2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

A Novel Adaptive NN Control for a Class of Strict-Feedback Nonlinear Systems

Tieshan Li*[1,2], Dan Wang[3] and Wei Li[1].

***Abstract*— An adaptive neural network control(ANNC) is proposed for a class of strict-feedback uncertain nonlinear systems with unknown system nonlinearities and unknown virtual control gain nonlinearities. Combining the dynamic surface control(DSC) technique with minimal-learning-parameters(MLP) algorithm, a systematic procedure for synthesis of ANNC is developed based on the universal approximation of neural networks. An important feature of the proposed algorithm is that the number of parameters updated on line for each subsystem is reduced only to one, both problems of "explosion of complexity" and "curse of dimension" are solved simultaneously, such that the computation load is reduced drastically and it is convenient to implement the controller in applications. It is shown that all closed-loop signals are semi-global uniform ultimate bound(SGUUB) via Lyapunov stability theory. Finally, simulation results are presented to demonstrate the effectiveness of the proposed scheme.**

***Index Terms*— Uncertain nonlinear systems, neural networks, adaptive control, dynamic surface control, minimal-learning parameters.**

I. INTRODUCTION

IN the past decades, the adaptive control of nonlinear systems with linearly parameterized uncertainty has achieved significant progress (see [1]∼[3] and references therein). For systems with high uncertainty, which cannot be modelled or repeatable, adaptive control approach obtained further development by means of neural network (NN) control schemes(e.g., [4]∼[7]) or fuzzy control schemes(e.g., [8]∼[10]) based on the idea of backstepping.

However, there is a substantial "dimension curse" restriction in the aforementioned works. That is, the number of hidden units becomes prohibitively large as we move to high dimensional systems, which imposes that there are many parameters need to be tuned in the approximator-based adaptive control schemes, such that the time-consuming process is unavoidable during the implementation of these schemes. This drawback restricts the applicability of these methods. This problem has been first researched in [11] and [12], and further discussed in [13]∼[15] when using adaptive fuzzy control schemes or NN control schemes.

1 Navigation College, Dalian Maritime University, Dalian, 116026, China; 2 School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai, 200030, China. 3 Marine Engineering College, Dalian Maritime University, Dalian, China. *Corresponding Author. E-mail:tieshanli@126.com; dwangdl@gmail.com; liweidmu@126.com

This work is supported in part by the National Natural Science Foundation of China (Grant Nos.60874056, 60674037 and 50779033), China Postdoctoral Science Foundation (Grant No. 20070420101), Shanghai Postdoctoral Scientific Program(Grant No.07R214128), and the 863 Plan (Grant No. 2007AA11Z250).

On the other hand, there is a well-known drawback of "explosion of complexity" with the conventional backstepping technique. That is, the complexity of a controller grows drastically as the order of the system increases. This "explosion of complexity" is caused by the repeated differentiations of certain nonlinear functions. In [16], a dynamic surface control (DSC) technique was proposed to eliminate this problem by introducing a first-order filtering of the synthetic input at each step of the traditional backstepping approach. In [17], this DSC method was extended to adaptive systems in which the nonlinearities are linear in the uncertain parameters. In [18], the DSC method was first extended to adaptive tracking control via neural networks for a class of strict-feedback uncertain systems without external disturbances, and the asymptotic semiglobal stability was achieved.

In this paper, incorporating the DSC technique into the MLP algorithm in [15], a new systematic procedure is developed for the synthesis of stable adaptive NN tracking controllers. RBF NNs are used to approximate the unknown functions. The controller guarantees that the resulting closed-loop system is SGUUB. The main features of the controllers are that 1) the adaptive mechanism with minimal learning parameterizations is achieved, i.e., the number of parameters updated on line for each subsystem is reduced to one, and 2) both problems of "explosion of complexity" and "curse of dimension" are solved simultaneously. Therefore, the computation burden is reduced dramatically and it is convenient to implement the algorithm in applications.

II. PRELIMINARIES

A. PROBLEM FORMULATION

Consider an uncertain nonlinear dynamic system in the following form

$$\left\{\begin{array}{l} \dot{x}_i = g_i(\bar{x}_i)x_{i+1} + f_i(\bar{x}_i)\,,\ 1 \le \mathrm{i} \le n-1 \\ \dot{x}_n = f_n(x) + g_n(x)u \\ y = x_1 \end{array}\right. \tag{1}$$

where $x = (x_1, x_2, ..., x_n)^T \in R^n$ is the system state, $u \in R$ is the control input, $y \in R$ is the output of the system and w is the model uncertainty belonging to a compact set, which includes uncertain parameter vector of the system. Let $\bar{x}_i = [x_1, x_2, ..., x_n]^T$. $f_i(\bar{x}_i)\, i = 1, 2, ..., n$ are uncertain smooth system functions with $f_i(0) = 0$ and $g_i(\bar{x}_i)\, i = 1, 2, ..., n$ are uncertain smooth virtual control gain functions, all of which may not be linearly parameterized.

The following assumption is introduced.

978-1-4244-4524-0/09/$25.00 ©2009 AACC

Assumption 1: The uncertain virtual control gain functions $g_i(\bar{x}_i)\, i = 1, 2, ..., n$ are confined within a certain range such that

$$0 < b_{min} \le |g_i(\bar{x}_i)| \le b_{max} \tag{2}$$

where b_{min} and b_{max} are the lower and upper bound of the gain functions, respectively.

