Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2010, Article ID 415895, 14 pages
doi:10.1155/2010/415895

Research Article

A New Robust Training Law for Dynamic Neural Networks with External Disturbance: An LMI Approach

Choon Ki Ahn

Department of Automotive Engineering, Seoul National University of Science and Technology,
172 Gongneung 2-dong, Nowon-gu, Seoul 139-743, Republic of Korea

Correspondence should be addressed to Choon Ki Ahn, hironaka@snut.ac.kr

Received 8 September 2010; Accepted 16 December 2010

Academic Editor: Chang-hong Wang

A new robust training law, which is called an input/output-to-state stable training law (IOSSTL), is proposed for dynamic neural networks with external disturbance. Based on linear matrix inequality (LMI) formulation, the IOSSTL is presented to not only guarantee exponential stability but also reduce the effect of an external disturbance. It is shown that the IOSSTL can be obtained by solving the LMI, which can be easily facilitated by using some standard numerical packages. Numerical examples are presented to demonstrate the validity of the proposed IOSSTL.

1. Introduction

The past few years have witnessed a remarkable progress in the research of neural networks due to their advantages of parallel computation, learning ability, function approximation, and fault tolerance. Many exciting applications have been developed in signal processing, control, combinatorial optimization, pattern recognition, and other areas. It has been shown that these applications greatly depend on the stability and performance of the underlying neural networks [1].

Stability of neural networks is a prerequisite for their successful applications as either associative memories or optimization solvers. Stability problems in neural networks can be classified into two main categories: stability of neural networks [2–9] and stability of learning algorithms [10–16]. In this paper, we focus on deriving a new learning algorithm for neural networks. Stable learning algorithms can be developed by analyzing identification or tracking errors in neural networks. In [11], the stability conditions of learning laws in a situation where neural networks were used to identify and control a nonlinear system were studied. In [17], the dynamic backpropagation was modified with NL_q stability constraints. In [13, 15], passivity-based learning laws were proposed for neural networks. Since neural networks cannot match the unknown nonlinear systems exactly, extensive modifications should be made either to the normal gradient algorithm or to the backpropagation algorithm [10, 11, 14, 16]. Despite these advances in neural network learning, most research results were restricted to learning algorithms for neural networks without external disturbance, but, in real physical systems, one is faced with model uncertainties and a lack of statistical information on the signals. Thus, knowledge of the learning algorithm for neural networks with external disturbance is of considerable practical importance. Unfortunately, with the existing works, it is impossible to design a learning algorithm for neural networks with external disturbance. In this paper, we propose a new robust learning algorithm for neural networks with external disturbance.

The input/output-to-state stability (IOSS) approach [18, 19] is an effective tool to analyze the stability of physical nonlinear systems affected by disturbances. It can deal with nonlinear systems using only the general characteristics of the input-output dynamics, and it offers elegant solutions for the proof of stability. The IOSS framework is a promising approach to the stability analysis of neural networks because it can lead to general conclusions on stability using only input-output characteristics. It has been widely accepted as an important concept in control engineering, and several research results have been reported in recent years [18–23]. A natural question arises: can we obtain an IOSS based robust training algorithm for dynamic neural networks with external disturbance? This paper gives an answer for this question. To the best of our knowledge, however, for the IOSS training law of dynamic neural networks with external disturbance, there is no result in the literature so far, which still remains open and challenging.

In this paper, for the first time, the IOSS approach is used to create a new robust training law for dynamic neural networks with external disturbance. This training law is called an IOSS training law (IOSSTL). The IOSSTL is a new contribution to the topic of neural network learning. A sufficient condition for the existence of the IOSSTL is presented such that neural networks are exponentially stable and input/output-to-state stable for any bounded disturbance. Based on linear matrix inequality (LMI) formulation, the design of the proposed training law can be realized by solving the LMI, which can be facilitated readily via standard numerical algorithms [24, 25]. The presented result of this paper opens a new path for application of the IOSS technique to weight learning of dynamic neural networks. It is expected that the proposed scheme can be extended to study learning algorithms for neural networks with timedelay.

This paper is organized as follows. In Section 2, we formulate the problem. In Section 3, an LMI problem for the IOSSTL of dynamic neural networks is proposed. In Section 4, numerical examples are given, and finally, conclusions are presented in Section 5.

2. Problem Formulation

Consider the following differential equation:

$$\begin{aligned} \dot{X}(t) &= F(X(t), U(t)), \\ Y(t) &= H(X(t), U(t)), \end{aligned} \tag{2.1}$$

where $X(t) \in R^n$ is the state variable, $U(t) \in R^m$ is the external input, and $Y(t) \in R^k$ is the output variable. $F : R^n \times R^m \to R^n$ and $H : R^n \times R^m \to R^k$ are continuously differentiable and satisfy $F(0,0) = 0$. We first recall the following definitions.

