Proceedings of the 7th Mediterranean Conference on Control and Automation (MED99) Haifa, Israel - June 28-30, 1999

PARAMETER IDENTIFICATION IN NONLINEAR SYSTEMS USING HOPFIELD NEURAL NETWORKS

Zhenning Hu[†] S.N. Balakrishnan[‡]
Department of Aerospace and Mechanical Engineering
and Engineering Mechanics
University of Missouri-Rolla
Rolla, MO 65401

ABSTRACT

In this study, we use the Hopfield neural network (HNN) for identifying parameters of a nonlinear system. We use a linearization process and develop the equations for a parameter identification algorithm. We use a scalar time varying problem and a complex nine-state nonlinear problem to demonstrate the potential of this method.

1 INTRODUCTION

Parameter estimation plays a crucial role in the field of control engineering. If the underlying physics does not lend itself to derive equations of motion (for example by using Newton's second law) easily, then parameter estimates are crucial in arriving at a model for the process to be studied. Even where equations of motion are readily derived, accurate estimation of the parameters associated with the process is crucial for the design of a control law. Another area where parameter identification is useful is in the area of post flight trajectory analysis; a more urgent need arises in a damaged process or aircraft where quick and accurate estimates of system parameters can mean the difference between recovery and total loss. Consequently, there has been and continues to be a lot of studies and development of new methods in the area of parameter estimation.

The capability of artificial neural networks (ANN) to model the behavior of large classes of uncertain nonlinear dynamical systems within a certain accuracy has made it very popular recently in the areas of signal processing, pattern recognition, system identification and optimal control. Neural networks have a natural advantage over other methods for online calculations in the sense that they are massively parallel in their processing structure and therefore, take less computation time. Thanks to Hopfield's distinguished work (Hopfield 1982, 1984, 1985), there has been a multitude of papers using recurrent neural networks for linear system solvers, control (Cetinkunt, 1993; Shen, 1997) and pattern recognition (Jagannathan, 1996; Bruyne, 1998; Habib, 1996; Shen, 1997). While feedforward networks are static mapping between two information domains, the structure of recurrent neural networks incorporates dynamical behavior through feedback connections. Because of the feedback, the input to the plant gets connected with the output of the plant. Many researcher and scientists have begun to use this new tool in their fields. Cichocki and Unbehauen's extensive and thorough research of linear systems (1992, No.2) concentrates on algebraic equations. They present several network configurations and compare their speeds in finding inverses and solutions to linear systems of

[†] Graduate Student

[‡] Professor and Contact Author, E-mail: bala@umr.edu

algebraic equations. Raol applies recurrent neural networks to linear and time invariant dynamical systems (1994, No.6; 1995, No.2; 1996, No.4). He has developed a system of equations for parameter identification using several recurrent networks. Our work is similar to Raol's, however, we present proofs of convergence and boundedness and extend the application to highly nonlinear problems. Lyashevskiy uses his network mapping strategy (1997; 1998) to identify parameters in a nonlinear system that can be written into a linear form. Amin, Gerhart and Rodin have special insight in Hopfield neural networks and network structure. Based on that, they proposed a new recurrent high-order neural networks (1997). They developed Lyapunov based theorems to show that convergence of their higher-order recurrent networks, which are more complex than the networks used in our study. A lot of network structures and their convergence or robustness analysis are also being done by other researchers, such as Kim and Liews (1997, No.8), Kambhampati (1998), Stubberud (1991). All these show their novel ideas about recurrent networks, but they are more complex to realize.

In this study, we use Hopfield recurrent neural network for online estimation of parameters. There are a few papers in the literature that shows this kind neural network's potential for parameter identification. The focus of this study is to develop an algorithm to deal with nonlinear systems, which are more common engineering processes and demonstrate its effectiveness through applications. We present nonlinear examples of stable and unstable systems where our methods yield accurate estimates. We also discuss the working mechanism of the network with respect to its parameters in achieving convergence.

2 HOPFIELD NEURAL NETWORK STRUCTURE

We present the dynamics of the Hopfield neural networks in this section. The dynamics of the network are defined by the following system of first-order differential equation

$$c_j \frac{du_j}{dt} = -g_j u_j(t) + \sum_{k=1}^{n} w_{jk} v_k(t) + i_j \tag{1}$$

where

$$v_j(t) = f_j(u_j(t)) \tag{2}$$

It is assumed that the nonlinear function $f(\cdot)$ relating the output $v_j(t)$ of neuron to its activation potential $u_j(t)$ is a continuous function and therefore differentiable. We also assume the inverse of the nonlinear activation function exists, so that one can write

$$u_j(t) = f_j^{-1}(v_j(t)) \tag{3}$$

According to (Hopfield, 1982, 1984, 1985), an energy (Lyapunov) function of this recurrent neural network is identified as

$$E = -\frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} w_{jk} v_k v_j + \sum_{j=1}^{n} g_j \int_0^T f_j^{-1}(v_j) dv_j - \sum_{j=1}^{n} i_j v_j \tag{4}$$

The differential equation E with respect to time, one can get

$$\begin{aligned} \frac{dE}{dt} &= -\sum_{j=1}^{n}\left(\sum_{k=1}^{n} w_{jk} v_k - g_j u_j + i_j\right)\frac{dv_j}{dt} = -\sum_{j=1}^{n} c_j \left(\frac{du_j}{dt}\right)\frac{dv_j}{dt} \\ &= -\sum_{j=1}^{n} c_j \left[\frac{d}{dt} f_j^{-1}(v_j)\right]\frac{dv_j}{dt} = \sum_{j=1}^{n} c_j \left[\frac{d}{dv_j} f_j^{-1}(v_j)\right]\left(\frac{dv_j}{dt}\right)^2 \end{aligned} \tag{5}$$

For log sigmoid, tangent sigmoid and linear activation function,

$$\frac{d}{dv_j} f_j^{-1}(v_j) \geq 0 \text{ for all } v_j(t) \tag{6}$$

Thus, for the energy function *E* defined above, it has

$$\frac{dE}{dt} < 0 \text{ for } v_j \neq 0 \tag{7}$$

From Lyapunov theory, we know that:

- This function *E* is a Lyapunov function of the Hopfield neural network (HNN)
- The model is stable in accordance with the Lyapunov's theorem

3 PARAMETER IDENTIFICATION OF DYNAMIC SYSTEMS

This section describe a general nonlinear dynamic system and then use a linearization method to linearize it. Based on Hopfield Neural Network theory, we compute the network's weights and biases. Boundedness and convergence are also given.

3.1 DYNAMIC SYSTEM DESCRIPTION

Consider a nonlinear dynamic system, which can be described in state space form as

$$\dot{x} = F(A,x) + Bu \tag{8}$$

where x is a nx1 vector, u is a px1 control input, and *F(A,x)* is a certain kind of nonlinear function that can be described as the following forms,

$$\begin{aligned} F(A,x) &= \begin{bmatrix} f_1(a_1,x) \\ f_2(a_2,x) \\ \vdots \\ f_n(a_n,x) \end{bmatrix} = \begin{bmatrix} f_{11}(a_{11},x)+\cdots+f_{1m}(a_{1m},x) \\ f_{21}(a_{21},x)+\cdots+f_{2m}(a_{2m},x) \\ \vdots \\ f_{n1}(a_{n1},x)+\cdots+f_{nm}(a_{nm},x) \end{bmatrix} \\ &= \begin{bmatrix} f_{11}(a_{11}h_{11}(x))+\cdots+f_{1m}(a_{1m}h_{1m}(x)) \\ f_{21}(a_{21}h_{21}(x))+\cdots+f_{2m}(a_{2m}h_{2m}(x)) \\ \vdots \\ f_{n1}(a_{n1}h_{n1}(x))+\cdots+f_{nm}(a_{nm}h_{nm}(x)) \end{bmatrix} \end{aligned} \tag{9}$$

and

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}_{(n\times m)} \tag{10}$$

$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{bmatrix}_{(n\times p)} \tag{11}$$

where $f_i(\bullet,\bullet)$ is nonlinear in *x*.

In *F(A, x)*, each row can have different number of term, like $m_1, m_2, \cdots, m_n$. One can pick the longest row, which has m_k terms. Make other rows also have m_k terms by adding zeros at the end. Here for simplicity but without losing generality, we suppose they all have equal number (m) of terms. To identify *A* and *B*, which are matrices of parameters associated with the system, the key point is to get the parameters out of every term in every row and make them coefficients of the terms in valued. In order to realize this transformation, we need to linearize the given nonlinear dynamic system.

3.2 NONLINEAR SYSTEM LINEARIZATION

Linear systems have been studied very extensively, and are always relatively easy to deal with. To identify parameters in system which contains nonlinear terms, like Eq.(8), we are going to change each nonlinear element $f_{ij}(a_{ij}, x)$ in *F(A, x)* into a linear form. However, the original system should satisfy some criteria in order to be able to use some linear representations. The following definitions and theorem are assumed to hold with respect to the nonlinear systems under study.

