

Neural Network Model Based on Predictive Control for Multivariable Nonlinear Systems

Jianfeng Yang Jun Zhao Jixin Qian Jian Niu

State Key Lab. of Industrial Control Technology, Zhejiang University, Hangzhou 310027, P. R. China

Abstract

A nonlinear model predictive control (NMPC) algorithm based on a BP-ARX combination model is proposed for multivariable nonlinear systems whose static nonlinearity between inputs and outputs could be obtained. The dynamic behavior of the system is described by a parameter varying ARX model, whose parameters are estimated on-line with recursive least-squares algorithm and rescaled properly according to a BP neural network representing the system static nonlinearity. The construction of the BP-ARX model and a constrained NMPC algorithm based on the BP-ARX model are elaborated. The effectiveness of the proposed method is demonstrated by simulation on a multivariable chemical reactor system.

Keywords: Neural networks, Model predictive control, Nonlinear systems, ARX model

1. Introduction

During the past decades, considerable progress has been made in the theory and practice of model predictive control (MPC) [1]. MPC is well suited for high performance control of constrained multivariable processes because explicit pairing of input and output variables is not required and constraints can be incorporated directly into the associated on-line optimal control problem. A major limitation of linear MPC is that plant behavior is described by linear dynamic models. As a result, linear MPC is inadequate for highly nonlinear process and moderately nonlinear process which has large operating regions. Increasing demands on throughput and product quality have spurred the development of nonlinear MPC (NMPC) in which a more accurate nonlinear dynamic model is used for process prediction and optimization [2][3].

It is estimated that, in a typical commissioning project, modeling efforts can take up to 90% of the cost and time in implementing a model predictive controller [4]. Consequently, the development of a suitable nonlinear dynamic model of the controlled process is of paramount importance to the efficient

implementation of NMPC. The available strategies to develop system dynamic model mainly include:

- Representing system by a linear model
- Representing system by several linear models [5], Hammerstein models or Wiener models [6]
- Representing system by neural networks [7], fuzzy models [8], or the combination of neural networks and fuzzy models [9]

This paper proposes a novel nonlinear dynamic model and elaborates its usage in NMPC. The proposed model combines a second order ARX (AutoRegressive with eXternal input) model identified on-line by recursive least-squares algorithm (RLS) and a BP (Back-Propagation) neural network trained off-line, referred to as BP-ARX model. The construction of the BP-ARX model is given in Section 2. A constrained NMPC algorithm based on the BP-ARX model is elaborated in Section 3. The simulation results of the proposed method and recursive generalized predictive control on a double-input double-output nonlinear chemical process are illustrated in Section 4. Section 5 draws the conclusions of this paper.

2. The BP-ARX model

2.1. BP neural network steady-state model

In the process industries, two types of process data are readily available: historical data and open-loop plant test data. The steady-state relation between inputs and outputs of most processes could always be extracted from those data. Thus the system static nonlinearity can be regarded as an independent factor, and constructed as a steady-state model. For systems whose first-principles models represented by ordinary differential equation (ODE) are known in advance, the steady-state data between inputs and outputs can be obtained by solving the ODE.

With inherent ability to approximate any nonlinear continuous function without requiring any prior knowledge, the neural networks theory has been greatly developed and widely used in process control.

Because there are only a nonlinear mapping between the system steady-state inputs and outputs, a static network without feedback elements can describe the mapping. BP neural networks are the most commonly used multilayer feed-forward networks and can be easily trained with back-propagation algorithm. Thus a three-layer BP network is used to represent the system static nonlinearity. The structure of the BP network is shown in Fig. 1.

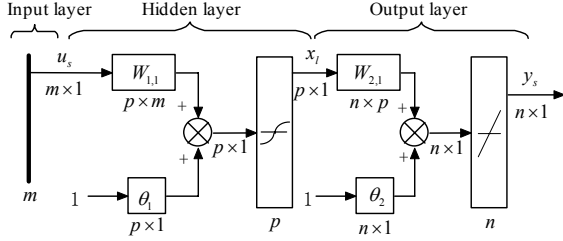


Fig. 1: Structure of the BP network.