Definition 2.1. A function $\gamma : R_{\geq 0} \to R_{\geq 0}$ is a $\mathcal{K}$ function if it is continuous and strictly increasing and if $\gamma(0) = 0$. It is a $\mathcal{K}_\infty$ function if it is a $\mathcal{K}$ function and $\gamma(s) \to \infty$ as $s \to \infty$. A function $\beta : R_{\geq 0} \times R_{\geq 0} \to R_{\geq 0}$ is a $\mathcal{KL}$ function if, for each fixed $t \geq 0$, the function $\beta(\cdot, t)$ is a $\mathcal{K}$ function and for each fixed $s \geq 0$, the function $\beta(s, \cdot)$ is decreasing and $\beta(s,t) \to 0$ as $t \to \infty$.

Definition 2.2. The system (2.1) is said to be input/output-to-state stable if there exist $\mathcal{K}$ functions $\gamma_1(s)$, $\gamma_2(s)$ and a $\mathcal{K}L$ function $\beta(s,t)$, such that, for each input $U(t)$, each output $Y(t)$, and each initial state $X(0)$, it holds that

$$\|X(t)\| \leq \max\left\{\beta(\|X(0)\|, t), \gamma_1\left(\sup_{0\leq\tau\leq t}\|U(\tau)\|\right), \gamma_2\left(\sup_{0\leq\tau\leq t}\|Y(\tau)\|\right)\right\} \tag{2.2}$$

for each $t \geq 0$.

Now we introduce a useful result that is employed for obtaining the IOSSTL.

Lemma 2.3 (see [18, 19]). *A continuous function $V(\cdot) : R^n \to R_{\geq 0}$ is called an IOSS-Lyapunov function for the system* (2.1) *if there exist $\mathcal{K}_\infty$ functions α_1, α_2, α_3, α_4, and α_5 such that*

$$\alpha_1(\|X(t)\|) \leq V(X(t)) \leq \alpha_2(\|X(t)\|) \tag{2.3}$$

for any $X(t) \in R^n$ and

$$\dot{V}(X(t)) \leq -\alpha_3(\|X(t)\|) + \alpha_4(\|U(t)\|) + \alpha_5(\|Y(t)\|) \tag{2.4}$$

for any $X(t) \in R^n$, any $U(t) \in R^m$, and any $Y(t) \in R^k$. Then the system (2.1) *is input/output-to-state stable if and only if it admits an IOSS-Lyapunov function.*

Consider the following dynamic neural network:

$$\dot{x}(t) = Ax(t) + W(t)\theta(x(t)) + \overline{W}(t)\phi(x(t))u(t) + Gd(t), \tag{2.5}$$

$$y(t) = Cx(t) + Dd(t), \tag{2.6}$$

where $x(t) = [x_1(t) \ \cdots \ x_n(t)]^T \in R^n$ is the state vector, $u(t) = [u_1(t) \ \cdots \ u_m(t)]^T \in R^m$ is the bounded control input, $y(t) = [y_1(t) \ \cdots \ y_p(t)]^T \in R^p$ is the output vector, $d(t) = [d_1(t) \ \cdots \ d_k(t)]^T \in R^k$ is the disturbance vector, $A = \text{diag}\{-a_1, \ldots, -a_n\} \in R^{n\times n}$ ($a_i > 0$, $i = 1, \ldots, n$) is the self-feedback matrix, $W(t) \in R^{n\times p}$ and $\overline{W}(t) \in R^{n\times p}$ are the weight matrices, $\theta(x) = [\theta_1(x), \ldots, \theta_p(x)]^T : R^n \to R^p$ is the nonlinear vector field, and $\phi(x) = [\phi_{ij}(x)] : R^n \to R^{p\times m}$ is the nonlinear matrix function, and $C \in R^{p\times n}$, $D \in R^{p\times k}$, and $G \in R^{n\times k}$ are known constant matrices. The element functions $\theta_i(x)$ and $\phi_{ij}(x)$ ($i = 1, \ldots, p$, $j = 1, \ldots, m$) are usually selected as sigmoid functions.

Remark 2.4. It can be seen that the Hopfield model [26] is the special case of the neural network (2.5) with $A = \operatorname{diag}\{a_i\}$, $a_i = -1/R_iC_i$, $R_i > 0$, and $C_i > 0$. R_i and C_i are the resistance and capacitance at the ith node of the network, respectively.

The purpose of this study is to derive a new training law such that the neural network (2.5)-(2.6) with $d(t) \neq 0$ is input/output-to-state stable and exponentially stable for $d(t) = 0$.

3. Main Results

In this section, we design a new input/output-to-state stable training law for dynamic neural networks with external disturbance. The following theorem gives an LMI condition for the existence of the desired training law.

Theorem 3.1. *Assume that there exist $P = P^T > 0$, $S_1 = S_1^T > 0$, $S_2 = S_2^T > 0$, $S_3 = S_3^T > 0$, and Y such that*

$$\begin{bmatrix} A^TP + Y^T + PA + Y - C^TS_2C + S_1 & PG - C^TS_2D \\ G^TP - D^TS_2C & -D^TS_2D - S_3 \end{bmatrix} < 0. \tag{3.1}$$

