Journal of Southeast University (English Edition) June 2001 Vol.17 No.1 ISSN 1003—7985

Identification of Hammerstein Model Using Hybrid Neural Networks

Li Shihua[1*] Li Qi[2] Li Jie[3]

([1] Department of Automatic Control Engineering, Southeast University, Nanjing 210096, China)
([2] Research Institute of Automation, Southeast University, Nanjing 210096, China)
([3] Nanjing Automation Research Institute, Nanjing 210003, China)

Abstract: The identification problem of Hammerstein model with extension to the multi-input multi-output (MIMO) case is studied. The proposed identification method uses a hybrid neural network (HNN) which consists of a multi-layer feed-forward neural network (MFNN) in cascade with a linear neural network (LNN). A unified back-propagation (BP) algorithm is proposed to estimate the weights and the biases of the MFNN and the LNN simultaneously. Numerical examples are provided to show the efficiency of the proposed method.

Key words: neural networks, nonlinear systems identification, Hammerstein model

The area of system identification has received a lot of attention over the last two decades. One of the nonlinear realizations frequently studied is the Hammerstein model, which is composed of a static nonlinearity in series with a linear dynamic part[1]. The Hammerstein model has proved successfully in providing a simple nonlinear model appropriate for a wide number of applications including actuator modeling, auditory and visual identification, non-Gaussian signal analysis and nonlinear communication filtering problems[1-5].

The identification methods for Hammerstein model can be divided into two directions. The first one[1,2] uses the polynomial to describe the nonlinearity. The problem is thus transformed into multi-input single-output (MISO) linear identification problem. The main drawbacks of this approach are the assumption that the nonlinearity is of a polynomial form and the increase of the inputs number in the linear identification problem. The other one is followed by different nonparametric methods. One of them[3,4] uses the kernels regression estimates to identify the nonlinearity. The identification of the nonlinearity is done separately from the identification of the linear part. Also there are methods using neural networks to identify the nonlinearity[5,6]. In [5], a hybrid model, which consists of an MFNN and an auto-regressive moving average model (ARMA), is employed to identify the Hammerstein model. The method proposed in [6] assumes that the linear part is completely known.

An identification method using hybrid neural networks, which was first proposed for identifying Wiener model[7], is now generalized to the identification of Hammerstein model. Here, the nonlinear part is approximated by an MFNN and the linear part is modeled by a linear neural network (LNN). Thus a method entirely using neural networks is proposed. The advantage of this method is that a unified back-propagation learning algorithm can be derived to estimate the weights of the MFNN and the LNN from the input-output data simultaneously. The method is also generalized to the MIMO case.

1 Identification of SISO Hammerstein Model

Fig.1 shows the structure of the hybrid neural network including the structure of the MFNN and the LNN. The HNN consists of an MFNN in cascade with an LNN model. The MFNN is a typical multi-layer network with one input, one output, and one hidden layer. Here, for simplicity of discussion, only one hidden layer is shown in Fig.1. More than one hidden layer can be constructed. For the SISO Hammerstein model, the LNN can be considered as a multi-input single-output neuron without bias. Note that the inputs of the neuron are composed of time delayed values of the outputs of the MFNN and the Hammerstein model. Moreover, the activation function of the neuron is selected as a linear transfer function. Thus, the coefficients of the linear part are represented by the weights of the LNN. Compared to the other identification methods for the linear dynamics, using an LNN to model the linear part is convenient to the derivation of a unified BP algorithm to estimate the weights and the biases of the MFNN and the LNN simultaneously.

Received 2000-06-20.

* Born in 1975, male, graduate.

