Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

RBF Neural Network Adaptive Backstepping Controllers for MIMO Nonaffine Nonlinear Systems

Wei-Yen Wang
Department of Applied Electronics Technology
National Taiwan Normal University
Taipei, Taiwan.
wywang@ntnu.edu.tw

Chin-Ming Hong
Department of Applied Electronics Technology
National Taiwan Normal University
Taipei, Taiwan
t07026@ntnu.edu.tw

Ming-Feng Kuo
Department of Applied Electronics Technology
National Taiwan Normal University
Taipei, Taiwan.
696750081@ntnu.edu.tw

Yih-Guang Leu
Department of Applied Electronics Technology
National Taiwan Normal University
Taipei, Taiwan.
leuyk@ntnu.edu.tw

Tsu-Tian Lee
Department of Electrical Engineering
National Taipei University of Technology
Taipei, Taiwan.
president@ntut.edu.tw

Abstract—This paper proposes a radial basis function neural network adaptive backstepping controller (RBFNN_ABC) for multiple-input multiple-output (MIMO) nonlinear systems in block-triangular form. The control scheme incorporates the adaptive neural backstepping design technique with a first-order filter at each step of the backstepping design to avoid the higher-order derivative problem, which is generated by the backstepping design. This problem may create an unpredictable and unfavorable influence on control performance because higher-order derivative term errors are introduced into the neural approximation model. Finally, simulation results demonstrate that the output tracking error between the plant output and the desired reference can be made arbitrarily small.

***Keywords*—Radial basis function (RBF) neural networks (NNs), adaptive, backstepping, MIMO nonlinear systems.**

I. INTRODUCTION

Adaptive control is a useful method for designing controllers for uncertain dynamic systems. The main idea in adaptive control is using output feedback to model-free the unknown system [1]-[2]. Adaptive controllers are classified into two types: direct and indirect. Direct adaptive control means that the parameters of the controller are directly adjusted to reduce the norm of the output error between the plant and the reference model. Indirect adaptive control means that the parameters of the plant are estimated and the controller is chosen assuming that the estimated parameters represent the true values of the plant parameters [3].

Compared to feedback linearization methods [4], the backstepping technique [5], [6] has the advantage of avoiding the cancellation of useful nonlinearities in the design process. Thus, in the past decade, the backstepping technique has been widely used for nonlinear control systems. In brief, to design a backstepping system and an appropriate state and virtual control are selected for each smaller subsystem. Then the state equation is rewritten in terms of them. Finally, Lyapunov functions are chosen for these subsystems so that the true controller integrating the individual controls of these subsystems guarantees the stability of the overall system. Recently, owing to the development of intelligent control methods such as fuzzy logic control, neural network control, etc., many intelligent backstepping methods [7-12] have been proposed
to control nonlinear systems with unknown system dynamics, combining intelligent control methods with an adaptive backstepping design.

A backstepping-based neural network controller for unknown nonlinear systems first came out in [13]. In [14], a robust and adaptive backstepping controller using a radial basis function neural network (RBFNN) was proposed. Adaptive neural control of uncertain MIMO nonlinear systems was proposed in [15]. In [16], an RBF neural network adaptive backstepping control system (RBFNN_ABC) was proposed to control a nonlinear system. However, the RBF adaptive backstepping control method results in the controller containing higher-order derivative terms as the order (n) of the system increases. The higher-order derivative terms introduced into the approximation model may produce an unpredictable and unfavorable influence on control performance. To solve the problems mentioned above, this paper proposes a filtered RBFNN adaptive backstepping control scheme. It consists of the backstepping design which achieves the desired control behavior, the RBF neural network which is utilized to estimate the unknown system dynamics, the adaptive control scheme which is utilized to adjust the controller parameters, and a first-order filter at each step of the backstepping design which is chosen to avoid producing higher-order derivative terms.

This paper is organized as follows. The system problem is formulated in section II. Design of the adaptive neural backstepping controller is described in section III. In section

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

IV, simulation results are given. Finally, we draw some conclusions in section V.

II. PROBLEM FORMULATION

The model of an uncertain MIMO block-triangular system can be described as shown in (1)

$$\begin{cases} \dot{x}_{1,1} = f_{1,1}(x_{1,1}, x_{2,1}, x_{2,2}, x_{1,2}) \\ \dot{x}_{1,2} = f_{1,2}(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}, x_{2,3}, u_1) \\ \dot{x}_{2,1} = f_{2,1}(x_{2,1}, x_{2,2}) \\ \dot{x}_{2,2} = f_{2,2}(x_{1,1}, x_{2,1}, x_{2,2}, x_{2,3}) \\ \dot{x}_{2,3} = f_{2,3}(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}, x_{2,3}, u_1, u_2) \\ y_k = x_{k,1} \qquad k = 1,2 \end{cases} \tag{1}$$

where $\{x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}, x_{2,3}\}$ are the states, $u_k \in R$ is the input, $y_k \in R$ is the output. Functions $f_{k,i}$,($k=1,2$, $i=1,2,3$), are unknown smooth continuous functions.

The control objective is to design an adaptive neural backstepping controller for system (1) such that all the signals in the closed-loop system are uniformly ultimately bounded and the state $x_{k,1}$ can track a bounded reference signal x_{dk} as close as possible.

