We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK
CITATION
INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Recurrent Neural Network-Based Adaptive Controller Design for Nonlinear Dynamical Systems

Hong Wei Ge and Guo Zhen Tan*

College of Computer Science and Technology, Dalian University Of Technology, Dalian, China

1. Introduction

The design goal of a control system is to influence the behavior of dynamic systems to achieve some pre-determinate objectives. A control system is usually designed on the premise that an accurate knowledge of a given object and environment cannot be obtained in advance. It usually requires suitable methods to address the problems related to uncertain and highly complicated dynamic system identification. As a matter of fact, system identification is an important branch of research in the automatic control domain. However, the majority of methods for system identification and parameters' adjustment are based on linear analysis: therefore it is difficult to extend them to complex non-linear systems. Normally, a large amount of approximations and simplifications have to be performed and, unavoidably, they have a negative impact on the desired accuracy. Fortunately the characteristics of the Artificial Neural Network (ANN) approach, namely non-linear transformation and support to highly parallel operation, provide effective techniques for system identification and control, especially for non-linear systems [1-9]. The ANN approach has a high potential for identification and control applications mainly because: (1) it can approximate the nonlinear input-output mapping of a dynamic system [10]; (2) it enables to model the complex systems' behavior and to achieve an accurate control through training, without a priori information about the structures or parameters of systems. Due to these characteristics, there has been a growing interest, in recent years, in the application of neural networks to dynamic system identification and control.

"Depth" and "resolution ratio" are the main characteristics to measure the dynamic memory performance of neural networks [11]. "Depth" denotes how far information can be memorized; "resolution ratio" denotes how much information in input sequences of neural networks can be retained. The memory of time-delay units is of lower depth and higher resolution ratio, while most recurrent neural networks, such as Elman neural networks, are higher depth and lower resolution ratio. The popular neural networks have much defect on dynamic memory performance. This chapter proposed a novel time-delay recurrent network model which has far more "depth" and "resolution ratio" in memory for

---

*Corresponding author

identifying and controlling dynamic systems. The proposed identification and control schemes are examined by the numerical experiments for identifying and controlling some typical nonlinear systems.

The rest of this chapter is organized as follows. Section 2 proposes a novel time-delay recurrent neural network (TDRNN) by introducing the time-delay and recurrent mechanism; moreover, a dynamic recurrent back propagation algorithm is developed according to the gradient descent method. Section 3 derives the optimal adaptive learning rates to guarantee the global convergence in the sense of discrete-type Lyapunov stability. Thereafter, the proposed identification and control schemes based on TDRNN models are examined by numerical experiments in Section 4. Finally, some conclusions are made in Section 5.

2. Time-delay recurrent neural network (TDRNN)

Figure 1 depicts the proposed time-delay recurrent neural network (TDRNN) by introducing the time-delay and recurrent mechanism. In the figure, Z^{-1} denotes a one-step time delay, the notation "▯" represents the memory neurons in the input layer with self-feedback gain γ $(0 \leq \gamma \leq 1)$, which improves the resolution ratio of the inputs.
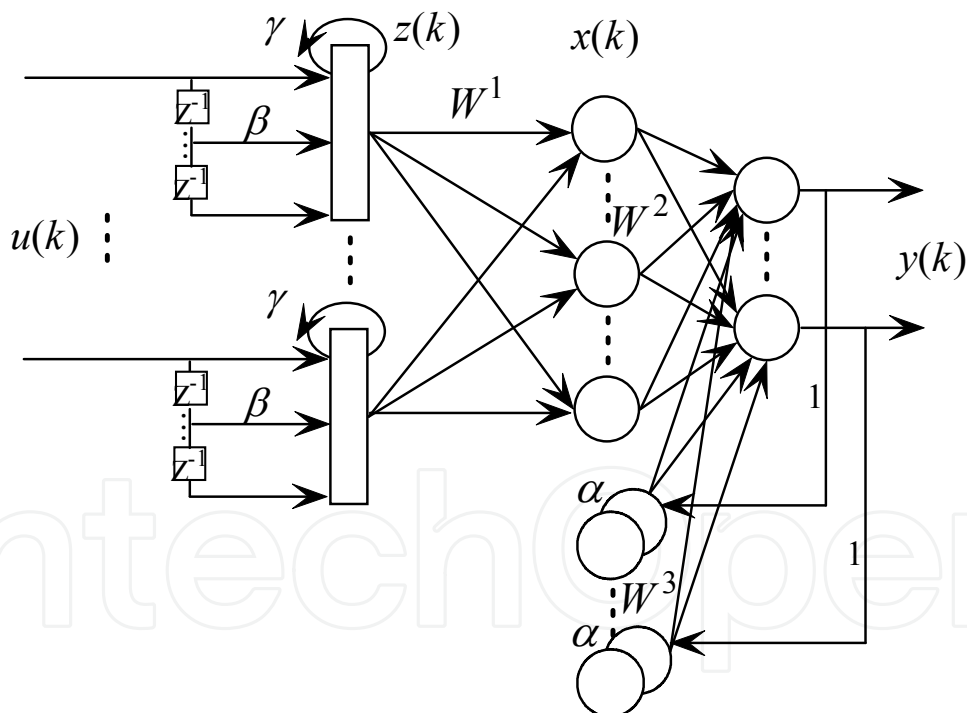

Fig. 1. Architecture of the modified Elman network

It is a type of recurrent neural network with different layers of neurons, namely: input nodes, hidden nodes, output nodes and, specific of the approach, context nodes. The input and output nodes interact with the outside environment, whereas the hidden and context nodes do not. The context nodes are used only to memorize previous activations of the output nodes. The feed-forward connections are modifiable, whereas the recurrent connections are fixed. More specifically, the proposed TDRNN possesses self-feedback links

with fixed coefficient α in the context nodes. Thus the output of the context nodes can be described by

$$y_{Cl}(k) = \alpha y_{Cl}(k-1) + y_l(k-1) \quad (l = 1, 2, \cdots, m). \tag{1}$$

where $y_{Cl}(k)$ and $y_l(k)$ are, respectively, the outputs of the lth context unit and the lth output unit and α ($0 \leq \alpha < 1$) is the self-feedback coefficient. If we assume that there are r nodes in the input layer, n nodes in the hidden layer, and m nodes in the output layer and context layers respectively, then the input u is an r dimensional vector, the output x of the hidden layer is n dimensional vector, the output y of the output layer and the output y_C of the context nodes are m dimensional vectors, and the weights W^1, W^2 and W^3 are $n \times r$, $m \times n$ and $m \times m$ dimensional matrices, respectively.

