Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Neurocomputing 70 (2006) 430–444

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Multiple recurrent neural networks for stable adaptive control

Wen Yu*

Departamento de Control Automatico, CINVESTAV-IPN, A.P. 14-740, Av. IPN 2508, México D.F. 07360, México

Received 8 March 2005; received in revised form 14 November 2005; accepted 19 December 2005
Communicated by D. Erdogmus
Available online 15 May 2006

Abstract

It is difficult to realize adaptive control for some complex nonlinear processes which are operated in different environments and when operation conditions are changed frequently. In this paper we propose an identifier-based adaptive control (or indirect adaptive control). The identifier uses two effective tools: multiple models and neural networks. A hysteresis switching algorithm is applied to select the best model. The adaptive controller also has a multi-model structure. We introduced three different multi-model neuro controllers. The convergence of the neuro identifier, switching property and the stability of neuro control are proved. Numerical simulations are given to illustrate the performances of multiple neural identifiers and neural adaptive control on a pH neutralization process.

Keywords: Multiple models; Neural control; Stability

1. Introduction

Adaptive control of nonlinear systems has been an active area in recent years. It is difficult to control unknown plants. A common approach to deal with this problem is to utilize the simultaneous identification technique. Neural networks have been employed in the identification and control of unknown nonlinear systems owing to their massive parallelism, fast adaptation and learning capability. Several neural adaptive approaches for nonlinear systems are developed. Lyapunov synthesis approach is most popular tool [14]. For examples, robust neural control based on a modified Lyapunov function is given in [31], the singularity issue is completely avoided. By Lyapunov–Krasovskii functions, adaptive neural control with unknown time delays is presented in [4]. An adaptive output feedback controller for a class of uncertain stochastic nonlinear systems is presented in [1], where the weights of the neural network are tuned adoptively by a Lyapunov design. Using parameter projection and high-gain observer, output feedback neural control is uniform bounded [20]. In our previous works [15,28], stability analysis of the identification error is performed by a Lyapunov analysis with dead-zone. In this paper, Lyapunov stability analysis methodology and the robustness analysis of neural information storage [25] are applied.

In many cases, the plant to be controlled is too complex to find the exact system dynamics, and the operating conditions in dynamic environments may be unexpected. Therefore, adaptive control technique has been combined with multiple models. The first multi-model approach may be found in [8] where multiple Kalman filters were used to improve the accuracy of the state estimation. The switching for multi-model was first introduced in [10] when the unknown linear systems can be stabilized by the use of adaptive schemes. More general versions of continuous-time and discrete-time multi-model adaptive controllers can be found in [12,13]. Stability analysis of multi-estimators for adaptive control with reduce model is proposed in [2]. Since the multiple models may describe more complex behavior of the dynamic systems, the transient performance of adaptive control can be improved [19]. A comprehensive survey on nonlinear process identification with multiple models may be found in [3].

The combination of multi-model and neural networks should be a rational approach for nonlinear system identification. Kikens and Karim [7] used several static

*Tel.: +52 55 5061 3734; fax: +52 55 5747 7089.
E-mail address: yuw@ctrl.cinvestav.mx.

0925-2312/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2005.12.122

neural networks as multi-model identifier, the switching algorithm was realized by a gating neural networks, but the stability analyses was not presented. Multi-model identification and failure detection using static neural networks is presented in [18]. In [6], a hierarchical mixture of experts method combining input/output space is employed. C-Y. Lee and J-J. Lee [9] proposes adaptive feedback linearizing controller where nonlinearity terms are approximated with multiple neural networks. They conclude that the closed-loop system is globally stable in the sense that all signals involved are uniformly bounded. Another type of multiple neural networks for adaptive control is adaptive critic neural networks [26]. The adaptive critic method determines optimal control laws for a system by successively adapting two neural networks, namely an action neural network (which dispenses the control signals) and a critic neural network (which "learns" the desired performance index for some function associated with the performance index). These two neural networks approximate the Hamilton–Jacobi equation associated with optimal control theory. During the adaptations, neither of the networks need any "information" of an optimal trajectory, only the desired cost needs to be known. This technique of neuro controller design does not require continual on-line training thus overcoming the risks of instability [16].

In this paper we propose novel identification and adaptive control approaches. To the best of our knowledge, multiple models approach for dynamic neural networks has not yet been published in the literature. Based on this multiple neuro identifier, we study indirect adaptive control. Three different multi-model neuro controllers are considered. First, an direct linearization controller is obtained from one neuro identifier, the uncertainties compensation control uses multi-model technique. Second, the linearization controller is obtained from multiple neuro identifiers, the uncertainties compensation control uses classical control technique. Third, the multiple neuro identifiers and multiple controller are used. These approaches can overcome the bad transient response and big steady-state error caused by unmodeled dynamic in compensators and identifiers. The simulation experiments illustrate the effectiveness of the multiple neuro identifiers and adaptive controllers.

2. System identification with single dynamic neural networks

Consider a nonlinear process given by

$$\dot{x}_t = f(x_t, u_t), \tag{1}$$

where $x_t \in \Re^n$ is the state, $u_t \in \Re^k$ is the input vector. $f : \Re^n \times \Re^k \to \Re^n$ is a bounded locally Lipschitz and general smooth function.

Let us consider the following dynamic neural network to identify the nonlinear process (1):

$$\dot{\widehat{x}}_t = A\widehat{x}_t + W_t\phi(V_t\widehat{x}_t) + U_t, \tag{2}$$

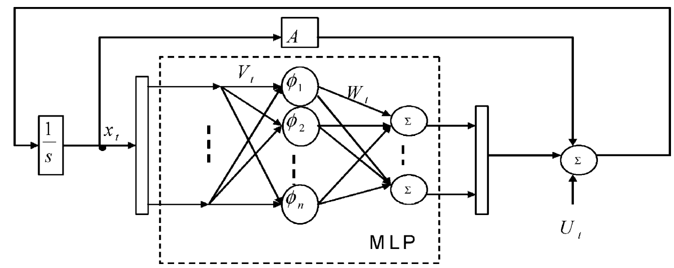


Fig. 1. The structure of multi-layer dynamic neural identifier.

where $\widehat{x}_t \in \Re^n$ is the state of the neural network, $A \in \Re^{n\times n}$ is a known stable matrix which will be specified. The matrix $W_t \in \Re^{n\times m}$ is the weight of the output layer, $V_t \in \Re^{m\times n}$ is the weight of the hidden layer. $U_t = [u_1, u_2, \ldots, u_k, 0, \ldots, 0]^{\mathrm{T}} \in \Re^n$. $\phi(\cdot) \in \Re^m$ is neural network activation function. The elements of $\phi_i(\cdot)$ can be any stationary, bounded and monotone increasing functions. In this paper we use sigmoid functions. The structure of this dynamic neural network is shown in Fig. 1. Generally the multi-layer dynamic neural networks (2) cannot match the given nonlinear system (1) exactly, the nonlinear system (1) can be represented as

$$\dot{x}_t = Ax_t + W^0\phi(V^0x_t) + U_t - \mu_t, \tag{3}$$

where μ_t is defined as modeling error, W^0 and V^0 are set of unknown weights which can minimize the modeling error μ_t. The identified nonlinear system (1) can also be written as

$$\dot{x}_t = Ax_t + W^*\phi(V^*x_t) + U_t - \widetilde{f}_t, \tag{4}$$

where $\widetilde{f}_t$ is modeling error, V^* and W^* are set of known weights chosen by the user. In general, $\|\widetilde{f}_t\| \geqslant \|\mu_t\|$. V^* does not effect the stability property of the neuro identification (see Theorem 1), but it influences the identification accuracy. We can use any off-line method to find a better value for V^*, more detail procedure can be found in [29]. For the modeling error $\widetilde{f}_t$, we make the following assumption [1,9,14,31].

A1:

$$\|\widetilde{f}_t\|^2_{\Lambda_1} = \widetilde{f}_t^{\mathrm{T}}\Lambda_1\widetilde{f}_t \leqslant \overline{\eta} < \infty, \tag{5}$$

here $\overline{\eta}$ is a known positive matrix, Λ_1 is any positive definite matrix.

Remark 1. For identification we assume that plant (1) is bounded-input and bounded-output (BIBO) stable system, i.e., U_t and x_t in (4) are bounded. By the bound of the active function ϕ, $\widetilde{f}_t$ is bounded.

One may see that Hopfield model [17] is a special case of this kind of neural networks with $m = n$ and $V = I$, $A = diag\{a_i\}$, $a_i = -1/R_iC_i$, $R_i > 0$ and $C_i > 0$. R_i and C_i are the resistance and capacitance at the ith node of the network, respectively. The single layer dynamic neural networks discussed by [15,17] are also the cases with $V = I$. The continuous-time feedforward multi-layer networks [25] are subnets of it, they are MLP in Fig. 1. (2) is linear in the input in order to simplify neuro controller, but may give a bigger $\widetilde{f}_t$ when u_t is nonlinear in (1).

Let us define the identification error as

$$\Delta_t = \widehat{x}_t - x_t. \tag{6}$$

It is clear that the sigmoid functions $\phi(\cdot)$ satisfy following generalized Lipschitz conditions:

$$\widetilde{\phi}_t^{\mathrm{T}} \Lambda_1 \widetilde{\phi}_t \leqslant \Delta_t^{\mathrm{T}} \Lambda_\phi \Delta_t, \quad \widetilde{\phi}'_t = D_\phi \widetilde{V}_t \widehat{x}_t + v_\phi, \tag{7}$$

where $\Lambda_\phi = \Lambda_\phi^{\mathrm{T}} > 0$ is a known matrix, $\widetilde{\phi}_t = \phi(V^* \widehat{x}_t) - \phi(V^* x_t)$, $\widetilde{\phi}'_t = \phi(V_t \widehat{x}_t) - \phi(V^* \widehat{x}_t)$, $\widetilde{V}_t = V_t - V^*$, and

$$D_\phi = \left.\frac{\partial \phi^{\mathrm{T}}(Z)}{\partial Z}\right|_{Z = V_t \widehat{x}_t}, \quad \|v_\phi\|^2_{\Lambda_1} \leqslant l_1 \|\widetilde{V}_t \widehat{x}_t\|^2_{\Lambda_1}, \tag{8}$$

l_1 is a positive constant. The first inequality of (7) is the matrix form of Lipschitz condition, the second equation of (7) can be regarded as Taylor formula.

Let us discuss the following Riccati equation:

$$A^{\mathrm{T}} P + PA + PRP + Q = 0. \tag{9}$$

It is known [27] that when the matrix A is stable, the pair $(A, R^{1/2})$ is controllable, the pair $(Q^{1/2}, A)$ is observable, and the local frequency condition

$$A^{\mathrm{T}} R^{-1} A - Q \geqslant \tfrac{1}{4}\left[A^{\mathrm{T}} R^{-1} - R^{-1} A\right] R \left[A^{\mathrm{T}} R^{-1} - R^{-1} A\right]^{\mathrm{T}}$$

is satisfied, then (9) has a positive definite solution $P = P^{\mathrm{T}} > 0$. This condition is easily fulfilled if we select A as a stable diagonal matrix. So we can choose a strictly positive matrix Q_1 such that the matrix Riccati equation (9) with

$$R = 2\overline{W} + \Lambda_1^{-1}, \quad Q = Q_1 + \Lambda_\phi \tag{10}$$

has a positive solution. Here $\overline{W} = W^* \Lambda_1^{-1} W^{*\mathrm{T}}$.

