Intelligent Automation And Soft Computing, 2019
Copyright © 2019, TSI® Press
Vol. 25, no. 3, 487–507
https://doi.org/10.31209/2018.100000037

MTN Optimal Control of SISO Nonlinear Time-varying Discrete-time Systems for Tracking by Output Feedback*

Hong-Sen Yan[1,2], Jiao-Jun Zhang[1,2] and Qi-Ming Sun[1,2]

[1]Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing 210096, China
[2]School of Automation, Southeast University, Nanjing 210096, China

* Both the first and second authors contributed equally to this work, and the third author has finished the original program implementation of MTN optimal control of the SISO non-linear constant systems for tracking.

ABSTRACT
MTN optimal control scheme of SISO nonlinear time-varying discrete-time systems based on multi-dimensional Taylor network (MTN) is proposed to achieve the real-time output tracking control for a given reference signal. Firstly, an ideal output signal is selected and Pontryagin minimum principle adopted to obtain the numerical solution of the optimal control law for the system relative to the ideal output signal, with the corresponding optimal output termed as desired output signal. Then, MTN optimal controller (MTNC) is generated automatically to fit the optimal control law, and the conjugate gradient (CG) method is employed to train the weight parameters of MTNC offline to acquire the initial weight parameters of MTNC for online training that guarantees the stability of closed-loop system. Finally, a four-term back propagation (BP) algorithm with a second order momentum term and error term is proposed to adjust the weight parameters of MTNC adaptively to implement the output tracking control of the systems in real time; the convergence conditions for the four-term BP algorithm are determined and proved. Simulation results show that the proposed MTN optimal control scheme is valid; the system's actual output response is capable of tracking the given reference signal in real time.

KEY WORDS: MTN optimal control, Discrete-time system, Multi-dimensional Taylor network, Nonlinear time-varying system, Output tracking, Real-time control

1 INTRODUCTION

MOST practical systems are of nonlinear and time-varying nature, and the output tracking control for them has found wide application in engineering fields. However, little work has been focused directly on the nonlinear time-varying discrete-time system described by the input-output model covering a wide class of non-linear systems. Therefore, researches on this topic are of great significance.

In recent years, control design of nonlinear time-varying systems has attracted much attention, and various control methods have also been proposed and practiced (Chen et al., 2010; Xu and Yan, 2004; Zhu et al., 2014). PID control strategy is the most commonly used method for industrial process control. Over 90% of industrial controllers implemented are still based on PID (Ang et al., 2005; Åström and Hägglund, 2001; Hušek, 2014; Papadopoulos et al., 2013), mainly due to its advantages such as sensitive intuition, simple implementation, good robustness and so on. However, as the adjustment of controller parameters is done by trial and error method (Tseng, 2001; Ziegler and Nichols, 1942), the controller has no desirable self-adaptive ability to the plants with complex nonlinear time-varying characteristics. Some scholars have been committed to the improvement of PID control (Cameron and Seborg, 1983; Fu and Chai, 2012; Rad et al., 1997), hoping to optimize the PID controller parameters online. Meanwhile, several intelligent algorithms (Nagarag and Vijayakumar, 2012; Zhang and Yang, 2016) have been developed

CONTACT Hong-Sen Yan hsyan@seu.edu.cn.

for PID control to enhance its adaptive ability. State feedback, which takes system states as feedback variables, is one of the popular means of control system design (Theodoridis et al., 2012), whereby the system states are required to be completely measurable. However, with the limitations of measuring equipment, nonlinear system states are hard to be measured directly and different results may be produced from the true and estimated states of the system in the process of control. Therefore, in engineering applications, output feedback control is necessary. Theoretically, neural network can be used for any nonlinear system for its strong nonlinear approximation ability and has been adopted in the design of adaptive controller for nonlinear systems with good control performance (Narendra and Parthasarathy, 1990; Zargarzadeh et al., 2014). However, its structure is mainly chosen by means of the trial and error method, its connection weights update needs dynamic learning algorithm in the process of control, and a complex network structure may lead it into complex computation, difficult parameters real-time update process and slow convergence speed in network training, all of which limit its practical application. There also turn up a variety of neural networks and learning algorithms applied to practical problems (Jin et al., 1995; Lu et al., 2007; Mon et al., 2008; Sanner and Slotine, 1992; Seshagiri and Khalil, 2000; Shu, 1999; Shu and Pi, 2000; Tseng, 2001; Wang et al., 2009). However, most of the literatures concerned just focus on the adaptive control problems without considering the initial values of the weight parameters for neural networks (Bezzaoucha et al., 2015; Shojaei, 2015). As divergence, oscillation or even instability may be caused by the time-delayed signals upon implementing artificial NNs in real application, the research on the stability problems of time-delayed NNs is in dire need (Saravanakumar and Ali, 2016; Saravanakumar et al., 2017a, 2017b).

The existing methods for tracking control of time-varying nonlinear systems mostly focus on the systems with special structures or the control based on neural networks which are unsuitable for real-time control due to their computation complexity. It is thus necessary to explore a new approach to the tracking control of time-varying nonlinear systems in real time. For this end, the MTN optimal control scheme of SISO nonlinear time-varying discrete-time systems based on multi-dimensional Taylor network (MTN, whose idea was proposed by Hong-Sen Yan in 2010 and its realization was completed by Bo Zhou, Yan's PhD student) is proposed here to achieve the optimal real-time output tracking control of the systems for the given reference signal. MTN, good at representing or approximating the general nonlinear dynamic system, reflects the dynamic characteristics of the system more directly with no need to know the order or other prior information of the system, and proves to be capable of approximating any nonlinear function by learning the weight coefficients in a bounded closed region (Lin, 2015). It has been applied to the stock and nonlinear time series prediction problems successfully (Lin, 2015; Lin et al., 2014a, 2014b, 2014c; Zhou, 2014; Zhou and Yan, 2013a, 2013b, 2014a, 2014b), more suitable to dynamic system control problem than feed-forward neural networks. However, the MTN mentioned in each of them is void of the control input item. For that sake, the idea of MTN optimal control was proposed by Yan (2010; 2019), and the control input item was introduced into the MTN for the optimal adjustment control of SISO nonlinear time-invariant systems with satisfactory control effects achieved (Sun and Yan, 2014). Asymptotic tracking and dynamic regulation of SISO nonlinear system based on discrete multi-dimensional Taylor network is considered in (Yan and Kang, 2017). Moreover, the studies and simulations of the simple MTNC (i.e., PID plus the sum of their second order monomials plus P.I.D, each item of which is multiplied by its corresponding parameter) for the cruise missile flight trajectory control (Zhang, 2015), the tank firing control in high speed motion (Jin, 2016), the ship roll stabilization (Yang, 2016), the flight of the plane (Zhou, 2016) and the axisymmetric cruise missile flight for attacking static targets (Xia, 2016) have been completed by the graduate students supervised by the first author. All of them are MIMO non-linear constant systems with strong disturbance. The simulation results show that far better dynamic performance, stronger anti-disturbance capability and larger region of attraction have been obtained by the simple MTNC than by PID (Jin, 2016; Xia, 2016; Yang, 2016; Zhang, 2015; Zhou, 2016), PID neural network (Xia, 2016; Zhou, 2016), neural network (Zhou, 2016), sliding mode control (Yang, 2016; Zhang, 2015; Zhou, 2016), and active disturbance rejection controller (Xia, 2016). However, the plants considered in the studies mentioned above are time-invariant. MTN contains polynomial (Yan, 2014). In fact, a kind of polynomial network has been applied for identification and control of nonlinear system (Patrikar and Provence, 1996). However, Patrikar and Provence just considered the constant system, and the algorithm they used for the identifier and controller is based on the gradient descent method which is slow and may converge to local minima. In addition, polynomial network cannot approximate any nonlinear function arbitrarily for the fact that there exist correlations between the high and low order aberration coefficients though fewer weight coefficients need adjusting. The uncertain systems with output feedback control were studied, and with robust stability results obtained in (Wei et al., 2014; Ali and Saravanakumar, 2014; Wei et al., 2017a, 2017b).

Due to the uncertainty (Tseng, 2008) of external environment and time-varying characteristics of the controlled plants, the controller parameters require online self-tuning in the process of control, which affects not merely the convergence speed but the convergence performance of the controller. That is, designing a good real-time self-tuning rule for controller parameters is a key problem. BP algorithm is the most widely used learning algorithm for training multilayer neural network (McClelland et al., 1986), but it has such drawbacks as slow convergence speed, easy falling into local optimal point and so on. Some researchers have been committed to the improvement of BP algorithm (Jacobs, 1988; Minai and Williams, 1990), but their improved BP algorithms are just based on the heuristics, most of them applicable only for specific problems. BP algorithm with a momentum term was analysed (Phansalkar and Sastry, 1994; Qian, 1999) and the necessary and sufficient conditions for the convergence of the algorithm were determined and proved. In (Pearlmutter, 1992), a second-order momentum term was introduced to the BP algorithm with a momentum term that improves the learning speed of the network. An extra term proportional to the error was added to the BP algorithm with a momentum term (Zweiri et al., 2003, 2005). The necessary and sufficient conditions for the convergence of the algorithm is determined and proved. The results show that the third term plays an important role in improving the learning speed of the network.

In this paper, MTN optimal control scheme is proposed for SISO nonlinear time-varying discrete-time systems, by which the output tracking control of the system relative to the given reference signal can be achieved. An ideal output signal is set as the given reference signal. Pontryagin minimum principle is applied to obtain the numerical solution of the optimal control law for the system relative to the ideal output signal, and the resulting optimal output is termed as the desired output signal. MTN optimal controller (MTNC) is generated automatically to fit the optimal control law, CG method is employed to train the weight parameters of MTNC offline as the initial values for online training. In view of the uncertainty of external environment and time-varying characteristics of the systems, a four-term BP algorithm with the second order momentum term and error term is developed to adjust the parameters adaptively for real-time output tracking control of the system, followed by analysis and verification of the convergence condition for the four-term BP algorithm.

The rest of the paper is organized as follows: Section 2 gives the problem statement. The procedure of MTN optimal controller design is presented in Section 3. In Section 4, selection of the initial values of MTN optimal controller parameters for online learning is discussed. Section 5 focuses on a scheme for MTN optimal controller parameters real-time self-tuning. Section 6 lists the algorithmic steps for MTN optimal control scheme. Stability of the four-term BP algorithm is confirmed in Section 7. Section 8 provides a simulation example. And concluding remarks are given in the last section.

2 PROBLEM STATEMENT

CONSIDER the SISO nonlinear time-varying discrete-time system described by the following input-output difference equation:

$$\begin{aligned} y(k+1) = f\,(&y(k), y(k-1), \cdots, y(k-n_y), \\ &u(k), u(k-1), \cdots, u(k-n_u), k), \end{aligned} \tag{1}$$

where $f(\cdot)$ is a nonlinear scalar function, $y(k) \in R$ is the system output, $u(k) \in R$ is the system input, and n_y and n_u are the corresponding maximum delays, $k = 0,1,2,\cdots$.

Our target is to automatically generate a controller which ensures that the output $y(k)$ of the system (1) tracks the given reference signal $r(k)$ in real time.

The control block diagram of the system (1) is shown in Figure 1.

3 AUTOMATIC GENERATION OF MTN OPTIMAL CONTROLLER

MTN optimal controller (MTNC) can be generated automatically as follows:

$$\begin{aligned} u(k) = h(&k_1 u(k-1), \cdots, k_{\bar{n}_u} u(k-\bar{n}_u), \\ &l_1 e(k-1), \cdots, l_{n_e} e(k-n_e), k), \end{aligned} \tag{2}$$

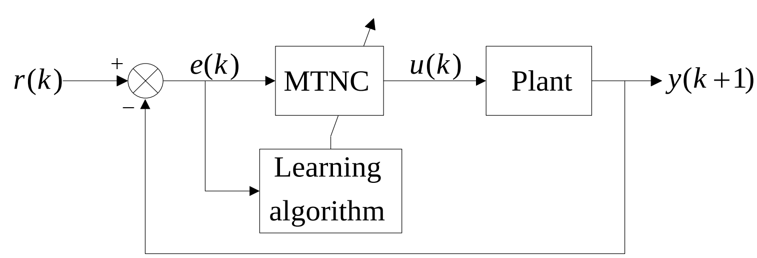

Figure 1. Control block diagram of system (1).

where $h(\cdot)$ is a nonlinear scalar function, $u(k)\in R$ is the output of MTNC, $e(k)\in R$ is the tracking error, i.e., the input of MTNC, $\bar{n}_u$ and n_e are the corresponding maximum delays, and l_i and k_j are positive constants for $i=1,2,\cdots,n_e$, $j=1,2,\cdots,\bar{n}_u$.

