Adaptive NN Backstepping With Considering Integral of Tracking Error

JIANGTAO XU1, YU FU2, GANGHUI CAO1, DEWEI ZHANG1, AND YA YANG1
1Department of Aerospace Engineering, Harbin Engineering University, Harbin 150001, China
2China Academy of Launch Vehicle Technology, Beijing Institute of Astronautical Systems Engineering, Beijing 100076, China
Corresponding author: Ganghui Cao (cao_g_h@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 11372080, in part by the Harbin Youth Innovation Talent Fund under Grant 2014RFQXJ121, and in part by the China Scholarship Council under Grant 201606665036.

ABSTRACT In order to simplify the design procedure of traditional neural network backstepping and improve the robustness and precision of control, an improved scheme is studied for a class of nonlinear systems. To avoid reconstructing virtual control inputs in each recursive step, RBF neural networks are utilized as approximators to estimate the desired feedback control of the whole system only. Meanwhile, the integral action of tracking error is introduced into the backstepping design procedure, which not only participates in updating the neural network weight, but also serves as a component part of the control input. This design may benefit the parameter tuning and make controller perform better sometimes. Based on the Lyapunov synthesis approach, theoretical analysis and simulation results are provided to show the feasibility of the improved scheme.

INDEX TERMS Backstepping, adaptive nonlinear control, neural networks.

I. INTRODUCTION
Backstepping technique can date back to [1]. As a systematic design approach, its recursive design procedure is suitable for complex strict-feedback systems. Owe to some significant works such as [2], backstepping technique has been further developed and perfected in theory. However, “explosion of complexity” arises with the growth of system order. So the application in practice may be limited for high order systems until dynamic surface control is proposed [3]. However, because of filtering technique, dynamic surface control doesn’t have the form of exponential convergence. In recent years, backstepping and techniques derived from it are increasingly common in various engineering fields. For air-breathing hypersonic vehicle control, dynamic surface control technique is involved in conjunction with the backstepping control approach [4]. A tracking controller is designed for autonomous helicopter based on a backstepping procedure [5]. Besides, backstepping tracking control law is applied to an autonomous underwater vehicle under a novel framework [6]. And [7] gives distributed controllers to solve flocking problem of multiple mobile robots with the aid of backstepping techniques.

In terms of engineering applications, precise modeling may be hard to establish for most systems. In practice, uncertainty is one of the important factors that affect the control performance and the closed-loop stability of the whole system [8]. To deal with the parametric uncertainties and unknown nonlinearities, much progress has been made by combining backstepping technique with robust control strategy. At the same time, increasing research interests focus on neural networks (NNs). For example, radial basis function (RBF) networks have been proposed [9]. For RBF networks, F. Girosi and T. Poggio have proved existence and uniqueness of best approximation [10]. This kind of neural networks have recently drawn much attention due to their good generalization ability and a simple network structure that avoids unnecessary and lengthy calculation [11]. In recent years, stability, dissipativity and extended dissipativity analysis problems have been investigated for NNs with time delay [12], [13]. With the development of NNs, neural network control of unknown nonlinear dynamic systems has caused wide public concern [14]. One of the great advantages is that exact values of the system base parameters are not required to be known a priori, such as [15]. Based on Lyapunov’s stability theory, stable adaptive NN controller can be designed, so that the neural network weight can achieve on-line adjustment. In this field, many significant...
works [14], [16]–[19] have been made by combining adaptive neural design with backstepping methodology.

Different from them, a systematic design of adaptive NN backstepping in this paper is developed inspired by [1], [11]. Instead of reconstructing virtual control inputs in each recursive step, the design process of RBF neural network is carried out until we step back to the whole system. By utilizing some properties of the system, the desired feedback control of the whole system has the form in which the unknown nonlinearities and known quantities are separated. It is not necessary for each subsystem to acquire a virtual control input estimated by neural network. In this scheme, the number of neural networks can be cut down and will probably no longer be the same as the order of the system. Meanwhile, we introduce the integral action of tracking error into the backstepping design procedure, which not only participates in updating the neural network weight, but also serves as a component part of the control input. Appropriate integral operation can improve the control effect sometimes. So a parameter is provided to adjust the intensity of integral operation can improve the control effect sometimes. It helps to enhance the disturbance suppression performance and meet higher precision requirement, but may not hold good for all cases.

This paper uses fundamental and simple mathematical knowledge, overall adaptive scheme is shown to guarantee the stability of the closed-loop system. The remaining part of the paper is organized as follows. The system under consideration is described in section II. In section III, the desired feedback control is presented without repeating the complex design procedure for system with arbitrary order. In section IV, an adaptive neural network controller is provided for controlling uncertain nonlinear systems. Section V demonstrates the simulation results to verify the feasibility.

II. SYSTEM DESCRIPTION

Consider the SISO nonlinear system with a simplified triangular structure [20]:

\[
\begin{align*}
\dot{x}_1 &= f_1(x_1, x_2, \cdots, x_i, x_{i+1}) + x_{i+1}, \quad 1 \leq i \leq n - 1 \\
\dot{x}_n &= f_n(x) + g(x)u \\
y &= x_1
\end{align*}
\]

where \(x = [x_1, x_2, \cdots, x_n]^T \in \mathbb{R}^n, u \in \mathbb{R}, y \in \mathbb{R}\) are the state variables, system input and output respectively; \(g\) and \(f_i, i = 1, 2, \cdots, n\) can be known or unknown smooth functions. Disturbance is included in \(f_n\). The objective is to force the output \(x_1\) follow a desired trajectory \(x_{1d}\). Note that many control problem (e.g., single inverted pendulum control, flight-path angle control [21], and manipulator control [22]) can be formulated in such structure.