The mathematical model of the proposed TDRNN can be described as follows.

$$y(k) = g(W^2 x(k) + W^3 y_C(k)), \tag{2}$$

$$y_C(k) = \alpha y_C(k-1) + y(k-1), \tag{3}$$

$$x(k) = f(W^1 z(k)), \tag{4}$$

$$z(k) = u(k) + \beta \sum_{i=1}^{\tau} u(k-i) + \gamma z(k-1). \tag{5}$$

where $0 \leq \alpha, \beta, \gamma \leq 1$, $\beta + \gamma = 1$, $z(0) = 0$, and τ is the step number of time delay. $f(x)$ is often taken as the sigmoidal function

$$f(x) = \frac{1}{1 + e^{-x}}. \tag{6}$$

and $g(x)$ is often taken as a linear function, that is

$$y(k) = W^2 x(k) + W^3 y_C(k). \tag{7}$$

Taking expansion for $z(k-1)$, $z(k-2)$,..., $z(1)$ by using Eq.(5), then we have

$$z(k) = \sum_{i=0}^{\tau} u(k-i) + \sum_{i=1}^{k-\tau} \gamma^i u(k-\tau-i) - \gamma^k u(0). \tag{8}$$

From Eq.(8) it can be seen that the memory neurons in the input layer include all the previous input information and the context nodes memorize previous activations of the output nodes, so the proposed TDRNN model has far higher memory depth than the popular neural networks. Furthermore, the neurons in the input layer can memory accurately the inputs from time $k - \tau$ to time k, and this is quite different from the memory performance of popular recurrent neural networks. If the delay step τ is moderate large, the TDRNN possesses higher memory resolution ratio.

Let the kth desired output of the system be $y_d(k)$. We can then define the error as

$$E = \frac{1}{2}(y_d(k) - y(k))^T(y_d(k) - y(k)).$$ (9)

Differentiating E with respect to W^3, W^2 and W^1 respectively, according to the gradient descent method, we obtain the following equations

$$\Delta w_{il}^3 = \eta_3 \delta_i^0 y_{C,l}(k) \qquad (i = 1, 2, \cdots, m; l = 1, 2, \cdots, m),$$ (10)

$$\Delta w_{ij}^2 = \eta_2 \delta_i^0 (x_j(k) + w_{ii}^3 \frac{\partial y_{C,i}(k)}{\partial w_{ij}^2}) \quad (i = 1, 2, \cdots, m; j = 1, 2, \cdots, n),$$ (11)

$$\Delta w_{jq}^1 = \eta_1 \sum_{t=1}^{n} \delta_t^0 w_{tj}^2 f_j'(\cdot) z_q(k) \qquad (j = 1, 2, \cdots, n; q = 1, 2, \cdots, r).$$ (12)

which form the learning algorithm for the proposed TDRNN, where η_1, η_2 and η_3 are learning steps of W^1, W^2 and W^3, respectively, and

$$\delta_i^0 = (y_{d,i}(k) - y_i(k))g_i'(\cdot),$$ (13)

$$\frac{\partial y_{C,i}(k)}{\partial w_{ij}^2} = \alpha \frac{\partial y_{C,i}(k-1)}{\partial w_{ij}^2} + \frac{\partial y_i(k-1)}{\partial w_{ij}^2}.$$ (14)

If $g(x)$ is taken as a linear function, then $g_i'(\cdot) = 1$. Clearly, Eqs. (11) and (14) possess recurrent characteristics.

3. Convergence of proposed time-delay recurrent neural network

In Section 2, we have proposed a TDRNN model and derived its dynamic recurrent back propagation algorithm according to the gradient descent method. But the learning rates in the update rules have a direct effect on the stability of dynamic systems. More specifically, a large learning rate can make the modification of weights over large in each update step, and this will induce non-stability and non-convergence. On the other hand, a small learning rate will induce a lower learning efficiency. In order to train neural networks more efficiently, we propose three criterions of selecting proper learning rates for the dynamic recurrent back propagation algorithm based on the discrete-type Lyapunov stability analysis. The following theorems give sufficient conditions for the convergence of the proposed TDRNN when the functions $f(\cdot)$ and $g(\cdot)$ in Eqs. (4) and (2) are taken as sigmoidal function and linear function respectively.

Suppose that the modification of the weights of the TDRNN is determined by Eqs. (10-14). For the convergence of the TDRNN we have the following theorems.

Theorem 1. The stable convergence of the update rule (12) on W^1 is guaranteed if the learning rate $\eta_1(k)$ satisfies that

$$0 < \eta_1(k) < \frac{8}{nr \left| \max_k z_k(k) \right| \left| \max_{ij}(W_{ij}^2(k)) \right|} . \tag{15}$$

Proof. Define the Lyapunov energy function as follows.

$$E(k) = \frac{1}{2} \sum_{i=1}^m e_i^2(k) . \tag{16}$$

Where

$$e_i(k) = y_{d,i}(k) - y_i(k) . \tag{17}$$

And consequently, we can obtain the modification of the Lyapunov energy function

$$\Delta E(k) = E(k+1) - E(k) = \frac{1}{2} \sum_{i=1}^m \left[e_i^2(k+1) - e_i^2(k) \right] . \tag{18}$$

Then the error during the learning process can be expressed as

$$e_i(k+1) = e_i(k) + \sum_{j=1}^n \sum_{q=1}^r \frac{\partial e_i(k)}{\partial W_{jq}^1} \Delta W_{jq}^1 = e_i(k) - \sum_{j=1}^n \sum_{q=1}^r \frac{\partial y_i(k)}{\partial W_{jq}^1} \Delta W_{jq}^1 . \tag{19}$$

Furthermore, the modification of weights associated with the input and hidden layers is

$$\Delta W_{jq}^1(k) = \eta_1(k) e_i(k) \frac{\partial e_i(k)}{\partial W_{jq}^1} = -\eta_1(k) e_i(k) \frac{\partial y_i(k)}{\partial W_{jq}^1} . \tag{20}$$

Hence, from Eqs.(18-20) we obtain

$$\begin{aligned}
\Delta E(k) &= \frac{1}{2} \sum_{i=1}^m e_i^2(k) \left[\left(1 - \eta_1(k) \left[\frac{\partial y_i(k)}{\partial W^1} \right]^T \left[\frac{\partial y_i(k)}{\partial W^1} \right] \right)^2 - 1 \right] \\
&= \frac{1}{2} \sum_{i=1}^m e_i^2(k) \left[\left(1 - \eta_1(k) \left\| \frac{\partial y_i(k)}{\partial W^1} \right\|^2 \right)^2 - 1 \right] \\
&= -\sum_{i=1}^m e_i^2(k) \beta_i^1(k)
\end{aligned} \tag{21}$$

Where

$$\begin{aligned}
\beta_i^1(k) &= \frac{1}{2} \left[1 - \left(1 - \eta_1(k) \left\| \frac{\partial y_i(k)}{\partial W^1} \right\|^2 \right)^2 \right] \\
&= \frac{1}{2} \eta_1(k) \left\| \frac{\partial y_i(k)}{\partial W^1} \right\|^2 \left(2 - \eta_1(k) \left\| \frac{\partial y_i(k)}{\partial W^1} \right\|^2 \right)
\end{aligned} \tag{22}$$

W^1 represents an $n \times r$ dimensional vector and $\|\cdot\|$ denotes the Euclidean norm.

