

ORIGINAL RESEARCH

Adaptive control for uncertain discrete-time systems with unknown disturbance based on RNN

Huimin Cui^{*1,2}, Jin Guo¹, Jianxin Feng³, Tingfeng Wang¹

¹ State Key Laboratory of Laser Interaction with Matter, Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, China

² University of Chinese Academy of Sciences, Beijing, China

³ Nanjing University of Aeronautics and Astronautics, Nanjing, China

Received: March 13, 2016

Accepted: May 12, 2016

Online Published: July 25, 2016

DOI: 10.5430/air.v5n2p102

URL: <http://dx.doi.org/10.5430/air.v5n2p102>

ABSTRACT

A new robust adaptive control algorithm is developed for a class of uncertain discrete-time SISO systems. Different from the existing investigated systems, the concerned discrete system here is with both uncertain smooth nonlinear functions and unknown disturbance. On the basis of the idea of neural network (NN) approximation, a novel recurrent neural network (RNN) is first proposed and used to approximate a backstepping control law following the transformation of the original system into a predictor form. According to Lyapunov stability theorem, a new on-line tuning law for parameters of RNN is obtained. Meanwhile, in order to achieve satisfying robust tracking performance, a novel controller is constructed by virtue of the approximation error of RNN. It has been proved that all the concerned signals are uniformly ultimately bounded. In addition, a very small tracking error can be obtained through appropriate selection of control parameters. Finally, we give a simulation example to demonstrate the validness of the newly proposed control algorithm for the investigated systems.

Key Words: Adaptive control, Backstepping control, Discrete-time nonlinear systems, Recurrent neural networks, External disturbance

1. INTRODUCTION

During the past few decades, neural networks (NNs) have obtained widespread attentions particularly in the area of identification and control for dynamic systems by virtue of their excellent universal approximation ability.^[1-12] For example, Sarma *et al.*^[4] propose an approach combined with ANN to make out primal phonemes of Assamese language. RBF neural network is employed as an accessorial method to weaken the impact of nonlinearity and uncertainty on the nonlinear system.^[5] Generally, according to structures, neu-

ral network (NN) can be categorized into two types, *i.e.*, feed-forward neural network (FNN)^[1,2,10,13,14] and recurrent neural network (RNN).^[6,9,11,15-18,20] We know that FNN can only represent static mappings and its approximation performance is easily influenced by training data because the scheme of weights update does not depend on internal network information. However, RNN can memorize the past knowledge in virtue of its delay feedback loops. Thus, when the inputs are time-varying RNN can also deal with them by use of its superior temporal operation. In practical appli-

* **Correspondence:** Huimin Cui; Email: cui_huimin2012@126.com; Address: State Key Laboratory of Laser Interaction with Matter, Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033; China and University of Chinese Academy of Sciences, Beijing 100049, China.

cations, that characteristic makes RNN change accordingly when control conditions change suddenly. Consequently, RNN can achieve better control property compared with FNN when the system contains un-modeled dynamics. That is also the main reason why RNN has been widely used in the control field. For example, Miao *et al.*^[9] propose a recurrent neural network control method in the case of having no information about the system dynamics. In addition, Lin *et al.*^[11] combine a robust adaptive backstepping control (RABC) algorithm and recurrent wavelet neural network to control the target system. From simulation results in those papers it can be concluded that control method combined with recurrent neural networks can achieve good control performance.

Through lots of literature investigation, recently we have found that, among various types of NN, radial basis function neural network owns the capability of approximating any function to arbitrarily small error range.^[24,25] However, so far, there has no discrete RNN found. In the study, for the purpose of enhancing the mapping capability of RBFNN, we add delay feedback links to the original NN forming the recurrent RBFNN. Then, we use the recurrent RBFNN as the main controller for the discrete nonlinear systems by use of the proposed neural network's dynamic characteristic and relative simple structure. Compared with the previous research, this study proposes a more general RNN for use to deal with nonlinearity and uncertainties for a more general control system.

We all know that compared with the continuous-time description the discrete-time description are more veritable when depicting practical problems in systems. From the investigated papers, we conclude that adaptive NN control is well developed for nonlinear continuous-time systems.^[9,10,16-20,27,28] However, the difference of Lyapunov function in discrete-time^[13] has no the linearity property of the derivative of a Lyapunov function in continuous-time. Consequently, adaptive NN control suitable for continuous-time systems may not be applied to discrete-time systems directly. Through years of progress, many researchers have been devoted to the research of adaptive NN control for discrete-time systems. So far, the research in this aspect has advanced significantly.^[13,14,31-34] Generally the adaptive NN control scheme for nonlinear uncertain discrete-time systems is on the basis of the backstepping technique and Lyapunov stability theory.^[20-23] For example, in Ref.,^[26] a NN control algorithm combined with the backstepping technique is provided for a class of strict-feedback systems. The controllers mentioned above can achieve bounded tracking error by means of neural networks as well as backstepping techniques. However, they have a drawback of complexity.

One aspect is the computational expansion^[27] which results from the repeated differentiations of the certain nonlinear functions. Furthermore, the designed controller becomes more complex when the system order grows. At present, through the introduction of dynamic surface control (DSC) technique^[28,29] this problem of complexity growing has been solved. The technique is to utilize first-order filters of the synthetic inputs at each intermediate step and it has been recently widely used in adaptive control literature such as Ref.^[30] The other aspect that results into design complexity is the utilization of multiple approximators as mentioned in the previous examples. For the sake of solving the problem, in literature,^[10] the author only utilizes one single NN to mimic the lumped unknown function. That approach effectively avoids use of multiple approximators and reduces the computational burden.

On the basis of above observations, in the study alleviating the complexity and lightening the computational burden of the discrete-time controllers design will be further considered. In the paper, a novel controller is constructed based on single recurrent RBFNN. In order to provide convenience for the following use of the backstepping technique, the original system is first transformed into an equivalent n-step ahead predictor. Then, all the unknown functions are passed down, and only the ideal backstepping control law at the last step is approximated through the proposed NN. Thus, the controller in this paper is much simplified and its computational burden is also much lightened. In addition, a robust adaptive controller based on the approximation error is constructed for the purpose of achieving satisfying tracking performance. It is obviously seen from the stability analysis that all the signals are bounded, and through appropriate chose of control parameters arbitrarily small state tracking error can be obtained. Finally, the simulation result of an example is given, that verifies the superiority of the newly investigated controller.

