Nonlinear System Identification and Control using Dynamic Multi-Time Scales Neural Networks

Xuan Han

A Thesis
In
The Department
of
Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science (Mechanical Engineering) at
Concordia University
Montreal, Quebec, Canada

April, 2010

Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-67204-4
Our file Notre référence
ISBN: 978-0-494-67204-4

Abstract

Nonlinear System Identification and Control using Dynamic Multi-Time Scales Neural Networks

In this thesis, on-line identification algorithm and adaptive control design are proposed for nonlinear singularly perturbed systems which are represented by dynamic neural network model with multi-time scales. A novel on-line identification law for the Neural Network weights and linear part matrices of the model has been developed to minimize the identification errors. Based on the identification results, an adaptive controller is developed to achieve trajectory tracking. The Lyapunov synthesis method is used to conduct stability analysis for both identification algorithm and control design. To further enhance the stability and performance of the control system, an improved dynamic neural network model is proposed by replacing all the output signals from the plant with the state variables of the neural network. Accordingly, the updating laws are modified with a dead-zone function to prevent parameter drifting. By combining feedback linearization with one of three classical control methods such as direct compensator, sliding mode controller or energy function compensation scheme, three different adaptive controllers have been proposed for trajectory tracking. New Lyapunov function analysis method is applied for the stability analysis of the improved identification algorithm and three control systems. Extensive simulation results are provided to support the effectiveness of the proposed identification algorithms and control systems for both dynamic NN models.

Acknowledgements

This thesis is finished during the research in the Mechanical and Industrial Engineering Department of Concordia University. It is a great opportunity to study here. I am very grateful to all the professors, technicians and fellow students whom I have worked with in this research field.

Firstly, I would like to express my deep gratitude to my supervisor Dr. Wenfang Xie. She provides me with the inspiration and direction in my research. Critical feedback from her keeps me progress throughout my graduate education. Her enthusiasm, creativity and concentration dramatically inspired me to improve the quality of the research and conquer the difficulties.

Secondly, I am deeply indebted to the colleagues and friends in Montreal for their help and advices.

Finally, I would like to thank my parents for always believing in me.

Xuan Han
Montreal Canada

Table of Contents

List of Figures

List of Tables

List of Symbols, Abbreviations and Nomenclature

Symbol	Definition
NN	Neural Network
ANN	Artificial Neural Network
SISO	Single Input Single Output
MIMO	Multi-Input Multi-Output
MLP	Multilayer Perceptron
RBF	Radial Basis Function
DMTSNN	Dynamic Multi-Time Scales Neural Networks
HH	Hodgkin-Huxley
RMS	Root Mean Square
ELF	Extremely Low Frequency
BP	Back Propagation
$x_{nn,} y_{nn}$	State variable of Neural network
$W_{1,2,3,4}$	Output layers weight
$V_{1,2,3,4}$	Hidden layers weight
$\sigma(\cdot)$	Activation function
$\phi(\cdot)$	Activation function
U	Control input
A,B	Linear part matrix

ε	Singular Perturbation Parameter
$\gamma(\cdot)$	Input-output mapping function
$W_1^*, W_2^*, W_3^*, W_4^*$	Nominal constant matrices
A^*, B^*	Nominal Hurwitz matrices
$\Delta x_x, \Delta y$	Identification error
$\Delta f_x, \Delta f_y$	Modeling error and disturbances
V_I	Lyapunov function for Identification
$\dot{A}, \dot{B}$	Updating law of linear part matrix
$\dot{W}_{1,2,3,4}$	Updating law of output layers weight
α	Compensation positive constant
V_c	Lyapunov function for control
λ	Eigenvalue
$\Lambda, \Lambda_x, \Lambda_y$	Positive definite matrix
$\alpha_x, \beta_x, \alpha_y, \beta_y$	K_∞ function
$\gamma_{A,}\gamma_B$	Eigenvalue of linear part matrix
RMS	Root mean square
x_d, y_d	Desired Trajectory
E_x, E_y	Tracking control error
$\Delta\bar{f}_x, \Delta\bar{f}_y$	Upper bound of modeling error

S_x, S_y	Dead-zone indicator
H_x, H_y	Identification threshold

Chapter 1 Introduction

1.1 Motivation

Numerous systems in the industrial fields demonstrate nonlinearities and uncertainties which can be considered as partial or total black-box. Dynamic neural networks have been applied in system identification and control of those systems for many years. Due to the fast adaptation and superb learning capability, the dynamic neural networks have transcendent advantages compared to the static ones [14], [15].

A wide class of nonlinear physical systems contains slow and fast dynamic processes that occur at different moments. Recent research results show that neural networks are very effective for modeling the complex nonlinear systems with different time-scales when one has incomplete model information, or even when the plant is considered as a black-box [14].

Dynamic neural networks with different time-scales can model the dynamics of the short-term memory of neural activity levels and the long-term memory on dynamics of unsupervised synaptic modifications [21]. The stability of equilibrium of competitive neural network with short and long-term memory was analyzed in [22] by a quadratic-type Lyapunov function. In [23-25], new methods of analyzing the dynamics of a system with different time scales are presented based on the theory of flow invariance. The K-monotone system theory was used for analyzing the dynamics of a competitive neural system with different time scales in [26].

Since system identification and control using dynamic neural networks (NN) was first introduced systematically in [27], the past decade has witnessed great activities in stability analysis, identification and control with continuous time dynamic neural networks with or without considering the time scales. In [28], Sandoval et al. developed new stability conditions by using a Lyapunov function and singularly perturbed technique. In [29], the passivity-based approach was used to derive stability conditions for dynamic neural networks with different time-scales. The passivity approach was used to prove that a gradient descent algorithm for weight adjustment was stable and robust to any bounded uncertainties, including the optimal network approximation error [8]. Many dynamic neural networks-based direct and indirect adaptive control algorithms for regulation and tracking have been published in the literatures [10, 30, and 31]. With consideration of the uncertainty of dynamic systems, the indirect method that adopts on-line identification via neural networks followed by controller design is developed and widely used. Several papers proposed adaptive nonlinear identification and trajectory [9] or velocity [11] tracking via dynamic neural networks without considering the multiple time scales. However, the above mentioned research has concentrated on the stability analysis instead of control for dynamic systems by using dynamic neural networks with time-scales or developed control scheme based on the neural network without considering time-scales.

In this thesis, on-line identification for nonlinear system with uncertainties using multi-time scales dynamic neural network are developed and then various controllers are designed for trajectory tracking based on the on-line identification results.

1.2 Literature review

1.2.1 Overview of system identification

Since the modeling and parameter estimation are initiated by the mathematical statistics and time series analysis, many disciplines like economic, social science and engineering have participated and contributed to this field. In 1956, Zadeh first introduced the term--System Identification for the problem of identifying a black box by its input-output relationship [32]. Since then, a lot of researches have been devoted to system identification which has become an established branch of control theory. Since systems with unknown linear parameters or unknown nonlinear characteristics cannot be controlled optimally, to identify these unknown linear parameters and nonlinear characteristics is essential in control domain. System identification is a process of estimating the architecture and parameters of a model from the input and output data. From different points of view, system identification can be classified as on-line and off-line identification, or grey box and black box, or linear system and nonlinear system identification.

a) On-line and Off-line identification

Conducting estimation process after collecting the data from the system are known as Off-line identification. On the contrary, these two steps are running at the same time for on-line identification. The main advantages of on-line identification are that the specified precision can be achieved by recursive process and real time identification for time varying system. Model reference techniques of the on-line identification problem are considered by Monopoli for nonlinear non-autonomous plants [34]. Daniel and Robert employed filtered version of the recursive least-squares as identification algorithm [35]. In [36], Kalman proposed a linear-quadratic estimator called Kalman filter. The extended Kalman filter is used to system identification problems of seismic structural systems in [37].

b) Black-box and grey box

Depending on the level of prior knowledge, the identification model can be catalogued into two groups. If absolutely no information about the process is available, the identification plants are notated as black-box. For the other cases, grey-box refers to the situations that considerable knowledge of the structure and/or parameters are already known.

For linear single input single output (SISO) Black-box models, Ljung summarized the general family structure which can rise to 32 different models [38].

$$\begin{aligned}
&A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t) \\
&A(q) = 1 + a_1 q^{-1} + \ldots + a_{na} q^{-na} \\
&B(q) = 1 + b_1 q^{-1} + \ldots + b_{nb} q^{-nb} \\
&C(q) = 1 + a_1 q^{-1} + \ldots + a_{nc} q^{-nc} \\
&D(q) = 1 + a_1 q^{-1} + \ldots + a_{nd} q^{-nd} \\
&F(q) = 1 + a_1 q^{-1} + \ldots + a_{nf} q^{-nf}
\end{aligned} \tag{1.1}$$

where u(t) and y(t) are scale input and output signal for a system, q is forward shift operator defined as $qu(t) = u(t+1)$ and q^{-1} is backward shift operator defined as $q^{-1}u(t) = u(t-1)$.

Some special cases are list below:

FIR – Finite Impulse Response (A=C=D=F=1)

ARX- AutoRegressive with Exogenous input (C=D=F=1)

ARMA - AutoRegressive Moving Average (B=D=F=1)

ARMAX – AutoRegressive Moving Average model with Exogenous inputs model (D=F=1)

ARARX - AutoRegressive AutoRegressive with Exogenous input (C=F=1)

OE- Output Error (A=C=D=1)

BJ – Box-Jenkins (A=1)

Recently the research results show that these methods have been extended for Multi-input Multi-output (MIMO) system. An identification algorithm using FIR model is proposed for multi-input, multi-output stochastic systems [49]. New evolutionary programming method is proposed to identify the ARMAX model for short term load forecasting [52]. Monin has derived a new identification algorithm for OE and ARMAX systems with exogenous nonstationary multiple input based on a hereditary computation [53]. Model identification and diagnostic checking using BJ method are developed for the control of prostheses for varied limb function, movement and circumstances [54].

The regressors for nonlinear system identification are similarly selected, hence the names are inherited with adding "N" representing nonlinear at beginning, like NFIR, NARX, NARMAX NOE and NBJ. In [55], NFIR Volterra model are utilized for a subspace approach of blind identification and equalization of nonlinear single-input multiple-output system. NARX time series are investigated with projections for estimating its endogenous and exogenous components [50]. In [51], a new linear and nonlinear ARMA identification algorithm is developed based on affine geometry. It is faster for the new algorithm to obtain the estimation results than the fast orthogonal search method. Subspace algorithms with a basis function are applied to identify a Hammerstein model expansion which is followed by modeling Wiener nonlinearity to build a model for ionospheric dynamics [56].

c) Linear system and nonlinear system identification

Linear systems obviously represent the most vita group of system identification. After years of extensive development in practice and in the literature, linear system identification techniques have been systematically presented in text books. Even though, there are still some improvement which are made for the past few decades. In [43], Guillaume et. al. discussed and analyzed a mathematical model of the Empirical Transfer Function Estimate with noisy input signals which is not deterministic and exactly known for Fourier analysis. Tugnait proposed a frequency-domain solution to the least-squares equation error identification problem using the power spectrum and the cross-spectrum of the time-domain input-output data to estimate the parametric input-output infinite impulse response transfer function [44]. Overschee and Moor find state-space models by subspace state space system identification algorithms from the input and output data [46].

Before 1980s, system identification techniques for nonlinear systems have received scant attention due to their inherent complexity and difficulty. As nonlinear systems are widely engaged in many different research fields, like control application, artificial intelligence, pattern recognition, signal processing etc., nonlinear system identification become more and more important.

Compared to linear systems, it is very difficult to obtain the precise physical model for nonlinear system and more distinct structural models can be chosen. On the other hand, the models need not to be true and accurate description of the real system. It is just a description of some of its properties to serve certain purpose [38]. Hence, many researchers tend to use reduced-order or linear model to represent the system. Transient response method is used to model the reduced-order transfer functions of power converters by analyzing the step response [39]. Linear models of Tokamak are created for control purpose and validating different models from physics principles by frequency response identification [40]. A continuous-time nonlinear unstable magnetic bearing systems are successfully identified by using a linear model and frequency response data [41].

Jean-Marc and René investigated the identification using nonlinear autoregressive models [47]. Volterra series serves as a generalization of the convolution integral to model a seventh order nonlinear model of a synchronous generator with saturation effect [48]. Nonlinear system identification is conducted by Reduced Volterra model with generalized orthonormal basis functions, which can overcome the huge estimation process [49]. Singh and Subramanian established a direct correspondence between the structure of a nonlinear system and the pattern of its frequency response [42].

1.2.2 Development of control strategies for nonlinear systems

Linear control as a mature topic with various effective methods has been systemically presented in textbook and successfully operated in industrial applications. In early age of development of control engineering, not a wide range of nonlinear analysis tools are available for researchers and engineers as today. Before 1940, some immature nonlinear control methods, like Tirrill regulator and the fly-ball governor have been successfully applied without systematically theoretical analysis [57]. Phase plane method, describing function method and Tsypkin's method for relay systems as the major nonlinear system analysis techniques during the two decades since 1940s are also discussed in [57]. Then the control engineering community boosts attention to Lyapunov's stability theory after more than 60 years since it is first published in 1892. Lyapunov's direct method has become the most fundamental and popular nonlinear system analysis tool for the major nonlinear control system design methods.

1.2.3 NN-based nonlinear system identification and control

When dealing with non-linear systems as well as linear systems with multiple inputs and multiple outputs, traditional identification methods need specific assumptions concerning the model structure. It is usually assumed that the system equations are known except for a number of parameters [45]. Neural networks have been proven to be effective for nonlinear system identification and control due to highly complexity and nonlinearity.

a) Static networks and dynamic networks

The architectures of neural networks can be categorized into two fundamental classes: feedforward (static) networks and recurrent (dynamic) networks. The major difference between them is that recurrent NN has at least one feedback loop.

In the literature, feedforward NNs are most popularly used for nonlinear system identification and control [4, 5, 6]. A typical example is the multilayer perceptron (MLP), which is utilized to identify the dynamic characteristics of a nonlinear system. The main characteristic of MLP—fast convergence makes it prime candidate for adaptive control of nonlinear systems. In [64], MLP NN is used to realize the position control of a Low Earth Orbit satellite. RBF neural networks are artificial neural networks with radial basis functions as activation functions. Similar to feedforward neural networks, RBFs are widely used in function approximation, time series prediction, control, pattern recognition and classification. Mark gave a systematic introduction about RBF neural networks [66]. RBF neural networks were also applied for diagnosis of diabetes mellitus [67], whose performance is evaluated with MLP neural networks and logistic regression. After comparing the performances of a multilayer MLP network and a RBF network for the online identification of a synchronous generator, Jung-wook et. al. claimed that the RBF network is simpler to implement, needs less computational memory, converges faster and better even in the changing operating conditions [65].

On the other hand, the recurrent neural networks have received considerable attention in recent two decades. Due to their strong nonlinear characteristics, dynamic NNs are more and more widely used for nonlinear system identification and control. On-line system identification based on modified recurrent neural network NARX model with three different validation algorithms are presented to serve the predictive controller [68]. Based on a recurrent neural network uncertainty observer, a back-stepping and adaptive combined controller is designed to perform position control of an induction servomotor [69]. The recurrent neural networks are trained based on the experimental data from a continuous biotechnological process for system identification and control [70]. Identification result of a class of control affine systems are used to synthesize the feedback linearization of the system which then can be controlled by a PID controller [71].

b) Learning algorithm

All the neural network topologies are supported by their corresponding training or learning algorithms. Years ago, back propagation is the dominating learning algorithms since it is first introduced by Paul J. Werbos in 1974 [7]. In addition, many efforts have been made to improve the traditional back propagation approach. New back propagation algorithm with optimization process for the slope of sigmoid function at each neuron is presented in [58], which accomplishes faster convergence rate and better accuracy model, especially for high level nonlinear systems, comparing to traditional back propagation method for system identification with neural networks. Various improved back propagation algorithm are presented for recurrent neural networks in [60]. An accelerated back propagation can remove the delay when the error is back-propagated through the adjoin model. Predictive back propagation and targeted back propagation with or without filtering are studied to update the weights of the network. Yue-Seng and Eng-Chong investigate various aspects like net pruning during training, adaptive learning rates for individual weights and biases, adaptive momentum, and extending the role of the neuron in learning and then combing them together to improve the performance of back propagation for multilayer feed-forward neural networks [61].

Now many new network topologies with the corresponding training algorithm are proposed for system identification and control purpose. Widely discussed recently are Evolutionary algorithms which can optimize neural network architecture to provide faster training speed [76]. In [62], evolutionary neural networks are proposed by combining the immune continuous ant colony algorithm and BP neural network. Structure and weights of static or recurrent neural networks can be simultaneously acquired by presented evolutionary programming [63]. In [59], a cascade learning architecture is developed to provide dynamic activation functions for neural networks. This results in faster learning speed, smoother process and simpler structure of the networks when it is used for identifying human control strategy.

c) Applications and experiment

There are numerous reports regarding the successful application and experiment of NN-based controllers in the real systems. The multi-loop nonlinear neural network tracking controller is implemented for a single flexible link [72]. Neural network controller combined with PID controller is tested on a wheeled drive mobile robot based inverted pendulum to maintain balance as well as track desired trajectories [73]. In [74], a 2-degrees-of-freedom inverted pendulum on an x-y plane is controlled by a decentralized neural network control scheme. Each axis is controlled by two separate neural network controllers since the decentralized controller can not only compensate the uncertainties but also decouple the system. In a word, numerous training processes are fast enough, which are suitable for real-time control implementation.

1.2.4 Multi-time scale system

Numerous nonlinear physical systems contain slow and fast dynamic processes that occur at different moments. The following models are used to describe such dynamic characteristic of the nonlinear systems.

a) Singularly perturbed system

We consider a system of $k_1 + k_2$ first-order autonomous ordinary differential equations for $k_1 + k_2$ dynamic variables, of which k_1 are slow variables and k_2 are fast variables. Therefore the vector of slow variables is $x_1 \in R^{k_1}$ and the vector of fast variables $x_2 \in R^{k_2}$. Then the system of equations is

$$\begin{aligned} \frac{dx_1}{dt} &= f_1(x_1, x_2) \\ \varepsilon \frac{dx_2}{dt} &= f_2(x_1, x_2) \end{aligned} \tag{1.2}$$

which is a slow-time system. $\varepsilon > 0$ is a small parameter. System (1.2) has asymptotic structure (k_1, k_2).

The transformation of time $t = \varepsilon T$ brings this system to the form of a fast-time system:

$$\begin{aligned} \frac{dx_1}{dT} &= \varepsilon f_1(x_1, x_2) \\ \frac{dx_2}{dT} &= f_1(x_1, x_2) \end{aligned} \tag{1.3}$$

Systems (1.2) and (1.3) are equivalent to each other for finite ε, but have different properties in the limit $\varepsilon \to 0^+$.

A typical example of multi-time scale system is DC servomotor as shown in Figure 1.1 [29]

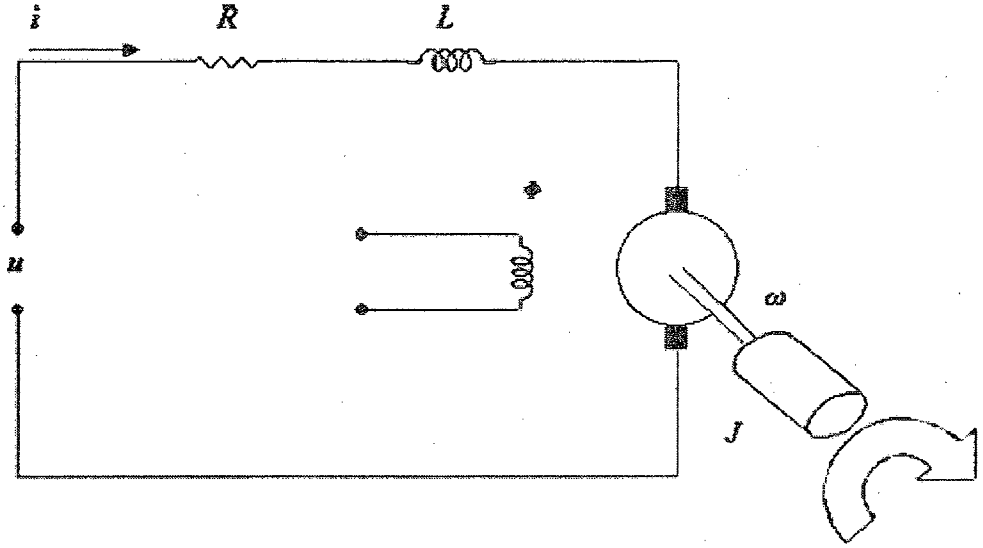


Figure 1-1 DC Servomotor

DC motor modeling can be separated into electrical and mechanical two subsystems. As we all know, the time constant of electrical system is much smaller then that of the mechanical system. Hence, the electrical subsystem is the fast subsystem.

Kirchhoff's voltage law is used to derive the electrical system:

$$L\frac{di}{dt} = -k\omega - Ri + u \tag{1.4}$$

where u is input voltage, i is armature current, R and L are the resistance and inductance of the armature, K is back EMF constant.

The mechanical subsystem follows

$$J\frac{d\omega}{dt} = ki \tag{1.5}$$

where J is the moment of inertia, k is torque constant of the motor.

Then we apply a transformation of $\omega_r = \frac{\omega}{J}, i_r = \frac{iR}{Jk}, u_r = \frac{u}{Jk}$ for (1.4) and (1.5)

$$\begin{aligned} \dot{\omega}_r &= i_r \\ \varepsilon \dot{i}_r &= -\omega_r - i_r + u_r \end{aligned} \qquad (1.6)$$

where $\varepsilon = \frac{Lk^2}{JR^2} \leq 1$, i_r is the fast state.

b) Parametric Embedding

Unlike the DC motor mentioned above, some system equations only consist of constants with certain values which are measured from experiments. To apply singular perturbation theorems to these models which do not contain any parameters that can tend to zero or infinity, we need to introduce the small parameters artificially.

We will call a system $\dot{x} = F(x;\varepsilon), x \in R^d$ depending on parameter ε , a one-parametric embedding of a system: $\dot{x} = f(x), x \in R^d$, if $f(x) \equiv F(x,1)$ for all $x \in R^d$. Similarly, we can define an n-parametric embedding, with right-hand sides in the form $f(x) \equiv F(x,1,...,1)$ and $\dot{x} = F(x;\varepsilon_1,...,\varepsilon_n), x \in R^d$. If an n-parametric embedding has a form of a fast-slow system with asymptotic structure $(k_1,...,k_n)$, we call it a $(k_1,...,k_n)$-asymptotic embedding.

We use this procedure to replace a small dimensionless constant a with an artificial small parameter εa , where $\varepsilon << 1$. The replacement εa constitutes a one-parametric embedding. There are numerous ways a system can be parametrically embedded, but only the one in which the qualitative features that we are interested in can be best preserved from the original system is the first choice.

1.2.5 Identification and control of systems with multi-time scale

The systems which contain fast and slow phenomena can be modeled by singularly perturbed model. This can decouple the high-order linear or nonlinear system into fast and slow subsystem that occurs at different moments, which simplifis the complicated dynamic process for numerical study and control design. In [77], a system identification strategy with two time scales is proposed for modeling the Tokamak process based on experiment data of static plasma response. Then two time-scales reduced-order models are used to test the optimal control scheme. For some cases, the centralized controller can't stabilize the fast and slow dynamic simultaneously. Then the singular perturbation theory is applied to separate the model into multiply time scale subsystems. Decentralized model predictive controller are developed based on the transfer function matrix of a kind of special systems which is decoupled into two models in different time scales [78]. Comprehensive discussion about singular perturbation and time scales in control can be found in [79-81].

Neural network are also applied for multi-time scale problem. In [82], a neural controller with two time scales is designed for trajectory tracking of robot manipulator. Only the fast subnet is learning when the linear parameter is changed, which can save the computation work. In [83], the flexible-link robot arm system is divided into two time scales to reduce the spillover effect. The optimal control technique is applied to fast subsystem, while the fuzzy logic controller guarantees the tracking control performance. The stability analysis of recurrent neural networks by singular perturbation method is presented in [28]. In [29], the passivity analysis of the system identification problem about multi-time scale neural network is developed for further application in control purpose.

1.3 Research objectives and main contributions of this thesis

Consider a class of nonlinear systems with different time scales. The overall objective of this thesis is to develop on-line identification and control strategies to achieve fast and accurate trajectory tracking performance for such class nonlinear system.

1.3.1 Research objectives

The main research objectives of this thesis are:

1) To develop new updating algorithms and stability analysis for dynamic neural networks with multi-time scales in the sense of minimizing the identification error for nonlinear systems with or without multi-time scales.
2) To develop NN-based adaptive controller to achieve fast and accurate trajectory tracking based on the on-line identification results.

1.3.2 Main contributions

In this research, system identification and control based on dynamic neural networks with multi-time scales are extensively studied for nonlinear black-box model. The main contributions are summarized as:

- The Lyapunov function and singularly perturbed techniques are used to develop the on-line update laws for both dynamic neural networks weights and the linear part matrices. The learning algorithm of the linear part matrices is applied to provide more flexibility and accuracy of nonlinear system identification [84].

- New stability conditions are determined for identification error by means of Lyapunov-like analysis with new dead-zone indicators which prevent the weights of neural network from drifting into infinity [86].
- Various control methods are applied for trajectory tracking based on the on-line multiple time scales neural networks identification results for the uncertain nonlinear dynamic systems [85].
- Simulations have been carried out to verify the effectiveness of these identification and control algorithms.

1.4 Thesis Outline

The thesis is organized as follows:

In Chapter 2, some mathematical preliminaries are introduced along with the mechanism and structure of dynamic neural network with multi-time scales.

In Chapter 3, the structure of dynamic neural networks with different time scales and the identification algorithm are discussed. Then the adaptive tracking control method and the error analysis are stated followed by simulation results.

In Chapter 4, the improved system identification and control schemes for multi-time scales neural network are presented.

In Chapter 5, conclusion and some possible future work are given.

1.5 Conclusion

In this chapter, first we review the system identification by classifying it into on-line and off-line identification, black-box and grey box and linear and nonlinear identification problem. After a brief discussion about the nonlinear control, we provide extensive literature review on NN-base system identification and control. The basic concept of multi-time scales system is introduced as well. The motivation, research objective and contribution are presented in the thesis.

Chapter 2 Mechanism and Structure of Neural Network

Artificial Neural Networks (ANN), commonly referred to as "neural networks", are mathematical models which are inspired from the structure and functions of the biological neural systems like human brain [1, 2]. Through the massively parallel distributed structure and the ability of learning and generalization, NN are computationally powerful enough to perform some capabilities of the biological neural networks (BNN), such as knowledge storing, information processing, learning and justification[1, 3]. As a result, the application areas of NN range from signal processing, patter recognition, data mining, classification, medicine, financial application, to system identification and control.

As mentioned in Chapter 1, the architectures of neural networks can be categorized into feedforward neural networks and recurrent neural networks (Dynamic Neural Network). In this study, dynamic neural networks are chosen candidates for modeling and control of nonlinear systems with multi-time scales which contain strong nonlinearity and uncertainty. The architecture of the recurrent neural networks and corresponding activation functions are introduced in this chapter.

2.1 Feedforward neural network

Multilayer feedforward neural network consists of an input layer, at least one hidden layer and an output layer. The general architecture of a multilayer perceptron (MLP) is shown in Figure 2-1.

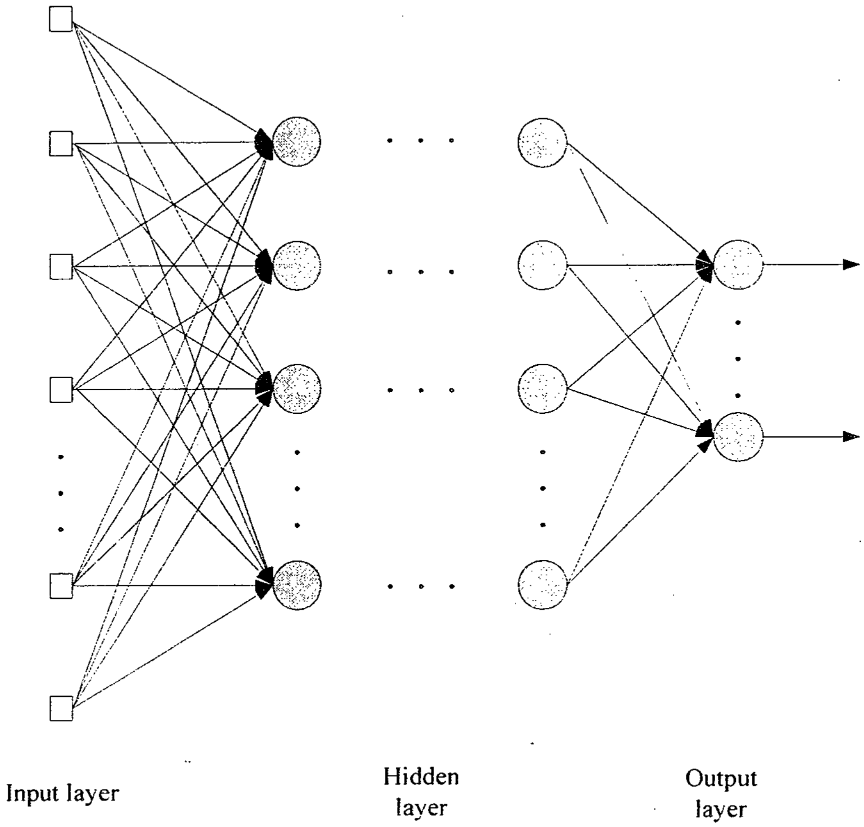


Figure 2-1 Structure of MLP

2.2 Dynamic Neural Network

A large class of dynamic systems can be represented in the form of a system of first-order differential equations written as follows:

$$\dot{x} = F(x(t)) \tag{2.2}$$

where F is a vector function, $x(t) = [x_1(t), x_2(t) \cdots x_N(t)]^T$ is the vector of the state variables, $\dot{x}$ denotes the derivative of state variables with respect to time t. The vector function F does not depend explicitly on time t, which makes the system (2.2) to be autonomous. In this paper, we only consider the systems in continuous time domain.