III. DESIGN OF ADAPTIVE NEURAL BACKSTEPPING CONTROLLER

A. The first subsystem

Step 1: Define the tracking error $z_{1,1} = x_{1,1} - x_{d1}$, where x_{d1} is the reference signal of the first subsystem. The derivative of $z_{1,1}$ is defined as

$$\dot{z}_{1,1} = \dot{x}_{1,1} - \dot{x}_{d1} = f_{1,1} - \dot{x}_{d1} = F_{1,1}(x_{1,1}, x_{2,1}, x_{1,2}, x_{2,2}, \dot{x}_{d1}) \tag{2}$$

Let a new function $\psi_{1,1}(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}, \dot{x}_{d1}, x_{d1}) = F_{1,1} + c_{1,1}z_{1,1}$, where $c_{1,1}$ is a positive design constant, then $\dot{z}_{1,1} = \psi_{1,1} - c_{1,1}z_{1,1}$. There exists an ideal virtual controller $\alpha^*_{1,1}(x_{1,1}, x_{2,1}, x_{2,2}, \dot{x}_{d1}, x_{d1})$, such that $\psi_{1,1}(x_{1,1}, \alpha^*_{1,1}, x_{2,1}, x_{2,2}, \dot{x}_{d1}, x_{d1}) = 0$

By using the mean value theorem, equation (2) can rewritten as

$$\begin{aligned} \dot{z}_{1,1} &= \psi_{1,1} - c_{1,1}z_{1,1} \\ &= \psi_{1,1}(x_{1,1}, \alpha^*_{1,1}, x_{2,1}, x_{2,2}, \dot{x}_{d1}, x_{d1}) + \frac{\partial \psi_{1,1}}{\partial x_{1,2}}\Big|_{x_{1,2}=x^{\lambda}_{1,2}} (x_{1,2} - \alpha^*_{1,1}) - c_{1,1}z_{1,1} \\ &= 0 + \frac{\partial \psi_{1,1}}{\partial x_{1,2}}\Big|_{x_{1,2}=x^{\lambda}_{1,2}} (x_{1,2} - \alpha^*_{1,1}) - c_{1,1}z_{1,1} \\ &= \frac{\partial (F_{1,1} + c_{1,1}z_{1,1})}{\partial x_{1,2}}\Big|_{x_{1,2}=x^{\lambda}_{1,2}} (x_{1,2} - \alpha^*_{1,1}) - c_{1,1}z_{1,1} \\ &= \frac{\partial f_{1,1}}{\partial x_{1,2}}\Big|_{x_{1,2}=x^{\lambda}_{1,2}} (x_{1,2} - \alpha^*_{1,1}) - c_{1,1}z_{1,1} \\ &= g^{\lambda}_{1,1}(x_{1,2} - \alpha^*_{1,1}) - c_{1,1}z_{1,1} \end{aligned} \tag{3}$$

where $x^{\lambda}_{1,2}$ is the point between $\alpha^*_{1,1}$ and $x_{1,2}$, such that $\psi_{1,1}(x_{1,2}) - \psi_{1,1}(\alpha^*_{1,1}) / x_{1,2} - \alpha^*_{1,1} = \partial \psi_{1,1} / \partial x_{1,2}\big|_{x_{1,2}=x^{\lambda}_{1,2}}$, and $g^{\lambda}_{1,1} = \partial f_{1,1} / \partial x_{1,2}\big|_{x_{1,2}=x^{\lambda}_{1,2}}$.

By employing an RBF neural network $\mathbf{W}^{*T}_{1,1}\boldsymbol{\varphi}_{1,1}(\mathbf{b}_{1,1})$ to approximate $\alpha^*_{1,1}$, $\alpha^*_{1,1}$ can be expressed as

$$\begin{aligned} \alpha^*_{1,1} &= \mathbf{W}^{*T}_{1,1}\boldsymbol{\varphi}_{1,1}(\mathbf{b}_{1,1}) - \delta_{1,1} \\ &= \mathbf{W}^{*T}_{1,1}\boldsymbol{\varphi}_{1,1}(\mathbf{b}_{1,1}) - c_{1,1}z_{1,1} + c_{1,1}z_{1,1} - \delta_{1,1} \\ &= \mathbf{W}^{*T}_{1,1}\boldsymbol{\varphi}_{1,1}(\mathbf{b}_{1,1}) - c_{1,1}z_{1,1} + \tau_{1,1} \end{aligned} \tag{4}$$

where $\tau_{1,1} = c_{1,1}z_{1,1} - \delta_{1,1}$,is the signal error and $\mathbf{W}^{*T}_{1,1}$ denotes ideal constant weights, $\boldsymbol{\varphi}_{1,1}$ is the basis function, $\mathbf{b}_{1,1}$=[$x_{1,1}$ $x_{2,1}$ $x_{2,2}$ x_{d1} $\dot{x}_{d1}$] denotes the RBF input, and $\delta_{1,1}$ is the approximation error.

The virtual controller, $\alpha_{1,1}$, is defined as

$$\alpha_{1,1} = \hat{\mathbf{W}}^{T}_{1,1}\boldsymbol{\varphi}_{1,1}(\mathbf{b}_{1,1}) - c_{1,1}z_{1,1} \tag{5}$$

where $\hat{\mathbf{W}}^{T}_{1,1}$ is the estimation of $\mathbf{W}^{*T}_{1,1}$.

The adaptation law for $\hat{\mathbf{W}}^{T}_{1,1}$ is

$$\dot{\hat{\mathbf{W}}}_{1,1} = \Gamma_{1,1}[-\boldsymbol{\varphi}_{1,1}(\mathbf{b}_{1,1})z_{1,1} - \sigma_{1,1}\hat{\mathbf{W}}_{1,1}] \tag{6}$$

where $\sigma_{1,1}$>0 and $\Gamma_{1,1}$>0 are design constants. Let $\hat{\mathbf{W}}^{T}_{1,1}\boldsymbol{\varphi}_{1,1}(\mathbf{b}_{1,1})$ pass through a first-order filter to obtain $\gamma_{1,1}$. Thus, we have

$$\xi_{1,1}\dot{\gamma}_{1,1} + \gamma_{1,1} = \hat{\mathbf{W}}^{T}_{1,1}\boldsymbol{\varphi}_{1,1}(\mathbf{b}_{1,1}) \tag{7}$$

where $\xi_{1,1}$ is the time constant. Then the virtual controller, $\alpha_{1,1}$, is redefined as

$$\alpha_{1,1} = -c_{1,1}z_{1,1} + \gamma_{1,1} \tag{8}$$

Step 2: Define the tracking error $z_{1,2} = x_{1,2} - \alpha_{1,1}$. The derivative of $z_{1,2}$ is defined as

$$\begin{aligned} \dot{z}_{1,2} &= \dot{x}_{1,2} - \dot{\alpha}_{1,1} = f_{1,2} - \dot{\alpha}_{1,1} \\ &= F_{1,2}(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}, x_{2,3}, u_1, \dot{\alpha}_{1,1}) \end{aligned} \tag{9}$$

