

# Direct Adaptive Backstepping Control for a Class of MIMO Non-affine Systems Using Recurrent Neural Networks

Ching-Hung Lee, *Member, IAENG*, Jen-Chieh Chien, Hao-Han Chang, Che-Ting Kuo, and Hua-Hsiang Chang

**Abstract**—This paper proposes a direct adaptive backstepping control scheme for a class of multi-input-multi-output nonlinear uncertain non-affine systems using output recurrent wavelet neural networks (ORWNNs), called DABC<sub>ORWNN</sub>. The proposed ORWNN combines the advantages of wavelet-based neural network, fuzzy neural network (FNN), and output feedback layer. For the tracking of nonlinear non-affine systems with non-triangular form, we first transform it into a strict-feedback-like form. Subsequently, the neural network based backstepping controller is developed. The ideal virtual controllers and actual controller are approximated by ORWNNs. In addition, a robust controller is designed to compensate the approximated error of ORWNNs. Based on the Lyapunov approach, the adaptive laws and stability analysis of closed-loop system is obtained. Finally, simulation results of non-affine double-pendulums system is shown to demonstrate the performance of our approaches.

**Index Terms**— Multiple-input-multiple-output, nonlinear non-affine system, backstepping, wavelet neural network, adaptive control, Lyapunov theorem

## I. INTRODUCTION

In this paper, the backstepping technique is used to design adaptive controller for a class of MIMO nonlinear uncertain non-affine systems. The backstepping technique provides a systematic framework and recursive design methodology for nonlinear systems [1, 4, 18]. The design procedure treats the state variables as virtual control inputs; then, the virtual controllers is designed step by step. Finally, the actual control input can be obtained. It illustrates the stability by Lyapunov stability theorem. However, the major constraint is that the system functions must be exactly known. If the internal uncertainty and external disturbance exist, then they may result in an unstable system. Therefore, we propose an output-recurrent wavelet neural network (ORWNN) system to approximate the unknown functions to solve this problem. Previous literatures developed fuzzy systems and neural networks to approximate the unknown functions [1-4, 6, 20]. Many researchers have shown that using wavelet basis can achieves superior performance in network size and learning ability [6, 13, 24]. Therefore, wavelet functions are combined with the fuzzy neural network to construct the wavelet based neural network. In order to meet our requirement, the output

feedback scheme is used to develop an output recurrent wavelet neural network (ORWNN) [22-23].

Recently, newly neural network based backstepping control schemes were proposed for nonlinear affine uncertain systems [1, 4]. According to the idea of literatures [3, 8, 21], the non-affine form system can be transferred into strict-feedback-like form system, in which each subsystem can be viewed as an affine-like system. Thus, there exists stabilizing controller for this type of transforming system by implicit function theorem [21, 30]. According to the results of [3, 8, 21], the MIMO nonlinear, non-affine, and non-triangular can be transferred into a like MIMO strict-feedback-like system; therefore, it can simplify the complexity of controller design, e.g., non-affine double-pendulum system [28]. Besides, these ideas can also cope with MIMO affine form systems. Literature [29] proposed a NN-based controller to deal with the state-feedback linearizable system, and two NNs to be used to approximate two unknown functions, i.e., gain matrix function and uncertain functions in system dynamic. The direct adaptive backstepping control using output-recurrent wavelet neural network, DABC<sub>ORWNN</sub>, is proposed to deal with a class of MIMO nonlinear uncertain non-affine systems. The ORWNNs are used to learn the ideal virtual controllers and actual controller [23, 25-26]. A robust controller is designed to attenuate effect of all the unmodeled dynamic, modeling errors, and external disturbances on tracking error. According to the Lyapunov stability approach, the adaptive laws and stability of closed-loop system are guaranteed.

This paper is organized as follows. Section II introduces the problem formulation and the proposed output recurrent wavelet neural network (ORWNN) system. The DABC<sub>ORWNN</sub> control schemes are introduced in Section III. Section IV shows the simulation results of three-order non-affine system in non-triangular form are shown to demonstrate the performance of the proposed DABC<sub>ORWNN</sub>. Finally, conclusion is given.

## II. PRELIMINARIES

### A. Problem Formulation

Consider the following MIMO nonlinear uncertain system in state-space representation

$$\begin{aligned}\dot{\mathbf{x}}_i &= \mathbf{F}_i(\bar{\mathbf{x}}_i, \mathbf{x}_{i+1}, \mathbf{d}_i), \text{ for } i=1, \dots, n-1, \\ \dot{\mathbf{x}}_n &= \mathbf{F}_n(\bar{\mathbf{x}}_n, \mathbf{u}, \mathbf{d}_n), \\ \mathbf{y} &= \mathbf{x}_1,\end{aligned}\quad (1)$$

where  $\mathbf{x}_i = [x_{1,i}, x_{2,i}, \dots, x_{m,i}]^T \in \mathfrak{R}^m$ ,  $i=1, \dots, n$ , are the vector of denoting the states of system (1),  $\mathbf{F}_i(\bar{\mathbf{x}}_i, \mathbf{x}_{i+1}, \mathbf{d}_i) = [F_{1,i}(\bar{x}_{1,i}, x_{1,i+1}, d_{1,i}), F_{2,i}(\bar{x}_{2,i}, x_{2,i+1}, d_{2,i}), \dots,$

This work was supported in part by the National Science Council, Taiwan, R.O.C., under contracts NSC-95-2221-E-155-068-MY2.

Ching-Hung Lee is with Department of Electrical Engineering, Yuan-Ze University, Chung-li, Taoyuan 320, Taiwan. (phone: +886-3-4638800, ext: 7119; fax: +886-3-4639355, e-mail: chlee@saturn.yzu.edu.tw).

$F_{m,i}(\bar{x}_{m,i}, x_{m,i+1}, d_{m,i})^T \in \mathfrak{R}^m$ ,  $i = 1, \dots, n-1$ , are smooth vector function,  $\mathbf{u} = [u_1, u_2, \dots, u_m]^T \in \mathfrak{R}^m$  is the control input,  $\mathbf{y} = [y_{1,1}, y_{2,1}, \dots, y_{m,1}]^T \in \mathfrak{R}^m$  is the system output,  $\mathbf{d}_i = [d_{1,i}, d_{2,i}, \dots, d_{m,i}]^T$ ,  $i = 1, 2, \dots, n$ , denotes the external bounded disturbance satisfying  $\|\mathbf{d}_i\| \leq \rho_i$ , in which  $\|\cdot\|$  is the Euclidean norm and  $\rho_i > 0$ . Herein, system state variables are also assumed to be measurable and  $\mathbf{x} = \mathbf{0}$  is an equilibrium point. Throughout this study, the following assumptions to be needed ensure the controllability of the system (1).

**Assumption 1:** The inequalities  $\frac{\partial F_{i,j}(\bar{x}_{i,j}, x_{i,j+1})}{\partial x_{i,j+1}} > 0$ , hold  $\forall \bar{x}_i \in \mathfrak{R}^i$ ,  $i = 1, 2, \dots, m-1, j = 1, \dots, n-1$ .