Recurrent neural network distinguishes itself from other neural networks like feedforward neural network in that it contains at least one feedback loop, which leads to fact that the neural network can be represented in the form of (2.2). On the other hand, the neural network, which is in form of dynamic system (2.2), has feedback loops congenitally. As a result, many researchers refer “recurrent” and “dynamic” as the same concept in neural network literature. A common recurrent neural network is show in Figure 2-2.

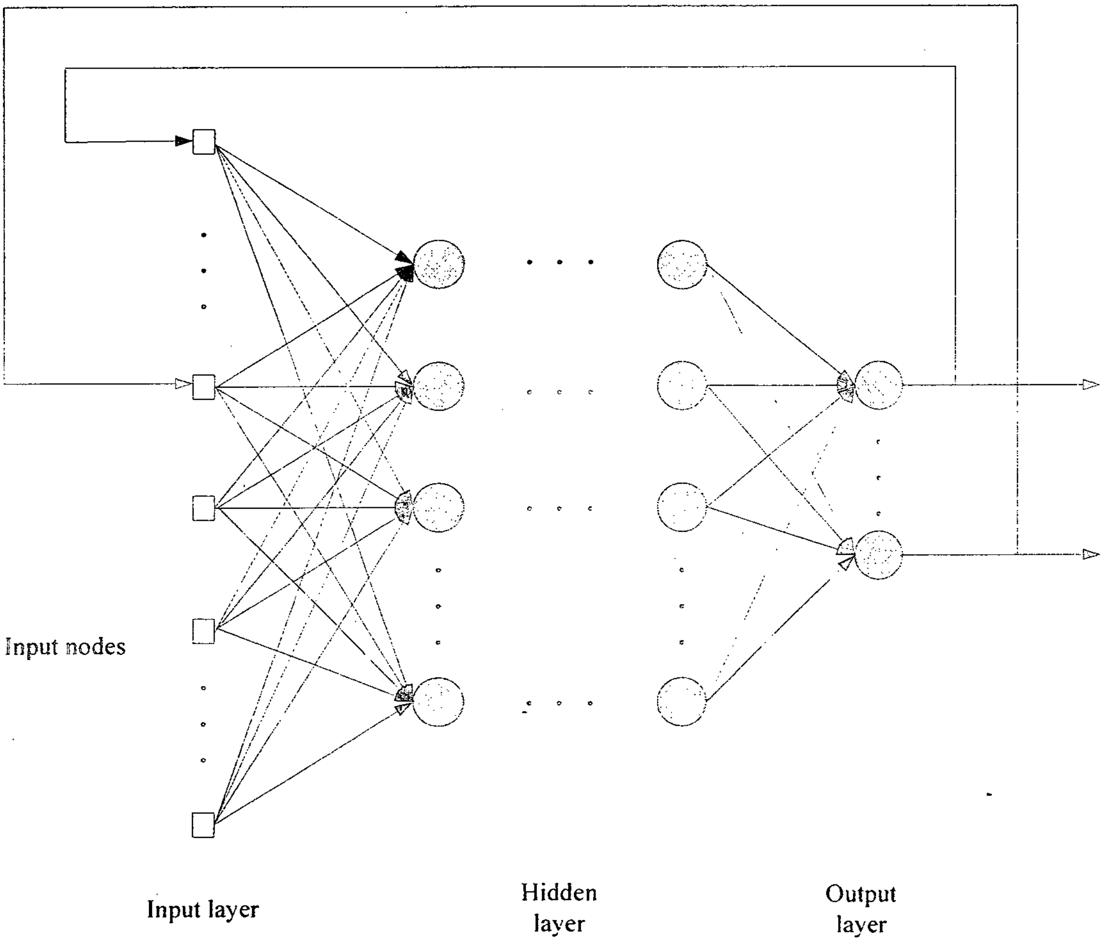


Figure 2-2 Structure of recurrent neural network

In this thesis, the architecture of the neural network is based on the following dynamic neural network.

$$\dot{x}_{nn} = Ax_{nn} + W_1\sigma(V_1 x_{nn}) + W_2\phi(V_2 x_{nn})\gamma(U) \tag{2.3}$$

where $x_{nn} \in R^n$ are state variables of neural networks, $W_1 \in R^{n \times p}, W_2 \in R^{n \times q}$ are the weight in the output layers, $V_1 \in R^{p \times n}, V_2 \in R^{q \times n}$ are the weight matrices describing hidden layers connection. $\sigma = [\sigma_1([V_1 x]_{1,1}) \cdots \sigma_p([V_1 x]_{p,1})]^T$ is vector function responsible for nonlinear state feedback. $\phi \in R^{q \times q}$ is diagonal matrix: $\phi = diag[\phi_1([V_2 x]_{1,1}) \cdots \phi_q([V_2 x]_{q,1})]^T$. $U \in R^m$ is the control input vector and $\gamma(\cdot): \Re^m \to \Re^n$ is a differentiable input-output mapping function. $A \in R^{n \times n}$ is a Hurwitz matrix for the linear part of neural networks.

As we can see that Hopfield-type neural network is the special case of neural network (2.3) with $A = diag\{a_i\}$, where $a_i = -1/R_i C_i, R_i > 0$ and $C_i > 0$. R_i and C_i are the resistance and capacitance at the ith node of the network respectively.

The three typical activation functions are shown as follows:

1. Threshold Function. The most common example is illustrated in Fig 2-3.

$$\theta(x) = \begin{cases} 1 & if \quad x \geq 0 \\ 0 & if \quad x < 0 \end{cases} \tag{2.4}$$

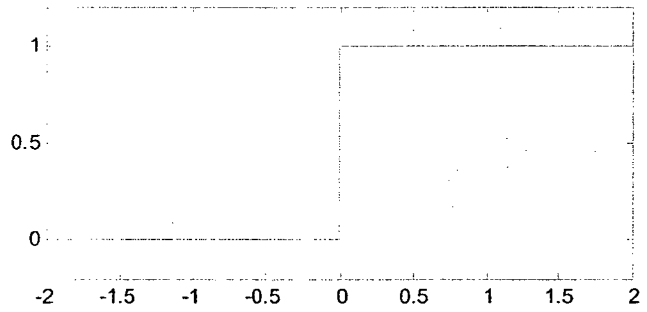


Figure 2-3 Threshold function

2. Piecewise-Linear Function (Figure 2-4)

$$\theta(x) = \begin{cases} 1 & if \quad x \geq h \\ \dfrac{1}{2h}x + 0.5 & if \quad -h < x < h \\ 0 & if \quad x \leq -h \end{cases} \tag{2.5}$$

where h is positive real number.

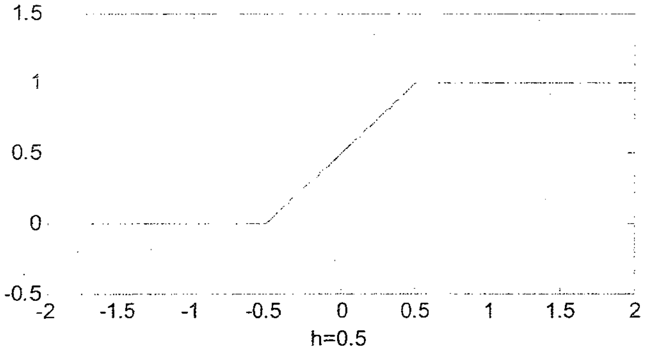


Figure 2-4 Piecewise-Linear function.

3. Sigmoid Function (Figure 2-5)

$$\theta(x) = \frac{1}{1+\exp(-ax)} \tag{2.6}$$

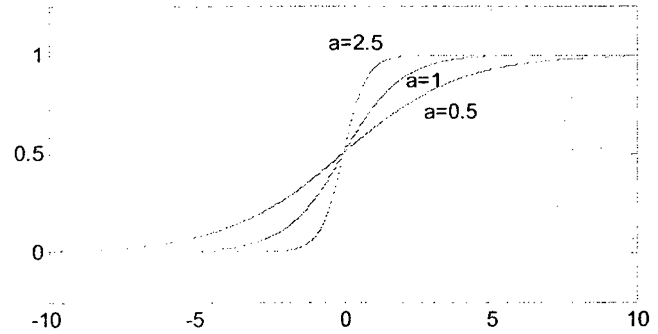


Figure 2-5 Sigmoid function.

According to many literature results [8, 9, 10, 11], sigmoid function is the most popular and suitable activation function for dynamic neural network with the structure like (2.3) due to its flexibility on the range and shape and the smoothness in the entire domain.

2.3 Multi-Time Scales Neural Networks

A wide class of nonlinear physical systems contains slow and fast dynamic processes that occur at different moments. In order to identify and control this kind of system we will utilize the Dynamic Multi-Time Scales Neural Networks (DMTSNN) as the modeling tool, which is inspired from neural network (2.3) with the perturbation parameter embedded.

$$\begin{aligned} \dot{x}_{nn} &= Ax_{nn} + W_1\sigma_1(V_1[x,y]^T) + W_2\phi_1(V_3[x,y]^T)U \\ \varepsilon\dot{y}_{nn} &= By_{nn} + W_3\sigma_2(V_2[x,y]^T) + W_4\phi_2(V_4[x,y]^T)U, \end{aligned} \qquad (2.7)$$

where $x_{nn} \in R^n, y_{nn} \in R^n$ are the slow and fast state variables of neural networks, $x \in R^n, y \in R^n$ are the state variables of the real system. $W_{1,2} \in R^{n\times 2n}$, $W_{3,4} \in R^{n\times 2n}$ are the weights in the output layers, $V_{1,2} \in R^{2n\times 2n}, V_{3,4} \in R^{2n\times 2n}$ are the weights in the hidden layer

$\sigma_k=[\sigma_k(x_1)\cdots\sigma_k(x_n),\sigma_k(y_1)\cdots\sigma_k(y_n)]^T\in R^{2n}$ $(k=1,2)$, are diagonal matrices, $\phi_k=diag[\phi_{1,2}(x_1)\cdots\phi_{1,2}(x_1),\phi_{1,2}(y_1)\cdots\phi_{1,2}(y_1)]^T\in R^{2n\times 2n}(k=1,2),U=[u_1,u_2,\cdots u_i,0\cdots 0]^T$ $\in R^{2n}$ is the control input vector, $A\in R^{n\times n}$ and $B\in R^{n\times n}$ are the unknown matrices for the linear part of neural networks, the parameter ε is a unknown small positive number. When ε is equal to 1, the neural network (2.7) becomes a normal one [12]. The typical presentations of the activation functions σ_k and ϕ_k are sigmoid functions.

In order to simplify the theory analysis, we make the hidden layer weight V to be an identity matrix, which makes DMTSNN (2.7) become a single layer neural network.

$$\begin{aligned}\dot{x}_{nn} &= Ax_{nn}+W_1\sigma_1(x,y)+W_2\phi_1(x,y)U\\ \varepsilon\dot{y}_{nn} &= By_{nn}+W_3\sigma_2(x,y)+W_4\phi_2(x,y)U\end{aligned} \tag{2.8}$$

The structure of the DMTSNN (2.8) is shown in Figure 2.6

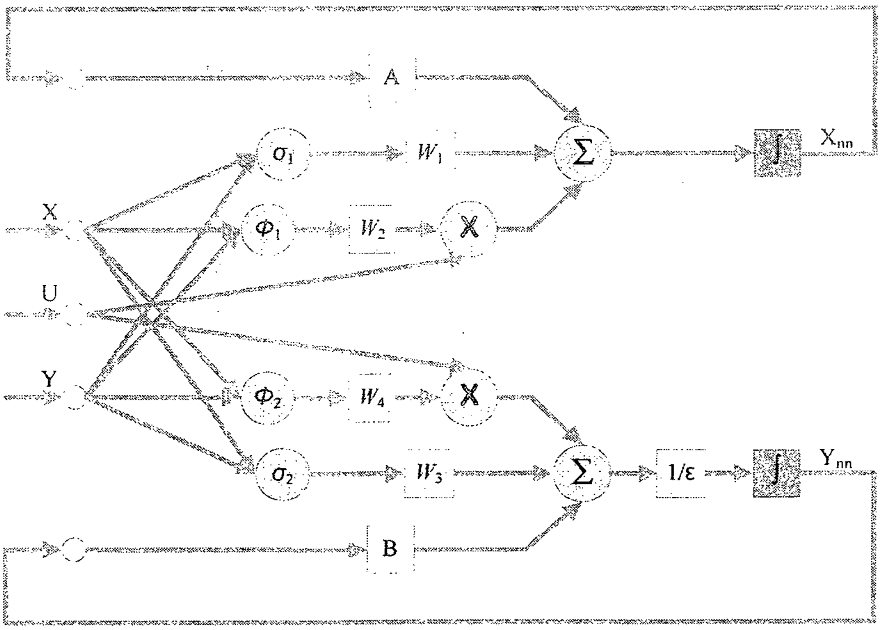


Figure 2-6 Structure of dynamic neural network with two time-scales

We apply the activation function σ_k and ϕ_k in DMTSNN (2.8) as the following types:

- Hyperbolic Tangent Function

$$\theta(x) = \tanh(x) \tag{2.9}$$

This activation function has the range from -1 to +1. The plot of Hyperbolic Tangent Function is show in Figure 2-7.

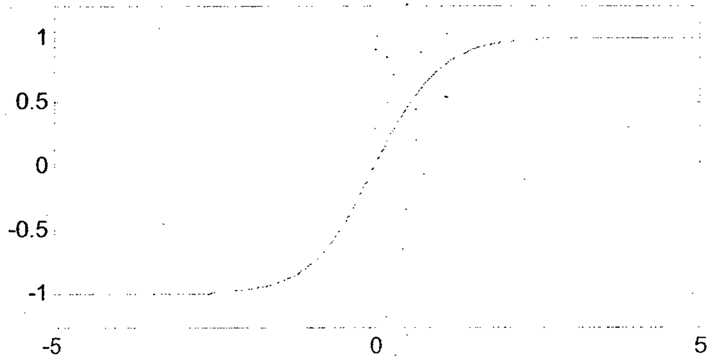


Figure 2-7 Hyperbolic Tangent Function

The architecture of neural network (2.8) is modified in Chapter 4.

$$\begin{aligned} \dot{x}_{nn} &= Ax_{nn} + W_1\sigma_1(x_{nn}, y_{nn}) + W_2\gamma(U) \\ \varepsilon\dot{y}_{nn} &= By_{nn} + W_3\sigma_2(x_{nn}, y_{nn}) + W_4\gamma(U) \end{aligned} \tag{2.10}$$

The structure of the DMTSNN (2.10) is shown in Figure 2.8

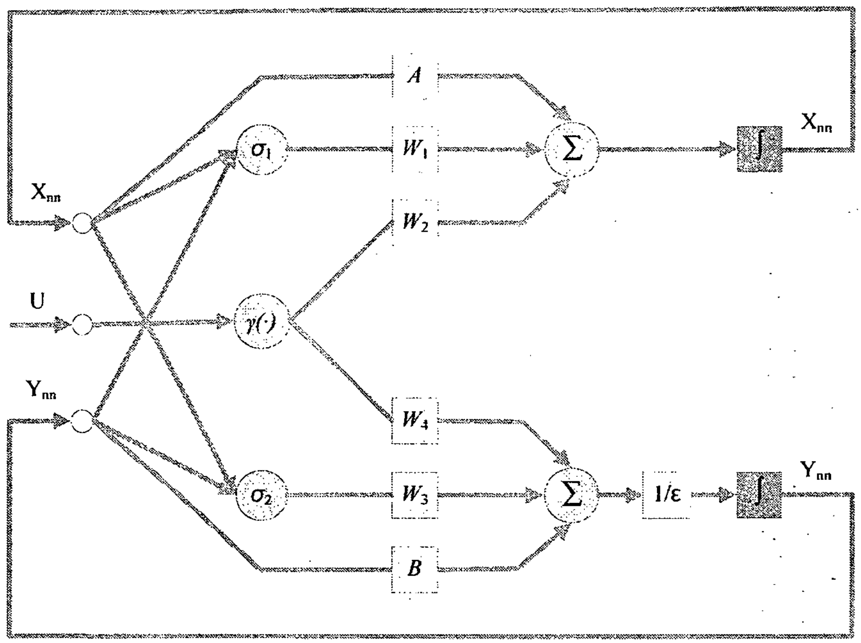


Figure 2-8 Structure of modified dynamic neural network with two time-scales

The activation functions defined in (2.4), (2.5), (2.6) and (2.9) range from 0 to +1. For the neural networks in Figure 2.8, we use the Logistic Function with multi-parameters (Figure 2-9), which can range differently and widely.

- Logistic Function with multi-parameters.

$$\theta(x) = \frac{a}{1+\exp(-bv)} - c \qquad (2.11)$$

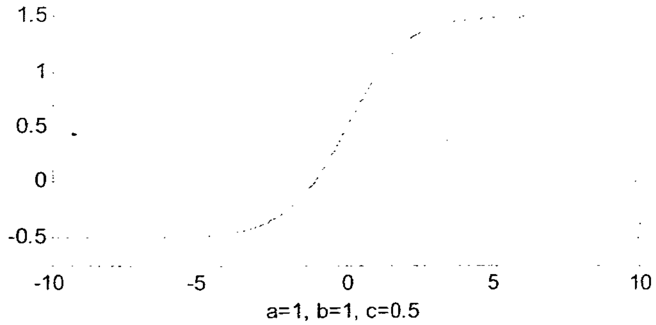


Figure 2-9 Logistic Function with multi-parameters

2.4 Conclusion

In this chapter, the mechanism and structure and the main definitions of the dynamic neural networks are introduced. An important mathematical preliminary is given for the further study. The basic mechanism and structure of DMTSNN used in system identification and control are given.

Chapter 3 Identification and Control for nonlinear systems with multi-time scales

Traditional linear control methods cannot deal with the nonlinear systems with incomplete or none information of dynamics. A common approach to deal with these problems is to utilize proper modeling and identification techniques in the control scheme. In this chapter, a dynamic neural network model is proposed for nonlinear system with multi-time scales and on-line identification algorithm is developed for its parameters so that the output of the model approaches to the output of the actual plant. By using the Lyapunov method and singularly perturbed techniques, an adaptive controller is designed based on the neural network model to control the states of nonlinear system to track reference trajectories.

3.1 On-line Identification

A large number of strategies have been proposed for the identification of dynamic systems with highly nonlinearities and uncertainties. Dynamic neural networks have been applied in system identification for those systems for many years. Due to the fast adaptation and superb learning capability, they have transcendent advantages compared to the traditional methods [14], [15].

3.1.1 Nonlinear systems with multi-time scale

Numerous systems in the industrial fields demonstrate nonlinearities and uncertainties which can be considered as partial or total black-box. A wide class of nonlinear physical systems contains slow and fast dynamic processes that occur at different moments. In this section we consider the problem of identifying this class of singular perturbation nonlinear systems with two different time scales described by

$$\begin{aligned} \dot{x} &= f_x(x,y,U,t) \\ \varepsilon\dot{y} &= f_y(x,y,U,t), \end{aligned} \tag{3.1}$$

where $x \in R^p$ and $y \in R^q$ are slow and fast state variables which are totally measureable, the functions f_x and f_y are partially or totally unknown but continuously differentiable, $U \in R^i$ is the control input vector and $\varepsilon > 0$ is a small parameter.

3.1.2 Dynamic NN model

In order to identify the nonlinear dynamical system (3.1), we employ the dynamical neural networks (2.8) with two time-scales:

$$\begin{aligned} \dot{x}_{nn} &= Ax_{nn} + W_1\sigma_1(x,y) + W_2\phi_1(x,y)U \\ \varepsilon\dot{y}_{nn} &= By_{nn} + W_3\sigma_2(x,y) + W_4\phi_2(x,y)U \end{aligned}$$

As we mentioned in Chapter 2 the slow and fast state variables of the DMTSNN are $x_{nn} \in R^n, y_{nn} \in R^n$. $W_{1.2} \in R^{n\times 2n}$, $W_{3,4} \in R^{n\times 2n}$ are the weights in the output layers. We use the state variables of the neural network to identify the object dynamic model respectively, where $n = \max < p,q >$.

Generally speaking, when the dynamic neural network (2.8) does not match the given nonlinear system (3.1) exactly, the nonlinear system can be represented as

$$\begin{aligned} \dot{x} &= A^*x + W_1^*\sigma_1(x,y) + W_2^*\phi_1(x,y)U + \Delta f_x \\ \varepsilon\dot{y} &= B^*y + W_3^*\sigma_2(x,y) + W_4^*\phi_2(x,y)U + \Delta f_y, \end{aligned} \tag{3.2}$$

where $W_1^*, W_2^*, W_3^*, W_4^*$ are unknown nominal constant matrices, the vector functions $\Delta f_x, \Delta f_y$ can be regarded as modeling error and disturbances, and A^*, B^* are the unknown nominal constant Hurwitz matrices.

Remark 3.1: In literature of system identification and control based on neural network like (2.3) or (2.7) [8 - 11], the authors coherently make a strong assumption that the linear part matrices A and B were posed as known Hurwitz matrices and such assumption is sometimes unrealistic for the black-box nonlinear system identification. Here we apply the on-line identification process to the linear part matrices dynamic to approximate to their nominal values.

It is assumed that the states in system (3.1) are completely measurable. And the number of the state variables of the plant is equal to that of the neural networks (2.8). The identification errors are defined by

$$\begin{aligned} \Delta x &= x - x_{nn} \\ \Delta y &= y - y_{nn}. \end{aligned} \tag{3.3}$$

From (2.8) and (3.2), we can obtain the error dynamics equations

$$\begin{aligned} \Delta\dot{x} &= A^*\Delta x + \widetilde{A}x_{nn} + \widetilde{W}_1\sigma_1(x,y) + \widetilde{W}_2\phi_1(x,y)U + \Delta f_x \\ \varepsilon\Delta\dot{y} &= B^*\Delta y + \widetilde{B}y_{nn} + \widetilde{W}_3\sigma_2(x,y) + \widetilde{W}_4\phi_2(x,y)U + \Delta f_y, \end{aligned} \tag{3.4}$$

where $\widetilde{W}_1 = W_1^* - W_1, \widetilde{W}_2 = W_2^* - W_2, \widetilde{W}_3 = W_3^* - W_3, \widetilde{W}_4 = W_4^* - W_4$ and $\widetilde{A} = A^* - A, \widetilde{B} = B^* - B$.

3.1.3 Adaptive Identification Algorithm

The Lyapunov synthesis method is used to derive the stable adaptive laws. Consider the Lyapunov function candidate:

$$\begin{aligned} V_I &= V_x + V_y \\ V_x &= \Delta x^T P_x \Delta x + tr\left\{\widetilde{W}_1^T P_x \widetilde{W}_1\right\} + tr\left\{\widetilde{W}_2^T P_x \widetilde{W}_2\right\} + tr\left\{\widetilde{A}^T P_x \widetilde{A}\right\} \\ V_y &= \Delta y^T P_y \Delta y + tr\left\{\widetilde{W}_3^T P_y \widetilde{W}_3\right\} + tr\left\{\widetilde{W}_4^T P_y \widetilde{W}_4\right\} + tr\left\{\widetilde{B}^T P_y \widetilde{B}\right\}. \end{aligned} \tag{3.5}$$

Since the matrices A^*, B^* are unknown nominal constant Hurwitz matrices, there definitely exist matrices P_x, P_y which can be chosen to satisfy the following equations, where Q_x, Q_y are positive definite symmetric matrices:

$$\begin{aligned} A^{*T} P_x + P_x A^* &= -Q_x \\ B^{*T} P_y + P_y B^* &= -Q_y. \end{aligned} \tag{3.6}$$

Hence, differentiating (3.5) and using (3.4) yield

$$\begin{aligned} \dot{V}_x &= -\Delta x^T Q_x \Delta x + 2\Delta x^T P \widetilde{A} x_{nn} + 2\Delta x^T P_x \widetilde{W}_1 \sigma_1(x,y) + 2\Delta x^T P_x \widetilde{W}_2 \phi_1(x,y) U \\ &\quad + 2\Delta x^T P_x f_x + 2tr\left\{\dot{\widetilde{A}}^T P_x \widetilde{A}\right\} + 2tr\left\{\dot{\widetilde{W}}_1^T P_x \widetilde{W}_1\right\} + 2tr\left\{\dot{\widetilde{W}}_2^T P_x \widetilde{W}_2\right\}, \\ \\ \dot{V}_y &= -(1/\varepsilon)\Delta y^T Q_y \Delta y + (1/\varepsilon) 2\Delta y^T P_y \widetilde{B} y_{nn} + (1/\varepsilon) 2\Delta y^T P_y \widetilde{W}_3 \sigma_2(x,y) \\ &\quad + (1/\varepsilon) 2\Delta y^T P_y \widetilde{W}_4 \phi_2(x,y) U + (1/\varepsilon) 2\Delta y^T P_y f_y + 2tr\left\{\dot{\widetilde{B}}^T P_y \widetilde{B}\right\} \\ &\quad + 2tr\left\{\dot{\widetilde{W}}_3^T P_y \widetilde{W}_3\right\} + 2tr\left\{\dot{\widetilde{W}}_4^T P_y \widetilde{W}_4\right\}, \end{aligned} \tag{3.7}$$

Theorem 3.1: Consider the identification model (3.2) for (3.1). If the modeling error and disturbances are assumed $\Delta f_x = 0$, $\Delta f_y = 0$, the updating laws

$$\begin{aligned} \dot{A} &= \Delta x x_{nn}^T & \dot{B} &= (1/\varepsilon)\Delta y y_{nn}^T \\ \dot{W}_1 &= \Delta x \sigma_1^T(x,y) & \dot{W}_3 &= (1/\varepsilon)\Delta y \sigma_2^T(x,y) \\ \dot{W}_2 &= \Delta x u^T \phi_1^T(x,y) & \dot{W}_4 &= (1/\varepsilon)\Delta y u^T \phi_2^T(x,y), \end{aligned} \tag{3.8}$$

can guarantee the following stability properties:

1) $\Delta x, \Delta y, W_{1,2,3,4}, A, B \in L_\infty$ and $\Delta x, \Delta y \in L_2$

2) $\lim_{t\to\infty} \Delta x = 0, \lim_{t\to\infty} \Delta y = 0$ and $\lim_{t\to\infty} \dot{W}_i = 0, i = 1, \cdots 4$.

Proof: Since the neural network's weights are adjusted as (3.8) and the derivatives of the neural network weights and matrices satisfy the following $\dot{W}_{1,2,3,4} = \dot{\tilde{W}}_{1,2,3,4}$, $\dot{A} = \dot{\tilde{A}}$, $\dot{B} = \dot{\tilde{B}}$ from (3.7), $\dot{V}_x, \dot{V}_y$ become

$$\begin{aligned} \dot{V}_x &= -\Delta x^T Q_x \Delta x + 2\Delta x^T P_x \Delta f_x \\ \dot{V}_y &= -(1/\varepsilon)\Delta y^T Q_y \Delta y + (1/\varepsilon)2\Delta y^T P_y \Delta f_y \end{aligned} \qquad (3.9)$$

If $f_x = 0$, $f_y = 0$, then one obtains

$$\dot{V}_x = -\|\Delta x\|^2_{Q_x} \le 0 \text{ , } \dot{V}_y = -(1/\varepsilon)\|\Delta y\|^2_{Q_y} \le 0$$

$$\dot{V} = \dot{V}_x + \dot{V}_y \le 0$$

where V_x, V_y are positive definite functions and $\dot{V}_x, \dot{V}_y \le 0$ can be achieved by using the updating laws (3.8) when $\Delta f_x = 0$, $\Delta f_y = 0$ which implies $\Delta x, \Delta y, W_{1,2,3,4}, A, B \in L_\infty$.

Furthermore, $x_{nn} = \Delta x + x, y_{nn} = \Delta y + y$ are also bounded. From the error equations (3.4), we can draw the conclusion that $\Delta\dot{x}, \Delta\dot{y} \in L_\infty$. Since V_x, V_y are non-increasing function of the time and bounded from below, the limits of V_x, V_y ($\lim_{t\to\infty} V_{x,y} = V_{x,y}(\infty)$) exist. Therefore by integrating $\dot{V}_x$, $\dot{V}_y$ on both sides from 0 to ∞, we have

$$\begin{aligned} \int_0^\infty \|\Delta x\|^2_{Q_x} &= [V_x(0) - V_x(\infty)] < \infty \\ \int_0^\infty \|\Delta y\|^2_{Q_y} &= \varepsilon[V_y(0) - V_y(\infty)] < \infty \end{aligned} \qquad (3.10)$$

The above inequalities imply that $\Delta x, \Delta y \in L_2$. Since $\Delta x, \Delta y \in L_2 \bigcap L_\infty$ and $\dot{\Delta x}, \dot{\Delta y} \in L_\infty$, using Barbalat's Lemma[13] we have $\lim_{t\to\infty} \Delta x = 0, \lim_{t\to\infty} \Delta y = 0$. Given that the control input U and $\sigma_{1,2}(\cdot), \phi_{1,2}(\cdot)$ are bounded, it is concluded that $\lim_{t\to\infty} \dot{W}_{1,2} = 0$, $\lim_{t\to\infty} \dot{W}_{3,4} = 0$. The identification scheme is illustrated in Figure 3.1.