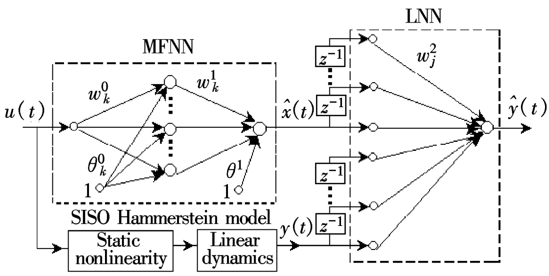


Fig.1 Structure of the HNN

The Hammerstein model is represented by the following equation:

$$A(q^{-1})y(t) = B(q^{-1})x(t) + \xi(t) \tag{1}$$

where $x(t) = f(u(t))$ is the output of the nonlinear part; $y(t)$ is the output; $u(t)$ is the input of the system; and $\xi(t)$ is the output noise of the system. The linear subsystems are defined as

$$\left.\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \cdots + a_n q^{-n} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + \cdots + b_m q^{-m} \end{aligned}\right\} \tag{2}$$

where q^{-1} is the delay operator; n, m are the orders. In this paper, the orders n, m are supposed to be known. The linear transfer function $B(q^{-1})/A(q^{-1})$ is assumed to be stable.

Due to the limitation of length, the learning algorithms of the weights and the biases of the HNN for the SISO Hammerstein model are omitted here. However, they can be easily concluded from the discussions on the identification of the MIMO case in the following section.

2 Generalizations to MIMO Hammerstein Model

Two possible structures can be used to describe MIMO Hammerstein model depending on whether the nonlinearities are separate or combined[5] (See Fig.2). HNN can be extended to MIMO systems with *nu* inputs and *ny* outputs. The corresponding HNN identification models for the two structures of MIMO Hammerstein model are shown in Fig.3. TDL in Fig.3 denotes a tapped delay line whose output vector has the delayed values of the input signal for its elements.

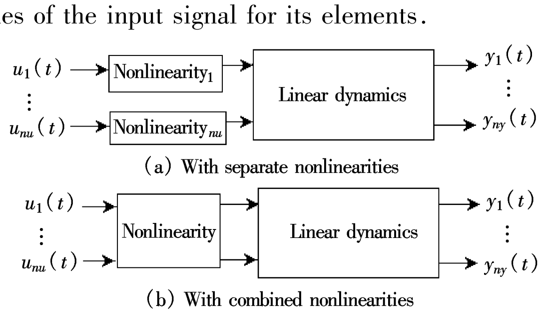


Fig.2 MIMO Hammerstein model

The difference between the structures of the two identification models lies in the nonlinearities. In Fig.3(a), *nu* MFNNs are exploited to estimate *nu* independent nonlinearities respectively. The structure of each MFNN is the same as Fig.1. Hence, the time delayed values of the output of *nu* MFNNs and the *ny* outputs of the MIMO Hammerstein model form the input vector to the LNN whose output $\hat{y}_1(t), \cdots, \hat{y}_{ny}(t)$ corresponds to the estimate of the plant at any instant of time t. As shown in Fig.3(b), an MFNN with *nu* inputs, single hidden layer, *nu* outputs is employed to approximate the combined nonlinearities. Hence, the time delayed values of the *nu* outputs of the MFNN and the *ny* outputs of the MIMO Hammerstein model form the input vector to the LNN. Note that the method using polynomial is not efficient for the identification of the general MIMO Hammerstein model especially when the nonlinearities are combined.

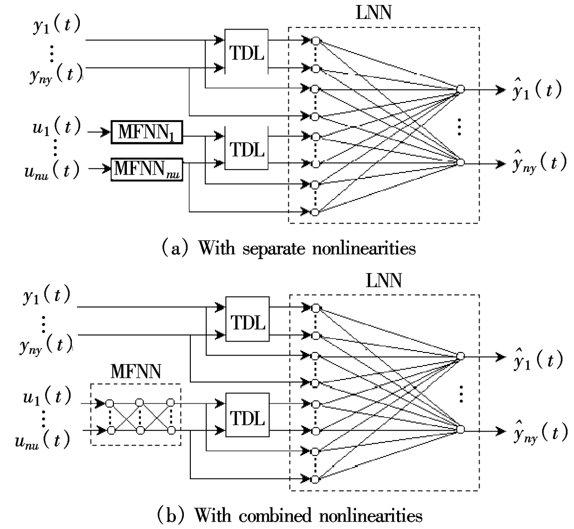


Fig.3 Structure of HNN identification model

The description of MIMO linear dynamic systems can be written as[8]