The first contribution of this paper is that a stable learning law for multi-layer dynamic neural networks (2) is proposed as

$$\dot{W}_t = -K_1 P \phi \Delta_t^{\mathrm{T}} - K_1 P D_\phi \widetilde{V}_t \widehat{x}_t \Delta_t^{\mathrm{T}},$$

$$\dot{V}_t = -K_2 P W_t D_\phi \Delta_t \widehat{x}_t^{\mathrm{T}} - K_2 \frac{l_1}{2} \Lambda_1 \widetilde{V}_t \widehat{x}_t \widehat{x}_t^{\mathrm{T}}, \tag{11}$$

where $K_1, K_2 \in \Re^{n \times n}$ are positive definite matrices, P is the solution of the matrix Riccati equation given by (9), ϕ is $\phi(V_t \widehat{x}_t)$. The initial conditions are W_0 and V_0.

Remark 2. From the updating law for the hidden layer V_t we can see that $W_t D_\phi \Delta_t$ is the backpropagation error, $\widehat{x}_t^{\mathrm{T}}$ and $\phi(V_t \widehat{x}_t)$ are the input and output of this layer. So the first parts $-K_1 P \phi \Delta_t^{\mathrm{T}}$ for W_t and $-K_2 P W_t D_\phi \Delta_t \widehat{x}_t^{\mathrm{T}}$ for V_t are the same as the backpropagation scheme of the multi-layer perceptrons (MLP in Fig. 1). The second parts $K_1 P D_\phi \widetilde{V}_t \widehat{x}_t \Delta_t^{\mathrm{T}}$ and $l_1/2 K_2 \Lambda_1 \widetilde{V}_t \widehat{x}_t \widehat{x}_t^{\mathrm{T}}$ are used to assure the stability properties of identification error. Even though the proposed learning law looks like the backpropagation algorithms, the stability of identification error in the sense L_∞ is guaranteed because of the fact that it is derived based on the Lyapunov approach (see the next theorem). The local minima problem (which is a major concern in static neural networks learning) does not arise in this case. However, we can only guarantee convergence to a region whose size can be very large and whose properties we do not know. For example, it could contain several local minima associated with the conventional cost function and it could be that the final solution in a Lyapunov based approach is actually no better than a corresponding local minima solution from conventional approaches.

The following theorem states the fact that the new learning law (11) suggested above with dead-zone technique is robust stable.

Theorem 1. *Under assumption* A1, *the weights are adjusted as follows*:

(a) *if* $\|\Delta_t\|^2 > (\overline{\eta}/\lambda_{\min}(Q_1))$ *then the updating law is given by* (11),
(b) *if* $\|\Delta_t\|^2 \leqslant (\overline{\eta}/\lambda_{\min}(Q_1))$ *then we stop the learning procedure* ($\dot{W}_t = \dot{V}_t = 0$)

then the identification error and weight matrices remain bounded, i.e.,

$$\|\Delta_t\| \in L_\infty, \quad W_t \in L_\infty, \quad V_t \in L_\infty \tag{12}$$

and for any $T > 0$ *the average of identification error fulfills the following tracking performance*:

$$\frac{1}{T}\int_0^T \|\Delta_t\|^2_{Q_1}\, \mathrm{d}t \leqslant \kappa \overline{\eta} + \frac{\Delta_0^{\mathrm{T}} P \Delta_0}{T}, \tag{13}$$

where κ *is condition number of* Q_1 *defined as* $\kappa = \lambda_{\max}(Q_1)/\lambda_{\min}(Q_1)$.

Proof. From (2) and (4) the error equation can be expressed as

$$\dot{\Delta}_t = A\Delta_t + \widetilde{W}_t \phi + W^* \widetilde{\phi}_t + W^* \widetilde{\phi}'_t + \widetilde{f}_t, \tag{14}$$

where $\widetilde{W}_t = W_t - W^*$. Defining Lyapunov function candidate as

$$V_t = \Delta_t^{\mathrm{T}} P \Delta_t + \mathrm{tr}\left[\widetilde{W}_t^{\mathrm{T}} K_1^{-1} \widetilde{W}_t\right] + \mathrm{tr}\left[\widetilde{V}_t K_2^{-1} \widetilde{V}_t^{\mathrm{T}}\right], \tag{15}$$

where P is a solution of (9). If the updating gains are defined as Π_1 and Π_2, here $\Pi_1 = K_1 P$, $\Pi_2 = K_2 P$, then K_1 and K_2 can be selected as

$$K_1 = P^{-1}\Pi_1, \quad K_2 = P^{-1}\Pi_2. \tag{16}$$

Calculating the derivative of (15), we obtain

$$\dot{V}_t = 2\Delta_t^{\mathrm{T}} P \dot{\Delta}_t + 2\mathrm{tr}\left[\dot{\widetilde{W}}_t^{\mathrm{T}} K_1^{-1} \widetilde{W}_t\right] + 2\mathrm{tr}\left[\dot{\widetilde{V}}_t K_2^{-1} \widetilde{V}_t^{\mathrm{T}}\right]. \tag{17}$$

Substitute (14) into (17), we get

$$\begin{aligned} 2\Delta_t^{\mathrm{T}} P \dot{\Delta}_t = {} & 2\Delta_t^{\mathrm{T}} P A \Delta_t + 2\Delta_t^{\mathrm{T}} P W^* \widetilde{\phi}_t + 2\Delta_t^{\mathrm{T}} P \widetilde{W}_t \phi \\ & + 2\Delta_t^{\mathrm{T}} P W^* \widetilde{\phi}'_t + 2\Delta_t^{\mathrm{T}} P \widetilde{f}_t. \end{aligned} \tag{18}$$

In view of the matrix inequality

$$X^{\mathrm{T}} Y + (X^{\mathrm{T}} Y)^{\mathrm{T}} \leqslant X^{\mathrm{T}} \Lambda^{-1} X + Y^{\mathrm{T}} \Lambda Y, \tag{19}$$

which is valid for any $X, Y \in \Re^{n\times k}$ and for any positive definite matrix $0<\Lambda=\Lambda^{\mathrm{T}}\in\Re^{n\times n}$, and using (7) $\Delta_t^{\mathrm{T}}PW^*\widetilde{\phi}_t$ in (18) can be concluded as

$$2\Delta_t^{\mathrm{T}}PW^*\widetilde{\phi}_t \leqslant \Delta_t^{\mathrm{T}}PW^*\Lambda_1^{-1}W^{*\mathrm{T}}P\Delta_t+\widetilde{\phi}_t^{\mathrm{T}}\Lambda_1\widetilde{\phi}_t \leqslant \Delta_t^{\mathrm{T}}(P\overline{W}P+\Lambda_\phi)\Delta_t. \tag{20}$$

Using (8), the last term in (18) can be rewritten as

$$2\Delta_t^{\mathrm{T}}PW^*\widetilde{\phi}_t' = 2\Delta_t^{\mathrm{T}}PW_tD_\phi\widetilde{V}_t\widehat{x}_t+2\Delta_t^{\mathrm{T}}P\widetilde{W}_tD_\phi\widetilde{V}_t\widehat{x}_t + 2\Delta_t^{\mathrm{T}}PW^*v_\phi. \tag{21}$$

The term $2\Delta_t^{\mathrm{T}}PW^*v_\phi$ in (21) can be formulated as

$$2\Delta_t^{\mathrm{T}}PW^*v_\phi \leqslant \Delta_t^{\mathrm{T}}PW^{*\mathrm{T}}\Lambda_1^{-1}W^*P\Delta_t+v_\phi^{\mathrm{T}}\Lambda_1v_\phi \leqslant \Delta_t^{\mathrm{T}}P\overline{W}P\Delta_t+l_1\|\widetilde{V}_t\widehat{x}_t\|^2_{\Lambda_1}. \tag{22}$$

From **A1**, $2\Delta_t^{\mathrm{T}}P\widetilde{f}_t$ can be estimated as

$$2\Delta_t^{\mathrm{T}}P\widetilde{f}_t \leqslant \Delta_t^{\mathrm{T}}P\Lambda_1^{-1}P\Delta_t+\widetilde{f}_t^{\mathrm{T}}\Lambda_1\widetilde{f}_t \leqslant \Delta_t^{\mathrm{T}}P\Lambda_1^{-1}P\Delta_t+\overline{\eta}. \tag{23}$$

Add and subtract $\Delta_t^{\mathrm{T}}Q\Delta_t$, (18), (20), (22) and (23) can be rewritten as

$$\dot{V}_t \leqslant \Delta_t^{\mathrm{T}}L\Delta_t+L_w+L_v-\Delta_t^{\mathrm{T}}Q_1\Delta_t+\overline{\eta}, \tag{24}$$

where

$$L=PA+A^{\mathrm{T}}P+P(2\overline{W}+\Lambda_1^{-1})P+(Q_1+\Lambda_\phi),$$

$$L_w=2\mathrm{tr}\left\{\dot{\widetilde{W}}_t^{\mathrm{T}}K_1^{-1}\widetilde{W}_t\right\}+2\Delta_t^{\mathrm{T}}P\widetilde{W}_t\left(\phi+D_\phi\widetilde{V}_t\widehat{x}_t\right),$$

$$L_v=2\mathrm{tr}\left\{\dot{\widetilde{V}}_t^{\mathrm{T}}K_2^{-1}\widetilde{V}_t\right\}+2\Delta_t^{\mathrm{T}}PW_tD_\phi\widetilde{V}_t\widehat{x}_t+l_1\widehat{x}_t^{\mathrm{T}}\widetilde{V}_t^{\mathrm{T}}\Lambda_1\widetilde{V}_t\widehat{x}_t.$$

Using (9), $L=0$. So

$$\dot{V}_t \leqslant L_w+L_v-\Delta_t^{\mathrm{T}}Q_1\Delta_t+\overline{\eta}.$$

(I) if $\|\Delta_t\|^2>\lambda_{\min}^{-1}(Q_1)\overline{\eta}$, using the updating law as (11) we can conclude that:

$$\dot{V}_t \leqslant -\Delta_t^{\mathrm{T}}Q_1\Delta_t+\overline{\eta} \leqslant -\lambda_{\min}(Q_1)\|\Delta_t\|^2+\overline{\eta}<0. \tag{25}$$

V_t is bounded. Integrating (25) from 0 up to T yields

$$V_T-V_0 \leqslant -\int_0^T \Delta_t^{\mathrm{T}}Q_1\Delta_t\,\mathrm{d}t+\overline{\eta}T.$$

Because $\kappa\geqslant 1$ and $V_T\geqslant 0$, we have

$$\int_0^T \Delta_t^{\mathrm{T}}Q_1\Delta_t\,\mathrm{d}t \leqslant V_0+\overline{\eta}T \leqslant V_0+\kappa\overline{\eta}T, \tag{26}$$

where κ is condition number of Q_1.

(II) If $\|\Delta_t\|^2\leqslant\lambda_{\min}^{-1}(Q_1)\overline{\eta}$, the weights become constants, V_t remains bounded. Since $V_0\geqslant 0$

$$\int_0^T \Delta_t^{\mathrm{T}}Q_1\Delta_t\,\mathrm{d}t \leqslant \int_0^T \lambda_{\max}(Q_1)\|\Delta_t\|^2\,\mathrm{d}t \leqslant \frac{\lambda_{\max}(Q_1)}{\lambda_{\min}(Q_1)}\overline{\eta}T \leqslant V_0+\kappa\overline{\eta}T. \tag{27}$$

From (I) and (II), V_t is bounded, (12) is realized. From (26) and (27), (13) is obtained. The theorem is proved. □

A summary of the system identification via single neural networks is as follows:

- Construct the dynamic neural networks as in (2). We choose a stable matrix A and sigmoid function ϕ, and give the initial conditions for the neural networks $\widehat{x}_0$ and the weights W_0 and V_0.
- Determine the constants in the learning law (11). We choose the updating gains Π_1 and Π_2, here $\Pi_1=K_1P$, $\Pi_2=K_2P$. So, we do not need to solve the Riccati equation (9) to determine the learning gain Π_1 and Π_2. We also should give l_1, Λ_1, the upper bound of uncertainties $\overline{\eta}$ and positive definite matrix Q_1. Use off-line algorithm [29] to determine V^*.
- On-line identification. We can get the system state x_t from the plant and the neural networks state $\widehat{x}_t$ from (2). Using the identification error Δ_t, we can update the weights of neural networks on-line.