The structure of the paper is given as below. In Section 2, the control problem to be investigated and preliminaries such as the architecture of RNN are presented. Section 3 describes the control design procedure for certain class of systems on the basis of NN approximation. Otherwise, all the closed-loop signals are also rigorously proved bounded through the constructed Lyapunov function in the section. In Section 4, a simulation example demonstrates the developed theory useful. Finally, Section 5 concludes the paper.

Notation. $\| \cdot \|$ denotes the Euclidean norm of vectors and induced norm of matrices.

$A := B$ means that B is defined as A.

$()^T$ represents the transpose of vector.

$\lambda_{max}()$ denotes the largest eigenvalue of a square matrix.

2. SYSTEM DESCRIPTION AND PRELIMINARIES

2.1 System description

The investigated systems are described as below:

$$\begin{aligned} \xi_j(\iota + 1) &= \xi_{j+1}(\iota) + f_j(\bar{\xi}_j(\iota)) & (1) \\ \xi_m(\iota + 1) &= f_m(\bar{\xi}_m(\iota)) + g_m(\bar{\xi}_m(\iota))u(\iota) + d(\iota) & (2) \\ y(\iota) &= \xi_1(\iota) & (3) \end{aligned}$$

where $\bar{\xi}_j(\iota) = [\xi_1(\iota), \xi_2(\iota), \dots, \xi_j(\iota)]^T \in R^j (j = 1, 2, \dots, m)$ is the system state variables which is assumed to be available for measurement, and $u(\iota) \in R, y(\iota) \in R$ are the system input and output, respectively; $f_j(\bar{\xi}_j(\iota)) (j = 1, 2, \dots, m)$ and $g_m(\bar{\xi}_m(\iota))$ are unknown smooth nonlinear functions; $d(\iota)$ is the external disturbance. Meanwhile, $|d(\iota)| \leq d_{max}$ where d_{max} is a known constant.

Our control goal is to design a controller that ensures the output tracks the known reference signal $y_d(\iota)$ and guarantees all the closed-loop signals in the system remain uniformly ultimately bounded.

To obtain our main results, we give the following assumptions:

Assumption 1 $\Omega_y := \{y|y = \xi_1\}$.

Assumption 2 $0 < g_m(\bar{\xi}_m(\iota)), 0 < g_{min}, 0 < g_{max}$ and $g_{min} < |g_m(\bar{\xi}_m(\iota))| < g_{max}$.

Definition 1 The solution of (1), (2) and (3) is semi-globally uniformly ultimately bounded (SGUUB), if for any Ω , which is a compact subset of R^n and all $\bar{\xi}_m(k_0) \in \Omega$, there exist an $\varepsilon > 0$ and a number $N(\varepsilon, \bar{\xi}_m(k_0))$ such that $\|\bar{\xi}_m(k)\| < \varepsilon$ for all $k \geq k_0 + N$.

2.2 Architecture of RNN

As we know many well-developed approaches can be utilized to emulate unknown nonlinear functions. However, among so many frequently used methods, only NN is known to own the capability of approximating any nonlinear unknown function to arbitrarily small error range. That is also the reason why it is often used for identification and control of nonlinear systems. In addition, we all know RBFNN owns simple structure and many other advantages. Thus, in order to enhance the approximation ability of NN, a RBFNN using radial basis function as basic activation function is proposed and depicted in Figure 1, where τ^{-1} means a time delay. The recurrent RBFNN comprises three layers, i.e., an input layer, a hidden layer, and an output layer. And the detailed description of the structure will be given as below:

(1) Layer 1 (input layer): The input and output in the first layer are respectively described as follows:

$$net_j^1(\chi) = \varphi_j(\chi) \tag{4}$$

$$o_j^1(\chi) = \theta_j^1(net_j^1(\chi)) = net_j^1(\chi) \tag{5}$$

$$j = 1, 2, \dots, m_1$$

where φ_j represents input to the j th node of input layer; χ denotes the number of iterations; θ_j^1 is activation function of the j th input node in this layer, which is set to be unit; m_1 denotes number of input nodes.

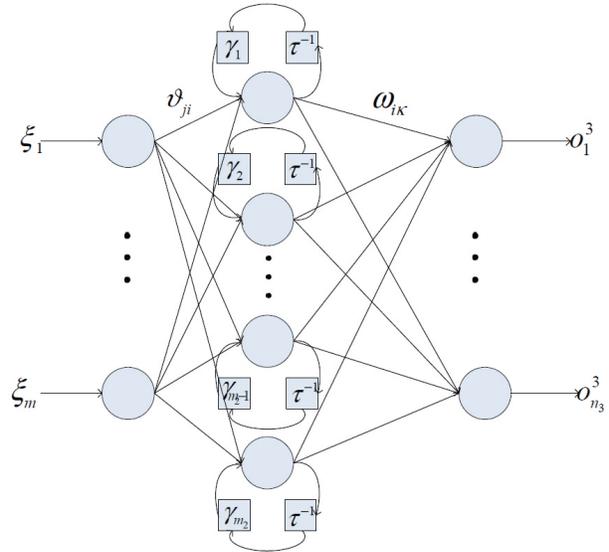


Figure 1. Structure of three-layer recurrent radial basis function neural network

(2) Layer 2 (hidden layer): Every recurrent loop is added to each corresponding layer. For the i -th node of the j -th input

$$net_i^2(\chi) = \gamma_i o_i^2(\chi - 1) + \sum_{j=1}^{m_1} \vartheta_{ji} \varphi_j^2(\chi) \tag{6}$$

$$o_i^2(\chi) = \theta_i^2(net_i^2(\chi)) = \exp\left(\frac{-\|net_i^2(\chi) - c_i\|^2}{b^2}\right) \tag{7}$$

$$i = 1, 2, \dots, m_2$$

where γ_i is the recurrent weight of the i th node; θ_i^2 is activation function of the i th node in this layer. For convenience of the following description, here we denote $\theta_i^2(\cdot)$ as $\theta(\cdot)$; ϑ_{ji} is the connective weight which is set to be 1 here; c_i represents center of the basis function of the i th node; $b > 0$ is the width of the basis function; m_2 denotes the number of nodes in hidden layer.

