Brief Papers

Direct Adaptive NN Control of a Class of Nonlinear Systems

Shuzhi S. Ge and Cong Wang

***Abstract*—In this paper, direct adaptive neural-network (NN) control is presented for a class of affine nonlinear systems in the strict-feedback form with unknown nonlinearities. By utilizing a special property of the affine term, the developed scheme avoids the controller singularity problem completely. All the signals in the closed loop are guaranteed to be semiglobally uniformly ultimately bounded and the output of the system is proven to converge to a small neighborhood of the desired trajectory. The control performance of the closed-loop system is guaranteed by suitably choosing the design parameters. Simulation results are presented to show the effectiveness of the approach.**

***Index Terms*—Adaptive control, backstepping, neural control, neural network (NN), uncertain strict-feedback system.**

I. Introduction

In recent years, adaptive neural control schemes have been found to be particularly useful for the control of nonlinear uncertain systems with unknown nonlinear functions. In the literature of adaptive neural control, neural networks (NNs) are primarily used as on-line approximators for the unknown nonlinearities due to their inherent approximation capabilities. With the help of NN approximation, it is not necessary to spend much effort on system modeling in case such a modeling is of great difficulty. In the earlier NN control schemes, optimization techniques were mainly used to derive parameter adaptation laws, which lack for analytical results about stability and performance. To overcome these problems, several elegant adaptive NN control approaches have been proposed based on Lyapunov's stability theory [2]–[12]. One main advantage of these schemes is that the parameter adaptation laws are derived based on Lyapunov synthesis and therefore stability of the closed-loop system is guaranteed. However, one limitation of these schemes is that they can only applied to nonlinear systems where certain types of matching conditions are required to be satisfied.

By using the idea of backstepping design [1], several interesting neural-based adaptive controllers [13]–[16] have been presented for some classes of uncertain nonlinear strict-feedback systems without the requirement of matching conditions

$$\begin{aligned}\dot{x}_i &= f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1}, \quad 1 \le i \le n-1\\ \dot{x}_n &= f_n(\bar{x}_n) + g_n(\bar{x}_n)u, \quad n \ge 2\\ y &= x_1\end{aligned} \tag{1}$$

where $\bar{x}_i = [x_1, \ldots, x_i]^T \in R^i, i = 1, \ldots, n, u \in R, y \in R$ are state variables, system input and output, respectively. In [14], an indirect adaptive NN control scheme was presented for system (1) with $f_i(\bar{x}_i), i = 1, \ldots, n$ being unknown smooth functions, $g_i(\bar{x}_i) = 1, i = 1, \ldots, n-1$, and $g_n(\bar{x}_n) = g$ being an unknown constant. The unknown smooth functions $f_i(\bar{x}_i), i = 1, \ldots, n$ are first approximated on-line by neural networks, then a stabilizing controller is constructed based on the approximation. In [15], a neural controller is proposed for a class of unknown, minimum phase, feedback linearizable nonlinear system with known relative degree. In [16], through the definition of a novel integral-type Lyapunov function, a direct adaptive neural control approach was proposed for the first time for strict-feedback system (1), where $f_i(\cdot)$ and $g_i(\cdot), i = 1, \ldots, n$ are all unknown smooth functions. In the approach, a desired feedback control law is first proved to be in existence, then neural networks are used to parameterize the desired feedback control law, finally adaptive techniques are used to tune the weights of neural networks for closed-loop stability. The possible controller singularity problem usually met in adaptive control is completely overcome.

Note that for the control of the uncertain strict-feedback system (1), one of the main difficulties comes from the uncertain affine terms $g_i(\cdot)(i = 1, \ldots, n)$. When $g_i(\cdot)(i = 1, \ldots, n)$ are known exactly, while $f_i(\cdot)(i = 1, \ldots, n)$ are composed of dominant uncertain nonlinearities and/or parametric uncertainties, both robust adaptive backstepping design (e.g., [1], [17]–[20]) and neural control scheme provided in [13] are applicable. When $g_i(\cdot)(i = 1, \ldots, n)$ are unknown nonlinearities, if feedback linearization type controllers $\alpha_i = (1/\hat{g}_i(\cdot))(-\hat{f}_i(\cdot) + \upsilon_i)$ are considered, where $\hat{f}_i(\cdot)$ and $\hat{g}_i(\cdot)$ are the estimates of $f_i(\cdot)$ and $g_i(\cdot)$, respectively, and υ_i is a new control to be defined, difficulties arise when $\hat{g}_i(\cdot) \to 0$, which is referred to the controller singularity problem. To the authors' knowledge, without using neural networks or fuzzy logic, or any other universal approximators, there is no results available in the literature to control the uncertain strict-feedback system (1). By combining adaptive neural design with backstepping methodology, a smooth adaptive neural controller was proposed in [16], where integral-type Lyapunov functions are introduced and play an important role in overcoming this singularity problem. However, due to the integral operation, this approach is complicated and difficult to use in practice.

In this paper, motivated by the fact that better results can be obtained when more properties of the studied systems are exploited, a direct adaptive NN control scheme is presented for nonlinear strict-feedback system (1), where $f_i(\cdot)$

Manuscript received May 18, 2000; revised February 26, 2001 and September 27, 2001.

The authors are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576, Singapore (e-mail: elegesz@nus.edu.sg).

Publisher Item Identifier S 1045-9227(02)00356-9.

and $g_i(\cdot), i = 1, \ldots, n$ are unknown smooth functions, and moreover, the affine term $g_n(\cdot)$ is assumed to be independent of state x_n [11]. By exploiting this property, the developed neural control scheme avoids the controller singularity problem completely, and the stability of the resulting adaptive system is guaranteed without the requirement for integral-type Lyapunov functions. On the other hand, since differentiation of the virtual controls has to be performed in backstepping design, the differentiation requirement becomes a technical problem when designing controllers for the uncertain strict-feedback system (1). The derivatives of virtual controls will involve the uncertainties $f_i(\cdot)$ and $g_i(\cdot)$, and thus are unknown nonlinear functions which are unavailable for implementation. Though these derivatives are smooth functions of the system states and neural weight estimates, and can also be approximated by NNs, the approximation might become computationally unacceptable when the large number of neural weight estimates are taken as inputs to the NNs. By introducing some intermediate variables as inputs to NNs, where the intermediate variables are available through the computation of system states and neural weight estimates, the NN approximation can be implemented by using the minimal number of NN inputs. Hence, the differentiation requirement can be satisfied by using NNs to approximate all the uncertainties, including the derivatives of virtual controls, while intermediate variables are introduced to make the NN approximation computationally feasible.

With the proposed neural control scheme, semiglobal uniform ultimate boundedness of all the signals in the closed-loop are guaranteed, and the output of the system is proven to converge to a small neighborhood of the desired trajectory. The control performance of the closed-loop system is guaranteed by suitably choosing the design parameters. The proposed direct adaptive NN controller can be applied to a large class of nonlinear strict-feedback systems in the form (1) without repeating the complex controller design procedure for different system nonlinearities. The rest of the paper is organized as follows: The problem formulation is presented in Section II. In Section III, a direct adaptive NN controller is presented for controlling uncertain nonlinear system (1). Simulation results performed on an illustrative example are demonstrated to show the effectiveness of the approach in Section IV. Section V contains the conclusion.