Suppose that there are m inputs and n outputs with a hidden layer consisting of p hidden nodes. Training the network using a gradient descent optimization procedure based on the system steady-state inputs and outputs data, the BP neural network steady-state model representing the system static nonlinearity can be represented as:

$$y_s = f_2 \left[W_{2,1} \cdot f_1 (W_{1,1} \cdot u_s + \theta_1) + \theta_2 \right] \quad (1)$$

where $y_s = [y_{1s}, y_{2s}, \dots, y_{ns}]^T$ is system steady-state output vector, $u_s = [u_{1s}, u_{2s}, \dots, u_{ms}]^T$ is system steady-state input vector, $W_{1,1}$, $W_{2,1}$, θ_1 , θ_2 are weighting matrices and bias vectors of the hidden layer and the output layer, respectively, f_1 is the hidden layer transfer function defined as the following tan-sigmoid style:

$$f_1 = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (2)$$

and f_2 is the output layer transfer function defined as the following linear style:

$$f_2 = x \quad (3)$$

Reversing the inputs and outputs data and retraining the BP network, the output-input steady-state model can be represented as:

$$u_s = f_2 \left[W_{2,1} \cdot f_1 (W_{1,1} \cdot y_s + \theta_1) + \theta_2 \right] \quad (4)$$

where the notations have the same meanings but different values as those of in Eq. (1).

The BP neural network steady-state model is trained off-line and can be retrained periodically with the steady-state data which is updated during industrial application.

2.2. Parameter varying ARX model

Test the system dynamic outputs at initial conditions with Pseudo-Random Binary Signal (PRBS), a second order multivariable ARX model with m inputs and n outputs can be easily identified with Least Squares (LS) method described in [10]:

$$y(k) = \sum_{i=1}^2 [A_i y(k-i) + B_i u(k-i)] \quad (5)$$

where $y(k) = [y_1(k), y_2(k), \dots, y_n(k)]^T$ is a vector of measured outputs or controlled variables, $u(k) = [u_1(k), u_2(k), \dots, u_m(k)]^T$ is a vector of process inputs or manipulated variables, $A_i \in R^{n \times n}$, $B_i \in R^{n \times m}$.

It may be necessary to perform on-line model adaptation when the process deviates significantly from the operating conditions. Thus, the parameters of above ARX model can be updated on-line with RLS. Let

$$\theta(k) = [A_1 \ A_2 \ B_1 \ B_2] \quad (6)$$

$$\varphi^T(k) = [y(k-1)^T \ y(k-2)^T \ u(k-1)^T \ u(k-2)^T] \quad (7)$$

where $\theta(k) \in R^{n \times 2(n+m)}$, $\varphi(k) \in R^{2(n+m) \times 1}$.

Then, the parameters of ARX model (5) can be updated on-line by:

$$\theta(k) = \theta(k-1) + [y(k) - \theta(k-1)\varphi(k)]L^T(k) \quad (8)$$

$$L(k) = \frac{P(k-1)\varphi(k)}{\rho + \varphi^T(k)P(k-1)\varphi(k)} \quad (9)$$

$$P(k) = \frac{1}{\rho} P(k-1) [I - L(k)\varphi^T(k)] \quad (10)$$

where $0 < \rho \leq 1$ is forgetting factor, $L(k)$ is correcting factor, $P(k)$ is positive definite matrix, $P(0) = \alpha I$, I is an identity matrix, α is commonly assigned a small value ($0 < \alpha \leq 1$) for high signal-to-noise ratio and a large value ($\alpha > 1$) for low signal-to-noise ratio [11].

Although the RLS system identification process can update the model parameters in such a way that the outputs computed from Eq. (5) are as close as possible with the measured outputs in the sense of least squares, ARX model (5) cannot reflect the future nonlinearity of the system and NMPC based on this model will show a performance of slow respond to great changes in set-points. Therefore, a new concept Adaptation Index ζ is brought forward to constrain the effect time of RLS and its definition is as following.

Definition: The Adaptation Index ζ is a positive constant and is used to determine the start and end of the RLS. If and only if ζ less than the set value provided by user, the RLS comes into effect. ζ is calculated on-line and satisfies the following formula:

$$\zeta = \max \left\{ \left| \frac{y_1(k) - y_{1sp}(k)}{y_{1sp}(k)} \right|, \dots, \left| \frac{y_n(k) - y_{nsp}(k)}{y_{nsp}(k)} \right| \right\} \quad (11)$$

where $|y|$ denotes the absolute value of y , $y_{sp} = [y_{1sp}, y_{2sp}, \dots, y_{nsp}]^T$ is system output set-points, if y_{jsp} equals zero, the ζ of j th output will not be calculated.

