Output Feedback Control of a Class of Discrete MIMO Nonlinear Systems With Triangular Form Inputs

Jin Zhang, Shuzhi Sam Ge, *Senior Member, IEEE*, and Tong Heng Lee, *Member, IEEE*

***Abstract*—In this paper, adaptive neural network (NN) control is investigated for a class of discrete-time multi-input–multi-output (MIMO) nonlinear systems with triangular form inputs. Each subsystem of the MIMO system is in strict feedback form. First, through two phases of coordinate transformation, the MIMO system is transformed into input–output representation with the triangular form input structure unchanged. By using high-order neural networks (HONNs) as the emulators of the desired controls, effective output feedback adaptive control is developed using backstepping. The closed-loop system is proved to be semiglobally uniformly ultimate bounded (SGUUB) by using Lyapunov method. The output tracking errors are guaranteed to converge into a compact set whose size is adjustable, and all the other signals in the closed-loop system are proved to be bounded. Simulation results show the effectiveness of the proposed control scheme.**

***Index Terms*—Discrete-time system, high-order neural networks (HONNs), multi-input–multi-output (MIMO) system, neural networks (NNs).**

I. Introduction

NEURAL networks (NNs) control of nonlinear systems has been extensively studied in the past decades. The universal approximation ability of NNs makes it one of the effective tool in nonlinear system identification and control [1]–[9].

Due to the difficulties in discrete-time systems, such as the noncausal problem [10] in backstepping design, the works in discrete-time domain are much less than those in continuous domain. In [1], input–output-based NN control was studied for a class of nonlinear discrete-time systems. In [11], multilayer NN was used in control of a class of discrete-time nonlinear systems with general relative degree. Offline training is needed to provide a good starting point for the online adaptive control. In [12], both state feedback and output feedback schemes were investigated for a class of discrete-time systems with general relative degree and bounded disturbances. For a class of discrete-time systems in strict feedback form, an effective backstepping design method was proposed in [10], in which, the noncausal problem was elegantly solved through the introduction of an n-step ahead descriptor. All these good NN controllers are designed for single-input–single-output (SISO) discrete-time nonlinear systems.

For multi-input–multi-output (MIMO) discrete-time systems, some results can be found in [13]–[16] for affine MIMO systems, i.e., the control inputs appear linearly, which makes feedback linearization method applicable. For nonaffine discrete-time MIMO systems, due to the inputs are in nonaffine form, feedback linearization method cannot be used. Therefore, how to find the "inverse" control, if there is one, is a problem that needs to be investigated. In [17], state feedback control scheme was investigated for a class of discrete-time nonlinear MIMO systems with triangular form inputs and bounded disturbances by using NNs. Though the method proposed is effective, all the system states are needed in order to construct the stable control. In [8], NN control was proposed for a class of nonaffine MIMO nonlinear autoregressive average with exogenous inputs (NARMAX) systems. First, SISO plants were studied, then the results were extended to MIMO cases. By first-order Taylor linearization, NNs were used to construct the inverse model for the linearized systems, which makes the results local. Additional restrictions include: 1) there is no input coupling in the system studied in [8], which avoided one of the major difficulties in MIMO nonlinear system control; and 2) NN identification should be carried out in advance in order to make the control implementable if the plant model is unknown.

In this paper, we are considering a class of MIMO nonlinear discrete-time systems with triangular form inputs [6]. Each subsystem of the MIMO system is in strict feedback form. Though the jth input appears linearly in the jth subsystem, the other $j-1$ inputs appear nonlinearly in the jth subsystem, which leads to the whole system in nonaffine form. First, through two phases of coordinate transformation, the system studied is transformed from state–space model into input–output representation, with each subsystem is in τ-step (τ is the system delay) predictor form and the triangular form inputs remains unchanged. Then, backstepping design is implemented. NNs and input–output sequences are used to construct the stable control. Comparing with the MIMO nonaffine system studied in the literature, we can see that: 1) there are complex inputs coupling; and 2) NN identification is not needed in this paper. The main contributions of this paper can be summarized as follows.

- An effective NN control scheme is developed for a class of complex nonlinear discrete-time nonaffine MIMO systems in state–space representation, for which, feedback linearization cannot be implemented.
- Only input and output sequences are used to construct the stable control, which is simple and easy to be implemented in practical applications.

Manuscript received November 10, 2003; revised February 16, 2005. This work was supported in part by the National Natural Science Foundation of China.

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

Digital Object Identifier 10.1109/TNN.2005.852242

1045-9227/$20.00 © 2005 IEEE

- System transformation is introduced, which transform the system from state–space description into input–output representation, and extends our previous works in [10] from SISO systems to MIMO systems.
- τ-step update laws are implemented, which are effective for this class of MIMO systems.

II. MIMO System Dynamics

Consider the following discrete-time MIMO system in state–space representation:

$$\Sigma_1 \begin{cases} x_{1,i_1}(k+1) = f_{1,i_1}\left(\bar{x}_{1,i_1}(k)\right) \\ \qquad\qquad + g_{1,i_1}\left(\bar{x}_{1,i_1}(k)\right) x_{1,i_1+1}(k) \\ \qquad\qquad 1 \le i_1 \le \tau - 1 \\ x_{1,\tau}(k+1) = f_{1,\tau}\left(X(k)\right) + g_{1,\tau}\left(X(k)\right) u_1(k) \end{cases}$$
$$\vdots$$
$$\Sigma_j \begin{cases} x_{j,i_j}(k+1) = f_{j,i_j}\left(\bar{x}_{j,i_j}(k)\right) \\ \qquad\qquad + g_{j,i_j}\left(\bar{x}_{j,i_j}(k)\right) x_{j,i_j+1}(k) \\ \qquad\qquad 1 \le i_j \le \tau - 1 \\ x_{j,\tau}(k+1) = f_{j,\tau}\left(X(k), \bar{u}_{j-1}(k)\right) \\ \qquad\qquad + g_{j,\tau}\left(X(k)\right) u_j(k) \end{cases}$$
$$\vdots$$
$$\Sigma_n \begin{cases} x_{n,i_n}(k+1) = f_{n,i_n}\left(\bar{x}_{n,i_n}(k)\right) \\ \qquad\qquad + g_{n,i_n}\left(\bar{x}_{n,i_n}(k)\right) x_{n,i_n+1}(k) \\ \qquad\qquad 1 \le i_n \le \tau - 1 \\ x_{n,\tau}(k+1) = f_{n,\tau}\left(X(k), \bar{u}_{n-1}(k)\right) \\ \qquad\qquad + g_{n,\tau}\left(X(k)\right) u_n(k) \end{cases}$$
$$y_j(k) = x_{j,1}(k), \quad 1 \le j \le n \tag{1}$$

where $X(k) = [x_1^T(k), x_2^T(k), \ldots, x_n^T(k)]^T$ with $x_j(k) = [x_{j,1}(k), x_{j,2}(k), \ldots, x_{j,\tau}(k)]^T \in R^\tau$ (τ is the system delay), $u_j(k) \in R^n$ and $y_j(k) \in R^n$ are the state variables, the system inputs and outputs, respectively, $\bar{u}_{j-1}(k) = [u_1(k), \cdots, u_{j-1}(k)]$ $(j = 2, \ldots, n)$; $\bar{x}_{j,i_j}(k) = [x_{j,1}(k), \ldots, x_{j,i_j}(k)]^T \in R^{i_j}$ denotes the first i_j states of the jth subsystem; $f_{j,i_j}(\cdot)$ and $g_{j,i_j}(\cdot)$ are smooth nonlinear functions. Noting that the control inputs of the whole system are in triangular form, then backstepping can be used to design stable controls for this class of systems. It is obvious that there are n subsystems in system (1), with the length of each subsystem being τ, and system (1) has n inputs and n outputs.

Assumption 1: The sign of $g_{j,i_j}(\cdot)$ ($j = 1, \ldots, n$ and $i_j = 1, \ldots, \tau$), are known and there exist two constants $\underline{g}_{j,i_j}, \bar{g}_{j,i_j} > 0$ such that $\underline{g}_{j,i_j} \le |g_{j,i_j}(\cdot)| \le \bar{g}_{j,i_j}, \forall X(k) \in \Omega \subset R^{n\times\tau}$.

Without losing generality, we shall assume that $g_{j,i_j}(\cdot)$ is positive in this paper.

The control objective is to design control input $u(k) = [u_1(k), \ldots, u_n(k)]^T$ to drive the system output $y(k) = [y_1(k), \ldots, y_n(k)]^T$ to follow a known and bounded trajectory $y_d(k) = [y_{d_1}(k), \ldots, y_{d_n}(k)]^T$.

Assumption 2: The desired trajectory $y_d(k) \in \Omega_y, \forall k > 0$ is smooth and known, where $\Omega_y \triangleq \{\chi | \chi = y(k)\}$.

Lemma 1: Consider the linear time varying discrete-time system given by

$$x(k+1) = A(k)x(k) + Bu(k), \quad y(k) = Cx(k) \tag{2}$$

where $A(k)$, B and C are appropriately dimensional matrices with B and C being constant matrices. Let $\Phi(k_1, k_0)$ be the state-transition matrix corresponding to $A(k)$ for system (2), i.e., $\Phi(k_1, k_0) = \prod_{k=k_0}^{k_1-1} A(k)$. If $\|\Phi(k_1, k_0)\| < 1$, $\forall k_1 > k_0 \ge 0$, then system (2) is 1) globally exponentially stable for the unforced system (i.e., $u(k) = 0$); and 2) bounded-input–bounded-output (BIBO) stable [12].

III. Function Approximation by HONN

In control engineering, NN has been successfully used as function approximators to solve different problems. The most frequently used NNs belongs to the linear-in-parameters function approximators which include radial basis function neural networks (RBFNNs) [18], polynomials [19], high-order neural networks (HONNs) [6] splines [20], fuzzy systems [21] and wavelet networks [22], among others. In terms of stability, they are also most equivalent. Their differences lie in the control performance, and difficulty of implementation [18]. The use of HONN is partially because of its relatively smaller size than that of RBFNN for its curse of dimensionality. For clarity, we consider the HONNs

$$\begin{aligned} \phi(W, z) &= W^T S(z), \quad W \in R^{l\times p} \quad \text{and} \quad S(z) \in R^l \\ S(z) &= [s_1(z), s_2(z), \ldots, s_l(z)]^T \\ s_i(z) &= \prod_{j \in I_i} [s(z_j)]^{d_j(i)}, \quad i = 1, 2, \ldots, l \end{aligned}$$

where $z = [z_1, z_2, \cdots, z_q]^T \in \Omega_z \subset R^q$, positive integer l denotes the NN node number, and p is the dimension of function vector, $\{I_1, I_2, \ldots, I_l\}$ is a collection of l not-ordered subsets of $\{1, 2, \ldots, q\}$ and $d_j(i)$ are nonnegative integers, W is an adjustable synaptic weight matrix, $s(z_j)$ is chosen as hyperbolic tangent function $s(z_j) = (e^{z_j} - e^{-z_j})/(e^{z_j} + e^{-z_j})$.

For a desired function $u^*(z)$, there exist ideal weights W^* such that the smooth function u^* can be approximated by an ideal NN on a compact set $\Omega_z \subset R^q$

$$u^* = W^{*T} S(z) + \epsilon_z \tag{3}$$

where ϵ_z is the bounded NN approximation error satisfying $|\epsilon_z| \le \epsilon_0$ on the compact set, which can be reduced by increasing the number of the adjustable weights. The ideal weight matrix W^* is an "artificial" quantity required for analytical purpose, and is defined as that minimizes $|\epsilon_z|$ for all $z \in \Omega_z \subset R^q$ in a compact region, i.e.

$$W^* \triangleq \arg \min_{W \in R^{l\times p}} \left\{ \sup_{z \in \Omega_z} \left| u^* - W^T S(z) \right| \right\}, \quad \Omega_z \subset R^q. \tag{4}$$

In general, the ideal NN weight matrix, W^*, is unknown though constant, its estimate $\hat{W}$ should be used for controller design which will be discussed later.

