第 29 卷 第 5 期 自 动 化 学 报 Vol. 29, No. 5
2003 年 9 月 ACTA AUTOMATICA SINICA Sept., 2003

SDIDRNN Based Dynamical Adaptive Tracking Identification of Robot Manipulators[1)]

JIANG Chun-Fu YU Yue-Qing
(*College of Mechanical Engineering, Beijing University of Technology, Beijing* 100022)
(E-mail: emailbox_cfj@163.com)

Abstract A new neural network model named state delay input dynamical recurrent neural network is presented. The model with new topological structure and learning algorithm has significance for weight matrices and makes training process of weights become more distinct and straightforward. Simulation shows that the speed of learning and convergence is improved by input of the priori input-output state knowledge. The new neural network is then applied to identification of robotic dynamics through strategy of combining information obtained from simplified model with identification of unknown nonlinear dynamical information. This method reduces computing time and accelerate the identification speed. Simulation results show the efficiency of the new neural network and the strategy.

Key words Dynamical recurrent neural network, tracking identification, robot manipulators

1 Introduction

Industrial robot is a multi-input-multi-output (MIMO) system with high nonlinearity and uncertainties in kinematics and dynamics. External disturbances and inherent complexities in a robot system, such as inertial payloads, interactions between joints, correlations between robot's positions, orientations, gravitational effects, etc., will definitely arouse error between theoretical and actual trajectories. This will lead to decreased working performance in a working cycle. Although intensive research has been done in literatures to develop control algorithms for a robot to improve its performance, its kinematic or dynamic model is simplified, excluding much information that would affect working performance. It is difficult to identify the lost modeling knowledge by conventional methods. Neural network has competitive ability of approximating any nonlinear continuous functions, and neural network's applications become new focus in identifying unformulated factors in robot study.

Identification of robot system using neural network can involove network structure and identification scheme. For neural network structures, researchers in different fields have proposed topological structures and learning algorithms that can be used in different research fields[1,2]. As to identification strategies, one can approximate the forward dynamics or inverse dynamics that are assumed totally unknown. However, doing so means heavy computational burden and decreased ability in online identification and control. So another way is dividing the nonlinearity of the system into two parts[3]. The first part can be expressed by formula, such as gravity and Coriolis. The second part includes all the nonlinearities that cannot be formularized, such as magnetic field of servo actuators and the damp of air. For the reason that the simplified model represents the system's primary motion property, it is more reasonable to identify the second part of nonlinearity, which is less time-consuming and is more possible for online identification and control.

1) Supported by National Natural Science Foundation of P. R. China(59975001) and Beijing Natural Science Foundation(3012003)
Received January 28, 2002; in revised form January 27, 2003
收稿日期 2002-01-28; 收修改稿日期 2003-01-27

2 State delay input dynamical recurrent neural network

In this section, the state delay input dynamical recurrent neural network (SDIDRNN) is proposed with topological structure shown in Fig. 1. The input and output of the network are added to the last hidden layer as one part of input. Different from CIDRNN[2], in SDIDRNN the input-output knowledge is not added to contexts shown by the dashed part in Fig. 1, but to a autonomous group of nodes named state contexts. Specially, these nodes are identical in structure with the input layer of Elman network if there are only three layers, but it is not the case for more layers. Generally, they are treated not as input layer nodes, but as the parallel neurons to context ones. Note that if there is no specific denotation, the "hidden layer" in this paper refers to the last hidden layer connecting with the output layer. The matrix between state context nodes and hidden layer nodes is called feed-forward filter weight matrix with definite meanings. In compound input dynamical recurrent neural networks(CIDRNN) the matrix is constant, making the network's generalization limited and the performance dependent on the filter factor. In SDIDRNN it is designed to be variable and weights are updated by BP update rule to improve adaptive capability. The number of state context neurons equals to the sum of nodes in both input layer and output layer. Each neuron holds the prior state information of one node in the two layers. In order to get satisfactory convergent and learning rate, one can preprocess the prior state knowledge by multiplying a filter factor λ. As a factor of experience, λ can be greater or less than 1. The state context influence upon the output layer is strengthened when $\lambda > 1$, while it is weakened if λ is less than 1. If $\lambda = 0$, then SDIDRNN is an Elman network. Thus, the Elman network can be considered as a special case of SDIDRNN.

