Design and Application of Stable Predictive Controller Using Recurrent Wavelet Neural Networks

Chi-Huang Lu, *Member, IEEE*

***Abstract*—This paper presents a design methodology for stable predictive control of nonlinear discrete-time systems via recurrent wavelet neural networks (RWNNs). This type of controller has its simplicity in parallelism to conventional generalized predictive control design and efficiency to deal with complex nonlinear dynamics. A mathematical model using RWNN is constructed, and a learning algorithm adopting a recursive least squares is employed to identify the unknown parameters in the consequent part of the RWNN. The proposed control law is derived based on the minimization of a modified predictive performance criterion. Two theorems are presented for the conditions of the stability analysis and steady-state performance of the closed-loop systems. Numerical simulations reveal that the proposed control gives satisfactory tracking and disturbance rejection performances. Experimental results for position control of a positioning mechanism show the efficacy of the proposed method with setpoint changes.**

***Index Terms*—Generalized predictive control (GPC), position control, recurrent wavelet neural network (RWNN), recursive least squares.**

I. Introduction

CLARKE *et al.* [1] proposed the generalized predictive control (GPC) method that has become one of the most popular model-based control methods both in industry and academia. This GPC strategy has been successfully implemented in many industrial applications [2]–[5]. The basic of GPC is to calculate a sequence of future control signals in such a way that it minimizes a multistage cost function defined over a prediction horizon. The index to be optimized is the expectation of a quadratic function measuring the distance between the predicted system output and some predicted reference sequence over the horizon plus a quadratic function measuring the control effort [2]. Many researchers have presented controllers involving the use of nonlinear predictive control. Nicolao *et al.* [6] presented a stabilizing predictive control with nonlinear ARX models for nonlinear discrete-time systems. Granado *et al.* [7] proposed a linear matrix inequality based model predictive controller. Lu and Tsai [8] developed fuzzy-based predictive controller for a variable-frequency oil-cooling machine. Ge *et al.* [9] addressed an adaptive predictive control using neural network (NN) for a class of pure-feedback systems in discrete time.

Feedforward NNs have been shown to obtain successful results in system identification and control [10]. Such NNs are static input/output mapping schemes that can approximate a continuous function to an arbitrary degree of accuracy. A recurrent NN (RNN) based on supervised learning which is a dynamic mapping network and is more suitable for describing dynamic systems than the NN. Of particular interested is that it can deal with time-varying input or output through its own natural temporal operation. For this ability to temporarily store information, the structure of the network is simplified. Namely, fewer nodes are required for system identification [11]. Hence, results have also been extended to RNNs [12]–[16]. In recent years, the concept of the wavelet NN (WNN) has become increasingly important. Much research has been done on applications of WNNs, which combine the capability of artificial NNs in learning from processes and the capability of wavelet decomposition [17], [18], for identification of the complex nonlinear systems. In [19], a WNN was proposed as an alternative to feedforward NNs for approximating arbitrary nonlinear functions based on the wavelet transform theory. In [28], it has been proven that a WNN can approximate any continuous function over a compact set and have high accuracy and fast learning ability. From these progresses, some researchers applied the WNN for the control of dynamic plants [20]–[22]. The recurrent WNN (RWNN) combines the properties of attractor dynamics of the RNN and good convergence performance of the WNN. The RWNN can deal with time-varying input or output through its own natural temporal operation because a mother wavelet layer composed of internal feedback neurons to capture the dynamic response of a system [24]. In most applications of GPC, linear models are used to predict the plant behavior over the prediction horizon and to evaluate future sequences of control signals. However, because the real plants usually exist a strong nonlinear behavior, the linear controller performs poorly when applied to nonlinear systems that operate over a wide range of operating conditions. Thus, the predictive control technique must be extended to incorporate the nonlinear models. The results demonstrated the feasibility and efficacy of GPC techniques for identification and control of nonlinear dynamic systems. For examples, Gu and Hu [23] presented a path tracking scheme for a carlike mobile robot based on neural predictive control, Yoo *et al.* [24] developed stable predictive control (SPC) of chaotic systems using self-RWNN, and a GPC based on self-RWNN for stable path tracking of mobile robots proposed in [25]. The method presented in this paper is

Manuscript received September 23, 2008; revised June 8, 2009. First published June 26, 2009; current version published August 12, 2009.

The author is with the Department of Electrical Engineering, Hsiuping Institute of Technology, Taichung 412, Taiwan (e-mail: chlu @mail.hit.edu.tw).

Digital Object Identifier 10.1109/TIE.2009.2025714

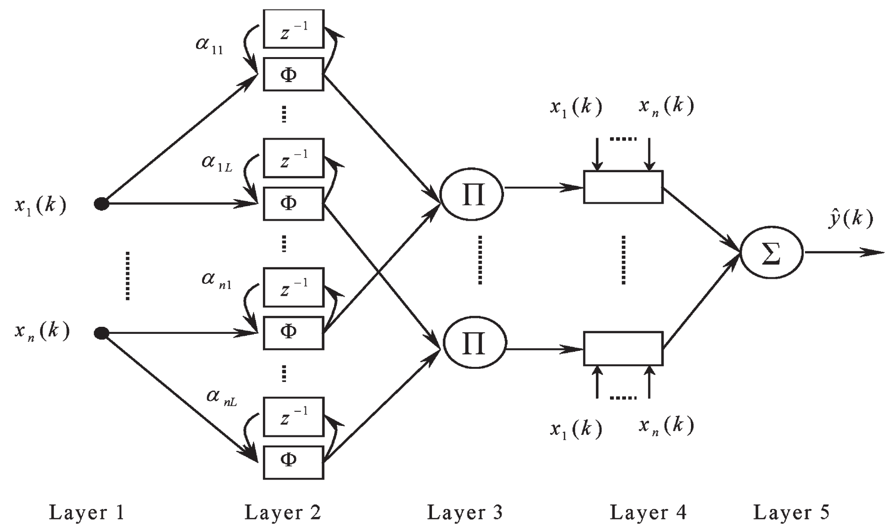

Fig. 1. Configuration of the proposed RWNN architecture.

an RWNN-based predictive controller, where the problems of the control performance, the stability analyses, and the computational complexity associated with generalized predictive controllers have not yet been addressed.

There are three objectives of this paper. The first is to propose a stable controller for a class of nonlinear discrete-time system using GPC with RWNN model, and derive the SPC controller' expression based on the minimization of a quadratic performance index. This type of controller has its simplicity in parallelism to conventional GPC design and efficiency to deal with complex nonlinear dynamics. The second is to establish the universal approximation of the RWNN using Stone–Weierstrass theorem and the convergence of the RWNN with adaptive learning rate (ALR) using Lyapunov stability theory. Two theorems are proved so that the condition for the stability of the closed-loop system can be determined and the proposed control law has zero-mean steady-state error performance. The third is to verify the feasibility and effectiveness of the proposed controller with its application to the position control of an aerostatic precision positioning stage with an air-lubricated capstan drive system.

The remainder of this paper is organized as follows. Section II presents the RWNN model for a class of nonlinear discrete-time systems. In Section III, the proposed control law is derived, and the theory analyses are studied. Section IV details the capabilities of the proposed algorithm for controlling a nonlinear system utilizing computer simulations. Experimental results for controlling a position control system to meet the desired performance specifications are presented. Section V concludes this paper.

II. RWNN

Consider a class of nonlinear discrete-time dynamic systems described by the following nonlinear autoregressive moving average model:

$$y(k) = f(y(k-1), \ldots, y(k-n_y), u(k-1), \ldots, u(k-n_u)) + \varepsilon(k) \quad (1)$$

where $u(\cdot) : Z^+ \to \Re$ and $y(\cdot) : Z^+ \to \Re$ are the system input and output, $f(\cdot) : \Re^n \to \Re$, $n = n_y + n_u$, is smooth valued nonlinear function, $n_y \in Z^+$ and $n_u \in Z^+$ are the orders of output and input, respectively, and $\varepsilon(k) \in \Re$ is a sequence of zero-mean Gaussian white noise. This section aims at developing a RWNN model for the nonlinear system (1), where a schematic diagram of the proposed RWNN structure is shown in Fig. 1.

Layer 1: This layer accepts the input variables, and then the nodes in this layer only transmit the input values to the next layer directly.