If the weight matrices $W(t)$ and $\overline{W}(t)$ are updated as

$$\begin{gathered} W(t) = \begin{cases} \dfrac{P^{-1}Yx(t)}{\|\theta(x(t))\|^2 + \|\phi(x(t))u(t)\|^2}\theta^T(x(t)), & \|\theta(x(t))\|^2 + \|\phi(x(t))u(t)\|^2 \neq 0, \\ 0, & \|\theta(x(t))\|^2 + \|\phi(x(t))u(t)\|^2 = 0, \end{cases} \\ \overline{W}(t) = \begin{cases} \dfrac{P^{-1}Yx(t)}{\|\theta(x(t))\|^2\|\phi(x(t))u(t)\|^2}u^T(t)\phi^T(x(t)), & \|\theta(x(t))\|^2 + \|\phi(x(t))u(t)\|^2 \neq 0, \\ 0, & \|\theta(x(t))\|^2 + \|\phi(x(t))u(t)\|^2 = 0, \end{cases} \end{gathered} \tag{3.2}$$

then the neural network (2.5)-(2.6) *is input/output-to-state stable.*

Proof. The neural network (2.5) can be represented by

$$\dot{x}(t) = Ax(t) + \begin{bmatrix} W(t) & \overline{W}(t) \end{bmatrix} \begin{bmatrix} \theta(x(t)) \\ \phi(x(t))u(t) \end{bmatrix} + Gd(t). \tag{3.3}$$

In order to guarantee the IOSS, the following relation must be satisfied:

$$Kx(t) = \begin{bmatrix} W(t) & \overline{W}(t) \end{bmatrix} \begin{bmatrix} \theta(x(t)) \\ \phi(x(t))u(t) \end{bmatrix}, \tag{3.4}$$

where $K \in R^{n\times n}$ is the gain matrix of the IOSSTL. Then we obtain

$$\dot{x}(t) = (A + K)x(t) + Gd(t). \tag{3.5}$$

One of the possible weight selections $[\, W(t)\ \overline{W}(t)\,]$ to fulfil (3.4) (perhaps excepting a subspace of a smaller dimension) is given by

$$\left[W(t)\ \ \overline{W}(t)\right] = Kx(t)\begin{bmatrix} \theta(x(t)) \\ \phi(x(t))u(t) \end{bmatrix}^{+}, \tag{3.6}$$

where $[\cdot]^{+}$ stands for the pseudoinverse matrix in the Moore-Penrose sense [27]. This training law is just an algebraic relation depending on $x(t)$ and $u(t)$, which can be evaluated directly. Taking into account that

$$x^{+} = \begin{cases} \dfrac{x^{T}}{\|x\|^{2}}, & x \neq 0, \\ 0, & x = 0, \end{cases} \tag{3.7}$$

the training law (3.6), see [27], can be rewritten as follows:

$$W(t) = \begin{cases} \dfrac{Kx(t)}{\|\theta(x(t))\|^{2} + \left\|\phi(x(t))u(t)\right\|^{2}}\theta^{T}(x(t)), & \|\theta(x(t))\|^{2} + \left\|\phi(x(t))u(t)\right\|^{2} \neq 0, \\ 0, & \|\theta(x(t))\|^{2} + \left\|\phi(x(t))u(t)\right\|^{2} = 0, \end{cases}$$
$$\overline{W}(t) = \begin{cases} \dfrac{Kx(t)}{\|\theta(x(t))\|^{2} + \left\|\phi(x(t))u(t)\right\|^{2}}u^{T}(t)\phi^{T}(x(t)), & \|\theta(x(t))\|^{2} + \left\|\phi(x(t))u(t)\right\|^{2} \neq 0, \\ 0, & \|\theta(x(t))\|^{2} + \left\|\phi(x(t))u(t)\right\|^{2} = 0. \end{cases} \tag{3.8}$$

To obtain the gain matrix K of the IOSSTL, consider a Lyapunov function $V(t) = x^{T}(t)Px(t)$. Note that $V(t)$ satisfies the following Rayleigh inequality [28]:

$$\lambda_{\min}(P)\|x(t)\|^{2} \leq V(t) \leq \lambda_{\max}(P)\|x(t)\|^{2}, \tag{3.9}$$

where $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ are the maximum and minimum eigenvalues of the matrix. The time derivative of $V(t)$ along with the trajectory of (3.5) is

$$\dot{V}(t) = x^{T}(t)\left[A^{T}P + K^{T}P + PA + PK\right]x(t) + x^{T}(t)PGd(t) + d^{T}(t)G^{T}Px(t). \tag{3.10}$$

Adding and subtracting $[Cx(t)+Dd(t)]^T S_2 [Cx(t)+Dd(t)]$, we obtain

$$\begin{aligned}\dot{V}(t) &= x^T(t)\left[A^TP + K^TP + PA + PK - C^TS_2C\right]x(t) - d^T(t)D^TS_2Dd(t)\\ &\quad + x^T(t)\left[PG - C^TS_2D\right]d(t) + d^T(t)\left[G^TP - D^TS_2C\right]x(t) + y^T(t)S_2y(t)\\ &= \begin{bmatrix} x(t)\\ d(t)\end{bmatrix}^T \begin{bmatrix} (1,1) & PG - C^TS_2D\\ G^TP - D^TS_2C & -D^TS_2D - S_3\end{bmatrix}\begin{bmatrix} x(t)\\ d(t)\end{bmatrix} - x^T(t)S_1x(t)\\ &\quad + y^T(t)S_2y(t) + d^T(t)S_3d(t),\end{aligned} \tag{3.11}$$

