

Modeling and Control of Nonlinear Discrete-time Systems Based on Compound Neural Networks^{*}

ZHANG Yan (张燕)^{1,**}, LIANG Xiuxia (梁秀霞)¹, YANG Peng (杨鹏)¹, CHEN Zengqiang (陈增强)² and YUAN Zhuzhi (袁著祉)²

¹ Department of Automation, Hebei University of Technology, Tianjin 300130, China

² Department of Automation, Nankai University, Tianjin 300071, China

Abstract An adaptive inverse controller for nonlinear discrete-time system is proposed in this paper. A compound neural network is constructed to identify the nonlinear system, which includes a linear part to approximate the nonlinear system and a recurrent neural network to minimize the difference between the linear model and the real nonlinear system. Because the current control input is not included in the input vector of recurrent neural network (RNN), the inverse control law can be calculated directly. This scheme can be used in real-time nonlinear single-input single-output (SISO) and multi-input multi-output (MIMO) system control with less computation work. Simulation studies have shown that this scheme is simple and affects good control accuracy and robustness.

Keywords adaptive inverse control, compound neural network, process control, reaction engineering, multi-input multi-output nonlinear system

1 INTRODUCTION

Today in process industries, an increasing number of process variables are being measured and stored. In particular, variables in most industrial processes exhibit strong temporal and spatial correlations. Techniques to capture these correlations from available multivariate data in the form of a mathematical model would be highly valuable. Process control of a multi-input multi-output (MIMO) system is difficult, especially when the system exhibits high non-linearity [1–5]. Adaptive inverse control or feedback control has been proposed as a major design method [6–8]. Neural network (NN) control combined with inverse control has been extensively studied for these days [9–12]. One typical inverse control approach is constructing a certainty equivalent control which is usually taken the form $u(k) = \hat{g}^{-1}(k)[y(k+1) - \hat{f}(k)]$, where $\hat{f}(k)$ and $\hat{g}(k)$ are estimations of the nonlinear functions $f(k)$ and $g(k)$. The state observer and critic feedback controller based on NN are designed [13]. In the control process, an action NN is used to minimize both the strategic utility function and the unknown dynamics estimation errors.

The other approach is by using a linear state observer and an error compensator to approximate the error between the linear model and the process based on NN. The nonlinear dynamics are inverted directly [14]. The extension of output feedback employing a high-gain observer is given [15]. An error observer approach with e -modification and a filter with nonlinearly parameterized NN and modification are introduced to decrease oscillation. A linear dynamic compensator is designed to stabilize the linearized system [16]. A temperature controller with Takagi-Sugeno-Kang-type recurrent fuzzy network (TRFN)

designed by direct inverse modeling approach is proposed [17]. The recurrent property of TRFN enables it to be directly applied to a dynamic plant control without a priori knowledge of the plant order. An application of neural networks based additive nonlinear autoregressive exogenous (NNANARX) structure for modeling of nonlinear MIMO systems is presented [18]. Then, the problem of the inverse function calculation in the control algorithm is solved and applied to control nonlinear MIMO systems. Most of these approaches are limited to system in strict-feedback form, and accuracy of the plant model is critical when an inverse is used.

Motivated by this, a novel NN based adaptive inverse control approach is proposed for both SISO and MIMO nonlinear nonaffine discrete systems in this paper. A compound neural network (CNN) is constructed to model the system and an adaptive inverse control strategy is presented. Simulation results can demonstrate the effectiveness and advantages.

2 SYSTEM DESCRIPTION AND PRELIMINARIES

2.1 System representation and identification

The following SISO nonaffine nonlinear discrete-time system is considered [16]:

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ x_2(k+1) &= x_3(k) \\ &\vdots \\ x_n(k+1) &= s[X(k), u(k)] \\ y(k+1) &= h[X(k)] \end{aligned} \quad (1)$$

Received 2008-06-24, accepted 2009-03-20.

^{*} Supported by the National Natural Science Foundation of China (60575009, 60574036).

^{**} To whom correspondence should be addressed. E-mail: yzhangzz@yahoo.com.cn

where $X(k) = [x_1(k), x_2(k), \dots, x_n(k)]^T \in R^n$ is the internal state of the system, $u(k) \in R$ is the input to the system, $y(k) \in R$ is the measured output, and $s(\cdot)$ and $h(\cdot)$ are unknown smooth nonlinear functions.

