

## Research Article

# An Output-Recurrent-Neural-Network-Based Iterative Learning Control for Unknown Nonlinear Dynamic Plants

**Ying-Chung Wang and Chiang-Ju Chien**

*Department of Electronic Engineering, Huaan University, Shihding, New Taipei City 223, Taiwan*

Correspondence should be addressed to Chiang-Ju Chien, cjc@cc.hfu.edu.tw

Received 31 July 2011; Revised 9 November 2011; Accepted 1 December 2011

Academic Editor: Isaac Chairez

Copyright © 2012 Y.-C. Wang and C.-J. Chien. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a design method for iterative learning control system by using an output recurrent neural network (ORNN). Two ORNNs are employed to design the learning control structure. The first ORNN, which is called the output recurrent neural controller (ORNC), is used as an iterative learning controller to achieve the learning control objective. To guarantee the convergence of learning error, some information of plant sensitivity is required to design a suitable adaptive law for the ORNC. Hence, a second ORNN, which is called the output recurrent neural identifier (ORNI), is used as an identifier to provide the required information. All the weights of ORNC and ORNI will be tuned during the control iteration and identification process, respectively, in order to achieve a desired learning performance. The adaptive laws for the weights of ORNC and ORNI and the analysis of learning performances are determined via a Lyapunov like analysis. It is shown that the identification error will asymptotically converge to zero and repetitive output tracking error will asymptotically converge to zero except the initial resetting error.

## 1. Introduction

Iterative learning control (ILC) system has become one of the most effective control strategies in dealing with repeated tracking control of nonlinear plants. The ILC system improves the control performance by a self-tuning process in the traditional PID-type ILC algorithms for linear plants or affine nonlinear plants with nonlinearities satisfying global Lipschitz continuous condition [1–3]. Recently, the ILC strategies combined with other control methodologies such as observer-based iterative learning control [4], adaptive iterative learning control [5], robust iterative learning control [6], or adaptive robust iterative learning control [7], have been widely studied in order to extend the applications to more general class of nonlinear systems. However, more and more restrictions are required in theory to develop these learning controllers. Among these ILC algorithms, PID-type ILC algorithms are still attractive to engineers since they are simple and effective for real implementations and industry applications. A main problem of the PID-type ILC algorithms is that a sufficient condition required to

guarantee learning stability and convergence will depend on plant's input/output coupling function (matrix). In general, it is hard to design the learning gain if the nonlinear dynamic plant is highly nonlinear and unknown. In order to get the input/output coupling function (matrix), the ILC using a neural or fuzzy system to solve the learning gain implementation problem can be found in [8, 9]. A neural network or a fuzzy system was used to approximate the inverse of plant's input/output coupling function (matrix). The inverse function (matrix) is claimed to be an optimal choice of the learning gain from a convergent condition point of view. As the nonlinear system is assumed to be unknown, some offline adaptive mechanisms are applied to update the network parameters in order to approximate the ideal optimal learning gain.

Actually, for control of unknown nonlinear systems, neural-network-based controller has become an important strategy in the past two decades. Multilayer neural networks, recurrent neural networks and dynamic neural network [10–16] were used for the design of adaptive controllers. On the other hand, fuzzy logic system, fuzzy neural network,

recurrent fuzzy neural networks and dynamic fuzzy neural network were also a popular tool for the design of adaptive controllers [17–22]. These concepts have also been applied to the design of adaptive iterative learning control of nonlinear plants [23–25]. However, few ILC works were developed for general unknown nonlinear plants, especially nonaffine nonlinear plants. As the authors can understand, a real-time recurrent network (RTRN) was developed in [26] for real-time learning control of general unknown nonlinear plants. But unfortunately, their learning algorithm depends on the generalized inverse of weight matrix in the RTRN. If the generalized inverse of weight matrix does not exist, the learning control scheme is not implementable.

In this paper, we consider the design of an iterative learning controller for a class of unknown nonlinear dynamic plants. Motivated by our previous work in [27], an improved version of an identifier-based iterative learning controller is proposed by using an output recurrent neural network (ORNN). Two ORNNs are used to design an ORNN-based iterative learning control system. The proposed ORNN-based ILC system includes an ORNN controller (ORNC) and an ORNN identifier (ORNI). The ORNC is used as an iterative learning controller to achieve the repetitive tracking control objective. The weights of ORNC are tuned via adaptive laws determined by a Lyapunov-like analysis. In order to realize the adaptive laws and guarantee the convergence of learning error, some information of the unknown plant sensitivity is required for the design of adaptive laws. Hence, the ORNI is then applied as an identifier to provide the required information from plant sensitivity. In a similar way, the weights of ORNI are tuned via some adaptive laws determined by a Lyapunov-like analysis. Both of the proposed ORNC and ORNI update their network weights along the control iteration and identification process, respectively. This ORNN-based ILC system can be used to execute a repetitive control task of a general nonlinear plant. It is shown that the identification error will asymptotically converge to zero and repetitive output tracking error will asymptotically converge to zero except the initial resetting error.

This paper is organized as follows. The structure of ORNN is introduced in Section 2. In Section 3, we present the design of ORNC and ORNI for the ORNN-based ILC system. The adaptation laws are derived and the learning performance is guaranteed based on a Lyapunov-like analysis. To illustrate the effectiveness of the proposed ILC system, a numerical example is used in Section 4 for computer simulation. Finally a conclusion is made in Section 5.

In the subsequent discussions, the following notations will be used in all the sections.

- (i)  $|z|$  denotes the absolute value of a function  $z$ .
- (ii)  $\|v\| = \sqrt{v^T v}$  denotes the usual Euclidean norm of a vector  $v = [v_1, \dots, v_n]^T \in \mathbb{R}^n$ .
- (iii)  $\|A\| = \max_{1 \leq i \leq n} \{\sum_{j=1}^m |a_{ij}|\}$  denotes the norm of a matrix  $A = \{a_{ij}\} \in \mathbb{R}^{n \times m}$ .