Let a new function $\psi_{1,2}(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}, x_{2,3}, u_1, \dot{\alpha}_{1,1}, \alpha_{1,1}, x_{d1}) = F_{1,2} + c_{1,2}z_{1,2} + g^{\lambda}_{1,2}z_{1,1}$ where $c_{1,2}$ is a design positive constant, and $g^{\lambda}_{1,2} = \partial f_{1,2} / \partial u_1\big|_{u_1=u^{\lambda}_1}$, then $\dot{z}_{1,2} = \psi_{1,2} - c_{1,2}z_{1,2} - g^{\lambda}_{1,2}z_{1,1}$. There exists an ideal controller $u^*_1(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}, x_{2,3}, \dot{\alpha}_{1,1}, \alpha_{1,1}, x_{d1})$, such that $\psi_{1,2}(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}, x_{2,3}, u^*_1, \dot{\alpha}_{1,1}, \alpha_{1,1}, x_{d1}) = 0$

By using the mean value theorem, equation (9) can rewritten as

$$
\begin{aligned}
\dot{z}_{1,2} =& \ \psi_{1,2} - c_{1,2}z_{1,2} - g_{1,2}^{\lambda} z_{1,1} \\
=& \ \psi_{1,2}(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}, x_{2,3}, u_1^*, \dot{\alpha}_{1,1}, \alpha_{1,1}, x_{d1}) \\
& + \frac{\partial \psi_{1,2}}{\partial u_1}\Big|_{u_1=u_1^{\lambda}} (u_1 - u_1^*) - c_{1,2}z_{1,2} - g_{1,2}^{\lambda} z_{1,1} \\
=& \ 0 + \frac{\partial \psi_{1,2}}{\partial u_1}\Big|_{u_1=u_1^{\lambda}} (u_1 - u_1^*) - c_{1,2}z_{1,2} - g_{1,2}^{\lambda} z_{1,1} \\
=& \ \frac{\partial (F_{1,2} + c_{1,2}z_{1,2})}{\partial u_1}\Big|_{u_1=u_1^{\lambda}} (u_1 - u_1^*) - c_{1,2}z_{1,2} - g_{1,2}^{\lambda} z_{1,1} \\
=& \ \frac{\partial f_{1,2}}{\partial u_1}\Big|_{u_1=u_1^{\lambda}} (u_1 - u_1^*) - c_{1,2}z_{1,2} - g_{1,2}^{\lambda} z_{1,1} \\
=& \ g_{1,2}^{\lambda}(u_1 - u_1^*) - c_{1,2}z_{1,2} - g_{1,2}^{\lambda} z_{1,1}
\end{aligned} \tag{10}
$$

where u_1^{λ} is the point between u_1^* and u_1, such that $\psi_{1,2}(u_1) - \psi_{1,2}(u_1^*) / u_1 - u_1^* = \partial \psi_{1,2} / \partial u_1 \big|_{u_1=u_1^{\lambda}}$.

By employing an RBF neural network, $\mathbf{W}_{1,2}^{*T}\boldsymbol{\varphi}_{1,2}(\mathbf{b}_{1,2})$ to approximate u_1^*, u_1^* can be expressed as

$$
\begin{aligned}
u_1^* &= \mathbf{W}_{1,2}^{*T}\boldsymbol{\varphi}_{1,2}(\mathbf{b}_{1,2}) - \delta_{1,2} \\
&= \mathbf{W}_{1,2}^{*T}\boldsymbol{\varphi}_{1,2}(\mathbf{b}_{1,2}) - c_{1,2}z_{1,2} + c_{1,2}z_{1,2} - \delta_{1,2} \\
&= \mathbf{W}_{1,2}^{*T}\boldsymbol{\varphi}_{1,2}(\mathbf{b}_{1,2}) - c_{1,2}z_{1,2} + \tau_{1,2}
\end{aligned} \tag{11}
$$

where $\tau_{1,2} = c_{1,2}z_{1,2} - \delta_{1,2}$, is the signal error and $\mathbf{W}_{1,2}^{*T}$ denotes ideal constant weights, $\boldsymbol{\varphi}_{1,2}$ is the basis function, $\mathbf{b}_{1,2}=[x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}, x_{2,3}, \dot{\alpha}_{1,1}, \alpha_{1,1}, x_{d1}]$ denotes the RBF input, and $\delta_{1,2}$ is the approximation error. The controller, u_1, is defined as

$$
u_1 = \mathbf{W}_{1,2}^{*T}\boldsymbol{\varphi}_{1,2}(\mathbf{b}_{1,2}) - c_{1,2}z_{1,2} \tag{12}
$$

where $\hat{\mathbf{W}}_{1,2}^{T}$ is the estimation of $\mathbf{W}_{1,2}^{*T}$.

The adaptation law for $\hat{\mathbf{W}}_{1,2}^{T}$ is

$$
\dot{\hat{\mathbf{W}}}_{1,2} = \Gamma_{1,2}[-\boldsymbol{\varphi}_{1,2}(\mathbf{b}_{1,2})z_{1,2} - \sigma_{1,2}\hat{\mathbf{W}}_{1,2}] \tag{13}
$$

where $\sigma_{1,2}$>0 and $\Gamma_{1,2}$>0 are design constants.