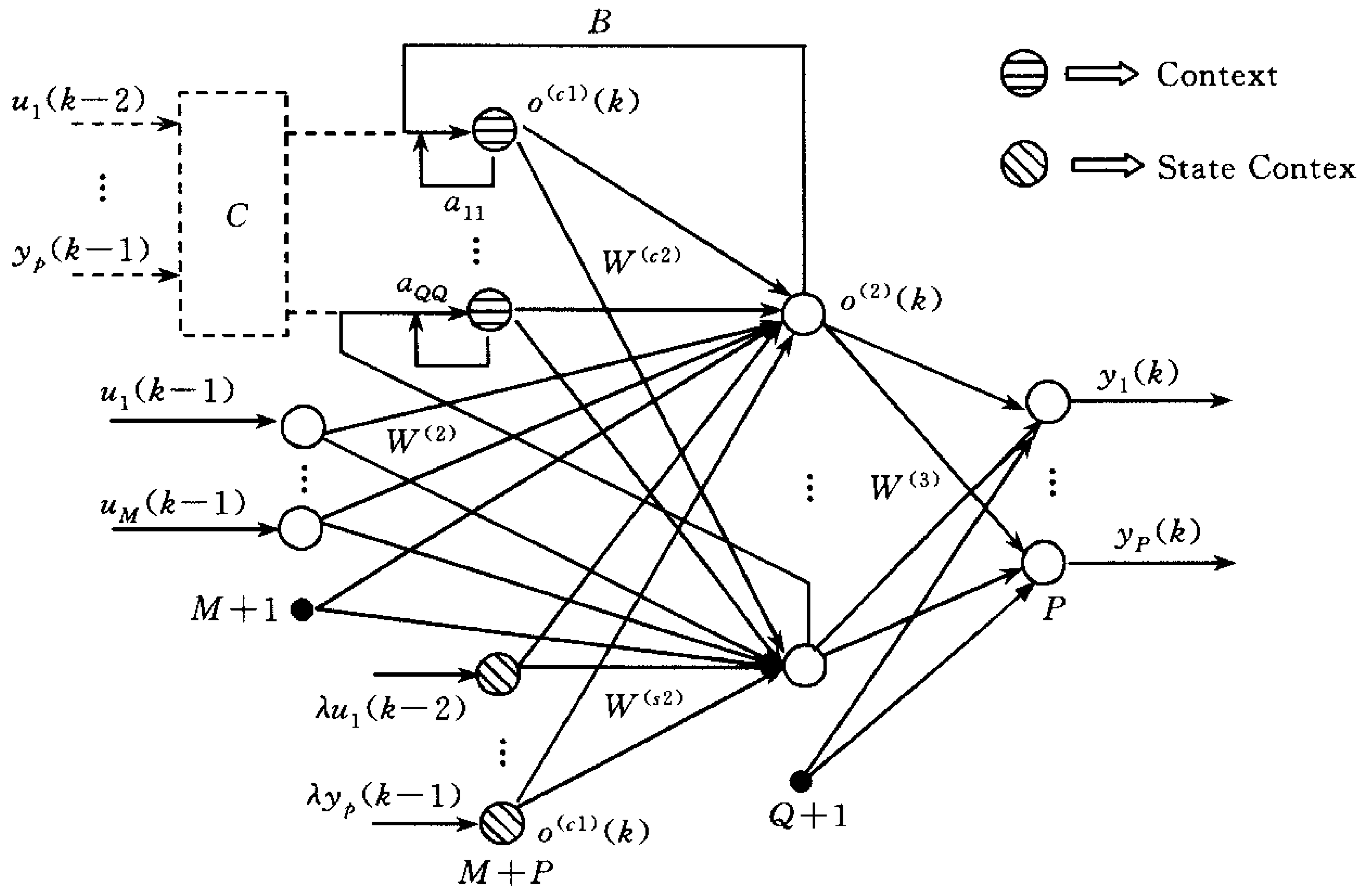

Fig. 1 Topological structure of SDIDRNN

The net input $net_i^{(2)}(k)$ of the ith hidden node is composed of three parts, i. e., knowledge transmitted from input layer, context layer and state context layer.

$$net_i^{(2)}(k) = \sum_{j=1}^{M+1} w_{ij}^{(2)} \cdot o_j^{(1)}(k) + \sum_{r=1}^{Q} w_{ir}^{(c2)} \cdot o_r^{(c1)}(k) + \sum_{t=1}^{M+P} w_{it}^{(s2)} \cdot o_t^{(s1)}(k) \tag{1}$$

where $W_{ij}^{(2)}$, $W_{ir}^{(c2)}$ and $W_{it}^{(s2)}$ are weights from input layer, context layer and state context layer to the hidden layer, respectively. $o_j^{(2)}$, $o_r^{(c1)}$, and $o_t^{(s1)}$ are output of input node j, context r and state context t, respectively. M, P and Q denote the numbers of nodes in the input layer, the hidden layer and the output layer, respectively. The most popular BP learning algorithm is chosen to train the neural network. The function of training signal is

$$J = \frac{1}{2}\sum_{n=1}^{S} E_n = \frac{1}{2}\sum_{n=1}^{S}\sum_{l=1}^{P}[y_{dl}(k) - y_l(k)]^2 \tag{2}$$

where S is the number of samples for training. The weights are tuned using gradient search method . In order to decrease the trend of oscillating and increase convergence capability, a momentum term is used in the generalized learing rule. The generalized equation used BP algorithm with momentum term is

$$\Delta w_{li}^{(n)}(k+1) = \eta\delta_l^{(n)}(k)\sigma'(net_l^{(n)}(k))o_i^{(n-1)}(k) + \alpha\Delta w_{li}^{(n)}(k) \tag{3}$$

where $\eta\in(0,1)$ is the update rate, $\alpha\in[0,1)$ is the momentum coefficient. $\delta_l^{(n)}(k)$ is the back propagation error signal for the lth node of the nth layer. $o_i^{(n-1)}(k)$ is the output of the ith node in the $(n-1)$th layer. $\sigma(\cdot)$ represents the activation functions.

$W_{ij}^{(2)}$ and $W_{li}^{(3)}$ can be tuned by the following standard BP algorithm, whereas more computaion should be done to adjust weigthts $W_{ir}^{(c2)}$ and $W_{it}^{(s2)}$. To tune weight $W_{ir}^{(c2)}$, one should get the partial derivative of J with respect to this variable. That is,

$$\frac{\partial J}{\partial w_{ir}^{(c2)}} = \sum_{p=1}^{P}(\delta_p^{(3)}w_{pi}^{(3)})\frac{\partial o_i^{(2)}(k)}{\partial w_{ir}^{(c2)}} \tag{4}$$

Note that

$$\frac{\partial o_i^{(2)}(k)}{\partial w_{ir}^{(c2)}} = f'(net_i^{(2)}(k))\frac{\partial(net_{i2}^{(2)}(k))}{\partial w_{ir}^{(c2)}} = f'(net_i^{(2)}(k))\left[o_r^{(c1)}(k) + \sum_{k=1}^{Q} w_{ik}^{(c2)}\frac{\partial o_k^{(c1)}(k)}{\partial w_{ik}^{(c2)}}\right] =$$
$$f'(net_i^{(2)}(k))\left[a_{rr}o_r^{(c1)}(k-1)+b_{rr}o_r^{(2)}(k-1)+\sum_{k=1}^{Q}w_{ik}^{(c2)}\left(a_{kk}\frac{\partial o_k^{(c1)}(k-1)}{\partial w_{ik}^{(c2)}}+b_{kk}\frac{\partial o_k^{(2)}(k-1)}{\partial w_{ik}^{(c2)}}\right)\right] \tag{5}$$

where a_{rr} and b_{rr} are the diagonal elements in self-feedback coefficient matrix A and feedback coefficient matrix B. $f(\cdot)$ is the activation function of neurons.