Remark 3. If we use other types of neural networks, for example the recurrent high-order neural networks [17]

$$y=WS(x).$$

Because the weights are linear with the data, gradient-like algorithm with dead-zone can also grant the identification error bounded. So these neural networks can also be applied for adaptive control with multiple neural networks which are discussed in this paper.

Remark 4. Similar to the other dynamic neural networks, the weights of the multi-layer dynamic neural networks cannot converge to the constants. So this dynamic neural network can only work on-line, it is maybe not effective for prediction, but it is useful for identification-based control (or adaptive control) [14,17]. It can be imaged that the weights of the dynamic neural networks are changeable with the different input sets. It appears that learning laws (11) require perfect knowledge of the process states. But in practice when process states are not known, and only the output y_t and input U_t are measurable

$$\dot{x}_t=f(x_t,U_t),\quad y_t=H(x_t). \tag{28}$$

Algorithm (11) cannot be used directly. A model-free observer, for example high-gain observer [20], has to be applied to get the full-state from the output measurement.

3. System identification with multiple dynamic neural networks

From (4) we know a neural network cannot match a nonlinear system exactly, the modeling error $\widetilde{f}_t$ depends on the structure of the network. For some nonlinear processes when the operation conditions are changed or its operation environment is complex, *one* model for these processes is not enough to follow the whole plant, *multiple* models can give a better identification accuracy. Although the single neural network (2) can identify any nonlinear process

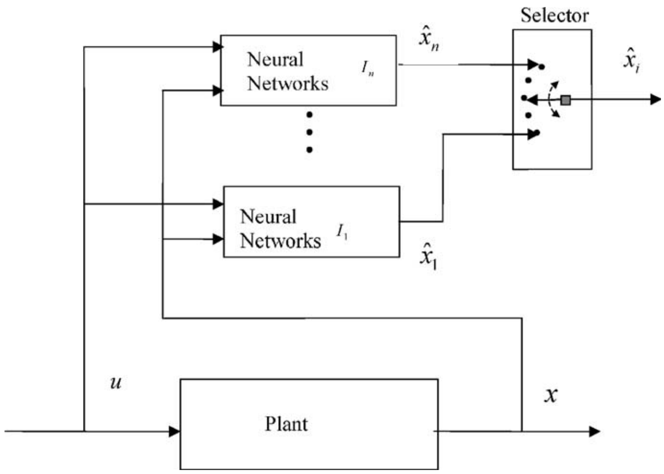


Fig. 2. The structure of multiple dynamic neural networks.

(black-box), the identification error is big if the network structure is not good. In general we cannot find the optimal network structure, but we can use several possible networks and select the best one by a proper switching algorithm.

The structure of multiple dynamic neural networks is shown in Fig. 2. Here $I_1, \ldots, I_n$ are neural identifiers, whose outputs are $\widehat{x}_1, \ldots, \widehat{x}_n$. We use a selector to choose a best identifier I_i such that the identification error $\widehat{x}_i - x$ is minimized. Let S be a closed and bounded set that represent a parameter space of finite dimension. Assume that the plant parameter vector p and the model parameter vector $\hat{p}$ (the weights of neural networks) belong to S. We assume that the plant and all of models can be parametrized in the same way as in (4). Each parameter vector $\hat{p}_i$ is associated to one neuro identifier I_i. The multiple dynamic neural networks are presented

$$\dot{\widehat{x}}_\sigma = A_\sigma \widehat{x}_\sigma + W_t^\sigma \phi_\sigma (V_t^\sigma \widehat{x}_\sigma) + U_t, \tag{29}$$

where $\sigma = \{1, 2, \ldots, N\}$. The objective of multiple neural networks is to improve a performance of the identification using a finite number of models $\{I_i\}_{i=1}^N$. N is selected according to the plant parameter vector p.

In each instant the identification error Δ_i ($\Delta_i = \widehat{x}_i - x$, $i = 1, \ldots, N$) which corresponds to each neuro identifier I_i is calculated. The multiple neural networks identification is to select suitable σ from all possible switching input set Ω such that the performance indexes is minimized. We can define identification error performance index J_i for each neuro identifier as

$$J_i(t) = \kappa_1 \Delta_i^2(t) + \kappa_2 \int_0^t \Delta_i^2(\tau)\,\mathrm{d}\tau, \tag{30}$$

where $\kappa_1 > 0$ and $\kappa_2 \geqslant 0$ are design parameters. κ_1 and κ_2 define the weights given to the instant and long term errors, respectively. This performance index is similar to [13].

From the view point of system identification, estimation error is caused by structure uncertainty and parameter uncertainty. Structure uncertainty is solved by switching in multiple neural networks. Parameter uncertainty is solved by weights updating of each neural network. When the weight updating (parameter) cannot make the error performance index smaller, we should change another neural network (structure). When we have a new neural network, we should first use parameter updating method to minimize the identification error until the parameters (weights) converge. So we should not change the neural network model while the weights are updated.

A switching scheme is needed to monitor the performance index of the multiple neural networks. To prevent an arbitrarily fast switching, a hysteresis switching algorithm [11] is needed. In this paper, we propose a new hysteresis switching algorithm for multiple neural networks, i.e., only when the weights of neural networks are almost convergence, we start to use the hysteresis switching algorithm. Let us define a selector function as

$$\rho(J) = \min\{i | J_i \leqslant J_j,\ i, j \in \Omega\},$$

where $J = [J_1, J_2, \ldots, J_N]$. And we define a weights change function

$$\omega_i(t) = \frac{1}{2}\left|\mathrm{tr}(W_t^{i\mathrm{T}} W_t^i) - \frac{1}{t}\int_0^t \mathrm{tr}(W_\tau^{i\mathrm{T}} W_\tau^i)\,\mathrm{d}\tau\right| + \frac{1}{2}\left|\mathrm{tr}(V_t^{i\mathrm{T}} V_t^i) - \frac{1}{t}\int_0^t \mathrm{tr}(V_\tau^{i\mathrm{T}} V_\tau^i)\,\mathrm{d}\tau\right|.$$

The new hysteresis switching algorithm is

$$\pi(i, J) = \begin{cases} i & \text{if } J_i(t) \leqslant J_{\rho(J)}(t) + h \\ & \quad \text{or } \omega_i(t) > l \text{ (no switch)}, \\ \rho(J) & \text{if } J_i(t) > J_{\rho(J)}(t) + h \\ & \quad \text{and } \omega_{\rho(J)}(t) \leqslant l \text{ (switch)}, \end{cases} \tag{31}$$

where $h > 0$ is hysteresis constant, $l > 0$ is the threshold of weights change. We choose l such that switching (31) can work after the parameters (weights) of the neural networks do not give the major influence on neuro identification. The switching input function σ in (29) is given as following:

$$\sigma(t) = \pi[\sigma^-(t), J], \quad \sigma^-(0) = i_0, \tag{32}$$

where i_0 is the initial condition in Ω and $\sigma^-(t)$ is the limit of $\sigma(\tau)$ from below, i.e., $\sigma^-(t) = \lim_{\tau \to t} \sigma(\tau)$. The switch process is that at $t = 0^-$ we start from a initial state i_0, σ will remain in this state until $t_1 \geqslant 0$ when $J_i \geqslant J_j + h$ and $\omega_i(t) \leqslant l$, at time t_1, σ switches to state j. Let us define $\mathscr{S}$ as the class of all piecewise-constant function $s : (0, \infty) \to \Omega$.

Lemma 1. (*a*) *For any T_s, there exists at least one integer $\sigma_0 \in \Omega$ such that for each $s \in \mathscr{S}$, the performance index $J_{\sigma_0}(t)$ in* (30) *is bounded on* $[0, T_s)$.

(*b*) *For each $s \in \mathscr{S}$ and each $j \in \Omega$, performance index $J_j(t)$ in* (30) *has a limit* (*may be infinity*) *as $t \to T_s$.*[1]

[1]Let f be a function defined on an open interval containing c (except possibly at c.) The statement "infinite limit" $\lim_{x \to c} f(x) = \infty$ means that for each $M > 0$ there exists a δ such that $f(x) > M$, whenever $0 < |x - c| < \delta$.

Proof. (29) can be expressed in normal nonlinear system form

$$\dot{\widehat{x}}_\sigma = A_\sigma \widehat{x}_\sigma + W_t^\sigma \phi_\sigma(V_t^\sigma \widehat{x}_\sigma) + U_t = f_\sigma(\widehat{x}_\sigma, U_t).$$

Since $\phi_\sigma(x)$ in (29) is sigmoid function, $f_\sigma(\widehat{x}_\sigma, u_t)$ is locally Lipschitz. The hysteresis constant h and the threshold on $\omega_i(t)$ assure that there exists a maximal interval $(0, t_1)$ on which σ is constant. If a switch occur at t_1, the hysteresis constant h and threshold l may also assure that maximal interval (t_1, t_2) on which σ still constant. So we can conclude that there must be a interval $[0, T_s)$ in which there is a unique pair $(\widehat{x}, \sigma)$ with $\widehat{x}$ continuous and σ piecewise constant, which satisfies (29)–(32). Moreover, on each strictly proper subinterval $[0, t) \subset [0, T_s)$, σ can switch at most a finite number of times.

On $[0, T_s)$ there are finite neuro identifiers participate the identification. Theorem 1 tells us the identification errors of neuro identifiers are bounded, so in finite time t there exists at least one $\sigma_0 \in \Omega$ such that (30) is bounded.

Since on $[0, T_s)$ there exist finite number of neuro identifiers, (30) has a limit as $t \to T_s$, (b) is established. □

Next theorem give the behavior of switch system (29)–(32).

Theorem 2. *If we use the multiple dynamic neural networks as in* (29) *and the hysteresis switching algorithm as in* (31), *then there exists a time T^* after which*

(a) *for all $T > T^*$, $J_{\sigma(T^*)}$ is bounded on* $[0, T)$;
(b) *the switching input function $\sigma(t)$ is constant.*

Proof. Lemma 1states that on $[0, T^*)$ there exist a bounded $J_{\sigma_0}(t)$. This theorem discusses on $(0^*, T)$, $T > T^*$, there exist a bounded $J_{\sigma(T^*)}$. Let us define $[0, T)$ is the maximal interval in which (29)–(32) have unique solution $(\widehat{x}, \sigma)$ from a fixed initial state $(\widehat{x}_0, i_0)$. From Lemma 1 (a) we know on $[0, T)$, there exist at least σ_0 such that J_i is bounded. On the other hand, there may exist another σ_1 such that on $[0, T)$ there are infinite neuro identifiers. In this case J_i is unbounded. On $[0, T)$ we define two disjoint subsets Ω_b and Ω_u, where J_i is bounded in Ω_b and unbounded in Ω_u. By Lemma 1(a), we know that the set Ω_b is nonempty, i.e.,

$$J_i(t) \leqslant c < \infty, \quad i \in \Omega_\mathrm{b}, \ t \in [0, T), \ c > 0. \tag{33}$$

When J_i is in Ω_u, J_i is unbounded. We can express it as

$$J_i(t) > c + h, \quad i \in \Omega_\mathrm{u}, \ t \in [0, T). \tag{34}$$

From Lemma 1(b) we know when $t \to T$, $J_i(t)$ has a limit. So there exist t_1 which is near to T such that

$$|J_i(t) - J_i(t_1)| < h/2, \quad i \in \Omega_\mathrm{b}, \ t \in [t_1, T). \tag{35}$$

(35) tells us that for each $i \in \Omega_\mathrm{b}$, there exists t_1 such that $J_i(t)$ has a small variation on $t \in [t_1, T)$, which do not grow more than half of h. Since l is selected such that the switch (31) cannot be effected by weights after the weights converge. So we only consider the case $\omega_i(t) \leqslant l$.