The aforementioned assumption implies that the smooth virtual control gain functions $g_i(\bar{x}_i)\, i = 1, 2, ..., n$ are strictly either positive or negative. From now on, without loss of generality, we will assume $0 < b_{min} \le g_i(\bar{x}_i), i = 1, 2, ..., n$. Assumption 1 is reasonable because $g_i(\bar{x}_i)$ being away from zero is the controllable conditions of (1).

Assumption 2: The reference signal $y_d(t)$ is a sufficiently smooth function of t and y_d, $\dot{y_d}$, $\ddot{y_d}$ are bounded, that is, there exists a positive constant B_0, such that $\Pi_0 := \{(y_d, \dot{y_d}, \ddot{y_d}) : {y_d}^2 + \dot{y_d}^2 + \ddot{y_d}^2 \le B_0\}$.

Remark 1: [18] In the traditional backstepping-based adaptive tracking control design, since the information of ith time derivative of $y_d(t)$ is needed when designing the virtual controller at Step $(i+1)$, so the requirement that the reference signal $y_d(t)$ is assumed to be available together with its n time derivatives is imposed ([13]).

The control objective is to develop an adaptive NN tracking controller such that all the solutions of the resulting closed-loop system are SGUUB, and the tracking error $z_1 = y(t) - y_d(t)$ can be rendered small.

B. RBF Neural Network

In control engineering, RBF neural networks are usually used as a tool for modelling nonlinear functions because of their good capabilities in function approximation. They belong to a class of linearly parameterized networks. For comprehensive treatment of neural networks approximation, see [4]. RBF neural networks can be described as $w^T S(z)$ with input vector $z \in R^n$, weight vector $w \in R^l$, node number l, and basis function vector $S(z) \in R^l$. Universal approximation results indicate that, if l is chosen sufficiently large, then $w^T S(z)$ can approximate any continuous function to any desired accuracy over a compact set. In this paper, we use the following RBF neural networks to approximate a smooth function $h(z) : R^q \to R$

$$h_{nn}(z) = w^T S(z) \tag{3}$$

where the input vector $z \in \Omega \subset R^n$, weight vector $w = [w_1, w_2, \dots, w_l]^T \in R^l$, the neural network node number $l > 1$, and $S(z) = [s_1(z), s_2(z), \dots, s_l(z)]^T$, with $s_i(z)$ being chosen as the commonly used Gaussian functions, which have the form

$$s_i(z) = \exp\left[\frac{-(z-\mu_i)^T(z-\mu_i)}{\eta_i^2}\right], i = 1, 2, \dots, l$$

where $\mu_i = [\mu_{i1}, \mu_{i2}, \dots, \mu_{in}]^T$ is the center of the receptive field and η_i is the width of the Gaussian function.

For the unknown nonlinear function $f(x)$, we have the following approximation over the compact sets Ω

$$f(x) = w^{*T} S(x) + \varepsilon. \quad \forall x \in \Omega \subseteq R^n \tag{4}$$

where $S(x)$ is the basis function vector, ε is the approximation error, and w^* is an unknown ideal constant weight vector.

The ideal weight vector w^* in (4) is an "artificial" quantity required only for analytical purposes. Typically, w^* is chosen as the value of w that minimizes $|\ \varepsilon\ |$ for all $x \in \Omega$, where $\Omega \subseteq R^n$ is a compact set, i.e.,

$$w^* := \arg\min_{w \in R^n} \left\{ \sup_{x \in \Omega} |\ f(x) - w^T S(x)\ | \right\}.$$

We make the following assumption on the approximation error.

Assumption 3: Over a compact region $\Omega \in R^n$

$$|\ \varepsilon\ | \le \varepsilon^*$$

where $\varepsilon^* > 0$ is an unknown bound.

The following lemma provides a new description for the continuous function by using continuous function separation technique and RBF NN approximation, which enables one to deal with nonlinear parameterization and will result in a solution to the robust adaptive NN control problem of nonlinear parameterized systems.

Lemma 1: [14] For any given real continuous function $f(x, \theta)$ with $f(0, \theta) = 0$, when the continuous function separation technique and RBF NN approximation technique are used, then $f(x, \theta)$ can be denoted as follows

$$f(x, \theta) = \bar{S}(x) A x \tag{5}$$

where $\bar{S}(x) = [1, S(x)] = [1, s_1(x), s_2(x), \cdots, s_l(x)]$, $s_i(x), i = 1, 2, \cdots, l$ are the RBF basis functions which are known and l is the node number. $A^T = [\varepsilon, W^T]$, $\varepsilon^T = [\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n]$ is a vector of the approximation error and $W = \begin{bmatrix} w_{11}^* & w_{12}^* & \cdots & w_{1n}^* \\ w_{21}^* & w_{22}^* & \cdots & w_{2n}^* \\ \vdots & \vdots & \cdots & \vdots \\ w_{l1}^* & w_{l2}^* & \cdots & w_{ln}^* \end{bmatrix}$ is a weight matrix.

III. DESIGN OF ROBUST ADAPTIVE FUZZY TRACKING CONTROL

A. Design of Robust Adaptive NN Tracking Control

Now we will incorporate the DSC technique into a RBF NN based robust adaptive tracking design scheme for (1). Similar to the traditional backstepping method, the recursive design procedure contains n steps. At each step, the virtual controller α_{i+1}, $i = 1, 2, \dots, n-1$ shall be developed. Finally an overall control law u is constructed at step n.