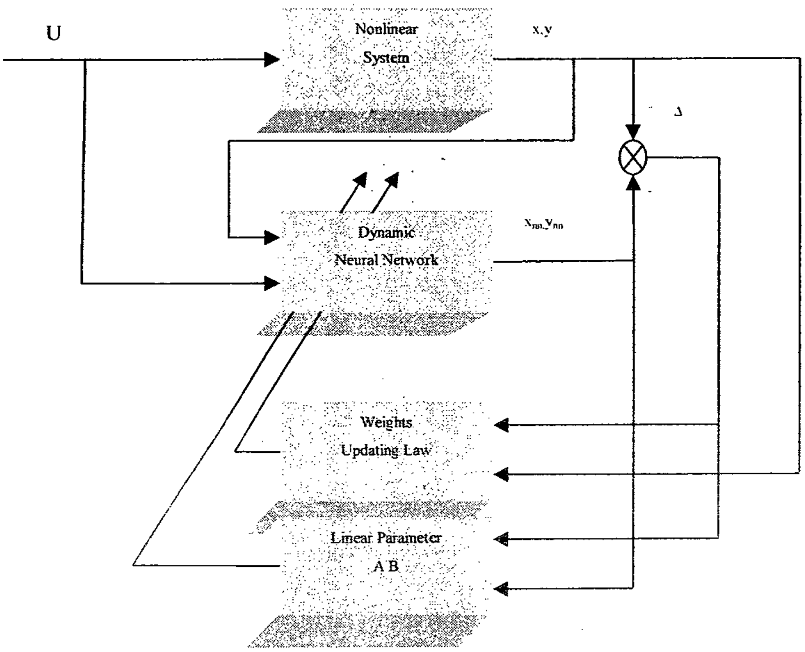


Figure 3-1 Identification scheme

Remark 3.2: When ε is very close to zero, both W_3 and W_4 exhibit a high-gain behavior, causing the instability of identification algorithm. The Lyapunov function (6) can be multiplied by any positive constant α, i.e., $B^{*T}(\alpha P_y) + (\alpha P_y)B^* = -\alpha Q_y$, the adaptation gains of W_3 and W_4 become $(1/\varepsilon)\alpha P_y$, which turn into small gains if α is chosen as a very small number.

Corollary 3.1: For the dynamics of error system equations (3.4), if we define f_x, f_y as the inputs, the update laws (3.8) can make (3.4) input-to-state stability (ISS) with the assumption that there exist positive definite matrixes Λ_x, Λ_y such that

$$\begin{aligned} &\lambda_{\min}(Q_x) \ge \lambda_{\max}(P_x \Lambda_x P_x) \\ &\lambda_{\min}(Q_y) \ge \lambda_{\max}(P_y \Lambda_y P_y). \end{aligned} \tag{3.11}$$

Proof: Using the following matrix inequality:

$$X^T Y + (X^T Y)^T \le X^T \Lambda^{-1} X + Y^T \Lambda Y \tag{3.12}$$

where $X, Y \in R^{j \times k}$ are any matrices, $\Lambda \in R^{j \times k}$ is any positive definite matrix.

We obtain

$$\begin{aligned} 2\Delta x^T P_x \Delta f_x &\le \Delta x^T P_x \Lambda_x P_x \Delta x + \Delta f_x^T \Lambda_x^{-1} \Delta f_x \\ (1/\varepsilon) 2\Delta y^T P_y \Delta f_y &\le (1/\varepsilon)\Delta y^T P_y \Lambda_y P_y \Delta y + (1/\varepsilon)\Delta f_y^T \Lambda_y^{-1} \Delta f_y \end{aligned} \tag{3.13}$$

Then equation (3.9) can be represented as

$$\begin{aligned} \dot{V}_x &= -\Delta x^T Q_x \Delta x + 2\Delta x^T P_x \Delta f_x \\ &\le -\lambda_{\min}(Q_x)\|\Delta x\|^2 + \Delta x^T P_x \Lambda_x P_x \Delta x + \Delta f_x^T \Lambda_x^{-1} \Delta f_x \\ &\le -\alpha_x(\|\Delta x\|) + \beta_x(\|\Delta f_x\|) \end{aligned} \tag{3.14}$$

$$\begin{aligned} \dot{V}_2 &= -(1/\varepsilon)\Delta y^T Q_y \Delta y + (1/\varepsilon) 2\Delta y^T P_y \Delta f_y \\ &\le -(1/\varepsilon)\lambda_{\min}(Q_y)\|\Delta y\|^2 + (1/\varepsilon)\Delta y^T P_y \Lambda_y P_y \Delta y + (1/\varepsilon)\Delta f_y^T \Lambda_y^{-1} \Delta f_y \\ &\le -\alpha_y(\|\Delta y\|) + \beta_y(\|\Delta f_y\|) \end{aligned} \tag{3.15}$$

where

$$\alpha_x(\|\Delta x\|) = (\lambda_{\min}(Q_x) - \lambda_{\max}(P_x \Lambda_x P_x))\|\Delta x\|^2, \ \beta_x(\|f_x\|) = \lambda_{\max}(\Lambda_x^{-1})\|\Delta f_x\|^2$$

$$\alpha_y(\|\Delta y\|) = (1/\varepsilon)(\lambda_{\min}(Q_y) - \lambda_{\max}(P_y \Lambda_y P_y))\|\Delta y\|^2, \ \beta_y(\|f_y\|) = (1/\varepsilon)\lambda_{\max}(\Lambda_y^{-1})\|\Delta f_y\|^2$$

We can select a positive matrix Λ_x and Λ_y such that (3.11) is established. Since α_x, β_x, α_y, β_y are K_∞ function, V_x, V_y are ISS-Lyapunov function. Using Theorem 1 in [16], the dynamics of the identification error (3.4) is input to state stability.

Theorem 3.2: If the model errors Δf_x, Δf_y, are bounded, then the updating law (3.8) can make the identification procedure stable [8]: $\Delta x, \Delta y \in L_\infty$ $W_{1,2,3,4}, A, B \in L_\infty$ $A, B \in L_\infty$.

Proof: The input to state stability means the behavior of neural network identification should remain bounded when its inputs are bounded [16].

3.1.4 Simulation Results of identification

To illustrate the theoretical results, we give the following two examples.

Example 1: Let us consider the nonlinear system

$$\begin{aligned} \dot{x}_1 &= \alpha_1 x_1 + \beta_1 sign(x_2) + u_1 \\ \varepsilon\dot{x}_2 &= \alpha_2 x_2 + \beta_2 sign(x_1) + u_2, \end{aligned} \qquad (3.16)$$

where we use the same parameter $\alpha_1 = -5$, $\alpha_2 = -10$, $\beta_1 = 3$, $\beta_2 = 2$, $x_1(0) = -5$, $x_2(0) = -5$. The given nonlinear system, even simple, is interesting enough, since it has multiple isolated equilibriums [9]. Using the parameter embedding technique [12], the model used here is singularly perturbed and the small parameter ε is positive and smaller than 1. The input signals are selected as: u_1 is a sinusoidal wave ($u_1 = 8\sin(0.05t)$) and u_2 is a saw-tooth function with the amplitude 8 and frequency 0.02Hertz.

a) We want to compare our result with that in [9]. For the fair comparison, we choose exactly the same model and input signal. Only one time scale (ε=1) is considered.

The activation function here we select hyperbolic tangent, $\sigma_{1,2}(\cdot) = \phi_{1,2}(\cdot) = \tanh(\cdot)$. This left the only difference from [9] are the neural network itself. Under the on-line adaptive updating algorithm (3.8), the identification process is conducted. The results are shown in the following Figures (3.2-3.8)

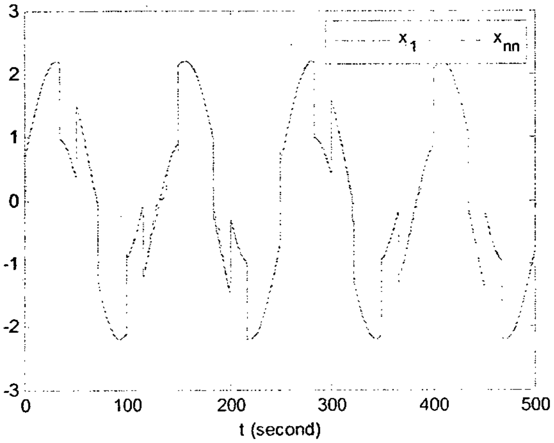


Figure 3-2 Identification result for x₁

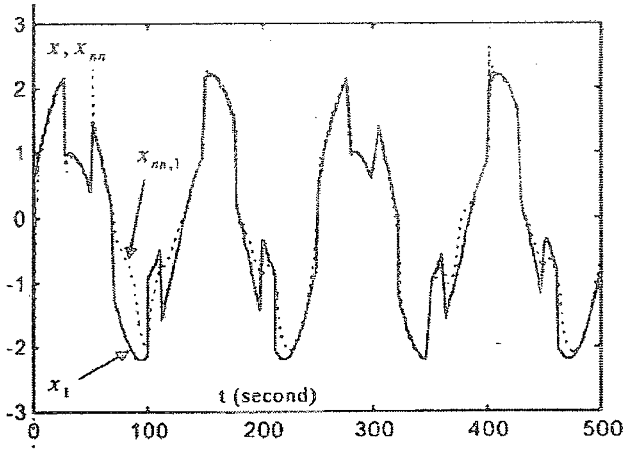


Figure 3-3 Identification result for x₁ in [15]

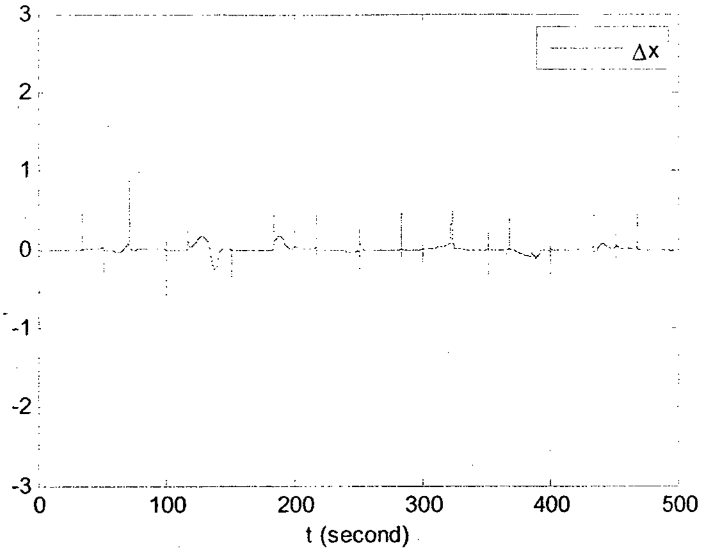


Figure 3-4 Identification error for x_1

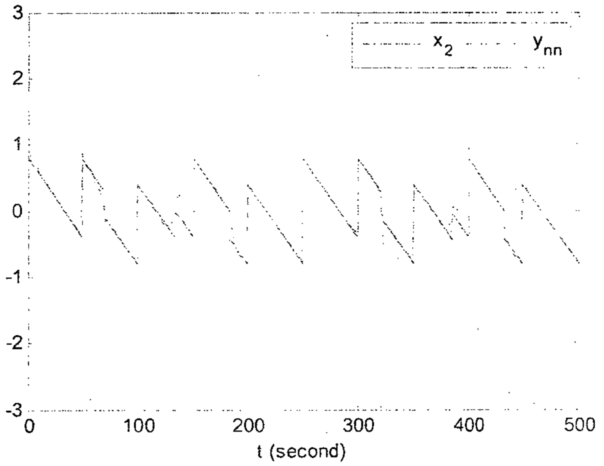


Figure 3-5 Identification result for x_2

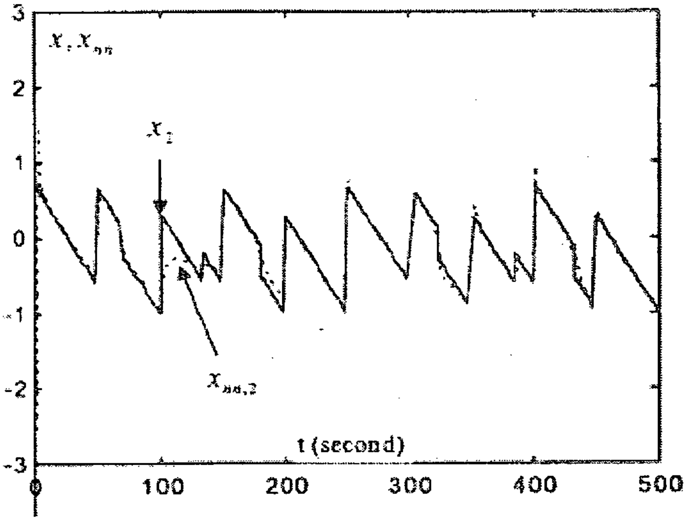


Figure 3-6 Identification result for x_2 in [15]

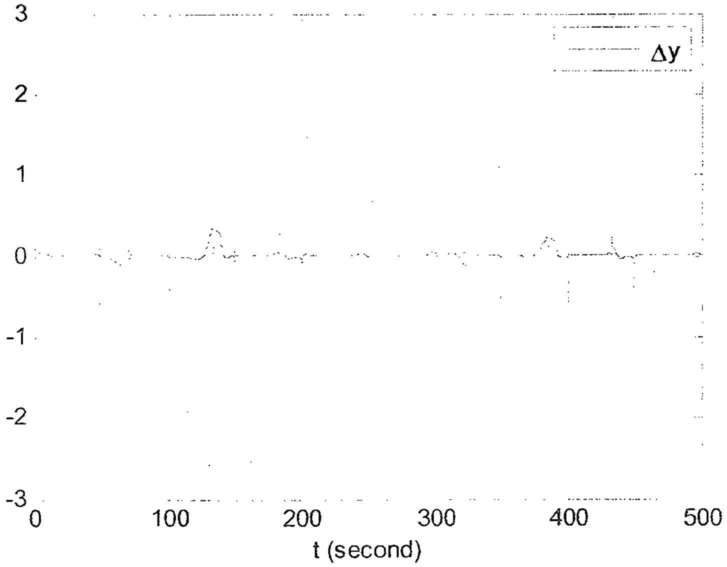


Figure 3-7 Identification error for x_2

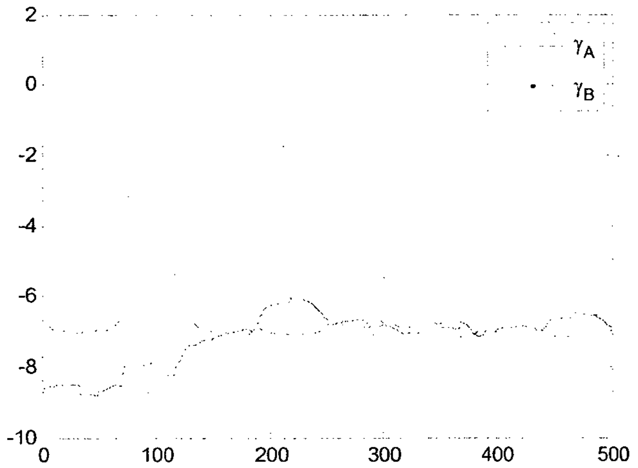


Figure 3-8 The eigenvalues of the linear parameter matrices

To show the identification performance of the proposed algorithm, the performance index --Root Mean Square (RMS) for the states error has been adopted for the purpose of comparison.

$$RMS = \sqrt{\sum_{i=1}^{n} e^2(i) \Big/ n}$$

where n is number of the simulation steps, $e(i)$ is the difference between the state variables in model and system at i^{th} step. For state variable x_1, the RMS value is 0.232782 and RMS for state variable x_2 is 0.149096.

The results in Figures 3.2-3.8 demonstrate that the identification performance has been improved compared to those in [15]. It can be seen that the state variables of dynamic multi-time scale NN follow those of the nonlinear system more accurately and quickly. The eigenvalues of the linear parameter matrix are shown in Figure 3.8. The eigenvalues for both A and B are universally smaller than zero, which means they are always stable matrices.

b) Now we consider model (3.16) with multi-time scales. The small parameter ε is selected as 0.2. The sigmoid functions are chosen as

$$\frac{a}{1+\exp(-bx)}-c \qquad (3.17)$$

The parameters for each sigmoid function in dynamic neural networks are listed in Table 3-1.

Table 3-1 Sigmoid function parameters

	a	b	c
$\sigma_1(x,y)$	2	2	0.5
$\phi_1(x,y)$	0.2	0.2	0.5
$\sigma_2(x,y)$	2	2	0.5
$\phi_2(x,y)$	0.2	0.2	-0.5

The results are shown in the following Figures (3.9-3.14).

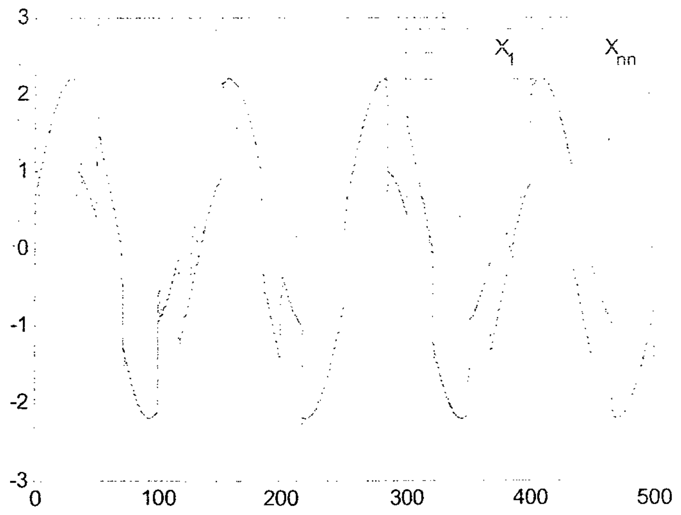


Figure 3-9 Identification result for x_1

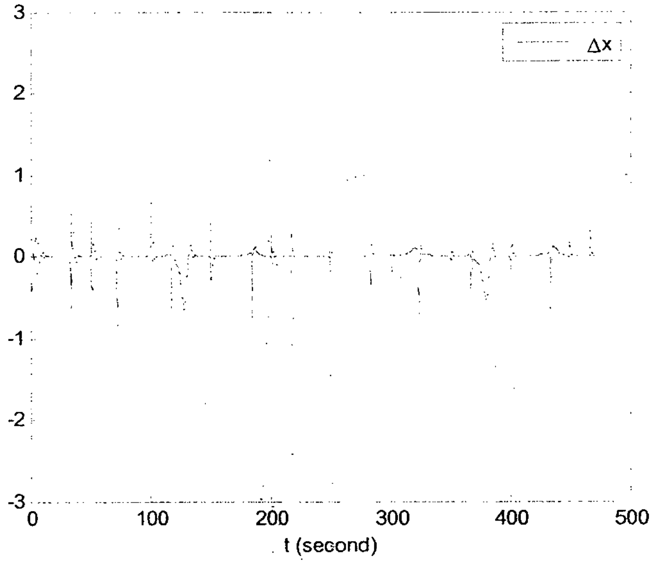


Figure 3-10 Identification error for x_1

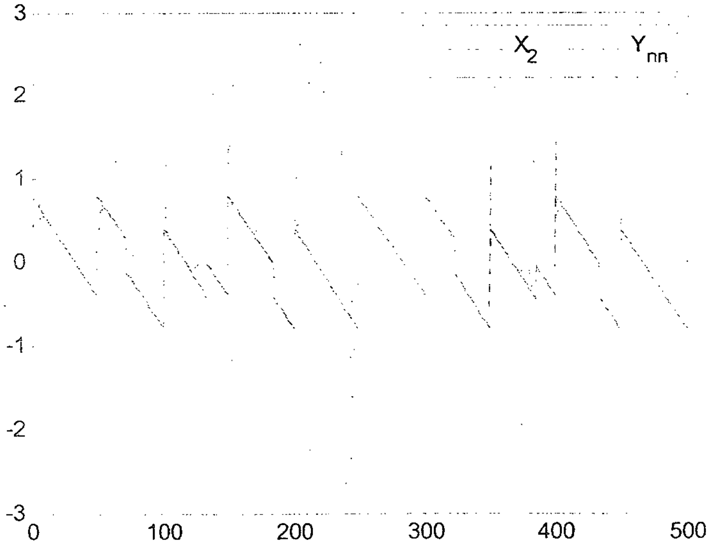


Figure 3-11 Identification result for x_2

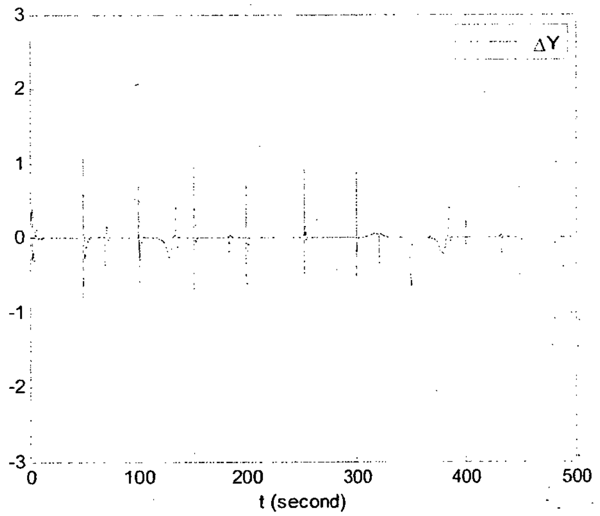


Figure 3-12 Identification error for x_2

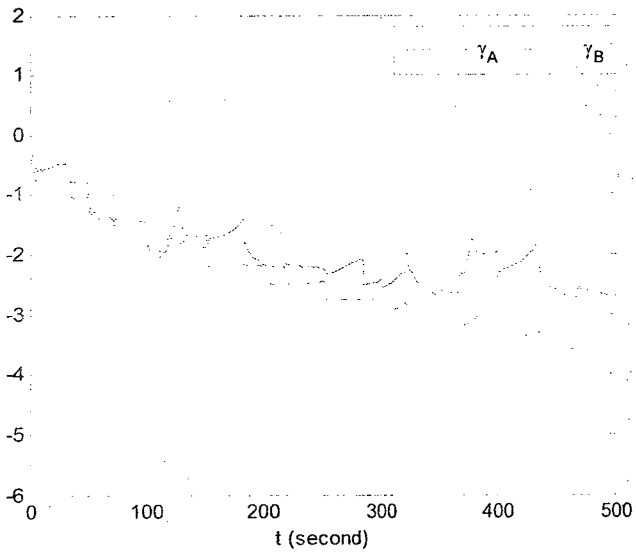


Figure 3-13 The eigenvalues of the linear part matrices

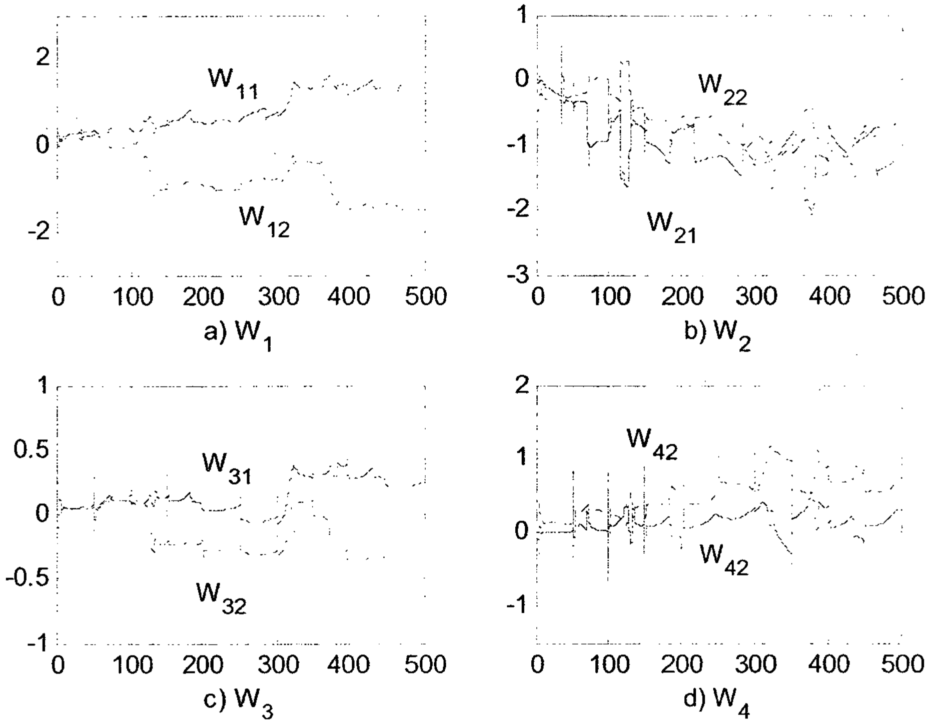


Figure 3-14 The learning process of the updating weight matrices

For state variable x_1, the RMS value is 0.139102 and RMS for state variable x_2 is 0.116635. The results in Figures 3.9-3.14 demonstrate that the state variables of dynamic multi-time scale NN follow those of the nonlinear system accurately and quickly. The eigenvalues of the linear parameter matrices are shown in Figure 3.13 The eigenvalues for both A and B are universally smaller than zero, which means they are kept as stable during the identification. Figure 3.14 shows the learning process of the updating weight matrices of the dynamic NNs.

Example 2: In 1952, Hodgkin & Huxley proposed a system of differential equations describing the flow of electric current through a surface membrane of a giant nerve fibre. Later this Hodgkin-Huxley (HH) model of the squid giant axon became one of the most important models in computational neuroscience and a prototype of a large family of mathematical models quantitatively describing electrophysiology of various living cells and tissues[12][17].

$$\begin{cases} \dfrac{dV}{dt} = \dfrac{1}{C_M}(Iext - \bar{g}_K n^4 (V + E_w - E_K) - \bar{g}_{Na} m^3 h (V + E_w - E_{Na}) \\ \qquad - \bar{g}_l (V + E_w - E_l)) \\ \dfrac{dn}{dt} = \dfrac{n_\infty - n}{\tau_n} \\ \varepsilon \dfrac{dm}{dt} = \dfrac{m_\infty - m}{\tau_m} \\ \varepsilon \dfrac{dh}{dt} = \dfrac{h_\infty - h}{\tau_h} \end{cases} \tag{3.18}$$

where time t is measured in ms, variable V is the membrane potential in mV, and n, m and h are dimensionless gating variables corresponding to K+, Na+ and leakage current channels respectively, which can vary between [0,1].

$$n_\infty = \frac{\alpha_n}{\alpha_n + \beta_n} \qquad m_\infty = \frac{\alpha_m}{\alpha_m + \beta_m} \qquad h_\infty = \frac{\alpha_h}{\alpha_h + \beta_h}$$

$$\tau_n = \frac{1}{\alpha_n + \beta_n} \qquad \tau_m = \frac{1}{\alpha_m + \beta_m} \qquad \tau_h = \frac{1}{\alpha_h + \beta_h}$$

$$\alpha_n = \frac{0.01(10 - V)}{e^{\frac{10-V}{10}} - 1} \qquad \alpha_m = \frac{0.1(25 - V)}{e^{\frac{25-V}{10}} - 1} \qquad \alpha_h = 0.07 e^{-\frac{V}{20}} .$$

$$\beta_n = 0.125 e^{-\frac{V}{80}} \qquad \beta_m = 4 e^{-\frac{V}{18}} \qquad \beta_h = \frac{1}{e^{\frac{30-V}{10}} + 1}$$

$$\bar{g}_K = 36 mS/cm^2 \qquad \bar{g}_{Na} = 120 mS/cm^2 \qquad \bar{g}_l = 0.3 mS/cm^2$$

$$E_K = -12 mv \quad E_{Na} = 115 mv \quad E_l = 10.599 mv \quad C_M = 1 \mu F/cm^2$$

From the electrophysiology point of view, the most important state of the HH system is the membrane potential V which has multifarious electro-physic phenomena and is also the core of the numerous former researches.

Instead of using the original HH model, we use the (2, 2) asymptotic embedded system [12]. We take the modified HH model with the effect of extremely low frequency (ELF) external electric field E_w which serves as the other control input besides the external applied stimulation current I_{ext}.

Since numerous researches have been carried out on applying various stimulations to HH model, whether the states of NN can still follow those of the HH system with these different stimulations becomes our first priority. So some classic inputs are applied to the system.

$$\begin{aligned} Iext &= \tfrac{1}{2} A_I (\cos\omega_I t + 1) \\ E_w &= \tfrac{1}{2} A_E \cos\omega_E t \end{aligned} \tag{3.19}$$

where $\omega_{I,E} = 2\pi f_{I,E}$, and all the initial conditions for the HH system are the equilibrium (quiescent). $V_o = 0.00002$, $m_o = 0.05293$, $h_o = 0.59612$, $n_o = 0.31768$.

We pick two typical stimulations which can result in significant and classic neuron excitation:

a) $E_w = 0$, $A_I = 30\,\mu A/cm^2$, $f_I = 10Hz$, $\varepsilon = 0.2$.