$$\begin{aligned} \boldsymbol{Y}(t) = &-\boldsymbol{A}_1\boldsymbol{Y}(t-1) - \cdots - \boldsymbol{A}_{na}\boldsymbol{Y}(t-na) \\ &+\boldsymbol{B}_0\boldsymbol{X}(t) + \cdots + \boldsymbol{B}_{nb}\boldsymbol{X}(t-nb) + \boldsymbol{\xi}(t) \end{aligned} \tag{3}$$

where $\boldsymbol{Y}(t)$ is the $ny \times 1$ output vector; $\boldsymbol{X}(t)$ is the $nu \times 1$ input vector; $\boldsymbol{\xi}(t)$ is the $ny \times 1$ noise vector; $\boldsymbol{A}_1, \cdots, \boldsymbol{A}_{na}$ are $ny \times ny$ matrices; $\boldsymbol{B}_0, \cdots, \boldsymbol{B}_{nb}$ are $ny \times nu$ matrices; and na, nb represent the order of the model. The input vector is defined as $\boldsymbol{U}(t) = [u_1(t) \cdots u_{nu}(t)]^{\mathrm{T}}$.

2.1 Identification of MIMO Hammerstein with separate nonlinearities

For MFNN_i $(i = 1, \cdots, nu)$, the input is $u_i(t)$ and the input to the k-th hidden unit is

$$\mathrm{net}_{ik}^0(t) = w_{ik}^0(t)u_i(t) + \theta_{ik}^0(t) \tag{4}$$

where $w_{ik}^0(t)$ are the weights from the input layer to the hidden layer; $\theta_{ik}^{0}(t)$ are the bias terms. The output of the MFNN_i is

$$\hat{x}_i(t) = \sum_{k=1}^{H} w_{ki}^{1}(t)\mathrm{sig}(\mathrm{net}_{ik}^{0}(t)) + \theta_i^{1}(t) \quad (5)$$

where $w_{ki}^{1}(t)$ are the weights from the hidden layer to the output layer; $\theta_i^{1}(t)$ are the bias terms; $\mathrm{sig}(\cdot)$ is the activation function of the hidden units. It can be defined as

$$\mathrm{sig}(x) = (1 - \mathrm{e}^{-x})/(1 + \mathrm{e}^{-x})$$

Define the vector $\hat{\boldsymbol{X}}(t) = [\hat{x}_1(t) \cdots \hat{x}_{nu}(t)]^{\mathrm{T}}$, $\boldsymbol{Y}(t) = [y_1(t) \cdots y_{ny}(t)]^{\mathrm{T}}$. The input vector to the LNN is represented by

$$\boldsymbol{O}^2(t) = [\hat{\boldsymbol{X}}^{\mathrm{T}}(t) \cdots \hat{\boldsymbol{X}}^{\mathrm{T}}(t-nb)\ \boldsymbol{Y}^{\mathrm{T}}(t-1) \cdots \boldsymbol{Y}^{\mathrm{T}}(t-na)]^{\mathrm{T}} \quad (6)$$

That is

$$\boldsymbol{O}^2(t) = [\hat{x}_1(t) \cdots \hat{x}_{nu}(t)\ \hat{x}_1(t-1) \cdots \hat{x}_{nu}(t-1) \cdots y_1(t-na) \cdots y_{ny}(t-na)]^{\mathrm{T}} \quad (7)$$

Define $o_l^2(l = 1,\cdots,N; N = nu \times (nb+1) + ny \times na)$ to represent the l-th element of the vector $\boldsymbol{O}^2$. Note $o_i^2(i = 1,\cdots,nu)$ are the same as $\hat{x}_i(t)(i = 1,\cdots,nu)$.