Layer 2: Each node of this layer has a mother wavelet and a self-feedback loop. In this layer, each node performs the first derivative of a Gaussian function as a mother wavelet function

$$\phi_{i\ell}(k) = \Phi\left(\frac{x_i(k) + \phi_{i\ell}(k-1)\alpha_{i\ell} - m_{i\ell}}{\sigma_{i\ell}}\right) \quad (2)$$

where $\Phi(\chi_{i\ell}) = -\chi_{i\ell}\exp(-\chi_{i\ell}^2/2)$ with $\chi_{i\ell} = (x_i(k) + \phi_{i\ell}(k-1)\alpha_{i\ell} - m_{i\ell})/\sigma_{i\ell}$. $x_i(k)$ is the ith input of the RWNN for discrete time k and $\alpha_{i\ell}$ denotes the feedback weight in the ℓth term of the ith input $x_i(k)$ to the node of wavelet layer. $m_{i\ell}$ and $\sigma_{i\ell}$ are the translation and the dilation of the wavelets, respectively.

Layer 3: Every node in this layer is labeled with Π, whose output is the product of all the incoming signals

$$\overline{\phi}_\ell(k) = \prod_{i=1}^{n} \phi_{i\ell}(k). \quad (3)$$

Layer 4: Every node ℓ in this layer is an adaptive node with a node function

$$o_\ell(k) = \overline{\phi}_\ell(k)\left(\sum_{i=1}^{n} \omega_{i\ell}x_i(k)\right) \quad (4)$$

where $\bar{\phi}_\ell(k)$ is the incoming signal from layer 3, $\omega_{i\ell}$ is the parameter of this node, and $x_i(k)$ is the ith input variable from layer 1. Parameters in this layer are referred to as consequent parameters.

Layer 5: The single node in this layer is labeled with Σ, which computes the output of RWNN as the summation of all incoming signals from layer 4

$$\hat{y}(k) = \sum_{\ell=1}^{L} o_\ell(k) = \sum_{\ell=1}^{L} \overline{\phi}_\ell(k) \left(\sum_{i=1}^{n} \omega_{i\ell} x_i(k) \right). \tag{5}$$

The aforementioned RWNN can be shown to be a universal uniform approximator for continuous functions over compact sets if it satisfies a certain condition. The condition is described as

$$\begin{aligned}\prod_{i=1}^{n} \frac{(\phi_{i\ell}(k)\alpha_{i\ell})^2}{\sigma_{i\ell}^2} \cdot \prod_{i=1}^{n} \frac{(\phi_{ij}(k)\alpha_{ij})^2}{\sigma_{ij}^2} &\neq \prod_{i=1}^{n} \frac{(m_{ij} + \phi_{i\ell}(k)\alpha_{i\ell} - m_{i\ell})^2}{\sigma_{i\ell}^2} \\ &\cdot \prod_{i=1}^{n} \frac{(m_{i\ell} + \phi_{ij}(k)\alpha_{ij} - m_{ij})^2}{\sigma_{ij}^2} \qquad \forall\, \ell \neq j. \end{aligned}\tag{6}$$

The detailed proof procedure can be referred to [30]–[32]. Then, one has the following result.

Theorem 1: Universal approximation theorem. For any real function $h : \Re^n \to \Re^m$ which is continuous on a compact set $K \subset \Re^n$ and for any given $\upsilon > 0$, there is an RWNN system $\int$ that satisfies condition (6), such that

$$\sup_{x \in K} \left\| \int(x) - h(x) \right\| < \upsilon$$

where $\| \bullet \|$ can be any norm.

This theorem shows that if the RWNN has a sufficiently large number of mother wavelets, then it can approximate any continuous function in $C(\Re^n)$ over a compact subset of $\Re^n$. For system identification, the theorem means that for any given continuous output trajectory $y(t)$ of any nonlinear dynamic system over any compact time-interval $t \in [t_0, T]$, the output $\hat{y}(t)$ of the RWNN can be approximate $y(t)$ uniformly with arbitrarily high accuracy.

To derive the learning algorithm for the parameters $m_{i\ell}$, $\sigma_{i\ell}$, and $\alpha_{i\ell}$ in the precondition part of the RWNN, the error function $E(k)$ of the RWNN is defined by

$$E(k) = \frac{1}{2} \left(y(k) - \hat{y}(k) \right)^2. \tag{7}$$

By applying the gradient decent method, and then the updating laws for $m_{i\ell}$, $\sigma_{i\ell}$, and $\alpha_{i\ell}$ are obtained by

$$\begin{aligned} m_{i\ell}(k+1) &= m_{i\ell}(k) - \eta \frac{\partial E(k)}{\partial m_{i\ell}} \\ &= m_{i\ell}(k) + \eta \left(y(k) - \hat{y}(k) \right) \frac{\partial \hat{y}(k)}{\partial m_{i\ell}} \end{aligned}\tag{8}$$

$$\begin{aligned} \sigma_{i\ell}(k+1) &= \sigma_{i\ell}(k) - \eta \frac{\partial E(k)}{\partial \sigma_{i\ell}} \\ &= \sigma_{i\ell}(k) + \eta \left(y(k) - \hat{y}(k) \right) \frac{\partial \hat{y}(k)}{\partial \sigma_{i\ell}} \end{aligned}\tag{9}$$

$$\begin{aligned} \alpha_{i\ell}(k+1) &= \alpha_{i\ell}(k) - \eta \frac{\partial E(k)}{\partial \vartheta_{i\ell}} \\ &= \alpha_{i\ell}(k) + \eta \left(y(k) - \hat{y}(k) \right) \frac{\partial \hat{y}(k)}{\partial \alpha_{i\ell}} \end{aligned}\tag{10}$$

where

$$\begin{aligned} \frac{\partial \hat{y}(k)}{\partial m_{i\ell}} &= \frac{\partial \hat{y}(k)}{\partial \overline{\phi}_\ell(k)} \frac{\partial \overline{\phi}_\ell(k)}{\partial \phi_{i\ell}(k)} \frac{\partial \phi_{i\ell}(k)}{\partial m_{i\ell}} \\ &= \left(\sum_{i=1}^{n} \omega_{i\ell} x_i(k) \right) \overline{\phi}_\ell(k) \frac{(\chi_{i\ell}^2 - 1)}{\chi_{i\ell}\sigma_{i\ell}} \end{aligned}\tag{11}$$

$$\begin{aligned} \frac{\partial \hat{y}(k)}{\partial \sigma_{i\ell}} &= \frac{\partial \hat{y}(k)}{\partial \overline{\phi}_\ell(k)} \frac{\partial \overline{\phi}_\ell(k)}{\partial \phi_{i\ell}(k)} \frac{\partial \phi_{i\ell}(k)}{\partial \sigma_{i\ell}} \\ &= \left(\sum_{i=1}^{n} \omega_{i\ell} x_i(k) \right) \overline{\phi}_\ell(k) \frac{(\chi_{i\ell}^2 - 1)}{\sigma_{i\ell}} \end{aligned}\tag{12}$$

$$\begin{aligned} \frac{\partial \hat{y}(k)}{\partial \alpha_{i\ell}} &= \frac{\partial \hat{y}(k)}{\partial \overline{\phi}_\ell(k)} \frac{\partial \overline{\phi}_\ell(k)}{\partial \phi_{i\ell}(k)} \frac{\partial \phi_{i\ell}(k)}{\partial \alpha_{i\ell}} \\ &= \left(\sum_{i=1}^{n} \omega_{i\ell} x_i(k) \right) \overline{\phi}_\ell(k) \phi_{i\ell}(k-1) \frac{(\chi_{i\ell}^2 - 1)}{\chi_{i\ell}\sigma_{i\ell}}. \end{aligned}\tag{13}$$