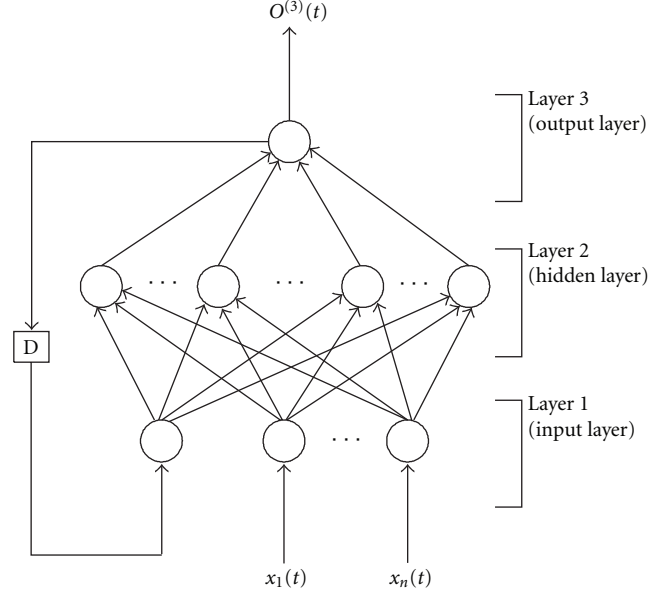


FIGURE 1: Structure of the ORNN.

## 2. The Output Recurrent Neural Network

In this paper, two ORNNs are used to design an iterative learning control system. The structure of the ORNN is shown in Figure 1, which comprises an input layer, a hidden layer, and an output layer.

- (i) Layer 1 (Input Layer): Each node in this layer represents an input variable, which only transmits input value to the next layer directly. For the  $i$ th input node,  $i = 1, \dots, n + 1$ ,

$$\begin{aligned} \text{net}_i^{(1)} &= \begin{cases} x_i, & i = 1, \dots, n \\ \mathbf{D}[O^{(3)}], & i = n + 1 \end{cases} \\ O_i^{(1)} &= f_i^{(1)}(\text{net}_i^{(1)}) = \text{net}_i^{(1)}, \end{aligned} \quad (1)$$

where  $x_i, i = 1, \dots, n$  represents the  $i$ th external input signal to the  $i$ th node of layer 1, and  $\mathbf{D}[O^{(3)}]$  denotes the delay of ORNN output  $O^{(3)}$  which can be further defined as  $x_{n+1} = \mathbf{D}[O^{(3)}]$ .

- (ii) Layer 2 (Hidden Layer): Each node in this layer performs an activation function whose inputs come from input layer. For the  $\ell$ th hidden node, a sigmoid function is adopted here such that the  $\ell$ th node,  $\ell = 1, \dots, M$  will be represented as

$$\begin{aligned} \text{net}_\ell^{(2)} &= \sum_{i=1}^{n+1} V_{i\ell} x_i^{(2)}, \\ O_\ell^{(2)} &= f_\ell^{(2)}(\text{net}_\ell^{(2)}) = \frac{1}{1 + \exp(-\text{net}_\ell^{(2)})}, \end{aligned} \quad (2)$$

where  $x_i^{(2)} = O_i^{(1)}$ ,  $V_{i\ell}$  is the connective weight between the input layer and the hidden layer,  $M$  is the number of neuron in the hidden layer.

(iii) Layer 3 (Output Layer): Each node in this layer represents an output node, which computes the overall output as the summation of all input signals from hidden layer. For the output node,

$$\begin{aligned} \text{net}^{(3)} &= \sum_{\ell=1}^M w_{\ell} \cdot x_{\ell}^{(3)}, \\ O^{(3)} &= f^{(3)}(\text{net}^{(3)}) = \text{net}^{(3)}, \end{aligned} \quad (3)$$

where  $x_{\ell}^{(3)} = O_{\ell}^{(2)}$  and  $w_{\ell}$  is the connective weight between the hidden layer and the output layer.

Let  $n$  denotes the dimension of input vector  $X = [x_1, \dots, x_n]^T \in \mathbb{R}^{n \times 1}$  of nonlinear function  $f(X)$  and  $M$  denotes the number of neurons in the hidden layer, the ORNN which performs as an approximator of the nonlinear function  $f(X)$  is now described in a matrix form as follows:

$$O^{(3)}(D[O^{(3)}], X, W, V) = W^T O^{(2)}(V^T X_a), \quad (4)$$

where  $W \in \mathbb{R}^{M \times 1}$  and  $V \in \mathbb{R}^{(n+1) \times M}$  are output-hidden wight matrix and hidden-input weight matrix, respectively,  $X \in \mathbb{R}^{n \times 1}$  is the external input vector,  $X_a \equiv [X^T, D[O^{(3)}]]^T \in \mathbb{R}^{(n+1) \times 1}$  is the augmented neural input vector, and  $D[O^{(3)}]$  denotes the delay of ORNN output  $O^{(3)}$ . The activation function vector is defined as  $O^{(2)}(V^T X_a) \equiv [O_1^{(2)}(V_1^T X_a), \dots, O_M^{(2)}(V_M^T X_a)]^T \in \mathbb{R}^{M \times 1}$  where  $V = [V_1, V_2, \dots, V_M]$  with  $V_{\ell} \in \mathbb{R}^{(n+1) \times 1}$  being the  $\ell$ th column vector, and  $O_{\ell}^{(2)}(V_{\ell}^T X_a) \equiv 1/(1 + \exp(-V_{\ell}^T X_a)) \in \mathbb{R}$ ,  $\ell = 1, \dots, M$  is a sigmoid function.

### 3. Design of Output-Recurrent-Neural- Network-Based Iterative Learning Control System

In this paper, we consider an unknown nonlinear dynamic plant which can perform a given task repeatedly over a finite time sequence  $t = \{0, \dots, N\}$  as follows:

$$y^j(t+1) = f(y^j(t), \dots, y^j(t-n+1), u^j(t)), \quad (5)$$

where  $j \in \mathbb{Z}_+$  denotes the index of control iteration number and  $t = \{0, \dots, N\}$  denotes the time index. The signals  $y^j(t)$  and  $u^j(t) \in \mathbb{R}$  are the system output and input, respectively.  $f: \mathbb{R}^{n+1} \rightarrow \mathbb{R}$  is the unknown continuous function,  $n$  represents the respective output delay order. Given a specified desired trajectory  $y_d(t)$ ,  $t \in \{0, \dots, N\}$ , the control objective is to design an output-recurrent-neural-network-based iterative learning control system such that when control iteration number  $j$  is large enough,  $|y_d(t) - y^j(t)|$  will converge to some small positive error tolerance bounds for all  $t \in \{0, \dots, N\}$  even if there exists an initial resetting error. Here the initial resetting error means that  $y_d(0) \neq y^j(0)$  for all  $j \geq 1$ . To achieve the control objective, an iterative learning control system based on ORNN design is proposed in Figure 2. In this figure,  $D$  denotes the delay in time domain and  $M$  denotes the memory in control iteration domain.