For convenience and with no loss of generality, let $t=\bar{n}_u+n_e$, and we have

$$\begin{aligned}\boldsymbol{z}(k)&=[z_1(k),\cdots,z_{\bar{n}_u}(k),z_{\bar{n}_u+1}(k),\cdots,z_t(k)]^{\mathrm{T}}\\&=[k_1u(k-1),\cdots,k_{\bar{n}_u}u(k-\bar{n}_u),\\&\quad l_1e(k-1),\cdots,l_{n_e}e(k-n_e)]^{\mathrm{T}}.\end{aligned}$$

As proved by Zhou and Yan (2013a), there exists a parameter vector $\boldsymbol{w}(k)=[w_1(k),w_2(k),\cdots,w_{N(t,m)}(k)]^{\mathrm{T}}$, whereby the output of MTNC, i.e., the input $u(k)$ of the system (1), can be expressed as

$$u(k)=\sum_{p=1}^{N(t,m)}w_p(k)\prod_{q=1}^{t}z_q^{\lambda(p,q)}(k), \tag{3}$$

where $N(t,m)$ denotes the total number of the product items for the t-ary function $h(\cdot)$ expanded into the approximate polynomial with m powers, $w_p(k)$ is the weight coefficient of the p th product item in formula (3), $\lambda(p,q)$ represents the power of the variable $z_q(k)$ in the p th product item, and $\sum_{q=1}^{t}\lambda(p,q)\le m$.

The structure diagram of MTNC is demonstrated in Figure 2 (Zhou and Yan, 2013a).

To obtain the mathematical expression of $N(t,m)$ and $\lambda(p,q)$, the product items in (3) are rearranged in a new way as shown in Figure 3 (Zhou and Yan, 2013a), i.e., the product items of the expansion are stored according to their powers respectively. Let (i,j) denote the i th rectangle, which is used to store the product items with the j th power obtained by adding one power to the i th variable $z_i(k)$ from the i th to the t th rectangle of the $(j-1)$ th power, so on and so forth, until the product items with the m th power obtained by adding one power to the t th variable $z_t(k)$ in $(t,m-1)$ have been stored in (t,m), where $i=1,2,\cdots,t$ and $j=2,3,\cdots,m$.

Listed below are the calculation process for $N(t,m)$ and $\lambda(p,q)$.

Let $P(i,j)$ $(i=1,2,\cdots,t;\ j=1,\cdots,m)$ denote the number of the product items in (i,j). From Figure 3 we derive

$$N(t,m)=\sum_{j=1}^{m}\sum_{i=1}^{t}P(i,j)+1, \tag{4}$$

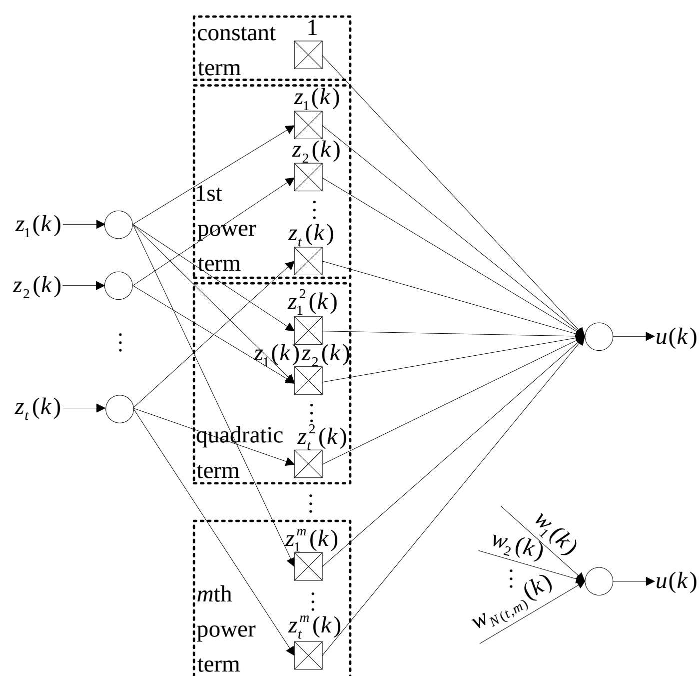

Figure 2. **Structure diagram of MTNC.**

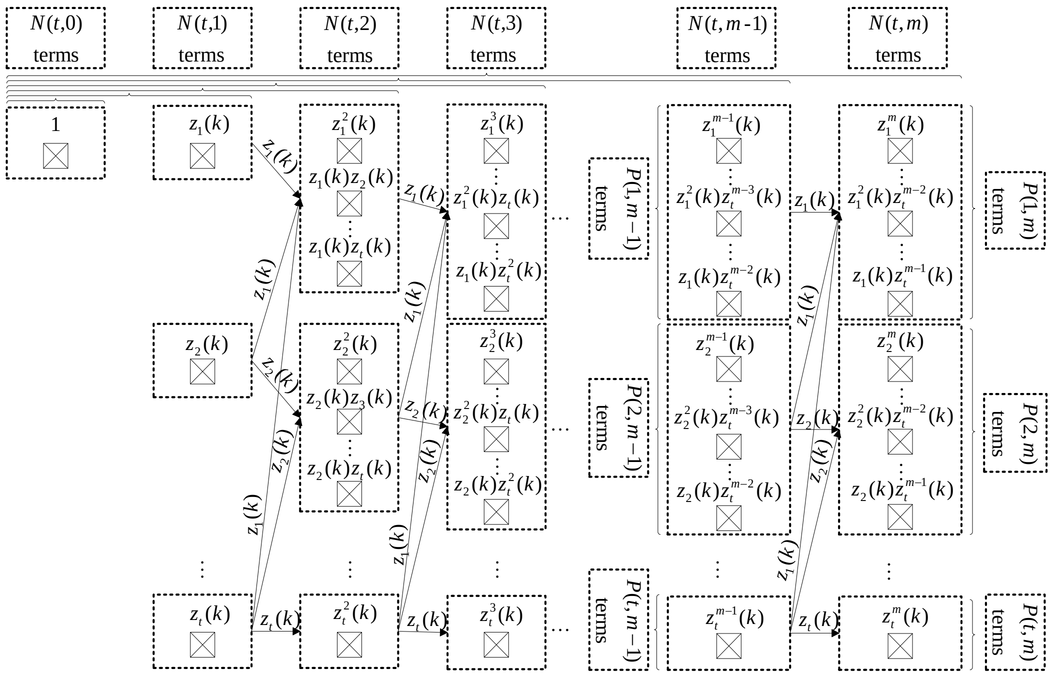

Figure 3. New arrangement of product items.

where

$$P(i,j)=\begin{cases}\sum_{l=i}^{t}P(l,j-1), \ j=2,3,\cdots,m, \\ 1, \ j=1,\end{cases} \quad (i=1,2,\cdots,t). \tag{5}$$

It is supposed that in Equation (3), from the second item, the p th $(p=2,\cdots,N(t,m))$ product item corresponds to the r th product item in (i,j) $(i=1,2,\cdots,t;\ j=1,\cdots,m)$ in Figure 3. The power of the variable $z_q(k)$ $(q=1,2,\cdots,t)$ is denoted by $\lambda_{i,j}(r,q)$. Let $Q_j(b,e)$ $(b,e=1,2,\cdots,t;j=1,2,\cdots,m)$ be the number of the product items from the b th to the e th rectangle with the j th power. From Figure 3, we have

$$\lambda_{i,j}(r,q)=\begin{cases}\lambda_{i+s,j-1}(r-Q_{j-1}(i,i+s-1),q)+1, \ q=i, \\ \lambda_{i+s,j-1}(r-Q_{j-1}(i,i+s-1),q), \quad q\neq i,\end{cases}$$
$$(j=2,3,\cdots,m), \tag{6}$$

$$Q_j(b,e)=\begin{cases}\sum_{i=b}^{e}P(i,j), \ b\le e, \\ 0, \quad b>e,\end{cases} \quad (j=2,3,\cdots,m). \tag{7}$$

The initial values are

$$\lambda_{i,j}(r,q)=\begin{cases}1, \ q=i, \\ 0, \ q\neq i,\end{cases} \quad (j=1),$$

$$Q_j(b,e)=\begin{cases}e-b+1, \quad b\le e, \\ 0, \quad b>e,\end{cases} \quad (j=1),$$

where $Q_{j-1}(i,i+s-1)<r\le Q_{j-1}(i,i+s)$, $s=0,1,2,\cdots,t-i$, and $i=1,2,\cdots,t$.

4 SELECTION OF INITIAL MTNC WEIGHT PARAMETER VALUES FOR ONLINE LEARNING

IT is crucial to choose proper initial values for network parameters that influence convergence speed as well as convergence performance of the network. One common practice in network training is to select the initial values randomly (Rubio and Yu, 2003; Savran, 2007; Yang et al., 2014). In order to raise the convergence speed and avoid falling into a local minimum, PSO algorithm is often adopted to identify the initial weight parameters (Fu and Chai, 2012). In the present study, selection of the initial weight parameters of MTNC for online learning is introduced in the two steps: 1) transform the input-output description form of the system (1) into its extended state space description form by variable substitution, select an ideal output signal $y_{id}(k)$ relative to the given reference signal $r(k)$, and employ Pontryagin minimum principle to obtain the numerical solution of the optimal control law $u^*(k)$ of the system (1) relative to the ideal output signal $y_{id}(k)$; the corresponding optimal output is termed as the desired output signal $y_{op}(k)$; 2) choose a group of weight parameter values randomly at the interval $(-1,1)$, and employ the CG method to train MTNC to approximate the optimal control law $u^*(k)$, and then a group of weight parameter values can be obtained as the initial values used to train MTNC online. The specific practices go as described below.

4.1 Optimal Control Law

Let $\bar{t} = n_y + n_u + 1$, and

$$\begin{aligned} \boldsymbol{x}(k) &= [x_1(k), x_2(k), \cdots, x_{n_y+1}(k), \\ &\quad x_{n_y+2}(k), \cdots, x_{\bar{t}}(k)]^{\mathrm{T}} \\ &= [y(k), y(k-1), \cdots, y(k-n_y), \\ &\quad u(k-1), \cdots, u(k-n_u)]^{\mathrm{T}}. \end{aligned}$$

Then, the input-output description form of the system (1) can be transformed into its extended state space description form via variable substitution, i.e.,

$$\begin{cases} x_1(k+1) = f(\boldsymbol{x}(k), u(k), k), \\ x_{\hat{j}}(k+1) = x_{\hat{j}-1}(k), \quad (2 \le \hat{j} \le n_y + 1), \\ x_{n_y+2}(k+1) = u(k), \\ x_{\hat{j}}(k+1) = x_{\hat{j}-1}(k), \quad (n_y + 3 \le \hat{j} \le \bar{t}), \end{cases} \tag{8}$$

$$y(k) = x_1(k). \tag{9}$$

Consider the following optimal control problem (Kirk, 2004):

$$\min_{u(k)} J = \frac{1}{2}\sum_{k=0}^{\mathrm{N}-1}(y_{id}(k) - y(k))^2, \tag{10}$$

where $y(k)$ is determined by (8) and (9).

Introduce the Hamilton function

$$\begin{aligned} &H(\boldsymbol{x}(k), u(k), \boldsymbol{\lambda}(k+1), k) \\ &= \frac{1}{2}(y_{id}(k) - x_1(k))^2 \\ &\quad + \lambda_1(k+1) f(\boldsymbol{x}(k), u(k), k) \\ &\quad + \cdots + \lambda_{\bar{t}}(k+1) x_{\bar{t}-1}(k), \end{aligned} \tag{11}$$

for convenience, suppose $H(k) = H(\boldsymbol{x}(k), u(k), \boldsymbol{\lambda}(k+1), k)$, and $\boldsymbol{x}(k)$, $\boldsymbol{\lambda}(k)$ satisfy the following difference equations:

$$\boldsymbol{x}(k+1) = \frac{\partial H(k)}{\partial \boldsymbol{\lambda}(k+1)}, \tag{12}$$

$$\boldsymbol{\lambda}(k) = \frac{\partial H(k)}{\partial \boldsymbol{x}(k)}. \tag{13}$$

If the control law is constrained, an extreme value of Hamilton function is taken on the optimal control sequence $u^*(k)$ and the corresponding optimal state series $\boldsymbol{x}^*(k)$ by Pontryagin minimum principle, i.e.,

$$\begin{aligned} &H(\boldsymbol{x}^*(k), u^*(k), \boldsymbol{\lambda}(k+1), k) \\ &= \min_{u(k)\in\Omega} H(\boldsymbol{x}^*(k), u(k), \boldsymbol{\lambda}(k+1), k), \end{aligned} \tag{14}$$

where Ω is a bounded closed set. If not, the value of $u(k)$ can be taken from the whole control space R; then, the extreme condition is

$$\frac{\partial H(k)}{\partial u(k)} = 0. \tag{15}$$

Here, we consider the second case, i.e., $u(k) \in R$.