II. Problem Formulation

The control objective is to design a direct adaptive NN controller for system (1) such that 1) all the signals in the closed-loop remain semiglobally uniformly ultimately bounded and 2) the output y follows a desired trajectory y_d generated from the following smooth bounded reference model:

$$\begin{aligned} \dot{x}_{di} &= f_{di}(x_d), \quad 1 \le i \le m \\ y_d &= x_{d1}, \quad m \ge n \end{aligned} \tag{2}$$

where $x_d = [x_{d1}, x_{d2}, \ldots, x_{dm}]^T \in R^m$ are the states, $y_d \in R$ is the system output, $f_{di}(\cdot), i = 1, 2, \ldots, m$ are known smooth nonlinear functions. Assume that the states of the reference model remain bounded, i.e., $x_d \in \Omega_d, \forall t \ge 0$.

Note that in the following derivation of the adaptive neural controller, NN approximation is only guaranteed within some compact sets. Accordingly, the stability results obtained in this work are semiglobal in the sense that, as long as the input variables of the NNs remain within some compact sets, where the compact sets can be made as large as desired, there exists controller(s) with sufficiently large number of NN nodes such that all the signals in the closed-loop remain bounded.

Since $g_i(\cdot), i = 1, \ldots, n$ are smooth functions, they are therefore bounded within some compact set. Accordingly, we can make the following assumption as commonly being done in the literature.

Assumption 1: The signs of $g_i(\cdot)$ are known, and there exist constants $g_{i1} \ge g_{i0} > 0$ such that $g_{i1} \ge |g_i(\cdot)| \ge g_{i0}, \forall \bar{x}_n \in \Omega \subset R^n$.

The above assumption implies that smooth functions $g_i(\cdot)$ are strictly either positive or negative. Without losing generality, we shall assume $g_{i1} \ge g_i(\bar{x}_i) \ge g_{i0} > 0, \forall \bar{x}_n \in \Omega \subset R^n$.

The derivatives of $g_i(\cdot)$ are given by

$$\begin{aligned} \dot{g}_i(\bar{x}_i) &= \sum_{k=1}^{i} \frac{\partial g_i(\bar{x}_i)}{\partial x_k} \dot{x}_k = \sum_{k=1}^{i} \frac{\partial g_i(\bar{x}_i)}{\partial x_k} \\ &\quad \times [g_k(\bar{x}_k) x_{k+1} + f_k(\bar{x}_k)], \quad i = 1, \ldots, n-1 \\ \dot{g}_n(\bar{x}_{n-1}) &= \sum_{k=1}^{n-1} \frac{\partial g_n(\bar{x}_{n-1})}{\partial x_k} \dot{x}_k = \sum_{k=1}^{n-1} \frac{\partial g_n(\bar{x}_{n-1})}{\partial x_k} \\ &\quad \times [g_k(\bar{x}_k) x_{k+1} + f_k(\bar{x}_k)]. \end{aligned} \tag{3}$$

Clearly, they only depend on states $\bar{x}_n$. Because $f_i(\cdot)$ and $g_i(\cdot)$ are assumed to be smooth functions, they are therefore bounded within the compact set Ω. Accordingly, we have the following assumption.

Assumption 2: There exist constants $g_{id} > 0$ such that $|\dot{g}_i(\cdot)| \le g_{id}, \forall \bar{x}_n \in \Omega \subset R^n$.

In control engineering, radial basis function (RBF) NN is usually used as a tool for modeling nonlinear functions because of their good capabilities in function approximation. The RBF NN can be considered as a two-layer network in which the hidden layer performs a fixed nonlinear transformation with no adjustable parameters, i.e., the input space is mapped into a new space. The output layer then combines the outputs in the latter space linearly. Therefore, they belong to a class of linearly parameterized networks. In this paper, the following RBF NN [22] is used to approximate the continuous function $h(Z): R^q \to R$

$$h_{\mathrm{nn}}(Z) = W^T S(Z) \tag{4}$$

where the input vector $Z \in \Omega_Z \subset R^q$, weight vector $W = [w_1, w_2, \ldots, w_l]^T \in R^l$, the NN node number $l > 1$; and $S(Z) = [s_1(Z), \ldots, s_l(Z)]^T$, with $s_i(Z)$ being chosen as the commonly used Gaussian functions, which have the form

$$s_i(Z) = \exp\left[\frac{-(Z - \mu_i)^T (Z - \mu_i)}{\eta_i^2}\right], \quad i = 1, 2, \ldots, l \tag{5}$$

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

It has been proven that network (4) can approximate any continuous function over a compact set $\Omega_Z \subset R^q$ to arbitrary any accuracy as

$$h(Z) = W^{*T}S(Z) + \epsilon, \quad \forall Z \in \Omega_Z \tag{6}$$

where W^* is ideal constant weights, and ϵ is the approximation error.

Assumption 3: There exist ideal constant weights W^* such that $|\epsilon| \le \epsilon^*$ with constant $\epsilon^* > 0$ for all $Z \in \Omega_Z$.

The ideal weight vector W^* is an "artificial" quantity required for analytical purposes. W^* is defined as the value of W that minimizes $|\epsilon|$ for all $Z \in \Omega_Z \subset R^q$, i.e.,

$$W^* \triangleq \arg\min_{W \in R^l} \left\{ \sup_{Z \in \Omega_Z} |h(Z) - W^T S(Z)| \right\}. \tag{7}$$

In the following, we let $\|\cdot\|$ denote the 2-norm, and $\lambda_{\max}(B)$ and $\lambda_{\min}(B)$ denote the largest and smallest eigenvalues of a square matrix B, respectively.

Note that the RBF NN represents a class of linearly parameterized approximators, and can be replaced by any other linearly parameterized approximators such as spline functions [21] or fuzzy systems [9]. Moreover, nonlinearly parameterized approximators, such as multilayer neural network (MNN), can be linearized as linearly parameterized approximators, with the higher order terms of Taylor series expansions being taken as part of the modeling error, as shown in [8] and [16]. The stability and performance properties of the adaptive system using nonlinearly parameterized approximators can be analyzed by combining the design procedures in [8] and [16] and those in this paper. It is omitted here for clarity and conciseness.

III. Direct Adaptive NN Control

In this section, by restricting our attention to uncertain strict-feedback system (1) where $g_n(\cdot)$ is unknown and independent of state x_n, we propose a singularity-free direct adaptive NN control scheme without using integral-type Lyapunov functions. In our scheme, direct adaptive NN design is combined with backstepping method. At each recursive step $i (1 \le i \le n)$ in backstepping design, the desired virtual control α_i^* and the desired control $u^* = \alpha_n^*$ are first shown to exist which possess some desired stabilizing properties. The desired virtual control $\alpha_i^* (i = 1, \ldots, n)$ contains uncertainties $f_i(\cdot)$ and $g_i(\cdot) (i = 1, \ldots, n)$, and thus cannot be implemented in practice. To solve this problem, the virtual control α_i and the practical control u are constructed by using RBF neural networks ${W_i}^T S_i(Z_i)$ to parameterize the unknown parts in the desired virtual control α_i^* and the desired control u^*. Then, adaptive techniques are used to update the weights of neural networks for closed-loop stability. Note that in our scheme, neural networks are not employed to approximate the unknown nonlinearities $f_i(\cdot)$ and $g_i(\cdot)$. By using the virtual control α_i, the ith-order subsystem is stabilized with respect to a Lyapunov function V_i. The control law u is designed in the last step to stabilize the whole closed-loop system with respect to an overall Lyapunov function V_n, which is the sum of all the sub-Lyapunov functions.

