ELSEVIER

Chemical Engineering Journal 68 (1997) 41–50

Chemical Engineering Journal

Dynamic neural network control for non-linear systems: optimal neural network structure and stability analysis

Masoud Nikravesh [a,*], Andrew E. Farell [b,1], Thomas G. Stanford [b]

[a] *Earth Sciences Division of Lawrence Berkeley National Laboratory, University of California at Berkeley, Berkeley, CA 94720, USA*

[b] *Department of Chemical Engineering, University of South Carolina, Columbia, SC 29208, USA*

Received 15 August 1996; accepted 3 February 1997

Abstract

Design techniques for non-linear dynamic systems are closely related to their stability properties. Stability results can be used to design a reliable controller. This paper discusses the stability analysis of the dynamic neural network control (DNNC). The results from DNNC stability analysis will be used to define the neural network stability index (NNSI). The NNSI is a practical index which in current form can only be used with DNNC structures. The NNSI can be used to determine the optimal DNNC network structure. In addition, we will provide guidelines for the design of an optimal DNNC network structure for the conventional neural network structure for model-based control strategies. In this study, DNNC will be designed for a non-isothermal CSTR as an example of a wide class of non-linear processes.

Keywords: Neural networks; Model-based control; CSTR; Time varying processes; Non-linear systems; Stability analysis; Dynamic neural network control

1. Introduction

During the last decade, application of neural networks for identification and control purposes has increased exponentially and applied widely and successfully in many areas [1–10]. These widespread applications have been due to several attractive features of neural networks. For example, neural networks have the potential to model very complicated non-linear systems [11–15]. They can be trained easily by using past data records from systems under study. They are readily applicable to multivariable systems [15]. They have the ability to infer general rules and extract typical patterns from specific examples and recognize input–output mapping parameters from complex multi-dimentional field data [16,17]. These facts suggest that neural networks, in conjunction with a suitable control strategy such as model-based control [4,10,18–20], differential-geometric control [6,21,22], and neuro-fuzzy control [23] can be used to control non-linear systems.

Neural networks are now widely used in many non-linear control applications [3,5,6,23,24]. Typical neural network models are complex and have several nodes in the input and hidden layers, as well as a large number of weights and bias terms. Since the neural network models are frequently complex, the calculation of the inverse of the process models for the design of the controller is oftentimes not trivial. Therefore, it is difficult to study the stability issue of neural network-based control systems. It is also difficult to design an optimal neural network structure for identification and control purposes for the same reason. Therefore, it is important to reduce the complexity of the mathematical expressions of the neural network models to analyze the stability of the neural-based control system and design an optimal network structure for control purposes. Jin et al. [25] and Narendra and Parthasarathy [3], provide stability analysis for a simple type of neural network and stress the importance of studying the stability properties of neural network models.

The stability analysis of neural-based control systems is an important issue which must be considered for the design of a good neural-based control system [15,25]. There are several methods which have been proposed to study the open-loop and closed-loop stability of processes and to analyze and design control systems [26]. State-space methods are best suited for analysis and synthesis of non-linear systems and they can be applied to the design of optimal control systems [26]. Once the systems are transformed into state-space models, non-linear model approaches such as geometric control [6,21,22], neuro-fuzzy control [23,24], fuzzy logic control

* Corresponding author. E-mail: nikraves@cs.berkeley.edu

[1] Present address: Union Camp Corp., P.O. Box B, Eastover, SC 29022, USA.