For convenience of analysis, the future output is determined by a number of past observations of the inputs and outputs. An equivalent input-output representation can be written as the nonlinear auto regressive moving average with exogenous inputs (NARMAX) system:

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n+1), u(k), u(k-1), \dots, u(k-m+1)] \quad (2)$$

Where n and m are the highest orders of the output and input, respectively, $f(\cdot)$ is the smooth nonlinear function.

Classically this controlled system can be realized by the NN-based model at moment k :

$$\hat{y}(k) = f_{NN}[y(k-1), y(k-2), \dots, y(k-n), u(k-1), u(k-2), \dots, u(k-m), W] \quad (3)$$

Where $f_{NN}(\cdot)$ is a nonlinear mapping function used to approximating the smooth nonlinear function $f(\cdot)$, W is the weight matrix of the whole network and $\hat{y}(k)$ is the output of NN.

2.2 Compound neural network

To identify the system shown in Eq. (2), many kinds of NNs can be used. Here, in order to get an inverse controller directly and avoid using another control neural network, a compound neural network (CNN) is proposed. Its structure is shown in Fig. 1. It is composed of two parts: a two-layer linear feedforward neural network (LFNN) and a recurrent neural network (RNN). The output of the identifier can be expressed as:

$$\hat{y}(k) = y_L(k) + y_N(k) \quad (4)$$

where $y_L(k)$ is the output of LFNN, and $y_L(k) = g_L[I_L(k-1)]$. Its input vector is defined as $I_L(k-1) = [y(k-1), \dots, y(k-n), u(k-1), \dots, u(k-m)]^T$. $y_N(k)$ is the output of RNN, whose input vector is defined as: $I_N(k-1) = [y(k-1), y(k-2), \dots, y(k-n), u(k-2), \dots, u(k-m)]^T$. RNN is used to approximate the nonlinear

character of the system and minimize the error between $y_L(k)$ and the system output $y(k)$, and $y_N(k) = g_N[I_N(k-1)]$.

Recurrent neural network (RNN) has widely been used because it is more applicable to control dynamic nonlinear problems in real time [19]. Suppose $\Psi_l(k)$ and $\Phi_l(k)$ represent the state and output vectors of the l th layer, respectively, with $\Phi_l(k) = [\phi_l^1, \phi_l^2, \dots, \phi_l^{N_l}]$ and $\Psi_l(k) = [\varphi_l^1, \varphi_l^2, \dots, \varphi_l^{N_l}]$, RNN can be described as follows:

$$\Psi_l(k) = W_{l,l-1}\Phi_{l-1}(k) + W_{l,l}\Phi_l(k-1) + b_l \quad l = 2, \dots, M \quad (5)$$

$$\Phi_l(k) = H[\Psi_l(k)] \quad l = 2, \dots, M \quad (6)$$

Where $l = 2, \dots, M$ is the number of layers. $W_{l,l-1}$ is the weight matrix associated with the link between the nodes of the $(l-1)$ th layer to the nodes of the l th layer. $W_{l,l}$ is the self-feedback links between the nodes of the l th layer. b_l is the bias vector to the nodes of the l th layer. $H(\cdot)$ is a sigmoidal activation function.

The identifier output is rewritten as follows:

$$\begin{aligned} \hat{y}(k) &= g_L[I_L(k-1)] + g_N[I_N(k-1)] \\ &= a_1 y(k-1) + a_2 y(k-2) + \dots + a_n y(k-n) + \\ &\quad b_1 u(k-1) + b_2 u(k-2) + \dots + b_m u(k-m) + \\ &\quad g_N[y(k-1), \dots, y(k-n), u(k-2), \dots, u(k-m)] \end{aligned} \quad (7)$$

where $a_i, i=1, \dots, n$ and $b_i, i=1, \dots, m$ are the weight coefficients of the LFNN in the CNN.