First let us verify the boundness of $J_{\sigma(T^*)}$.

- Suppose that after t_1 we have two neuro identifiers i and j,

  $$J_j(t) \leqslant J_i(t) + h, \quad t \in [t_1, T).$$

  From (31) we know σ does not switch after t_1, so $J_i(t)$ satisfies (33). Let $T^* = t_1$ and $\sigma^* = \sigma(T^*)$, using (33) we have

  $$J_{\sigma(T^*)} \leqslant c + h, \quad t \in [T^*, T).$$

  So $J_{\sigma(T^*)}$ is bounded on $[0, T)$.
- Suppose at time $t_2 \in [t_1, T)$,

  $$J_j(t_2) > J_i(t_2) + h,$$

  σ switches in t_2 then remains constant in $t \in [t_2, T)$. Let $T^* = t_2$ and $\sigma^* = \sigma(T^*)$.

  $$J_j(t_2) > J_{\sigma^*}(t_2) + h > J_{\sigma^*}(t_2), \quad j \in \Omega. \tag{36}$$

  Since Ω_b is nonempty, there exits one bounded J before t_2, such that

  $$c > J_j(t_2).$$

  Therefore by (36), J_{σ^*} is bounded in $[0, T)$ and $\sigma^* \in \Omega_\mathrm{b}$.

Next we are going to review the infinite switching case.

- If $j \in \Omega_\mathrm{b}$, for any $t \in [t_2, T)$, where t_2 is near to T, we have

  $$J_{\sigma^*}(t) - J_j(t) = [J_{\sigma^*}(t) - J_{\sigma^*}(t_2)] + [J_{\sigma^*}(t_2) - J_j(t_2)] + [J_j(t_2) - J_j(t)].$$

  Using (36) we have $J_{\sigma^*}(t_2) - J_j(t_2) \leqslant 0$, so

  $$\begin{aligned} &J_{\sigma^*}(t) - J_j(t) \\ &\quad \leqslant [J_{\sigma^*}(t) - J_{\sigma^*}(t_2)] + [J_j(t_2) - J_j(t)] \\ &\quad \leqslant |J_{\sigma^*}(t) - J_{\sigma^*}(t_2)| + |J_j(t_2) - J_j(t)|. \end{aligned} \tag{37}$$

  Following from (35), (37) becomes

  $$J_{\sigma^*}(t) - J_j(t) \leqslant h, \quad t \in [t_2, T), \ j \in \Omega_\mathrm{b}.$$

- If $j \in \Omega_\mathrm{u}$, from above we know $J_{\sigma^*}(t)$ is bounded, (33) and (34) imply that

  $$J_{\sigma^*}(t) - J_j(t) \leqslant c - (c + h) \leqslant h, \quad t \in [t_2, T), \ j \in \Omega_\mathrm{u}.$$

  So for all $j \in \Omega$,

  $$J_{\sigma^*}(t) \leqslant J_j(t) + h, \quad t \in [t_2, T). \tag{38}$$

  (31) and (38) imply that no more switches occur no $[t_2, T)$, so σ is constant on $[T^*, T)$, that is (b). □

Remark 5. The proof of this theorem is different from the hysteresis switching lemma in [11]. First we use multiple neural networks as identification models. So we do not need the "open-loop assumptions". These conditions are proved by Lemma 1 of this paper. Second the hysteresis switching algorithm is changed according to the special property of neuro identifiers, so the proof also considers the weight conditions.

4. Indirect adaptive control using multiple neural networks

In this section we give a general multiple neural controller which consists of four subsystems (see Fig. 3).

1. Multiple neuro estimator. This subsystem can give a best estimation such that the output of the multiple identifier can follow the plant with a minimum error.
2. Multiple adaptive controller. This subsystem is an estimator-based controller, the controller produces a suitable control signed for the plant.
3. Performance index generator. This subsystem collects identification information, such as identification error and model number, and generates a performance index signal.
4. Switching Logic. This subsystem uses the performance index signal and inner logics, and gives a switching command to control model.

From last section we know the nonlinear system can be modeled by one dynamic neural network as

$$\dot{x}_t = Ax_t + W_t\phi(V_t\widehat{x}_t) + W_t[\phi(V_tx_t) - \phi(V_t\widehat{x}_t)] + U_t + \widetilde{f}_t. \tag{39}$$

Since we use the updated laws (11) for W_t and V_t, by Theorem 1 we know W_t and V_t are bounded. Using the assumption A1, (39) can be rewritten as

$$\dot{x}_t = Ax_t + W_t\phi(V_t\widehat{x}_t) + U_t + d_t, \tag{40}$$

where $d_t = \widetilde{f}_t + W_t[\phi(V_tx_t) - \phi(V_t\widehat{x}_t)]$ is bounded, $\|d_t\| \leqslant \overline{d}$.

The object of the indirect adaptive control discussed in this paper is to track a optimal trajectory $x_t^* \in \Re^r$ which is assumed to be smooth enough. This trajectory is regarded as a solution of a nonlinear reference model:

$$\dot{x}_t^* = \varphi(x_t^*, t)$$

with a fixed initial condition. Let us define the state tracking error as

$$e_t = x_t - x_t^*.$$

The error equation is

$$\dot{e}_t = Ax_t + W_t\phi(V_t\widehat{x}_t) + U_t + d_t - \varphi(x_t^*, t).$$

Now let us assume the control action U_t is made up of two parts:

$$U_t = u_{1,t} + u_{2,t}, \tag{41}$$

where $u_{1,t} \in \Re^n$ is direct linearization part and $u_{2,t} \in \Re^n$ is a compensation of unmodeled dynamic d_t. As $\varphi(x_t^*, t)$, x_t^* and $W_t\phi(V_t\widehat{x}_t)$ are available, we can select $u_{1,t}$ as

$$u_{1,t} = \varphi(x_t^*, t) - Ax_t^* - W_t\phi(V_t\widehat{x}_t). \tag{42}$$

So the tracking error dynamic is

$$\dot{e}_t = Ae_t + u_{2,t} + d_t. \tag{43}$$

Theorem 3. *The tracking error is bounded when the neural control is* (41) *and* (42), *the compensator $u_{2,t}$ and disturbance d_t are bounded.*

Proof. Let us define Lyapunov-like function as

$$V_t = e_t^{\mathrm{T}} P_1 e_t, \tag{44}$$

where P_1 is a positive definite matrix. Using (43), the derivative of (44) is

$$\dot{V}_t = e_t^{\mathrm{T}}(A^{\mathrm{T}}P_1 + P_1A)e_t + 2e_t^{\mathrm{T}}P_1u_{2,t} + 2e_t^{\mathrm{T}}P_1d_t, \tag{45}$$

$2e_t^{\mathrm{T}}P_1u_{2,t}$ and $2e_t^{\mathrm{T}}P_1d_t$ can be estimated as

$$2e_t^{\mathrm{T}}P_1u_{2,t} \leqslant e_t^{\mathrm{T}}P_1\Lambda_{\mathrm{u}}^{-1}P_1e_t + u_{2,t}^{\mathrm{T}}\Lambda_{\mathrm{u}}u_{2,t},$$

$$2e_t^{\mathrm{T}}P_1d_t \leqslant e_t^{\mathrm{T}}P_1\Lambda_p^{-1}P_1e_t + d_t^{\mathrm{T}}\Lambda_pd_t, \tag{46}$$

where Λ_{u} and Λ_p are positive definite matrices. So

$$\begin{aligned}\dot{V}_t \leqslant{} & e_t^{\mathrm{T}}(A^{\mathrm{T}}P_1 + P_1A + P_1(\Lambda_{\mathrm{u}}^{-1} + \Lambda_p^{-1})P_1 + Q_c)e_t \\ & + 2e_t^{\mathrm{T}}P_1u_{2,t} + 2e_t^{\mathrm{T}}P_1d_t - e_t^{\mathrm{T}}Q_ce_t.\end{aligned} \tag{47}$$

Because A is stable, there exit $\Lambda_{\mathrm{u}}^{-1}$, Λ_p^{-1} and Q_c such that the matrix Riccati equation

$$A^{\mathrm{T}}P_1 + P_1A + P_1(\Lambda_{\mathrm{u}}^{-1} + \Lambda_p^{-1})P_1 + Q_c = 0 \tag{48}$$

has a solution $P_1 = P_1^{\mathrm{T}} > 0$. (47) can be represented as

$$\begin{aligned}\dot{V}_t & \leqslant -e_t^{\mathrm{T}}Q_ce_t + u_{2,t}^{\mathrm{T}}\Lambda_{\mathrm{u}}u_{2,t} + d_t^{\mathrm{T}}\Lambda_pd_t \\ & \leqslant -\alpha\|e_t\| + \beta_1\|u_{2,t}\| + \beta_2\|d_t\|,\end{aligned}$$

where $\alpha = [\lambda_{\min}(Q_c)]\|e_t\|$, $\beta_1 = \lambda_{\max}(\Lambda_{\mathrm{u}})\|u_{2,t}\|$, $\beta_2 = \lambda_{\max}(\Lambda_p)\|d_t\|$. We see that α, β_1 and β_2 are $\mathscr{K}_\infty$ functions, V_t is an ISS-Lyapunov function. Using Theorem 1 of [22], the dynamic of tracking error (43) is input to state stability, i.e., when the inputs $\|u_{2,t}\|$ and $\|d_t\|$ are bounded, the state e_t is bounded. $\square$

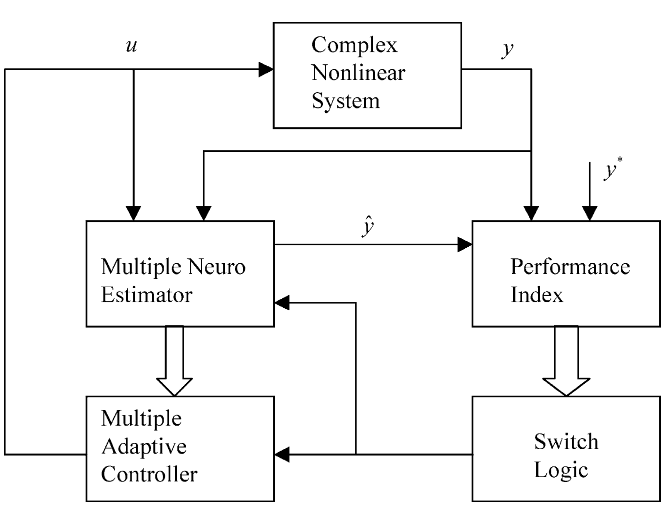


Fig. 3. General structure of multiple neuro adaptive control.

The design of $u_{2,t}$ is based on multiple model technique in order to cancel d_t effectively. To compensate d_t following two techniques can be applied.