Step 1: Define the 1st error variable $z_1 = x_1 - y_d$, then

$$\dot{z}_1 = g_1(x_1) x_2 + f_1(x_1) - \dot{y}_d \tag{6}$$

Since $f_1(x_1)$ is an unknown continuous function, according to Lemma 1, RBF NN $\hat{f}_1(x_1, A_1)$ with input vector $x_1 \in U_{x_1}$, where U_{x_1} is some compact set, is proposed here to approximate uncertain function $f_1(x_1)$ with A_1 a matrix containing unknown constants. Then $f_1(x_1)$ can be expressed as

$$f_1(x_1) = \xi_1(x_1) A_1 x_1 + \varepsilon_1$$

$$\begin{aligned}&= \xi_1(x_1) A_1 z_1 + \xi_1(x_1) A_1 y_d + \varepsilon_1 \\ &= c_{\theta 1}\xi_1(x_1)\omega_1 + \xi_1(x_1) A_1 y_d + \varepsilon_1 \end{aligned} \tag{7}$$

where ε_1 is a parameter denoting approximating accuracy, and $c_{\theta 1} = \|A_1\|$, $A_1^m = A_1/\|A_1\|$. Thus one has $\|A_1^m\| \le 1$ and $\omega_1 = A_1^m z_1$.

Substituting (7) into (6) , we get

$$\dot{z}_1 = g_1(x_1)x_2 + c_{\theta 1}\xi_1(x_1)\omega_1 + v_1 - \dot{y}_d \tag{8}$$

where $v_1 = \xi_1(x_1) A_1 y_d + \varepsilon_1$ and $c_{\theta 1}$ is an unknown constant, and there exists a bound for v_1 as follows

$$\|v_1\| \le \|\xi_1(x_1) A_1 y_d + \varepsilon_1\| \le b_{min}\theta_1\psi_1(x_1) \tag{9}$$

where $\theta_1 = b_{min}^{-1}\max(\|A_1 y_d\|, \|\varepsilon_1\|)$ and $\psi_1(x_1) = 1 + \|\xi_1\|$.

Now we choose a virtual controller α_2 for x_2 in the subsystem (8) and the update law for $\hat{\lambda}_1$ as

$$\alpha_2 = -\left(k_1 + \hat{\lambda}_1\Phi_1(x_1)\right)z_1 + \dot{y}_d \tag{10}$$

$$\dot{\hat{\lambda}}_1 = \Gamma_1[\Phi_1(x_1)z_1^2 - \sigma_1(\hat{\lambda}_1 - \lambda_1^0)] \tag{11}$$

where k_1, Γ_1, σ_1 and λ_1^0 are positive design constants. $\hat{\lambda}_1$ are the estimates of $\lambda_1 = b_{min}^{-1}max(c_{\theta 1}^2, {\theta_1}^2)$. $\Phi_1(x_1)$ and $\Phi_i(\bar{x}_i)$, $i = 2, ..., n$ in the sequel are defined after Eq.(34).

Introduce a new variable s_2 and let α_2 pass through a first-order filter with time constant η_2

$$\eta_2\dot{s}_2 + s_2 = \alpha_2,\ \ s_2(0) = \alpha_2(0). \tag{12}$$

Step i $(2 \le i \le n-1)$: A similar procedure is employed recursively for each step $i\,(2 \le i \le n-1)$. Define the ith error variable $z_i = x_i - s_i$, and we have

$$\dot{z}_i = g_i(\bar{x}_i)x_{i+1} + f_i(\bar{x}_i) - \dot{s}_i \tag{13}$$

We also use RBF NN to approximate the unknown function $f_i(\bar{x}_i)$ leading to

$$\begin{aligned} f_i(\bar{x}_i) &= \xi_i(\bar{x}_i) A_i \bar{x}_i^T + \varepsilon_i \\ &= \xi_i A_i \begin{bmatrix} z_1 + y_d \\ z_2 + s_2 \\ \vdots \\ z_i + s_i \end{bmatrix}^T + \varepsilon_i = c_{\theta i}\xi_i\omega_i + v_i \end{aligned} \tag{14}$$

where $c_{\theta i} = \|A_i^1\| = \lambda_{\max}^{1/2}\left(A_i^{1T}A_i^1\right)$, $A_i^m = A_i^1/\|A_i^1\|$, thus $\|A_i^m\| \le 1$ and $\omega_i = A_i^m\bar{z}_i$. $v_i = \xi_i A_i^1 y_d + \xi_i\sum\limits_{j=2}^{i} A_i^j s_j + \varepsilon_i$.

Then (13) can be converted as follows

$$\dot{z}_i = g_i(\bar{x}_i)x_{i+1} + c_{\theta i}\xi_i\omega_i + v_i - \dot{s}_i \tag{15}$$

and note that

$$\|v_i\| \le \left\|\xi_i A_i^1 y_d + \xi_i\sum_{j=2}^{i} A_i^j s_j + \varepsilon_i\right\| \le b_{min}\theta_i\psi_i \tag{16}$$

with $\theta_i = b_{min}^{-1}\max\left(\|A_i^1 y_d\|, \left\|\sum\limits_{j=2}^{i} A_i^j s_j\right\|, \|\varepsilon_i\|\right)$, $\psi_i = 1 + \|\xi_i\|$.

Similarly, choose a virtual controller α_{i+1} and the update law for $\hat{\lambda}_i$ as follows

$$\alpha_{i+1} = -\left(k_i + \hat{\lambda}_i\Phi_i(\bar{x}_i)\right)z_i + \dot{s}_i \tag{17}$$

$$\dot{\hat{\lambda}}_i = \Gamma_i[\Phi_i(\bar{x}_i)z_i^2 - \sigma_i(\hat{\lambda}_i - \lambda_i^0)] \tag{18}$$

where k_i, Γ_i, σ_i and λ_i^0 are positive design constants. $\hat{\lambda}_i$ are the estimates of $\lambda_i = b_{min}^{-1}max(c_{\theta_i}^2, {\theta_i}^2)$.