**Assumption 2:** The inequality  $\frac{\partial F_{i,n}(\bar{x}_{i,n}, \mathbf{u})}{\partial u_i} > 0$ ,  $i = 1, \dots, m$ , holds for all  $(\bar{x}_{i,n}, \mathbf{u}) \in \Omega_x \times \mathfrak{R}^m$  with a controllability region  $\Omega_x$ , i.e.,

$$i = 1, \frac{\partial F_{1,n}(\bar{x}_{1,n}, u_1)}{\partial u_1} > 0, i = 2, \frac{\partial F_{2,n}(\bar{x}_{2,n}, u_1, u_2)}{\partial u_1} > 0, \\ \frac{\partial F_{2,n}(\bar{x}_{2,n}, u_1, u_2)}{\partial u_2} > 0, \text{ and } \dots i = m, \\ \frac{\partial F_{m,n}(\bar{x}_{m,n}, u_1, \dots, u_m, d_{m,n})}{\partial u_1} > 0, \dots, \\ \frac{\partial F_{m,n}(\bar{x}_{m,n}, u_1, \dots, u_m, d_{m,n})}{\partial u_m} > 0.$$

**Assumption 3:** The designed trajectory vector  $\mathbf{y}_d$  is smooth and bounded.

The control objective is to design the control input  $\mathbf{u}$  such that the output  $\mathbf{y}$  follows a desired trajectory vector  $\mathbf{y}_d$ . Our proposed controller design method is based on the concept of [3, 8, 21]. Herein, we extend the SISO non-affine nonlinear system to MIMO non-affine nonlinear system in triangular form. It can be represented as follows:

$$\mathbf{y}^{(n)} = \mathbf{F}(\mathbf{y}, \mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n-1)}, \mathbf{u}) \quad (2)$$

where  $\mathbf{x} = [x_1, x_2, \dots, x_n] = [\mathbf{y}, \mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n-1)}] \in \mathfrak{R}^n$ ,  $\mathbf{u} \in \mathfrak{R}^m$  and  $\mathbf{F}$  is smooth vector function. The reference signal  $\mathbf{y}_d$  and its time derivatives  $\mathbf{y}_d^{(1)}, \mathbf{y}_d^{(2)}, \dots, \mathbf{y}_d^{(n)}$  are assumed to be bounded. Define the tracking error as  $\mathbf{e}_1 = \mathbf{y}_d - \mathbf{y}$  and the corresponding error vector as  $\mathbf{e} = [\mathbf{e}_1, \mathbf{e}_1^{(1)}, \dots, \mathbf{e}_1^{(n-1)}]$ . Herein, we rewrite (2)

$$\mathbf{y}^{(n)} = \mathbf{F}(\mathbf{x}, \mathbf{u}) = c\mathbf{u} + \{\mathbf{F}(\mathbf{x}, \mathbf{u}) - c\mathbf{u}\} = c\mathbf{u} + \mathbf{H}(\mathbf{x}, \mathbf{u}) \quad (3)$$

where  $c$  is a design constant and  $\mathbf{H}(\mathbf{x}, \mathbf{u}) = \mathbf{F}(\mathbf{x}, \mathbf{u}) - c\mathbf{u}$ . The feedback linearization control input of (4) can be determined as

$$\mathbf{u} = \frac{1}{c}(\mathbf{u}_{flc} - \mathbf{u}_{rnn} + \mathbf{u}_{rc}) \quad (4)$$

where  $\mathbf{u}_{flc}$  is a control input to stabilize linearized dynamic,  $\mathbf{u}_{rnn}$  is an adaptive recurrent neural network (RNN) control signal designed to cancel  $\mathbf{H}(\mathbf{x}, \mathbf{u})$ , and  $\mathbf{u}_{rc}$  is an additional robust control input to compensate the approximation error. Substituting (4) into (3) yields

$$\mathbf{y}^{(n)} = \mathbf{u}_{flc} + \{\mathbf{H}(\mathbf{x}, \mathbf{u}) - \mathbf{u}_{rnn}\} + \mathbf{u}_{rc} \quad (5)$$

We design  $\mathbf{u}_{flc}$  as

$$\mathbf{u}_{flc} = \mathbf{y}_d^{(n)} + \mathbf{k}\mathbf{e} \quad (6)$$

where  $\mathbf{k} = [\mathbf{k}_n \dots \mathbf{k}_1]$ , substituting (4) and (6) into (3) and yields

$$\mathbf{e}^n + \mathbf{k}_1 \mathbf{e}^{n-1} + \dots + \mathbf{k}_n = 0 \quad (7)$$

which implies that  $\lim_{t \rightarrow \infty} \mathbf{e}(t) = 0$ . This can be done by choosing proper  $\mathbf{k}$  so that all roots of the polynomial that  $\mathbf{s}^n + \mathbf{k}_1 \cdot \mathbf{s}^{(n-1)} + \dots + \mathbf{k}_n = 0$  are located in the open left-half plane. Thus, if  $\mathbf{H}(\mathbf{x}, \mathbf{u})$  is perfectly canceled by  $\mathbf{u}_{rnn}$ , i.e.,  $\mathbf{H}(\mathbf{x}, \mathbf{u}) = \mathbf{u}_{rnn}$ , and  $\mathbf{u}_{rc} = 0$ , the closed-loop system is stable. As the discussion above, a RNN controller should be employed to approximate  $\mathbf{H}(\mathbf{x}, \mathbf{u})$ . The inputs to the RNN are  $\mathbf{x}$  and  $\mathbf{u}$  ( $\mathbf{u} = \mathbf{u}_{flc} - \mathbf{u}_{rnn} + \mathbf{u}_{rc}$ ). Obviously, the output of the RNN  $\mathbf{u}_{rnn}$  is directly fed into RNN to produce the control input  $\mathbf{u}$ . According to the implicit function theorem [21, 30], there exists a set  $\Omega_x \subset \mathfrak{R}^n$  and unique  $\mathbf{u}_{rnn}^*$  which is a function of  $\mathbf{x}$  and  $\mathbf{u}_\alpha = \mathbf{u}_{flc} + \mathbf{u}_{rc}$ , such that  $\mathbf{u}_{rnn}^*(\mathbf{x}, \mathbf{u}_\alpha)$  satisfies for all  $(\mathbf{x}, \mathbf{u}_\alpha) \in \Omega_x \times \mathfrak{R}^m$ .

## B. Output Recurrent Wavelet Neural Network (ORWNN)

In this paper, to achieve highly approximated accuracy and speed up the convergence, the FNN is modified as a novel wavelet-based NN. Herein, we combine the advantages of FNN with wavelet functions to propose a four-layer output recurrent wavelet neural network (ORWNN). The schematic diagram is depicted in Fig. 1, in which  $z^{-1}$  denotes a unit time delay. This ORWNN is composed of an input layer, a wavelet layer, a hidden layer, an output layer, and a recurrent layer.