Besides, systems in a more general form may be transformed into (1). A case study of three order system is provided here and similar procedure may be extended to \(n\) order systems. Consider the following nonlinear dynamic system:

\[
\begin{align*}
\dot{x}_1 &= f_1(x_1) + g_1(x_1, x_2) \\
\dot{x}_2 &= f_2(x_1, x_2) + g_2(x_1, x_2, x_3) \\
\dot{x}_3 &= f_3(x_1, x_2, x_3) + g_3(x_1, x_2, x_3)u \\
y &= x_1
\end{align*}
\]

Define \(\xi_1 = x_1, \xi_2 = g_1(x_1, x_2)\). Supposing that there exists an explicit solution \(x_2 = \alpha_2(\xi_1, \xi_2)\) (e.g. \(x_2 = \frac{x_2}{g_1(x_1)}\) if \(g_2(x_1, x_2) \neq 0\)), then let \(\frac{\alpha_1}{\alpha_3} = \beta_21(\xi_1, \xi_2), \frac{\alpha_2}{\alpha_3} = \beta_22(\xi_1, \xi_2)\), we obtain that

\[
\begin{align*}
\dot{\xi}_2 &= \beta_21\dot{\xi}_1 + \beta_22\dot{\xi}_2 \\
&= \beta_21(\xi_1, \xi_2) \cdot [f_1(\xi_1) + \xi_2] \\
&+ \beta_22(\xi_1, \xi_2) \cdot [f_2(\xi_1, \alpha_2) + g_2(\xi_1, \alpha_2, x_3)]
\end{align*}
\]

Define \(\xi_3 = \beta_31(\xi_1, \xi_2, \xi_3)\). Supposing that there exists an explicit solution \(x_3 = \alpha_3(\xi_1, \xi_2, \xi_3)\), then let \(\frac{\alpha_3}{\alpha_4} = \beta_32(\xi_1, \xi_2, \xi_3), \frac{\alpha_3}{\alpha_4} = \beta_33(\xi_1, \xi_2, \xi_3)\), we obtain that

\[
\begin{align*}
\dot{\xi}_3 &= \beta_31\dot{\xi}_1 + \beta_32\dot{\xi}_2 + \beta_33\dot{\xi}_3 \\
&= \beta_31(\xi_1, \xi_2, \xi_3) \cdot [f_1(\xi_1) + \xi_2] \\
&+ \beta_32(\xi_1, \xi_2, \xi_3) \cdot [f_2(\xi_1, \alpha_2) + g_2(\xi_1, \alpha_2, x_3)] \\
&+ \beta_33(\xi_1, \xi_2, \xi_3) \cdot [f_3(\xi_1, \alpha_2, x_3) + g_3(\xi_1, \alpha_2, x_3)u]
\end{align*}
\]

All signals in the closed-loop system must remain bounded, so the derivation above is not available for all cases.

III. DESIRED FEEDBACK CONTROL

For the sake of brevity, the following step-by-step procedure only shows Step 1, 2, i, n, in detail with redundant equations and repetitive steps being omitted.

Step 0: Define tracking error \(e_1 = x_1 - x_{1d}\), and denote by \(\xi, k_1, k_2, \cdots, k_n\) constant coefficients to be chosen later.

Step 1: Define \(s_1 = e_1 + \xi \int e_1 dt\), whose derivative is

\[
\dot{s}_1 = f_1 + \dot{x}_2 - \dot{x}_{1d} + \xi e_1
\]

View \(x_2\) as a virtual control input, the desired value can be expressed as

\[
x_{2d} = -k_1 s_1 - f_1 + \dot{x}_{1d} - \xi e_1 + \xi \int e_2 dt
\]

where, \(e_2 = x_2 - x_{2d}\).
Taking $V_1 = \frac{1}{2}s_1^2$ as a Lyapunov function candidate, its derivative is
\[ \dot{V}_1 = s_1\dot{s}_1 = -k_1s_1^2 + s_1\left(e_2 + \xi \int e_2\,dt\right) \]  \hspace{1cm} (8)

**Step 2:** Define $s_2 = e_2 + \xi \int e_2\,dt$, whose derivative is
\[ \dot{s}_2 = f_2 + x_3 - \dot{x}_2d + \xi e_2 \]  \hspace{1cm} (9)

View $x_3$ as a virtual control input, the desired value can be expressed as
\[ x_{3d} = -k_2s_2 - s_1 - f_2 + \dot{x}_2d - \xi e_2 + \xi \int e_3\,dt \]  \hspace{1cm} (10)

where, $e_3 = x_3 - x_{3d}$.

Taking $V_2 = V_1 + \frac{1}{2}s_2^2$ as a Lyapunov function candidate, its derivative is
\[ \dot{V}_2 = -k_1s_1^2 + s_1s_2 + s_2\dot{s}_2 \]
\[ = -k_1s_1^2 - k_2s_2^2 + s_2\left(e_3 + \xi \int e_3\,dt\right) \]  \hspace{1cm} (11)

**Step i(2 ≤ i ≤ n − 1):** Define $s_i = e_i + \xi \int e_i\,dt$, whose derivative is
\[ \dot{s}_i = f_i + x_{i+1} - \dot{x}_{id} + \xi e_i \]  \hspace{1cm} (12)