Now, introduce a variable s_{i+1} and let α_{i+1} pass through a first-order filter with time constant η_{i+1}

$$\eta_{i+1}\dot{s}_{i+1} + s_{i+1} = \alpha_{i+1},\ \ s_{i+1}(0) = \alpha_{i+1}(0). \tag{19}$$

Step n: Define the nth error variable $z_n = x_n - s_n$, then

$$\dot{z}_n = g_n(x)u + f_n(x) - \dot{s}_n \tag{20}$$

Similarly, $f_n(x)$ can be expressed as

$$\begin{aligned} f_n(x) &= \xi_n(x) A_n x^T + \varepsilon_n \\ &= c_{\theta n}\xi_n\omega_n + v_n \end{aligned} \tag{21}$$

where $c_{\theta n} = \|A_n^1\| = \lambda_{\max}^{1/2}\left(A_n^{1T}A_n^1\right)$, $A_n^m = A_n^1/\|A_n^1\|$, thus $\|A_n^m\| \le 1$ and $\omega_n = A_n^m z$ with $z = [z_1, ..., z_n]^T$. $v_n = \xi_n A_n^1 y_d + \xi_n\sum\limits_{j=2}^{n} A_n^j s_j + \varepsilon_n$. Then one has

$$\dot{z}_n = g_n(x)u + c_{\theta n}\xi_n\omega_n + v_n - \dot{s}_n \tag{22}$$

and

$$\|v_n\| \le \left\|\xi_n A_n^1 y_d + \xi_n\sum_{j=2}^{n} A_n^j s_j + \varepsilon_n\right\| \le b_{min}\theta_n\psi_n \tag{23}$$

with $\theta_n = b_{min}^{-1}\max\left(\|A_n^1 y_d\|, \left\|\sum\limits_{j=2}^{n} A_n^j s_j\right\|, \|\varepsilon_n\|\right)$, $\psi_n = 1 + \|\xi_n\|$.

Now, we choose the control input u and the update law for $\hat{\lambda}_n$ as follows

$$u = -\left(k_n + \hat{\lambda}_n\Phi_n(x)\right)z_n + \dot{s}_n \tag{24}$$

$$\dot{\hat{\lambda}}_n = \Gamma_n[\Phi_n(x)z_n^2 - \sigma_n(\hat{\lambda}_n - \lambda_n^0)] \tag{25}$$

where k_n, Γ_n, σ_n and λ_n^0 are positive design constants. $\hat{\lambda}_n$ are the estimates of $\lambda_n = b_{min}^{-1}max(c_{\theta_n}^2, {\theta_n}^2)$.

Remark 2: It can be observed from the form of the virtual controllers $\alpha_{i+1}, i = 1, ..., n-1$ and the controller u proposed in this paper, together with the properties of equation (34), that we does not estimate the unknown gain functions $g_i(x), i = 1, ..., n$. In such a way we can not only avoid the possible controller singularity problem usually met with feedback linearization design when the adaptive NN controller is executed, but also removed the number of parameters needed to be updated on-line for $\hat{g}_i(x), i = 1, ..., n$.

Remark 3: It is worth noting in our algorithm that, the RBF NNs are only used to approximate those unstructured system functions, especially, the number of parameters updated on line for each RBF NN in each subsystem is reduced to only one, which thus avoid the problem of "dimension curse". In addition, the proposed scheme avoids both problems of "dimension curse" and "explosion of complexity" simultaneously, which result in a minimal learning parameterizations algorithm with a much simpler structure. Consequently, the computation burden of the adaptive control algorithm is reduced drastically and the algorithm is easy to be implemented in applications.

B. Stability Analysis

Define new error variables

$$y_{i+1} = s_{i+1} - \alpha_{i+1}, i = 1, 2, \ldots, n-1 \tag{26}$$

Note that $\dot{s}_i = -(s_i + \alpha_i)/\eta_i = -y_i/\eta_i$, then

$$\begin{aligned} \dot{y}_2 &= \dot{s}_2 - \dot{\alpha}_2 \\ &= -\tfrac{y_2}{\eta_2} + \left(-\tfrac{\partial\alpha_2}{\partial z_1}\dot{z}_1 - \tfrac{\partial\alpha_2}{\partial x_1}\dot{x}_1 - \tfrac{\partial\alpha_2}{\partial\hat{\lambda}_1}\dot{\hat{\lambda}}_1 + \ddot{y}_r\right) \\ &= -\tfrac{y_2}{\eta_2} + B_2\left(z_1, z_2, y_2, \hat{\lambda}_1, y_r, \dot{y}_r, \ddot{y}_r\right) \end{aligned} \tag{27}$$

Obviously, $B_2(\cdot)$ is a continuous function with respect to variables $\left(z_1, z_2, y_2, \hat{\lambda}_1, y_r, \dot{y}_r, \ddot{y}_r\right)$.

Similarly, we have

$$\begin{aligned} &\dot{y}_{i+1} = \dot{s}_{i+1} - \dot{\alpha}_{(i+1)} = -\tfrac{y_{i+1}}{\eta_{i+1}} + \\ &B_{i+1}\left(z_1, \ldots, z_{i+1}, y_2, \ldots, y_i, \hat{\lambda}_1, \ldots, \hat{\lambda}_i, y_r, \dot{y}_r, \ddot{y}_r\right) \end{aligned} \tag{28}$$

where $i = 2, \ldots, n-1$.