Unlike the Gaussian membership functions used in conventional FNNs, wavelet functions are spatially localized. Therefore, the learning of ORWNN is more efficient than FNNs in function approximation. Herein, the Gaussian membership functions are replaced by wavelet basis functions, and the self-recurrent layer is replaced by output recurrent layer. We indicate the signal propagation and the function of every node in each layer.

### Layer 1: Input layer & feedback layer

The inputs of this layer are the current network input  $\mathbf{x}(t)$  and past network output  $\mathbf{y}(t-1)$  with weighting vectors  $\boldsymbol{\theta}_r$ , where  $\mathbf{x} = [x_1, x_2, \dots, x_i, \dots, x_m]^T \in \mathfrak{R}^m$ ,  $m$  is the input number. The output of this layer is

$$\mathbf{x}_r(t) = \mathbf{x}(t) + \boldsymbol{\theta}_r \mathbf{y}(t-1) \quad (8)$$

where  $\mathbf{x}_r = [x_{r1}, x_{r2}, \dots, x_{ri}, \dots, x_{rm}]^T \in \mathfrak{R}^m$  and  $\boldsymbol{\theta}_r = [\theta_{r1}, \dots, \theta_{ri}, \dots, \theta_{rm}]$ ,  $m_o$  denotes the number of network output. It is clear that ORWNN contains the output term  $\mathbf{y}(t-1)$  which stores the past information of network.

### Layer 2: Hidden layer 1 (Wavelet Layer)

Each node in this layer performs a wavelet function. Herein, the Gaussian wavelet function  $\mu(z) = \cos \omega z \exp(-z^2/2)$  is adopted as the activation function, where  $\omega$  is the selected frequency. Hence,

$$\mu_{ik} \left( \frac{x_{ri} - m_{ik}}{\sigma_{ik}} \right) = \cos(\omega \cdot \frac{x_{ri} - m_{ik}}{\sigma_{ik}}) \cdot \exp\left[-\frac{1}{2} \left( \frac{x_{ri} - m_{ik}}{\sigma_{ik}} \right)^2\right],$$

$$k = 1, 2, \dots, m_b \quad (9)$$

where  $m_{ik}$  and  $\sigma_{ik}$  are the translations and dilation in the  $k$ th term of the  $i$ th input  $x_{ri}$  to the node of the mother wavelet layer, respectively.

### Layer 3: Hidden layer 2

In this layer, each node calculates the product of all input signals, i.e.,

$$\psi_k(\mathbf{x}_r, \mathbf{m}_k, \boldsymbol{\sigma}_k, \boldsymbol{\theta}_r) = \prod_{i=1}^{m_b} \mu_{ik}, \quad k = 1, \dots, m_b \quad (10)$$

where  $\mathbf{m}_k = [m_{1k}, \dots, m_{2k}, \dots, m_{m_b k}]^T \in \mathcal{R}^{m_b}$ ,

$\boldsymbol{\sigma}_k = [\sigma_{1k}, \dots, \sigma_{2k}, \dots, \sigma_{m_b k}]^T \in \mathcal{R}^{m_b}$ . It can be expressed in a vector notation as

$$\boldsymbol{\psi}(x_r, m, \sigma, \theta_r) = [\psi_1, \psi_2, \dots, \psi_{m_b}]^T \in \mathcal{R}^{m_b} \quad (11)$$

where  $\mathbf{m} = [m_1^T, m_2^T, \dots, m_k^T, \dots, m_{m_b}^T] \in \mathcal{R}^{m \times m_b}$  and

$\boldsymbol{\sigma} = [\sigma_1^T, \sigma_2^T, \dots, \sigma_k^T, \dots, \sigma_{m_b}^T] \in \mathcal{R}^{m \times m_b}$ .

### Layer 4: Output layer

Each node calculates the linear combination of input variables. Therefore, the  $p$ th output is

$$y_p = w_p^T \boldsymbol{\psi}(x, m, \sigma, w_r) = \sum_{k=1}^{m_b} w_{kp} \psi_k, \quad p = 1, 2, \dots, m_o \quad (12)$$

where  $w_{kp}$  denotes the connecting activated weight value of the  $p$ th output associated with the  $k$ th layer. In vector representation

$$\mathbf{y} = [y_1, \dots, y_p, \dots, y_{m_o}]^T = \mathbf{w}^T \boldsymbol{\psi} \quad (13)$$

According to the above introduction, the ORWNN has adjustable parameters  $\mathbf{m}$ ,  $\boldsymbol{\sigma}$ ,  $\boldsymbol{\theta}_r$ , and  $\mathbf{w}$ . The architecture of ORWNN used in this paper is designed to have the advantages of network with dynamic characteristics.

## III. DIRECT ADAPTIVE ORWNNs CONTROL VIA BACKSTEPPING DESIGN TECHNIQUE

Most of literatures using backstepping approach are limited to the feedback linearizable nonlinear systems, i.e., the unknown nonlinearities must satisfy matching condition [1-2, 26]. For nonlinear non-affine system, they are not valid. As above discussion, we rewrite nonlinear non-affine system (2) as follows affine-like form to simplify our design approach

$$\dot{\mathbf{x}}_i = \mathbf{H}_i(\bar{\mathbf{x}}_i, \mathbf{x}_{i+1}, \mathbf{d}_i) + \mathbf{x}_{i+1}, \quad \text{for } i = 1, \dots, n-1,$$

$$\dot{\mathbf{x}}_n = \mathbf{H}_n(\bar{\mathbf{x}}_n, \mathbf{u}, \mathbf{d}_n) + \mathbf{u} \quad (14)$$

$$\mathbf{y} = \mathbf{x}_1$$

where  $\mathbf{H}_i(\cdot)$ ,  $i=1, \dots, n$  are uncertain nonlinear functions.

Since the system dynamic functions may be unknown or perturbed by external disturbance in practical application, the ideal virtual controllers  $\mathbf{x}_{(i+1)d}$ ,  $i=1, 2, \dots, n-1$  and the ideal actual control law  $\mathbf{u}$  cannot be precisely obtained. Therefore, the stability of the controlled system cannot be guaranteed. The dynamic neural network- ORWNNs are adopted to estimate the ideal virtual controllers and actual controller and to ensure the stability of the controlled system despite the existence of the uncertain system dynamic. The direct adaptive backstepping control scheme employs ORWNNs to

approximate the virtual controllers in each step. Finally, the actual controller is obtained by ORWNN using backstepping approach again. Thus, the ORWNN based adaptive backstepping control laws are designed as

$$\mathbf{x}_{(i+1)d} = \hat{\mathbf{x}}_{(i+1)d(\text{ORWNN})} + \mathbf{u}_{ri}, \quad i=1, \dots, n-1 \quad (15)$$