In the approximation process, suppose $W(k)$ is the weight matrix of the whole CNN. It is trained by minimizing the following index function.

$$J = \frac{1}{2} [y(k) - \hat{y}(k)]^2 \quad (8)$$

Define $e(k) = y(k) - \hat{y}(k)$. Then, the gradient descent method can be applied as follows:

$$\begin{aligned} \Delta W(k) &= -\eta \cdot \partial J(k) / \partial W(k) \\ &= -\eta \partial J(k) / \partial e(k) \cdot \partial e(k) / \partial W(k) \\ &= \eta \cdot e(k) \cdot \partial \hat{y}(k) / \partial W(k). \end{aligned} \quad (9)$$

2.3 Error correction

The error between the identifier and the actual system exists inevitably. To solve the problem as the output at time $k+1$ is predicted, we add the error at time k , and form a one-step predictive error. For the SISO case, the one-step predictive output of the system can be written as:

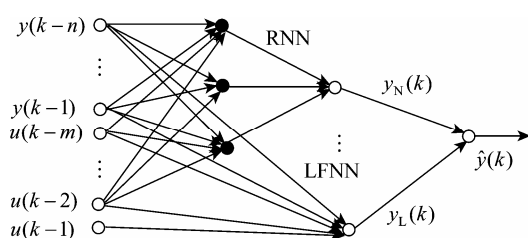


Figure 1 The structure of compound neural network

$$y(k+1) = \hat{y}(k+1) + c \cdot e(k) \\ = \hat{y}(k+1) + c[y(k) - y_L(k) - y_N(k)] \quad (10)$$

Where $c > 0$ is the error proportional coefficient.

3 ADAPTIVE INVERSE CONTROLLER

The control objective is to find a control input $u(k)$ to assure that the system output tracks the desired trajectory $y_d(k)$ with an acceptable accuracy, while all signals in the closed-loop system remain bounded. The system output at time $k+1$ can be written as:

$$y(k+1) = \hat{y}(k+1) + c \cdot e(k) \\ = y_L(k+1) + y_N(k+1) + c \cdot e(k) \\ = a_1 y(k) + a_2 y(k-1) + \dots + a_n y(k-n+1) + \\ b_1 u(k) + b_2 u(k-1) + \dots + b_m u(k-m+1) + \\ g_N[y(k-1), \dots, y(k-n), u(k-2), \dots, \\ u(k-m)] + c[y(k) - y_L(k) - y_N(k)] \quad (11)$$

Unlike the general NARMAX model, this compound structure can be easily controlled by the dynamic feedback theory. If the model is given by Eq. (2), the control signal is calculated by the following equation:

$$u(k) = f^{-1}[y(k+1), I_N(k)] \quad (12)$$

From Eq. (11), the desired output feedback is given by

$$u(k) = \frac{1}{b_1} \{ y_d(k+1) - g_N[I_N(k)] - \\ a_1 y(k) - a_2 y(k-1) - \dots - a_n y(k-n+1) - \\ b_2 u(k-1) - \dots - b_m u(k-m+1) - ce(k) \} \quad (13)$$

where $y_d(k+1)$ is the reference trajectory at time $k+1$ to replace the actual output $y(k+1)$. Fig. 2 is the architecture of the proposed adaptive inverse control for nonlinear system. To sum up, the objectives of the control process include the following:

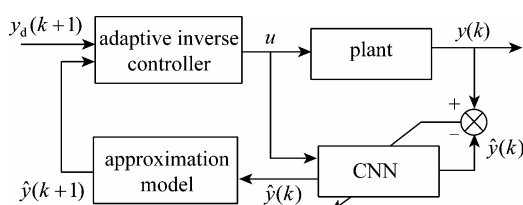


Figure 2 Control system architecture

Step 1 Initialize the structure of the compound neural network. Select the relative coefficient: the learning rate η , the initial weight vector of CNN, and the error proportional coefficient c .

Step 2 Firstly, train the LFNN off-line before the error between its output and the system output satisfies the settled value. Secondly, the LFNN com-

bined with the RNN whose input vector doesn't include the control signal $u(k-1)$ is built to approximate the real process off-line. The weight of CNN is updated according to Eq.(9).

Step 3 Put the trained CNN into the closed loop control, and then, the control signal at time k can be calculate by Eq. (13).

Step 4 Update the CNN weight vector by Eq. (9).

Step 5 Let $k = k+1$ and go to Step 2.

