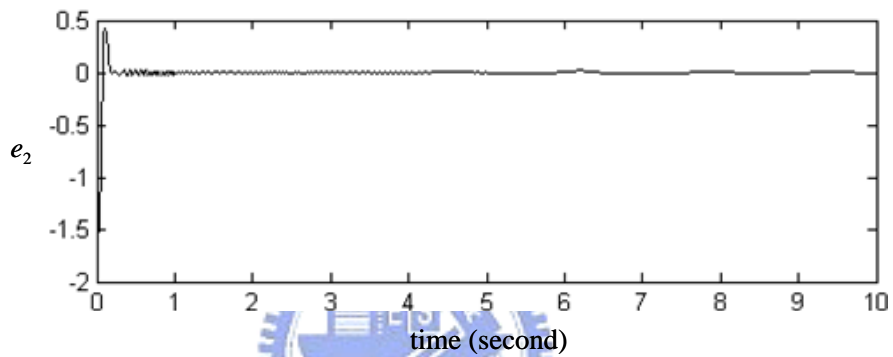


Fig. 3-7. Behavior of identification system.

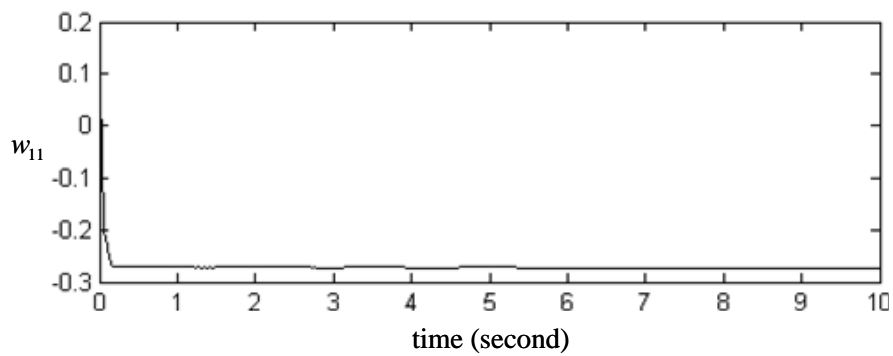


(a)

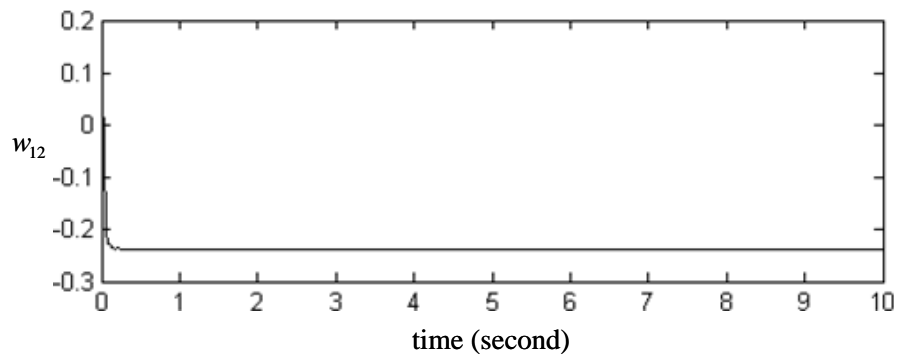


(b)

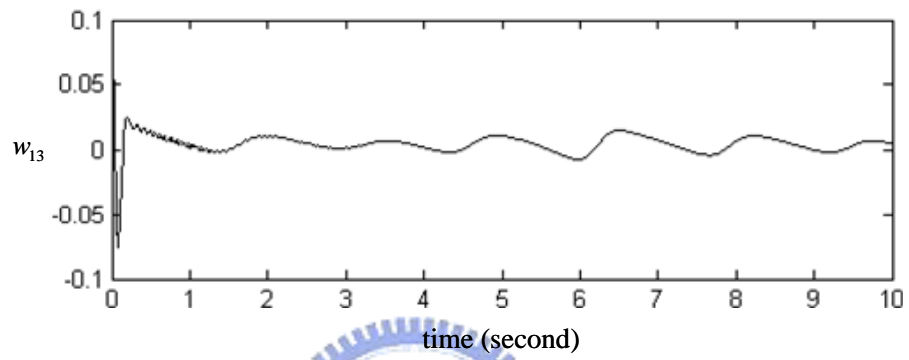
Fig. 3-8. The error of identification.



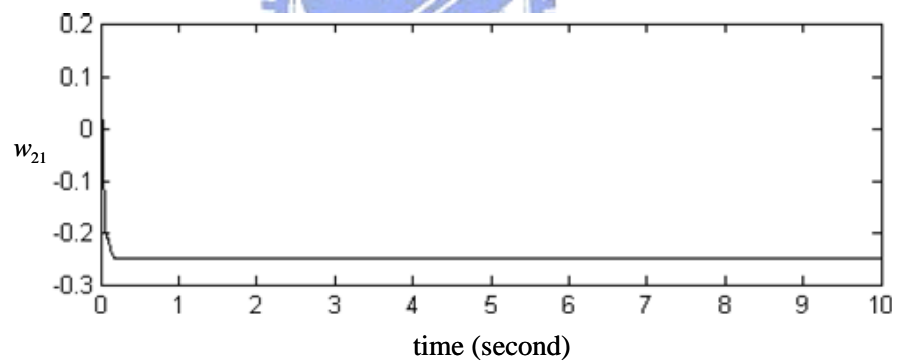
(a)



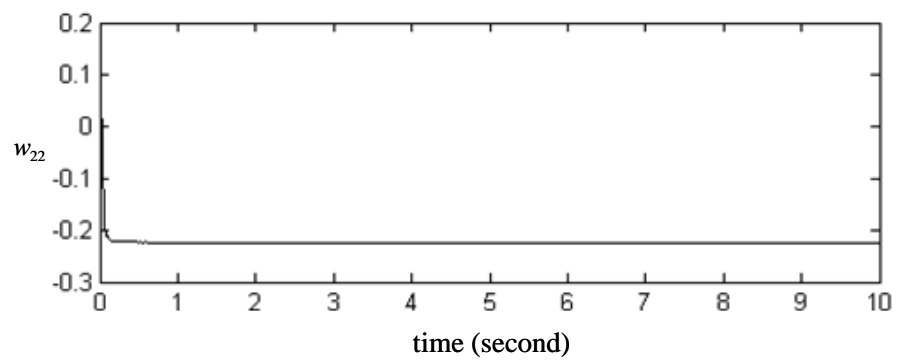
(b)