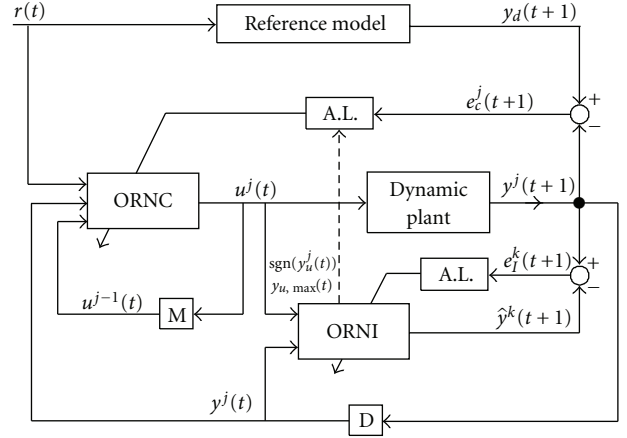


FIGURE 2: Block diagram of the ORNN-based ILC system.

Before we state the design steps of the proposed control structure, some assumptions on the unknown nonlinear system and desired trajectories are given as follows.

- (A1) The nonlinear dynamic plant is a relaxed system whose input  $u^j(t)$  and output  $y^j(t)$  are related by  $y^j(t) = 0$  for all  $t \in \{-\infty, \dots, -1\}$ .
- (A2) There exists a bounded unknown upper bounding function  $y_{u, \max}(t) = \max_{\forall j} |y_u^j(t)|$  such that  $0 < |y_u^j(t)| \leq y_{u, \max}(t)$ , where the factor  $y_u^j(t) = \partial y^j(t+1)/\partial u^j(t)$  represents the sensitivity of the plant with respect to its input.
- (A3) The reference model is designed to generate the bounded desired trajectory  $y_d(t+1) = f_d(y_d(t), \dots, y_d(t-n+1), r(t))$ , which is based on a specified bounded reference input  $r(t)$  with  $f_d: \mathbb{R}^{n+1} \rightarrow \mathbb{R}$  being a continuous function.

The design of the ORNN-based iterative learning control system is divided into two parts.

**3.1. Part 1: Design of ORNC and Corresponding Adaptive Laws.** Based on the assumptions on the nonlinear plant (5), we define a tracking error  $e_c^j(t)$  at  $j$ th control iteration as follows:

$$e_c^j(t) = y_d(t) - y^j(t). \quad (6)$$

It is noted that there exist bounded constants  $\epsilon_c^j$ ,  $j \in \mathbb{Z}_+$  such that the initial value of  $e_c^j(t)$  will satisfy

$$|e_c^j(0)| = \epsilon_c^j. \quad (7)$$

The difference of  $e_c^j(t)$  between two successive iterations can be computed as [28]

$$\begin{aligned}\Delta e_c^j(t+1) &= e_c^{j+1}(t+1) - e_c^j(t+1) \\ &= -(y^{j+1}(t+1) - y^j(t+1)) \\ &\approx -\frac{\partial y^j(t+1)}{\partial u^j(t)}(u^{j+1}(t) - u^j(t)) \\ &\equiv -y_u^j(t)(u^{j+1}(t) - u^j(t)).\end{aligned}\quad (8)$$

The ORNN is used to design an ORNC in order to achieve the iterative learning control objective. Let  $n_c$  be the dimension of the external input vector  $X_c^j(t) = [r(t), y^j(t), u^{j-1}(t), ]^\top \in \mathbb{R}^{n_c \times 1}$  and  $M_c$  denote the number of neurons in hidden layer of the ORNC. The ORNC which performs as an iterative learning controller is described in a matrix form as follows:

$$\begin{aligned}u^j(t) &= O_c^{(3)}(\mathbf{D}[O_c^{(3)j}(t)], X_c^j(t), W_c^j(t), V_c^j(t)) \\ &= W_c^j(t)^\top O_c^{(2)}(V_c^j(t)^\top X_{ca}^j(t)),\end{aligned}\quad (9)$$

where  $W_c^j(t) \in \mathbb{R}^{M_c \times 1}$  and  $V_c^j(t) \in \mathbb{R}^{(n_c+1) \times M_c}$  are output-hidden weight matrix and hidden-input weight matrix to be tuned via some suitable adaptive laws, respectively, and  $X_{ca}^j(t) \equiv [X_c^j(t)^\top, \mathbf{D}[O_c^{(3)j}(t)]]^\top \in \mathbb{R}^{(n_c+1) \times 1}$  is the augmented neural input vector. For the sake of convenience, we define  $O_c^{(2)}(V_c^j(t)^\top X_{ca}^j(t)) \equiv O_c^{(2)j}(t)$ . Now substituting (9) into (8), we will have

$$\Delta e_c^j(t+1) = -y_u^j(t)(W_c^{j+1}(t)^\top O_c^{(2)j+1}(t) - W_c^j(t)^\top O_c^{(2)j}(t)).\quad (10)$$

For simplicity, we define  $\Delta X_{ca}^j(t) = X_{ca}^{j+1}(t) - X_{ca}^j(t)$ ,  $\Delta W_c^j(t) = W_c^{j+1}(t) - W_c^j(t)$ ,  $\Delta V_c^j(t) = V_c^{j+1}(t) - V_c^j(t)$ . After adding and subtracting  $W_c^j(t)^\top O_c^{(2)j+1}(t)$  to (10), we can find that

$$\begin{aligned}\Delta e_c^j(t+1) &= -y_u^j(t)\Delta W_c^j(t)^\top O_c^{(2)j+1}(t) \\ &\quad - y_u^j(t)W_c^j(t)^\top (O_c^{(2)j+1}(t) - O_c^{(2)j}(t)).\end{aligned}\quad (11)$$

Investigating the second term in the right hand side of (11) by using the mean-value theorem, we have

$$\begin{aligned}O_c^{(2)j+1}(t) - O_c^{(2)j}(t) &= O_c^{(2)j}(t)(V_c^{j+1}(t)^\top X_{ca}^{j+1}(t) - V_c^j(t)^\top X_{ca}^j(t)) \\ &= O_c^{(2)j}(t)(V_c^{j+1}(t)^\top X_{ca}^{j+1}(t) - V_c^j(t)^\top X_{ca}^{j+1}(t) \\ &\quad + V_c^j(t)^\top X_{ca}^{j+1}(t) - V_c^j(t)^\top X_{ca}^j(t)) \\ &= O_c^{(2)j}(t)(\Delta V_c^j(t)^\top X_{ca}^{j+1}(t) + V_c^j(t)^\top \Delta X_{ca}^j(t)),\end{aligned}\quad (12)$$

where  $O_c^{(2)j}(t) = \text{diag}[O_{c,1}^{(2)j}(t), \dots, O_{c,M_c}^{(2)j}(t)] \in \mathbb{R}^{M_c \times M_c}$  with  $O_{c,\ell}^{(2)j}(t) \equiv dO_{c,\ell}^{(2)j}(Z_{c,\ell}(t))/dZ_{c,\ell}(t)|_{Z_{c,\ell}(t)}$ ,  $Z_{c,\ell}(t)$  has a value between  $V_{c,\ell}^{j+1}(t)^\top X_{ca}^{j+1}(t)$  and  $V_{c,\ell}^j(t)^\top X_{ca}^j(t)$ ,  $\ell = 1, \dots, M_c$ .