For any given series of the control sequence $u(k)$, improve it by repeated iteration in the direction lowering the gradient of Hamilton function $H(k)$, until the necessary condition (15) is met and the numerical solution of the optimal control law $u^*(k)$ $(k = 0,1,\cdots,\mathrm{N}-1)$ is obtained. For convenience and without loss of generality, let $\boldsymbol{u} = (u(0), u(1), \cdots, u(\mathrm{N}-1))^{\mathrm{T}}$, and the calculation steps go as below:

Algorithm 1. Step 1. Select an initial value of the state vector $\boldsymbol{x}(0)$ and a nominal control history $\boldsymbol{u}_0$. Set the iteration index M =0.

Step 2. Calculate the state trajectory $\boldsymbol{x}_M(k)$ by (8) based on $\boldsymbol{x}(0)$ and $\boldsymbol{u}_M$, where $k = 1,2,\cdots,\mathrm{N}$.

Step 3. Calculate the partial derivative of $H(k)$ with respect to $u(k)$ at $u_M(k)$, and set it as $g_M(k) = \left.\frac{\partial H(k)}{\partial u(k)}\right|_{u(k)=u_M(k)}$, where $k = 0,1,\cdots,\mathrm{N}-1$. Let $\boldsymbol{g}_M = (g_M(0), g_M(1), \cdots, g_M(\mathrm{N}-1))^{\mathrm{T}}$.

Step 4. Calculate $\|\boldsymbol{g}_M\|$. If $\|\boldsymbol{g}_M\| < \varepsilon$, where ε is a given positive constant, and then stop; or else, revise the control history: $\boldsymbol{u}_{M+1} = \boldsymbol{u}_M - \sigma\boldsymbol{g}_M$, i.e., $u_{M+1}(k) = u_M(k) - \sigma g_M(k)$, where σ is a fixed step size and $k = 0,1,\cdots,\mathrm{N}-1$, let $M = M+1$, and go to Step 2.

4.2 Selection of Initial MTNC Weight Parameter Values for Online Learning

Generate the MTN optimal controller (MTNC) automatically to approximate the numerical solution of the optimal control law $u^*(k)$ for the system (1) relative to the ideal output signal $y_{id}(k)$. The initial weight parameters $\boldsymbol{w}_0^*$ for online learning can be obtained through offline learning. The following describe how to choose these parameters for online training by the CG method. Let $u(k)$ denote the output of MTNC at time k.

The fitting error is defined as

$$\bar{e}(k) = u^*(k) - u(k), \tag{16}$$

and the corresponding mean square error is

$$\bar{E} = \frac{1}{2}\sum_{k=0}^{N-1}\bar{e}^2(k). \tag{17}$$

Substituting Equations (3) and (16) into (17) yields

$$\bar{E} = \frac{1}{2}\sum_{k=0}^{N-1}\left(u^*(k) - \sum_{p=1}^{N(t,m)} w_p \prod_{q=1}^{t} z_q^{\lambda(p,q)}(k)\right)^2. \tag{18}$$

Let

$$\boldsymbol{\alpha}(k) = \left(\prod_{q=1}^{t} z_q^{\lambda(1,q)}(k), \prod_{q=1}^{t} z_q^{\lambda(2,q)}(k), \cdots, \prod_{q=1}^{t} z_q^{\lambda(N(t,m),q)}(k)\right)^{\mathrm{T}},$$

$$\boldsymbol{A} = (\boldsymbol{\alpha}(0), \boldsymbol{\alpha}(1), \cdots, \boldsymbol{\alpha}(\mathrm{N}-1)),$$

$$\boldsymbol{w} = (w_1, w_2, \cdots, w_{N(t,m)})^{\mathrm{T}}.$$

Then, Equation (18) can be rewritten as

$$\begin{aligned}\bar{E} &= \frac{1}{2}\sum_{k=0}^{N-1}(u^*(k) - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\alpha}(k))^2 \\ &= \frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\sum_{k=0}^{N-1}(\boldsymbol{\alpha}(k)\boldsymbol{\alpha}^{\mathrm{T}}(k))\boldsymbol{w} \\ &\quad - (\sum_{k=0}^{N-1}u^*(k)\boldsymbol{\alpha}^{\mathrm{T}}(k))\boldsymbol{w} + \frac{1}{2}\sum_{k=0}^{N-1}u^{*2}(k).\end{aligned} \tag{19}$$

Calculating the partial derivative of $\bar{E}$ with respect to the weight parameter vector $\boldsymbol{w}$, we get

$$\frac{\partial \bar{E}}{\partial \boldsymbol{w}} = \sum_{k=0}^{N-1}(\boldsymbol{\alpha}(k)\boldsymbol{\alpha}^{\mathrm{T}}(k))\boldsymbol{w} - \sum_{k=0}^{N-1}u^*(k)\boldsymbol{\alpha}(k).$$

Assuming $\boldsymbol{g} = \frac{\partial \bar{E}}{\partial \boldsymbol{w}}$, $\boldsymbol{Q} = \sum_{k=0}^{N-1}(\boldsymbol{\alpha}(k)\boldsymbol{\alpha}^{\mathrm{T}}(k))$, $\boldsymbol{b} = -\sum_{k=0}^{N-1}u^*(k)\boldsymbol{\alpha}(k)$, the partial derivative can be rewritten as

$$\boldsymbol{g} = \boldsymbol{Q}\boldsymbol{w} + \boldsymbol{b}. \tag{20}$$

Let $\boldsymbol{u}^* = (u^*(0), u^*(1), \cdots, u^*(\mathrm{N}-1))^{\mathrm{T}}$, and $\boldsymbol{Q}$, $\boldsymbol{b}$ and $\boldsymbol{g}$ can be obtained by

$$\boldsymbol{Q} = \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}},\ \boldsymbol{b} = -\boldsymbol{A}\boldsymbol{u}^*,\ \boldsymbol{g} = \boldsymbol{A}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{w} - \boldsymbol{u}^*).$$

For the given numerical solution of the optimal control law $u^*(k)$, the value of the weight parameter vector $\boldsymbol{w}$ for MTNC can be updated in the negative gradient direction of $\bar{E}$. Let $\boldsymbol{w}_\tau$ denote the weight parameter vector after the τ th training, and we have $\boldsymbol{w}_{\tau+1} = \boldsymbol{w}_\tau - \mu_\tau \boldsymbol{g}_\tau$, where μ_τ is step size, and $\mu_\tau = \frac{\boldsymbol{g}_\tau^{\mathrm{T}}\boldsymbol{g}_\tau}{\boldsymbol{g}_\tau^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{g}_\tau}$. Using the gradient method, the search direction, as always vertical to the last, takes a zigzag course; however, the problem can be solved effectively by the CG method. By the latter, the weight can be updated as $\boldsymbol{w}_{\tau+1} = \boldsymbol{w}_\tau + \mu_\tau \boldsymbol{p}_\tau$, where $\boldsymbol{p}_\tau = -\boldsymbol{g}_\tau + a_{\tau-1}\boldsymbol{p}_{\tau-1}$, $\mu_\tau = \frac{\boldsymbol{g}_\tau^{\mathrm{T}}\boldsymbol{g}_\tau}{\boldsymbol{p}_\tau^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{p}_\tau}$, and $a_{\tau-1} = \frac{\boldsymbol{g}_\tau^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{p}_{\tau-1}}{\boldsymbol{p}_{\tau-1}^{T}\boldsymbol{Q}\boldsymbol{p}_{\tau-1}}$, $(\tau > 1)$. The initial value is $\boldsymbol{p}_1 = -\boldsymbol{g}_1$.

5 REAL-TIME SELF-TUNING OF MTNC PARAMETERS

AS the adaptive controller is required to adjust its parameters automatically in real time, BP algorithm is the most widely used training algorithm for the networks, but it has such drawbacks as slow convergence speed, easy falling into local optimal point, etc. To raise the convergence speed, a momentum term is often added to the BP algorithm for accelerating and stabilizing the learning procedure (Phansalkar and Sastry, 1994; Rummelhart et al., 1986), followed by the introduction of a second-order momentum term (Pearlmutter, 1992) and an extra term proportional to the error (Zweiri et al., 2003, 2005). Good results have been attained by the improvement of BP algorithms. For further enhancement of the learning speed, a four-term BP algorithm developed by introducing the second-order momentum term into the three-term BP algorithm is proposed here for MTNC online training as follows:

$$\begin{aligned}\Delta\boldsymbol{w}(k) &= -\alpha\nabla\boldsymbol{E}(\boldsymbol{w}(k)) + \beta_1\Delta\boldsymbol{w}(k-1) \\ &\quad + \beta_2\Delta\boldsymbol{w}(k-2) + \gamma\boldsymbol{e}(\boldsymbol{w}(k)),\end{aligned} \tag{21}$$

where $\nabla\boldsymbol{E}(\boldsymbol{w}(k)) = \frac{\partial E(\boldsymbol{w}(k))}{\partial \boldsymbol{w}(k)}$ denotes the partial derivative of the performance indicator $E(\boldsymbol{w}(k))$ with respect to the weight parameter vector $\boldsymbol{w}(k)$ at time $k+1$, $\Delta\boldsymbol{w}(k-1)$ is the first-order momentum term, $\Delta\boldsymbol{w}(k-2)$ is the second-order momentum term, $e(\boldsymbol{w}(k))$ is the tracking error with $\boldsymbol{w}(k)$, $\boldsymbol{e}(\boldsymbol{w}(k)) = (e(\boldsymbol{w}(k)), e(\boldsymbol{w}(k)), \cdots, e(\boldsymbol{w}(k)))^{\mathrm{T}}$, α is the learning factor, β_1 is the first-order momentum factor, β_2 is the second-order momentum factor, and γ is the proportional factor. The performance indicator is

$E(\boldsymbol{w}(k)) = \frac{1}{2}e^2(\boldsymbol{w}(k))$, and the tracking error is $e(k+1) = e(\boldsymbol{w}(k)) = r(k+1) - y(k+1)$.

From Equation (21), we acquire

$$\Delta \boldsymbol{w}(k) = \boldsymbol{w}(k+1) - \boldsymbol{w}(k), \tag{22}$$

$$\nabla \boldsymbol{E}(\boldsymbol{w}(k)) = -e(\boldsymbol{w}(k))\frac{dy(k+1)}{d\boldsymbol{w}(k)}, \tag{23}$$

where

$$\frac{dy(k+1)}{d\boldsymbol{w}(k)} = \frac{\partial y(k+1)}{\partial u(k)}\frac{du(k)}{d\boldsymbol{w}(k)}. \tag{24}$$

With the system model not precisely known, the first term on the right side of Equation (24) can be replaced by

$$\frac{\partial y(k+1)}{\partial u(k)} = sign\left(\frac{y(k+1) - y(k)}{u(k) - u(k-1)}\right), \tag{25}$$

and the influence due to computational inaccuracy can be compensated by adjusting the learning rate.

Meanwhile, the second term on the right side of Equation (24) can be calculated as

$$\frac{du(k)}{d\boldsymbol{w}(k)} = \boldsymbol{\alpha}(k), \tag{26}$$

where $\boldsymbol{\alpha}(k)$ refers to the same as mentioned before.

6 ALGORITHMIC STEPS FOR MTN OPTIMAL CONTROL SCHEME

ALGORITHM steps for MTN optimal control scheme are summarized as below:

Algorithm 2. Step 1. Select an ideal output signal $y_{id}(k)$ to replace the given reference signal $r(k)$.

Step 2. Call Algorithm 1 to calculate the optimal control law $u^*(k)$ of the system (1) relative to the ideal output signal $y_{id}(k)$, and the corresponding optimal output signal $y^*(k)$ is taken as the desired output signal.

Step 3. Generate MTNC automatically to fit the optimal control law $u^*(k)$, select a group of initial weight parameter vector $\boldsymbol{w}_0$ at the interval $(-1,1)$ in a random way for offline training, and train MTNC by CG method to acquire the initial weight parameter vector $\boldsymbol{w}_0^*$ for online training.

Step 4. Obtain $r(k)$, $u(k)$ and $y(k)$, and take the real-time tracking error $e(k)$ as $e(k) = r(k) - y(k)$, where $k = 0,1,\cdots$.

Step 5. Obtain the input signal $u(k)$ of the system (1) by putting $\boldsymbol{z}(k)$ into MTNC controller (2), and $u(k)$ into (1).

Step 6. Employ Equations (21)-(26) to adjust the weight parameter vector for MTNC online to track the given reference signal $r(k)$, and then go to Step 1.

Step 7. Go to Step 4 and continue the process.