The output vector of the LNN is $\hat{\boldsymbol{Y}}(t) = [\hat{y}_1(t) \cdots \hat{y}_{ny}(t)]^{\mathrm{T}}$, among which

$$\hat{y}_j(t) = \sum_{l=1}^{N} w_{lj}^{2}(t) o_l^{2}(t) \qquad j = 1,\cdots,ny \quad (8)$$

where $w_{lj}^2(t)$ are the weights.

To develop updating rules for the weights and biases of the HNN, the performance function has the form

$$J_m = \frac{1}{2}\sum_j (y_j(t) - \hat{y}_j(t))^2 \quad (9)$$

Thus, according to the BP algorithm, the weights and the biases of the HNN are updated according to the following rules:

$$w_{lj}^2(t+1) = w_{lj}^2(t) + \eta(y_j(t) - \hat{y}_j(t))o_l^2(t) \quad (10)$$

$$w_{ki}^1(t+1) = w_{ki}^1(t) + \eta\mathrm{sig}(\mathrm{net}_{ki}^0(t)) \times \sum_{j=1}^{ny} w_{ij}^2(y_j(t) - \hat{y}_j(t)) \quad (11)$$

$$\theta_i^1(t+1) = \theta_i^1(t) + \eta\sum_{j=1}^{ny} w_{ij}^2(y_j(t) - \hat{y}_j(t)) \quad (12)$$

$$w_{ik}^0(t+1) = w_{ik}^0(t) + \eta(w_{ki}^1(t)\mathrm{sig}'(\mathrm{net}_{ik}^0(t)) \times u_i(t))\sum_{j=1}^{ny} w_{ij}^2(y_j(t) - \hat{y}_j(t)) \quad (13)$$

$$\theta_{ik}^0(t+1) = \theta_{ik}^0(t) + \eta(w_{ki}^1(t)\mathrm{sig}'(\mathrm{net}_{ik}^0(t))) \times \sum_{j=1}^{ny} w_{ij}^2(y_j(t) - \hat{y}_j(t)) \quad (14)$$

where $\mathrm{sig}'(x) = 0.5(1 - \mathrm{sig}^2(x))$ is the derivative of the activation function.

2.2 Identification of MIMO Hammerstein with combined nonlinearities

The input vector to the MFNN is $\boldsymbol{U}(t) = [u_1(t) \cdots u_{nu}(t)]^{\mathrm{T}}$. The input to the k-th hidden unit is

$$\mathrm{net}_k^0(t) = \sum_{i=1}^{nu} w_{ik}^0(t)u_i(t) + \theta_k^0(t) \quad (15)$$

where $w_{ik}^0(t)$ are the weights from the input layer to the hidden layer, $\theta_k^0(t)$ are the bias terms.

The output vector of the MFNN is defined as $\hat{\boldsymbol{X}}(t) = [\hat{x}_1(t) \cdots \hat{x}_{nu}(t)]^{\mathrm{T}}$ among which

$$\hat{x}_i(t) = \sum_{k=1}^{H} w_{ki}^1\mathrm{sig}(\mathrm{net}_k^0(t)) + \theta_i^1(t) \quad (16)$$

where $w_{ki}^1(t)$ are the weights from the hidden layer to the output layer; $\theta_i^1(t)$ are the bias terms.

