JOURNAL OF AIRCRAFT
Vol. 42, No. 1, January–February 2005

Parameter Estimation in Nonlinear Systems Using Hopfield Neural Networks

Zhenning Hu* and S. N. Balakrishnan†
University of Missouri–Rolla, Rolla, Missouri 65401

A method using the Hopfield neural network is developed for estimating the parameters of a nonlinear system whose theoretical model is assumed to exist. A linearization procedure is presented, and the errors between the dynamics of the plant and its model are minimized through a cost function that is equated to the energy function of a Hopfield neural network. The minimization process yields the weights and biases of the neural network. Proof of convergence of the modeled parameters to their true values and boundedness of parameter estimates at each step are provided. Numerical results from a scalar time-varying problem and a complex nine-state aircraft problem are presented to demonstrate the potential of this method.

I. 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 estimation is useful is in the area of postflight 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 have been and continue to be numerous studies and development of new methods in the area of parameter estimation.

The capability of artificial neural networks to model the behavior of large classes of uncertain nonlinear dynamic systems within a certain accuracy has made them 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 work,[1–3] there has been a multitude of papers using recurrent neural networks for linear system solvers, control[4–5] and pattern recognition.[5–8] Whereas feedforward networks are static mapping between two information domains, the structure of recurrent neural networks incorporates dynamic behavior through feedback connections. Many researchers and scientists have begun to use this new tool in their fields. Cichocki and Unbehauen's[9] extensive and thorough research of linear systems concentrates on algebraic equations. They present several network configurations and compare their speeds in finding inverses and solutions to linear systems of algebraic equations. Raol applies recurrent neural networks to linear and time invariant dynamic systems.[10–12] He has developed a system of equations for parameter estimation using several recurrent networks. The development in this study is similar to Raol's[10]; however, proofs of convergence and boundedness are presented and applications include a highly nonlinear problem. Lyshevsky uses his network mapping strategy[13,14] to identify parameters in a nonlinear system that can be written into a linear form. Amin et al.[15] show special insight into Hopfield[1] neural networks and network structure. With their observations, they propose new recurrent high-order neural networks (see Ref. 5). They develop Lyapunov-based theorems (see Ref. 13) to show analytical results with their higher-order recurrent networks, which are more complex than the networks used in this study. A lot of network structures and their convergence or robustness analysis are also being studied by other researchers, such as Kim et al.,[16] Kambhampati et al.,[17] and Stubberud et al.[18]

In this study, Hopfield recurrent neural networks (HNN) are used for online estimation of parameters of dynamic systems. There are a few papers in the literature that show this kind of neural network's potential for parameter estimation. The focus of this study is to develop an algorithm to deal with nonlinear systems that are more common in engineering processes and to demonstrate its effectiveness through applications. Nonlinear and time-varying systems, including an aircraft, are used as example applications. The working mechanism of the network with respect to its parameters in achieving convergence is also discussed.

II. HNN Structure

Dynamics of the HNN are discussed briefly in this section. The HNN dynamics are characterized by the following system of first-order differential equations

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

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 a neuron to its activation potential $u_j(t)$ is a continuous function and, therefore, differentiable, and that 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 Refs. 1–3, an energy (Lyapunov) (see Ref. 13) function of this recurrent neural network is identified as

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

Received 18 June 2003; revision received 15 December 2003; accepted for publication 16 December 2003.

*Graduate Student, Department of Mechanical and Aerospace Engineering.

†Professor, Department of Mechanical and Aerospace Engineering; bala@umr.edu. Associate Fellow AIAA.

The differential equation of $E$with respect to time is

$$\frac{\mathrm{d}E}{\mathrm{d}t} = -\sum_{j=1}^{n}\left(\sum_{k=1}^{n} w_{jk}v_k - g_j u_j + i_j\right)\frac{\mathrm{d}v_j}{\mathrm{d}t}$$

$$= -\sum_{j=1}^{n} c_j\left(\frac{\mathrm{d}u_j}{\mathrm{d}t}\right)\frac{\mathrm{d}v_j}{\mathrm{d}t} = -\sum_{j=1}^{n} c_j\left[\frac{\mathrm{d}}{\mathrm{d}t} f_j^{-1}(v_j)\right]\frac{\mathrm{d}v_j}{\mathrm{d}t}$$

$$= -\sum_{j=1}^{n} c_j\left[\frac{\mathrm{d}}{\mathrm{d}v_j} f_j^{-1}(v_j)\right]\left(\frac{\mathrm{d}v_j}{\mathrm{d}t}\right)^2 \tag{5}$$

For a log sigmoid, tangent sigmoid, and linear activation function,

$$\frac{\mathrm{d}}{\mathrm{d}v_j} f_j^{-1}(v_j) \geq 0 \tag{6}$$

for all $v_j(t)$. Thus, for the energy function E just defined,

$$\frac{\mathrm{d}E}{\mathrm{d}t} < 0 \tag{7}$$

for $v_j \neq 0$. Note the following:

1) This function E is a Lyapunov (see Ref. 13) function of the HNN.[1]

2) The model is stable in accordance with Lyapunov's (see Ref. 13) theorem.

III. Parameter Estimation of Dynamic Systems

This section describes a general nonlinear dynamic system and then presents the steps to linearize it. It then shows how the states of the system can be used to derive the values for the weights and biases for an HNN. Boundedness of the parameter estimates and convergence of the proposed method are also shown.

A. Dynamic System Description

Consider the following nonlinear dynamic system described in a state-space form:

$$\dot{x} = F(A, \boldsymbol{x}) + B\boldsymbol{u} \tag{8}$$

where $\boldsymbol{x}$ is a $n \times 1$ vector, $\boldsymbol{u}$ is a $p \times 1$ control input, and $F(A, \boldsymbol{x})$ is a certain kind of nonlinear function that can be described in the following forms,

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

$$= \begin{bmatrix} f_{11}[a_{11}h_{11}(\boldsymbol{x})] + \cdots + f_{1m}[a_{1m}h_{1m}(\boldsymbol{x})] \\ f_{21}[a_{21}h_{21}(\boldsymbol{x})] + \cdots + f_{2m}[a_{2m}h_{2m}(\boldsymbol{x})] \\ \vdots \\ f_{n1}[a_{n1}h_{n1}(\boldsymbol{x})] + \cdots + f_{nm}[a_{nm}h_{nm}(\boldsymbol{x})] \end{bmatrix} \tag{9}$$

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

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

where $f_i(\bullet, \bullet)$is nonlinear in $\boldsymbol{x}$.

In $F(A, \boldsymbol{x})$, each row can have a different number of terms, $m_1, m_2, \ldots, m_n$. Here, for simplicity, but without losing generality, it is assumed that they all have an 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 inside any nonlinear function as coefficients of individual terms. To realize this transformation, the given nonlinear dynamic system needs to be linearized.

B. Nonlinear System Linearization

Linear systems have been studied extensively and are relatively easy to deal with. To identify parameters in a system that contains nonlinear terms, like Eq. (8), each nonlinear element $f_{ij}(a_{ij}, \boldsymbol{x})$ in $F(A, \boldsymbol{x})$ is going to be changed into a linear form. However, the original system should satisfy some criteria 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 (Ref. 19): For an autonomous system, $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$, $f(0) = 0$, that is, $\boldsymbol{x} = 0$ is an equilibrium f is continuously differentiable. Let

$$A = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{x=0}$$

that is, let A denote the Jacobian matrix of f evaluated at $\boldsymbol{x} = 0$, $R(\boldsymbol{x}) = f(\boldsymbol{x}) - A\boldsymbol{x}$. Then, if it turns out that $\lim_{\|x\|=0}[\|R(\boldsymbol{x})\|/\|\boldsymbol{x}\|] = 0$, that is, the Taylor series expansion of $f(\boldsymbol{x}) = A\boldsymbol{x} + R(\boldsymbol{x})$, the system $\dot{z} = Az$, is called the linearization of the nonlinear system around the equilibrium $\boldsymbol{x} = 0$.

Definition 2 (Ref. 19): Given the nonautonomous system,

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

Suppose that

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

and that f is a C^1 function. Define

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

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

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

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

then the system

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

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

Theorem (Ref. 19): Consider the system (12). Suppose that Eq. (13) holds and that $f()$ is continuously differentiable. Define $A(t)$, $f_1(t, \boldsymbol{x})$ as in Eqs. (14) and (15), respectively, and assume that 1) Eq. (16) holds, and 2) $A()$ 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, that this theorem and these definitions are all based on the equilibrium being at the origin. However, this is only for convenience and can be relaxed. Note that we can use the results of Ref. 19 to extend its validity to where the equilibrium need not be the origin.