The detailed design procedure is described in the following steps. For clarity and conciseness of presentation, Step 1 and 2 are described with detailed explanations, while Step i and Step n are simplified, with redundant equations and explanations being omitted.

Step 1: Define $z_1 = x_1 - x_{d1}$. Its derivative is

$$\dot{z}_1 = f_1(x_1) + g_1(x_1)x_2 - \dot{x}_{d1}.$$

By viewing x_2 as a virtual control input, i.e., if we choose $\alpha_1^* \triangleq x_2$ as the control input for the z_1-subsystem in the above equation, and consider the Lyapunov function candidate $V_{z_1} = (1/2)z_1^2$, whose derivative is

$$\dot{V}_{z_1} = z_1\dot{z}_1 = z_1[f_1(x_1) + g_1(x_1)\alpha_1^* - \dot{x}_{d1}]$$

then there exists a desired feedback control

$$\alpha_1^* = -c_1 z_1 - \frac{1}{g_1}[f_1 - \dot{x}_{d1}] \tag{8}$$

where $c_1 > 0$ is a design constant, such that $\dot{V}_{z_1} = -c_1 g_1 z_1^2 \le 0$. Therefore V_{z_1} is a Lyapunov function, and $z_1 = 0$ is asymptotically stable.

Since $f_1(x_1)$ and $g_1(x_1)$ are unknown smooth functions, the desired feedback control α_1^* cannot be implemented in practice. From (8), it can be seen that the unknown part $(1/g_1(x_1))(f_1(x_1) - \dot{x}_{d1})$ is a smooth function of x_1 and $\dot{x}_{d1}$. Denote

$$h_1(Z_1) \triangleq \frac{1}{g_1(x_1)}(f_1(x_1) - \dot{x}_{d1}) \tag{9}$$

where $Z_1 \triangleq [x_1, \dot{x}_{d1}]^T \subset R^2$. By employing an RBF neural network $W_1^T S_1(Z_1)$ to approximate $h_1(Z_1), \alpha_1^*$ can be expressed as

$$\alpha_1^* = -c_1 z_1 - {W_1^*}^T S_1(Z_1) - \epsilon_1 \tag{10}$$

where W_1^* denotes the ideal constant weights, and $|\epsilon_1| \le \epsilon_1^*$ is the approximation error with constant $\epsilon_1^* > 0$. Since W_1^* is unknown, let $\hat{W}_1$ be the estimate of W_1^*.

Remark 1: The principle for designing the neural network $W_1^T S_1(Z_1)$ is to use as few neurons as possible to approximate the unknown functions $h_1(Z_1)$. Since $\dot{x}_{d1} = f_{r1}(x_d)$ is available through the computation of x_d, we use $\dot{x}_{d1} \in R$ instead of $x_d \in R^m$ as an input to NN $W_1^T S_1(Z_1)$, i.e., we define $Z_1 = [x_1, \dot{x}_{d1}]^T$, rather than $Z_1 = [x_1, x_d]^T$ to approximate $h_1(Z_1)$. Thus, minimal inputs to the neural network $W_1^T S_1(Z_1)$ are employed to implement the approximation.

Since x_2 is only taken as a virtual control, not as the real control input for the z_1-subsystem, by introducing the error variable $z_2 = x_2 - \alpha_1$ and choosing the virtual control

$$\alpha_1 = -c_1 z_1 - \hat{W}_1^T S_1(Z_1) \tag{11}$$

the $\dot{z}_1$ equation becomes

$$\begin{aligned} \dot{z}_1 &= f_1(x_1) + g_1(x_1)(z_2 + \alpha_1) - \dot{x}_{d1} \\ &= g_1(x_1)\left(z_2 - c_1 z_1 - \tilde{W}_1^T S_1(Z_1) + \epsilon_1\right) \end{aligned} \tag{12}$$

where $\tilde{W}_1 = \hat{W}_1 - W_1^*$. Through out this paper, we shall define $\tilde{(\cdot)} = \hat{(\cdot)} - (\cdot)^*$.

Consider the following Lyapunov function candidate:

$$V_1 = \frac{1}{2g_1(x_1)} z_1^2 + \frac{1}{2}\tilde{W}_1^T \Gamma_1^{-1} \tilde{W}_1 \tag{13}$$

where $\Gamma_1 = \Gamma_1^T > 0$ is an adaptation gain matrix.

The derivative of V_1 is

$$\begin{aligned}\dot{V}_1 &= \frac{z_1 \dot{z}_1}{g_1} - \frac{\dot{g}_1 z_1^2}{2g_1^2} + \tilde{W}_1^T \Gamma_1^{-1} \dot{\hat{W}}_1 \\ &= \frac{z_1}{g_1}\left[g_1\left(z_2 - c_1 z_1 - \tilde{W}_1^T S_1(Z_1) + \epsilon_1\right)\right] \\ &\quad - \frac{\dot{g}_1}{2g_1^2} z_1^2 - \tilde{W}_1^T S_1(Z_1) z_1 + \tilde{W}_1^T \Gamma_1^{-1} \dot{\hat{W}}_1 \\ &= z_1 z_2 - c_1 z_1^2 - \frac{\dot{g}_1}{2g_1^2} z_1^2 + z_1 \epsilon_1 \\ &\quad + \tilde{W}_1^T \Gamma_1^{-1}[\dot{\hat{W}}_1 - \Gamma_1 S_1(Z_1) z_1].\end{aligned} \tag{14}$$

Consider the following adaptation law:

$$\dot{\hat{W}}_1 = \dot{\tilde{W}}_1 = \Gamma_1[S_1(Z_1) z_1 - \sigma_1 \hat{W}_1] \tag{15}$$

where $\sigma_1 > 0$ is a small constant. The σ-modification term $\sigma_1 \hat{W}_1$ is introduced to improve the robustness in the presence of the NN approximation error ϵ_1 [23]. Without such a modification term, the NN weight estimates $\hat{W}_1$ might drift to very large values, which will result in a variation of a high-gain control scheme [11].

Let $c_1 = c_{10} + c_{11}$, with c_{10} and $c_{11} > 0$. Then, (14) becomes

$$\dot{V}_1 = z_1 z_2 - \left(c_{10} + \frac{\dot{g}_1}{2g_1^2}\right) z_1^2 - c_{11} z_1^2 + z_1 \epsilon_1 - \sigma_1 \tilde{W}_1^T \hat{W}_1. \tag{16}$$

By completion of squares, we have

$$\begin{aligned}-\sigma_1 \tilde{W}_1^T \hat{W}_1 &= -\sigma_1 \tilde{W}_1^T (\tilde{W}_1 + W_1^*) \\ &\le -\sigma_1 \|\tilde{W}_1\|^2 + \sigma_1 \|\tilde{W}_1\| \|W_1^*\| \\ &\le -\frac{\sigma_1 \|\tilde{W}_1\|^2}{2} + \frac{\sigma_1 \|W_1^*\|^2}{2} \\ -c_{11} z_1^2 + z_1 \epsilon_1 &\le -c_{11} z_1^2 + z_1 |\epsilon_1| \le \frac{{\epsilon_1}^2}{4c_{11}} \le \frac{\epsilon_1^{*2}}{4c_{11}}.\end{aligned} \tag{17}$$

Because $-(c_{10} + (\dot{g}_1/2g_1^2)) z_1^2 \le -(c_{10} - (g_{1d}/2g_{10}^2)) z_1^2$, by choosing c_{10} such that $(c_{10}^* \triangleq c_{10} - (g_{1d}/2g_{10}^2)) > 0$, we have the following inequality:

$$\dot{V}_1 < z_1 z_2 - c_{10}^* z_1^2 - \frac{\sigma_1 \|\tilde{W}_1\|^2}{2} + \frac{\sigma_1 \|W_1^*\|^2}{2} + \frac{\epsilon_1^{*2}}{4c_{11}} \tag{18}$$

where the coupling term $z_1 z_2$ will be canceled in the next step.