View $x_{i+1}$ as a virtual control input, the desired value can be expressed as
\[ x_{(i+1)d} = -k_is_i - s_{i-1} - f_i + \dot{x}_{id} - \xi e_i + \xi \int e_{i+1}\,dt \]  \hspace{1cm} (13)

where, $e_{i+1} = x_{i+1} - x_{(i+1)d}$

Taking $V_i = V_{i-1} + \frac{1}{2}s_i^2$ as a Lyapunov function candidate, its derivative is
\[ \dot{V}_i = \sum_{m=1}^{i-1} k_ms_m^2 + s_{i-1}s_i + s_i\dot{s}_i \]
\[ = -\sum_{m=1}^{i-1} k_ms_m^2 + s_{i-1}s_i + \left(e_{i+1} + \xi \int e_{i+1}\,dt\right) \]  \hspace{1cm} (14)

**Step n:** Define $s_n = e_n + \xi \int e_n\,dt$, whose derivative is
\[ \dot{s}_n = f_n + gu - \dot{x}_{nd} + \xi e_n \]  \hspace{1cm} (15)

Desired feedback control input is chosen as
\[ u^* = \frac{1}{g}\left(-k_sn_s - s_{n-1} - f_n + \dot{x}_{nd} - \xi e_n\right) \]  \hspace{1cm} (16)

Choose overall Lyapunov function $V_n = V_{n-1} + \frac{1}{2}s_n^2$. As a positive definite function, it is differentiable with a negative time derivative, that is
\[ \dot{V}_n = \sum_{m=1}^{n-1} k_ms_m^2 + s_{n-1}s_n + s_n\dot{s}_n \]
\[ = -\sum_{m=1}^{n-1} k_ms_m^2 + s_{n-1}s_n + s_n\left(f_n + gu^* - \dot{x}_{nd} + \xi e_n\right) \]
\[ = -\sum_{m=1}^{n-1} k_ms_m^2 \]  \hspace{1cm} (17)

The desired feedback control input $u^*$ is given in the last step. By suitably choosing value of $\xi, k_1, k_2, \ldots, k_n$, stability of the whole close-loop system can be obtained with good performance. Then, supposing that nonlinear function $g$ and $f_i, i = 1, 2, \ldots, n$ are known exactly, the realizable and practicable form of $u^*$ is derived as follows.

Differentiating (7) and (13) yields
\[ -\dot{x}_{2d} + \xi e_2 = k_1s_1 + \dot{f}_1 - \dot{x}_{1d} + \xi \dot{e}_1 \]
\[ -\dot{x}_{3d} + \xi e_3 = k_2s_2 + \dot{f}_1 + \dot{f}_2 - \dot{x}_{2d} + \xi \dot{e}_2 \]
\[ \vdots \]
\[ -\dot{x}_{id} + \xi e_i = k_{i-1}s_{i-1} + \dot{f}_{i-2} - \dot{x}_{(i-1)d} + \xi \dot{e}_{i-1}, \quad 3 \leq i \leq n \]  \hspace{1cm} (18)

It follows that
\[ -\dot{x}_{i+1d} + \xi e_{i+1} = f_i + x_{i+1} - \dot{x}_{id} + \xi \dot{e}_i \]
\[ -\dot{x}_{id} + \xi e_i = \sum_{m=1}^{i-1} (f_m + k_ms_m\xi e_i) + s_{i-1} - s_{i-2} + \xi e_{i-2}, \quad 3 \leq i \leq n + 1 \]  \hspace{1cm} (19)

The above equation can be rewritten as
\[ u^* = \frac{1}{g}\left(-\sum_{m=1}^{n} f_m(n-m) - \sum_{m=1}^{n} k_ms_m(n-m)\right) \]
\[ -\sum_{m=1}^{n} s_m(n-m) + x_{1d} - \xi e_1(n-1) \]  \hspace{1cm} (20)

Substituting the above equation into (16) to have the form
\[ u^* = \frac{1}{g}\left(-\sum_{m=1}^{n} f_m(n-m) - \sum_{m=1}^{n} k_ms_m(n-m)\right) \]
\[ -\sum_{m=1}^{n} s_m(n-m) + x_{1d} - \xi e_1(n-1) \]  \hspace{1cm} (21)

System (1) can be transformed into the following equations:
\[ f_1 + x_2 = \dot{x}_1 \]
\[ f_2 + x_3 = \dot{x}_2 - \dot{f}_1 \]
\[ f_3 + x_4 = \dot{x}_3 - \dot{f}_2 \]
\[ \vdots \]
\[ f_i + x_{i+1} = \dot{x}_i - \sum_{m=1}^{i-1} f_m(i-m), \quad 3 \leq i \leq n + 1 \]  \hspace{1cm} (22)

The above equation can be rewritten as
\[ f_{i-1} + x_i = x_i^{(i-1)} - \sum_{m=1}^{i-2} f_m(i-m-1), \quad 3 \leq i \leq n \]  \hspace{1cm} (23)
Combining (7) and (13) with the define of $e_{i+1}$ and $s_{i+1}$, we obtain
\[ s_2 = f_1 + x_2 + k_1 s_1 - \dot{x}_{1d} + \xi e_1 \]
\[ s_3 = f_2 + x_3 + s_1 + k_2 s_2 - \dot{x}_{2d} + \xi e_2 \]
\[ \vdots \]
\[ s_i = f_{i-1} + x_i + s_{i-2} + k_{i-1} s_{i-1} - \dot{x}_{(i-1)d} + \xi e_{i-1}, \]
\[ 3 \leq i \leq n \] (24)

Substituting (19) and (22) into (24) leads to
\[ s_1 = e_1 + \xi \int e_1 dt \]
\[ s_2 = k_1 s_1 + \dot{s}_1 \]
\[ s_3 = \ddot{s}_1 + k_1 \dot{s}_1 + s_1 + k_2 \ddot{s}_2 \]
\[ \vdots \]
\[ s_i = \ddot{s}_{i-1} + \sum_{m=1}^{i} k_m s_{m-i} + \sum_{m=1}^{i-2} s_{m-i-2}, \]
\[ 3 \leq i \leq n \] (25)