1385-8947/97/$17.00
PII S1385-8947(97)00041-7

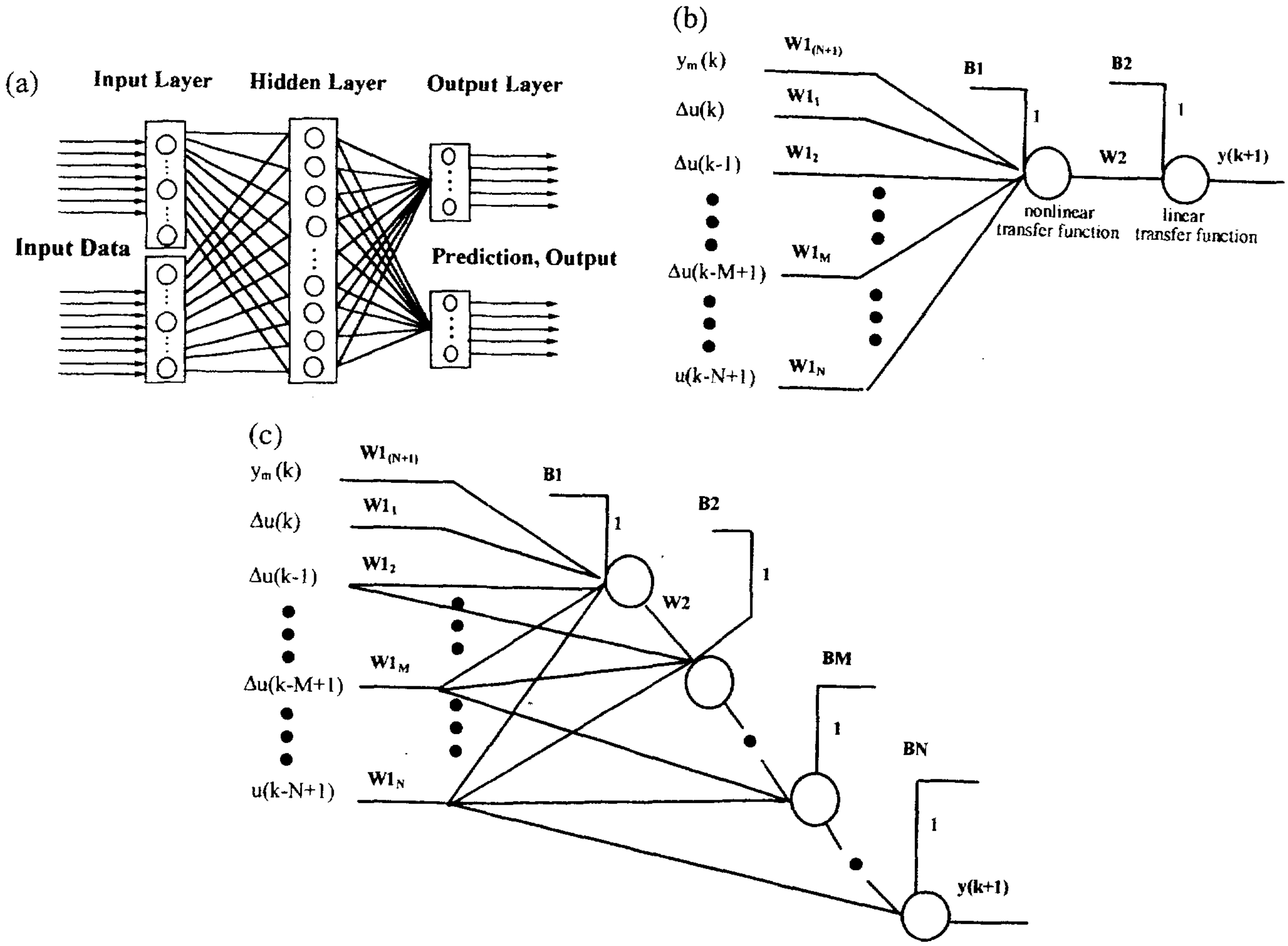

Fig. 1. Comparison between conventional neural network structure and DNNC network structure: (a) typical neural network model; (b) typical multi-layer DNNC process model; (c) typical single-layer DNNC process model.

[27], and model-based control [4,6,10,18–20] can be used to design and analyze the controller performance. In addition, the Liapunov theory [26,28] can be used for stability analysis [15,25,29,30]. Liapunov stability theory plays an important role in the stability analysis of control systems described by state-space equations. The second method of Liapunov [26,28] is most commonly used and is applicable to both linear and non-linear systems. This method is also suited for the stability of non-linear systems for which exact solutions may be unobtainable such as neural network models. Although the second method of Liapunov is applicable to a wide class of non-linear systems including neural network systems for stability analysis, obtaining successful results is not trivial. Therefore, experience may be necessary to interpret correctly the results from the stability analysis of non-linear systems.

Recently, Nikravesh [15] and Nikravesh et al. [5] proposed dynamic neural network control (DNNC) as a control strategy. DNNC is a simple neural network model-based control strategy. Although the DNNC network structure is simple, it has demonstrated the potential for controlling a wide class of non-linear systems. The objectives of this paper are to study the stability of DNNC, to determine the optimal DNNC network structure for identification and control purposes, and to analyze the controller performance.

The structure of the paper is as follows. First, a brief introduction on stability analysis will be given. Next, the stability analysis of DNNC will be discussed according to the Liapunov theory, and the neural network stability index (NNSI) will be introduced. The NNSI index will be used to determine an optimal DNNC network structure, to design an optimal DNNC controller, and to analyze the controller performance. Finally, an optimal DNNC neural network structure will be designed for the non-isothermal CSTR as an example of a wide class of non-linear processes.

2. Dynamic neural network control (DNNC)

Fig. 1(a) shows the structure of a conventional neural network. The typical neural network has an input layer, an output layer, and at least one hidden layer. Each layer is fully connected to the succeeding layer with corresponding weights. In this case, only the neighboring layers are connected to each other. The weights represent the current state of knowledge of the network and are adjusted to improve the network per-

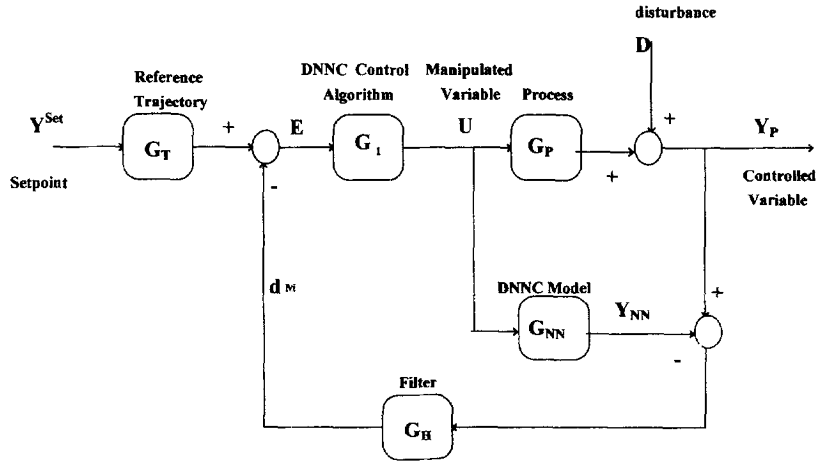

Fig. 2. Block diagram of simplified DNNC strategy in IMC framework.

formance. In general, the performance of neural networks is a function of hidden layer topology. During the past several years, it has been shown that the performance of conventional neural networks for prediction can be increased by connecting certain nodes in a specific layer to certain nodes in the non-succeeding layer. For example, one can also connect certain nodes in the input layer to certain nodes in the output layer. Using this concept, the multi-layer DNNC model (Fig. 1(b)) has been introduced. It is clear that complex networks are capable of modeling more complicated behavior than our simple model. However, it is our contention that many highly non-linear systems can be handled with the simple structure of DNNC. In this study, a single layer DNNC (Fig. 1(c)) is used. If desired, it is easy to expand the new methodology to the multi-layer DNNC case.

In the following section, a brief overview of DNNC will be given. Then the stability analysis of the DNNC process model and controller will be discussed. In this paper, we present the DNNC controller in an IMC framework as shown in Fig. 2. In Fig. 2, Y_P is the controlled variable (measured), Y_{NN} is neural network prediction, U is the manipulated variable (predicted), Y^{set} is the setpoint, and D is the disturbance. G_P is the actual process model, G_{NN} is the neural network model of the process, G_I is the inverse of the process model (for the ideal case), and G_H is the filter transfer function. Details of IMC and IMC filter design are available throughout the literature [15,19,31–34]. However, DNNC can be employed in a more general model predictive control (MPC) framework. For example, DNNC can be employed in DMC framework. Details of such a DNNC controller are given in ref. [15].