From the previous discussion we know that the LNN here has the same form as described in Eqs. (6) – (8). Moreover, the performance function is selected the same as Eq. (9) and the updated rule of the weights of the LNN is the same as Eq. (10). Thus, according to the BP algorithm, the weights and the biases of the MFNN are adjusted according to the following rules:

$$w_{ki}^1(t+1) = w_{ki}^1(t) + \eta\mathrm{sig}(\mathrm{net}_k^0(t)) \times \sum_{j=1}^{ny} w_{ij}^2(y_j(t) - \hat{y}_j(t)) \quad (17)$$

$$\theta_i^1(t+1) = \theta_i^1(t) + \eta\sum_{j=1}^{ny} \times w_{ij}^2(y_j(t) - \hat{y}_j(t)) \quad (18)$$

$$w_{ik}^0(t+1) = w_{ik}^0(t) + \eta\mathrm{sig}'(\mathrm{net}_k^0(t))u_i(t) \times \sum_{i=1}^{nu}\left(\sum_{j=1}^{ny} w_{ij}^2(y_j(t) - \hat{y}_j(t))\right)w_{ki}^1(t) \quad (19)$$

$$\theta_{ik}^0(t+1) = \theta_{ik}^0(t) + \eta\mathrm{sig}'(\mathrm{net}_k^0(t)) \times \sum_{i=1}^{nu}\left(\sum_{j=1}^{ny} w_{ij}^2(y_j(t) - \hat{y}_j(t))\right)w_{ki}^1(t) \quad (20)$$

As we all know, due to the complex nonlinear nature of the neural networks, it is difficult to perform a convergence analysis of BP algorithm. Most of the results in the neural networks literature are based on simulation results. It can be sure that the choice of the step size may be very important in the convergence of the BP algorithm. A small step size will make the convergence very slow while large step size may cause divergence in the algorithm. Due to the different structure complexity, the step sizes of the MFNN and the LNN may be chosen differently to increase the speed of convergence.

3 Simulation Results

Example 1 Consider a process with two inputs and two outputs described by the following equation[5]:

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 0.5 & -0.1 \\ 0.8 & -0.7 \end{bmatrix}\begin{bmatrix} y_1(t-1) \\ y_2(t-1) \end{bmatrix} + \begin{bmatrix} -0.3 & 0.2 \\ 0.9 & -0.5 \end{bmatrix}\begin{bmatrix} y_1(t-2) \\ y_2(t-2) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} s_1(u_1(t)) \\ s_2(u_2(t)) \end{bmatrix} + \begin{bmatrix} \zeta_1(t) \\ \zeta_2(t) \end{bmatrix}$$

where the nonlinearities are given by

$$s_1(u_1(t)) = (1 - e^{-4u_1(t)})/(1 + e^{-4u_1(t)})$$

$$s_2(u_2(t)) = 0.5u_2^3(t)$$

$\zeta_1(t), \zeta_2(t)$ are white, zero-mean Gaussian distributed noises of variances 0.5. The inputs to the plant and the HNN are random signals whose amplitude is uniformly distributed in the interval [−2,2]. Both MFNNs employ 10 hidden nodes and the number of input node of the LNN is 6. Two identification procedures of the plant, with noise or not, are implemented respectively. The BP algorithm employs two different step sizes, which are 0.1, 0.01 (the plant without noise) and 0.02, 0.001 (the plant with noise), for the MFNN and the LNN, respectively. The numerical results of example 1 are located in the third row and the fourth row of Tab.1. The numerical results of the method in [5] are in the second row. Fig.4 shows the actual and identified nonlinearities when the noise is not added to the plant.

Example 2 Consider a process the same as the process of example 1 except that the nonlinearities are given by

$$s_1(u_1(t), u_2(t)) = \frac{(1 - e^{-(u_1(t)+u_2(t))})}{(1 + e^{-(u_1(t)+u_2(t))})}$$

