Inverse Identification and Closed-Loop Control of Dynamic Systems Using Neural Networks

MA Xiaomin

(Automation Department, Petroleum University (East China) · Shandong Dongying, 257062, PRC)

Abstract: A structure for inverse identification of dynamic linear or non-linear system using neural network is presented. Two types of closed-loop control schemes that combine the neural network inverse model with PID are proposed. The dynamic feed forward multilayer network for identification and control is trained by a novel learning algorithm based on U-D factorization Kalman filter (UDK). The potentials of the proposed structure and schemes are demonstrated by simulation studies.

Key words: neural networks; system identification; adaptive control

1 Introduction

Inverse models of dynamic systems play a crucial role in many control strategies. Characteristics of neural networks such as the approximation of nonlinear functions, the ability of learning from experiences, fault tolerance have made them one of the most effective ways to identification and control of non-linear systems. There have been some methods of inverse identification and control by neural networks[1~3]. Inverse identifications of many nonlinear systems have been satisfactorily achieved. However most of the structures of neural networks for inverse identification are static without taking account of possible time delay of the system or are relatively complex. There exist some drawbacks in the BP algorithm widely used to train the feed forward neural network for identification and control: low learning speed, local minimum. etc.. Therefore it is difficult to ensure the practicability and the reliability of these kinds of strategies.

In order to improve the speed, the precision, the robustess of inverse identification and control, a scheme of inverse identification of dynamic system using dynamic feed forward multilayer perceptron is presented and discussed. In place of the BP algorithm, a more effective learning algorithm based on the U-D factorization Kalman filter is used for training of the dynamic neural network. Combination of the trained inverse model with the PID evolves two types of feasible, stable closed-loop controllers. The PID is added to compensate for the remaining control error caused by inverse identification error, meanwhile providing teacher signal for on-line training of the inverse model.

2 Structure of Inverse Identification Using Neural Network

For a single-input, single-output (SISO) controllable and observable, n-dimension dynamic system, its difference equation can be written as:

$$y(k) = \mathrm{F}[y(k-1), y(k-2), \cdots, y(k-n), \\ u(k-d), u(k-d-1), \cdots, u(k-d-m+1)] \quad (1)$$

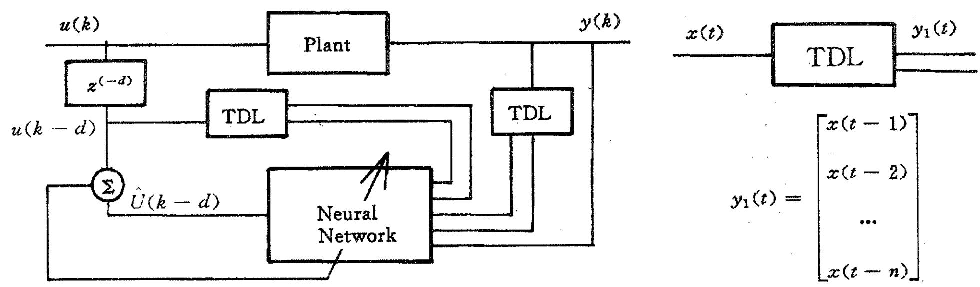
Manuscript received Nov. 9, 1995, revised Oct. 21, 1996.

where F[ ] is a linear or nonlinear transfer function, $u(k)$, $y(k)$ are inputs, outputs of the system respectively, d is a pure time delay of the system.

If the system described by equation (1) is invertible, a stucture of inverse identification using neural network can be established according to the following equation:

$$\hat{U}(k-d) = \mathrm{IF}[y(k), y(k-1), \cdots, y(k-n), u(k-d-1), \cdots, u(k-d-m+1)] \tag{2}$$

where IF[ ] is an inverse transfer function which represents output-input map of the system. A block diagram of the identification structure is shown in Fig. 1. Where TDL is a tapped delay line (Fig. 1(b)) which expresses various discrete time delays between output and input of system.



(a) (b)

Fig. 1 Structure of inverse identification using neural network

Here, when the system input with delay $u(k-d)$ is used as the neural output and the system output and its past outputs $y(k), y(k-1), \cdots, y(k-n)$, the system past inputs $u(k-d-1), u(k-d-2), \cdots, u(k-d-m+1)$ are used as neural inputs, the inverse transfer function of the unknown plant is obtained in the neural network. The input $u(k)$ is assumed to be random signal or PRBS signal uniformly distributed over certain interval. The error signal between the network output and the system input with d delay time $e(k) = u(k-d) - \hat{U}(k-d)$ is used to train the network. The square error (error energy function) is minimized and neural network output converges with the plant input as learning progresses. As mentioned above, since the neural network introduces TDL feedback it can approximate the inverse model of dynamic system.