$$\mathbf{u} = \hat{\mathbf{u}}_{\text{ORWNN}} + \mathbf{u}_m \quad (16)$$

where  $\hat{\mathbf{x}}_{(i+1)d(\text{ORWNN})}$ ,  $i=1, \dots, n-1$  and  $\hat{\mathbf{u}}_{\text{ORWNN}}$  are virtual and actual controllers generated by ORWNNs, respectively. They are used to learn the ideal controllers. Robust controllers,  $\mathbf{u}_{ri}$ ,  $i=1, \dots, n$  are designed to compensate the approximated errors of ORWNN controllers. Based on the universal approximation theorem [5], there exists an optimal approximation  $\mathbf{x}_{(i+1)d(\text{ORWNN})}^*$ ,  $i=1, \dots, n-1$  and  $\mathbf{u}_{(\text{ORWNN})}^*$  of ORWNN such that

$$\mathbf{x}_{(i+1)d}^* = \mathbf{x}_{(i+1)d(\text{ORWNN})}^* + \boldsymbol{\varepsilon}_i$$

$$= \mathbf{w}^{*T} \boldsymbol{\psi}^*(\mathbf{x}_r, \mathbf{m}^*, \boldsymbol{\sigma}^*, \boldsymbol{\theta}_r^*) + \boldsymbol{\varepsilon}_i, \quad i = 1, 2, \dots, n \quad (17)$$

$$\mathbf{u}^* = \mathbf{u}_{(\text{ORWNN})}^* + \boldsymbol{\varepsilon}_n = \mathbf{w}^{*T} \boldsymbol{\psi}^*(\mathbf{x}_r, \mathbf{m}^*, \boldsymbol{\sigma}^*, \boldsymbol{\theta}_r^*) + \boldsymbol{\varepsilon}_n \quad (18)$$

where  $\boldsymbol{\varepsilon}_i$ ,  $i=1, 2, \dots, n$  denote approximation error vector;  $\mathbf{w}^*$ ,  $\mathbf{m}^*$ ,  $\boldsymbol{\sigma}^*$ ,  $\boldsymbol{\theta}_r^*$ , and  $\boldsymbol{\psi}^*$  are optimal parameters of  $\mathbf{w}$ ,  $\mathbf{m}$ ,  $\boldsymbol{\sigma}$ ,  $\boldsymbol{\theta}_r$ , and  $\boldsymbol{\psi}$ , respectively. From (13), the ORWNNs' output can be represented as

$$\hat{\mathbf{x}}_{(i+1)d(\text{ORWNN})} = \hat{\mathbf{w}}^T \hat{\boldsymbol{\psi}} = \hat{\mathbf{w}}^T \hat{\boldsymbol{\psi}}(\hat{\mathbf{m}}, \hat{\boldsymbol{\sigma}}, \hat{\boldsymbol{\theta}}_r, \bar{\mathbf{x}}_n), \quad i=1, \dots, n-1 \quad (19)$$

$$\hat{\mathbf{u}}_{\text{ORWNN}} = \hat{\mathbf{w}}^T \hat{\boldsymbol{\psi}}(\hat{\mathbf{m}}, \hat{\boldsymbol{\sigma}}, \hat{\boldsymbol{\theta}}_r, \bar{\mathbf{x}}_n). \quad (20)$$

The estimated error  $\tilde{\mathbf{x}}_{(i+1)d}$ ,  $i=1, 2, \dots, n-1$  and  $\tilde{\mathbf{u}}$  satisfy

$$\tilde{\mathbf{x}}_{(i+1)d} = \mathbf{x}_{(i+1)d} - \hat{\mathbf{x}}_{(i+1)d}$$

$$= \mathbf{w}_i^{*T} \boldsymbol{\psi}_i^* + \boldsymbol{\varepsilon}_i - (\hat{\mathbf{x}}_{(i+1)d(\text{ORWNN})} + \mathbf{u}_{ri}) \quad (21)$$

$$= \mathbf{w}_i^{*T} \tilde{\boldsymbol{\psi}}_i + \tilde{\mathbf{w}}_i^T \hat{\boldsymbol{\psi}}_i + \boldsymbol{\varepsilon}_i - \mathbf{u}_{ri}$$

and

$$\tilde{\mathbf{u}} = \mathbf{u} - \hat{\mathbf{u}}$$

$$= \mathbf{w}_n^{*T} \boldsymbol{\psi}_n^* + \boldsymbol{\varepsilon}_n - (\hat{\mathbf{u}}_{\text{ORWNN}} + \mathbf{u}_m) \quad (22)$$

$$= \mathbf{w}_n^{*T} \tilde{\boldsymbol{\psi}}_n + \tilde{\mathbf{w}}_n^T \hat{\boldsymbol{\psi}}_n + \boldsymbol{\varepsilon}_n - \mathbf{u}_m$$

where  $\tilde{\mathbf{w}} = \mathbf{w}^* - \hat{\mathbf{w}}$  and  $\tilde{\boldsymbol{\psi}} = \boldsymbol{\psi}^* - \hat{\boldsymbol{\psi}}$ . The linearization technique is employed to have the following Taylor expansion of  $\tilde{\boldsymbol{\psi}}$

$$\tilde{\boldsymbol{\psi}} = \begin{bmatrix} \tilde{\psi}_1 \\ \tilde{\psi}_2 \\ \vdots \\ \tilde{\psi}_{m_b} \end{bmatrix} = \begin{bmatrix} \frac{\partial \tilde{\psi}_1}{\partial \mathbf{m}} \\ \frac{\partial \tilde{\psi}_2}{\partial \mathbf{m}} \\ \vdots \\ \frac{\partial \tilde{\psi}_{m_b}}{\partial \mathbf{m}} \end{bmatrix}_{\mathbf{m}=\hat{\mathbf{m}}} (\mathbf{m}^* - \hat{\mathbf{m}}) + \begin{bmatrix} \frac{\partial \tilde{\psi}_1}{\partial \boldsymbol{\sigma}} \\ \frac{\partial \tilde{\psi}_2}{\partial \boldsymbol{\sigma}} \\ \vdots \\ \frac{\partial \tilde{\psi}_{m_b}}{\partial \boldsymbol{\sigma}} \end{bmatrix}_{\boldsymbol{\sigma}=\hat{\boldsymbol{\sigma}}} (\boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}}) \quad (23)$$

$$+ \begin{bmatrix} \frac{\partial \tilde{\psi}_1}{\partial \boldsymbol{\theta}_r} \\ \frac{\partial \tilde{\psi}_2}{\partial \boldsymbol{\theta}_r} \\ \vdots \\ \frac{\partial \tilde{\psi}_{m_b}}{\partial \boldsymbol{\theta}_r} \end{bmatrix}_{\boldsymbol{\theta}_r=\hat{\boldsymbol{\theta}}_r} (\boldsymbol{\theta}_r^* - \hat{\boldsymbol{\theta}}_r) + \mathbf{O}_r.$$