Notice that the activation function of the hidden neurons in the TDRNN is the sigmoidal type, we have $0 < f'(x) \leq 1/4$. Thus,

$$\left|\frac{\partial y_i(k)}{\partial W_{jq}^1}\right| = \left|W_{ij}^2(k)f_j'(\cdot)z_q(k)\right| \leq \frac{1}{4}\left|\max_q z_q(k)\right|\left|\max_{ij}(W_{ij}^2(k))\right| .$$

$$(i = 1,2,\cdots,m; j = 1,2,\cdots,n; q = 1,2,\cdots,r)$$

$$(23)$$

According to the definition of the Euclidean norm we have

$$\left\|\frac{\partial y(k)}{\partial \boldsymbol{W}^1}\right\| \leq \sqrt{\frac{nr}{4}}\left|\max_q z_q(k)\right|\left|\max_{ij}(W_{ij}^2(k))\right| . \tag{24}$$

Therefore, while $0 < \eta_1(k) < \dfrac{8}{nr\left|\max\limits_q z_q(k)\right|\left|\max\limits_{ij}(W_{ij}^2(k))\right|}$, we have $\beta_i^1(k) > 0$, then from

Eq.(21) we obtain $\Delta E(k) < 0$. According to the Lyapunov stability theory, this shows that the training error will converges to zero as $t \to \infty$. This completes the proof.

Theorem 2. The stable convergence of the update rule (11) on W^2 is guaranteed if the learning rate $\eta_2(k)$ satisfies that

$$0 < \eta_2(k) < \frac{2}{n}. \tag{25}$$

Proof. Similarly, the error during the learning process can be expressed as

$$e_i(k+1) = e_i(k) + \sum_{j=1}^n \frac{\partial e_i(k)}{\partial W_{ij}^2}\Delta W_{ij}^2 = e_i(k) - \sum_{j=1}^n \frac{\partial y_i(k)}{\partial W_{ij}^2}\Delta W_{ij}^2 . \tag{26}$$

Therefore,

$$\begin{aligned}
\Delta E(k) &= \frac{1}{2}\sum_{i=1}^m e_i^2(k)\left[\left(1 - \eta_2(k)\left[\frac{\partial y_i(k)}{\partial W_i^2}\right]^T\left[\frac{\partial y_i(k)}{\partial W_i^2}\right]\right)^2 - 1\right] \\
&= \frac{1}{2}\sum_{i=1}^m e_i^2(k)\left[\left(1 - \eta_2(k)\left\|\frac{\partial y_i(k)}{\partial W_i^2}\right\|^2\right)^2 - 1\right] \\
&= -\sum_{i=1}^m e_i^2(k)\beta_i^2(k)
\end{aligned} \tag{27}$$

Where

$$\beta_i^2(k) = \frac{1}{2}\left[1 - \left(1 - \eta_2(k)\left\|\frac{\partial y_i(k)}{\partial W_i^2}\right\|^2\right)^2\right].$$ (28)

Notice that the activation function of the hidden neurons in the TDRNN is the sigmoidal type, and neglect the dependence relation between $y_C(k)$ and the weights w_{ij}^2, we obtain

$$\frac{\partial E}{\partial w_{ij}^2} = -\delta_i^0 x_j(k) .$$ (29)

Hence,

$$\left|\frac{\partial y_i(k)}{\partial W_{ij}^2}\right| = \left|x_j(k)\right| < 1 \quad (i = 1,2,\cdots,m; j = 1,2,\cdots,n) .$$ (30)

According to the definition of the Euclidean norm we have

$$\left\|\frac{\partial y_i(k)}{\partial W_i^2}\right\| < \sqrt{n} .$$ (31)

Therefore, while $0 < \eta_2(k) < \dfrac{2}{n}$, we have $\beta_i^2(k) > 0$, then from Eq.(27) we obtain $\Delta E(k) < 0$. According to the Lyapunov stability theory, this shows that the training error will converges to zero as $t \to \infty$. This completes the proof.

Theorem 3. The stable convergence of the update rule (10) on W^3 is guaranteed if the learning rate $\eta_3(k)$ satisfies that

$$0 < \eta_3(k) < \frac{2}{m\left|\max_l(y_{C,l}(k))\right|^2} .$$ (32)

Proof. Similarly, as the above proof, we have

$$\begin{aligned}
\Delta E(k) &= \frac{1}{2}\sum_{i=1}^m e_i^2(k)\left[\left(1 - \eta_3(k)\left[\frac{\partial y_i(k)}{\partial W^3}\right]^T\left[\frac{\partial y_i(k)}{\partial W^3}\right]\right)^2 - 1\right] \\
&= \frac{1}{2}\sum_{i=1}^m e_i^2(k)\left[\left(1 - \eta_3(k)\left\|\frac{\partial y_i(k)}{\partial W^3}\right\|^2\right)^2 - 1\right] \\
&= -\sum_{i=1}^m e_i^2(k)\beta_i^3(k)
\end{aligned}$$ (33)

Where

$$\beta_i^3(k) = \frac{1}{2}\left[1 - \left(1 - \eta_3(k)\left\|\frac{\partial y_i(k)}{\partial W^3}\right\|^2\right)^2\right].$$ (34)

Furthermore, according to the learning algorithm we have

$$\left|\frac{\partial y_s(k)}{\partial W_{il}^3}\right| = \left|\delta_{is}y_{C,h}(k)\right| = \delta_{is}\left|y_{C,l}(k)\right| \le \delta_{is}\left|\max_l(y_{C,l}(k))\right|.$$ (35)

$$(i = 1,2,\cdots,m; s = 1,2,\cdots,m; l = 1,2,\cdots,m)$$

Where

$$\delta_{is} = \begin{cases} 1 & i = s \\ 0 & i \ne s \end{cases}.$$ (36)

According to the definition of the Euclidean norm we have

$$\left\|\frac{\partial y(k)}{\partial W^3}\right\| \le \sqrt{m}\left|\max_l(y_{C,l}(k))\right|.$$ (37)

Therefore, from Eq.(34), we have $\beta_i^3(k) > 0$, while $0 < \eta_3(k) < \dfrac{2}{m\left|\max\limits_l(y_{C,l}(k))\right|^2}$. Then from

Eq.(33) we obtain $\Delta E(k) < 0$. According to the Lyapunov stability theory, this shows that the training error will converges to zero as $t \to \infty$. This completes the proof.