We can have

$$f_{ij}[a_{ij}h_{ij}(\boldsymbol{x})] = f_{ij}\left[a_{ij}h_{ij}(\boldsymbol{x}^*)\right] + f'_{ij}a_{ij}h'_{ij}(\boldsymbol{x}^*)\cdot(\boldsymbol{x}-\boldsymbol{x}^*) + \Delta(\boldsymbol{x}-\boldsymbol{x}^*)^2 \tag{18}$$

Because $f_{ij}[a_{ij}h_{ij}(\boldsymbol{x}^*)]=0$ (where $\boldsymbol{x}^*$ is the equilibrium point), and $\Delta(\boldsymbol{x}-\boldsymbol{x}^*)\to 0$ because $\boldsymbol{x}$ is in the neighborhood of $\boldsymbol{x}^*$. Then,

$$f_{ij}[a_{ij}h_{ij}(\boldsymbol{x})] \approx f'_{ij}\big[a_{ij}h_{ij}(\boldsymbol{x}^*)\big]a_{ij}h'_{ij}(\boldsymbol{x}^*)\cdot(\boldsymbol{x}-\boldsymbol{x}^*) + \Delta(\boldsymbol{x}-\boldsymbol{x}^*)^2 \equiv a_{ij}g_{ij}(\boldsymbol{x}) \tag{19}$$

where $g_{ij}(\boldsymbol{x})=f'_{ij}[a_{ij}h_{ij}(\boldsymbol{x}^*)]h'_{ij}(\boldsymbol{x}^*)\cdot(\boldsymbol{x}-\boldsymbol{x}^*)$. Thus, the whole nonlinear vector becomes

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

Each row in $F(A,\boldsymbol{x})$ is now a sum of linear terms in $\boldsymbol{x}$. Note, that all of the unknown parameters appear as coefficients of the terms in $F(A,\boldsymbol{x})$. Now, the term $F(A,\boldsymbol{x})$ will be related to the energy function and weights of the neural networks.

C. Computation of Weights and Biases

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

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

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

$$E(A,B,\boldsymbol{x}) = \frac{1}{2T}\int_0^T \boldsymbol{e}^T\boldsymbol{e}\,\mathrm{d}t = \frac{1}{2T}\int_0^T [\dot{x} - F_s(A,\boldsymbol{x}) - B_s u]^T \times [\dot{x} - F_s(A,\boldsymbol{x}) - B_s u]\,\mathrm{d}t \tag{22}$$

where T is time period over which data are collected.

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

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

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

If A_i and B_i are assumed to represent the ith row of A, and B, respectively, and $\boldsymbol{V}$ as a vector consisting of columns of matrices A and B, then we get

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

Equations (23) and (24) can be rewritten in terms of the elements of $\boldsymbol{V}$ (Ref. 10), for $1\le s\le mn$, as

$$\begin{aligned}\frac{\partial E}{\partial v_s} &= \frac{\partial E}{\partial a_{ij}} \\ \frac{\partial E}{\partial v_s} &= \sum_{k=1}^{m} v_{(i-1)m+k}(t)\frac{1}{T}\int_0^T g_{ik}(x)g_{ij}(x)\,\mathrm{d}t + \sum_{k=1}^{p} v_{mn+(i-1)p+k}(t)\frac{1}{T}\int_0^T u_k g_{ij}(x)\,\mathrm{d}t - \frac{1}{T}\int_0^T \dot{x}_i g_{ij}(x)\,\mathrm{d}t\end{aligned} \tag{26}$$

and, for $mn+1\le s\le mn+np$, as

$$\begin{aligned}\frac{\partial E}{\partial v_s} &= \frac{\partial E}{\partial b_{ij}} \\ \frac{\partial E}{\partial v_s} &= \sum_{k=1}^{m} v_{(i-1)m+k}(t)\frac{1}{T}\int_0^T g_{ik}(x)u_j\,\mathrm{d}t + \sum_{k=1}^{p} v_{mn+(i-1)p+k}(t)\frac{1}{T}\int_0^T u_k u_j\,\mathrm{d}t - \frac{1}{T}\int_0^T \dot{x}_i u_j\,\mathrm{d}t\end{aligned} \tag{27}$$

Now the parameter estimation formulations in Eqs. (26) and (27) can be related in terms of weights of the HNN as

$$\frac{\mathrm{d}E}{\mathrm{d}V_s} = \sum_{r=1}^{mn+np} W_{sr}V_r + I_{sr} \tag{28}$$

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

For $1\le s\le mn$,

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

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

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

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

$$I_{sr} = -\frac{1}{T}\int_0^T \dot{x}_i u_j\,\mathrm{d}t \tag{32}$$

For a more compact representation, define

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

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

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

and the parameter estimation formulation becomes

$$\frac{\mathrm{d}E}{\mathrm{d}V} = WV + I \tag{35}$$

where W (weight) and I(bias) are set as follows:

$$W = \frac{1}{T}\int_0^T \begin{bmatrix} (G_1G_1^T) & 0 & \ldots & 0 & (G_1u^T) & 0 & \ldots & 0 \\ 0 & (G_2G_2^T) & \ldots & 0 & 0 & (G_2u^T) & \ldots & 0 \\ & \vdots & & \vdots & & \vdots & & \vdots \\ 0 & 0 & \ldots & (G_nG_n^T) & 0 & 0 & \ldots & (G_nu^T) \\ (uG_1^T) & 0 & \ldots & 0 & (uu^T) & 0 & \ldots & 0 \\ 0 & (uG_2^T) & \ldots & 0 & 0 & (uu^T) & \ldots & 0 \\ & \vdots & & \vdots & & \vdots & & \vdots \\ 0 & 0 & \ldots & (uG_n^T) & 0 & 0 & \ldots & (uu^T) \end{bmatrix}_{(mn+np)\times(mn+np)} \mathrm{d}t \tag{36}$$

$$I = -\frac{1}{T}\int_0^T \left[(\dot{x}_1G_1^T)\ (\dot{x}_2G_2^T) \ldots (\dot{x}_nG_n^T)\ (\dot{x}_1u^T)\ (\dot{x}_2u^T) \ldots (\dot{x}_nu^T) \right]^T_{(mn+np)\times 1} \mathrm{d}t \tag{37}$$

Now, this formulation has to be related to the dynamics of an HNN. The network dynamics can be written in the following form[9]:

$$\frac{\mathrm{d}V}{\mathrm{d}t} = -\mu\frac{\mathrm{d}E}{\mathrm{d}V} = -\mu(WV + I) \tag{38}$$

To find the expression for μ, a tangent sigmoid activation function is chosen, and Eq. (5) is rewritten to obtain

$$\begin{aligned} \frac{\mathrm{d}v_j}{\mathrm{d}t} &= -\left\{1\bigg/\left[\frac{\mathrm{d}}{\mathrm{d}t}f^{-1}(v_j)\right]\right\}\frac{1}{c_j}\frac{\mathrm{d}E}{\mathrm{d}t} \\ &= -\left\{1\bigg/\left[\frac{\mathrm{d}}{\mathrm{d}v_j}f^{-1}(v_j)\right]\right\}\frac{1}{c_j}\frac{\mathrm{d}E}{\mathrm{d}v_j} \end{aligned} \tag{39}$$

where

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

From Eq. (40),

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

By the insertion of Eq. (41) into Eq. (38), it can be shown that

$$\frac{\mathrm{d}v_j}{\mathrm{d}t} = -\frac{\lambda_i\left(\rho^2 - v_j^2\right)}{2\rho c_j}\frac{\mathrm{d}E}{\mathrm{d}v_j} \tag{42}$$

A comparison of Eq. (41) with Eq. (37) shows that

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

and $\mu = \mathrm{diag}[\mu_1\ \mu_2\ \ldots\ \mu_{mn+np}]$.