Rewrite (23), it can be represented as

$$\tilde{\boldsymbol{\psi}} = \boldsymbol{\Psi}_m^T \tilde{\mathbf{m}} + \boldsymbol{\Psi}_\sigma^T \tilde{\boldsymbol{\sigma}} + \boldsymbol{\Psi}_{\theta_r}^T \tilde{\boldsymbol{\theta}}_r + \mathbf{O}_r \quad (24)$$

where  $\Psi_m = [\frac{\partial \tilde{\psi}_1}{\partial \mathbf{m}} \frac{\partial \tilde{\psi}_2}{\partial \mathbf{m}} \dots \frac{\partial \tilde{\psi}_{m_b}}{\partial \mathbf{m}}]$ ,  $\Psi_\sigma = [\frac{\partial \tilde{\psi}_1}{\partial \boldsymbol{\sigma}} \frac{\partial \tilde{\psi}_2}{\partial \boldsymbol{\sigma}} \dots \frac{\partial \tilde{\psi}_{m_b}}{\partial \boldsymbol{\sigma}}]$   
 $\Psi_{\theta_r} = [\frac{\partial \tilde{\psi}_1}{\partial \boldsymbol{\theta}_r} \frac{\partial \tilde{\psi}_2}{\partial \boldsymbol{\theta}_r} \dots \frac{\partial \tilde{\psi}_{m_b}}{\partial \boldsymbol{\theta}_r}]$ ,  $\mathbf{O}_r$  is the high-order term.

Substituting (26) into (23) and (22), then we have the estimated error of  $\tilde{\mathbf{x}}_{(i+1)d}$ ,  $i=1,2, \dots, n-1$ , and  $\tilde{\mathbf{u}}$  are

$$\begin{aligned} \tilde{\mathbf{x}}_{(i+1)d} &= \mathbf{x}_{(i+1)d} - \hat{\mathbf{x}}_{(i+1)d} \\ &= (\mathbf{w}_i^{*T} \Psi_i^{*T} + \boldsymbol{\varepsilon}_i) - (\hat{\mathbf{x}}_{(i+1)d}(\text{ORWNN}) + \mathbf{u}_{ri}) \\ &= \tilde{\mathbf{w}}_i^T (\hat{\Psi}_i - \Psi_{mi}^T \hat{\mathbf{m}}^{(i)} - \Psi_{\sigma i}^T \hat{\boldsymbol{\sigma}}^{(i)} - \Psi_{\theta_i}^T \hat{\boldsymbol{\theta}}_r^{(i)}) + \hat{\mathbf{w}}_i^T (\Psi_{mi}^T \tilde{\mathbf{m}}^{(i)} \\ &\quad + \Psi_{\sigma i}^T \tilde{\boldsymbol{\sigma}}^{(i)} + \Psi_{\theta_i}^T \tilde{\boldsymbol{\theta}}_r^{(i)}) + \mathbf{D}_i - \mathbf{u}_{ri} \end{aligned} \quad i=1,2, \dots, n-1, \quad (25)$$

$$\begin{aligned} \tilde{\mathbf{u}} &= \mathbf{u} - \hat{\mathbf{u}} \\ &= (\mathbf{w}_n^{*T} \Psi_n^{*T} + \boldsymbol{\varepsilon}_n) - (\hat{\mathbf{u}}_{\text{ORWNN}} + \mathbf{u}_m) \\ &= \tilde{\mathbf{w}}_n^T (\hat{\Psi}_n - \Psi_{mn}^T \hat{\mathbf{m}}^{(n)} - \Psi_{\sigma n}^T \hat{\boldsymbol{\sigma}}^{(n)} - \Psi_{\theta_n}^T \hat{\boldsymbol{\theta}}_r^{(n)}) \\ &\quad + \hat{\mathbf{w}}_n^T (\Psi_{mn}^T \tilde{\mathbf{m}}^{(n)} + \Psi_{\sigma n}^T \tilde{\boldsymbol{\sigma}}^{(n)} + \Psi_{\theta_n}^T \tilde{\boldsymbol{\theta}}_r^{(n)}) + \mathbf{D}_n - \mathbf{u}_m \end{aligned} \quad (26)$$

where  $\mathbf{D}_i$ ,  $i=1, \dots, n$ , are the uncertain terms and assumed to be bounded by  $\|\mathbf{D}_i\| \leq \delta_i$ ,  $i=1,2, \dots, n$ ;  $\delta_i$  are unknown finite positive constant. Usually,  $\delta_i$  cannot be obtained in practical applications. Herein, however, an adaptive scheme is used to estimate it. Therefore, the following theorem can be obtained.

**Theorem 1:** Consider the MIMO nonlinear uncertain non-affine system (14) satisfying Assumptions 1, 2, and 3. The adaptive laws of the ORWNN backstepping controllers are designed as

$$\dot{\hat{\mathbf{w}}}_i = -\gamma_{wi} (\hat{\Psi}_i - \Psi_{mi}^T \hat{\mathbf{m}}^{(i)} - \Psi_{\sigma i}^T \hat{\boldsymbol{\sigma}}^{(i)} - \Psi_{\theta_i}^T \hat{\boldsymbol{\theta}}_r^{(i)}) \mathbf{e}_i, \quad i=1, \dots, n \quad (27)$$

$$\dot{\hat{\mathbf{m}}}^{(i)} = -\gamma_{mi} \Psi_{mi} \hat{\mathbf{w}}_i \mathbf{e}_i, \quad (28)$$

$$\dot{\hat{\boldsymbol{\sigma}}}^{(i)} = -\gamma_{\sigma i} \Psi_{\sigma i} \hat{\mathbf{w}}_i \mathbf{e}_i, \quad (29)$$

$$\dot{\hat{\boldsymbol{\theta}}}_r^{(i)} = -\gamma_{\theta_i} \Psi_{\theta_i}^{(i)} \hat{\mathbf{w}}_i \mathbf{e}_i. \quad (30)$$

where  $\gamma_{wi}$ ,  $\gamma_{mi}$ ,  $\gamma_{\sigma i}$ ,  $\gamma_{\theta_i}$ , and  $\gamma_{\gamma_i}$ ,  $i=1, \dots, n$ , are positive adaptive parameters, and the robust controllers with an adaptive bounded estimator are designed as follows

$$\mathbf{u}_{ri} = -[\mathbf{e}_i^T]^+ \|\mathbf{e}_i\| \hat{\delta}_i, \quad (31)$$

$$\dot{\hat{\delta}}_i = \gamma_{\delta_i} \|\mathbf{e}_i\|, \quad i=1, \dots, n, \quad (32)$$

where  $[\cdot]^+$  denotes the pseudo inverses and  $\hat{\delta}_i$  is an on-line estimated value of the uncertain term bounded. Hence, the asymptotically convergence of tracking error and the system stability can be guaranteed.

*Proof:* As the above discussion.

The control scheme using Theorem 1 is called DABC<sub>ORWNN</sub>. Figure 2 shows the configuration of DABC<sub>ORWNN</sub>.

