Batch-to-batch Control of Batch Processes Based on Multilayer Recurrent Fuzzy Neural Network

He Liu[1] **Li Jia**[2] **Qing Liu**[1] **Dao Huang**[1]

[1]School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, P. R. China

[2]Shanghai Key Laboratory of Power Station Automation Technology, Shanghai University, Shanghai 200072, P. R. China

Abstract

The batch-to-batch model-based iterative optimal control strategy for batch processes is realized based on multilayer recurrent fuzzy neural network (MRFNN) and chaotic search. MRFNNs are used to model batch processes. Modeling and optimization problems are mainly solved by chaotic search. Due to model-plant mismatches and disturbances, the calculated optimal control profile may not be optimal when applied to the actual process. Current predictions are improved by prediction errors from previous batches, and the model errors are gradually reduced from batch-to-batch. Furthermore, the control strategy is developed for temperature tracking control. The effectiveness is verified on simulated batch reactors.

Keywords: Batch-to-batch control, Iterative learning control, Recurrent neural network, Batch process

1. Introduction

In contrast to continuous processes, batch processes have strong nonlinear behaviors and always operate in transient states. A further difficulty in batch process control of final product quality is that product quality variables usually cannot be measured on-line and can only be obtained through laboratory analysis after a batch has finished [1]. Therefore, to build an accurate model that can provide accurate long term prediction is the base for optimal batch process control. To construct a mechanistic model for a batch system, challenging difficulties occur in determining coefficients of the reaction. Once the expression is available, the unknown correlated parameters have to be fitted by optimization or estimated [2].

To overcome these difficulties, neural network models based upon process operational data have been widely proposed as an alternative to the traditional mathematical models. To deal with temporal problems of dynamic systems, recurrent networks are commonly used. With feedbacks to the network input layer or hidden layers, recurrent neural network (RNN) can capture the nonlinearity and time-variance of dynamic system well. Mixed order locally recurrent neural network, global recurrent neural network and aggregated recurrent neural network are applied to the modeling and control of batch polymerization process by Zhang etc [1], [3], [4]. Interest in recurrent fuzzy neural networks (RFNNs) has been growing in recent years. RFNNs combined with T-S fuzzy model [5] and compensation-based fuzzy model [6] were proposed for dynamic system identification. In this paper, a multilayer recurrent neural network (MRFNN) with local feedbacks in both membership layer and rule layer is used to model batch reactors for providing long-term predictions.

Due to model-plant mismatches and unknown disturbances, the calculated optimal control profile may not be optimal when applied to the actual process. Because the batch processes are repetitive, the general idea of batch-to-batch or run-to-run optimization is using results from previous batches to find iteratively the optimal operating conditions, while performing the smallest number of suboptimal runs and preferably no unacceptable ones [7]. Various strategies have been proposed for batch-to-batch optimization. Recently, iterative learning control (ILC) has been introduced into batch process control. Lee and his co-workers proposed a model predictive control for batch processes (BMPC) approach with quadratic criterion for temperature control in batch processes [8]. Xiong and co-workers adopted the idea of ILC, proposed a batch-to-batch iterative optimal control strategy based on recurrent neural network models to improve product quality from batch-to-batch [7], [9]. The model predictions are iteratively modified by using errors of the network model during previous batch runs, and then updated control policy is calculated by SQP for each batch using the modified model predictions.

In this paper, the batch-to-batch iterative optimal control strategy for product quality control proposed by Xiong is realized based on multilayer recurrent fuzzy neural network (MRFNN) and optimization is performed by chaotic search algorithm. And then the control strategy for temperature control of batch processes is developed. In section 2, MRFNN architecture is presented for modeling batch processes. In section 3, chaotic search algorithm is presented for MRFNN parameters learning and updating control policy after each batch run. Section 4 introduces the control strategy for product quality control and the developed strategy for temperature control of batch processes. Applications of control strategies to simulated batch reactors are given in Section 5. In Section 6, conclusion is drawn.

2. MRFNN architecture

The *j*th fuzzy if-then rule with n inputs of T-S model is as follows:

Rule-*j*: IF $x_1(t)$ is A_{1j} and ... and $x_n(t)$ is A_{nj}

Then $y^j(t+1) = a_0^j + a_1^j x_1(t) + \cdots + a_n^j x_n(t)$

for $j = 1,\cdots,R$, where Rule-*j* denotes the *j*th fuzzy rule, $\underline{x}(t) = [x_1(t),\cdots,x_n(t)]^T$ is the input vector the to model at time t, $y^j(t+1)$ is the output of the *j*th rule, $A_{1j},\cdots,A_{nj}$ are the fuzzy sets, $a_0^j,\cdots,a_n^j$ are the consequent coefficients of Rule-*j*.

The proposed multilayer recurrent neural network has six layers. The MRFNN combines T-S fuzzy model with local feedbacks in both membership layer and rule layer. The structure is shown in Fig.1.

The Layer1 is the input layer. The inputs are received and transmitted to the next layer directly.