Consider $x_{i+1} = z_{i+1} + s_{i+1}$ and $s_{i+1} = y_{i+1} + \alpha_{i+1}$, the overall error systems can be expressed as

$$\begin{aligned} &\dot{z}_1 = g_1 z_2 + g_1 y_2 + g_1\alpha_2 + c_{\theta 1}\xi_1(x_1)\omega_1 + v_1 - \dot{y}_d \\ &\dot{z}_i = g_i z_{i+1} + g_i y_{i+1} + g_i\alpha_{i+1} + c_{\theta i}\xi_i(\bar{x}_i)\omega_i + v_i - \dot{s}_i \\ &\vdots \qquad i = 2, \ldots, n-1, \\ &\dot{z}_n = g_n u + c_{\theta n}\xi_n(x)\omega_n + v_n - \dot{s}_n \end{aligned} \tag{29}$$

We are now in a position to state our main result on semiglobal stable robust adaptive NN controller.

Theorem 1: Consider the closed-loop system composed of (27)∼(29), the controllers (10),(17) and (24), and the updated laws (11),(18) and (25), given any positive number p, for all initial conditions satisfying $\Pi := \left\{\sum_{j=1}^n \left(z_j^2 + \tilde{\lambda}_j^T b_{min}\Gamma_{j2}^{-1}\tilde{\lambda}_j\right) + \sum_{j=2}^n y_j^2 < 2p\right\}$, $i = 1, \ldots, n$, there exist k_i, γ_i, δ_i, η_i, σ_i and Γ_i, such that the solution of the closed-loop control system is uniformly ultimately bounded. Furthermore, given any $\mu > 0$, we can tune our controller parameters such that the output error $z_1 = y(t) - y_d(t)$ satisfies $\lim_{t\to\infty}|z_1(t)| \le \mu$.

Proof: Choosing the Lyapunov function candidate as

$$V = \frac{1}{2}\sum_{i=1}^n \left(z_i^2 + \tilde{\lambda}_i^T b_{min}\Gamma_i^{-1}\tilde{\lambda}_i\right) + \frac{1}{2}\sum_{i=1}^{n-1} y_{i+1}^2 \tag{30}$$

where $\tilde{\theta}_i = \theta_i - \hat{\theta}_i$, $\tilde{\lambda}_i = \lambda_i - \hat{\lambda}_i$, Γ_{i1} and $\Gamma_{i2}, i = 1, 2, \ldots, n$ are positive definite matrix to be determined later. The time derivative of V along the system trajectories is

$$\begin{aligned} \dot{V} =& \sum_{i=1}^n \left(z_i\dot{z}_i - \tilde{\lambda}_i^T b_{\min}\Gamma_i^{-1}\dot{\hat{\lambda}}_i\right) + \sum_{i=1}^{n-1} y_{i+1}\dot{y}_{i+1} \\ \le& \sum_{i=1}^{n-1}\Big(-b_{\min}k_i z_i^2 + g_i z_{i+1}z_i + c_{\theta i}\xi_i(\bar{x}_i)w_i z_i + v_i z_i \\ &- b_{\min}\hat{\lambda}_i\Phi_i(\bar{x}_i)z_i^2 - \tilde{\lambda}_i^T b_{\min}\Gamma_i^{-1}\dot{\hat{\lambda}}_i + g_i y_{i+1}z_i\Big) \\ &- b_{\min}k_n z_n^2 + \sum_{i=2}^n (g_i\dot{s}_i z_i - \dot{s}_i z_i) + g_1\dot{y}_d z_1 \\ &- \dot{y}_d z_1 + c_{\theta n}\xi_n(x)w_n z_n + v_n z_n - b_{\min}\hat{\lambda}_n\Phi_n(x)z_n^2 \\ &- \tilde{\lambda}_n^T b_{\min}\Gamma_n^{-1}\dot{\hat{\lambda}}_n + \sum_{i=1}^{n-1}\left(-\frac{y_{i+1}^2}{\eta_{i+1}} + |y_{i+1}B_{i+1}|\right) \end{aligned} \tag{31}$$

It is noted that

$$\begin{aligned} c_{\theta i}\xi_i(\bar{x}_i)w_i z_i =& c_{\theta i}\xi_i(\bar{x}_i)w_i z_i - \gamma_i^2 w_i^T w_i + \gamma_i^2 w_i^T w_i \\ \le& \frac{c_{\theta i}^2}{4\gamma_i^2}\xi_i\xi_i^T z_i^2 + \gamma_i^2 w_i^T w_i \end{aligned} \tag{32}$$

and

$$\nu_i z_i \le \theta_i\psi_i(\bar{x}_i) \parallel z_i \parallel \le \frac{\theta_i^2}{4\iota_i^2}\psi_i^2(\bar{x}_i)z_i^2 + \iota_i^2 \tag{33}$$

with ι_i and γ_i being any given positive constant, and $\psi_i(\bar{x}_i) = 1 + \parallel \xi_i \parallel$.

Combining (32) and (33) yields

$$\begin{aligned} &c_{\theta i}\xi_i w_i z_i + \nu_i z_i \\ \le& b_{\min}\lambda_i\Phi_i(\bar{x}_i)z_i^2 + \gamma_i^2\omega_i^T\omega_i + \iota_i^2 \\ \le& b_{\min}\hat{\lambda}_i\Phi_i(\bar{x}_i)z_i^2 + b_{\min}\widetilde{\lambda}_i\Phi_i(\bar{x}_i)z_i^2 + \gamma_i^2\omega_i^T\omega_i + \iota_i^2 \end{aligned} \tag{34}$$

where $\Phi_i(\bar{x}_i) = \frac{1}{4\gamma_i^2}\xi_i\xi_i^T + \frac{1}{4\iota_i^2}\psi_i^2$, $\lambda_i = \max(b_{\min}^{-1}c_{\theta 1}^2, b_{\min}^{-1}\theta_i^2)$, $\widetilde{\lambda}_i = (\lambda_i - \hat{\lambda}_i)$ and $\hat{\lambda}_i$ is the estimate of λ_i.

