# Neural Predictive Control Design for Uncertain Nonlinear Systems

**Wei Wu[1,*] and Jun-Xian Chang[2]**

[1]*Department of Chemical Engineering, National Yunlin University of Science and Technology, Touliu, Yunlin 640, Taiwan, R.O.C.*
[2]*Department of Electrical Engineering, National Yunlin University of Science and Technology, Touliu, Yunlin 640, Taiwan, R.O.C.*

Abstract

Abstract ：In this paper, adaptive neural-network predictive control strategies for general nonlinear systems are presented. The system is described by an unknown NARMAX model and neuro model is used to on-line learn the system. Despite state/parameter estimation, the neural predictive control scheme associated with the constrained optimization framework is implemented in a straightforward manner. Through the Lyapunov stability analysis, the network weight adaptation rule is derived, and guarantees the minimum error between the neuro output and plant output. An unstable reactor system is given to demonstrate the effectiveness of the proposed control schemes. 

Keywords: Predictive control; Neural network; Input constraints; Dynamic backpropagation algorithm; Reactor systems

## 1. Introduction

Research on neural-network-based control systems has received a considerable attention over the past several years [14]. It is because many control systems often exhibit strongly nonlinearities, and the implicit programming of neural network computing architectures is able to accurately approximate any nonlinear function. Since uncertain nonlinear systems without complete model information are usually called as “black boxes”, the widely used structures of neural-network-based control designs lead to problems in nonlinear and adaptive control fashions. Ge et al. [5] provided a framework for structured dynamic modelling and adaptive control design for robots using neural networks, Kulawski and Brdyś [9] developed adaptive control technique for nonlinear stable systems using recurrent neural networks, and Calise et al. [2] presented an adaptive output feedback design procedure, where the design employed the feedback linearization coupled with an on-line neural network.

As it is mentioned above, the neuro identification process turns out to be of the central parts in constructing feasible adaptive controllers. However, the adaptive control combined with neuro-identifier would reduce the efficient and reliable computation due to the hybrid convergence manner. Recently, Poznyak et al. [17] indicated that ‘dynamic’ neural network architectures were successfully used to enhance tracking controllers, Poznyak et al. [18] proposed optimal adaptive controller based on on-line learning of the neural-network parameters to realize the good tracking performance, Ku and Lee [10] developed a diagonal recurrent neural network to generalize the structure of the tracking controllers, and Gao et al. [3] provided a qualified real-time implementation to elaborate the merit of the diagonal recurrent neural network. Obviously, the adaptive neuro-based identifier can create robust training algorithms and reinforce the robustness of adaptive control techniques [23].

Recently, the study of neuro-optimizers has been mentioned in order to make the system identification more efficient, and the neural network is more expected to determine controller parameters. Whereas the predictive control technique is a dynamic optimization approach to control problems, and the flexible constraint-handling capability makes it most suitable for process control problems, Lazar and Pastravanu [11] indicated that the neural predictive control could reduce the significant obstacles encountered in conventional model predictive control (MPC) applications, Noriega and Wang [16] developed the adaptive neural-network control scheme within the unconstrained optimization framework, Vila and Wanger [20] used the feedforward neural network (FNN) to generate a constrained optimal control at each time step, and Wang and Wan [21] provided the high computational demand in solving the optimization problems associated with the MPC technique and structured neural network approach. In other respects, Hoo et al. [6] established a reliable estimate by incorporating directional information to improve the predictive capability of neural network models, and Ahmed and Tasadduq [1] developed neural servocontroller for a nonlinear multi-input multi-output (MIMO) system with the strong interaction of the input-output pairs.

*Author to whom correspondence should be addressed. Email: weiwu@yuntech.edu.tw

In this paper, the flexible predictive control strategy using on-line neuro-based adaptation is developed. The control design procedure is implemented to highly nonlinear systems in the presence of parameter uncertainties and input constraints, that does not rely on state/parameter estimation. The resulting implementation of neural predictive controller is able to ensure a rapid and reliable solution to the control algorithm, so it is superior to 'deterministic' model-based predictive control applications.