D. Boundedness Analysis

In this section, a proof of boundedness of estimates at each step is outlined. By the discretization of Eq. (42), the use of Eqs. (23), (24), and (34) in parameter v_i, and under the assumption of a small time step t and $t + \Delta$, it can be shown that

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

If v_i belongs to b_{ij}, then g_{ij} can be changed to u_j to get

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

Because $\|A_i(t)\| < \infty$ and $\|F_i(A, \boldsymbol{x}) - A_i^T G_i\|/\|\boldsymbol{x}\| \to \mathrm{O}$ as $\|\boldsymbol{x}\| \to \boldsymbol{x}^*$ (from the linearization theorem), $\dot{\boldsymbol{x}}_i - A_i^T G_i - B_i^T u = o(G_i^T G_i)$ is a second-order term. Because $\rho \gg v_j[v_j = f(u_j) = \rho[(1 - e^{-\lambda_j u_j})/(1 + e^{-\lambda_j u_j})]$, where $f \to \pm p$ as $u_j \to \pm\infty$ and $\lambda_i \ll \rho$, $\mu_j = -[\lambda_i(\rho^2 - v_j^2)/2\rho c_j]$ is bounded. (Usually the parameter ρ is picked as a high value but can be different for each row.) Because $g_{ij}(\boldsymbol{x})$is bounded at each $\boldsymbol{x}$ and u_j, the control signal is bounded. This implies that $\Delta v_i(\Delta t) = \|v_i(t + \Delta t) - v_i(t)\|$ is bounded.

E. Convergence Analysis

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

$$\begin{aligned} F_s(A, \boldsymbol{x}) &= \begin{bmatrix} A_1^T G_1 \\ A_2^T G_2 \\ \vdots \\ A_n^T G_n \end{bmatrix} \\ &= \begin{bmatrix} (A_1^T)_{1\times m} & 0 & \ldots & 0 \\ 0 & (A_2^T)_{1\times m} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & (A_n^T)_{1\times m} \end{bmatrix}_{n\times(m\times n)} \\ &\quad \times \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_n \end{bmatrix}_{(m\times n)\times 1} \equiv A_s G_s \end{aligned} \tag{46}$$

omitting the argument x with G_s, and where

$$A_s = \begin{bmatrix} \left(A_1^T\right)_{1\times m} & 0 & \dots & 0 \\ 0 & \left(A_2^T\right)_{1\times m} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \left(A_n^T\right)_{1\times m} \end{bmatrix}_{n\times(m\times n)}$$

$$G_s = \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_n \end{bmatrix}_{(m\times n)\times 1}$$

The cost function in terms of these parameters is

$$E = \frac{1}{2T}\int_0^T (\dot{x} - A_s G_s - B_s \boldsymbol{u})^T (\dot{x} - A_s G_s - B_s \boldsymbol{u})\, \mathrm{d}t \qquad (47)$$

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

$$\begin{aligned} E = \frac{1}{2T}\int_0^T &\Big(G_s^T A_s^T A_s G_s + \boldsymbol{u}^T B_s^T \boldsymbol{u} + G_s^T A_s^T B_s \boldsymbol{u} \\ &+ \boldsymbol{u}^T B_s^T A_s G_s \boldsymbol{x} - \dot{x} A_s G_s - G_s^T A_s^T \dot{x} - \dot{x}^T B_s \boldsymbol{u} \\ &- \boldsymbol{u}^T B_s^T \dot{x} + \dot{x}^T \dot{x}\Big)\, \mathrm{d}t \end{aligned} \qquad (48)$$

To help with derivations further, the following trace operations are used:

$$(A^T B^T C D)_{1\times 1} = \mathrm{tr}(BAD^T C^T)_{1\times 1} \qquad (49)$$

By observation that the left part of Eq. (49) is a scalar,

$$(D^T C^T B A)_{1\times 1} = \mathrm{tr}(BAD^T C^T)_{1\times 1} \qquad (50)$$

It is also easy to see from Eqs. (49) and (50) that

$$(A^T B^T D)_{1\times 1} = \mathrm{tr}(BAD^T)_{1\times 1} \qquad (51)$$

$$(D^T B A)_{1\times 1} = \mathrm{tr}(BAD^T)_{1\times 1} \qquad (52)$$

Thus, Eq. (48) is rearranged by the use of results produced by Eqs. (49–52) as

$$\begin{aligned} E = {} & \mathrm{tr}\left[A_s\left(\frac{1}{2T}\int_0^T G_s G_s^T \mathrm{d}t\right)A_s^T\right] + \mathrm{tr}\left[B_s\left(\frac{1}{2T}\int_0^T \boldsymbol{u}\boldsymbol{u}^T \mathrm{d}t\right)B_s^T\right] \\ & + \mathrm{tr}\left[A_s\left(\frac{1}{2T}\int_0^T G_s \boldsymbol{u}^T \mathrm{d}t\right)B_s^T\right] - \mathrm{tr}\left[A_s\left(\frac{1}{2T}\int_0^T G_s \dot{\boldsymbol{x}}^T \mathrm{d}t\right)\right] \\ & - \mathrm{tr}\left[B_s\left(\frac{1}{2T}\int_0^T \boldsymbol{u}\dot{\boldsymbol{x}}^T \mathrm{d}t\right)\right] + \frac{1}{2T}\int_0^T \dot{\boldsymbol{x}}^T \dot{\boldsymbol{x}}\, \mathrm{d}t \end{aligned} \qquad (53)$$

Two other important trace operations are given by

$$\frac{\partial}{\partial A}\mathrm{tr}(ABA^T) = 2AB \qquad (54)$$

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

By the use of Eqs. (54) and (55), and with $\dot{\boldsymbol{x}} = F_p(A, \boldsymbol{x}) + B_p \boldsymbol{u} = A_p G_s + B_p \boldsymbol{u}$, the following equations can be obtained:

$$\frac{\partial}{\partial A_s}\mathrm{tr}\left[A_s\left(\frac{1}{2T}\int_0^T G_s G_s^T \mathrm{d}t\right)A_s^T\right] = A_s\left(\frac{1}{T}\int_0^T G_s G_s^T\, \mathrm{d}t\right) \qquad (56)$$

$$\frac{\partial}{\partial A_s}\mathrm{tr}\left[B_s\left(\frac{1}{2T}\int_0^T \boldsymbol{u}\boldsymbol{u}^T\, \mathrm{d}t\right)B_s^T\right] = 0 \qquad (57)$$

$$\frac{\partial}{\partial A_s}\mathrm{tr}\left[A_s\left(\frac{1}{T}\int_0^T G_s \boldsymbol{u}^T\, \mathrm{d}t\right)B_s^T\right] = B_s\left(\frac{1}{T}\int_0^T \boldsymbol{u} G_s^T\, \mathrm{d}t\right) \qquad (58)$$

$$\begin{aligned} &\frac{\partial}{\partial A_s}\mathrm{tr}\left[A_s\left(\frac{1}{T}\int_0^T G_s(A_p G_s + B_p \boldsymbol{u})\, \mathrm{d}t\right)\right] \\ &\quad = \frac{1}{T}\int_0^T (A_p G_s + B_p \boldsymbol{u}) G_s^T\, \mathrm{d}t \end{aligned} \qquad (59)$$

$$\frac{\partial}{\partial A_s}\mathrm{tr}\left[B_s\left(\frac{1}{T}\int_0^T \boldsymbol{u}\dot{x}^T\, \mathrm{d}t\right)\right] = 0 \qquad (60)$$

$$\frac{\partial}{\partial A_s}\mathrm{tr}\left(\frac{1}{T}\int_0^T \dot{x}^T \dot{x}\, \mathrm{d}t\right) = 0 \qquad (61)$$

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

$$\frac{\partial E}{\partial A_s} = (A_s - A_p)\left(\frac{1}{T}\int_0^T G_s G_s^T\, \mathrm{d}t\right) + (B_s - B_p)\left(\frac{1}{T}\int_0^T \boldsymbol{u} G_s^T\, \mathrm{d}t\right) \qquad (62)$$

Similarly,

$$\frac{\partial E}{\partial B_s} = (A_s - A_p)\left(\frac{1}{T}\int_0^T G_s \boldsymbol{u}^T\, \mathrm{d}t\right) + (B_s - B_p)\left(\frac{1}{T}\int_0^T \boldsymbol{u}\boldsymbol{u}^T\, \mathrm{d}t\right) \qquad (63)$$

By the combination of Eqs. (61) and (62) and setting them to be zero for minimum error,

$$\begin{bmatrix} A_s - A_p & B_s - B_p \end{bmatrix} \begin{bmatrix} \dfrac{1}{T}\displaystyle\int_0^T G_s G_s^T\, \mathrm{d}t & \dfrac{1}{T}\displaystyle\int_0^T G_s \boldsymbol{u}^T\, \mathrm{d}t \\ \dfrac{1}{T}\displaystyle\int_0^T \boldsymbol{u} G_s^T\, \mathrm{d}t & \dfrac{1}{T}\displaystyle\int_0^T \boldsymbol{u}\boldsymbol{u}^T\, \mathrm{d}t \end{bmatrix} = 0 \qquad (64)$$

or

$$\begin{bmatrix} A_s - A_p & B_s - B_p \end{bmatrix} \cdot \frac{1}{T}\int_0^T \begin{bmatrix} G_s G_s^T & G_s \boldsymbol{u}^T \\ \boldsymbol{u} G_s^T & \boldsymbol{u}\boldsymbol{u}^T \end{bmatrix} \mathrm{d}t = 0 \qquad (65)$$

or

$$\begin{bmatrix} A_s - A_p & B_s - B_p \end{bmatrix} \cdot \frac{1}{T}\int_0^T \begin{bmatrix} G_s \\ \boldsymbol{u} \end{bmatrix} \begin{bmatrix} G_s^T & \boldsymbol{u}^T \end{bmatrix} \mathrm{d}t = 0 \qquad (66)$$

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

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

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

QED

IV. Simulation Results

In this section, two numerical examples are presented that demonstrate the potential of the HNN-based parameter estimation method for time-varying and nonlinear systems.