(3) Layer 3 (output layer): The output node \sum represents the summation of all incoming signals

$$net_\kappa^3(\chi) = \sum_{i=1}^{m_2} \omega_{i\kappa} \varphi_i^3(\chi) \tag{8}$$

$$o_\kappa^3(\chi) = \theta_\kappa^3(net_\kappa^3(\chi)) = net_\kappa^3(\chi) \tag{9}$$

$$\kappa = 1, 2, \dots, m_3$$

where $\omega_{i\kappa}$ is the connective weight between the i th node in the hidden layer and the κ th node in the output layer; θ_{κ}^3 is the activation function of the κ th node in the output layer, which is set to be unit; and o_{κ}^3 is the κ th output of the output layer; m_3 denotes the number of output nodes.

Moreover, we denote

$$\gamma = (\gamma_1 \gamma_2 \dots \gamma_{m_2})^T \tag{10}$$

$$\omega = (\omega_{11} \omega_{21} \dots \omega_{m_2 1}, \dots, \omega_{1 m_3} \omega_{2 m_3} \dots \omega_{m_2 m_3})^T \tag{11}$$

$$\varrho = (\gamma^T, \omega^T)^T \tag{12}$$

Then, the final output is

$$\begin{aligned} o^3 &= \Psi(\varphi, \gamma, \omega) \\ &= \Psi(\varphi | \varrho) \end{aligned} \tag{13}$$

where $\varphi = (\varphi_1, \varphi_2, \dots, \varphi_{m_1})$ are the inputs of the RNN, $\Psi = (\Psi_1, \Psi_2, \dots, \Psi_{m_3})$ are the outputs of the RNN. For a smooth function $F(\varphi) : R^{m_1} \rightarrow R^{m_3}$, it can be expressed as $F(\varphi) = \hat{F}(\varphi | \varrho^*) + \epsilon$, where ϵ is the functional reconstruction error. In the study the ideal weights ϱ^* can be denoted as $\varrho^* = \underset{\varrho}{\operatorname{argmin}} \{ \sup_{\varphi} \| F(\varphi) - \hat{F}(\varphi | \varrho) \| \}$, and $\hat{\varrho}$ is often denoted as the estimation of ϱ^* .

Assumption 3 On a compact set $\Omega_{\varphi} \in R^m$, the ideal RNN weight ϱ^* satisfies $\| \varrho^* \| \leq \varrho_m$ where ϱ_m is a positive constant.

Lemma 1 Consider φ being the input vector. The properties, i.e., $\lambda_{\max} [\theta(\varphi(\kappa)) \theta^T(\varphi(\kappa))] < 1$ and $\theta^T(\varphi(\kappa)) \theta(\varphi(\kappa)) < n$, will be used in the following system stability proof.

3. SINGLE NEURAL NETWORK APPROXIMATION BASED ADAPTIVE ROBUST CONTROL DESIGN

We are often confronted with the problem of causality contradiction when constructing a backstepping controller for systems under consideration. However, when the original system (1)-(3) is transformed into an ahead predictor^[28] the aforementioned problem can be naturally avoided. According to the transformation process in Ref.^[28] the initial strict-feedback form (1)-(3) can be transformed into

$$\begin{aligned} \xi_1(\iota + m) &= \xi_2(\iota + m - 1) + F_1(\bar{\xi}_m(\iota)) \\ &\vdots \\ \xi_{m-1}(\iota + 2) &= \xi_m(\iota + 1) + F_{m-1}(\bar{\xi}_m(\iota)) \\ \xi_m(\iota + 1) &= F_m(\bar{\xi}_m(\iota)) + G_m(\bar{\xi}_m(\iota))u(\iota) + d(\iota) \end{aligned} \tag{14}$$

where $F_j(\bar{\xi}_m(\iota))$ and $G_m(\bar{\xi}_m(\iota))$ depend on $f_j(\cdot)$ ($j = 1, 2, \dots, m$) and $g_m(\cdot)$ respectively. We should be aware of the fact that functions $F_j(\bar{\xi}_m(\iota))$ ($j = 1, 2, \dots, m$) become highly nonlinear. Actually, when j decreases, $F_j(\bar{\xi}_m(\iota))$ becomes more entangled and complex. The reason is that $F_{m-1}(\bar{\xi}_m(\iota))$ is obtained through one-step substitution, while $F_1(\bar{\xi}_m(\iota))$ is obtained through $(m - 1)$ -step substitution. Fortunately, a RNN can approximate an unknown smooth function to arbitrarily small error tolerance as discussed in section 2. Thus, employing RNN as our main controller may be a good choice when there is no knowledge of the exact structures of $F_j(\bar{\xi}_m(\iota))$ ($j = 1, 2, \dots, m$) and $G_m(\bar{\xi}_m(\iota))$. The construction procedure of the proposed controller will be presented as follows.

Firstly, in order to make it convenient for analysis and discussion, let $F_j(\iota) = F_j(\bar{\xi}_m(\iota))$ ($j = 1, 2, \dots, m$) and $G_m(\iota) = g_m(\bar{\xi}_m(\iota))$.

Step 1: From equations (1)-(3) and (14), we obtain that

$$\begin{aligned} e_1(\iota + m) &= \xi_1(\iota + m) - y_d(\iota + m) \\ &= \xi_2(\iota + m - 1) + F_1(\iota) - y_d(\iota + m) \end{aligned} \tag{15}$$

If we consider $\xi_2(\iota + m - 1)$ as a virtual control for the concerned system and $\delta_2(\iota + m - 1)$ as the ideal intermediate function, the following error variable can be introduced, i.e.,

$$e_2(\iota + m - 1) = \xi_2(\iota + m - 1) - \delta_2(\iota + m - 1) \tag{16}$$

It is obviously seen that $e_1(\iota + m) = 0$ if choosing

$$\delta_2(\iota + m - 1) = -F_1(\iota) + y_d(\iota + m) \tag{17}$$

Substituting (17) into (16) leads to

$$\begin{aligned} \xi_2(\iota + m - 1) &= e_2(\iota + m - 1) + \delta_2(\iota + m - 1) \\ &= e_2(\iota + m - 1) - F_1(\iota) + y_d(\iota + m) \end{aligned} \tag{18}$$

Substituting (18) into (15), we can achieve $e_1(\iota + m) = e_2(\iota + m - 1)$.