IV. System Coordinate Transformation

In this section, the procedure of how to transform system (1) from state–space description into input–output description is illustrated. In general, the transformation procedure can be divided into two phases.

A. *Coordinate Transformation: Phase I*

Consider the ith $(1 \leq i \leq n)$ subsystem of system (1)

$$\Sigma_i \begin{cases} x_{i,1}(k+1) = f_{i,1}\left(\bar{x}_{i,1}(k)\right) \\ \qquad\qquad + g_{i,1}\left(\bar{x}_{i,1}(k)\right) x_{i,2}(k) \\ \qquad\vdots \\ x_{i,\tau-1}(k+1) = f_{i,\tau-1}\left(\bar{x}_{i,\tau-1}(k)\right) \\ \qquad\qquad + g_{i,\tau-1}\left(\bar{x}_{i,\tau-1}(k)\right) x_{i,\tau}(k) \\ x_{i,\tau}(k+1) = f_{i,\tau}\left(X(k), \bar{u}_{i-1}(k)\right) \\ \qquad\qquad + g_{i,\tau}\left(X(k)\right) u_i(k) \end{cases}. \tag{5}$$

Define new coordinates $(1 \leq i \leq n)$

$$\xi_i = [\xi_{i,1}, \xi_{i,2}, \ldots, \xi_{i,\tau}]^T \tag{6}$$

with each element of ξ_i is defined as follows:

$$\begin{cases} \xi_{i,1}(k) = x_{i,1}(k) \\ \qquad\vdots \\ \xi_{i,\tau}(k) = x_{i,1}(k+\tau-1) \end{cases}. \tag{7}$$

Thus, the original system state $X = [x_1, x_2, \ldots, x_n]^T \in R^{n\times\tau}$ can be transformed into Ξ defined as

$$\Xi = \left[\xi_1^T, \xi_2^T, \ldots, \xi_n^T\right]^T \in R^{n\times\tau}. \tag{8}$$

Define this mapping as

$$T(X) : X \rightarrow \Xi. \tag{9}$$

In order to guarantee that this transformation is valid, in the following, we prove that the mapping is diffeomorphism [23], [24]. Considering (7), it can be obtained that

$$\begin{cases} \xi_{i,1}(k+1) = \xi_{i,2}(k) \\ \qquad\vdots \\ \xi_{i,\tau-1}(k+1) = \xi_{i,\tau}(k) \\ \xi_{i,\tau}(k+1) = x_{i,1}(k+\tau) \end{cases}. \tag{10}$$

Define

$$\begin{aligned} p_{i,1}\left(\bar{x}_{i,1}(k)\right) &\triangleq f_{i,1}\left(\bar{x}_{i,1}(k)\right) \\ q_{i,1}\left(\bar{x}_{i,1}(k)\right) &\triangleq g_{i,1}\left(\bar{x}_{i,1}(k)\right). \end{aligned}$$

Considering the first equation in (5), we obtain

$$\begin{aligned} x_{i,1}(k+1) &= f_{i,1}\left(\bar{x}_{i,1}(k)\right) + g_{i,1}\left(\bar{x}_{i,1}(k)\right) x_{i,2}(k) \\ &= p_{i,1}\left(\bar{x}_{i,1}(k)\right) + q_{i,1}\left(\bar{x}_{i,1}(k)\right) x_{i,2}(k) \end{aligned}$$

Noting that $x_{i,1}(k+1)$ depends on $\bar{x}_{i,2}(k)$, we can define $x_{i,1}(k+1) \triangleq \alpha_{i,1}(\bar{x}_{i,2}(k))$ with $\alpha_{i,1}(\cdot)$ being a nonlinear function. Considering (5) and (10), similarly, we can define nonlinear functions $p_{i,j}(\cdot)$, $q_{i,j}(\cdot)$ and $\alpha_{i,j}(\cdot)$ $(j = 2, \ldots, \tau-1)$ as follows:

$$\begin{aligned} x_{i,1}(k+2) &= f_{i,1}\left(\bar{x}_{i,1}(k+1)\right) \\ &\quad + g_{i,1}\left(\bar{x}_{i,1}(k+1)\right) x_{i,2}(k+1) \\ &= f_{i,1}\left(\alpha_{i,1}\left(\bar{x}_{i,2}(k)\right)\right) + g_{i,1}\left(\alpha_{i,1}\left(\bar{x}_{i,2}(k)\right)\right) \\ &\quad \times \left[f_{i,2}\left(\bar{x}_{i,2}(k)\right) + g_{i,2}\left(\bar{x}_{i,2}(k)\right) x_{i,3}(k)\right] \\ &= p_{i,2}\left(\bar{x}_{i,2}(k)\right) + q_{i,2}\left(\bar{x}_{i,2}(k)\right) x_{i,3}(k) \\ &= \alpha_{i,2}\left(\bar{x}_{i,3}(k)\right) \end{aligned}$$

with

$$\begin{aligned} p_{i,2}\left(\bar{x}_{i,2}(k)\right) &\triangleq f_{i,1}\left(\alpha_{i,1}\left(\bar{x}_{i,2}(k)\right)\right) \\ &\quad + g_{i,1}\left(\alpha_{i,1}\left(\bar{x}_{i,2}(k)\right)\right) f_{i,2}\left(\bar{x}_{i,2}(k)\right) \\ q_{i,2}\left(\bar{x}_{i,2}(k)\right) &\triangleq g_{i,1}\left(\alpha_{i,1}\left(\bar{x}_{i,2}(k)\right)\right) \times g_{i,2}\left(\bar{x}_{i,2}(k)\right) \\ \alpha_{i,2}\left(\bar{x}_{i,3}(k)\right) &\triangleq p_{i,2}\left(\bar{x}_{i,2}(k)\right) + q_{i,2}\left(\bar{x}_{i,2}(k)\right) x_{i,3}(k). \end{aligned}$$

Repeating this procedure recursively, we have

$$\begin{aligned} x_{i,1}(k+\tau-1) &\triangleq p_{i,\tau-1}\left(\bar{x}_{i,\tau-1}(k)\right) \\ &\quad + q_{i,\tau-1}\left(\bar{x}_{i,\tau-1}(k)\right) x_{i,\tau}(k) \\ &\triangleq \alpha_{i,\tau-1}\left(\bar{x}_{i,\tau}(k)\right) \end{aligned} \tag{11}$$

with $p_{i,j}(\cdot)$, $q_{i,j}(\cdot)$ and $\alpha_{i,j}(\cdot)$ $(j = 1, \ldots, \tau-1)$ being nonlinear functions.

Remark 1: Due to the boundedness of $g_{j,i_j}(\cdot)$ $(1 \leq j \leq n, 1 \leq i_j \leq \tau)$ in Assumption 1, $q_{i,j}(\cdot)$ is also bounded.

Now considering $x_{i,1}(k+\tau-1)$, we know that

$$x_{i,1}(k+\tau-1) = p_{i,\tau-1}\left(\bar{x}_{i,\tau-1}(k)\right) + q_{i,\tau-1}\left(\bar{x}_{i,\tau-1}(k)\right) x_{i,\tau}(k) \tag{12}$$

with $p_{i,\tau-1}(\cdot)$ and $q_{i,\tau-1}(\cdot)$ being highly entangled nonlinear functions. Proceeding one more step and noting the last equation in (5), we have

$$\begin{aligned} x_{i,1}(k+\tau) &= p_{i,\tau-1}\left(\bar{x}_{i,\tau-1}(k+1)\right) \\ &\quad + q_{i,\tau-1}\left(\bar{x}_{i,\tau-1}(k+1)\right) x_{i,\tau}(k+1) \\ &\triangleq p_{i,\tau}\left(\bar{x}_{i,\tau}(k)\right) + q_{i,\tau}\left(\bar{x}_{i,\tau}(k)\right) \\ &\quad \times \left[f_{i,\tau}\left(X(k), \bar{u}_{i-1}(k)\right)\right. \\ &\quad \left. + g_{i,\tau}\left(X(k)\right) u_i(k)\right] \\ &\triangleq p_i\left(X(k), \bar{u}_{i-1}(k)\right) + q_i\left(X(k)\right) u_i(k) \\ p_i\left(X(k), \bar{u}_{i-1}(k)\right) &\triangleq p_{i,\tau}\left(\bar{x}_{i,\tau}(k)\right) + q_{i,\tau}\left(\bar{x}_{i,\tau}(k)\right) \\ &\quad \times f_{i,\tau}\left(X(k), \bar{u}_{i-1}(k)\right) \\ q_i\left(X(k)\right) &\triangleq q_{i,\tau}\left(\bar{x}_{i,\tau}(k)\right) g_{i,\tau}\left(X(k)\right). \end{aligned} \tag{13}$$

Remark 2: Noting Assumption 1, Remark 1 and that we have assumed the positiveness of $g_{j,i_j}(\cdot)$, it can be easily obtained that $q_i(X(k)) = q_{i,\tau}(\bar{x}_{i,\tau}(k)) g_{i,\tau}(X(k))$ is also bounded. Specifically, there are two positive constants $\underline{q}_i$ and $\bar{q}_i$ such that, $\underline{q}_i \leq q_i(\cdot) \leq \bar{q}_i$ $(1 \leq i \leq n)$.

Therefore, the original system (1) becomes $(1 \leq i \leq n)$

$$\Sigma_i : \begin{cases} \xi_{i,1}(k+1) = \xi_{i,2}(k) \\ \quad\vdots \\ \xi_{i,\tau-1}(k+1) = \xi_{i,\tau}(k) \\ \xi_{i,\tau}(k+1) = p_i\left(X(k), \bar{u}_{i-1}(k)\right) \\ \qquad\qquad + q_i\left(X(k)\right) u_i(k) \end{cases} \tag{14}$$

provided that the coordinate transformation, $T(X)$, is diffeomorphism. In the next, we will show that the mapping $T(X)$ is indeed diffeomorphism.

Considering (7) and (11), the mapping from $x_i(k) = [x_{i,1}(k), x_{i,2}(k), \ldots, x_{i,\tau}(k)]^T$ to $\xi_i(k) = [\xi_{i,1}(k), \xi_{i,2}(k), \ldots(k), \xi_{i,\tau}(k)]^T$ can be expressed as follows:

$$\begin{cases} \xi_{i,1}(k) = x_{i,1}(k) \\ \xi_{i,2}(k) = x_{i,1}(k+1) \\ \qquad = p_{i,1}\left(\bar{x}_{i,1}(k)\right) + q_{i,1}\left(\bar{x}_{i,1}(k)\right) x_{i,2}(k) \\ \xi_{i,3}(k) = x_{i,1}(k+2) \\ \qquad = p_{i,2}\left(\bar{x}_{i,2}(k)\right) + q_{i,2}\left(\bar{x}_{i,2}(k)\right) x_{i,3}(k) \\ \qquad\vdots \\ \xi_{i,\tau-1}(k) = x_{i,1}(k+\tau-2) \\ \qquad = p_{i,\tau-2}\left(\bar{x}_{i,\tau-2}(k)\right) \\ \qquad\quad + q_{i,\tau-2}\left(\bar{x}_{i,\tau-2}(k)\right) x_{i,\tau-1}(k) \\ \xi_{i,\tau}(k) = x_{i,1}(k+\tau-1) \\ \qquad = p_{i,\tau-1}\left(\bar{x}_{i,\tau-1}(k)\right) \\ \qquad\quad + q_{i,\tau-1}\left(\bar{x}_{i,\tau-1}(k)\right) x_{i,\tau}(k) \end{cases} . \tag{15}$$

From (15), it can be seen that the coordinate transformation is the following.