(c)



(d)



(e)

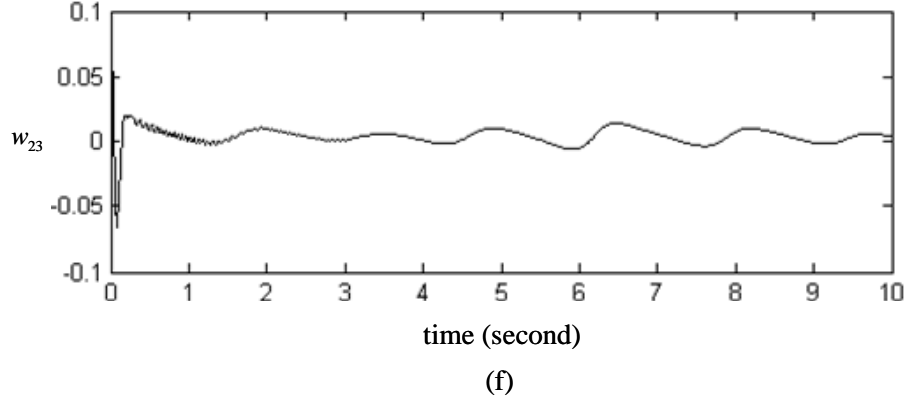


Fig. 3-9. The training conditions of weighting factors.

3.5 Simulation Results of Non-affine System

The model of this example is referred to as Reference [26, 61] and described as

$$\begin{aligned} \dot{x} &= 1.1(1-x-2xu-x^2)e^{-0.5x^2-u^2} \\ u &= \cos t \end{aligned} \quad (3-51)$$

where the initial condition is set as $x(0) = 0$. Fig. 3-10 shows the training data obtained from the closed-loop identification experiment in advance. The parameters of the DNN based on the HNN are selected as $R = 100 \Omega$, $C = 0.01F$, $a_i = 1$, $q = 1$, and $\mathbf{Q} = \text{diag}[2 \ 2]$. The learning rates of weighting factors are selected as $\eta_\phi = 1.5$ and $\eta_u = 0.3$. The initial weight values are all set as 0.2. The following simulation results show the identification ability of the proposed network. The simulation results are shown in Figs. 3-11 and 3-12. The command trajectory x_c and the response of state x is shown in Fig. 3-11(a); Fig. 3-11(b) is the enlarging drawing of Fig. 3-11(a) from $t = 0$ second to $t = 2$ second; and Fig. 3-11(c) shows the approximated errors. Fig. 3-12 shows the training process of weighting factors. In order to examine the robustness of the proposed scheme, a disturbance is added to the system after 10 seconds and the nonlinear system (3-51) is become as:

$$\begin{aligned} \dot{x} &= 1.1(1-x-2xu-x^2)e^{-0.5x^2-u^2} + \zeta \\ u &= \cos t \end{aligned} \quad (3-52)$$

where ζ is defined as $0.5 \sin(2t)$. The simulation results are shown in Figs. 3-13 and 3-14. The command trajectory x_c and the response of state x is shown in Fig. 3-13(a); Fig. 3-13(b) is the enlarging drawing of Fig. 3-13(a) from $t = 9.8$ second to $t = 11$ second; and Fig. 3-13(c) shows the approximated errors. Fig. 3-14 shows the training process of weighting

factors.

According to two simulation results, it is obvious that the proposed scheme can display good identified performance. Even though the nonlinear dynamic system encounters a disturbance suddenly, the proposed structure still maintains good identified property. The modified structure of the dynamic neural network also shows its flexibility. The simulation results also show a good robust ability of the proposed scheme in the *section 3.5*.

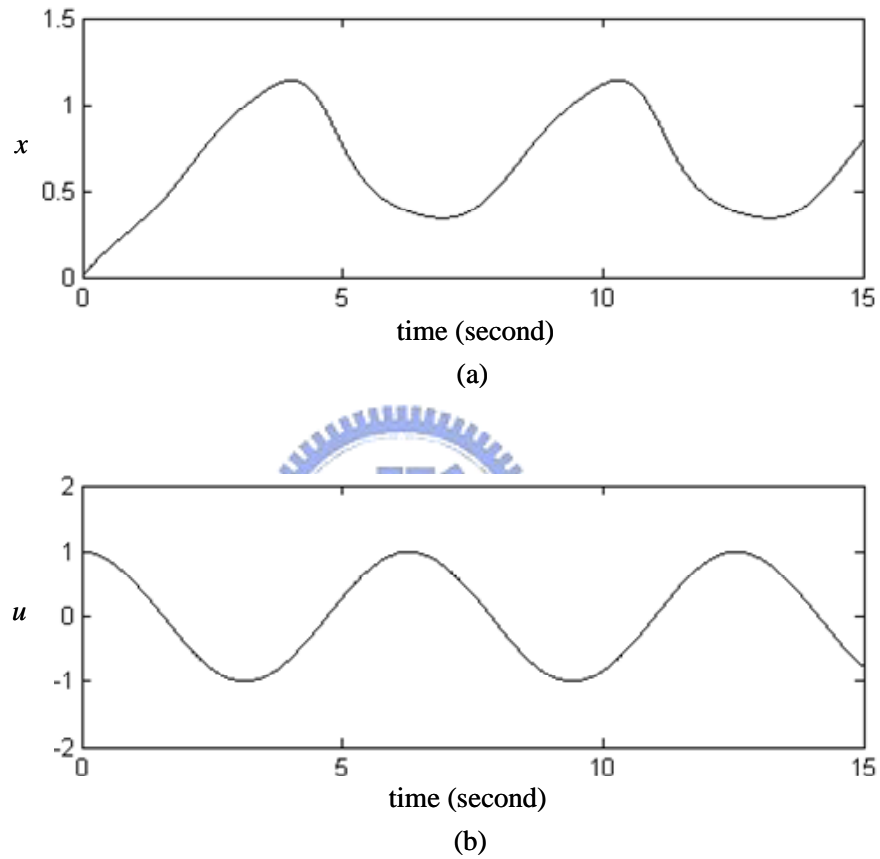


Fig. 3-10. Training data obtained from experimenting in advance.