Now if we substitute (12) into (11), we will have

$$\begin{aligned}\Delta e_c^j(t+1) &= -y_u^j(t)\Delta W_c^j(t)^\top O_c^{(2)j+1}(t) \\ &\quad - y_u^j(t)W_c^j(t)^\top O_c^{(2)j}(t) \\ &\quad \times (\Delta V_c^j(t)^\top X_{ca}^{j+1}(t) + V_c^j(t)^\top \Delta X_{ca}^j(t)).\end{aligned}\quad (13)$$

The adaptation algorithms for weights  $W_c^{j+1}(t)$  and  $V_c^{j+1}(t)$  of ORNC at (next)  $j+1$ th control iteration to guarantee the error convergence are given as follows:

$$W_c^{j+1}(t) = W_c^j(t) + \frac{\text{sgn}(y_u^j(t))e_c^j(t+1)O_c^{(2)j+1}(t)}{y_{u,\max}(t)M_c},\quad (14)$$

$$V_c^{j+1}(t) = V_c^j(t) - \frac{X_{ca}^{j+1}(t)\Delta X_{ca}^j(t)^\top V_c^j(t)}{\|X_{ca}^{j+1}(t)\|^2},\quad (15)$$

where  $y_{u,\max}(t)$  is defined in assumption (A2). If we substitute adaptation laws (14) and (15) into (13), we can find that

$$\begin{aligned}e_c^{j+1}(t+1) &= e_c^j(t+1) - e_c^j(t+1) \\ &\quad \times \frac{|y_u^j(t)|O_c^{(2)j+1}(t)^\top O_c^{(2)j+1}(t)}{y_{u,\max}(t)M_c}.\end{aligned}\quad (16)$$

**Theorem 1.** Consider the nonlinear plant (5) which satisfies assumptions (A1)–(A3). The proposed ORNC (9) and adaptation laws (14) and (15) will ensure the asymptotic convergence of tracking error as control iteration approaches infinity.

*Proof.* Let us choose a discrete-type Lyapunov function as

$$E_c^j(t+1) = \frac{1}{2}(e_c^j(t+1))^2,\quad (17)$$

then the change of Lyapunov function is

$$\begin{aligned}\Delta E_c^j(t+1) &= E_c^{j+1}(t+1) - E_c^j(t+1) \\ &= \frac{1}{2}[(e_c^{j+1}(t+1))^2 - (e_c^j(t+1))^2].\end{aligned}\quad (18)$$

Taking norms on (16), it yields

$$\begin{aligned}|e_c^{j+1}(t+1)| &= \left| e_c^j(t+1) \left[ 1 - \frac{|y_u^j(t)|\|O_c^{(2)j+1}(t)\|^2}{y_{u,\max}(t)M_c} \right] \right| \\ &< |e_c^j(t+1)|\end{aligned}\quad (19)$$

for iteration  $j \geq 1$ . This further implies that  $E_c^j(t+1) > 0$ ,  $\Delta E_c^j(t+1) < 0$ , for all  $t \in \{0, \dots, N\}$  for  $j \geq 1$ . Using Lyapunov stability of  $E_c^j(t+1) > 0$ ,  $\Delta E_c^j(t+1) < 0$  and (7), the tracking error  $e_c^j(t)$  will satisfy

$$\lim_{j \rightarrow \infty} |e_c^j(t)| = \begin{cases} \epsilon_c^\infty, & t = 0 \\ 0, & t \neq 0. \end{cases}\quad (20)$$

This proves Theorem 1.  $\square$

*Remark 2.* If the plant sensitivity  $y_u^j(t)$  is completely known so that  $\text{sgn}(y_u^j(t))$  and  $y_{u,\max}(t)$  are available, then the control objective can be achieved by using the adaptation algorithms (14) and (15). However, the plant sensitivity  $y_u^j(t)$  is in general unknown or only partially known. In part 2, we will design an ORNN-based identifier (ORNI) to estimate the unknown plant sensitivity  $y_u^j(t)$  and then provide the sign function and upper bounding function of  $y_u^j(t)$  for adaptation algorithms of ORNC.

**3.2. Part 2: Design of ORNI and Corresponding Adaptive Laws.** After each control iteration, the ORNI subsequently begins to perform identification process. The trained ORNI will then provide the approximated plant sensitivity to the ORNC to start the next control iteration. We would like to emphasize that the ORNI only identifies the nonlinear plant after each control iteration. This concept is quite different from traditional control tasks [29] and very important to the proposed ORNN-based ILC structure.

The structure of ORNN is further applied to design an ORNI to identify the nonlinear plant after the  $j$ th control iteration. The identification process is stated as follows. After each trial of controlling the nonlinear system, we collect the input output data  $u^j(t)$  and  $y^j(t)$ ,  $t = 0, 1, \dots, N + 1$  as the training data for the identifier. When discussing the identification, we omit the control iteration index  $j$  and introduce a new identification iteration index  $k \in \mathbb{Z}_+$  to represent the number of identification process. That is, the notation for the training data  $u^j(t)$ ,  $y^j(t)$  and the ORNI output  $\hat{y}^{j,k}(t)$  are simplified as  $u(t)$ ,  $y(t)$ , and  $\hat{y}^k(t)$ , respectively. For the ORNI, let  $n_I$  be the dimension of external input vector  $X_I(t) = [u(t), y(t)]^\top \in \mathbb{R}^{n_I \times 1}$  and  $M_I$  denote the number of neurons in hidden layer of the ORNI. The ORNI which performs as an iterative learning identifier for nonlinear plant (5) is now described in a matrix form as follows:

$$\begin{aligned} \hat{y}^k(t+1) &= O_I^{(3)} \left( \mathbf{D} \left[ O_I^{(3)k}(t) \right], X_{Ia}^k(t), W_I^k(t), V_I^k(t) \right) \\ &= W_I^k(t)^\top O_I^{(2)} \left( V_I^k(t)^\top X_{Ia}^k(t) \right), \end{aligned} \quad (21)$$

where  $W_I^k(t) \in \mathbb{R}^{M_I \times 1}$  and  $V_I^k(t) \in \mathbb{R}^{(n_I+1) \times M_I}$  are output-hidden weight matrix and hidden-input weight matrix to be tuned via some suitable adaptive laws, respectively, and  $X_{Ia}^k(t) \equiv [X_I(t)^\top, \mathbf{D}[O_I^{(3)k}(t)]]^\top \in \mathbb{R}^{(n_I+1) \times 1}$  is the augmented neural input vector. For the sake of convenience, we define  $O_I^{(2)}(V_I^k(t)^\top X_{Ia}^k(t)) \equiv O_I^{(2)k}(t)$ .