where

$$(1,1) = A^TP + K^TP + PA + PK - C^TS_2C + S_1. \tag{3.12}$$

If the following matrix inequality is satisfied

$$\begin{bmatrix} (1,1) & PG - C^TS_2D\\ G^TP - D^TS_2C & -D^TS_2D - S_3\end{bmatrix} < 0, \tag{3.13}$$

we have

$$\dot{V}(t) < -x^T(t)S_1x(t) + y^T(t)S_2y(t) + d^T(t)S_3d(t) \tag{3.14}$$

$$\le -\lambda_{\min}(S_1)\|x(t)\|^2 + \lambda_{\max}(S_2)\left\|y(t)\right\|^2 + \lambda_{\max}(S_3)\|d(t)\|^2. \tag{3.15}$$

Define functions $\alpha_1(r)$, $\alpha_2(r)$, $\alpha_3(r)$, $\alpha_4(r)$, and $\alpha_5(r)$ as

$$\begin{aligned}\alpha_1(r) &\triangleq \lambda_{\min}(P)r^2,\\ \alpha_2(r) &\triangleq \lambda_{\max}(P)r^2,\\ \alpha_3(r) &\triangleq \lambda_{\min}(S_1)r^2,\\ \alpha_4(r) &\triangleq \lambda_{\max}(S_2)r^2,\\ \alpha_5(r) &\triangleq \lambda_{\max}(S_3)r^2.\end{aligned} \tag{3.16}$$

Note that $\alpha_1(r)$, $\alpha_2(r)$, $\alpha_3(r)$, $\alpha_4(r)$, and $\alpha_5(r)$ are $\mathcal{K}_\infty$ functions. From (3.9) and (3.15), we can obtain

$$\begin{gathered}\alpha_1(\|x(t)\|) \le V(t) \le \alpha_2(\|x(t)\|),\\ \dot{V}(t) \le -\alpha_3(\|x(t)\|) + \alpha_4\left(\left\|y(t)\right\|\right) + \alpha_5(\|d(t)\|).\end{gathered} \tag{3.17}$$

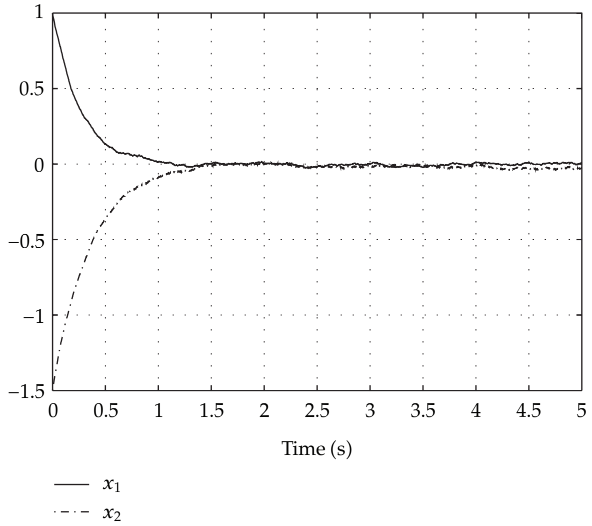

Figure 1: State trajectories.

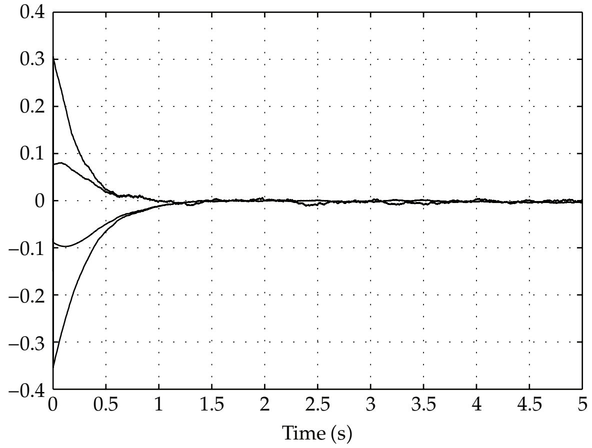

Figure 2: Elements of the weight matrix $W(t)$.

According to Lemma 2.3, we can conclude that $V(t)$ is an IOSS-Lyapunov function. Introducing a change of variable $PK = Y$, (3.13) is equivalently changed into the LMI (3.1). Then the gain matrix of the IOSSTL is given by $K = P^{-1}Y$. The training laws (3.8) are changed to (3.2). This completes the proof. □