To achieve the precise mapping to inverse model of the system in real time, a fast learning algorithm with good performance for training the neural network must be needed. The neural network chosen for inverse identification is a feed forward multilayer network trained by the UDK algorithm.

3 Neural Network for Identification and Control

Compared with the BP algorithm, the UDK learning algorithm has the advantages of fast learning speed, good numeric stability, insensitivity of learning parameters; avoidance of local minimum. The Kalman filter can be used to train the multilayer network. The U-D factorization technique also can be introduced to enhance the reliability and precision of training for inverse identification and control. So it is suitable for tasks of identification and control.

The multilayer perceptron is shown in Fig. 2, where $S(k) = \{s_{jp}\}$, $Y(k) = \{y_{jp}\}$ respectively represents input vector and summation output vector for each layer j, $W(k)=\{w_{jpi}\}$ is a weight vector of the network. For an inverse identification task, the UDK algorithm can be realized as the following steps:

1) Initialize:

a) Randomize all weights W_{jpi} as small numbers

b) Initialize the $U(0|0)$, $D(0|0)$

c) Equate the node offset $S_{j-1,0}=1$

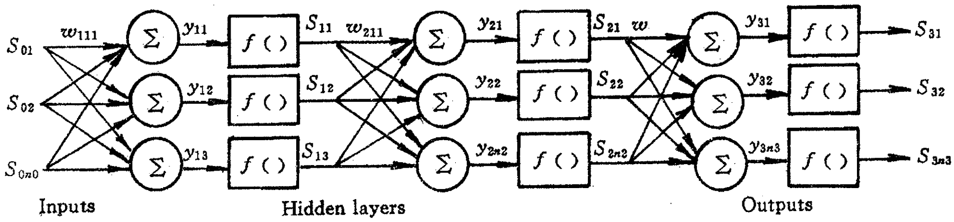


Fig. 2 Feedforward multilayer neural network

2) Select training pattern.

Randomly select an input/output pair of the network S_{jp}, O_p according to dimensions of the system and the range of the inputs and outputs.

$$\begin{cases} O_p = u(k-d), \\ s_{0p} = [s_{01}, s_{02}, \cdots, s_{0n}] \\ \quad = [y(k), y(k-1), \cdots, y(k-n), u(k-d-1), u(k-d-2), \cdots, u(k-d-m+1)], \\ s_{Lp} = \hat{U}(k-d). \end{cases} \tag{3}$$

3) Run selected pattern through the network

$$y_{jp} = \sum_{j=0}^{n_j-1} (s_{j-1,i} w_{jpi}) \tag{4}$$

$$S_{jp} = f(y_{jp}) = c_j(1-\exp(-ay_{jp}))/(1+\exp(-ay_{jp})), \tag{5}$$

where for every node p, n_j is the number of inputs to a node, constant a is the sigmoid slope and values of c_j are determined by the range of the input and output of the identified system.

4) Compute the backpropagate error signals

the error signal in the output layer L:

$$e_{Lp} = f'(y_{Lp})(O_p - S_{Lp}), \tag{6}$$

the error signal in the hidden layers:

$$e_{jp} = f'(y_{jp}) e_{j+1,i} w_{j+1,i,p}, \tag{7}$$

where $\quad f(y) = \mathrm{d}f/\mathrm{d}y = c_j 2a[\exp(-ay)]/[1+\exp(ay)]^2.$

5) The desired summation output:

$$d_p = f^{-1}(O_p) = \frac{1}{a}\ln((1+O_p/c_j)/(1-O_p/c_j)). \tag{8}$$

6) Compute the Kalman gain and U-D factor

a) iteration of state estimation:

$$\hat{W}(k|k-1) = \hat{W}(k-1|k-1), \tag{9}$$

b) iteration of covariance:

$$P(k|k-1) = P(k-1|k-1), \tag{10}$$

c) U-D factorization:

$$P(k|k-1) = U_{k|k-1}D(k|k-1)U^{T}_{k|k-1}, \tag{11}$$

d) the lately U-D factor and the Kalman gain matrix: for $i = 1,2\cdots,n$:

$$t(i) = U^{T}_{k|k-1}(i,j_1)h(j_1), \tag{12}$$

$$\gamma(i) = D_{k|k-1}(i,i)t(i), \tag{13}$$

$$\alpha(i+1) = \alpha(i) + t(i)\gamma(i), \tag{14}$$

$$\alpha(0) = 1(\text{Variance of measurement})$$

$$D_{k|k}(i,i) = D_{k|k-1}(i,i)\alpha(i)/\alpha(i+1), \tag{15}$$

$$B(i,j) = U_{k|k-1}(i,m)\gamma(m), \quad j = 1,2,\cdots,n, \tag{16}$$

$$\lambda(i) = -t(i)/\alpha(i), \tag{17}$$

$$U_{k|k}(i,j_1) = U_{k|k-1}(i,j_1) + B(i,j_1)\lambda(j_1), \quad j_1 > i, \tag{18}$$

$$K(k) = B(k,n)/\alpha(n). \tag{19}$$

7) Update the weights

For output layer L :

$$\hat{W}_L(k|k) = \hat{W}_L(k|k-1) + K_L(k)(d_p - y_{Lp}). \tag{20}$$

For hidden layer j :

$$\hat{W}_j(k|k) = \hat{W}_j(k|k-1) + K_j(k)e_{jp}\mu_j, \tag{21}$$

where μ_j is the learning step size of the neural network;

8) Test for completion: Repeat step 2)~7) until the mean-squared error of the network output of a fixed number of iteration is satisficd.

4 Closed - Loop Control Based on the Trained Inverse Model

Once the inverse identification of the plant has been completed, the control system can be constructed based on the trained inverse model.

4.1 Direct Inverse Control

One possible control law is shown in Fig. 3. The inverse model is simply cascaded with the controlled system in order that the composed system results in an identity mapping between desired response (i, e, the network inputs) and the controlled system output. Thus the network acts directly as the controller:

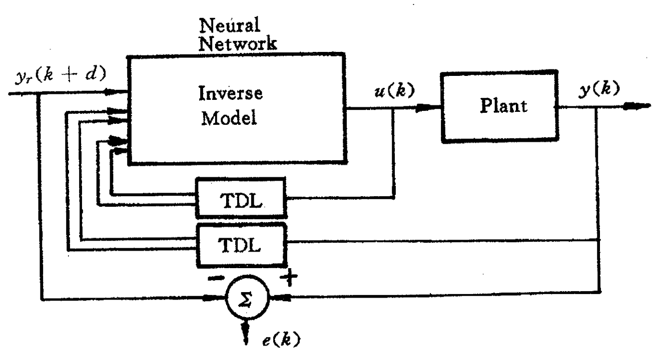


Fig. 3 Direct inverse control scheme

$$\hat{U}(k) = \text{IF}[y_r(k+d), y(k-1), y(k-2), \cdots, \hat{U}(k-1), \cdots, \hat{U}(k-m+1)] \tag{22}$$

or

$$\hat{U}(k) = \text{IF}[y_r(k+d), y_r(k+d-1), y_r(k+d-2), \cdots, \hat{U}(k-1), \cdots, \hat{U}(k-m+1)] \tag{23}$$

where $y_r(k)$ is a reference input of the controller. To achieve high control quality, the reference is added ahead of d discrete time due to the pure time delay of the plant. The objective of control is to minimize $\mathrm{MSE}=\frac{1}{N}\sum_k(y(k)-y_r(k))^2$.

Clearly this control law relies heavily on the fidelity of the inverse model used as the controller. The error of the inverse identification will always keep during the control period and it is difficult to guarantee the stability of the control system. To enhance robustness and accuracy of the control, two simple and realistic closed-loop inverse control designs in combination with the PID are introduced.