And noting that

$$g_i\dot{s}_i z_i - \dot{s}_i z_i \le \frac{1 + b_{max}}{4\eta_i}z_i^2 + \frac{b_{max} + 1}{\eta_i}y_i^2, \tag{35}$$

$$g_1\dot{y}_d z_1 - \dot{y}_d z_1 \le \frac{b_{max} + 1}{4}z_1^2 + (b_{max} + 1)B_0^2, \tag{36}$$

and the fact that $g_i z_{i+1} z_i \le z_i^2 + \frac{b_{max}}{4}z_{i+1}^2$ and $g_i y_{i+1} z_i \le z_i^2 + \frac{b_{max}}{4}y_{i+1}^2$, then (31) becomes

$$\begin{aligned}\dot{V} \leq & \sum_{i=2}^{n-1}\left(-\left(b_{min}k_i - 2 - \frac{1+b_{max}}{\eta_i}\right)z_i^2 + \frac{b_{max}}{4}z_{i+1}^2\right) \\ & -(b_{\min}k_1 - 2 - \frac{b_{max}+1}{4})z_1^2 \\ & -(b_{min}k_n - \frac{1+b_{max}}{\eta_i})z_n^2 + \sum_{i=1}^{n}\left(\gamma_i^2\omega_i^T\omega_i + \delta_i'\right) \\ & -\sum_{i=1}^{n}\left(\frac{\sigma_i}{2\lambda_{\max}\left(b_{min}\Gamma_i^{-1}\right)}\tilde{\lambda}_i^T\Gamma_i^{-1}\tilde{\lambda}_i\right) \\ & +\sum_{i=1}^{n-1}\left(\frac{b_{max}}{4}y_{i+1}^2 - \frac{3-b_{max}}{4\eta_{i+1}}y_{i+1}^2 + |y_{i+1}B_{i+1}|\right)\end{aligned} \tag{37}$$

where $\delta_i' = (b_{max}+1)B_0^2 + b_{max}\delta_i + \frac{\sigma_i}{2}\left|\lambda_i^* - \lambda_i^0\right|^2$.

Now we investigate the characteristics of B_{i+1}, $i = 1, 2, \ldots, n-1$.

Since the sets $\Pi_0 \in \mathrm{R}^3$ and $\Pi_i \in \mathrm{R}^{\left(\sum_{j=1}^{i} N_j + 2i - 1\right)}$, where N_j is the dimension of $\tilde{\theta}_j$, are compact, $\Pi_0 \times \Pi_i \in \mathrm{R}^{\left(\sum_{j=1}^{i} N_j + 2i + 2\right)}$ is also compact. Therefore, $|B_{i+1}|$ has a maximum M_{i+1} on $\Pi_0 \times \Pi_i$.

Let $\frac{1}{\eta_{i+1}} = (\frac{3-b_{max}}{4})^{-1}(\frac{b_{max}}{4} + \frac{M_{i+1}^2}{2\alpha} + \alpha_0)$, and note that $|B_{i+1}y_{i+1}| \leq \frac{y_{i+1}^2 B_{i+1}^2}{2\alpha} + \frac{\alpha}{2}$, where α_0 and α are positive constants. Then we arrive at

$$\left(\frac{b_{max}}{4} - \frac{3-b_{max}}{4\eta_{i+1}}\right)y_{i+1}^2 + |B_{i+1}y_{i+1}| \leq -\alpha_0 y_{i+1}^2 + \frac{\alpha}{2} \tag{38}$$

Let $\sigma_{i1}/2\lambda_{\max}\left(b_{min}\Gamma_{i1}^{-1}\right) = \sigma_{i2}/2\lambda_{\max}\left(b_{min}\Gamma_{i2}^{-1}\right) = \alpha_0$, and $k_1 = b_{min}^{-1}(2 + \frac{b_{max}+1}{4} + \alpha_0)$, $k_i = b_{min}^{-1}(2 + \frac{1+b_{max}}{4\eta_i} + \frac{b_{max}}{4} + \alpha_0)(i = 2, \ldots, n-1)$, $k_n = b_{min}^{-1}(\frac{b_{max}+1}{4\eta_i} + \alpha_0)$, then (37) can be further expressed as

$$\begin{aligned}\dot{V} \leq & -\alpha_0\sum_{i=1}^{n} z_i^2 - \alpha_0\sum_{i=1}^{n}\left(\tilde{\lambda}_i^T b_{min}\Gamma_i^{-1}\tilde{\lambda}_i\right) \\ & -\alpha_0\sum_{i=1}^{n-1} y_{i+1}^2 + \sum_{i=1}^{n}\left(\gamma_i^2\omega_i^T\omega_i\right) + \rho \\ \leq & -2\alpha_0 V + \gamma^2\|\omega\|^2 + \rho\end{aligned} \tag{39}$$

where $\rho = \sum\limits_{i=1}^{n}(\delta_i') + \sum\limits_{i=1}^{n-1}(\alpha/2)$, $\gamma = (\gamma_1^2 + \gamma_2^2 + \cdots + \gamma_n^2)^{1/2}$, $\omega = [\omega_1, \omega_2, ..., \omega_n]^T$.