In order to reflect the future nonlinearity of the system by ARX model (5), the parameters of the ARX model are rescaled by setting the model's steady-state gain be equal to the gains calculated according to BP neural network steady-state model.

The steady-state gain matrix of ARX model (5) is:

$$K = \frac{B_1 + B_2}{I - A_1 - A_2} = (I - A_1 - A_2)^{-1} \cdot (B_1 + B_2) \quad (12)$$

The gain between the steady-state input u_i and the steady-state output y_j is:

$$K^s [u_i] = \frac{\partial y_j}{\partial u_i} = \frac{y_j(u_i + \Delta u_i) - y_j(u_i)}{\Delta u_i} \quad (13)$$

where $i=1, 2, \dots, m, j=1, 2, \dots, n, y_j(u_i + \Delta u_i)$ and $y_j(u_i)$ are calculated according to Eq. (1), Δu_i is an arbitrary small positive number.

The future gains during the prediction horizon are approximate to the linear interpolation of the current steady-state gain K_1^s and the next steady-state gain K_2^s .

$$K_j [u_i(k+q)] = K_{1,ji}^s + \frac{K_{2,ji}^s - K_{1,ji}^s}{u_i^n - u_i^c} \Delta u_i(k+q) \quad (14)$$

where u_i^c, u_i^n are the current and the next i th input, respectively, which are calculated according to Eq. (4) based on the current measurement outputs $y(k)$ and the next set-points $y_{sp}(k+1)$, K_1^s, K_2^s are gains calculated according to Eq. (13) based on u^c, u^n , respectively, $\Delta u_i(k+q)$ is the increment of i th input at $k+q$ time instant, $q=0, 1, \dots, P-1, P$ is prediction horizon.

Multiply $\Delta=1-q^{-1}$ at both side of Eq. (5), the corresponding incremental form of ARX model (5) is

$$\Delta y(k) = \sum_{i=1}^2 [A_i \Delta y(k-i) + B_i \Delta u(k-i)] \quad (15)$$

where $\Delta y(k) = y(k) - y(k-1), \Delta u(k) = u(k) - u(k-1)$.

Setting the gain calculated from Eq. (12) be equal to the gain calculated from Eq. (14) by rescaling the B_1, B_2 in the ARX model (15), the following parameter varying ARX model describing the dynamic behavior of nonlinear system over their entire operating regions can be obtained:

$$\Delta y(k) = \sum_{i=1}^2 [A_i \Delta y(k-i) + B_{1i} \Delta u(k-i) + B_{2i} \Delta u^2(k-i)] \quad (16)$$

where

$$\begin{aligned} B_{1i} &= (I - A_1 - A_2) \cdot [B_i' / (B_1' + B_2')] \cdot K_1^s \\ B_{2i} &= (I - A_1 - A_2) \cdot [B_i' / (B_1' + B_2')] \cdot K_2^s \\ B_i' &= (I - A_1 - A_2) \cdot [B_i' / (B_1 + B_2)] \\ K' &= (K_2^s - K_1^s) / (u^n - u^c) \end{aligned} \quad (17)$$

and A_i, B_i have the same value as those in Eq. (6), other notation have the same value as those in Eq. (14), $B_1./B_2$ denotes dividing every two number at the same row and the same column in matrix B_1 and B_2 , $B_1.*K$ denotes multiplying every two number at the same row and the same column in matrix B_1 and K .

3. Constrained NMPC

Define the variables $S(n), T(n)$ and $H(n)$ as:

$$\begin{aligned} S(n) &= A_1 S(n-1) + A_2 S(n-2) \\ T(n-1) &= S(n-1) B_{11} + S(n-2) B_{12} \end{aligned} \quad (18)$$

$$H(n-1) = S(n-1) B_{21} + S(n-2) B_{22}$$

where $n=3, 4, \dots, P+1$, and $S(1)=I, S(2)=A_1, T(1)=B_{11}, H(1)=B_{21}$.

