Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Stability criteria for three-layer locally recurrent networks

Krzysztof Patan *

* *Institute of Control and Computation Engineering University of Zielona Góra, ul. Podgórna 50, 65-246 Zielona Góra, Poland (e-mail: k.patan@issi.uz.zgora.pl)*

Abstract: The paper deals with a discrete-time recurrent neural network designed with dynamic neuron models. Dynamics are reproduced within each single neuron, hence the considered network is a locally recurrent globally feedforward. In the paper, conditions for global stability of the neural network considered are derived using the Lyapunov's second method.

Keywords: Neural networks, stability, modelling, Lyapunov's method.

1. INTRODUCTION

In the last decade, a growing interest in locally recurrent networks has been observed. This class of neural networks, due to their interesting properties, has been successfully applied to solve problems from different scientific and engineering areas. Cannas and co-workers [2001] applied a locally recurrent network to train the attractors of Chua's circuit, as a paradigm for studying chaos. The modelling of continuous polymerisation and neutralisation processes is reported in Zhang et al. [1998]. In turn, a three-layer locally recurrent neural network was succesfully applied to the control of non-linear systems in Gupta and Rao [1993]. In the framework of fault diagnosis, the literature reports many applications, e.g. an observer based fault detection and isolation system of a three-tank laboratory system Marcu et al. [1999], or model based fault diagnosis of sensor and actuator faults in a sugar evaporator Patan and Parisini [2005]. Tsoi and Back [1994] compared and applied different architectures of locally recurrent networks to the prediction of speech utterance. Finally, Campolucci and Piazza [2000] elaborated an intristic stability control method for a locally recurrent network designed for signal processing.

Stability plays an important role in both control theory and system identification. Furthermore, the stability issue is of crucial importance in relation to training algorithms adjusting the parameters of neural networks. If the predictor is unstable for certain choices of neural model parameters, serious numerical problems can occur during training. Stability criteria should be universal, applicable to as broad a class of systems as possible and at the same time computationally efficient. The majority of well-known approaches are based on Lyapunov's method Matsuoka [1992], Gupta et al. [2003], Ensari and Arik [2005], Cao et al. [2006], Forti et al. [2005]. Fang and Kincaid [1996] applied the matrix measure technique to study global exponential stability of asymmetrical Hopfield type networks. Jin and colleagues Jin et al. [1994] derived sufficient conditions for absolute stability of a general class of discrete-time recurrent networks by using Ostrowski's theorem. Recently, global asymptotic as well exponential stability conditions for discrete-time recurrent networks with globally Lipschitz continuous and monotone nondecreasing activation functions were introduced by Hu and Wang Hu and Wang [2002].

The stability analysis for locally recurrent networks with only one hidden layer is given in the work of Patan [2007a]. Unfortunately, approximation abilities of such networks are limited Patan [2007b]. Therefore, this paper presents stability criteria for locally recurrent networks with two hidden layers based on Lyapunov's second method. Moreover, some aspects concerning computational burden of stability checking are also discussed. Moreover, based on the elaborated stability conditions, a stabilization procedure is proposed, which quarantees the stability of the trained neural model.

The paper is organized as follows: in Section 2, the dynamic neural networks and its representation in the state-space are described. Stability analysis of the neural network considered as well as illustrative examples of stability checking are given in Section 3. Section 4 includes conlusions and final remarks.

2. DYNAMIC NEURAL NETWORKS

The topology of the neural network considered is analogous to that of the multi-layer feedforward one and the dynamics are reproduced by the so-called dynamic neuron models Patan and Parisini [2005]. Such neural networks are called locally recurrent globally feedforward Tsoi and Back [1994]. Dynamic properties of the model are achieved by introducing an Infinite Impulse Response (IIR) filter into a neuron structure. As a consequence of incorporating an IIR filter between input weights and an activation function, the neuron can reproduce its own past inputs and activations using two signals: the input $\boldsymbol{u}(k)$ and the output $y(k)$. In order to analyze the properties of the neuron model considered, it is convenient to represent it in

⋆ This work was supported in part by the Ministry of Science and Higher Education in Poland under the grant 3378/T02/2007/32 *Artificial Neural Networks in Robust Diagnostic Systems.*

the state-space. The states of the neuron can be described by the following state equation:

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{W}\boldsymbol{u}(k), \quad (1)$$

where $\boldsymbol{x}(k) \in \mathbb{R}^r$ is the state vector, $\boldsymbol{W} = \boldsymbol{1}\boldsymbol{w}^T$ is the weight matrix ($\boldsymbol{w} \in \mathbb{R}^n$, $\boldsymbol{1} \in \mathbb{R}^r$ is the vector with one in the first place and zeros elsewhere), $\boldsymbol{u}(k) \in \mathbb{R}^n$ is the input vector, n is the number of inputs, and the state matrix $\boldsymbol{A}$ has the form

$$\boldsymbol{A} = \begin{bmatrix} -a_1 & -a_2 & \dots & -a_{r-1} & -a_r \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}. \quad (2)$$

Finally, the neuron output is described by

$$y(k) = \sigma\big(g_2(\boldsymbol{b}\boldsymbol{x}(k) + \boldsymbol{d}\boldsymbol{u}(k) - g_1)\big), \quad (3)$$

where $\sigma(\cdot)$ is a non-linear activation function, $\boldsymbol{b} = [b_1, \dots, b_r]$ is the vector of feedforward filter parameters, $\boldsymbol{d} = [b_0 w_1, \dots, b_0 w_n]$.

2.1 State-space representation of the network

A locally recurrent network with only one hidden layer is represented by the linear state equation Patan [2007a]. Thus, its ability to approximate nonlinear mappings is limited. Therefore, in this paper, a network with two hidden layers has been taken into account. Let us consider a discrete-time dynamic neural network with n inputs and m outputs, with two hidden layers with v_1 neurons in the first layer and v_2 neurons in the second layer, and each neuron consisting of r-th order IIR filter.