Based on the assumptions on the nonlinear plant (5), we define an identification error  $e_I^k(t)$  at  $k$ th identification process as follows:

$$e_I^k(t) = y(t) - \hat{y}^k(t). \quad (22)$$

The difference of  $e_I^k(t)$  between two successive identification process can be computed as

$$\begin{aligned} \Delta e_I^k(t+1) &= e_I^{k+1}(t+1) - e_I^k(t+1) \\ &= -(\hat{y}^{k+1}(t+1) - \hat{y}^k(t+1)). \end{aligned} \quad (23)$$

Now substituting (21) into (23), we will have

$$\Delta e_I^k(t+1) = - \left( W_I^{k+1}(t)^\top O_I^{(2)k+1}(t) - W_I^k(t)^\top O_I^{(2)k}(t) \right). \quad (24)$$

For simplicity, we define  $\Delta X_{Ia}^k(t) = X_{Ia}^{k+1}(t) - X_{Ia}^k(t)$ ,  $\Delta W_I^k(t) = W_I^{k+1}(t) - W_I^k(t)$ ,  $\Delta V_I^k(t) = V_I^{k+1}(t) - V_I^k(t)$ . After adding and subtracting  $W_I^k(t)^\top O_I^{(2)k+1}(t)$  to (24), we can find

$$\begin{aligned} \Delta e_I^k(t+1) &= -\Delta W_I^k(t)^\top O_I^{(2)k+1}(t) \\ &\quad - W_I^k(t)^\top \left( O_I^{(2)k+1}(t) - O_I^{(2)k}(t) \right). \end{aligned} \quad (25)$$

Investigating the second term in the right hand side of (25) by using the mean-value theorem, we can derive

$$\begin{aligned} O_I^{(2)k+1}(t) - O_I^{(2)k}(t) &= O_I^{(2)k}(t) \left( V_I^{k+1}(t)^\top X_{Ia}^{k+1}(t) - V_I^k(t)^\top X_{Ia}^k(t) \right) \\ &= O_I^{(2)k}(t) \left( V_I^{k+1}(t)^\top X_{Ia}^{k+1}(t) - V_I^k(t)^\top X_{Ia}^{k+1}(t) \right. \\ &\quad \left. + V_I^k(t)^\top X_{Ia}^{k+1}(t) - V_I^k(t)^\top X_{Ia}^k(t) \right) \\ &= O_I^{(2)k}(t) \left( \Delta V_I^k(t)^\top X_{Ia}^{k+1}(t) + V_I^k(t)^\top \Delta X_{Ia}^k(t) \right), \end{aligned} \quad (26)$$

where  $O_I^{(2)k}(t) = \text{diag}[O_{I,1}^{(2)k}(t), \dots, O_{I,M_I}^{(2)k}(t)] \in \mathbb{R}^{M_I \times M_I}$  with  $O_{I,\ell}^{(2)k}(t) \equiv dO_{I,\ell}^{(2)k}(Z_{I,\ell}(t))/dZ_{I,\ell}(t)|_{Z_{I,\ell}(t)}$ ,  $Z_{I,\ell}(t)$  has a value between  $V_{I,\ell}^{k+1}(t)^\top X_{Ia}^{k+1}(t)$  and  $V_{I,\ell}^k(t)^\top X_{Ia}^k(t)$ ,  $\ell = 1, \dots, M_I$ .

Now if we substitute (26) into (25), we will have

$$\begin{aligned} \Delta e_I^k(t+1) &= -\Delta W_I^k(t)^\top O_I^{(2)k+1}(t) \\ &\quad - W_I^k(t)^\top O_I^{(2)k}(t) \\ &\quad \times \left( \Delta V_I^k(t)^\top X_{Ia}^{k+1}(t) + V_I^k(t)^\top \Delta X_{Ia}^k(t) \right). \end{aligned} \quad (27)$$

The adaptation algorithms for weights  $W_I^{k+1}(t)$  and  $V_I^{k+1}(t)$  of ORNI at (next)  $k+1$ th identification process are given as follows:

$$\begin{aligned} W_I^{k+1}(t) &= W_I^k(t) + \frac{e_I^k(t+1) O_I^{(2)k+1}(t)}{M_I} \\ V_I^{k+1}(t) &= V_I^k(t) - \frac{X_{Ia}^{k+1}(t) \Delta X_{Ia}^k(t)^\top V_I^k(t)}{\|X_{Ia}^{k+1}(t)\|^2}. \end{aligned} \quad (28)$$

If we substitute adaptation laws (28) into (27), we have

$$e_I^{k+1}(t+1) = e_I^k(t+1) - e_I^k(t+1) \frac{O_I^{(2)k+1}(t)^\top O_I^{(2)k+1}(t)}{M_I}. \quad (29)$$

**Theorem 3.** Consider the nonlinear dynamic plant (5) which satisfies assumptions (A1)–(A3). The proposed ORNI (21) and adaptation laws (28) will ensure that the asymptotic convergence of identification error is guaranteed as the numbers of identification approach infinity.