Corollary 3.2. *Assume that there exist $P = P^T > 0$, $S_1 = S_1^T > 0$, $S_2 = S_2^T > 0$, $S_3 = S_3^T > 0$, and Y such that*

$$\begin{bmatrix} A^T P + Y^T + PA + Y - C^T S_2 C + S_1 & PG - C^T S_2 D \\ G^T P - D^T S_2 C & -D^T S_2 D - S_3 \end{bmatrix} < 0, \tag{3.18}$$

$$S_1 - C^T S_2 C > 0. \tag{3.19}$$

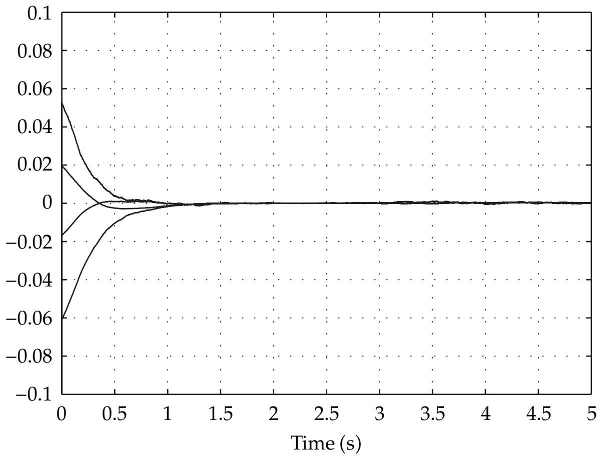

Figure 3: Elements of the weight matrix $\overline{W}(t)$.

If the weight matrices $W(t)$ *and* $\overline{W}(t)$ *are updated as* (3.2)*, the neural network* (2.5)-(2.6) *with* $d(t) \neq 0$ *is input/output-to-state stable and exponentially stable for* $d(t) = 0$*.*

Proof. From Theorem 3.1, the LMI (3.18) guarantees that the neural network (2.5)-(2.6) with $d(t) \neq 0$ is input/output-to-state stable. Next, we show that under the additional LMI condition (3.19), the neural network (2.5)-(2.6) with $d(t) = 0$ is exponentially stable. When $d(t) = 0$, from (3.14) and (2.6), we obtain

$$\begin{aligned} \dot{V}(t) &< -x^T(t)\left[S_1 - C^T S_2 C\right]x(t) \\ &\leq -\lambda_{\min}\left(S_1 - C^T S_2 C\right)\|x(t)\|^2 \\ &\leq -\frac{\lambda_{\min}(S_1 - C^T S_2 C)}{\lambda_{\max}(P)}V(t). \end{aligned} \tag{3.20}$$

From (3.20), we have

$$V(t) < V(0)e^{-\kappa t}, \tag{3.21}$$

where

$$\kappa = \frac{\lambda_{\min}(S_1 - C^T S_2 C)}{\lambda_{\max}(P)} > 0. \tag{3.22}$$

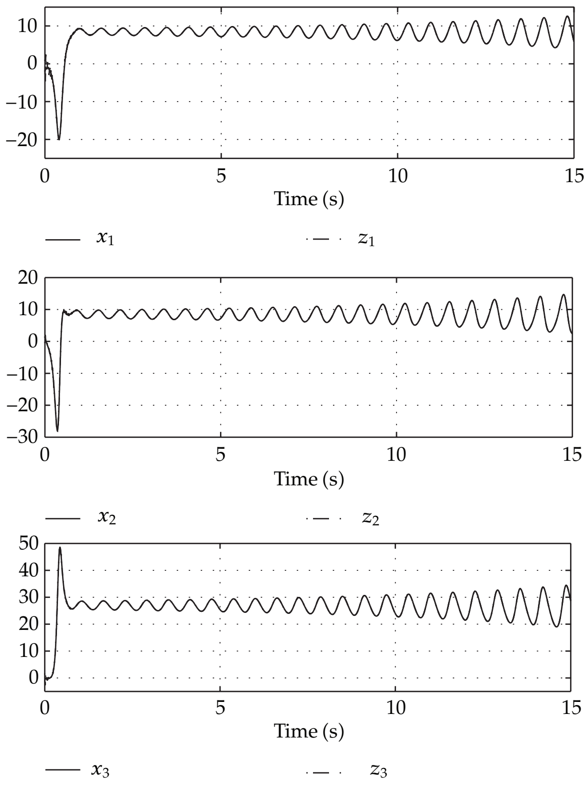

Figure 4: State trajectories.

Using (3.9), we have

$$\|x(t)\| < \sqrt{\frac{V(0)}{\lambda_{\min}(P)}e^{-\kappa t}} = \sqrt{\frac{x^T(0)Px(0)}{\lambda_{\min}(P)}}e^{-(\kappa/2)t}. \tag{3.23}$$

Since

$$\sqrt{\frac{x^T(0)Px(0)}{\lambda_{\min}(P)}} > 0, \qquad \frac{\kappa}{2} > 0, \tag{3.24}$$

the relation (3.23) guarantees the exponential stability. This completes the proof. □

Remark 3.3. The LMI problems given in Theorem 3.1 and Corollary 3.2 are to determine whether solutions exist or not. They are called the feasibility problems. The LMI problems can be solved efficiently by using recently developed convex optimization algorithms [24]. In this paper, in order to solve the LMI problems, we utilize MATLAB LMI Control Toolbox [25], which implements state-of-the-art interior-point algorithms.