Sliding mode compensation. If $\dot{x}_t$ is not available, the sliding mode technique can be applied. According to sliding mode technique, we can select $u_{2,t}$ as

$$u_{2,t} = -kP_1^{-1} sgn(e_t), \quad k > 0, \tag{49}$$

where k is positive constant,

$$sgn(\Delta_t^*) = [sgn(e_{1,t}), \ldots, sgn(e_{n,t})]^{\mathrm{T}} \in \Re^n.$$

Substitute (49) into (45)

$$\begin{aligned}\dot{V}_t &= -\|e_t\|^2 - 2k\|e_t\| + 2e_t^{\mathrm{T}} P_1 d_t \\ &\leqslant -\|e_t\|^2 - 2k\|e_t\| + 2\lambda_{\max}(P_1)\|e_t\|\|d_t\| \\ &= -\|e_t\|^2 - 2\|e_t\|(k - \lambda_{\max}(P_1)\|d_t\|).\end{aligned}$$

If we select

$$k > \lambda_{\max}(P_1)\overline{d}, \tag{50}$$

where $\overline{d}$ is define as (40), then $\dot{V}_t < 0$. So

$$\lim_{t\to\infty} e_t = 0.$$

Local optimal control: The control goal is to minimize

$$\begin{aligned}J_{op} &= \frac{1}{T}\int_0^T e_t^{\mathrm{T}} Q_2 e_t \, \mathrm{d}t + \frac{1}{T}\int_0^T u_{2,t}^{\mathrm{T}} R_2 u_{2,t} \, \mathrm{d}t \\ &= \|e_t\|_{Q_2}^2 + \|u_{2,t}\|_{R_2}^2,\end{aligned}$$

where Q_2 and R_2 are positive definite matrices. Because e_t and $u_{2,t}$ satisfy (43), we will use the following positive function Lyapunov function:

$$V_t = e_t^{\mathrm{T}} P_2 e_t, \quad P_2 = P_2^{\mathrm{T}} > 0.$$

By (43), whose time derivative is

$$\dot{V}_t = e_t^{\mathrm{T}}(A^{\mathrm{T}} P_2 + P_2 A)e_t + 2e_t^{\mathrm{T}} P_2 u_{2,t} + 2e_t^{\mathrm{T}} P_2 d_t. \tag{51}$$

Substituting (46) in (51), adding and subtracting the term $e_t^{\mathrm{T}} Q_2 e_t$ and $u_{2,t}^{\mathrm{T}} R_2 u_{2,t}$, we formulate

$$\begin{aligned}\dot{V}_t \leqslant\ & e_t^{\mathrm{T}} L e_t + 2e_t^{\mathrm{T}} P_2 u_{2,t} + u_{2,t}^{\mathrm{T}} R_2 u_{2,t} + d_t^{\mathrm{T}} \Lambda_2^{-1} d_t \\ & - e_t^{\mathrm{T}} Q_2 e_t - u_{2,t}^{\mathrm{T}} R_2 u_{2,t}^d.\end{aligned} \tag{52}$$

Because A is stable, there exit Λ_2 and Q_2 such that the matrix Riccati equation

$$L = A^{\mathrm{T}} P_2 + P_2 A + P_2 \Lambda_2 P_2 + Q_2 = 0 \tag{53}$$

has a solution $P_2 = P_2^{\mathrm{T}} > 0$. So (52) is

$$\dot{V}_t \leqslant \Psi(u_{2,t}) + d_t^{\mathrm{T}} \Lambda_2^{-1} d_t - \|e_t\|_{Q_2}^2 - \|u_{2,t}\|_{R_2}^2, \tag{54}$$

where $\Psi(u_{2,t}) = 2e_t^{\mathrm{T}} P_2 u_{2,t} + u_{2,t}^{\mathrm{T}} R_2 u_{2,t}$. Integrating each term from 0 to T, dividing each term by T, (54) becomes

$$\begin{aligned}&\|e_t\|_{Q_2}^2 + \|u_{2,t}\|_{R_2}^2 \\ &\quad\leqslant \frac{1}{T}\int_0^T d_t^{\mathrm{T}} \Lambda_2^{-1} d_t \, \mathrm{d}t + \frac{1}{T}\int_0^T \Psi(u_{2,t}) \, \mathrm{d}t + \frac{1}{T}(V_0 - V_T) \\ &\quad\leqslant \frac{1}{T}\int_0^T d_t^{\mathrm{T}} \Lambda_2^{-1} d_t \, \mathrm{d}t + \frac{1}{T} V_0 + \frac{1}{T}\int_0^T \Psi(u_{2,t}) \, \mathrm{d}t,\end{aligned}$$

$(1/T)V_0$ is a constant. $(1/T)\int_0^T d_t^{\mathrm{T}} \Lambda_2^{-1} d_t \, \mathrm{d}t$ is unknown disturbance. So, the control goal now is to minimize $\Psi(u_{2,t}) = 2e_t^{\mathrm{T}} P_2 u_{2,t} + u_{2,t}^{\mathrm{T}} R_2 u_{2,t}$. Without restriction, u^* is selected according to the linear squares optimal control law

$$u_{2,t}^* = -2R_2^{-1} P_2 e_t.$$

Because R_2 can be chosen a positive constant, we can select a gain such that $K = R_2^{-1} P_2$. We do not need to solve the Riccati equation (53). We name the $u_{2,t}^*(t)$ as the locally optimal control, because it is calculated based only on "local" information. We consider following three different architectures of the multi-model neuro control.

4.1. Multiple neuro controllers

The structure is shown in Fig. 4. Here "Neural Networks 1" is the unique estimator, the controller is "Direct linearization" with "Compensation 1" or "Compensation 2", here "Compensation 1" is sliding mode control, "Compensation 2" is local optimal control. The sliding mode control uses high-gain and variable structure technique, it can compensate the uncertainties in any accuracy, but it is sensitive to the noise and the chattering prevent its application. The local optimal control is smooth, but the control accuracy is not so well. If we use multi-model technique, we can take the advantages of the two controllers and overcome their disadvantages.

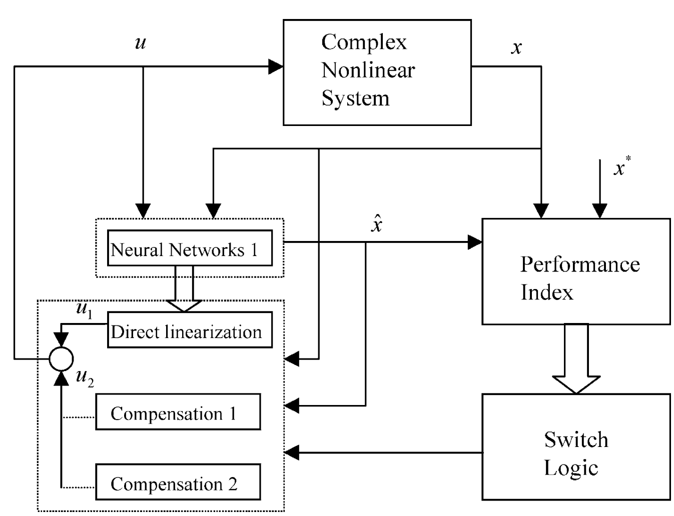


Fig. 4. One estimator and multiple controllers.

The switch logic is as follows:

- If the identification error $\|\Delta_t\| < S$ ($S > 0$ is the threshold constant for the switch), then $u_{2,t}$ is coming from the sliding mode control,

  $u_{2,t} = -kP_1^{-1}sgn(e_t), \quad k > 0.$

- If the identification error $\|\Delta_t\| \geqslant S$, then $u_{2,t}$ is come from the local optimal control

  $u_{2,t} = -2R_2^{-1}P_2e_t.$

4.2. Multiple neural estimators

The structure is shown in Fig. 5. Here "Neural Networks 1" and "Neural Networks N" are multiple estimators, the controller is "direct linearization" with "compensation". This control is also indirect adaptive control. First, we use multiple dynamic neural networks to estimate the nonlinear plant, the neuro identifier is

$$\dot{\hat{x}}_t = A_\sigma \hat{x}_t + W_t^\sigma \phi_\sigma(V_t^\sigma \hat{x}_t) + U_t, \tag{55}$$

where σ is the switching input. We select the performance index as in (30). The switching logic is an hysteresis (31). The nonlinear plant can be written as

$$\dot{x}_t = A_\sigma x_t + W_t^\sigma \phi_\sigma(V_t^\sigma \hat{x}_t) + W_t^\sigma[\phi_\sigma(V_t^\sigma x_t) - \phi_\sigma(V_t^\sigma \hat{x}_t)] + U_t + \widetilde{f}_t$$

or

$$\dot{x}_t = A_\sigma x_t + W_t^\sigma \phi_\sigma(V_t^\sigma \hat{x}_t) + U_t + d_{t,\sigma},$$

where $d_{t,\sigma} = \widetilde{f}_t + W_t^\sigma[\phi_\sigma(V_t^\sigma x_t) - \phi_\sigma(V_t^\sigma \hat{x}_t)]$. The control is

$$u_{1,t} = \varphi(x_t^*, t) - Ax_t^* - W_t\phi(V_t\hat{x}_t),$$

$$u_{2,t} = -2R_2^{-1}P_2e_t.$$

4.3. Multiple neural estimators and multiple neuro controllers

The structure is shown in Fig. 6. Here "Neural Networks 1" and "Neural Networks N" are multiple estimators, the controller is "Direct linearization" with "Compensation 1" or "Compensation 2". This structure is more flexible to design the multiple controller. It is a combination of the upper two approaches. The nonlinear plant can be written as

$$\dot{x}_t = A_\sigma x_t + W_t^\sigma \phi_\sigma(V_t^\sigma \hat{x}_t) + U_t + d_t,$$

where $d_t = \widetilde{f}_t + W_t^\sigma[\phi_\sigma(V_t^\sigma x_t) - \phi_\sigma(V_t^\sigma \hat{x}_t)]$. The direct linearization control is

$$u_{1,t} = \varphi(x_t^*, t) - A_\sigma x_t^* - W_t^\sigma \phi_\sigma(V_t^\sigma \hat{x}_t).$$

The design of $u_{2,t}$ is based on multiple model technique in order to cancel d_t effectively. The switch logic for the controller is the same as approach 4.1.

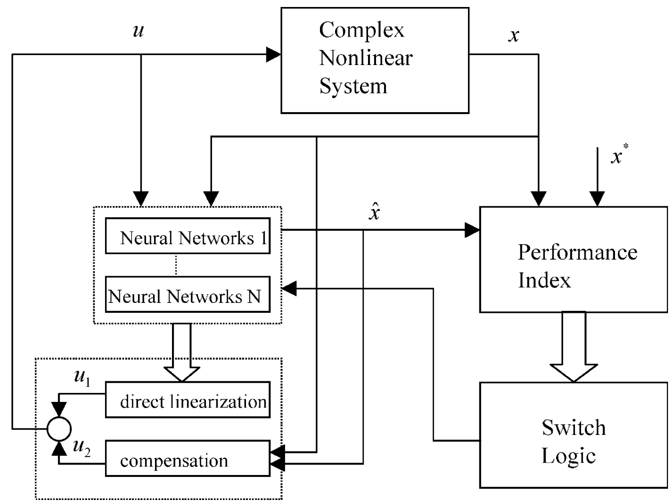


Fig. 5. Multiple estimators and one controller.