*Proof.* Let us choose a discrete-type Lyapunov function as

$$E_I^k(t+1) = \frac{1}{2} \left( e_I^k(t+1) \right)^2, \quad \forall t \in \{0, \dots, N\}, \quad (30)$$

then we can derive the change of Lyapunov function as

$$\begin{aligned} \Delta E_I^k(t+1) &= E_I^{k+1}(t+1) - E_I^k(t+1) \\ &= \frac{1}{2} \left[ \left( e_I^{k+1}(t+1) \right)^2 - \left( e_I^k(t+1) \right)^2 \right]. \end{aligned} \quad (31)$$

Taking norms on (29), we have

$$\begin{aligned} \left| e_I^{k+1}(t+1) \right| &= \left| e_I^k(t+1) \right| \left| 1 - \frac{\left\| O_I^{(2)k+1}(t) \right\|^2}{M_I} \right| \\ &< \left| e_I^k(t+1) \right| \end{aligned} \quad (32)$$

for iteration  $k \geq 1$ . This implies that  $E_I^k(t+1) > 0$ ,  $\Delta E_I^k(t+1) < 0$ , for all  $t \in \{0, \dots, N\}$  for  $k \geq 1$ , and hence the identification error  $e_I^k(t)$  will satisfy  $\lim_{k \rightarrow \infty} |e_I^k(t)| = 0$ , for all  $t \in \{0, 1, \dots, N\}$ . This proves Theorem 3.  $\square$

*Remark 4.* The ORNN is a promising tool for identification because it can approximate any “well-behaved” nonlinear function to any desired accuracy. This good function approximation is applied to estimate the unknown plant sensitivity in this paper. The plant sensitivity  $y_u^j(t)$  in (8) can be approximated as follows:

$$y_u^j(t) \equiv \frac{\partial y^j(t+1)}{\partial u^j(t)} \approx \frac{\partial \hat{y}^j(t+1)}{\partial u^j(t)}. \quad (33)$$

Note that the index  $k$  in the identifier output  $\hat{y}^k(t)$  is removed once the identification process stops. Applying the chain rule to (21), it yields

$$\begin{aligned} \frac{\partial \hat{y}^j(t+1)}{\partial u^j(t)} &= \frac{\partial O_I^{(3)j}(t)}{\partial u^j(t)} = \sum_{\ell=1}^{M_I} \frac{\partial \hat{y}^j(t+1)}{\partial O_{I,\ell}^{(2)j}(t)} \frac{\partial O_{I,\ell}^{(2)j}(t)}{\partial u^j(t)} \\ &= \sum_{\ell=1}^{M_I} w_{I,\ell}^j(t) \frac{\partial O_{I,\ell}^{(2)j}(t)}{\partial u^j(t)}. \end{aligned} \quad (34)$$

Also from (21), we have

$$\frac{\partial O_{I,\ell}^{(2)j}(t)}{\partial u^j(t)} = f_{I,\ell}^{(2)'} \left( \text{net}_{I,\ell}^{(2)j}(t) \right) \frac{\partial \text{net}_{I,\ell}^{(2)j}(t)}{\partial u^j(t)}. \quad (35)$$

Since the inputs to ORNI are  $u^j(t)$ ,  $y^j(t)$  and  $\mathbf{D}[O_I^{(3)j}(t)]$ , we further have

$$\begin{aligned} \text{net}_{I,\ell}^{(2)j}(t) &= V_{I,1\ell}^j(t) u^j(t) + V_{I,2\ell}^j(t) y^j(t) \\ &\quad + V_{I,3\ell}^j(t) \mathbf{D} \left[ O_I^{(3)j}(t) \right]. \end{aligned} \quad (36)$$

Thus,

$$\frac{\partial \text{net}_{I,\ell}^{(2)j}(t)}{\partial u^j(t)} = V_{I,1\ell}^j(t). \quad (37)$$

From (34), (35) and (37), we obtain

$$\hat{y}_u^j(t) = \frac{\partial \hat{y}^j(t+1)}{\partial u^j(t)} = \sum_{\ell=1}^{M_I} w_{I,\ell}^j(t) f_{I,\ell}^{(2)'} \left( \text{net}_{I,\ell}^{(2)j}(t) \right) V_{I,1\ell}^j(t), \quad (38)$$

where  $0 < f_{I,\ell}^{(2)'} \left( \text{net}_{I,\ell}^{(2)j}(t) \right) < 0.5$ . If we define  $\|w_I^j(t)\| \equiv \max_{\ell} |w_{I,\ell}^j(t)|$ , and  $\|V_{I,1}^j(t)\| \equiv \max_{\ell} |V_{I,1\ell}^j(t)|$ , then

$$\begin{aligned} \left| \hat{y}_u^j(t) \right| &\leq M_I \left\| w_I^j(t) \right\| \left\| f_{I,\ell}^{(2)'} \left( \text{net}_{I,\ell}^{(2)j}(t) \right) V_{I,1}^j(t) \right\| \\ &\leq \frac{M_I}{2} \left\| w_I^j(t) \right\| \left\| V_{I,1}^j(t) \right\| \equiv \hat{y}_{u,\max}^j(t). \end{aligned} \quad (39)$$

The sign function and upper bounding function of plant sensitivity after finishing the identification process at  $j$ th control iteration can be obtained as follows:

$$\begin{aligned} \text{sgn} \left( y_u^j(t) \right) &= \text{sgn} \left( \hat{y}_u^j(t) \right) \\ y_{u,\max}(t) &= \max \left\{ \hat{y}_{u,\max}^j(t), \hat{y}_{u,\max}^{j-1}(t) \right\}. \end{aligned} \quad (40)$$

It is noted that we do not need the exact plant sensitivity  $y_u^j(t)$  for the design of adaptive law (14). Even though there may exist certain approximation error between  $y_u^j(t)$  and  $\hat{y}_u^j(t)$ , we can still guarantee the convergence of learning error since only an upper bounding function is required. Also note that the value of  $\text{sgn}(y_u^j(t))$  (+1 or -1) can be easily determined from the identification result.

## 4. Simulation Example

In this section, we use the proposed ORNN-based ILC to iteratively control an unknown non-BIBO nonlinear dynamic plant [26, 29]. The difference equation of the nonlinear dynamic plant is given as

$$\begin{aligned} y^j(t+1) &= 0.2 \left( y^j(t) \right)^2 + 0.2 y^j(t-1) \\ &\quad + 0.4 \sin \left( 0.5 \left( y^j(t-1) + y^j(t) \right) \right) \\ &\quad \times \cos \left( 0.5 \left( y^j(t-1) + y^j(t) \right) \right) \\ &\quad + 1.2 u^j(t), \end{aligned} \quad (41)$$

where  $y^j(t)$  is the system output,  $u^j(t)$  is the control input. The reference model is chosen as

$$y_d(t+1) = 0.6 y_d(t) + r(t), \quad y_d(0) = 0, \quad (42)$$

where  $r(t) = \sin(2\pi t/25) + \sin(2\pi t/10)$  is a bounded reference input. The control objective is to force  $y^j(t)$  to track the desired trajectory  $y_d(t)$  as close as possible over a finite time interval  $t \in \{1, \dots, 200\}$  except the initial point. The network weight adaptation for the ORNI and ORNC is designed according to (14), (15), and (28), respectively. In the ORNC, we set  $W_c^j(t) \in \mathbb{R}^{2 \times 1}$  and  $V_c^j(t) \in \mathbb{R}^{4 \times 2}$ , that is, only two hidden nodes in layer 2 are used to construct the ORNC. In a similar way, we let  $W_I^k(t) \in \mathbb{R}^{2 \times 1}$