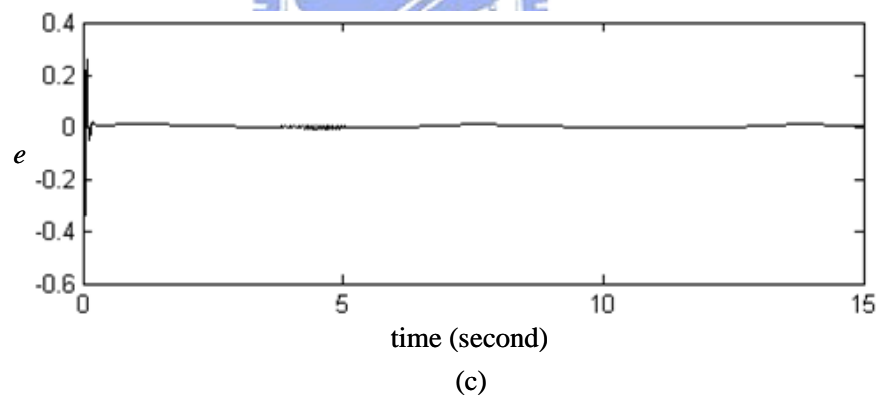
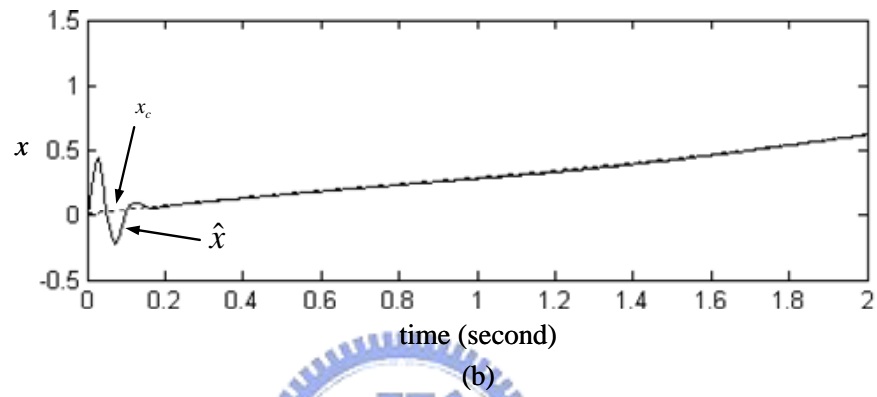
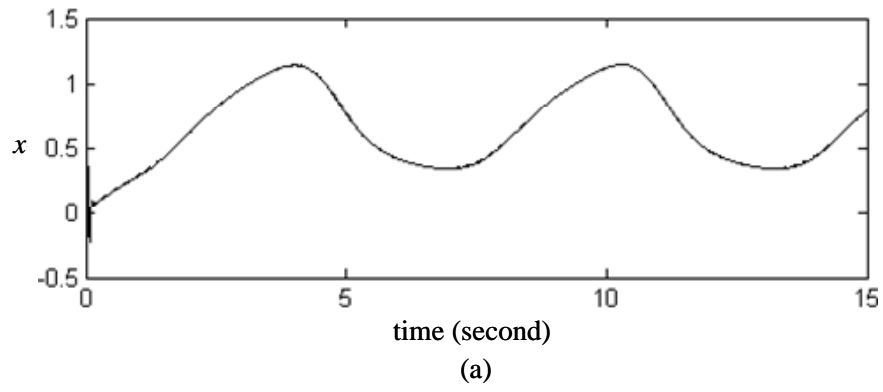


Fig. 3-11. Information of identification system. (a) is the response of state x ; (b) is the enlarging drawing of (a); and (c) shows the approximated errors.

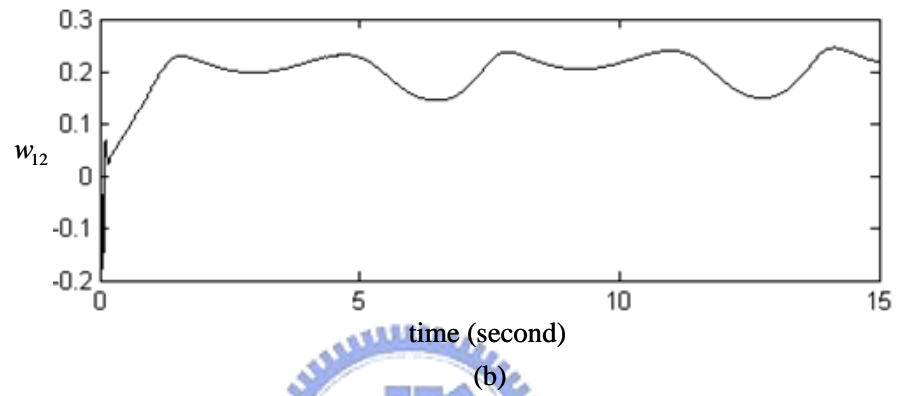
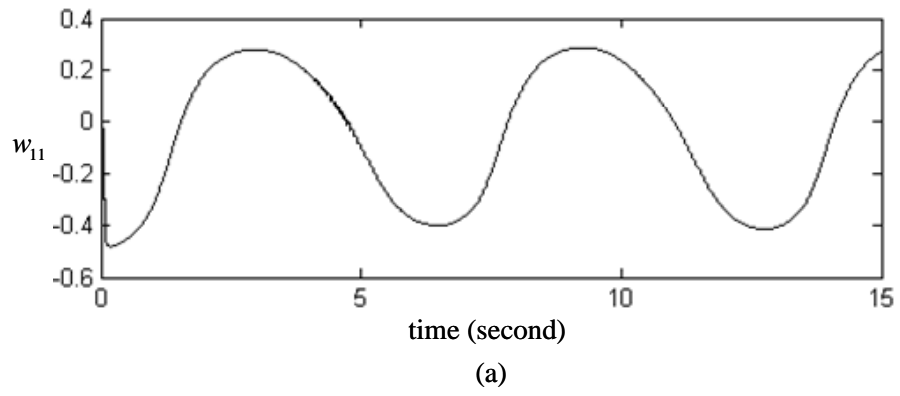
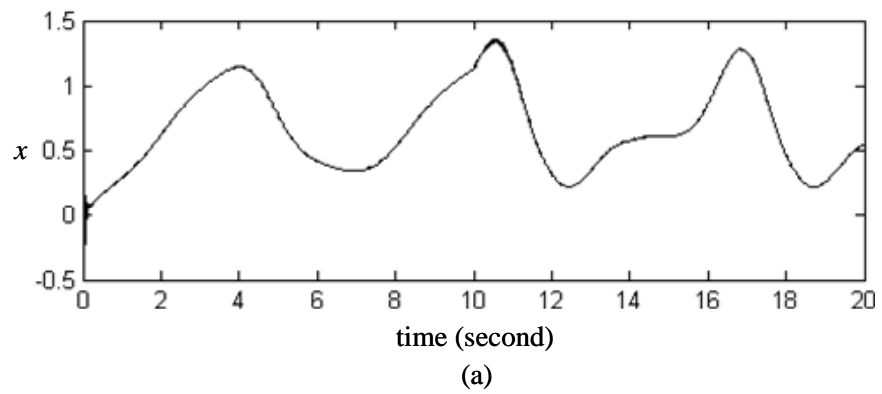