**Remarks:**
1. It should be noticed that $s_1$, $i = 2, 3, \ldots, n$ is forced to follow the desired trajectory $x_{id}$, $i = 2, 3, \ldots, n$ which contains an integral term. In this way, singularity-free desired feedback controller can be designed by taking advantage of the integral operation.
2. Actually, (7) and (13) are first order differential equations for $x_{id}$, $i = 2, 3, \ldots, n$, respectively. It is not necessary to solve them, though analytical solutions are available.
3. Considering (25), $s_i$, $i = 1, 2, \ldots, n$ depends on constant coefficients, $s_1$ and its derivative, $-\sum_{m=1}^{n} k_{m} s_{m-n}$ in (21) can be expressed by $s_1$ and its derivative.
4. There exists $\eta = 2 \min(k_1, k_2, \ldots, k_n)$ such that $\dot{V}_n \leq -\eta \dot{V}_n$, then we have $V_n(\tau) \leq V_n(0) \cdot e^{-\eta \tau}$, that is
\[ \sum_{m=1}^{n} \left( e_m(t) + \xi \int_0^t e_m(\tau) d\tau \right)^2 \leq \sum_{m=1}^{n} e_m^2(0) \cdot e^{-\eta t} \] (26)

For high order systems, improved computing power nowadays may be qualified for the task. However, because of filtering technique, dynamic surface control doesn’t have the form of exponential convergence.

### IV. ADAPTIVE NN CONTROL

There have been several studies concerning neural network approximation, using hidden units described by so-called “radial basis functions,” $h(||x - c||)$, where $h$ is some smooth real function of the distance $||x - c||$ from the “center” vector $c$ in the input space [23]. Consider RBF neural networks
\[ S(W, z) = W^T h(z) \] (27)
\[ h_1(z) = [h_1(z), h_2(z), \ldots, h_l(z)]^T \] (28)
\[ h_j(z) = e^{-\frac{||z - c_j||^2}{2\sigma_j^2}}, \quad j = 1, 2, \ldots, l \] (29)

where $b_j$ notes a positive scalar called a width, positive integer $l$ denotes the neural network node number, $W$ is an adjustable network weight vector. An assumption should be made that the reconstruction errors of RBF neural networks are bounded in the following discussions.

Supposing that nonlinear function $g$ and $f_n$ are unknown, let $\dot{W}$ be an estimate of the ideal NN weight $W^*$, and define $\dot{W} = \dot{\dot{W}} - W^*$. It can be proved that the following function approximation holds:
\[ f_n(x) = W_{fa}^T h_n(z) + \mu_{fa} \] (30)
\[ \hat{f}_n(x) = W_{fa}^T h_n(z) \] (31)
\[ \dot{W}_{fa} = \dot{W}_{fa} - W^* \] (32)
\[ g(x) = W_{g}^T h_g(z) + \mu_g \] (33)
\[ \dot{g}(x) = W_{g}^T h_g(z) \] (34)
\[ \dot{W}_{g} = \dot{W}_{g} - W^* \] (35)

where, $z = [x_1, x_2, \ldots, x_n]^T$ is the input of RBF neural network.

In the above section, a desired feedback control input is given
\[ u^* = \frac{1}{\xi} \left[ -k_n s_n - f_n - \sum_{m=1}^{n-1} f_m^{(n-m)} s_m^{(n-m)} + s_m^{(n-m-1)} \right] + \dot{x}_{id} - \xi e_{i-1} \] (36)

Based on it, an adaptive NN controller is designed as follows
\[ u = \frac{1}{\xi} \left[ -k_n s_n - \hat{f}_n - \sum_{m=1}^{n-1} f_m^{(n-m)} s_m^{(n-m)} + s_m^{(n-m-1)} \right] + \dot{x}_{id} - \xi e_{i-1} - M \text{sgn}(s_n) \] (37)

Considering (20), the derivative of $s_n$ can be expressed as
\[ \dot{s}_n = f_n + \hat{g} + (g - \hat{g}) - \dot{x}_{id} + \xi e_n \]
\[ = -k_n s_n - s_{n-1} + f_n - \hat{f}_n + (g - \hat{g}) u - M \text{sgn}(s_n) \]
\[ = -k_n s_n - s_{n-1} - \dot{W}_{fa} h_n(z_1) - \dot{W}_{g} h_g(z) u + \mu_{fa} + \mu_g u - M \text{sgn}(s_n) \] (38)

Choose the following Lyapunov function candidate for the design of the adaptive and control laws
\[ V = V_n + \frac{1}{2} \dot{W}_{fa}^T \Gamma_{fa}^{-1} \dot{W}_{fa} + \frac{1}{2} \dot{W}_{g}^T \Gamma_{g}^{-1} \dot{W}_{g} \] (39)

Using (17) and (38), the time derivative of $V$ satisfies
\[ \dot{V} = -\sum_{m=1}^{n} k_m s_m^2 + s_{n-1} s_n + s_n \dot{s}_n + \dot{W}_{fa}^T \Gamma_{fa}^{-1} \dot{W}_{fa} + \dot{W}_{g}^T \Gamma_{g}^{-1} \dot{W}_{g} \]
\[ = -\sum_{m=1}^{n} k_m s_m^2 + \dot{W}_{fa}^T \left[ \Gamma_{fa}^{-1} \dot{W}_{fa} - s_n h_n(z) \right] \]
\[ + \dot{W}_{g}^T \left[ \Gamma_{g}^{-1} \dot{W}_{g} - s_n u h_g(z) \right] + s_n (\mu_{fa} + \mu_g u) - M |s_n| \] (40)
Then, the updating algorithms for NN weights are chosen as

\[
\dot{\hat{W}}_{f_a} = s_n \Gamma_{f_a} h_f(z) \tag{41}
\]
\[
\dot{\hat{W}}_g = s_n u_1 \Gamma_g h_g(z) \tag{42}
\]

where \( \Gamma_{f_a}, \Gamma_g > 0 \) are diagonal adaptation rate matrices. The parameter of robust term can be set as \( M \geq |\mu_{f_a} + \mu_g u| \), so that asymptotic tracking can be retained.