## 2. Problem Statement

Consider the discrete-time uncertain system described by the NARMAX model [4] is shown as

$$\begin{aligned} y(k+1) = f_p(&y(k), y(k-1), \cdots, y(k-n+1), \\ &u(k), u(k-1), \cdots, u(k-m+1)) \end{aligned} \tag{1}$$

where $u(k)$ and $y(k)$ are measurable scalar input and output, respectively. $n$ is the number of past outputs used, $m$ is the number of past inputs used, and $f_p(\bullet)$ is the unknown nonlinear difference equation representing the system dynamics. Since the neural network architecture can provide the accurate approximation, the off-line model identification is recommended that the training procedure for a class of FNN architectures corresponding to the NARMAX model associated with adequately delayed inputs and optimum weights is addressed [16].

### *2.1 Off-line neuro identification*

The neural NARMAX model is given by

$$y_{\mathrm{NN}}(k+1) = \hat{f}(Y(k-d_y), U(k-d_u)) \tag{2}$$

where $\hat{f}(\bullet)$ represents the input-output transfer function of the neural network, $d_y$ and $d_u$ are the transport delays of the input space, and the vector forms are shown as

$$\begin{aligned} U(k-d_u) &= [u(k-d_u), \cdots, u(k-d_u-m+1)]^T \\ Y(k-d_y) &= [y(k-d_y), \cdots, y(k-d_y-n+1)]^T \end{aligned} \tag{3}$$

*Remark 1.* The model orders for the input and output and delays need to be determined for the non-recurrent neural modeling [22]. In this paper, a three-layer FNN with one net output and $n$+$m$+1 input is given by [13]

$$\hat{f}(z) = (W^O)^T \sigma[(W^I)^T z + W^B] \tag{4}$$

where $z = \left[U^T, Y^T, 1\right]^T$, and the activation function $\sigma(\bullet)$ is set as

$$\sigma(z_a) = \tanh(z_a),\ z_a \in \Re \tag{5}$$

$W^I \in \Re^{m\times(m+n+1)}$ is the first-to-second layer interconnection weight, $W^O \in \Re^{1\times m}$ is the second-to-third layer interconnection weight, and $W^B \in \Re^{m\times 1}$ is the threshold offset. In general, the learning algorithm consists in sequentially adjusting the network weights for the minimal mean squared error between the nominal NARMAX model $f_p(\bullet)$ and the neural NARMAX model $\hat{f}(\bullet)$. Moreover, the time validation test of the FNN model is employed for new input signals required. Through the off-line training and validation algorithms, the nominal FNN model is established and provides a stable input-output representation of neural networks.

### *2.2 Neural predictive control algorithm*

The objective of the predictive control strategy using FNN model is twofold: (i) to fulfill the output feedback control design and (ii) to minimize an objective function with a quadratic penalty function. Thus, the FNN model-based control problem is described by

$$\min_{u(k),\ldots,u(k+N_u-1)} \sum_{i=k}^{k+N_y} [y_{NN}(i) - r(i)]^2 + \lambda \sum_{j=k}^{k+N_u-1} [\Delta u(j)]^2 \tag{6}$$

subject to input constraint

$$u_{\min} \le u(j) \le u_{\max},\ j{=}k,\ \ldots,\ k+N_u-1\,. \tag{7}$$

where $\Delta u$ presents the manipulated variable increment, $N_u$ is the control horizon, $N_y$ is the prediction horizon, $r$ is the reference trajectory, and $\lambda$ is the weight factor. $u_{\min}$ and $u_{\max}$ represent lower and upper bounds, respectively. The output of FNN model is shown as

$$y_{NN}(i) = (\widetilde{W}^O)^T \sigma[(\widetilde{W}^I)^T z + \widetilde{W}^B],\ i{=}k,\ldots,\ k+N_y \tag{8}$$

($\widetilde{W}^O, \widetilde{W}^I, \widetilde{W}^B$) in (11) represent optimal values of the network weights. Note that the update of weights is corrected during the off-line training procedure.