A. Case 1

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

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

where $a = t - 5$ and $b = -8$. To identify parameter a in this nonlinear dynamic system, the linearization process is used. Because its equilibrium is at 0 when unforced, its linearization form is

$$\dot{\boldsymbol{x}} = a\boldsymbol{x} + bu \tag{68}$$

Then, we need to check whether Eq. (67) can be linearized to Eq. (68). Because

$$\lim_{\|x\| \to 0} \frac{\|\sin(a\boldsymbol{x}) - a\boldsymbol{x}\|}{\|\boldsymbol{x}\|} = 0 \tag{69}$$

With reference to Definition 1, it can be concluded that Eq. (68) is indeed a linearization of Eq. (67). 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 earlier, we can compute the values of parameter a and b. Histories of a and b calculations with time are presented in Figs. 1 and 2. It can be observed that both a and b approach their correct values very quickly and that the time-varying a is captured very well. These are shown in Figs. 1–3. The state history is presented in Fig. 3.

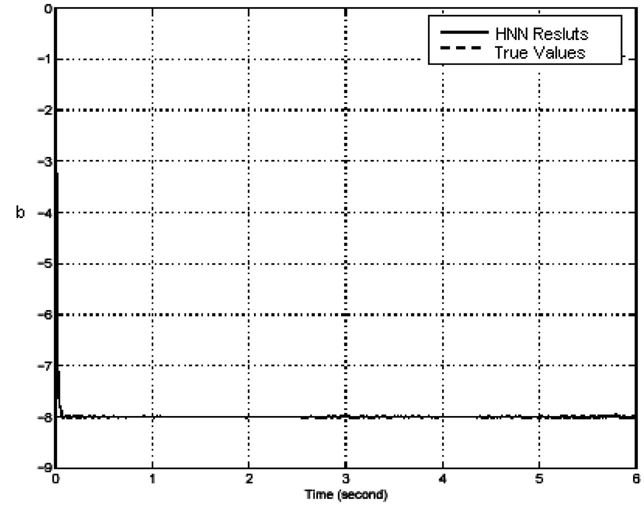

Fig. 2 History of parameter *b* (case 1).

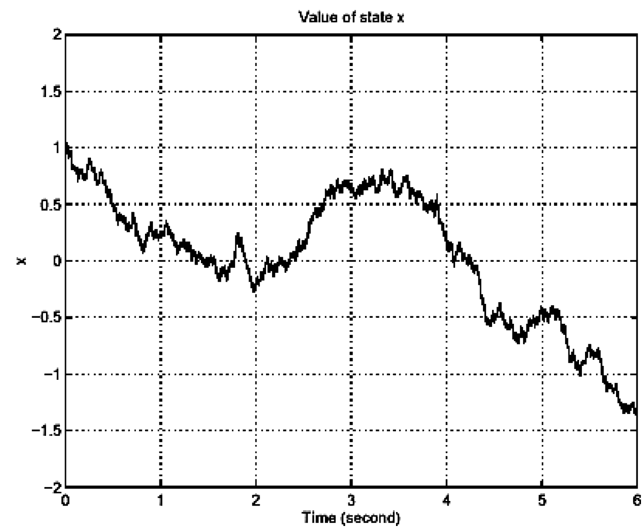

Fig. 3 State trajectory (case 1).

B. Case 2

Now a more difficult practical nonlinear higher dimensional problem in aircraft dynamics is considered.[14] The nonlinear estimation concept developed in Secs. III.B and III.C is used 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.

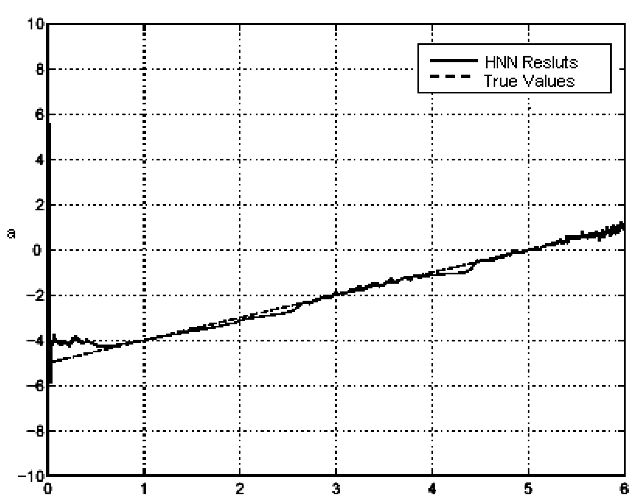

Fig. 1 History of parameter *a* (case 1).

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} \dot{v} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \\ \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = A\boldsymbol{x} + B\boldsymbol{u} + F(\boldsymbol{x})$$

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

where

$A =$

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

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

The problem is to identify the unknown matrices A, B, and C. In accordance with the method outlined in Sec. III, set the weight W and bias I with Eqs. (36) and (37) as

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

$$I = -\frac{1}{T}\int_0^T \left[\left(i_1G_1^T\right) \quad \left(i_2G_2^T\right) \quad \dots \quad \left(i_9G_9^T\right) \quad \left(i_1u^T\right) \quad \left(i_2u^T\right) \quad \dots \quad \left(i_9u^T\right)\right]_{175\times 1}^T \mathrm{d}t \tag{74}$$

$$G_1 = \begin{bmatrix} x_1 & x_2 & \dots & x_9 & 0 & 0 \end{bmatrix}^T, \qquad G_2 = \begin{bmatrix} x_1 & x_2 & \dots & x_9 & 0 & 0 \end{bmatrix}^T, \qquad G_3 = \begin{bmatrix} x_1 & x_2 & \dots & x_9 & x_6x_7 & x_7^2 - x_6^2 \end{bmatrix}^T$$

$$G_4 = \begin{bmatrix} x_1 & x_2 & \dots & x_9 & 0 & 0 \end{bmatrix}^T, \qquad G_5 = \begin{bmatrix} x_1 & x_2 & \dots & x_9 & 0 & 0 \end{bmatrix}^T, \qquad G_6 = \begin{bmatrix} x_1 & x_2 & \dots & x_9 & x_3x_6 & x_3x_7 \end{bmatrix}^T$$

$$G_7 = \begin{bmatrix} x_1 & x_2 & \dots & x_9 & x_3x_6 & -x_3x_7 \end{bmatrix}^T, \qquad G_8 = \begin{bmatrix} x_1 & x_2 & \dots & x_9 & 0 & \end{bmatrix}^T, \qquad G_9 = \begin{bmatrix} x_1 & x_2 & \dots & x_9 & 0 & \end{bmatrix}^T$$

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

$B =$

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

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

where

$$i_1 = \dot{x}_1, \qquad i_2 = \dot{x}_2 + x_6\cos x_2 \tan x_5 + x_7 \sin x_2 \tan x_5$$

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

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

$$i_5 = \dot{x}_5 - x_6\sin x_2 + x_7\cos x_2, \qquad i_6 = \dot{x}_6$$

$$i_7 = \dot{x}_7, \qquad i_8 = \dot{x}_8 - x_3\tan x_4 \sin x_8 - x_7 \tan x_4 \cos x_8$$

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

In the numerical experiments, it was observed that the estimated values are nearly the same as the true parameter values (differences lie between 0–3%). Figures 4–29 shows the plots of some parameters in A and B. For some of them that do not show full convergence, the last values at the end of the first iteration are used as initial values for the next iteration. All parameters converge after the third pass to steady-state values.