Note that $\omega_i = A_i^m \bar{z}_i^T$ and $\|A_i^m\| \leq 1$, $i = 1, ..., n$, so we obtain

$$\omega = \begin{bmatrix}\omega_1 \\ \omega_2 \\ \vdots \\ \omega_n\end{bmatrix} = \begin{bmatrix} A_1^m & 0 & \cdots & 0 \\ A_2^{m1} & A_2^{m2} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ A_n^{m1} & A_n^{m2} & \cdots & A_n^{mn}\end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n\end{bmatrix} = Az$$

and

$$\| \omega \| \leq \| A \| \| z \| \leq \| z \| \tag{40}$$

Now, if choosing $\gamma < 1$, then (39) can be converted into

$$\dot{V} \leq -2\alpha_0 V + \|z\|^2 + \rho \leq -c_1 V + \rho \tag{41}$$

where $c_1 = (2\alpha_0 - 1)$.

From (41), we obtain

$$V(t) \leq \frac{\rho}{c_1} + \left(V(t_0) - \frac{\rho}{c_1}\right)e^{-(t-t_0)}$$

It follows that , for any $\mu_1 > (\rho/c_1)^{1/2}$, there exists a constant $T > 0$ such that $\|z_1(t)\| \leq \mu_1$ for all $t \geq t_0 + T$, and the tracking error can be made small since $(\rho/c_1)^{1/2}$ can be made arbitrarily small if the design parameters γ_1, δ_1, η_2, σ_1 and Γ_1 are chosen appropriately. Theorem 1 is thus proved.

■

IV. APPLICATION EXAMPLES

In this section, we will present an example of an one-link robot system with the inclusion of motor dynamics to reveal the control performance of the proposed algorithm.

Consider an one-link manipulator with the inclusion of motor dynamics[13]. The robot model is

$$\begin{cases} D\ddot{q} + B\dot{q} + N sin(q) = \tau + \tau_d \\ M\dot{\tau} + H\tau = u - K_m\dot{q}\end{cases} \tag{42}$$

where q, $\dot{q}$ and $\ddot{q}$ denote the link position, velocity and acceleration, respectively. τ and $\dot{\tau}$ are the motor shaft angle and velocity. τ_d represents the torque disturbance. u is the control input used to represent the motor torque. Eq.(42) can be expressed in the form (1) by noting that

$x_1 = q$, $x_2 = \dot{q}$, $x_3 = \frac{\tau}{D}$, $f_1(x_1, w) = 0$, $f_2(x_1, x_2, w) = \frac{-N sin(x_1) - Bx_2}{D}$, $f_3(x_1, x_2, x_3, w) = \frac{-K_m x_2 - HDx_3}{MD}$, $d_1 = 0$, $d_2 = \frac{\tau_d}{D}$, $d_3 = 0$.

The parameter values with appropriate units are given by $D = 1$, $B = 1$, $M = 0.05$, $H = 0.5$, $N = 10$, $K_m = 10$. The torque disturbance $\tau_d = sin(t)$.

In simulation, we choose Gaussian function as the form aforementioned, the RBF NNs for f_2 and f_3 contain 25 nodes, respectively, with centers evenly spaced in $[-2.5, 2.5] \times [-2, 2]$ for f_2 and $[-2.5, 2.5] \times [-2, 2] \times [-2, 2]$ for f_3 and widths $a_{i,l} = 5$, $i = 2, 3$, $l = 1, ..., 25$. The initial conditions for x_1, x_2 and x_3 are 0.2, 2π and 0.

Remark 4: We found a fact in the simulation that the nodes number in each RBF NN does not have obvious effect on the controller performance and the time consumed when executing the algorithm proposed in this paper. For example, the control performance and the consumed time of the controller with fewer nodes(such as five nodes) are similar to those of the controller with much more nodes(such as twenty-five or one-hundred nodes). This is one of the significant advantages of our algorithm against those NN-based adaptive control schemes in the literature.

Figs.1∼2 illustrate the simulation results of the one-link robot system for tracking a reference signal $y_d = sin(\pi t)$.

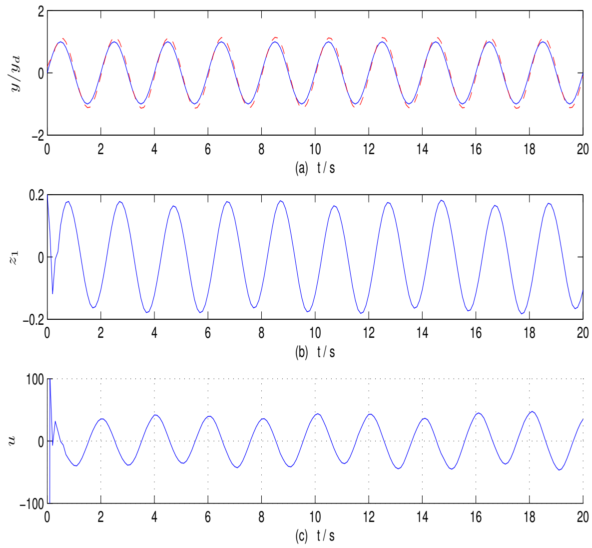

Fig. 1. Simulation results for one-link robot system: (a) system output y (dot line) and reference signal y_d (solid line), (b) tracking error z_1, (c) control signal u.