Step 2: Let $e_2(\iota) = \xi_2(\iota) - \delta_2(\iota)$, then its $(m-1)$ th difference is

$$\begin{aligned} e_2(\iota + m - 1) &= \xi_2(\iota + m - 1) - \delta_2(\iota + m - 1) \\ &= \xi_2(\iota + m - 1) + F_1(\iota) - y_d(\iota + m) \\ &= \xi_3(\iota + m - 2) + F_2(\iota) + F_1(\iota) - y_d(\iota + m) \\ &= \xi_3(\iota + m - 2) + F_2^*(\iota) - y_d(\iota + m) \end{aligned} \tag{19}$$

where $F_2^*(\iota) = F_1(\iota) + F_2(\iota)$. And $\xi_3(\iota + m - 2)$ is likewise considered as a virtual control for the investigated system.

Trough the introduction of the error variable

$$e_3(\iota + m - 2) = \xi_3(\iota + m - 2) - \delta_3(\iota + m - 2) \quad (20)$$

and chose of

$$\delta_3(\iota + m - 2) = -F_2^*(\iota) + y_d(\iota + m) \quad (21)$$

it can be easily obtained that $e_2(\iota + m - 1) = 0$. Substituting

(21) into (20) leads to

$$\begin{aligned} \xi_3(\iota + m - 2) &= e_3(\iota + m - 2) + \delta_3(\iota + m - 2) \\ &= e_3(\iota + m - 2) - F_2^*(\iota) + y_d(\iota + m) \end{aligned} \quad (22)$$

From (20), (21) and (22), $e_2(\iota + m - 1) = e_3(\iota + m - 2)$ can be obtained.

Step i: As in step 1 and step 2, for $e_j(\iota) = \xi_j(\iota) - \delta_j(\iota)$, its (m-j+1)th difference is presented as below

$$\begin{aligned} e_j(\iota + m - j + 1) &= \xi_j(\iota + m - j + 1) - \delta_j(\iota + m - j + 1) \\ &= \xi_{j+1}(\iota + m - j) + F_j(\iota) + F_{j-1}^*(\iota) - y_d(\iota + m) \\ &= \xi_{j+1}(\iota + m - j) + F_j^*(\iota) - y_d(\iota + m) \end{aligned} \quad (23)$$

where $F_j^*(\iota) = F_j(\iota) + F_{j-1}^*(\iota)$ is an unknown function. As the previous design step, $\xi_{j+1}(\iota + m - j)$ is also considered as a virtual control for the investigated system. Through the introduction of the error variable

$$\begin{aligned} e_{j+1}(\iota + m - j) &= \xi_{j+1}(\iota + m - j) \\ &\quad - \delta_{j+1}(\iota + m - j) \end{aligned} \quad (24)$$

and selection of

$$\delta_{j+1}(\iota + m - j) = -F_j^*(\iota) + y_d(\iota + m) \quad (25)$$

it is apparent to obtain that $e_j(\iota + m - j + 1) = 0$. Substituting (25) into (24) leads to

$$\begin{aligned} \xi_{j+1}(\iota + m - j) &= e_{j+1}(\iota + m - j) \\ &\quad - F_j^*(\iota) + y_d(\iota + m) \end{aligned} \quad (26)$$

Substituting (26) into (23), the error (23) is re-written as

$$e_j(\iota + m - j + 1) = e_{j+1}(\iota + m - j) \quad (27)$$

Step n: As for $e_m(\iota) = \xi_m(\iota) - \delta_m(\iota)$, its first difference is expressed as below

$$\begin{aligned} e_m(\iota + 1) &= \xi_m(\iota + 1) - \delta_m(\iota + 1) \\ &= F_m(\bar{\xi}_m(\iota)) + G_m(\bar{\xi}_m(\iota))u(\iota) + d(\iota) + F_{m-1}^*(\iota) - y_d(\iota + m) \\ &= F_m(\iota) + F_{m-1}^*(\iota) + G_m(\iota)u(\iota) + d(\iota) - y_d(\iota + m) \\ &= F_m^*(\iota) + G_m(\iota)u(\iota) + d(\iota) - y_d(\iota + m) \end{aligned} \quad (28)$$

where $F_m^*(\iota) = F_m(\iota) + F_{m-1}^*(\iota)$ and $G_m(\iota)$ are unknown functions. Apparently, $e_m(\iota + 1) = 0$ if the expected control law $u^*(\iota)$ is chosen as follows:

$$u(\iota) = u^*(\iota) = \frac{-F_m^*(\iota) - d(\iota) + y_d(\iota + m)}{G_m(\iota)} \quad (29)$$

Since $F_m^*(\iota)$ and $G_m(\iota)$ are unknown, we can utilize the recurrent RBFNN to emulate $u^*(\iota)$ as below:

$$u^*(\iota) = \Psi(\bar{\xi}_m(\iota)|\varrho^*) + \epsilon(\bar{\xi}_m(\iota)) \quad (30)$$

where $\bar{\xi}_m(\iota) = \varphi(\iota)$. In this paper, let the recurrent weights to be constant and from the aforementioned description, (30) can be easily converted to the following form:

$$\begin{aligned} u^*(\iota) &= \Psi(\varphi(\iota)|\varrho^*) + \epsilon(\varphi(\iota)) \\ &= \omega^{*T}\theta(\varphi(\iota)) + \epsilon(\varphi(\iota)) \end{aligned} \quad (31)$$

where $\theta = \theta_j^2$.

Assumption 4 For compact set $\Omega_\varphi, \epsilon(\varphi(\iota))$ satisfies $\|\epsilon(\varphi(\iota))\| \leq \varsigma$, where $\varsigma > 0$ is unknown constant.