4. Numerical Examples

4.1. Example 1

Consider the neural network (2.5)-(2.6) with the following parameters:

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \quad d(t) = \begin{bmatrix} d_1(t) \\ d_2(t) \end{bmatrix}, \quad A = \begin{bmatrix} -4 & 0 \\ 0 & -3 \end{bmatrix}, \quad G = \begin{bmatrix} 1 & 0.24 \\ 0 & 1 \end{bmatrix},$$
$$\theta(x(t)) = \begin{bmatrix} \dfrac{1}{1+e^{-x_1(t)}} \\ \dfrac{1}{1+e^{-x_2(t)}} \end{bmatrix}, \quad \phi(x(t)) = \begin{bmatrix} \dfrac{2}{1+e^{-2x_1(t)}} - 0.5 & \dfrac{2}{1+e^{-2x_1(t)}} - 0.5 \\ \dfrac{2}{1+e^{-2x_2(t)}} - 0.5 & \dfrac{2}{1+e^{-2x_2(t)}} - 0.5 \end{bmatrix},$$
$$u(t) = 0.1\begin{bmatrix} \sin(t) \\ \cos(t) \end{bmatrix}, \quad C = \begin{bmatrix} -0.5 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{4.1}$$

Applying Corollary 3.2 yields

$$P = \begin{bmatrix} 7.6834 & 0.0796 \\ 0.0796 & 9.4285 \end{bmatrix}, \quad Y = \begin{bmatrix} 2.6817 & 0.5495 \\ -1.0798 & 1.0493 \end{bmatrix},$$
$$S_1 = \begin{bmatrix} 28.8674 & -1.0876 \\ -1.0876 & 30.4987 \end{bmatrix}, \quad S_2 = 4.3502, \quad S_3 = \begin{bmatrix} 26.1485 & 0 \\ 0 & 28.3236 \end{bmatrix}. \tag{4.2}$$

Figure 1 shows state trajectories when the initial conditions are given by

$$x(0) = \begin{bmatrix} 1 \\ -1.5 \end{bmatrix}, \quad W(0) = \begin{bmatrix} 0.08 & -0.15 \\ 0.15 & -0.08 \end{bmatrix}, \quad \overline{W}(0) = \begin{bmatrix} -0.21 & 0.15 \\ -0.15 & -0.13 \end{bmatrix}, \tag{4.3}$$

and the external disturbance $d_i(t)$ $(i = 1, 2)$ is given by $w(t)$, where $w(t)$ means a Gaussian noise with mean 0 and variance 1. Figure 1 shows that the IOSSTL reduces the effect of the external disturbance $d(t)$ on the state vector $x(t)$. The evolutions of the weights $W(t)$ and $\overline{W}(t)$ are shown in Figures 2 and 3, respectively. Although the weights are not convergent, they are bounded. This result is very important for robust identification and control using neural networks [29].

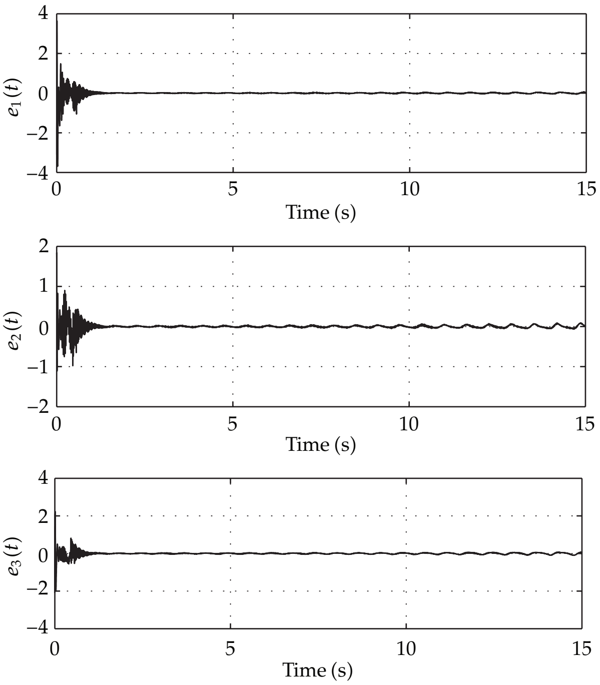

Figure 5: Identification errors.

4.2. Example 2

In this subsection, we apply the proposed scheme to the identification problem of Lorenz model presented in [30]. The Lorenz model is used for the fluid conviction description, especially for some feature of the atmospheric dynamic. The uncontrolled model is given by

$$\begin{aligned}\dot{z}_1(t) &= \sigma(z_2(t) - z_1(t)) + u_1(t),\\ \dot{z}_2(t) &= \rho z_1(t) - z_2(t) - z_1(t)z_3(t) + u_2(t),\\ \dot{z}_3(t) &= -\beta z_3(t) + z_1(t)z_2(t) + u_3(t),\end{aligned} \tag{4.4}$$

where $z_1(t)$, $z_2(t)$, and $z_3(t)$ represent measures of fluid velocity and horizontal and vertical temperature variations, respectively. σ, ρ, and β are positive parameters representing the Prandtl number, Rayleigh number, and geometric factor, respectively. As in the commonly studied case, we select $\sigma = 10$, $\beta = 8/3$, and $\rho = 28$. Consider the neural network (2.5)-(2.6) with the following parameters:

$$A = \begin{bmatrix} -8 & 0 & 0 \\ 0 & -8 & 0 \\ 0 & 0 & -8 \end{bmatrix}, \qquad \theta(x(t)) = \begin{bmatrix} \dfrac{2}{1+e^{-2x_1(t)}} - 0.5 \\ \dfrac{2}{1+e^{-2x_2(t)}} - 0.5 \\ \dfrac{2}{1+e^{-2x_3(t)}} - 0.5 \end{bmatrix},$$