4 MIMO SYSTEM INVERSE CONTROLLER

The controlled nonlinear MIMO system with r inputs and r outputs is considered. The controlled system can be represented by the NARMAX model as follows:

$$Y(k) = F[Y(k-1), \dots, Y(k-n), U(k-1), \dots, U(k-m)] \quad (14)$$

Where $F(\cdot)$ is a smooth nonlinear function vector,

$$Y(k) = [y_1(k), \dots, y_r(k)]^T \text{ and } U(k) = [u_1(k), \dots, u_r(k)]^T.$$

Similarly as SISO nonlinear system, a compound neural network can be used to approximate the MIMO NARMAX model. The input vector of the LFNN and RNN are defined as:

$$I_L(k-1) = [Y^T(k-1), \dots, Y^T(k-n), U^T(k-1), \dots, U^T(k-m)]^T \\ I_N(k-1) = [Y^T(k-1), \dots, Y^T(k-n), U^T(k-2), \dots, U^T(k-m)]^T$$

The output of CNN can be written as follows:

$$\hat{Y}(k) = G_L[I_L(k-1)] + G_N[I_N(k-1)] \quad (15)$$

In the approximation process, $W(k)$ is supposed to be the weight matrix of the whole CNN. The gradient decent method is used to train $W(k)$.

In the control process, as SISO system, the error at time k is introduced to predict the output of the system. The predictive output can be written as:

$$Y(k+1) = \hat{Y}(k+1) = c \cdot E(k) \\ = G_L[I_L(k)] + G_N[I_N(k)] + c \cdot E(k) \quad (16)$$

where $E(k)$ is the error vector at time k , $E(k) = Y(k) - \hat{Y}(k) = Y(k) - Y_L(k) - Y_N(k)$ and $c > 0$ is the error proportional coefficient.

Because the current control signal $U(k)$ is one part of the input vector of LFNN in the CNN, the control law for the MIMO case can be determined straightforwardly under the inverse control thought:

$$U(k) = F^{-1}[Y_d(k+1), I_N(k)] \quad (17)$$

where $Y_d(k+1)$ is the reference trajectory at time $k+1$.

5 SIMULATION RESULTS

Example 1 A liquid level system is described by the following equations [16]:

$$\begin{aligned} x_1(k+1) &= x_2(k) \\ x_2(k+1) &= 0.9722x_2(k) - 0.3515x_2^2(k)u(k-1) - \\ &\quad 0.3103x_2(k)u(k) + 0.1633x_1(k)u(k-1) - \\ &\quad 0.03259x_2^2(k)x_1(k) - 0.1295u(k-1) + \\ &\quad 0.3084x_2(k)x_1(k)u(k) - 0.04228x_1^2(k) + \\ &\quad 0.1087x_1(k)u(k)u(k-1) + 0.3578u(k) \\ y(k) &= x_2(k) \end{aligned} \quad (18)$$

The NARMAX model of the process should be written as:

$$\begin{aligned} y(k+1) &= 0.9722y(k) + 0.3578u(k) - 0.1295u(k-1) + \\ &\quad 0.3084y(k)y(k-1)u(k-1) - \\ &\quad 0.04228y^2(k-1) + 0.1633y(k-1)u(k-1) - \\ &\quad 0.03259y^2(k)y(k-1) + \\ &\quad 0.3513y^2(k)u(k-1) - 0.3103y(k)u(k) + \\ &\quad 0.1087y(k-1)u(k)u(k-1) \end{aligned} \quad (19)$$

To control this nonlinear process, the control process is divided into the following important steps. Assume the input signal applied to the controlled system is a finite sequence of uniformly distributed random variables with range $[-1,1]$, which generates input/output samples (patterns) to train the identification neural network CNN. There are 9 nodes in the hidden layer of RNN in the CNN is used. The learning rate η is defined as 0.18 and the initial network weight values are in the intervals $[-1,1]$. The CNN training capability is shown in Fig. 3.

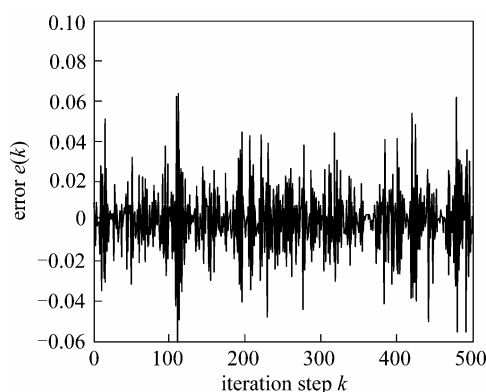
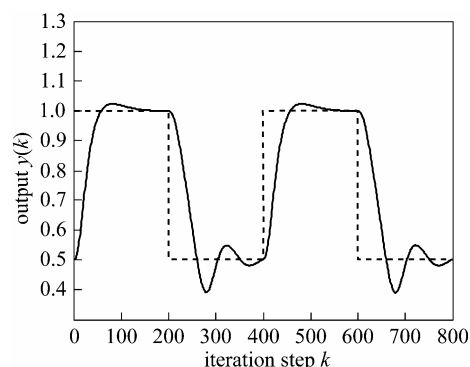