Similarly, when nonlinear function \( g \) and \( f_i \), \( i = 1, 2, \ldots, n \) are unknown, adaptive NN controller can be designed. Let

\[
f = \sum_{m=1}^{n} f_m^{(n-m)}
\]

we have

\[
f(x) = W_f^T h_f(z) + \mu_f \tag{43}
\]
\[
\hat{f}(x) = \hat{W}_f^T h_f(z) \tag{44}
\]
\[
\hat{W}_f = \hat{W}_f - W_f^* \tag{45}
\]
\[
g(x) = W_g^T h_g(z) + \mu_g \tag{46}
\]
\[
\hat{g}(x) = \hat{W}_g^T h_g(z) \tag{47}
\]
\[
\hat{W}_g = \hat{W}_g - W_g^* \tag{48}
\]

where, \( z = \left[ x_1, \hat{x}_1, \ldots, x_1^{(n-1)} \right]^T \) is the input of RBF neural network.

The design of adaptive and control laws and Lyapunov function candidate are given directly

\[
u = \frac{1}{g} \left[ -\hat{\xi} - \sum_{m=1}^{n} k_m s_m^{(n-m)} - \sum_{m=1}^{n-1} s_m^{(n-m-1)}
\right.
\]
\[
\left. + x_1^{(n)} - \xi_1^{(n-1)} - M \text{sgn} (s_n) \right] \quad \text{(49)}
\]

\[
s_n = -k_n s_n - s_{n-1} + \sum_{m=1}^{n} f_m^{(n-m)} \hat{f} + (g - \hat{g}) u - M \text{sgn} (s_n)
\]
\[
= -k_n s_n - s_{n-1} - \hat{W}_f^T h_f(z) - \hat{W}_g^T h_g(z) u
\]
\[
+ \mu_f + \mu_g u - M \text{sgn} (s_n) \quad \text{(50)}
\]

\[
V = V_n + \frac{1}{2} \hat{W}_f^T \Gamma_f^{-1} \hat{W}_f + \frac{1}{2} \hat{W}_g^T \Gamma_g^{-1} \hat{W}_g
\]
\[
\hat{V} = -\sum_{m=1}^{n} k_m s_m^2 + s_{n-1} s_n + s_n \xi_{n-1} + \hat{W}_f^T \Gamma_f^{-1} \hat{W}_f + \hat{W}_g^T \Gamma_g^{-1} \hat{W}_g
\]
\[
= -\sum_{m=1}^{n} k_m s_m^2 + \hat{W}_f^T \Gamma_f^{-1} \hat{W}_f - s_n h_f(z)
\]
\[
+ \hat{W}_g^T \Gamma_g^{-1} \hat{W}_g - s_n u h_g(z) + s_n \left( \mu_f + \mu_g u \right) - M |s_n| \quad \text{(52)}
\]

\[
\dot{\hat{f}}_f = s_n \Gamma_f h_f(z) \quad \text{(53)}
\]
\[
\dot{\hat{f}}_g = s_n u_1 \Gamma_g h_g(z) \quad \text{(54)}
\]

\[M \geq |\mu_{f_a} + \mu_g u| \quad \text{(55)}
\]

It’s worth mentioning that the chattering phenomena in sliding modes have been studied for many years. Various methods of chattering suppression is introduced in [24].

After RISE-based control [25], [26] came out, the method has been improved and creatively applied in industry [27].

In this paper, by rising \( k_n \) and cutting down \( M \), chattering can be reduced and the stability can be guaranteed in the presence of great disturbance. It’s uniformly ultimate bounded result in theory, but good control effect can be achieved. Simulation results show the feasibility when \( M = 0 \).

Besides, another approach is provided as follows to solve the chattering problem. In (37) and (49), robust term may cause chattering, so a low pass filter can be designed to deal with the drawback brought by the sign function.

The desired continuous control input is given by (16). Let \( e_{n+1} = u - u^* \) denotes the difference between actual and desired control input, then (15) can be rewritten as

\[
\dot{s}_n = f_n + g u^* + g e_{n+1} - \dot{x}_{nd} + \xi e_n
\]
\[
= -k_n s_n - s_{n-1} + g e_{n+1} \quad \text{(56)}
\]

The low pass filter can be expressed as

\[
\dot{\hat{u}} = -(u - r) / T \quad \text{(57)}
\]

where \( r \) is the input of the filter, and \( T \) is time constant.