$$\phi(x(t)) = \begin{bmatrix} \dfrac{0.2}{1+e^{-0.2x_1(t)}} - 0.05 & \dfrac{0.2}{1+e^{-0.2x_1(t)}} - 0.05 & \dfrac{0.2}{1+e^{-0.2x_1(t)}} - 0.05 \\ \dfrac{0.2}{1+e^{-0.2x_2(t)}} - 0.05 & \dfrac{0.2}{1+e^{-0.2x_2(t)}} - 0.05 & \dfrac{0.2}{1+e^{-0.2x_2(t)}} - 0.05 \\ \dfrac{0.2}{1+e^{-0.2x_3(t)}} - 0.05 & \dfrac{0.2}{1+e^{-0.2x_3(t)}} - 0.05 & \dfrac{0.2}{1+e^{-0.2x_3(t)}} - 0.05 \end{bmatrix}. \tag{4.5}$$

For the numerical simulation, we use the following parameters:

$$C = G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{bmatrix} = 0.01 \begin{bmatrix} \sin(t) \\ \cos(t) \\ 0 \end{bmatrix}. \tag{4.6}$$

Solving the LMIs (3.18)-(3.19) by the convex optimization technique software [24, 25] gives

$$\begin{gathered} P = \begin{bmatrix} 7.8455 & 0 & 0 \\ 0 & 7.8455 & 0 \\ 0 & 0 & 7.8455 \end{bmatrix}, \quad Y = \begin{bmatrix} 43.1503 & 0 & 0 \\ 0 & 43.1503 & 0 \\ 0 & 0 & 43.1503 \end{bmatrix}, \\ S_1 = \begin{bmatrix} 31.3821 & 0 & 0 \\ 0 & 31.3821 & 0 \\ 0 & 0 & 31.3821 \end{bmatrix}, \quad S_2 = \begin{bmatrix} 15.691 & 0 & 0 \\ 0 & 15.691 & 0 \\ 0 & 0 & 15.691 \end{bmatrix}, \\ S_3 = \begin{bmatrix} 23.5366 & 0 & 0 \\ 0 & 23.5366 & 0 \\ 0 & 0 & 23.5366 \end{bmatrix}. \end{gathered} \tag{4.7}$$

Figure 4 shows state trajectories when the initial conditions are given by

$$\begin{gathered} \begin{bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{bmatrix} = \begin{bmatrix} 1.1 \\ -0.3 \\ 1.7 \end{bmatrix}, \quad \begin{bmatrix} z_1(0) \\ z_2(0) \\ z_3(0) \end{bmatrix} = \begin{bmatrix} -2 \\ 1.5 \\ -0.4 \end{bmatrix}, \\ W(0) = \begin{bmatrix} 0.1 & -0.25 & 0.5 \\ 0.25 & -0.1 & 0.2 \\ 0.1 & -0.1 & 0.2 \end{bmatrix}, \quad \overline{W}(0) = \begin{bmatrix} -0.1 & -0.8 & 1 \\ 1.2 & 0.1 & -0.4 \\ -0.5 & 0.15 & -0.42 \end{bmatrix}, \end{gathered} \tag{4.8}$$

and the external disturbance $d_i(t)$ $(i = 1, 2, 3)$ is given by a Gaussian noise with mean 0 and variance 1. Figure 5 shows, by the proposed scheme, that the identification errors ($e_1(t) = z_1(t) - x_1(t)$, $e_2(t) = z_2(t) - x_2(t)$, and $e_3(t) = z_3(t) - x_3(t)$) are bounded around the origin.

These results appear to be of little use for the storage of patterns, but they are very important for identification and control using neural networks [29].

5. Conclusion

This paper has proposed the IOSSTL, which is a new robust training law, for weight adjustment of dynamic neural networks with external disturbance. Based on LMI approach, the IOSSTL was developed to guarantee exponential stability and reduce the effect of the external disturbance on the state vector. It was also shown that the IOSSTL can be obtained by solving the LMI. A numerical simulation was performed to demonstrate the effectiveness of the proposed IOSSTL. It is expected that the results obtained in this study can be extended to discrete-time dynamic neural networks.

The scheme proposed in this paper can be used in several control applications. For example, with the proposed learning law, a dynamic neural network is applied to model an unknown nonlinear system, and the neuron states estimated by some filters can be then utilized to achieve certain design objectives by the control law. This application of the proposed scheme remains as a future work.

Acknowledgment

This work was supported by the grant of the Korean Ministry of Education, Science and Technology (The Regional Core Research Program/Center for Healthcare Technology Development).

References

[1] M. M. Gupta, L. Jin, and N. Homma, *Static and Dynamic Neural Networks*, Wiley-Interscience, New York, NY, USA, 2003.

[2] D. G. Kelly, “Stability in contractive nonlinear neural networks,” *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 3, pp. 231–242, 1990.

[3] K. Matsuoka, “Stability conditions for nonlinear continuous neural networks with asymmetric connection weights,” *Neural Networks*, vol. 5, no. 3, pp. 495–500, 1992.