Step 2: The derivative of $z_2 = x_2 - \alpha_1$ is $\dot{z}_2 = f_2(\bar{x}_2) + g_2(\bar{x}_2) x_3 - \dot{\alpha}_1$. By viewing x_3 as a virtual control input to stabilize the (z_1, z_2)-subsystem, there exists a desired feedback control

$$\alpha_2^* = -z_1 - c_2 z_2 - \frac{1}{g_2(\bar{x}_2)}(f_2(\bar{x}_2) - \dot{\alpha}_1) \tag{19}$$

where c_2 is a positive constant to be specified later. From (11), it can be seen that α_1 is a function of x_1, x_d and $\hat{W}_1$. Thus, $\dot{\alpha}_1$ is given by

$$\begin{aligned}\dot{\alpha}_1 &= \frac{\partial \alpha_1}{\partial x_1}\dot{x}_1 + \frac{\partial \alpha_1}{\partial x_d}\dot{x}_d + \frac{\partial \alpha_1}{\partial \hat{W}_1}\dot{\hat{W}}_1 \\ &= \frac{\partial \alpha_1}{\partial x_1}(g_1(x_1) x_2 + f_1(x_1)) + \phi_1\end{aligned} \tag{20}$$

where $\phi_1 = (\partial \alpha_1/\partial x_d)\dot{x}_d + (\partial \alpha_1/\partial \hat{W}_1)[\Gamma_1(S_1(Z_1) z_1 - \sigma_1 \hat{W}_1)]$ is introduced as an intermediate variable which is computable. Since $f_1(x_1)$ and $g_1(x_1)$ are unknown, $\dot{\alpha}_1$ is in fact a scalar unknown nonlinear function. Let

$$h_2(Z_2) \triangleq \frac{1}{g_2(\bar{x}_2)}(f_2(\bar{x}_2) - \dot{\alpha}_1) \tag{21}$$

denote the unknown part of α_2^* (19), with $Z_2 \triangleq [\bar{x}_2^T, (\partial \alpha_1/\partial x_1), \phi_1]^T \subset R^4$. (Please see Remark 2 for the definition of Z_2). By employing an RBF neural network $W_2^T S_2(Z_2)$ to approximate $h_2(Z_2)$, α_2^* can be expressed as

$$\alpha_2^* = -z_1 - c_2 z_2 - W_2^{*T} S_2(Z_2) - \epsilon_2. \tag{22}$$

Define the error variable $z_3 = x_3 - \alpha_2$ and choose the virtual control

$$\alpha_2 = -z_1 - c_2 z_2 - \hat{W}_2^T S_2(Z_2). \tag{23}$$

Then, we have

$$\begin{aligned}\dot{z}_2 &= f_2(\bar{x}_2) + g_2(\bar{x}_2)(z_3 + \alpha_2) - \dot{\alpha}_1 \\ &= g_2\left[z_3 - z_1 - c_2 z_2 - \tilde{W}_2^T S_2(Z_2) + \epsilon_2\right].\end{aligned} \tag{24}$$

Remark 2: Though the unknown function $h_2(Z_2)$ (21) is a function of $\bar{x}_2, x_d$ and $\hat{W}_1$, the large number of neural weight estimates $\hat{W}_1$ are not recommended to be taken as inputs to the NN because of the curse of dimensionality of RBF NN [22]. By defining intermediate variables $(\partial \alpha_1/\partial x_1)$ and ϕ_1, which are available through the computation of $\bar{x}_2, x_d$ and $\hat{W}_1$, the NN approximation $\hat{W}_2^T S_2(Z_2)$ of the unknown function $h_2(Z_2)$ can be computed by using the minimal number of NN inputs $Z_2 = [\bar{x}_2^T, (\partial \alpha_1/\partial x_1), \phi_1]^T$. The introductions of intermediate variables help to avoid curse of dimensionality, and make the proposed neural control scheme computationally implementable. The same idea of choosing the input variables of NNs is also used in the following design steps.

Consider the Lyapunov function candidate

$$V_2 = V_1 + \frac{1}{2g_2(\bar{x}_2)} z_2^2 + \frac{1}{2}\tilde{W}_2^T \Gamma_2^{-1} \tilde{W}_2 \tag{25}$$

where $\Gamma_2 = \Gamma_2^T > 0$ is an adaptation gain matrix. The derivative of V_2 is

$$\begin{aligned}\dot{V}_2 &= \dot{V}_1 + \frac{z_2 \dot{z}_2}{g_2} - \frac{\dot{g}_2 z_2^2}{2g_2^2} + \tilde{W}_2^T \Gamma_2^{-1} \dot{\hat{W}}_2 \\ &= \dot{V}_1 - z_1 z_2 + z_2 z_3 - c_2 z_2^2 - \frac{\dot{g}_2}{2g_2^2} z_2^2 + z_2 \epsilon_2 \\ &\quad - \tilde{W}_2^T S_2(Z_2) z_2 + \tilde{W}_2^T \Gamma_2^{-1} \dot{\hat{W}}_2.\end{aligned} \tag{26}$$

Consider the following adaptation law:

$$\dot{\hat{W}}_2 = \dot{\tilde{W}}_2 = \Gamma_2[S_2(Z_2)z_2 - \sigma_2 \hat{W}_2] \tag{27}$$

where $\sigma_2 > 0$ is a small constant. Let $c_2 = c_{20} + c_{21}$, where c_{20} and $c_{21} > 0$. By using (18), (24), and (27), and with some completion of squares and straightforward derivation similar to those employed in Step 1, the derivative of V_2 becomes

$$\dot{V}_2 < z_2 z_3 - \sum_{k=1}^{2} c_{k0}^* z_k^2 - \sum_{k=1}^{2} \frac{\sigma_k \|\tilde{W}_k\|^2}{2} + \sum_{k=1}^{2} \frac{\sigma_k \|W_k^*\|^2}{2} + \sum_{k=1}^{2} \frac{\epsilon_k^{*2}}{4c_{k1}} \tag{28}$$

where c_{20} is chosen such that $c_{20}^* \triangleq c_{20} - (g_{2d}/2g_{20}^2)^2 > 0$.