Next, the recursive least squares method is adopted to identify the unknown parameters in the consequent part of the RWNN. Thus, $\hat{y}(k)$ can be rewritten as

$$\hat{y}(k) = \zeta^{\mathrm{T}}(k)\theta \tag{14}$$

where

$$\begin{aligned} \zeta(k) &= [\, \zeta_1^{\mathrm{T}}(k) \quad \zeta_2^{\mathrm{T}}(k) \quad \cdots \quad \zeta_L^{\mathrm{T}}(k) \,]^{\mathrm{T}} \\ \theta &= [\, \theta_1^{\mathrm{T}} \quad \theta_2^{\mathrm{T}} \quad \cdots \quad \theta_L^{\mathrm{T}} \,]^{\mathrm{T}} \\ \zeta_\ell(k) &= [\, \overline{\phi}_\ell x_1(k) \quad \overline{\phi}_\ell x_2(k) \quad \cdots \quad \overline{\phi}_\ell x_n(k) \,]^{\mathrm{T}} \\ \theta_\ell &= [\, \omega_{1\ell} \quad \omega_{2\ell} \quad \cdots \quad \omega_{n\ell} \,]^{\mathrm{T}}. \end{aligned}$$

Define the error covariance matrix $P(k) = (\Theta^{\mathrm{T}}(k)\Theta(k))^{-1}$ where $\Theta(k) = \sum_k \zeta^{\mathrm{T}}(k)\zeta(k)$, and assume that the matrix $\Theta(k)$ has full rank for $\forall\, k \geq 0$. Given an initial error estimate $\theta(0)$ and an initial error covariance matrix $P(0)$, the least-squares estimate of $\theta(k)$ satisfies the following recursive equations:

$$\theta(k) = \theta(k-1) + \frac{P(k-1)\zeta^{\mathrm{T}}(k) \left(y(k) - \zeta(k)\theta(k-1) \right)}{1 + \zeta(k)P(k-1)\zeta^{\mathrm{T}}(k)} \tag{15}$$

$$P(k) = P(k-1) - \frac{P(k-1)\zeta^{\mathrm{T}}(k)\zeta(k)P(k-1)}{1 + \zeta(k)P(k-1)\zeta^{\mathrm{T}}(k)}. \tag{16}$$

Note that the aforementioned recursive least-squares equations can be easily derived by the method in [26].

Assume that the parameters $m_{i\ell}$, $\sigma_{i\ell}$, $\alpha_{i\ell}$, and $\omega_{i\ell}$ for the RWNN (5) are updated along with (8)–(10), (15), and (16).

Then, the proposed RWNN algorithm is convergent, provided that its learning rate η satisfies the following condition:

$$0 < \eta < \frac{2}{\sum_{\ell=1}^{L}\sum_{i=1}^{n}\left(\left(\frac{\partial \hat{y}(k)}{\partial m_{i\ell}}\right)^2 + \left(\frac{\partial \hat{y}(k)}{\partial \sigma_{i\ell}}\right)^2 + \left(\frac{\partial \hat{y}(k)}{\partial \alpha_{i\ell}}\right)^2 + \left(\frac{\partial \hat{y}(k)}{\partial \omega_{i\ell}}\right)^2\right)}. \tag{17}$$

The convergent condition (17) of the RWNN algorithm can be shown by choosing a Lyapunov function $L(k) = (y(k) - \hat{y}(k))^2$ and the detailed proof procedure can be referred to [14].

To guarantee this selecting learning rate η inside the stable region, an ALR for training the weights of the RWNN is given by

$$\eta = 1 \Bigg/ \sum_{\ell=1}^{L}\sum_{i=1}^{n}\left(\left(\frac{\partial \hat{y}(k)}{\partial m_{i\ell}}\right)^2 + \left(\frac{\partial \hat{y}(k)}{\partial \sigma_{i\ell}}\right)^2 + \left(\frac{\partial \hat{y}(k)}{\partial \alpha_{i\ell}}\right)^2 + \left(\frac{\partial \hat{y}(k)}{\partial \omega_{i\ell}}\right)^2\right) \tag{18}$$

where $\partial \hat{y}(k)/\partial m_{i\ell}$, $\partial \hat{y}(k)/\partial \sigma_{i\ell}$, and $\partial \hat{y}(k)/\partial \alpha_{i\ell}$ are obtained from (11)–(13), and $\partial \hat{y}(k)/\partial \omega_{i\ell} = \partial(\zeta^{\mathrm{T}}(k)\theta)/\partial \omega_{i\ell} = \bar{\phi}_\ell(k)x_i(k)$.

III. Derivation and Analysis of Predictive Control Law

This section is devoted to the derivation of the proposed controller based on the RWNN model for this class of nonlinear discrete-time systems. The proposed control law is derived to minimize the mathematical expectation of the following predictive performance criterion:

$$J(k) = E\left\{\sum_{j=1}^{N_{\mathrm{p}}}\left(D(z^{-1})y(k+j) - \beta N(z^{-1})r(k+j)\right)^2\right\} \tag{19}$$

where $D(z^{-1}) = 1 - \sum_{i=1}^{m} d_i z^{-i}$ and $N(z^{-1}) = \sum_{i=1}^{m} n_i z^{-i}$. N_{p} is the prediction output horizon, β is the weight parameter, and $r(k+j) \in \Re$ is a known bounded reference input for the discrete time $k+j$. In general, N_{p} is chosen to encompass all the responses that are significantly affected by the present control, and more typically, $N_{\mathrm{p}}T_{\mathrm{s}}$ is the same magnitude as the rise time of the controlled system using the sampling time T_{s} [26].

To design the RWNN-based predictive controller, the global operation of the nonlinear system (1) is divided into several local operating regions. Within each local region, a linear ARMA model is used to represent the local system behavior [29]. In doing so, one defines

$$\{y(k-1), \ldots, y(k-n_y), u(k-1), \ldots, u(k-n_u)\} \equiv \left\{x_1(k), \ldots, x_{n_y}(k), x_{n_y+1}(k), \ldots, x_n(k)\right\} \tag{20}$$

$$\left\{a_{1\ell}, \ldots, a_{n_y\ell}, b_{1\ell}, \ldots, b_{n_u\ell}\right\} \equiv \left\{\omega_{1\ell}, \ldots, \omega_{n_y\ell}, \omega_{(n_y+1)\ell}, \ldots, \omega_{n\ell}\right\}. \tag{21}$$

Then, the RWNN model (5) for the nonlinear system (1) can be expressed mathematically as

$$y(k) = \sum_{\ell=1}^{L}\overline{\phi}_\ell(k)\left(\overline{a}_\ell(z^{-1})y(k-1) + \overline{b}_\ell(z^{-1})u(k-1)\right) + \varepsilon(k) \tag{22}$$

where

$$\overline{a}_\ell(z^{-1}) = a_{1\ell} + a_{2\ell}z^{-1} + \cdots + a_{n_y\ell}z^{-(n_y-1)}$$
$$\overline{b}_\ell(z^{-1}) = b_{1\ell} + b_{2\ell}z^{-1} + \cdots + b_{n_u\ell}z^{-(n_u-1)}.$$

To derive the GPC law and to find the j step-ahead prediction of $y(k)$, the RWNN model (22) is rewritten as

$$a(z^{-1})y(k) = b(z^{-1})u(k-1) + \varepsilon(k) \tag{23}$$

where

$$a(z^{-1}) = 1 - a_1z^{-1} - a_2z^{-2} - \cdots - a_{n_y}z^{-n_y}, \qquad a_i = \sum_{\ell=1}^{L}\overline{\phi}_\ell(k)a_{i\ell}$$
$$b(z^{-1}) = b_1 + b_1z^{-1} + \cdots + b_{n_u}z^{-(n_u-1)}, \qquad b_i = \sum_{\ell=1}^{L}\overline{\phi}_\ell(k)b_{i\ell}.$$

In order to optimize the cost function $J(k)$, the prediction $y(k+j)$ for $j \geq 1$ and $j \leq N_{\mathrm{p}}$ will be obtained. Consider the following Diophantine equation:

$$D(z^{-1}) = \Delta e_j(z^{-1})a(z^{-1}) + z^{-j}f_j(z^{-1}). \tag{24}$$

The polynomials $e_j(z^{-1})$ and $f_j(z^{-1})$ are uniquely defined with degrees $j-1$ and n_y, respectively. They can be obtained dividing $D(z^{-1})$ by $\Delta a(z^{-1}) \equiv (1-z^{-1})a(z^{-1})$ until the remainder can be factorized as $z^{-j}f_j(z^{-1})$. The quotient of the division is the polynomial $e_j(z^{-1})$.