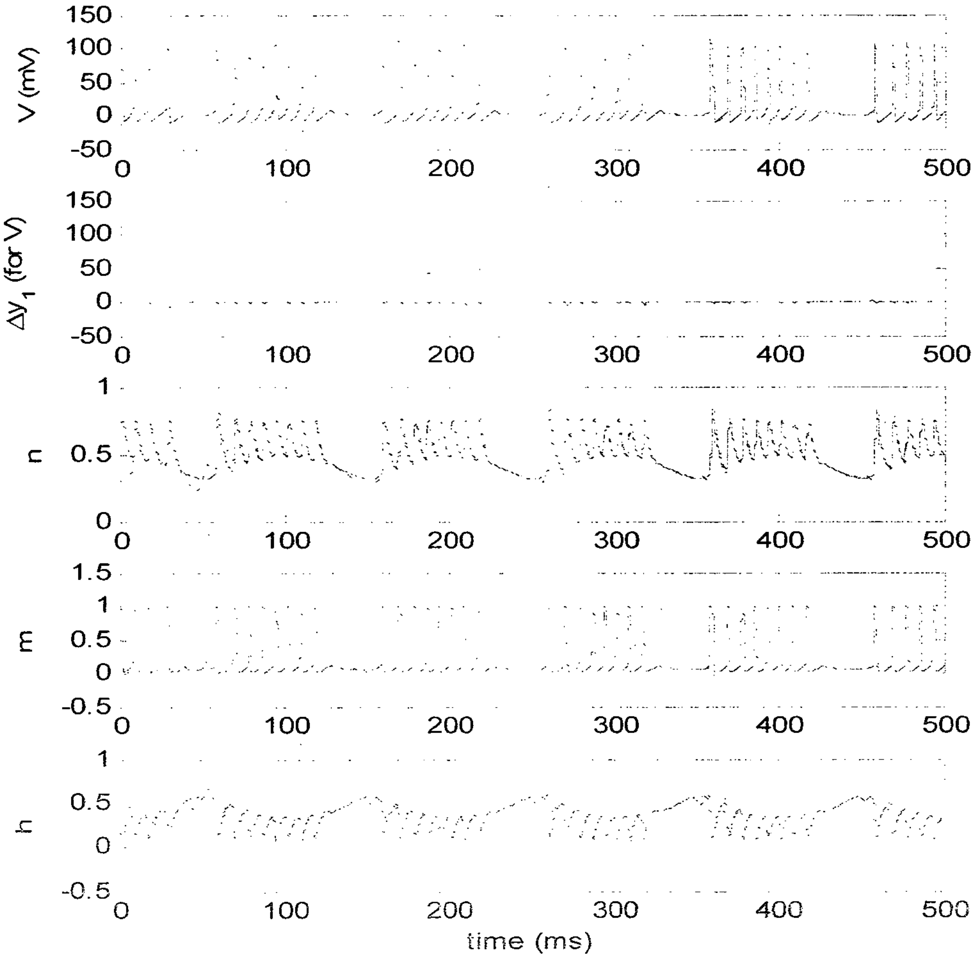


Figure 3-15 Identification results

In the plot for state V, n, m, h, the real lines are the real state variables for the HH system and the dot lines represent the identification state in the NN. The second plot is the identification error for the membrane potential.

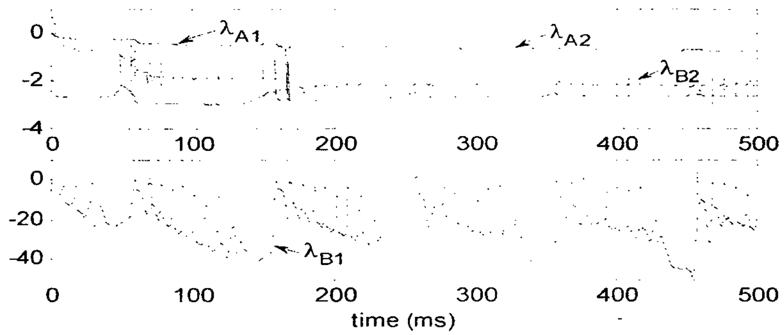


Figure 3-16 Eigenvalues of the linear matrices A, B

b) $I_{ext} = 0$, $A_E = 10mV$, $f_E = 115Hz$, $\varepsilon = 0.2$.

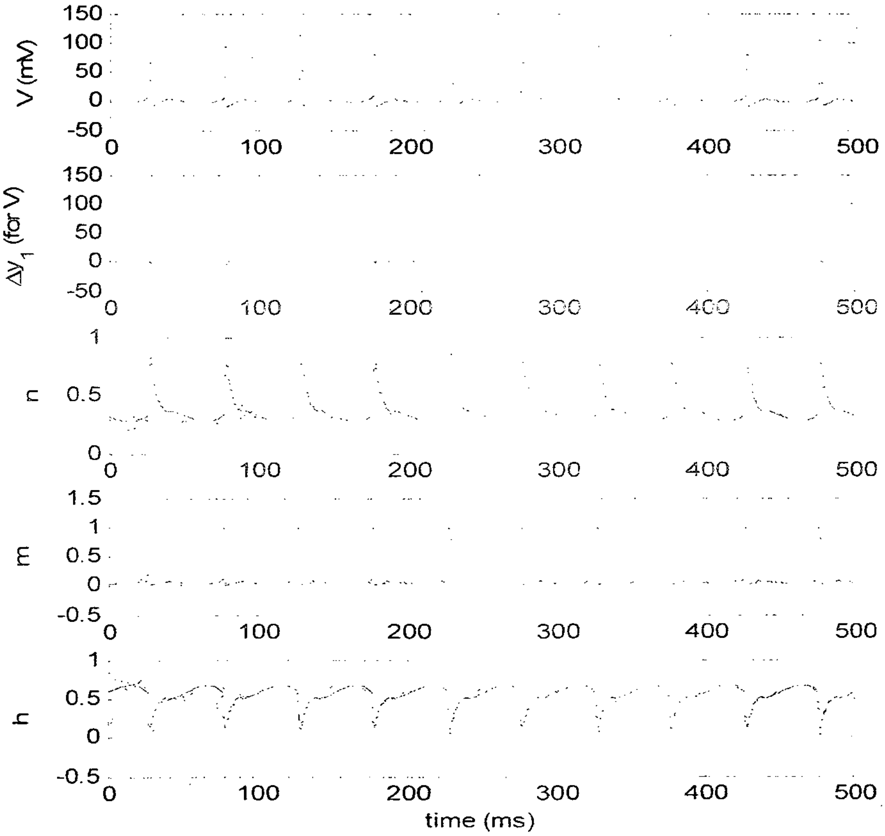


Figure 3-17 Identification results

In the plot for state V, n, m, h, the real lines are the real state variables for the HH system and the dot lines represent the identification state in the NN. The second plot is the identification error for the membrane potential.

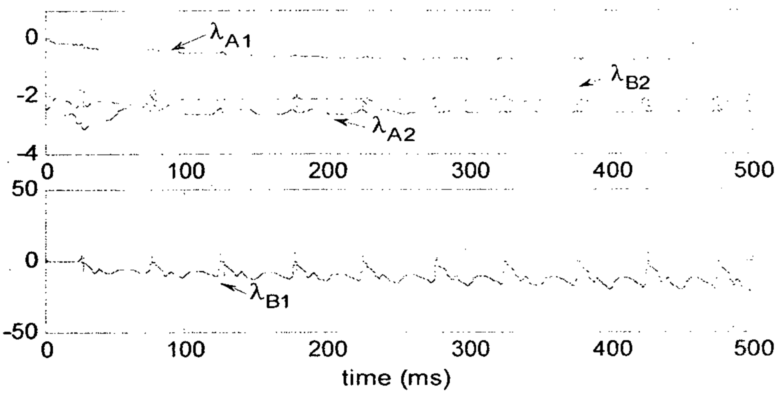


Figure 3-18 Eigenvalues of the linear matrices A, B

In simulation a), System is in 8/1 phase locked oscillation periodic bursting. RMS value of the state variables are RMS_n=0.074642, RMS_h=0.083497, RMS_V=0.438275, RMS_m=0.035473. In b), System is in same frequency periodic spiking. RMS value of the state variables are RMS_n=0.05695, RMS_h=0.061458, RMS_V=0.86327, RMS_m=0.060288. The time scale is considered by putting $\varepsilon = 0.2$. From Figures 3.15-3.18, we can see that the states of NN model can follow those of HH model very closely. The identification performance of the proposed algorithm is very good, especially for the membrane potential. The eigenvalues of A and B for a) and b) converge to the same steady values since the nominal linear matrices A^* and B^* do not change with different inputs.

3.2 NN-based adaptive control design

Traditional nonlinear control techniques have been developed and applied for many decades, but they are not efficient when facing the plants with incomplete information. The past decade has witnessed great activities in neural networks based control for the models with nonlinearity and uncertainty. The tracking problem is investigated based on the identification results from Section 3.1.

3.2.1 Tracking error analysis

From section 3.1 we know the nonlinear system may be modeled by dynamic neural networks with the updating laws (3.8):

$$\begin{aligned} \dot{x} &= Ax + W_1\sigma_1(x,y) + W_2\phi_1(x,y)U + \Delta f_x \\ \varepsilon\dot{y} &= By + W_3\sigma_2(x,y) + W_4\phi_2(x,y)U + \Delta f_y, \end{aligned} \tag{3.20}$$

where the model error and disturbances Δf_x , Δf_y, are still assumed to be constrained as before. And also $W_{1,2,3,4}$ are bounded as well as other stability properties in Section 3.1.

The model error in most cases could be zero or negligible, however, even if the dynamic neural networks have superb learning ability to represent the nonlinear dynamic process, the model error are sometimes inevitable or even may affect the stability of the system. The following controller design considers this model error for more general situations.

Hence, the control goal is to force the system states to track the desired signals, which are generated by a nonlinear reference model

$$\begin{aligned} \dot{x}_d &= g_x(x_d, y_d, t) \\ \dot{y}_d &= g_y(x_d, y_d, t) \end{aligned} \tag{3.21}$$

The overall structure of the neural networks identification and controller is shown in Figure 3.19.

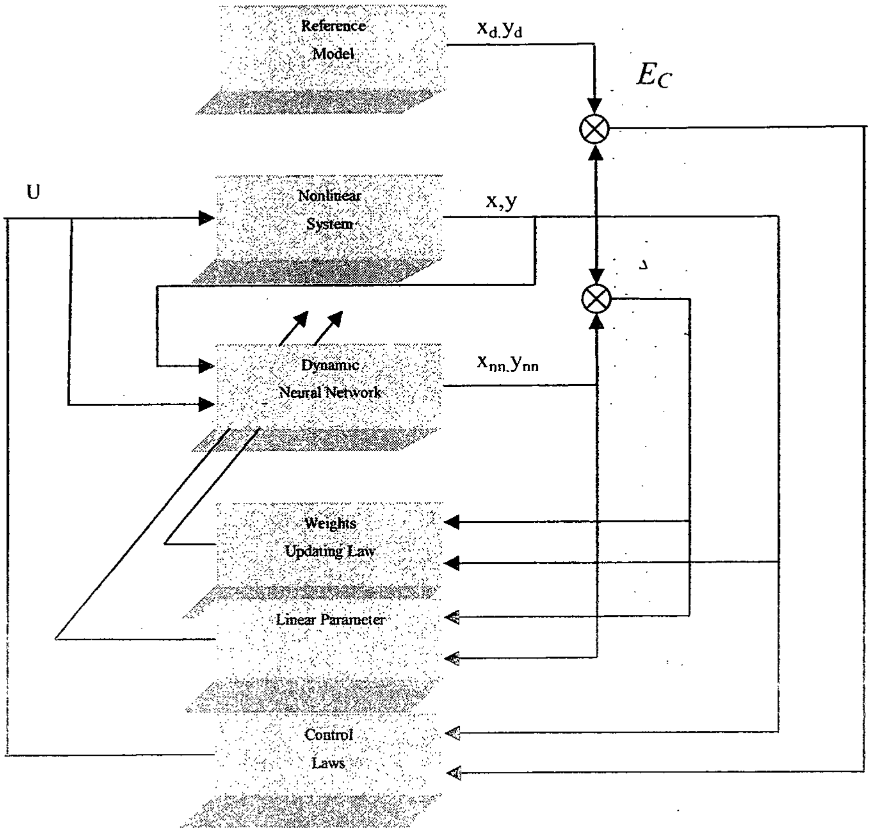


Figure 3-19 Identification and control scheme

We define the state tracking error as

$$
\begin{aligned}
E_x &= x - x_d \\
E_y &= y - \dot{y}_d .
\end{aligned}
\tag{3.22}
$$

Then the error dynamic equations become:

$$
\begin{aligned}
\dot{E}_x &= Ax + W_1\sigma_1(x,y) + W_2\phi_1(x,y)U + \Delta f_x - g_x \\
\varepsilon\dot{E}_y &= By + W_3\sigma_2(x,y) + W_4\phi_2(x,y)U + \Delta f_y - g_y .
\end{aligned}
\tag{3.23}
$$

Then the control action U is designed as

$$U = u_L + u_f, \tag{3.24}$$

where u_L is a compensation for the known nonlinearity and u_f are dedicated to deal with the model errors, which can be left open if it is zero or ignorable. Let u_L be

$$\begin{aligned} u_L &= \begin{bmatrix} W_2\phi_1(x,y) \\ (1/\varepsilon)W_4\phi_2(x,y) \end{bmatrix}^{-1} u_L' \\ u_L' &= -\begin{bmatrix} Ax_d \\ (1/\varepsilon)By_d \end{bmatrix} - \begin{bmatrix} W_1\sigma_1(x,y) \\ (1/\varepsilon)W_3\sigma_2(x,y) \end{bmatrix} + \begin{bmatrix} g_x \\ (1/\varepsilon)g_y \end{bmatrix} \end{aligned} \tag{3.25}$$

The control action u_f is to compensate the unknown dynamic modeling error. The sliding mode control methodology is applied to accomplish the task. So let u_f be,

$$u_f = \begin{bmatrix} W_2\phi_1(x,y) \\ (1/\varepsilon)W_4\phi_2(x,y) \end{bmatrix}^{-1} u_f' \tag{3.26}$$

$$u_f' = \begin{bmatrix} u_{fx}' \\ u_{fy}' \end{bmatrix} = \begin{bmatrix} -AE_x - k_x \operatorname{sgn}(E_x) \\ -(1/\varepsilon)BE_y - (1/\varepsilon)k_y \operatorname{sgn}(E_y), \end{bmatrix} \tag{3.27}$$

The modeling error and disturbances are assumed to be bounded. Hence we have

$$\|\Delta f_x\| \le \Delta\bar{f}_x, \|\Delta f_y\| \le \Delta\bar{f}_y, \tag{3.28}$$

Theorem 3.3: Consider nonlinear system (3.1) and the identification model (3.2). With the updating laws (3.8) and control strategy (3.24), we can guarantee the following stability properties:

1) $\Delta x, \Delta y, W_{1.2.3.4}, A, B \in L_\infty$ and $\Delta x, \Delta y \in L_2$

2) $\lim_{t\to\infty} \Delta x = 0, \lim_{t\to\infty} \Delta y = 0$ and $\lim_{t\to\infty} \dot{W}_i = 0, i = 1, \cdots 4.$

3) $\lim_{t\to\infty} E_x = 0, \lim_{t\to\infty} E_y = 0$

Proof : If we consider the identification and control as a whole process, then we can apply the strategy to real applications by generating the final Lyapunov function candidate as $V = V_I + V_C$.

In section 3.1, we had already proved $\dot{V}_I \le 0$ and the stability properties in Theorem 3.1. Now let's consider the Lyapunov function candidate for control purpose

$$V_c = E_x^T E_x + E_y^T E_y. \tag{3.29}$$

First rewrites (3.23) as

$$\begin{bmatrix} \dot{E}_x \\ \dot{E}_y \end{bmatrix} = \begin{bmatrix} Ax \\ (1/\varepsilon)By \end{bmatrix} + \begin{bmatrix} W_1\sigma_1(x,y) \\ (1/\varepsilon)W_3\sigma_2(x,y) \end{bmatrix} + \begin{bmatrix} W_2\phi_1(x,y) \\ (1/\varepsilon)W_4\phi_2(x,y) \end{bmatrix} U + \begin{bmatrix} \Delta f_x \\ (1/\varepsilon)\Delta f_y \end{bmatrix} - \begin{bmatrix} g_x \\ (1/\varepsilon)g_y \end{bmatrix} \tag{3.30}$$

Then substituting (3.25) into (3.30) obtains

$$\begin{bmatrix} \dot{E}_x \\ \dot{E}_y \end{bmatrix} = \begin{bmatrix} AE_x \\ (1/\varepsilon)BE_y \end{bmatrix} + \begin{bmatrix} W_2\phi_1(x,y) \\ (1/\varepsilon)W_4\phi_2(x,y) \end{bmatrix} u_f + \begin{bmatrix} \Delta f_x \\ (1/\varepsilon)\Delta f_y \end{bmatrix} \tag{3.31}$$

If the model error and disturbances are zero or negligible which, from the control point of view, means it won't devastate the stability of the system, u_f can be chosen to be zero which will lead the error dynamics converge to the origin. Proof is quite straightforward since A and B are stable matrices and ε is positive.

Then substituting (3.26) into (3.31) yields

$$\begin{aligned} \dot{E}_x &= AE_x + u'_{fx} + \Delta f_x \\ \dot{E}_y &= (1/\varepsilon)BE_y + u'_{fy} + (1/\varepsilon)\Delta f_y. \end{aligned} \tag{3.32}$$

By using (3.32) and (3.27), we obtain the derivative of (3.29) as

$$\begin{aligned}\dot{V}_c &= 2E_x^T\dot{E}_x + 2E_y^T\dot{E}_y \\ &= 2E_x^T(AE_x + u'_{fx} + f_x) + 2E_y^T((1/\varepsilon)BE_y + u'_{fy} + (1/\varepsilon)\Delta f_y) \\ &= -2k_x\|E_x\| + 2E_x^T f_x - 2(1/\varepsilon)k_y\|E_y\| + 2(1/\varepsilon)\Delta cy^T\Delta f_y \\ &\leq -2k_x\|E_x\| + 2\|E_x\|\|\Delta f_x\| - 2(1/\varepsilon)k_y\|E_y\| + 2(1/\varepsilon)\|E_y\|\|\Delta f_y\| \\ &= -2(k_x - \|\Delta f_x\|)\|E_x\| - 2(1/\varepsilon)(k_y - \|\Delta f_y\|)\|E_y\|\end{aligned}$$

If we choose $k_x > \Delta\bar{f}_x, k_y > \Delta\bar{f}_y$, then $\dot{V}_c < 0$. Hence, we have stability properties 3) $\lim_{t\to\infty} E_x = 0, \lim_{t\to\infty} E_y = 0$, and $\dot{V} = \dot{V}_I + \dot{V}_C \leq 0$.

With consideration of the modeling error and disturbances, the sliding mode control logic (3.27) can guarantee the tracking stability without the assumption that A and B are stable matrices.

The controller involves matrix inversion which can guarantee the non-singularities by choosing the proper initial values of the parameters in the updating law and the activation function.

3.2.2 Simulation results of control scheme

We continue the process in Section 3.1 for nonlinear system (3.16). Instead of using input signals sinusoidal wave and saw-tooth function, we implement the control law to obtain the control signal to the nonlinear system (3.24). It constitutes a feedback linearization and a sliding mode compensator. The desired trajectories are generated by the reference model

$$\begin{aligned}\dot{x}_d &= y_d \\ \dot{y}_d &= \sin x_d,\end{aligned} \qquad (3.33)$$

with the initial value $x_d(0) = 1$, $y_d(0) = 0$.

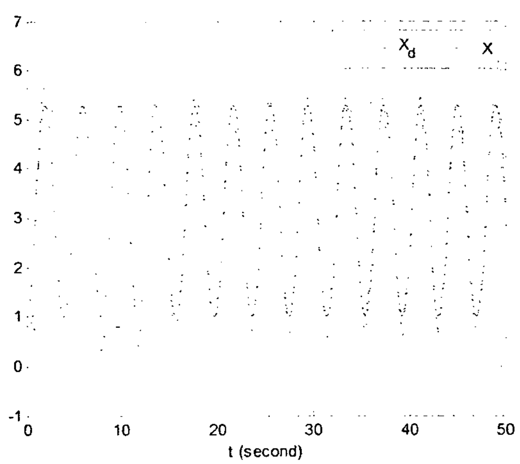


Figure 3-20 Trajectory tracking of x

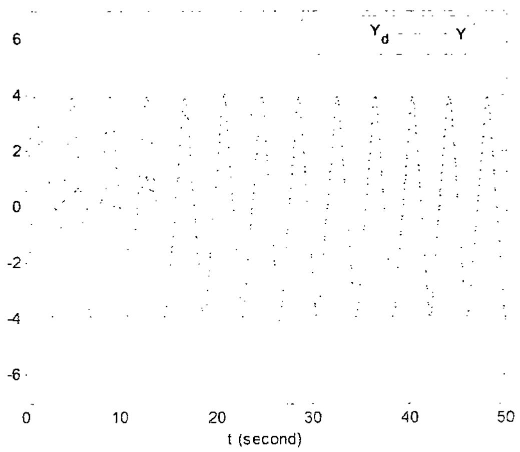


Figure 3-21 Trajectory tracking of y

The time scale is considered by putting ε = 0.2. From Figures 3.20 and 3.21, we can see that the states of the nonlinear system can track the desired trajectories in 20 seconds. For state variable x, the RMS value is 0.5246 and RMS for state variable y is 1.641411 Since the small parameter accelerate the state y, it takes relative more time for the state of the system x to track the reference signal. The simulation results demonstrate that the proposed identification and control algorithm can guarantee the tracking performance of nonlinear and uncertain dynamic systems.

3.3 Conclusion

In this chapter we propose a new on-line dynamic multi-time scale neural networks identification algorithm for both dynamic neural networks weights and the linear part matrices for nonlinear systems with multi-time scales. The proposed algorithms are applied to identify a second order nonlinear system with multiple equilibriums and the famous well-studied HH model which has complicated and multifarious system performance when different inputs applied. Both identification results show the effectiveness of the proposed identification algorithms.

Furthermore, we propose an adaptive control method based on dynamic multiple time scales neural networks. The learning algorithm of the linear part matrices is applied to provide more flexibility and accuracy of nonlinear system identification. The controller consists of a feedback linearization and a sliding mode-based compensator to deal with the unknown identification error and disturbance. Simulation results show the effectiveness of the proposed identification and control algorithms.

Chapter 4 Improved NN based Adaptive Control Design

In Chapter 3, we introduced an on-line dynamic updating law for a selected multi-time scale neural networks structure via Lyapunov method and then proposed a control strategy by combining feedback linearization and sliding mode methods. To further improve the performance and stability of the controller, we propose a new identification and control scheme with a modified neural network structure in this chapter.

4.1 Improved system identification

In chapter 3, we use the signals from the actual system in the neuron networks to identify the nonlinear system (3.1). This may simplifies the identification and control procedure, but the control law will depend on the actual signals of the nonlinear system. Also, this may risk the stability of the neural network because it is related to the output of the real system. In order to conquer this flaw and also simplify the identification scheme, we replace all the output signals from nonlinear system with the state variables of the neural networks in the construction of NN identifier and add constraint to the control signal as well.

Consider the nonlinear system (3.1). In order to identify the system, we employ the dynamical neural networks with two time-scales:

$$\begin{aligned} \dot{x}_{nn} &= Ax_{nn} + W_1\sigma_1(V_1[x_{nn}, y_{nn}]^T) + W_2\phi_1(V_3[x_{nn}, y_{nn}]^T)\gamma(U) \\ \varepsilon\dot{y}_{nn} &= By_{nn} + W_3\sigma_2(V_2[x_{nn}, y_{nn}]^T) + W_4\phi_2(V_4[x_{nn}, y_{nn}]^T)\gamma(U), \end{aligned} \tag{4.1}$$

where $x_{nn} \in \Re^n, y_{nn} \in \Re^n$ are the slow and fast state variables of neural networks. $W_{1,2} \in \Re^{n\times 2n}, W_{3,4} \in \Re^{n\times 2n}$ are the weights in the output layers, $V_{1,2} \in \Re^{2n\times 2n}, V_{3,4} \in \Re^{2n\times 2n}$

are the weights in the hidden layer and $\sigma_k=[\sigma_k(x_1)\cdots\sigma_k(x_n),\sigma_k(y_1)\cdots\sigma_k(y_n)]^T \in {}^{2n}$ $(k=1,2)$ are diagonal matrices, $\phi_k = diag[\phi_{1,2}(x_1)\cdots\phi_{1,2}(x_1),\phi_{1,2}(y_1)\cdots\phi_{1,2}(y_1)]^T \in \Re^{2n\times 2n}$ $(k=1,2)$, $U=[u_1,u_2,\cdots u_i,0,\cdots 0]^T \in \Re^{2n}$ is the control input vector, $\gamma(\cdot):\Re^m \to \Re^n$ is a differentiable input-output mapping function. $A\in\Re^{n\times n}$ and $B\in\Re^{n\times n}$ are the unknown matrices for the linear part of neural networks and the parameter ε is a unknown small positive number. The activation functions σ_k and ϕ_k are still kept as sigmoid function.

In order to simplify the analysis process, we consider the simplest structure this time which means: $p=q=n \quad V_1=V_2=I \quad \phi(\cdot)=I$

$$\begin{aligned} \dot{x}_{nn} &= Ax_{nn}+W_1\sigma_1(x_{nn},y_{nn})+W_2\gamma(U) \\ \varepsilon\dot{y}_{nn} &= By_{nn}+W_3\sigma_2(x_{nn},y_{nn})+W_4\gamma(U), \end{aligned} \tag{4.2}$$

4.1.1 Identification with precise structure of NN identifier

In this section, we deal with the situation that the dynamic neural networks can represent the plant precisely, which means that there exist nominal constant values of the weight W_1^*,W_2^*,W_3^*,W_4^* and unknown nominal constant Hurwitz matrices A^*,B^* such that the nonlinear system (3.1) can be described by following neural network model:

$$\begin{aligned} \dot{x} &= A^*x+W_1^*\sigma_1(x,y)+W_2^*\gamma(U) \\ \varepsilon\dot{y} &= B^*y+W_3^*\sigma_2(x,y)+W_4^*\gamma(U) \end{aligned} \tag{4.3}$$

Assumption 4.1: The difference of the activation function $\sigma_k(\cdot)$, which is $\tilde{\sigma}_k=\sigma_k(x,y)-\sigma_k(x_{nn},y_{nn})$, satisfies generalized Lipshitz condition

$$\begin{aligned}\tilde{\sigma}_1{}^T \Lambda_1 \tilde{\sigma}_1 &< \begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix}^T D_1 \begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix} = \Delta x^T D_1 \Delta x + \Delta y^T D_1 \Delta y \\ \tilde{\sigma}_2{}^T \Lambda_2 \tilde{\sigma}_2 &< \begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix}^T D_2 \begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix} = \Delta x^T D_2 \Delta x + \Delta y^T D_2 \Delta y\end{aligned} \qquad (4.4)$$

where $\sigma_k(x,y) = [\sigma_k(x_1)\cdots\sigma_k(x_n), \sigma_k(y_1)\cdots\sigma_k(y_n)]^T \in \Re^{2n} (k=1,2)$ $D_1 = D_1{}^T > 0$ $D_2 = D_2{}^T > 0$ are known normalizing matrices.

Assumption 4.2: The nominal values $W_1^*, W_2^*, W_3^*, W_4^*$ are bounded as

$$\begin{aligned}&W_1^* \Lambda_1^{-1} W_1^{*T} \le \overline{W}_1 && W_3^* \Lambda_3^{-1} W_3^{*T} \le \overline{W}_3 \\ &W_2^* \Lambda_2^{-1} W_2^{*T} \le \overline{W}_2 && W_4^* \Lambda_4^{-1} W_4^{*T} \le \overline{W}_4\end{aligned} \qquad (4.5)$$

where $\Lambda_1^{-1}, \Lambda_2^{-1}, \Lambda_3^{-1}, \Lambda_4^{-1}$ are any positive definite symmetric matrices, $\overline{W}_1, \overline{W}_2, \overline{W}_3, \overline{W}_4$ are prior known matrices bounds.

As we assumed in Chapter 3 that the state variables in system (3.1) are completely measurable and still the number of the state variables of the plant is equal to that of the neural networks (4.3). The identification error is defined as:

$$\begin{aligned}\Delta x &= x - x_{nn} \\ \Delta y &= y - y_{nn}.\end{aligned} \qquad (4.6)$$

From (4.2) and (4.3), we can obtain the error dynamics

$$\begin{aligned}\Delta \dot{x} &= A^* \Delta x + \tilde{A} x_{nn} + W_1^* \tilde{\sigma}_1 + \tilde{W}_1 \sigma_1(x_{nn}, y_{nn}) + \tilde{W}_2 \gamma(U) \\ \varepsilon \Delta \dot{y} &= B^* \Delta y + \tilde{B} y_{nn} + W_3^* \tilde{\sigma}_2 + \tilde{W}_3 \sigma_2(x_{nn}, y_{nn}) + \tilde{W}_4 \gamma(U)\end{aligned} \qquad (4.7)$$

where $\tilde{W}_1 = W_1^* - W_1$, $\tilde{W}_2 = W_2^* - W_2$, $\tilde{W}_3 = W_3^* - W_3$, $\tilde{W}_4 = W_4^* - W_4$ and $\tilde{A} = A^* - A$, $\tilde{B} = B^* - B$.