- Decoupled subsystems: the coordinate transformation from $x_i(k) = [x_{i,1}(k), \ldots, x_{i,\tau}(k)]^T$ to $\xi_i(k) = [\xi_{i,1}(k), \ldots, \xi_{i,\tau}(k)]^T$ are independent of other subsystems.
- Independent of the control input $u_i(k)$: the coordinate transformation has nothing to do with the control inputs.

Define this mapping as follows $(1 \leq i \leq n)$:

$$T_i\left(x_i(k)\right) : x_i(k) \rightarrow \xi_i(k) \tag{16}$$

then we know that the whole system coordinate transformation from $X(k)$ to $\Xi(k)$ defined in (9) can be written as follows:

$$T\left(X(k)\right) = \begin{bmatrix} T_1\left(x_1(k)\right) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & T_2\left(x_2(k)\right) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & T_n\left(x_n(k)\right) \end{bmatrix} . \tag{17}$$

If we can verify that the mapping $T_i(x_i(k))$ in (15) is diffeomorphism [23], [24], then owing to the independent property of $T_i(x_i(k))$ $(1 \leq i \leq n)$, we know that the whole system coordinate transformation, $T(X(k))$ in (17) is also diffeomorphism.

Lemma 2: Let U be an open subset of R^n and let $\varphi = (\varphi_1, \ldots, \varphi_n)$: $U \rightarrow R^n$ be a smooth map. If the Jacobian Matrix

$$\frac{d\varphi}{dx} = \begin{bmatrix} \frac{\partial \varphi_1}{\partial x_1} & \cdots & \frac{\partial \varphi_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \varphi_n}{\partial x_1} & \cdots & \frac{\partial \varphi_n}{\partial x_n} \end{bmatrix}$$

is nonsingular at some point $p \in U$, or equivalently, $\text{Rank}(d\varphi/dx) = n$ at some point $p \in U$, then there exists a neighborhood $V \subset U$ of p such that φ: $V \rightarrow \varphi(V)$ is a diffeomorphism [10], [23], [24].

Lemma 3: The mapping $T(X(k))$: $X(k) \rightarrow \Xi(k)$, defined in (9) and (16) as

$$T\left(X(k)\right) = \text{diag}\left[T_1\left(x_1(k)\right), T_2\left(x_2(k)\right), \ldots, T_n\left(x_n(k)\right)\right]$$

is a diffeomorphism.

Proof: The proof for the diffeomorphism of $T_i(x_i(k))$ can be found in [10]. For completeness, it is also detailed here.

Considering the ith subsystem $(1 \leq i \leq n)$, we have $\xi_i(k) = T_i(x_i(k))$. It is shown in Lemma 2 that once 1) the map $T_i(\cdot)$ is invertible, and 2) $T_i(\cdot)$ and $T_i^{-1}(\cdot)$ are both continuously differentiable, then the map $T_i(\cdot)$ is a diffeomorphism.

Noting that

$$\begin{aligned} x_{i,1}(k) &= \xi_{i,1}(k) \\ x_{i,1}(k+1) &= \xi_{i,2}(k) \end{aligned}$$

and that

$$x_{i,2}(k) = \frac{x_{i,1}(k+1) - f_{i,1}\left(\bar{x}_{i,1}(k)\right)}{g_{i,1}\left(\bar{x}_{i,1}(k)\right)}$$

we have

$$x_{i,2}(k) = \frac{\xi_{i,2}(k) - f_{i,1}\left(\bar{x}_{i,1}(k)\right)}{g_{i,1}\left(\bar{x}_{i,1}(k)\right)} = \frac{\xi_{i,2}(k) - f_{i,1}\left(\bar{\xi}_{i,1}(k)\right)}{g_{i,1}\left(\bar{\xi}_{i,1}(k)\right)} .$$

Therefore, we can define that

$$x_{i,2}(k) \triangleq t_{i,2}\left(\xi_{i,1}(k), \xi_{i,2}(k)\right) \triangleq t_{i,2}(k) \tag{18}$$

with

$$t_{i,2}(k) = \frac{\xi_{i,2}(k) - f_{i,1}\left(\bar{\xi}_{i,1}(k)\right)}{g_{i,1}\left(\bar{\xi}_{i,1}(k)\right)}$$

It is clear that

$$\begin{aligned} x_{i,2}(k+1) &= t_{i,2}(k+1) \\ \frac{\partial t_{i,2}(k)}{\partial \xi_{i,2}(k)} &= \frac{1}{g_{i,1}\left(\bar{\xi}_{i,1}(k)\right)} \\ \frac{\partial t_{i,2}(k+1)}{\partial \xi_{i,3}(k)} &= \frac{\partial\left[\frac{\xi_{i,2}(k+1) - f_{i,1}\left(\bar{\xi}_{i,1}(k+1)\right)}{g_{i,1}\left(\bar{\xi}_{i,1}(k+1)\right)}\right]}{\partial \xi_{i,3}(k)} \\ &= \frac{1}{g_{i,1}\left(\bar{\xi}_{i,2}(k)\right)} \\ &\quad \left(\text{noting } \xi_{i,2}(k+1) = \xi_{i,3}(k)\right) . \end{aligned}$$

Because

$$x_{i,3}(k) = \frac{x_{i,2}(k+1) - f_{i,2}\left(\bar{x}_{i,2}(k)\right)}{g_{i,2}\left(\bar{x}_{i,2}(k)\right).}$$

Therefore, we can obtain

$$\begin{aligned} x_{i,3}(k) =& \frac{t_{i,2}(k+1) - f_{i,2}\left([\xi_{i,1}(k), t_{i,2}(k)]^T\right)}{g_{i,2}\left([\xi_{i,1}(k), t_{i,2}(k)]^T\right)} \\ \triangleq& \, t_{i,3}(k). \end{aligned}$$

It is obvious that

$$\begin{aligned} \frac{\partial t_{i,3}(k)}{\partial \xi_{i,3}(k)} =& \frac{\frac{\partial t_{i,2}(k+1)}{\partial \xi_{i,3}(k)}}{g_{i,2}\left([\xi_{i,1}(k), t_{i,2}(k)]^T\right)} \\ =& \frac{1}{g_{i,1}\left(\bar{\xi}_{i,2}(k)\right) \times g_{i,2}\left([\xi_{i,1}(k), t_{i,2}(k)]^T\right).} \end{aligned}$$

Continuing this process recursively, finally, we can obtain

$$\begin{aligned} x_{i,\tau}(k) =& \, t_{i,\tau}\left(\bar{\xi}_{i,\tau}(k)\right) \triangleq t_{i,\tau}(k) \\ \frac{\partial t_{i,\tau}(k)}{\partial \xi_{i,\tau}(k)} =& \frac{1}{g_{i,\tau}(k)} \end{aligned}$$

with $g_{i,\tau}(k) = g_{i,1}(\bar{\xi}_{i,2}(k)) \times g_{i,2}([\xi_{i,1}(k), t_{i,2}(k)]^T) \times \cdots \times g_{i,\tau}([\xi_{i,1}(k), t_{i,2}(k), \ldots, t_{i,\tau-1}(k)]^T)$.

Therefore, we can see that for the ith subsystem, the inverse transformation $T_i^{-1}(\xi_i(k))$ can be denoted as

$$T_i^{-1}\left(\xi_i(k)\right) = \begin{bmatrix} x_{i,1}(k) \\ t_{i,2}(k) \\ \vdots \\ t_{i,\tau}(k) \end{bmatrix} = \begin{bmatrix} x_{i,1}(k) \\ x_{i,2}(k) \\ \vdots \\ x_{i,\tau}(k) \end{bmatrix}$$

and, consequently, we have

$$\frac{\partial T_i^{-1}\left(\xi_i(k)\right)}{\partial \xi_i(k)} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \star & \frac{1}{g_{1,1}\left(\bar{\xi}_{1,2}(k)\right)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \star & \star & \star & \star \end{bmatrix}. \tag{19}$$

Similarly, for the other subsystems, this coordinate transformation still holds. Therefore, for the whole system, the inverse transformation from $\Xi(k)$ to $X(k)$ can be expressed as

$$T^{-1}(\Xi) = \begin{bmatrix} T_1^{-1}(\xi_1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & T_2^{-1}(\xi_2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & T_n^{-1}(\xi_n) \end{bmatrix}. \tag{20}$$

Noting (19), it can be concluded that the Jacobian matrix of $T^{-1}(\Xi)$ is both nonsingular and differentiable. Therefore, we conclude that both the mapping $T(X(k))$ and its inverse, $T^{-1}(\Xi(k))$ are all nonsingular and differentiable. Therefore, we have the following equation:

$$X(k) = T^{-1}\left(\Xi(k)\right) \tag{21}$$

and $T(X(k))$ is a diffeomorphism actually. This completes the proof. ■

Therefore, considering (14), we know that the ith subsystem of (1) is in the following form:

$$\begin{cases} \xi_{i,1}(k+1) = \xi_{i,2}(k) \\ \qquad \vdots \\ \xi_{i,\tau-1}(k+1) = \xi_{i,\tau}(k) \\ \xi_{i,\tau}(k+1) = p_i\left(X(k), \bar{u}_{i-1}(k)\right) \\ \qquad\qquad + q_i\left(X(k)\right) u_i(k) \end{cases}. \tag{22}$$

Noting (21), (22) can be written as

$$\begin{cases} \xi_{i,1}(k+1) = \xi_{i,2}(k) \\ \qquad \vdots \\ \xi_{i,\tau-1}(k+1) = \xi_{i,\tau}(k) \\ \xi_{i,\tau}(k+1) = f_i\left(\Xi(k), \bar{u}_{i-1}(k)\right) \\ \qquad\qquad + g_i\left(\Xi(k)\right) u_i(k) \end{cases} \tag{23}$$

with

$$\begin{aligned} f_i\left(\Xi(k), \bar{u}_{i-1}(k)\right) \triangleq& \, p_i\left(T^{-1}\left(\Xi(k)\right), \bar{u}_{i-1}(k)\right) \\ g_i\left(\Xi(k)\right) \triangleq& \, q_i\left(T^{-1}\left(\Xi(k)\right)\right). \end{aligned}$$

This completes the first phase of system coordinate transformation.