7 STABILITY ANALYSIS FOR LEARNING ALGORITHM

SIMILAR to Zweiri et al. (2005), let $\boldsymbol{\rho}_1(k) = \boldsymbol{w}(k)$, $\boldsymbol{\rho}_2(k) = \Delta \boldsymbol{w}(k-1)$ and $\boldsymbol{\rho}_3(k) = \Delta \boldsymbol{w}(k-2)$, changing Equations (21) into

$$\begin{aligned}\boldsymbol{\rho}_1(k+1) = \boldsymbol{\rho}_1(k) - \alpha\nabla \boldsymbol{E}(\boldsymbol{\rho}_1(k)) + \beta_1\boldsymbol{\rho}_2(k)\\ + \beta_2\boldsymbol{\rho}_3(k) + \gamma\boldsymbol{e}(\boldsymbol{\rho}_1(k)),\end{aligned} \tag{27}$$

$$\begin{aligned}\boldsymbol{\rho}_2(k+1) = -\alpha\nabla \boldsymbol{E}(\boldsymbol{\rho}_1(k)) + \beta_1\boldsymbol{\rho}_2(k)\\ + \beta_2\boldsymbol{\rho}_3(k) + \gamma\boldsymbol{e}(\boldsymbol{\rho}_1(k)),\end{aligned} \tag{28}$$

$$\boldsymbol{\rho}_3(k+1) = \boldsymbol{\rho}_2(k). \tag{29}$$

Lemma 1. If $\boldsymbol{c} = (\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3)$ is an equilibrium point of the system shown in Equations (27)-(29), then $\boldsymbol{c}_2 = \boldsymbol{0}$, $\boldsymbol{c}_3 = \boldsymbol{0}$ and $\alpha\nabla \boldsymbol{E}(\boldsymbol{\rho}_1(k)) = \gamma\boldsymbol{e}(\boldsymbol{\rho}_1(k))$.

Proof. If $\boldsymbol{c} = (\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3)$ is an equilibrium point, there exist $\boldsymbol{\rho}_1(k) = \boldsymbol{c}_1$, $\boldsymbol{\rho}_2(k) = \boldsymbol{c}_2$, $\boldsymbol{\rho}_3(k) = \boldsymbol{c}_3$ and

$$\boldsymbol{\rho}_1(k+1) - \boldsymbol{\rho}_1(k) = \boldsymbol{0}, \tag{30}$$

$$\boldsymbol{\rho}_2(k+1) - \boldsymbol{\rho}_2(k) = \boldsymbol{0}, \tag{31}$$

$$\boldsymbol{\rho}_3(k+1) - \boldsymbol{\rho}_3(k) = \boldsymbol{0}. \tag{32}$$

Substituting Equations (30)-(32) into Equations (27)-(29) gives

$$\begin{aligned}-\alpha\nabla \boldsymbol{E}(\boldsymbol{\rho}_1(k)) + \beta_1\boldsymbol{\rho}_2(k)\\ + \beta_2\boldsymbol{\rho}_3(k) + \gamma\boldsymbol{e}(\boldsymbol{\rho}_1(k)) = \boldsymbol{0},\end{aligned} \tag{33}$$

$$\begin{aligned}\boldsymbol{\rho}_2(k) = -\alpha\nabla \boldsymbol{E}(\boldsymbol{\rho}_1(k)) + \beta_1\boldsymbol{\rho}_2(k)\\ + \beta_2\boldsymbol{\rho}_3(k) + \gamma\boldsymbol{e}(\boldsymbol{\rho}_1(k)),\end{aligned} \tag{34}$$

$$\boldsymbol{\rho}_3(k) = \boldsymbol{\rho}_2(k). \tag{35}$$

By summarization, we get

$$\alpha\nabla \boldsymbol{E}(\boldsymbol{\rho}_1(k)) = \gamma\boldsymbol{e}(\boldsymbol{\rho}_1(k)), \tag{36}$$

$$\boldsymbol{\rho}_2(k) = 0, \tag{37}$$

$$\boldsymbol{\rho}_3(k) = 0, \tag{38}$$

that is,

$$\alpha\nabla \boldsymbol{E}(\boldsymbol{\rho}_1(k)) = \gamma\boldsymbol{e}(\boldsymbol{\rho}_1(k)), \tag{39}$$

$$\boldsymbol{c}_2 = \boldsymbol{0}, \tag{40}$$

$$\boldsymbol{c}_3 = \boldsymbol{0}. \tag{41}$$

That completes the proof of Lemma 1.

Let $\boldsymbol{\varphi}_1(k) = \boldsymbol{\rho}_1(k) - \boldsymbol{c}_1$, $\boldsymbol{\varphi}_2(k) = \boldsymbol{\rho}_2(k) - \boldsymbol{c}_2$ and $\boldsymbol{\varphi}_3(k) = \boldsymbol{\rho}_3(k) - \boldsymbol{c}_3$; then Equations (42)-(44) hold:

$$\begin{aligned}\boldsymbol{\varphi}_1(k+1) = {} & \boldsymbol{\varphi}_1(k) - \alpha\nabla \boldsymbol{E}(\boldsymbol{c}_1 + \boldsymbol{\varphi}_1(k)) \\ & + \beta_1(\boldsymbol{c}_2 + \boldsymbol{\varphi}_2(k)) + \beta_2(\boldsymbol{c}_3 + \boldsymbol{\varphi}_3(k)) \\ & + \gamma \boldsymbol{e}(\boldsymbol{c}_1 + \boldsymbol{\varphi}_1(k)),\end{aligned} \tag{42}$$

$$\begin{aligned}\boldsymbol{\varphi}_2(k+1) = {} & -\alpha\nabla \boldsymbol{E}(\boldsymbol{c}_1 + \boldsymbol{\varphi}_1(k)) + \beta_1(\boldsymbol{c}_2 + \boldsymbol{\varphi}_2(k)) \\ & + \beta_2(\boldsymbol{c}_3 + \boldsymbol{\varphi}_3(k)) + \gamma \boldsymbol{e}(\boldsymbol{c}_1 + \boldsymbol{\varphi}_1(k)),\end{aligned} \tag{43}$$

$$\boldsymbol{\varphi}_3(k+1) = \boldsymbol{\varphi}_2(k). \tag{44}$$

At the equilibrium point $\boldsymbol{c} = (\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3)$, expanding Equations (42) and (43) by Taylor expansion, we obtain

$$\begin{aligned}\boldsymbol{\varphi}_1(k+1) \approx {} & \boldsymbol{\varphi}_1(k) - \alpha\nabla^2 \boldsymbol{E}(\boldsymbol{c}_1)\boldsymbol{\varphi}_1(k) + \beta_1\boldsymbol{\varphi}_2(k) \\ & + \beta_2\boldsymbol{\varphi}_3(k) + \gamma\nabla \boldsymbol{e}(\boldsymbol{c}_1)\boldsymbol{\varphi}_1(k),\end{aligned} \tag{45}$$

$$\begin{aligned}\boldsymbol{\varphi}_2(k+1) \approx {} & -\alpha\nabla^2 \boldsymbol{E}(c_1)\boldsymbol{\varphi}_1(k) + \beta_1\boldsymbol{\varphi}_2(k) \\ & + \beta_2\boldsymbol{\varphi}_3(k) + \gamma\nabla \boldsymbol{e}(\boldsymbol{c}_1)\boldsymbol{\varphi}_1(k),\end{aligned} \tag{46}$$

$$\boldsymbol{\varphi}_3(k+1) = \boldsymbol{\varphi}_2(k). \tag{47}$$

Let $\boldsymbol{B} = \nabla^2 \boldsymbol{E}(\boldsymbol{c}_1) \in R^{N(t,m)\times N(t,m)}$ and $\boldsymbol{D} = \nabla \boldsymbol{e}(\boldsymbol{c}_1) \in R^{N(t,m)\times N(t,m)}$, and then we have

$$\begin{bmatrix}\boldsymbol{\varphi}_1(k+1)\\ \boldsymbol{\varphi}_2(k+1)\\ \boldsymbol{\varphi}_3(k+1)\end{bmatrix} = \begin{bmatrix}\boldsymbol{I} - \alpha\boldsymbol{B} + \gamma\boldsymbol{D} & \beta_1\boldsymbol{I} & \beta_2\boldsymbol{I}\\ -\alpha\boldsymbol{B} + \gamma\boldsymbol{D} & \beta_1\boldsymbol{I} & \beta_2\boldsymbol{I}\\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{\varphi}_1(k)\\ \boldsymbol{\varphi}_2(k)\\ \boldsymbol{\varphi}_3(k)\end{bmatrix}. \tag{48}$$

Let $\boldsymbol{\Theta} = \begin{bmatrix}\boldsymbol{I} - \alpha\boldsymbol{B} + \gamma\boldsymbol{D} & \beta_1\boldsymbol{I} & \beta_2\boldsymbol{I}\\ -\alpha\boldsymbol{B} + \gamma\boldsymbol{D} & \beta_1\boldsymbol{I} & \beta_2\boldsymbol{I}\\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0}\end{bmatrix}$ and $\boldsymbol{\varphi}(k) = \begin{bmatrix}\boldsymbol{\varphi}_1(k)\\ \boldsymbol{\varphi}_2(k)\\ \boldsymbol{\varphi}_3(k)\end{bmatrix}$, and then Equation (48) can be simplified into

$$\boldsymbol{\varphi}(k+1) = \boldsymbol{\Theta}\boldsymbol{\varphi}(k). \tag{49}$$

Let ψ_i be an eigenvalue of $\boldsymbol{\Theta}$, known from Leigh (1985), the system (49) is stable if and only if $|\psi_i| < 1$ for all i, where $i = 1, 2, \cdots, 2N(t,m)$.

Lemma 2. Let $\boldsymbol{F} = \boldsymbol{B}/\gamma - \boldsymbol{D}/\alpha$. If λ_i is an arbitrary eigenvalue of $\boldsymbol{F}$, the corresponding eigenvalue ψ_i of $\boldsymbol{\Theta}$ can be determined by the following cubic equation:

$$\psi_i^3 + (\alpha\gamma\lambda_i - \beta_1 - 1)\psi_i^2 + (\beta_1 - \beta_2)\psi_i + \beta_2 = 0, \tag{50}$$

Proof. For any $\boldsymbol{B}$ and $\boldsymbol{D}$, $\boldsymbol{\Theta}$ is reversible if $\beta_2 \neq 0$. Let ψ_i be an eigenvalue of $\boldsymbol{\Theta}$; then we have $\psi_i \neq 0$ if $\boldsymbol{\Theta}$ is reversible. Let $\boldsymbol{\eta} = (\boldsymbol{\eta}_1^{\mathrm{T}}, \boldsymbol{\eta}_2^{\mathrm{T}}, \boldsymbol{\eta}_3^{\mathrm{T}})^{\mathrm{T}}$ be an eigenvector corresponding to the eigenvalue ψ_i of $\boldsymbol{\Theta}$, and we get

$$\boldsymbol{\Theta}\boldsymbol{\eta} = \psi_i\boldsymbol{\eta}, \tag{51}$$

which leads to

$$\boldsymbol{\eta}_1 - \alpha\boldsymbol{B}\boldsymbol{\eta}_1 + \gamma\boldsymbol{D}\boldsymbol{\eta}_1 + \beta_1\boldsymbol{\eta}_2 + \beta_2\boldsymbol{\eta}_3 = \psi_i\boldsymbol{\eta}_1, \tag{52}$$

$$-\alpha\boldsymbol{B}\boldsymbol{\eta}_1 + \gamma\boldsymbol{D}\boldsymbol{\eta}_1 + \beta_1\boldsymbol{\eta}_2 + \beta_2\boldsymbol{\eta}_3 = \psi_i\boldsymbol{\eta}_2, \tag{53}$$

$$\boldsymbol{\eta}_2 = \psi_i\boldsymbol{\eta}_3. \tag{54}$$

As revealed by Equations (52)-(54), the following equations hold.

$$\boldsymbol{\eta}_2 = \frac{\psi_i - 1}{\psi_i}\boldsymbol{\eta}_1, \tag{55}$$

$$\boldsymbol{\eta}_3 = \frac{\psi_i - 1}{\psi_i^2}\boldsymbol{\eta}_1. \tag{56}$$

Substituting Equations (55) and (56) into Equation (52) yields

$$(-\alpha\boldsymbol{B} + \gamma\boldsymbol{D})\boldsymbol{\eta}_1 = (\psi_i - 1)\left(1 - \frac{\beta_1}{\psi_i} - \frac{\beta_2}{\psi_i^2}\right)\boldsymbol{\eta}_1, \tag{57}$$

Substituting $\boldsymbol{F} = \boldsymbol{B}/\gamma - \boldsymbol{D}/\alpha$ into Equation (57) gives

$$\boldsymbol{F}\boldsymbol{\eta}_1 = \frac{1}{-\alpha\gamma}(\psi_i - 1)\left(1 - \frac{\beta_1}{\psi_i} - \frac{\beta_2}{\psi_i^2}\right)\boldsymbol{\eta}_1, \tag{58}$$

Known from Equation (58), η_1 is an eigenvector corresponding to the eigenvalue $\frac{1}{-\alpha\gamma}(\psi_i - 1)\left(1 - \frac{\beta_1}{\psi_i} - \frac{\beta_2}{\psi_i^2}\right)$ of $\boldsymbol{F}$. Without loss of generality, let

$$\lambda_i = \frac{1}{-\alpha\gamma}(\psi_i - 1)\left(1 - \frac{\beta_1}{\psi_i} - \frac{\beta_2}{\psi_i^2}\right), \tag{59}$$

Summarizing Equation (59) confirms that the equation $\psi_i^3 + (\alpha\gamma\lambda_i - \beta_1 - 1)\psi_i^2 + (\beta_1 - \beta_2)\psi_i + \beta_2 = 0$ holds, that is, ψ_i can be determined by Equation (50).