Step $i (3 \le i \le n-1)$: The derivative of $z_i = x_i - \alpha_{i-1}$ is $\dot{z}_i = f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1} - \dot{\alpha}_{i-1}$. By viewing x_{i+1} as a virtual control input to stabilize the $(z_1, \ldots, z_i)$-subsystem, there exists a desired feedback control $\alpha_i^* = x_{i+1}$

$$\alpha_i^* = -z_{i-1} - c_i z_i - \frac{1}{g_i(\bar{x}_i)}(f_i(\bar{x}_i) - \dot{\alpha}_{i-1}) \tag{29}$$

where

$$\dot{\alpha}_{i-1} = \sum_{k=1}^{i-1} \frac{\partial \alpha_{i-1}}{\partial x_k}(g_k(\bar{x}_k)x_{k+1} + f_k(\bar{x}_k)) + \phi_{i-1} \tag{30}$$

with $\phi_{i-1} = \sum_{k=1}^{i-1} (\partial \alpha_{i-1}/\partial x_d)\dot{x}_d + \sum_{k=1}^{i-1} (\partial \alpha_{i-1}/\partial \hat{W}_k) [\Gamma_k(S_k(Z_k)z_k - \sigma_k \hat{W}_k)]$ computable.

Let

$$h_i(Z_i) \triangleq \frac{1}{g_i(\bar{x}_i)}(f_i(\bar{x}_i) - \dot{\alpha}_{i-1}) \tag{31}$$

denote the unknown part in α_i^* (29), with

$$Z_i \triangleq \left[\bar{x}_i^T, \frac{\partial \alpha_{i-1}}{\partial x_1}, \ldots, \frac{\partial \alpha_{i-1}}{\partial x_{i-1}}, \phi_{i-1}\right]^T \subset R^{2i}. \tag{32}$$

Note that the number of inputs to the NN is kept minimal by the introduction of intermediate variables $(\partial \alpha_{i-1}/\partial x_1), \ldots, (\partial \alpha_{i-1}/\partial x_{i-1}), \phi_{i-1}$, as discussed in Remark 2. By employing an RBF neural network $W_i^T S_i(Z_i)$ to approximate $h_i(Z_i), \alpha_i^*$ can be expressed as

$$\alpha_i^* = -z_{i-1} - c_i z_i - W_i^{*T} S_i(Z_i) - \epsilon_i. \tag{33}$$

Define the error variable $z_{i+1} = x_{i+1} - \alpha_i$ and choose the virtual control

$$\alpha_i = -z_{i-1} - c_i z_i - \hat{W}_i^T S_i(Z_i). \tag{34}$$

Then, we have

$$\begin{aligned} \dot{z}_i &= f_i(\bar{x}_i) + g_i(\bar{x}_i)(z_{i+1} + \alpha_i) - \dot{\alpha}_{i-1} \\ &= g_i \left[z_{i+1} - z_{i-1} - c_i z_i - \tilde{W}_i^T S_i(Z_i) + \epsilon_i\right]. \end{aligned} \tag{35}$$

Consider the Lyapunov function candidate

$$V_i = V_{i-1} + \frac{1}{2g_i(\bar{x}_i)} z_i^2 + \frac{1}{2}\tilde{W}_i^T \Gamma_i^{-1} \tilde{W}_i. \tag{36}$$

Consider the following adaptation law:

$$\dot{\hat{W}}_i = \dot{\tilde{W}}_i = \Gamma_i[S_i(Z_i)z_i - \sigma_i \hat{W}_i] \tag{37}$$

where $\sigma_i > 0$ is a small constant. Let $c_i = c_{i0} + c_{i1}$, where c_{i0} and $c_{i1} > 0$. By using (28), (35), and (37), and with some completion of squares and straightforward derivation similar to those employed in the former steps, the derivative of V_i becomes

$$\dot{V}_i < z_i z_{i+1} - \sum_{k=1}^{i} c_{k0}^* z_k^2 - \sum_{k=1}^{i} \frac{\sigma_k \|\tilde{W}_k\|^2}{2} + \sum_{k=1}^{i} \frac{\sigma_k \|W_k^*\|^2}{2} + \sum_{k=1}^{i} \frac{\epsilon_k^{*2}}{4c_{k1}} \tag{38}$$

where c_{i0} is chosen such that $c_{i0}^* :\triangleq (c_{i0} - (g_{id}/2g_{i0}^2)) > 0$.

Step n: This is the final step. The derivative of $z_n = x_n - \alpha_{n-1}$ is $\dot{z}_n = f_n(\bar{x}_n) + g_n(\bar{x}_{n-1})u - \dot{\alpha}_{n-1}$. To stabilize the whole system $(z_1, \ldots, z_n)$, there exists a desired feedback control

$$u^* = -z_{n-1} - c_n z_n - \frac{1}{g_n(\bar{x}_{n-1})}(f_n - \dot{\alpha}_{n-1}) \tag{39}$$

where

$$\dot{\alpha}_{n-1} = \sum_{k=1}^{n-1} \frac{\partial \alpha_{n-1}}{\partial x_k}(g_k(\bar{x}_k)x_{k+1} + f_k(\bar{x}_k)) + \phi_{n-1} \tag{40}$$

with $\phi_{n-1} = \sum_{k=1}^{n-1} (\partial \alpha_{n-1}/\partial x_d)\dot{x}_d + \sum_{k=1}^{n-1} (\partial \alpha_{n-1}/\partial \hat{W}_k) [\Gamma_k(S_k(Z_k)z_k - \sigma_k \hat{W}_k)]$ computable.

Let

$$h_n(Z_n) = \frac{1}{g_n(\bar{x}_{n-1})}(f_n(\bar{x}_n) - \dot{\alpha}_{n-1}) \tag{41}$$

denote the unknown part in u^* (39), with

$$Z_n = \left[\bar{x}_n^T, \frac{\partial \alpha_{n-1}}{\partial x_1}, \ldots, \frac{\partial \alpha_{n-1}}{\partial x_{n-1}}, \phi_{n-1}\right]^T \subset R^{2n}. \tag{42}$$

By employing an RBF neural network $W_n^T S_n(Z_n)$ to approximate $h_n(Z_n), u^*$ can be expressed

$$u^* = -z_{n-1} - c_n z_n - W_n^{*T} S_n(Z_n) - \epsilon_n. \tag{43}$$

Choosing the practical control law as

$$u = -z_{n-1} - c_n z_n - \hat{W}_n^T S_n(Z_n) \tag{44}$$

we have

$$\begin{aligned} \dot{z}_n &= f_n(\bar{x}_n) + g_n(\bar{x}_{n-1})u - \dot{\alpha}_{n-1} \\ &= g_n(\bar{x}_{n-1}) \left[-z_{n-1} - c_n z_n - \tilde{W}_n^T S_n(Z_n) + \epsilon_n\right]. \end{aligned} \tag{45}$$

Consider the overall Lyapunov function candidate

$$V_n = V_{n-1} + \frac{1}{2g_n(\bar{x}_{n-1})} z_n^2 + \frac{1}{2}\tilde{W}_n^T \Gamma_n^{-1} \tilde{W}_n. \tag{46}$$

Consider the following adaptation law:

$$\dot{\hat{W}}_n = \dot{\tilde{W}}_n = \Gamma_n[S_n(Z_n)z_n - \sigma_n \hat{W}_n] \tag{47}$$

where $\sigma_n > 0$ is a small constant. Let $c_n = c_{n0} + c_{n1}$, where c_{n0} and $c_{n1} > 0$. By using (38), (45), and (47), and with some completion of squares and straightforward derivation similar to those employed in the former steps, the derivative of V_i becomes

$$\dot{V}_n < -\sum_{k=1}^{n} c_{k0}^* z_k^2 - \sum_{k=1}^{n} \frac{\sigma_k \|\tilde{W}_k\|^2}{2} + \sum_{k=1}^{n} \frac{\sigma_k \|W_k^*\|^2}{2} + \sum_{k=1}^{n} \frac{\epsilon_k^{*2}}{4c_{k1}} \tag{48}$$

where c_{n0} is chosen such that $c_{n0}^* \stackrel{\triangle}{:=} c_{n0} - (g_{nd}/2g_{n0}^2) > 0$.