B. The second subsystem

Step 1: Define the tracking error $z_{2,1} = x_{1,1} - x_{d2}$, where x_{d2} is the reference signal of the second subsystem. The derivative of $z_{2,1}$ is defined as

$$
\dot{z}_{2,1} = \dot{x}_{2,1} - \dot{x}_{d2} = f_{2,1} - \dot{x}_{d2} = F_{2,1}(x_{2,1}, x_{2,2}, \dot{x}_{d2}) \tag{14}
$$

Let a new function $\psi_{2,1}(x_{2,1}, x_{2,2}, \dot{x}_{d2}, x_{d2}) = F_{2,1} + c_{2,1}z_{2,1}$, where $c_{2,1}$ is a design positive constant, then $\dot{z}_{2,1} = \psi_{2,1} - c_{2,1}z_{2,1}$.There exists an ideal virtual controller $\alpha_{2,1}^*(x_{2,1}, \dot{x}_{d2}, x_{d2})$, such that $\psi_{2,1}(x_{2,1}, \alpha_{2,1}^*, \dot{x}_{d2}, x_{d2}) = 0$

By using the mean value theorem, equation (14) can rewritten as

$$
\begin{aligned}
\dot{z}_{2,1} &= \psi_{2,1} - c_{2,1}z_{2,1} \\
&= \psi_{2,1}(x_{2,1}, \alpha_{2,1}^*, \dot{x}_{d2}, x_{d2}) + \frac{\partial \psi_{2,1}}{\partial x_{2,2}}\Big|_{x_{2,2}=x_{2,2}^{\lambda}} (x_{2,2} - \alpha_{2,1}^*) - c_{2,1}z_{2,1} \\
&= 0 + \frac{\partial \psi_{2,1}}{\partial x_{2,2}}\Big|_{x_{2,2}=x_{2,2}^{\lambda}} (x_{2,2} - \alpha_{2,1}^*) - c_{2,1}z_{2,1} \\
&= \frac{\partial (F_{2,1} + c_{2,1}z_{2,1})}{\partial x_{2,2}}\Big|_{x_{2,2}=x_{2,2}^{\lambda}} (x_{2,2} - \alpha_{2,1}^*) - c_{2,1}z_{2,1} \\
&= \frac{\partial f_{2,1}}{\partial x_{2,2}}\Big|_{x_{2,2}=x_{2,2}^{\lambda}} (x_{2,2} - \alpha_{2,1}^*) - c_{2,1}z_{2,1} \\
&= g_{2,1}^{\lambda}(x_{2,2} - \alpha_{2,1}^*) - c_{2,1}z_{2,1}
\end{aligned} \tag{15}
$$

where $x_{2,2}^{\lambda}$ is the point between $\alpha_{2,1}^*$ and $x_{2,2}$, such that $\psi_{2,1}(x_{2,2}) - \psi_{2,1}(\alpha_{2,1}^*) / x_{2,2} - \alpha_{2,1}^* = \partial \psi_{2,1} / \partial x_{2,2} \big|_{x_{2,2}=x_{2,2}^{\lambda}}$, and $g_{2,1}^{\lambda} = \partial f_{2,1} / \partial x_{2,2} \big|_{x_{2,2}=x_{2,2}^{\lambda}}$.

By employing an RBF neural network $\mathbf{W}_{2,1}^{*T}\boldsymbol{\varphi}_{2,1}(\mathbf{b}_{2,1})$ to approximate $\alpha_{2,1}^*$, $\alpha_{2,1}^*$ can be expressed as

$$
\begin{aligned}
\alpha_{2,1}^* &= \mathbf{W}_{2,1}^{*T}\boldsymbol{\varphi}_{2,1}(\mathbf{b}_{2,1}) - \delta_{2,1} \\
&= \mathbf{W}_{2,1}^{*T}\boldsymbol{\varphi}_{2,1}(\mathbf{b}_{2,1}) - c_{2,1}z_{2,1} + c_{2,1}z_{2,1} - \delta_{2,1} \\
&= \mathbf{W}_{2,1}^{*T}\boldsymbol{\varphi}_{2,1}(\mathbf{b}_{2,1}) - c_{2,1}z_{2,1} + \tau_{2,1}
\end{aligned} \tag{16}
$$

where $\tau_{2,1} = c_{2,1}z_{2,1} - \delta_{2,1}$, is the signal error and $\mathbf{W}_{2,1}^{*T}$ denotes ideal constant weights, $\boldsymbol{\varphi}_{2,1}$ is the basis function, $\mathbf{b}_{2,1}=[x_{2,1}, \dot{x}_{d2}, x_{d2}]$ denotes the RBF input, and $\delta_{2,1}$ is the approximation error.

The virtual controller, $\alpha_{2,1}$, is defined as

$$
\alpha_{2,1} = \hat{\mathbf{W}}_{2,1}^{T}\boldsymbol{\varphi}_{2,1}(\mathbf{b}_{2,1}) - c_{2,1}z_{2,1} \tag{17}
$$

where $\hat{\mathbf{W}}_{2,1}^{T}$ is the estimation of $\mathbf{W}_{2,1}^{*T}$.

The adaptation law for $\hat{\mathbf{W}}_{2,1}^{T}$ is

$$
\dot{\hat{\mathbf{W}}}_{2,1} = \Gamma_{2,1}[-\boldsymbol{\varphi}_{2,1}(\mathbf{b}_{2,1})z_{2,1} - \sigma_{2,1}\hat{\mathbf{W}}_{2,1}] \tag{18}
$$

where $\sigma_{2,1}$>0 and $\Gamma_{2,1}$>0 are design constants.

Let $\hat{\mathbf{W}}_{2,1}^{T}\boldsymbol{\varphi}_{2,1}(\mathbf{b}_{2,1})$ pass through a first-order filter to obtain $\gamma_{2,1}$. Thus, we have

$$
\xi_{2,1}\dot{\gamma}_{2,1} + \gamma_{2,1} = \hat{\mathbf{W}}_{2,1}^{T}\boldsymbol{\varphi}_{2,1}(\mathbf{b}_{2,1}) \tag{18}
$$

where $\xi_{2,1}$ is the time constant. Then the virtual controller, $\alpha_{2,1}$, is redefined as

$$
\alpha_{2,1} = -c_{2,1}z_{2,1} + \gamma_{2,1}
$$

Step 2: Define the tracking error $z_{2,2} = x_{2,2} - \alpha_{2,1}$. The derivative of $z_{2,2}$ is defined as

$$
\dot{z}_{2,2} = \dot{x}_{2,2} - \dot{\alpha}_{2,1} = f_{2,2} - \dot{\alpha}_{2,1} = F_{2,2}(x_{1,1}, x_{2,1}, x_{2,2}, x_{2,3}, \dot{\alpha}_{2,1}) \tag{19}
$$