Then, the system output increments $\Delta \hat{y}(k+j|k)$, ($j=1, 2, \dots, P$) in prediction horizon can be calculated according to Eq. (16)-(18).

$$\begin{aligned} \Delta \hat{Y}(k) &= G_{11} \Delta U(k) + G_{12} \Delta U^2(k) + G_{21} \Delta u(k-1) \\ &+ G_{22} \Delta u^2(k-1) + F_1 \Delta y(k) + F_2 \Delta y(k-1) \end{aligned} \quad (19)$$

where

$$\Delta \hat{Y}(k) = [\Delta \hat{y}(k+1|k) \quad \Delta \hat{y}(k+2|k) \quad \dots \quad \Delta \hat{y}(k+P|k)]^T,$$

$$\Delta U(k) = [\Delta u(k) \quad \Delta u(k+1) \quad \dots \quad \Delta u(k+M-1)]^T,$$

$$\Delta U^2(k) = [\Delta u^2(k) \quad \Delta u^2(k+1) \quad \dots \quad \Delta u^2(k+M-1)]^T,$$

$$\Delta u(k-1) = [\Delta u_1(k-1) \quad \Delta u_2(k-1) \quad \dots \quad \Delta u_m(k-1)]^T,$$

$$\Delta u^2(k-1) = [\Delta u_1^2(k-1) \quad \Delta u_2^2(k-1) \quad \dots \quad \Delta u_m^2(k-1)]^T,$$

$$G_{11} = \begin{bmatrix} T(1) \\ T(2) & T(1) \\ \vdots & \vdots & \ddots \\ T(M) & T(M-1) & \dots & T(1) \\ \vdots & \vdots & \vdots & \vdots \\ T(P) & T(P-1) & \dots & T(P-M+1) \end{bmatrix},$$

$$G_{12} = \begin{bmatrix} H(1) \\ H(2) & H(1) \\ \vdots & \vdots & \ddots \\ H(M) & H(M-1) & \dots & H(1) \\ \vdots & \vdots & \vdots & \vdots \\ H(P) & H(P-1) & \dots & H(P-M+1) \end{bmatrix},$$

$$G_{21} = [S(1) \quad S(2) \quad \dots \quad S(P)]^T \cdot B_{12},$$

$$G_{22} = [S(1) \quad S(2) \quad \dots \quad S(P)]^T \cdot B_{22},$$

$$F_1 = [S(2) \quad S(3) \quad \dots \quad S(P+1)]^T,$$

$$F_2 = [S(1) \quad S(2) \quad \dots \quad S(P)]^T \cdot A_2.$$

Then, the system predictive outputs will be:

$$\hat{Y}(k) = V \cdot \Delta \hat{Y}(k) + W \cdot y(k) \quad (20)$$

$$\text{where } V = \begin{bmatrix} I \\ I & I \\ \vdots & \vdots & \ddots \\ I & I & \dots & I \end{bmatrix}_{(n \times P) \times (n \times P)}, \quad W = \begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix}_{(n \times P) \times n}.$$

Correcting predictive outputs with the error $e_p(k)$ between predictive outputs and system real outputs at time instant k :

$$Y_c(k) = \hat{Y}(k) + Z \cdot e_p(k) \quad (21)$$

where $e_p(k) = y(k) - y(k-1) - \Delta\hat{y}(k|k-1)$, $Z \in R^{(n \times P) \times n}$ is a correction matrix, $y(k)$ is the system output vector at time instant k , $\Delta\hat{y}(k|k-1)$ is the predictive output increment at time instant k calculated at time instant $k-1$.

Setting reference trajectory as:

$$Y_r(k) = [y_r^T(k+1) \quad y_r^T(k+2) \quad \dots \quad y_r^T(k+P)]^T \quad (22)$$

where $y_r(k+j) = C y_r(k+j-1) + (I-C)y_{sp}$, $j=1, 2, \dots, P$, $y_r(k) = y(k)$ is system measurement output at time instant k , $C = \text{diag}\{c^1, c^2, \dots, c^n\}$ ($0 \leq c \leq 1$) is soft coefficient matrix.