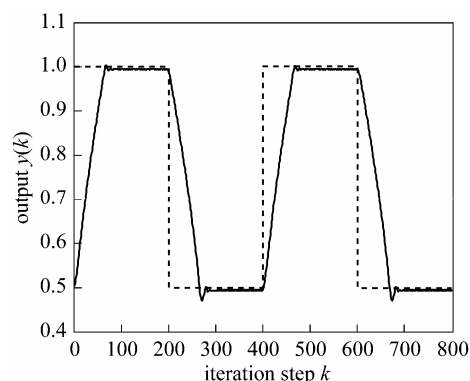
Figure 3 CNN training error

From Fig. 3, it can be seen that the CNN has the accurate identification capability. The nonlinear model of the plant was placed into the inverse control loop.

The initial conditions of the plant are set to random variables. Here, the parameters of the CNN used in the inverse control are the same as in the CNN training. The error proportional coefficient c is set as 0.7. Fig. 4 presents the control results and the control signal in the process obtained by the proposed inverse control method for tracking square signal. It shows a good tracking result.



(a) The minimum variance control based on RNN



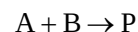
(b) Adaptive inverse control based on CNN

Figure 4 System response of Example 1

— $y(k)$; ---- $y_d(k)$

Example 2 Isothermal reactor [20]

The following reaction occurs in an ideal stirred tank reactor:



where A is in excess. The reaction rate equation is given by the following equation:

$$r_B = \frac{K_1 c_B}{(1.0 + K_2 c_B)^2} \quad (20)$$

where c_B is the concentration of component B. The concentration of B in the two inlet streams are assumed constant at $c_{B1} = 24.9 \text{ mol} \cdot \text{L}^{-1}$ and $c_{B2} = 0.1 \text{ mol} \cdot \text{L}^{-1}$, and contain an excess amount of A. The flow of liquid from the tank is determined by Eq. (21).

$$F(h) = 0.2 \cdot h^{0.5} \quad (21)$$

The cross-sectional area of the tank is 1.0 m^2 and the sampling time is 1.0 min. After simplification, the model becomes:

$$\frac{dc_1}{dt} = v_1 + v_2 - 0.2c_1^{0.5}$$

$$\frac{dc_2}{dt} = \frac{(24.9 - c_2)v_1}{c_1} + \frac{(0.1 - c_2)v_2}{c_1} - \frac{c_2}{(1 + c_2)^2}$$

$$y_1 = c_1, \quad y_2 = c_2 \quad (22)$$

where c_1 is the liquid height in tank in meters, c_2 , the concentration of B in the reactor in $\text{mol} \cdot \text{L}^{-1}$, v_1 , the inlet flow rate with condensed B in $\text{m}^3 \cdot \text{s}^{-1}$, and v_2 , the inlet flow rate with dilute B in $\text{m}^3 \cdot \text{s}^{-1}$. A factor of 1000 m^{-3} cancels out in Eq. (22). Initially, the process is at steady-state with $c_1 = 40 \text{ m}$ and $c_2 = 0.1 \text{ mol} \cdot \text{L}^{-1}$.

In the process of the model for the MIMO nonlinear system, 12 nodes are chosen in the hidden layer of RNN for the CNN identifier. First, the following input signals are used to approximate the MIMO nonlinear discrete systems:

$$u_1(k) = 0.3 + \sin(2\eta k/10) + \sin(2\eta k/25)$$

$$u_2(k) = 0.2 + \cos(2\eta k/10) + \cos(2\eta k/25)$$

In this case, the learning rate η is selected as 0.2. From Fig. 5, it can be seen that the CNN model can describe the MIMO nonlinear system with high accuracy.