Taking into account the layered topology of the network, one can decompose the state vector as follows: $\boldsymbol{x}(k) = [\boldsymbol{x}^1(k) \quad \boldsymbol{x}^2(k)]^T$, where $\boldsymbol{x}^1(k) \in \mathbb{R}^{N_1}$ ($N_1 = v_1 \times r$) represents the states of the first layer, and $\boldsymbol{x}^2(k) \in \mathbb{R}^{N_2}$ ($N_2 = v_2 \times r$) represents the states of the second layer. Then the state equation can be rewritten in the following form:

$$\boldsymbol{x}^1(k+1) = \boldsymbol{A}^1\boldsymbol{x}^1(k) + \boldsymbol{W}^1\boldsymbol{u}(k) \quad (4a)$$

$$\boldsymbol{x}^2(k+1) = \boldsymbol{A}^2\boldsymbol{x}^2(k) + \boldsymbol{W}^2\boldsymbol{\sigma}\big(\boldsymbol{G}_2^1(\boldsymbol{B}^1\boldsymbol{x}^1(k), +\boldsymbol{D}^1\boldsymbol{u}(k) - \boldsymbol{g}_1^1)\big) \quad (4b)$$

where $\boldsymbol{u} \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^m$ are inputs and outputs, respectively, $\boldsymbol{A}^1 \in \mathbb{R}^{N_1 \times N_1}$ and $\boldsymbol{A}^1 \in \mathbb{R}^{N_2 \times N_2}$ are the block diagonal state matrices of the first and second layers, respectively, $\boldsymbol{W}^1 \in \mathbb{R}^{N_1 \times n}$ is the input weight matrix, $\boldsymbol{W}^2 \in \mathbb{R}^{N_2 \times v_1}$ is the weight matrix between the first and second layers, $\boldsymbol{B}^1 \in \mathbb{R}^{v_1 \times N_1}$ is the block diagonal matrix of feedforward filter parameters of the first layer, $\boldsymbol{D}^1 \in \mathbb{R}^{v_1 \times n}$ is the transfer matrix, $\boldsymbol{g}_1^1$ denotes the vector of biases of the first layer, $\boldsymbol{G}_2^1 \in \mathbb{R}^{v_1 \times v_1}$ is the diagonal matrix of slope parameters of the first layer, and $\boldsymbol{\sigma} : \mathbb{R}^{v_1} \to \mathbb{R}^{v_1}$ is the non-linear vector-valued function. A detailed form of the network matrices can be found in Patan [2007b]. Finally, the output of the model is represented by the equation

$$\boldsymbol{y}(k) = \boldsymbol{C}^2\boldsymbol{\sigma}\big(\boldsymbol{G}_2^2(\boldsymbol{B}^2\boldsymbol{x}^2(k) + \boldsymbol{D}^2\boldsymbol{\sigma}\big(\boldsymbol{G}_2^1(\boldsymbol{B}^1\boldsymbol{x}^1(k) + \boldsymbol{D}^1\boldsymbol{u}(k) - \boldsymbol{g}_1^1)\big) - \boldsymbol{g}_1^2)\big), \quad (5)$$

where $\boldsymbol{C}^2 \in \mathbb{R}^{m \times v_2}$ is the output matrix, $\boldsymbol{B}^2 \in \mathbb{R}^{v_2 \times N_2}$ is the block diagonal matrix of second layer feedforward filter parameters, $\boldsymbol{D}^2 \in \mathbb{R}^{v_2 \times v_1}$ is the transfer matrix of second layer, $\boldsymbol{g}_1^2 \in \mathbb{R}^{v_2}$ is the vector of second layer biases, $\boldsymbol{G}_2^2 \in \mathbb{R}^{v_2 \times v_2}$ represents the diagonal matrix of the second layer activation function slope parameters.

3. STABILITY ANALYSIS – NETWORKS WITH TWO HIDDEN LAYERS

Let us consider the locally recurrent neural network (4) and (5) with two hidden layers containing v_1 neurons in the first layer and v_2 neurons in the second layer, where each neuron consists of the r-th order IIR filter, and an output layer with linear static elements. For further analysis let us assume that the activation function of each neuron is chosen as the hyperbolic tangent one $\sigma(x) = \tanh(x)$ satisfying the following conditions:

$$\begin{array}{ll} \text{(i)} & \sigma(x) \to \pm 1 \text{ as } x \to \pm\infty, \\ \text{(ii)} & \sigma(x) = 0 \text{ at a unique point } x = 0, \\ \text{(iii)} & \sigma'(x) > 0 \text{ and } \sigma'(x) \to 0 \text{ as } x \to \pm\infty, \\ \text{(iv)} & \sigma'(x) \text{ has a global maximum equal to 1.} \end{array} \quad (6)$$

In this case the state equation has a non-linear form. From the decomposed state equation (4), it is clearly seen that the states of the first layer of the network are independent of the states of the second layer and have a linear form (4a). The states of the second layer are described by the non-linearity (4b). Let $\boldsymbol{\Psi} = \boldsymbol{G}_2^1\boldsymbol{B}^1$ and $\boldsymbol{s}_1 = \boldsymbol{G}_2^1\boldsymbol{D}^1\boldsymbol{u}(k) - \boldsymbol{G}_2^1\boldsymbol{g}_1^1$, where $\boldsymbol{s}_1$ can be regarded as a threshold or a fixed input; then (4b) takes the form

$$\boldsymbol{x}^2(k+1) = \boldsymbol{A}^2\boldsymbol{x}^2(k) + \boldsymbol{W}^2\boldsymbol{\sigma}\big(\boldsymbol{\Psi}\boldsymbol{x}^1(k) + \boldsymbol{s}_1\big). \quad (7)$$