Definition 1 (Vidyasagar, 1993) *For an autonomous system,* $\dot{x} = f(x)$, $f(0) = 0$ *(i.e. x=0 is an equilibrium), f is continuously differentiable. Let* $A = \left.\frac{\partial f}{\partial x}\right|_{x=0}$ *(i. e. let A denote the Jacobian matrix of f evaluated at x=0), R(x)=f(x)-Ax, then if it turns out that* $\lim_{\|x\|=0} \frac{\|R(x)\|}{\|x\|} = 0$ *i.e. the Taylor series expansion of f(x) =Ax + R(x), the system* $\dot{z} = Az$ *is called the linearization of the nonlinear system around the equilibrium x=0.*

Definition 2 (Vidyasagar, 1993) *Given the non-autonomous system,*

$$\dot{x}(t) = f[t, x(t)] \tag{12}$$

Suppose that

$$f(t,0) = 0 \quad \forall t \geq 0 \tag{13}$$

and that f is a C^1 *function. Define*

$$A \equiv \left[\frac{\partial f(t,x)}{\partial x}\right]_{x=0} \tag{14}$$

$$f_1(t,x) \equiv f(t,x) - A(t)x \tag{15}$$

Then by the definition of the Jacobian, it follows that for fixed t≥ 0, if it is true that

$$\lim_{\|x\|=0} \sup_{t\geq 0} \frac{\|f_1(t,x)\|}{\|x\|} = 0 \tag{16}$$

then the system

$$\dot{z} = A(t)z(t) \tag{17}$$

is called the linearization or linearized system of (12) *around the origin.*

Theorem (Vidyasagar, 1993) *Consider the system* (12). *Suppose that* (13) *holds and that* $f(\bullet)$ *is continuously differentiable. Define A(t),* $f_1(t,x)$ *as in* (14), (15), *respectively, and assume that (1) Eq.*(16) *holds, and (2)* $A(\bullet)$ *is bounded. Under these conditions, if 0 is an exponentially stable equilibrium of the linear system* $\dot{z} = A(t)z(t)$, *then it is also an exponentially stable equilibrium of the system* (12).

Note, above theorems and definitions are all based on the equilibrium being at the origin. However, it is only for convenience and can be relaxed. Note that we can follow the proof of this theorem outlined in (Vidyasagar, 1993), to extend its validity to where the equilibrium need not be the origin.

Thus, based on the definitions and the theorem, we can have

$$f_{ij}(a_{ij}h_{ij}(x)) = f_{ij}(a_{ij}h_{ij}(x^*)) + f_{ij}^{'}a_{ij}h_{ij}^{'}(x^*)\cdot(x-x^*) + \Delta(x-x^*)^2 \tag{18}$$

Since $f_{ij}(a_{ij}h_{ij}(x^*)) = 0$ (x^* is the equilibrium point)

And $\Delta(x-x^*) \to 0$ as x is in the neighborhood of x^*. Then,

$$f_{ij}(a_{ij}h_{ij}(x)) \approx f_{ij}^{'}(a_{ij}h_{ij}(x^*))a_{ij}h_{ij}^{'}(x^*)\cdot(x-x^*) + \Delta(x-x^*)^2 \equiv a_{ij}g_{ij}(x) \tag{19}$$

Where $g_{ij}(x) = f_{ij}^{'}(a_{ij}h_{ij}(x^*))h_{ij}^{'}(x^*)\cdot(x-x^*)$

So, the whole nonlinear vector becomes

$$F(A,x) = \begin{bmatrix} a_{11}g_{11}(x) + a_{12}g_{12}(x) + \cdots + a_{1m}g_{1m}(x) \\ a_{21}g_{21}(x) + a_{22}g_{22}(x) + \cdots + a_{2m}g_{2m}(x) \\ \vdots \\ a_{n1}g_{n1}(x) + a_{n2}g_{n2}(x) + \cdots + a_{nm}g_{nm}(x) \end{bmatrix} \tag{20}$$

Each row in *F(A,x)* is now a sum of linear terms in *x*. Note, that all the unknown parameters appear as coefficients of the terms in *F(A,x)*. Now, we will relate *F(A,x)* to the energy function and weights of the neural networks.

3.3 COMPUTATION OF WEIGHTS AND BIASES

The error dynamics between the plant and the model with unknown parameters are given by

$$e(A,B,x) = \dot{x} - F_s(A,x) - B_s u \tag{21}$$

The subscript "s" denotes the system containing estimated parameters. The energy function of the neural network is defined as

$$E(A,B,x) = \frac{1}{2T}\int_0^T e^T e \quad dt = \frac{1}{2T}\int_0^T \left[\dot{x} - F_s(A,x) - B_s u\right]^T \cdot \left[\dot{x} - F_s(A,x) - B_s u\right] \; dt \tag{22}$$

where *T* is time period during which data are collected.

The equilibrium point for the energy function occurs when the partial derivatives $\partial E/\partial A_s$, $\partial E/\partial B_s$ are zero. The derivatives of the energy function *E* with respect to parameters a_{ij} and b_{ij} are given by

$$\begin{aligned} \frac{\partial E}{\partial a_{ij}} &= \frac{1}{T}\int_0^T \left[\dot{x}_i - (a_{i1}g_{i1} + a_{i2}g_{i2} + \cdots + a_{im}g_{im}) - \sum_{k=1}^{p} b_{ik}u_k\right]\cdot[-g_{ij}] \; dt \\ &= \frac{1}{T}\int_0^T \left[a_{i1}g_{i1}g_{ij} + a_{i2}g_{i2}g_{ij} + \cdots + a_{im}g_{im}g_{ij} + \sum_{k=1}^{p} b_{ik}u_k g_{ij} - \dot{x}_i g_{ij}\right] \; dt \end{aligned} \tag{23}$$

$$\begin{aligned} \frac{\partial E}{\partial b_{ij}} &= \frac{1}{T}\int_0^T \left[\dot{x}_i - (a_{i1}g_{i1} + a_{i2}g_{i2} + \cdots + a_{im}g_{im}) - \sum_{k=1}^{p} b_{ik}u_k\right]\cdot[-u_k] \; dt \\ &= \frac{1}{T}\int_0^T \left[a_{i1}g_{i1}u_k + a_{i2}g_{i2}u_k + \cdots + a_{im}g_{im}u_k + \sum_{k=1}^{p} b_{ik}{u_k}^2 - \dot{x}_i u_k\right] \; dt \end{aligned} \tag{24}$$

If we define A_i , B_i to represent *i*th row of *A* and *B* respectively, and *V* as vector consisting of columns of matrices *A* and *B*, then we get

$$\begin{aligned} V &= \left[A_1^T, A_2^T, \cdots, A_n^T, B_1^T, B_2^T, \cdots, B_n^T\right]^T \\ &= \left[a_{11}, \cdots a_{1m}, \cdots, a_{n1}, \cdots a_{nm}, b_{11}, \cdots b_{1p}, \cdots b_{n1}, \cdots b_{np}\right]^T \end{aligned} \tag{25}$$

We can rewrite Eq.(23) and (24) in terms of the elements of V, like (Raol, 1996) as

$$\text{For } 1 \le s \le mn, \quad \frac{\partial E}{\partial v_s} = \frac{\partial E}{\partial a_{ij}}$$

$$\frac{\partial E}{\partial v_s} = \sum_{k=1}^{m} v_{(i-1)m+k}(t) \frac{1}{T}\int_0^T g_{ik}(x) g_{ij}(x) dt + \sum_{k=1}^{p} v_{mn+(i-1)p+k}(t) \frac{1}{T}\int_0^T u_k g_{ij}(x) dt - \frac{1}{T}\int_0^T \dot{x}_i g_{ij}(x) dt \tag{26}$$

$$\text{For } mn+1 \le s \le mn+np, \quad \frac{\partial E}{\partial v_s} = \frac{\partial E}{\partial b_{ij}}$$

$$\frac{\partial E}{\partial v_s} = \sum_{k=1}^{m} v_{(i-1)m+k}(t) \frac{1}{T}\int_0^T g_{ik}(x) u_j dt + \sum_{k=1}^{p} v_{mn+(i-1)p+k}(t) \frac{1}{T}\int_0^T u_k u_j dt - \frac{1}{T}\int_0^T \dot{x}_i u_j dt \tag{27}$$

Now we can relate the parameter identification formulations in Eq.(26) and (27) in terms of weights of the Hopfield neural network, as

$$\frac{dE}{dV_s} = \sum_{r=1}^{mn+np} W_{sr} V_r + I_{sr} \tag{28}$$

where W_{sr} are the weights of Hopfield neural network to be used in parameter identification; I_{sr} represents the biases in the neural network.