B. Coordinate Transformation: Phase II

Now, the original system (1) has been transferred into the following form:

$$\begin{aligned} &\Sigma_1 \begin{cases} \xi_{1,i_1}(k+1) = \xi_{1,i_1+1}(k), \quad 1 \le i_1 \le \tau - 1 \\ \xi_{1,\tau}(k+1) = f_1\left(\Xi(k)\right) \\ \qquad\qquad + g_1\left(\Xi(k)\right) u_1(k) \end{cases} \\ &\vdots \\ &\Sigma_j \begin{cases} \xi_{j,i_j}(k+1) = \xi_{j,i_j+1}(k), \quad 1 \le i_j \le \tau - 1 \\ \xi_{j,\tau}(k+1) = f_j\left(\Xi(k), \bar{u}_{j-1}(k)\right) \\ \qquad\qquad + g_j\left(\Xi(k)\right) u_j(k) \end{cases} \\ &\vdots \\ &\Sigma_n \begin{cases} \xi_{n,i_n}(k+1) = \xi_{n,i_n+1}(k), \quad 1 \le i_n \le \tau - 1 \\ \xi_{n,\tau}(k+1) = f_n\left(\Xi(k), \bar{u}_{n-1}(k)\right) \\ \qquad\qquad + g_n\left(\Xi(k)\right) u_n(k) \end{cases} \\ &y_j(k) = \xi_{j,1}(k), \quad 1 \le j \le n \end{aligned} \tag{24}$$

with $f_j(\cdot)$ and $g_j(\cdot)$ $(1 \le j \le n)$ being smooth nonlinear functions. Noting Remark 2, we know that there are two positive constants $\underline{g}_i$ and $\bar{g}_i > 0$, such that $\underline{g}_i \le g_i(\cdot) \le \bar{g}_i$ $(1 \le i \le n)$, $\forall \Xi(k) \in \Omega \subset R^{n \times \tau}$.

However, each subsystem in state–space representation (24) consists of τ states in its first $\tau - 1$ equations, therefore, how to guarantee the stability of each state by choosing appropriate control is a problem that needs to be solved.

Owing to the cascade property of each subsystem, we can find that for the jth subsystem, we have $\xi_{j,1}(k) = y_j(k)$, $\xi_{j,2}(k) = y_j(k+1), \ldots, \xi_{j,\tau}(k) = y_j(k+\tau-1)$. This motivates us to seek for input–output representation (38) for each subsystem, for which, the states will be stabilized once the output sequence is stabilized. In the following, the coordinate transformation procedure is outlined in the following:

1) for each subsystem, use its corresponding output sequence to represent their states;
2) transform each subsystem from state–space representation to τ-step predictor input–output presentation, at the same time, the triangular form inputs structure remains unchanged.

Motivated by the design procedure in [10], coordinate transformation is used to transform system (24) from state–space description to input–output description. Consider the jth $(1 \le j \le n)$ subsystem in system (24)

$$\Sigma_j : \begin{cases} \xi_{j,1}(k+1) = \xi_{j,2}(k) \\ \qquad\vdots \\ \xi_{j,\tau-1}(k+1) = \xi_{j,\tau}(k) \\ \xi_{j,\tau}(k+1) = f_j\left(\Xi(k), \bar{u}_{j-1}(k)\right) \\ \qquad\qquad + g_j\left(\Xi(k)\right) u_j(k) \end{cases} . \tag{25}$$

In order to develop the output feedback control scheme, define the following new variables:

$$\begin{aligned} \underline{y}_1(k) &= \left[y_1(k-\tau+1), \cdots, y_1(k-1), y_1(k)\right]^T \\ &\vdots \\ \underline{y}_n(k) &= \left[y_n(k-\tau+1), \cdots, y_n(k-1), y_n(k)\right]^T \end{aligned}$$

and

$$\begin{aligned} \underline{u}_1^{k-1}(k) &= \left[u_1(k-1), \cdots, u_1(k-\tau+1)\right]^T \\ &\vdots \\ \underline{u}_n^{k-1}(k) &= \left[u_n(k-1), \cdots, u_n(k-\tau+1)\right]^T . \end{aligned}$$

Furthermore, define

$$\begin{aligned} \underline{z}_1(k) &= \left[\underline{y}_1^T(k), \underline{u}_1^{k-1^T}(k)\right]^T \\ &\vdots \\ \underline{z}_n(k) &= \left[\underline{y}_n^T(k), \underline{u}_n^{k-1^T}(k)\right]^T \\ \underline{z}(k) &= \left[\underline{z}_1^T(k), \underline{z}_2^T(k), \cdots, \underline{z}_n^T(k)\right]^T \in R^{(2\tau-1)\times n}. \end{aligned}$$

According to the definition of the new states, we know that

$$\begin{aligned} \underline{y}_1(k) &= \left[\xi_{1,1}(k-\tau+1), \cdots, \xi_{1,1}(k-1), \xi_{1,1}(k)\right]^T \\ &\vdots \\ \underline{y}_n(k) &= \left[\xi_{n,1}(k-\tau+1), \cdots, \xi_{n,1}(k-1), \xi_{n,1}(k)\right]^T . \end{aligned}$$

Noting (25), we obtain

$$\begin{aligned} y_j(k+1) &= \xi_{j,2}(k) = \xi_{j,3}(k-1) = \cdots \\ &= \xi_{j,\tau}(k-\tau+2) \\ &= f_j\left(\Xi(k-\tau+1), \bar{u}_{j-1}(k-\tau+1)\right) \\ &\quad + g_j\left(\Xi(k-\tau+1)\right) u_j(k-\tau+1). \end{aligned} \tag{26}$$

Noting that

$$\begin{aligned} \Xi(k-\tau+1) &= [\xi_{1,1}(k-\tau+1), \ldots, \xi_{1,\tau}(k-\tau+1), \ldots, \\ &\qquad \xi_{n,1}(k-\tau+1), \ldots, \xi_{n,\tau}(k-\tau+1)]^T \\ &= [y_1(k-\tau+1), \ldots, \\ &\qquad y_1(k), \ldots, y_n(k-\tau+1), \ldots, y_n(k)]^T \\ &= \left[\underline{y}_1^T(k), \underline{y}_2^T(k), \ldots, \underline{y}_n^T(k)\right]^T \end{aligned}$$

and define

$$Y(k) = \left[\underline{y}_1^T(k), \underline{y}_2^T(k), \cdots, \underline{y}_n^T(k)\right]^T$$

we have

$$\Xi(k-\tau+1) = Y(k). \tag{27}$$

Now, (26) becomes

$$y_j(k+1) = f_j\left(Y(k), \bar{u}_{j-1}(k-\tau+1)\right) + g_j\left(Y(k)\right) u_j(k-\tau+1). \tag{28}$$

This means that $\xi_{j,2}(k)$ is a function of $Y(k)$, $\bar{u}_{j-1}(k-\tau+1)$ and $u_j(k-\tau+1)$. It should be noted that although the right-hand side of (28) does not contain all the elements of $\underline{z}(k)$, for convenience of analysis, we can denote (28) as follows without any ambiguity:

$$\begin{aligned} y_j(k+1) &= \xi_{j,2}(k) \\ &= f_j\left(Y(k), \bar{u}_j(k-\tau+1)\right) \\ &\quad + g_j\left(Y(k)\right) u_j(k-\tau+1) \\ &\triangleq \psi_{j,2}\left(\underline{z}(k)\right). \end{aligned} \tag{29}$$

It is obvious that

$$\begin{aligned} y_1(k+1) &= \xi_{1,2}(k) \\ &= f_1\left(Y(k)\right) + g_1\left(Y(k)\right) u_1(k-\tau+1) \\ &\triangleq \psi_{1,2}\left(\underline{z}(k)\right) \\ y_2(k+1) &= \xi_{2,2}(k) \\ &= f_2\left(Y(k), u_1(k-\tau+1)\right) \\ &\quad + g_2\left(Y(k)\right) u_2(k-\tau+1) \\ &\triangleq \psi_{2,2}\left(\underline{z}(k)\right) \\ &\ \ \vdots \\ y_n(k+1) &= \xi_{n,2}(k) \\ &= f_n(Y(k), u_1(k-\tau+1), \ldots, u_{n-1}(k-\tau+1)) \\ &\quad + g_n\left(Y(k)\right) u_n(k-\tau+1) \\ &\triangleq \psi_{n,2}\left(\underline{z}(k)\right). \end{aligned}$$

Thus, we obtain

$$Y(k+1) = \left[\underline{y}_1^T(k+1), \cdots, \underline{y}_n^T(k+1)\right]^T \triangleq \Psi_1\left(\underline{z}(k)\right). \tag{30}$$

Similarly, noting (27) and (28), we can obtain

$$\begin{aligned} y_j(k+2) &= \xi_{j,3}(k) \\ &= f_j\left(\Xi(k-\tau+2), \bar{u}_j(k-\tau+2)\right) \\ &\quad + g_j\left(\Xi(k-\tau+2)\right) u_j(k-\tau+2) \\ &= f_j\left(Y(k+1), u_j(k-\tau+2)\right) \\ &\quad + g_j\left(Y(k+1)\right) u_j(k-\tau+2). \end{aligned} \tag{31}$$

Substituting (30) into (31), we obtain

$$y_j(k+2) = \xi_{j,3}(k) \triangleq \psi_{j,3}\left(\underline{z}(k)\right). \tag{32}$$

Therefore, we can obtain

$$\begin{cases} y_1(k+2) = \xi_{1,3}(k) \triangleq \psi_{1,3}\left(\underline{z}(k)\right) \\ \qquad \vdots \\ y_n(k+2) = \xi_{n,3}(k) \triangleq \psi_{n,3}\left(\underline{z}(k)\right). \end{cases}$$

Noting (33), it can be easily obtained that

$$Y(k+2) = \left[\underline{y}_1^T(k+2), \cdots, \underline{y}_n^T(k+2)\right]^T \triangleq \Psi_2\left(\underline{z}(k)\right). \tag{33}$$

Repeating the previous procedure recursively, we can prove that

$$y_j(k+\tau-1) = \xi_{j,\tau}(k) \triangleq \psi_{j,\tau}\left(\underline{z}(k)\right). \tag{34}$$

This implies that $\xi_{j,\tau}(k)$ is a function of $\underline{z}(k)$. Similarly, the following equations hold:

$$\begin{aligned} y_1(k+\tau-1) &= \xi_{1,\tau}(k) \triangleq \psi_{1,\tau}\left(\underline{z}(k)\right) \\ &\vdots \\ y_n(k+\tau-1) &= \xi_{n,\tau}(k) \triangleq \psi_{n,\tau}\left(\underline{z}(k)\right). \end{aligned} \tag{35}$$

By noting (29), (32), and (34), we conclude that

$$\begin{aligned} \xi_j(k) &= \left[\xi_{j,1}(k), \xi_{j,2}(k), \cdots, \xi_{j,\tau}(k)\right]^T \\ &= \left[y_j(k), \psi_{j,2}\left(\underline{z}(k)\right), \cdots, \psi_{j,\tau}\left(\underline{z}(k)\right)\right]^T \\ &\triangleq \psi_j\left(\underline{z}(k)\right). \end{aligned}$$

Therefore, we have

$$\begin{aligned} \xi_1(k) &= \left[\xi_{1,1}(k), \xi_{1,2}(k), \cdots, \xi_{1,\tau}(k)\right]^T \triangleq \psi_1\left(\underline{z}(k)\right) \\ &\vdots \\ \xi_n(k) &= \left[\xi_{n,1}(k), \xi_{n,2}(k), \cdots, \xi_{n,\tau}(k)\right]^T \triangleq \psi_n\left(\underline{z}(k)\right). \end{aligned}$$

The system state $\Xi(k) = \left[\xi_1^T(k), \xi_2^T(k), \cdots, \xi_n^T(k)\right]^T$ also depends on $\underline{z}(k)$, that means

$$\Xi(k) = \mathbf{\Psi}\left(\underline{z}(k)\right) \tag{36}$$

with $\mathbf{\Psi}(\cdot)$ being a vector nonlinear function. Up to this step, $\mathbf{\Psi}(\cdot)$ contains all the elements of $\underline{z}(k)$.