Desired input of the filter is chosen as

\[
r^* = u - T \left( k_{n+1} e_{n+1} + g s_n - \hat{u}^* \right) \quad \text{(58)}
\]

Let \( p = -u / T - \hat{u}^* \), the following function approximation holds:

\[
p = W^* h(z) + \mu \quad \text{(59)}
\]
\[
\hat{p} = \hat{W}^* h(z) \quad \text{(60)}
\]
\[
\tilde{W} = \hat{W} - W^* \quad \text{(61)}
\]

By utilizing the reconstruction of NN, input of the filter is

\[
r = -T \left[ k_{n+1} e_{n+1} + g s_n + \hat{p} + M \text{sgn} (e_{n+1}) \right] \quad \text{(62)}
\]

where \( e_{n+1} \) is accessible. According to (15) and (16)

\[
e_{n+1} = u - u^* = \left( \frac{1}{g} (s_n + k_n s_n + s_{n-1}) \right) \quad \text{(63)}
\]

Stability analysis is given as follows

\[
\dot{e}_{n+1} = -(u - r) / T - \hat{u}^* \quad \text{(64)}
\]

Choose Lyapunov function candidate

\[
V = V_n + \frac{1}{2} e_{n+1}^2 + \frac{1}{2} \hat{W}^T \Gamma^{-1} \hat{W} \quad \text{(65)}
\]

The time derivative of \( V \) satisfies

\[
\dot{V} = -\sum_{m=1}^{n} k_m s_m^2 + s_{n-1} s_n + s_n \xi_{n+1} + e_{n+1} e_{n+1} + \hat{W}^T \Gamma^{-1} \hat{W}
\]
\[
= -\sum_{m=1}^{n} k_m s_m^2 - k_{n+1} e_{n+1}^2 + \hat{W}^T \left[ \Gamma^{-1} \hat{W} - e_{n+1} h(z) \right]
\]
\[
+ \mu e_{n+1} - M |e_{n+1}| \quad \text{(66)}
\]
Then, the updating algorithm for NN weight is chosen as
\[ \dot{W} = e_{n+1} \Gamma h(z) \]  
(67)
If \( g \) is known, it can be achieved. We can see that input of the filter contains the sign function, but output of the filter is a relatively smooth signal, which is the actual control input.

Remarks: 1. If the unknown nonlinearities are only \( g \) and \( f_i \), the system is observable. So the input of RBF neural network is chosen as \( z = [x_1, x_2, \ldots, x_n]^T \). However, if \( g \) and \( f_i, i = 1, 2, \ldots, n \) are unknown, \( z = \left[ x_1, \dot{x}_1, \ldots, x_1^{(n-1)} \right]^T \) is chosen because every state variable is a function of \( x_1 \) and its derivatives according to system (1).  

2. For systems having local precise model, substituting the accurate parts into \( f \) directly makes it easy for RBF design. For high order systems, the estimation of \( f \) using one RBF neural network may not reconstruct the complex dynamic characteristics well. Number of the RBF neural network depends on the specific system and control effect, though approximation term by term is theoretically possible.

V. SIMULATION STUDY

In this section, two second order plants are considered to verify the effectiveness of the improved scheme.

In the first example, simulation results of our improved scheme are provided in contrast to a traditional method. Consider the following strict-feedback system [17]:

\[
\begin{align*}
\dot{x}_1 &= 0.5x_1 + \left(1 + 0.1x_1^2\right)x_2 \\
\dot{x}_2 &= x_1x_2 + (2 + \cos x_1)u \\
y &= x_1
\end{align*}
\]

The desired trajectory \( y_d \) is generated from the following van der Pol oscillator system:

\[
\begin{align*}
\dot{x}_{d1} &= x_{d2} \\
\dot{x}_{d2} &= -x_{d1} + \beta \left(1 - x_{d1}^2\right)x_{d2} \\
y_d &= x_{d1}
\end{align*}
\]

The initial conditions \([x_1(0), x_2(0)]^T = [1.2, 1.0]^T\) and \([x_{d1}(0), x_{d2}(0)]^T = [1.5, 0.8]^T\). In [17], an ingenious adaptive NN controller is designed using two neural networks containing 25 and 135 nodes respectively. Fig. 1 shows the control effect of applying the controller to the system for tracking desired signal \( y_d \) with \( \beta = 0.2 \).

Based on section II, define \( \bar{x}_1 = x_1, \bar{x}_2 = (1 + 0.1x_1^2)x_2 \). The above system can be transformed into a simplified triangular structure:

\[
\begin{align*}
\dot{\bar{x}}_1 &= 0.5\bar{x}_1 + \bar{x}_2 \\
\dot{\bar{x}}_2 &= \bar{x}_1\bar{x}_2 \left[1 + \frac{0.1(\bar{x}_1 + 2\bar{x}_2)}{1 + 0.1\bar{x}_1^2}\right] + \left(1 + 0.1\bar{x}_1^2\right)(2 + \cos \bar{x}_1)u \\
y &= \bar{x}_1
\end{align*}
\]

As the plant is a second-order system, according to (49), the adaptive NN backstepping shall be chosen as
\[
u = \frac{1}{\xi} \left[-k_2s_2 - s_1 - \hat{f} - k_1\dot{s}_1 + \dot{s}_{id} - \xi\dot{\bar{s}}_1 - M \text{sgn}(s_2)\right]
\]
where, \( s_2 = k_1s_1 + \dot{s}_1 \), \( z = [x_1, \dot{x}_1]^T \).
FIGURE 2. (Continued.) Responses of the adaptive NN controller in this paper.

Fixed-step size is chosen as 0.01s. Fig. 2 shows the simulation results for our designed adaptive NN controller with two neural networks containing 7 and 7 nodes respectively.

Through comparing Fig. 1 with Fig. 2(a), it can be seen that RBF neural networks while simpler in structure, offer a better tracking performance under the improved scheme. The system output \( y \) tracks the desired trajectory \( y_d \) more quickly and precisely.