The on-line optimization problem to be solved in NMPC can be formulated as:

$$\begin{aligned} \min J(k) &= [\hat{Y}(k) - Y_r(k)]^T Q [\hat{Y}(k) - Y_r(k)] \\ &\quad + \Delta U(k)^T R \Delta U(k) \\ \text{s.t. } \Delta u_{\min} &\leq \Delta u(k+i) \leq \Delta u_{\max} \\ u_{\min} &\leq u(k+i) \leq u_{\max}, (i=0, 1, \dots, M-1) \\ y_{\min} &\leq \hat{y}(k+j) \leq y_{\max}, (j=1, 2, \dots, P) \end{aligned} \quad (23)$$

where P is prediction horizon, M is control horizon, $Q=qI$, $R=rI$ are output error weighing matrix and control input weighing matrix.

Because predictive model (16) has quadratic parts, it is a nonlinear predictive model. The optimization problem (23) must be solved by nonlinear programming algorithm. SQP (Sequential Quadratic Programming) is one of the most important methods for the nonlinear programming (NLP) problems. It has been shown that SQP requires the fewest function evaluations to solve NLP problems and can be applied to a wide range of process systems engineering problems with different structures. The convergence rate of SQP has been proved to be super linear.

Then the optional control inputs $u(k)$ will be:

$$u(k) = u(k-1) + \Delta u(k) \quad (24)$$

where $\Delta u(k)$ are the optional control input increments at time instant k which are obtained by solving the on-line optimization problem (23) using SQP.

The block diagram of the constrained NMPC based on the BP-ARX model is shown in Fig. 2

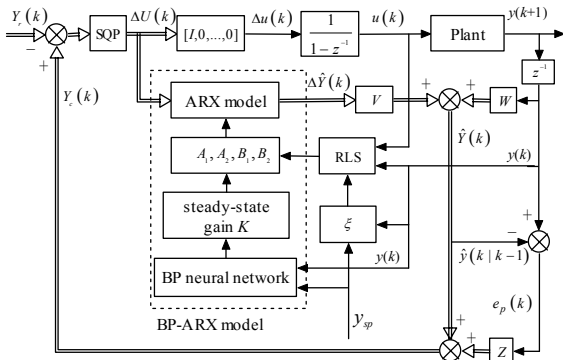


Fig. 2: Block diagram of the constrained NMPC.

4. Simulation example

Because of their highly nonlinear behavior, chemical reactors are the most common application and have become a benchmark problem for NMPC simulation studies. To demonstrate the effectiveness of the NMPC based on the BP-ARX model, a nonlinear MIMO system (a reactor plant) is considered for simulation purpose [12].

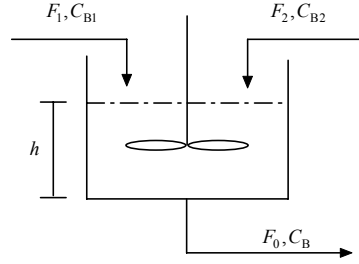


Fig. 3: Chemical reactor system.

For the reaction $A + B \rightarrow P$, where A is an excess component, C_B is the concentration of component B . The reaction occurs in an ideal stirred tank as shown in Fig. 3. Concentration of component B is assumed to be constant in both inlet flows. Both inlet flows contain an excess amount of A . The tank is well stirred with a liquid outflow rate determined by the liquid height in the tank. The system parameters are $C_{B1} = 24.9$, $C_{B2} = 0.1$.

The simplified dynamic model of the reactor is:

$$\begin{aligned} \dot{x}_1 &= u_1 + u_2 - 0.2x_1^{0.5} \\ \dot{x}_2 &= \frac{(24.9 - x_2)u_1 + (0.1 - x_2)u_2}{x_1} - \frac{x_2}{(1 + x_2)^2} \end{aligned} \quad (25)$$

$$y_1 = x_1, \quad y_2 = x_2$$

where u_1 is inlet flow rate with condensed B (F_1), u_2 is inlet flow rate with dilute B (F_2), x_1 is liquid height in the tank (h) and x_2 is concentration of B in the reactor.

The control objective is to minimize the difference between the reactor output and set-points. Initial conditions are selected as $x_{10} = 40$, $x_{20} = 0.17$. Unlike these variables in reference [12], the sampling time is 10 second here, and u_1, u_2 are constraint as $[0, 2]$.