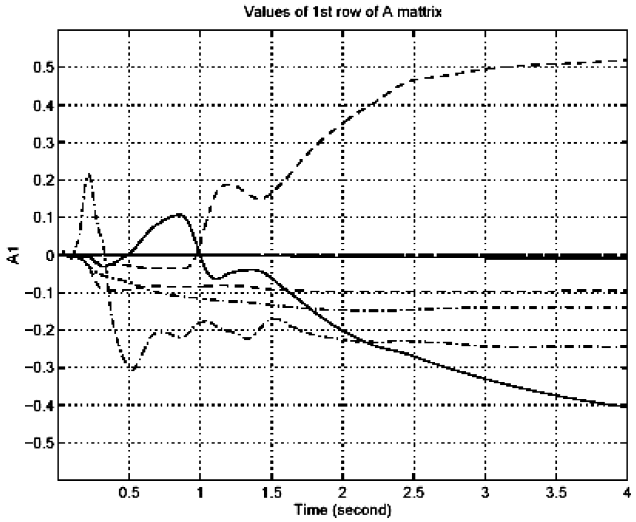

Fig. 4 Values of $A1$ from first iteration.

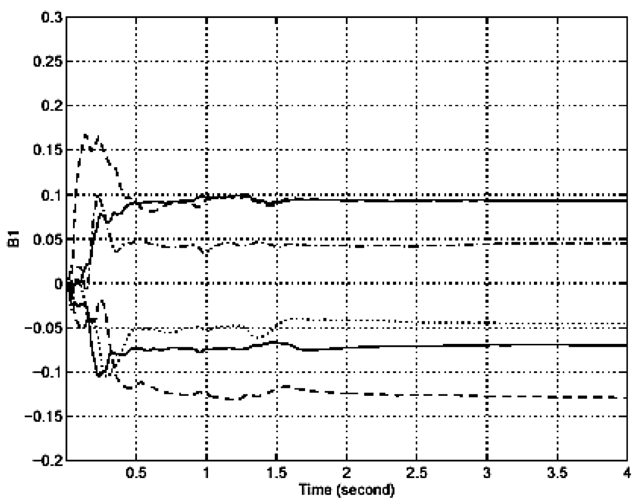

Fig. 5 Values of $B1$ from first iteration.

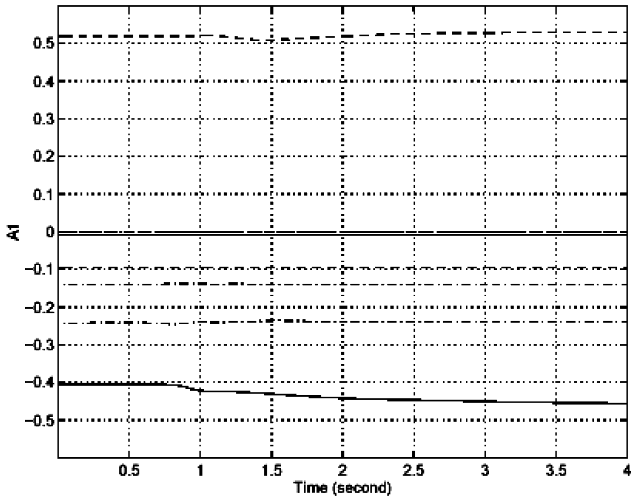

Fig. 6 Values of $A1$ from second iteration.

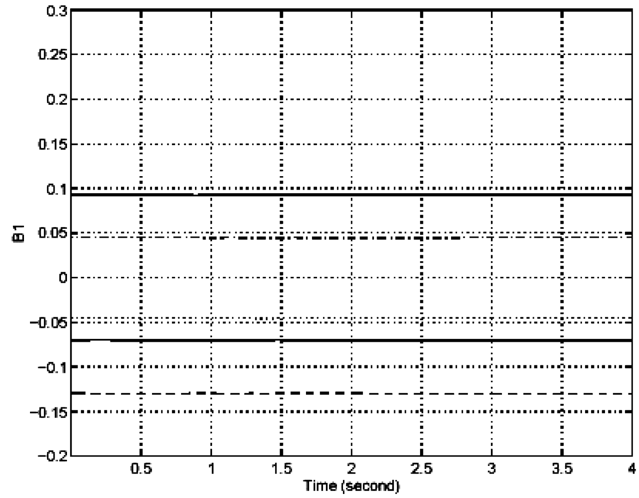

Fig. 7 Values of $B1$ from second iteration.

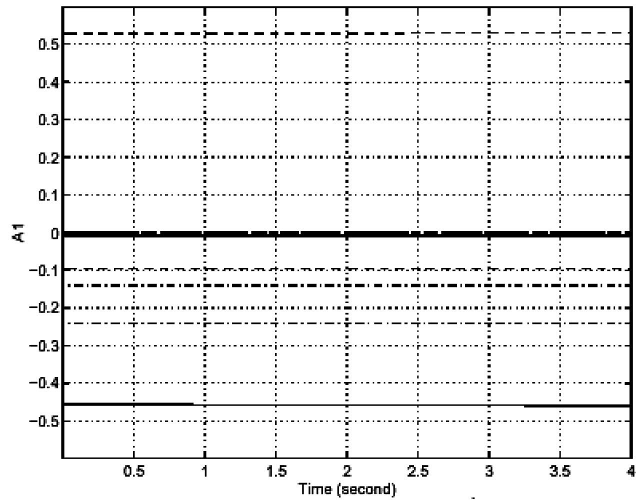

Fig. 8 Values of $A1$ from third iteration.

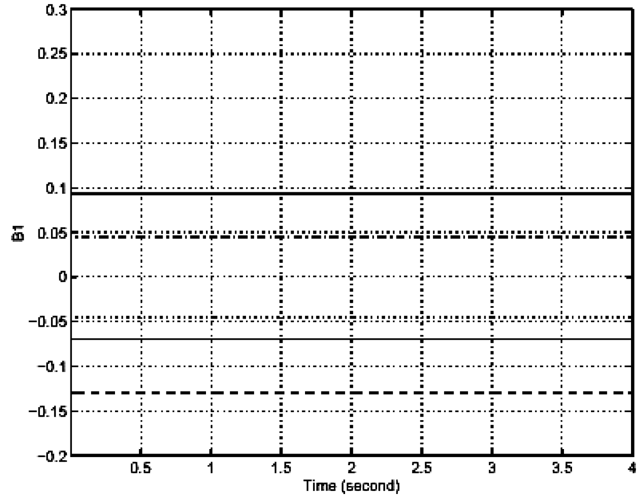

Fig. 9 Values of $B1$ from third iteration.

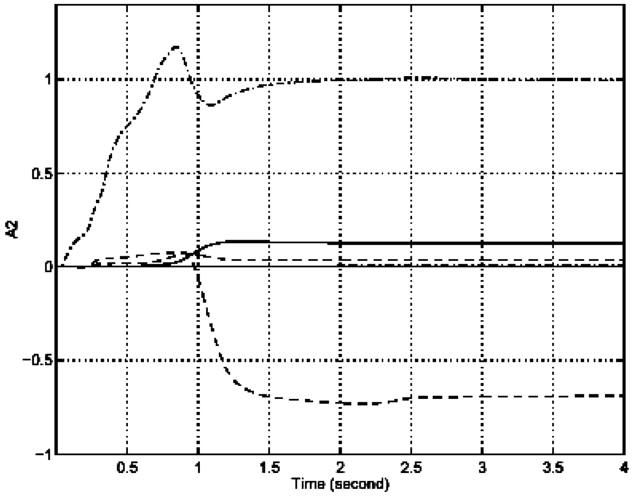

Fig. 10 Values of $A2$ from first iteration.

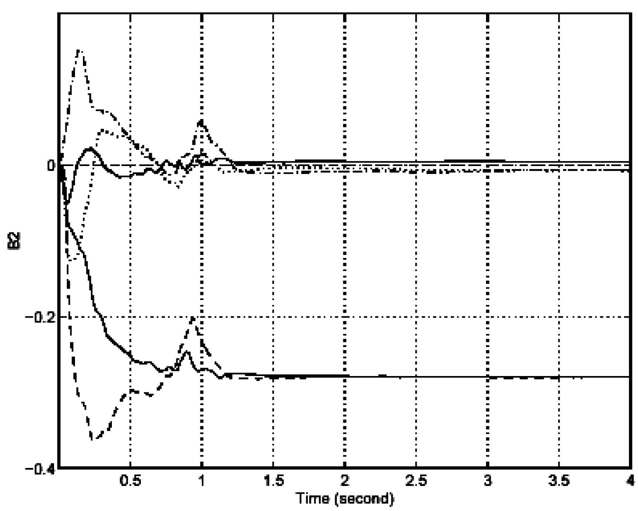

Fig. 11 Values of $B2$ from first iteration.