Premultiplying (23) by $\Delta e_j(z^{-1})z^j$ gives the following equality:

$$\Delta e_j(z^{-1})a(z^{-1})y(k+j) = \Delta e_j(z^{-1})b(z^{-1})u(k+j-1) + \Delta e_j(z^{-1})\varepsilon(k+j).$$

By using (24), the aforementioned equation becomes

$$D(z^{-1})y(k+j) = f_j(z^{-1})y(k) + g_j(z^{-1})\Delta u(k+j-1) + \Delta e_j(z^{-1})\varepsilon(k+j) \tag{25}$$

where

$$g_j(z^{-1}) = e_j(z^{-1})b(z^{-1}). \tag{26}$$

Thus, the j step-ahead prediction of $y(k)$ is now given by

$$D(z^{-1})\hat{y}(k+j) = f_j(z^{-1})y(k) + g_j(z^{-1})\Delta u(k+j-1). \tag{27}$$

With the j step-ahead prediction of $y(k)$, the optimal control which minimizes the cost function $J(k)$ should be simultaneously minimize the following new cost function $\breve{J}(k)$ given by

$$\begin{aligned}\breve{J}(k) &= \sum_{j=1}^{N_{\mathrm{p}}} \left(D(z^{-1})\hat{y}(k+j) - \beta N(z^{-1})r(k+j)\right)^2 \\ &= \left(Fy(k) + GU + \Lambda - \beta N(z^{-1})R\right)^{\mathrm{T}} \\ &\quad \times \left(Fy(k) + GU + \Lambda - \beta N(z^{-1})R\right) \end{aligned} \tag{28}$$

where

$$F = [\, f_1(z^{-1}) \quad f_2(z^{-1}) \quad \cdots \quad f_{N_{\mathrm{p}}}(z^{-1}) \,]^{\mathrm{T}}$$

$$G = \begin{bmatrix} g_{1,0} & 0 & \cdots & 0 \\ g_{2,1} & g_{2,0} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_{\mathrm{p}},N_{\mathrm{p}}-1} & g_{N_{\mathrm{p}},N_{\mathrm{p}}-2} & \cdots & g_{N_{\mathrm{p}},0} \end{bmatrix}$$

$$U = [\, \Delta u(k) \quad \Delta u(k+1) \quad \cdots \quad \Delta u(k+N_{\mathrm{p}}-1) \,]^{\mathrm{T}}$$

$$R = [\, r(k+1) \quad r(k+2) \quad \cdots \quad r(k+N_{\mathrm{p}}) \,]^{\mathrm{T}}$$

and Λ is expressed as shown at the bottom of the page.

By using $u(k+N_{\mathrm{p}}-1) = \cdots = u(k+1) = u(k)$, the proposed controller takes a less computational requirement to overcome a major obstacle to the solution of the conventional GPC that demands for large matrix inversion and numerous matrix multiplications in solving the optimization problems. The cost function $\breve{J}(k)$ in (27) can be equivalently expressed in following form:

$$\begin{aligned}\overline{J}(k) &= \left(Fy(k) + \overline{G}\Delta u(k) + \overline{\Lambda} - \beta N(z^{-1})R\right)^{\mathrm{T}} \\ &\quad \times \left(Fy(k) + \overline{G}\Delta u(k) + \overline{\Lambda} - \beta N(z^{-1})R\right) \end{aligned} \tag{29}$$

where

$$\overline{G} = [\, g_{1,0} \quad g_{2,0} \quad \cdots \quad g_{N_{\mathrm{p}},0} \,]^{\mathrm{T}}$$

and $\overline{\Lambda}$ is expressed as shown at the bottom of the page.

Because the cost function $\overline{J}(k)$ is quadratic in $\Delta u(k)$, a minimum solution for $\Delta u(k)$ is easily obtained from

$$\frac{\partial \overline{J}(k)}{\partial\left(\Delta u(k)\right)} = 0 \tag{30}$$

which yields the following equation:

$$\overline{G}^T \left(Fy(k) + \overline{G}\Delta u(k) + \overline{\Lambda} - \beta N(z^{-1})R\right) = 0. \tag{31}$$

Consequently, the increment control output can be obtained by

$$\Delta u(k) = \frac{\overline{G}^T \left(\beta N(z^{-1})R - Fy(k) - \overline{\Lambda}\right)}{\overline{G}^T\overline{G}}.$$

Note that the control signal must be limited to satisfy the input constraint because in practice the actuators have a limited range of action. Hence, the control signal $u(k)$ and the weight parameter β are expressed by

$$\begin{aligned} u(k) &= u(k-1) + \frac{\overline{G}^T \left(\beta N(z^{-1})R - Fy(k) - \overline{\Lambda}\right)}{\overline{G}^T\overline{G}} \\ \beta &= \begin{cases} \frac{\overline{G}(u_{\max} - u(k-1)) + Fy(k) + \overline{\Lambda}}{N(z^{-1})R}, & \text{if } u(k) > u_{\max} \\ 1, & \text{if } |u(k)| \le u_{\max} \\ \frac{\overline{G}(-u_{\max} - u(k-1)) + Fy(k) + \overline{\Lambda}}{N(z^{-1})R}, & \text{if } u(k) < -u_{\max} \end{cases} \end{aligned} \tag{32}$$

where $|u(k)|$ denotes the absolute value of $u(k)$ and $u_{\max} \in \Re$ is the upper bound for the control input $u(k)$.

The following theorems state that the resulting overall system can be stable and the aforementioned control algorithm has zero-mean steady-state tracking errors.

Theorem 2: Assume that the RWNN model (23) is controlled by (32), the upper bounds for n_y and n_u are known, the estimated parameters of $m_{i\ell}$, $\sigma_{i\ell}$, $\alpha_{i\ell}$, and $\omega_{i\ell}$ are convergent and uniformly bounded. Then, the convergence of the closed-loop system is guaranteed if a transfer function $T(z^{-1}) = N(z^{-1})/D(z^{-1})$ is chosen to satisfy a stable model.

Proof: Multiplying $\overline{G}^T[\, \Delta e_1(z^{-1})z \quad \Delta e_2(z^{-1})z^2 \quad \cdots \quad \Delta e_{N_{\mathrm{p}}}(z^{-1})z^{N_{\mathrm{p}}}]^{\mathrm{T}}$ into both sides of the RWNN model (23) and substituting the control law (32) into (23) yields the following equation:

$$\begin{aligned} &\overline{G}^T \begin{bmatrix} \Delta e_1(z^{-1})za(z^{-1}) \\ \Delta e_2(z^{-1})z^2a(z^{-1}) \\ \vdots \\ \Delta e_{N_{\mathrm{p}}}(z^{-1})z^{N_{\mathrm{p}}}a(z^{-1}) \end{bmatrix} y(k) \\ &= \overline{G}^T \begin{bmatrix} \beta N(z^{-1})z \\ \beta N(z^{-1})z^2 \\ \vdots \\ \beta N(z^{-1})z^{N_{\mathrm{p}}} \end{bmatrix} r(k) - \overline{G}^T \begin{bmatrix} f_1(z^{-1}) \\ f_2(z^{-1}) \\ \vdots \\ f_{N_{\mathrm{p}}}(z^{-1}) \end{bmatrix} y(k) \end{aligned}$$

$$\Lambda = \left[\sum_{i=1}^{n_u-1} g_{1,i}\Delta u(k-i) \quad \sum_{i=2}^{n_u} g_{2,i}\Delta u(k-i) \quad \cdots \quad \sum_{i=N_{\mathrm{p}}-1}^{N_{\mathrm{p}}+n_u-2} g_{N_{\mathrm{p}},i}\Delta u(k-i) \right]^{\mathrm{T}}$$

$$\overline{\Lambda} = \left[\sum_{i=1}^{n_u-1} g_{1,i}\Delta u(k-i) \quad \sum_{i=1}^{n_u} g_{2,i}\Delta u(k-i) \quad \cdots \quad \sum_{i=1}^{N_{\mathrm{p}}+n_u-2} g_{N_{\mathrm{p}},i}\Delta u(k-i) \right]^{\mathrm{T}}$$

$$+\,\overline{G}^T\begin{bmatrix}\Delta e_1(z^{-1})z\\ \Delta e_2(z^{-1})z^2\\ \vdots\\ \Delta e_{N_\mathrm{P}}(z^{-1})z^{N_\mathrm{P}}\end{bmatrix}\varepsilon(k) \tag{33}$$

which leads to the following polynomial equation:

$$\sum_{j=1}^{N_\mathrm{P}} g_{j,0}z^j\left(\Delta e_j(z^{-1})a(z^{-1})+z^{-j}f_j(z^{-1})\right)y(k) = \sum_{j=1}^{N_\mathrm{P}} g_{j,0}z^j\beta N(z^{-1})r(k)+\sum_{j=1}^{N_\mathrm{P}} g_{j,0}z^j\Delta e_j(z^{-1})\varepsilon(k). \tag{34}$$