$$s_2(u_1(t), u_2(t)) = 0.25u_1^2(t)u_2(t)$$

Tab.1 Numerical results of example 1 and example 2

	A_1	A_2	B_0	Time steps	$RMSE_1$	$RMSE_2$
Desired value	$\begin{bmatrix} 0.5 & -0.1 \\ 0.8 & -0.7 \end{bmatrix}$	$\begin{bmatrix} -0.3 & 0.2 \\ 0.9 & -0.5 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$			
Result of [5]	$\begin{bmatrix} 0.5027 & -0.0951 \\ 0.7958 & -0.7015 \end{bmatrix}$	$\begin{bmatrix} -0.3034 & 0.2044 \\ 0.8963 & -0.5018 \end{bmatrix}$	$\begin{bmatrix} -0.3155 & 0.1846 \\ -0.0679 & 0.2464 \end{bmatrix}$		0.0115	0.0132
Example 1 (without noise)	$\begin{bmatrix} 0.5000 & -0.1001 \\ 0.7982 & -0.7011 \end{bmatrix}$	$\begin{bmatrix} -0.3002 & 0.2000 \\ 0.8993 & -0.5005 \end{bmatrix}$	$\begin{bmatrix} 0.3187 & 0.0000 \\ 0.0012 & -0.5102 \end{bmatrix}$	5000	0.0043	0.0092
Example 1 (with noise)	$\begin{bmatrix} 0.5029 & -0.1088 \\ 0.7724 & -0.6903 \end{bmatrix}$	$\begin{bmatrix} -0.3111 & 0.2185 \\ 0.9005 & -0.4887 \end{bmatrix}$	$\begin{bmatrix} -0.3126 & -0.0039 \\ 0.0016 & 0.7575 \end{bmatrix}$	20000	0.0076	0.0563
Example 2 (without noise)	$\begin{bmatrix} 0.5001 & -0.1002 \\ 0.7997 & -0.7003 \end{bmatrix}$	$\begin{bmatrix} -0.3000 & 0.2001 \\ 0.9004 & -0.5019 \end{bmatrix}$	$\begin{bmatrix} 0.3197 & 0.0141 \\ 0.0078 & 1.0460 \end{bmatrix}$	20000	0.0057	0.0083
Example 2 (with noise)	$\begin{bmatrix} 0.4577 & -0.1170 \\ 0.8066 & -0.7292 \end{bmatrix}$	$\begin{bmatrix} -0.2807 & 0.1972 \\ 0.8765 & -0.4939 \end{bmatrix}$	$\begin{bmatrix} 0.6455 & 0.0242 \\ 0.0492 & 0.6589 \end{bmatrix}$	50000	0.0225	0.0821

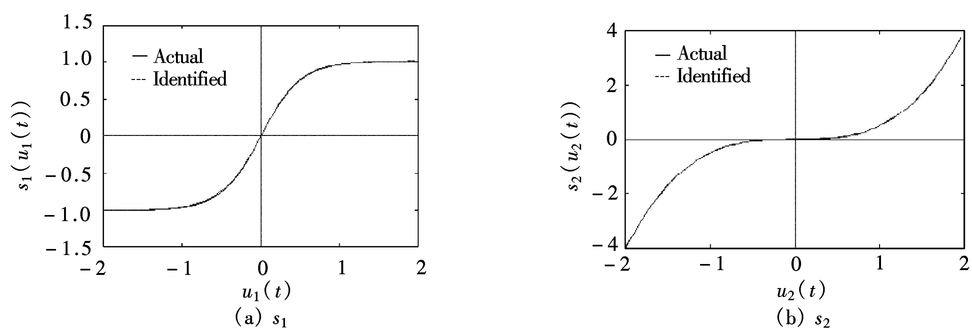
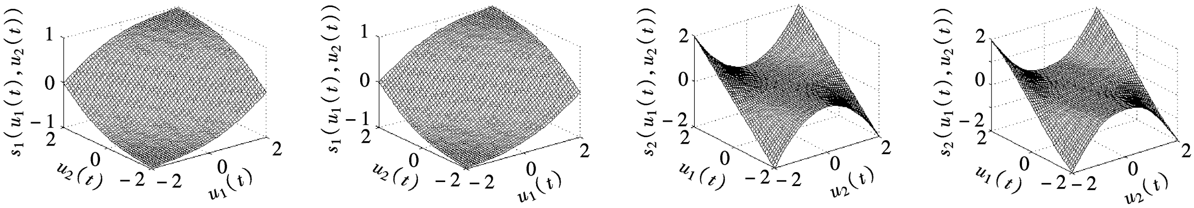