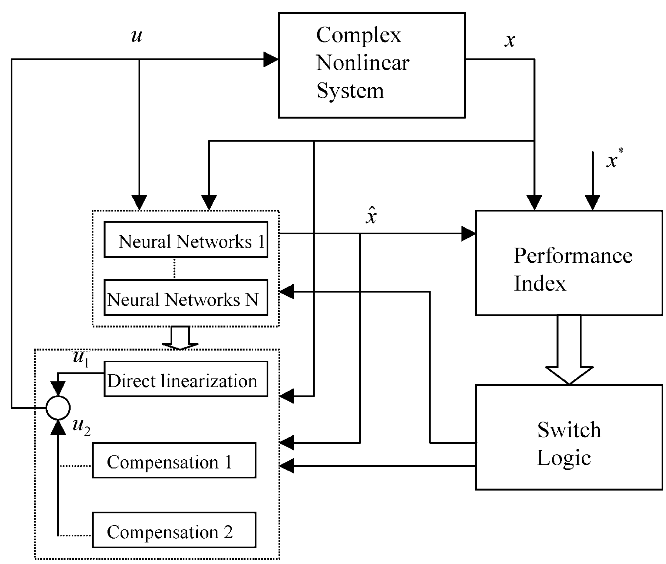


Fig. 6. Multiple estimators and multiple controllers.

Remark 6. The multiple neuro controllers are indirect adaptive control. First, we need to get a model from neuro identification, then based on this neuro model we can design adaptive controllers. The control is made up of two parts: direct linearization and uncertainties compensation.

- In approach 4.1, the direct linearization control is from one neuro identifier, the uncertainties compensation control uses multi-model technique. This approach can overcome the bad transient response caused by the compensator.
- In approach 4.2, the direct linearization control is from multiple neuro identifiers, the uncertainties compensation control uses classical control technique. This approach can overcome the bad transient response caused by the single neuro identifier

Table 1
Comparison of five multi-model controllers

	Multi-model adaptive control [19,12]	Nonlinear PID control [21]	Multi-model static neural networks [18]	TS fuzzy control [23]	Multi-model dynamic neural networks
Model	Linear	Linear/nonlinear	Nonlinear	Linear	Nonlinear
Adaptive	Yes	No	Yes	No	Yes
Algorithm	Simple	Very simple	Complex	Simple	Normal
Speed	Fast	Very fast	Very slow	Normal	Slow
Stability analysis	Yes	No	No	No	Yes
Transient performance	Good	Normal	Good	Normal	Good
Steady-state error	Small	Normal	Small	Normal	Small

- In approach 4.3, the direct linearization control is from multiple neuro identifiers, the uncertainties compensation control also uses multi-model technique. This approach can overcome the bad transient response caused by both of the compensator and the single neuro identifier. But the design process is complex.

Takagi–Sugeno (TS) fuzzy system [23] is proposed as an alternative to multiple model systems. Since the THEN parts are linear combinations of the input variables, where change from one output to the other is smooth rather than abrupt of multiple neural networks. TS fuzzy system can be viewed as a somewhat piecewise *linear* function, and multiple dynamic neural networks are combination of several *dynamic nonlinear* systems.

Table 1 presents the concrete comparison between our multiple dynamic neural networks with the other four multi-model controllers. We see that multi-model neural networks use nonlinear models, but multi-model adaptive control and TS fuzzy control use linear models. Multiple neural networks and multi-model adaptive control can select models and parameters automatically (adaptive). The algorithm for multi-model dynamic neural networks is more simple than multi-model static neural networks, because dynamic neural networks have less neurons. All of the multi-model controllers can improve transient and steady-state performances.

5. Simulations

Controlling pH neutralization systems are very important in the chemical industry. Usually we use the logarithmic behavior to present pH characteristic, this nonlinearity make the identification and control more difficult. A neutralization stirred tank is shown in Fig. 7, it has three influent streams: acid stream $q_1(HNO_3)$, buffer stream $q_2(NaHCO_3)$ and base stream $q_3(NaOH)$, and one effluent stream: q_4. The units for q is m^3/min.

Under the assumptions of perfect mixing, constant density, completely soluble ions and fast reactions, we

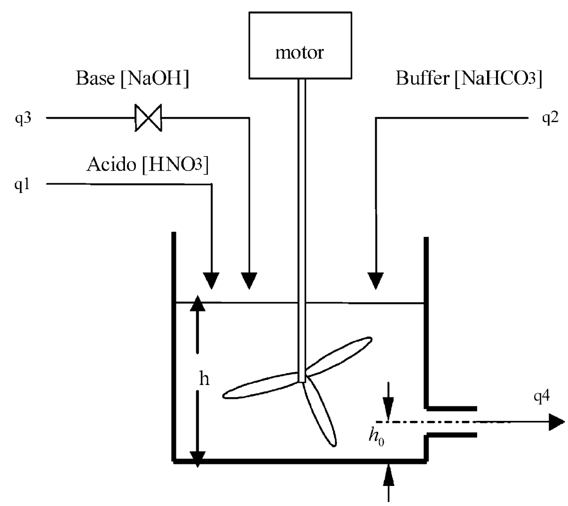


Fig. 7. Neutralization stirred tank.

can give the following chemical reaction models [5]:

$$H_2O \leftrightharpoons OH^- + H^+, \quad H_2CO_3 \leftrightharpoons HCO_3^- + H^+,$$
$$HCO_3^- \leftrightharpoons CO_3^= + H^+ \tag{56}$$

Based on neglecting the buffer stream (q_1) the only reaction considered is

$$H_2O \leftrightharpoons OH^- + H^+$$

The total mass balance is

$$D\dot{h} = q_1 + q_3 - q_4 - p\sqrt{h - h_0}, \tag{57}$$

where h and h_0 are tank level and tank outlet level, the unit is meter. D is tank area, the unit is m^2. p is valve constant. The component balance is

$$hD\dot{s} = q_1(s_1 - s) + q_3(s_3 - s), \tag{58}$$

where s is reaction invariant $s = [H^+] - [OH^-]$, s_1 and s_3 are chemical reaction invariants of HNO_3 and NaOH. The pH value can be calculated by

$$pH = -\log_{10}[H^+], \quad s = [H^+] - \frac{K_w}{[H^+]}, \tag{59}$$

where K_w is a positive constant $K_w = [H^+][OH^-]$. Since $[H^+] > 0$, we should use the positive solution of (59), i.e.,

$$\mathrm{pH} = -\log_{10}\left(\frac{s + \sqrt{s^2 + 4K_w}}{2}\right). \tag{60}$$

The neutralization stirred tank is modeled by Eqs. (57), (58) and (60). The inputs can be q_1, q_3 and q_4. The output is pH. We take q_3 as input and keep q_1 and q_4 as constants. One can see that pH neutralization is a strong nonlinear process. If we use the normal nonlinear form (1) to express the pH process, x_t is corresponded to pH, U_t is q_3.

There are many techniques of controlling pH neutralization systems, for example, time optimal controllers, linear adaptive control and nonlinear adaptive control. They did not consider "mass balance", only (58) was used. It is linear model. Even we consider the nonlinear transformation of the pH value, the parameters are linear with the data vector as in [5]. In this example the water level is changeable (see (57)), so (57)–(59) represent a strong nonlinear system. Normal adaptive control is difficult to be applied on it.

Let us use the same parameters as in [5] for the simulation. $D = 207$, $h_0 = 5$, $q_1 = 16.6$, $q_4 = 1$, $p = 8$, $s_1 = 0.003$, $s_3 = -0.003$, $K_w = 10^{-14}$. The initial conditions for h and s are $h(0) = 14$, $s(0) = 0.002$. We assume the ranges of the input and the output are $0 \leqslant s_3 \leqslant 40$, $0 \leqslant \mathrm{pH} \leqslant 14$. In this simulation we use Random Number block of Matlab/Simulink as input with sampling time 7 min. Before $t = 230$, the mean value of random number is 35, after $t = 230$ the mean value is 15.

Because neuro identification is in the sense of black-box, we do not need to know the nonlinear function $f(\mathrm{pH}, w_3)$. The study of [5] showed that the nonlinearity of the pH process are different in different ranges of pH values. So the unmodeled dynamics $\widetilde{f}_t$ in (4) are changed in different ranges of pH values. An object of multiple neuro identifiers is to use different neuro models to identify different ranges of pH values such that the unmodeled dynamic $\|\widetilde{f}_t\|$ is minimized in the whole range.

Because (57)–(60) represent a single-input single-output system, as in (29) let us construct three neuro identifiers ($\sigma = \{1, 2, 3\}$),

$$I_1: \quad \dot{\widehat{x}}_1 = A_1\widehat{x}_1 + W_t^1\phi_1(V_t^1\widehat{x}_1) + U_t,$$

$$I_2: \quad \dot{\widehat{x}}_2 = A_2\widehat{x}_2 + W_t^2\phi_2(V_t^2\widehat{x}_2) + U_t,$$

$$I_3: \quad \dot{\widehat{x}}_3 = A_3\widehat{x}_3 + W_t^3\phi_3(V_t^3\widehat{x}_3) + U_t, \tag{61}$$

where $A_1 = -1$, $A_2 = -0.3$, $A_3 = -15$, $W_t^1 \in R^1$, $W_t^2 \in R^3$, $W_t^3 \in R^5$, $V_0^1 = V_0^{1*} = 1$, $V_0^2 = V_0^{2*} = [1, 3, 1]^{\mathrm{T}}$, $V_0^2 = V_0^{2*} = [1, 3, 1, 0.1, 0.5]^{\mathrm{T}}$, $U_t = q_3$. The initial condition is $\widehat{x}_i(0) = 0$,

$$\phi_i = \frac{2}{1 + \exp(-2V_i\widehat{x}_i)} - \frac{1}{2}, \quad i = 1, 2, 3. \tag{62}$$

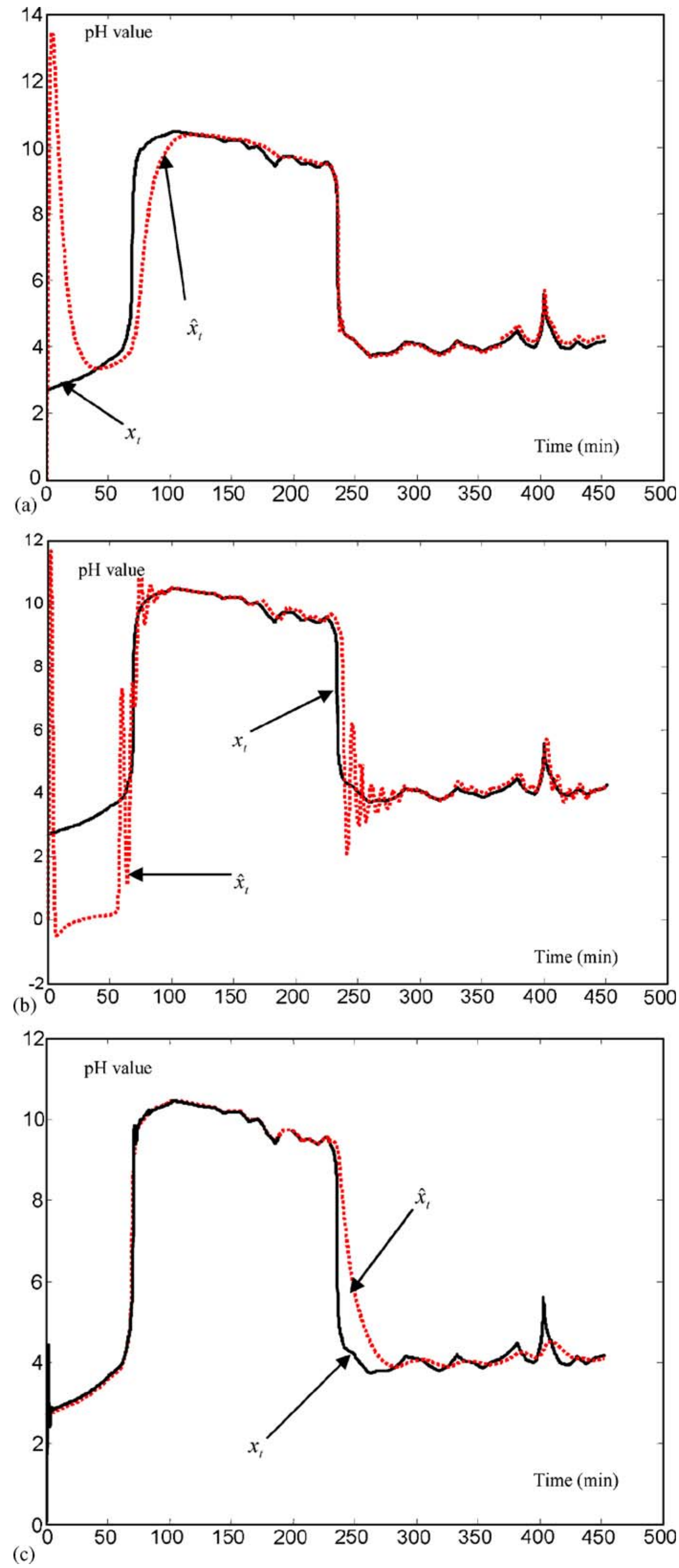


Fig. 8. Identification via I_1, identification via I_2, identification via I_3.