If the knowledge of input-output states is added to context nodes, namely, the feed-forward filter weight matrix is constant set as in CIDRNN, a certain term should be contained in (5) which includes the output $o_l^{(3)}(k)$ of the system. Hence, $\sum_{u=1}^{M+P}C_{ru}\frac{\partial o_u^{(s1)}}{\partial w_{ik}^{(c2)}}$ should be included in the last summing term. The learning rule becomes much complex because the output $o_k^{(3)}$ of the neural network in term $\sum_{u=1}^{M+P}C_{ru}\frac{\partial o_u^{(s1)}}{\partial w_{ik}^{(c2)}}$ is a function of variables $w_{li}^{(3)}$ and $w_{ir}^{(c2)}$. Therefore, this computation process is recurrent and it makes the learning algorithm very lengthy, complex and uneasy to understand. It will also cause the computation burden more heavy if the feed-forward filter weight matrix is variable. However, this adjusting process will be independent if it follows the strategy proposed in this paper which makes the hierarchy clearer and the learning rule easier to understand.

If dependency relationship between $o_r^{(c1)}(k)$ and $w_{ir}^{(c2)}$ can be neglected, (5) can be simplified as

$$\frac{\partial o_i^{(2)}(k)}{\partial w_{ir}^{(c2)}} = f'(net_i^{(2)}(k))[a_{rr}o_r^{(c1)}(k-1) + b_{rr}o_r^{(2)}(k-1)] \tag{6}$$

As to weight matrix $W^{(s2)}$

$$\frac{\partial J}{\partial w_{it}^{(s2)}} = \sum_{p=1}^{P}(\delta_p^{(3)}w_{pi}^{(3)})\frac{\partial o_i^{(2)}(k)}{\partial w_{it}^{(s2)}} \tag{7}$$

where
$$\frac{\partial o_i^{(2)}(k)}{\partial w_{it}^{(s2)}} = f'(net_i^{(2)}(k))\left[o_t^{(s1)}(k) + \sum_{k=1}^{M+P}w_{ik}^{(s2)}\frac{\partial o_k^{(s1)}(k)}{\partial w_{ik}^{(s2)}}\right] \tag{8}$$

because when $t=1,2,\cdots,M$, $o_t^{(s1)}(k)$ and $w_{it}^{(s2)}$ is independent of each other, (8) can be rewritten as follows.

$$\frac{\partial o_i^{(2)}(k)}{\partial w_{it}^{(s2)}} = f'(net_i^{(2)}(k))\left[o_t^{(s1)}(k) + \sum_{k=M+1}^{M+P} w_{ik}^{(s2)} \frac{\partial o_k^{(s1)}(k)}{\partial w_{ik}^{(s2)}}\right] = f'(net_i^{(2)}(k))\left[o_t^{(s1)}(k) + \sum_{k=1}^{P} w_{ik}^{(s2)} \frac{\partial y_k^{(3)}(k)}{\partial w_{ik}^{(s2)}}\right] \quad (9)$$

Without consideraton of the dependency between $o_t^{(s1)}(k)$ and $w_{it}^{(s2)}$, (9) is simplified as

$$\frac{\partial o_i^{(2)}(k)}{\partial w_{it}^{(s2)}} = f'(net_i^{(2)}(k))o_t^{(s1)}(k) \quad (10)$$

Thus, we have so far introduced the topological structure and learning algorithm of the state delay input dynamical recurrent neural network in details. The following contents are to prove the superiorities of SDIDRNN through a nonlinear dynamical system.

Consider a nonlinear system in discrete representation $y(k+1)=\frac{y(k)y(k-1)[y(k)+1.5]}{1+y(k)^2+y(k-1)^2}+u(k)$, and 400 samples are generated randomly in the closed interval $[-0.5, 0.5]$ as training set. The training destination is root-mean-square error (*r. m. s*). SDIDRNN is compared with the other three types of neural network, this is, BP, improved Elman and CIDRNN. In the process of learning, all parameters needed by these four types are set to be identical, namely weight matrix $W^{(1)}$ between the input and the output layer, $W^{(2)}$ between the hidden and the output layer, self-feedback coefficient matrix A and feedback coefficient matrix B. Assume that $A=0.7I$ and $B=I$, where I is n identity matrix. The feed-forward filter coefficient C is equal to 0.6 in CIDRNN. The learning algorithm applied to the four networks is error back propagation algorithm (EBP) plus a momentum term. Learning rate $\eta=0.1$, momentum coefficient $\alpha=0.1$, filter factor $\lambda=1.0$. Training cycle is 50. The curves of root-mean-square error are shown in Fig. 2. Note that 1 denotes BP, 2 denotes Elman, 3 denotes CIDRNN, 4 denotes SDIDRNN.