Fig. 3-12. The training conditions of weighting factors.



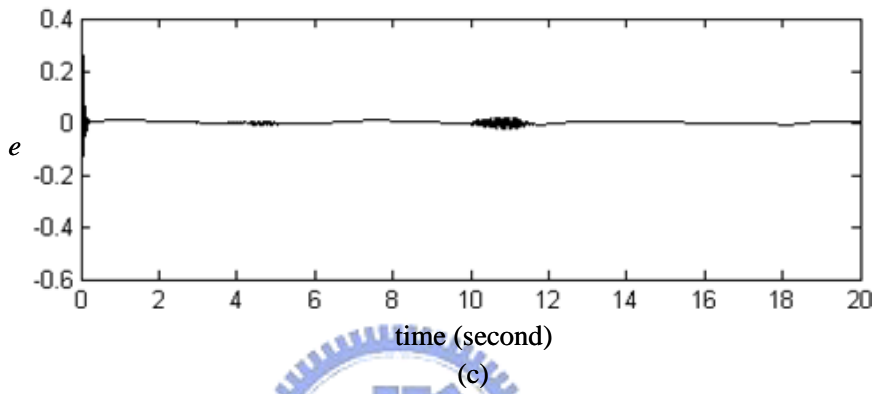
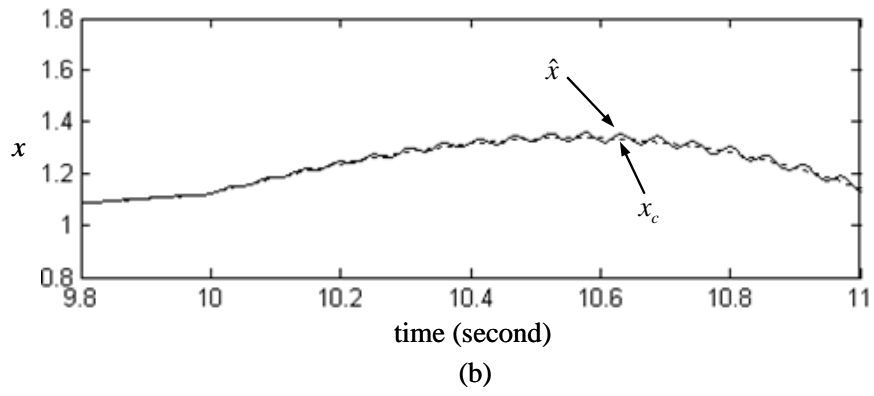
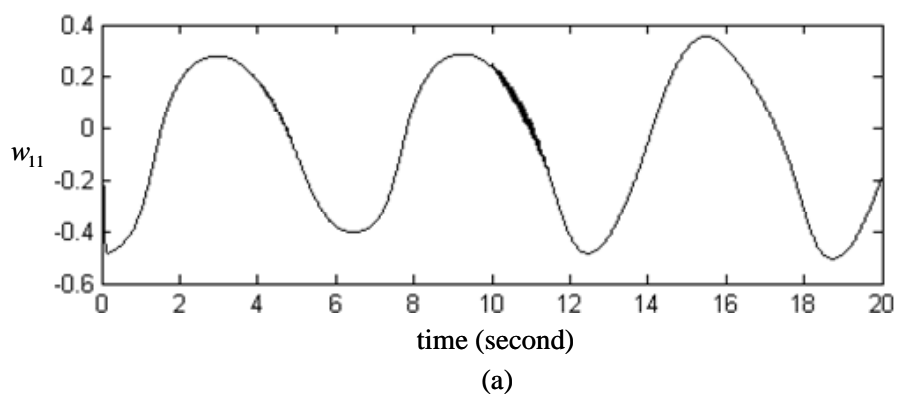


Fig. 3-13. Information of identification system. (a) is the response of state x ; (b) is the enlarging drawing of (a); and (c) shows the approximated errors.



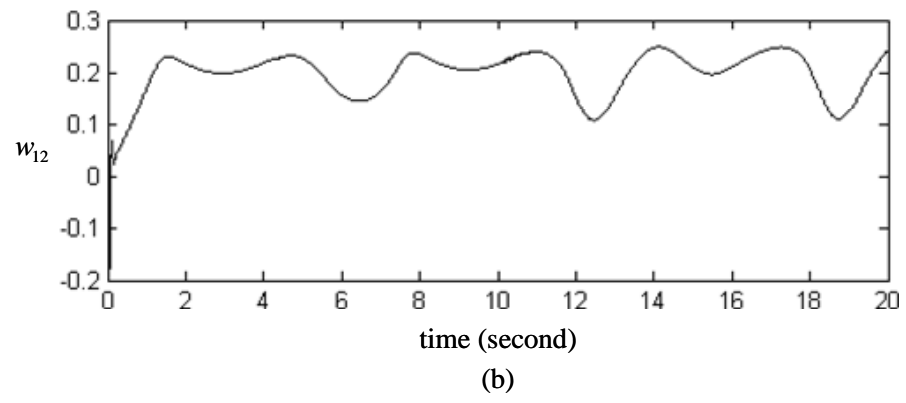


Fig. 3-14. The training conditions of weighting factors.