That completes the proof of Lemma 2.

Theorem 1. The necessary and sufficient conditions for the stability of the system (49) are

$$\max\left(0, \beta_1\left(1+\frac{1}{\beta_2}\right)+\beta_2\left(1-\frac{1}{\beta_2^2}\right)\right)<\alpha\gamma\lambda_i$$
$$<\min\left(2(\beta_1-\beta_2)+2, \beta_1\left(1+\frac{1}{\beta_2}\right)-\beta_2\left(1-\frac{1}{\beta_2^2}\right)\right),$$

$\beta_1>0$ and $0<\beta_2<1$.

Proof. $\beta_1>0$ and $\beta_2>0$ are known to be the momentum factors. Let $\varphi(z)=a_3z^3+a_2z^2+a_1z+a_0$, the roots of the cubic equation $\varphi(z)=0$ lie in the unit circle if and only if $\varphi(1)>0$, $(-1)^3\varphi(-1)>0$, $|a_0|<a_3$ and $|b_0|>|b_2|$ are satisfied (Phillips & Habor, 1996), where $a_3=1$, $a_2=\alpha\gamma\lambda_i-\beta_1-1$, $a_1=\beta_1-\beta_2$, $a_0=\beta_2$ and $b_j=\begin{vmatrix}a_0 & a_{3-j}\\ a_3 & a_j\end{vmatrix}$, $(j=0,1,2)$.

Then, the necessary and sufficient conditions for the stability of the system (49) are as follows:

$$1+(\alpha\gamma\lambda_i-\beta_1-1)+(\beta_1-\beta_2)+\beta_2>0,$$

$$-1+(\alpha\gamma\lambda_i-\beta_1-1)-(\beta_1-\beta_2)+\beta_2<0,$$

$$|\beta_2|<1,$$

$$|\beta_2^2-1|>|\beta_2(\alpha\gamma\lambda_i-\beta_1-1)-(\beta_1-\beta_2)|.$$

Summarizing the above confirms that the following formulas hold:

$$\alpha\gamma\lambda_i>0,$$

$$\alpha\gamma\lambda_i<2(\beta_1-\beta_2)+2,$$

$$-1<\beta_2<1,$$

$$\beta_1\left(1+\frac{1}{\beta_2}\right)+\beta_2\left(1-\frac{1}{\beta_2^2}\right)<\alpha\gamma\lambda_i$$
$$<\beta_1\left(1+\frac{1}{\beta_2}\right)-\beta_2\left(1-\frac{1}{\beta_2^2}\right).$$

Therefore, Theorem 1 holds.

That completes the proof of Theorem 1.

8 SIMULATION EXAMPLE

IN this section, an example is provided to demonstrate the effectiveness of MTN optimal control scheme. The example is a modification of example 2 in (Li et al., 2011). Consider the following SISO nonlinear time-varying discrete system:

$$y(k+1)=\frac{1.2(1-e^{-0.8k})y(k)+0.1\lg(k+1)}{(1+y^2(k))} + e^{-0.8k}y(k-1)+u(k)+e^{-0.8k}u(k-1). \tag{60}$$

The given reference signal is $r(k)=1$.

Set

$$x(k)=[x_1(k),x_2(k),x_3(k)]^{\mathrm{T}} =[y(k),y(k-1),u(k-1)]^{\mathrm{T}}.$$

We obtain the corresponding extended state space description as follows:

$$\begin{cases} x_1(k+1)=\dfrac{1.2(1-e^{-0.8k})x_1(k)+0.1\lg(k+1)}{1+x_1^2(k)} \\ \qquad +e^{-0.8k}x_2(k)+u(k)+e^{-0.8k}x_3(k), \\ x_2(k+1)=x_1(k), \\ x_3(k+1)=u(k), \end{cases} \tag{61}$$

$$y(k)=x_1(k). \tag{62}$$

a) Step response experiments

Here, MTN optimal controller is chosen as 3-10-1 with the structure of 3 input nodes, 1 output node and 2 powers. Its input vector is $\boldsymbol{z}(k)=[k_1u(k-1),l_1e(k-1),l_2e(k-2)]^{\mathrm{T}}$, and $k_1=l_1=l_2$ =0.001. Firstly, an ideal output signal $y_{\mathrm{id}}(k)$ was selected relative to the given reference signal $r(k)$, and Pontryagin minimum principle employed to obtain the numerical solution of the optimal control law $u^*(k)$ of the formulas (61) and (62) relative to the ideal output signal $y_{\mathrm{id}}(k)$, with the corresponding optimal output taken as the desired output signal $y_{\mathrm{op}}(k)$. A set of weight parameter values were determined in a random way at the interval $(-1,1)$, and the CG method was used to train MTNC offline to approximate the optimal control law $u^*(k)$ already obtained. Then a set of parameter values were secured and taken as the initial parameters for online MTNC training. In the off-line training process, the iterations of fitting were set as ite =100; finally, a four-term BP algorithm was adopted to adjust the weight parameters of MTNC for the real-time output tracking control of the system relative to the given reference signal $r(k)$. In the online training process, the learning rate was set as α_{M} =0.2, the first-order momentum factor β_{M_1} =0.15, the second-order momentum factor β_{M_2} =0.15, and the proportional factor γ_{M} =0.015. In the traditional NN adaptive control scheme, an NN controller (NNAC) is established by a three-layer NN with the 3-50-1 structure of 3 input neurons, 50 hidden neurons and 1 output neuron. Repeated experiments showed that a better control result can be obtained when 50 hidden neurons were chosen for NNAC. The activation functions for the hidden layer and output layer were set as $a_h(x)=\dfrac{1}{1+e^{-x}}$ and $a_o(x)=\dfrac{1-e^{-x}}{1+e^{-x}}$.

The initial values of the weight parameters were chosen in a random way at the interval $(-1,1)$, and BP algorithm was employed to train NNAC. For the training, the learning rate was set as α_N =0.2. The PID neural networks (PIDNN) control scheme was introduced to demonstrate the effectiveness of MTN optimal control scheme. In the PIDNN scheme, a PIDNN controller (PIDNNAC) was established by a three-layer network with the 2-3-1 structure of 2 input PID neurons, 3 hidden PID neurons and 1 output PID neuron (Shu (1999); Shu and Pi (2000)). The initial values of the weight parameters were set as w_{1j} =1 and $w_{2j} = -1$ for the weights between the input and hidden layers, and v_j =0 for the weights between the hidden and output layers with $j = 1,2,3$. BP algorithm was implemented to train PIDNNAC. For the training, the learning factor was set as α_P =0.3. Simulation results are presented in Figures 4-6.

Remark 1: In the following figures, r represents the given reference signal; yMC, yNAC and yPNAC are the actual output responses of the MTN optimal control scheme, the traditional NN adaptive control scheme and the PIDNN control scheme respectively; eMC, eNAC and ePNAC denote the corresponding tracking errors.

As shown in Figures 4-6, the overshoots are 28.83%, 52.74% and 42.74%; the performance index goes as E<10^{-3} after the iterations 13, 17 and 21; the steady state error obtained by taking the average of the absolute errors from the iterations 13 to 1000 is 0.0025 with the MTN optimal control scheme, that obtained from 17 to 1000 with the traditional NN adaptive control scheme is 8.4625×10^{-4}, and that from 21 to 1000 with the PIDNN control scheme is 5.8137×10^{-4}. Simulation results do confirm the feasibility and validity of the proposed control scheme.

b) Noise interference experiments

At the time instant 200, a Gaussian white noise with the mean of 0 and the standard deviation of 0.2 is added to the system (60), and the simulation results are illustrated by Figures 7-9.

As indicated by Figures 7-9, all of the three control schemes are of good robustness for noise interference. For better illustration of the robustness of the proposed control scheme, noise interference is expanded 30 times based on the above discussion, and the simulation results are given in Figures 10-12, which further verify its superior robustness to the other two upon noise interference expansion.

c) Input superposition experiments.

Figures 13-15 give the results from the simulation of adding an external input disturbance $d(k) = 0.1\sin(0.004\pi k)$ to the given reference signal $r(k)$.

As shown in Figures 13-15, the overshoots are 28.06%, 51.92% and 42.14%; the performance index E<10^{-3} after the iterations 13, 17 and 19; the steady state error obtained by taking the average of the absolute errors from the iterations 13 to 1000 is 0.0037 with the MTN optimal control scheme, that from 17 to 1000 is 0.0022 with the traditional NN adaptive control scheme, and that from 19 to 1000 is 0.0019 with the PIDNN control scheme. Simulation results do confirm the feasibility and validity of the control scheme proposed here.

Remark 2: For each control period, 19-time multiplication and 9-time addition operations are needed for the MTN optimal control scheme. However, with the exponential function expanded into finite terms with 2 powers, double addition operations are needed for the input layer, 8-time addition and 7-time multiplication operations are required for each hidden node; for the output layer node, there need 61-time addition and 57-time multiplication operations. For each control period, 463-time addition and 407-time multiplication operations are required with the traditional neural network adaptive control scheme. With the PIDNN control scheme, 9-time multiplication and 7-time addition operations are demanded. Compared with the other two, there are fewer operations needed by the MTN and PIDNN control schemes for each control period; compared with the TMS320F28335 DSP with the dominant frequency of 150MHz, for each control period the computation takes 186.7ns and 106.7ns with the MTN optimal control scheme and the PIDNN control scheme respectively, whereas 5800ns is required by the traditional neural network adaptive control scheme. To conclude, seen from the simulations, real-time control is better ensured by the MTN optimal and PIDNN control schemes than by the traditional neural network adaptive control scheme.

9 CONCLUSIONS

FOR SISO nonlinear time-varying discrete systems, an optimal control scheme based on MTN has been proposed to achieve the real-time output tracking control for the system relative to a given reference signal. Simulation results show that the MTN optimal control scheme is feasible and effective, and the system’s actual output response can track the given reference signal well in real time.

The main contributions of this paper are as follows:

(1) A feasible real-time self-tuning control scheme is proposed for the control design of nonlinear time-varying discrete systems; in this scheme, all the parameters of MTNC are adjusted automatically and simultaneously.

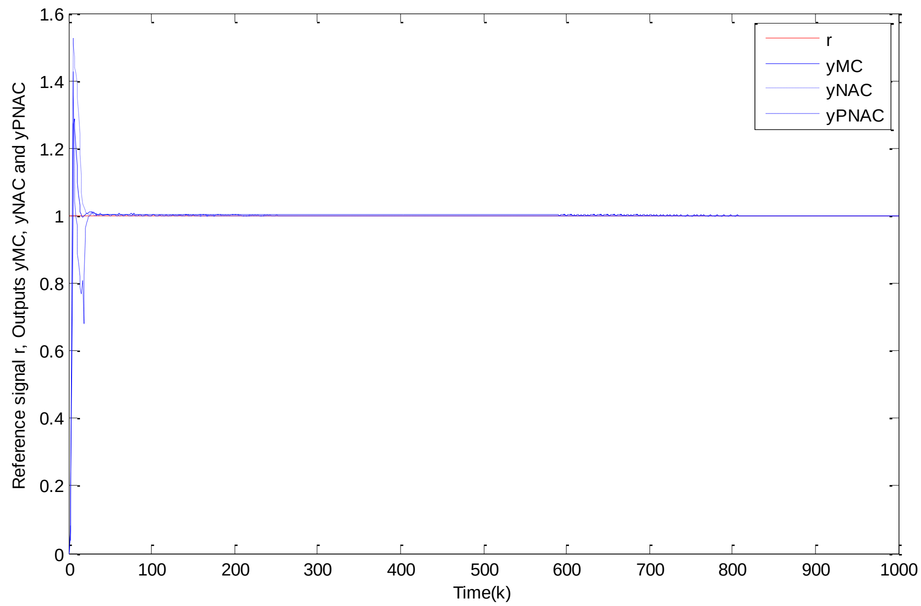

Figure 4. Tracking performance of output feedback.