For $1 \le s \le mn$

$$\begin{aligned} W_{sr} &= \frac{1}{T}\int_0^T g_{ik}(x) g_{ij}(x) dt \qquad 1 \le k \le m, \quad r = (i-1)m + k \\ &= \frac{1}{T}\int_0^T u_k g_{ij}(x) dt \qquad 1 \le k \le p, \quad r = mn + (i-1)p + k \end{aligned} \tag{29}$$

$$I_{sr} = -\frac{1}{T}\int_0^T \dot{x}_i g_{ij}(x) dt \tag{30}$$

For $mn+1 \le s \le mn+np$

$$\begin{aligned} W_{sr} &= \frac{1}{T}\int_0^T g_{ik}(x) u_j dt \qquad 1 \le k \le m, \quad r = (i-1)m + k \\ &= \frac{1}{T}\int_0^T u_k u_j dt \qquad 1 \le k \le p, \quad r = mn + (i-1)p + k \end{aligned} \tag{31}$$

$$I_{sr} = -\frac{1}{T}\int_0^T \dot{x}_i u_j dt \tag{32}$$

For a more compact representation, define

$$G_j = \left[g_{j1}, g_{j2}, \cdots, g_{jm}\right]^T \tag{33}$$

Note that, in terms of G_j, the linearized model representation becomes

$$\dot{x}_j = A_j^T G_j + B_j^T u \tag{34}$$

We can write the parameter identification formulation as

$$\frac{dE}{dV} = WV + I \tag{35}$$

where W (weight) and I (bias) are set as following,

$$W = \frac{1}{T}\int_0^T \begin{bmatrix} \left(G_1 G_1^T\right) & 0 & \cdots & 0 & \left(G_1 u^T\right) & 0 & \cdots & 0 \\ 0 & \left(G_2 G_2^T\right) & \cdots & 0 & 0 & \left(G_2 u^T\right) & \cdots & 0 \\ & \vdots & & \vdots & & \vdots & & \vdots \\ 0 & 0 & \cdots & \left(G_n G_n^T\right) & 0 & 0 & \cdots & \left(G_n u^T\right) \\ \left(u G_1^T\right) & 0 & \cdots & 0 & \left(u u^T\right) & 0 & \cdots & 0 \\ 0 & \left(u G_2^T\right) & \cdots & 0 & 0 & \left(u u^T\right) & \cdots & 0 \\ & \vdots & & \vdots & & \vdots & & \vdots \\ 0 & 0 & \cdots & \left(u G_n^T\right) & 0 & 0 & \cdots & \left(u u^T\right) \end{bmatrix}_{(mn+np)\times(mn+np)} dt \tag{36}$$

$$I = -\frac{1}{T}\int_0^T \left[\left(\dot{x}_1 G_1^T\right) \quad \left(\dot{x}_2 G_2^T\right) \quad \cdots \quad \left(\dot{x}_n G_n^T\right) \quad \left(\dot{x}_1 u^T\right) \quad \left(\dot{x}_2 u^T\right) \quad \cdots \quad \left(\dot{x}_n u^T\right)\right]^T_{(mn+np)\times 1} dt \tag{37}$$

Now, we will relate this formulation to the dynamics of a Hopfield neural network. The network dynamics can be written in the following form (Cichocki, 1992)

$$\frac{dV}{dt} = -\mu \frac{dE}{dV} = -\mu(WV + I) \tag{38}$$

In order to find the expression for μ, we choose tangent sigmoid as our nonlinear activation function and rewrite Eq. (5) to get

$$\frac{dv_j}{dt} = -\frac{1}{\left[\frac{d}{dt} f^{-1}(v_j)\right]}\frac{1}{c_j}\frac{dE}{dt} = -\frac{1}{\left[\frac{d}{dv_j} f^{-1}(v_j)\right]}\frac{1}{c_j}\frac{dE}{dv_j} \tag{39}$$

where
$$v_j = f(u_j) = \rho\frac{1-e^{-\lambda_j u_j}}{1+e^{-\lambda_j u_j}} \tag{40}$$

From Eq.(39), we get

$$\frac{d}{dv_j} f^{-1}(v_j) = \frac{2\rho}{\lambda_i\left(\rho^2 - v_j^2\right)} \tag{41}$$

By using Eq.(40) into Eq.(38), we get

$$\frac{dv_j}{dt} = -\frac{\lambda_i\left(\rho^2 - v_j^2\right)}{2\rho c_j}\frac{dE}{dv_j} \tag{42}$$

Comparing Eq. (41) with Eq. (37), we observe that

$$\mu_j = -\frac{\lambda_i\left(\rho^2 - v_j^2\right)}{2\rho c_j} \tag{43}$$

and $\mu = \operatorname{diag}\left[\mu_1 \quad \mu_2 \quad \cdots \quad \mu_{mn+np}\right]$.

3.4 BOUNDEDNESS ANALYSIS

In this section, we outline a proof of boundedness of estimates at each step. Discretization of Eq.(42) and using Eq.(23), (24) and (34) in parameter v_i, assuming a small time step t and $t+\Delta t$, we have

$$v_j(t+\Delta t) - v_j(t) = \frac{\lambda_i\left(\rho^2 - v_j^2\right)}{2\rho c_j T}\left[\left(\dot{x}_i - A_i^T G_i - B_i^T u\right)\cdot\left(-g_{ij}\right)\right]\cdot \Delta t \tag{44}$$

If v_i belongs to b_{ij}, g_{ij} will be changed to u_j, we get

$$v_j(t+\Delta t) - v_j(t) = \frac{\lambda_i\left(\rho^2 - v_j^2\right)}{2\rho c_j T}\left[\left(\dot{x}_i - A_i^T G_i - B_i^T u\right)\cdot\left(-u_j\right)\right]\cdot \Delta t \tag{45}$$

Since $\left\|A_i(t)\right\| < \infty$ and $\dfrac{\left\|F_i(A,x) - A_i^T G_i\right\|}{\|x\|} \to 0$ as $\|x\| \to x^*$ (From linearization theorem)

Then $\dot{x}_i - A_i^T G_i - B_i^T u = o(G_i^T G_i)$ a second-order term.

Since

$$\rho >> v_j \quad \left(v_j = f(u_j) = \rho\frac{1-e^{-\lambda_j u_j}}{1+e^{-\lambda_j u_j}}, \quad \text{where} \quad f \to \pm p \quad \text{as} \quad u_j \to \pm\infty \right) \quad \text{and} \quad \lambda_i << \rho,$$

(Usually the parameter ρ is picked as a high value but can be different for each row.)

$\mu_j = -\dfrac{\lambda_i\left(\rho^2 - v_j^2\right)}{2\rho c_j}$ is bounded; Since $g_{ij}(x)$ is bounded at each x, and u_j, the control a dither signal is bounded. This implies that

$\Delta v_i(\Delta t) = \left\|v_i(t+\Delta t) - v_i(t)\right\|$ is bounded.

3.5 CONVERGENCE ANALYSIS

In this section, we present a proof convergence of modeled parameters to their true values. Note, that this proof is applicable for both time-invariant and time-varying parameters. From Eq.(20), we rewrite the nonlinear function as

$$F_s(A,x) = \begin{bmatrix} A_1^T G_1 \\ A_2^T G_2 \\ \vdots \\ A_n^T G_n \end{bmatrix} = \begin{bmatrix} \left(A_1^T\right)_{1\times m} & 0 & \cdots & 0 \\ 0 & \left(A_2^T\right)_{1\times m} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \left(A_n^T\right)_{1\times m} \end{bmatrix}_{n\times(m\times n)} \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_n \end{bmatrix}_{(m\times n)\times 1} \tag{46}$$

$$\equiv A_s G_s$$

omitting the argument x with G_s, and where

$$A_s = \begin{bmatrix} \left(A_1^T\right)_{1\times m} & 0 & \cdots & 0 \\ 0 & \left(A_2^T\right)_{1\times m} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \left(A_n^T\right)_{1\times m} \end{bmatrix}_{n\times(m\times n)} \quad \text{and} \quad G_s = \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_n \end{bmatrix}_{(m\times n)\times 1}$$

then, the cost function is

$$E = \frac{1}{2T}\int_0^T \left(\dot{x} - A_s G_s - B_s u\right)^T \left(\dot{x} - A_s G_s - B_s u\right) dt \tag{47}$$

The right hand side of Eq.(46) can be expanded as

$$\begin{aligned} E = \frac{1}{2T}\int_0^T (&G_s^T A_s^T A_s G_s + u^T B_s^T u + G_s^T A_s^T B_s u + u^T B_s^T A_s G_s x \\ &- \dot{x} A_s G_s - G_s^T A_s^T \dot{x} - \dot{x}^T B_s u - u^T B_s^T \dot{x} + \dot{x}^T \dot{x}) dt \end{aligned} \tag{48}$$

In order to help with derivations further, we introduce trace operations:

$$\left(A^T B^T C D\right)_{1\times 1} = tr\left(BAD^T C^T\right)_{1\times 1} \tag{49}$$

By observing that the left part of Eq.(49) is a scalar, we can write another equation

$$\left(D^T C^T B A\right)_{1\times 1} = tr\left(BAD^T C^T\right)_{1\times 1} \tag{50}$$

Also it is easy to see from Eq.(49) and (50)

$$\left(A^T B^T D\right)_{1\times 1} = tr\left(BAD^T\right)_{1\times 1} \tag{51}$$

$$\left(D^T B A\right)_{1\times 1} = tr\left(BAD^T\right)_{1\times 1} \tag{52}$$

Thus, Eq.(48) will be rearranged using results produced by Eq.(49-52) as

$$\begin{aligned} E = {}& tr\left(A_s \left(\frac{1}{2T}\int_0^T G_s G_s^T dt \right) A_s^T \right) + tr\left(B_s \left(\frac{1}{2T}\int_0^T u u^T dt \right) B_s^T \right) \\ &+ tr\left(A_s \left(\frac{1}{2T}\int_0^T G_s u^T dt \right) B_s^T \right) - tr\left(A_s \left(\frac{1}{2T}\int_0^T G_s \dot{x}^T dt \right) \right) \\ &- tr\left(B_s \left(\frac{1}{2T}\int_0^T u \dot{x}^T dt \right) \right) + \frac{1}{2T}\int_0^T \dot{x}^T \dot{x} dt \end{aligned} \tag{53}$$