Chapter 4

Performance Comparison between SFNN and Hopfield-based DNN

After successfully completing the scheme developments and simulations for the static and dynamic neural networks, we will discuss the following two issues on the implementations of software and hardware.

4.1 Software Analysis with Implementation

For the fuzzy system or the fuzzy neural network, membership functions and fuzzy rules must be pre-determined by the trial-and-error method. This is a time-consuming work because we must consider the balance between the rule number and the desired performance. When the SFNN scheme is proposed, this annoying work must be solved first. In addition, for the control theories, the problem of acquiring effectively the unknown system information must be solved when the DNN identifier is developed. In the processes of two computer simulations, the program code of the Hopfield-based DNN is simpler than one of the SFNN because the Hopfield-based DNN is a single-layer structure, but the SFNN has four layers. Moreover, the SFNN has the structure adaptation algorithm. This algorithm also results in complex codes. Although membership functions and fuzzy rules for the SFNN are not determined in advance, there are several parameters of the structure adaptation algorithm which need to be predetermined by the designers in the beginning of simulation. If many membership functions and fuzzy rules are grown at a moment, the computation load may become heavy instantly.

For the implementation of computer simulation of the Hopfield-based DNN, the code is brief because it just has a single layer. However, the selection of values of capacitance and resistance is an annoying problem. In Chapter 3, the matrices, \mathbf{A} and \mathbf{B} , are consisted of capacitance and resistance. If the values of these two matrices are too large, the sampling time of simulation must be set very short and the learning rates must be chosen very small to avoid

the system output diverging immediately. Perhaps, the Hopfield-based DNN is more suitable to be realized in the real-time environment than to be simulated by the computer. In addition, the Hopfield-based DNN can be a de-stabilized system by improper initial conditions. As long as the little problem of selecting the values of parameters is solved, the simulation of the Hopfield-based DNN can be accomplished easily.

In summary, although the SFNN and the Hopfield-based DNN both have some drawbacks for the software implementation, they can both show good desired performance based on the simulation results in the above chapters.

4.2 Hardware Analysis with Implementation

For the hardware implementation, the Hopfield-based DNN is still simpler than the SFNN if the learning algorithm is neglected. The network structure of SFNN consists of many membership functions and fuzzy rules. It is difficult to implement the complex computation process including fuzzifying, inferring, and defuzzifying. We must adopt a single-chip microcontroller 8051, a semiconductor device FPGA (field-programmable gate array), or a DSP (digital signal processing) to implement the structure of the SFNN. We also need a great number of memories to store the data of means, variances, and weights. Moreover, it is also a hard work to implement the structure adaptation algorithm of the SFNN. The problems, which include the complex computation and a great number of memories, must be considered. Because the growing or pruning of membership functions and fuzzy rules must be decided in time, we need the faster CPU to execute these programs. When the procedure of the growing of membership functions and fuzzy rules is executed, more memories are required. When the procedure of the pruning of membership functions and fuzzy rules is executed, more memories, which are assigned in advance, are wasted. Therefore, the implementation of the SFNN is really a difficult work.

For the Hopfield-based DNN, we just need some capacitance, resistance, and summing junctions to implement the identifier circuit if the learning algorithm is neglected. It is a simple network which is implemented possibly. All algorithms of the Hopfield-based DNN can be executed by the circuit. According to the voltage of capacitance, we can obtain the solution. However, the selection of capacitance and resistance is an important issue. The values of capacitance and resistance will affect the dynamic behavior of the Hopfield-based DNN. When we consider the learning algorithms for both networks, the implementations may

become difficult works because the computation load and the implemented approaches are a big problem. However, the implementation of the Hopfield-based DNN is still simpler than one of the SFNN because there are two adjustable parameters for the Hopfield-based DNN, but the SFNN has four adjustable parameters. Therefore, the implementation of the Hopfield-based DNN is a more possible work than one of the SFNN.

4.3 Summary

TABLE 4-1. The comparison result between SFNN and Hopfield-based DNN for the software and hardware.

	SFNN	Hopfield-based DNN
Program Code	Long	Brief
Memory requirement	More	Medium
Performance	Good	Good
Implementation by software	Hard	Easy
Implementation by hardware	Hard (by the faster CPU)	Easy (by a simple circuit)
Cost	High	Low

Chapter 5

Conclusions with Future Works

For the fuzzy neural network (FNN) control design, the structure of the FNN should be determined in advance by the empiricism. It is difficult to consider the balance between the rule number and the desired performance. Therefore, in Chapter 2 of this dissertation, we develop an adaptive self-structuring asymmetric fuzzy neural-network control (ASAFNC) system, which consists of a self-structuring fuzzy neural network (SFNN) controller and a robust controller. In the SFNN controller, SFNN, which adopts asymmetric Gaussian membership functions in the structure and parameter learning phases, is utilized to mimic an ideal controller. The structure learning phase of SFNN is used to find how many rules and membership functions are necessary, and the parameter learning phase of SFNN is concerned with the parameter values of membership functions in the premise part and the crisp value in the consequence part. The robust controller is designed to compensate for the modeling error between the SFNN controller and the ideal controller. An online training methodology is developed in the Lyapunov sense, and thus the stability of the closed-loop control system can be guaranteed. The simulation results of a chaotic dynamics system show that the ASAFNC can achieve favorable tracking performance without control system dynamics.