4.2 Modified Control 1: Inverse Model Cascades with the PID

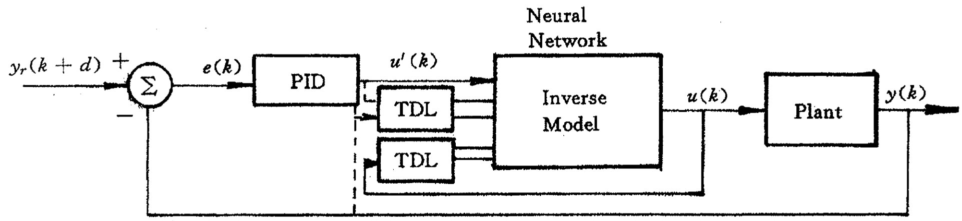


Fig. 4 Block diagram of modified control 1

The block diagram of the modified control 1 is shown in Fig. 4 PID, the trained inverse model. The plant are cascaded orderly. PID is used to control the composed system. Since mapping between the input and the output of the composed system is:

$$y(k)=\mathrm{F}[u(k)]=\mathrm{F}[\mathrm{IF}[u(k)]]=u'(k) \tag{24}$$

PID controller regards the controlled plant as a linear system with unit gain although the plant may be a nonlinear system. If the inverse of the model is not perfect, the colsed-loop PID helps to reduce the sensitivity of the whole system against this type of error and provides zero steady state ereor.

$$u'(k)=q_0[e(k)-e(k-1)]+q_1e(k)+q_2[e(k)-2e(k-1)+e(k-2)] \tag{25}$$

where q_0, q_1, q_2 are PID parameters respectively. $e(k)$ is the error between the reference signal and the output of the system.

4.3 Modified Control 2: Inverse Model Parallels with the PID

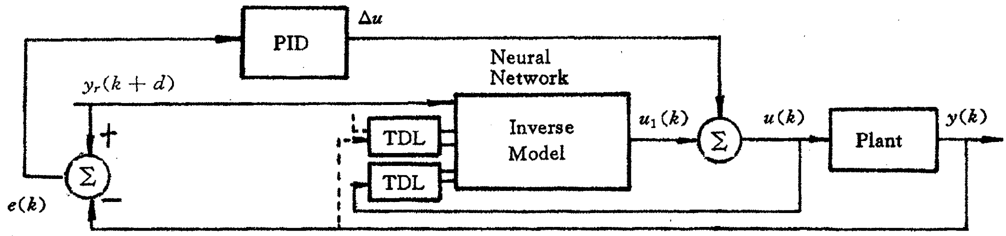


Fig. 5 Block diagram of modified control 2

The block diagram of the modified control 2 is shown in Fig. 5. A trained inverse dynamic model also cascades with the plant, but the PID controller, the trained inverse model are paralleled as the controller of the plant. A PID signal from the error between the reference signal and the output of the plant reduces the error of inverse identification at the output of the neural network:

$$u(k) = \Delta u + u_1(k), \tag{26}$$

$$\Delta u = u(k) - u_1(k). \tag{27}$$

For this architecture, the function of the inverse model and the function of PID that aims to decrease the control error are combined and made up mutually. The PID signal can also be used as teacher signal or the neural network for on-line training of the inverse identification to adapt model variations caused by distrubances or other unexpected factors. In this case the learned weights of the inverse identification are used as the initial weights of the on-line controller.

5 Simulation Results

In this section two examples are presented. The first example is chosen to emphasize the inverse identification and the control of a nonlinear dynamic plant, the second example is chosen to emphasize those of a linear dynamic plant with pure time delays. These experiments were programmed in C^{++} and run on a PC 486.

Example 1 The plant to be identified and controlled is described by the following difference equation:

$$y_p(k) = 6y_p(k-1)/(1 + y_p^2(k-1)) + u(k). \tag{28}$$

It can be verified that the output of the plant satisfies the inequality $|y_p(k)| \leqslant 5$ for any input $|u(k)| \leqslant 2$. According to the description in Fig. 1 the inverse identification can be selected by the equation:

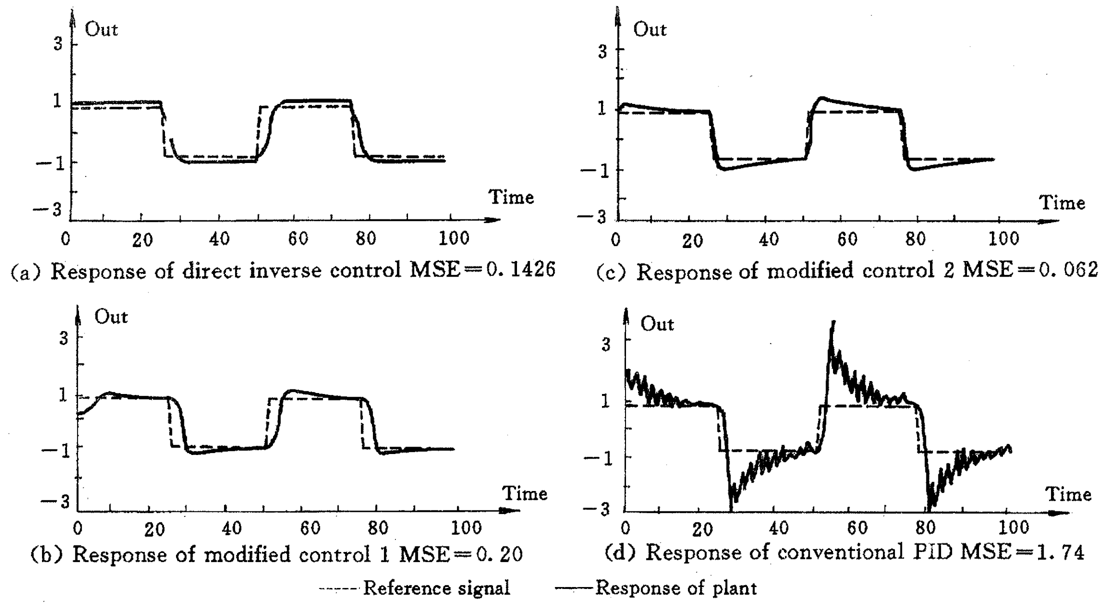
$$\hat{U}(k) = N[y_p(k), y_p(k-1)] \tag{29}$$

where N is a three layer network with 2 inputs, 20 hidden nodes, 1 output. The learning parameters are selected as: $c_j = 2.5$, $\mu_j = 140$, $a = 0.2$, $D(0|0) = 100\,I$. The input to the plant is a Gaussian random signal uniformly distributed in the interval $[-2, 2]$ as the desired output of the neural network. After training with the UDK algorithm in 2000 iterations the learned inverse model is added to different control architecture described in Fig. 3～Fig. 5 as controllers. When reference signals of the control are square waveform, the responses of the plant are shown in Fig. 6.

It may be seen from Fig. 6(a) that the inverse nonlinear dynamic model can correspond to the real plant well, which verifies the availability of the strategy as outlined in the previous paragraph. But the static error of the direct inverse control due to incompleteness of the inverse identification may always keep during control process. Fig. (6)(b), (c) show two modified control schemes which have improved the performance of the direct control since the PID and the feedback are successfully introduced. PID parameters for two cases are : $q_0 = 0.1$, $q_1 = 0.3$, $q_2 = 0.1$. Comparing with the modified control 1, the modified control 2 can trace the reference quickly and can be trained on-line to adapt some time-varying factors. The modified control 2, however, is not suitable for non-monotone system. Fig. 6(d) shows that the response of nonlinear system using PID can not converge well.

Example 2 The plant is described by the following difference equation:

$$y_p(k) = 0.3\,y_p(k-1) + 0.4\,y_p(k-2) + 1.25\,u(k-1) + 0.75\,u(k-2). \tag{30}$$



(a) Response of direct inverse control MSE=0. 1426
(c) Response of modified control 2 MSE=0. 062
(b) Response of modified control 1 MSE=0. 20
(d) Response of conventional PID MSE=1. 74