Two other important trace operations are given by

$$\frac{\partial}{\partial A} tr\left(ABA^T\right) = 2AB \tag{54}$$

$$\frac{\partial}{\partial A} tr(ABD) = D^T B^T \tag{55}$$

By using Eq.(54) and (55), and with $\dot{x} = F_p(A, x) + B_p u = A_p G_s + B_p u$ we can get the following equations:

$$\frac{\partial}{\partial A_s} tr\left(A_s \left(\frac{1}{2T}\int_0^T G_s G_s^T dt \right) A_s^T \right) = A_s \left(\frac{1}{T}\int_0^T G_s G_s^T dt \right) \tag{56}$$

$$\frac{\partial}{\partial A_s} tr\left(B_s \left(\frac{1}{2T}\int_0^T u u^T dt \right) B_s^T \right) = 0 \tag{57}$$

$$\frac{\partial}{\partial A_s} tr\left(A_s \left(\frac{1}{T}\int_0^T G_s u^T dt \right) B_s^T \right) = B_s \left(\frac{1}{T}\int_0^T u G_s^T dt \right) \tag{58}$$

$$\frac{\partial}{\partial A_s} tr\left(A_s \left(\frac{1}{T}\int_0^T G_s \left(A_p G_s + B_p u\right) dt \right) \right) = \frac{1}{T}\int_0^T \left(A_p G_s + B_p u\right) G_s^T dt \tag{59}$$

$$\frac{\partial}{\partial A_s} tr\left(B_s \left(\frac{1}{T}\int_0^T u \dot{x}^T dt \right) \right) = 0 \tag{60}$$

$$\frac{\partial}{\partial A_s} tr\left(\frac{1}{T}\int_0^T \dot{x}^T \dot{x} dt\right) = 0 \tag{61}$$

So, the derivatives of cost function Eq.(53) with respect to parameter matrix A_s and B_s are

$$\frac{\partial E}{\partial A_s} = \left(A_s - A_p\right)\left(\frac{1}{T}\int_0^T G_s G_s^T dt\right) + \left(B_s - B_p\right)\left(\frac{1}{T}\int_0^T u G_s^T dt\right) \tag{62}$$

Similarly,

$$\frac{\partial E}{\partial B_s} = \left(A_s - A_p\right)\left(\frac{1}{T}\int_0^T G_s u^T dt\right) + \left(B_s - B_p\right)\left(\frac{1}{T}\int_0^T u u^T dt\right) \tag{63}$$

Combine Eq.(61) and (62) and letting them to be zero for minimum error,

$$\begin{bmatrix} A_s - A_p & B_s - B_p \end{bmatrix} \begin{bmatrix} \frac{1}{T}\int_0^T G_s G_s^T dt & \frac{1}{T}\int_0^T G_s u^T dt \\ \frac{1}{T}\int_0^T u G_s^T dt & \frac{1}{T}\int_0^T u u^T dt \end{bmatrix} = 0 \tag{64}$$

Or

$$\begin{bmatrix} A_s - A_p & B_s - B_p \end{bmatrix} \cdot \frac{1}{T}\int_0^T \begin{bmatrix} G_s G_s^T & G_s u^T \\ u G_s^T & u u^T \end{bmatrix} dt = 0 \tag{65}$$

Or

$$\begin{bmatrix} A_s - A_p & B_s - B_p \end{bmatrix} \cdot \frac{1}{T}\int_0^T \begin{bmatrix} G_s \\ u \end{bmatrix} \begin{bmatrix} G_s^T & u^T \end{bmatrix} dt = 0 \tag{66}$$

From Eq.(65), we find so long as

$$\frac{1}{T}\int_0^T \begin{bmatrix} G_s \\ u \end{bmatrix} \begin{bmatrix} G_s^T & u^T \end{bmatrix} dt \neq 0$$

$A_s \to A_p$ and $B_s \to B_p$ asymptotically.

Hence, the proof.

4 SIMULATION RESULTS

In this section, we present two numerical examples, which demonstrate the potential of the Hopfield neural network based parameter estimation method.

4.1 CASE 1

This case is a scalar example, where the dynamics is nonlinear and the time-varying parameter is embedded inside the nonlinearity.

$$\dot{x} = \sin(ax) + bu \tag{67}$$

where $a = t - 5, \quad b = -8$

To identify parameter a in this nonlinear dynamic system, the linearization process is used. Since its equilibrium is 0, its linearization form is

$$\dot{x} = ax + bu \tag{68}$$

Then, we need to check whether Eq.(66) can be linearized to Eq.(68). Since

$$\lim_{\|x\| \to 0} \frac{\left\|\sin(ax) - ax\right\|}{\|x\|} = 0 \tag{69}$$

Referencing to Definition 1, it can be concluded that Eq.(68) is indeed a linearization of Eq.(66). To calculate the relevant weight and bias, we get, by defining the identified parameters as $V = [a,b]^T$,

$$W = \begin{bmatrix} x^2 & xu \\ ux & u^2 \end{bmatrix} \tag{70}$$

$$I = \begin{bmatrix} \dot{x}x & \dot{x}u \end{bmatrix}^T \tag{71}$$

By using the scheme given by section 3, we can compute the values of parameter *a*, *b* and the state trajectory. These are shown in Figures 1 to 3. Both of the parameters *a* and *b* show similar convergence.

4.2 CASE 2

Now consider a more difficult practical nonlinear higher dimensional problem in aircraft dynamics as shown (Lyshevski, 1998). We apply this nonlinear identification concept developed in section 3.2 and 3.3, to identify the unknown parameters of a twin-tail supercritical swept wing aircraft. The relevant dynamics are described by a set of nonlinear differential equations.

$$\dot{x}(t) = \begin{bmatrix} \dot{v} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \\ \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = Ax + Bu + F(x) \tag{72}$$

$$= A\begin{bmatrix} v \\ \alpha \\ q \\ \theta \\ \beta \\ p \\ r \\ \phi \\ \psi \end{bmatrix} + B\begin{bmatrix} \delta_{HR} \\ \delta_{HL} \\ \delta_{FR} \\ \delta_{FL} \\ \delta_{C} \\ \delta_{R} \end{bmatrix} + \begin{bmatrix} 0 \\ -p\cos\alpha\tan\beta - r\sin\alpha\tan\beta \\ c_{31}pr + c_{32}(r^2 - p^2) \\ q\cos\phi - r\sin\phi \\ p\sin\alpha - r\cos\alpha \\ c_{61}qp + c_{62}qr \\ c_{71}qp - c_{72}qr \\ q\tan\theta\sin\phi + r\tan\theta\cos\phi \\ q\cos^{-1}\theta\sin\phi + r\cos^{-1}\theta\cos\phi \end{bmatrix}$$

where

$$A = \begin{bmatrix} -0.009 & 0.53 & -0.24 & -9.8 & -0.46 & -0.095 & -0.14 & 0 & 0 \\ -0.001 & -0.68 & 1 & 0 & 0.12 & 0.037 & 0.005 & 0 & 0 \\ 0.0002 & 2.7 & -0.53 & 0 & 0.009 & 0.0062 & 0.028 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.001 & 0.69 & 0.42 & 0.18 & -0.72 & 0.086 & -0.15 & 0 & 0 \\ 0.00002 & 1.1 & 0.041 & 0.007 & -26 & -4.9 & 0.53 & 0 & 0 \\ 0.00001 & -1.7 & 0.098 & 0.011 & 7.4 & -0.037 & 0.82 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.093 & 0.093 & 0.045 & -0.045 & -0.07 & -0.13 \\ -0.28 & -0.28 & -0.0068 & -0.0068 & 0.0049 & 0 \\ -25 & -25 & -0.59 & -0.59 & 3.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.015 & -0.015 & -0.36 & 0.36 & 0.083 & -0.051 \\ -0.24 & 0.24 & -9.8 & 9.8 & 0.26 & -0.37 \\ 0.38 & -0.38 & 0.19 & -0.19 & 0.52 & -4.6 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ c_{31} & c_{32} \\ 0 & 0 \\ 0 & 0 \\ c_{61} & c_{62} \\ c_{71} & c_{72} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1.0667 & 0.0156 \\ 0 & 0 \\ 0 & 0 \\ 0.0319 & -1.4713 \\ -0.7087 & 0.0319 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

C has components which are products of moments of inertia I_x, I_y, I_z, I_{xz}, which are all constants. v is the forward velocity [m/sec]; α is the angle of attack [rad]; q is the pitch rate [rad/sec]; θ is the pitch angle [rad]; β is the sideslip angle [rad]; p is the roll rate [rad/sec]; r is the yaw rate [rad/sec]; ϕ is the roll angle [rad]; ψ is the yaw angle [rad]; δ_{HR} and δ_{HL} are the deflections of the right and left horizontal stabilizers [rad]; δ_{FR} and δ_{FL} are the deflections of the right and left flaps [rad]; δ_C and δ_R are the canard and rudder deflections [rad].