Lemma 4.1[9]: $A \in \Re^{n\times n}$ is a Hurwitz matrices, $R, Q \in \Re^{n\times n}$, $R = R^T > 0$, $Q = Q^T > 0$ if $(A, R^{1/2})$ is controllable, $(A, Q^{1/2})$ is observable, and

$$A^T R^{-1} A - Q \geq \frac{1}{4}\left(A^T R^{-1} - R^{-1} A\right) R \left(A^T R^{-1} - R^{-1} A\right)^T$$

is satisfied, the algebraic Riccati equation $A^T X + XA + XRX + Q = 0$ has a unique positive definite solution $X = X^T > 0$

Assumption 4.3: The matrices A^*, B^* are unknown nominal constant Hurwitz matrices. Define

$$\begin{aligned} R_x &= \overline{W}_1 \qquad & Q_x &= D_1 + (1/\varepsilon) D_3 + Q_{xo} \\ R_y &= \overline{W}_3 \qquad & Q_y &= D_3 + \varepsilon D_1 + Q_{yo} \end{aligned}$$

If one can select proper Q_{xo} , Q_{yo} satisfying the conditions in Lemma 1, there exist matrices P_x, P_y satisfying the following equations:

$$\begin{aligned} A^{*T} P_x + P_x A^* + P_x R_x P_x + Q_x = 0 \\ B^{*T} P_y + P_y B^* + P_y R_y P_y + Q_y = 0 \end{aligned} \tag{4.8}$$

Theorem 4.1: Consider the nonlinear system (3.1) and identification model (4.2) the updating laws for the parameters in the model

$$\begin{aligned} \dot{A} &= k_A \Delta x x_{nn}^T & \dot{B} &= (1/\varepsilon) k_B \Delta y y_{nn}^T \\ \dot{W}_1 &= k_1 \Delta x \sigma_1^T (x_{nn}, y_{nn}) & \dot{W}_3 &= (1/\varepsilon) k_3 \Delta y \sigma_2^T (x_{nn}, y_{nn}) \\ \dot{W}_2 &= k_2 \Delta x u^T \phi_1^T (x_{nn}, y_{nn}) & \dot{W}_4 &= (1/\varepsilon) k_4 \Delta y u^T \phi_2^T (x_{nn}, y_{nn}), \end{aligned} \tag{4.9}$$

where k_A, k_B, k_1, k_2, k_3, k_4 are positive constants

can guarantee the following stability properties:

1) $\Delta x, \Delta y, W_{1.2.3,4}, A, B \in L_\infty$ and $\Delta x, \Delta y \in L_2$

2) $\lim_{t \to \infty} \Delta x = 0, \lim_{t \to \infty} \Delta y = 0$ and $\lim_{t \to \infty} \dot{W}_i = 0, i = 1, \cdots 4.$

Proof: The Lyapunov synthesis method is used to derive the stable adaptive laws. Consider the Lyapunov function candidate:

$$
\begin{aligned}
V_l &= V_x + V_y \\
V_x &= \Delta x^T P_x \Delta x + \frac{1}{k_1} tr\left\{\widetilde{W}_1^T P_x \widetilde{W}_1\right\} + \frac{1}{k_2} tr\left\{\widetilde{W}_2^T P_x \widetilde{W}_2\right\} + \frac{1}{k_A} tr\left\{\widetilde{A}^T P_x \widetilde{A}\right\} \qquad (4.10) \\
V_y &= \Delta y^T P_y \Delta y + \frac{1}{k_3} tr\left\{\widetilde{W}_3^T P_y \widetilde{W}_3\right\} + \frac{1}{k_4} tr\left\{\widetilde{W}_4^T P_y \widetilde{W}_4\right\} + \frac{1}{k_B} tr\left\{\widetilde{B}^T P_y \widetilde{B}\right\}.
\end{aligned}
$$

Hence, differentiating (4.10) and using (4.7) yield

$$\begin{aligned}
\dot{V}_x &= (\Delta\dot{x}^T P_x \Delta x + \Delta x^T P_x \Delta\dot{x}) + \frac{2}{k_A} tr\left\{\dot{\tilde{A}}^T P_x \tilde{A}\right\} + \frac{2}{k_1} tr\left\{\dot{\tilde{W}}_1^T P_x \tilde{W}_1\right\} + \frac{2}{k_2} tr\left\{\dot{\tilde{W}}_2^T P_x \tilde{W}_2\right\} \\
&= \left[\Delta x^T A^{*T} + x_{nn}{}^T \tilde{A}^T + \tilde{\sigma}_1^T W_1^{*T} + \sigma_1^T(x_{nn}, y_{nn})\tilde{W}_1^T + \gamma(U)^T \tilde{W}_2^T\right] P_x \Delta x \\
&\quad + \Delta x^T P_x \left[A^* \Delta x + \tilde{A} x_{nn} + W_1^* \tilde{\sigma}_1 + \tilde{W}_1 \sigma_1(x_{nn}, y_{nn}) + \tilde{W}_2 \gamma(U)\right] \\
&\quad + \frac{2}{k_A} tr\left\{\dot{\tilde{A}}^T P_x \tilde{A}\right\} + \frac{2}{k_1} tr\left\{\dot{\tilde{W}}_1^T P_x \tilde{W}_1\right\} + \frac{2}{k_2} tr\left\{\dot{\tilde{W}}_2^T P_x \tilde{W}_2\right\} \\
&= \Delta x^T (A^{*T} P_x + P_x A^*) \Delta x \\
&\quad + x_{nn}{}^T \tilde{A}^T P_x \Delta x + \tilde{\sigma}_1^T W_1^{*T} P_x \Delta x + \sigma_1^T(x_{nn}, y_{nn}) \tilde{W}_1^T P_x \Delta x + \gamma(U)^T \tilde{W}_2^T P_x \Delta x \\
&\quad + \Delta x^T P_x \tilde{A} x_{nn} + \Delta x^T P_x W_1^* \tilde{\sigma}_1 + \Delta x^T P_x \tilde{W}_1 \sigma_1(x_{nn}, y_{nn}) + \Delta x^T P_x \tilde{W}_2 \gamma(U) \\
&\quad + \frac{2}{k_A} tr\left\{\dot{\tilde{A}}^T P_x \tilde{A}\right\} + \frac{2}{k_1} tr\left\{\dot{\tilde{W}}_1^T P_x \tilde{W}_1\right\} + \frac{2}{k_2} tr\left\{\dot{\tilde{W}}_2^T P_x \tilde{W}_2\right\} \\
\dot{V}_y &= (\Delta\dot{y}^T P_y \Delta y + \Delta y^T P_y \Delta\dot{y}) + \frac{2}{k_B} tr\left\{\dot{\tilde{B}}^T P_y \tilde{B}\right\} + \frac{2}{k_3} tr\left\{\dot{\tilde{W}}_3^T P_y \tilde{W}_3\right\} + \frac{2}{k_4} tr\left\{\dot{\tilde{W}}_4^T P_y \tilde{W}_4\right\} \\
&= (1/\varepsilon)\left[\Delta y^T B^{*T} + y_{nn}{}^T \tilde{B}^T + \tilde{\sigma}_2^T W_3^{*T} + \sigma_2^T(x_{nn}, y_{nn}) \tilde{W}_3^T + \gamma(U)^T \tilde{W}_4^T\right] P_y \Delta y \\
&\quad + (1/\varepsilon) \Delta y^T P_y \left[B^* \Delta y + \tilde{B} y_{nn} + W_3^* \tilde{\sigma}_2 + \tilde{W}_3 \sigma_2(x_{nn}, y_{nn}) + \tilde{W}_4 \gamma(U)\right] \\
&\quad + \frac{2}{k_B} tr\left\{\dot{\tilde{B}}^T P_y \tilde{B}\right\} + \frac{2}{k_3} tr\left\{\dot{\tilde{W}}_3^T P_y \tilde{W}_3\right\} + \frac{2}{k_4} tr\left\{\dot{\tilde{W}}_4^T P_y \tilde{W}_4\right\} \\
&= (1/\varepsilon) \Delta y^T (B^{*T} P_y + P_y B^*) \Delta y \\
&\quad + (1/\varepsilon)\left[y_{nn}{}^T \tilde{B}^T P_y \Delta y + \tilde{\sigma}_2^T W_3^{*T} P_y \Delta y + \sigma_2^T(x_{nn}, y_{nn}) \tilde{W}_3^T P_y \Delta y + \gamma(U)^T \tilde{W}_4^T P_y \Delta y\right] \\
&\quad + (1/\varepsilon)\left[\Delta y^T P_y \tilde{B} y_{nn} + \Delta y^T P_y W_3^* \tilde{\sigma}_2 + \Delta y^T P_y \tilde{W}_3 \sigma_2(x_{nn}, y_{nn}) + \Delta y^T P_y \tilde{W}_4 \gamma(U)\right] \\
&\quad + \frac{2}{k_B} tr\left\{\dot{\tilde{B}}^T P_y \tilde{B}\right\} + \frac{2}{k_3} tr\left\{\dot{\tilde{W}}_3^T P_y \tilde{W}_3\right\} + \frac{2}{k_4} tr\left\{\dot{\tilde{W}}_4^T P_y \tilde{W}_4\right\}
\end{aligned} \tag{4.11}$$

Since all the terms here are scalar, therefore one has

$$\begin{aligned}
\dot{V}_x &= \Delta x^T (A^{*^T} P_x + P_x A^*)\Delta x + 2\Delta x^T P_x W_1^* \widetilde{\sigma} \\
&\quad + 2\Delta x^T P_x \widetilde{A} x_{nn} + 2\Delta x^T P_x \widetilde{W}_1 \sigma_1 (x_{nn}, y_{nn}) + 2\Delta x^T P_x \widetilde{W}_2 \gamma(U) \\
&\quad + \frac{2}{k_A} tr\left\{\dot{\widetilde{A}}^T P_x \widetilde{A}\right\} + \frac{2}{k_1} tr\left\{\dot{\widetilde{W}}_1^T P_x \widetilde{W}_1\right\} + \frac{2}{k_2} tr\left\{\dot{\widetilde{W}}_2^T P_x \widetilde{W}_2\right\} \\
\dot{V}_y &= (1/\varepsilon)\Delta y^T (B^{*^T} P_y + P_y B^*)\Delta y + (1/\varepsilon) 2\Delta y^T P_y W_3^* \widetilde{\sigma}_2 \\
&\quad + (1/\varepsilon) 2\Delta y^T P_y \widetilde{B} y_{nn} + (1/\varepsilon) 2\Delta y^T P_y \widetilde{W}_3 \sigma_2 (x, y) + (1/\varepsilon) 2\Delta y^T P_y \widetilde{W}_4 \phi_2 (x, y) \\
&\quad + \frac{2}{k_B} tr\left\{\dot{\widetilde{B}}^T P_y \widetilde{B}\right\} + \frac{2}{k_3} tr\left\{\dot{\widetilde{W}}_3^T P_y \widetilde{W}_3\right\} + \frac{2}{k_4} tr\left\{\dot{\widetilde{W}}_4^T P_y \widetilde{W}_4\right\}
\end{aligned} \tag{4.12}$$

If we applying the adaptive law as (4.9) and taking the following facts into consideration:

$$\dot{\widetilde{A}} = -\dot{A}, \dot{\widetilde{B}} = -\dot{B}, \dot{\widetilde{W}}_{1,2,3,4} = -\dot{W}_{1,2,3,4}$$

$$\begin{aligned}
2\Delta x^T P_x \widetilde{A} x_{nn} &= 2tr\left\{x_{nn} \Delta x^T P_x \widetilde{A}\right\} \\
2\Delta x^T P_x \widetilde{W}_1 \sigma_1 (x_{nn}, y_{nn}) &= 2tr\left\{\sigma_1 (x_{nn}, y_{nn}) \Delta x^T P_x \widetilde{W}\right\} \\
2\Delta x^T P_x \widetilde{W}_2 \gamma(U) &= 2tr\left\{\gamma(U) \Delta x^T P_x \widetilde{W}_2\right\} \\
2\Delta y^T P_y \widetilde{B} y_{nn} &= 2tr\left\{y_{nn} \Delta y^T P_y \widetilde{B}\right\} \\
2\Delta y^T P_y \widetilde{W}_3 \sigma_2 (x, y) &= 2tr\left\{\sigma_2 (x, y) \Delta y^T P_y \widetilde{W}_3\right\} \\
2\Delta y^T P_y \widetilde{W}_4 \phi_2 (x, y) &= 2tr\left\{\phi_2 (x, y) \Delta y^T P_y \widetilde{W}_4\right\}
\end{aligned}$$

(4.12) becomes

$$
\begin{aligned}
\dot{V}_x &= \Delta x^T (A^{*T} P_x + P_x A^*)\Delta x + 2\Delta x^T P_x W_1^* \tilde{\sigma} \\
&= 2tr\left\{\left(x_{nn}\Delta x^T + \frac{1}{k_A}\dot{\tilde{A}}^T \right) P_x \tilde{A} \right\} + 2tr\left\{\left(\sigma_1(x_{nn}, y_{nn})\Delta x^T + \frac{1}{k_1}\dot{\tilde{W}}_1^T \right) P_x \tilde{W} \right\} \\
&\quad + 2tr\left\{\left(\gamma(U)\Delta x^T + \frac{1}{k_2}\dot{\tilde{W}}_2^T \right) P_x \tilde{W}_2 \right\} \\
&= \Delta x^T (A^{*T} P_x + P_x A^*)\Delta x + 2\Delta x^T P_x W_1^* \tilde{\sigma} \\
\dot{V}_y &= (1/\varepsilon)\Delta y^T (B^{*T} P_y + P_y B^*)\Delta y + (1/\varepsilon) 2\Delta y^T P_y W_3^* \tilde{\sigma}_2 \\
&\quad (1/\varepsilon) 2tr\left\{\left(y_{nn}\Delta y^T + \frac{\varepsilon}{k_B}\dot{\tilde{B}}^T \right) P_y \tilde{B} \right\} + (1/\varepsilon) 2tr\left\{\left(\sigma_2(x, y)\Delta y^T + \frac{\varepsilon}{k_3}\dot{\tilde{W}}_3^T \right) P_y \tilde{W}_3 \right\} \\
&\quad + (1/\varepsilon) 2tr\left\{\left(\phi_2(x, y)\Delta y^T + \frac{\varepsilon}{k_4}\dot{\tilde{W}}_4^T \right) P_y \tilde{W}_4 \right\} \\
&= (1/\varepsilon)\Delta y^T (B^{*T} P_y + P_y B^*)\Delta y + (1/\varepsilon) 2\Delta y^T P_y W_3^* \tilde{\sigma}_2
\end{aligned} \tag{4.13}
$$

Using the following matrix inequality:

$$
X^T Y + (X^T Y)^T \le X^T \Lambda^{-1} X + Y^T \Lambda Y \tag{4.14}
$$

where $X, Y \in R^{j\times k}$ are any matrices, $\Lambda \in R^{j\times k}$ is any positive definite matrix. From Assumptions 4.1, 4.2, one obtains

$$
\begin{aligned}
2\Delta x^T P_x W_1^* \tilde{\sigma}_1 &\le \Delta x^T P_x W_1^* \Lambda_1^{-1} W_1^* P_x \Delta x + \tilde{\sigma}_1^T \Lambda_1 \tilde{\sigma}_1 \\
&\le \Delta x^T P_x \overline{W}_1 P_x \Delta x + \Delta x^T D_1 \Delta x + \Delta y^T D_1 \Delta y \\
2\Delta y^T P_y W_3^* \tilde{\sigma}_2 &\le \Delta y^T P_y W_3^* \Lambda_3^{-1} W_3^* P_y \Delta y + \tilde{\sigma}_2^T \Lambda_3 \tilde{\sigma}_2 \\
&\le \Delta y^T P_y \overline{W}_3 P_y \Delta y + \Delta x^T D_2 \Delta x + \Delta y^T D_2 \Delta y
\end{aligned} \tag{4.15}
$$

Hence, from (4.13), one has

$$
\begin{aligned}
\dot{V}_x &\le \Delta x^T [A^{*T} P_x + P_x A^* + P_x \overline{W}_1 P_x + D_1 + (1/\varepsilon) D_3]\Delta x \\
&= \Delta x^T [A^{*T} P_x + P_x A^* + P_x \overline{W}_1 P_x + D_1 + (1/\varepsilon) D_3 + Q_{xo}]\Delta x - \Delta x^T Q_{xo} \Delta x \\
\dot{V}_y &\le (1/\varepsilon) S_y \Delta y^T [B^{*T} P_y + P_y B^* + P_y \overline{W}_3 P_y + D_3 + \varepsilon D_1]\Delta y \\
&= (1/\varepsilon) S_y \Delta y^T [B^{*T} P_y + P_y B^* + P_y \overline{W}_3 P_y + D_3 + \varepsilon D_1 + Q_{yo}]\Delta y - (1/\varepsilon)\Delta y^T Q_{yo} \Delta y
\end{aligned} \tag{4.16}
$$

Then applying the Assumption 4.3, we can get

$$\begin{aligned}&\dot{V}_x = -\Delta x^T Q_{xo} \Delta x = -\|\Delta x\|^2_{Q_x} \le 0, \dot{V}_y = -(1/\varepsilon)\Delta y^T Q_{yo}\Delta - (1/\varepsilon)\|\Delta y\|^2_{Q_y} \le 0 \\ &\dot{V} = \dot{V}_x + \dot{V}_y \le 0\end{aligned} \qquad (4.17)$$

where V_x, V_y are positive definite functions and $\dot{V}_x, \dot{V}_y \le 0$ can be achieved by using the learning laws (4.9) which implies $\Delta x, \Delta y, W_{1,2,3,4}, A, B \in L_\infty$.

Furthermore, $x_{nn} = \Delta x + x, y_{nn} = \Delta y + y$ are also bounded. From the error dynamics (4.7), we can draw the conclusion that $\dot{\Delta x}, \dot{\Delta y} \in L_\infty$. Since V_x, V_y are non-increasing function of the time and bounded from below, the limits of V_x, V_y ($\lim_{t\to\infty} V_{x,y} = V_{x,y}(\infty)$) exist. Therefore by integrating $\dot{V}_x$, $\dot{V}_y$ on both sides from 0 to ∞, we have

$$\begin{aligned}&\int_0^\infty \|\Delta x\|^2_{Q_x} = [V_x(0) - V_x(\infty)] < \infty \\ &\int_0^\infty \|\Delta y\|^2_{Q_y} = \varepsilon[V_y(0) - V_y(\infty)] < \infty\end{aligned} \qquad (4.18)$$

which imply that $\Delta x, \Delta y \in L_2$. Since $\Delta x, \Delta y \in L_2 \bigcap L_\infty$ and $\dot{\Delta x}, \dot{\Delta y} \in L_\infty$, using Barbalat's Lemma [13] we have $\lim_{t\to\infty} \Delta x = 0, \lim_{t\to\infty} \Delta y = 0$. Given that the control input U and $\sigma_{1,2}(\cdot), \phi_{1,2}(\cdot)$ are bounded, it is concluded that $\lim_{t\to\infty} \dot{W}_{1,2} = 0,\ \lim_{t\to\infty} \dot{W}_{3,4} = 0$.

The structure of improved identification scheme is illustrated in Figure 4.1.

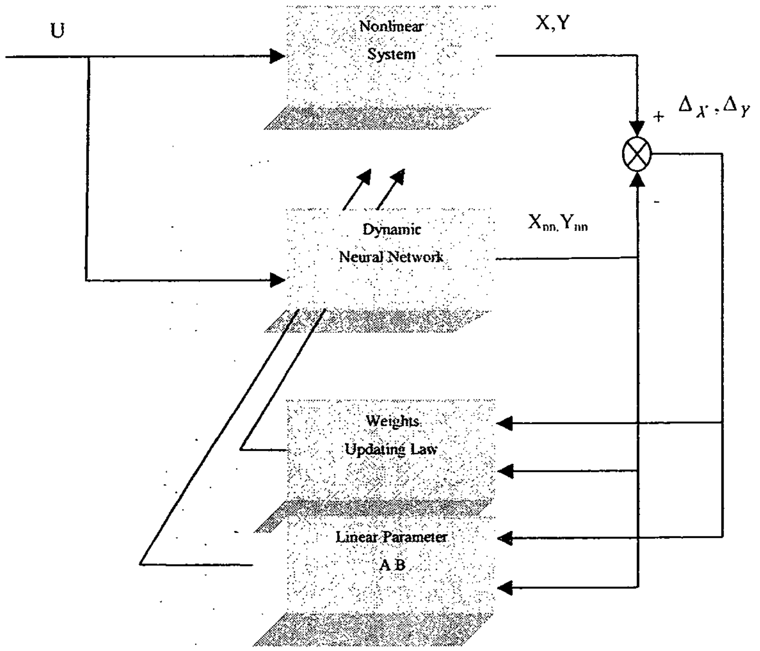


Figure 4-1 Improved Identification Scheme

4.1.2 Identification for nonlinear systems with bounded un-modeled dynamics

For more general and realistic situations, we will consider the case where the dynamic neural network (4.2) does not match the given nonlinear system (3.1) exactly. Then we can define the modeling error as

$$\begin{aligned} \Delta f_x &= f_x - \left(A^* x + W_1^* \sigma_1(x,y) + W_2^* \gamma(U)\right) \\ \Delta f_y &= (1/\varepsilon)\left(f_y - \left(B^* y + W_3^* \sigma_2(x,y) + W_4^* \gamma(U)\right)\right) \end{aligned} \tag{4.19}$$

Now the nonlinear system can be represented as

$$\begin{aligned} \dot{x} &= A^* x + W_1^* \sigma_1(x,y) + W_2^* \gamma(U) - \Delta f_x \\ \varepsilon \dot{y} &= B^* y + W_3^* \sigma_2(x,y) + W_4^* \gamma(U) - \Delta f_y \end{aligned} \tag{4.20}$$

where $W_1^*, W_2^*, W_3^*, W_4^*$ are unknown nominal constant matrices, the vector functions Δf_x

,Δf_y can be regarded as modeling error and disturbances which are assumed to be bounded as $\|\Delta f_x\| \leq \Delta \bar{f}_x, \|\Delta f_y\| \leq \Delta \bar{f}_y$ and A^*, B^* are unknown nominal constant Hurwitz matrices.

If we defined the identification error same as in (4.6), the error dynamic equations become

$$\begin{aligned} \Delta \dot{x} &= A^* \Delta x + \widetilde{A} x_{nn} + W_1^* \widetilde{\sigma}_1 + \widetilde{W}_1 \sigma_1(x_{nn}, y_{nn}) + \widetilde{W}_2 \gamma(U) + \Delta f_x \\ \varepsilon \Delta \dot{y} &= B^* \Delta y + \widetilde{B} y_{nn} + W_3^* \widetilde{\sigma}_2 + \widetilde{W}_3 \sigma_2(x_{nn}, y_{nn}) + \widetilde{W}_4 \gamma(U) + \Delta f_y \end{aligned} \qquad (4.21)$$

where $\widetilde{W}_1 = W_1^* - W_1$, $\widetilde{W}_2 = W_2^* - W_2$, $\widetilde{W}_3 = W_3^* - W_3$, $\widetilde{W}_4 = W_4^* - W_4$ and $\widetilde{A} = A^* - A$, $\widetilde{B} = B^* - B$.

Assumption 4.4: The matrices A^*, B^* are unknown nominal constant Hurwitz matrices. Define

$$\begin{aligned} R_x &= \overline{W}_1 + \Lambda_2^{-1} \qquad & Q_x &= D_1 + (1/\varepsilon)(S_y / S_x) D_3 + Q_{xo} \\ R_y &= \overline{W}_3 + \Lambda_4^{-1} \qquad & Q_y &= D_3 + \varepsilon (S_x / S_y) D_1 + Q_{yo} \end{aligned}$$

where the function S_x, S_y are defined as

$$S_x = \left[1 - \frac{H_x}{\left\| P_x^{1/2} \Delta x \right\|} \right]_+ \qquad S_y = \left[1 - \frac{H_y}{\left\| P_y^{1/2} \Delta y \right\|} \right]_+ \qquad (4.22)$$

where $[\,\circledast\,]_+ = \max\{\circledast, 0\}$, and

$$\begin{aligned} H_x &= \left| \sqrt{\frac{\lambda_{\max}(P_x)}{\lambda_{\min}(Q_{xo})} (\lambda_{\max}(\Lambda_2) \Delta \bar{f}_x^2 + \frac{\lambda_{\max}(D_1) H_y^2}{\lambda_{\min}(P_y)})} \right| \\ H_y &= \left| \sqrt{\frac{\lambda_{\max}(P_y)}{\lambda_{\min}(Q_{yo})} (\lambda_{\max}(\Lambda_4) \Delta \bar{f}_y^2 + \frac{\lambda_{\max}(D_3) H_x^2}{\lambda_{\min}(P_x)})} \right| \end{aligned} \qquad (4.23)$$

If one can select proper Q_{xo} , Q_{yo} satisfying the conditions in Lemma 1, there exist matrices P_x, P_y satisfying the following equations:

$$\begin{aligned} A^{*T}P_x + P_x A^* + P_x R_x P_x + Q_x = 0 \\ B^{*T}P_y + P_y B^* + P_y R_y P_y + Q_y = 0 \end{aligned} \tag{4.24}$$

Theorem 4.2: Consider the nonlinear system (3.1) and identification model (4.2) the updating laws for the dynamic neural networks

$$\begin{aligned} \dot{A} &= S_x k_A \Delta x x_{nn}^T & \dot{B} &= (1/\varepsilon) S_y k_B \Delta y y_{nn}^T \\ \dot{W}_1 &= S_x k_1 \Delta x \sigma_1^T(x_{nn}, y_{nn}) & \dot{W}_3 &= (1/\varepsilon) S_y k_3 \Delta y \sigma_2^T(x_{nn}, y_{nn}) \\ \dot{W}_2 &= S_x k_2 \Delta x \gamma(U)^T & \dot{W}_4 &= (1/\varepsilon) S_y k_4 \Delta y \gamma(U) \end{aligned} \tag{4.23}$$

where k_A, k_B, k_1, k_2, k_3, k_4 are positive constants

can guarantee the following stability properties:

1) $\Delta x, \Delta y, W_{1,2,3,4}, A, B \in L_\infty$ and $\lim\limits_{t\to\infty} \dot{\widetilde{W}}_{1,2} = 0, \lim\limits_{t\to\infty} \dot{\widetilde{W}}_{3,4} = 0$

2) The identification error satisfies the following tracking performance

$$\begin{aligned} \int_0^T S_x \Delta x^T Q_{xo} \Delta x + (1/\varepsilon)\int_0^T S_y \Delta y^T Q_{yo} \Delta y \le V_o + T\Bigg(\Bigg(\lambda_{\max}(\Lambda_2)\Delta\bar{f}_x^2 + \frac{\lambda_{\max}(D_1)H_y^2}{\lambda_{\min}(P_y)}\Bigg) \\ + (1/\varepsilon)\Bigg(\lambda_{\max}(\Lambda_4)\Delta\bar{f}_y^2 + \frac{\lambda_{\max}(D_3)H_x^2}{\lambda_{\min}(P_x)}\Bigg)\Bigg) \end{aligned} \tag{4.26}$$

Proof: The Lyapunov synthesis method is used to derive the stable adaptive laws. Consider the Lyapunov function candidate:

$$\begin{aligned} V &= V_x + V_y \\ V_x &= \left[\left\|P_x^{1/2}\Delta x\right\| - H_x\right]_+^2 + \frac{1}{k_1} tr\left\{\widetilde{W}_1^T P_x \widetilde{W}_1\right\} + \frac{1}{k_2} tr\left\{\widetilde{W}_2^T P_x \widetilde{W}_2\right\} + \frac{1}{k_A} tr\left\{\widetilde{A}^T P_x \widetilde{A}\right\} \\ V_y &= \left[\left\|P_y^{1/2}\Delta y\right\| - H_y\right]_+^2 + \frac{1}{k_3} tr\left\{\widetilde{W}_3^T P_y \widetilde{W}_3\right\} + \frac{1}{k_4} tr\left\{\widetilde{W}_4^T P_y \widetilde{W}_4\right\} + \frac{1}{k_B} tr\left\{\widetilde{B}^T P_y \widetilde{B}\right\} \end{aligned} \tag{4.27}$$

Using Exercise 1 in section 4.2 in [18] and differentiating (4.27) yield

$$
\begin{aligned}
\dot{V}_x &= 2\left[\left\|P_x^{1/2}\Delta x\right\| - H_x\right]_+ \frac{\left(P_x^{1/2}\Delta x\right)^T}{\left\|P_x^{1/2}\Delta x\right\|} P_x^{1/2}\dot{\Delta} x \\
&\quad + \frac{2}{k_A} tr\left\{\dot{\widetilde{A}}^T P_x \widetilde{A}\right\} + \frac{2}{k_1} tr\left\{\dot{\widetilde{W}}_1^T P_x \widetilde{W}_1\right\} + \frac{2}{k_B} tr\left\{\dot{\widetilde{W}}_2^T P_x \widetilde{W}_2\right\} \\
&= 2S_x \Delta x^T P_x \Delta\dot{x} + \frac{2}{k_A} tr\left\{\dot{\widetilde{A}}^T P_x \widetilde{A}\right\} + \frac{2}{k_1} tr\left\{\dot{\widetilde{W}}_1^T P_x \widetilde{W}_1\right\} + \frac{2}{k_B} tr\left\{\dot{\widetilde{W}}_2^T P_x \widetilde{W}_2\right\} \\
\dot{V}_y &= 2\left[\left\|P_y^{1/2}\Delta y\right\| - H_y\right]_+ \frac{\left(P_y^{1/2}\Delta y\right)^T}{\left\|P_y^{1/2}\Delta y\right\|} P_y^{1/2}\dot{\Delta} y \\
&\quad + \frac{2}{k_B} tr\left\{\dot{\widetilde{B}}^T P_y \widetilde{B}\right\} + \frac{2}{k_3} tr\left\{\dot{\widetilde{W}}_3^T P_y \widetilde{W}_3\right\} + \frac{2}{k_4} tr\left\{\dot{\widetilde{W}}_4^T P_y \widetilde{W}_4\right\} \\
&= 2S_y \Delta y^T P_y \Delta\dot{y} + \frac{2}{k_B} tr\left\{\dot{\widetilde{B}}^T P_y \widetilde{B}\right\} + \frac{2}{k_3} tr\left\{\dot{\widetilde{W}}_3^T P_y \widetilde{W}_3\right\} + \frac{2}{k_4} tr\left\{\dot{\widetilde{W}}_4^T P_y \widetilde{W}_4\right\}
\end{aligned}
\tag{4.28}
$$

Since the neural network's weights are adjusted as (4.23), the derivatives of the neural network weights and matrices satisfy the following $\dot{W}_{1,2,3,4} = \dot{\widetilde{W}}_{1,2,3,4}$, $\dot{A} = \dot{\widetilde{A}}, \dot{B} = \dot{\widetilde{B}}$.