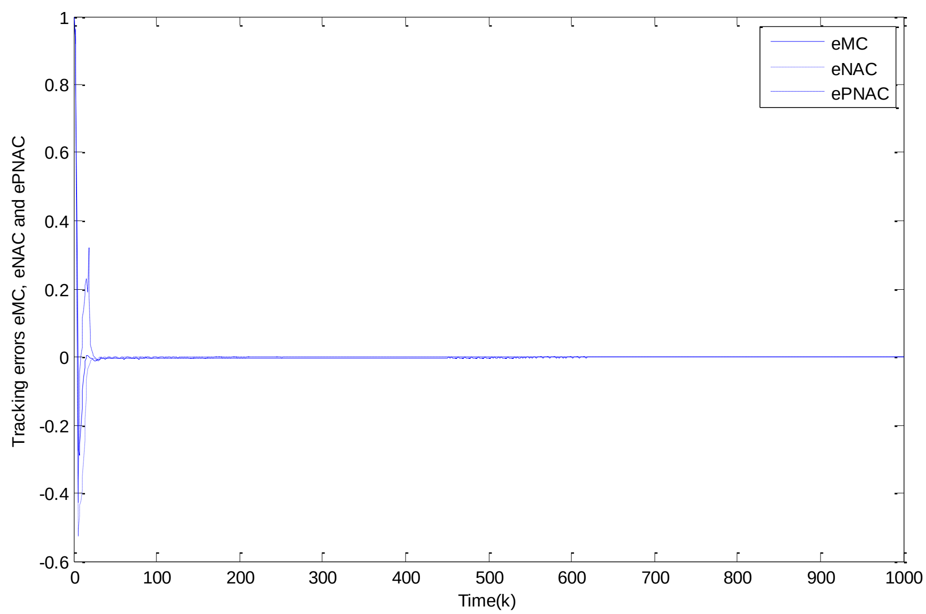

Figure 5. Tracking errors.

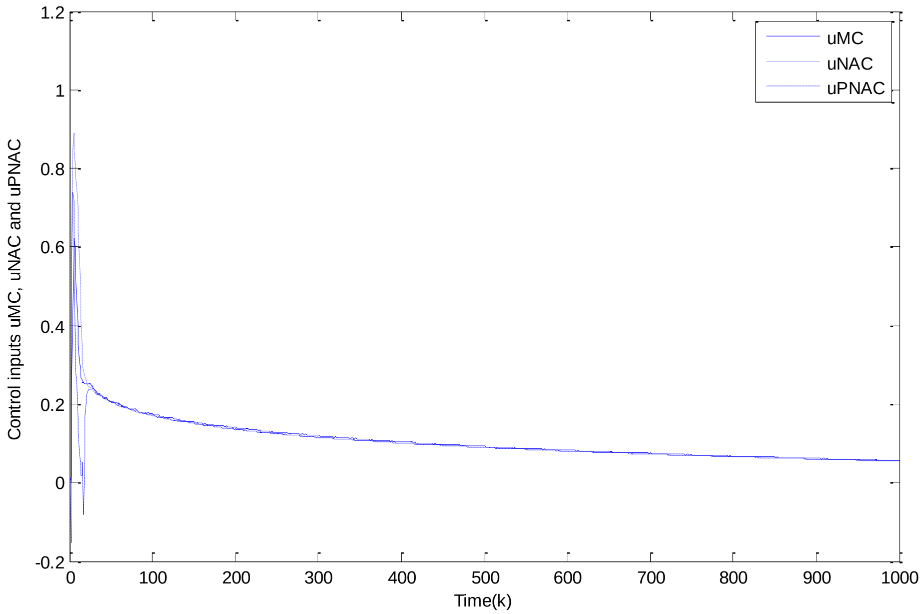

Figure 6. **Control inputs.**

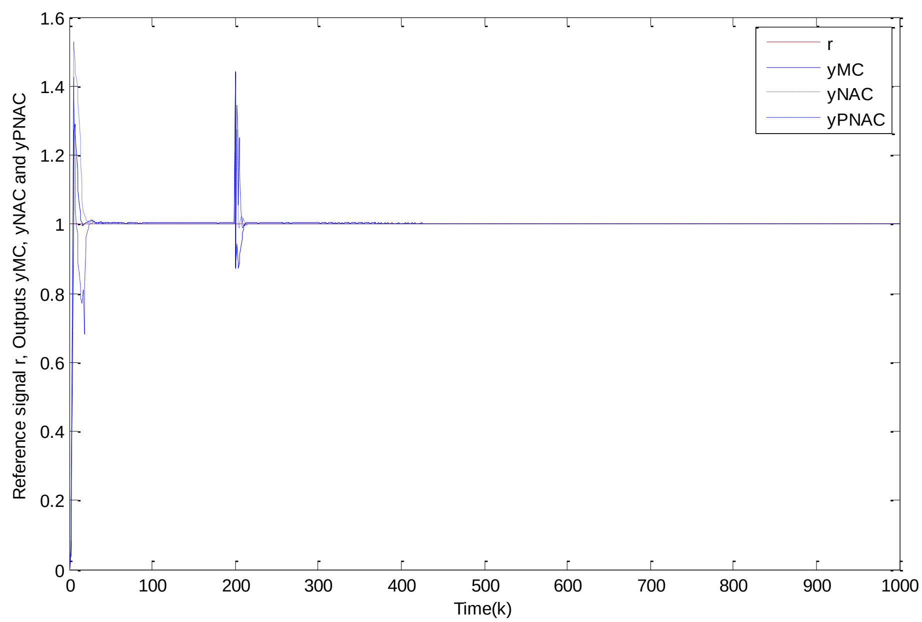

Figure 7. **Tracking performance of output feedbacks with noise interference.**

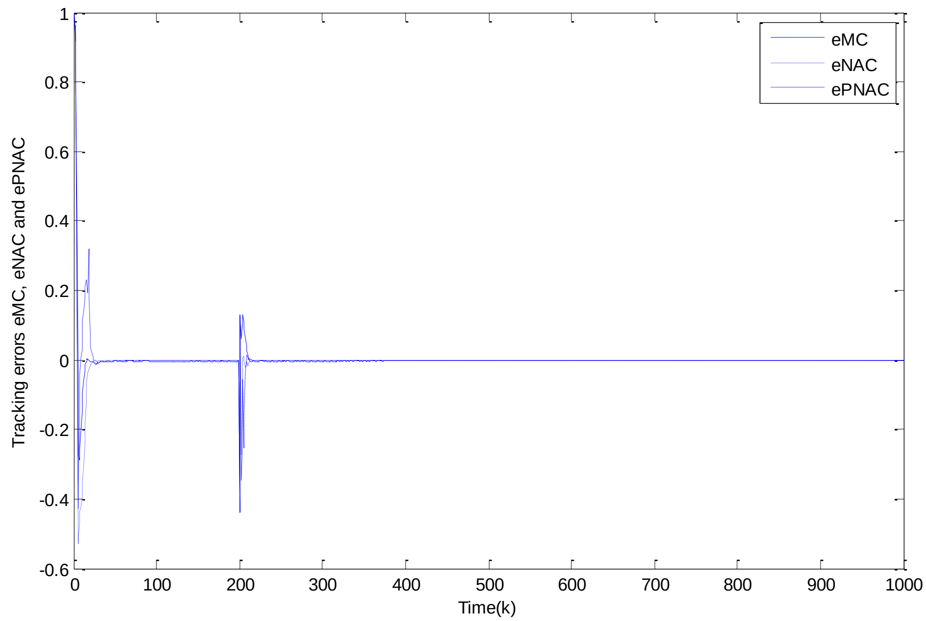

Figure 8. **Tracking errors with noise interference.**

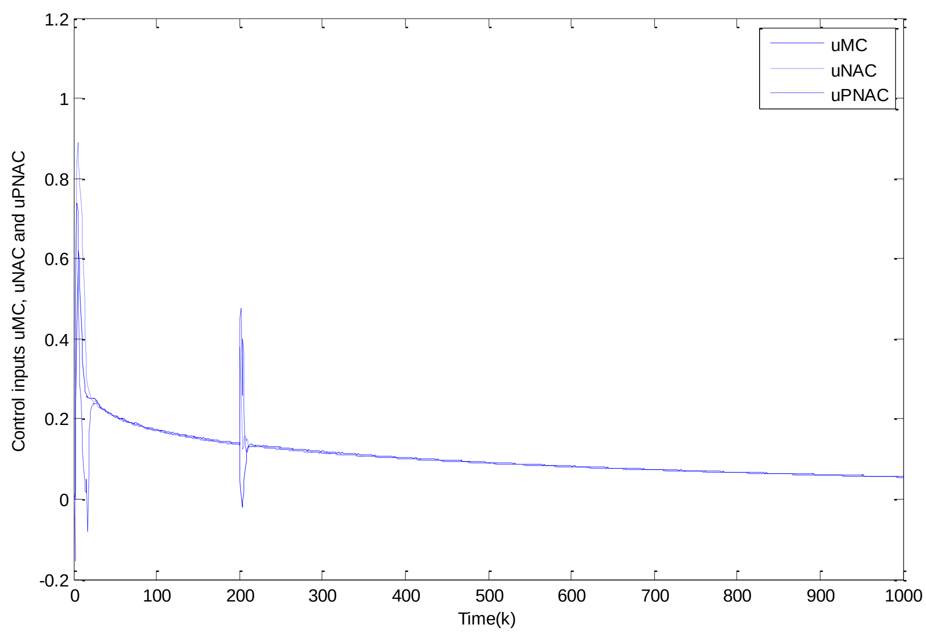

Figure 9. **Control inputs with noise interference.**

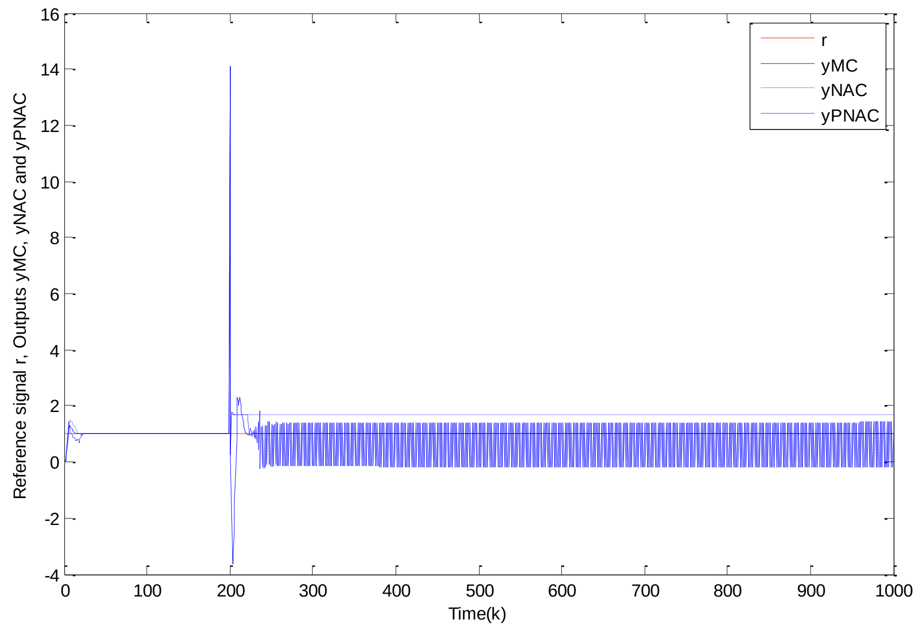

Figure 10. **Tracking performance of output feedbacks with expanded noise interference.**