[4] X.-B. Liang and L.-D. Wu, “A simple proof of a necessary and sufficient condition for absolute stability of symmetric neural networks,” *IEEE Transactions on Circuits and Systems I*, vol. 45, no. 9, pp. 1010–1011, 1998.

[5] E. N. Sanchez and J. P. Perez, “Input-to-state stability (ISS) analysis for dynamic neural networks,” *IEEE Transactions on Circuits and Systems I*, vol. 46, no. 11, pp. 1395–1398, 1999.

[6] T. Chu, C. Zhang, and Z. Zhang, “Necessary and sufficient condition for absolute stability of normal neural networks,” *Neural Networks*, vol. 16, no. 8, pp. 1223–1227, 2003.

[7] T. Chu and C. Zhang, “New necessary and sufficient conditions for absolute stability of neural networks,” *Neural Networks*, vol. 20, no. 1, pp. 94–101, 2007.

[8] L. Zhang and E. Boukas, “Stability and stabilization of Markovian jump linear systems with partly unknown transition probabilities,” *Automatica*, vol. 45, no. 2, pp. 463–468, 2009.

[9] L. Zhang and J. Lam, “Necessary and sufficient conditions for analysis and synthesis of Markov jump linear systems with incomplete transition descriptions,” *IEEE Transactions on Automatic Control*, vol. 55, no. 7, pp. 1695–1701, 2010.

[10] G. A. Rovithakis and M. A. Christodoulou, “Adaptive control of unknown plants using dynamical neural networks,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 3, pp. 400–412, 1994.

[11] S. Jagannathan and F. L. Lewis, “Identification of nonlinear dynamical systems using multilayered neural networks,” *Automatica*, vol. 32, no. 12, pp. 1707–1712, 1996.

[12] J. A. K. Suykens, J. Vandewalle, and B. De Moor, "Lur'e systems with multilayer perceptron and recurrent neural networks: absolute stability and dissipativity," *IEEE Transactions on Automatic Control*, vol. 44, no. 4, pp. 770–774, 1999.

[13] W. Yu and X. Li, "Some stability properties of dynamic neural networks," *IEEE Transactions on Circuits and Systems I*, vol. 48, no. 2, pp. 256–259, 2001.

[14] I. Chairez, A. Poznyak, and T. Poznyak, "New sliding-mode learning law for dynamic neural network observer," *IEEE Transactions on Circuits and Systems II*, vol. 53, no. 12, pp. 1338–1342, 2006.

[15] W. Yu and X. Li, "Passivity analysis of dynamic neural networks with different time-scales," *Neural Processing Letters*, vol. 25, no. 2, pp. 143–155, 2007.

[16] J. de Jesús Rubio and W. Yu, "Nonlinear system identification with recurrent neural networks and dead-zone Kalman filter algorithm," *Neurocomputing*, vol. 70, no. 13–15, pp. 2460–2466, 2007.

[17] J. A. K. Suykens, J. Vandewalle, and B. L. R. De Moor, "NL_q theory: checking and imposing stability of recurrent neural networks for nonlinear modeling," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2682–2691, 1997.

[18] E. D. Sontag and Y. Wang, "Output-to-state stability and detectability of nonlinear systems," *Systems & Control Letters*, vol. 29, no. 5, pp. 279–290, 1997.

[19] M. Krichman, E. D. Sontag, and Y. Wang, "Input-output-to-state stability," *SIAM Journal on Control and Optimization*, vol. 39, no. 6, pp. 1874–1928, 2001.

[20] D. S. Laila and D. Nešić, "Changing supply rates for input-output to state stable discrete-time nonlinear systems with applications," *Automatica*, vol. 39, no. 5, pp. 821–835, 2003.

[21] D. Angeli, B. Ingalls, E. D. Sontag, and Y. Wang, "Separation principles for input-output and integral-input-to-state stability," *SIAM Journal on Control and Optimization*, vol. 43, no. 1, pp. 256–276, 2004.

[22] C. Cai and A. R. Teel, "Asymptotic characterizations of input–output-to-stable stability for discrete-time systems," *Systems & Control Letters*, vol. 56, no. 6, pp. 408–415, 2007.

[23] C. Cai and A. R. Teel, "Input-output-to-state stability for discrete-time systems," *Automatica*, vol. 44, no. 2, pp. 326–336, 2008.

[24] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, vol. 15 of *SIAM Studies in Applied Mathematics*, SIAM, Philadelphia, Pa, USA, 1994.

[25] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *LMI Control Toolbox*, The Mathworks Inc., 1995.

[26] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 81, no. 10, pp. 3088–3092, 1984.

[27] A. Albert, *Regression and the Moore-Penrose Pseudoinverse*, vol. 9 of *Mathematics in Science and Engineering*, Academic Press, New York, NY, USA, 1972.

[28] G. Strang, *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Wellesley, Mass, USA, 1986.

[29] K. J. Hunt, D. Sbarbaro, R. Żbikowski, and P. J. Gawthrop, "Neural networks for control systems—a survey," *Automatica*, vol. 28, no. 6, pp. 1083–1112, 1992.

[30] Y. Zeng and S. N. Singh, "Adaptive control of chaos in Lorenz system," *Dynamics and Control*, vol. 7, no. 2, pp. 143–154, 1997.