Let $\delta \stackrel{\triangle}{=} \sum_{k=1}^{n} (\sigma_k \|W_k^*\|^2/2) + \sum_{k=1}^{n} (\epsilon_k^{*2}/4c_{k1})$. If we choose c_{k0}^* such that $c_{k0}^* \geq (\gamma/2g_{k0})$, i.e., choose c_{k0} such that $c_{k0} > (\gamma/2g_{k0}) + (g_{kd}/2g_{k0}^2), k = 1, \ldots, n$, where γ is a positive constant, and choose σ_k and Γ_k such that $\sigma_k \geq \gamma \lambda_{\max}\{\Gamma_k^{-1}\}, k = 1, \ldots, n$, then from (48) we have the following inequality:

$$\begin{aligned} \dot{V}_n &< -\sum_{k=1}^{n} c_{k0}^* z_k^2 - \sum_{k=1}^{n} \frac{\sigma_k \|\tilde{W}_k\|^2}{2} + \delta \\ &\leq -\sum_{k=1}^{n} \frac{\gamma}{2g_{k0}} z_k^2 - \sum_{k=1}^{n} \frac{\gamma \tilde{W}_k^T \Gamma_k^{-1} \tilde{W}_k}{2} + \delta \\ &\leq -\gamma \left[\sum_{k=1}^{n} \frac{1}{2g_k} z_k^2 + \sum_{k=1}^{n} \frac{\tilde{W}_k^T \Gamma_k^{-1} \tilde{W}_k}{2} \right] + \delta \\ &\leq -\gamma V_n + \delta. \end{aligned} \tag{49}$$

The following theorem shows the stability and control performance of the closed-loop adaptive system.

Theorem 1: Consider the closed-loop system consisting of the plant (1), the reference model (2), the controller (44), and the NN weight updating laws (15), (27), (37), and (47). Assume there exists sufficiently large compact sets $\Omega_i \in R^{2i}, i = 1, \ldots, n$ such that $Z_i \in \Omega_i$ for all $t \geq 0$. Then, for bounded initial conditions, we have the following.

1) All signals in the closed-loop system remain bounded, and the states $\bar{x}_n$ and the neural weight estimates $\hat{W}_1^T, \ldots, \hat{W}_n^T$ eventually converge to the compact set

$$\Omega_{s1} \stackrel{\triangle}{=} \left\{ \bar{x}_n, \hat{W}_1, \ldots, \hat{W}_n \,\middle|\, V < \frac{\delta}{\gamma}, x_d \in \Omega_d \right\}. \tag{50}$$

2) The output tracking error $y(t) - y_{d1}(t)$ converges to a small neighborhood around zero by appropriately choosing design parameters.

Proof: 1) From (49), using the boundedness theorem (e.g., [20, Th. 2.14]), we have that all z_i and $\hat{W}_i (i = 1, \ldots, n)$ are uniformly ultimately bounded. Since $z_1 = x_1 - x_{d1}$ and x_{d1} are bounded, we have that x_1 is bounded. From $z_i = x_i - \alpha_{i-1}, i = 2, \ldots, n$, and the definitions of virtual controls α_i (11), (23), (34) we have that $x_i, i = 2, \ldots, n$ remain bounded. Using (44), we conclude that control u is also bounded. Thus, all the signals in the closed-loop system remain bounded.

To provide some estimates of the regions of attraction of (49), we consider the following two conditions.

1) If

$$(\bar{x}_n(0), \hat{W}_1(0), \ldots, \hat{W}_n(0)) \in \Omega_{01} \subseteq \Omega_{s1} \tag{51}$$

where Ω_{s1} is given in (50), then according to Theorem 2.14 in [20], all the states $\bar{x}_n$ and the neural weights $\hat{W}_1, \ldots, \hat{W}_n$ will remain in Ω_{s1}, i.e.,

$$(\bar{x}_n(t), \hat{W}_1(t), \ldots, \hat{W}_n(t)) \in \Omega_{s1}, \quad \forall t \geq 0.$$

2) If

$$(\bar{x}_n(0), \hat{W}_1(0), \ldots, \hat{W}_n(0)) \in \Omega_{02} \subset \Omega_{s1}^c$$

where Ω_{s1}^c denotes the complimentary set of Ω_{s1}, then $\dot{V}_n$ remains negative definite until the states $\bar{x}_n$ and the neural weights $\hat{W}_1, \ldots, \hat{W}_n$ enter and stay inside Ω_{s1}, i.e.,

$$(\bar{x}_n(t), \hat{W}_1(t), \ldots, \hat{W}_n(t)) \in (\Omega_{02} \cup \Omega_{s1}), \quad \forall t \geq 0.$$

Thus, for bounded initial conditions, all signals in the closed-loop system remain bounded, and the states $\bar{x}_n$ and the neural weights $\hat{W}_1, \ldots, \hat{W}_n$ will eventually converge to the compact set Ω_{s1}.

3) Let $\rho \stackrel{\triangle}{=} \delta/\gamma > 0$, then (49) satisfies

$$0 \leq V_n(t) < \rho + (V_n(0) - \rho)\exp(-\gamma t). \tag{52}$$

From (52), we have

$$\begin{aligned} \sum_{k=1}^{n} \frac{1}{2g_k} z_k^2 &< \rho + (V_n(0) - \rho)\exp(-\gamma t) \\ &< \rho + V_n(0)\exp(-\gamma t). \end{aligned} \tag{53}$$

Let $g^* = \max_{1 \leq i \leq n}\{g_{i1}\}$. Then, we have

$$\frac{1}{2g^*} \sum_{k=1}^{n} z_k^2 \leq \sum_{k=1}^{n} \frac{1}{2g_k} z_k^2 < \rho + V_n(0)\exp(-\gamma t) \tag{54}$$

that is

$$\sum_{k=1}^{n} z_k^2 < 2g^*\rho + 2g^* V_n(0)\exp(-\gamma t) \tag{55}$$

which implies that given $\mu > \sqrt{2g^*\rho}$, there exists T such that for all $t \geq T$, the tracking error satisfies

$$|z_1(t)| = |x_1(t) - x_{d1}(t)| = |y(t) - y_d(t)| < \mu \tag{56}$$

where μ is the size of a small residual set which depends on the NN approximation error ϵ_i and controller parameters c_i, σ_i, and Γ_i. It is easily seen that the increase in the control gain c_i, adaptive gain Γ_i and NN node number l_j will result in a better tracking performance. $\diamond$

Remark 3: In the above analysis, it is clear that the uniform ultimate boundedness of all the signals are guaranteed by choosing $c_i = c_{i0} + c_{i1}$ large enough, such that $c_{i0}^* \stackrel{\triangle}{=} c_{i0} - (g_{id}/2g_{i0}^2) > 0$. Moreover, it can be seen that 1) increasing c_{i0} might lead to larger γ, and increasing c_{i1} will reduce δ, thus, increasing c_i will lead to smaller Ω_{s1} and 2) decreasing σ_i will help to reduce δ, and increasing the NN node number l_j will help to reduce ϵ_i^*, both of which will help to reduce the size of Ω_{s1}. However, increasing c_i will lead to a high gain control scheme. On the other hand, though σ_i is required to be chosen as a small positive constant when applying σ-modification [23], a very small σ_i may not be enough to prevent the NN weight estimates from drifting to very large values in the presence of the NN approximation errors [11], where the large $\hat{W}_i$ might result in a variation of a high-gain control. Therefore, in practical applications, the design parameters should be adjusted carefully for achieving suitable transient performance and control action.