2.1. State-space representation of DNNC

Fig. 1(c) shows DNNC network structure. The input–output mapping of the DNNC [15] can be represented by,

$$y(k+1)=w_2\Gamma(w_1^{\mathrm{T}}\Delta uy+B_1)+B_2 \tag{1}$$

$$\Delta uy=[\Delta u(k)\Delta u(k-1)\cdots\Delta u(k-N+2)u(k-N+1)y_m(k)]^{\mathrm{T}}$$

$$w_1=[w1_1w1_2\cdots w1_Nw1_{N+1}]^T$$

$$\Gamma(z)=\frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$$

where w_1 is neural network weight, input-hidden layer; B_1 is neural network bias, hidden layer node; w_2 is neural network weight, hidden-output layer; B_2 is neural network bias, output layer; y_m is controlled variable, measured value; y is controlled variable, neural network prediction; u is manipulated variable, measured value; Γ is neuron transfer fuction.

Substituting $\Delta u(k-j+2)=u(k-j+2)-u(k-j+1)$ into Eq. (1) and $y(k)=y_m(k)$ gives,

$$\begin{aligned} &y(k+1)=w_2\Gamma(\Delta w_1^{\mathrm{T}}uy+B_1)+B_2\\ &uy=[u(k)u(k-1)\cdots u(k-N+2)u(k-N+1)y(k)]^{\mathrm{T}}\\ &\Delta w_1=[w1_1(w1_2-w1_1)\cdots(w1_N-w1_{N-1})w1_{N+1}]^{\mathrm{T}} \end{aligned} \tag{2}$$

Eq. (2) can be written in the following discrete state-space form,

$$\begin{aligned} &x(k+1)=f(x(k))+g(x(k),u(k))\\ &y(k)=h(z^{-1}(x(k),u(k)))\\ &y(k+1)=h(x(k),u(k)) \end{aligned} \tag{3}$$

with $x(k)$ given by,

$$\begin{aligned} x(k)&=[x_1(k)x_2(k)\cdots x_{N-1}(k)x_N(k)]^{\mathrm{T}}\\ &=[u(k-1)u(k-2)\cdots u(k-N+1)y(k)]^{\mathrm{T}} \end{aligned} \tag{4}$$

Therefore, $x(k+1)$ is given by,

$$\begin{aligned} x(k+1)&=[x_1(k+1)x_2(k+1)\cdots x_{N-1}(k+1)x_N(k+1)]^{\mathrm{T}}\\ &=[u(k)u(k-1)\cdots u(k-N+2)y(k+1)]^{\mathrm{T}}\\ &=[u(k)x_1(k)\cdots x_{N-2}(k)h(x(k),u(k))]^{\mathrm{T}} \end{aligned} \tag{5}$$

with,

$$f(x(k)) = [0\ x_1(k)\ x_2(k) \cdots x_{N-2}(k)\ 0]^{\mathrm{T}}$$
$$g(x(k),u(k)) = [u(k)\ 0\ 0 \cdots 0\ 0\ h(x(k),u(k))]^{\mathrm{T}} \qquad (6)$$
$$h(x(k),u(k)) = y(k+1) = w_2 \Gamma(\Delta w_1^{\mathrm{T}} uy + B_1) + B_2$$

2.2. Closed-loop stability under the DNNC

In this study, we are interested in the stability of the overall process. In the DNNC strategy in the IMC framework (Fig. 2) and with an exact model for the process, the stability of both the process and controller is sufficient for overall system stability [15]. In this section, the stability of the DNNC process model and controller model will be presented.

2.2.1. Stability of the DNNC process mode

The stability of the process model according to Liapunov theory is guaranteed if the eigenvalues of the Jacobian of $x(k+1)$ with respect to $x(k)$ are inside of the unit circle. In this case, the Jacobian is given by,

$$J^{\mathrm{Y}} = \begin{bmatrix} 0 & 0 & 0 & \bullet & 0 & 0 & 0 \\ 1 & 0 & 0 & \bullet & 0 & 0 & 0 \\ 0 & 1 & 0 & \bullet & 0 & 0 & 0 \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ 0 & 0 & 0 & \bullet & 0 & 0 & 0 \\ 0 & 0 & 0 & \bullet & 1 & 0 & 0 \\ \alpha_1 & \alpha_2 & \alpha_3 & \bullet & \alpha_{N-2} & \alpha_{N-1} & w_2 \cdot w1_{N+1} \cdot (1 - y1(k)^2) \end{bmatrix} \qquad (7)$$

where,

$$y1(k) = \Gamma(\Delta w_1^{\mathrm{T}} uy + B_1)$$
$$\alpha_j = w_2 \cdot \Delta w_{1,j} \cdot (1 - y1(k)^2)$$
$$\Delta w_{1,j} = w1_{j+1} - w1_j \qquad (8)$$
$$j = 1,\ldots,N-1$$

The non-zero eigenvalue of the Jacobian matrix J^{Y} is given by,

$$\lambda = (1 - y1(k)^2) w1_{N+1} w_2 \qquad (9)$$

2.2.2. Stability of the DNNC controller

The DNNC controller will be defined using the inverse of the DNNC process model and is given by the following equations [15]:

$$u(k) = \frac{\left[\Gamma^{-1}\left(\dfrac{\nu(k) - B_2}{w_2}\right) - B_1 - (\Delta w_1^{\mathrm{C}})^{\mathrm{T}} uy^{\mathrm{C}}\right]}{w_{1,1}} \qquad (10)$$

with,

$$\Delta w_1^{\mathrm{C}} = [\Delta w_{1,1} \Delta w_{1,2} \ldots \Delta w_{1,N-1} w_{1,N+1}]^{\mathrm{T}}$$
$$uy^{\mathrm{C}} = [u(k-1) u(k-2) \ldots u(k-N+2) u(k-N+1) y(k)]^{\mathrm{T}}$$
$$\nu(k) = \nu(k-1) + (1-\varphi)[y^{\mathrm{set}}(k) - d(k) - y(k)]$$
$$d(k) = y_{\mathrm{m}}(k) - y(k)$$
$$\Gamma^{-1}(z) = -0.5 \ln\left[\frac{1-z}{1+z}\right]$$