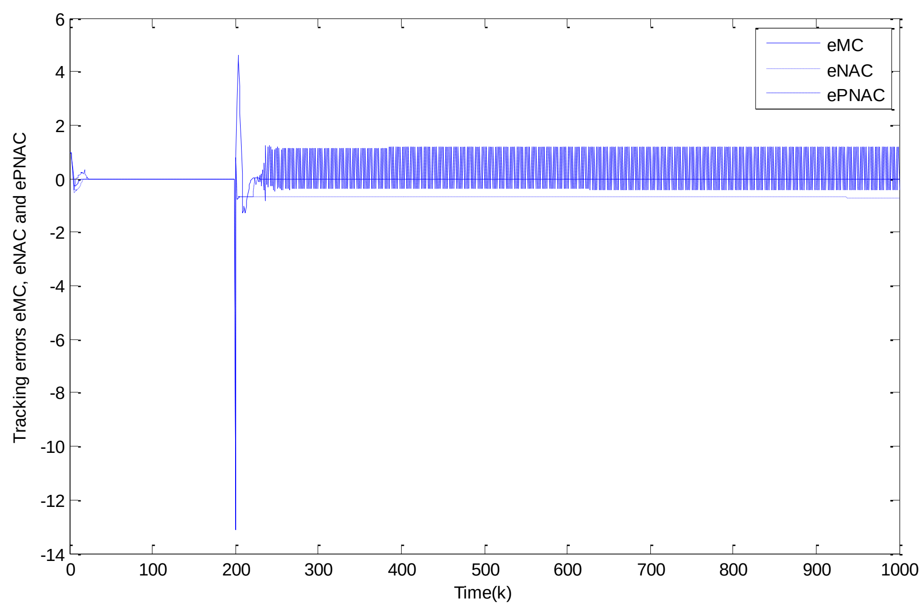

Figure 11. **Tracking errors with expanded noise interference.**

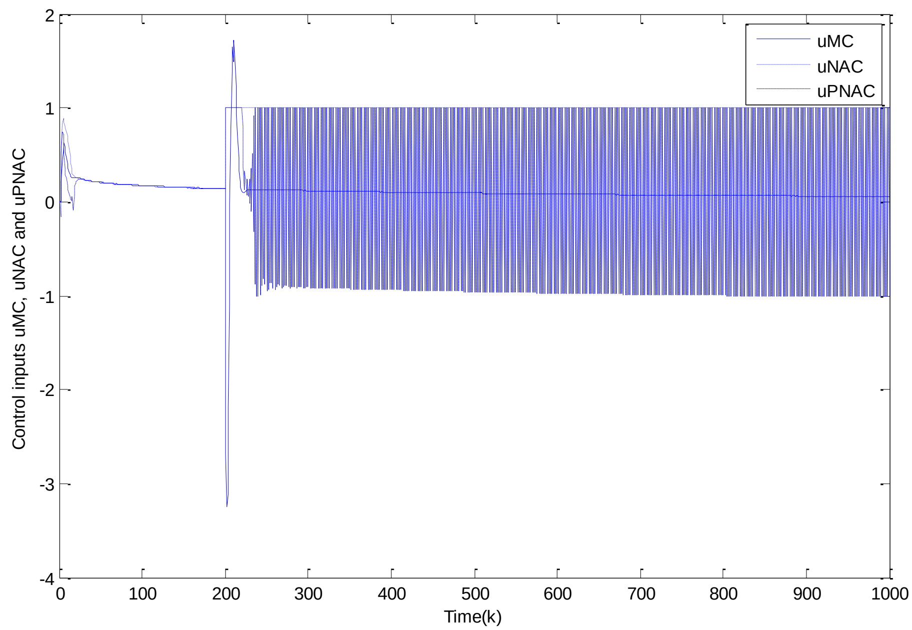

Figure 12. Control inputs with expanded noise Interference.

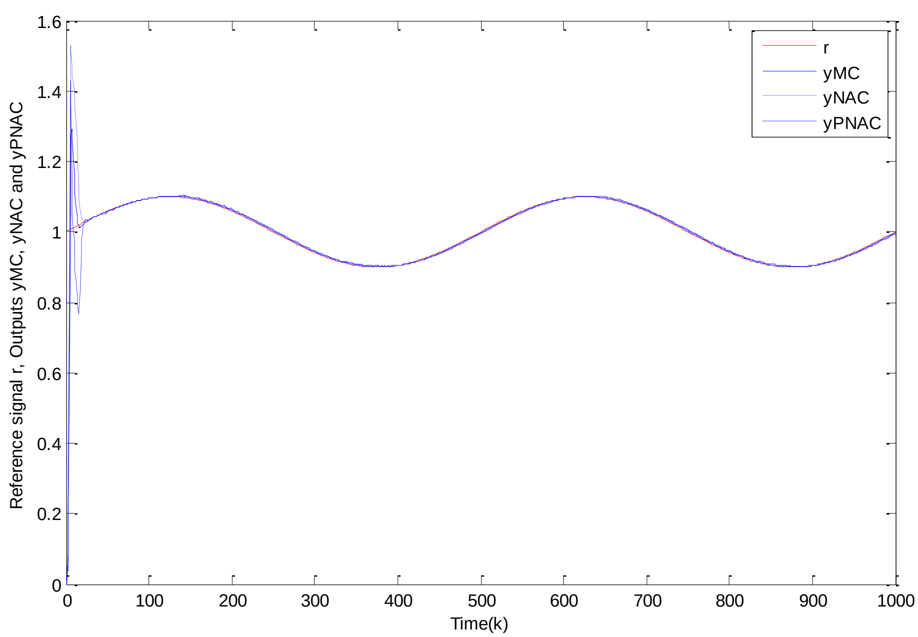

Figure 13. Tracking performance of output feedbacks with signal superposition.

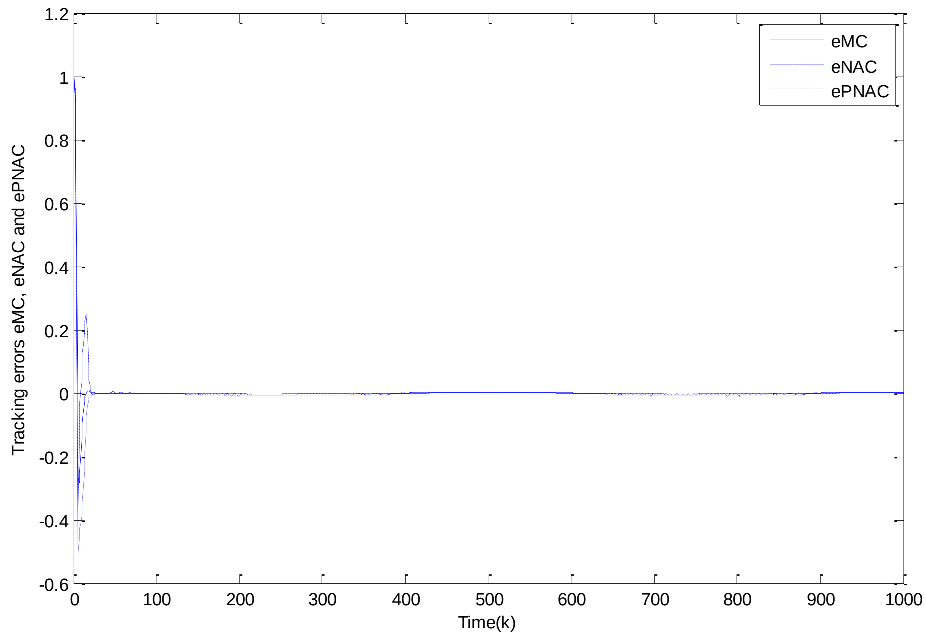

Figure 14. **Tracking errors with signal superposition.**

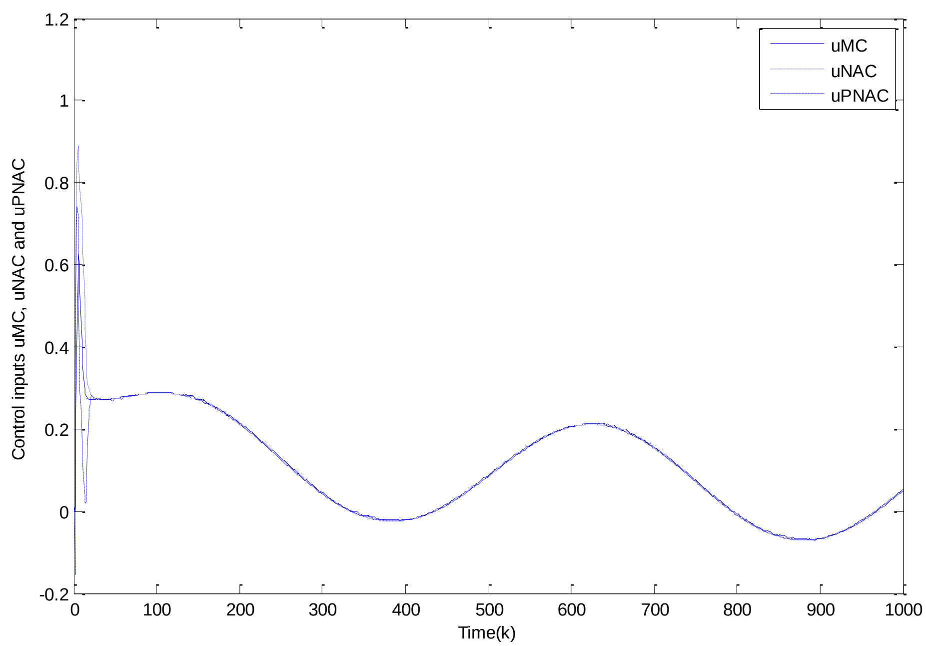

Figure 15. **Control inputs with signal superposition.**

(2) Unlike neural networks, MTNC can be generated automatically, which simplifies the network structure and raises the convergence speed by a great deal.

(3) A good network initial weight parameter identification scheme is developed by fitting the optimal control law, and the desirable control effects can be attained by the selection of initial weight parameter values of the controller.

(4) Apparently superior to the traditional BP algorithm with the momentum term, a four-term BP algorithm is adopted to adjust the weight parameters in real time, promising a faster convergence speed.

(5) The convergence conditions for the four-term BP algorithm are identified, and the ideal control effect can be achieved by proper selection of the learning, momentum and proportional factors.

10 ACKNOWLEDGMENT

THIS work was supported in part by the National Natural Science Foundation of China under Grants 61673112 and 60934008, the Fundamental Research Funds for the Key Universities of China under Grants 2242017K10003 and 2242014K10031, and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD). The authors would like to thank the anonymous reviewers and Professor Li Lu for their valuable comments that improve the quality of the paper.

11 DISCLOSURE STATEMENT

NO potential conflict of interest was reported by the authors.

12 REFERENCES

M. S. Ali and R. Saravanakumar, (2014). Novel delay-dependent robust H_∞ control of uncertain systems with distributed time-varying delay, *Applied Mathematics and Computation*. 249, 510-520.

K. H. Ang, G. Chong, and Y. Li, (2005). PID control system analysis, design, and technology, *IEEE Transactions on Control Systems Technology*. 13(4), 559-576.

K. J. Åström and T. Hägglund, (2001). The future of PID control, *Control Engineering Practice*. 9, 1163-1175.

S. Bezzaoucha, B. Marx, D. Maquin, and J. Ragot, (2015). State and output feedback control for Takagi-Sugeno systems with saturated actuators, *International Journal of Adaptive Control and Signal Processing*. 30(6), 888-905.

F. Cameron and D. E. Seborg, (1983). A self-tuning controller with a PID structure, *International Journal of Control*. 38(2), 401-417.

W. Chen, W. Li, and Q. Miao, (2010). Backstepping control for periodically time-varying systems using high-order neural network and Fourier series expansion, *ISA Transactions*. 49, 283-292.

Y. Fu and T. Chai, (2012). Robust self-tuning PI decoupling control of uncertain multivariable systems, *International Journal of Adaptive Control and Signal Processing*. 26(4), 316-332.

P. Hušek, (2014). Decentralized PI controller design based on phase margin specifications, *IEEE Transactions on Control Systems Technology*. 22(1), 346-351.

R. A. Jacobs, (1988). Increased rates of convergence through learning rate adaptation, *Neural Networks*. 1(4), 295-307.

L. Jin, M. M. Gupta, and N. Nikiforuk, (1995). Intelligent control for nonlinear systems using dynamic neural networks with robotic applications, *Intelligent Automation and Soft Computing*. 1(2), 123-144.

T. Jin, (2016). *Research and simulation of multi-dimensional Taylor network optimal control for the tank firing control in high speed motion*, Masters Thesis, Southeast University, Nanjing, China. (in Chinese).

D. E. Kirk, (2004). *Optimal control theory: an introduction* (eds). NJ: Dover Publications.

J. R. Leigh, (1985). *Applied Digital Control*. NJ: Prentice-Hall.

S. Li, J. Li, J. Qiu, H. Ji, and K. Zhu, (2011). Control design for arbitrary complex nonlinear discrete-time systems based on direct NNMRAC strategy, *Journal of Process Control*. 21, 103-110.

Y. Lin, (2015). *Research into modeling and prediction of nonlinear time series based on napofics (alternate positive negative feedbackics) and multi-dimensional Taylor network*, PhD Thesis, Southeast University, Nanjing, China. (in Chinese).

Y. Lin, H. S. Yan, and B. Zhou, (2014a). A novel modeling method based on multi-dimensional Taylor network and its application in time series prediction, *Advanced Materials Research*. 940, 480-484.

Y. Lin, H. S. Yan, and B. Zhou, (2014b). A novel nonlinear time series prediction method and its application in structural vibration response prediction, *Applied Mechanics and Materials*. 599, 1918-1921.

Y. Lin, H. S. Yan, and B. Zhou, (2014c). Nonlinear time series prediction method based on multi-dimensional Taylor network and its applications, *Control and Decision*. 29(5), 795-801. (in Chinese).

Z. Lu, L. S. Shieh, G. Chen, and J. Chandra, (2007). Identification and control of chaotic systems via recurrent high-order neural networks, *Intelligent Automation and Soft Computing*. 13(4), 357-372.

J. L. McClelland and D. E. Rumelhart, (1986). *Group TPR. Parallel distributed Processing:*

Psychological and Biological Models. Cambridge, MA: MIT Press.

A. A. Minai and R. D. Williams, (1990). Back-propagation heuristic: a study of the extended delta bar delta algorithm, *Paper presented at the 1990 IJCNN international Joint Conference on Networks*. 595-600.

Y. J. Mon, C. M. Lin, and C. H. Leng, (2008). Recurrent fuzzy neural network control for mimo nonlinear systems, *Intelligent Automation and Soft Computing*. 14(4), 395-415.