Let $\hat{\omega}(\iota)$ be the estimate of ω^* , and $\hat{\zeta}(\iota)$ be the estimate of ς . Then,

$$u(\iota) = \hat{\omega}^T\theta(\varphi(\iota)) + \hat{\zeta}(\iota) \quad (32)$$

The updating algorithms are as below:

$$\begin{aligned} \hat{\omega}(\iota + 1) &= \hat{\omega}(\iota) - \lambda(\theta(\varphi(\iota))e_m(\iota + 1)) \\ \hat{\zeta}(\iota + 1) &= \hat{\zeta}(\iota) - \zeta(e_m(\iota + 1) + \mu\hat{\zeta}(\iota)) \end{aligned} \quad (33)$$

Substituting (32) into (28), the error (28) can be transformed into

$$\begin{aligned} e_m(\iota + 1) &= G_m(\iota)(\hat{\omega}^T(\iota)\theta(\varphi(\iota)) + \hat{\zeta}(\iota)) \\ &\quad + F_m^*(\iota) - y_d(\iota + m) + d(\iota) \end{aligned} \quad (34)$$

Combining (31), (34) equals to

$$e_m(\iota + 1) = G_m(\iota)(\hat{\omega}^T(\iota)\theta(\varphi(\iota)) + \hat{\zeta}(\iota)) + F_m^*(\iota) - y_d(\iota + m) + d(\iota) + G_m(\iota)u^*(\iota) - G_m(\iota)(\omega^{*T}\theta(\varphi(\iota)) + \epsilon(\varphi(\iota))) \tag{35}$$

Substituting (29) into (35) leads to

$$\begin{aligned} e_m(\iota + 1) &= G_m(\iota)(\hat{\omega}^T(\iota)\theta(\varphi(\iota)) + \hat{\zeta}(\iota)) - G_m(\iota)(\omega^{*T}\theta(\varphi(\iota)) + \epsilon(\varphi(\iota))) \\ &= G_m(\iota)(\tilde{\omega}^T(\iota)\theta(\varphi(\iota)) + \hat{\zeta}(\iota) - \epsilon(\varphi(\iota))) \end{aligned} \tag{36}$$

where $\tilde{\omega}(\iota) = \hat{\omega}(\iota) - \omega^*$.

Proof: The Lyapunov function is chosen as below

$$V = \sum_{j=1}^m e_j^2(\iota) + \tilde{\omega}^T(\iota)\lambda^{-1}\tilde{\omega}(\iota) + \zeta^{-1}\zeta^2(\iota) \tag{37}$$

Theorem 1 Taking the second-order nonlinear system depicted in (1)-(3) into consideration, we provide the control law as (32) and the adaptation laws as (33). Then, under any bounded initial conditions, i.e., $\tilde{\xi}_m(0)$ is initialized in Ω , all the closed-loop system signals preserve SGUUB and a small tracking tolerance can be achieved through appropriate selection of control parameters.

From equation (36), we know that

$$\tilde{\omega}^T(\iota)\theta(\varphi(\iota)) = \frac{e_m(\iota + 1)}{G_m(\iota)} - \hat{\zeta}(\iota) + \epsilon(\varphi(\iota)) \tag{38}$$

The first difference of (37) is given by

$$\Delta V = \sum_{j=1}^m [e_j^2(\iota + 1) - e_j^2(\iota)] + \tilde{\omega}^T(\iota + 1)\lambda^{-1}\tilde{\omega}(\iota + 1) - \tilde{\omega}^T(\iota)\lambda^{-1}\tilde{\omega}(\iota) + \zeta^{-1}\zeta^2(\iota + 1) - \zeta^{-1}\zeta^2(\iota) \tag{39}$$

According to (27), we have

$$e_1(\iota + 1) = e_2(\iota), e_2(\iota + 1) = e_3(\iota), \dots, e_j(\iota + 1) = e_{j+1}(\iota), j = 1, 2, \dots, m - 1 \tag{40}$$

Then, its difference along (33) and (36) is as follows:

$$\begin{aligned} \Delta V &= e_m^2(\iota + 1) - e_1^2(\iota) + \tilde{\omega}^T(\iota + 1)\lambda^{-1}\tilde{\omega}(\iota + 1) - \tilde{\omega}^T(\iota)\lambda^{-1}\tilde{\omega}(\iota) + \zeta^{-1}\zeta^2(\iota + 1) - \zeta^{-1}\zeta^2(\iota) \\ &= e_m^2(\iota + 1) - e_1^2(\iota) - 2\tilde{\omega}^T(\iota)\theta(\varphi(\iota))e_m(\iota + 1) + (\theta(\varphi(\iota))e_m(\iota + 1))^T \\ &\quad \lambda\theta(\varphi(\iota))e_m(\iota + 1) - 2\tilde{\zeta}(\iota)(e_m(\iota + 1) + \mu\hat{\zeta}(\iota)) + \zeta(e_m(\iota + 1) + \mu\hat{\zeta}(\iota))^2 \\ &= e_m^2(\iota + 1) - e_1^2(\iota) - 2\frac{e_m^2(\iota + 1)}{G_m(\iota)} + 2\hat{\zeta}(\iota)e_m(\iota + 1) - 2\epsilon(\varphi(\iota))e_m(\iota + 1) \\ &\quad + \theta^T(\varphi(\iota))\lambda\theta(\varphi(\iota))e_m^2(\iota + 1) + \zeta e_m^2(\iota + 1) - 2\tilde{\zeta}(\iota)e_m(\iota + 1) - 2\mu\tilde{\zeta}(\iota)\hat{\zeta}(\iota) \\ &\quad + 2\zeta\mu\hat{\zeta}(\iota)e_m(\iota + 1) + \zeta\mu^2\hat{\zeta}^2(\iota) \\ &= (\zeta + 1 - \frac{2}{G_m(\iota)})e_m^2(\iota + 1) - e_1^2(\iota) - 2\epsilon(\varphi(\iota))e_m(\iota + 1) + 2\zeta e_m(\iota + 1) \\ &\quad + \theta^T(\varphi(\iota))\lambda\theta(\varphi(\iota))e_m^2(\iota + 1) - 2\mu\tilde{\zeta}(\iota)\hat{\zeta}(\iota) + 2\zeta\mu\hat{\zeta}(\iota)e_m(\iota + 1) + \zeta\mu^2\hat{\zeta}^2(\iota) \end{aligned} \tag{41}$$