The steady-state data between inputs and outputs can be obtained by solving the differential Eq. (25). Choosing 50 number of u_1, u_2 respectively between constraints $[0, 2]$, 2500 pairs of sample data can be obtained to train the BP neural network. With Neural Network Toolbox in MATLAB, the BP neural network with 15 hidden nodes can be easily obtained. The steady-state characteristic of the chemical reactor system described by the sample data is shown in Fig. 4.

The original second order ARX model can be obtained by testing the system dynamic outputs with PRBS at initial conditions. Let $P=5$, $M=3$, $c=0.5$, $q=1$,

$r=1$, $\alpha=100$, $\rho=0.98$, the control effects of different Adaptation Index on NMPC are shown in Fig. 5, where the dashed line are set-points, the dotted line are NMPC without RLS, the dash-dot line are NMPC always having RLS and the solid line are NMPC partially having RLS. It can be seen that NMPC partially having RLS can quickly respond to set-points change, and has good stability when approaches set-points. The great fluctuation in outputs for $\zeta=2$ is due to the combined effect of the RLS updating model's parameters with Eq. (8) and gain scheduling strategy rescaling the model's parameters with Eq. (17) at the middle of system operating point change, in which case the system real dynamic gain differs greatly from the gain calculating from Eq. (13). Consequently, the effect time of RLS should be constrained and a suitable value of Adaptation Index ζ can improve the prediction accuracy of the BP-ARX model.

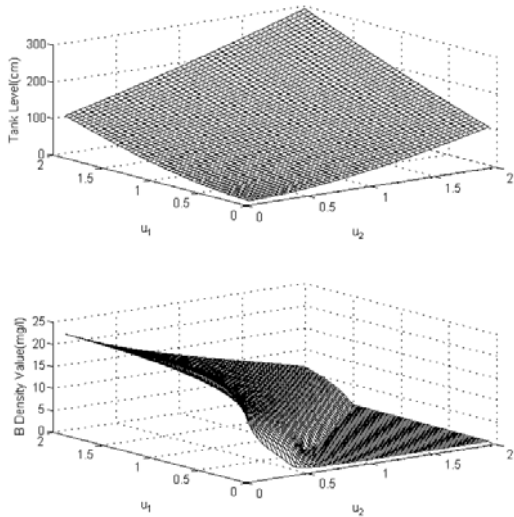


Fig. 4: Steady-state characteristic of chemical reactor system.

Generalized Predictive Control (GPC) [13] is one of the most popular members of the MPC family, which has quickly received widespread acceptance. Because there are no on-line estimation is used, with the multivariable GPC algorithm proposed in [14], this chemical reactor system can't achieve its control objective. Moon *et al.* [11] developed a recursive GPC (RGPC), which combines the process of the RLS system identification and GPC design. The advantages of RGPC are that no prior system information is required, and that the controller is updated adaptively in the presence of a changing operating environment [11]. Thus a comparison of the control effects between NMPC and RGPC is processed here.

With the same parameters and reference trajectory as above, output error weighing $q=15$ and

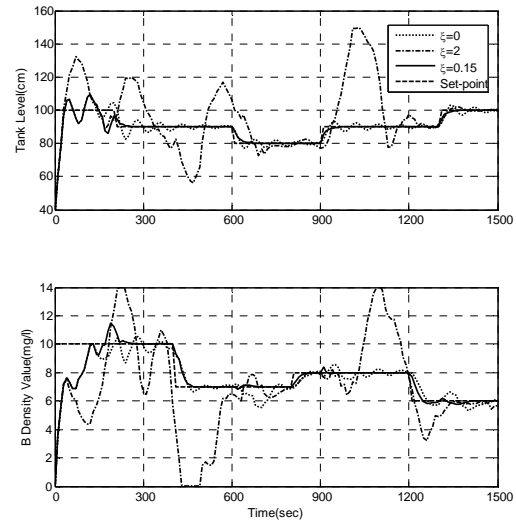


Fig. 5: Effects of different Adaptation Index on NMPC.

quadratic programming being added to RGPC to deal with input constraints, the control effect of NMPC ($\zeta=0.15$) and RGPC are demonstrated in Fig. 6. The corresponding manipulated variables are shown in Fig. 7. It can be seen that NMPC based on BP-ARX model provides better set-points tracking than RGPC from the point of quickly respond to set-points change and faster settling time. The poor tracking performance of the RGPC is due to the large changes in the process gain and model mismatch at the beginning of set-points change. On the other hand, the BP-ARX model can keep up with the system gain change over their entire operating regions, the model mismatch is completely eliminated.