#### IV. SIMULATION RESULTS

In this section, the simulation results of a non-affine double-pendulums system (as shown in Fig. 3) are presented. This illustration example shows the performances of our proposed approaches. Consider the tracking control of two degree-of-freedom double pendulums [28]. As shown in Fig. 3, the two rods in the vertical plane and two connecting joints are derived by torque control. All frictional forces are

ignored here. The following equations of motion can be derived by

$$\begin{aligned} M_1(t) - M_2(t) &= \frac{1}{3} l_1^2 (m_1 + 3m_2) \ddot{\theta}_1 + \frac{1}{2} g l_1 (m_1 + 2m_2) \sin(\theta_1(t)) \\ &\quad + \frac{1}{2} l_1 l_2 m_2 \cos(\theta_2(t) - \theta_1(t)) \ddot{\theta}(t) \\ &\quad - \frac{1}{2} l_1 l_2 m_2 \dot{\theta}_2^2(t) \sin(\theta_2(t) - \theta_1(t)) \\ M_2(t) &= \frac{1}{2} l_1 l_2 m_2 \cos(\theta_2(t) - \theta_1(t)) \ddot{\theta}_1(t) + \frac{1}{3} m_2 l_2^2 \ddot{\theta}_2(t) \\ &\quad + \frac{1}{2} g l_2 m_2 \sin(\theta_2(t)) - \frac{1}{2} l_1 l_2 m_2 \dot{\theta}_1^2(t) \sin(\theta_2(t) - \theta_1(t)) \end{aligned} \quad (33)$$

where  $\theta_1(t)$  and  $\theta_2(t)$  are the generalized coordinates, and  $M_1(t)$ ,  $M_2(t)$  are the torques acting on the connecting joints of the rod 1 and rod 2. Then, equations in (33) can be rewritten with respect to  $\ddot{\theta}_1(t)$  and  $\ddot{\theta}_2(t)$

$$\begin{aligned} \ddot{\theta}_1(t) &= f_{11}(\theta_1, \theta_2) M_1(\theta_1(t), \theta_2(t), u_1(t)) \\ &\quad + f_{12} M_2(\theta_1(t), \theta_2(t), u_2(t)) + f_{13}(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) \\ \ddot{\theta}_2(t) &= f_{21}(\theta_1, \theta_2) M_1(\theta_1(t), \theta_2(t), u_1(t)) \\ &\quad + f_{22} M_2(\theta_1(t), \theta_2(t), u_2(t)) + f_{23}(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) \end{aligned} \quad (34)$$

where

$$\begin{aligned} f_{11}(\theta_1, \theta_2) &= \frac{12}{l_1^2 [4m_1 + 12m_2 - 9m_2 \cos^2(\theta_2 - \theta_1)]}, \\ f_{12}(\theta_1, \theta_2) &= \frac{12l_2 + 18l_1 \cos(\theta_2 - \theta_1)}{l_1^2 l_2 [9m_2 \cos^2(\theta_2 - \theta_1) - 4m_1 - 12m_2]}, \\ f_{21}(\theta_1, \theta_2) &= \frac{18 \cos(\theta_2 - \theta_1)}{l_1 l_2 [9m_2 \cos^2(\theta_2 - \theta_1) - 4m_1 - 12m_2]}, \\ f_{22}(\theta_1, \theta_2) &= \frac{12m_1 + 36m_2}{l_2^2 m_2 [4m_1 + 12m_2 - 9m_2 \cos^2(\theta_2 - \theta_1)]} \\ &\quad + \frac{18l_1 \cos(\theta_2 - \theta_1)}{l_1 l_2^2 [4m_1 + 12m_2 - 9m_2 \cos^2(\theta_2 - \theta_1)]}, \\ f_{13}(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2) &= \frac{9gm_2 \sin(2\theta_2 - \theta_1)}{l_1 [15m_2 + 8m_1 - 9m_2 \cos(2\theta_2 - 2\theta_1)]} \\ &\quad - \frac{9l_1 m_2 \dot{\theta}_1^2 \sin(2\theta_2 - 2\theta_1)}{l_1 [15m_2 + 8m_1 - 9m_2 \cos(2\theta_2 - 2\theta_1)]} \\ &\quad + \frac{12l_2 m_2 \dot{\theta}_2^2 \sin(\theta_2 - \theta_1)}{l_1 [15m_2 + 8m_1 - 9m_2 \cos(2\theta_2 - 2\theta_1)]} \\ &\quad - \frac{(15gm_2 + 12gm_1) \sin(\theta_1)}{l_1 [15m_2 + 8m_1 - 9m_2 \cos(2\theta_2 - 2\theta_1)]}, \\ f_{23}(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2) &= \frac{12l_1^2 \dot{\theta}_1^2 \sin(\theta_2 - \theta_1)(m_1 + 3m_2)}{l_1 [15m_2 + 8m_1 - 9m_2 \cos(2\theta_2 - 2\theta_1)]} \\ &\quad - \frac{9l_2 m_2 \dot{\theta}_2^2 \sin(2\theta_2 - 2\theta_1)}{l_1 [15m_2 + 8m_1 - 9m_2 \cos(2\theta_2 - 2\theta_1)]} \\ &\quad - \frac{9g \sin(\theta_2 - 2\theta_1)(m_1 + 2m_2)}{l_1 [15m_2 + 8m_1 - 9m_2 \cos(2\theta_2 - 2\theta_1)]} \\ &\quad + \frac{3g \sin(\theta_2)(m_1 + 6m_2)}{l_1 [15m_2 + 8m_1 - 9m_2 \cos(2\theta_2 - 2\theta_1)]}. \end{aligned}$$

The double pendulums system in (33) can be rewritten in state-space representation, where  $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4]^T$  is the vector of measurable states,  $\mathbf{u} = [u_1 \ u_2]^T$  is the vector control inputs, and

$$\begin{aligned} f_1(\mathbf{x}, \mathbf{u}) &= f_{11}(\mathbf{x}) M_1(\mathbf{x}, u_1) + f_{12}(\mathbf{x}) M_2(\mathbf{x}, u_2) + f_{13}(\mathbf{x}) \\ f_2(\mathbf{x}, \mathbf{u}) &= f_{21}(\mathbf{x}) M_1(\mathbf{x}, u_1) + f_{22}(\mathbf{x}) M_2(\mathbf{x}, u_2) + f_{23}(\mathbf{x}) \end{aligned} \quad (35)$$

where  $f_{ij}, i=1, 2, j=1, 2, 3$  are well defined for all  $\mathbf{x} \in \mathfrak{R}^4$ .