------ Reference signal ——Response of plant

Fig. 6 Example 1: control of Nonlinear plant using NNS

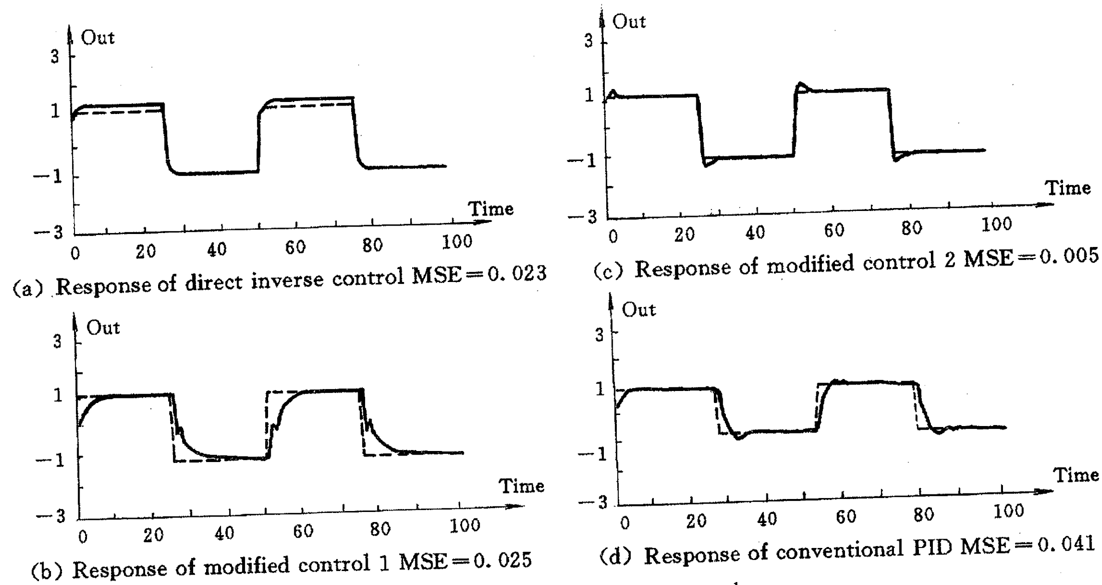
This is a two-phase dynamic linear system with time delay. The inverse identifier can be selected by the equation:

$$\hat{U}\ (k-1) = N[y_p(k), y_p(k-1), y_p(k-2), u(k-2)] \tag{31}$$

where N is a three layer neural network with 4 inputs, 10 hidden nodes and 1 output. The learning parameters are same as example 1 except $c_j = 1.1$. The input to the plant is a Gauss random signal uniformly distributed in the interval $[-1, 1]$. After training with the UDK algorithm in 2000 iteration, the trained model is respectively add to different control architecture described in Fig. 3~Fig. 5. The responses of the plant are shown in Fig. 7. It can be seen from Fig. 7 that inverse model of the dynamic linear plant with pure time delay can also correspond to the real plant, and two modified control laws remarkably decrease the error of the inverse identification thereby improve the performance of the control. Comparing with the conventional PID controller with optimum parameters, the modified control laws have higher response speed, ability of anti-disturbance, and the characteristics of the modified control systems are not sensitive to the PID parametes.

6 Conclusions

The basic concept of inverse identification of dynamic system using neural network has been discussed. A fast stable learning algorithm was introduced to train feedforward multilayer network. In particular two hybrid closed-loop control structures consisting of the direct inverse controller which parallels or cascades with a PID controller were proposed to improve the performance of the control. Here the control of nonlinear plant is transformed into the linear control, which provided a clue for control of nonlinear system. Simulation results have showed that with these methods of the inverse identification and two modified control, inverse models can be satisfactorily achieved, the accuracy, the response speed and the stability of the control can be evidently improved.



(a) Response of direct inverse control MSE=0.023

(b) Response of modified control 1 MSE=0.025

(c) Response of modified control 2 MSE=0.005

(d) Response of conventional PID MSE=0.041

-----Reference signal ——Response plant

Fig. 7 Example 2: control of linear plant

References

1 Hunt, K. J. et al.. Neural networks for control systems——A survey. Automatica, 1992, 28(6): 1083−1112

2 Ma Xiaomin. A learning algorithm based on the U-D factorized Kalman filter fo training multilayer perceptrons. Signal Processing, 1995, 11(4): 276−282

3 Yamada, T. and Yabuta, T.. Dynamic system identification using neural networks. IEEE Trans. Systems, Man. and Cyberntics, 1993, SMC-23(1): 204

基于神经网络的动态系统逆模型辨识及闭环控制

马晓敏

(石油大学(华东)自动化系・山东东营,257062)

摘要:本文提出一种动态线性或非线性系统的神经网络逆模型辨识结构.并引出两种 PID 与神经网络逆模型相结合的自适应控制方案.神经网络模型采用基于 U-D 分解卡尔曼滤波学习算法(UDK)的动态前向多层网.仿真结果表明了所述辨识方案的有效性及特点.

关键词:神经网络;系统辨识;自适应控制

本文作者简介

马晓敏 1964 年生.副教授.1984 年安徽大学无线电系毕业,获学士学位,1989 年北京航空航天大学二系毕业,获硕士学位.目前在石油大学自动化系从事教学、科研工作.主要研究领域为智能信息处理及自适应控制.现已在国内外发表论文二十余篇.