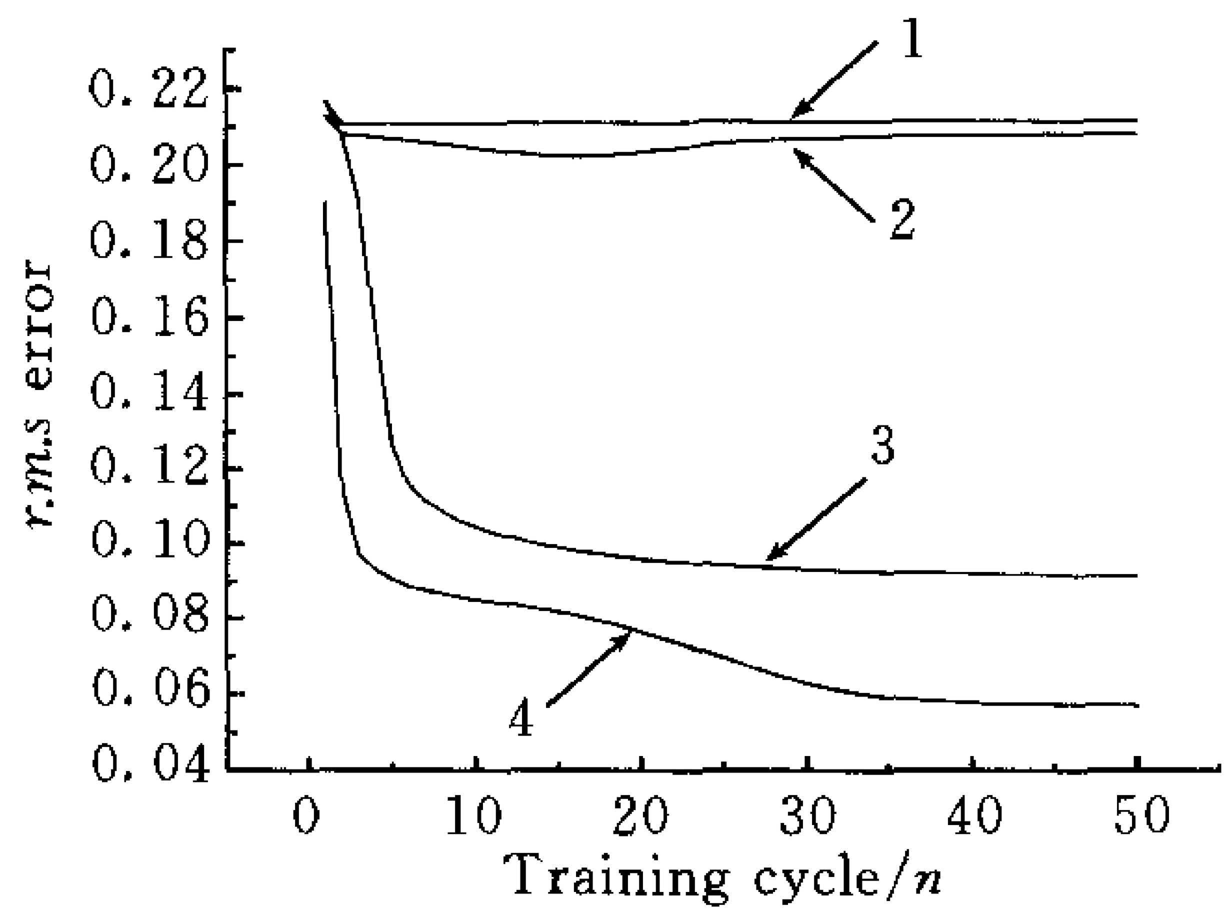

Fig. 2 Curves of *r. m. s* Error

By analyzing the four curves of r. m. s error in Fig. 2, one can find that on one hand, curves of BPNN and improved Elman are in little difference. Their convergence rates are far from satisfying compared with CIDRNN and SDIDRNN. On the other hand, CIDRNN and SDIDRNN converge relatively rapidly but the latter is superior to the former in convergence rate and steady state accuracy, thus SDIDRNN is of more competitive learning capability than the other three types of network mentioned above.

3 Online adaptive tracking identification of robot based on SDIDRNN

As mentioned above, the robot system can be expressed by two parts, i. e., the known and the unknown. In order to decrease the computation burden and improve the possibility of online utilization, we utilize the known part as the primary model, while the output of the neural network identifier, approximating the unknown part of the system, works as supplement for the primary model. This strategy is introduced in literature[4,5]. The neural network identifier keeps tuning weights until the error between the output of the identifier plus the mechanism model and actual output of the system is less than a preset error scale.

A two-link robot is used to study the problem of adaptive identification and to prove the efficiency of SDIDRNN in online adaptive tracking identification. The scheme is shown in Fig. 3 where τ_{sm} is the torque defined by the mechanism model, τ_{nn} is the output of SDIDRNN, representing the torque of the system's unknown part. τ is the actual joint torque equal to τ_{sm} plus τ_{nn}.