*Remark 3.* Obviously, the above augmented objective function has been shown in [8, 13]. Because the quadratic problem (QP) framework is a nonlinear algebraic equation, the optimization toolbox in the Matlab environment provides the accurate computation. Inspired by the issue in [11], the one-step-ahead predictive control approach for the QP framework with $N_y$=1 and the single parameter $\lambda$ in (9) would induce a uniformly convex function so that a rapid, reliable solution to the control algorithm can be achieved.

*Remark 4.* The tuning scheme of FNN model-based predictive control usually depends on the number of output prediction $N_y$ and the weight factor $\lambda$. However, the tuning parameter $N_y$ cannot effectively improve the control performance when the FNN model cannot be faithful representation of the uncertain system. Consider that the control horizon $N_u$ is one and the control action is bounded by (10), Kambhampati et al. [7] indicated that the one-step-ahead predictive control design with the addition of the penalty function could be implemented to an open-loop unstable process.

## 3. Adaptive Neural-Network Predictive Controller

In fact, the above FNN model-based identification cannot instantly capture the real dynamic behavior. In this section, the on-line training procedure associated with network weight adaptation rule is utilized to establish an adaptive neural-network predictive control.

### *3.1 On-line neuro identification*

An approach for on-line identification using the FNN-based predictive model in (11) is presented. The current square error between the plant output and the neural output is defined as

$$E_m(k) = \frac{1}{2}(y(k) - y_{NN}(k))^2 \tag{9}$$

and the learning law by a gradient method [17] for the minimization of $E_m(k)$ is written as

$$\pi(k+1) = \pi(k) + \eta(-\frac{\partial E_m}{\partial \pi}) \tag{10}$$

where $\eta$ is a learning rate, and

$$\frac{\partial E_m(k)}{\partial \pi} = -(y(k) - y_{NN}(k))\frac{\partial y_{NN}(k)}{\partial \pi} \tag{11}$$

Obviously, from (13) and (14) it is a typically dynamic backpropagation (DBP) algorithm. According to the FNN architecture in (11), the learning rules of the weights are obtained as follows:

(1) Output weights:

$$w_i^O(k+1) = w_i^O(k) + \eta(y(k) - y_{NN}(k))\sigma(s_i(k)) \tag{12}$$

where $s_i(k) \equiv \sum_j w_{i,j}^I z_j(k) + w_i^B$.

(2) Input weights:

$$w_{i,j}^I(k+1) = w_{i,j}^I(k) + \eta(y(k) - y_{NN}(k))[w_i^O Q_{ij}(k)] \tag{13}$$

where $Q_{ij}(k) \equiv \sigma'(s_i(k))z_j(k)$ and $Q_{ij}(0) = 0$.

(3) Bias weights:

$$w_i^B(k+1) = w_i^B(k) + \eta(y(k) - y_{NN}(k))[w_i^O R_i(k)] \tag{14}$$

where $R_i(k) \equiv \sigma'(s_i(k))$ and $R_i(0) = 0$.

*Remark 5.* All weights are split up into the three-layer FNN architecture, Those tuning algorithms are used in the on-line fashion. The on-line update of the weights captures changes in the process dynamics. Apparently, the DBP algorithm can be treated as an adaptation approach. Referring the traditional adaptive mechanism in [19], the performance of those learning rules will rely critically on the choice of the learning rate $\eta$. However, the development of guideline in selecting the learning rate is omitted here due to the similar results which have been stated in [10]. For a small value of $\eta$, the convergence is guaranteed but its rate is slow. Oppositely, the too big $\eta$ will induce the unstable updating algorithm. In the next demonstrated examples depend on the on-line tuning scheme for the parameter $\lambda$ but the learning rate $\eta$ is fixed.