4. Numerical results and discussion

The performance of the proposed time-delay recurrent neural network for identifying and controlling dynamic systems is examined by some typical test problems. We provide four examples to illustrate the effectiveness of the proposed model and algorithm.

4.1 Nonlinear time-varying system identification

We have carried out the identification for the following nonlinear time-varying system using the TDRNN model as an identifier.

$$y(k+1) = \frac{y(k)}{1 + 0.68\sin(0.0005\pi k)y^2(k)} + 0.78u^3(k) + v(k).$$ (38)

Where $v(k)$ is Gauss white noise with zero mean and constant variance 0.1. The input of system is taken as $u(k) = \sin(0.01\pi k)$.

To evaluate the performance of the proposed algorithm, the numerical results are compared with those obtained by using Elman neural network (ENN). The Elman network is a typical

recurrent network proposed by Elman [12]. Some parameters on the TDRNN in our experiments are taken as follows. The number of hidden nodes is taken as 6, the weights are initialized in the interval [-2, 2] randomly, besides, α, β and γ are set as 0.4, 0.6, 0.4 respectively. The number of hidden nodes in the ENN is also taken as 6.

Figure 2 shows the identification result, where the "Actual curve" is the real output curve of the dynamic system, represented by the solid line; the "Elman curve" is the output curve identified using the ENN model, and represented by the dash line; the "TDRNN curve" is the output curve identified by the proposed TDRNN model, and represented by the dash dot line. Figure 3 shows the identification error curves obtained with the TDRNN and ENN respectively, in which the error is the absolute value of the difference between identification result and the actual output. From the two figures it can be seen that the proposed method is superior to the ENN method. These results demonstrate the power and potential of the proposed TDRNN model for identifying nonlinear systems.
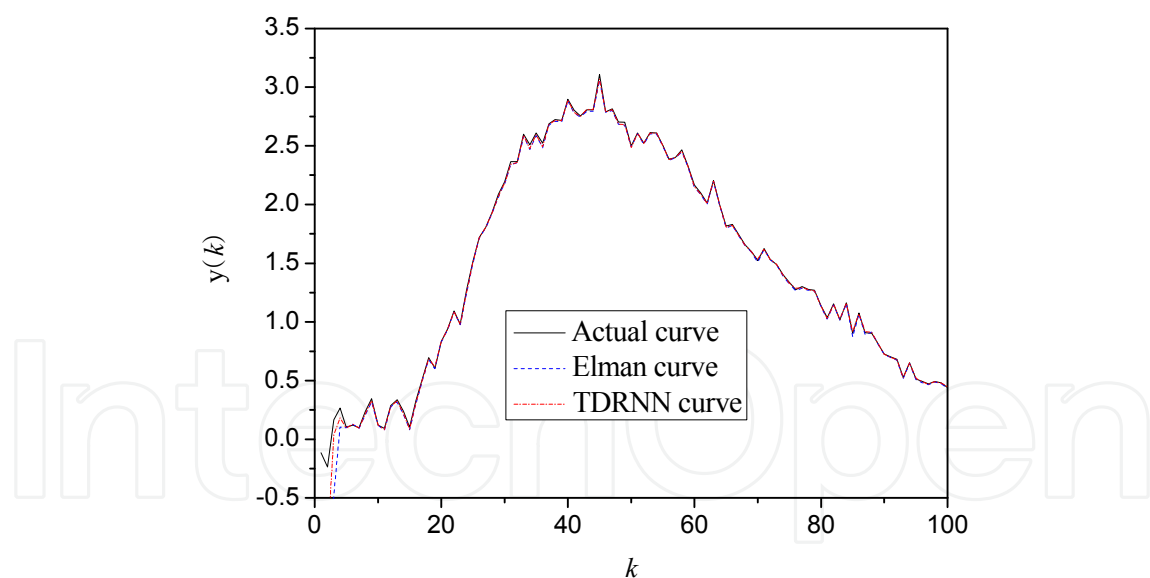
Fig. 2. Identification curves with different methods
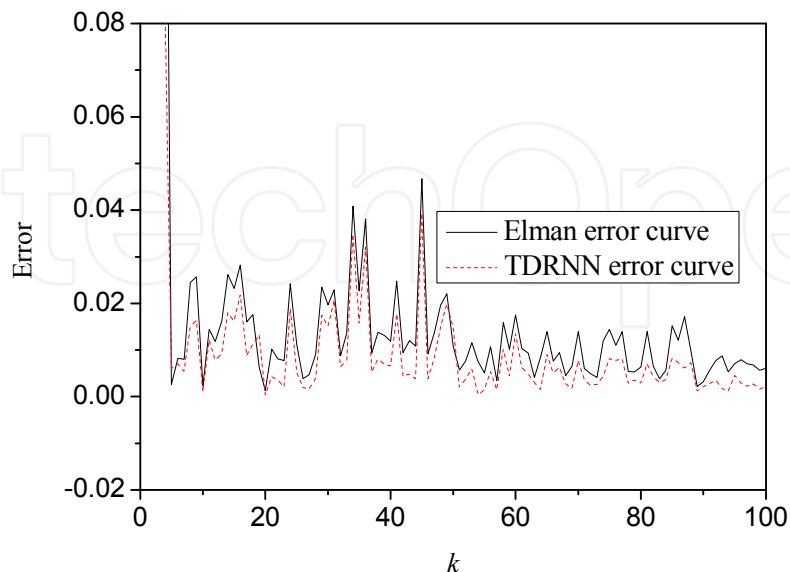
Fig. 3. Comparison of error curves obtained by different methods

4.2 Bilinear DGP system control

In this section, we control the following bilinear DGP system using the TDRNN model as a controller.

$$z(t) = 0.5 - 0.4z(t-1) + 0.4z(t-1)u(t-1) + u(t) . \tag{39}$$

The system output at an arbitrary time is influenced by all the past information. The control reference curves are respectively taken as:

1. Line type:

$$z(t) = 1.0 ; \tag{40}$$

2. Quadrate wave:

$$z(t) = \begin{cases} 0.0 & (2k \cdot 5 \le t < (2k+1) \cdot 5, (k = 0,1,2,\cdots)) \\ 1.0 & ((2k-1) \cdot 5 \le t < 2k \cdot 5, (k = 1,2,3,\cdots)) \end{cases} \tag{41}$$