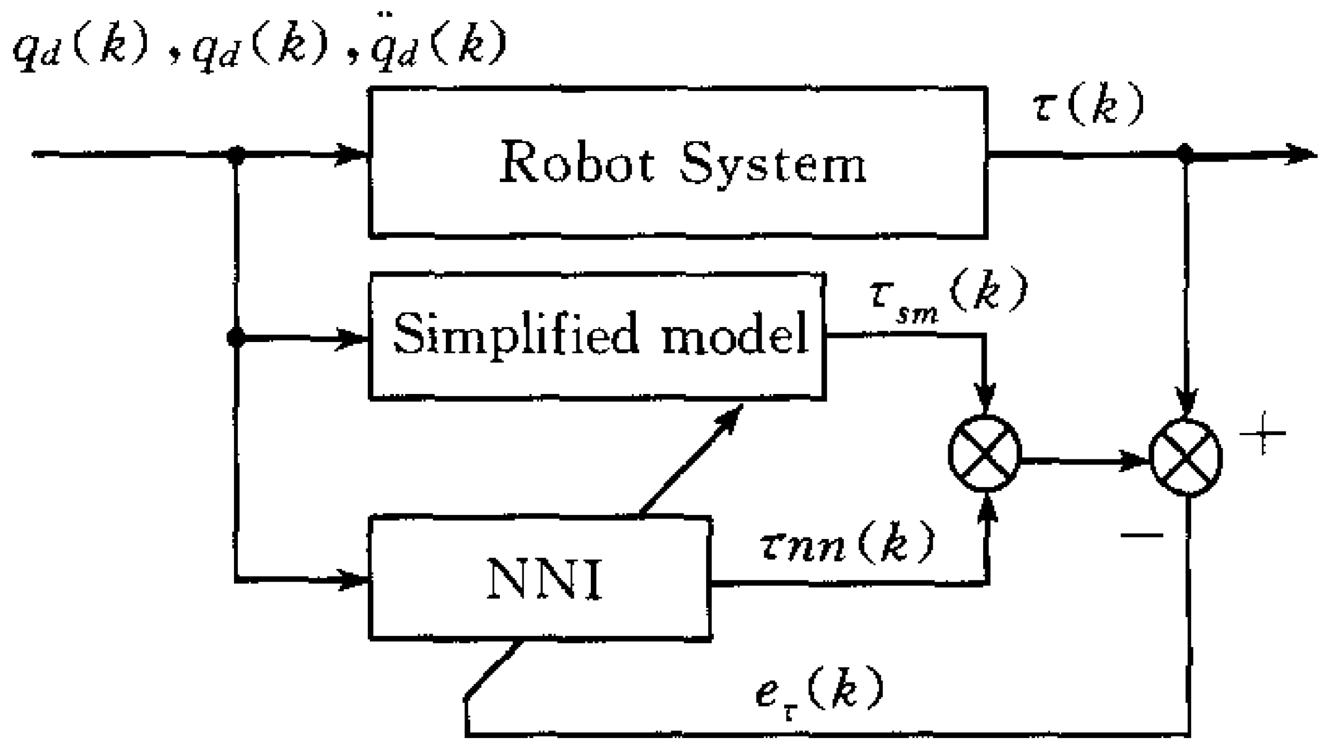

Fig. 3 Scheme for online adaptive identification

Trajectory settings are listed in table 1.

The simplified dynamic model of the two-joint robot is

$$\tau = M(q)\ddot{q} + H(q,\dot{q}) + G(q) \tag{11}$$

where $M \in R^{n\times n}$, $H \in R^{n\times 1}$, $G \in R^{n\times 1}$ denote the mass matrix, vector of Coriolis and centrifugal torques, and gravitational torques, respectively. $\tau \in R^{n\times 1}$ is the vector of joint actuator torques, the vectors $q, \dot{q}, \ddot{q}$ are the joint angle, joint velocity and joint acceleration, respectively.

Table 1 Desired trajectory

Trajectory	T	T_s	T_b
$x=0.2\cos\alpha+0.1$ $y=0.1\sin\alpha+0.6$	4s	0.5s	0.5s
Accelerating$[0, T_s]$	Uniform$[T_s, T-T_b]$	Decelerating$[T-T_b, T]$	
$\dot{\alpha} = \frac{(\cos(t/T_s \cdot \pi+\pi)+1)}{2}V_a$	$\dot{\alpha} = V_a$	$\dot{\alpha} = \frac{\cos\{[t-(T-T_b)]\pi/T_b\}+1}{2}V_a$	

In order to identify the unknown model of the robot system, one can rewrite (11) as

$$\tau = M(q)\ddot{q} + H(q,\dot{q}) + G(q) + F(q,\dot{q},\ddot{q}) \tag{12}$$

where

$$F(q,\dot{q},\ddot{q}) = M_e(\ddot{q}) + M_f(\dot{q}) + D(t) \tag{13}$$

In (13), $M_f(\dot{q}) = [0.6\mathrm{sgn}(\dot{q}_1)+1.2\dot{q}_1, 0.6\mathrm{sgn}(\dot{q}_2)+1.2\dot{q}_2]^T$ is the vector of friction torques, $D(t)=0.008\sin(2t)I$ is sine disturbance, $M_e(\ddot{q}) = (\ddot{q}_1+0.06\ddot{q}_2, 0.06\ddot{q}_2)^T$ is mass compensation vector for links. Suppose the system is unknown. What we want to do is to approximate the unformulated part of $F(q,\dot{q},\ddot{q})$ by training SDIDRNN. $l_1=0.4$m, $l_2=0.6$m, $m_1=6.0$kg, $m_2=4.0$kg, sample time is 4s, sample rate is 0.01s, and training cycle is 100.

From Fig. 4 to Fig. 7 are curves representing the identification results of the two configurations. One can find that using SDIDRNN and the identification strategy proposed in this paper, we could get satisfactory result. especially, the computational consumption is greatly deduced. During each identification cycle, the network tunes weights for a small number of times (only 100 for this sample) and outputs satisfactory result. The neural network after learning then can be used as the model representing the unknown dynamics and applied to robot control.