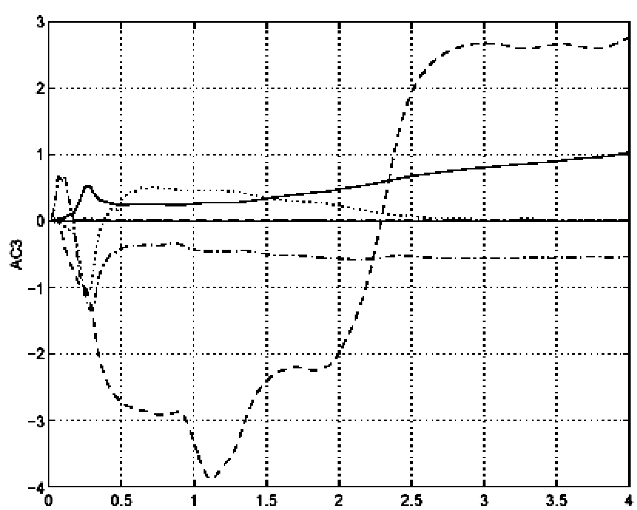

Fig. 12 Values of $A3$ and $C3$ of the first iteration.

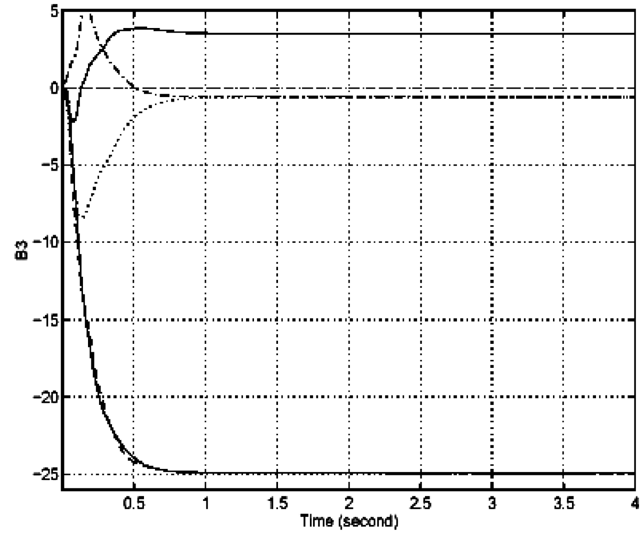

Fig. 13 Values of $B3$ from first iteration.

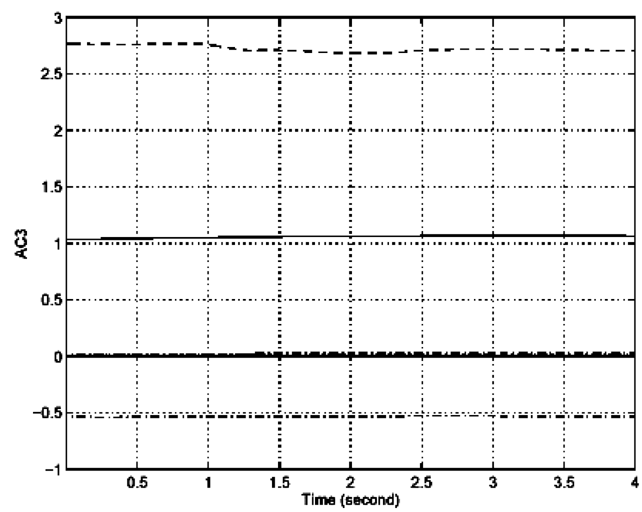

Fig. 14 Values of $A3$ and $C3$ of the second iteration.

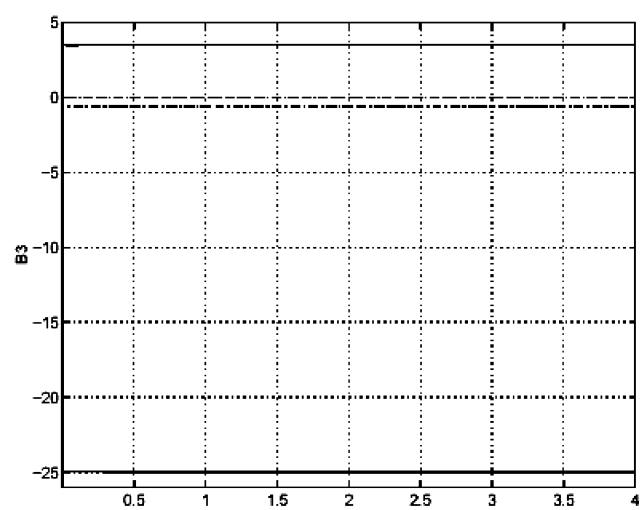

Fig. 15 Values of $B3$ from second iteration.

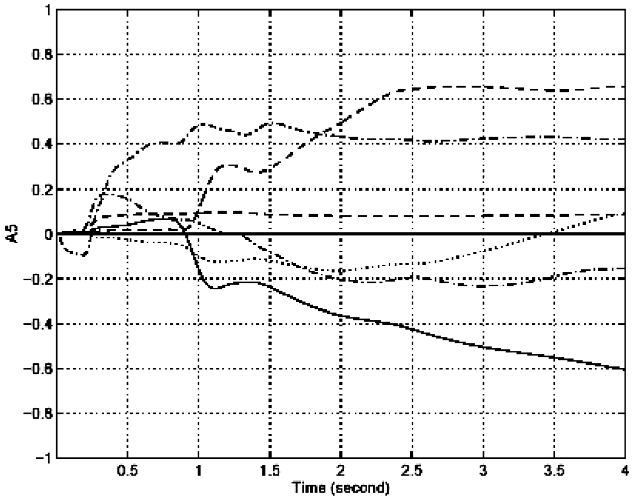

Fig. 16 Values of $A5$ from first iteration.

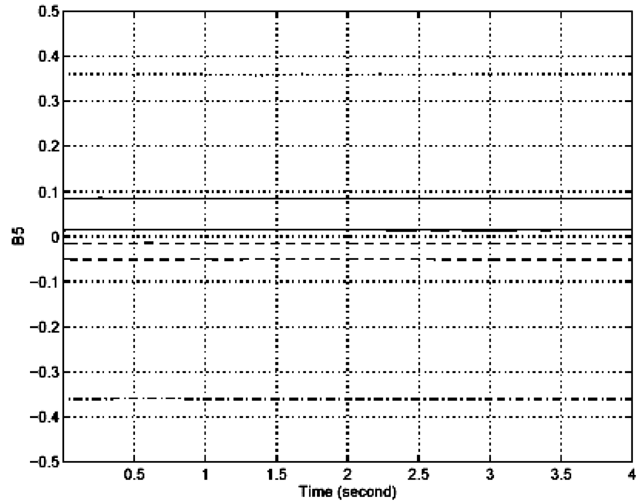

Fig. 19 Values of $B5$, second iteration.

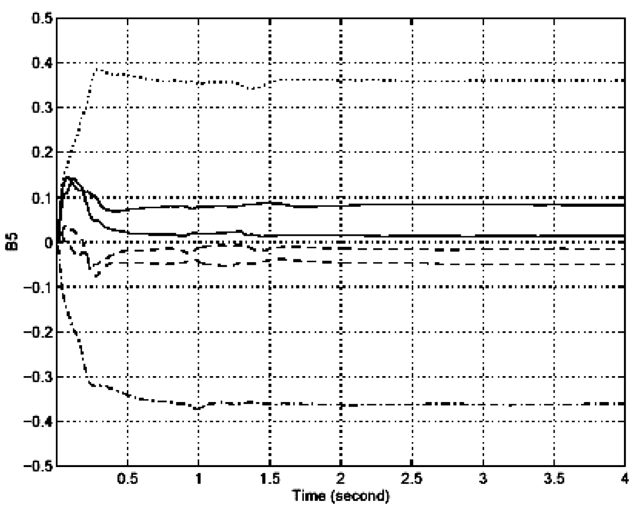

Fig. 17 Values of $B5$, first iteration.