Using the linear transformation $\boldsymbol{v}^1(k) = \boldsymbol{\Psi}\boldsymbol{x}^1(k) + \boldsymbol{s}_1$ and $\boldsymbol{v}^2(k) = \boldsymbol{x}^2(k)$, one obtains an equivalent system:

$$\begin{cases} \boldsymbol{v}^1(k+1) = & \boldsymbol{\Psi}\boldsymbol{A}^1\boldsymbol{\Psi}^-\boldsymbol{v}^1 - \boldsymbol{\Psi}\boldsymbol{A}^1\boldsymbol{\Psi}^-\boldsymbol{s}_1 + \boldsymbol{s}_2 \\ \boldsymbol{v}^2(k+1) = & \boldsymbol{A}_i^2\boldsymbol{v}^2(k) + \boldsymbol{W}_i^2\boldsymbol{\sigma}\big(\boldsymbol{v}^1(k)\big) \end{cases}, \quad (8)$$

where $\boldsymbol{\Psi}^-$ is a pseudoinverse of the matrix $\boldsymbol{\Psi}$ (e.g. in a Moore-Penrose sense), and $\boldsymbol{s}_2 = \boldsymbol{\Psi}\boldsymbol{W}^1\boldsymbol{u}(k) + \boldsymbol{s}_1$ is a threshold or a fixed input. Let $\boldsymbol{v}^* = [\boldsymbol{v}^{1*} \ \boldsymbol{v}^{2*}]^T$ be an equilibrium point of (8). Introducing an equivalent coordinate transformation $\boldsymbol{z}(k) = \boldsymbol{v}(k) - \boldsymbol{v}^*(k)$, the system (8) can be transformed to the form

$$\begin{cases} \boldsymbol{z}^1(k+1) = & \boldsymbol{\Psi}\boldsymbol{A}^1\boldsymbol{\Psi}^-\boldsymbol{z}^1(k) \\ \boldsymbol{z}^2(k+1) = & \boldsymbol{A}^2\boldsymbol{z}^2(k) + \boldsymbol{W}^2\boldsymbol{f}(\boldsymbol{z}^1(k)) \end{cases}, \quad (9)$$

where $\boldsymbol{f}(\boldsymbol{z}^1(k)) = \boldsymbol{\sigma}(\boldsymbol{z}^1(k) + \boldsymbol{v}^{1*}(k)) - \boldsymbol{\sigma}(\boldsymbol{v}^{1*}(k))$. Substituting $\boldsymbol{z}(k) = [\boldsymbol{z}^1(k) \ \boldsymbol{z}^2(k)]^T$, one finally obtains

$$\boldsymbol{z}(k+1) = \mathcal{A}\boldsymbol{z}(k) + \mathcal{W}\boldsymbol{f}(\boldsymbol{z}(k)), \quad (10)$$

where

$$\mathcal{A} = \begin{bmatrix} \boldsymbol{\Psi}\boldsymbol{A}^1\boldsymbol{\Psi}^- & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}^2 \end{bmatrix}, \quad \mathcal{W} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{W}^2 & \boldsymbol{0} \end{bmatrix}. \quad (11)$$

In order to obtain stability conditions for the system (10), the second Lyapunov's method will be applied.

Lemma 1. (Global stability theorem of Lyapunov, Gupta et al. [2003]) Let $\boldsymbol{x} = 0$ be an equilibrium point of the system

$$\boldsymbol{x}(k+1) = \boldsymbol{f}(\boldsymbol{x}(k)), \quad (12)$$

and $\boldsymbol{V} : \mathbb{R}^n \to \mathbb{R}$ a continuously differentiable function such that

(1) $\boldsymbol{V}(\boldsymbol{x}(k)=0)=0$,
(2) $\boldsymbol{V}(\boldsymbol{x}(k))>0$, for $\boldsymbol{x}\neq 0$,
(3) $\boldsymbol{V}(\boldsymbol{x})\rightarrow\infty$ for $\|\boldsymbol{x}\|\rightarrow\infty$,
(4) $\Delta\boldsymbol{V}(\boldsymbol{x}(k))=\boldsymbol{V}(\boldsymbol{x}(k+1))-\boldsymbol{V}(\boldsymbol{x}(k))<0$ for $\boldsymbol{x}\neq 0$.

Then, the equilibrium point $\boldsymbol{x}=0$ is globally asymptotically stable and $\boldsymbol{V}(\boldsymbol{x})$ is a global Lyapunov function.

Theorem 2. The neural system represented by (10) is globally asymptotically stable if the following condition is satisfied:

$$\|\mathcal{A}\|+\|\mathcal{W}\|<1. \tag{13}$$

Proof. Let $\boldsymbol{V}(\boldsymbol{z})=\|\boldsymbol{z}\|$ be a Lyapunov function for the system (10). This function is positive definite with the minimum at $\boldsymbol{x}(k)=0$. The difference along the trajectory of the system is given as follows:

$$\begin{aligned}\Delta\boldsymbol{V}(\boldsymbol{z}(k))&=\|\boldsymbol{z}(k+1)\|-\|\boldsymbol{z}(k)\|\\&=\|\mathcal{A}\boldsymbol{z}(k)+\mathcal{W}\boldsymbol{f}(\boldsymbol{z}(k))\|-\|\boldsymbol{z}(k)\|\\&\leqslant\|\mathcal{A}\boldsymbol{z}(k)\|+\|\mathcal{W}\boldsymbol{f}(\boldsymbol{z}(k))\|-\|\boldsymbol{z}(k)\|.\end{aligned} \tag{14}$$

The activation function $\boldsymbol{\sigma}$ is a short map with the Lipschitz constant $L=1$. Then $\boldsymbol{f}$ is also a short map, with the property $\|\boldsymbol{f}(\boldsymbol{z}(k))\|\leqslant\|\boldsymbol{z}(k)\|$, and (14) can be expressed in the form

$$\begin{aligned}\Delta\boldsymbol{V}(\boldsymbol{z}(k))&\leqslant\|\mathcal{A}\|\|\boldsymbol{z}(k)\|+\|\mathcal{W}\|\|\boldsymbol{z}(k)\|-\|\boldsymbol{z}(k)\|\\&\leqslant(\|\mathcal{A}\|+\|\mathcal{W}\|-1)\|\boldsymbol{z}(k)\|.\end{aligned} \tag{15}$$

From (15) one can see that if

$$\|\mathcal{A}\|+\|\mathcal{W}\|<1, \tag{16}$$

then $\Delta\boldsymbol{V}(\boldsymbol{z}(k))$ is negative definite and the system (10) is globally asymptotically stable, which completes the proof.