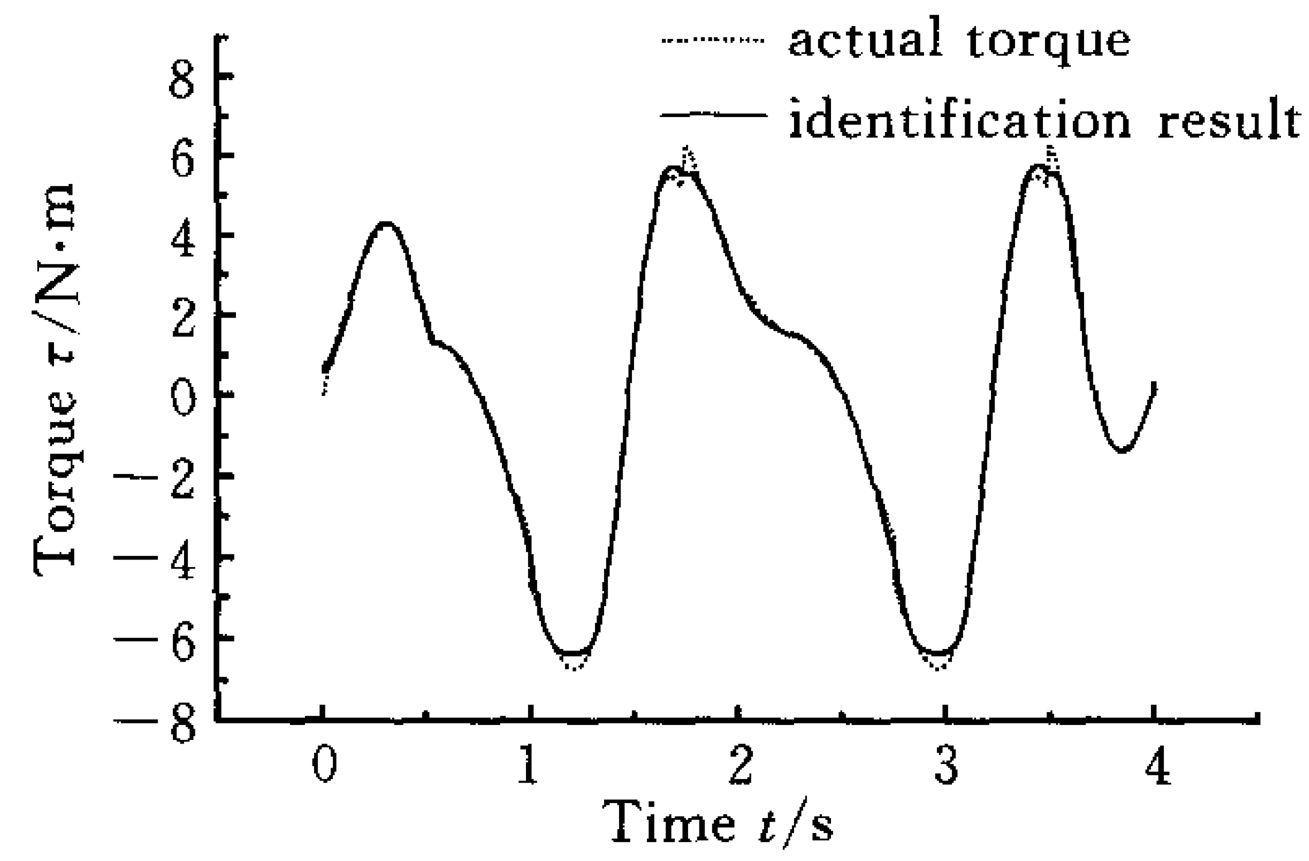

Fig. 4 Identification result of Joint 1(config. 1)