### *3.2 Control algorithm*

Inspired by the neural predictive control strategy in (9) and the DBP algorithm in (15)-(17), the one-step-ahead predictive controller using the on-line neuro identification is described by

$$\min_{u(k)} J_1(\lambda) = [\hat{y}_{NN}(k+1) - r(k+1)]^2 + \lambda \Delta u(k)^2 \tag{15}$$

$$\min_{\pi} J_2(\eta) = [\hat{y}_{NN}(k) - y(k)]^2 \tag{16}$$

subject to the current input constraint

$$u_{\min} \le u(k) \le u_{\max} \tag{17}$$

where the one-step-ahead FNN model is written as

$$\hat{y}_{NN}(k+1) = (W^O)^T \sigma[(W^I)^T z + W^B] \tag{18}$$

($W^O, W^I, W^B$) in (18) represent updated network weights connected to optimal network weights in (11) as initials.

*Remark 6.* Obviously, the control scheme is dynamic and adaptable. Since the solution for minimization of $J_2$ is explicit, the DBP algorithm can provide the asymptotic convergence by adjusting a proper parameter $\eta$. Under the one-step-ahead predictive horizon and input constraint, the optimizer not only carries out the minimal tracking error, i.e., $\lim_{k\to\infty} \hat{y}_{NN}(k) \approx y(k) \approx r(k)$, but also the tuning parameter $\lambda$ directly affects the closed-loop tracking performance and stability. Moreover, this adaptive neural-network predictive control (ANNPC) architecture is depicted in Fig. 1.

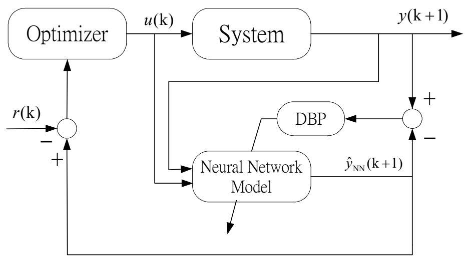


Fig. 1 Adaptive neural-network predictive control scheme.

## 4. Application to a Reactor System

A continuous stirred tank reactor (CSTR), in which an exothermic irreversible first-order reaction $(A \rightarrow B)$ takes place, is covered with a coolant stream flowing in a cocurrent fashion. The coolant flow rate is chosen to be manipulated input and the coolant temperature is allowed to vary along the length of the cooling coil [15]. Under the mass and energy balance, the dynamics of this CSTR model can be described as

$$\begin{aligned} \dot{C}_A &= \frac{q}{V}(C_{Af} - C_A) - k_0 C_A \exp(-\frac{E}{RT}) \\ \dot{T} &= \frac{q}{V}(T_f - T) + \frac{(-\Delta H) k_0 C_A}{\rho C_p} \exp(-\frac{E}{RT}) \\ &\quad + \frac{\rho_c C_{pc}}{\rho C_p V} q_c [1 - \exp(-\frac{hA}{q_c \rho_c C_{pc}})](T_{cf} - T) \end{aligned} \tag{19}$$

where $C_A$ is the effluent concentration of component A, $T$ is the reactor temperature, $q$ is the feed flowrate, and $q_c$ is the coolant flowrate. The nominal system parameters and operating point for the reactor are given in Table 1. Under input constraint by $80(\mathrm{l\,min^{-1}}) \le q_c \le 120(\mathrm{l\,min^{-1}})$, the control objective is to regulate $C_A$ by manipulating $q_c$.