In this case, the state-space representation of the controller (Eq. (10)) is given by,

$$x(k+1) = \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 & \bullet & \beta_{N-2} & \beta_{N-1} & \dfrac{-w1_{N+1}}{w1_1} \\ 1 & 0 & 0 & \bullet & 0 & 0 & 0 \\ 0 & 1 & 0 & \bullet & 0 & 0 & 0 \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ 0 & 0 & 0 & \bullet & 0 & 0 & 0 \\ 0 & 0 & 0 & \bullet & 1 & 0 & 0 \\ 0 & 0 & 0 & \bullet & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \bullet \\ x_{N-2}(k) \\ x_{N-1}(k) \\ x_N(k) \end{bmatrix} + \begin{bmatrix} R \\ 0 \\ 0 \\ \bullet \\ 0 \\ 0 \\ \nu^{\mathrm{set}}(k) \end{bmatrix} \qquad (11)$$

$$\beta_j = \frac{-\Delta w_{1,j}}{w1_1}$$

$$R = \frac{\Gamma^{-1}\left(\dfrac{\nu^{\mathrm{set}}(k) - B2}{w_2}\right) - B1}{w1_1}$$

The stability of the controller according to Liapunov is guaranteed if the eigenvalues of the Jacobian of $x(k+1)$ with respect to $x(k)$ are inside of the unit circle. In this case, the Jacobian is given by,

$$J^{\mathrm{U}} = \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 & \bullet & \beta_{N-2} & \beta_{N-1} & \dfrac{-w1_{N+1}}{w1_1} \\ 1 & 0 & 0 & \bullet & 0 & 0 & 0 \\ 0 & 1 & 0 & \bullet & 0 & 0 & 0 \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ 0 & 0 & 0 & \bullet & 0 & 0 & 0 \\ 0 & 0 & 0 & \bullet & 1 & 0 & 0 \\ 0 & 0 & 0 & \bullet & 0 & 0 & 0 \end{bmatrix} \qquad (12)$$

The non-zero eigenvalues of the Jacobian matrix J^{U} are the solutions of the following characteristic equation:

$$\lambda^{N-1}-\beta_1\cdot\lambda^{N-2}-\beta_2\cdot\lambda^{N-3}-\cdots-\beta_{N-2}\cdot\lambda-\beta_{N-1}=0 \tag{13}$$

2.3. Optimum DNNC network structure

The results from DNNC stability analysis will be used to define the neural network stability index (NNSI). In this study, several indexes have been examined. It has been found that one of these indexes has a clear and one-to-one relationship with the closed-loop stability performance of DNNC. Neural network stability index (NNSI) is given in Table 1. As shown in Table 1, the NNSI is a function of the number of eigenvalues of the process model (N_Y), the number of eigenvalues of the inverse of the process model (N_U), eigenvalues of the process model (J_Y), and eigenvalues of the inverse of the process model (J_U). The NNSI is a practical index which in current form can only be used with DNNC of the structure. The NNSI can be used to determine the optimal DNNC network structure. In addition, NNSI provides guidelines for the design of an optimal DNNC network structure for the conventional neural network structure for model-based control strategies.

In this section, we would like to answer the following question, "What is the optimum DNNC network structure for an acceptable controller performance?" To answer this question, the neural network stability index (NNSI) is defined (Table 1). Based on NNSI, the optimum number of nodes in the input layer (past information of manipulated variable, N) for an acceptable controller performance is defined as follows,

Guideline 1. N is equal to 70% of the maximum number of nodes required to model the process for which the NNSI attains some constant value and if this constant value for the NNSI is less than 0.50.

Guideline 2. N is equal to the maximum number of nodes required to model the process for which the NNSI attains some constant value and if this constant value for NNSI is greater than 0.50.

Therefore, the optimum number of nodes required to model the process for an acceptable controller performance is given by the maximum value for N which is obtained from guidelines 1 and 2. In addition, as we will show later, increasing the number of nodes in the input layer (DNNC network structure) results in the smaller value for the NNSI which implies a more stable process model and has a smoother and slower response.

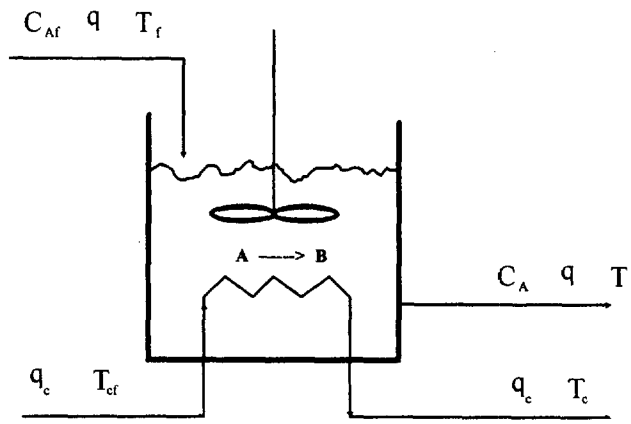

Fig. 3. Non-isothermal CSTR.