The unknown matrices A, B and C are to be identified. Following the method outlined in section 3, we set the weight W and bias I using Eq.(36) and (37) as

$$W = \frac{1}{T}\int_0^T \begin{bmatrix} \left(G_1G_1^T\right) & 0 & \cdots & 0 & \left(G_1u^T\right) & 0 & \cdots & 0 \\ 0 & \left(G_2G_2^T\right) & \cdots & 0 & 0 & \left(G_2u^T\right) & \cdots & 0 \\ & \vdots & & \vdots & & \vdots & & \vdots \\ 0 & 0 & \cdots & \left(G_9G_9^T\right) & 0 & 0 & \cdots & \left(G_9u^T\right) \\ \left(uG_1^T\right) & 0 & \cdots & 0 & \left(uu^T\right) & 0 & \cdots & 0 \\ 0 & \left(uG_2^T\right) & \cdots & 0 & 0 & \left(uu^T\right) & \cdots & 0 \\ & \vdots & & \vdots & & \vdots & & \vdots \\ 0 & 0 & \cdots & \left(uG_9^T\right) & 0 & 0 & \cdots & \left(uu^T\right) \end{bmatrix}_{175\times175} dt \tag{73}$$

$$I = -\frac{1}{T}\int_0^T \left[\left(i_1G_1^T\right) \quad \left(i_2G_2^T\right) \quad \cdots \quad \left(i_9G_9^T\right) \quad \left(i_1u^T\right) \quad \left(i_2u^T\right) \quad \cdots \quad \left(i_9u^T\right)\right]^T_{175\times1} dt \tag{74}$$

And

$$G_1 = \begin{bmatrix} x_1 & x_2 & \cdots & x_9 & 0 & 0 \end{bmatrix}^T$$

$$G_2 = \begin{bmatrix} x_1 & x_2 & \cdots & x_9 & 0 & 0 \end{bmatrix}^T$$

$$G_3 = \begin{bmatrix} x_1 & x_2 & \cdots & x_9 & x_6\,x_7 & x_7^2 - x_6^2 \end{bmatrix}^T$$

$$G_4 = \begin{bmatrix} x_1 & x_2 & \cdots & x_9 & 0 & 0 \end{bmatrix}^T$$

$$G_5 = \begin{bmatrix} x_1 & x_2 & \cdots & x_9 & 0 & 0 \end{bmatrix}^T$$

$$G_6 = \begin{bmatrix} x_1 & x_2 & \cdots & x_9 & x_3\,x_6 & x_3\,x_7 \end{bmatrix}^T$$

$$G_7 = \begin{bmatrix} x_1 & x_2 & \cdots & x_9 & x_3\,x_6 & -x_3\,x_7 \end{bmatrix}^T$$

$$G_8 = \begin{bmatrix} x_1 & x_2 & \cdots & x_9 & 0 & 0 \end{bmatrix}^T$$

$$G_9 = \begin{bmatrix} x_1 & x_2 & \cdots & x_9 & 0 & 0 \end{bmatrix}^T$$

$I = [i_1 \quad i_2 \quad \cdots \quad i_9]^T$ where

$$i_1 = \dot{x}_1$$

$$i_2 = \dot{x}_2 + x_6 \cos x_2 \tan x_5 + x_7 \sin x_2 \tan x_5$$

$$i_3 = \dot{x}_3$$

$$i_4 = \dot{x}_4 - x_3 \cos x_8 + x_7 \sin x_8$$

$$i_5 = \dot{x}_5 - x_6 \sin x_2 + x_7 \cos x_2$$

$$i_6 = \dot{x}_6$$

$$i_7 = \dot{x}_7$$

$$i_8 = \dot{x}_8 - x_3 \tan x_4 \sin x_8 - x_7 \tan x_4 \cos x_8$$

$$i_9 = \dot{x}_9 - x_3 \cos^{-1} x_4 \sin x_8 - x_7 \cos^{-1} x_4 \cos x_8$$

Compared with the true values, we find that the estimated values are nearly the same (differences lie between 0-3%). Figure 4 to Figure 30 represent the plots of some parameters in A and B and trajectories of states. For some of them that do not show full convergence, we use the last values at the end of one iteration as initial values for the next iteration and recompute. All converge after the third pass to reach equilibrium.

The converged values of A , B and C are given below.

$$A = \begin{bmatrix} -0.0090 & 0.5299 & -0.2400 & -9.8 & -0.4598 & -0.0950 & -0.1400 & 0 & 0 \\ -0.001 & -0.6874 & 0.9976 & 0 & 0.1236 & 0.0370 & 0.0050 & 0 & 0 \\ 0.0002 & 2.7027 & -0.5299 & 0 & 0.0090 & 0.0062 & 0.0279 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.001 & 0.6888 & 0.4201 & 0.1763 & -0.7159 & 0.0860 & -0.1503 & 0 & 0 \\ 0.00002 & 1.0999 & 0.0411 & 0.0069 & -26 & -4.9046 & 0.5273 & 0 & 0 \\ 0.00001 & -1.7015 & 0.0981 & 0.0109 & 7.4 & -0.0370 & 0.8226 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0930 & 0.0930 & 0.0450 & -0.0450 & -0.0700 & -0.1300 \\ -0.2793 & -0.2797 & -0.0067 & -0.0069 & 0.0048 & 0 \\ -24.9999 & -24.9999 & -0.5900 & -0.5900 & 3.5000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0150 & -0.0150 & -0.3600 & 0.3600 & 0.0849 & -0.0510 \\ -0.2400 & 0.2401 & -9.8000 & 9.8001 & 0.2599 & -0.3701 \\ 0.3800 & -0.3799 & 0.1900 & -0.1890 & 0.5199 & -4.6001 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1.0666 & 0.0156 \\ 0 & 0 \\ 0 & 0 \\ 0.0316 & -1.4706 \\ -0.7090 & 0.0312 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Let *A*1 denote the estimates of first row in *A* matrix, *B*1 denotes the estimates of the first row in *B* matrix and so on. The histories of the parameter values are presented versus time. It can easily be observed that some of the parameters reach convergences after just one pass but some of them converge after two or three passes. For convergence, the simulation show that the neural network parameters ρ and λ can be chosen arbitrarily as long as $\rho > v_j$. Otherwise, v_j will converges to ρ. The bigger ρ, λ are, the faster the results will be. But they can lead to instability. For completion, the history of states is also provided in Figure 4.