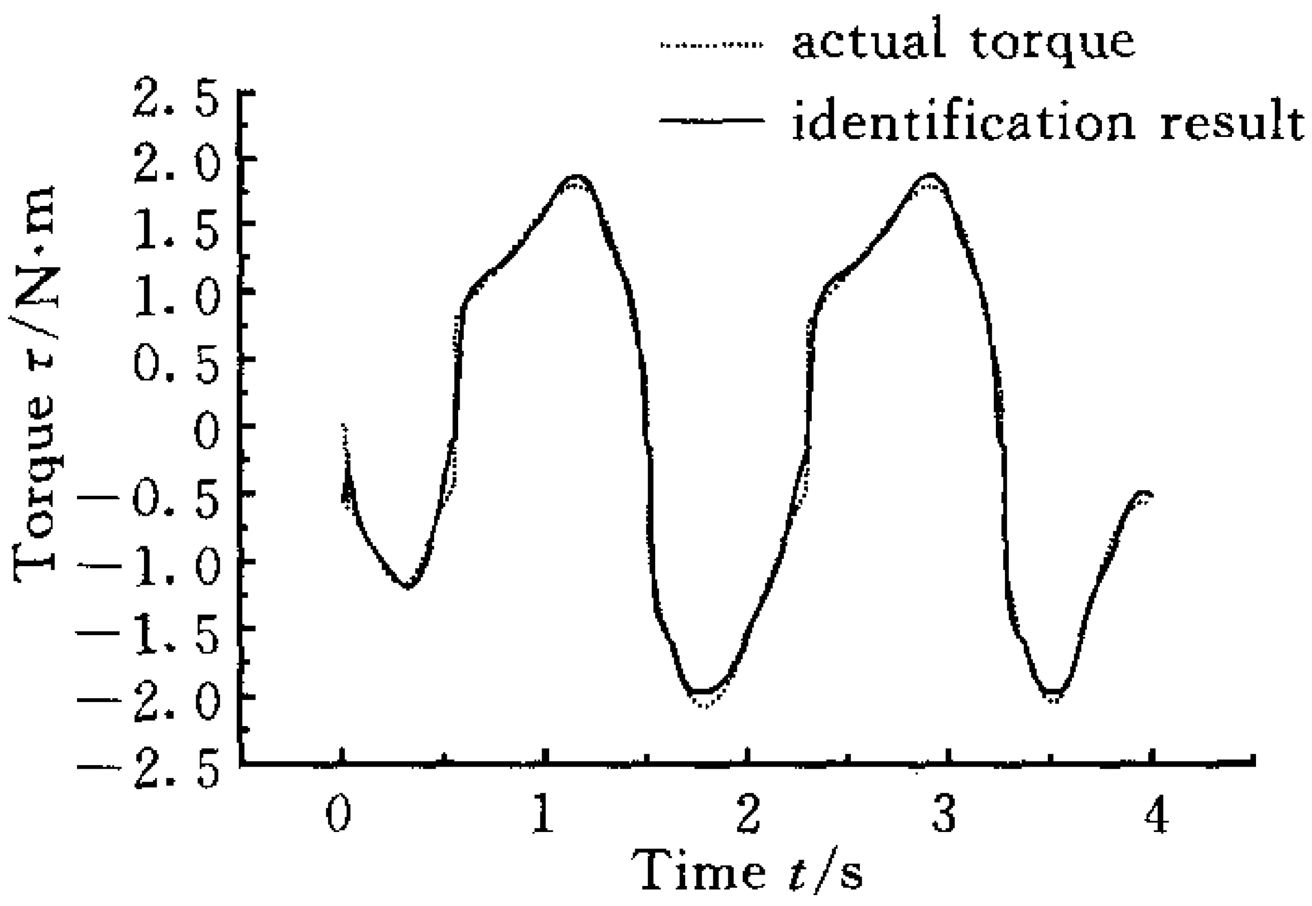

Fig. 5 Identification result of Joint 2(config. 1)

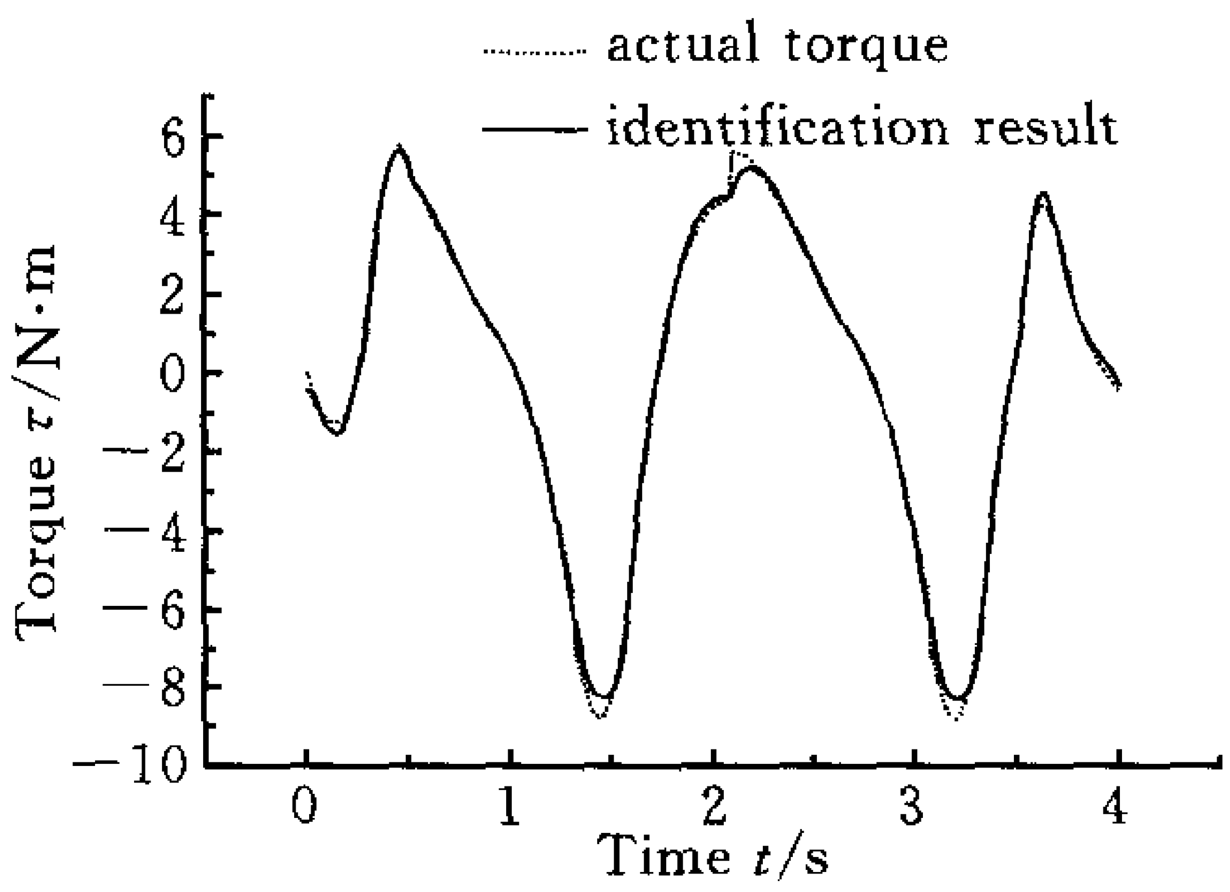

Fig. 6 Identification result of Joint 1(config. 2)