Using the facts that

$$\begin{aligned} \theta^T(\varphi(\iota))\theta(\varphi(\iota)) &< n; \\ \theta^T(\varphi(\iota))\lambda\theta(\varphi(\iota)) &\leq \tau n; \\ -2\epsilon e_m(\iota + 1) &\leq \tau e_m^2(\iota + 1) + \frac{\epsilon^2}{\tau}; \\ 2\zeta\mu\hat{\zeta}(\iota)e_m(\iota + 1) &\leq \zeta e_m^2(\iota + 1) + \zeta\mu^2\hat{\zeta}^2(\iota); \\ 2\tilde{\zeta}(\iota)\hat{\zeta}(\iota) &= \tilde{\zeta}^2(\iota) + \hat{\zeta}^2(\iota) - \zeta^2; \\ 2\zeta e_m(\iota + 1) &\leq \tau e_m^2(\iota + 1) + \frac{\zeta^2}{\tau}; \end{aligned} \tag{42}$$

we obtain

$$\begin{aligned}
 \Delta V &\leq \left(\zeta + 1 - \frac{2}{G_m(\iota)}\right)e_m^2(\iota + 1) - e_1^2(\iota) + \tau e_m^2(\iota + 1) + \frac{\epsilon^2}{\tau} \\
 &\quad + \tau e_m^2(\iota + 1) + \frac{\zeta^2}{\tau} + \tau n e_m^2(\iota + 1) \\
 &\quad - \mu \hat{\zeta}^2(\iota) - \mu \zeta^2(\iota) + \mu \zeta^2 + \zeta e_m^2(\iota + 1) + \zeta \mu^2 \hat{\zeta}^2(\iota) + \zeta \mu^2 \zeta^2(\iota) \\
 &\leq \left(2\zeta + 2\tau + \tau n + 1 - \frac{2}{g_{max}}\right)e_m^2(\iota + 1) - e_1^2(\iota) \\
 &\quad + \left(\frac{1}{\tau} + 2\mu\right)\zeta^2 + (2\zeta\mu^2 - \mu)\hat{\zeta}^2(\iota) - \mu\zeta^2(\iota) + \frac{\epsilon^2}{\tau}
 \end{aligned} \tag{43}$$

where $\rho = (\frac{1}{\tau} + 2\mu)\zeta^2 + \frac{\epsilon^2}{\tau}$ is nonnegative. The design parameters are chosen as below

$$0 < \mu < \frac{1}{2\zeta}, \zeta < \frac{1}{g_{max}} - \tau\left(\frac{n}{2} + 1\right) - \frac{1}{2} \tag{44}$$

Then, as long as the error $e_m(\iota)$ is larger than $\sqrt{\rho}$, $\Delta V \leq 0$. This implies for all $\iota \geq 0$, $V(\iota)$ is bounded. This indicates $e_m(\iota)$ is bounded and satisfies that $\Omega_e \in R := \{e | e \leq \sqrt{\rho}\}$. Consequently, the theorem is completely proved.

4. SIMULATION EXAMPLE

To illustrate the superiority of the proposed approach an example is given in the following description:

$$\begin{aligned}
 \xi_1(\iota + 1) &= \xi_2(\iota) + f_1(\bar{\xi}_1(\iota)) \\
 \xi_2(\iota + 1) &= f_2(\bar{\xi}_2(\iota)) + g_2(\bar{\xi}_2(\iota))u(\iota) + d(\iota) \\
 y_\iota &= \xi_1(\iota)
 \end{aligned} \tag{45}$$

where $f_1(\bar{\xi}_1(\iota))$, $f_2(\bar{\xi}_2(\iota))$ and $g_2(\bar{\xi}_2(\iota))$ are unknown smooth functions. According to (32) the controller is chosen as below:

$$u(\iota) = \hat{\omega}^T(\iota)\theta(\varphi(\iota)) + \hat{\zeta}(\iota) \tag{46}$$

For the purpose of simulation, the unknown system functions are assumed as below:

$$\begin{aligned}
 f_1(\bar{\xi}_1(\iota)) &= \frac{1.1\xi_1^2(\iota)}{1 + \xi_1^2(\iota)} \\
 f_2(\bar{\xi}_2(\iota)) &= \frac{\xi_1(\iota)}{1 + \xi_1^2(\iota) + \xi_2^2(\iota)} \\
 g_2(\bar{\xi}_2(\iota)) &= 1 \\
 d(\iota) &= 0.1 \times \sin(\iota \times \pi/10)
 \end{aligned} \tag{47}$$

At beginning let $\xi(0) = [1 \ 0]^T$. The node number of the first layer, the hidden layer and the output layer is $m_1 = 2, m_2 = 9, m_3 = 1$, respectively. The initial weights for RNN are $\hat{\omega}(0) = 1.5 \times \text{ones}(L, 1)$ and $\gamma(0) = 1.8 \times \text{ones}(L, 1)$. That is, the feedforward weights are initialized to be 1.5, and the recurrent weights are initialized

to be 1.8. The controller parameters chosen for simulation are $\lambda = 0.01, \zeta = 0.01, \mu = 0.001$. The reference signal is $y_d(\iota) = (1/2) \sin(\iota \times \pi/20) + (1/2) \sin(\iota \times \pi/10)$.

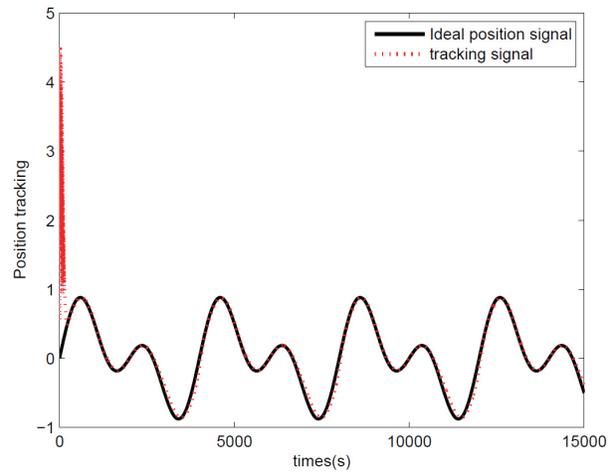


Figure 2. The tracking performance of system

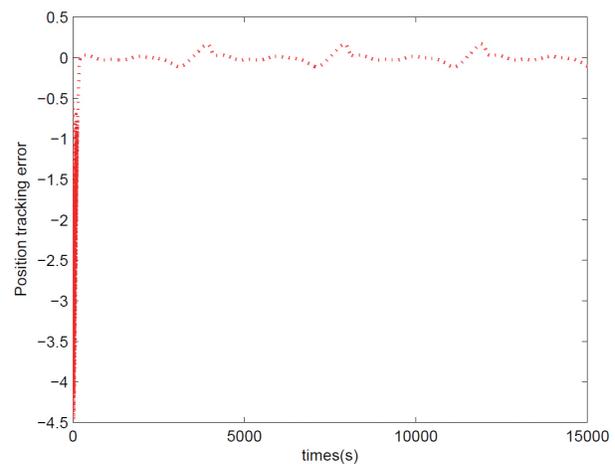


Figure 3. The tracking error of system

Simulation results obtained in this situation are presented in Figures 2-5. The proposed control approach is utilized to

track the reference signal $y_d(t)$ for the system (47). Figures 2 and 3 apparently demonstrate that the final tracking performance is favorable with high tracking precision. Figure 4 gives the control input signal. It can be obviously observed that the control input is not smooth but bounded. Figure 5 depicts the study behavior of the RNN weights. The four figures show that the concerned signals are bounded.