Table 1 Nominal CSTR operating conditions

| | |
|---|---|
| $q = 100\ \mathrm{l\,min^{-1}}$ | $E/R = 9.95 \times 10^3\ \mathrm{K}$ |
| $C_{Af} = 1\ \mathrm{mol\,l^{-1}}$ | $-\Delta H = 2 \times 10^5\ \mathrm{cal\,mol^{-1}}$ |
| $T_f = 350\ \mathrm{K}$ | $\rho, \rho_c = 1000\ \mathrm{g\,l^{-1}}$ |
| $T_{cf} = 350\ \mathrm{K}$ | $C_p, C_{pc} = 1\ \mathrm{cal\,g^{-1} \cdot K^{-1}}$ |
| $V = 100\ \mathrm{l}$ | $q_c = 103.41\ \mathrm{min^{-1}}$ |
| $hA = 7 \times 10^5\ \mathrm{cal\,min^{-1} \cdot K^{-1}}$ | $T = 440.2\ \mathrm{K}$ |
| $k_0 = 7.2 \times 10^{10}\ \mathrm{min^{-1}}$ | $C_A = 8.36 \times 10^{-2}\ \mathrm{mol\,l^{-1}}$ |

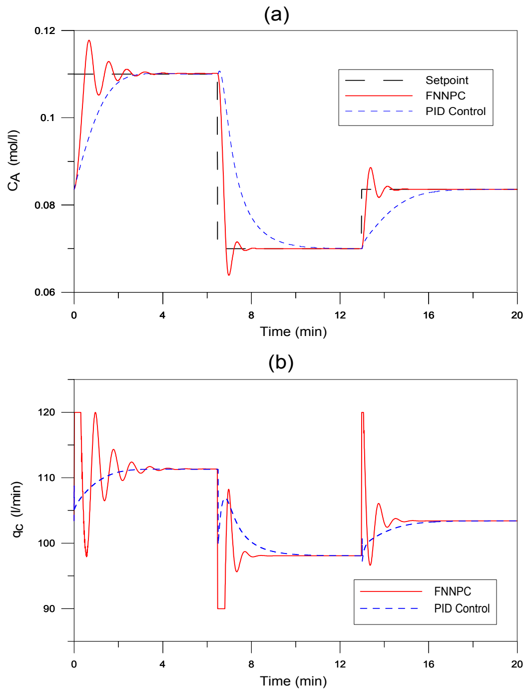


Fig. 2 Closed-loop output tracking responses for the reactor system, in the case of using the unconstrained FNNPC and PID controllers

According to the neural predictive control strategy in Section 2, Fig. 2 shows that the FNN-based predictive control (FNNPC) design can confirm better tracking performance than the conventional PID control approach. By the issue in [15], the PID controller parameters are $K_c = 190\ \mathrm{l^2/mol^{-1} \cdot min^{-1}}$, $\tau_I = 0.556$ min, and $\tau_D = 0.827$ min. For the output regulation of perturbed systems, the adaptive neural-network predictive control (ANNPC) in Section 3 is used. Under a small learning rate $\eta = 0.002$ for the on-line neuro adaptation, we find that the update of weights can keep a rapid, convergent manner. Fig. 3(a) depicts that the unconstrained ANNPC design can carry out the satisfactory disturbance attenuation for +20% change in the inlet concentration $C_{Af}$. However, Fig. 3(b) shows that the optimization-based control connected to the on-line adaptation procedure could cause the over-large control action. Within the constrained OP framework in (18), Fig. 4 shows that by adjusting $\lambda$ the satisfactory output tracking subject to an unsaturated control action is superior to the PID control. Moreover, the same ANNPC design for unknown disturbance attenuation is depicted in Fig. 5. Obviously, the simulation demonstrates that the proposed ANNPC method is relatively robust against unknown disturbances.

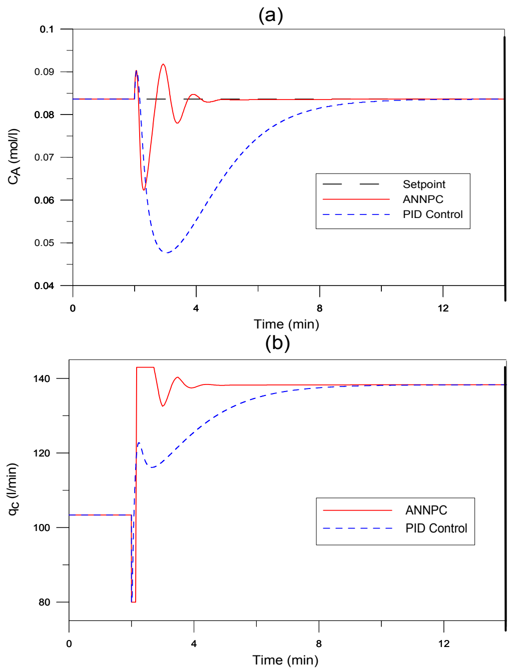


Fig. 3 Disturbance rejection for step change in feed concentration, in the case of using the unconstrained ANNPC and PID controllers