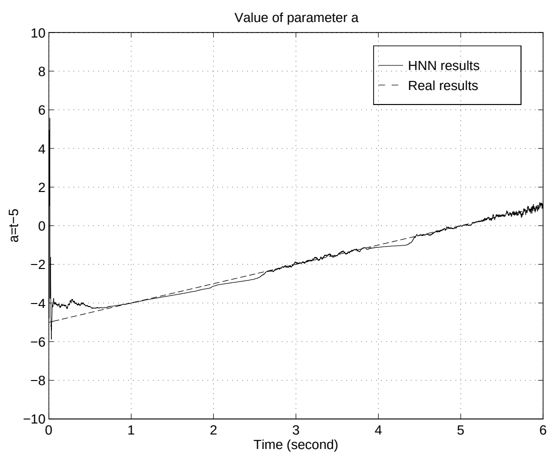

Figure 1 Parameter *a* trajectory of Case2

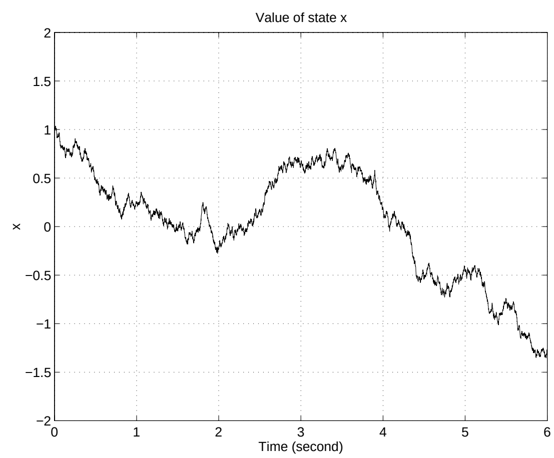

Figure 2 Trajectory of scalar state

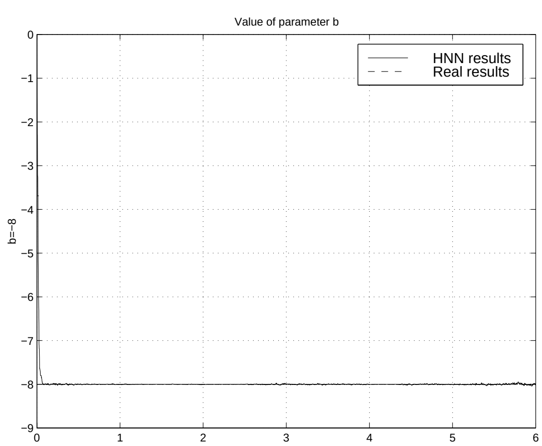

Figure 3 Parameter *b* trajectory of Case2

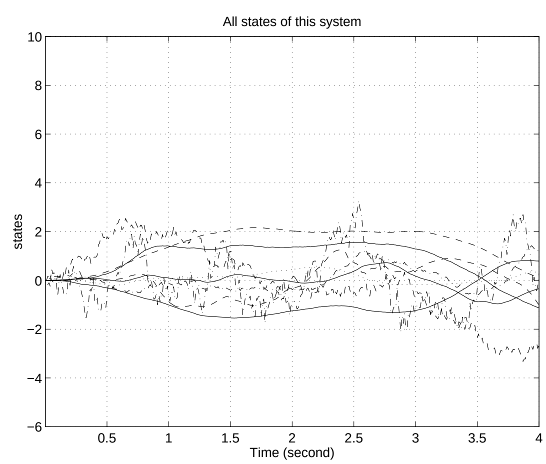

Figure 4 Trajectories of 9-state

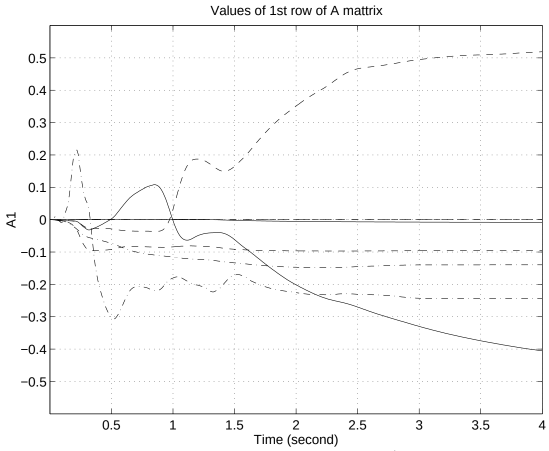

Figure 5 Values of $A1$ from 1st iteration

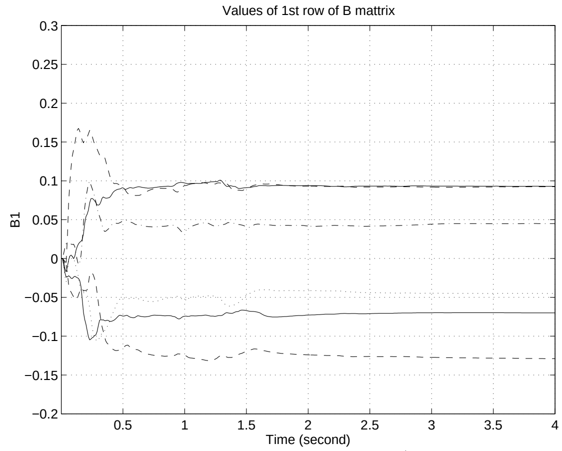

Figure 6 Values of $B1$ from 1st iteration

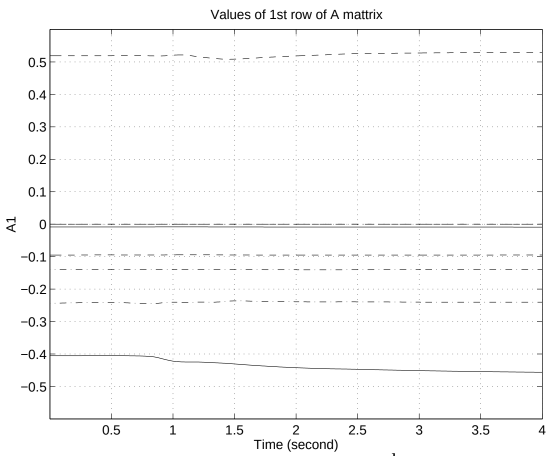

Figure 7 Values of $A1$ from 2nd iteration

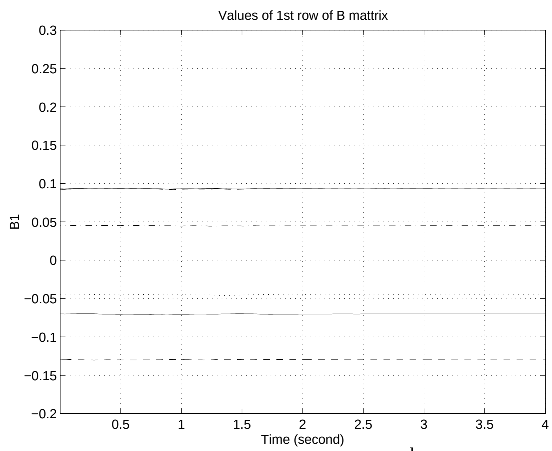

Figure 8 Values of $B1$ from 2nd iteration

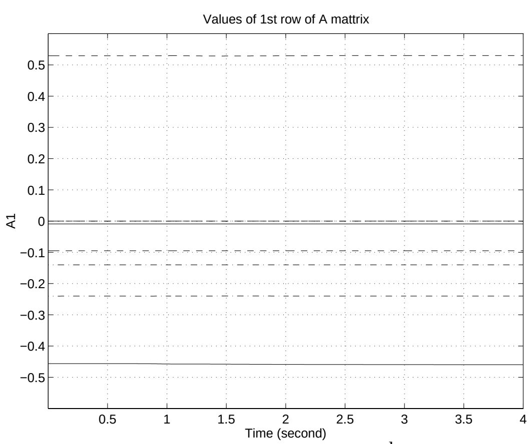

Figure 9 Values of $A1$ from 3rd iteration

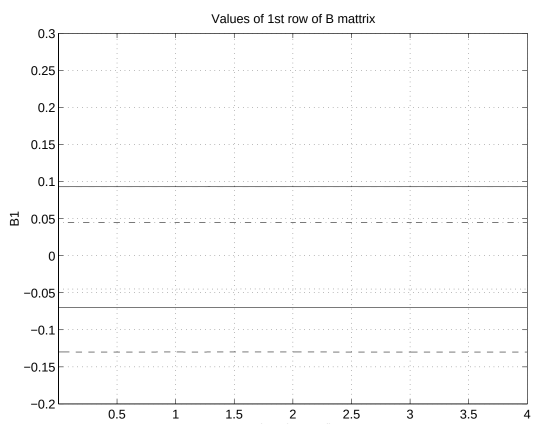

Figure 10 Values of $B1$ from 3rd iteration

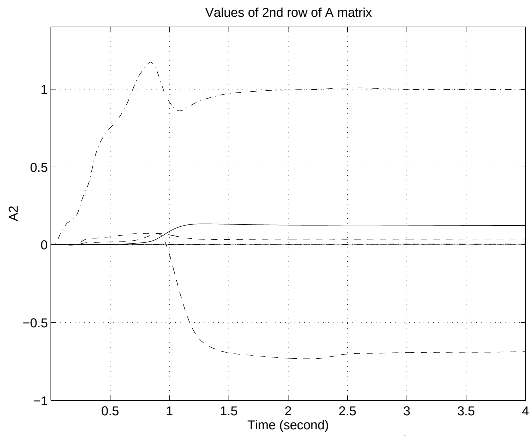

Figure 11 Values of $A2$ from 1st iteration

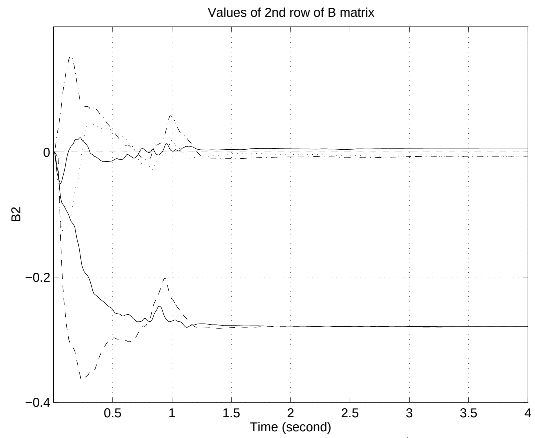

Figure 12 Values of $B2$ from 1st iteration

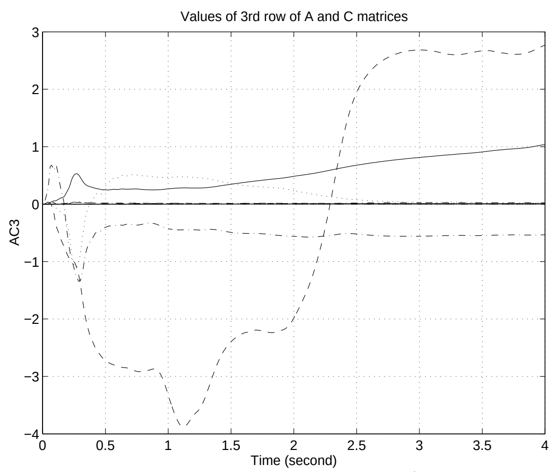

Figure 13 Values of $A3$, $C3$ of 1st iteration

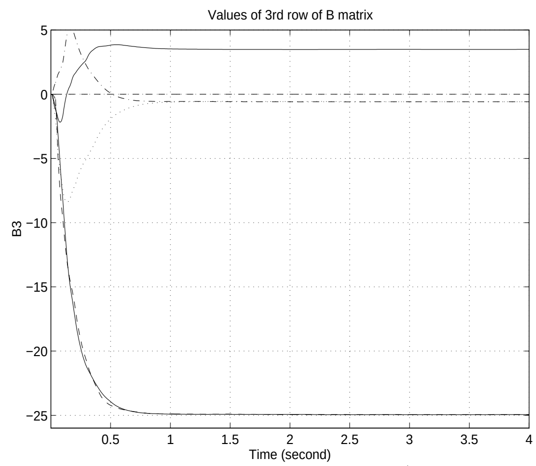

Figure 14 Values of $B3$ from 1st iteration

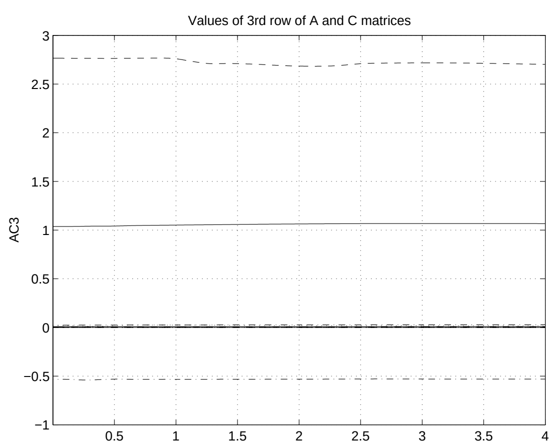

Figure 15 Values of $A3$, $C3$ of 2nd iteration

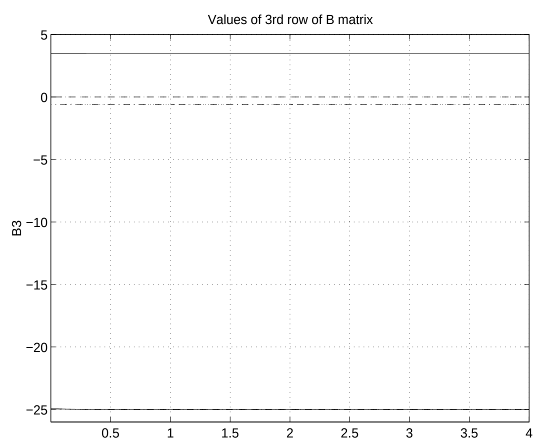

Figure 16 Values of $B3$ from 2nd iteration

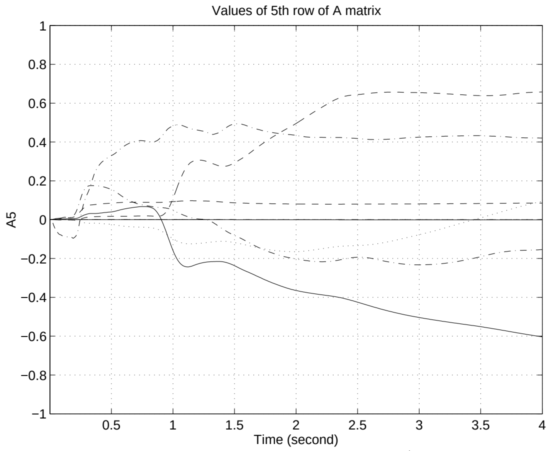

Figure 17 Values of *A*5 from 1st iteration

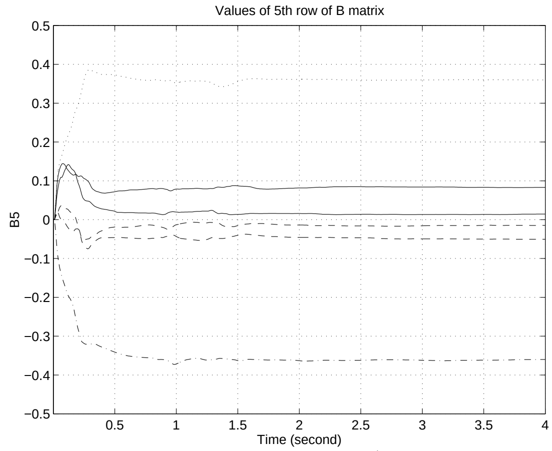

Figure 18 Values of *B*5 1st iteration

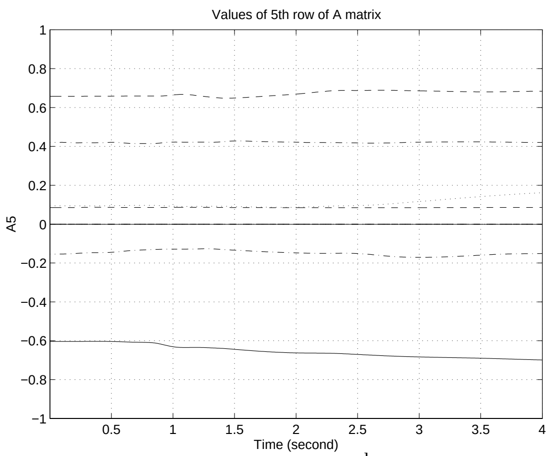

Figure 19 Values of *A*5 2nd iteration

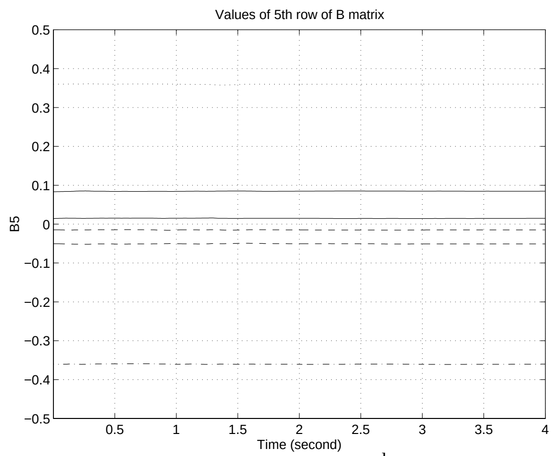

Figure 20 Values of *B*5 2nd iteration

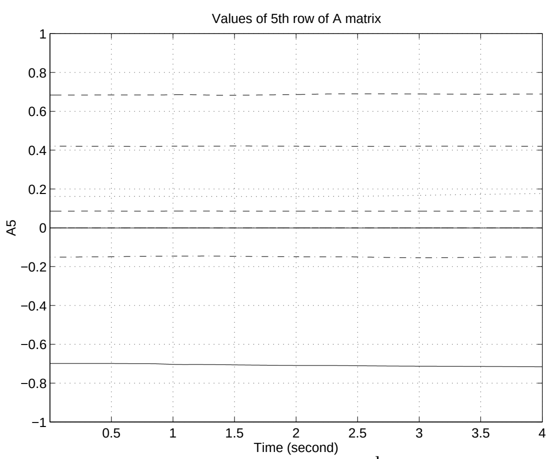

Figure 21 Values of *A*5 3rd iteration

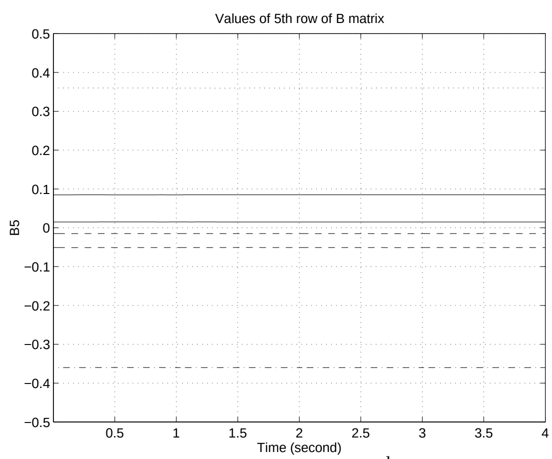

Figure 22 Values of *B*5 3rd iteration

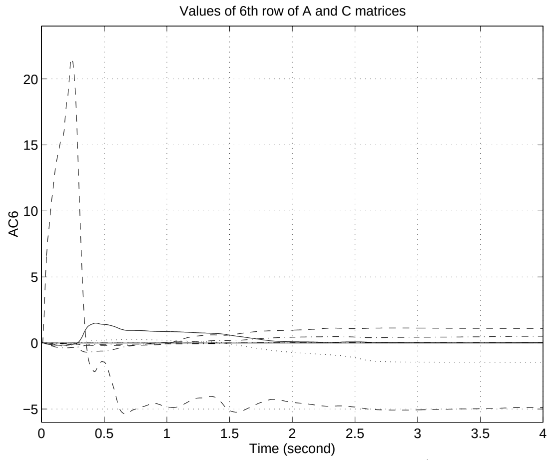

Figure 23 Values of *A*6, *C*6 of 1st iteration

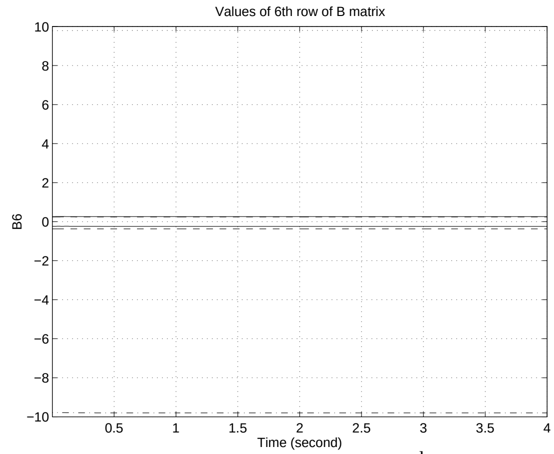

Figure 24 Values of *B*6 from 2nd iteration

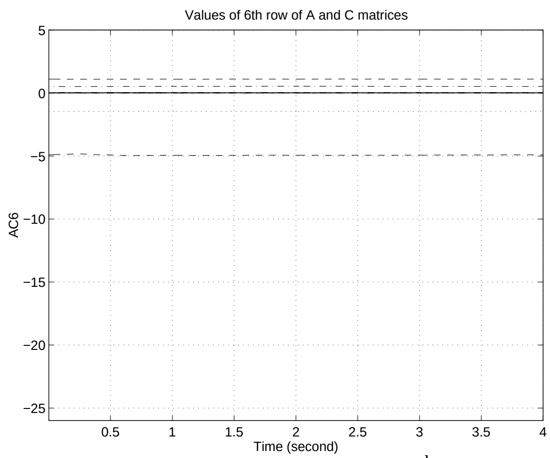