Hence, the resulting closed-loop system is

$$\left(\Delta e_j(z^{-1})a(z^{-1})+z^{-j}f_j(z^{-1})\right)y(k) = \beta N(z^{-1})r(k)+\Delta e_j(z^{-1})\varepsilon(k). \tag{35}$$

By using the Diophantine equation (24), (35) becomes

$$y(k)=\frac{\beta N(z^{-1})}{D(z^{-1})}r(k)+\frac{\Delta e_j(z^{-1})}{D(z^{-1})}\varepsilon(k). \tag{36}$$

Since $T(z^{-1})=N(z^{-1})/D(z^{-1})$ can be given stable model, the roots of the closed-loop characteristic equation $D(z^{-1})=\Delta e_j(z^{-1})a(z^{-1})+z^{-j}f_j(z^{-1})$ lie within the unit circle of the z-plane. Then, the closed-loop system is shown to be stable.

Theorem 3: Assume that the SPC law (32) is applied to the RWNN model (23), the estimated parameters of $m_{i\ell}$, $\sigma_{i\ell}$, $\alpha_{i\ell}$, and $\omega_{i\ell}$ are convergent and uniformly bounded, the dc gain of $T(z^{-1})$ is one, and all setpoints are constant, i.e., $r(k)=r$. Then, the closed-loop system has the following property:

$$\lim_{k\to\infty} E\{y(k)-r\}=0. \tag{37}$$

Proof: By using the resulting closed-loop system (36), (37) becomes

$$\lim_{k\to\infty} E\{y(k)-r\} = \lim_{k\to\infty} E\left\{\frac{\beta N(z^{-1})r(k)}{D(z^{-1})}+\frac{\Delta e_j(z^{-1})\varepsilon(k)}{D(z^{-1})}-r\right\}.$$

Since the closed-loop system is stable from *Theorem 2*, then the parameter β of the control signal $u(k)$ can be 1 when it is working around the equilibrium point of the control system. Thus, one has

$$\lim_{k\to\infty} E\{y(k)-r\}=\lim_{k\to\infty} E\left\{T(z^{-1})r(k)\right\} + \lim_{k\to\infty} E\left\{\frac{\Delta e_j(z^{-1})}{D(z^{-1})}\varepsilon(k)\right\}-r. \tag{38}$$

With the properties $T(z^{-1})|_{z=1}=T(1)=1$ and $E\{\varepsilon(k)\}=0$, (38) is expressed by

$$\begin{aligned}\lim_{k\to\infty} E\{y(k)-r\} &= T(z^{-1})\big|_{z=1}r(k)+\left.\frac{\Delta e_j(z^{-1})}{D(z^{-1})}\right|_{z=1}E\{\varepsilon(k)\}-r\\ &= T(1)r+\frac{\Delta e_j(1)}{D(1)}0-r=0\end{aligned}$$

which completes the proof of Theorem 3.

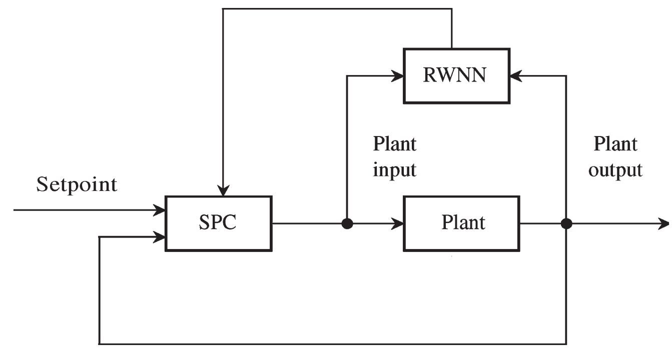

Fig. 2. Block diagram of the proposed control system.

IV. Illustrative Examples

In this section, three illustrative examples are provided to demonstrate the performance of the proposed SPC controller. The examples also show the effect of setpoint changes and load disturbances on the control systems employing the proposed controller.

A. Adaptive Control Algorithm

Fig. 2 shows a conceptual diagram of the proposed control system. An adaptive control procedure is employed recursively to update the RWNN and therefore the proposed controller's parameters. The computations at each time instant k can be outlined as the following steps.

Step 1) Measure the plant output $y(k)$.
Step 2) Update $m_{i\ell}$, $\sigma_{i\ell}$, $\alpha_{i\ell}$, $\omega_{i\ell}$, and η of the RWNN using (8)–(10), (15) and (18).
Step 3) Compute the control signal $u(k)$ via (32).
Step 4) Output the control signal to the controlled plant.
Step 5) Repeat steps 1)–4).

B. Simulation of a Nonlinear Discrete-Time System

Example 1: Consider the control of a nonlinear discrete-time dynamical system given by [27] and [28]. The system model is described as follows:

$$\begin{aligned}y(k) = {}&0.9722y(k-1)+0.3578u(k-1)-0.1295u(k-2)\\ &-0.3103y(k-1)u(k-1)-0.04228y^2(k-2)\\ &+0.1663y(k-2)u(k-2)\\ &-0.03259y^2(k-1)y(k-2)\\ &-0.3513y^2(k-1)u(k-2)\\ &+0.3084y(k-1)y(k-2)u(k-2)\\ &+0.1087y(k-2)u(k-1)u(k-2)+v(k).\end{aligned} \tag{39}$$

The objective is to make the system output $y(k)$ tracks a reference input using the proposed controller, and the reference input $r(k)$ and the external disturbances $v(k)$ specified by

$$r(k)=\begin{cases}1, & 0<k\le 200\\ 0, & 200<k\le 400\end{cases}$$

$$v(k)=\begin{cases}0, & 0<k\le 100\\ 0.05, & 100<k\le 300\\ 0.2, & 300<k\le 400.\end{cases}$$

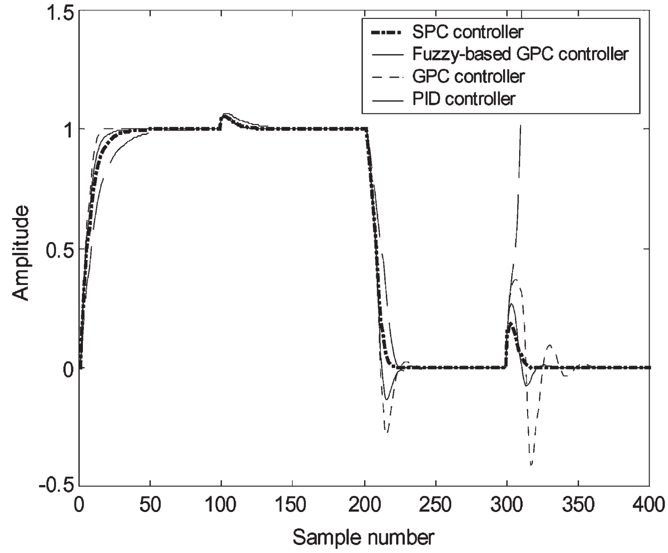

Fig. 3. Output responses (under external disturbances).

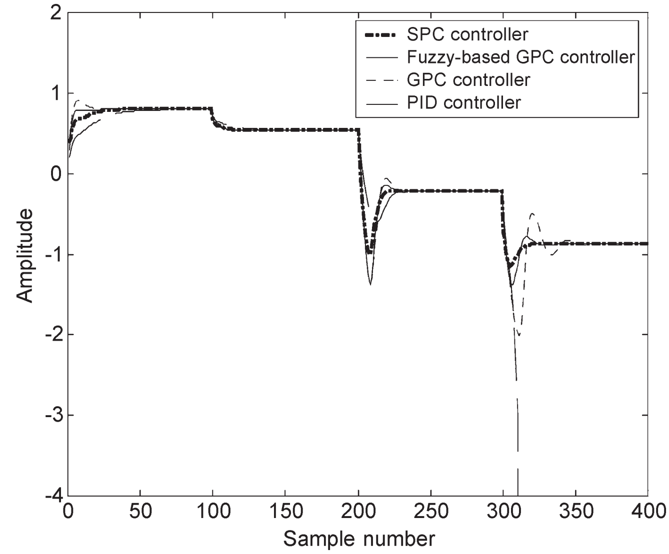

Fig. 4. Control signals.