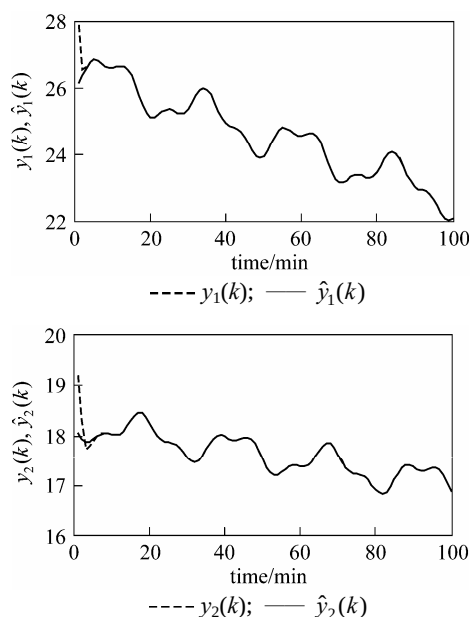
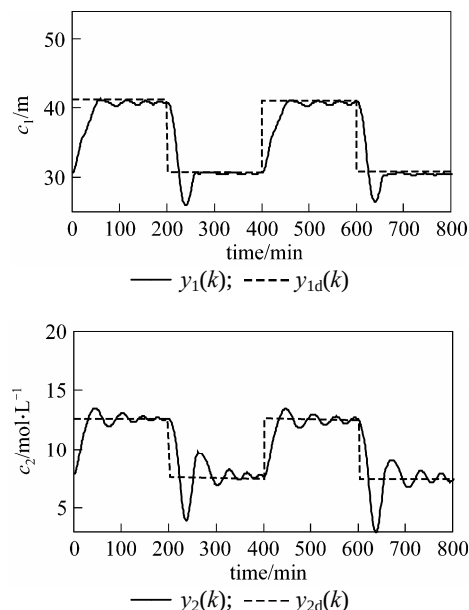


Figure 5 Identification results of Example 2

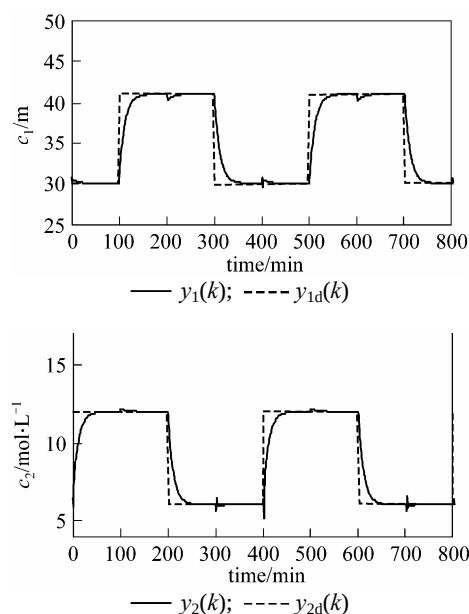
Second, the trained CNN is put into the closed-loop inverse control. The control results are illustrated in Fig. 6. It shows that the system can track the reference signals with acceptable approximation errors. The proposed CNN inverse control method shows good tracking performance for the MIMO nonlinear chemical system.

6 CONCLUSIONS

An adaptive inverse control scheme has been



(a) The minimum variance control based on RNN



(b) Adaptive inverse control based on CNN

Figure 6 Tracking performance of Example 2

proposed in this paper. To use inverse theory directly, a new compound neural network is proposed. The linear feed-forward neural network is used to approximate the nonlinear controlled process. The recurrent neural network is used to minimize the error between the LFNN and the real nonlinear process. Based on this kind of neural network to approximate the controlled process, an adaptive inverse control scheme can be directly implemented. During this process, an error correction method is proposed to reduce the predictive error. The less computation work is needed since no further training task is required for the neural inverse controller in the I/O domain. This

scheme can be used to control both nonlinear dynamic discrete-time SISO and MIMO systems in real time. Simulation results exploit that the proposed scheme is effective and practical.

NOMENCLATURE

c	the error proportional coefficient
c_B	concentration of B in the reactor example
e	error between the system output and the identifier
F	a smooth nonlinear function vector
f	a smooth nonlinear function
G_E	a smooth linear function vector
G_N	a smooth nonlinear function vector
g_L	a smooth linear function
g_N	a smooth nonlinear function
H	a sigmodal activation function
I_L	the input vector of the LFNN in CNN
I_N	the input vector of the RNN in CNN
M	the number of layers of RNN
m	the degree of system input u
n	the degree of system output y
s	a smooth nonlinear function
u	the system input
W	weight matrix of CNN
$\hat{Y}$	the CNN's output in case of MIMO
y	the system output
$\hat{y}$	the CNN's output
y_L	the LFNN's output
y_N	the RNN's output
η	the learning rate for the weight vector W
Φ_l	the output vector of the l th layer for RNN
Ψ_l	the state vector of the l th layer for RNN