Remark 3. The theorem formulates the sufficient condition only, not a necessary one. Therefore, if the condition (13) is not satisfied, one cannot judge the stability of the system.

Remark 4. The condition (13) is very restrictive. The matrix $\mathcal{A}$ is a block diagonal one with the entries $\boldsymbol{\Psi}\boldsymbol{A}^1\boldsymbol{\Psi}^-$ and $\boldsymbol{A}_i^2$, for $i=1,\ldots,v_2$. For block diagonal matrices, the following relation holds:

$$\|\mathcal{A}\|=\max_{i=1,\ldots,n}\{\|\mathcal{A}_i\|\}. \tag{17}$$

The entries of $\mathcal{A}$ for $i=2,\ldots,v_2$ have the form (2). For such matrices, the norm is greater than or equal to one. Thus, Theorem 2 is useless, because there is no network (10) able to satisfy (13). One way to make (13) applicable to the system (10) is to use the modified neuron state matrix of the form (18) with the parameter $\nu<1$

$$\boldsymbol{A}_i=\begin{bmatrix}-a_{1_i} & -a_{2_i} & \ldots & -a_{r-1_i} & -a_{r_i}\\ \nu & 0 & \ldots & 0 & 0\\ 0 & \nu & \ldots & 0 & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \ldots & \nu & 0\end{bmatrix}. \tag{18}$$

The parameter ν can be selected experimentally by the user or can be adapted by a training procedure.

Remark 5. In spite of its shortcomings, the condition (13) is very attractive because of its simplicity and ease of use.

Remark 6. The theorem is also valid for other activation functions with the Lipschitz constant $L\leqslant 1$, satisfying the conditions (6).

Example 7. Consider the neural network described by (4) and (5) with 7 neurons in the first hidden layer and 4 neurons in the second hidden layer. Each neuron consists of the second order IIR filter and a hyperbolic tangent activation function. The network is applied to model the process described in Experiment 4 in the outstanding paper of Narendra and Parthasarathy [1990]. Training was carried out for 5000 steps using the SPSA algorithm with the settings $a=0.002$, $c=0.01$, $\alpha=0.302$, $\gamma=0.101$, $A=100$. The training set consists of 100 samples generated randomly using the uniform distribution. The sum of squared errors for the training set is 0.6943, and for the testing set containing another 100 samples it is 1.2484. The stability of the trained network was tested using the norm stability condition (13) as follows:

$$\|\mathcal{A}\|_2+\|\mathcal{W}\|_2=4.0399>1 \tag{19}$$

$$\|\mathcal{A}\|_1+\|\mathcal{W}\|_1=4.7221>1\ , \tag{20}$$

$$\|\mathcal{A}\|_\infty+\|\mathcal{W}\|_\infty=6.4021>1 \tag{21}$$

where

$$\begin{aligned}\|\boldsymbol{X}\|_2&=\sqrt{\lambda_{max}\boldsymbol{X}^T\boldsymbol{X}}\\ \|\boldsymbol{X}\|_1&=\max_{1\leqslant i\leqslant n}\sum_{j=1}^{n}|x_{ij}|\\ \|\boldsymbol{X}\|_\infty&=\max_{1\leqslant j\leqslant n}\sum_{i=1}^{n}|x_{ij}|\end{aligned}\ .$$

Unfortunately, based on (19)–(21), the norm stability condition cannot judge the stability of the system. On the other hand, observing the convergence of the network states one can see that the system is stable. Figures 1(a) and (b) present convergence of the states of the first and second layers of the system (4). All states converge to set-up values what means that the network is stable.

This experiment clearly shows that the norm stability condition is very restrictive. Moreover, to satisfy the condition (13) the entries of both matrices $\mathcal{A}$ and $\mathcal{W}$ should have relatively small values. The following procedure proposes the training of the network assuring the stability of the model. Assuming that each neuron in the network is represented by the modified state transition matrix (18), the norm of the matrix $\mathcal{W}$ is checked after each training step. If the norm stability condition is not satisfied, the entries of $\mathcal{W}$ are decreased iteratively.

Table 1. Outline of norm stability checking

Step 0: Initiation
Choose the network parameters is such a way that $\|\boldsymbol{A}\|<1$, set $\nu<1$.

Step 1: Update the network parameters using a training algorithm.

Step 2: Assure the stability of the network:

set $x:=1$;
while ($\|\boldsymbol{A}\|+\|\boldsymbol{W}\|>1$) **do**
  $x:=x+1$;
  $\boldsymbol{W}:=\boldsymbol{W}/(x\cdot\|\boldsymbol{W}\|)$;
end

Step 3: Termination criteria
if (termination criterion satisfied) **then** STOP
else go to Step 1.