In addition, in many researches of control theories and system analyses, it is very important to understand the system model. In order to acquire the sufficient system information, a new dynamic neural network (DNN) based on the Hopfield neural network (HNN) is proposed to perform the nonlinear system identification in Chapter 3 of this dissertation. Lyapunov's method is applied to derive the adaptive laws of weighting factors of Hopfield-based DNN. The guarantee of convergence for the identification process is examined by Lyapunov stability theory. The simulation results demonstrate that the proposed identification scheme can achieve good identified performance which is consistent with the convergent analysis discussed in Chapter 3 of this dissertation. The modified structure of the DNN also shows its flexibility.

Although the structure of the SFNN has been developed well, the universal approximation theorem of the SFNN is not still explored. Thus, one of the future works is to derive the universal approximation theorem of the SFNN. In order to avoid the possible

situation about the high degree of overlapping of membership functions, the similarity measure method and the merged algorithm for membership functions and fuzzy rules also need to be developed and the weighting factor needs to be determined after merging fuzzy rules. In addition, due to the simple architecture of the Hopfield-based DNN, we will try to implement the hardware of the identifier scheme based on the proposed scheme and theorem. Because the Hopfield-based DNN identifier has been developed successfully in this dissertation, we will tend towards the control design by the Hopfield-based DNN in the future.



Reference

- [1] C. C. Ku and K. Y. Lee, "Diagonal recurrent neural networks for dynamic systems control," *IEEE Trans. Neural Networks*, Vol. 6, No. 1, pp. 144-156, Jan. 1995.
- [2] L. A. Zadeh, "Fuzzy sets," *Information and Control*, Vol. 8, pp. 338-353, 1965.
- [3] Y. T. Hsu and C. M. Chen, "A novel fuzzy logic system based on N-version programming," *IEEE Trans Fuzzy Systems*, Vol. 8, No. 2, pp. 155-170, April 2000.
- [4] C. T. Lin and C. S. George Lee, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*, Prentice Hall International, 1999.
- [5] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115-133, 1943.
- [6] Y. Q. Zhang and A. Kandel, "Compression and expansion of fuzzy rule based by using crisp-fuzzy neural networks," *Cybernetics and Systems: An International Journal*, Vol. 29, No. 1, pp. 5-34, 1998.
- [7] J. Lee and G. Lee, "Gait angle prediction for lower limb orthotics and prostheses using an EMG signal and neural networks," *International Journal of Control, Automation and System*, Vol. 3, No. 2, pp. 152-158, 2005.
- [8] H. C. Cho, M. S. Fadali, M. S. Saiidi, and K. S. Lee, "Neural network active control of structures with earthquake excitation," *International Journal of Control, Automation and System*, Vol. 3, No. 2, pp. 202-210, 2005.
- [9] S. Huh, K. Lee, D. Kim, I. Choy, and G. Park, "Sensorless speed control system using a neural network," *International Journal of Control, Automation and System*, Vol. 3, No. 4, pp. 612-619, 2005.
- [10] D. Nauck, F. Klawonn, and R. Kruse, *Foundations of Neuro-Fuzzy Systems*, New York: Wiley, 1997.
- [11] A. Chatterjee, K. Pulasinghe, K. Watanabe, and K. Izumi, "A particle-swarm-optimized fuzzy-neural network for voice-controlled robot systems," *IEEE Trans. Industrial Electronics*, Vol. 52, No. 6, pp. 1478-1489, Dec. 2005.
- [12] Y. G. Leu, W. Y. Wang, and T. T. Lee, "Observer-based direct adaptive fuzzy-neural control for nonaffine nonlinear systems," *IEEE Trans. Neural Networks*, Vol. 16, No. 4, pp. 853-861, July 2005.
- [13] C. M. Lin and C. F. Hsu, "Supervisory recurrent fuzzy neural network control of wing

- rock for slender delta wings,” *IEEE Trans. Fuzzy Systems*, Vol. 12, No. 5, pp. 733-742, Oct. 2004.
- [14] C. H. Wang, H. L. Liu, and T. C. Lin, “Direct adaptive fuzzy-neural control with state observer and supervisory controller for unknown nonlinear dynamical systems,” *IEEE Trans. Fuzzy Systems*, Vol. 10, No. 1, pp. 39-49, Feb. 2002.
- [15] G. B. Huang, P. Saratchandran, and N. Sundararajan, “An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks,” *IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 34, No. 6, pp. 2284-2292, Dec. 2004.
- [16] G. Leng, T. M. McGinnity, and G. Prasad, “An approach for on-line extraction of fuzzy rules using a self-organising fuzzy neural network,” *Fuzzy Sets and Systems*, Vol. 150, No.2, pp. 211-243, March 2005.
- [17] C. Li, C. Y. Lee, and K. H. Cheng, “Pseudo-error-based self-organizing neuro-fuzzy system,” *IEEE Trans. Fuzzy Systems*, Vol. 12, No. 6, pp. 812-819, Dec, 2004.
- [18] C. J. Lin and Y. J. Xu, “A self-adaptive neural fuzzy network with group-based symbiotic evolution and its prediction applications,” *Fuzzy Sets and Systems*, Vol. 157, No.8, pp. 1036-1056, April 2006.
- [19] C. T. Lin, W. C. Cheng, and S. F. Liang, “An on-line ICA-mixture-model-based self-constructing fuzzy neural network,” *IEEE Trans. Circuits and Systems I*, Vol. 52, No. 152, pp. 207-221, Jan. 2005.
- [20] Y. Gao and M. J. Er, “Online adaptive fuzzy neural identification and control of a class of MIMO nonlinear systems,” *IEEE Trans. Fuzzy Systems*, Vol. 11, No. 4, pp. 462-477, Aug. 2003.
- [21] C. F. Hsu, “Self-organizing adaptive fuzzy neural control for a class of nonlinear systems,” *IEEE Trans. Neural Networks*, Vol. 18, No. 4, pp. 1232-1241, July 2007.
- [22] F. J. Lin, C. H. Lin, and P. H. Shen, “Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive,” *IEEE Trans. Fuzzy Systems*, Vol. 9, No. 5, pp. 751-759, Oct. 2001.
- [23] 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, March 2004.
- [24] S. Wu and M. J. Er, “Dynamic fuzzy neural networks - a novel approach to function approximation,” *IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 30, No. 2, pp. 358-364, April 2000.
- [25] S. Wu, M. J. Er, and Y. Gao, “A fast approach for automatic generation of fuzzy rules by