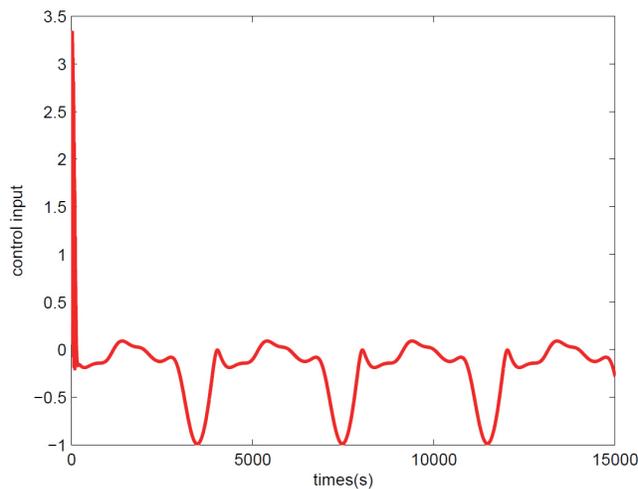


Figure 4. The control input of system

5. CONCLUSIONS

Through combining the advantage of backstepping technique and NN, a novel approach for certain systems with external unknown disturbance is proposed in the study. In order to enhance mapping ability of NNs, feedback and delay loops are added to the original RBFNN. Then, following the system

transformation and combining with backstepping technique, single recurrent radial basis function neural network is utilized to emulate the lumped nonlinear functions. Through this method, the structure of the controller can be simplified observably, and the computational burden can be alleviated drastically. Another adaptive controller is also constructed in this article so as to weaken the negative impact of the approximation error on the investigated system. Stability analysis shows all the closed-loop system signals are ensured uniformly ultimately bounded, and arbitrarily small tracking error can be obtained through appropriate parameters selection. Finally, a simulation example demonstrates the proposed controller is feasible and effective.

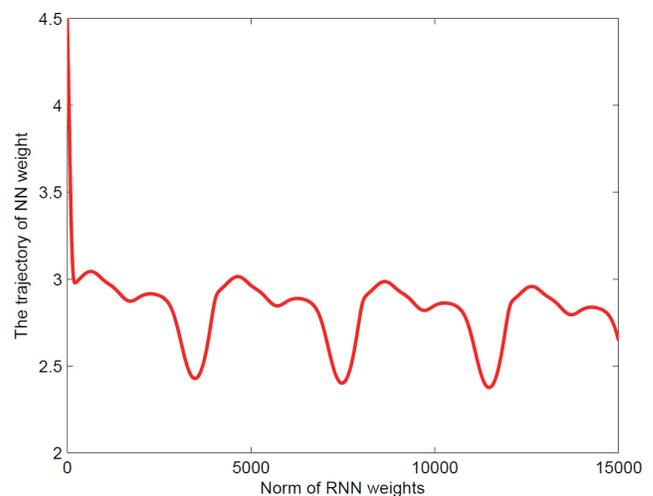


Figure 5. The trajectory of RNN weights

REFERENCES

- [1] Polycarpou M. Stable adaptive neural control scheme for nonlinear systems. *Automatic Control*. 1996; 41: 447-451. <http://dx.doi.org/10.1109/9.486648>
- [2] Zhang T, Ge S, Hang C. Adaptive neural network control for strict-feedback nonlinear systems using backstepping design. *Automatica*. 2000; 36: 1835-1846. [http://dx.doi.org/10.1016/S0005-1098\(00\)00116-3](http://dx.doi.org/10.1016/S0005-1098(00)00116-3)
- [3] Ho D, Zhang P, Xu J. Fuzzy wavelet networks for function learning. *IEEE Transactions on Neural Networks*. 2001; 9(1): 200-211.
- [4] Sarma M, Sarma K. An ANN based approach to recognize initial phonemes of spoken words of Assamese language. *Applied Soft Computing*. 2013; 13(5): 2281-2291. <http://dx.doi.org/10.1016/j.asoc.2013.01.004>
- [5] Miao B, Li T. A novel neural network-based adaptive control for a class of uncertain nonlinear systems in strict-feedback form. *Nonlinear Dynamics*. 2015; 79: 1005-1013. <http://dx.doi.org/10.1007/s11071-014-1717-2>
- [6] Hsu C, Cheng K. Recurrent fuzzy-neural approach for nonlinear control using dynamic structure learning scheme. *Neurocomputing*. 2008; 71: 3447-3459. <http://dx.doi.org/10.1016/j.neucom.2007.10.014>
- [7] Fu Z, Xie W, Luo W. Robust on-line nonlinear systems identification using multilayer dynamic neural networks with two-time scales. *Neurocomputing*. 2013; 113(3): 16-26.
- [8] Peng J, Dubay R. Identification and adaptive neural network control of a DC motor system with dead-zone characteristics. *ISA Transactions*. 2011; 50(4): 588-598. PMID:21788017. <http://dx.doi.org/10.1016/j.isatra.2011.06.005>
- [9] Miao Z, Wang Y, Yang Y. Robust tracking control of uncertain dynamic nonholonomic systems using recurrent neural networks. *Neurocomputing*. 2014; 142: 216-227. <http://dx.doi.org/10.1016/j.neucom.2014.03.061>
- [10] Sun G, Wang D, Li T, *et al.* Single neural network approximation based adaptive control for a class of uncertain strict-feedback non-