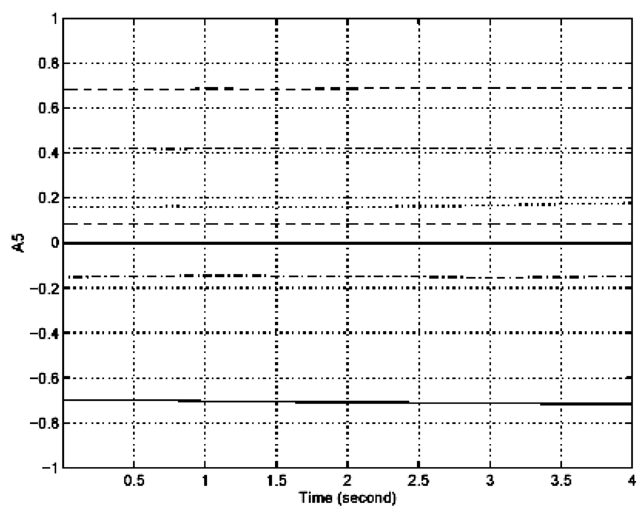

Fig. 20 Values of $A5$, third iteration.

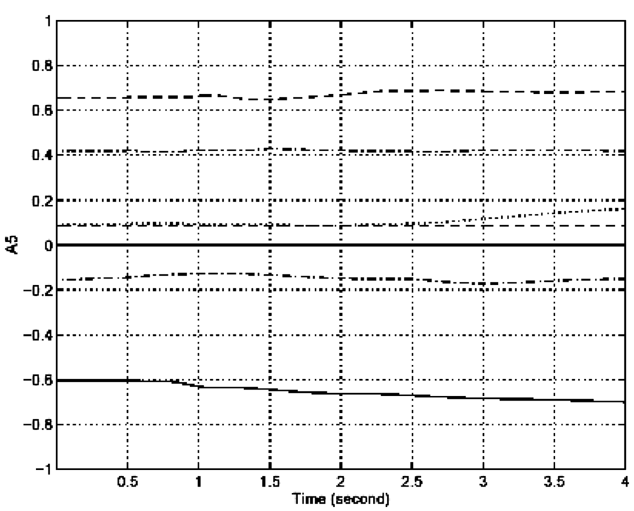

Fig. 18 Values of $A5$, second iteration.

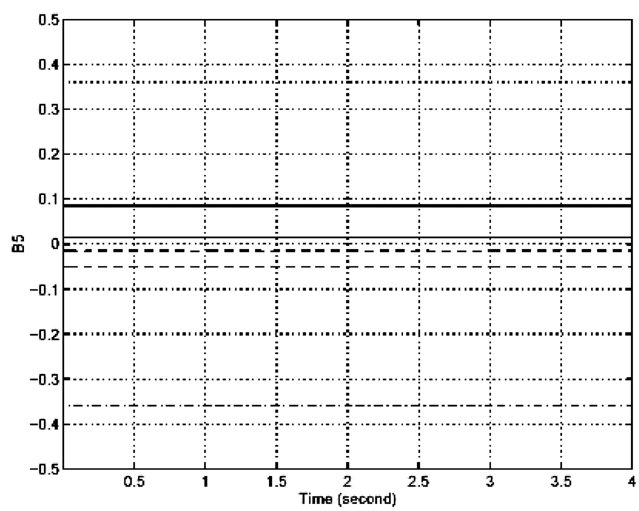

Fig. 21 Values of $B5$, third iteration.

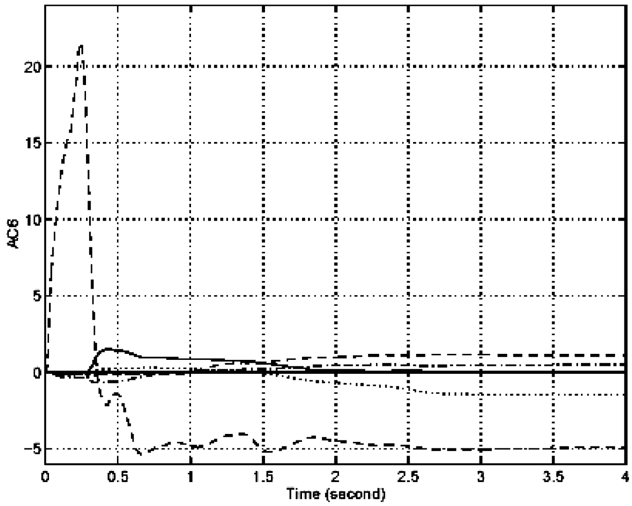

Fig. 22 Values of $A6$ and $C6$ of the first iteration.

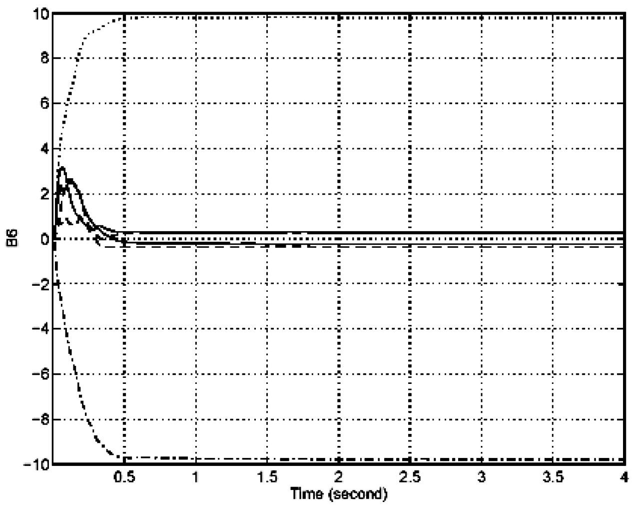

Fig. 23 Values of $B6$ from first iteration.

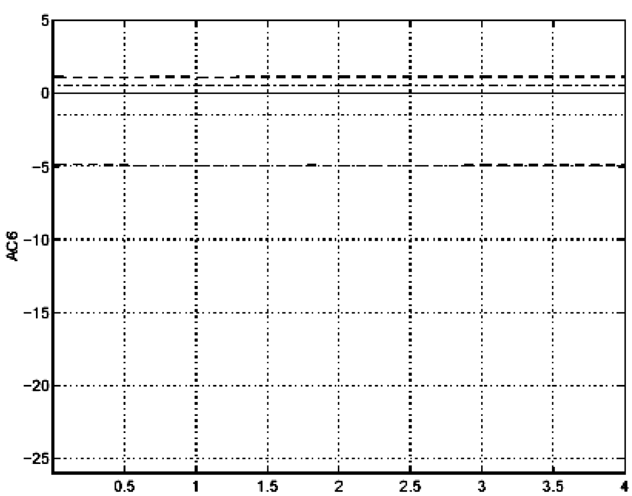

Fig. 24 Values of $A6$ and $C6$ of the second iteration.

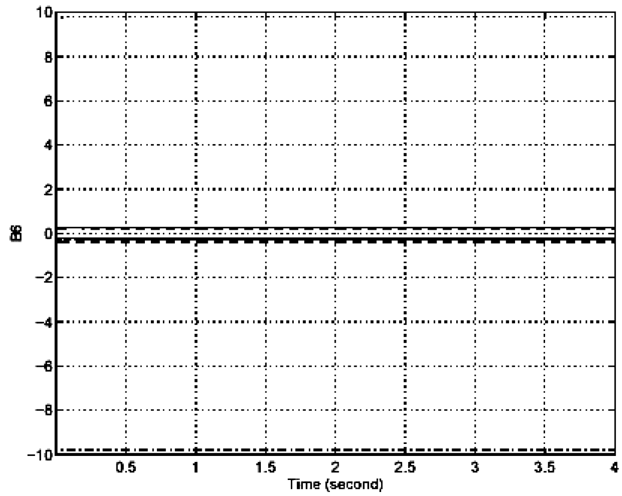

Fig. 25 Values of $B6$ from second iteration.

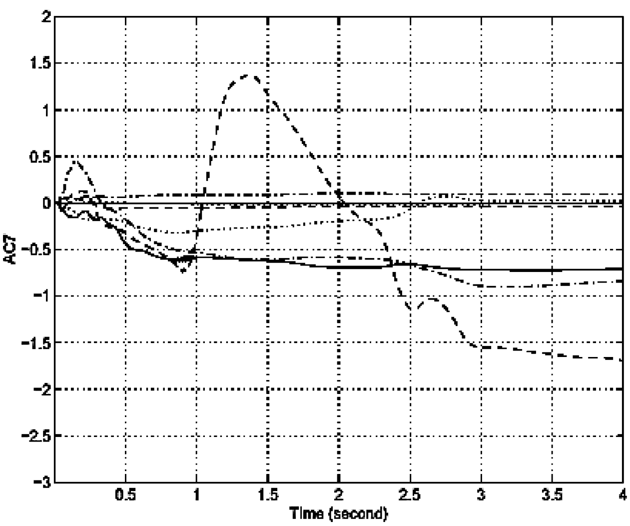

Fig. 26 Values of $A7$ and $C7$ of the first iteration.