The update rules are as in (11) with $\Delta_i = \widehat{x}_i - x_t$, $K_{1,1} = 0.8$, $K_{1,2} = 0.5$, $K_{1,3} = 1.5$

First, we use the three neuro identifiers (61) to approximate the neutralization stirred tank separately. The identification results are shown in Fig. 8.

We can find that neuro identifier I_1 is good when pH value is low, I_3 is good when pH value is high, I_2 is good in whole range of pH value, but transient performance is bad. Then, we let these three identifiers run in parallel to identify the neutralization stirred tank, and use the hysteresis switching algorithm to select the best estimator at each time. The hysteresis switching algorithm (31) is

$$\pi(i,J)=\begin{cases} i & \text{if } J_i(t)\leqslant J_{\rho(J)}(t)+h \\ & \quad\text{or } \omega_i(t)>l \text{ (no switch)}, \\ \rho(J) & \text{if } J_i(t)>J_{\rho(J)}(t)+h \\ & \quad\text{and } \omega_{\rho(J)}(t)\leqslant l \text{ (switch)}, \end{cases}$$

$$J_1=\Delta_1^2,\quad J_2=\Delta_2^2,\quad J_3=\Delta_3^2,$$

$$\omega_i(t)=\frac{1}{2}\left|\operatorname{tr}(W_t^{iT}W_t^i)-\frac{1}{t}\int_0^t \operatorname{tr}(W_\tau^{iT}W_\tau^i)\,d\tau\right| + \frac{1}{2}\left|\operatorname{tr}(V_t^{iT}V_t^i)-\frac{1}{t}\int_0^t \operatorname{tr}(V_\tau^{iT}V_\tau^i)\,d\tau\right|, \tag{63}$$

where the hysteresis constant is selected $h=0.05$, the threshold of weights change is $l=1.5$. We start from I_3, i.e., $\sigma^-(0)=3$.

The identification result with multiple neural networks is shown in Fig. 9. We can see that at time 0–7, 70–80 and 230–240, the identification error is big and the hysteresis switch algorithm in (31) does not work, because in these periods the changes of the weights are bigger than the threshold l. Since multiple neural networks can switch identification models to match the large changes in the plant, the identification performance is better than a single neural network.

The structure of multiple neuro identifiers is very important for identification quality. The following items are helpful for the application:

1. In this example we only study two ranges of pH values, so we use three identifiers. For real application some prior knowledge can help us to decide how many neuro identifiers we should use. In general we should use a little more models than the number of operation ranges.
2. A_i should be selected as a stable matrix (or scalar), A_i will influence the dynamic response of the neural network. The bigger eigenvalues of A_i will make the neural network slower.
3. The constants of the sigmoid functions in (62) are chosen by simulations or experiments. From neural networks theory we know that the form of the sigmoid function does not influence the stability of the neural network, but for a special nonlinear process some functions have better approximate abilities.
4. The dimensions of the weight matrices W_t^i, V_t^i are structure problems for neural networks. It is well known that increasing the dimension of the weights can cause "overlap" problem and give more computation burden. What is the best dimension of the weights is still an open problem for neural network society. In this example we use three kinds of weights: small dimension I_1 (one-dimension), normal dimension I_2 (three-dimension) and large dimension I_3 (five-dimension). Since it is difficult to obtain the neural structure from prior knowledge, we can put several neuro identifiers in parallel and select the best one by the switching algorithm. This is another way to find a suitable structure for neuro identifiers.
5. For the three neuro identifiers we use the same update rules (11), almost all of single layer neural networks (feedforward or recurrent) use this kind of gradient descent algorithm. Because the structures are different, the same updating law will make the weights converge to different values. The learning gains $K_{1,\sigma}$ and $K_{2,\sigma}$ will influence the learning speed, very large gains can cause unstable learning, very small gains can make a slow learning process.

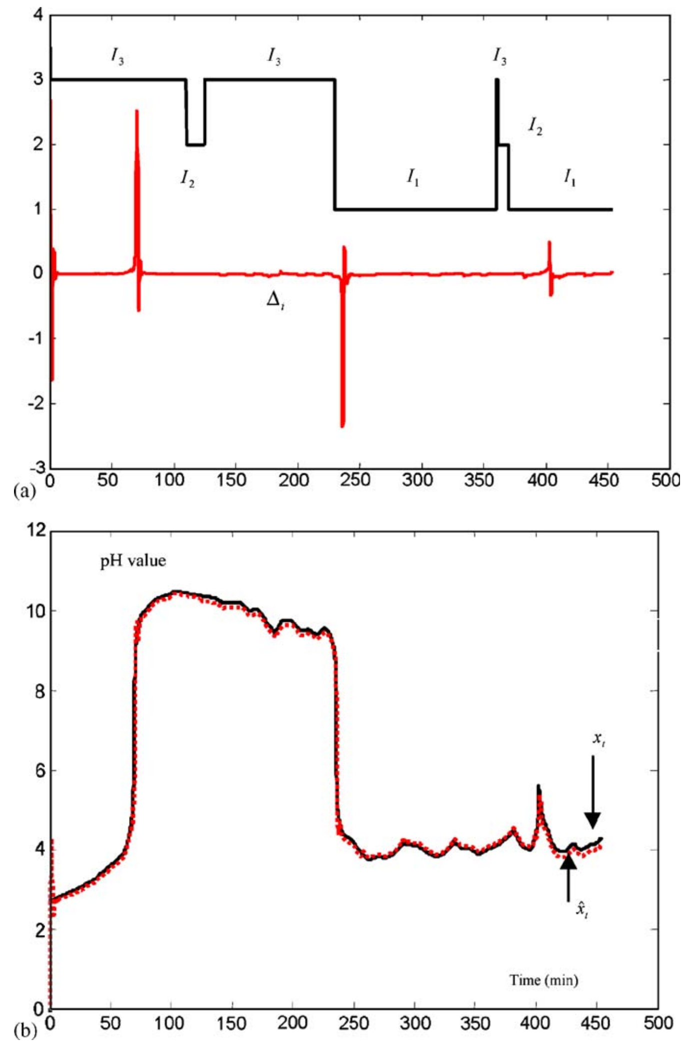


Fig. 9. Switching between identifiers. Identification via multiple neural networks.

Now we discuss adaptive control with neural networks. First, we use the same parameters as in [5], $D=207$, $h_0=5$, $q_1=16.6$, $q_4=1$, $p=8$, $s_1=0.003$, $s_3=-0.003$,

$K_w = 10^{-14}$. Based on the multiple neuro identifier, we use the indirect adaptive to force the pH value to 7. We compare two types of controllers: single controller with multi-identifier and multi-controller with multi-identifier. Single controller with multi-identifier is approach 4.2, the adaptive controller is

$$u_{1,t} = \varphi(x_t^*, t) - A_i x_t^* - W_t^i \phi_\phi(V_t^i \widehat{x}_t), \quad i = 1, 2, 3,$$

$$u_{2,t} = -2R_2^{-1} P_2 e_t,$$

where $\varphi(x_t^*, t) = x^* = 7$ (regulating to pH = 7), i is determined by (63), $e_t = x_t - x_t^*$, we choose $2R_2^{-1}P_2 = 100$. Multi-controller with multi-identifier is approach 4.3, the adaptive controller is

$$u_{1,t} = \varphi(x_t^*, t) - A_i x_t^* - W_t^i \phi_\phi(V_t^i \widehat{x}_t), \quad i = 1, 2, 3,$$

$$u_{2,t} = \begin{cases} u_{2,t} = -12 sgn(e_t) & \|e_t\| < 0.05, \\ u_{2,t} = -100 e_t & \|e_t\| \geqslant 0.05. \end{cases} \tag{64}$$

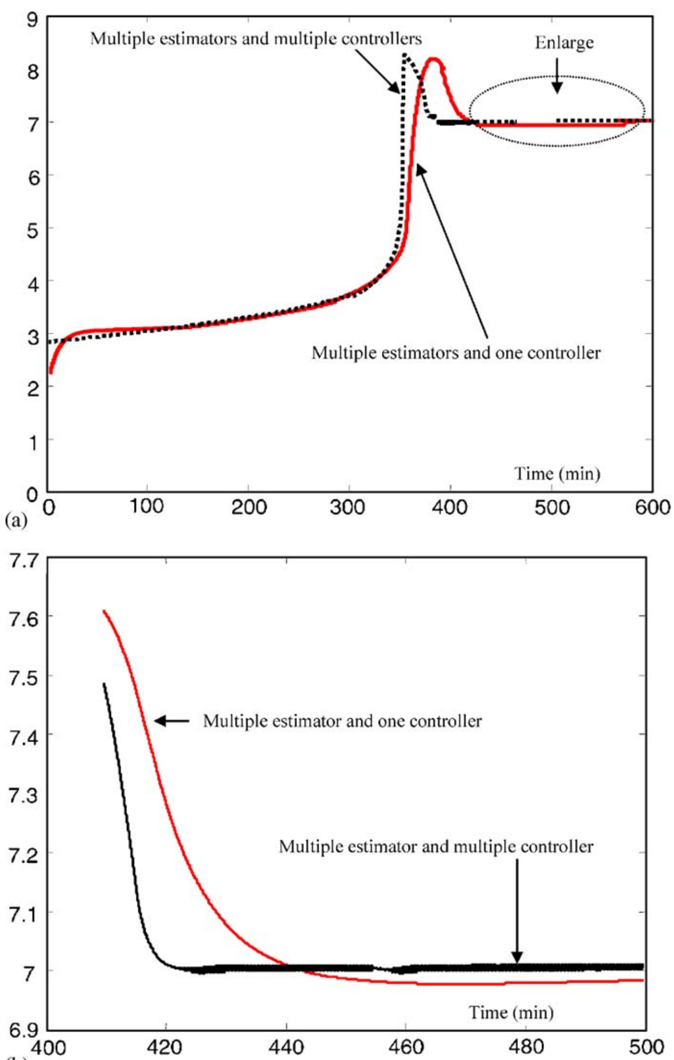


Fig. 10. Neural controllers. Enlarge the steady state.

The results of multi-model neural control are shown in Fig. 10.