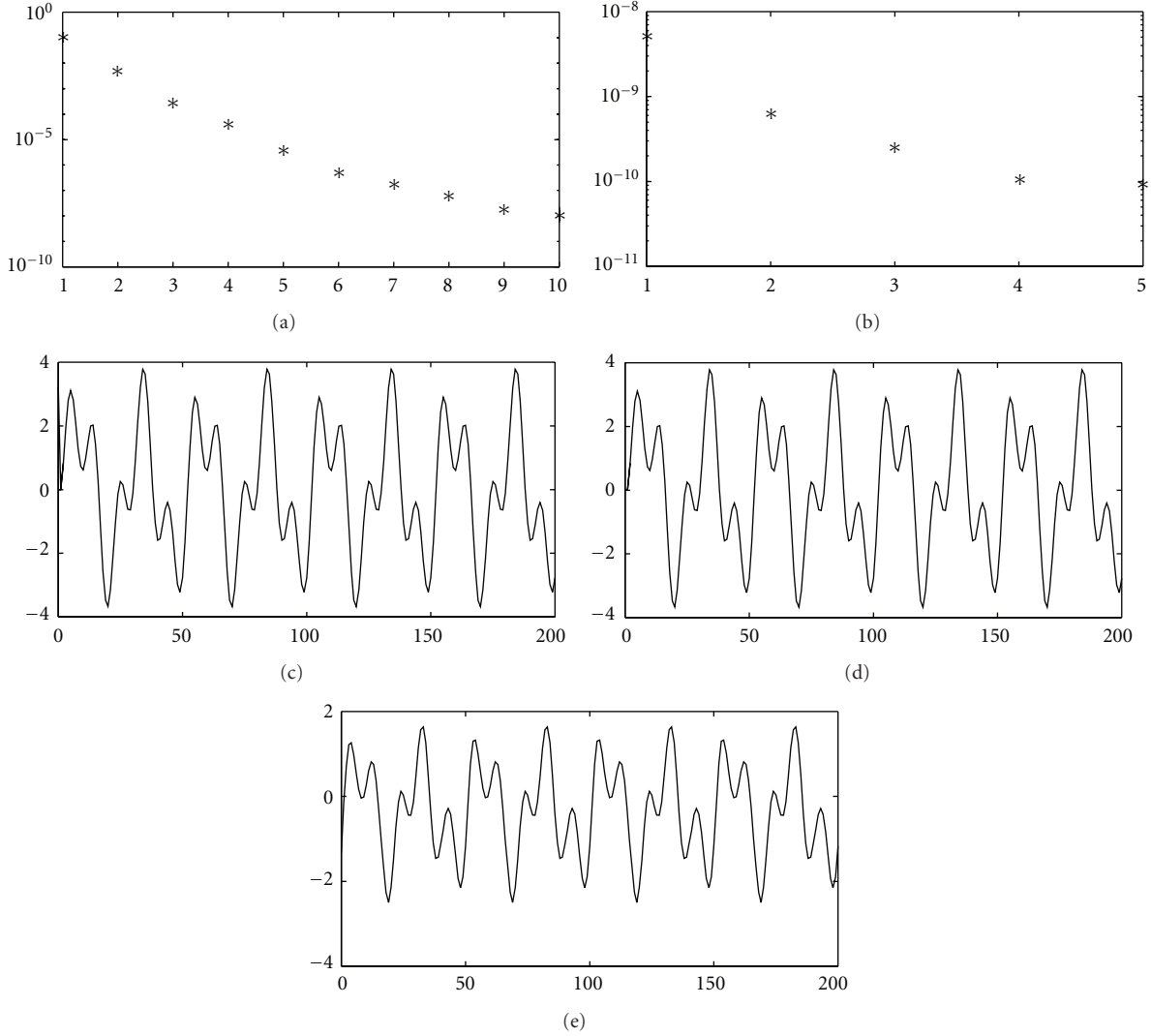


FIGURE 3: (a)  $\max_{t \in \{1, \dots, 200\}} |e_c^j(t)|$  versus control iteration  $j$ . (b)  $\max_{t \in \{0, \dots, 200\}} |e_t^{10,k}(t)|$  versus identification process  $k$  at the 10th control iteration. (c)  $y^{10}(t)$  (dotted line) and  $y_d(t)$  (solid line) versus time  $t$  at the 10th control iteration. (d)  $\hat{y}^{10}(t)$  (dotted line) and  $y^{10}(t)$  (solid line) versus time  $t$  at the 10th control iteration. (e)  $u^{10}(t)$  at the 10th control iteration versus time  $t$ .

and  $V_I^k(t) \in \mathbb{R}^{3 \times 2}$ , that is, only two hidden nodes in layer 2 are used to set up the ORNI. For simplicity, all the initial conditions of ORNC parameters are set to be 0 at the first control iteration. In addition, the initial ORNI parameters are set to be 0 at the first identification process which begins after the first control iteration. We assume that the plant initial condition satisfies  $y^j(0) = 2 + \text{randn}$  where  $\text{randn}$  is a generator of random number with normal distribution, mean = 0 and variance = 1. To study the effects of learning performances, we first show the maximum value of tracking error  $|e_c^j(t)|$ ,  $t \in \{1, \dots, 200\}$  with respect to control iteration  $j$  in Figure 3(a). It is noted that  $|e_c^j(0)|$  is omitted in calculating the maximum value of tracking error since it is not controllable. The identification error at 10th control iteration  $|e_t^{10,k}(t)|$  with respect to identification process  $k$  is shown in Figure 3(b). According to the simulation results, it is clear that the asymptotic

convergence proved in Theorems 1 and 3 is achieved. Since a reasonable tracking performance is almost observed at the 10th control iteration, the trajectories between the desired output  $y_d(t)$  and plant output  $y^{10}(t)$  at the 10th control iteration are shown to demonstrate the control performance in Figure 3(c). Figure 3(d) shows the comparison between the identification result of  $\hat{y}^{10}(t)$  and the plant output  $y^{10}(t)$ . The nice identification result enables the ORNI to provide the required information for the design of ORNC. Finally, the bounded control input  $u^{10}(t)$  is plotted in Figure 3(e).

## 5. Conclusion

For controlling a repeatable nonaffine nonlinear dynamic plant, we propose an output-recurrent-neural-network-based iterative learning control system in this paper. The control structure consists of an ORNC used as an iterative learning controller and an ORNI used as an identifier.