Using error dynamics (4.21), Equation (4.28) becomes

$$
\begin{aligned}
\dot{V}_x &= S_x[\Delta x^T (A^{*T} P_x + P_x A^*)\Delta x + 2\Delta x^T P_x W_1^* \widetilde{\sigma}_1 + 2\Delta x^T P_x \Delta f_x] \\
\dot{V}_y &= S_y(1/\varepsilon)[\Delta y^T (B^{*T} P_y + P_y B^*)\Delta y + 2\Delta y^T P_y W_3^* \widetilde{\sigma}_2 + 2\Delta y^T P_y \Delta f_y]
\end{aligned}
\tag{4.29}
$$

Using the matrix inequality (4.14) and Assumptions 4.1, 4.2, one obtains

$$
\begin{aligned}
2\Delta x^T P_x W_1^* \widetilde{\sigma}_1 &\le \Delta x^T P_x W_1^* \Lambda_1^{-1} W_1^* P_x \Delta x + \widetilde{\sigma}_1^T \Lambda_1 \widetilde{\sigma}_1 \\
&\le \Delta x^T P_x \overline{W}_1 P_x \Delta x + \Delta x^T D_1 \Delta x + \Delta y^T D_1 \Delta y \\
2\Delta y^T P_y W_3^* \widetilde{\sigma}_2 &\le \Delta y^T P_y W_3^* \Lambda_3^{-1} W_3^* P_y \Delta y + \widetilde{\sigma}_2^T \Lambda_3 \widetilde{\sigma}_2 \\
&\le \Delta y^T P_y \overline{W}_3 P_y \Delta y + \Delta x^T D_2 \Delta x + \Delta y^T D_2 \Delta y
\end{aligned}
\tag{4.30}
$$

and

$$\begin{aligned} &2\Delta x^T P_x \Delta f_x \le \Delta x^T P_x \Lambda_2^{-1} P_x \Delta x + \Delta f_x^T \Lambda_2 \Delta f_x \\ &2(1/\varepsilon)\Delta y^T P_y \Delta f_y \le (1/\varepsilon)\left(\Delta y^T P_y \Lambda_4^{-1} P_y \Delta y + \Delta f_y^T \Lambda_4 \Delta f_y\right) \end{aligned} \tag{4.31}$$

Hence, from (4.29) one has

$$\begin{aligned} \dot{V}_x &\le S_x \Delta x^T [A^{*T} P_x + P_x A^* + P_x(\overline{W}_1 + \Lambda_2^{-1})P_x + D_1]\Delta x \\ &\quad + S_x \Delta y^T D_1 \Delta y + S_x \Delta f_x^T \Lambda_2 \Delta f_x \\ \dot{V}_y &\le (1/\varepsilon) S_y \Delta y^T [B^{*T} P_y + P_y B^* + P_y(\overline{W}_3 + \Lambda_4^{-1})P_y + D_3]\Delta y \\ &\quad + (1/\varepsilon) S_y \Delta x^T D_3 \Delta x + (1/\varepsilon) S_y \Delta f_y^T \Lambda_4 \Delta f_y \end{aligned} \tag{4.32}$$

a) When the identification errors are both larger than the thresholds (i.e. $S_x>0$, $S_y>0$).

One has

$$\begin{aligned} \dot{V}_x &\le S_x \Delta x^T [A^{*T} P_x + P_x A^* + P_x(\overline{W}_1 + \Lambda_2^{-1})P_x + D_1 + (1/\varepsilon)(S_y/S_x)D_3 + Q_{xo}]\Delta x \\ &\quad - S_x \Delta x^T Q_{xo} \Delta x + S_x \Delta f_x^T \Lambda_2 \Delta f_x \\ \dot{V}_y &\le (1/\varepsilon) S_y \Delta y^T [B^{*T} P_y + P_y B^* + P_y(\overline{W}_3 + \Lambda_4^{-1})P_y + D_3 + \varepsilon(S_x/S_y)D_1 + Q_{yo}]\Delta y \\ &\quad - (1/\varepsilon) S_y \Delta y^T Q_{yo} \Delta y + (1/\varepsilon) S_y \Delta f_y^T \Lambda_4 \Delta f_y \end{aligned} \tag{4.33}$$

Then applying the Assumptions 4.4, one can obtain

$$\begin{aligned} \dot{V} &= \dot{V}_x + \dot{V}_y \\ &\le -S_x\left(\Delta x^T Q_{xo} \Delta x - \Delta f_x^T \Lambda_2 \Delta f_x\right) - (1/\varepsilon) S_y \left(\Delta y^T Q_{yo} \Delta y - \Delta f_y^T \Lambda_4 \Delta f_y\right) \\ &\le -S_x\left(\lambda_{\min}(Q_{xo})\|\Delta x\|^2 - \lambda_{\max}(\Lambda_2)\|\Delta f_x\|^2\right) - (1/\varepsilon) S_y \left(\lambda_{\min}(Q_{yo})\|\Delta y\|^2 - \lambda_{\max}(\Lambda_4)\|\Delta f_y\|^2\right) \\ &\le -S_x\left(\lambda_{\min}(Q_{xo})\|\Delta x\|^2 - \lambda_{\max}(\Lambda_2)\Delta \bar{f}_x^2\right) - (1/\varepsilon) S_y \left(\lambda_{\min}(Q_{yo})\|\Delta y\|^2 - \lambda_{\max}(\Lambda_4)\Delta \bar{f}_y^2\right) \\ &= -S_x \frac{\lambda_{\min}(Q_{xo})}{\lambda_{\max}(P_x)}\left(\lambda_{\max}(P_x)\|\Delta x\|^2 - \frac{\lambda_{\max}(P_x)}{\lambda_{\min}(Q_{xo})}\lambda_{\max}(\Lambda_2)\Delta \bar{f}_x^2\right) \\ &\quad - (1/\varepsilon) S_y \frac{\lambda_{\min}(Q_{yo})}{\lambda_{\max}(P_y)}\left(\lambda_{\max}(P_y)\|\Delta y\|^2 - \frac{\lambda_{\max}(P_y)}{\lambda_{\min}(Q_{yo})}\lambda_{\max}(\Lambda_4)\Delta \bar{f}_y^2\right) \\ &\le -S_x \frac{\lambda_{\min}(Q_{xo})}{\lambda_{\max}(P_x)}\left(\|P_x^{1/2}\Delta x\|^2 - H_x^2\right) - (1/\varepsilon) S_y \frac{\lambda_{\min}(Q_{yo})}{\lambda_{\max}(P_y)}\left(\|P_y^{1/2}\Delta y\|^2 - H_y^2\right) \le 0 \end{aligned} \tag{4.34}$$

b) When the identification error of Y is smaller than the threshold, ($S_x>0$, $S_y=0$)

From (4.32) one has $\left\|P_y^{1/2}\Delta y\right\| \le H_y$ and $\dot{V}_y = 0$

$$\dot{V} = \dot{V}_x \le S_x \Delta x^T [A^{*T} P_x + P_x A^* + P_x(\overline{W}_1 + \Lambda_2^{-1})P_x + D_1 + (1/\varepsilon)(S_y/S_x)D_3 + Q_{xo}]\Delta x \\ - S_x \Delta x^T Q_{xo} \Delta x + S_x \Delta f_x^T \Lambda_2 \Delta f_x + S_x \Delta y^T D_1 \Delta y \tag{4.35}$$

Then applying the Assumptions 4, one can obtain

$$\begin{aligned}\dot{V} &\le -S_x\left(\Delta x^T Q_{xo} \Delta x - \Delta f_x^T \Lambda_2 \Delta f_x - \Delta y^T D_1 \Delta y\right) \\ &\le -S_x\left(\lambda_{\min}(Q_{xo})\|\Delta x\|^2 - \lambda_{\max}(\Lambda_2)\|\Delta f_x\|^2 - \lambda_{\max}(D_1)\|\Delta y\|^2\right) \\ &\le -S_x\left(\lambda_{\min}(Q_{xo})\|\Delta x\|^2 - \lambda_{\max}(\Lambda_2)\Delta \bar{f}_x^2 - \frac{\lambda_{\max}(D_1)H_y^2}{\lambda_{\min}(P_y)}\right) \\ &= -S_x \frac{\lambda_{\min}(Q_{xo})}{\lambda_{\max}(P_x)}\left(\lambda_{\max}(P_x)\|\Delta x\|^2 - \frac{\lambda_{\max}(P_x)}{\lambda_{\min}(Q_{xo})}\left(\lambda_{\max}(\Lambda_2)\Delta \bar{f}_x^2 + \frac{\lambda_{\max}(D_1)H_y^2}{\lambda_{\min}(P_y)}\right)\right) \\ &\le -S_x \frac{\lambda_{\min}(Q_{xo})}{\lambda_{\max}(P_x)}\left(\left\|P_x^{1/2}\Delta x\right\|^2 - H_x^2\right) \le 0\end{aligned} \tag{4.36}$$

c) When the identification error of X is smaller than the threshold (S_x=0, S_y>0)

From (4.32) one has $\left\|P_x^{1/2}\Delta x\right\| \le H_x$ and $\dot{V}_x = 0$

$$\dot{V} = \dot{V}_y \le (1/\varepsilon)S_y \Delta y^T [B^{*T} P_y + P_y B^* + P_y(\overline{W}_3 + \Lambda_4^{-1})P_y + D_3 + \varepsilon(S_x/S_y)D_1 + Q_{yo}]\Delta y \\ - (1/\varepsilon)S_y \Delta y^T Q_{yo} \Delta y + (1/\varepsilon)S_y \Delta f_y^T \Lambda_4 \Delta f_y + (1/\varepsilon)S_y \Delta x^T D_3 \Delta x \tag{4.37}$$

Then applying the Assumptions 4.4 one can obtain

$$
\begin{aligned}
\dot{V} &\le -(1/\varepsilon)S_y\left(\Delta y^T Q_{yo}\Delta y - \Delta f_y^T \Lambda_4 \Delta f_y - \Delta x^T D_3 \Delta x\right) \\
&\le -(1/\varepsilon)S_y\left(\lambda_{\min}(Q_{yo})\|\Delta y\|^2 - \lambda_{\max}(\Lambda_4)\|\Delta f_y\|^2 - \lambda_{\max}(D_{31})\|\Delta x\|^2\right) \\
&\le -(1/\varepsilon)S_y\left(\lambda_{\min}(Q_{yo})\|\Delta y\|^2 - \lambda_{\max}(\Lambda_4)\Delta\bar{f}_y^2 - \frac{\lambda_{\max}(D_3)H_x^2}{\lambda_{\min}(P_x)}\right) \\
&= -(1/\varepsilon)S_y\frac{\lambda_{\min}(Q_{yo})}{\lambda_{\max}(P_y)}\left(\lambda_{\max}(P_y)\|\Delta y\|^2 - \frac{\lambda_{\max}(P_y)}{\lambda_{\min}(Q_{yo})}\left(\lambda_{\max}(\Lambda_4)\Delta\bar{f}_y^2 + \frac{\lambda_{\max}(D_3)H_x^2}{\lambda_{\min}(P_x)}\right)\right) \\
&\le -(1/\varepsilon)S_y\frac{\lambda_{\min}(Q_{yo})}{\lambda_{\max}(P_y)}\left(\|P_y^{1/2}\Delta y\|^2 - H_y^2\right) \le 0
\end{aligned}
\tag{4.38}
$$

d) When the identification errors are both smaller than the thresholds (S_x=0, S_y=0)

One has $\|P_x^{1/2}\Delta x\| \le H_x$ $\|P_y^{1/2}\Delta y\| \le H_y$ and $\dot{V} = 0$.

Since $V=V_x+V_y$ are positive definite, $\dot{V} = \dot{V}_x + \dot{V}_y \le 0$ can be achieved by using the update laws (4.23). This implies $\Delta x, \Delta y, W_{1,2,3,4}$, $A, B \in L_\infty$. Furthermore, $x_{nn} = \Delta x + x, y_{nn} = \Delta y + y$ are also bounded. From the error equations (4.21), with the assumption that error and disturbances are bounded, we can draw the conclusion that $\dot{\Delta x}, \dot{\Delta y} \in L_\infty$. Since the control input $\gamma(U)$ and $\sigma_{1,2}(\cdot)$ are bounded, it is concluded that $\lim_{t\to\infty}\dot{\tilde{W}}_{1,2} = 0,\ \lim_{t\to\infty}\dot{\tilde{W}}_{3,4} = 0$

In Case a), one has

$$
\begin{aligned}
\dot{V} &\le -S_x\left(\Delta x^T Q_{xo}\Delta x - \Delta f_x^T \Lambda_2 \Delta f_x\right) - (1/\varepsilon)S_y\left(\Delta y^T Q_{yo}\Delta y - \Delta f_y^T \Lambda_4 \Delta f_y\right) \\
&\le -S_x \Delta x^T Q_{xo}\Delta x + S_x \lambda_{\max}(\Lambda_2)\|\Delta f_x\|^2 - (1/\varepsilon)S_y \Delta y^T Q_{yo}\Delta y + (1/\varepsilon)S_y \lambda_{\max}(\Lambda_4)\|\Delta f_y\|^2 \\
&\le -S_x \Delta x^T Q_{xo}\Delta x + S_x\left(\lambda_{\max}(\Lambda_2)\Delta\bar{f}_x^2 + \frac{\lambda_{\max}(D_1)H_y^2}{\lambda_{\min}(P_y)}\right) \\
&\quad - (1/\varepsilon)S_y \Delta y^T Q_{yo}\Delta y + (1/\varepsilon)S_y\left(\lambda_{\max}(\Lambda_4)\Delta\bar{f}_y^2 + \frac{\lambda_{\max}(D_3)H_x^2}{\lambda_{\min}(P_x)}\right)
\end{aligned}
\tag{4.39}
$$

In Case b), one has

$$\begin{aligned}\dot{V} &\le -S_x\left(\Delta x^T Q_{xo}\Delta x - \Delta f_x^T \Lambda_2 \Delta f_x - \Delta y^T D_1 \Delta y\right)\\ &\le -S_x \Delta x^T Q_{xo}\Delta x + S_x\left(\lambda_{\max}(\Lambda_2)\|\Delta f_x\|^2 + \lambda_{\max}(D_1)\|\Delta y\|^2\right)\\ &\le -S_x \Delta x^T Q_{xo}\Delta x + S_x\left(\lambda_{\max}(\Lambda_2)\Delta \bar{f}_x^2 + \frac{\lambda_{\max}(D_1)H_y^2}{\lambda_{\min}(P_y)}\right)\end{aligned} \tag{4.40}$$

In Case c), one has

$$\begin{aligned}\dot{V} &\le -(1/\varepsilon)S_y\left(\Delta y^T Q_{yo}\Delta y - \Delta f_y^T \Lambda_4 \Delta f_y - \Delta x^T D_3 \Delta x\right)\\ &\le -(1/\varepsilon)S_y \Delta y^T Q_{yo}\Delta y + (1/\varepsilon)S_y\left(\lambda_{\max}(\Lambda_4)\|\Delta f_y\|^2 + \lambda_{\max}(D_{31})\|\Delta x\|^2\right)\\ &\quad -(1/\varepsilon)S_y \Delta y^T Q_{yo}\Delta y + (1/\varepsilon)S_y\left(\lambda_{\max}(\Lambda_4)\Delta \bar{f}_y^2 + \frac{\lambda_{\max}(D_3)H_x^2}{\lambda_{\min}(P_x)}\right)\end{aligned} \tag{4.41}$$

From the analysis above, we can get the conclusion that equation (4.39) can be used to represent the derivative of Lyapunov function for all the Cases a), b), c), d). Since $0 \le S_x \le 1$, $0 \le S_y \le 1$, one infers

$$\begin{aligned}&\dot{V} \le -S_x \Delta x^T Q_{xo}\Delta x + \left(\lambda_{\max}(\Lambda_2)\Delta \bar{f}_x^2 + \frac{\lambda_{\max}(D_1)H_y^2}{\lambda_{\min}(P_y)}\right)\\ &-(1/\varepsilon)S_y \Delta y^T Q_{yo}\Delta y + (1/\varepsilon)\left(\lambda_{\max}(\Lambda_4)\Delta \bar{f}_y^2 + \frac{\lambda_{\max}(D_3)H_x^2}{\lambda_{\min}(P_x)}\right)\end{aligned} \tag{4.42}$$

Since *V_x, V_y* are non-increasing function of the time and bounded, *$V_x(t)$, $V_y(0)$, $V_x(t)$, $V_y(t)$* are bounded. Therefore by integrating $\dot{V}$ on both sides from 0 to *T*, one obtains

$$\begin{aligned}V_T - V_o &\le -\int_0^T S_x \Delta x^T Q_{xo}\Delta x + T\left(\lambda_{\max}(\Lambda_2)\Delta \bar{f}_x^2 + \frac{\lambda_{\max}(D_1)H_y^2}{\lambda_{\min}(P_y)}\right)\\ &\quad -(1/\varepsilon)\int_0^T S_y \Delta y^T Q_{yo}\Delta y + (1/\varepsilon)T\left(\lambda_{\max}(\Lambda_4)\Delta \bar{f}_y^2 + \frac{\lambda_{\max}(D_3)H_x^2}{\lambda_{\min}(P_x)}\right)\end{aligned} \tag{4.43}$$

Hence, the following inequality is held

$$\begin{aligned}&\int_0^T S_x \Delta x^T Q_{xo} \Delta x + (1/\varepsilon)\int_0^T S_y \Delta y^T Q_{yo} \Delta y \\ &\le V_o - V_T + T\left(\lambda_{\max}(\Lambda_2)\Delta\bar{f}_x^2 + \frac{\lambda_{\max}(D_1)H_y^2}{\lambda_{\min}(P_y)}\right) + (1/\varepsilon)T\left(\lambda_{\max}(\Lambda_4)\Delta\bar{f}_y^2 + \frac{\lambda_{\max}(D_3)H_x^2}{\lambda_{\min}(P_x)}\right) \qquad (4.44) \\ &\le V_o + T\left(\left(\lambda_{\max}(\Lambda_2)\Delta\bar{f}_x^2 + \frac{\lambda_{\max}(D_1)H_y^2}{\lambda_{\min}(P_y)}\right) + (1/\varepsilon)\left(\lambda_{\max}(\Lambda_4)\Delta\bar{f}_y^2 + \frac{\lambda_{\max}(D_3)H_x^2}{\lambda_{\min}(P_x)}\right)\right)\end{aligned}$$

Remark 4.1: S_x and S_y are the dead-zone functions which prevent the weights drifting into infinity when the modeling error presents [19]. This is known as "parameters drift" [20] phenomenon.

It is noticed that H_x, H_y are thresholds for the identification error. For the case (a), where $S_x>0$, $S_y>0$, i.e. $\left\|P_x^{1/2}\Delta x\right\| > H_x, \left\|P_y^{1/2}\Delta y\right\| > H_y$, smaller thresholds as in (4.34) could be used, but we extend those to H_x , H_y to unify the thresholds for all the possible Cases a), b), c), d) during the entire identification process.

4.1.3 Simulation results

To illustrate the theoretical results, we use the same systems in Chapter 3 for demonstration.

Example 1: Let us consider the nonlinear system (3.16) where we use the same parameters $\alpha_1 = -5$, $\alpha_2 = -10$, $\beta_1 = 3$, $\beta_2 = 2$, $x_1(0) = -5$, $x_2(0) = -5$, and same input signals are adopted as where u_1 is a sinusoidal wave (u_1=8sin (0.05t)) and u_2 is a saw-tooth function with the amplitude 8 and frequency 0.02Hertz. The small parameter ε is selected as 0.5.

The Sigmoid functions $\sigma_{1,2}(\cdot)$ are chosen as $\frac{a}{1+\exp(-bx)}-c$ and the parameters for each sigmoid function in dynamic neural networks are listed in Table 4.1

Table 4-1 Sigmoid function parameters

	a	b	c
$\sigma_1(\cdot)$	2	2	0.5
$\sigma_2(\cdot)$	2	2	0.5

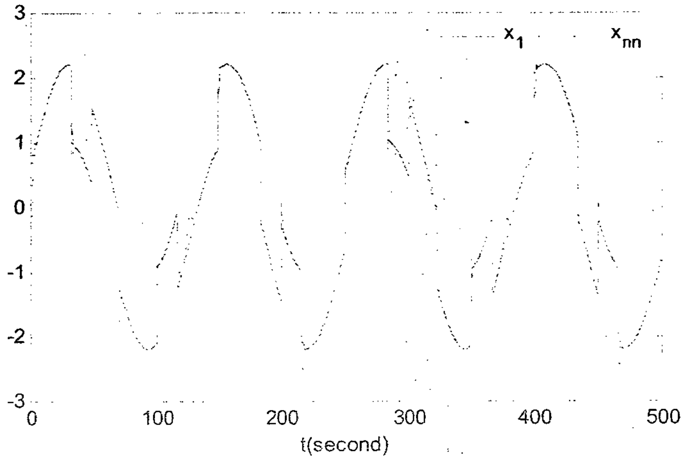


Figure 4-2 Identification result for x_1

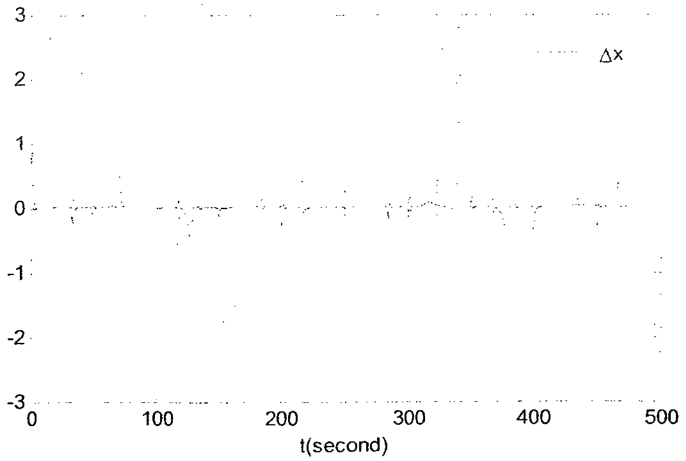


Figure 4-3 Identification error for x_1

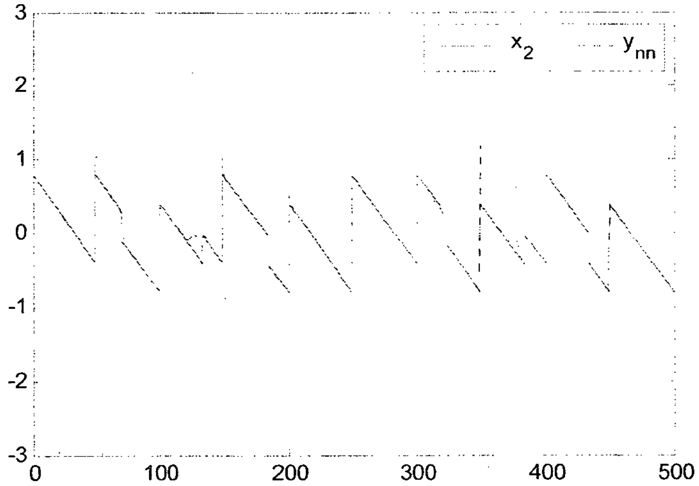


Figure 4-4 Identification result for x_2

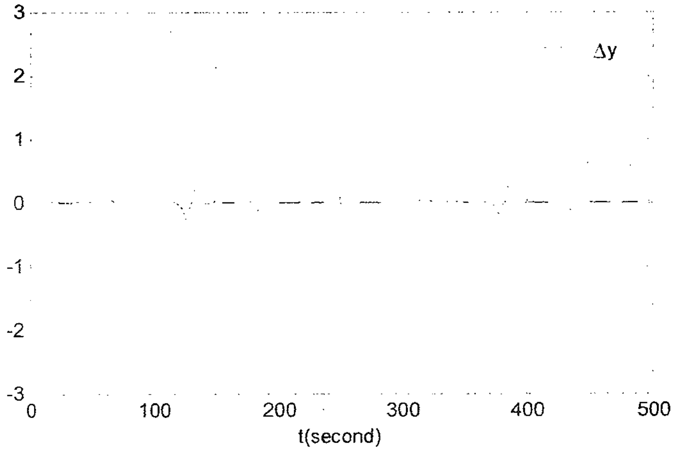


Figure 4-5 Identification error for x_2

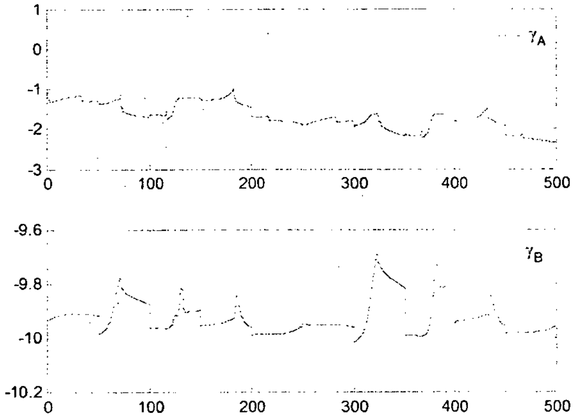


Figure 4-6 The eigenvalues of the linear parameter matrices A, B

The on-line identification results on the system are shown in Figs 4.2-4.6. From these figures, it can be seen that the state variables of the dynamic multi-time scale NN follow those of the nonlinear system accurately and quickly. The eigenvalues of the linear parameter matrix are shown in Fig.4-6. The eigenvalues for both A and B are universally smaller than zero, which means they are always stable matrices. For state variable x_1, the RMS value is 0.049168 and RMS for state variable x_2 is 0.022158. The identification results are better than those in Chapter 3.

Example 2: We consider the Hodgkin-Huxley system (3.18) with same parameters. We also focus on membrane potential E and use ELF external electric field E_w and stimulation current $\bar{I}_{ext}$ in (3.19) as the control inputs.

Case A: $E_w = 0$, $A_I = 30\mu A/cm^2$, $f_I = 10Hz$, $\varepsilon = 0.2$.

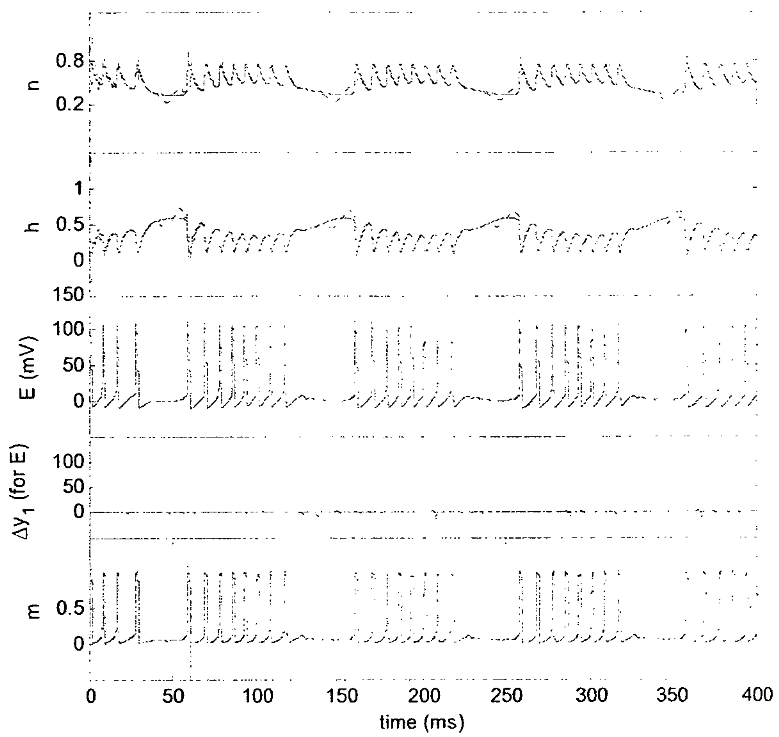


Figure 4-7 Identification results in Case A

The solid lines are the real state variables (n, h, E, m) for the HH system and the dot lines represent the identified states of the NN. The forth plot is the identification error for the membrane potential.

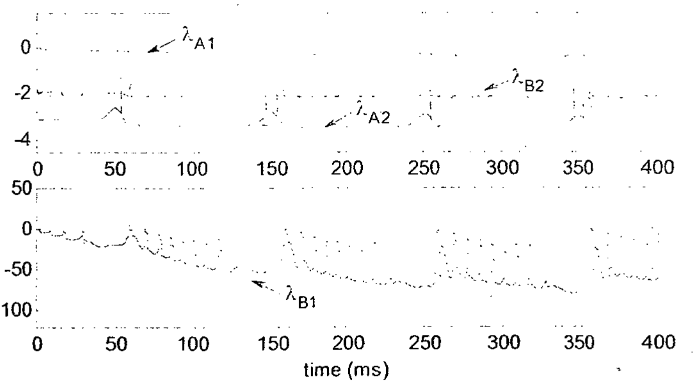


Figure 4-8 Eigenvalues of the linear matrices A, B in Case A

Case B: I_{ext}=0, A_E=10mV, f_E=115Hz, ε = 0.2.

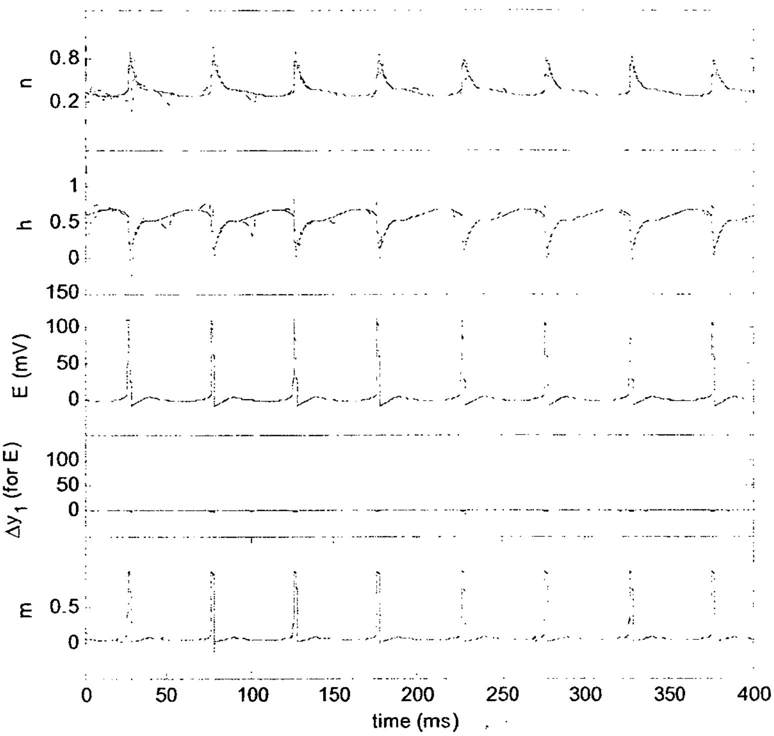


Figure 4-9 Identification results in Case B

The solid lines are the real state variables (n, h, E, m) for the HH system and the dot lines represent the identified states of the NN. The forth plot is the identification error for the membrane potential.