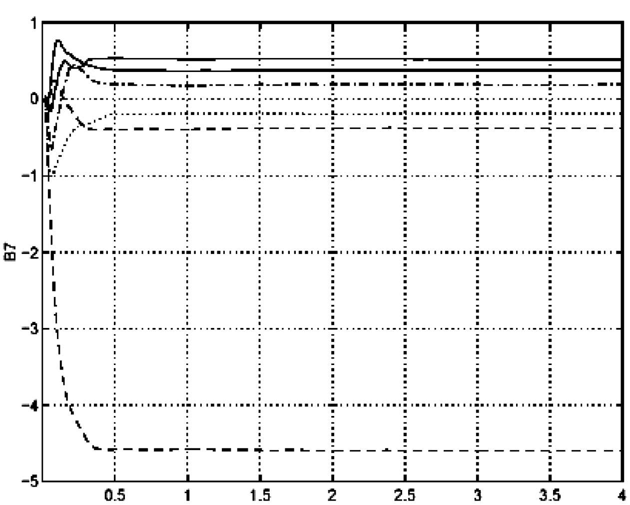

Fig. 27 Values of $B7$ from first iteration.

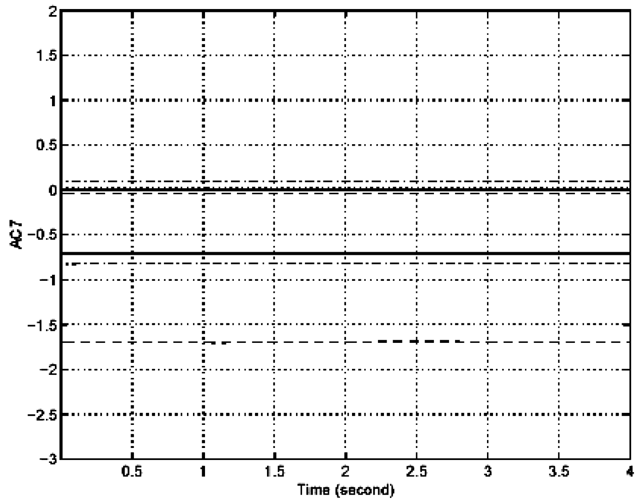

Fig. 28 Values of $A7$ and $C7$ of the second iteration.

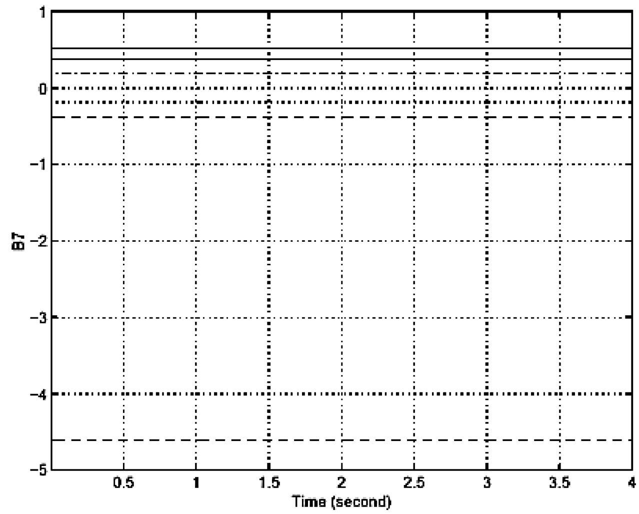

Fig. 29 Values of $B7$ from second iteration.

The converged values of A, B, and C are given hereafter:

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

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

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

Let $A1$ denote the estimates of first row in the A matrix, $B1$ denotes the estimates of the first row in the B matrix, and so on. The histories of the parameter values are presented vs 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. To reach convergence, the simulation shows that the neural network parameters ρ and λ can be chosen arbitrarily, as long as $\rho > v_j$. Usually, it was found that larger values of ρ and λ led to faster convergence. Sometimes, however, they exhibited instability.

V. Conclusions

A method to estimate parameters of model-based time-varying and time-invariant nonlinear systems has been presented. Effectiveness of the method has been demonstrated through applications. Because of the massively parallel nature of neural networks, this HNN-based technique is a good candidate for online parameter estimation. Note, however, that the underlying model should accurately reflect the physics of the system. Furthermore, a state estimator may need to be added in the parameter estimation loop if the measurements are noisy or all of the states are not measured.

References

[1]Hopfield, J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proceedings of the National Academy of Science*, Vol. 79, 1982, pp. 2554–2558.

[2]Hopfield, J., "Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons," *Proceedings of the National Academy of Science*, Vol. 81, 1984, pp. 3088–3092.

[3]Hopfield, J., and Tank, D., "Neuroal Computation of Decisions In Optimization Problems," *Biological Cybernetics*, Vol. 52, 1985, pp. 141–152.

[4]Cetinkunt, S., and Hsin-Tan, C., "Estimation of Model Parameters of Linear Structural Systems Using Hopfield Networks," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 3, 1993, pp. 621–624.

[5]Shen, J., and Balakrishnan, S. N., "A Class of Modified Hopfield Networks for Aircraft Identification and Control," AIAA Paper 96-3428, 1997.

[6]Jagannathan, S., and Lewis, F. L., "Identification of Nonlinear Dynamic Systems Using Multilayered Neural Networks," Vol. 132, No. 12, *Automatica*, 1996, pp. 1707–1712.

[7]Bruyne, F. D., "A Closed-Loop Relevant Identification of Input-Output and Noise Dynamics," *Proceedings of the American Control Conference*, American Automatic Control Council, IEEE, Piscataway, NJ, 1998, pp. 3199–3203.

[8]Habib, S., and Zaghloul, M., "Concurrent Identification and Control of System by Applying Standardized Neurocontroller and Neuroplant," AIAA Paper 96-3773, 1996.

[9]Cichocki, A., and Unbehauen, R., "Neural Networks for Solving Systems of Linear Equations and Related Problems," *IEEE Transactions on Circuits and Systems*, Vol. 39, No. 2, 1992, pp. 124–138.

[10]Raol, J. R., "Neural Network Based Parameter Estimation of Unstable Aerospace Dynamic Systems," *IEE Proceedings on Control Theory Application*, Vol. 141, No. 6, 1994, pp. 114–118.

[11]Raol, J. R., "Parameter Estimation of State Space Models by Recurrent Neural Networks," *IEE Proceedings on Control Theory Application*, Vol. 142, No. 2, 1995, pp. 385–388.

[12]Raol, J. R., "Neural Network Architectures for Parameter Estimation of Dynamic Systems," *IEE Proceedings on Control Theory Application*, Vol. 143, No. 4, 1996, pp. 387–394.

[13]Lyshevski, S., "Nonlinear Mapping Technology in System Identification," *Proceedings of the American Control Conference*, American Automatic Control Council, IEEE, Piscataway, NJ, 1997, pp. 605, 606.

[14]Lyshevski, S., "Nonlinear Identification of Control System," *Proceedings of the American Control Conference*, American Automatic Control Council, IEEE, Piscataway, NJ, 1998, pp. 2366–2370.

[15]Amin, S. M., Volker, G., and Rodin, E. Y., "System Identification via Artificial Neural Networks: Applications to On-line Aircraft Parameter Estimation," AIAA Paper 97-5612, 1997.

[16]Kim, Y. H., Liews, F. L., and Chaouki, T. A., "A Dynamic Recurrent Neural Network-Based Adaptive Observer for a Class of Nonlinear Systems," *Automatica*, Vol. 133, No. 8, 1997, pp. 1539–1543.

[17]Kambhampati, C., Craddock, R., Tham, M., and Warwick, K., "An Inverting Recurrent Neural Network for Internal Model Control of Nonlinear Systems," *Proceedings of the American Control Conference*, American Automatic Control Council, IEEE, Piscataway, NJ, 1998, pp. 975–979.

[18]Stubberud, A., Wabgaonkar, H., and Stubberud, S., "A Neural-Network-Based System Identification Technique," *Proceedings of IEEE Conference on Decision and Control*, IEEE, Piscataway, NJ, 1991, pp. 869, 870.

[19]Vidyasagar, M., *Nonlinear System Analysis*, Prentice–Hall, Upper Saddle River, NJ, 1993.