Let a new function $\psi_{2,2}(x_{1,1}, x_{2,1}, x_{2,2}, x_{2,3}, \dot{\alpha}_{2,1}, \alpha_{2,1}, x_{d2}) = F_{2,2} + c_{2,2}z_{2,2}$, where $c_{2,2}$ is a design positive constant, then $\psi_{2,2} = F_{2,2} + c_{2,2}z_{2,2} + g_{2,2}^{\lambda}z_{2,1}$.There exists an ideal virtual controller $\alpha_{2,2}^{*}(x_{1,1},x_{1,2},x_{2,1},x_{2,2},\dot{\alpha}_{2,1},\alpha_{2,1},x_{d2})$, such that $\psi_{2,2}(x_{1,1},x_{2,1},x_{2,2},\alpha_{2,2}^{*},\dot{\alpha}_{2,1},\alpha_{2,1},x_{d2})=0$

By using the mean value theorem, equation (19) can rewritten as

$$\begin{aligned}\dot{z}_{2,2}&=\psi_{2,2}-c_{2,2}z_{2,2}-z_{2,1}\\&=\psi_{2,2}(x_{1,1},x_{2,1},x_{2,2},\alpha_{2,2}^{*},\dot{\alpha}_{2,1},\alpha_{2,1},x_{d2})+\frac{\partial\psi_{2,2}}{\partial x_{2,3}}\Big|_{x_{2,3}=x_{2,3}^{\lambda}}(x_{2,3}-\alpha_{2,2}^{*})\\&\quad-c_{2,2}z_{2,2}-g_{2,2}^{\lambda}z_{2,1}\\&=0+\frac{\partial\psi_{2,2}}{\partial x_{2,3}}\Big|_{x_{2,3}=x_{2,3}^{\lambda}}(x_{2,3}-\alpha_{2,2}^{*})-c_{2,2}z_{2,2}-g_{2,2}^{\lambda}z_{2,1}\\&=\frac{\partial(F_{2,2}+c_{2,2}z_{2,2})}{\partial x_{2,3}}\Big|_{x_{2,3}=x_{2,3}^{\lambda}}(x_{2,3}-\alpha_{2,2}^{*})-c_{2,2}z_{2,2}-g_{2,2}^{\lambda}z_{2,1}\\&=\frac{\partial f_{2,2}}{\partial x_{2,3}}\Big|_{x_{2,3}=x_{2,3}^{\lambda}}(x_{2,3}-\alpha_{2,2}^{*})-c_{2,2}z_{2,2}-g_{2,2}^{\lambda}z_{2,1}\\&=g_{2,2}^{\lambda}(x_{2,3}-\alpha_{2,2}^{*})-c_{2,2}z_{2,2}-g_{2,2}^{\lambda}z_{2,1}\end{aligned}\tag{20}$$

where $x_{2,3}^{\lambda}$ is the point between $\alpha_{2,2}^{*}$ and $x_{2,3}$, such that $\psi_{2,2}(x_{2,3})-\psi_{2,2}(\alpha_{2,2}^{*})/x_{2,3}-\alpha_{2,2}^{*}=\partial\psi_{2,2}/\partial x_{2,3}\Big|_{x_{2,3}=x_{2,3}^{\lambda}}$, and $g_{2,2}^{\lambda}=\partial f_{2,2}/\partial x_{2,3}\Big|_{x_{2,3}=x_{2,3}^{\lambda}}$.

By employing an RBF neural network $\mathbf{W}_{2,2}^{*T}\boldsymbol{\varphi}_{2,2}(\mathbf{b}_{2,2})$ to approximate $\alpha_{2,2}^{*},\alpha_{2,2}^{*}$ can be expressed as

$$\begin{aligned}\alpha_{2,2}^{*}&=\mathbf{W}_{2,2}^{*T}\boldsymbol{\varphi}_{2,2}(\mathbf{b}_{2,2})-\delta_{2,2}\\&=\mathbf{W}_{2,2}^{*T}\boldsymbol{\varphi}_{2,2}(\mathbf{b}_{2,2})-c_{2,2}z_{2,2}+c_{2,2}z_{2,2}-\delta_{2,2}\\&=\mathbf{W}_{2,2}^{*T}\boldsymbol{\varphi}_{2,2}(\mathbf{b}_{2,2})-c_{2,2}z_{2,2}+\tau_{2,2}\end{aligned}\tag{21}$$

where $\tau_{2,2}=c_{2,2}z_{2,2}-\delta_{2,2}$, is the signal error and $\mathbf{W}_{2,2}^{*T}$ denotes ideal constant weights, $\boldsymbol{\varphi}_{2,2}$ is the basis function, $\mathbf{b}_{2,2}$=$[x_{1,1},x_{1,2},x_{2,1},x_{2,2},\dot{\alpha}_{2,1},\alpha_{2,1},x_{d2}]$ denotes the RBF input, and $\delta_{2,2}$ is the approximation error.

The virtual controller, $\alpha_{2,2}$, is defined as

$$\alpha_{2,2}=\hat{\mathbf{W}}_{2,2}^{T}\boldsymbol{\varphi}_{2,2}(\mathbf{b}_{2,2})-c_{2,2}z_{2,2}\tag{22}$$

where $\hat{\mathbf{W}}_{2,2}^{T}$ is the estimation of $\mathbf{W}_{2,2}^{*T}$.

The adaptation law for $\hat{\mathbf{W}}_{2,2}^{T}$ is

$$\dot{\hat{\mathbf{W}}}_{2,2}=\Gamma_{2,2}[-\boldsymbol{\varphi}_{2,2}(\mathbf{b}_{2,2})z_{2,2}-\sigma_{2,2}\hat{\mathbf{W}}_{2,2}]\tag{23}$$

where $\sigma_{2,2}$>0 and $\Gamma_{2,2}$>0 are design constants.