To study the performance of the NMPC when there is unmeasured disturbances, disturbances in both C_{B1} and C_{B2} are considered. In the process block, C_{B1} and C_{B2} are changed to 20.0 and 0.2 respectively at $t=0$, while nominal values of 24.9 and 0.1 was used in the model. The control parameters are mostly the same as above except for $r=10$, $\zeta=0.015$ in NMPC and $q=10$ in RGPC. The control effects of NMPC and RGPC are demonstrated in Fig. 8, which shows that NMPC can track the tank level faster and smoother, track the concentration of B with smaller peak error and faster settling time than RGPC. Fig. 9 shows that NMPC has less fluctuation in manipulated variables than RGPC.

5. Conclusions

A novel BP-ARX nonlinear dynamic model is constructed for the purpose of NMPC. By estimating a second order ARX model on-line with RLS and rescaled the model's parameters according to the off-

line trained BP neural network steady-state model, the BP-ARX model can efficiently represent the dynamic behavior of the nonlinear processes over their entire operating regions. The Adaptation Index is defined to constrain the effect time of the RLS and improve the prediction accuracy of the BP-ARX model. Simulation results on a multivariable chemical process demonstrate that NMPC based on BP-ARX model can quickly respond to set-points change, together with the performance of smaller peak error, faster settling time and less fluctuation in manipulated variables for modeling errors.

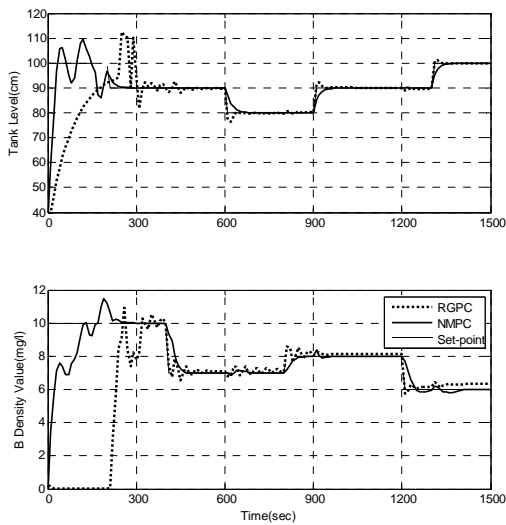


Fig. 6: Comparison of the control results obtained from NMPC and RGPC on chemical reactor system with set-points change.

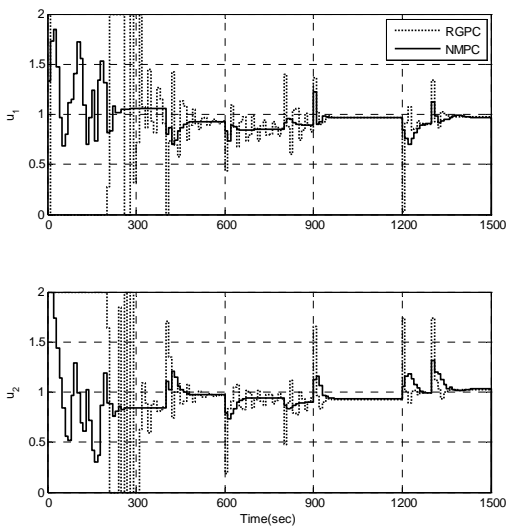


Fig. 7: Manipulated variables.