- linear systems. *Nonlinear Dynamics*. 2013; 72: 175-184. <http://dx.doi.org/10.1007/s11071-012-0701-y>
- [11] Lin C, Hsueh C, Chen C. Robust adaptive backstepping control for a class of nonlinear systems using recurrent wavelet neural network. *Optimal Control Applications and Methods*. 2014; 142: 372-382.
- [12] Hartman E, Keeler J, Kowalski J. Layered neural networks with Gaussian hidden units as universal approximation. *Neural Computing*. 1990; 2(2): 210-215. <http://dx.doi.org/10.1162/neco.1990.2.2.210>
- [13] Huang Z, Yang Q, Luo X. Adaptive output-feedback control of a class of discrete-time nonlinear systems. *Proceedings of American Control Conference*. 1993; 1359-1364.
- [14] Ge S, Li G, Lee T. Adaptive NN control for a class of strict-feedback discrete-time nonlinear systems. *Automatica*. 2003; 39: 807-819. [http://dx.doi.org/10.1016/S0005-1098\(03\)00032-3](http://dx.doi.org/10.1016/S0005-1098(03)00032-3)
- [15] Hsu C. Adaptive recurrent neural network control using a structure adaptation algorithm. *Neural Computing Application*. 2009; 18: 115-125. <http://dx.doi.org/10.1007/s00521-007-0164-0>
- [16] Lin F, Wai R. Robust recurrent fuzzy neural network control for linear synchronous motor drive system. *Neurocomputing*. 2003; 50: 365-390. [http://dx.doi.org/10.1016/S0925-2312\(02\)00572-6](http://dx.doi.org/10.1016/S0925-2312(02)00572-6)
- [17] Lin F, Wai R, Chou W, *et al.* Adaptive backstepping control using recurrent neural network for linear induction motor drive. *IEEE Transactions on Industrial Electronics*. 2002; 49(1): 134-146. <http://dx.doi.org/10.1109/41.982257>
- [18] Lin F, Shieh H, Shieh P, *et al.* An adaptive recurrent-neural-network motion controller for X-Y table in CNC machine. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, Cybernetics*. 2006; 36(2): 286-299. PMID:16602590. <http://dx.doi.org/10.1109/TSMCB.2005.856719>
- [19] Hsu C, Cheng K. Recurrent fuzzy-neural approach for nonlinear control using dynamic structure learning scheme. *Neurocomputing*. 2008; 71: 3447-3459. <http://dx.doi.org/10.1016/j.neucom.2007.10.014>
- [20] Zhang T, Ge S, Hang C. Adaptive neural network control for strict feedback nonlinear systems using backstepping design. *Automatica*. 2000; 36(12): 1835-1846. [http://dx.doi.org/10.1016/S0005-1098\(00\)00116-3](http://dx.doi.org/10.1016/S0005-1098(00)00116-3)
- [21] Zhang T, Ge S, Hang C. Adaptive neural network control for a class of lowtriangular-structured nonlinear systems. *IEEE Transactions on Neural Networks*. 2006; 17(2): 509-514. PMID:16566476. <http://dx.doi.org/10.1109/TNN.2005.863403>
- [22] Choi J, Farrell J. Adaptive observer backstepping control using neural networks. *IEEE Transactions on Neural Networks*. 2001; 12(5): 1103-1112. PMID:18249937. <http://dx.doi.org/10.1109/72.950139>
- [23] Wang G. Adaptive NN control of uncertain nonlinear pure-feedback systems. *Automatica*. 2002; 38(4): 671-682. [http://dx.doi.org/10.1016/S0005-1098\(01\)00254-0](http://dx.doi.org/10.1016/S0005-1098(01)00254-0)
- [24] Park J, Sandberg L. Universal approximation using radial-basis-function networks. *Neural Computing*. 1991; 3(2): 246-257. <http://dx.doi.org/10.1162/neco.1991.3.2.246>
- [25] Hartman E, Keeler J, Kowalski J. Robust and adaptive backstepping control for nonlinear systems using RBF neural networks. *IEEE Transactions on Neural Networks*. 2004; 15: 693-701. PMID:15384556. <http://dx.doi.org/10.1109/TNN.2004.826215>
- [26] Wen G, Liu Y, Chen C. Direct adaptive robust NN control for a class of discrete-time nonlinear strict-feedback SISO systems. *Neural Computing Applications*. 2012; 21: 1423-1431. <http://dx.doi.org/10.1007/s00521-011-0596-4>
- [27] Zhang C, Yang F, Wu D. Adaptive neural network tracking control for a class of nonlinear systems. *International Journal of Systems Science*. 2010; 2(2): 143-158.
- [28] Yip P, Hedrick J. Adaptive dynamic surface control: a simplified algorithm for adaptive backstepping control of nonlinear systems. *International Journal of Control*. 1998; 71(5): 959-979. <http://dx.doi.org/10.1080/002071798221650>
- [29] Wang D, Huang J. Neural network based adaptive dynamic surface control for nonlinear systems in strictfeedback form. *IEEE Transactions on Neural Networks*. 2005; 16(1): 195-202. PMID:15732399. <http://dx.doi.org/10.1109/TNN.2004.839354>
- [30] Li T, Wang D, Feng G, *et al.* A DSC approach to robust adaptive NN tracking control for strict-feedback nonlinear systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, Cybernetics*. 2010; 40(3): 915-927. PMID:19887321. <http://dx.doi.org/10.1109/TSMCB.2009.2033563>
- [31] Ge S, Lee T, Li G, *et al.* Adaptive NN control for a class of discrete-time nonlinear systems. *International Journal of Control*. 2003; 76(4): 334-354. <http://dx.doi.org/10.1080/0020717031000073063>
- [32] Ge S, Zhang J, Lee T. Adaptive neural networks control for a class of MIMO nonlinear systems with disturbances in discrete-time. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, Cybernetics*. 2004; 34(4): 1630-1645. <http://dx.doi.org/10.1109/TSMCB.2004.826827>
- [33] Yang C, Ge S, Xiang C, *et al.* Output feedback NN control for two classes of discrete-time systems with unknown control directions in a unified approach. *IEEE Transactions on Neural Networks*. 2008; 19(11): 1873-1886. PMID:18990642. <http://dx.doi.org/10.1109/TNN.2008.2003290>
- [34] Zhang J, Ge S, Lee T. Output feedback control of a class of discrete MIMO nonlinear systems with triangular form inputs. *IEEE Transactions on Neural Networks*. 2005; 16(6): 1491-1503. PMID:16342490. <http://dx.doi.org/10.1109/TNN.2005.852242>