The ORNC is the main controller utilized to achieve the repetitive control task. The ORNI is an auxiliary component utilized to provide some useful information from plant sensitivity for the design of ORNC's adaptive laws. All the network weights of ORNC and ORNI will be tuned during the control iteration and identification process so that no prior plant knowledge is required. The adaptive laws for the weights of ORNC and ORNI and the analysis of learning performances are determined via a Lyapunov-like analysis. We show that if the ORNI can provide the knowledge of plant sensitivity for ORNC, then output tracking error will asymptotically converge to zero except an initial resetting error. We also show that the objective of identification can be achieved by the ORNI if the number of identifications is large enough.

## Acknowledgment

This work is supported by National Science Council, R.O.C., under Grant NSC99-2221-E-211-011-MY2.

## References

- [1] K. L. Moore and J. X. Xu, "Special issue on iterative learning control," *International Journal of Control*, vol. 73, no. 10, pp. 819–823, 2000.
- [2] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.
- [3] H. S. Ahn, Y. Q. Chen, and K. L. Moore, "Iterative learning control: brief survey and categorization," *IEEE Transactions on Systems, Man and Cybernetics. Part C*, vol. 37, no. 6, pp. 1099–1121, 2007.
- [4] J. X. Xu and J. Xu, "Observer based learning control for a class of nonlinear systems with time-varying parametric uncertainties," *IEEE Transactions on Automatic Control*, vol. 49, no. 2, pp. 275–281, 2004.
- [5] I. Rotariu, M. Steinbuch, and R. Ellenbroek, "Adaptive iterative learning control for high precision motion systems," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 5, pp. 1075–1082, 2008.
- [6] A. Tayebi, S. Abdul, M. B. Zaremba, and Y. Ye, "Robust iterative learning control design: application to a robot manipulator," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 5, pp. 608–613, 2008.
- [7] J. X. Xu and B. Viswanathan, "Adaptive robust iterative learning control with dead zone scheme," *Automatica*, vol. 36, no. 1, pp. 91–99, 2000.
- [8] J. Y. Choi and H. J. Park, "Neural-based iterative learning control for unknown systems," in *Proceedings of the 2nd Asian Control Conference*, pp. II243–II246, Seoul, Korea, 1997.
- [9] C. J. Chien, "A sampled-data iterative learning control using fuzzy network design," *International Journal of Control*, vol. 73, no. 10, pp. 902–913, 2000.
- [10] J. H. Park, S. H. Huh, S. H. Kim, S. J. Seo, and G. T. Park, "Direct adaptive controller for nonaffine nonlinear systems using self-structuring neural networks," *IEEE Transactions on Neural Networks*, vol. 16, no. 2, pp. 414–422, 2005.
- [11] J. S. Wang and Y. P. Chen, "A fully automated recurrent neural network for unknown dynamic system identification and control," *IEEE Transactions on Circuits and Systems I*, vol. 53, no. 6, pp. 1363–1372, 2006.
- [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, pp. 1–9, 2006.
- [13] C. M. Lin, L. Y. Chen, and C. H. Chen, "RCMAC hybrid control for MIMO uncertain nonlinear systems using sliding-mode technology," *IEEE Transactions on Neural Networks*, vol. 18, no. 3, pp. 708–720, 2007.
- [14] Z. G. Hou, M. M. Gupta, P. N. Nikiforuk, M. Tan, and L. Cheng, "A recurrent neural network for hierarchical control of interconnected dynamic systems," *IEEE Transactions on Neural Networks*, vol. 18, no. 2, pp. 466–481, 2007.
- [15] Z. Liu, R. E. Torres, N. Patel, and Q. Wang, "Further development of input-to-state stabilizing control for dynamic neural network systems," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 38, no. 6, pp. 1425–1433, 2008.
- [16] Z. Liu, S. C. Shih, and Q. Wang, "Global robust stabilizing control for a dynamic neural network system," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 39, no. 2, pp. 426–436, 2009.
- [17] Y. G. Leu, W. Y. Wang, and T. T. Lee, "Observer-based direct adaptive fuzzy-neural control for nonaffine nonlinear systems," *IEEE Transactions on Neural Networks*, vol. 16, no. 4, pp. 853–861, 2005.
- [18] 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, 2007.
- [19] B. Chen, X. Liu, and S. Tong, "Adaptive fuzzy output tracking control of MIMO nonlinear uncertain systems," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 2, pp. 287–300, 2007.
- [20] Y. J. Liu and W. Wang, "Adaptive fuzzy control for a class of uncertain nonaffine nonlinear system," *Information Sciences*, vol. 177, no. 18, pp. 3901–3917, 2007.
- [21] R. J. Wai and C. M. Liu, "Design of dynamic petri recurrent fuzzy neural network and its application to path-tracking control of nonholonomic mobile robot," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 7, pp. 2667–2683, 2009.
- [22] C. H. Lee, Y. C. Lee, and F. Y. Chang, "A dynamic fuzzy neural system design via hybridization of EM and PSO algorithms," *IAENG International Journal of Computer Science*, vol. 37, no. 3, 2010.
- [23] W. G. Seo, B. H. Park, and J. S. Lee, "Adaptive fuzzy learning control for a class of nonlinear dynamic systems," *International Journal of Intelligent Systems*, vol. 15, no. 12, pp. 1157–1175, 2000.
- [24] C. J. Chien and L. C. Fu, "Iterative learning control of nonlinear systems using neural network design," *Asian Journal of Control*, vol. 4, no. 1, pp. 21–29, 2002.
- [25] C. J. Chien, "A combined adaptive law for fuzzy iterative learning control of nonlinear systems with varying control tasks," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 1, pp. 40–51, 2008.
- [26] T. W. S. Chow and Y. Fang, "A recurrent neural-network-based real-time learning control strategy applying to nonlinear systems with unknown dynamics," *IEEE Transactions on Industrial Electronics*, vol. 45, no. 1, pp. 151–161, 1998.
- [27] Y. C. Wang, C. J. Chien, and D. T. Lee, "An output recurrent fuzzy neural network based iterative learning control for nonlinear systems," in *Proceedings of the IEEE International*



*Conference on Fuzzy Systems*, pp. 1563–1569, Hong Kong, 2008.

- [28] Y. C. Wang and C. C. Teng, “Output recurrent fuzzy neural networks based model reference control for unknown nonlinear systems,” *International Journal of Fuzzy Systems*, vol. 6, no. 1, pp. 28–37, 2004.
- [29] C. C. Ku and K. Y. Lee, “Diagonal recurrent neural networks for dynamic systems control,” *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 144–156, 1995.