We can see that multiple controllers have faster transient response. Then we compare our multi-controller with multi-identifier to nonlinear PID controller. We let the neutralization stirred tank be operated in two conditions, before $t = 800$ min the condition (parameters) is the same as in [5] (Condition A), after 800 min the parameter s_3 is changed as $s_3 = -0.01$ (Condition B). The nonlinear PID controller has the following form [21]

$$U_t = \begin{cases} k_{p1} e_t + k_{d1} \dot{e}_t + k_{i1} \int_0^1 e_t \, dt & \|e_t\| < 1, \\ k_{p2} e_t + k_{d2} \dot{e}_t + k_{i1} \int_0^1 e_t \, dt & \|e_t\| \geqslant 1. \end{cases}$$

After several simulations, we found the following PID parameters are optimal for Condition A:

$$k_{p1} = 2.5, \quad k_{d1} = 18, \quad k_{i1} = 0.01,$$

$$k_{p2} = 15, \quad k_{d2} = 10, \quad k_{i2} = 0.6.$$

Because PID controller has no adaptive mechanism, it does not work well for Condition B. On the other hand, multiple neural controllers can adjust their controllers (models) and decide the best controller for each condition. The comparison results are shown in Fig. 11.

We can see that for pH control, nonlinear PID is faster than multiple neural controllers in the case of fixed operation condition. When operation condition is changed, multiple neural controllers can adjust their control structure, they are better than fixed PID control. But our multiple controllers have bigger chattering because when steady-state error is small we switch to sliding mode control, see (64).

Remark 7. The method proposed in this paper is not convenient to apply on systems with fast dynamics (e.g., manipulators, motor drives). There are two cases to prevent its application on fast systems. First, the algorithm is in the form of continuous time, we use discrete-time approximate (several software such as Matlab-Simulink and LabView use this technique) to realize the neural control. Within each sampling time, we have to finish weights updating (11), models selection (30) and controller design (41). So for any computer, the algorithm proposed in this paper requires the computational time of the above three jobs is faster than sampling time. Second, neural adaptive control cannot arrive optimal results before the weights converge. So the convergence rate of the weight adaptation process should be faster than the speed of system state variation. Since weights updating law (11) is gradient algorithm, the convergence rate depends on gains K_1 and K_2. The bigger gain is, the faster convergence rate is, but less stability the learning process is. How to improve the convergence rate which can also guarantee stability is a hot topic in neural networks learning. In [24], the authors proposed a genetic search algorithm to get a stable and optimal learning rate. In our previous work [30] time-varying learning rate can accelerate the learning procedure.

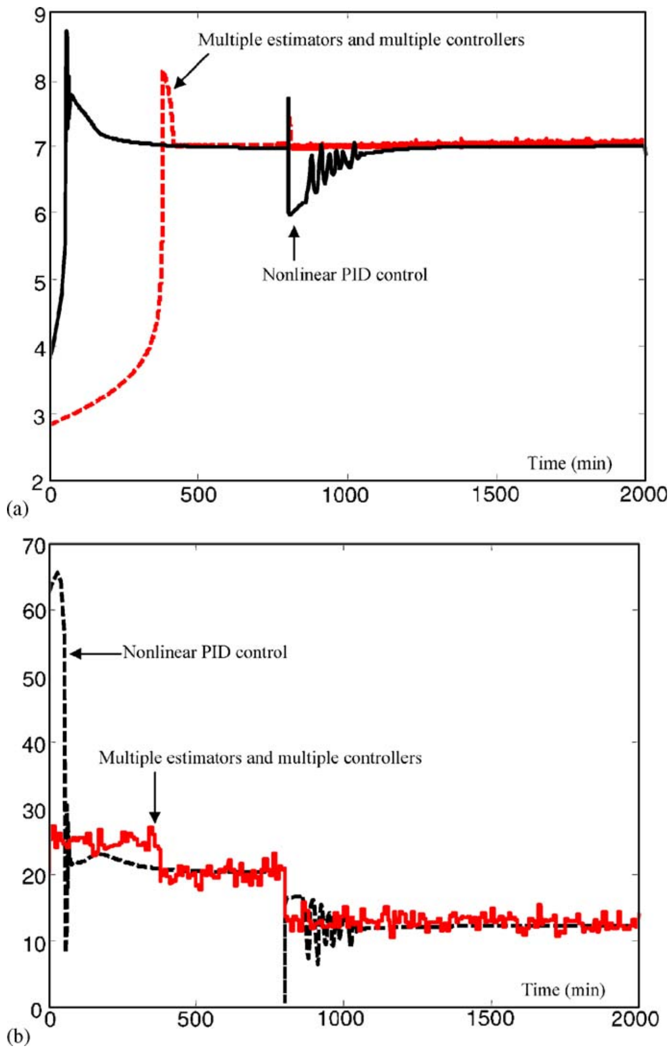


Fig. 11. Neural control and PID control. Control performance (q_3).

Another question is what maximum speed of state variations can be tracked with the given learning gains K_1 and K_2, it is a difficult problem because the exact convergence speed of weights updating is depended on gains K_1 and K_2, input data (persistent exciting) and the structure of neural networks (nonlinear model). For linear model, there are some discussions in [5].

6. Conclusion

In this paper we propose a new indirect adaptive control for complex unknown nonlinear system. Both identifier and controller are multiple models. The main contributions of this papers are: (1) A robust learning algorithm for single neural network is proposed. (2) A hysteresis switching algorithm is used to select the best neuro identifier, and the convergence of the multiple neuro identifier is proved. (3) Three indirect adaptive control are proposed. These neuro identifiers based controllers are also multiple. The approach presented in this paper is suitable for engineers, because they have more opportunities to select identification models and adaptive controllers. In the future, we will use some statistical analysis techniques, such as Monte Carlo runs, to study noise sensitivity of the multiple neural controllers.

References

[1] S. Battilotti, A. De Santis, Robust output feedback control of nonlinear stochastic systems using neural networks, IEEE Trans. Neural Networks 14 (1) (2003) 103–116.

[2] A. Bilbao-Guillerna, M. de la Sen, A. Ibeas, S. Alonso-Quesada, Robustly stable multiestimation scheme for adaptive control and identification with model reduction issues, Discrete Dyn. Nat. Soc. 10 (1) (2005) 31–67.

[3] A. Boukhris, G. Mourot, J. Ragot, Nonlinear dynamic system identification: a multi-model approach, Int. J. Control 72 (1999) 591–604.

[4] S.S. Ge, F. Hong, T.H. Lee, Adaptive neural network control of nonlinear systems with unknown time delays, IEEE Trans. Autom. Control 48 (11) (2003) 2004–2010.

[5] G.C. Goodwin, K.S. Sin, Adaptive Filtering, Prediction and Control, Prentice-Hall, Englewood Cliffs, NJ 07632, 1984.

[6] M.I. Jordan, R.A. Jacobs, Hierarchical mixtures of experts and the EM algorithm, Neural Comput. 6 (1994) 181–214.

[7] B. Kikens, M.N. Karim, Process identification with multiple neural network models, Int. J. Control 72 (1999) 576–590.

[8] D.G. Lainiotis, Partitioning: an unifying framework for adaptive systems, I: estimation, II: control, Proc. IEEE 54 (1976) 1126–1143.

[9] C-Y. Lee, J-J. Lee, Adaptive control for uncertain nonlinear systems based on multiple neural networks, IEEE Trans. Syst. Man Cybern. B 34 (1) (2004) 325–333.

[10] B. Maartensson, Adaptive stabilization, Ph.D. Thesis, Lund Institute of Technology, Sweden 1986.

[11] A.S. Morse, D.Q. Mayne, G.C. Goodwin, Applications of hysteresis switching in parameter adaptive control, IEEE Trans. Autom. Control 37 (9) (1992).

[12] K.S. Narendra, J. Balakrishnan, Adaptive control using multiple models, IEEE Trans. Autom. Control 42 (2) (1997) 171–187.

[13] K.S. Narendra, X. Cheng, Adaptive control of discrete-time systems using multiple models, IEEE Trans. Autom. Control 45 (9) (2000) 1669–1686.

[14] M.M. Polycarpou, Stable adaptive neural control scheme for nonlinear systems, IEEE Trans. Autom. Control 416 (2) (1996) 447–451.

[15] A.S. Poznyak, W. Yu, E.N. Sanchez, J.P. Perez, Nonlinear adaptive trajectory tracking using dynamic neural networks, IEEE Trans. Neural Networks 6 (10) (1999) 1402–1411.

[16] D. Prokhorov, D. Wunsch, Adaptive critic designs, IEEE Trans. Neural Networks 8 (5) (1997) 997–1007.

[17] G.A. Rovithakis, M.A. Christodoulou, Adaptive Control With Recurrent High-Order Neural Networks, Springer, Berlin, Germany, 2000.

[18] R.R. Selmic, F.L. Lewis, Multimodel neural networks identification and failure detection of nonlinear systems, in: 40th IEEE Conference on Decision and Control, Orlando, USA, 2001, pp. 3128–3133.

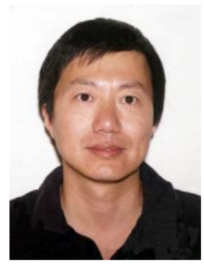
[19] M. de la Sen, J.J. Minambres, A.J. Garrido, A. Almansa, J.C. Soto, Basic theoretical results for describing expert systems. Application to the supervision of adaptation transients in a planar robot, Artif. Intell. 152 (2) (2004) 173–211.

[20] S. Seshagiri, H.K. Khalil, Output feedback control of nonlinear systems using RBF neural networks, IEEE Trans. Neural Networks 11 (1) (2000) 69–79.

[21] F.G. Shinskey, Process control systems, Application, Design, and Tuning, third ed., McGraw-Hill, New York, 1988.

[22] E.D. Sontag, Y. Wang, On characterization of the input-to-state stability property, Syst. Control Lett. 24 (1995) 351–359.

[23] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Syst. Man Cybern. 15 (1985) 116–132.
[24] C.H. Wang, H.L. Liu, C.T. Lin, Dynamic optimal learning rates of a certain class of fuzzy neural networks and its applications with genetic algorithm, IEEE Trans. Syst. Man Cybern. B 31 (2001) 467–475.
[25] K. Wang, A.N. Mitchel, Robustness and perturbation analysis of a class of artificial neural networks, Neural Networks 7 (2) (1994) 251–257.
[26] P. Werbos, Approximate dynamic programming for real-time control and neural modeling, in: D.A. White, D.A. Sofge (Eds.), Handbook of Intelligent Control, Van Nostrand Reinhold, New York, 1992, pp. 493–525.
[27] J.C. Willems, Least squares optimal control and algebraic Riccati equations, IEEE Trans. Autom. Control 16 (1971) 621–634.
[28] W. Yu, X. Li, Some new results on system identification with dynamic neural networks, IEEE Trans. Neural Networks 12 (2) (2001) 412–417.
[29] W. Yu, X. Li, Discrete-time neuro identification without robust modification, IEE Proc. Control Theory Appl. 150 (3) (2003) 311–316.
[30] W. Yu, X. Li, Fuzzy identification using fuzzy neural networks with stable learning algorithms, IEEE Trans. Fuzzy Syst. 12 (3) (2004) 411–420.
[31] T. Zhang, S.S. Ge, C.C. Hang, Stable adaptive control for a class of nonlinear systems using a modified Lyapunov function, IEEE Trans. Autom. Control 45 (2000) 129–132.



Wen Yu received the B.S. degree from Tsinghua University, Beijing, China in 1990 and the M.S. and Ph.D. degrees, both in Electrical Engineering, from Northeastern University, Shenyang, China, in 1992 and 1995, respectively. From 1995 to 1996, he served as a Lecture in the Department of Automatic Control at Northeastern University, Shenyang, China. In 1996, he joined CINVESTAV-IPN, México, where he is a professor in the Departamento de Control Automático. He has held a research position with the Instituto Mexicano del Petróleo, from December 2002 to November 2003. His research interests include adaptive control, neural networks, and fuzzy Control.