Using the results of [28], the dynamic equations of motion are

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} f_1(\mathbf{x}, \mathbf{u}) \\ f_2(\mathbf{x}, \mathbf{u}) \end{bmatrix}$$

$$\mathbf{y}(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}(t), \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (36)$$

where  $x_1 = \theta_1, x_2 = \dot{\theta}_1, x_3 = \theta_2, x_4 = \dot{\theta}_2$ ,  $\mathbf{x} = [x_1, x_2, x_3, x_4]^T$ ,  $\mathbf{u} = [u_1, u_2]^T$ . Obviously, system (36) is nonlinear non-affine, we cannot have the stabilizing controller by feedback linearization approach. Herein, the control objective is to use our adaptive backstepping controller such that the states follow a designed bounded reference trajectory  $x_{1d}, x_{3d}$

asymptotically. The initial conditions is  $\mathbf{x}_0 = [\frac{\pi}{18}, 0, -\frac{\pi}{18}, 0]^T$

and the design parameter is  $\mathbf{k}_1 = \begin{bmatrix} 25 & 0 \\ 0 & 25 \end{bmatrix}$ , the adaptive parameter rates of DABC<sub>ORWNN</sub>-2 are  $\gamma_\delta = 15$ ,  $\gamma_w = 10$ ,  $\gamma_m = \gamma_\sigma = 0.9, \gamma_{\theta_i} = 0.01$ . The network structure is selected as [2 -8 -4 -2] and the initial value was set that  $w_{ij}$  is 0,  $m_{ij}$  is [-1.5,-0.5,0.5,1.5],  $\sigma_{ij}$  is 1.0, and  $\theta_{\epsilon_i}$  is 0. The external disturbance is  $\mathbf{D} = \frac{1}{2}[\sin(2t) \cos(2t)]^T$ . The simulation

results of DABC<sub>ORWNN</sub>-2 are shown in Figs. 4 and 5. State trajectories are shown in Fig. 4 (solid line: actual outputs; dashed line: reference trajectories). Figure 5 shows the corresponding control forces and tracking errors. It can be found that DABC<sub>ORWNN</sub>-2 approach performs well and has fast stabilizing time (about 0.8 second).

### V.CONCLUSION

This paper has successfully presented the DABC<sub>ORWNN</sub> control scheme for a class of MIMO nonlinear uncertain non-affine systems in non-triangular form. In the DABC<sub>ORWNN</sub> control systems, ORWNNs were used to learn the ideal virtual controllers and actual controller. In addition, the robust controllers are designed to compensate the approximated errors of ORWNNs. According to the Lyapunov stability approach, the adaptive laws of online tuning parameters are obtained, and the stability of the control system is guaranteed. To verify the effectiveness of the proposed control scheme, numerical simulation of non-affine double-pendulums system in non-triangular form have been presented to illustrate the effectiveness and performances of our approach.

### REFERENCES

[1] B. Chen, X. Liu, and S. Tong, "Adaptive fuzzy output tracking control of MIMO nonlinear uncertain systems," *IEEE Trans. on Fuzzy System*, Vol. 15, No. 2, pp. 287-300, 2007.  
 [2] C. S. Chiu, "Robust adaptive control of uncertain MIMO nonlinear systems- feedforward Takagi-Sugeno fuzzy approximation based approach," *IEE Proc. Control Theory Appl.*, Vol. 152, No. 2, pp. 157-164, 2005.  
 [3] B. R. Chung, *Fuzzy Neural Network Based Adaptive Backstepping Controller Design for a Class of Nonlinear Systems*, Master Thesis of Yuan Ze University, Taiwan, 2007.  
 [4] E. V. Grinitis and C. P. Bottura, "Adaptive neural-based backstepping

control of uncertain MIMO nonlinear systems," *IEEE International Joint Conference on Neural Networks*, pp. 4468-4475, 2006.  
 [5] H. Han, C. Y. Su, and Y. Stepanenko, "Adaptive control of a class of nonlinear systems with nonlinearly parameterized fuzzy approximators," *IEEE Trans. on Fuzzy Systems*, Vol. 9, No. 2, pp. 315-323, 2001.  
 [6] C. F. Hsu, C. M. Lin, and T. T. Lee, "Wavelet adaptive backstepping control for a class of nonlinear systems," *IEEE Trans. on Neural Networks*, Vol. 17, No. 5, pp. 1-9, 2006.  
 [7] N. Hovakimyan, E. Lavretsky, and A. Sasane, "Dynamic inversion for nonaffine-in-control systems via time-scale separation part 1," *Journal of Dynamical and Control Systems*, Vol. 13, No. 4, pp. 451-465, 2007.  
 [8] Y. Q. Jin, H. Wu, and W. J. Gu, "Adaptive control for a class of nonaffine systems based on fuzzy-neural approach," *IMACS Multiconference on Computational Engineering in Systems Applications*, Vol. 1, pp. 519-523, 2006.  
 [9] M. Krstic, I. Kanllakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*, New York: Wiley, 1995.  
 [10] Y. Li, S. Qiang, X. Zhuang, and O. Kaynak, "Robust and adaptive backstepping control for nonlinear systems using RBF neural networks," *IEEE Trans. on Neural Networks*, Vol. 15, No. 3, pp. 693-701, 2004.  
 [11] C. H. Lee and C. C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural networks," *IEEE Trans. on Fuzzy Systems*, Vol. 8, No. 4, pp. 349-366, 2000.  
 [12] C. H. Lee and C. C. Teng, "Fine tuning of membership functions for fuzzy neural systems," *Asian Journal of Control*, Vol. 3, No. 3, pp. 216-225, 2001.  
 [13] C. J. Lin and C. C. Chin, "A wavelet-based neural-fuzzy system and its applications," in *Proc. IEEE Int. Joint Conf. Neural Networks*, pp. 1921-1926, 2003.  
 [14] C. M. Lin, L. Y. Chen, and C. H. Chen, "RCMAC hybrid control for MIMO uncertain nonlinear systems using sliding-mode technology," *IEEE Trans. on Neural Networks*, Vol. 18, No. 3, pp. 708-720, 2007.  
 [15] C. M. Lin and Y. J. Mon, "Decoupling control by hierarchical fuzzy sliding-mode controller," *IEEE Trans. on Fuzzy Systems*, Vol. 13, No. 4, pp. 593-598, 2005.  
 [16] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*, Prentice-Hall, NJ, 1996.  
 [17] F. J. Lin and C. H. Lin, "On-line gain-tuning IP controller using RFNN," *IEEE Trans. on Aerospace and Electronic Systems*, Vol. 37, No. 2, pp. 655-670, 2001.  
 [18] Y. J. Liu and W. Wang, "Adaptive fuzzy control for a class of uncertain nonaffine nonlinear system," *Information Sciences*, Vol. 177, Issue 18, pp. 3901-3917, 2007.  
 [19] S. Labiod and T. M. Guerra, "Adaptive fuzzy control of a class of SISO nonaffine nonlinear systems," *Fuzzy Sets and Systems*, Vol. 158, No. 10, pp. 1126-1137, 2006.  
 [20] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. on Neural Networks*, Vol. 1, No. 1, pp. 4-27, 1990.  
 [21] J. K. Park, S. H. Huh, and S. H. Kim, "Direct adaptive controller for nonaffine nonlinear systems using self-structuring neural networks," *IEEE Trans. on Neural Networks*, Vol. 16, No. 2, pp. 414-422, 2005.  
 [22] Y. F. Peng and C. M. Lin, "Intelligent hybrid control for uncertain nonlinear systems using a recurrent cerebellar model articulation controller," *IEE Proc. Control Theory Appl.*, Vol. 151, pp. 589-600, 2004.  
 [23] Y. F. Peng, M. H. Lin, and C. M. Chong, "Design of output recurrent CMAC backstepping control system for tracking periodic trajectories," *IEEE International Joint Conference on Neural Networks*, pp. 3108-3113, 2006.  
 [24] C. D. Sousa, J. E. M. Hemerly, and R. H. Calvao, "Adaptive control for mobile robot using wavelet networks," *IEEE Trans. on Systems, Man and Cybernetics, Part B: Cybernetics*, Vol. 32, No. 4, pp. 493-504, 2002.  
 [25] S. Tong and Y. Li, "Direct adaptive fuzzy backstepping control for nonlinear systems," *IEEE on Proceeding of the First International Conference on Innovative Computing, Information and Control*, Vol. 2, pp. 623-627, 2006.  
 [26] R. J. Wai and K. Y. Hsieh, "Tracking control design for robot manipulator via fuzzy neural network," *IEEE Trans. on Fuzzy Systems*, Vol. 2, pp. 1422-1427, 2002.  
 [27] L. X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*, Englewood Cliffs, Prentice-Hall, NJ, 1994.  
 [28] A. Young, C. Cao, N. Hovakimyan, and E. Lavretsky, "Control of a nonaffine double-pendulum system via dynamic inversion and