3. Simulation studies

The performance of the DNNC strategy was tested on a non-isothermal CSTR with irreversible reaction (A→B) (Fig. 3). The process model consists of two non-linear ordinary differential equations and is given by [15],

$$\frac{dC_A}{dt}=\frac{q}{V}(C_{Af}-C_A)-k_0C_A\exp\left(-\frac{E}{RT}\right)\phi_c(t) \tag{14a}$$

$$\frac{dT}{dt}=\frac{q}{V}(T_f-T)+\frac{(-\Delta H)k_0C_A}{\rho C_p}\exp\left(-\frac{E}{RT}\right)\phi_c(t)+\frac{\rho_cC_{pc}}{\rho C_pV}q_c\left[1-\exp\left(-\frac{hA}{q_c\rho C_{pc}}\phi_h(t)\right)\right](T_{cf}-T) \tag{14b}$$

with ϕ_c and ϕ_h given by [35,36]

$$\phi_c(t)=\exp(-\alpha_c t) \tag{14c}$$

$$h_d=\phi_h(t)h=(1-\alpha_h t)h \tag{14d}$$

where h_d is the heat transfer coefficient, scaled; $\phi_h(t)$ is the fouling coefficient, $0<\phi_h<1$; α_h is the fouling constant; $\phi_c(t)$ is the deactivation coefficient; α_c is the deactivation constant; C_A is the effluent concentration, the controlled var-

Table 1
Neural network stability index (NNSI)

$$\frac{\frac{N_U}{N_Y}\sum_{j=1}^{N_Y}J_{Y,j}+\sum_{j=1}^{N_U}J_{U,j}}{2N_U}=\frac{\sum_{j=1}^{N_Y}J_{Y,j}+\frac{N_Y}{N_U}\sum_{j=1}^{N_U}J_{U,j}}{2N_Y}=\frac{N_U\sum_{j=1}^{N_Y}J_{Y,j}+N_Y\sum_{j=1}^{N_U}}{2N_UN_Y}=\frac{\frac{1}{N_Y}\sum_{j=1}^{N_Y}J_{Y,j}+\frac{1}{N_U}\sum_{j=1}^{N_U}J_{U,j}}{2}$$

N_Y	Number of eigenvalues; process model
N_U	Number of eigenvalues; inverse of the process model
J_Y	Eigenvalues; process model
J_U	Eigenvalues; inverse of the process model

Table 2
Nominal CSTR operating condition [15,19]

$q = 100\ \mathrm{l\ min^{-1}}$	$E/R = 9.95\times10^{3}$ K
$C_{Af} = 1\ \mathrm{mol\ l^{-1}}$	$-\Delta H = 2\times10^{5}$ cal mol^{-1}
$T_f = 350$ K	$\rho,\rho_c = 1000$ g l^{-1}
$T_{cf} = 350$ K	$C_p, C_{pc} = 1$ cal g^{-1} K^{-1}
$V = 100$ l	$q_c = 103.41$ l min^{-1}
$h_A = 7\times10^{5}$ cal min^{-1} K^{-1}	$T = 440.2$ K
$k_0 = 7.2\times10^{10}$ min^{-1}	$C_A = 8.36\times10^{-2}$ mol l^{-1}
$\phi_h(t) = (1-\alpha_h t)$	
$\phi_c(t) = \exp(-\alpha t)$	

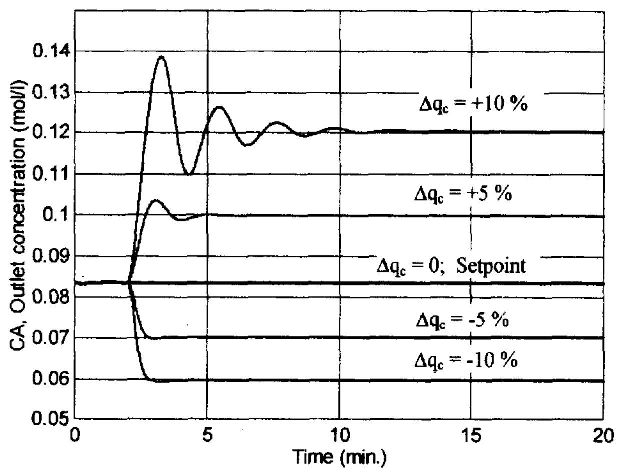

Fig. 4. Open-loop response of the CSTR for step changes in the coolant flow rate q_c.

iable; q_c is the coolant flow rate, the manipulated variable; q is the feed flow rate, disturbance; C_{Af} is the feed concentration; T_f is the feed temperature; T_{cf} is the coolant inlet temperature. The remaining model parameters and operating conditions are presented in Table 2.

The open-loop step response for a series of step changes in q_c is shown in Fig. 4. It is seen that the process is highly non-linear. Table 3 shows the neural network structures for DNNC and a conventional neural network (neural network internal model control (NIMC) as an example [19]) which were used for this study. The DNNC and NIMC models are trained using the backpropagation algorithm with data generated by making random changes in q_c. Table 4 shows the quantitative comparison between DNNC-2, DNNC-5, DNNC-10, DNNC-15, DNNC-25 and NIMC. In this study, DNNC-N refers to DNNC neural network model structure with $N+2$ input nodes representing current values for the manipulated variable $(u(k))$ and the controlled variable $(y(k))$ and N past information of the manipulated variable $(u(k-1)$ to $u(k-N))$. DNNC-2 provides very good output prediction for a randomly generated disturbance in q_c. Further improvements will be obtained by increasing the number of nodes (manipulated input) in the input layer. NIMC provides excellent output prediction for a randomly generated disturbance in q_c. Several performance measures show that the performance of DNNC-15 for model identification is the same as NIMC. Although this is true, we note that the DNNC structure is very simple and on the average, has a small number of nodes (total number of weights and bias terms is 7 to

Table 3
Comparison between DNNC-N and NIMC neural network structures

Network structure	Total number of weights and bias terms	Input layer	Hidden layer	Output layer
Dynamic neural network control [15] (DNNC-N)	$N+5$	Number of nodes $N+2$ (current C_A and current and N past values for q_c)	Number of nodes 1, transfer function non-linear	Number of nodes 1, transfer function linear prediction of C_A
Dynamic neural network control [15] (DNNC-10)	$N=10$ $N+5=15$	$N=10$, Number of nodes $N+2=12$ (current C_A and current and ten past values for q_c)	Number of nodes 1, transfer function non-linear	Number of nodes 1, transfer function linear prediction of C_A
Neural network internal model control [19] (NIMC)	70	Number of nodes 6 (current and two past values for q_c and C_A)	Transfer function non-linear	Number of nodes 1, transfer function linear prediction of C_A