Let $\hat{\mathbf{W}}_{2,2}^{T}\boldsymbol{\varphi}_{2,2}(\mathbf{b}_{2,2})$ pass through a first-order filter to obtain $\gamma_{2,2}$. Thus, we have

$$\xi_{2,2}\dot{\gamma}_{2,2}+\gamma_{2,2}=\hat{\mathbf{W}}_{2,2}^{T}\boldsymbol{\varphi}_{2,2}(\mathbf{b}_{2,2})\tag{24}$$

where $\xi_{2,2}$ is the time constant. Then the virtual controller, $\alpha_{2,2}$, is redefined as

$$\alpha_{2,2}=-c_{2,2}z_{2,2}+\gamma_{2,2}\tag{25}$$

Step 3: Define the tracking error $z_{2,3}=x_{2,3}-\alpha_{2,2}$. The derivative of $z_{2,3}$ is defined as

$$\dot{z}_{2,3}=\dot{x}_{2,3}-\dot{\alpha}_{2,2}=f_{2,3}-\dot{\alpha}_{2,2}=F_{2,3}(x_{1,1},x_{1,2},x_{2,1},x_{2,2},x_{2,3},u_{1},u_{2},\dot{\alpha}_{2,2})\tag{26}$$

Let a new function $\psi_{2,3}(x_{1,1},x_{1,2},x_{2,1},x_{2,2},x_{2,3},u_{1},u_{2},\dot{\alpha}_{2,2},\alpha_{2,2},\alpha_{2,1})=F_{2,3}+c_{2,3}z_{2,3}+g_{2,3}^{\lambda}z_{2,2}$, where $c_{2,3}$ is a design positive constant, then $\dot{z}_{2,3}=\psi_{2,3}-c_{2,3}z_{2,3}-g_{2,3}^{\lambda}z_{2,2}$. There exists an ideal controller $u_{2}^{*}\{x_{1,1},x_{1,2},x_{2,1},x_{2,2},x_{2,3},u_{1},\dot{\alpha}_{2,2},\alpha_{2,2},\alpha_{2,1}\}$, such that $\psi_{2,3}(x_{1,1},x_{1,2},x_{2,1},x_{2,2},x_{2,3},u_{1},u_{2}^{*},\dot{\alpha}_{2,2},\alpha_{2,2},\alpha_{2,1})$=0

By using the mean value theorem, equation (26) can rewritten as

$$\begin{aligned}\dot{z}_{2,3}&=\psi_{2,3}-c_{2,3}z_{2,3}-g_{2,3}^{\lambda}z_{2,3}\\&=\psi_{2,3}(x_{1,1},x_{1,2},x_{2,1},x_{2,2},x_{2,3},u_{1},u_{2}^{*},\dot{\alpha}_{2,2},\alpha_{2,2},\alpha_{2,1})\\&\quad+\frac{\partial\psi_{2,3}}{\partial u_{2}}\Big|_{u_{2}=u_{2}^{\lambda}}(u_{2}-u_{2}^{*})-c_{2,3}z_{2,3}-g_{2,3}^{\lambda}z_{2,2}\\&=0+\frac{\partial\psi_{2,3}}{\partial u_{2}}\Big|_{u_{2}=u_{2}^{\lambda}}(u_{2}-u_{2}^{*})-c_{2,3}z_{2,3}-g_{2,3}^{\lambda}z_{2,2}\\&=\frac{\partial(F_{2,3}+c_{2,3}z_{2,3})}{\partial u_{2}}\Big|_{u_{2}=u_{2}^{\lambda}}(u_{2}-u_{2}^{*})-c_{2,3}z_{2,3}-g_{2,3}^{\lambda}z_{2,2}\\&=\frac{\partial f_{2,3}}{\partial u_{2}}\Big|_{u_{2}=u_{2}^{\lambda}}(u_{2}-u_{2}^{*})-c_{2,3}z_{2,3}-g_{2,3}^{\lambda}z_{2,2}\\&=g_{2,3}^{\lambda}(u_{2}-u_{2}^{*})-c_{2,3}z_{2,3}-g_{2,3}^{\lambda}z_{2,2}\end{aligned}\tag{27}$$

where u_{2}^{λ} is the point between u_{2}^{*} and u_{2}, such that $\psi_{2,3}(u_{2})-\psi_{2,3}(u_{2}^{*})/u_{2}-u_{2}^{*}=\partial\psi_{2,3}/\partial u_{2}\Big|_{u_{2}=u_{2}^{\lambda}}$.

By employing an RBF neural network, $\mathbf{W}_{2,3}^{*T}\boldsymbol{\varphi}_{2,3}(\mathbf{b}_{2,3})$ to approximate u_{2}^{*},u_{2}^{*} can be expressed as

$$\begin{aligned}u_{2}^{*}&=\mathbf{W}_{2,3}^{*T}\boldsymbol{\varphi}_{2,3}(\mathbf{b}_{2,3})-\delta_{2,3}\\&=\mathbf{W}_{2,3}^{*T}\boldsymbol{\varphi}_{2,3}(\mathbf{b}_{2,3})-c_{2,3}z_{2,3}+c_{2,3}z_{2,3}-\delta_{2,3}\\&=\mathbf{W}_{2,3}^{*T}\boldsymbol{\varphi}_{2,3}(\mathbf{b}_{2,3})-c_{2,3}z_{2,3}+\tau_{2,3}\end{aligned}\tag{28}$$

where $\tau_{2,3}=c_{2,3}z_{2,3}-\delta_{2,3}$, is the signal error and $\mathbf{W}_{2,3}^{*T}$ denotes ideal constant weights, $\boldsymbol{\varphi}_{2,3}$ is the basis function, $\mathbf{b}_{2,3}$=$[x_{1,1},x_{1,2},x_{2,1},x_{2,2},x_{2,3},u_{1},\dot{\alpha}_{2,2},\alpha_{2,2},\alpha_{2,1}]$ denotes the RBF input, and $\delta_{2,3}$ is the approximation error.