- generalized dynamic fuzzy neural networks,” *IEEE Trans. Fuzzy Systems*, Vol. 9, No. 4, pp. 578-594, Aug. 2001.
- [26] G. P. Liu, V. Kadiramanathan, and S. A. Billings, “Stable sequential identification of continuous nonlinear dynamic systems by growing radial basis function networks,” *International Journal of Control*, Vol. 65, No. 1, pp. 53-69, 1996.
- [27] K. S. Narendra and K. Parthasarathy, “Identification and control of dynamical systems using neural networks,” *IEEE Trans. Neural Networks*, Vol. 1, No. 1, pp. 4-27, March 1990.
- [28] A. S. Poznyak, Y. Wen, E. N. Sanchez, and J. P. Perez, “Nonlinear adaptive trajectory tracking using dynamic neural networks,” *IEEE Trans. Neural Networks*, Vol. 10, No. 6, pp. 1402-1411, Nov. 1999.
- [29] S. Haykin, *Neural Networks - A Comprehensive Foundation* (2nd Edition), Prentice Hall, 1999.
- [30] I. Chairez, A. Poznyak, and T. Poznyak, “New sliding-mode learning law for dynamic neural network observer,” *IEEE Trans. Circuits and Systems II*, Vol. 53, No. 12, pp. 1338-1342, Dec. 2006.
- [31] Y. M. Park, M. S. Choi, and K. Y. Lee, “An optimal tracking neuro-controller for nonlinear dynamic systems,” *IEEE Trans. Neural Networks*, Vol. 7, No. 5, pp. 1099-1110, Sept. 1996.
- [32] D. T. Pham and X. Liu, “Dynamic system identification using partially recurrent neural networks,” *Journal of Systems Engineering*, Vol. 2, No. 2, pp. 90-97, 1992.
- [33] A. Poznyak, E. Sanchez, and W. Yu, *Dynamic Neural Networks for Nonlinear Control: Identification, State Estimation and Trajectory Tracking*, London, U.K.: World Scientific, 2001.
- [34] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” *Proceedings of National Academy of sciences, USA*, Vol. 79, pp. 2554-2558, April 1982.
- [35] J. J. Hopfield, “Neurons with graded response have collective computational properties like those of two-state neurons,” *Proceedings of National Academy of sciences, USA*, Vol. 81, pp. 3088-3092, May 1984.
- [36] M. A. Li and X. G. RUAN, “Optimal control with continuous Hopfield neural network,” *IEEE. Proceedings of International Conference on Robotics, Intelligent Systems and Signal*, pp. 758-762, 2003.
- [37] J. J. Hopfield and D. W. Tank, “Neural computation of decisions in optimization

- problems,” *Biological Cybernetics*, Vol. 52, pp. 1-25, 1985.
- [38] L. Wang, Y. S. Xiao, G. Zhou, and Q. Wu, “Further discussion of Hopfield neural network based DC drive system identification and control,” *IEEE. Proceedings of Congress on Intelligent Control and Automation*, pp. 1990-1993, 2002.
- [39] W. Y. Wang, I. H. Li, S. F. Su, and C. W. Tao, “Identification of four types of high-order discrete-time nonlinear systems using Hopfield neural networks,” *Dynamics of Continuous, Discrete and Impulsive Systems, Series A: Mathematical Analysis*, Special Issue: Advances in Neural Networks--Theory and Applications, Vol. 14, pp. 57-66, 2007.
- [40] J. S. Lin, K. S. Chen, and C. W. Mao, “A fuzzy Hopfield neural network for medical image segmentation,” *IEEE. Trans. Nuclear Science*, Vol. 43, No. 4, pp. 2389-2398, Aug. 1996.
- [41] G. Pajares, “A Hopfield neural network for image change detection,” *IEEE Trans. Nuclear Science*, Vol. 17, No. 5, pp. 1250-1264, Sept. 2006.
- [42] C. S. Velayutham and S. Kumar, “Asymmetric subethood-product fuzzy neural inference system (ASuPFuNIS),” *IEEE Trans. Neural Networks*, Vol. 16, No. 1, pp. 160-174, Jan. 2005.
- [43] J. J. E. Slotine and W. P. Li, *Applied Nonlinear Control*, Englewood Cliffs, NJ: Prentice Hall, 1991.
- [44] L. X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*, Englewood Cliffs, NJ: Prentice-Hall, 1994.
- [45] J. H. Park, G. T. Park, S. H. Kim, and C. J. Moon, “Direct adaptive self-structuring fuzzy controller for nonaffine nonlinear system,” *Fuzzy Sets and Systems*, Vol. 153, No. 3, pp. 429-445, Aug. 2005.
- [46] C. F. Hsu, C. M. Lin, and T. T. Lee, “Wavelet adaptive backstepping control for a class of nonlinear systems,” *IEEE Trans. Neural Networks*, Vol. 17, No. 5, pp. 1175-1183, Sept. 2006.
- [47] W. Y. Wang, M. L. Chan, C. C. J. Hsu, and T. T. Lee, “ H^∞ tracking-based sliding mode controller for uncertain nonlinear systems via an adaptive fuzzy-neural approach,” *IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 32, No. 4, pp. 483-492, Aug. 2002.
- [48] Y. G. Leu, T. T. Lee, and W. Y. Wang, “Observer-based adaptive fuzzy-neural control for unknown nonlinear dynamical systems,” *IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 29, No. 5, pp. 583-591, Oct. 1999.