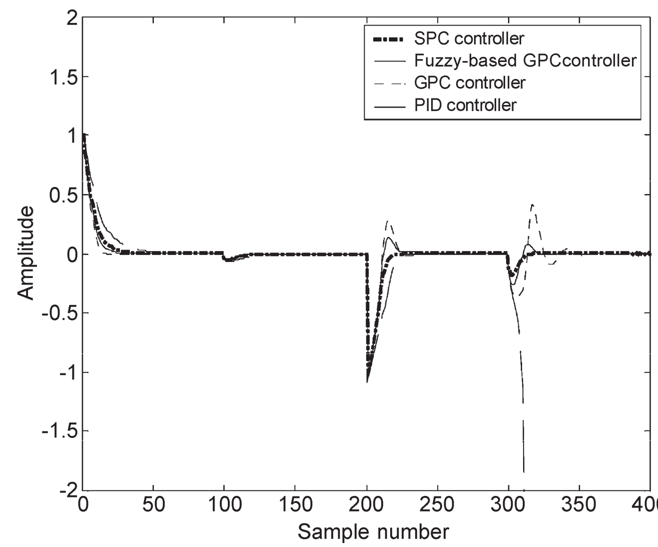

Fig. 5. Tracking errors.

The system model (39) indicates that the input variables of the RWNN model are specified by $\{y(k-1), y(k-2), u(k-1), u(k-2)\}$. Train the network parameters of the RWNN using input–output data of the nonlinear system, the key parameter of the RWNN is chosen as $L = 3$ that the RWNN model can be effective under this nonlinear system. The prediction horizon of the proposed control law was selected as $N_\mathrm{p} = 20$, and the desired closed-loop dynamic polynomials $N(z^{-1})$ and $D(z^{-1})$ were chosen as

$$N(z^{-1}) = 0.1z^{-1} \quad D(z^{-1}) = 1 - 0.9z^{-1}. \tag{40}$$

To evaluate the performance of the control system, one defines the root-mean-square error (rmse) as follows:

$$\Psi = \sqrt{\frac{\sum_{k=1}^{S} (r(k) - y(k))^2}{S}} \tag{41}$$

where S is the total number of samples.

It is desirable to compare a proposed SPC controller, a fuzzy-based GPC controller [8], conventional GPC controller (predictive output horizon $N_\mathrm{p} = 20$, control horizon $N_u = 1$ and control weighting factor $\lambda = 10$), and a velocity-type PID controller ($K_P = 1.5$, $K_I = 0.2$, and $K_D = 0.01$).

Figs. 3–5 show the output responses, the control signals, and tracking errors of the SPC controller, the fuzzy-based GPC controller, the conventional GPC controller, and PID controller under setpoint changes and constant load disturbances, respectively. The results indicate that both the SPC controller and fuzzy-based GPC controller have the short setting times and the small maximum overshoots, which exhibit that the controllers have good disturbance rejections. In the presence of disturbances $v(k)$ for the duration $k \geq 300$, Fig. 3 shows that the GPC controller has a big maximum overshoot and a long setting time. The PID controller has an unstable tracking performance, which represents that the fixed PID controller is not capable of controlling nonlinear system under disturbance changes.

TABLE I
SIMULATION RESULTS OF THE CONTROL SYSTEMS

Controllers	Network parameters	Maximum Overshoots	RMSE
SPC	24	0.1836	0.0866
Fuzzy-based GPC	24	0.2655	0.1192
GPC	0	0.3696	0.1649
PID	0	∞	∞

In Table I, the rmse value for the SPC control system is less than that of the other approaches. As can be seen, the performance of the SPC controller is significantly better than those that can be obtained with the fuzzy-based GPC controller, the conventional GPC controller, and the fixed PID controller.

Example 2: The plant to be controlled is described by the following difference equation [10], [21]:

$$y(k) = \frac{y(k-1)y(k-2)\left(y(k-1) + 2.5\right)}{1 + y^2(k-1) + y^2(k-2)} + u(k-1). \tag{42}$$

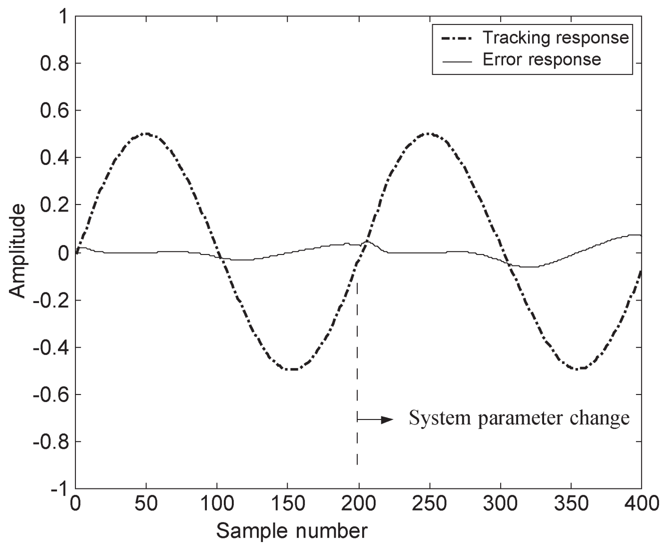

Fig. 6. Tracking/error responses (under parameter variations).

For the control of the plant, the time-varying reference input $r(k)$, prediction horizon N_{p}, and the desired closed-loop dynamic polynomials $N(z^{-1})$ and $D(z^{-1})$ are used

$$r(k) = 0.5\sin(k\pi/100), \qquad 0 < k \leq 400$$
$$N_{\mathrm{p}} = 3 \quad N(z^{-1}) = 0.7z^{-1} \quad D(z^{-1}) = 1 - 0.3z^{-1}.$$

To test the robustness of the SPC controller, the simulation was performed in which the system parameters were artificially changed. The system parameters a_i and b_i with $\pm 10\%$ variations were applied to the plant after the 200th sample. Fig. 6 shows that the proposed SPC controller could adapt to change in the plant parameters and that the error response remained within 0.1.

The results of illustrative examples reveal that the RWNN-based SPC controller demonstrates the satisfied tracking performance and system robustness for this class of nonlinear discrete-time systems.

C. Real-Time Application to a Position Control System

Fig. 7 shows a schematic diagram of the position control system, while Fig. 8 shows a picture of experimental mechanism in this paper. The plant was mounted on an aerostatic bearing slide and driven by the air-lubricated capstan drive. The slide has a travel stroke of 150 mm. A flexible coupling has been used for avoiding misalignment error between slide and capstan drive. To obtain high stability characteristics and reduce geometric error, the whole system was supported by a granite block base (900 mm $\times$ 600 mm $\times$ 150 mm). The DMC-1842 motion controller (Galil Motion Control, Inc.) for control of the positioning mechanism is inserted in the slot of an industrial computer. The power amplifier (MSA 12-80) converts a ± 10 V signal from the controller into current to drive the brushless dc motor, the resolution of the linear encoder is 1 nm, and the sampling period in this control system adopted here is 10 ms. The industrial computer is responsible for editing, compiling, and running C program codes for the controller.

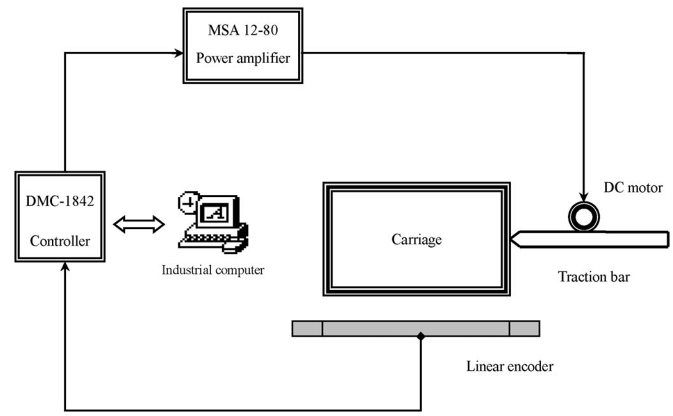

Fig. 7. Schematic diagram of the position control system.