The controller, u_{2}, is defined as

$$u_{2}=\hat{\mathbf{W}}_{2,3}^{\mathrm{T}}\boldsymbol{\varphi}_{2,3}(\mathbf{b}_{2,3})-c_{2,3}z_{2,3}\tag{29}$$

where $\hat{\mathbf{W}}_{2,3}^{\mathrm{T}}$ is the estimation of $\mathbf{W}_{2,3}^{*T}$.

The adaptation law for $\hat{\mathbf{W}}_{2,3}^{\mathrm{T}}$ is

$$\dot{\hat{\mathbf{W}}}_{2,3}=\Gamma_{2,3}[-\boldsymbol{\varphi}_{2,3}(\mathbf{b}_{2,3})z_{2,3}-\sigma_{2,3}\hat{\mathbf{W}}_{2,3}]\tag{30}$$

where $\sigma_{2,3}$>0 and $\Gamma_{2,3}$>0 are design constants.

Theorem 1: Consider the closed-loop system (1), The control input is defined as (12) and (29), and the adaptation law is given as (13) and (30). Then all the signals in the closed-loop system remain bound and the state $x_{k,1}$ can track a bound reference signal $x_{d,k}$ arbitrarily closely.

Proof: To be omitted for matching the requirement of the length.

IV. SIMULATION RESULTS

This section presents the simulation results of the proposed controller, showing that the tracking error of the closed-loop system can be made arbitrarily small.

Example1 : Consider an MIMO nonlinear system

$$\begin{cases} \dot{x}_{1,1} = 2x_{1,2} + \sin(x_{1,1}x_{1,2}x_{2,1}x_{2,2}) \\ \dot{x}_{1,2} = 2u_1 + \cos(x_{1,1}x_{1,2}x_{2,1}x_{2,2}x_{2,3}u_1) \\ \dot{x}_{2,1} = 2x_{2,2} + \sin(x_{2,1}x_{2,2}) \\ \dot{x}_{2,2} = 2x_{2,3} + \cos(x_{1,1}x_{2,1}x_{2,2}x_{2,3}) \\ \dot{x}_{2,3} = 2u_1 + \cos(x_{1,1}x_{1,2}x_{2,1}x_{2,2}x_{2,3}u_1u_2) \\ y_i = x_{i,1} \qquad i = 1,2 \end{cases} \tag{31}$$

The reference model is a van der Pol oscillator

$$\begin{cases} \dot{x}_{d1} = x_{d2} \\ \dot{x}_{d2} = -x_{d1} + \beta(1 - x_{d1}^2)x_{d2} \\ y_{di} = x_{di} \qquad i = 1,2 \end{cases} \tag{32}$$

The reference model is taken from [15], where u_1 and u_2 are the control inputs. The initial state is $[x_{1,1}(0), x_{1,2}(0), x_{2,1}(0), x_{2,2}(0), x_{2,3}(0)]^T = [0.1, 0.2, 0.7, 0.1, 0.2]^T$, and $[x_{d1}(0), x_{d2}(0)]^T = [1.5, 0.8]^T$. The initial weights are $\hat{\mathbf{W}}_{1,1}(0) = \hat{\mathbf{W}}_{1,2}(0) = \hat{\mathbf{W}}_{2,1}(0) = \hat{\mathbf{W}}_{2,2}(0) = \hat{\mathbf{W}}_{2,3}(0) = 0$. The design parameters are selected as $c_{1,1} = 8$, $c_{1,2} = 12$, $c_{2,1} = 12$, $c_{2,2} = 22$, $c_{2,3} = 22$, $\Gamma_{1,1} = \Gamma_{1,2} = \Gamma_{2,1} = \Gamma_{2,2} = \Gamma_{2,3} = diag\{2\}$, and $\sigma_{1,1} = \sigma_{1,2} = \sigma_{2,1} = \sigma_{2,2} = \sigma_{2,3} = 0.1$. The first neural network $\hat{W}_{1,1}^T\phi_{1,1}(b_{1,1})$ has 32 centers which range over the interval [-2.5,2.5]×[-2.5,2.5]×[-2.5,2.5]×[-2.5,2.5] ×[-2.5,2.5]. The second neural network $\hat{W}_{1,2}^T\phi_{1,2}(b_{1,2})$ has 256 centers which range over the interval [-2.5,2.5]×[-2.5,2.5]×[-2.5,2.5]×[-2.5,2.5] ×[-2.5,2.5]×[-2.5,2.5]×[-2.5,2.5]×[-2.5,2.5]. The third neural network $\hat{W}_{2,1}^T\phi_{2,1}(b_{2,1})$ has 8 centers which range over the interval [-2.5,2.5]×[-2.5,2.5]×[-2.5,2.5]. The fourth neural network $\hat{W}_{2,2}^T\phi_{2,2}(b_{2,2})$ has 64 centers which range over the interval [-2.5,2.5]×[-2.5,2.5]×[-2.5,2.5] ×[-5,5] ×[-2.5,2.5]×[-2.5,2.5]. The fifth neural network $\hat{W}_{2,3}^T\phi_{2,3}(b_{2,3})$ has 512 centers which range over the interval [-2.5,2.5]×[-2.5,2.5] ×[-2.5,2.5]×[-2.5,2.5] ×[-2.5,2.5] ×[-2.5,2.5] ×[-5,5] ×[-2.5,2.5]×[-2.5,2.5]. The state $x_{1,1}$ of this system and the model reference signal x_{d1} are shown in Fig.1. The state $x_{2,1}$ of this system and the model reference signal x_{d2} are shown in Fig. 2. The responses of control inputs u_1 and u_2 are shown in Fig. 3. The simulation results show that the states $x_{1,1}$ and $x_{2,1}$ can track the model reference signals x_{d1} and x_{d2} **arbitrarily** closely.

V. CONCLUSION

In this paper, **a** radial basis function neural network **adaptive backstepping control** scheme has been proposed to control **MIMO nonlinear systems in** block-triangular **form.** We added a first-order filter at each step of the backstepping design to avoid the higher-order derivative problem generated by the backstepping design. The proposed controller possesses the advantages of avoiding the higher-order derivative effect on the approximation model and eliminating the need to compute complicated derivatives. Simulation results have shown that the proposed RBF **adaptive backstepping control** scheme can rapidly learn unknown system dynamics, and achieve favorable tracking performance.