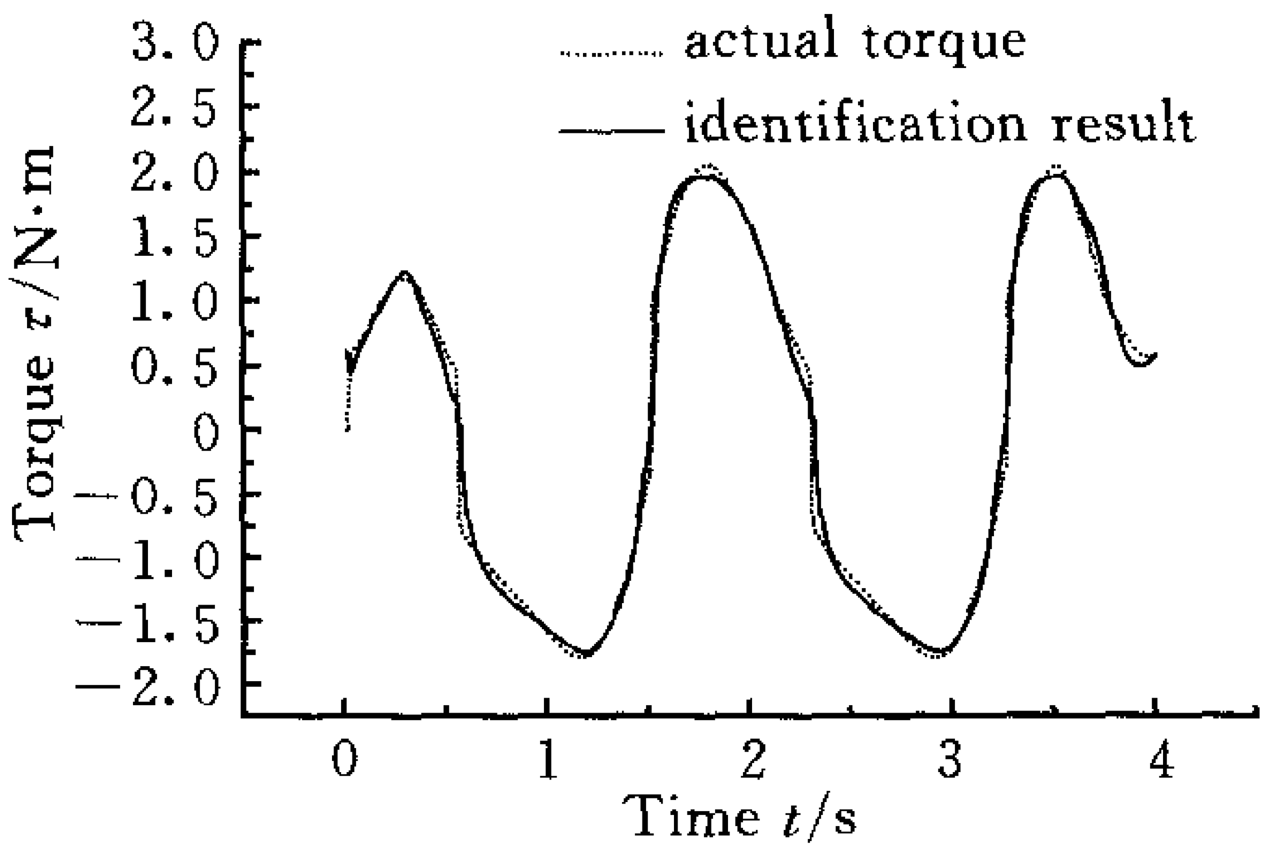

Fig. 7 Identification result of Joint 2(config. 2)

4 Summary

A new type of neural network SDIDRNN is proposed based on Elman network and the learning algorithm is derived. The learning ability that is crucial in online identification is improved. In this paper, by combining the formularized dynamics with the output of neural network identifier representing the unpredicted dynamics of the system, the computation of the network identifier is reduced, and the identification speed is up. This property is significant in online system identification and approximation of unpredicted dynamics.

References

1 Sun Zeng-Qi, Zhang Zai-Xing, Deng Zhi-Dong. Theory and Technology of Intelligent Control. Beijing: Tsinghua University Press, 1997(in Chinese)

2 He Yu-Bin, Li Xin-Zhong. Technology and Application of Neural Network Control. Beijing: Science Press, 2000(in Chinese)

3 Fujimoto Shinsaku, Ohsaka Kazumasa, Ono Toshiro. Robust trajectory tracking control of robot manipulators for high precision. *Transactions of the Japan Society of Mechanical Engineers*, Part C, 1994, **60**(574): 2071～2077

4 Johansen Tor A, Foss Bjarme A. Representing and learning unmodeled dynamics with neural network memories. In: Proceedings of the American Control Conference. Green Valley: American Automatic Control Council, 1992, **4**: 3037～3043

5 Kramer Mark A, Thompson Michael L, Bhagat Phiroz. Embedding theoretical models in neural networks. In: Proceedings of the American Control Conference, Green Valley: American Automatic Control Council, 1992, **1**: 475～479

JIANG Chun-Fu Ph. D. candidate in the College of Mechanical Engineering, Beijing University of Technology, received his master degree from Beijing University of Technology in 2000. His main research interests include the neural network's application in robot and relative study in experiment.

YU Yue-Qing Professor of the College of Mechanical Engineering, Beijing University of Technology. His main research interests include robot mechanism, flexible manipulator, and coordinated manipulators.

基于状态延迟动态递归神经网络的机器人动态自适应跟踪辨识

姜春福 余跃庆

(北京工业大学机电学院 北京 100022)

(E-mail: emailbox_cfj@163.com)

摘 要 对一种在 Elman 动态递归网络基础上发展而来的复合输入动态递归网络 (CIDRNN)作了改进，提出一种新的动态递归神经网络结构，称为状态延迟动态递归神经网络 (State Delay Input Dynamical Recurrent Neural Network). 具有这种新的拓扑结构和学习规则的动态递归网络，不仅明确了各权值矩阵的意义，而且使权值的训练过程更为简洁，意义更为明确. 仿真实验表明，这种结构的网络由于增加了网络输入输出的前一步信息，提高了收敛速度，增强了实时控制的可能性. 然后将该网络用于机器人未知非线性动力学的辨识中，使用辨识实际输出与机理模型输出之间的偏差，来识别机理模型或简化模型所丢失的信息，既利用了机器人现有的建模方法，又可以减小网络运算量，提高辨识速度. 仿真结果表明了这种改进的有效性.

关键词 动态递归网络，跟踪辨识，机器人

中图分类号 TP24