Table 4
Comparison between DNNC-N and NIMC neural network structures

Neural network structure for the process model	Mean (error)	Standard (error)	Sum squares (error)
DNNC-2	0.0015	0.0069	0.013
DNNC-5	0.00058	0.0066	0.0115
DNNC-10	0.00042	0.0062	0.0101
DNNC-15	0.0017	0.0054	0.0082
DNNC-25	0.0017	0.0054	0.0082
NIMC [19]	0.0017	0.0054	0.0082

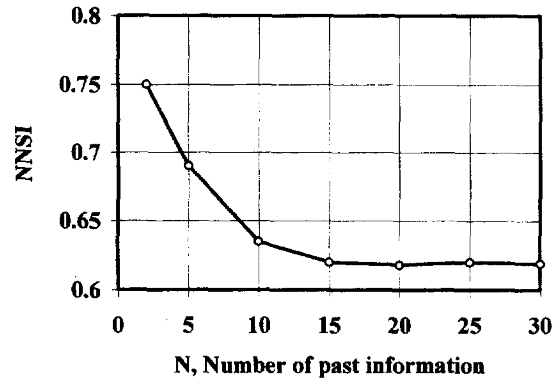

Fig. 5. Neural network stability index (NNSI).

20) in comparison with NIMC (total number of weights and bias terms is 70).

3.1. Optimum DNNC network structure

Eqs. (9) and (13) and Table 1 are used to calculate the NNSI for the DNNC process models and is shown in Fig. 5. Referring to Fig. 5, one can see that increasing the number of nodes in the input layer results in the smaller value for the NNSI which implies a more stable process model. In addition, referring to Fig. 5, one can see that for $N \geq 15$ the NNSI remains fairly constant. Therefore, the maximum number of nodes needed to model this process would be 15. The optimum number of nodes required to model the process with an acceptable controller performance is given by the maximum value for N obtained from guidelines 1 and 2, and is equal to 15. (i.e. maximum nodes 15 and final value for NNSI 0.61, therefore for this case study $N = 15$).

3.2. Controller performance based on the results from stability analysis

To test the NNSI for interpreting controller performance without applying the controller, we considered the DNNC network model with different numbers of nodes in the input layer. Based on Fig. 5 we predict that DNNC-2 will have a faster but more oscillatory response than other DNNC structures. In addition, we predict that DNNC-2 will be the least stable and DNNC-15 will be the most stable controller.

To illustrate the usefulness of the NNSI for predicting the DNNC controller performance, DNNC is applied to control the CSTR. DNNC was tuned with a filter constant value of 0.95 ($\varphi = 0.95$). Fig. 6(a) and (b) show the disturbance rejection performance of the DNNC. In comparison to DNNC-15 (for 10% and 20% change in inlet flow rate as disturbance), DNNC-2 exhibits a faster response toward setpoint but with oscillatory performance. Comparing DNNC-15 to DNNC-10, one can find that DNNC-15 shows a slower, but smoother response.

Comparing the results extracted from Fig. 5 and the performance of the DNNC controllers in Fig. 6(a) and (b), one can see that the conclusions are in good agreement. In brief, the results show that the controller with the smaller value for NNSI is more stable and has a smoother and slower response.

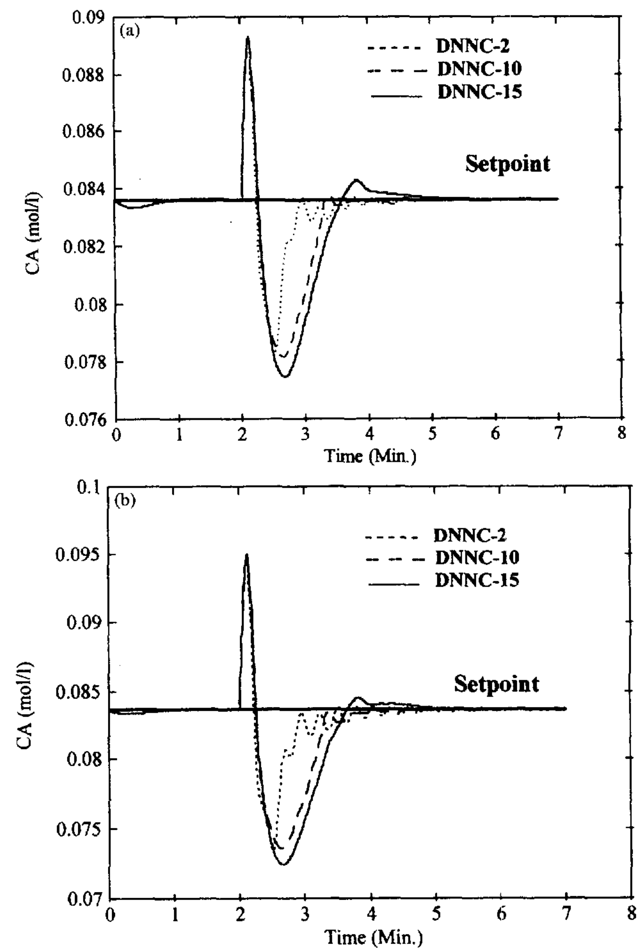

Fig. 6. (a) Disturbance rejection performance of DNNC, %10 changes in q_c; (b) disturbance rejection performance of DNNC, %20 changes in q_c.