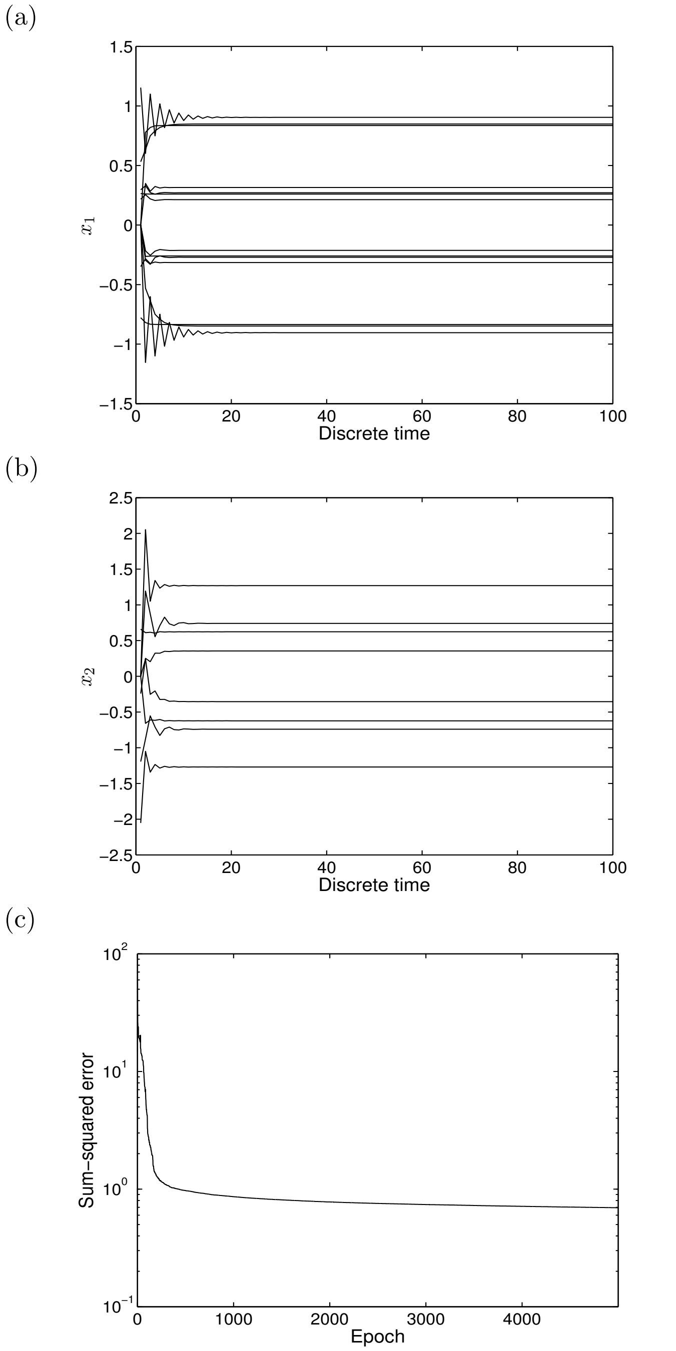

Fig. 1. Convergence of network states (a) and (b), learning track (c)

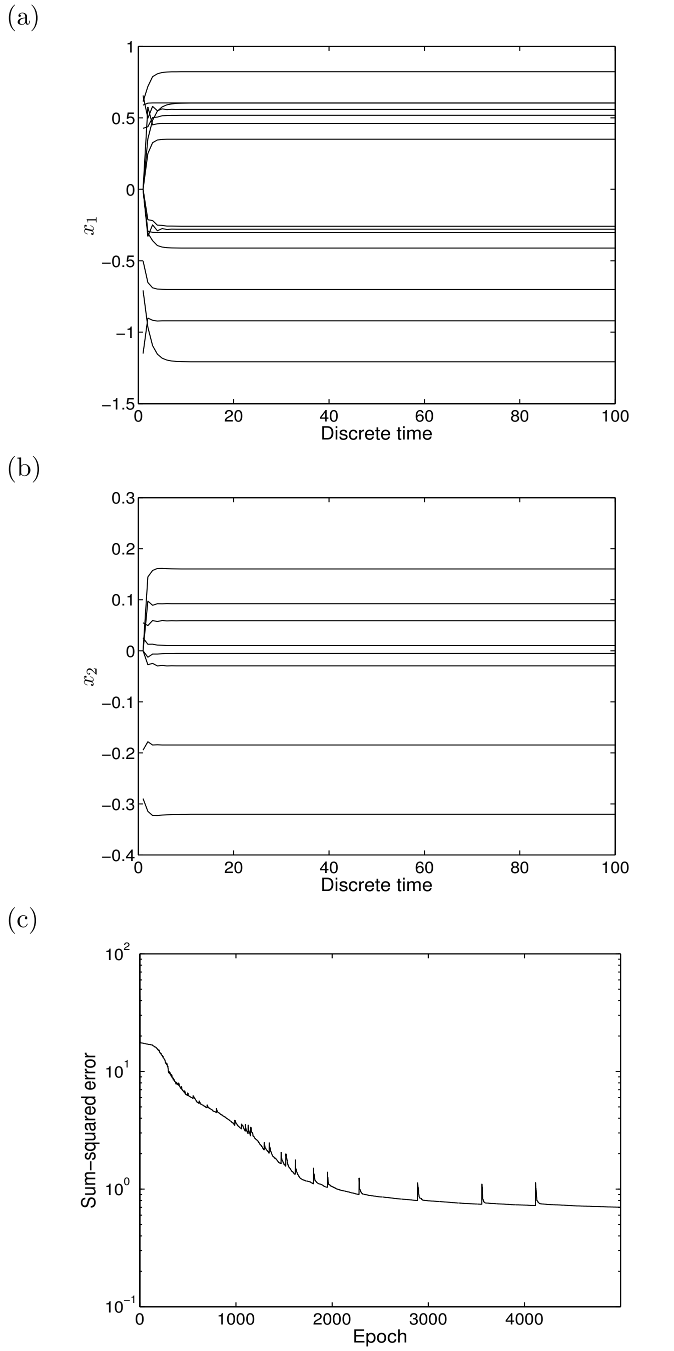

Fig. 2. Convergence of network states (a) and (b), learning track (c)

Example 8. Le us revisit the problem considered in the example 7, but this time with each neuron in the network represented by the modified state transition matrix (18) with the parameter $\nu = 0.5$. The training is carried out using the procedure shown in Table 1. In this case, the sum of squared errors for the training set is 0.7008, and for the testing set containing another 100 samples it is 1.2924. The stability of the already trained network was tested using the norm stability condition (13) as follows:

$$\|\mathcal{A}\|_2 + \|\mathcal{W}\|_2 = 0.9822 < 1. \tag{22}$$

In this case, the criterion is satisfied and the neural network is globally asymptotically stable. Similarly as in the previous example, Figs 2(a) and (b) present the convergence of the states of the first and second layers of the system (4). In turn, in Figs 2(c) and (d) the convergence of the transformed autonomous system (10) is shown. All states converge to zero, which means that the network is stable. The procedure presented in Table 1 guarantees the stability of the model. Recalculating the weights $\mathcal{W}$ can introduce perturbations to the training in the form of spikes, as illustrated in Fig. 2(e), but the training is, in general, convergent.