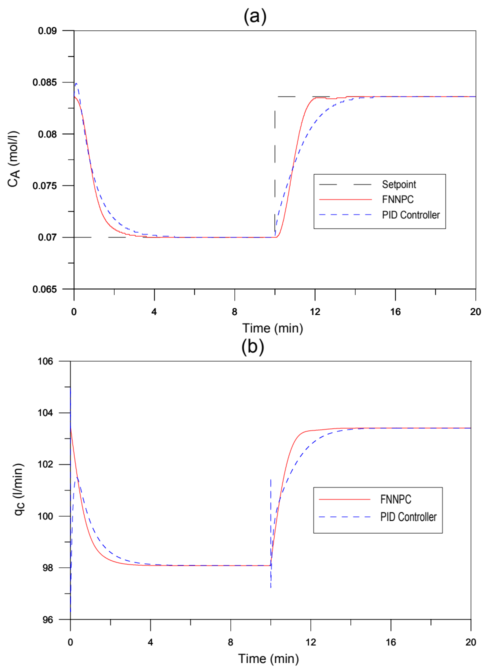


Fig. 4 Closed-loop output tracking responses for the reactor system, in the case of using the constrained FNNPC and PID controllers

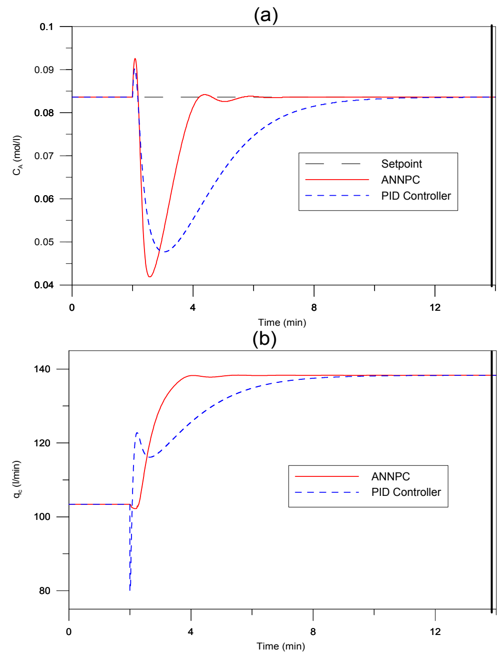


Fig. 5 Disturbance rejection for step change in feed concentration, in the case of using the constrained ANNPC and PID controllers:

## 5. Conclusion

In this work, predictive control strategies associated with FNN-based adaptation mechanism for constrained SISO and MIMO nonlinear systems are addressed. The discrete-time learning procedure enforce the minimal error between the FNN model output and plant output, and the one-step-ahead predictive control design can carry out the stable output regulation in spite of unknown disturbances. Definitely, the performance and robustness of the proposed control schemes have been successfully verified by a strongly nonlinear reactor system.

**Acknowledgments**：This work is supported by the National Science Council of the Republic of China under grant number NSC-92-2214-E-224-002.