Remark 4: Note that in backstepping design, differentiation of the virtual controls has to be performed. For the control of uncertain strict-feedback system (1), the differentiation requirement becomes a technical problem due to the presence of uncertainties. The derivative of virtual control α_{i-1} (30) involves the unknown nonlinear functions $f_1, \ldots, f_{i-1}(\cdot)$ and $g_1, \ldots, g_{i-1}(\cdot)(i = 2, \ldots, n)$, and thus is a unknown nonlinear function which is unavailable for implementation. Because derivative $\dot{\alpha}_{i-1}$ appears in the desired virtual control α_i^* (29), an RBF neural network can be employed to approximate all the unknown part of α_i^* (29), including the unknown $\dot{\alpha}_{i-1}$. However, since $\dot{\alpha}_{i-1}$ is a smooth function of system states $\bar{x}_i$ and neural weight estimates $\hat{W}_1, \ldots, \hat{W}_{i-1}$, as can be seen from (30), approximating the unknown nonlinear function $\dot{\alpha}_{i-1}$ might become computationally unacceptable when the large number of neural weight estimates $\hat{W}_1, \ldots, \hat{W}_{i-1}$ are taken as inputs to the RBF NN, which will lead to curse of dimensionality [22]. By introducing some intermediate variables $(\partial\alpha_{i-1}/\partial x_1), \ldots, (\partial\alpha_{i-1}/\partial x_{i-1})$ and ϕ_{i-1} as inputs to RBF NN $W_i^T S_i(Z_i)$, where the intermediate variables are available through the computation of system states $\bar{x}_i$ and neural weight estimates $\hat{W}_1, \ldots, \hat{W}_{i-1}$, the NN approximation can be implemented by using the minimal number of NN input variables. Hence, the differentiation requirement can be satisfied by using RBF NNs to approximate all the uncertainties in the desired virtual controls and practical control, while intermediate variables are introduced to make the NN approximation computationally feasible.

Remark 5: The adaptive NN controller (44) adaptation laws (15), (37), and (47) are highly structural, and independent of the complexities of the system nonlinearities. Thus, it can be applied to other similar plants without repeating the complex controller design procedure for different system nonlinearities. In addition, such a structural property is particularly suitable for parallel processing and hardware implementation in practical applications.

IV. Simulation Studies

In practical application of the proposed scheme, the selection of the centers and widths of RBF has a great influence on the performance of the designed controller. According to [3], Gaussian RBF NNs arranged on a regular lattice on R^n can uniformly approximate sufficiently smooth functions on closed, bounded subsets. Furthermore, given only crude estimates of the smoothness of the function being approximated, it is feasible to select the centers and variances of a finite number of Gaussian nodes, so that the resulting NNs are capable of uniformly approximating the required function to a chosen tolerance everywhere on a prespecified subset. Accordingly, in the following simulation studies, we will select the centers and widths on a regular lattice in the respective compact sets.

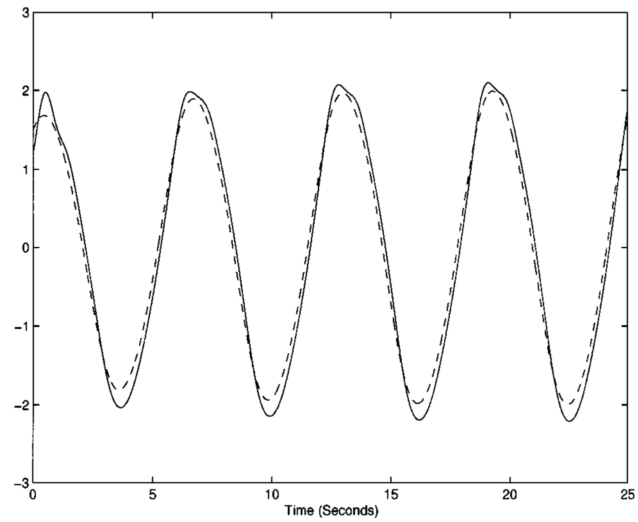

Fig. 1. Output tracking performance (y—solid line and y_d—dashed line).

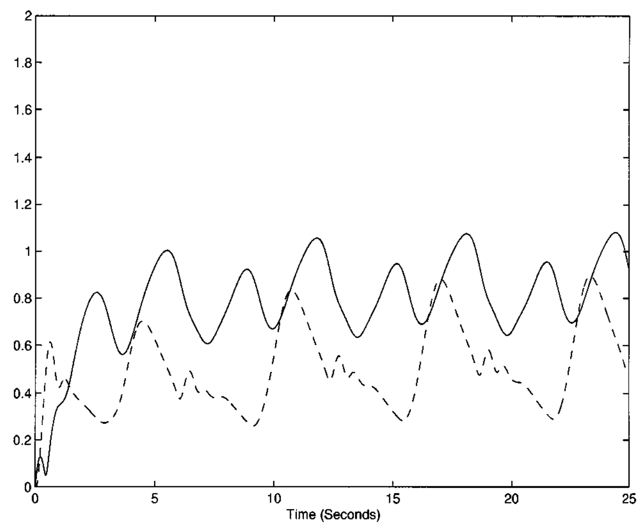

Fig. 2. L_2 norms of the NN weights: $\hat{W}_1$ (solid line) and $\hat{W}_2$ (dashed line).

The proposed adaptive NN controller is applied to the following strict-feedback system:

$$\begin{aligned} \dot{x}_1 &= 0.5x_1 + (1 + 0.1x_1^2)x_2 \\ \dot{x}_2 &= x_1x_2 + [2 + \cos(x_1)]u \\ y &= x_1. \end{aligned} \tag{57}$$

The control objective is to guarantee 1) all the signals in the closed-loop system remain bounded and 2) the output y follows a desired trajectory y_d generated from the following van der Pol oscillator system:

$$\begin{aligned} \dot{x}_{d1} &= x_{d2} \\ \dot{x}_{d2} &= -x_{d1} + \beta(1 - x_{d1}^2)x_{d2} \\ y_d &= x_{d1}. \end{aligned} \tag{58}$$

As shown in [24], the phase-plane trajectories of the van der Pol oscillator, starting from an initial state other than $(0, 0)$, approach a limit cycle when $\beta > 0$.

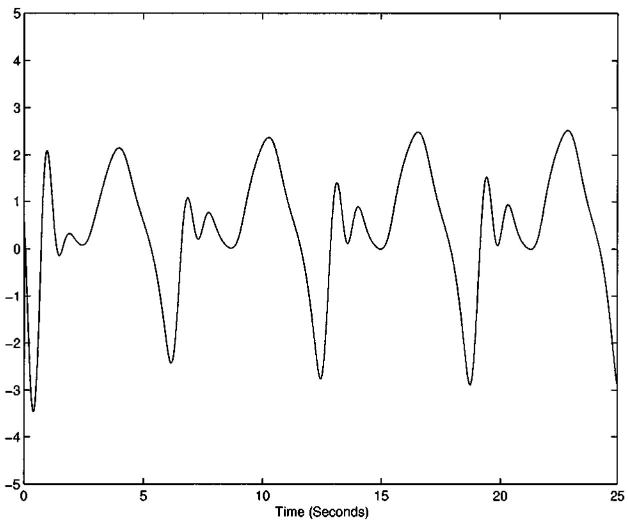

Fig. 3. Boundedness of the control u.