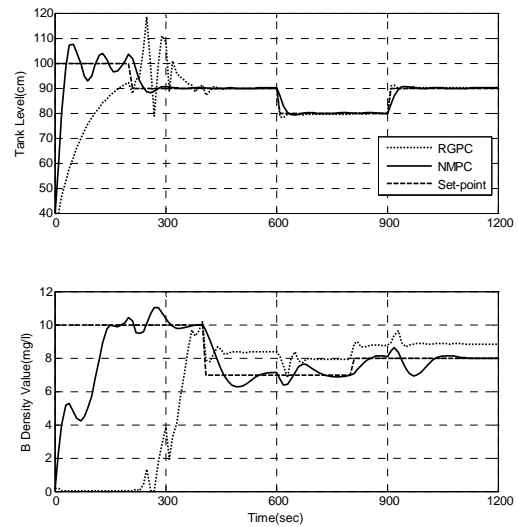


Fig. 8: Comparison of the control results obtained from NMPC and RGPC with set-points change in the presence of a modeling error in C_{B1} and C_{B2} .

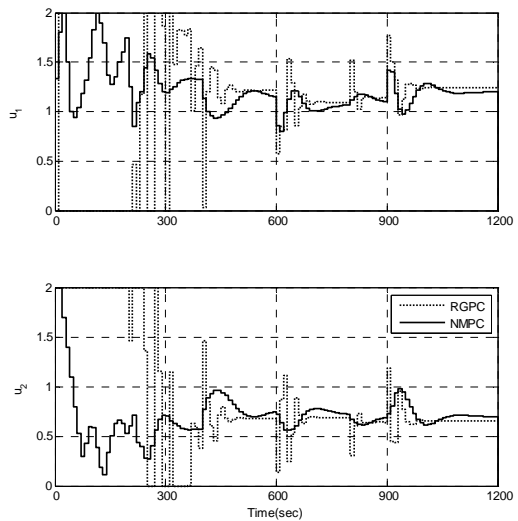


Fig. 9: Manipulated variables.

Acknowledgement

This work is partially supported by National Nature Science Foundation of China (Grant No. 60504004).

References

- [1] S.J. Qin and T. A. Badgwell, A survey of industrial model predictive control technology, *Control Engineering Practice*, 11:733-764, 2003.
- [2] M.A. Henson, Nonlinear model predictive control: current status and future directions,

- Computers and Chemical Engineering*, 23:187-202, 1998.
- [3] R.K. Pearson, Selecting nonlinear model structures for computer control, *Journal of Process Control*, 13:1-26, 2003.
 - [4] M. Morari and J. H. Lee, Model predictive control: past, present and future, *Computers and Chemical Engineering*, 23:667-682, 1999.
 - [5] F. D. Palma and L. Magni, A multi-model structure for model predictive control, *Annual Reviews in Control*, 28:47-52, 2004.
 - [6] E. Perez, X. Blasco, S. Garcia-Nieto and J. Sanchis, Diesel Engine Identification and Predictive Control using Wiener and Hammerstein Models, *Proc. Of the IEEE International Conference on Control Applications*, pp. 2417-2423, 2006.
 - [7] S. W. Wang, D. L. Yu, J. B. Gomm, G. F. Page and S. S. Douglas, Adaptive neural network model based predictive control for air-fuel ratio of SI engines, *Engineering Applications of Artificial Intelligence*, 19:189-200, 2006.
 - [8] B. Kadmiry and D. Driankov, A fuzzy flight controller combining linguistic and model-based fuzzy control, *Fuzzy Sets and Systems*, 146:313-347, 2004.
 - [9] M. Jalili-Kharaajoo, Predictive control of a continuous stirred tank reactor based on neuro-fuzzy model of the process. *Proc. Of the SICE 2003 Annual Conference*, pp. 3277-3282, 2003.
 - [10] L. Ljung, *System Identification: Theory for the User*, Second Edition, Beijing: Tsinghua University Press, 2002.
 - [11] S.M. Moon, R.L. Clark and D. G. Cole, The recursive generalized predictive feedback control: theory and experiments, *Journal of Sound and Vibration*, 279:171-199, 2005.
 - [12] M. Haeri and H.Z. Beik, Application of extended DMC for nonlinear MIMO systems, *Computers and Chemical Engineering*, 29:1867-1874, 2005.
 - [13] D.W. Clarke, C. Mohtadi and P. S. Tuffs, Generalized Predictive Control--Part I. The Basic Algorithm, *Automatica*, 23:137-148, 1987.
 - [14] C.M. Chow, A.G. Kuznetsov and D. W. Clark, Application of generalised predictive control to the paper machine benchmark, *Control Engineering Practice*, 3:1483-1486, 1995.