Fig.4 Actual and identified nonlinearities of example 1

The input to the plant and the HNN is random signals whose amplitude is uniformly distributed in the interval [−2,2]. The MFNN employs 30 hidden nodes and the number of input node of the LNN is 6. Two identification procedures of the plant, with noise or not, are implemented respectively. The BP method employs two different step sizes, which are 0.04, 0.007 (the plant without noise) and 0.01, 0.01 (the plant without noise), for the MFNN and the LNN respectively. Numerical results are also shown in Tab.1. Fig.5 shows the actual and identified nonlinearities when the noise is not added to the plant.



(a) Actual s_1 (b) Identified s_1 (c) Actual s_2 (d) Identified s_2

Fig.5 Actual and identified nonlinearities of example 2

4 Conclusion

In this study, a method for the identification of the Hammerstein model using a hybrid neural network, which is consisted of a static MFNN in cascade with an LNN, has been developed. The use of the MFNN to model the nonlinearity, compared to polynomial approximation used in the literatures, makes it possible to model any kind of nonlinearities. The use of the LNN to model the linear dynamics is especially convenient to the exploitation of a unified back propagation method for the estimate of the weights and the biases of the hybrid neural network. The two parts of the Hammerstein can be well estimated simultaneously. Moreover, the proposed method is applicable to MIMO Hammerstein systems with separate or combined nonlinearities. Simulation results show the efficiency of the proposed method.

References

1 K. S. Narendra, and P. G. Gallman, An iterative method for the identification of nonlinear system using the Hammerstein model, *IEEE Trans Automat Control*, vol.11, no.2, pp. 546 – 550, 1966

2 F. H. I. Chang, and R. Luus, A noniterative method for identification using Hammerstein model, *IEEE Trans Automat Control*, vol. 16, no.5, pp. 464 – 468, 1971

3 W. Greblicki, and M. Pawlak, Identification of discrete Hammerstein systems using kernel regression estimates, *IEEE Trans Automat Control*, vol. 31, no. 1, pp. 74 – 77, 1986

4 W. Greblicki, and M. Pawlak, Nonparametric identification of Hammerstein systems, *IEEE Trans Inform Theory*, vol. 35, no.2, pp. 409 – 418, 1989

5 H. Al-Duwaish, and M. N. Karim, A new method for the identification of Hammerstein model, *Automatica*, vol. 33, no. 10, pp.1871 – 1875, 1997

6 K. S. Narendra, and K. Parthasarathy, Identification and control of dynamic systems using neural networks, *IEEE Trans Neural Networks*, vol.1, no.1, pp. 4 – 27, 1990

7 S. Li, F. Wu, and Q. Li, Identification of Wiener model using dynamic artificial neural networks, *Control Theory and Applications*, vol.17, no.1, pp. 92 – 95, 2000

8 L. Ljung, and T. Soderstrom, *Theory and practice of recursive identification*, MIT Press, Cambridge, MA,1983

利用混合神经网络辨识 Hammerstein 模型的方法

李世华[1] 李 奇[2] 李 捷[3]

([1] 东南大学自动控制系,南京 210096)
([2] 东南大学自动化研究所,南京 210096)
([3] 南京自动化研究所,南京 210003)

摘 要 研究了 Hammerstein 模型的辨识问题,并考虑了多输入多输出(MIMO)情况.提出一种混合神经网络辨识模型,该模型由一个多层前馈神经网络(MFNN)与一个线性神经网络(LNN)串联而成.给出了一个反向传播(BP)算法同步训练该混合神经网络的权值和阈值.仿真结果表明了该方法的有效性.

关键词 神经网络, 非线性系统辨识, Hammerstein 模型

中图分类号 TP14