Clearly, system (58) is in strict-feedback form (1) and satisfies Assumptions 1 and 2. As system (58) is of second order, the adaptive NN controller is chosen according to (44) as follows:

$$u = -z_1 - c_2 z_2 - \hat{W}_2^T S_2(Z_2) \tag{59}$$

where $z_1 = x_1 - y_d, z_2 = x_2 - \alpha_1$ and $Z_2 = [x_1, x_2, (\partial\alpha_1/\partial x_1), \phi_1]^T$ with

$$\alpha_1 = -c_1 z_1 - \hat{W}_1^T S_1(Z_1), Z_1 = [x_1, \dot{x}_{d1}]^T$$
$$\phi_1 = \frac{\partial\alpha_1}{\partial x_{d1}}\dot{x}_{d1} + \frac{\partial\alpha_1}{\partial x_{d2}}\dot{x}_{d2} + \frac{\partial\alpha_1}{\partial\hat{W}_1}\dot{\hat{W}}_1 \tag{60}$$

and NN weights $\hat{W}_1$ and $\hat{W}_2$ are updated by (15) and (27) correspondingly.

Neural networks $\hat{W}_1^T S_1(Z_1)$ contains 25 nodes (i.e., $l_1 = 25$), with centers $\mu_l(l = 1, \ldots, l_1)$ evenly spaced in $[-4, 4] \times [-4, 4]$, and widths $\eta_l = 2(l = 1, \ldots, l_1)$. Neural networks $\hat{W}_2^T S_2(Z_2)$ contains 135 nodes (i.e., $l_2 = 135$), with centers μ_l $(l = 1, \ldots, l_2)$ evenly spaced in $[-4, 4] \times [-4, 4] \times [-4, 4] \times [-6, 6]$, and widths $\eta_l = 2$ $(l = 1, \ldots, l_2)$. The design parameters of the above controller are $c_1 = 3.5$, $c_2 = 3.5, \Gamma_1 = \Gamma_2 = \mathrm{diag}\{2.0\}, \sigma_1 = \sigma_2 = 0.2$. The initial weights $\hat{W}_1(0) = 0.0, \hat{W}_2(0) = 0.0$. The initial conditions $[x_1(0), x_2(0)]^T = [1.2, 1.0]^T$ and $[x_{d1}(0), x_{d2}(0)]^T = [1.5, 0.8]^T$.

Figs. 1–3 shows the simulation results of applying controller (59) to system (58) for tracking desired signal y_d with $\beta = 0.2$. From Fig. 1, we can see that fairly good tracking performance is obtained. The boundedness of NN weights $\hat{W}_1, \hat{W}_2$ and control signal u are shown in Figs. 2–3, respectively.

V. Conclusion

In this paper, a direct adaptive neural control scheme is presented for a class of uncertain nonlinear strict-feedback systems. By utilizing a special property of the affine term, the developed scheme avoids the controller singularity problem completely. All the signals of the closed-loop system are guaranteed to be semiglobally uniformly ultimately bounded, and the output of the system is proven to converge to a small neighborhood of the desired trajectory. The proposed direct adaptive NN scheme can be applied to a large class of nonlinear strict-feedback systems without repeating the complex controller design procedure for different system nonlinearities.

References

[1] M. Krstič, I. Kanellakopoulos, and P. Kokotovič, *Nonlinear and Adaptive Control Design*. New York: Wiley, 1995.

[2] M. M. Polycarpou and P. A. Ioannou, "Identification and Control Using Neural Network Models: Design and Stability Analysis," Dept. Elect. Eng. Syst., Univ. Southern California, Los Angeles, Tech. Rep. 91-09-01, 1991.

[3] R. M. Sanner and J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Trans. Neural Networks*, vol. 3, pp. 837–863, Nov. 1992.

[4] F. C. Chen and C. C. Liu, "Adaptively controlling nonlinear continuous-time systems using multilayer neural networks," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 1306–1310, 1994.

[5] G. A. Rovithakis and M. A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, pp. 400–412, Mar. 1994.

[6] F. C. Chen and H. K. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks," *IEEE Trans. Automat. Contr.*, vol. 40, pp. 791–801, 1995.

[7] A. Yesidirek and F. L. Lewis, "Feedback linearization using neural networks," *Automatica*, vol. 31, no. 11, pp. 1659–1664, 1995.

[8] F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Networks*, vol. 7, pp. 388–398, Mar. 1996.

[9] J. T. Spooner and K. M. Passino, "Stable adaptive control using fuzzy systems and neural networks," *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 339–359, 1996.

[10] S. S. Ge, T. H. Lee, and C. J. Harris, *Adaptive Neural Network Control of Robotic Manipulators*. London, U.K.: World Scientific, 1998.

[11] T. Zhang, S. S. Ge, and C. C. Hang, "Design and performance analysis of a direct adaptive controller for nonlinear systems," *Automatica*, vol. 35, pp. 1809–1817, 1999.

[12] G. A. Rovithakis, "Tracking control of multi-input affine nonlinear dynamical systems with unknown nonlinearities using dynamical neural networks," *IEEE Trans. Syst., Man, Cybern.*, vol. 29, pp. 179–189, 1999.

[13] M. M. Polycarpou, "Stable adaptive neural control scheme for nonlinear systems," *IEEE Trans. Automat. Contr.*, vol. 41, pp. 447–451, 1996.

[14] M. M. Polycarpou and M. J. Mears, "Stable adaptive tracking of uncertain systems using nonlinearly parametrized on-line approximators," *Int. J. Contr.*, vol. 70, no. 3, pp. 363–384, 1998.

[15] Y. Zhang, P. Y. Peng, and Z. P. Jiang, "Stable neural controller design for unknown nonlinear systems using backstepping," *IEEE Trans. Neural Networks*, vol. 11, pp. 1347–1359, 2000.

[16] T. Zhang, S. S. Ge, and C. C. Hang, "Adaptive neural network control for strict-feedback nonlinear systems using backstepping design," *Automatica*, vol. 36, pp. 1835–1846, 2000.

[17] B. Yao and M. Tomizuka, "Adaptive robust control of SISO nonlinear systems in a semi-strict feedback form,," *Automatica*, vol. 33, pp. 893–900, 1997.

[18] Z. P. Jiang and L. Praly, "Design of robust adaptive controllers for nonlinear systems with dynamic uncertainties," *Automatica*, vol. 34, no. 7, pp. 825–840, 1998.

[19] Z. Pan and T. Basar, "Adaptive controller design for tracking and disturbance attenuation in parametric strict-feedback nonlinear systems," *IEEE Trans. Automat. Contr.*, vol. 43, pp. 1066–1083, 1998.

[20] Z. Qu, *Robust Control of Nonlinear Uncertain Systems*. New York: Wiley, 1998.

[21] G. Nurnberger, *Approximation by Spline Functions*. New York: Springer-Verlag, 1989.

[22] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.

[23] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[24] M. Vidyasagar, *Nonlinear Systems Analysis*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1993.