- [49] C. M. Lin and Y. F. Peng, "Adaptive CMAC-based supervisory control for uncertain nonlinear systems," *IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 34, No. 2, pp. 1248-1260, April 2004.
- [50] R. J. Wai, C. M. Lin, and C. F. Hsu, "Adaptive fuzzy sliding-mode control for electrical servo drive," *Fuzzy Sets and Systems*, Vol. 143, No. 2, pp. 295-310, 2004.
- [51] K. H. Cheng, C. F. Hsu, C. M. Lin, T. T. Lee, and C. Li, "Fuzzy-neural sliding-mode control for DC-DC converters using asymmetric Gaussian membership functions," *IEEE Trans. Industrial Electronics*, Vol. 54, No. 3, pp. 1528-1536, June 2007.
- [52] A. N. Michel, J. A. Farrel, and W. Porod, "Qualitative analysis of neural networks," *IEEE Trans. Circuits and Systems*, Vol. 36, No. 2, pp. 229-243, Feb. 1989.
- [53] H. Khalil, *Nonlinear Systems*, New York: Macmillan, 1992.
- [54] J. J. Hopfield and D. W. Tank, "Computing with neural circuits: A model," *Science*, Vol. 233, No. 4764, pp. 625-633, 1986.
- [55] D. Liu and A. N. Michel, "Sparsely interconnected neural networks for associative memories with applications to cellular neural networks," *IEEE Trans. Circuits and Systems II*, Vol. 41, No 4, pp. 295-307, April 1994.
- [56] L. O. Chua and L. Yang, "Cellular neural networks: theory," *IEEE Trans. Circuits and Systems*, Vol. 35, No, 10, pp. 1257-1272, Oct. 1988.
- [57] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: perceptron, Madaline, and backpropagation," *Proceedings of the IEEE*, Vol. 78, No. 9, pp. 1415-1442, Sept. 1990.
- [58] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two states neurons," *Proceedings of National Academy of sciences, USA*, Vol. 81, pp. 3088-3092, 1982.
- [59] E. B. Kosmatopoulos, M. M. Polycarpou, M. A. Christodoulou, and P. A. Ioannou, "High-order neural network structures for identification of dynamical systems," *IEEE Trans. Neural Networks*, Vol. 6, No. 2, pp. 422-431, Mar 1995.
- [60] V. M. Becerra, F. R. Garces, S. J. Nasuto, and W. Holderbaum, "An efficient parameterization of dynamic neural networks for nonlinear system identification," *IEEE Trans. Neural Networks*, Vol. 16, No. 4, pp. 983-988, July 2005.
- [61] W. F. Xie and A. B. Rad, "Fuzzy on-line identification of SISO nonlinear systems," *Fuzzy Sets and Systems*, Vol. 107, No. 3, pp. 323-334, Nov. 1999.

Vita

Name: Ping-Zong Lin

Personal:

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

Day of Birth: July 27, 1978

Gender: Male

Education background:

<u>Degree</u>	<u>Date</u>	<u>School</u>
B.S. E.E.	1997/9 - 2001/6	Fu-Jen Catholic University, Department of Electronic Engineering
M.S. E.E.	2001/9 - 2003/6	Fu-Jen Catholic University, Department of Electronic Engineering
Ph.D. E.C.E.	2003/9 - 2008/7	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] P. Z. Lin, C. M. Lin, C. F. Hsu, and T. T. Lee, "Type-2 fuzzy controller design using a sliding-mode approach for application to DC-DC converters," *IEE Proc. Electric Power Applications*, Vol. 152, No. 6, pp. 1482-1488, Nov. 2005.
- [2] P. Z. Lin and T. T. Lee, "Robust self-organizing fuzzy-neural control using asymmetric Gaussian membership functions," *International Journal of Fuzzy Systems*, Vol. 9, No. 2, pp. 77-86, 2007.
- [3] G. M. Chen, P. Z. Lin, W. Y. Wang, T. T. Lee, and C. H. Wang, "Image-based fuzzy control system," *IET Electronics Letters*, Vol. 44, No. 7, pp. 461-462, March 2008.
- [4] C. F. Hsu, P. Z. Lin, T. T. Lee, and C. H. Wang, "Adaptive asymmetric fuzzy neural network controller design via network structuring adaptation," accepted to be published in *Fuzzy Sets and Systems*, 2008.
- [5] P. Z. Lin, C. F. Hsu, T. T. Lee, and C. H. Wang, "Robust fuzzy-neural sliding-mode controller design via network structure adaptation," accepted to be published in *IET Control Theory & Applications*, 2008.

Submitted Journal Papers:

- [1] P. Z. Lin, W. Y. Wang, T. T. Lee, and C. H. Wang, "On-line genetic algorithm-based fuzzy-neural sliding mode controller using improved adaptive bound reduced-form genetic algorithm," submitted to *International Journal of Systems Science*, Oct. 2007.
- [2] C. H. Wang, P. Z. Lin, P. C. Chen, and T. T. Lee, "A dynamic neural network model for nonlinear system identification," submitted to *IEEE Trans. Systems, Man, and Cybernetics*, July 2008.

International Conference Papers:

- [1] P. Z. Lin, W. Y. Wang, T. T. Lee, and Y. G. Leu, "GA-based fuzzy-neural sliding mode controllers of robot manipulators," *2004 The Twelfth National Conference on Fuzzy Theory and It's Applications*, 2004.

- [2] C. F. Hsu, T. T. Lee, P. Z. Lin, C. M. Lin, and S. J. Liu, “Self-organizing fuzzy control for DC-DC converters,” in *Proc. of 11th International Fuzzy Systems Association World Congress*, pp. 788-793, 2005.
- [3] P. Z. Lin, C. F. Hsu, and T. T. Lee, “Type-2 fuzzy logic controller design for buck DC-DC converters,” in *Proc. of 2005 IEEE International Conference on Fuzzy System, Reno, USA*, pp. 365-370, 2005.
- [4] P. Z. Lin, W. Y. Wang , T. T. Lee, and G. M. Chen, “On-line genetic fuzzy-neural sliding mode controller design,” in *Proc. of 2005 IEEE International Conference on Systems, Man and Cybernetics, Waikoloa, Hawaii*, pp245-250, Oct. 2005.
- [5] C. F. Hsu, T. T. Lee, and P. Z. Lin, “Intelligent tracking controller design using dynamic fuzzy neural networks,” in *Proc. of 2007 IEEE International Joint Conference on Neural Networks, Florida, Orlando, USA*, pp. 148-153, Aug. 2007.
- [6] G. M. Chen, P. Z. Lin, W. Y. Wang, T. T. Lee, and C. H. Wang, “Fuzzy control using intuitive image analysis,” *Conference on Human System Interaction, Krakow, Poland*, May 2008.