Noting (34) and the last equation in system (25), we have

$$\begin{aligned} y_j(k+\tau) &= \xi_{j,\tau}(k+1) \\ &= f_j\left(\Xi(k), \bar{u}_{j-1}(k)\right) + g_j\left(\Xi(k)\right) u_j(k). \end{aligned} \tag{37}$$

Substituting (36) into (37), we have

$$y_j(k+\tau) = f_j\left(\mathbf{\Psi}\left(\underline{z}(k)\right), \bar{u}_{j-1}(k)\right) + g_j\left(\mathbf{\Psi}\left(\underline{z}(k)\right)\right).u_j(k)$$

Thus, we obtain the input–output representation of system (24) as follows:

$$\begin{cases} y_1(k+\tau) = f_1\left(\mathbf{\Psi}\left(\underline{z}(k)\right)\right) \\ \qquad\qquad + g_1\left(\mathbf{\Psi}\left(\underline{z}(k)\right)\right) u_1(k) \\ \qquad \vdots \\ y_j(k+\tau) = f_j\left(\mathbf{\Psi}\left(\underline{z}(k)\right), \bar{u}_{j-1}(k)\right) \\ \qquad\qquad + g_j\left(\mathbf{\Psi}\left(\underline{z}(k)\right)\right) u_j(k) \\ \qquad \vdots \\ y_n(k+\tau) = f_n\left(\mathbf{\Psi}\left(\underline{z}(k)\right), \bar{u}_{n-1}(k)\right) \\ \qquad\qquad + g_n\left(\mathbf{\Psi}\left(\underline{z}(k)\right)\right) u_n(k) \end{cases}. \tag{38}$$

For the convenience of analysis, define

$$\begin{aligned} f_1(k) &= f_1\left(\mathbf{\Psi}\left(\underline{z}(k)\right)\right) \\ g_1(k) &= g_1\left(\mathbf{\Psi}\left(\underline{z}(k)\right)\right) \\ f_2\left(k, \bar{u}_1(k)\right) &= f_2\left(\mathbf{\Psi}\left(\underline{z}(k)\right), \bar{u}_1(k)\right) \\ g_2(k) &= g_2\left(\mathbf{\Psi}\left(\underline{z}(k)\right)\right) \\ &\vdots \\ f_n\left(k, \bar{u}_{n-1}(k)\right) &= f_n\left(\mathbf{\Psi}\left(\underline{z}(k)\right), \bar{u}_{n-1}(k)\right) \\ g_n(k) &= g_n\left(\mathbf{\Psi}\left(\underline{z}(k)\right)\right). \end{aligned}$$

Remark 3: By now, we have successfully transformed the original MIMO system from state–space representation (24) into input–output representation (38), with the triangular form inputs structure unchanged. Considering the input–output representation (38), it can be taken as a τ-step ahead predictor model [10], in which, the current outputs are determined by system information of τ steps earlier. Thus, different from those traditional one step parameter update law [25] used for one-step ahead predictor, high-order update laws should be used to deal with this τ-step predictor model, which will be discussed later.

V. Controller Design and Stability Analysis

The closed-loop system structure is shown in Fig. 1. At the beginning of Section IV-B, we have shown that if (38) is stable, then the stability of (24) is guaranteed. In this section, we will develop stable adaptive NN controls and corresponding weight

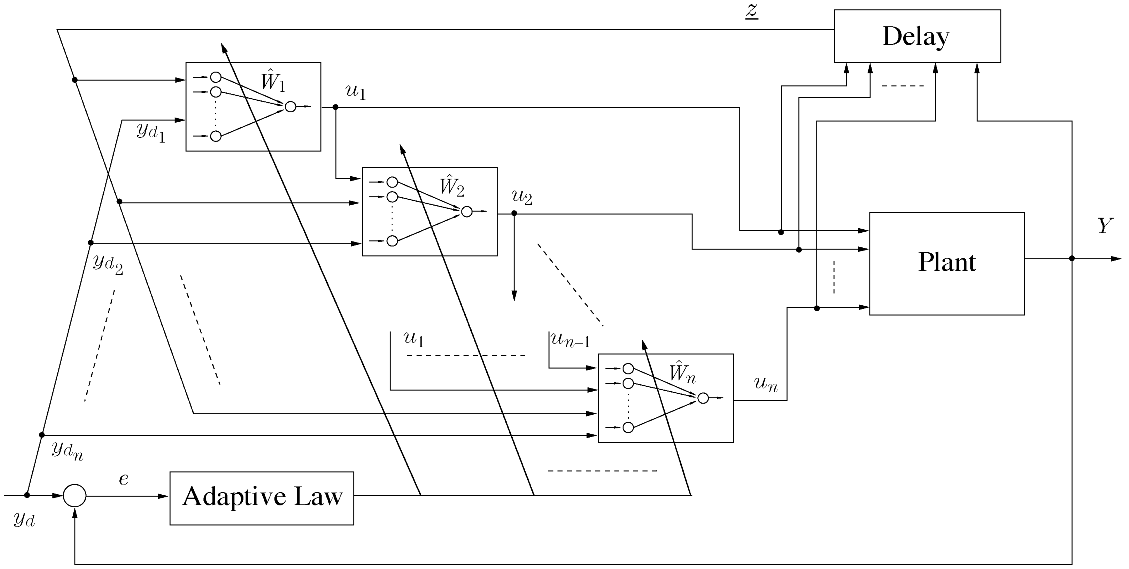

Fig. 1. Control system structure.

tuning laws for system (38), which will subsequently stabilize system (24).

Define tracking error as $e(k) = [e_1(k), \ldots, e_n(k)]^T$, with

$$e_i(k) = y_i(k) - y_{d_i}(k), \quad i = 1, \ldots, n \tag{39}$$

then the error dynamics can be obtained

$$\begin{aligned} e_1(k+\tau) &= f_1(k) + g_1(k)u_1(k) - y_{d_1}(k+\tau) \\ e_2(k+\tau) &= f_2(k, \bar{u}_1(k)) + g_2(k)u_2(k) - y_{d_2}(k+\tau) \\ &\vdots \\ e_n(k+\tau) &= f_n(k, \bar{u}_{n-1}(k)) + g_n(k)u_n(k) \\ &\quad - y_{d_n}(k+\tau). \end{aligned} \tag{40}$$

Considering the first equation in error dynamics (40), if we choose the desired control $u_1^*(k)$ as

$$u_1^*(k) = \frac{[y_{d_1}(k+\tau) - f_1(k)]}{g_1(k)} \tag{41}$$

then we can obtain $e_1(k+\tau) = 0$. Therefore, the tracking error $e_1(k)$ will reach zero in τ steps. However, in practical applications, normally, an exact system model cannot be obtained. Therefore, the desired control $u_1^*(k)$ is not applicable. Instead, we can use HONNs to approximate $u_1^*(k)$

$$\begin{aligned} u_1^*(k) &= W_1^{*T} S_1(z_1(k)) + \epsilon_{z_1}(z_1(k)) \\ z_1(k) &= [\underline{z}(k), y_{d_1}(k+\tau)]^T \in \Omega_{z_1} \subset R^{1+(2\tau-1)\times n}. \end{aligned} \tag{42}$$

Choose the practical adaptive control input $u_1(k)$ and robust updating algorithm for NN weights as

$$u_1(k) = \hat{W}_1^T(k) S_1(z_1(k)) \tag{43}$$

$$\begin{aligned} \hat{W}_1(k) &= \hat{W}_1(k-\tau) - \Gamma_1 \\ &\quad \times \left[S_1(z_1(k-\tau)) e_1(k) + \sigma_1 \hat{W}_1(k-\tau) \right] \end{aligned} \tag{44}$$

where $\Gamma_1^T = \Gamma_1 > 0$ is the adaptation diagonal gain matrix, $\hat{W}_1(k)$ denotes the estimate of $W_1^*(k)$, and σ_1 is a small positive constant.

Once $u_1(k)$ is confirmed, the desired control $u_2^*(k)$ can be chosen as

$$u_2^*(k) = \frac{[y_{d_2}(k+\tau) - f_2(k, \bar{u}_1(k))]}{g_2(k)} \tag{45}$$

which will drive $e_2(k+\tau) = 0$. Similarly, we can choose the corresponding HONN approximator and weight update law. Repeating the previous procedure recursively, at step i, we know that the desired control $u_i^*(k)$ is

$$u_i^*(k) = \frac{[y_{d_i}(k+\tau) - f_i(k, \bar{u}_{i-1}(k))]}{g_i(k)}. \tag{46}$$

Its HONN approximation is

$$\begin{aligned} u_i^*(k) &= W_i^{*T} S_i(z_i(k)) + \epsilon_{z_i}(z_i(k)) \\ z_i(k) &= [\underline{z}(k), \bar{u}_{i-1}(k), y_{d_i}(k+\tau)]^T \\ &\in \Omega_{z_i} \subset R^{i+(2\tau-1)\times n}. \end{aligned} \tag{47}$$

Accordingly, the practical control input $u_i(k)$ and its NN weight update law are chosen as follows:

$$u_i(k) = \hat{W}_i^T(k) S_i(z_i(k)) \tag{48}$$

$$\begin{aligned} \hat{W}_i(k) &= \hat{W}_i(k-\tau) - \Gamma_i \\ &\quad \times \left[S_i(z_i(k-\tau)) e_i(k) + \sigma_i \hat{W}_i(k-\tau) \right] \end{aligned} \tag{49}$$

where $\Gamma_i^T = \Gamma_i > 0$ is the adaptation diagonal gain matrix and $\hat{W}_i(k)$ denotes the estimation of $W_i^*(k)$. In the final step, we know that the desired control $u_n^*(k)$ is

$$u_n^*(k) = \frac{[y_{d_n}(k+\tau) - f_n(k, \bar{u}_{n-1}(k))]}{g_n(k)}. \tag{50}$$

Its HONN approximation is

$$\begin{aligned} u_n^*(k) &= W_n^{*T} S_n\left(z_n(k)\right) + \epsilon_{z_n}\left(z_n(k)\right) \\ z_n(k) &= \left[\underline{z}(k), \bar{u}_{n-1}(k), y_{d_n}(k+\tau)\right]^T \in \Omega_{z_n} \subset R^{2\tau n}. \end{aligned} \tag{51}$$

Accordingly, the practical control input $u_n(k)$ and its NN weight update law are chosen as follows:

$$u_n(k) = \hat{W}_n^T(k) S_n\left(z_n(k)\right) \tag{52}$$

$$\begin{aligned} \hat{W}_n(k) = {} & \hat{W}_n(k-\tau) - \Gamma_n \\ & \times \left[S_n\left(z_n(k-\tau)\right) e_n(k) + \sigma_n \hat{W}_n(k-\tau)\right] \end{aligned} \tag{53}$$

where $\Gamma_n^T = \Gamma_n > 0$ is the adaptation diagonal gain matrix and $\hat{W}_n(k)$ denotes the estimation of $W_n^*(k)$.