B. Nagarag and P. Vijayakumar, (2012). Tuning of a PID controller using soft computing methodologies applied to moisture control in paper machine, *Intelligent Automation and Soft Computing*. 18(4), 399-411.

K. S. Narendra, and K. Parthasarathy, (1990). Identification and control of dynamical systems using neural networks, *IEEE Transactions on Neural Networks*. 1(1), 4-27.

K. G. Papadopoulos, (2013). Papastefanaki EN, Margaris NI. Explicit Analytic PID tuning rules for the design for the design of type-III control loops, *IEEE Transactions on Industrial Electronics*. 60(10), 4650-4664.

A. Patrikar and J. Provence, (1996). Nonlinear system identification and adaptive control using polynomial networks, *Mathematical and computer modelling*. 23(1-2), 159-173.

B. Pearlmutter, (1992). Gradient descent: second-order momentum and saturating error, *Advances in Neural Information Processing Systems (NIPS)*. 887-894.

V. V. Phansalkar and P. S. Sastry, (1994). Analysis of the back-propagation algorithm with momentum, *IEEE Transactions on Neural Networks*. 5(3), 505-506.

C. L. Phillips and R. D. Habor, (1996). *Feedback Control Systems*. NJ: Prentice-Hall.

N. Qian, (1999). On the momentum term in gradient descent learning algorithms, *Neural Networks*. 12, 145-151.

A. B. Rad, W. L. Lo, and K. M. Tsang, (1997). Self-tuning PID controller using Newton-Raphson search method, *IEEE Transactions on Industrial Electronics*. 44(5), 717-725.

J. J. Rubio and W. Yu, (2007). Nonlinear system identification with recurrent neural networks and dead-zone Kalman filter algorithm, *Neurocomputing*. 70, 2460-2466.

D. E. Rummelhart, G. E. Hinton, and R. J. Williams, (1988). Learning representations by back-propagating errors. *Nature*. *323*, 533-536.

R. M. Sanner, and J. J. E. Slotine, (1992). Gaussian networks for direct adaptive control, *IEEE Transactions on Neural Networks*. 3(6), 837-863.

R. Saravanakumar and M. S. Ali, (2016). Robust H_∞ control for uncertain Markovian jump systems with mixed delays, *Chin. Phys. B*, 25(7), 1-6.

R. Saravanakumar, G. Rajchakit, M. S. Ali, Z. Xiang, and Y. H. Joo, (2017). Robust extended dissipativity criteria for discrete-time uncertain neural networks with time-varying delays, *Neural Computing and Applications*. 1-12.

R. Saravanakumar, M. S. Ali, and H. R. Karimi, (2017). Robust H_∞ control of uncertain stochastic Markovian jump systems with mixed time-varying delays. *International Journal of Systems Science*. 48(4), 862-872.

A. Savran, (2007). Multifeedback-Layer neural network, *IEEE Transactions on Neural Networks*. 18(2), 373-384.

S. Seshagiri and H. K. Khalil, (2000). Output feedback control of nonlinear systems using RBF neural networks, *IEEE Transactions on Neural Networks*. 11(1), 69-79.

K. Shojaei, (2015). Neural adaptive robust output feedback control of wheeled mobile robots with saturating actuators, *International Journal of Adaptive Control and Signal Processing*. 29(7), 855-876.

H. Shu, (1999). Analysis of PID neural network multivariable control systems, *Acta Automatica Sinica*. 25(1), 105-111. (in Chinese).

H. Shu and Y. Pi, (2000). PID neural networks for time-delay systems, *Computers and Chemical Engineering*. 24, 859-862.

Q. M. Sun and H. S. Yan, (2014). Optimal adjust control of SISO nonlinear systems based on multi-dimensional Taylor network only by output feedback, *Advanced Materials Research*. 1049-1050, 1389-1391.

D. C. Theodoridis, Y. S. Boutalis, and M. A. Christodoulou, (2012). Direct adaptive neuro-fuzzy trajectory tracking of uncertain nonlinear systems, *International Journal of Adaptive Control and Signal Processing*. 26(7), 660-688.

C. S. Tseng and B. S. Chen, (2001). Multiobjective PID control design in uncertain robotic systems using neural network elimination scheme, *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*. 31(6), 632-644.

C. S. Tseng, (2008). Robust tracking control design for uncertain robotic systems with persistent bounded disturbances, *Asian Journal of Control*. 10(4), 420-429.

Y. Wang, W. Sun, Y. Xiang, and S. Miao, (2009). Neural network-based robust tracking control for robots, *Intelligent Automation and Soft Computing*. 15(2), 211-222.

Y. Wei, J. Qiu, and H. R. Karimi, (2017a). Reliable output feedback control of discrete-time fuzzy affine systems with actuator faults, *IEEE*

Transactions on Circuits and Systems-I: Regular Papers. 64(1), 170-181.

Y. Wei, J. Qiu, H. K. Lam, and L. Wu, (2017b). Approaches to T-S fuzzy-affine-model-based reliable output feedback control for nonlinear to Stochastic Systems, *IEEE Transactions on fuzzy systems*. 25(3), 569-583.

Y. Wei, J. Qiu, H. R. Karimi, and M. Wang, (2014). New results on H_∞ dynamic output feedback control for Markovian jump systems with time-varying delay and defective mode information, *Optimal Control Applications and Methods*. 35(6), 656-675.

L. C. Xia, (2016). *Research into multi-dimensional Taylor network optimal control of the axisymmetric cruise missile flight for attacking static target*, Masters Thesis, Southeast University, Nanjing, China. (in Chinese).

J. X. Xu and R. Yan, (2004). Iterative learning control design without a priori knowledge of the control direction, *Automatica*. 40, 1803-1809.

H. S. Yan and A. M. Kang, (2017). Asymptotic tracking and dynamic regulation of SISO nonlinear system based on discrete multi-dimensional Taylor network, *IET Control Theory Applications.* 11(10), 1619-1626.

H. S. Yan, (2010). *MTN (multi-dimensional Taylor network) optimal control*, Technical Report, Research Institute of Control and Optimization of Manufacturing Systems, School of Automation, Southeast University, Nanjing, China. (in Chinese).

H. S. Yan, (2019). Multi-dimensional Taylor network optimal control. Retrieved from https://automation.seu.edu.cn/2019/0528/c24504a275249/page.htm.

H. S. Yan, (2014). *Comparison between neural network, multi-variable polynomial regression and multi-dimensional Taylor network*, Technical Report, Research Institute of Control and Optimization of Manufacturing Systems, School of Automation, Southeast University, Nanjing, China. (in Chinese).

X. Yang, D. Liu, D. Wang, and Q. Wei, (2014). Discrete-time online learning control for a class of unknown nonaffine nonlinear systems using reinforcement learning, *Neural Networks*. 55, 30-41.

Y. Yang, (2016). *Study on multi-dimensional Taylor network optimal control for ship roll stabilization*, Masters Thesis, Southeast University, Nanjing, China. (in Chinese).

J. Zhang and S. Yang, (2016). A novel PSO algorithm based on an incremental-PID-controlled search strategy, *Soft Computing*, *20*, 991-1005.

J. X. Zhang, (2015). *Research and simulation of the cruise missile flight trajectory control*, Masters Thesis, Southeast University, Nanjing, China. (in Chinese).

H. Zargarzadeh, T. Dierks, and S. Jagannathan, (2014). Adaptive neural network-based optimal control of nonlinear continuous-time systems in strict-feedback form, *International Journal of Adaptive Control and Signal Processing*. 28(3-5), 305-324.

B. Zhou and H. S. Yan, (2013a). Financial time series forecasting based on wavelet and multi-dimensional Taylor network dynamics model, *Systems Engineering—Theory & Practice*. 33(10), 2654-2662. (in Chinese).

B. Zhou and H. S. Yan, (2013b). Time series forecasting based on the empirical mode decomposition multi-dimensional Taylor network model, *Paper presented at the 2013 Ninth International Conference on Natural Computation (ICNC)*. 1194-1198.

B. Zhou and H. S. Yan, (2014a). A dynamics model based on intermittent feedback multi-dimensional Taylor network, *ACTA Automatica Sinica*. 40(7), 1517-1521. (in Chinese).

B. Zhou and H. S. Yan, (2014b). Nonlinear system identification and prediction based on dynamics cluster multi-dimensional Taylor network model, *Control and Decision*. 29(1), 33-38. (in Chinese).

B. Zhou, (2014). *Research into modeling and forecasting of financial data based on multi-dimensional Taylor network and its extension*, PhD Thesis, Southeast University, Nanjing, China. (in Chinese).

Y. Zhou, (2016). *Research and simulation of multi-dimensional Taylor network optimal control for the flight of the plane*, Masters Thesis, Southeast University, Nanjing, China. (in Chinese).

Y. Zhu, C. Wen, H. Su, and X. M. Liu, (2014). Modular-based adaptive control of uncertain nonlinear time-varying systems with piecewise jumping parameters, *International Journal of Adaptive Control and Signal Processing*. 28(11), 1266-1289.

J. G. Ziegler and N. B. Nichols, (1942). Optimum settings for automatic controllers, *Transactions of the ASME*. 759-765.

Y. H. Zweiri, J. F. Whidborne, and L. D. Seneviratne, (2003). A three-term backpropagation algorithm, *Neurocomputing*. 50, 305-318.

Y. H. Zweiri, L. D. Seneviratne, and K. Althoefer, (2005). Stability analysis of a three-term backpropagation algorithm, *Neural Networks*. 18, 1341-1347.

13 NOTES ON CONTRIBUTORS

Hong-sen Yan received his B.S. degree in automatic control from Harbin Ship Building Engineering Institute in 1982, M.S. degree in industrial automation and Ph.D. degree in automatic control theory and application from Harbin Institute of Technology in 1989 and 1992 respectively.

From February 1982 to August 1986, he was an Assistant Engineer with Yangzhou Marine Electronic Instruments Institute. From July 1992 to June 1994, he was a Postdoctoral Fellow with the CIMS Laboratory, Department of Mechanical Engineering, Nanjing University of Aeronautics and Astronautics. From July 1994 to March 2007, he worked as associate professor and then full professor with the Research Institute of Automation, Southeast University. Since October 2007, he has been with the MOE Key Laboratory of Measurement and Control of Complex Systems of Engineering, as Superintendent of the Research Institute of Control and Optimization of Manufacturing Systems, School of Automation, Southeast University. He is the founder of multi-dimensional Taylor network (MTN), MTN optimal control, knowledge meshes and knowledgeable manufacturing theories. He has been engaged in or finished the design and development of four computer-integrated manufacturing systems in enterprises with his colleagues. He has authored or coauthored more than 240 research papers in refereed journals such as *Applied Artificial Intelligence*, *Computers and Industrial Engineering*, *COMPEL*, *Computers in Industry*, *Concurrent Engineering: Research & Applications*, *Entropy*, *European Journal of Operational Research*, *Expert Systems with Applications*, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON NEURAL NETWORKS, IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS, PART A, *IET Control Theory and Applications*, *IIE Transactions*, *Information Sciences*, *International Journal of Advanced Manufacturing Technology*, *International Journal of Computer Integrated Manufacturing*, *International Journal of Industrial Engineering*, *International Journal of Production Research*, *ISA Transactions*, *Journal of Grey System*, *Journal of Intelligent and Fuzzy Systems*, *Journal of Intelligent Manufacturing*, *Journal of Optimization Theory and Applications*, *Journal of the Operational Research Society*, *Mathematical Problems in Engineering*, *Optimal Control Applications & Methods*, *Proceedings of IMechE Part B: Journal of Engineering Manufacture*, *Robotics and Computer-Integrated Manufacturing*, and *Transactions of the Institute of Measurement and Control*. His research interests include MTN, MTN optimal control, time series forecasting, flexible manufacturing systems, computer integrated manufacturing systems (CIMS), agile manufacturing, concurrent engineering, and knowledgeable manufacturing systems.

Dr. Yan is a senior member of the Institute of Industrial Engineers, director of the Jiangsu Association of Automation in China, senior member of the Chinese Mechanical Engineering Society, and was a director of the Chinese Association for Artificial Intelligence. He won the Best Journal Paper Reward for one of the best 200 papers selected from the 2001–2005 years' papers in the 283 journals sponsored by 105 institutes, societies, institutions, and associations of the China Association for Science and Technology, and also seven other rewards from the State, the ministry, the province, or universities.

Jiao-Jun Zhang is a PhD candidate, studying and researching at the MOE Key Laboratory of Measurement and Control of Complex Systems of Engineering, School of Automation, Southeast University, China. His major research interest is MTN optimal control and adaptive control.

Qi-Ming Sun is a PhD candidate in the MOE Key Laboratory of Measurement and Control of Complex Systems of Engineering, School of Automation, Southeast University, China. His major research interest is MTN optimal control of nonlinear systems.