The discussed examples show that the norm stability condition is a restrictive one. In order to successfully apply this criterion to network training, several modifications are required. Firstly, the form of the state transition matrix

$\mathcal{A}$ is modified and, secondly, the update of the network weight matrix $\mathcal{W}$ should be performed during training. In the further part of the section less restrictive stability conditions are investigated.

Theorem 9. The neural system (10) is globally asymptotically stable if there exists a matrix $\boldsymbol{P} \succ 0$ such that the following condition is satisfied:

$$(\mathcal{A}+\mathcal{W})^T \boldsymbol{P}(\mathcal{A}+\mathcal{W}) - \boldsymbol{P} \prec 0. \tag{23}$$

Proof. Let us consider a positive definite candidate Lyapunov function:

$$V(\boldsymbol{z}) = \boldsymbol{z}^T \boldsymbol{P}\boldsymbol{z}. \tag{24}$$

The difference along the trajectory of the system (10) is given as follows:

$$\begin{aligned}\Delta V(\boldsymbol{z}(k)) =& V(\boldsymbol{z}(k+1)) - V(\boldsymbol{z}(k))\\ =& \left(\mathcal{A}\boldsymbol{z}(k)+\mathcal{W}\boldsymbol{f}(\boldsymbol{z}(k))\right)^T \boldsymbol{P}\left(\mathcal{A}\boldsymbol{z}(k)+\mathcal{W}\boldsymbol{f}(\boldsymbol{z}(k))\right)\\ &- \boldsymbol{z}^T(k)\boldsymbol{P}\boldsymbol{z}(k)\\ =& \boldsymbol{z}^T(k)\mathcal{A}^T\boldsymbol{P}\mathcal{A}\boldsymbol{z}(k) + \boldsymbol{f}^T(\boldsymbol{z}(k))\mathcal{W}^T\boldsymbol{P}\mathcal{A}\boldsymbol{z}(k)\\ &+ \boldsymbol{z}^T(k)\mathcal{A}^T\boldsymbol{P}\mathcal{W}\boldsymbol{f}(\boldsymbol{z}(k)) - \boldsymbol{z}^T(k)\boldsymbol{P}\boldsymbol{z}(k)\\ &+ \boldsymbol{f}^T(\boldsymbol{z}(k))\mathcal{W}^T\boldsymbol{P}\mathcal{W}\boldsymbol{f}(\boldsymbol{z}(k)).\end{aligned} \tag{25}$$

For activation functions satisfying the conditions (6) it holds that $|f(z)| \leqslant |z|$ and

$$f = \begin{cases} f & z>0 \\ -f & z<0 \end{cases}; \tag{26}$$

then

$$\boldsymbol{f}^T(\boldsymbol{z}(k))\mathcal{W}^T\boldsymbol{P}\mathcal{A}\boldsymbol{z}(k) \leqslant \boldsymbol{z}^T(k)\mathcal{W}^T\boldsymbol{P}\mathcal{A}\boldsymbol{z}(k) \tag{27}$$

$$\boldsymbol{z}^T(k)\mathcal{A}^T\boldsymbol{P}\mathcal{W}\boldsymbol{f}(\boldsymbol{z}(k)) \leqslant \boldsymbol{z}^T(k)\mathcal{A}^T\boldsymbol{P}\mathcal{W}\boldsymbol{z}(k) \tag{28}$$

and

$$\boldsymbol{f}^T(\boldsymbol{z}(k))\mathcal{W}^T\boldsymbol{P}\mathcal{W}\boldsymbol{f}(\boldsymbol{z}(k)) \leqslant \boldsymbol{z}^T(k)\mathcal{W}^T\boldsymbol{P}\mathcal{W}\boldsymbol{z}(k). \tag{29}$$

Substituting the inequalities (27), (28) and (29) into (25), one obtains

$$\begin{aligned}\Delta V(\boldsymbol{z}(k)) \leqslant& \boldsymbol{z}^T(k)\mathcal{A}^T\boldsymbol{P}\mathcal{A}\boldsymbol{z}(k) + \boldsymbol{z}^T(k)\mathcal{W}^T\boldsymbol{P}\mathcal{A}\boldsymbol{z}(k)\\ &+ \boldsymbol{z}^T(k)\mathcal{A}^T\boldsymbol{P}\mathcal{W}\boldsymbol{z}(k) + \boldsymbol{z}^T(k)\mathcal{W}^T\boldsymbol{P}\mathcal{W}\boldsymbol{z}(k)\\ &- \boldsymbol{z}^T(k)\boldsymbol{P}\boldsymbol{z}(k)\\ \leqslant& \boldsymbol{z}^T(k)(\mathcal{A}^T\boldsymbol{P}\mathcal{A} + \mathcal{W}^T\boldsymbol{P}\mathcal{A} + \mathcal{A}^T\boldsymbol{P}\mathcal{W}\\ &+ \mathcal{W}^T\boldsymbol{P}\mathcal{W} - \boldsymbol{P})\boldsymbol{z}(k)\\ \leqslant& \boldsymbol{z}^T(k)\left((\mathcal{A}+\mathcal{W})^T\boldsymbol{P}(\mathcal{A}+\mathcal{W}) - \boldsymbol{P}\right)\boldsymbol{z}(k).\end{aligned} \tag{30}$$

From (30) one can see that if

$$(\mathcal{A}+\mathcal{W})^T \boldsymbol{P}(\mathcal{A}+\mathcal{W}) - \boldsymbol{P} \prec 0, \tag{31}$$

then $\Delta\boldsymbol{V}(\boldsymbol{z}(k))$ is negative definite and the system (10) is globally asymptotically stable.

Remark 10. From the practical point of view, the selection of a proper matrix $\boldsymbol{P}$, in order to satisfy the condition (23), can be troublesome. Therefore, the corollary presented below allows us to verify the stability of the system in an easier manner. The corollary is formulated in the form of the Linear Matrix Inequality (LMI). Recently, LMI methods have become quite popular among researchers from the control community due to their simplicity and effectiveness taking into account numerical complexity.