Summarizing (43) and (44), (48) and (49), (52) and (53), we propose the HONN controls and weight update laws for system (38) as follows:

$$u_i(k) = \hat{W}_i^T(k) S_i\left(z_i(k)\right) \tag{54}$$

$$\hat{W}_i(k) = \hat{W}_i(k-\tau) - \Gamma_i[S_i\left(z_i(k-\tau)\right) e_i(k) \tag{55}$$

$$+ \sigma_i \hat{W}_i(k-\tau)] \tag{56}$$

where $i = 1, \ldots, n$, $z_i(k) = [\underline{z}(k), \bar{u}_{i-1}(k), y_{d_i}(k+\tau)]^T \in \Omega_{z_i} \subset R^{i+(2\tau-1)\times n}$, $\Gamma_i = \mathrm{diag}[\gamma_{i_1}, \gamma_{i_2}, \ldots, \gamma_{i_n}] > 0$ is diagonal adaptation gain matrix, $0 < \gamma_{i_1}, \gamma_{i_2}, \ldots, \gamma_{i_n} < 1$ and $0 < \sigma_i < 1$ are positive constants. It should be noticed that, in the NN weights update laws (55), σ-modification [26] is used to improve the robustness of the controller. For the ease of analysis, (55) can also be written as

$$\hat{W}_i(k+\tau) = \hat{W}_i(k) - \Gamma_i\left[S_i\left(z_i(k)\right) e_i(k+\tau) + \sigma_i \hat{W}_i(k)\right]. \tag{57}$$

Theorem 1: The closed-loop nonlinear MIMO system consisting of system (1), NN controls (54) and NN weight update laws (55) is semiglobally uniformly ultimately bounded (SGUUB), and has an equilibrium at $[e_1(k), \ldots, e_n(k)]^T = 0$ provided that the design parameters are properly chosen as

$$\bar{\gamma}_i < \frac{1}{(1 + l_i + \bar{g}_i l_i)}, \quad \sigma_i < \frac{1}{[(1+\bar{g}_i)\bar{\gamma}_i]} \quad (i = 1, \ldots, n).$$

This guarantees that all the signals include the state vector $X(k)$, the control inputs $u_i(k)$ and NN weight estimates $\hat{W}_i(k)$, $i = 1, \ldots, n$ are all bounded, subsequently

$$\lim_{k\to\infty} \|y(k) - y_d(k)\| \le \epsilon$$

where ϵ is a small positive number.

Proof: The proof procedure is as follows.

1) In the first step, for the first subsystem, by choosing NN controller $u_1(k)$, its stability is guaranteed by using Lyapunov analysis.
2) In the second step, once $u_1(k)$ is determined, by choosing $u_2(k)$, we prove the SGUUB stability for the first two subsystems Σ_1 and Σ_2.
3) Repeating this procedure recursively, in step i, choose $u_i(k)$ to stabilize subsystems Σ_1 to Σ_i.
4) Finally, in step n, choose $u_n(k)$ to guarantee the stability of the whole system.

Suppose that $Y(k-\tau), Y(k-\tau+1), \ldots, Y(k-1) \in \Omega, \forall k \ge 0$ and Ω denotes the compact set in which NN approximation (42), (47), (51) are valid. Now, we prove that $Y(k) \in \Omega$ and $u(k)$ is bounded by backstepping.

Step 1: Noting that $e_1(k) = y_1(k) - y_{d_1}(k)$, its τth difference is given by

$$\begin{aligned} e_1(k+\tau) &= y_1(k+\tau) - y_{d_1}(k+\tau) \\ &= f_1(k) + g_1(k) u_1(k) - y_{d_1}(k+\tau). \end{aligned} \tag{58}$$

Adding and subtracting $g_1(k) u_1^*(k)$ on the right-hand side of (58) and noting (41), we have

$$\begin{aligned} e_1(k+\tau) &= g_1(k)\left(u_1(k) - u_1^*(k)\right) \\ &= g_1(k)\left[\tilde{W}_1^T(k) S_1\left(z_1(k)\right) - \epsilon_{z_1}\right] \end{aligned}$$

with $\tilde{W}_1^T(k) = \hat{W}_1^T(k) - W_1^*(k)$ denotes the estimation error of the NN weight. Consequently, we obtain

$$\tilde{W}_1^T(k) S_1\left(z_1(k)\right) = \frac{e_1(k+\tau)}{g_1(k)} + \epsilon_{z_1}. \tag{59}$$

Choose the following Lyapunov function candidate:

$$V_1(k) = \frac{1}{\bar{g}_1} \sum_{j=0}^{\tau-1} e_1^2(k+j) + \sum_{j=0}^{\tau-1} \tilde{W}_1^T(k+j) \Gamma_1^{-1} \tilde{W}_1(k+j). \tag{60}$$

Its first difference is

$$\begin{aligned} \Delta V_1(k) &= V_1(k+1) - V_1(k) \\ &= \frac{1}{\bar{g}_1} e_1^2(k+\tau) - \frac{1}{\bar{g}_1} e_1^2(k) + \tilde{W}_1^T(k+\tau) \\ &\quad \times \Gamma_1^{-1} \tilde{W}_1(k+\tau) - \tilde{W}_1^T(k) \Gamma_1^{-1} \tilde{W}_1(k). \end{aligned} \tag{61}$$

Noting weight update algorithm (57) and (59), we have

$$\begin{aligned} \Delta V_1(k) &= \frac{1}{\bar{g}_1} e_1^2(k+\tau) - \frac{1}{\bar{g}_1} e_1^2(k) - 2\tilde{W}_1^T(k) S_1\left(z_1(k)\right) \\ &\quad \times e_1(k+\tau) - 2\sigma_1 \tilde{W}_1^T(k) \hat{W}_1(k) + S_1^T\left(z_1(k)\right) \\ &\quad \times \Gamma_1^T S_1\left(z_1(k)\right) e_1^2(k+\tau) + 2\sigma_1 \hat{W}_1^T(k) \\ &\quad \times \Gamma_1^T S_1\left(z_1(k)\right) e_1(k+\tau) + \sigma_1^2 \hat{W}_1^T(k) \Gamma_1^T \hat{W}_1(k) \\ &= \frac{1}{\bar{g}_1} e_1^2(k+\tau) - \frac{1}{\bar{g}_1} e_1^2(k) - 2\frac{1}{g_1(k)} e_1^2(k+\tau) \\ &\quad - 2\epsilon_{z_1} e_1(k+\tau) - 2\sigma_1 \tilde{W}_1^T(k) \hat{W}_1(k) \\ &\quad + S_1^T\left(z_1(k)\right) \Gamma_1^T S_1\left(z_1(k)\right) e_1^2(k+\tau) \\ &\quad + 2\sigma_1 \hat{W}_1^T(k) \Gamma_1^T S_1\left(z_1(k)\right) e_1(k+\tau) \\ &\quad + \sigma_1^2 \hat{W}_1^T(k) \Gamma_1^T \hat{W}_1(k) \\ &\le -\frac{1}{\bar{g}_1} e_1^2(k+\tau) - \frac{1}{\bar{g}_1} e_1^2(k) - 2\epsilon_{z_1} e_1(k+\tau) \\ &\quad - 2\sigma_1 \tilde{W}_1^T(k) \hat{W}_1(k) + S_1^T\left(z_1(k)\right) \Gamma_1^T S_1\left(z_1(k)\right) \\ &\quad \times e_1^2(k+\tau) + 2\sigma_1 \hat{W}_1^T(k) \Gamma_1^T S_1\left(z_1(k)\right) e_1(k+\tau) \\ &\quad + \sigma_1^2 \hat{W}_1^T(k) \Gamma_1^T \hat{W}_1(k). \end{aligned}$$

Using the following facts:

$$\begin{aligned} S_1^T\left(z_1(k)\right)S_1\left(z_1(k)\right) &< l_1 \\ S_1^T\left(z_1(k)\right)\Gamma_1^T S_1\left(z_1(k)\right) &\le \bar{\gamma}_1 S_1^T\left(z_1(k)\right)S_1\left(z_1(k)\right) \\ &\le \bar{\gamma}_1 l_1 \\ 2\epsilon_{z_1} e_1(k+\tau) &\le \frac{\bar{\gamma}_1 e_1^2(k+\tau)}{\bar{g}_1} + \frac{\bar{g}_1 \epsilon_{z_1}^2}{\bar{\gamma}_1} \end{aligned}$$

$$\begin{aligned} &2\tilde{W}_1^T(k)\hat{W}_1(k) \\ &\quad = \left\|\tilde{W}_1(k)\right\|^2 + \left\|\hat{W}_1(k)\right\|^2 - \|W_1^*\|^2 \\ &2\sigma_1\hat{W}_1^T(k)\Gamma_1 S_1\left(z_1(k)\right)e_1(k+\tau) \\ &\quad \le \frac{\bar{\gamma}_1 l_1 e_1^2(k+\tau)}{\bar{g}_1} + \bar{g}_1\sigma_1^2\bar{\gamma}_1\|\hat{W}_1\|^2 \end{aligned}$$

where $\bar{\gamma}_1 = \max\{\gamma_{1_1}, \gamma_{1_2}, \ldots, \gamma_{1_n}\}$ denotes the biggest eigenvalue of Γ_1 and l_1 denotes the neurons used, we obtain

$$\begin{aligned} \Delta V_1 \le & -\frac{\rho_1}{\bar{g}_1}e_1^2(k+\tau) - \frac{1}{\bar{g}_1}e_1^2(k) + \beta_1 \\ & -\sigma_1(1-\sigma_1\bar{\gamma}_1 - \bar{g}_1\sigma_1\bar{\gamma}_1)\left\|\hat{W}_1(k)\right\|^2 - \sigma_1\left\|\tilde{W}_1(k)\right\|^2 \end{aligned}$$

where $\rho_1 = 1 - \bar{\gamma}_1 - \bar{\gamma}_1 l_1 - \bar{g}_1\bar{\gamma}_1 l_1$ and $\beta_1 = (\bar{g}_1\epsilon_{z_1}^2/\bar{\gamma}_1) + \sigma_1\|W_1^*\|^2$. If we choose the design parameters as follows:

$$\bar{\gamma}_1 < \frac{1}{1 + l_1 + \bar{g}_1 l_1}, \quad \sigma_1 < \frac{1}{(1+\bar{g}_1)\bar{\gamma}_1} \tag{62}$$

then we have

$$\Delta V_1 \le -\frac{1}{\bar{g}_1}e_1^2(k) + \beta_1.$$

Then $\Delta V_1 \le 0$ once the error $|e_1(k)|$ is larger than $\sqrt{\bar{g}_1\beta_1}$. This implies the boundedness of $e_1(k)$, and we know that the tracking error $e_1(k)$ will be bounded in a compact set.

Subtracting W_1^* to both sides of (44), we can obtain

$$\tilde{W}_1(k) = \tilde{W}_1(k-\tau) - \Gamma_1\left[S_1\left(z_1(k-\tau)\right)e_1(k) + \sigma_1\hat{W}_1(k-\tau)\right].$$

It can be rewritten as

$$\tilde{W}_1(k+\tau) = \tilde{W}_1(k) - \Gamma_1\left[S_1\left(z_1(k)\right)e_1(k+\tau) + \sigma_1\hat{W}_1(k)\right].$$

Noting that $W_1^* = \hat{W}_1(k) - \tilde{W}_1(k) = \hat{W}_1(k+\tau) - \tilde{W}_1(k+\tau)$, the estimation error (44) can be written as

$$\begin{aligned} \tilde{W}_1(k+\tau) &= (I - \Gamma_1\sigma_1)\tilde{W}_1(k) \\ &\quad - \Gamma_1\left[S_1\left(z_1(k)\right)e_1(k+\tau) + \sigma_1 W_1^*\right] \\ &= A_1(k)\tilde{W}_1(k) \\ &\quad - \Gamma_1\left[S_1\left(z_1(k)\right)e_1(k+\tau) + \sigma_1 W_1^*\right]. \end{aligned}$$

Because $0 < \gamma_{1_1}, \gamma_{1_2}, \ldots, \gamma_{1_n} < 1$ and $0 < \sigma_1 < 1$, we know that the transition matrix $\|\Phi(k_1, k_0)\|$ of $A_1(k)$ always satisfies $\|\Phi(k_1, k_0)\| < 1$. Furthermore, noting $S_1(z_1(k))$, $e_1(k+\tau)$ and $\sigma_1 W_1^*$ are all bounded, by applying Lemma 1, $\tilde{W}_1(k)$ is bounded in a compact set denoted by Ω_{w_1} and, hence, the boundedness of $\hat{W}_1(k)$ is assured.