The parameters on the TDRNN in the experiments are taken as follows. The number of hidden nodes is taken as 6, the weights are initialized in the interval [-2, 2] randomly, besides, α, β and γ are set as 0.3, 0.6, 0.4 respectively. Figures 4 and 5 show the control results. Figure 4 shows the control curve using the proposed TDRNN model when the control reference is taken as a line type. Figure 5 shows the control curve when the reference is taken as a quadrate wave type. From these results it can be seen that the proposed control model and algorithm possess a satisfactory control precision.
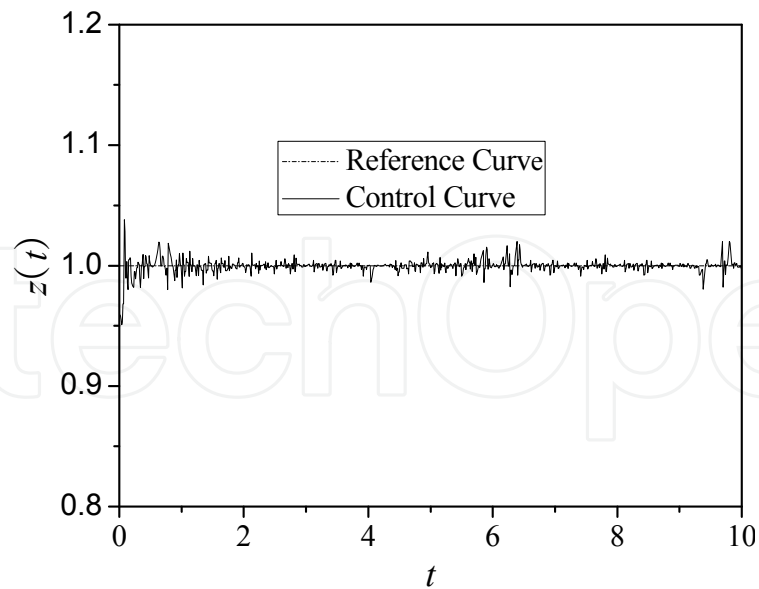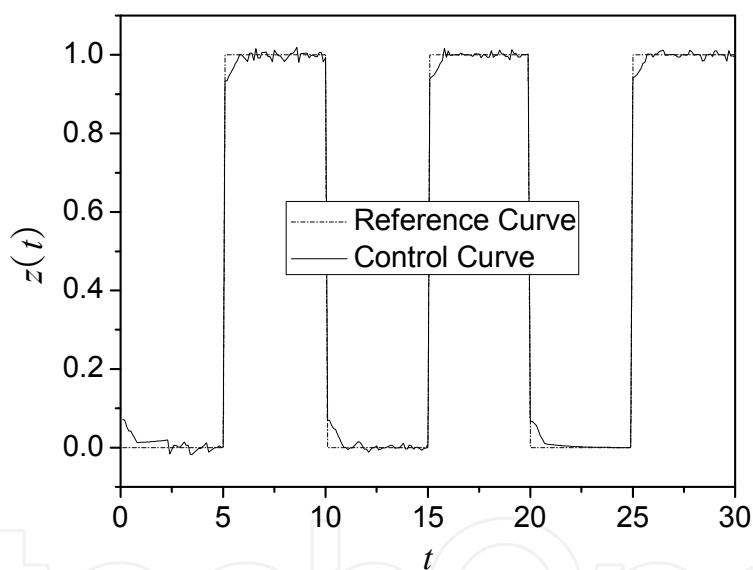
Fig. 4. Control curves with line type reference

Fig. 5. Control curves with quadrate wave reference

4.3 Inverted pendulum control

The inverted pendulum system is one of the classical examples used in many experiments dealing with classical as well as modern control, and it is often used to test the effectiveness of different controlling schemes [13-16]. So in this chapter, to examine the effectiveness of the proposed TDRNN model, we investigate the application of the TDRNN to the control of inverted pendulums.

The inverted pendulum system used here is shown in Fig.6, which is formed from a cart, a pendulum and a rail for defining position of cart. The Pendulum is hinged on the center of the top surface of the cart and can rotate around the pivot in the same vertical plane with the rail. The cart can move right or left on the rail freely.
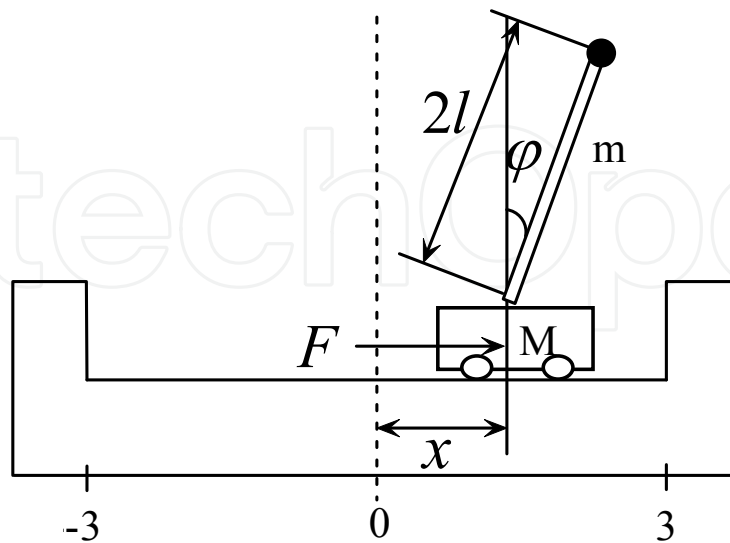
Fig. 6. Schematic diagram of inverted pendulum system

The dynamic equation of the inverted pendulum system can be expressed as the following two nonlinear differential equations.

$$\ddot{\varphi} = \frac{g\sin\varphi + \cos\varphi[-F - ml\dot{\varphi}^2\sin\varphi + \mu_c \operatorname{sgn}(\dot{x})\cdot(m+M)^{-1}] - \frac{\mu_p\dot{\varphi}}{ml}}{\frac{4}{3}l - \frac{m\cos^2\varphi}{m+M}\cdot l} \ , \tag{42}$$

$$\ddot{x} = \frac{F + ml(\dot{\varphi}^2\sin\varphi - \ddot{\varphi}\cos\varphi) - \mu_c \operatorname{sgn}(\dot{x})}{m+M} \ . \tag{43}$$

Where the parameters, M and m are respectively the mass of the cart and the mass of the pendulum in unit (kg), $g = 9.81\,m/s^2$ is the gravity acceleration, l is the half length of the pendulum in unit (m), F is the control force in the unit (N) applied horizontally to the cart, u_c is the friction coefficient between the cart and the rail, u_p is the friction coefficient between the pendulum pole and the cart. The variables φ, $\dot{\varphi}$, $\ddot{\varphi}$ represent the angle between the pendulum and upright position, the angular velocity and the angular acceleration of the pendulum, respectively. Moreover, given that clockwise direction is positive. The variables x, $\dot{x}$, $\ddot{x}$ denote the displacement of the cart from the rail origin, its velocity, its acceleration, and right direction is positive.