3.3. Optimum DNNC controller for NIMC control strategy

Referring to Table 4, one can see that the performance of DNNC-15 for model identification is the same as NIMC. Therefore, we predict that if we use the DNNC-15 controller with the NIMC process model, the performance of this new hybrid model would be very close and even the same as the DNNC strategy with the DNNC-15 structure. The same prediction will be expected for any NIMC–DNNC-N hybrid controller with $N \geq 15$. Figs. 7 and 8 show the controller performances for DNNC-15 and NIMC–DNNC-15. Comparing the controller performances of DNNC-15 to NIMC–DNNC-15, one can clearly see that these controllers have exactly the same overall performance.

4. Conclusions

In this paper, detailed guidelines for the stability analysis of DNNC and conventional neural networks were presented.

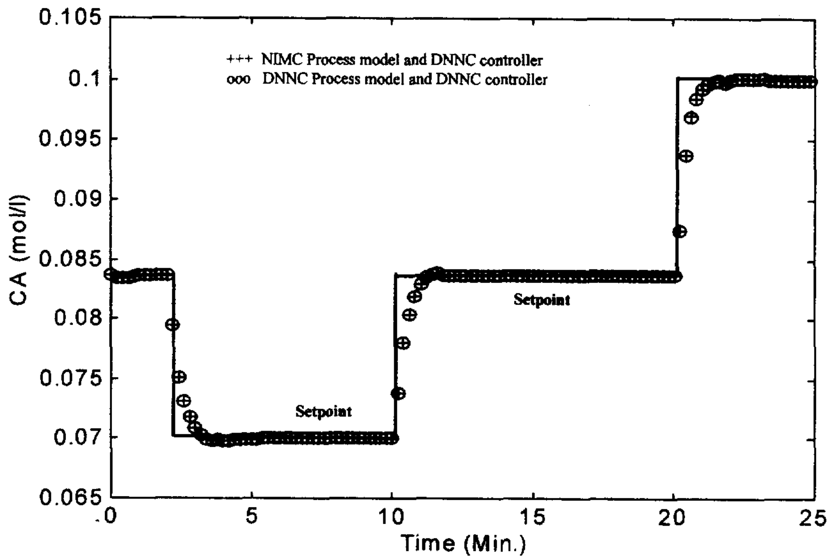

Fig. 7. Setpoint tracking performance, NIMC–DNNC hybrid and DNNC.

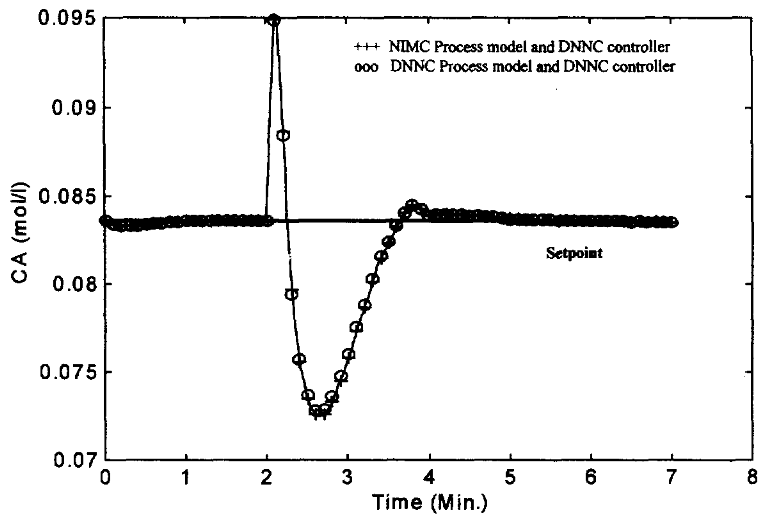

Fig. 8. Disturbance rejection performance, NIMC–DNNC hybrid and DNNC.

The results from the stability analysis were used to define the neural network stability index (NNSI). The NNSI was used to determine the optimal DNNC network structure, to analyze the DNNC controller performance and to design a controller with an acceptable or predefined performaance. In addition, the results from the stability analysis were used to design an optimal DNNC neural network structure for identification and controller purposes for a conventional neural network process model (NIMC as an example) with an equivalent controller performance but less network structure complexity.

The controller strategies, DNNC and the hybrid DNNC–conventional neural network (DNNC–NIMC), were applied to a non-isothermal CSTR as an example of a wide class of non-linear processes. Based on the stability analysis, an optimal neural network structure for an acceptable or predefined controller performance was designed for this process.

5. Notation

B	bias
f	transfer function
J	Jacobian matrix
u	manipulated input
$\Delta u(k)$	change in the input (manipulated variable) defined as $u(k)-(k-1)$
Δuy	as defined in Eq. (1)
x	state variables
y_1	output from hidden layer nodes
y_m	current feedback measurement
y^{set}	setpoint
W	network weights

Subscript

1 input-hidden layer
2 hidden-output layer

Superscript

Y process model
U controller
T transpose
C as defined in Eq. (10)

Greek letters

α as defined in Eq. (8)
β as defined in Eq. (11)
λ eigenvalues
Δ difference
Γ neurons transfer function, as defined by Eq. (2)

Non-isothermal CSTR

A area
C_A concentration of component A
C_p heat capacity
E activation energy
h heat transfer coefficient for CSTR
h_d heat transfer coefficient, scale
k_0 rate constant
q flow rate
T temperature
V volume of the tank

Greek letters

α_c deactivation constant
α_h fouling constant
ΔH heat of reaction
$\phi_c(t)$ deactivation coefficient
$\phi_h(t)$ fouling coefficient, $0<\phi_h<1$
ρ density of reactor content

Subscripts

c coolant
f feed, inlet condition

References

[1] N.V. Bhat, P.A. Minderman, J.T. McAvoy and N.S. Wang, Modeling chemical process systems via neural computaion, IEEE Control System Magazine (1990) 24.

[2] S. Chen, S.A. Billings, P.M. Grant, Non-linear systems identification using neural networks, Int. J. Control 51 (6) (1990) 1191.

[3] K.S. Narendra, K. Parthasarathy, Identification and control of dynamical systems, using neural networks, IEEE Trans. Neural Networks 1 (1990) 4–27.