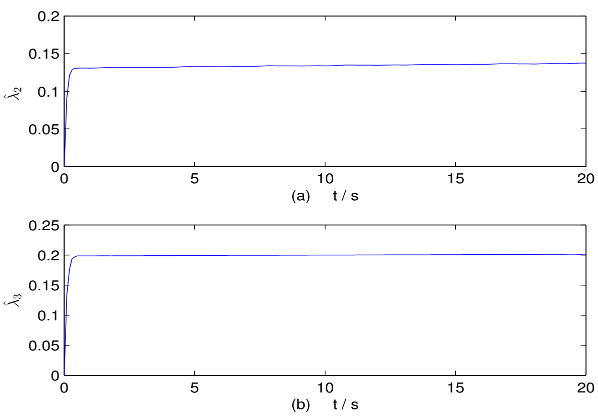

Fig. 2. Simulation results for one-link robot system: (a) parameter estimate λ_2; (b) parameter estimate λ_3.

V. CONCLUSION

In this paper, the tracking control problem has been considered for a class of strict-feedback uncertain nonlinear systems. Combining the DSC technique with the MLP algorithms, an adaptive NN tracking control scheme is developed based on Lyapunov direct method. It is shown that the closed-loop system is SGUUB. The main features of the proposed algorithms are that the adaptive mechanism with minimal learning parameterizations is achieved, and both problems of "explosion of complexity" and "curse of dimension", as well as the possible controller singularity problem in some of the existing adaptive control schemes with feedback linearization techniques, are circumvented. The proposed adaptive control algorithm is in a much simpler form and its computation load is reduced dramatically, and thus it is much easier to implement this algorithm for applications.

REFERENCES

[1] M. Krstic, I. Kanellakopoulos and P. V. Kokotovic, *Nonlinear and adaptive control design*. Wiley, New York. 1995.

[2] A. Isidori, *Nonlinear Control Systems*, Berlin: Springer-Verlag, 1989.

[3] R. Marino and P. Tomei, "Global adaptive output-feedback control of nonlinear systems, part I: Linear parameterization; part II: nonlinear parameterization", *IEEE Trans. on Automatic Control*, 1993, 38: 33-49.

[4] S. S. Ge, C. C. Hang and T. Zhang, *Stable adaptive neural network control*, Kluwer Academic Publishers, 2002.

[5] T P Zhang, S S Ge. "Direct Adaptive NN Control of Nonlinear Systems in Strict-Feedback Form Using Dynamic Surface Control", *IEEE 22nd Int. Symposium on Intelligent Control*, 1-3 Oct. 2007, p315-320.

[6] C. Wang and D. J. Hill, "Learning from neural control", *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 2006, 17 (1): 130-146.

[7] J. Zhou, J. E Meng and J. M. Zurada, "Adaptive neural network control of uncertain nonlinear systems with nonsmooth actuator nonlinearities", *Neurocomputing*, 2007,70(4-6): 1062-1070.

[8] W. Y. Wang, M. L. Chan, T. T. Lee and C. H. Liu, "Adaptive fuzzy control for strict-feedback cononical nonlinear systems with H_∞ tracking performance", *IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics*, 2000, 30: 878-885.

[9] T. S. Li, Y.S. Yang, J.Q. Hu and L.J. Yang. "Robust Adaptive Fuzzy Tracking Control for a Class of Perturbed Uncertain Nonlinear Systems with UVCGF". *Int. J. Wavelets, Multiresolution and Information Processing* , 2007, 5(1): 227-239

[10] G. Feng, S.G. Cao, and N.W. Rees, "Stable Adaptive Control of Fuzzy Dynamic Systems", *Fuzzy Sets and Systems*, 2002, 131: 217-224.

[11] Y. S. Yang and J. S. Ren, "Adaptive fuzzy robust tracking controller design via small gain approach and its application,"*IEEE Trans. Fuzzy Syst.*, 2003, 11(6): 783-795.

[12] Y. S. Yang, C. J. Zhou and T. S. Li , "Small gain method for adaptive robust fuzzy control of a class of nonlinear systems", *Proc. of the 2003 IEEE Int. Symp. on Intel. Control*, Houston, Texas, 2003, p40-45.

[13] Y. S. Yang, G. Feng and J. S. Ren, "A combined backstepping and small-gain approach to robust adaptive fuzzy control for strict-feedback nonlinear systems", *IEEE Trans. SMC-Part A: Syst. and Humans,* 2004, 34(3): 406-420.

[14] Y. S. Yang, T. S. Li, and X. F. Wang. "Robust adaptive neural network control for strict-feedback nonlinear systems via small-gain approaches". *Lecture Notes in Computer Science*, 2006, vol.3972, p888-897.

[15] T. S. Li, Z. J. Zou, and X. M. Zhou. "Adaptive NN Control for a Class of Strict-feedback Nonlinear Systems". //Proceedings of the 2008 American Control Conference(ACC08). June 11-13, 2008, Seattle, Washington, USA: IEEE Press, 2008, p81-86.

[16] D. Swaroop, J. K. Hedrick, P. P. Yip, and J. C. Gerdes. "Dynamic surface control of nonlinear systems", *IEEE Trans. on Automatic Control*, 2000, 45(10): 1893-1899

[17] P. P. Yip and J. K.Hedrick. "Adaptive dynamic surface control: a simplified algorithm for adaptive backstepping control of nonlinear systems". *Int. J. Control*, 1998, 71(5): 959-979.

[18] D. Wang and J. Huang. "Neural Network-Based Adaptive Dynamic Surface Control for a Class of Uncertain Nonlinear Systems in Strict-Feedback Form". *IEEE Trans. on neural networks*, 2005, 16(1): 195-202.