We use the variables φ and x to control inverted pendulum system. The control goal is to make φ approach to zero by adjusting F, with the constraint condition that x is in a given interval. The control block diagram of the inverted pendulum system is shown in Figure 7. The TDRNN controller adopts variables φ and x as two input items.
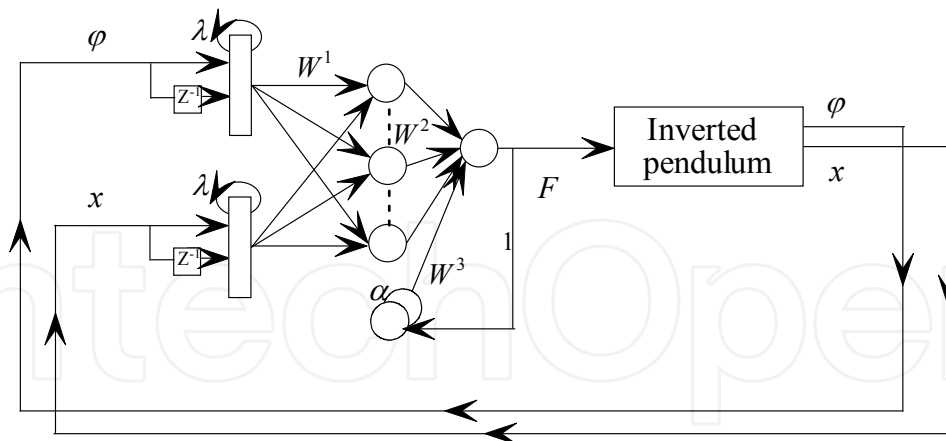
Fig. 7. Control block diagram of inverted pendulum system

In the numerical experiments, the motion of the inverted pendulum system is simulated by numerical integral. The parameter setting is listed in the Table 1.

parameter	g	M	m	l	μ_c	μ_p	φ	$\dot{\varphi}$	x	$\dot{x}$
value	9.81	1.0	0.1	0.6	0.002	0.00002	5°	0	0	0

Table 1. Parameter Setting of Inverted Pendulum

Besides, the number of hidden nodes is taken as 6, the weights are initialized in the interval [-3, 3] randomly, the parameters α, β and γ on the TDRNN are set as 0.3, 0.6, 0.4 respectively. The control goals are to control the absolute value of φ within 10° and make it approximate to zero as closely as possible, with the constraint condition of the absolute value of x within 3.0m.
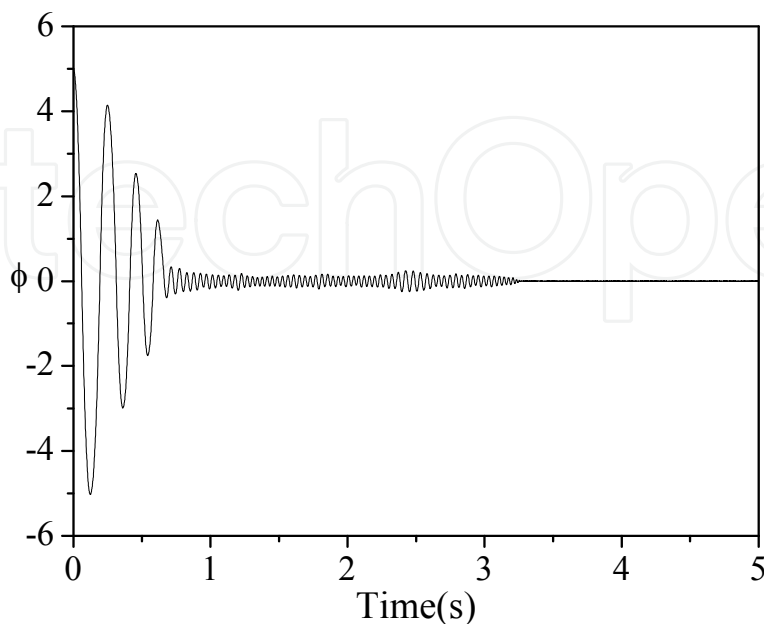
Fig. 8. Control curve of the angle φ
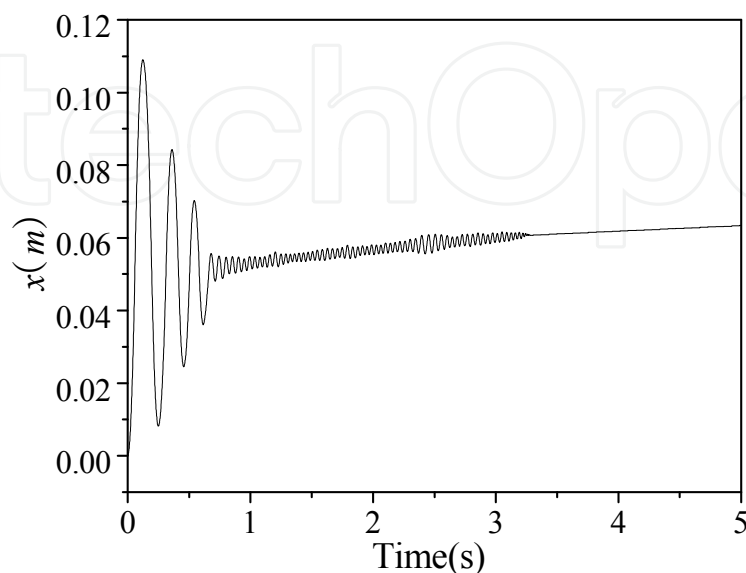
Fig. 9. Control curve of the displacement x

The control results are shown in Figures 8 and 9, and the sampling interval is taken as $T = 1ms$. Figures 8 and 9 respectively show the control curve of the angle φ and the control curve of the displacement x. From Figure 8, it can be seen that the fluctuation degree of φ is large at the initial stage, as time goes on, the fluctuation degree becomes smaller and smaller, and it almost reduces to zero after 3 seconds. Figure 9 shows that the change trend of x is similar to that of φ except that it has a small slope. These results demonstrate the proposed control scheme based on the TDRNN can effectively perform the control for inverted pendulum system.