Subscripts

B	the tank B
L	the LFNN in the CNN
N	the RNN in the CNN

REFERENCES

- 1 Fu, Y., Chai, T.Y., "Nonlinear multivariable adaptive control using multiple models and neural Networks", *Automatica*, **43**, 1101–1110 (2007).
- 2 Zhang, Y., Chen, Z.Q., Yang, P., Yuan, Z.Z., "Multivariable nonlinear proportional-integral-derivative decoupling control based on recurrent neural net works", *Chin. J. Chem. Eng.*, **12** (5), 677–681 (2004).
- 3 Wang, Z., Chen, Z.Z., Sun, Q.L., Yuan, Z.Z., "Multivariable decoupling predictive control based on QFT theory and application in CSTR chemical process", *Chin. J. Chem. Eng.*, **14** (6), 765–769 (2006).
- 4 Zhang, Q., Li, S., "Performance monitoring and diagnosis of multi-variable model predictive control using statistical analysis", *Chin. J. Chem. Eng.*, **14** (2), 207–215 (2006).
- 5 Su, B.L., Chen, Z.Z., Yuan, Z.Z., "Multivariable decoupling predictive control with input constraints and its application on chemical process", *Chin. J. Chem. Eng.*, **14** (2), 216–222 (2006).
- 6 Widrow, B., W alach, E., *Adaptive Inverse Control*, Prentice Hall, New Jersey, US (1986).
- 7 Alolinwi, B., Khalil, H.k., "Robust adaptive output feedback control of nonlinear systems without persistence of excitation condition", *Automatica*, **33**, 2025–2032 (1997).
- 8 Tong, S.C., Chai, T.Y., "Direct adaptive fuzzy output feedback control for uncertain nonlinear systems", *Control and Decision*, **19** (3), 257–261 (2004).
- 9 Ge, S.S., Li, Y., Lee, T.H., "daptive NN control for a class of strict-feedback discrete-time nonlinear systems" *Automatica*, **39** (5), 807–819 (2003).
- 10 Miguel, A.B., Ton, J.J., Van, D.B., "Predictive control based on neural network model with I/O feedback linearization", *Int. J. Control*, **72** (17), 1358–1554 (1999).
- 11 Song, Y., Chen, Z.Q., Yuan, Z.Z., "Neural network nonlinear predictive control based on tent-map chaos optimization", *Chin. J. Chem. Eng.*, **15** (4), 539–544 (2007).
- 12 Deng, H., Li, H.X., "A novel neural Approximate inverse control for unknown nonlinear discrete dynamical Systems", *IEEE Tran. System, Man, Cybernetics B: Cybernetics*, **35** (1), 115–123 (2005).
- 13 He, P., Jagannathan, S., "Reinforcement learning-based output feedback control of nonlinear systems with input constraints", *IEEE Trans. Systems, Man, Cybernetics B*, **35** (1), 150–154 (2005).
- 14 Hovakimyan, N., Nardi, F., Calise, A.J., "A novel error observer-based adaptive output feedback approach for control of uncertain systems", *IEEE Trans. Automatic Control*, **47** (8), 1310–1314 (2002).
- 15 Kim, N., Calise, A.J., "Several extensions in methods for adaptive output feedback control", *IEEE Trans. Neural Networks*, **18** (2), 482–494 (2007).
- 16 Zhai, L.F., Chai, T.Y., Ge, S.S., "Stable adaptive neural network control of nonaffine nonlinear discrete-time systems and application", In: 22th IEEE International Symposium on Intelligent Control, Singapore, 602–607 (2007).
- 17 Juang, C.F., Chen, J.S., "A recurrent fuzzy-network-based inverse modelling method for a temperature system control", *IEEE trans. Systems, Man, Cybernetics C*, **37** (3), 410–417 (2007).
- 18 Petlenkov, E., "NN-ANARX structure based dynamic output feedback linearization for control of nonlinear MIMO systems", In: Mediterranean Conference on Control and Automation, Athens, Greece, T22–009 (2007).
- 19 Delgado, A., "Dynamic recurrent neural networks for system identification and control", *IEE Proceedings Control Theory Applications*, **142** (4), 307–314 (1995).
- 20 Li, W.C., Biegler, L.T., "Process control strategies for constrained nonlinear system", *Ind. Eng. Chem. Res.*, **27**, 1611–1622 (1988).