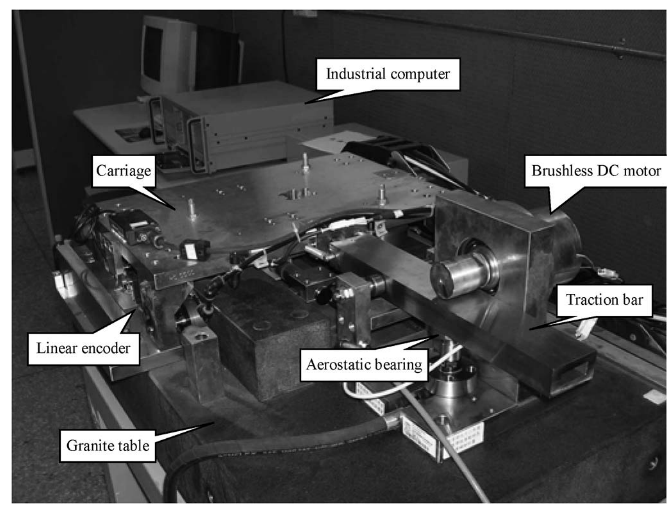

Fig. 8. Picture of experimental mechanism.

The first experiment was conducted to test the displacement regulation and disturbance rejection capability of the proposed method. The carriage of the experiment mechanism was put into steel with 500 g as load disturbance. The parameter and the input variable obtained from the experimental input/output data of the position control system are $L = 3$ and $\{y(k-1)\ u(k-1)\}$. The prediction horizon of the SPC control law was selected as $N_{\mathrm{p}} = 5$, and the desired closed-loop dynamic polynomials were chosen as $D(z^{-1}) = 1 - d_1 z^{-1} = 1 - 0.97z^{-1}$ and $N(z^{-1}) = n_1 z^{-1} = 0.03z^{-1}$. Consequently, the control signal $u(k)$ for the controlled plant is formulated as

$$\begin{aligned} u(k) = u(k-1) &+ \left(\sum_{j=1}^{N_{\mathrm{p}}} g_{j,o} n_1 r(k+j-1) \right. \\ &- \sum_{j=1}^{N_{\mathrm{p}}} g_{j,0} f_{j,0} y(k) - \sum_{j=1}^{N_{\mathrm{p}}} g_{j,0} f_{j,1} y(k-1) \\ &\left. - \sum_{j=1}^{N_{\mathrm{p}}} \sum_{i=1}^{j+n_u-2} g_{j,0} g_{j,i} \Delta u(k-i) \right) \Bigg/ \sum_{j=1}^{N_{\mathrm{p}}} g_{j,0}^2 \quad (43) \end{aligned}$$

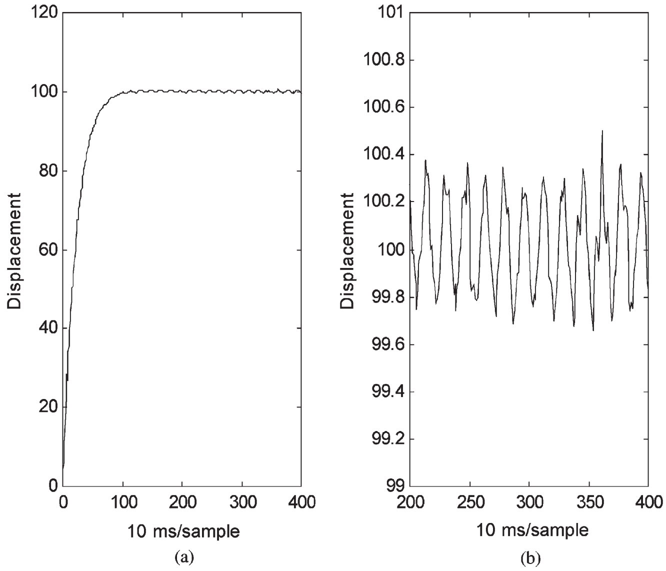

Fig. 9. (a) Step response (under external disturbance). (b) Detail behavior of the position control system for $200 < k \leq 400$.

where

$$g_{j,0} = \sum_{\ell=1}^{L} \omega_\ell(k) b_{0\ell}$$

$$g_{j,i} = \left(\sum_{r=0}^{i} \left(\sum_{\ell=1}^{L} \omega_\ell(k) a_{1\ell} \right)^{r} - \left(\sum_{r=1}^{i} \left(\sum_{\ell=1}^{L} \omega_\ell(k) a_{1\ell} \right)^{(r-1)} \right) d_1 \right) \left(\sum_{\ell=1}^{L} \omega_\ell(k) b_{0\ell} \right)$$

$$f_{j,0} = \sum_{r=0}^{j} \left(\sum_{\ell=1}^{L} \omega_\ell(k) a_{1\ell} \right)^{r} - \left(\sum_{r=0}^{j-1} \left(\sum_{\ell=1}^{L} \omega_\ell(k) a_{1\ell} \right)^{r} \right) d_1$$

$$f_{j,1} = -\sum_{r=1}^{j} \left(\sum_{\ell=1}^{L} \omega_\ell(k) a_{1\ell} \right)^{r} + \left(\sum_{r=1}^{j-1} \left(\sum_{\ell=1}^{L} \omega_\ell(k) a_{1\ell} \right)^{r} \right) d_1.$$

Fig. 9 shows the 100-μm step response and the steady-state error, respectively, of the position control system. The resultant maximum overshoot was less than 5 μm, and the steady-state errors remained within ± 1 μm. The results indicate that the proposed controller is capable of giving a good closed-loop output response.

The second experiment was performed to examine tracking capability of the SPC controller. The experiment is conducted using employing the time-varying command $r(k)$, the prediction horizon N_p, and the desired closed-loop dynamic polynomials $N(z^{-1})$ and $D(z^{-1})$

$$r(k) = 100 \sin(0.001 \times k)\ \mu\mathrm{m}, \qquad 0 < k \leq 1000$$
$$N_\mathrm{p} = 5 \quad N(z^{-1}) = 0.6 z^{-1} \quad D(z^{-1}) = 1 - 0.4 z^{-1}.$$

The identified parameters a_i and b_i with $\pm 10\%$ artificial variations were applied to the position control system. Figs. 10 and 11 show the satisfactory output response and the tracking

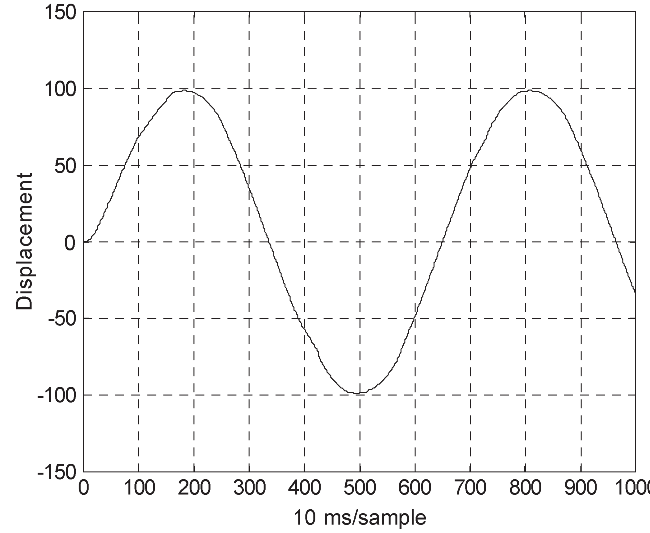

Fig. 10. Output response (under parameter variations).