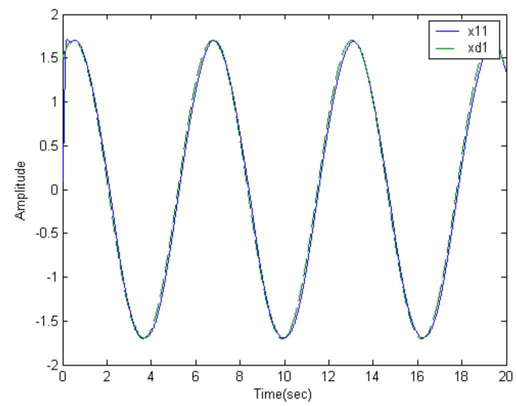

Fig. 1 the state of system $x_{1,1}$ and model reference signal x_{d1}

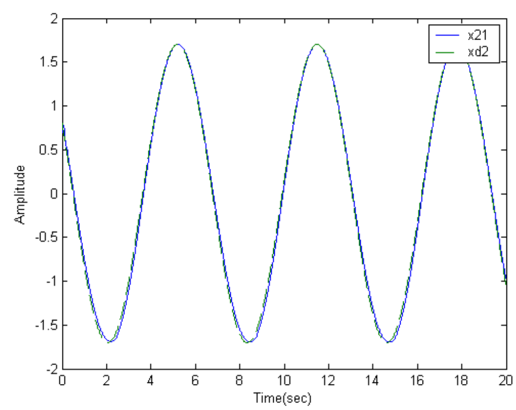

Fig. 2 the state of system $x_{2,1}$ and model reference signal x_{d2}

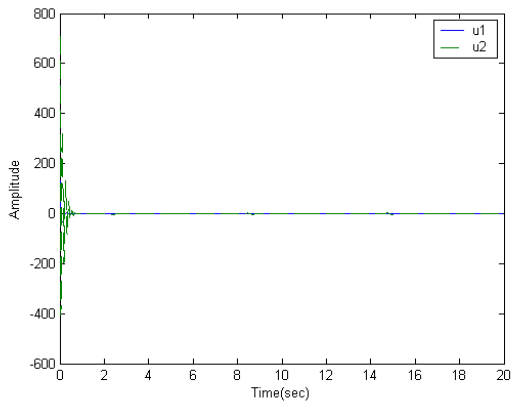

Fig. 3 control inputs u_1 and u_2

REFERENCES

[1] M.Hojati and S.Gazor, "Hybrid Adaptive Fuzzy Identification and Control of Nonlinear Systems". *IEEE Transactions On Fuzzy Systems*, vol. 10, no. 2, April 2002

[2] Choi, J.Y.; Farrell, J.A."Nonlinear adaptive control using networks of piecewise linear approximators".Proceedings of 38thConference on Decision & Control Phoenix, Arizona USA. December 1999

[3] L.X.Wang,"Stable adaptive fuzzycontrollers with application to Inverted pendulum tracking "*IEEE Transactions On Fuzzy Systems*, Man, And Cybernetics-Part B: Cybernetics, vol. 26, no. 5, October 1996

[4] A. Isidori, *Nonlinear Control System*. New York: Springer-Verlag, 1989.

[5] M. Krstic, I. Kanellakopoulos, and P.V.Kokotovic, *Nonlinear and Adaptive Control Design*. New York: Wiley, 1995.

[6] I. Kanellakopoulos, P. V. Kokotovic, and A. S. Morse, "Systematic design of adaptive controller for feedback linearizable system," *IEEE Transactions Automat. Contr.*, vol. 36, pp. 1241–1253, 1991.

[7] C. Kwan and F. L. Lewis, "Robust backstepping control of nonlinear systems using neural networks," *IEEE Transactions Syst., Man, Cybern. A*, vol. 30, pp. 753–765, 2000.

[8] T. Knohl and H. Unbehauen, "ANNNAC—extension of adaptive backstepping algorithm with artificial neural networks," *Inst. Elect. Eng. Proc. Contr. Theory Appl.*, vol. 147, pp. 177–183, 2000.

[9] C. M. Kwan and F. L. Lewis, "Robust backstepping control of induction motors using neural networks," *IEEE Transactions Neural Networks*, vol. 11, pp. 1178–1187, 2000.

[10] J. Y. Choi and J. A. Farrell, "Adaptive observer backstepping control using neural networks," *IEEE Transactions Neural Networks*, vol. 12, pp. 1103–1112, 2001.

[11] Y. Zhang, P.Y. Peng, and Z.P. Jiang, "Stable Neural Controller Design for Unknown Nonlinear Systems Using Backstepping," *IEEE Transactions on Neural Networks*, vol. 11, no. 6, November 2000.

[12] C.F. Hsu, C.M. Lin, and T.T. Lee, "Wavelet Adaptive Backstepping Control for a Class of Nonlinear Systems," *IEEE Transactions on Neural Networks*, vol. 17, no. 5, September 2006.

[13] Y.Zhang, P.Y.Peng, Z.P.Jiang, "Stable Neural Controller Design for Unknown Nonlinear Systems Using Backstepping," *IEEE Transactions on Neural Networks*, vol. 11, no. 6, November 2000.

[14] Y.Li, S.Qiang, X.Zhuang, and O.Kaynak, "Robust and Adaptive Backstepping Control for Nonlinear Systems Using RBF Neural Networks," *IEEE Transactions on Neural Networks,* vol. 15, no. 3, May 2004.

[15] S.S.Ge, C.Wang, "Adaptive Neural Control of Uncertain MIMO Nonlinear Systems," *IEEE Transactions on Neural Networks*, vol. 15, no.3, may 2004.

[16] D.Wang and J.Hung, "Neural Network-Based Adaptive Dynamic Surface Control for a Class of Uncertain Nonlinear Systems in Strict-Feedback Form," *IEEE Transactions on Neural Networks*, vol. 15, no.1, January 2005.