The second example is used to show the function of integral terms. For a variable length pendulum [28], the plant dynamics can be expressed as follows:

\[
\dot{x}_1 = x_2, \\
\dot{x}_2 = f(x_1, x_2) + g(x_1, x_2)u + d(t), \\
y = x_1, 
\]

where

\[
f = \frac{0.5 \sin(x_1(1 + 0.5 \cos(x_1)))^2 - 10 \sin(x_1(1 + \cos(x_1)))}{0.25(2 + \cos(x_1))}, \\
g = \frac{1}{0.25(2 + \cos(x_1))}, \\
d(t) = D \cdot \cos(3t) \cos(x_1).
\]

The reference signal is

\[
y_d = \begin{cases} 
\pi/6, & t \leq 4\pi s \\
\pi/4 \sin t, & t > 4\pi s.
\end{cases}
\]

As the plant is a second order system, according to (21) and (49), the desired feedback control input and the adaptive NN control input shall be chosen as

\[
u^* = \frac{1}{g} \left[ -k_2 s_2 - s_1 - f - k_1 \dot{s}_1 + \ddot{x}_1d - \dot{\xi} \dot{\dot{e}}_1 - M \text{sgn}(s_2) \right],
\]

where \( s_2 = k_1 s_1 + \dot{s}_1, z = [x_1, \dot{x}_1]^T \).

Take the physical meaning into consideration, \( x_1 \in [-\pi, \pi] \). If circular frequency of \( y_d \) is 1 rad/s, then \( x_2 \in [-\pi, \pi] \). So choose parameters of Gaussian function as

\[
e = \begin{bmatrix} -1.5 & -1.0 & -0.5 & 0.5 & 0.0 & 1.0 & 1.5 \end{bmatrix}^T, \\
b = 15.
\]

Restrict the extent of \( 1/\dot{g}(x) \) for simulation by forcing \( \dot{g}(x) \geq 0.2 \), so that program can run to completion and produce a result. Fixed-step size is chosen as 0.01s.

Based on dynamic characteristics when \( x(0) = [0, 0]^T \), \( k_1 = k_2 = 20, \xi = D = M = 0 \), the parameters of adaptive laws are chosen as \( \Gamma_f = 200I_7, \Gamma_g = 0.1I_7 \), \( \dot{\hat{W}}_f(0) = 0, \dot{\hat{W}}_g(0) = 0 \) to obtain good control effects shown in Fig. 7.

Change \( \xi \) by setting \( \xi = 0 \) and 15 respectively with other parameters being fixed at \( D = 5, k_1 = k_2 = 20, M = 0 \). Fig. 3 and 4 present the simulation results for the desired feedback control. The tracking error given in Fig. 4(a) is restricted in a smaller region than that given in Fig. 3(a). The control input and system states are shown in Fig. 3(b)-(d) and 4(b)-(d).
Keeping design parameters as above, Fig. 5 and 6 present the simulation results for the adaptive NN control with \( \xi = 0 \) and 15 respectively. The tracking error given in Fig. 6(a) is reduced greatly compared with that given in Fig. 5(a). The control input and learning ability of RBF neural networks are shown in Fig. 5(b)-(f) and 6(b)-(f).

The simulation results with \( \xi = 0 \) correspond to the traditional backstepping design, where no integral of tracking error takes effect. As indicated by Fig. 3-6(a), smaller position tracking error can be obtained by suitably choosing the coefficient of integral terms. The controller with \( \xi = 15 \) in Fig. 4 and 6 performs better than the cases where \( \xi = 0 \) in Fig. 3 and 5. The integral terms can assist to meet the requirements of tracking commands in the presence of great disturbance.
Remarks: 1. In the first example, choose $\xi = 0$ because the adding of integral term will go against the fast convergence and less steady residual error. The difference between desired and practical initial conditions may be one of the main causes. Use simpler neural networks with fewer nodes, fairly good
J. Xu et al.: Adaptive NN Backstepping With Considering Integral of Tracking Error

2. Due to many factors, it is hard to set optimal control parameters for controller in engineering practice. When the design of control parameters is inappropriate, adding integral action of tracking error may significantly improve the control effect. When suitably choosing the control parameters, integral terms can still contribute to the precision of control. The second example belong to the latter. Note that simulation results are sensitive to the parameters of neural network. It is hard to just simply compare the adaptive NN backstepping control effects with different $\xi$. Desired feedback control effects with different $\xi$ can be convincing.

3. The output of RBF may be far from the exact value of $f$ affected by large disturbance in the second example. Actually it is the estimate of $f + d(t)$. Neural network control makes breakthrough in control of nonlinear uncertain systems. But it’s difficult to acquire accurate estimation of unknown dynamics in many cases [11]. Reasons for this are manifold. Fundamentally, the adaptive law cannot guarantee the convergence of approximation error. And parameters determination of the neural network mainly depends on experience in system debugging. Moreover, actual operation may not always run on nominal condition due to modeling error as well as occurrence of external disturbance. So the deviation of estimation and tracking error that followed are hard to avoid. Increase of tracking error will further cause deterioration of the approximation effect. Such positive feedback behavior could lead to divergence of the system. Integral action of tracking error introduced into each step of backstepping may help to solve the problem mentioned above sometimes.

4. The reason for the jumps in Fig. 3-7 is the switching of the command signal. So there is a jump for tracking the abruptly altered desired trajectory.

VI. CONCLUSION

In this paper, for a class of nonlinear systems with a simplified triangular structure, a systematic design of adaptive neural network controllers has been developed. We incorporate backstepping technique into neural network based adaptive control design framework in a different way. During this process, integral action of tracking error takes effect and the Lyapunov function with integral term contained in it can still converge to zero exponentially. Two simulation examples indicate that our improved scheme can reduce the design complexity of neural network controller, and improve the precision of control. System stability and better asymptotic tracking are shown to be maintained in the improved scheme.