Lemma 11. (Schur complement Boyd et al. [1994]). Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{C} \in \mathbb{R}^{m\times m}$ be symmetric matrices, and $\boldsymbol{A} \succ 0$; then

$$\boldsymbol{C} + \boldsymbol{B}^T\boldsymbol{A}^{-1}\boldsymbol{B} \prec 0, \tag{32}$$

iff

$$\boldsymbol{U} = \begin{bmatrix} -\boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^T & \boldsymbol{C} \end{bmatrix} \prec 0 \quad \text{or} \quad \boldsymbol{U} = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{B}^T \\ \boldsymbol{B} & -\boldsymbol{A} \end{bmatrix} \prec 0. \tag{33}$$

Corollary 12. The neural system (10) is globally asymptotically stable if there exists a matrix $\boldsymbol{Q} \succ 0$ such that the following LMI holds:

$$\begin{bmatrix} -\boldsymbol{Q} & (\mathcal{A}+\mathcal{W})\,\boldsymbol{Q} \\ \boldsymbol{Q}\,(\mathcal{A}+\mathcal{W})^T & -\boldsymbol{Q} \end{bmatrix} \prec 0. \tag{34}$$

Proof. From Theorem 9 one knows that the system (10) is globally asymptotically stable if the following condition is satisfied:

$$(\mathcal{A}+\mathcal{W})^T \boldsymbol{P}(\mathcal{A}+\mathcal{W}) - \boldsymbol{P} \prec 0. \tag{35}$$

Applying the Schur complement formula to (35) yields

$$\begin{bmatrix} -\boldsymbol{P}^{-1} & \mathcal{A}+\mathcal{W} \\ (\mathcal{A}+\mathcal{W})^T & -\boldsymbol{P} \end{bmatrix} \prec 0. \tag{36}$$

In order to transform (36) into the LMI, let us introduce the substitution $\boldsymbol{Q} = \boldsymbol{P}^{-1}$ and then multiply the result from the left and the right by $\mathrm{diag}(\boldsymbol{I}, \boldsymbol{Q})$ to obtain

$$\begin{bmatrix} -\boldsymbol{Q} & (\mathcal{A}+\mathcal{W})\,\boldsymbol{Q} \\ \boldsymbol{Q}\,(\mathcal{A}+\mathcal{W})^T & -\boldsymbol{Q} \end{bmatrix} \prec 0.$$

Remark 13. The LMI (34) defines the so-called feasibility problem Boyd et al. [1994]. This convex optimisation problem can be solved effectively using poly-nomial-time algorithms, e.g. interior point methods. Interior point algorithms are computationally efficient and nowadays widely used for solving LMIs.

Example 14. Consider again the problem presented in Example 7. As is shown in that example, the norm stability condition cannot ensure the stability of the neural model (4). In this example, the condition given in Corollary 12 is used to check the stability of the neural network. The problem was solved with the LMI solver implemented in the LMI Control Toolbox under Matlab 7.0. After 4 iterations the solver found the feasible solution represented by a positive definite matrix $\boldsymbol{Q}$ what means that the neural network is globally asymptotically stable. This example shows that the condition presented in Theorem 9 is less restrictive than the norm stability condition. Moreover, representing a stability condition in the form of LMIs renders it possible to easily check the stability of the neural system.

Lemma 15. (Gahinet and Apkarian [1994]). Let $\boldsymbol{A} \in \mathbb{R}^{q\times q}$ be a symmetric matrix, and $\boldsymbol{P} \in \mathbb{R}^{r\times q}$ and $\boldsymbol{Q} \in \mathbb{R}^{s\times q}$ real matrices; then there exists a matrix $\boldsymbol{B} \in \mathbb{R}^{r\times s}$ such that

$$\boldsymbol{A} + \boldsymbol{P}^T\boldsymbol{B}^T\boldsymbol{Q} + \boldsymbol{Q}^T\boldsymbol{B}\boldsymbol{P} \prec 0 \tag{37}$$

iff the inequalities $\boldsymbol{W}_P^T\boldsymbol{A}\boldsymbol{W}_P \prec 0$ and $\boldsymbol{W}_Q^T\boldsymbol{A}\boldsymbol{W}_Q \prec 0$ both hold, where $\boldsymbol{W}_P$, and $\boldsymbol{W}_Q$ are full rank matrices satisfying $Im(\boldsymbol{W}_P) = ker(\boldsymbol{P})$ and $Im(\boldsymbol{W}_Q) = ker(\boldsymbol{Q})$.

Example 16. The term (34) can be rewritten as

$$\begin{aligned}\begin{bmatrix} -\boldsymbol{Q} & (\mathcal{A}+\mathcal{W})\,\boldsymbol{Q} \\ \boldsymbol{Q}\,(\mathcal{A}+\mathcal{W})^T & -\boldsymbol{Q} \end{bmatrix} =& \begin{bmatrix} -\boldsymbol{Q} & \mathcal{A}\boldsymbol{Q} \\ \boldsymbol{Q}\mathcal{A}^T & -\boldsymbol{Q} \end{bmatrix}\\ &+ \begin{bmatrix} \boldsymbol{W} \\ \boldsymbol{0} \end{bmatrix}\boldsymbol{Q}\left[\,\boldsymbol{0}\;\boldsymbol{I}\,\right] + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix}\boldsymbol{Q}^T\left[\,\boldsymbol{W}^T\;\boldsymbol{0}\,\right].\end{aligned} \tag{38}$$

Using Lemma 15 one obtains

Table 2. Comparison of methods.