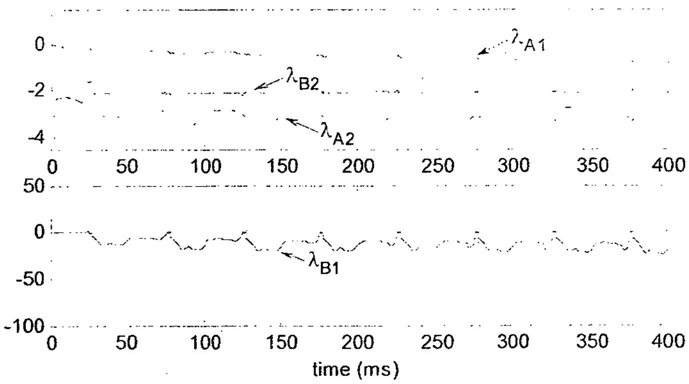


Figure 4-10 Eigenvalues of the linear matrices A, B

The identification results are presented in Figures 7-10 for (2, 2) asymptotic embedded HH model. In Case A, System is in 8/1 phase locked oscillation periodic bursting. RMS value of the state variables are RMS_n=0.333511, RMS_h=0.335092, RMS_V=0.639899, RMS_m=0.322476. In Case B, system is in the same frequency periodic spiking. RMS value of the state variables are RMS_n=0.140449, RMS_h=0.147785, RMS_V=0.784985, RMS_m=0.07245. The time scale is considered by putting $\varepsilon = 0.2$. The flexibility of linear part matrix A and B enhance the identification ability of the neural identifier. Even the single layer structure is powerful enough to successfully follow the complicated electro-physic phenomena from HH model.

The simulation results of two nonlinear systems demonstrate the states of dynamic multi-scale neural networks can track the nonlinear system state variables on-line. The identification errors approach to the thresholds. The eigenvalues of A and B converge to the steady values in both system identifications.

4.2 Multiple control methods based on Neural Network

In this section, multiple control methods are applied to accomplish the tracking task. We will utilize direct compensation, Sliding Mode Control and feedback linearization as our main control tools. The tracking problem is investigated based on the identification results from Section 4.1.

4.2.1 Tracking Error Analysis

As we mentioned before, even if the dynamic neural networks have superb learning ability to represent the nonlinear dynamic process, the modeling error are sometimes inevitable or even may affect the stability of the system. So the nonlinear system can be represented by dynamic neural networks with the updating laws (4.23):

$$\begin{aligned} \dot{x} &= Ax_{nn} + W_1\sigma_1(x_{nn}, y_{nn}) + W_2\gamma(U) + \Delta f_x \\ \varepsilon\dot{y} &= By_{nn} + W_3\sigma_2(x_{nn}, y_{nn}) + W_4\gamma(U) + \Delta f_y, \end{aligned} \tag{4.45}$$

where the model error and disturbances Δf_x , Δf_y, are still assumed to be constrained as $\|\Delta f_x\| \le \Delta\bar{f}_x, \|\Delta f_y\| \le \Delta\bar{f}_y$. Also $\Delta x, \Delta y$ and $W_{1,2,3,4}$ are bounded as well as other stability properties in Section 4.1.2.

Then we can reform (4.45)

$$\begin{aligned} \dot{x} &= Ax + W_1\sigma_1(x_{nn}, y_{nn}) + W_2\gamma(U) + d_x \\ \varepsilon\dot{y} &= By + W_3\sigma_2(x_{nn}, y_{nn}) + W_4\gamma(U) + d_y \end{aligned} \tag{4.46}$$

where $d_x = \Delta f_x + Ax_{nn} - Ax = \Delta f_x - A\Delta x, d_y = \Delta f_y + By_{nn} - By = \Delta f_y - B\Delta y$. If $\Delta f_x, \Delta f_y$ and $\Delta x, \Delta y$ are all bounded, d_x, d_y can summed to be bounded as well, like $\|d_x\| \le \bar{d}_x, \|d_y\| \le \bar{d}_y$.

The desired time-varying trajectory is defined as (3.21) in Chapter 3 with time-scale parameter embedded.

As the new structure of neural network consists of the state variables of neural network itself only, the overall structure of the neural networks identification and controller is shown in the following figure. The control law is independent on the actual signals from the real system.

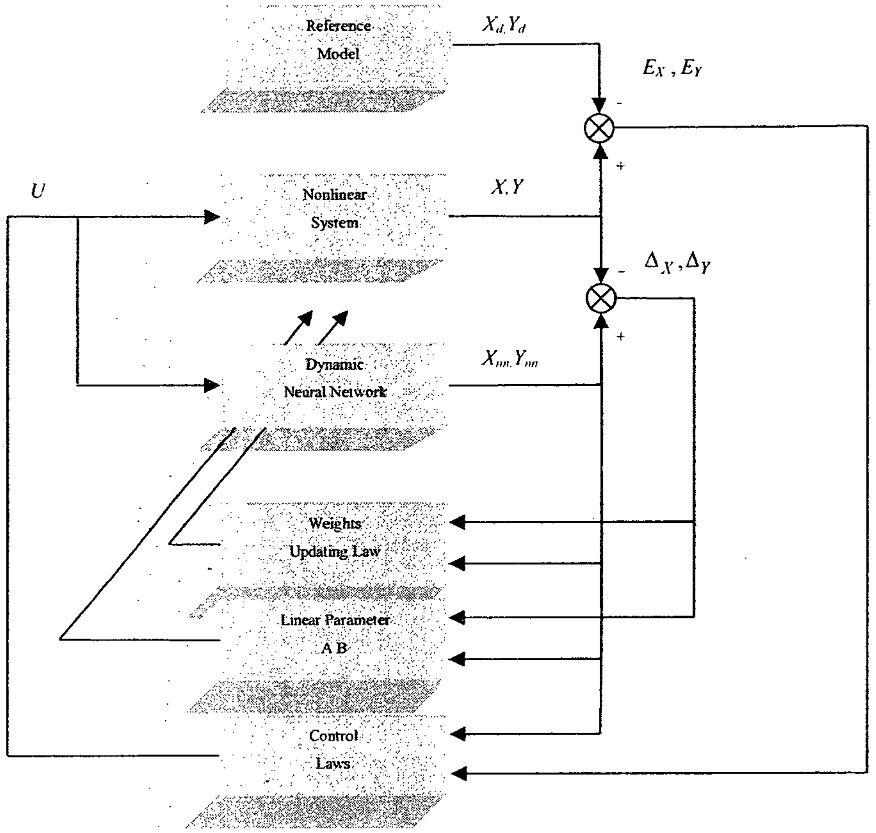


Figure 4-11 New Identification and control scheme

We define the state tracking error as

$$\begin{aligned} E_x &= x - x_d \\ E_y &= y - y_d . \end{aligned} \tag{4.47}$$

Then the error dynamic equations become:

$$\begin{aligned} \dot{E}_x &= Ax_{nn} + W_1\sigma_1(x_{nn}, y_{nn}) + W_2\gamma(U) + d_x - g_x \\ \varepsilon\dot{E}_y &= By_{nn} + W_3\sigma_2(x_{nn}, y_{nn}) + W_4\gamma(U) + d_y - g_y . \end{aligned} \tag{4.48}$$

If we consider the identification and control as a whole process, then we can apply the strategy to real applications by generating the final Lyapunov function candidate as

$V = V_I + V_C$. Since we had already proved $\dot{V}_I \le 0$ and the stability properties in Theorem 4.2. Now let's consider the stability analysis for tracking purpose.

4.2.2 Improved controller design

Again we the control action U is designed as

$$U = u_L + u_f, \tag{4.49}$$

where u_L is a compensation for the known nonlinearity and u_f are dedicated to deal with the model errors, which can be left open if it is zero or ignorable. Let u_L be

$$\begin{aligned} u_L &= \begin{bmatrix} W_2 \\ (1/\varepsilon)W_4 \end{bmatrix}^{-1} u'_L \\ u'_L &= -\begin{bmatrix} Ax_d \\ (1/\varepsilon)By_d \end{bmatrix} - \begin{bmatrix} W_1\sigma_1(x_{nn}, y_{nn}) \\ (1/\varepsilon)W_3\sigma_2(x_{nn}, y_{nn}) \end{bmatrix} + \begin{bmatrix} g_x \\ (1/\varepsilon)g_y \end{bmatrix} \end{aligned} \tag{4.50}$$

The control action u_f is to compensate the unknown dynamic modeling error. The sliding mode control methodology is applied to accomplish the task. So let u_f be,

$$u_f = \begin{bmatrix} W_2 \\ (1/\varepsilon)W_4 \end{bmatrix}^{-1} u'_f = \begin{bmatrix} W_2 \\ (1/\varepsilon)W_4 \end{bmatrix}^{-1} \begin{bmatrix} u'_{fx} \\ u'_{fy} \end{bmatrix} \tag{4.51}$$

First rewrite (4.48) as

$$\begin{bmatrix} \dot{E}_x \\ \dot{E}_y \end{bmatrix} = \begin{bmatrix} Ax_{nn} \\ (1/\varepsilon)By_{nn} \end{bmatrix} + \begin{bmatrix} W_1\sigma_1(x_{nn}, y_{nn}) \\ (1/\varepsilon)W_3\sigma_2(x_{nn}, y_{nn}) \end{bmatrix} + \begin{bmatrix} W_2 \\ (1/\varepsilon)W_4 \end{bmatrix} U + \begin{bmatrix} d_x \\ (1/\varepsilon)d_y \end{bmatrix} - \begin{bmatrix} g_x \\ (1/\varepsilon)g_y \end{bmatrix} \tag{4.52}$$

Then substituting (4.50) into (4.52) obtains

$$\begin{bmatrix} \dot{E}_x \\ \dot{E}_y \end{bmatrix} = \begin{bmatrix} AE_x \\ (1/\varepsilon)BE_y \end{bmatrix} + \begin{bmatrix} W_2 \\ (1/\varepsilon)W_4 \end{bmatrix} u_f + \begin{bmatrix} d_x \\ (1/\varepsilon)d_y \end{bmatrix} \tag{4.53}$$

If the model error and disturbances are zero or negligible which, from the control point of view, it won't devastate the stability of the system, u_f can be chosen to be zero which will lead the error dynamics converge to the origin. Proof is quite straightforward since A and B are stable matrices and ε is positive.

Then substituting (4.51) into (4.53) yields

$$\begin{aligned}\dot{E}_x &= AE_x + u'_{fx} + d_x \\ \dot{E}_y &= (1/\varepsilon)BE_y + u'_{fy} + (1/\varepsilon)d_y.\end{aligned} \tag{4.54}$$

a) Direct Compensation

If the identification process is stabilized as we proofed in Section 4.1.2, the modeling errors can be calculated as $\Delta f_x = \dot{x} - \dot{x}_{nn}, \Delta f_y = \dot{y} - \dot{y}_{nn}$. So if the derivatives of all real signals are available, we can compensated the dynamic modeling errors with

$$u'_f = \begin{bmatrix} u'_{fx} \\ u'_{fy} \end{bmatrix} = -\begin{bmatrix} d_x \\ (1/\varepsilon)d_y \end{bmatrix} = -\begin{bmatrix} \Delta f_x - A\Delta x \\ (1/\varepsilon)(\Delta f_y - B\Delta y) \end{bmatrix} = -\begin{bmatrix} \dot{x} - \dot{x}_{nn} - A\Delta x \\ (1/\varepsilon)(\dot{y} - \dot{y}_{nn} - B\Delta y) \end{bmatrix} \tag{4.55}$$

Theorem 4.3: With the control strategy (4.52), we can guarantee the control errors are globally asymptotically stable as $\lim_{t\to\infty} E_x = 0, \lim_{t\to\infty} E_y = 0$.

Proof: By using (4.54) and (4.55), the error dynamics become

$$\begin{aligned}\dot{E}_x &= AE_x \\ \dot{E}_y &= (1/\varepsilon)BE_y\end{aligned} \tag{4.56}$$

Hence, we have stability properties $\lim_{t\to\infty} E_x = 0, \lim_{t\to\infty} E_y = 0$.

The controller involves matrix inversion which can be guaranteed non-singularities by choosing the proper initial value of the updating law and the parameter of the activation function.

b) Sliding Mode Compensation

If the derivatives of real signals are not available, we can compensate the dynamic modeling errors with sliding mode technique with the assumption that d_x, d_y are assumed to be bounded as $\|d_x\| \leq \bar{d}_x, \|d_y\| \leq \bar{d}_y$.

$$u'_f = \begin{bmatrix} u'_{fx} \\ u'_{fy} \end{bmatrix} = \begin{bmatrix} -k_x P_x^{-1} \operatorname{sgn}(E_x) \\ -(1/\varepsilon) P_y^{-1} \operatorname{sgn}(E_y), \end{bmatrix} \tag{4.57}$$

Where $k_x > \lambda_{\max}(P_x)\bar{d}_x, k_y > \lambda_{\max}(P_y)\bar{d}_y$, P_x, P_y are the solutions of (4.58).

Since the matrices A, B are unknown nominal constant Hurwitz matrices, there definitely exist matrices P_x, P_y which can be chosen to satisfy the following equations, where Q_x, Q_y are positive definite symmetric matrices:

$$\begin{aligned} A^T P_x + P_x A &= -Q_x \\ B^T P_y + P_y B &= -Q_y. \end{aligned} \tag{4.58}$$

Theorem 4.4: With the control strategy (4.56), we can guarantee that the control errors are globally asymptotically stable as $\lim_{t \to \infty} E_x = 0, \lim_{t \to \infty} E_y = 0$.

Proof: Since we had already proved $\dot{V}_I \leq 0$ and the identification stability properties in Theorem 4.2. Now let's consider the Lyapunov function candidate for control design purpose:

$$V_c = E_x^T P_x E_x + E_y^T P_y E_y \tag{4.59}$$

By using (4.54) and (4.57), we obtain the derivative of Lyapunov candidate (4.59) as

$$\begin{aligned}
\dot{V}_c &= \dot{E}_x^T P_x E_x + E_x^T P_x \dot{E}_x + \dot{E}_y^T P_y E_y + E_y^T P_y \dot{E}_y \\
&= (E_x^T A^T + u_{fx}'^T + d_x^T) P_x E_x + E_x^T P_x (A E_x + u'_{fx} + d_x) \\
&\quad + (1/\varepsilon)(E_y^T B^T + \varepsilon u_{fy}'^T + d_y^T) P_y E_y + (1/\varepsilon)(B E_y + \varepsilon u'_{fy} + d_y) \\
&= E_x^T (A^T P_x + P_x A) E_x + 2 E_x^T P_x u'_{fx} + 2 E_x^T P_x d_x \\
&\quad + (1/\varepsilon) E_y^T (B^T P_y + P_y B) E_y + 2 E_y^T P_y u'_{fy} + 2(1/\varepsilon) E_y^T P_y d_y \\
&= -E_x^T Q_{xo} E_x - 2k_x |E_x| + 2 E_x^T P_x d_x - (1/\varepsilon) E_y^T Q_{yo} E_y - 2(1/\varepsilon) k_y |E_y| + 2(1/\varepsilon) E_y^T P_y d_y \\
&\le -E_x^T Q_{xo} E_x - 2k_x \|E_x\| + 2\lambda_{\max}(P_x) \|E_x\| \|d_x\| \\
&\quad - (1/\varepsilon) E_y^T Q_{yo} E_y - 2(1/\varepsilon) k_y \|E_y\| + 2(1/\varepsilon) \lambda_{\max}(P_y) \|E_y\| \|d_y\| \\
&\le -E_x^T Q_{xo} E_x - 2(k_x - \lambda_{\max}(P_x)\bar{d}_x) \|E_x\| - (1/\varepsilon) E_y^T Q_{yo} E_y - 2(1/\varepsilon)(k_y - \lambda_{\max}(P_y)\bar{d}_y) \|E_y\| \\
&\le 0
\end{aligned}$$

Hence, we have stability properties $\lim_{t \to \infty} E_x = 0, \lim_{t \to \infty} E_y = 0$.

c) Energy function Compensation

In this case, we use the assumption that the modeling errors are bounded. Then we define

$$u'_f = \begin{bmatrix} u'_{fx} \\ u'_{fy} \end{bmatrix} = \begin{bmatrix} -2R_x^{-1} P_x E_x \\ -2(1/\varepsilon) R_y^{-1} P_y E_y \end{bmatrix} \tag{4.60}$$

where $R_x = R_x^T > 0, R_y = R_y^T > 0$

Assumption 4.5: Since the matrices *A, B* are Hurwitz matrices. If we define $R_x = \Lambda_x^{-1}, R_y = \Lambda_y^{-1}$ one can select a proper Q_x, Q_y satisfying the conditions in Lemma 1, then there exist matrices P_x, P_y satisfying the following equations:

$$\begin{aligned}&A^T P_x + P_x A + P_x R_x P_x + Q_x = 0\\&B^T P_y + P_y B + P_y R_y P_y + Q_y = 0\end{aligned}\tag{4.61}$$

Theorem 4.5: With the control strategy (4.60) and assumption 4.5, we can guarantee the following stability properties:

$$\begin{aligned}&\|E_x\|_{Q_x}^2 + \|u'_{fx}\|_{R_x}^2 \le \|d_x\|_{\Lambda_x}^2 + \limsup_{\tau\to\infty}\frac{1}{\tau}\int_0^\tau \Psi(u'_{fx})dt\\&\|E_y\|_{Q_y}^2 + \|u'_{fy}\|_{R_y}^2 \le \frac{1}{\varepsilon}\|d_y\|_{\Lambda_y}^2 + \limsup_{\tau\to\infty}\frac{1}{\tau}\int_0^\tau \Psi(u'_{fy})dt\end{aligned}\tag{4.62}$$

where $\Psi(u'_{fx}) = 2E_x^T P_x u'_{fx} + u'^T_{fx} R_x u'_{fx}, \Psi(u'_{fy}) = 2E_y^T P_y u'_{fy} + (1/\varepsilon) u'^T_{fy} R_y u'_{fy}$ are defined as energy functions.

Proof: Since we had already proved $\dot{V}_I \le 0$ and the identification stability properties in **Theorem 4.2**. Now let's consider the Lyapunov function candidate for control purpose

$$\begin{aligned}&V_c = V_x + V_y\\&V_x = E_x^T P_x E_x, V_y = E_y^T P_y E_y\end{aligned}\tag{4.63}$$

By using (4.54) and (4.60), we obtain the derivatives of Lyapunov candidate (4.63) as

$$\begin{aligned}\dot{V}_x &= \dot{E}_x^T P_x E_x + E_x^T P_x \dot{E}_x\\&= (E_x^T A^T + u'^T_{fx} + d_x^T)P_x E_x + E_x^T P_x (A E_x + u'_{fx} + d_x)\\&= E_x^T (A^T P_x + P_x A)E_x + 2E_x^T P_x u'_{fx} + 2E_x^T P_x d_x\\\dot{V}_y &= \dot{E}_y^T P_y E_y + E_y^T P_y \dot{E}_y\\&= (1/\varepsilon)(E_y^T B^T + \varepsilon u'^T_{fy} + d_y^T)P_y E_y + (1/\varepsilon)(B E_y + \varepsilon u'_{fy} + d_y)\\&= (1/\varepsilon)E_y^T (B^T P_y + P_y B)E_y + 2E_y^T P_y u'_{fy} + 2(1/\varepsilon)E_y^T P_y d_y\end{aligned}\tag{4.64}$$

Using the matrix inequality (4.14) one obtains

$$\begin{aligned}2E_x^T P_x d_x &\le E_x^T P_x \Lambda_x^{-1} P_x E_x + d_x^T \Lambda_x d_x\\2(1/\varepsilon)E_y^T P_y d_y &\le (1/\varepsilon)\left(E_y^T P_y \Lambda_y^{-1} P_y E_y + d_y^T \Lambda_y d_y\right)\end{aligned}\tag{4.65}$$

Hence from (4.64) one obtains

$$\begin{aligned}
\dot{V}_x &\le E_x^T(A^T P_x + P_x A + P_x \Lambda_x^{-1} P_x + Q_x)E_x - E_x^T Q_x E_x + d_x^T \Lambda_x d_x + 2E_x^T P_x u'_{fx} \\
&= -E_x^T Q_x E_x - u_{fx}'^T R_x u'_{fx} + d_x^T \Lambda_x d_x + 2E_x^T P_x u'_{fx} + u_{fx}'^T R_x u'_{fx} \\
&= -E_x^T Q_x E_x - u_{fx}'^T R_x u'_{fx} + d_x^T \Lambda_x d_x + \Psi(u'_{fx}) \\
\dot{V}_y &\le (1/\varepsilon)E_y^T(B^T P_y + P_y B + P_y \Lambda_y^{-1} P_y + Q_y)E_y - (1/\varepsilon)E_y^T Q_y E_y + (1/\varepsilon)d_y^T \Lambda_y d_y + 2E_y^T P_y u'_{fy} \\
&= -(1/\varepsilon)E_y^T Q_y E_y - (1/\varepsilon)u_{fy}'^T R_y u'_{fy} + (1/\varepsilon)d_y^T \Lambda_y d_y + 2E_y^T P_y u'_{fy} + (1/\varepsilon)u_{fy}'^T R_y u'_{fy} \\
&= -(1/\varepsilon)E_y^T Q_y E_y - (1/\varepsilon)u_{fy}'^T R_y u'_{fy} + (1/\varepsilon)d_y^T \Lambda_y d_y + \Psi(u'_{fy})
\end{aligned}$$

$$(4.66)$$

We reformulate (4.66) as

$$\begin{aligned}
E_x^T Q_x E_x + u_{fx}'^T R_x u'_{fx} &\le d_x^T \Lambda_x d_x + \Psi(u'_{fx}) - \dot{V}_x \\
(1/\varepsilon)E_y^T Q_y E_y + (1/\varepsilon)u_{fy}'^T R_y u'_{fy} &\le (1/\varepsilon)d_y^T \Lambda_y d_y + \Psi(u'_{fy}) - \dot{V}_y
\end{aligned} \qquad (4.67)$$

Then integrating each term from 0 to τ, averaging them by τ and taking the limit of these integrals' upper bound, we obtain:

$$\begin{aligned}
&\|E_x\|_{Q_x}^2 + \|u'_{fx}\|_{R_x}^2 \le \|d_x\|_{\Lambda_x}^2 + \lim_{\tau\to\infty}\sup\frac{1}{\tau}\int_0^\tau \Psi(u'_{fx})dt + \lim_{\tau\to\infty}\sup(-\frac{1}{\tau}\int_0^\tau \dot{V}_x dt) \\
&\frac{1}{\varepsilon}\|E_y\|_{Q_y}^2 + \frac{1}{\varepsilon}\|u'_{fy}\|_{R_y}^2 \le \frac{1}{\varepsilon}\|d_y\|_{\Lambda_y}^2 + \lim_{\tau\to\infty}\sup\frac{1}{\tau}\int_0^\tau \Psi(u'_{fy})dt + \lim_{\tau\to\infty}\sup(-\frac{1}{\tau}\int_0^\tau \dot{V}_y dt)
\end{aligned} \qquad (4.68)$$

where

$$\begin{aligned}
\|E_x\|_{Q_x}^2 &= \lim_{\tau\to\infty}\sup\frac{1}{\tau}\int_0^\tau E_x^T Q_x E_x dt & \|E_y\|_{Q_y}^2 &= \lim_{\tau\to\infty}\sup\frac{1}{\tau}\int_0^\tau E_y^T Q_y E_{yx} dt \\
\|u'_{fx}\|_{R_x}^2 &= \lim_{\tau\to\infty}\sup\frac{1}{\tau}\int_0^\tau u_{fx}'^T R_x u'_{fx} dt & \|u'_{fy}\|_{R_y}^2 &= \lim_{\tau\to\infty}\sup\frac{1}{\tau}\int_0^\tau u_{fy}'^T R_y u'_{fy} dt \\
\|\Delta f_x\|_{\Lambda_x}^2 &= \lim_{\tau\to\infty}\sup\frac{1}{\tau}\int_0^\tau d_x^T \Lambda_x d_x dt & \|\Delta f_y\|_{\Lambda_y}^2 &= \lim_{\tau\to\infty}\sup\frac{1}{\tau}\int_0^\tau d_y^T \Lambda_y d_y dt
\end{aligned}$$

Considering the Lyapunov functions V_x, V_y are always positive, one can have

$$\begin{aligned}\limsup_{\tau\to\infty}\Big(-\frac{1}{\tau}\int_0^\tau \dot{V}_x dt\Big) &= \limsup_{\tau\to\infty}\left(-\frac{1}{\tau}V_x(\tau)+\frac{1}{\tau}V_x(0)\right) \le \limsup_{\tau\to\infty}\left(\frac{1}{\tau}V_x(0)\right)=0 \\ \limsup_{\tau\to\infty}\Big(-\frac{1}{\tau}\int_0^\tau \dot{V}_y dt\Big) &= \limsup_{\tau\to\infty}\left(-\frac{1}{\tau}V_y(\tau)+\frac{1}{\tau}V_y(0)\right) \le \limsup_{\tau\to\infty}\left(\frac{1}{\tau}V_y(0)\right)=0\end{aligned} \tag{4.69}$$

Hence, we have stability properties (4.62). Then the right sides of (4.62) decide the threshold of the trajectory tracking errors. Now the task is to minimize the energy function

$$\Psi(u'_{fx}) = 2E_x^T P_x u'_{fx} + u'^T_{fx} R_x u'_{fx}, \Psi(u'_{fy}) = 2E_y^T P_y u'_{fy} + (1/\varepsilon) u'^T_{fy} R_y u'_{fy}$$

If we chose

$$\begin{aligned} u'_{fx} &= -2R_x^{-1} P_x E_x \\ u'_{fy} &= -2(1/\varepsilon) R_y^{-1} P_y E_y \end{aligned} \tag{4.70}$$

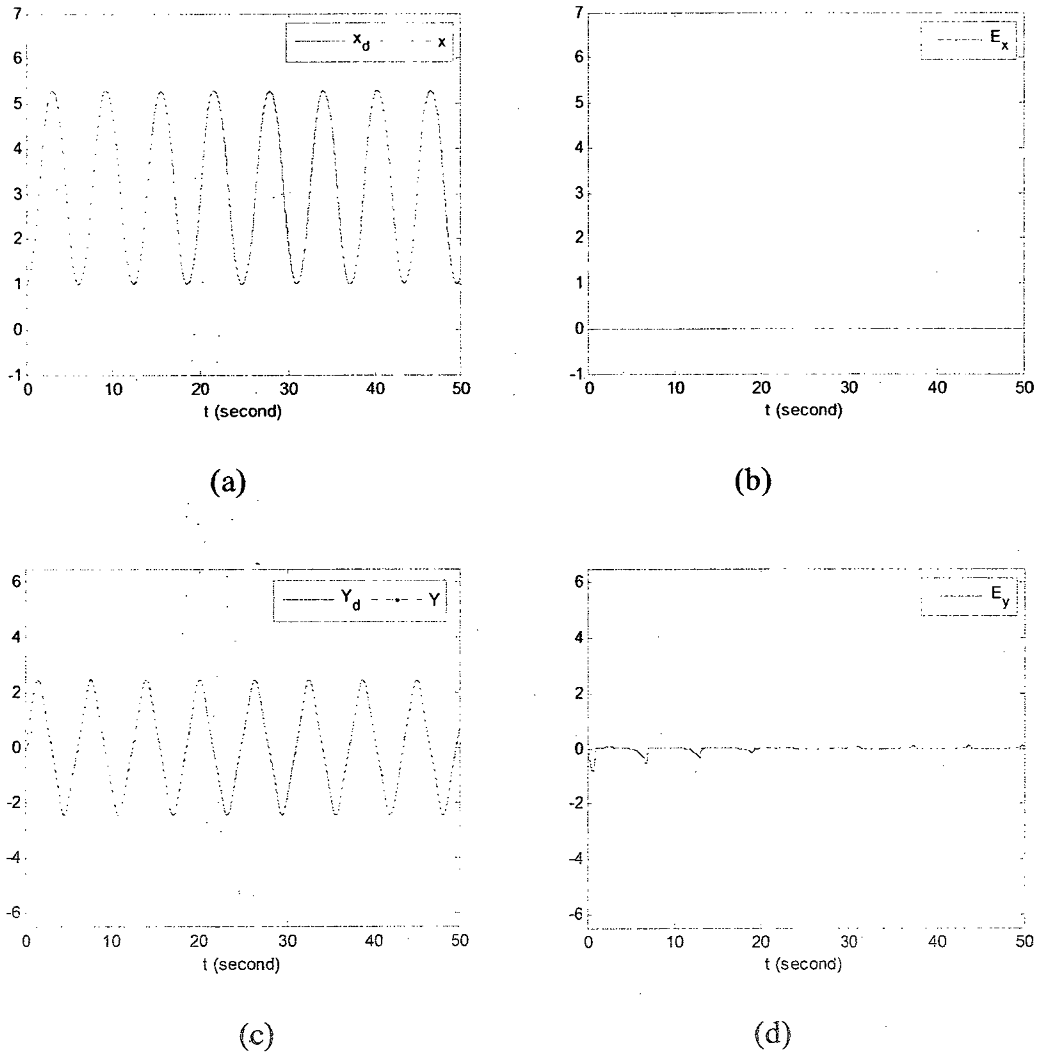
The energy function stays at zero.