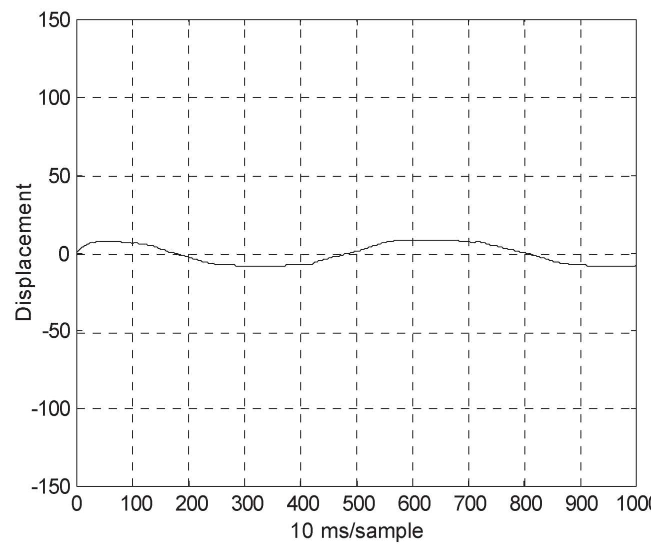

Fig. 11. Tracking error of the position control system.

error of the proposed controller under parameter variations. The results indicate that the tracking errors remained within ± 10 μm. Through experimental results, the proposed control algorithm has been successfully show to meet tracking performance for a positioning mechanism under setpoint changes, and gives satisfactory control performance.

V. Conclusion

This paper has proposed a systematic design methodology to develop the SPC controller for control of the nonlinear plants. The RWNN has been used for the discrete-time mathematical model of the nonlinear system and the proposed controller has been designed for control of the nonlinear systems. The setpoint tracking and load disturbance rejection capabilities of the proposed method can be improved by the modified predictive performance criterion. The sufficient conditions have been established for securing both stability and steady-state performance of the control system. The adaptive control algorithm, including the ALR for the RWNN model, has been applied to two nonlinear systems and a positioning mechanism, respectively. Through the results from computer simulations and experiments, the proposed control system has been proven useful and pragmatic for achieving setpoint tracking either in the absence or in the presence of external disturbances.

Acknowledgment

The author would like to thank the reviewers for their many helpful comments and suggestions.

References

[1] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control—Part I. The basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, Mar. 1987.
[2] E. F. Camacho and C. Bordons, *Model Predictive Control*. London, U.K.: Springer-Verlag, 1999.
[3] B. Kouvaritakis and M. Cannon, *Nonlinear Predictive Control—Theory and Practice*. London, U.K.: IEE, 2001.
[4] S. Huang, K. K. Tan, and T. H. Lee, *Applied Predictive Control*. London, U.K.: Springer-Verlag, 2002.
[5] K. SoonLow and R. Cao, "Model predictive control of parallel-connected inverters for uninterruptible power supplies," *IEEE Trans. Ind. Electron.*, vol. 55, no. 8, pp. 2884–2893, Aug. 2008.
[6] G. D. Nicolao, L. Magi, and R. Scattolini, "Stabilizing predictive control of nonlinear ARX models," *Automatica*, vol. 33, no. 9, pp. 1691–1697, Sep. 1997.
[7] E. Granado, W. Colmenares, J. Bernussou, and G. García, "Linear matrix inequality based model predictive controller," *Proc. Inst. Elect. Eng.—Control Theory Appl.*, vol. 150, no. 5, pp. 528–533, Sep. 2003.
[8] C. H. Lu and C. C. Tsai, "Generalized predictive control using recurrent fuzzy neural networks for industrial processes," *J. Process Control*, vol. 17, no. 1, pp. 83–92, Jan. 2007.
[9] S. S. Ge, C. Yang, and T. H. Lee, "Adaptive predictive control using neural network for a class of pure-feedback systems in discrete time," *IEEE Trans. Neural Netw.*, vol. 19, no. 9, pp. 1599–1614, Sep. 2008.
[10] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical system using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, Mar. 1990.
[11] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, 1989.
[12] K. Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural network," *Neural Netw.*, vol. 6, no. 6, pp. 801–806, 1993.
[13] L. Jin, P. N. Nikiforuk, and M. Gupta, "Approximation of discrete-time state-space trajectories using dynamic recurrent networks," *IEEE Trans. Autom. Control*, vol. 40, no. 7, pp. 1266–1270, Jul. 1995.
[14] C. C. Ku and K. Y. Lee, "Diagonal recurrent neural networks for dynamical system control," *IEEE Trans. Neural Netw.*, vol. 6, no. 1, pp. 144–156, Jan. 1995.
[15] C. H. Lu and C. C. Tsai, "Adaptive predictive control with recurrent neural network for industrial processes: An application to temperature control of a variable-frequency oil-cooling machine," *IEEE Trans. Ind. Electron.*, vol. 55, no. 3, pp. 1366–1375, Mar. 2008.
[16] F. J. Lin, L. T. Teng, J. W. Lin, and S. Y. Chen, "Recurrent functional-link-based fuzzy-neural-network-controlled induction-generator system using improved particle swarm optimization," *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp. 1557–1577, May 2009.
[17] Y. C. Pati and P. S. Krishnaprasad, "Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations," *IEEE Trans. Neural Netw.*, vol. 4, no. 1, pp. 73–85, Jan. 1993.
[18] B. Delyon, A. Juditsky, and A. Benveniste, "Accuracy analysis for wavelet approximations," *IEEE Trans. Neural Netw.*, vol. 6, no. 2, pp. 332–348, Mar. 1995.
[19] Q. Zhang and A. Benveniste, "Wavelet networks," *IEEE Trans. Neural Netw.*, vol. 3, no. 6, pp. 889–898, Nov. 1992.
[20] F. J. Lin, R. J. Wai, and M. P. Chen, "Wavelet neural network control for linear ultrasonic motor drive via adaptive sliding-mode technique," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 50, no. 6, pp. 686–697, Jun. 2003.
[21] R. H. Abiyev and O. Kaynak, "Fuzzy wavelet neural networks for identification and control of dynamic plants—A novel structure and a comparative study," *IEEE Trans. Ind. Electron.*, vol. 55, no. 8, pp. 3133–3140, Aug. 2008.
[22] S. H. Ling, H. H. C. Iu, F. H. F. Leung, and K. Y. Chan, "Improved hybrid particle swarm optimized wavelet neural network for modeling the development of fluid dispensing for electronic packaging," *IEEE Trans. Ind. Electron.*, vol. 55, no. 9, pp. 3447–3460, Sep. 2008.
[23] D. Gu and H. Hu, "Neural predictive control for a car-like mobile robot," *Robot. Auton. Syst.*, vol. 39, no. 2, pp. 73–86, May 2002.
[24] S. J. Yoo, J. B. Park, and Y. H. Choi, "Stable predictive control of chaotic systems using self-recurrent wavelet neural network," *Int. J. Control Autom. Syst.*, vol. 3, no. 1, pp. 43–55, 2005.
[25] S. J. Yoo, Y. H. Choi, and J. B. Park, "Generalized predictive control based on self-recurrent wavelet neural network for stable path tracking of mobile robots: Adaptive learning rates approach," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 6, pp. 1381–1394, Jun. 2006.
[26] K. J. Åström and B. Wittenmark, *Adaptive Control*. Upper Saddle River, NJ: Addison-Wesley, 1995.
[27] K. R. Sales and S. A. Billings, "Self-tuning control of nonlinear ARMAX model," *Int. J. Control*, vol. 51, pp. 753–769, 1990.
[28] J. Xu, D. W. C. Ho, and D. Zhou, "Adaptive wavelet networks for nonlinear system identification," in *Proc. Amer. Control Conf.*, 1999, vol. 5, pp. 3472–3473.
[29] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1999.
[30] W. Rudin, *Principles of Mathematical Analysis*. New York: McGraw-Hill, 1976.
[31] C. H. Lee and C. C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural networks," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 4, pp. 349–366, Aug. 2000.
[32] C. J. Lin and C. H. Chen, "A compensation-based recurrent fuzzy neural network for dynamic system identification," *Eur. J. Oper. Res.*, vol. 172, no. 2, pp. 696–715, Jul. 2006.

Chi-Huang Lu (M'98) received the B.S. degree in automatic engineering from Feng-Chia University, Taichung, Taiwan, in 1989, and the M.S. and Ph.D. degrees in electrical engineering from National Chung-Hsing University, Taichung, in 2000 and 2007, respectively.

He is currently an Associate Professor in the Department of Electrical Engineering, Hsiuping Institute of Technology, Taichung. His research interests include predictive control, intelligent control, and their applications to industrial systems.