Step $i(1 < i < n)$: Following the same procedures in Step 1 or Step 2, for $e_i(k) = y_i(k) - y_{d_i}(k)$, its τth difference is given by

$$\begin{aligned} e_i(k+\tau) &= y_i(k+\tau) - y_{d_i}(k+\tau) \\ &= f_i\left(k, \bar{u}_{i-1}(k)\right) + g_i(k)u_i(k) - y_{d_i}(k+\tau). \end{aligned} \tag{63}$$

Adding and subtracting $g_i(k)u_i^*(k)$ on the right-hand side of (63) and noting (46), we have

$$\begin{aligned} e_i(k+\tau) &= g_i(k)\left(u_i(k) - u_i^*(k)\right) \\ &= g_i(k)\left[\tilde{W}_i^T(k)S_i\left(z_i(k)\right) - \epsilon_{z_i}\right] \end{aligned}$$

with $\tilde{W}_i^T(k) = \hat{W}_i^T(k) - W_i^*(k)$ denotes the estimation error of the NN weight $W_i^*(k)$. Consequently, we obtain

$$\tilde{W}_i^T(k)S_i\left(z_i(k)\right) = \frac{e_i(k+\tau)}{g_i(k)} + \epsilon_{z_i}. \tag{64}$$

Similarly, choosing the following Lyapunov function candidate for subsystems Σ_1 to Σ_i

$$\begin{aligned} V_i(k) = \sum_{j=1}^{i-1} V_j(k) + \frac{1}{\bar{g}_i}\sum_{j=0}^{\tau-1} e_i^2(k+j) \\ + \sum_{j=0}^{\tau-1}\tilde{W}_i^T(k+j)\Gamma_i^{-1}\tilde{W}_i(k+j). \end{aligned} \tag{65}$$

By following the same procedure as in Step 1, we have

$$\begin{aligned} \Delta V_i(k) \le \sum_{j=1}^{i-1}\Delta V_j(k) - \frac{\rho_i}{\bar{g}_i}e_i^2(k+\tau) - \frac{1}{\bar{g}_i}e_i^2(k) + \beta_i \\ -\sigma_i(1 - \sigma_i\bar{\gamma}_i - \bar{g}_i\sigma_i\bar{\gamma}_i)\left\|\hat{W}_i(k)\right\|^2 - \sigma_i\left\|\tilde{W}_i(k)\right\|^2 \end{aligned}$$

where $\rho_i = 1 - \bar{\gamma}_i - \bar{\gamma}_i l_i - \bar{g}_i\bar{\gamma}_i l_i$ and $\beta_i = (\bar{g}_i\epsilon_{z_i}^2/\bar{\gamma}_i) + \sigma_i\|W_i^*\|^2$. If we choose the design parameters as follows:

$$\bar{\gamma}_i < \frac{1}{1 + l_i + \bar{g}_i l_i}, \quad \sigma_i < \frac{1}{(1+\bar{g}_i)\bar{\gamma}_i} \tag{66}$$

then we have

$$\Delta V_i(k) \le \sum_{j=1}^{i}\left\{-\frac{1}{\bar{g}_j}e_j^2(k)\right\} + \sum_{j=1}^{i}\beta_j.$$

Thus, $\Delta V_i(k) \le 0$ once the error $|e_j(k)|$ $(j = 1, 2, \ldots, i)$ is larger than $\sqrt{\bar{g}_j(\beta_1 + \cdots + \beta_i)}$. This implies the boundedness of $e_j(k)$ $(j = 1, 2, \ldots, i)$. Furthermore, the tracking error $e_j(k)$ $(j = 1, 2, \ldots, i)$ will bounded in a compact set.

By following the similar procedure as in Step 1, we know that $\tilde{W}_i(k)$ is bounded in a compact set denoted by Ω_{w_i} and, hence, the boundedness of $\hat{W}_i(k)$ is assured.

Step n: In the final step, following the same procedure as in Step i, we have the following Lyapunov function candidate (for clarity of presentation, details are omitted here)

$$V_n(k) = \sum_{j=1}^{n-1} V_j(k) + \frac{1}{\bar{g}_n}\sum_{j=0}^{\tau-1} e_n^2(k+j) + \sum_{j=0}^{\tau-1} \tilde{W}_n^T(k+j)\Gamma_n^{-1}\tilde{W}_n(k+j). \quad (67)$$

Its first difference is

$$\Delta V_n(k) \le \sum_{j=1}^{n-1} \Delta V_j(k) - \frac{\rho_n}{\bar{g}_n} e_n^2(k+\tau) - \frac{1}{\bar{g}_n} e_n^2(k) + \beta_n - \sigma_n(1 - \sigma_n\bar{\gamma}_n - \bar{g}_n\sigma_n\bar{\gamma}_n)\left\|\hat{W}_n(k)\right\|^2 - \sigma_n\left\|\tilde{W}_n(k)\right\|^2$$

with $\rho_n = 1 - \bar{\gamma}_n - \bar{\gamma}_n l_n - \bar{g}_n\bar{\gamma}_n l_n$ and $\beta_n = (\bar{g}_n\epsilon_{z_n}^2/\bar{\gamma}_n) + \sigma_n\|W_n^*\|^2$. If we choose the design parameters as follows:

$$\bar{\gamma}_n < \frac{1}{1 + l_n + \bar{g}_n l_n}, \quad \sigma_n < \frac{1}{(1+\bar{g}_n)\bar{\gamma}_n} \quad (68)$$

then we have

$$\Delta V_n(k) \le \sum_{j=1}^{n}\left\{-\frac{1}{\bar{g}_j}e_j^2(k)\right\} + \sum_{j=1}^{n}\beta_j.$$

Thus, $\Delta V_n(k) \le 0$ once the error $e_j(k)$ $(j = 1, 2, \ldots, n)$ is larger than $\sqrt{\bar{g}_j(\beta_1 + \cdots + \beta_n)}$. This implies the boundedness of $e_j(k)$ $(j = 1, 2, \ldots, n)$. Furthermore, the tracking error $e_j(k)$ $(j = 1, 2, \ldots, n)$ will bounded in a compact set.

Following the procedures in previous steps, we know that $\tilde{W}_n(k)$ is bounded in a compact set denoted by Ω_{w_n} and, hence, the boundedness of $\hat{W}_n(k)$ is assured.

In summary, for the closed-loop nonlinear MIMO system consists of system (1), controller (54) and adaptive law (55), if the design parameters are chosen as

$$\bar{\gamma}_i < \frac{1}{(1 + l_i + \bar{g}_i l_i)}, \quad \sigma_i < \frac{1}{(1+\bar{g}_i)\bar{\gamma}_i} \quad (i = 1, \ldots, n)$$

then the closed-loop system is SGUUB and has an equilibrium at $[e_1(k), \ldots, e_n(k)]^T = 0$. This guarantees that all the signals include the state vector $X(k)$, the control input $u(k)$ and NN weight estimates $\hat{W}_i(k)$, $i = 1, \ldots, n$ are all bounded. Subsequently

$$\lim_{k\to\infty}\|y(k) - y_d(k)\| \le \epsilon$$

where ϵ is a small positive number. ■

VI. Simulation

Consider the following MIMO discrete-time system with triangular form inputs:

$$\begin{cases} x_{1,1}(k+1) = f_{1,1}\left(\bar{x}_{1,1}(k)\right) + g_{1,1}\left(\bar{x}_{1,1}(k)\right)x_{1,2}(k) \\ x_{1,2}(k+1) = f_{1,2}\left(\bar{x}_{1,2}(k)\right) + g_{1,2}\left(\bar{x}_{1,2}(k)\right)u_1(k) \\ x_{2,1}(k+1) = f_{2,1}\left(\bar{x}_{2,1}(k)\right) + g_{2,1}\left(\bar{x}_{2,1}(k)\right)x_{2,2}(k) \\ x_{2,2}(k+1) = f_{2,2}\left(\bar{x}_{2,2}(k), u_1(k)\right) + g_{2,2}\left(\bar{x}_{2,2}(k)\right)u_2(k) \\ \quad y_1(k) = x_{1,1}(k) \\ \quad y_2(k) = x_{2,1}(k) \end{cases}$$

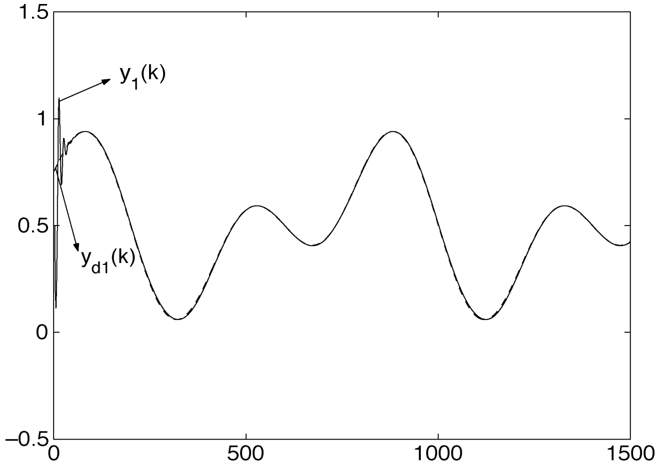

Fig. 2. Tracking performance $y_1(k)$ and $y_{d_1}(k)$.

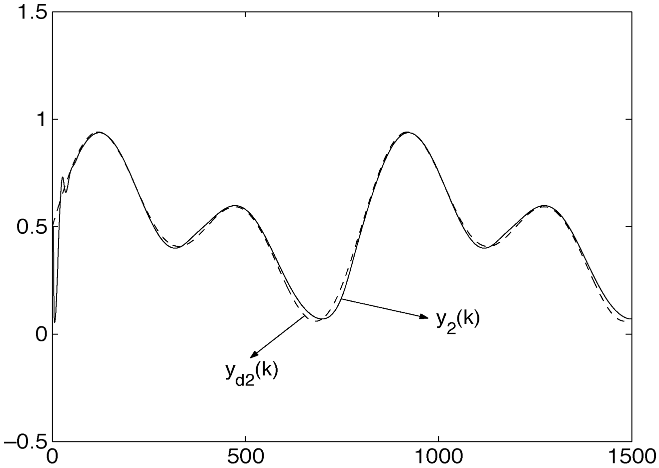

Fig. 3. Tracking performance $y_2(k)$ and $y_{d_2}(k)$.

where $f_{1,1}(\bar{x}_{1,1}(k)) = x_{1,1}^2(k)/[1 + x_{1,1}^2(k)]$, $g_{1,1}(\bar{x}_{1,1}(k)) = 0.3$, $f_{1,2}(\bar{x}_{1,2}(k)) = x_{1,1}^2(k)/[1 + x_{1,2}^2(k) + x_{2,1}^2(k) + x_{2,2}^2(k)]$, $g_{1,2}(\bar{x}_{1,2}(k)) = 1$, $f_{2,1}(\bar{x}_{2,1}(k)) = x_{2,1}^2(k)/[1 + x_{2,1}^2(k)]$, $g_{2,1}(\bar{x}_{2,1}(k)) = 0.2$, $f_{2,2}(\bar{x}_{2,2}(k), u_1(k)) = u_1^2(k)x_{2,1}^2(k)/[1+x_{1,1}^2+x_{1,2}^2(k)+x_{2,2}^2(k)]$ and $g_{2,2}(\bar{x}_{2,2}(k)) = 1$.