time-scale separation," *America Control Conference*, pp. 1820-1825, 2006.

- [29] A. Yesildirek and F. L. Lewis, "Feedback linearization using neural networks," *Automatica*, Vol. 31, No. 11, pp. 1659-1664, 1995.
- [30] W. Zhang and S. S.Ge, "A global implicit function theorem without initial point and its applications to control of non-affine systems of high dimensions," *Journal of Mathematical Analysis and Applications*, Vol. 313, No. 1, pp. 251-261, 2006.

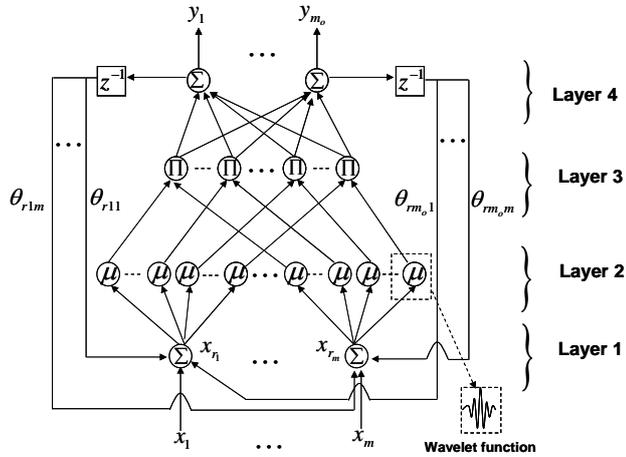


Figure 1: Schematic diagram of output recurrent wavelet neural network (ORWNN).

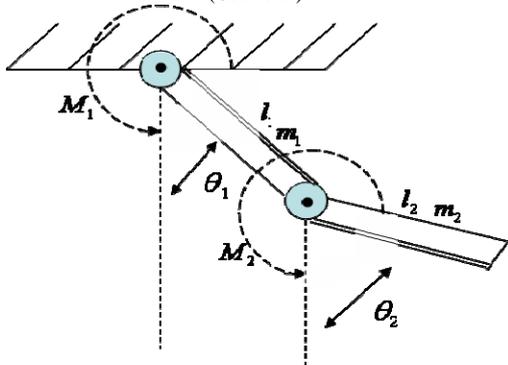


Figure 3: Double pendulums system [28].

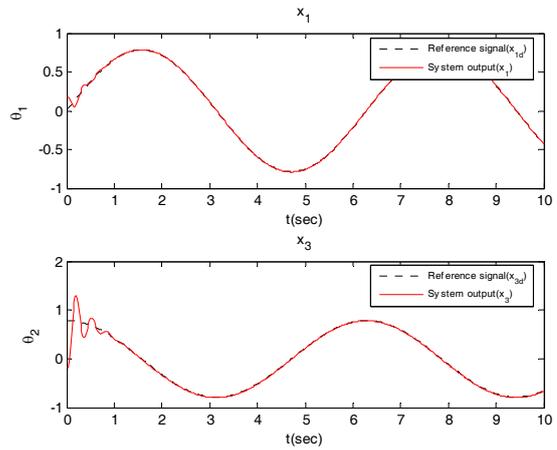


Figure 4: Simulation results - State trajectories (dashed line:  $(x_{1d}, x_{3d})$ ; solid line: DABC<sub>ORWNN-2</sub>).

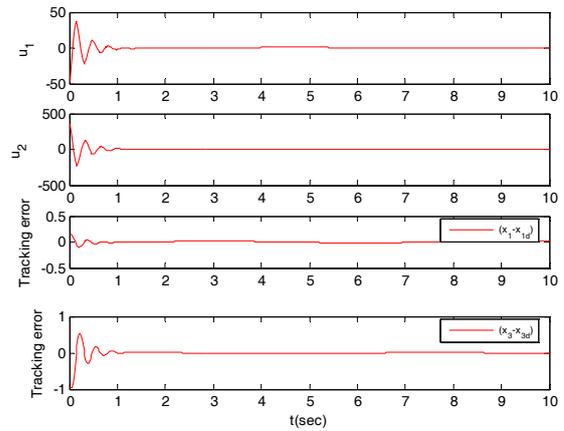


Figure 5: Simulation results- Control force and tracking errors.

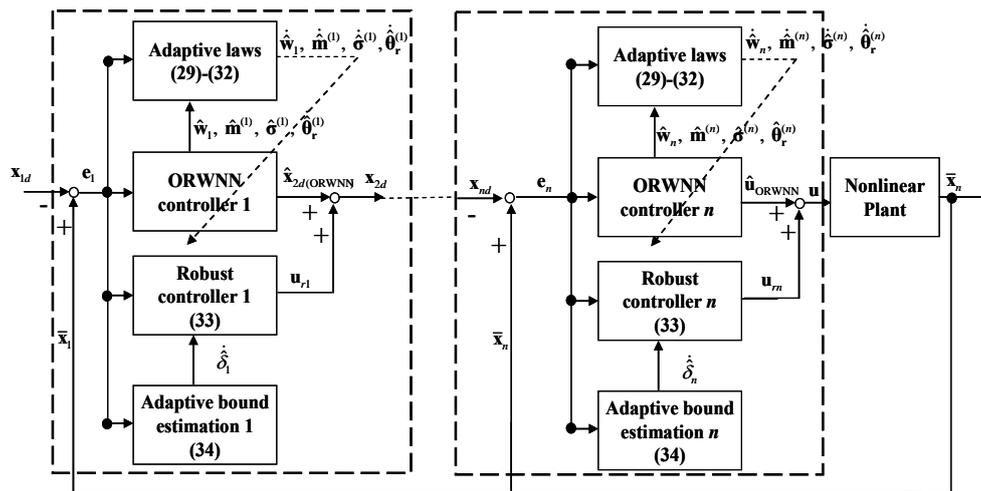


Figure 2: Direct adaptive ORWNNs control via backstepping control scheme (DABC<sub>ORWNNs</sub>).