Figure 25 Values of *A*6, *C*6 of 2nd iteration

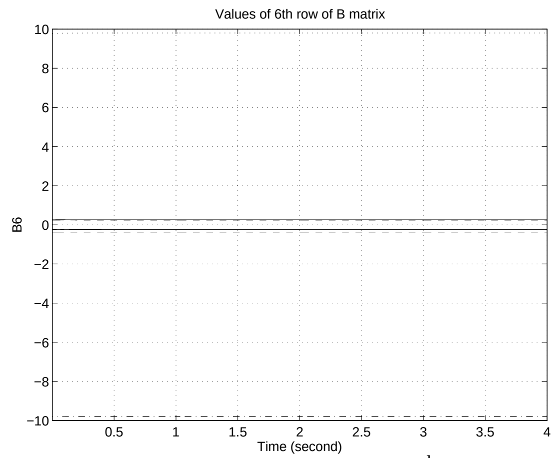

Figure 26 Values of *B*6 from 2nd iteration

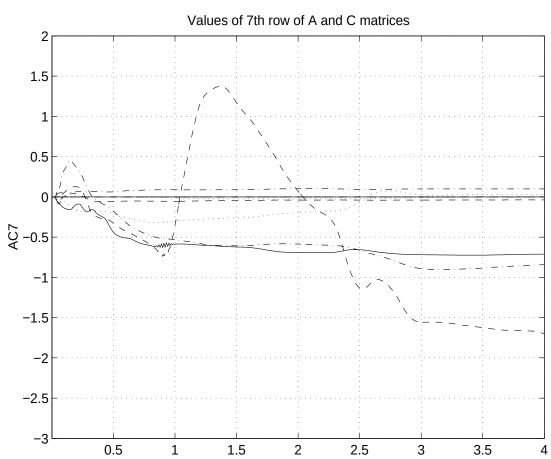

Figure 27 Values of *A*7, *C*7 of 1st iteration

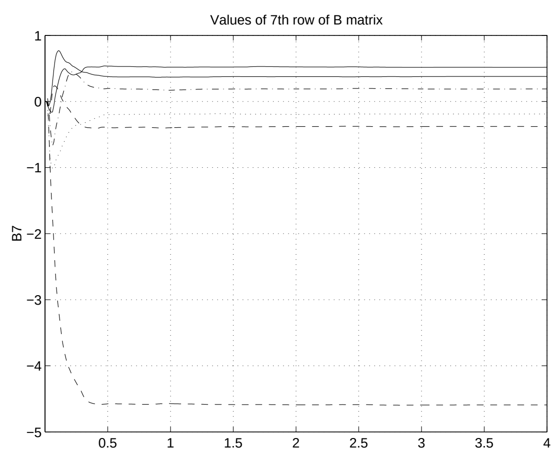

Figure 28 Values of *B7* from 1st iteration

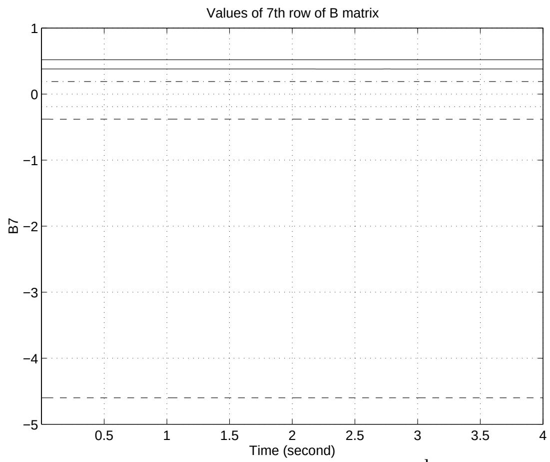

Figure 29 Values of $A7$, $C7$ of 2nd iteration

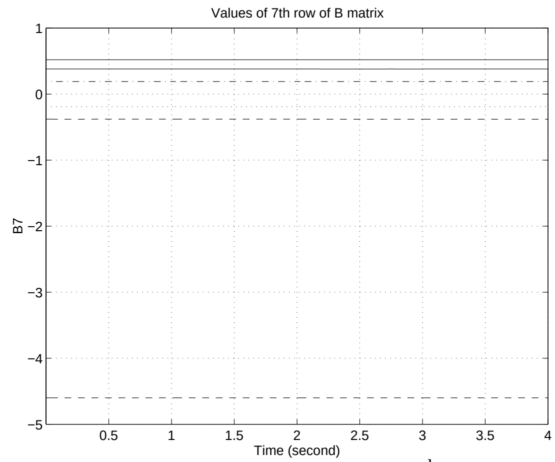

Figure 30 Values of $B7$ from 2nd iteration

5 CONCLUSIONS

A method to identify parameters of time-varying and time-invariant nonlinear systems has been presented. Effectiveness of the method has been demonstrated through applications. Due to the massively parallel nature of neural networks, this is a good candidate for online parameter estimation.

REFERENCES

Shen, Jie and S.N.Balakrishnan (1997). "A Class of Modified Hopfield Networks For Aircraft Identification and Control", Atmospheric Flight Mechanics Conference, San Diego, CA.

Raol, J.R. (1994). "Neural Network Based Parameter Estimation of Unstable Aerospace Dynamic Systems", IEE Proc. Control Theory Application, Vol.141, No.6, pp.114-118.

Raol, J.R. (1995). "Parameter Estimation o f State Space Models By Recurrent Neural Network", IEE Proc.-Control Theory Application., Vol.142, No.2, pp.385-388.

Raol, J.R. (1996). "Neural Network Architectures For Parameter Estimation of Dynamic Systems", IEE Proc. Control Theory Application., Vol.143, No.4, pp.387-394.

Cetinkunt, Sabri and Hsin-Tan Chiu (1993). "Estimation of Model Parameters of Linear Structural Systems Using Hopfield Networks", Journal of Guidance, Control and Dynamics, Vol.17, No.3, pp.621-624, Engineering notes.

Hopfield, J. (1982). "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", Proc. of the National Academy of Science, 79, pp.2554-2558.

Hopfield, J. (1984). "Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons", Proc. of the National Academy of Science, 81, pp.3088-3092.

Hopfield, J. and D.Tank (1985). "Neuroal Computation of Decisions In Optimization Problems", Biological Cybernetics, 52, pp.141-152.