The control objective is to drive the output $y(k) = [y_1(k), y_2(k)]^T$ of the system to follow desired reference signals

$$y_{d_1}(k) = 0.5 + \frac{1}{4}\cos\left(\frac{\pi Tk}{4}\right) + \frac{1}{4}\sin\left(\frac{\pi Tk}{2}\right)$$
$$y_{d_2}(k) = 0.5 + \frac{1}{4}\sin\left(\frac{\pi Tk}{4}\right) + \frac{1}{4}\sin\left(\frac{\pi Tk}{2}\right)$$

with $T = 0.01$.

The initial condition for system states is $x_{1,1}(0) = x_{1,1}(1) = 0.5$, $x_{1,2}(0) = x_{1,2}(1) = 0$, $x_{2,1}(0) = x_{2,1}(1) = 0.5$, and $x_{2,2}(0) = x_{2,2}(1) = 0$. The neurons used are $l_1 = 28$ and $l_2 = 36$. All the elements of the NN weights $\hat{W}_1(0)$, $\hat{W}_1(1)$, $\hat{W}_2(0)$ and $\hat{W}_2(1)$ are initialized to be random numbers between 0.00 and 0.01, and the active functions $S_1(z_1(0))$, $S_1(z_1(1))$, $S_2(z_2(0))$, and $S_2(z_2(1))$ are initialized to be random numbers between 0.00 and 0.02. The σ modification gains are $\sigma_1 = \sigma_2 = 0.01$, and adaptive gain matrices are $\Gamma_1 = \Gamma_2 = 0.015\mathrm{I}$.

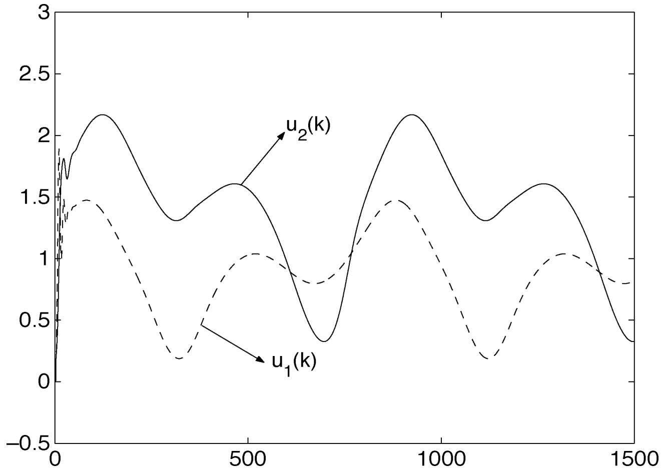

Fig. 4. Control inputs $u_1(k)$ and $u_2(k)$.

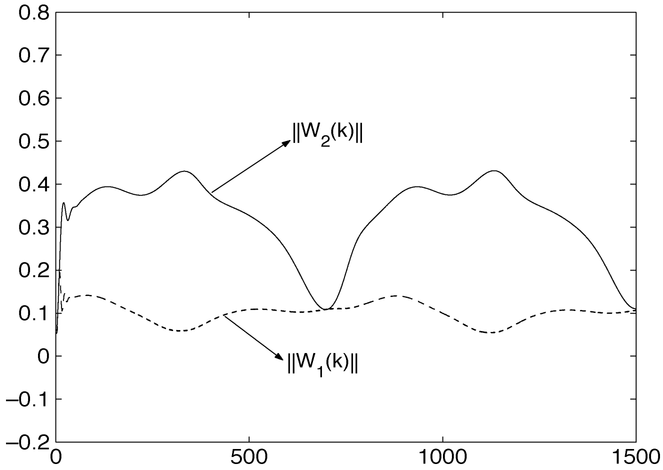

Fig. 5. Control inputs weights norm $\|\hat{W}_1(k)\|$ and $\|\hat{W}_2(k)\|$.

Simulation results are shown in Figs. 2–5. Figs. 2 and 3 show the tracking performances of the first subsystem and the second subsystem, respectively. It can be seen that, in the initial period of simulation, the tracking errors are large. Then, as the time increases, the practical outputs converge to the neighborhoods of the desired signals. The control input trajectories $u_1(k) = \hat{W}_1(k)S_1(z_1(k))$ and $u_2(k) = \hat{W}_2(k)S_2(z_2(k))$ are shown in Fig. 4. Their corresponding NN weights norms $\|\hat{W}_1(k)\|$ and $\|\hat{W}_2(k)\|$ are shown in Fig. 5. From Figs. 4 and 5, we can see that both the control inputs and their corresponding weights norms are all bounded.

From Fig. 5, it can be seen that there is no convergence of the weights actually, which corresponds to our analysis in the proof of closed-loop system stability, that the NN weights are only guaranteed to be bounded.

VII. Conclusion

In this paper, adaptive NN control has been developed for a class of discrete-time nonlinear MIMO systems. Through coordinate transformation, the system was first transformed into input–output description. Then the input and output sequences were used to construct the effective NN control. HONNs were used to approximate the desired controls. The closed-loop system was proved to be SGUUB based on Lyapunov analysis.

References

[1] A. U. Levin and K. S. Narendra, "Control of nonlinear dynamical systems using neural networks-part ii: observability, identification, and control," *IEEE Trans. Neural Netw.*, vol. 7, no. 1, pp. 30–42, Jan. 1996.

[2] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamic systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, Jan. 1990.

[3] F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Netw.*, vol. 7, no. 2, pp. 388–398, Mar. 1996.

[4] M. M. Polycarpou, "Stable adaptive neural control scheme for nonlinear systems," *IEEE Trans. Autom. Control*, vol. 41, no. 3, pp. 447–451, Mar. 1996.

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

[6] S. S. Ge, C. C. Hang, T. H. Lee, and T. Zhang, *Stable Adaptive Neural Network Control*. Norwell, MA: Kluwer, 2001.

[7] J. B. D. Cabrera and K. S. Narendra, "Issues in the application of neural networks for tracking based on inverse control," *IEEE Trans. Autom. Control*, vol. 44, no. 11, pp. 2007–2027, Nov. 1999.

[8] O. Adetona, E. Garcia, and L. H. Keel, "A new method for the control of discrete nonlinear dynamic systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 11, no. 1, pp. 102–112, Jan. 2000.

[9] J. X. Zhao and I. Kanellakopoulos, "Active identification for discrete-time nonlinear control-part ii: strict-feedback systems," *IEEE Trans. Autom. Control*, vol. 47, no. 2, pp. 225–240, Feb. 2002.

[10] S. S. Ge, G. Y. Li, and T. H. Lee, "Adaptive NN control for a class of strict-feedback discrete-time nonlinear systems," *Automatica*, vol. 39, no. 5, pp. 807–819, May 2003.

[11] F. C. Chen and H. K. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks," *IEEE Trans. Autom. Control*, vol. 72, no. 7, pp. 791–807, Jul. 1995.

[12] S. S. Ge, T. H. Lee, G. Y. Li, and J. Zhang, "Adaptive NN control for a class of discrete-time nonlinear systems," *Int. J. Control*, vol. 76, no. 4, pp. 334–354, 2003.

[13] S. Jagannathan and F. L. Lewis, "Discrete-time neural net controller for a class of nonlinear dynamical systems," *IEEE Trans. Autom. Control*, vol. 41, no. 11, pp. 1693–1699, Nov. 1996.

[14] ——, "Multilayer discrete-time neural-net controller with guaranteed performance," *IEEE Trans. Neural Netw.*, vol. 7, no. 1, pp. 107–130, Jan. 1996.

[15] S. S. Ge, G. Y. Li, J. Zhang, and T. H. Lee, "Direct adaptive control for a class of MIMO nonlinear systems using neural networks," *IEEE Trans. Autom. Control*, vol. 49, no. 11, pp. 2001–2006, Nov. 2004.

[16] F. Sun, Z. Sun, and P. Y. Woo, "Stable neural-network-based adaptive control for sampled-data nonlinear systems," *IEEE Trans. Neural Netw.*, vol. 9, no. 5, pp. 956–968, Sep. 1998.

[17] S. S. Ge, J. Zhang, and T. H. Lee, "State feedback neural network control of a class of discrete MIMO nonlinear systems with disturbances," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 4, pp. 1630–1645, Aug. 2004.

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

[19] S. Chen and S. A. Billings, "Representations of nonlinear systems: the narmax model," *Int. J. Control*, vol. 49, no. 3, pp. 1013–1032, 1989.

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

[21] J. T. Spooner and K. M. Passino, "Stable adaptive control using fuzzy systems and neural networks," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 3, pp. 339–359, 1996.

[22] G. P. Liu, V. Kadirkamanathan, and S. A. Billings, "Nonlinear system identification using wavelet networks," *Int. J. Syst. Sci.*, vol. 12, pp. 1531–1541, 2000.

[23] R. Marino and P. Tomei, *Nonlinear Control Design*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[24] H. K. Khalil, *Nonlinear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[25] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

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

Jin Zhang was born in Xi'an, Shaanxi Province, China, in 1974. He received the B.Eng. degree from the Department of Automatic Control, Beijing University of Aeronautics and Astronautics (BUAA), China, in 1997 and the Ph.D. degree from the Department of Electrical and Computer Engineering, National University of Singapore, in 2004.

He is currently a Postdoctoral Fellow in the School of Automation, Beijing University of Aeronautics and Astronautics. His research interests include adaptive nonlinear control, neural network control, system simulation, and virtual reality technology.

Shuzhi Sam Ge (S'90–M'92–SM'00) received the B.Sc. degree from Beijing University of Aeronautics and Astronautics (BUAA), Beijing, China, in 1986, and the Ph.D. degree and the Diploma of Imperial College (DIC) from Imperial College of Science, Technology and Medicine, University of London, London, U.K., in 1993.

He has been with the Department of Electrical and Computer Engineering, National University of Singapore, since 1993, where he is currently a Full Professor. He has authored and coauthored over 200 international journal and conference papers, three monographs and co-invented three patents. He serves as a technical consultant local industry. His curent research interests are control of nonlinear systems, neural/fuzzy systems, robotics, hybrid systems, sensor fusion, and system development.

Dr. Ge has served as Editor of the *International Journal of Control, Automation, and Systems* since 2003, as a chair of the Technical Committee on Intelligent Control since 2005, and as a member of Board of Governors of the IEEE Control Systems Society since 2004. He served as an Associate Editor for the Conference Editorial Board of the IEEE Control Systems Society in 1998 and 1999. He has been serving as Associate Editor of the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY since 1999, Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, since 2004, Corresponding Editor for Asia and Australia of the *IEEE Control Systems Magazine* since 2004, and Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS, since 2004. He received the 1999 National Technology Award, the 2001 University Young Research Award, and the 2002 Temasek Young Investigator Award (Singapore).

Tong Heng Lee (M'90) received the B.A. degree (with first class honors) in engineering tripos from Cambridge University, Cambridge, U.K., in 1980 and the Ph.D. degree from Yale University, New Haven, CT, in 1987.

He is a Professor in the Department of Electrical and Computer Engineering, National University of Singapore. He is also currently Head of the Drives, Power and Control Systems Group in this Department. His research interests are in the areas of adaptive systems, knowledge-based control, intelligent mechatronics, and computational intelligence. He has also coauthored three research monographs, and holds four patents (two of which are in the technology area of adaptive systems, and the other two are in the area of intelligent mechatronics). He is an Associate Editor of *Control Engineering Practice*, the *International Journal of Systems Science*, and the *Mechatronics Journal*

Dr. Lee received the Cambridge University Charles Baker Prize in Engineering. He is an Associate Editor of the IEEE TRANSACTIONS IN SYSTEMS, MAN AND CYBERNETICS.