4.4 Ultrasonic motor control

In this section, a dynamic system of the ultrasonic motor (USM) is considered as an example of a highly nonlinear system. The simulation and control of the USM are important problems in the applications of the USM. According to the conventional control theory, an accurate mathematical model should be set up. But the USM has strongly nonlinear speed characteristics that vary with the driving conditions and its operational characteristics depend on many factors. Therefore, it is difficult to perform effective control to the USM using traditional methods based on mathematical models of systems. Our numerical experiments are performed using the model of TDRNN for the speed control of a longitudinal oscillation USM [17] shown in Figure 10.
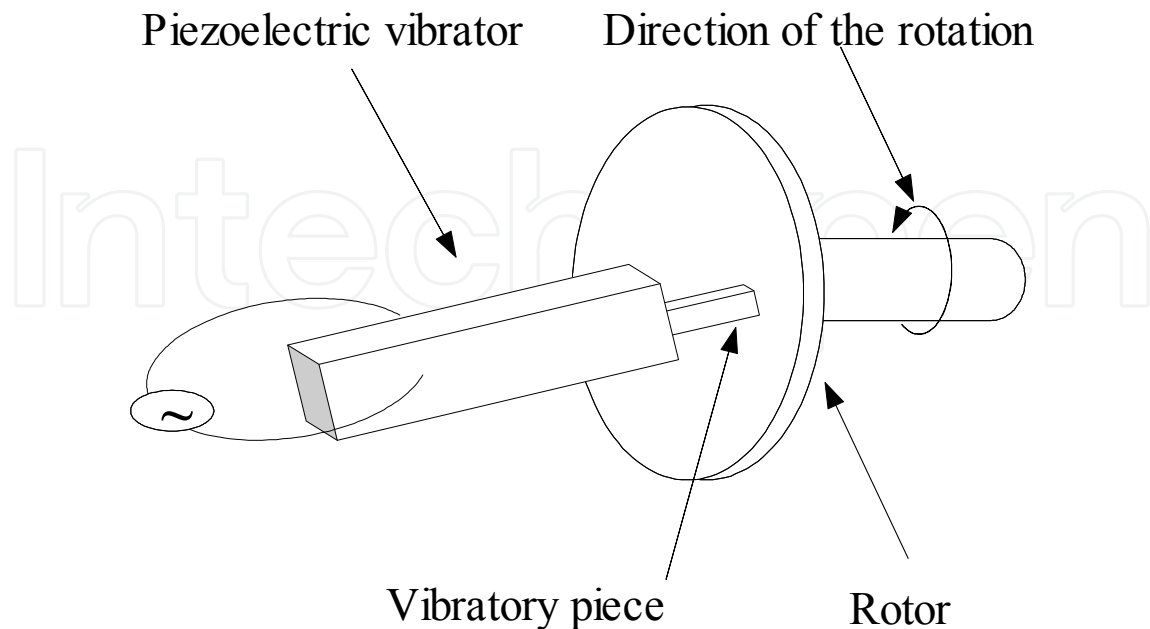
Fig. 10. Schematic diagram of the motor

Some parameters on the USM model are taken as: driving frequency $27.8\ kHZ$, amplitude of driving voltage $300\ V$, allowed output moment $2.5\ kg \cdot cm$, rotation speed $3.8\ m/s$. Besides, the number of hidden nodes of the TDRNN is taken as 5, the weights are initialized in the interval [-3, 3] randomly, the parameters α, β and γ on the TDRNN are taken as 0.4, 0.6, 0.4 respectively. The input of the TDRNN is the system control error in the last time, and the output of the TDRNN, namely the control parameter of the USM is taken as the frequency of the driving voltage.

Figure 11 shows the speed control curves of the USM using the three different control strategies when the control speed is taken as $3.6\ m/s$. In the figure the dotted line a represents the speed control curve based on the method presented by Senjyu et al.[18], the solid line b represents the speed control curve using the method presented by Shi et al.[19] and the solid line c represents the speed curve using the method proposed in this paper. Simulation results show the stable speed control curves and the fluctuation amplitudes obtained by using the three methods. The fluctuation degree is defined as

$$\zeta = (V_{\max} - V_{\min}) / V_{ave} \times 100\% \tag{44}$$

where $V_{\max}$, $V_{\min}$ and V_{ave} represent the maximum, minimum and average values of the speeds. From Figure 11 it can be seen that the fluctuation degrees when using the methods proposed by Senjyu and Shi are 5.7% and 1.9% respectively, whereas, it is just 1.1% when using the method in this paper. Figure 12 shows the speed control curves of the reference speeds vary as step types. From the figures it can be seen that this method possesses good control precision.
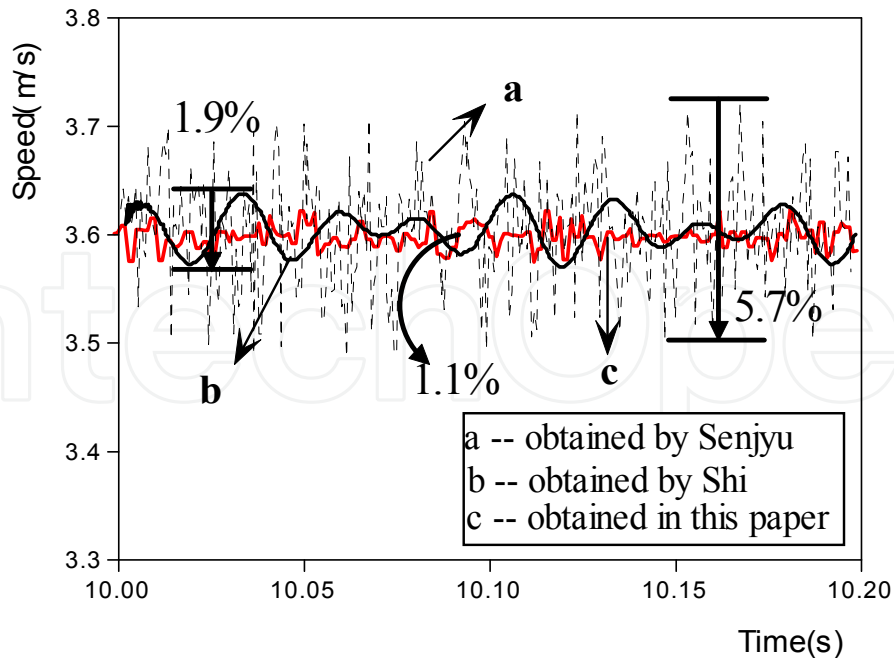
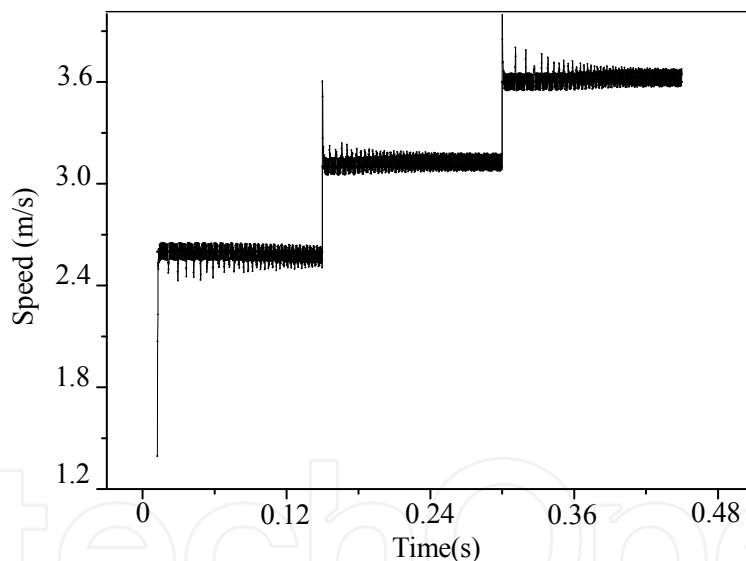Fig. 11. Comparison of speed control curves using different schemes