Cichocki, Andrzej and Rolf Unbehauen (1992). "Neural Networks For Solving Systems of Linear Equations and Related Problems", IEEE Trans. On Circuits and Systems, Vol.39, No.2, pp.124-138.

Lyashevskiy, Sergey (1997). "Nonlinear Mapping Technology In System Identification", Proc. of the American Control Conference, pp.605-606, Albuquerque, NM.

Lyshevski, Sergey (1998). "Nonlinear Identification of Control System", Proc. of the American Control Conference, pp.2366-2370, Philadelphia, PA.

Vidyasagar, M. (1993). "Nonlinear System Analysis", Prentice-Hall.

Amin, S. Massoud, Volker Gerhart and Ervin Y.Rodin (1997). "System Identification Via Artificial Neural Networks: Applications to On-line Aircraft Parameter Estimation", 1997 World Aviation Congress, Anaheim, CA.

Kim, Young H., Frank L.liews and Chaouki T. Abdallah (1997). "A dynamic Recurrent Neural Network-Based Aaptive Observer For A Class of Nonlinear Systems", Automatica, vol.133, No.8, pp.1539-1543.

Kambhampati, C., R.Craddock, M.Tham and K. Warwick (1998). "Inverting Recurrent Neural Network For Internal Model Control of Nonlinear Systems", Proc. of the American Control Conference, pp.975-979, Philadelphia, PA.

Jagannathan, S. and F.L.Lewis (1996). "Identification of Nonlinear Dynamic Systems Using Multilayered Neural Networks", Automatica, Vol.132, No.12, pp.1707-1712.

Bruyne, Franky De (1998). "Closed-Loop Relevant Identification Of Input-Output And Noise Dynamics", Proc. of the American Control Conference, pp3199-3203, Philadelphia, PA.

Habib, S. and M.Zaghloul (1996). "Concurrent Identification and Control of System by applying Standardized Neurocontroller and Neuroplant", AIAA Guidance Navigation and Control Conference, San Diego, CA.

Habib, S. and M.Zaghloul (1006). "System Identification Using Time Dependent Neural Networks: A Simple Approach", AIAA Guidance Navigation and Control Conference, San Diego, CA.

Stubberud, A., H.Wabgaonkar and S**.** Stubberud (1991). "A Neural-Network-Based System Identification Technique", Proceedings of 30th Conference in Decision and Control, pp869-870, Brighton, England.