## REFERENCE


1. Ahmed, M. S., and Tasadduq, I. A.: ‘Neural servocontroller for nonlinear MIMO plant’, *IEE Proc.-Control Theory Appl.*, 1998, **145**, (3), pp. 277-290
2. Calise, A. J., Hovakimyan, N., and Idan, M., ‘Adaptive output feedback control of nonlinear systems using neural networks’, *Automatica*, 2001, **37**, (8), pp. 1201-1211
3. Gao, F., Wang, F., and Li, M.: ‘A simple nonlinear controller with diagonal recurrent neural network’, *Chem. Eng. Sci.*, 2001, **55**, (7), pp. 1283-1288
4. Chen, S., and Billings, S. A.: ‘Representations of nonlinear systems: the NARMAX model’, *Int. J. Control*, 1989, **49**, (10), pp. 1013-1032

5. Ge, S. S., Lee, T. H., and Harris, C. J.: 'Adaptive neural network control of robotic manipulators' (World Scientific Publishing Co., Singapore, 1998)
6. Hoo, K. A., Sinzinger E. D., and Piovoso, M. J.: "Improvements in the predictive capability of neural network', *Journal of Process Control*, 2002, **12**, (1), pp. 193-202
7. Jiang, Z. P., and Wang, Y.: 'Input-to-state stability for discrete-time nonlinear systems', *Automatica*, 2001, **37**, (2), pp. 857-869
8. Kambhampati, C., Mason, J. D., and Warwick, K.: 'A stable one-step-ahead predictive control of nonlinear systems', *Automatica*, 2000, **36**, (4), pp. 485-495
9. Kulawski, G. J., and Brdyś, M. A.: 'Stable adaptive control with recurrent networks', *Automatica*, 2000, **36**, (1), pp. 5-22
10. Ku, C. C., and Lee, K. Y.: 'Diagonal recurrent neural networks for dynamic systems control', *IEEE Transactions on Neural Networks*, 1995, **6,** (1), pp. 144-156
11. Lazar, M., and Pastravanu, O.: 'A neural predictive controller for nonlinear systems', *Mathematics and Computers in Simulation*, 2002, **60**, (2), pp. 315-324
12. Lewis, F. L., Jagannathan, S., and Yeşildirek, A.: 'Neural network control of robot manipulators and nonlinear systems' (Taylor & Francis, London, 1999)
13. Maciejowaki, J. M.: 'Predictive control with constraints' (Pearson Education Limited, England, 2002)
14. Narendra, K. S., and Lewis, F. L.: 'Introduction to the special issue on neural network feedback control', *Automatica*, 2001, **37**, (8), pp. 1147-1148
15. Nahas, E. P., Henson, M., and Seborg, D. E.: 'Nonlinear internal model control strategy for neural network models,' *Computers & Chem. Eng.*, 1992, **16**, (12), pp. 1039-1057
16. Noriega, J., and Wang, H.: 'A direct adaptive neural-network control for unknown nonlinear systems and its application', *IEEE Transactions on Neural Networks*, 1998, **9**, (1), pp. 27-33
17. Poznyak, A. S., Sanchez, E. N., and Yu, W.: 'Differential neural networks for robust nonlinear control' (World Scientific Publishing Co., Singapore, 2001)
18. Poznyak, A. S., Yu, W., Sanchez, E. N., and PEREZ, J. P.: 'Nonlinear adaptive trajectory tracking using dynamic neural networks', *IEEE Transactions on Neural Networks*, 1999, **10**, (6), pp. 1402-1411
19. Slotine, J.-J. E., and Li, W.: 'Applied nonlinear control' (Prentice Hall Englewood Cliffs, New Jersey, 1991)
20. Vila, J. P., and Wagner, V.: 'Predictive neurocontrol of uncertain systems: design and use of a neuro-optimizer', *Automatica*, 2003, **39**, (5), pp. 767-777
21. Wang, L. X., and Wan, F.: 'Structured neural networks for constrained model predictive control', *Automatica*, 2001, **37**, (8), pp. 1235-1243
22. Yu, D. L., Gomm, J. B., and Williams, D.: 'On-linear predictive control of a chemical process using neural network models', *14th World Congress of IFAC*, 1999, Beijing, P.R. China, pp. 121-126
23. Yu, W., and Li, X.: 'Discrete-time neuro identification without robust modification', IEE *Proc.-Control Theory Appl.*, 2003, **150**, (3), pp. 311-316