Fig. 12. Speed control curve for step type

5. Conclusions

This chapter proposes a time-delay recurrent neural network (TDRNN) with better performance in memory than popular neural networks by employing the time-delay and recurrent mechanism. Subsequently, the dynamic recurrent back propagation algorithm for the TDRNN is developed according to the gradient descent method. Furthermore, to train neural networks more efficiently, we propose three criterions of selecting proper learning rates for the dynamic recurrent back propagation algorithm based on the discrete-type Lyapunov stability analysis. Besides, based on the TDRNN model, we have described, analyzed and discussed an identifier and an adaptive controller designed to identify and

control nonlinear systems. Our numerical experiments show that the TDRNN has good effectiveness in the identification and control for nonlinear systems. It indicates that the methods described in this chapter can provide effective approaches for nonlinear dynamic systems identification and control.

6. Acknowledgment

7. References

[1] M. Han, J.C. Fan and J. Wang, A Dynamic Feed-forward Neural Network Based on Gaussian Particle Swarm Optimization and its Application for Predictive Control, IEEE Transactions on Neural Networks, 22 (2011) 1457-1468.

[2] G. Puscasu and B. Codres, Nonlinear System Identification and Control Based on Modular Neural Networks, International Journal of Neural Systems, 21(2011), 319-334.

[3] L.Li, G. Song, and J.Ou, Nonlinear Structural Vibration Suppression Using Dynamic Neural Network Observer and Adaptive Fuzzy Sliding Mode Control. Journal of Vibration and Control, 16 (2010), 1503-1526.

[4] T. Hayakawa, W.M. Haddad, J.M. Bailey and N. Hovakimyan, Passivity-based Neural Network Adaptive Output Feedback Control for Nonlinear Nonnegative Dynamical Systems, IEEE Transactions on Neural Networks, 16 (2005) 387-398.

[5] M. Sunar, A.M.A. Gurain and M. Mohandes, Substructural Neural Network Controller, Computers & Structures, 78 (2000) 575-581.

[6] D. Wang and J. Huang, Neural Network-based Adaptive Dynamic Surface Control for a Class of Uncertain Nonlinear Systems in Strict-feedback Form, IEEE Transactions on Neural Networks, 16 (2005) 195-202.

[7] Y.M. Li, Y.G. Liu and X.P. Liu, Active Vibration Control of a Modular Robot Combining a Back-propagation Neural Network with a Genetic Algorithm, Journal of Vibration and Control, 11 (2005) 3-17.

[8] J.C. Patra and A.C. Kot, Nonlinear Dynamic System Identification Using Chebyshev Functional Link Artificial Neural Networks, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 32 (2002) 505-511.

[9] R.J. Wai, Hybrid Fuzzy Neural-network Control for Nonlinear Motor-toggle Servomechanism, IEEE Transactions on Control Systems Technology, 10 (2002) 519-532.

[10] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signals and System, 2 (1989) 303–314.

[11] S. Haykin, Neural Networks: A Comprehensive Foundation (Englewood Cliffs, NJ: Prentice Hall, 1999).

[12] J.L. Elman, Finding Structure in Time, Cognitive Science, 14 (1990) 179-211.

[13] C.H. Chiu, Y.F. Peng, and Y.W. Lin, Intelligent backstepping control for wheeled inverted pendulum, Expert Systems With Applications, 38 (2011) 3364-3371.

[14] M.I. El-Hawwary, A.L. Elshafei and H.M. Emara, Adaptive Fuzzy Control of the Inverted Pendulum Problem, IEEE Transactions on Control Systems Technology, 14 (2006) 1135-1144.

[15] P.J. Gawthrop and L.P. Wang, Intermittent Predictive Control of An Inverted Pendulum, Control Engineering Practice, 14 (2006) 1347-1356.

[16] R.J. Wai and L.J. Chang, Stabilizing and Tracking Control of Nonlinear Dual-axis Inverted-pendulum System Using Fuzzy Neural Network, IEEE Transactions on Fuzzy systems, 14 (2006) 145-168.

[17] X. Xu, Y.C. Liang, H.P. Lee, W.Z. Lin, S.P. Lim and K.H. Lee, Mechanical modeling of a longitudinal oscillation ultrasonic motor and temperature effect analysis, Smart Materials and Structures, 12 (2003) 514-523.

[18] T. Senjyu, H. Miyazato, S. Yokoda, and K. Uezato, Speed control of ultrasonic motors using neural network, IEEE Transactions on Power Electronics, 13 (1998) 381-387.

[19] X.H. Shi, Y.C. Liang, H.P. Lee, W.Z. Lin, X.Xu and S.P. Lim, Improved Elman networks and applications for controlling ultrasonic motors, Applied Artificial Intelligence 18 (2004) 603-629.
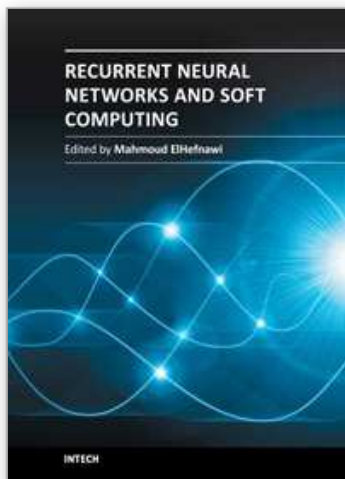
Recurrent Neural Networks and Soft Computing

Edited by Dr. Mahmoud ElHefnawi

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Hong Wei Ge and Guo Zhen Tan (2012). Recurrent Neural Network-Based Adaptive Controller Design For Nonlinear Dynamical Systems, Recurrent Neural Networks and Soft Computing, Dr. Mahmoud ElHefnawi (Ed.), ISBN: 978-953-51-0409-4, InTech, Available from: http://www.intechopen.com/books/recurrent-neural-networks-and-soft-computing/recurrent-neural-network-based-adaptive-controller-design-for-nonlinear-systems

INTECH
open science | open minds