4.2.3 Simulation result

Following the identification process in Section 4.1 for nonlinear system (3.16), we implement the developed control laws. It constitutes a feedback linearization with sliding mode controller. The desired trajectories are generated by the reference model

$$\begin{aligned} \dot{x}_d &= y_d \\ \varepsilon \dot{y}_d &= \sin x_d, \end{aligned} \tag{4.70}$$

with the initial value $x_d(0) = 1$, $y_d(0) = 0$.



(a)

(b)

(c)

(d)

Figure 4-12 Trajectory tracking results using direct compensation

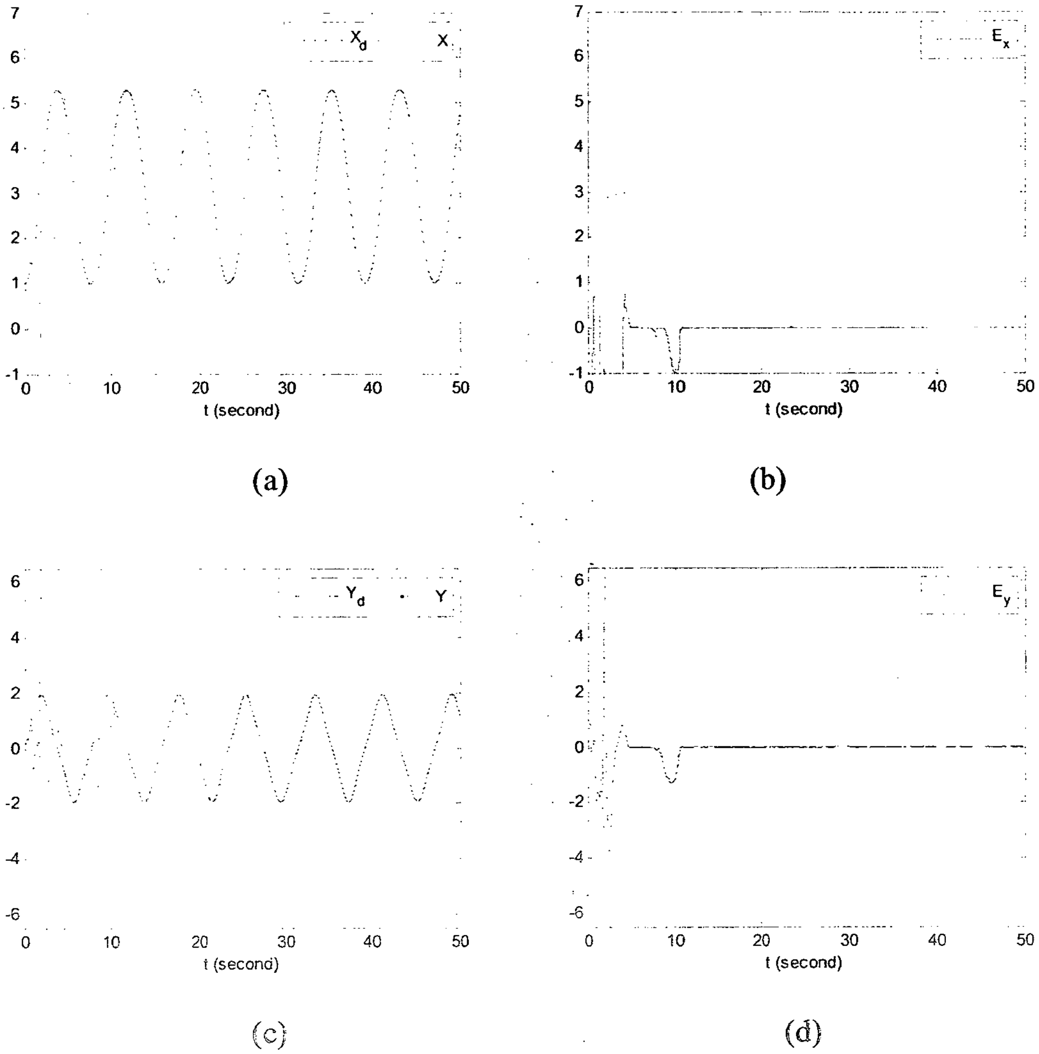
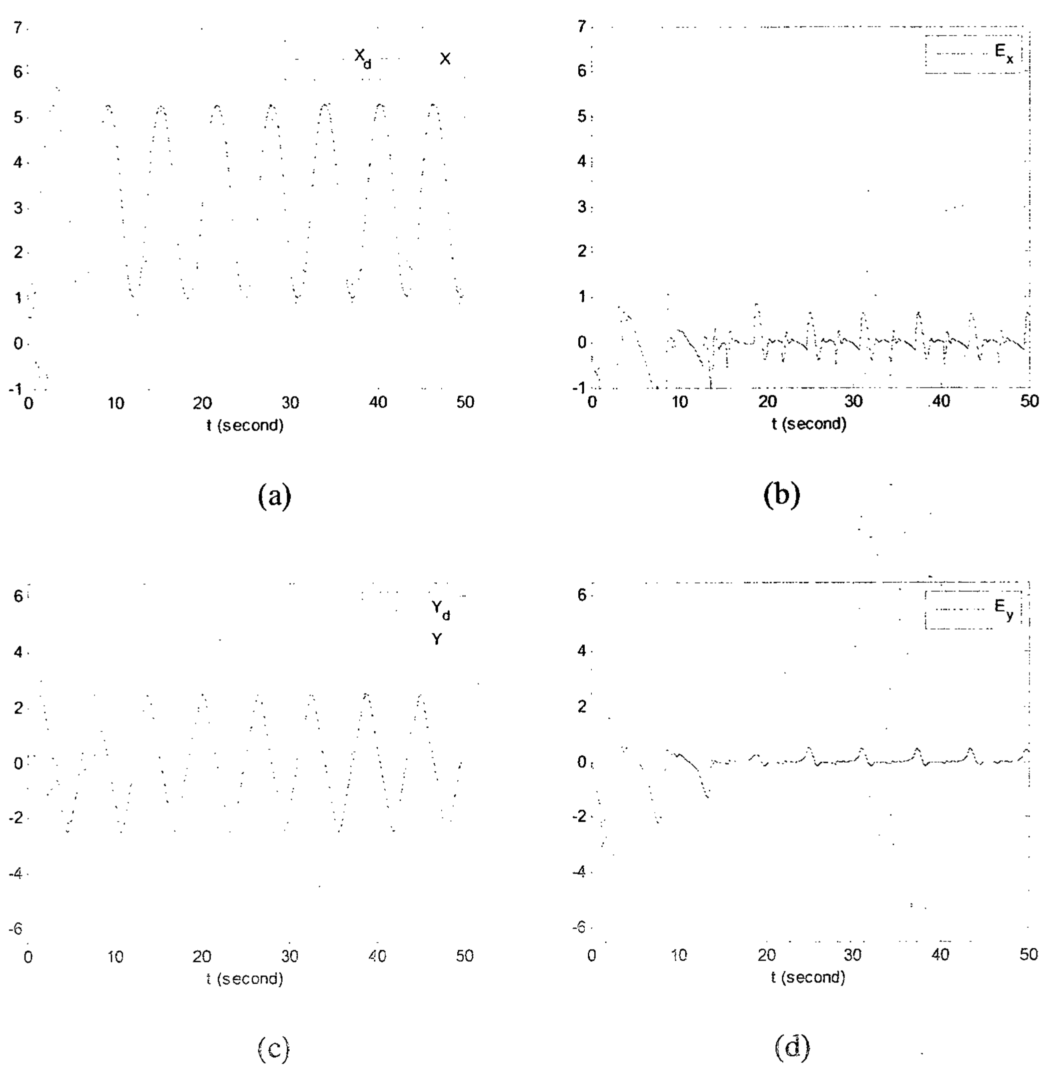


Figure 4-13 Trajectory tracking results using Sliding Mode Compensation



(a)

(b)

(c)

(d)

Figure 4-14 Trajectory tracking results using energy function compensation

The time scale is considered by putting ε = 0.5. From Figures 4.12-4.14, we can see that the states of the nonlinear system can track the desired trajectories for the three different control methods. Although the first method needs full information about the derivative of the real signals, the tracking performance is the best according to our theoretical analysis, since it directly compensate the modeling errors. The other methods have extensive disturbance at beginning, but the tracking errors will be adjusted to stable or under the threshold. Therefore, we can draw the conclusion that the proposed identification and control algorithm can guarantee the tracking stability of nonlinear and uncertain dynamic systems.

To compare three methods, we list the RMS information in the following table.

Table 4-2 RMS for Control Strategies

	X	Y
Direct Compensation	0.000895	0.102215
Sliding Mode	1.361007	0.879036
Energy Function	0.997788	0.678837

4.3 Conclusion

In this chapter we propose a new on-line identification algorithm for both dynamic neural networks weights and the linear part matrices for nonlinear systems with multi-time scales. New structure of the dynamic neural network simplifies the identification and control schemes. Then we propose three different control methods based on dynamic multiple time scales neural networks. The controller consists of a feedback linearization and one of three classical control methods such as direct, sliding mode or energy function compensator to deal with the unknown identification error and disturbance. Simulation results show the effectiveness of the proposed identification and control algorithms.

Chapter 5 Conclusion and future work

In this study, an in-depth research has been carried out on the identification algorithm and controller design for nonlinear singularly perturbed system by using dynamic multi-time scale neural network.

The main contributions of this research are summarized as:

1) An on-line identification algorithm is proposed for dynamic neural networks with different time scales. Updating laws are developed for both the linear part and weights of dynamic neural network.
2) Then tracking problem based on the identification results is investigated. Feedback linearization method is utilized with additional sliding mode controller in case of modelling error presented.
3) The dead-zoon functions are design with the updating algorithm to prevent from parameter drifting. The stability of the on-line identification algorithm is proved by using Lyapunov function analysis for the modified structure of the dynamic neural works which result in simplified the identification scheme.
4) The controller design consists of a feedback linearization and one of three classical control methods such as direct, sliding mode or energy function compensator to deal with the unknown identification error and disturbance.

Simulation results are compared with the existing works, which reveals that our algorithm achieves better identification performance. The eigenvalues for linear part matrix are all negative which supports the stability of the neural networks. In addition, the dynamic neural networks successfully follow the complicated electro-physic phenomena the singularly perturbed HH system. The simulation results demonstrate the fast and accurate convergent property of the proposed on-line identification algorithms. Three control strategies are applied to the system based on the identification results. The simulation results show that the controller can make the system satisfy the tracking performance.

Possible future works are list as follows:

1) The vector function of the plant does not depend explicitly on time t, which makes the system (2.2) to be autonomous. In this paper, we only consider this kind of system. Future work can focus on dealing with nonautonomous system.
2) For the black-box models in this thesis, all the signals of stated variables are assumed to be directly observable. If some of them are not available, observer technique could be used.
3) For system identification, the input singles are also assumed to be available. How can we identify a nonlinear system that operates in closed-loop and is stabilized by an unknown regulator?
4) The structure of the neural network is pre-selected for general black box problem. Evolutionary algorithm could be combined to achieve simpler optimal structure and faster calculation speed.

References

[1] S. Haykin, "Neural Networks: A Comprehensive Foundation", NJ, Prentice-Hall, 1999.

[2] Phil Picton, "Neural Networks", New York, PALGRAVE, 2000.

[3] Tommy W S Chow, Siu-Yeung Cho, "Neural Networks and Comuting: Learning Algorithms and Applications", Imperial College Press, 2007.

[4] Canelon, J.I., L.S.Shieh, et al, "A new approach for the neural control of nonlinear discrete dynamic systems", Information Sciences, 2004.

[5] Hornik, K., M. Stinchcombe, et al, "Multilayer feedforward networks are universal approximators", Neural Networks 2, pp. 359-366, 1989.

[6] Seul Jung, Hyun-Taek Cho and T. C. Hsia, "Neural Network Control for Position Tracking of a Two-Axis Inverted pendulum System: Experimental Studies", Neural Networks, Vol. 18 No. 4, 2007.

[7] Paul J. Werbos, "The Roots of Backpropagation", Wiley-Interscience, 1994.

[8] Wen Yu and Xiaoou Li, "Some New Results on system Identification with Dynamic Neural Networks", IEEE Trans. on Neural Networks, Vol. 12, No. 2, 2001.

[9] Alexander S. Poznyak, Wen Yu, Edgar N. Sanchez and Jose P. Perez, "Nonlinear Adaptive trajectory Tracking Using Dynamic Neural Networks", IEEE Trans. on Neural Networks, Vol. 10, No. 6, 1999.

[10] Geotge A. Rovithakis and Manolis A. Christodoulou, "Direct Adaptive Regulation of Unknown Nonlinear Dynamical Systems via Dynamic Neural Networks", IEEE Trans. on systems, Man, and Cybernetics, Vol.25, No. 12, 1995.

[11] Geotge A. Rovithakis and Manolis A. Christodoulou, "Adaptive Control of Unknown Plants Using Dynamical Neural Networks", IEEE Trans. on systems, Man, and Cybernetics, Vol.24, No. 3, 1994.

[12] Rebecca Suckley and Vadim N. Biktashev, "The asymptotic structure of the Hodgkin-Huxley equations", International Journal of Bifurcation and Chaos (IJBC) 2, Vol. 13, Issue. 12, pp. 3805–3825, 2003.

[13] Hassan K. Khall, "Nonlinear systems", 3rd ed. Prentice Hall, pp. 323, Inc 2002.

[14] K. J. Hunt, D. Sbarbaro, R. Zbikowski, and P. J. Gawthrop, "Neural networks for control systems—A survey", Automatica, vol. 28, Issue. 6, pp. 1083–1112, 1992.

[15] D.H. Rao, M.M. Gupta and H.C. Wood, "Neural networks in control systems", Proceeding of IEEE Conference on Communications, Computers and Power in the Modern Environment, pp. 282–290, 1993.

[16] E. D. Sontag and Y. Wang, "On characterization of the input-to-state stability property", System and Control Letters, Vol. 24, pp. 351–359, 1995.

[17] Eugene M. Izhikevich, "Dynamical Systems in Neuroscience: The geometry of excitability and bursting", The Neurosciences Institute: The MIT Press 2005.

[18] Boris T. Polyak, "Introduction to Optimization", Optimization Software, Inc., New York, 1987.

[19] W. Yu and A.S. Poznyak, "Indirect Adaptive Control via Parallel Dynamic Neural Networks", IEE Proceeding of Control Theory and Applications, Vol.146, No.1, pp. 25-30, 1999.

[20] Ioannou, P.A., and Sun, J., "Robust Adaptive Control", Prentice-Hall, Upper Saddle River, NJ, 1996.

[21] Meyer Base, A. Ohl. F. and Scheich, H, "Singular perturbation analysis of competitive neural networks with different time-scales", Neural Computation 8, pp. 545–563, 1996.

[22] Anke Meyer Base, Frank Ohl, Henning Scheich, "Quadratic-Type Lyapunov functions for competitive neural networks with different time-Scales", IEEE Trans. on Automatic Control, Vol. 29, Issue 6, pp. 542 – 550, 1984.

[23] Anke Meyer, "Flow invariance for competitive neural networks with different timescales"; International Joint Conference on Neural Networks, Vol.1, pp. 858-861, Honolulu, Hawaii, 2002.

[24] A. Meyer Baese, S. S. Pilyugin, and Y. Chen, "Global exponential stability of competitive neural networks with different time Scales", IEEE Trans. on Neural Networks, Vol. 14, No. 3, May 2003.

[25] Meyer Base, A., Pilyugin, S.S. and Wismuller, A, "Stability analysis of a self-organizing neural network with feedforward and feedback dynamics", Proceedings of IEEE International Joint Conference on Neural Networks, Vol. 2, pp. 1505-1509, Budapest Hungary, July 2004.

[26] Meyer Base, A., Ohl, F. and Scheich, H., "Stability analysis techniques for competitive neural networks with different time-scales", Proceedings of IEEE International Conference on Neural Networks, Vol. 6, pp. 3215–3219, Perth, WA, Australia, 27 Nov.-1 Dec. 1995.

[27] K. S. Narendra and K. Parthasarathv, "Identification and control of dynamical systems using neural networks", IEEE Trans. on Neural Networks, Vol. 1, No. 1, pp. 4-27, 1990.

[28] Alejandro Cruz Sandoval, Wen Yu and Xiaoou Li, "Some stability properties of dynamic neural networks with different time-scales", International Joint Conference on Neural Networks, pp. 4218-4224, 2006.

[29] Wen Yu and Xiaoou Li, "Passivity analysis of dynamic neural networks with different time-scales", Neural Processing Letters, Vol. 25, Issue 2, pp. 143–155, 2007.

[30] George A. Rovithakis, and Manolis A. Christodoulou, "Regulation of unknown nonlinear dynamical systems via dynamical neural networks", Journal of Intelligent and Robotic Systems, Vol. 12, No. 3, pp. 259-275, 1995.

[31] Alexander S. Poznyak, Edgar N.Sanchez, Jose P.Perez and Wen Yu, "Robust adaptive nonlinear system identification and trajectory tracking by dynamic neural networks", Processings of the American Control Conference, Vol. 1, pp. 242-246, 1997.

[32] L. A. Zadeh, "On the identification problem", IRE Trans. on Circuit Theory, pp. 277–281, 1956.

[33] L. A. Zadeh, "From circuit theory to system theory", IRE Trans. on Circuit Theory, pp. 856-865, 1962.

[34] Monopoli, R., "Model reference adaptive control with an augmented error signal", IEEE Trans. on Automatic Control, Vol.19, No.5, pp. 474- 484, Oct 1974.

[35] Daniel M. Rovner and Robert H. Cannon, Jr, "Experiments Toward On-Line Identification and Control of a Very Flexible One-Link Manipulator", The International Journal of Robotics Research, Vol. 6, No. 4, pp. 3-19, 1987.

[36] Ho, B. and R. Kalman, "Effective construction of linear state-variable models from input /output functions", REGELUNGSTECHNIK. Vol. 14, No. 12, pp. 545-548, 1966.

[37] Masaru Hoshiya and Etsuro Saito, "Structural Identification by Extended Kalman Filter", Journal of Engineering Mechanics, Vol. 110, No. 12, pp. 1757-1770, 1984.

[38] Lennart Ljung, "system identification-Theory for the user", second edition, Prentice Hall, 1999.

[39] Valdivia, V., Barrado, A., Laazaro, A., Zumel, P., Raga, C. and Fernandez, C., "Simple Modeling and Identification Procedures for Black-Box Behavioral Modeling of Power Converters Based on Transient Response Analysis", IEEE Trans. on Power Electronics, Vol.24, No.12, pp. 2776-2790, Dec. 2009.

[40] Coutlis, A., Limebeer, D.J.N., Wainwright, J.P., Lister, J.B. and Vyas, P., "Frequency response identification of the dynamics of a Tokamak plasma", IEEE Trans. on Control Systems Technology, Vol.8, No.4, pp. 646-659, Jul 2000.

[41] Mohd-Mokhtar, R., Wang, L., Qin, L. and Barry, T., "Continuous time system identification of magnetic bearing systems using frequency response data", 5th Asian Control Conference, Vol. 3, pp. 2066- 2072, July 2004.

[42] Singh, Y. P. and Subramanian, S., "Frequency-response identification of structure of nonlinear systems", IEE Proceedings of Control Theory and Applications, Vol. 127, No. 3, pp. 77-82, 1980.

[43] Guillaume, P., Kollar, I., Pintelon, R., "Statistical analysis of nonparametric transfer function estimates", IEEE Trans. on Instrumentation and Measurement, Vol. 45, No. 2, pp. 594-600, April 1996.

[44] Tugnait, J.K., "Identification of multivariable stochastic linear systems via spectral analysis given time-domain data", IEEE Trans. on Signal Processing, Vol. 46, No. 5, pp. 1458-1463, May 1998.

[45] K.J. Astrom and P. Eykhoff, "System Identification - A Survey", Automatica, Pergamon Press, Vol. 7, pp. 123-162, 1971.

[46] Overschee, P. V. and B. D. Moor, "N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems", Automatica 30, pp. 75-93, 1994.

[47] Jean-Marc Le Caillec and Rene Garello, "Nonlinear system identification using autoregressive quadratic models", Signal Processing, Vol. 81, Issue 2, pp. 357-379, 2001.

[48] Fard, R. D., Karrari, M. and Malik, O. P., "Synchronous generator model identification for control application using volterra series", IEEE Trans. on Energy Conversion, Vol. 20, No. 4, pp. 852- 858, Dec. 2005.

[49] Feng Ding and Tongwen Chen, "Identification of multivariable systems based on finite impulse response models with flexible orders", IEEE International Conference on Mechatronics and Automation, Vol. 2, No. 29, pp. 770-775, July-1 Aug. 2005.

[50] Masry, E. and Tjostheim, D., "Nonlinear autoregressive exogenous time series: structural identification via projection estimates", IEEE Signal Processing Workshop on Statistical Signal and Array Processing, pp. 368-370, Jun 1996.

[51] Sheng Lu, Ki Hwan Ju, Chon, K.H., "A new algorithm for linear and nonlinear ARMA model parameter estimation using affine geometry [and application to blood flow/pressure data]", IEEE Trans. on Biomedical Engineering, Vol. 48, No. 10, pp. 1116-1124, Oct. 2001.

[52] Hong-Tzer Yang, Chao-Ming Huang and Ching-Lien Huang, "Identification of ARMAX model for short term load forecasting: an evolutionary programming approach", IEEE Trans. on Power Systems, Vol. 11, No. 1, pp. 403-408, Feb 1996.

[53] Monin, A., "ARMAX identification via hereditary algorithm", IEEE Trans. on Automatic Control, Vol. 49, No. 2, pp. 233- 238, Feb. 2004.

[54] Triolo, R.J., Nash, D.H. and Moskowitz, G. D., "The identification of time series models of lower extremity EMG for the control of prostheses using Box-Jenkins criteria", IEEE Trans. on Biomedical Engineering, Vol. 35, No. 8, pp. 584-594, Aug. 1988.

[55] Fang Yangwang and Jiao Licheng, "Blind identification of nonlinear FIR Volterra channels", Proceedings of WCCC-ICSP 5th International Conference on Signal Processing, Vol. 1, pp. 294-297, 2000.

[56] Palanthandalam-Madapusi, H.J., Ridley, A.J. and Bernstein, D.S., "Identification and prediction of ionospheric dynamics using a Hammerstein-Wiener model with radial basis functions", Proceedings of the American Control Conference, Vol. 7, pp. 5052-5057, June 2005.

[57] Atherton, D.P., "Early developments in nonlinear control", Proceedings of the 33rd IEEE Conference on Decision and Control, Vol. 3, pp. 2106-2111, Dec 1994.

[58] Chinmoy Pal, Ichiro Hagiwara, Naoki Kayaba and Shin Morishita, "Dynamic System Identification by Neural Network: A New, Fast Learning Method Based on Error Back Propagation", Journal of Intelligent Material Systems and Structures, Vol. 5, No. 1, pp. 127-135, 1994.

[59] Michael C. Nechyba and Yangsheng Xu, "Neural Network Approach to Control System Identification with Variable Activation Functions", Proceeding of IEEE International Symposium on Intelligent Control, pp. 358-363, August 1994.

[60] Srinivasan, B., Prasad, U.R. and Rao, N.J., "Back propagation through adjoints for the identification of nonlinear dynamic systems using recurrent neural models", IEEE Trans. on Neural Networks, Vol.5, No.2, pp. 213-228, Mar 1994.

[61] Yue-Seng Goh and Eng-Chong Tan, "An integrated approach to improving back-propagation neural networks", Proceedings of IEEE Ninth International Conference on Frontiers of Computer Technology, Vol. 2, pp. 801-804, Aug 1994.

[62] Wei Gao, "Evolutionary Neural Network Based on New Ant Colony Algorithm", International Symposium on Computational Intelligence and Design, Vol.1, pp. 318-321, Oct. 2008.

[63] Jian Fang and Yugeng Xi, "Neural network design based on evolutionary programming", Artificial Intelligence in Engineering, Vol. 11, Issue 2, pp. 155-161, 1997.

[64] Razi, A., Menhaj, M.B. and Mohebbi, A., "Intelligent position control of earth station antennas with backlash compensation based on MLP neural network", Innovative Technologies in Intelligent Systems and Industrial Applications, pp. 265-270, July 2009.

[65] Jung-Wook Park, Harley, R.G. and Venayagamoorthy, G.K., "Comparison of MLP and RBF neural networks using deviation signals for on-line identification of a synchronous generator", IEEE Power Engineering Society Winter Meeting, Vol.1, pp. 274- 279, 2002.

[66] Mark J. L. Orr, "Introduction to Radial Basis Function Networks", Centre for Cognitive Science, University of Edinburgh, 2, Buccleuch Place, Edinburgh EH8 9LW, Scotland, April 1996.

[67] P. Venkatesan and S. Anitha, "Application of a radial basis function neural network for diagnosis of diabetes mellitus", Current Science, Vol. 91, No. 9, 10 November 2006.

[68] Akpan, V.A. and Hassapis, G., "Adaptive predictive control using recurrent neural network identification", Mediterranean Conference on Control and Automation, pp. 61-66, June 2009.

[69] Chih-Min Lin and Chun-Fei Hsu, "Recurrent-neural-network-based adaptive-backstepping control for induction servomotors", IEEE Trans. on Industrial Electronics, Vol.52, No.6, pp. 1677- 1684, Dec. 2005.

[70] Barbu, M., Caraman, S. and Ceanga, E., "Bioprocess Control Using a Recurrent Neural Network Model", Proceedings of IEEE International Symposium on Intelligent Control, pp. 479-484, 27-29 June 2005.

[71] Delgado, A., Kambhampati, C. and Warwick, K., "Dynamic recurrent neural network for system identification and control", IEE Proceedings of Control Theory and Applications, Vol.142, No.4, pp. 307-314, Jul 1995.

[72] Gutierrez, L.B., Lewis, F.L. and Lowe, J.A., "Implementation of a neural network tracking controller for a single flexible link: comparison with PD and PID controllers", IEEE Trans. on Industrial Electronics, Vol. 45, No. 2, pp. 307-318, Apr 1998.

[73] Seul Jung and Sung Su Kim, "Control Experiment of a Wheel-Driven Mobile Inverted Pendulum Using Neural Network", IEEE Trans. on Control Systems Technology, Vol. 16, No. 2, pp. 297-303, March 2008.

[74] Seul Jung, Hyun-Taek Cho and Hsia, T.C., "Neural Network Control for Position Tracking of a Two-Axis Inverted Pendulum System: Experimental Studies", IEEE Trans. on Neural Networks, Vol.18, No.4, pp. 1042-1048, July 2007.

[75] Noh, J.S., Lee, G.H., Jung, S., "Position control of a mobile inverted pendulum system using radial basis function network", IEEE International Joint Conference on Neural Networks, pp. 370-376, June 2008.

[76] Xie W. F., Zhu Y.Q., Zhao Z.Y. and Wong Y.K., "Nonlinear System Identification using Optimized Dynamic Neural Network", Vol. 72, Issue 13-15, pp. 3277-3287, 2009.

[77] D. Moreau1, M. Ariola, E. Bouvier, V. Cordoliani, L. Laborde, D. Mazon, T. Tala and JET EFDA contributors, "Identification of the dynamic plasma response for integrated profile control in advanced scenarios on JET", 33rd EPS Conference on Plasma Physics, Vol. 301, pp. 1069, Rome, June 2006.

[78] Jian Niu, Jun Zhao, Zuhua Xu and Jixin Qian, "A Two-Time Scale Decentralized Model Predictive Controller Based on Input and Output Model", Journal of Automated Methods and Management in Chemistry, pp. 11, 2009.

[79] P. V. Kokotovi´c, "Applications of singular perturbation techniques to control problems", SIAM Review 26, pp. 501–550, 1984.

[80] V.R. Saksena, J. O'Reilly and P.V. Kokotovic, "Singular perturbations and time-scale methods in control theory: Survey 1976-1983", Automatica, Vol. 20, Issue 3, pp. 273-293, May 1984.

[81] D. Subbaram Naidu, "Singular Perturbations And Time Scales In Control Theory And Applications: An Overview", Dynamics of Continuous, Discrete and Impulsive Systems, Series B: Applications & Algorithms 9, pp. 233-278, 2002.

[82] Weiying Cheng, Wen, J.T.-Y, "A two-time-scale neural controller for the tracking control of rigid manipulators", IEEE Trans. on Systems, Man and Cybernetics, Vol. 24, No.7, pp. 991-1000, Jul 1994.

[83] Lin, J., Lewis, F.L., "Fuzzy controller for flexible-link robot arm by reduced-order techniques", IEE Proceedings of Control Theory and Applications, Vol. 149, No.3, pp. 177-187, May 2002.

[84] Han X. and Xie W. F., "Nonlinear Systems Identification using Dynamic Multi-Time Scales Neural Networks", Proceedings of the 4th Annual IEEE Conference on Automation Science and Engineering, (CASE 2008), August 23-26, 2008, Key Bridge Marriott, Washington DC, USA.

[85] Han X., and Xie W.F., "Nonlinear Systems Identification and Control using Dynamic Multi-Time Scales Neural Networks", Proceedings of 2009 IEEE International Conference on Automation and Logistics, Shenyang, China, August 5-8, 2009.

[86] Han X., Xie W.F. and Mei R.X., "On-line Nonlinear Systems Identification via Dynamic Neural Networks with Multi-Time Scales", submitted to 49th IEEE Conference on Decision and Control, 2010.