Network	LMI (34)		LMIs (40)	
structure	time [sec]	iterations	time [sec]	iterations
7-4	0.0845	4	0.0172	1
15-7	0.545	3	0.189	1
25-10	6.5	4	1.7	1

$$\boldsymbol{W}_P^T \begin{bmatrix} -\boldsymbol{Q} & \mathcal{A}\boldsymbol{Q} \\ \boldsymbol{Q}\mathcal{A}^T & -\boldsymbol{Q} \end{bmatrix} \boldsymbol{W}_P \prec 0, \quad \boldsymbol{W}_Q^T \begin{bmatrix} -\boldsymbol{Q} & \mathcal{A}\boldsymbol{Q} \\ \boldsymbol{Q}\mathcal{A}^T & -\boldsymbol{Q} \end{bmatrix} \boldsymbol{W}_Q \prec 0, \tag{39}$$

where $\boldsymbol{W}_P = \mathrm{diag}(\ker(W), \boldsymbol{I})$ and $\boldsymbol{W}_Q = \mathrm{diag}(\boldsymbol{I}, \boldsymbol{0})$. Multiplying the second inequality in (39) gives $\boldsymbol{Q} \succ 0$. Then (39) can be rewritten as

$$\boldsymbol{W}_P^T \boldsymbol{R} \boldsymbol{W}_P \prec 0, \quad \boldsymbol{Q} \succ 0, \tag{40}$$

where

$$\boldsymbol{R} = \begin{bmatrix} -\boldsymbol{Q} & \mathcal{A}\boldsymbol{Q} \\ \boldsymbol{Q}\mathcal{A}^T & -\boldsymbol{Q} \end{bmatrix}. \tag{41}$$

These LMI conditions can be solved with a less computational burden than the LMI condition 34. The results of computations, for different network structures, are presented in Table 2. The experiments were performed using the LMI Control Toolbox under Matlab 7.0 on a PC with Intel Centrino 1.4 GHz and a 512MB RAM.

4. CONCLUSIONS

The paper proposes stability criteria for locally recurrent networks with two hidden layers. The norm stability condition is a restrictive one, but introducing a modified neuron structure makes this condition applicable to real-life problems. Moreover, the norm stability criterion can be adopted to design stable training of the dynamic neural network, which guarantees the stability of the model. On the other hand, the condition presented in Theorem 9 is less restrictive than the norm stability condition, but there are problems with finding a proper matrix $\boldsymbol{P}$ able to satisfy the condition. Therefore, it is proposed to formulate this condition in the form of LMIs, and then the stability can be easily checked using suitable numerical packages. Theorems based on the Lyapunov's second method give sufficient conditions for global asymptotic stability. If these conditions are not satisfied, one cannot judge the stability of the system. Moreover, stability criteria developed using the Lyapunov's second method cannot be used as a starting point to determime constraints on the network parameters. Thus, the optimisation problem with constraints cannot be determined and application of the elaborated stability conditions to real-time training is limited. A solution of the problem can be achieved based on the Lyapunov's first method.

REFERENCES

S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* SIAM Studies in Applied and Numerical Mathematics. SIAM, Philadelphia, USA, 1994. Vol. 15.

P. Campolucci and F. Piazza. Intrinsic stability-control method for recursive filters and neural networks. *IEEE Trans. Circuit and Systems – II: Analog and Digital Signal Processing*, 47(8):797–802, 2000.

B. Cannas, S. Cincotti, M. Marchesi, and F. Pilo. Learnig of Chua's circuit attractors by locally recurrent neural networks. *Chaos Solitons & Fractals*, 12:2109–2115, 2001.

J. Cao, K. Yuan, and H.X. Li. Global asymptotical stability of recurrent neural networks with multiple discrete delays and distributed delays. *IEEE Trans. Neural Networks*, 17:1646–1651, 2006.

T. Ensari and S. Arik. Global stability analysis of neural networks with multiple time varying delays. *IEEE Trans. Automatic Control*, 50:1781–1785, 2005.

Y. Fang and T. G. Kincaid. Stability analysis of dynamical neural networks. *IEEE Trans. Neural Networks*, 7:996–1006, 1996.

M. Forti, P. Nistri, and D. Papini. Global exponential stability and global convergence in finite time of delayed neural networks with infinite gain. *IEEE Trans. Neural Networks*, 16:1449–1463, 2005.

P. Gahinet and P. Apkarian. A linear matrix inequality approach to H_∞ control. *International Journal of Robust and Nonlinear Control*, 4:421–448, 1994.

M. M. Gupta and D. H. Rao. Dynamic neural units with application to the control of unknown nonlinear systems. *Journal of Intelligent and Fuzzy Systems*, 1:73–92, 1993.

M. M. Gupta, L. Jin, and N. Homma. *Static and Dynamic Neural Neetworks. From Fundamentals to Advanced Theory.* John Wiley & Sons, New Jersey, 2003.

S. Hu and J. Wang. Global stability of a class of discrete-time recurrent neural networks. *IEEE Trans. Circuits and Systems-I: Fundamental Theory and Applications*, 49:1104–1117, 2002.

L. Jin, P. N. Nikiforuk, and M. M. Gupta. Absolute stability conditions for discrete-time recurrent neural networks. *IEEE Trans. Neural Networks*, 5:954–963, 1994.

T. Marcu, L. Mirea, and P. M. Frank. Development of dynamical neural networks with application to observer based fault detection and isolation. *International Journal of Applied Mathematics and Computer Science*, 9 (3):547–570, 1999.

K. Matsuoka. Stability conditions for nonlinear continuous neural networks with asymmetric connection weights. *Neural Networks*, 5:495–500, 1992.

K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Netwoks*, 1:12–18, 1990.

K. Patan. Stability analysis and the stabilization of a class of discrete-time dynamic neural networks. *IEEE Trans. Neural Networks*, 18:660–673, 2007a.

K. Patan. Aproximation of state-space trajectories by locally recurrent globally feed-forward neural networks. *Neural Networks*, 2007b. DOI: 10.1016/j.neunet.2007.10.004.

K. Patan and T. Parisini. Identification of neural dynamic models for fault detection and isolation: the case of a real sugar evaporation process. *Journal of Process Control*, 15:67–79, 2005.

A. Ch. Tsoi and A. D. Back. Locally recurrent globally feedforward networks: A critical review of architectures. *IEEE Trans. Neural Networks*, 5:229–239, 1994.

J. Zhang, A. J. Morris, and E. B. Martin. Long term prediction models based on mixed order locally recurrent neural networks. *Computers Chem. Engng*, 22:1051–1063, 1998.