REFERENCES

[1] I. Kanellakopoulos, P. V. Kokotovic, and A. S. Morse, “Systematic design of adaptive controllers for feedback linearizable systems,” IEEE Trans. Autom. Control, vol. 36, no. 11, pp. 1241–1253, Nov. 1991.
[2] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, Nonlinear and Adaptive Control Design. New York, NY, USA: Wiley, 1995.
[3] D. Swaroop, J. K. Hedrick, P. P. Yip, and J. C. Gerdes, “Dynamic surface control for a class of nonlinear systems,” IEEE Trans. Autom. Control, vol. 45, no. 10, pp. 1893–1899, Oct. 2000.
[4] Q. Hu and Y. Meng, “Adaptive backstepping control for air-breathing hypersonic vehicle with actuator dynamics,” *Aerosp. Sci. Technol.*, vol. 67, pp. 412–421, Aug. 2017.

[5] E. Frazzoli, M. A. Dahleh, and E. Feron, “Trajectory tracking control design for autonomous helicopters using a backstepping algorithm,” in *Proc. Amer. Control Conf.*, Chicago, IL, USA, 2000, pp. 4102–4107.

[6] C. Shen, Y. Shi, and B. Buckham, “Trajectory tracking control of an autonomous underwater vehicle using Lyapunov-based model predictive control,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5796–5805, Jul. 2018.

[7] W. Dong, “Flocking of multiple mobile robots based on backstepping,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 2, pp. 414–424, Apr. 2011.

[8] W. Chen, S. S. Ge, J. Wu, and M. Gong, “ Globally stable adaptive backstepping neural network control for uncertain strict-feedback systems with tracking accuracy known a priori,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 1842–1854, Sep. 2015.

[9] D. S. Broomhead and D. Lowe, “ Multivariable functional interpolation and adaptive networks,” *Complex Syst.*, vol. 2, no. 3, pp. 321–355, 1988.

[10] F. Girosi and T. Poggio, “ Networks and the best approximation property,” *Biol. Cybern.*, vol. 63, no. 3, pp. 169–176, Jul. 1990.

[11] J. Liu, *RBF Neural Network Control for Mechanical Systems*, 2nd ed. Beijing, China: Tsinghua Univ. Press, 2018, p. 2, 109–111, 179–185, and 337–347.

[12] Z.-G. Wu, J. Lam, H. Su, and J. Chu, “ Stability and dissipativity analysis of static neural networks with time delay,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 2, pp. 199–210, Feb. 2012.

[13] Z. Feng and W. X. Zheng, “ On extended dissipativity of discrete-time neural networks with time delay,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3293–3300, Dec. 2015.

[14] T. Zhang, “ Adaptive neural network control for strict-feedback nonlinear systems using backstepping design,” *Automatica*, vol. 36, no. 12, pp. 1835–1846, Dec. 2000.

[15] P. Liu, H. Yu, and S. Cang, “ Adaptive neural network tracking control for underactuated systems with matched and mismatched disturbances,” *Nonlinear Dyn.*, pp. 1447–1464, Oct. 2019.

[16] Y. Guo and J. Liu, “ Neural network based adaptive dynamic surface control for flight path angle,” in *Proc. IEEE 51st IEEE Conf. Decis. Control*, Maui, HI, USA, Dec. 2012, pp. 5374–5379.

[17] S. S. Ge and C. Wang, “ Direct adaptive NN control of a class of nonlinear systems,” *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 214–221, Jan. 2002.

[18] X. Shi, Y. Cheng, C. Yin, X. Huang, and S.-M. Zhong, “Design of adaptive backstepping dynamic surface control method with RBF neural network for uncertain nonlinear system,” *Neurocomputing*, vol. 330, pp. 490–503, Feb. 2019.

[19] 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. Neural Netw.*, vol. 16, no. 1, pp. 195–202, Jan. 2005.

[20] A. Kojić and A. M. Annaswamy, “Adaptive control of nonlinearly parameterized systems with a triangular structure,” *Automatica*, vol. 38, no. 1, pp. 115–123, Jan. 2002.

[21] M. Sharma, “Flight-path angle control via neuro-adaptive backstepping,” in *Proc. AIAA Guid., Navigat., Control Conf. Exhib.*, Monterey, CA, USA, Aug. 2002, p. 4451.

[22] J. Liu, *Robot Control System Design and MATLAB Simulation*, 1st ed. Beijing, China: Tsinghua Univ. Press, 2011.

[23] E. J. Hartman, J. D. Keeler, and J. M. Kowalski, “Layered neural networks with Gaussian hidden units as universal approximations,” *Neural Comput.*, vol. 2, no. 2, pp. 210–215, Jun. 1990.

[24] H. Lee and V. I. Utkin, “Chattering suppression methods in sliding mode control systems,” *Annu. Rev. Control*, vol. 31, no. 2, pp. 179–188, Jan. 2007.

[25] B. Xian, D. M. Dawson, M. S. de Queiroz, and J. Chen, “A continuous asymptotic tracking control strategy for uncertain nonlinear systems,” *IEEE Trans. Autom. Control*, vol. 49, no. 7, pp. 1206–1211, Jul. 2004.

[26] C. Makkar, G. Hu, W. G. Sawyer, and W. E. Dixon, “Lyapunov-based tracking control in the presence of uncertain nonlinear parameterizable friction,” *IEEE Trans. Autom. Control*, vol. 52, no. 10, pp. 1988–1994, Oct. 2007.

[27] Z. Yao, J. Yao, and W. Sun, “Adaptive RISE control of hydraulic systems with multilayer neural-networks,” *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8638–8647, Nov. 2019.

[28] S. S. Ge, C. C. Hang, and T. Zhang, “A direct method for robust adaptive nonlinear control with guaranteed transient performance,” *Syst. Control Lett.*, vol. 37, no. 5, pp. 275–284, Aug. 1999.