[4] M. Nikravesh, A.E. Farell, Modeling and controlling non-linear chemical processes using hybrid Neural Network Dynamic Matrix Control algorithm (NNDMC), AIChE Annu. Meet., St. Louis, MO, 1993, paper 213b8.

[5] M. Nikravesh, A.E. Farell, T.G. Stanford, Model identification of nonlinear time variant processes via artifitial neural network, Comput. Chem. Eng. 20 (11) (1996) 1277–1290.

[6] M. Nikravesh, M. Soroush, A.R. Kovscek, T.W. Patzek, Identification and control of industrial-scale processes via neural networks, presented at Chemical Process Control V, Tahoe City, CA, 5–12 January 1996.

[7] M. Nikravesh, M. Soroush, R.M. Johnston, T.W. Patzek, Design of smart wellhead controllers for optimal fluid injection policy and producibility in petroleum reservoirs: a neuro-geometric approach, SPE# 37557, 1997 Int. Thermal Operations and Heavy Oil Symp., 10–12 February, Bakersfield, CA, 1997.

[8] J.F. Pollard, M.R. Broussartd, D.B. Garrison and K.Y. San, Process identification using neural networks, Comput. Chem. Eng. 4 (1992) 253.

[9] C.D. Pschogios, L.H. Ungar, A hybrid neural network—first principles approach to process modeling, AIChE J. 10 (1992) 1499.

[10] C.D. Pschogios, L.H. Ungar, Direct and indirect model based control using artifitial neural networks, Ind. Eng. Chem. Res. 30 (1991) 2564.

[11] M.R. Azimi, R.J. Liou, Fast learning process of multilayer neural networks using recursive least squares method, IEEE Trans. Signal Process. 40 (1992) 446–450.

[12] G. Cybenko, Approximation by superposition of a sigmoidal function, Math. Control Signal System 2 (1989) 303.

[13] R. Hecht-Nielsen, Theroy of bavkpropagation neural networks, IEEE Proc. Int. Conf. on Neural Networks, Washington, DC, 1989, I-593.

[14] R.J. Jannaron, Concurrent Information Processing: Real Time Neurocomputing for Rapidly Changing World, book draft in review/preparation, 1994.

[15] M. Nikravesh, Dynamic neural network control, Ph.D. Dissertation, University of South Carolina, Columbia, SC, 1994.

[16] M. Nikravesh, A.R. Kovscek, T.W. Patzekand, R.M. Johnston, Prediction of formation damage during fluid injection into fractured, low permeability reservoirs via neural networks, SPE 31103, SPE Formation Damage Symp., Lafayette, LA, 1996.

[17] A.R. Kovscek, M. Nikravesh, T.W. Patzek, Smart neural network wellhead control systems for oil fields, DOE (Patent Pending), June 14, 1996.

[18] M. Lee, S. Park, A new scheme combining neural feeddforward control with model predictive control, AIChE J. 38 (2) (1992) 193–200.

[19] E. Nahas, M. Henson, D. Seborg, Non-linear internal model control strategy for neural network models, Comput. Chem. Eng. 16 (12) (1992) 1039.

[20] J. Saint-Donat, N. Bhat, T.J. McAvoy, Neural net based model predictive control, Int. J. Control 54 (6) (1991) 1453.

[21] M. Soroush, C. Kravaris, Discrete-time nonlinear controller synthesis by input/output linearization, AIChE J. 38 (1992) 1923–1945.

[22] M. Soroush, C. Kravaris, Feedforward/feedback control of discrete-time nonlinear systems, 1992 AIChE Annu. Meet., 1–6 November, Miami, FL, 1992, Paper 126e.

[23] M.M. Gupta, Fuzzy logic and fuzzy systems: recent developments and directions, 1996 Biennial Conf. of the North American Fuzzy Information Processing—NAFIPS, Berkeley, CA, 1996, pp. 155–159.

[24] M.M. Gupta, D.H. Rao, Dynamic neural units in the control of linear and nonlinear systems, Int. Joint Conf. on Neural Networks, Baltimore, MD, 7–11 June 1992, Vol. II, pp. 100–105.

[25] L. Jin, M.M. Gupta, P.N. Nikiforuk, 13th Triennial World Congress, San Francisco, CA, 1996, Vol. F, pp. 187–192.

[26] K. Ogata, Discrete Time Control Systems, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[27] R. Langari, W. Li, Analysis and efficient implementation of fuzzy logic control algorithms, 1996 Biennial Conf. of the North American Fuzzy Information Processing—NAFIPS, Berkeley, CA, 1996, pp. 1–4.

[28] J.P. LaSalle, The Stability and Control of Discrete Processes, Springer, New York, 1986.

[29] L. Jin, P.N. Nikiforuk, M.M. Gupta, Absolute stability conditions for discrete-time recurrent neural networks, IEEE Trans. Neural Networks 5 (1994) 945–955.
[30] L. Jin, P.N. Nikiforuk, M.M. Gupta, Dynamics and stability of multilayered recurrent neural networks, Proc. 1993 IEEE Int. Conf. on Neural Networks (ICNN'93), Vol. II, IEEE, New York, 1993, pp. 1135–1140.
[31] C.E. Garcia, M. Morari, Internal model control. 1. A unifying review and some new results, Ind. Eng. Chem. Process Des. 21 (1982) 308.
[32] M. Morari, E. Zafiriou, Robust Process Control, Prentice Hall, Englewood Cliffs, NJ, 1989.
[33] P.B. Deshpande, Multivariable Control Method, ISA, 1988.
[34] P.B. Deshpande, Computer Process Control with Advanced Control Application, ISA, 1988.
[35] G.F. Froment, K.B. Bischoff, Chemical Reactor Analysis and Design, Wiley, New York, 1990, 2nd edn.
[36] R.H. Perry, C.H. Chilton, Chemical Engineering Handbook, McGraw-Hill, 1973, 5th edn.