The Layer2 is the membership layer with a local recurrent structure. The output of this layer node is fed back to the membership layer node. The Gaussian-type membership function can be written as follows:

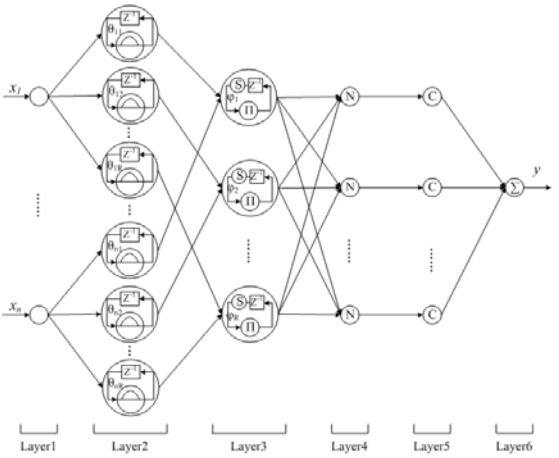

Fig.1: Structure of the proposed MRFNN.

$$\mu_{A_{ij}}(x_i(t)) = \exp\left\{\frac{-(x_i(t)-m_{ij}-\theta_{ij}\cdot\mu_{A_{ij}}(x_i(t-1)))}{\sigma_{ij}^2}\right\} \qquad (1)$$

where m_{ij} and σ_{ij} are the mean and variance of the Gaussian membership function of the *j*th term of the *i*th input x_i, respectively; θ_{ij} is the feedback weight. Therefore, the node output of this layer is the function of the current input, the node parameters and the previous node output.

The Layer3 is the rule layer with a local recurrent structure. There are R rules in this layer. A fuzzy region A_j is formed by "fuzzy-and" operation:

$$\mu_{A_j}(\underline{x}(t)) = \prod_{i=1}^{n}\mu_{A_{ij}}(x_i(t))\cdot s_j \qquad (2)$$

The above equation provides the degree to which a particular input vector $\underline{x}(t)$ belongs to the fuzzy region A_j. For the internal variable s_j, the following Sigmoid function is used:

$$s_j = 1/(1+\exp(-\mu_{A_j}(\underline{x}(t-1))\cdot\varphi_j)) \qquad (3)$$

where φ_j is the feedback weight.

The Layer4 is the normalization layer.

$$\overline{\mu_{A_j}}(\underline{x}(t)) = \mu_{A_j}(\underline{x}(t)) / \sum_{j=1}^{R}\mu_{A_j}(\underline{x}(t)) \qquad (4)$$

The Layer5 is the consequent layer. The node output of this layer is the output of each rule.

$$y^j(t+1) = \overline{\mu_{A_j}}(\underline{x}(t))\cdot(a_0^j + a_1^j x_1(t) + \cdots + a_n^j x_n(t)) \quad (5)$$

The Layer6 is the output layer.

$$y = \sum_{j=1}^{R} y^j(t+1) \qquad (6)$$

Therefore, modeling with MRFNN is the problem of optimizing premise parameters $\{m_{ij},\sigma_{ij},\theta_{ij},\varphi_j\}$ and consequent coefficients $\{a_i^j\}$.

In this study, MRFNN is used to model the nonlinear relationship between the input $U_k = [u_k(0),\cdots,u_k(N-1)]^T$ and the output $Y_k = [y_k(1),\cdots y_k(N)]^T$, where k is the batch index, u is the input (manipulated) variable for the output variable, y is the output variable of batch processes such as product quality, The batch run length t_f is fixed and divided into N intervals (i.e. $N = t_f / h$, h is the sampling time). After training, the current or previous predictions of MRFNN, in stead of the batch process output, are used as inputs of MRFNN for predicting next output. Therefore, predictions from MRFNN are multi-step ahead / long-term predictions.

3. Chaotic search

A chaotic motion can go non-repeatedly through every state in a certain domain. Chaos optimization algorithm was introduced by Li and Jiang [10] by incorporating chaotic dynamics into the optimized variables and then the optimum can be found based on chaotic search (CS). Because CS is simple and high effective, in recent years, it has drawn more and more attention. In this paper, modified CS is used not only as an optimization algorithm for optimization problem but also used for premise parameters learning of MRFNN modeling. The general idea of modified CS is the same. Here, we just describe the parameter learning algorithm, because it combines CS together with least square estimation (LSE), where CS is for optimization of premise parameters while LSE is for updating consequent coefficients for fast convergence.

For training MRFNN, the object is to find an appropriate set of parameters for minimum root mean square error (RMSE) between the desired outputs and the outputs of MRFNN. When the premise parameters are defined, the consequent parameters can be defined by LSE. Therefore, the object of parameter learning can be written as:

$$\min f(z,a) \tag{7}$$

where f denotes the RMSE between the desired outputs and the outputs of MRFNN; z represents for optimized premise parameters, $z_{i\min} \le z_i \le z_{i\max}$, $i = 1,\cdots,(3n+1)R$; a represents for consequent coefficients.

In this paper, the modified CS is based on the well-known Logistic map:

$$v^{t+1} = \mu v^t(1-v^t) \tag{8}$$

where μ is a control parameter, v is the chaotic variable represents for premise parameters, $v \in (0,1)$. The system represented by Eq. (8) is in a chaotic state when $\mu = 4$.

The parameter learning process is described as follows:

Step1. Initialization. Set iteration counter $t = 0$; initialize number M_1, M_2 of chaotic search and initial value of chaos variables v^0 (cannot be the fixed points of Logistic map i.e. 0.25, 0.5, 0.75); the best RMSE f^* is initialized as a big number; set disturbing coefficient $\alpha_1 = 0.05$ and decreasing coefficient $\beta = 0.9999$.

Step2. Chaotic searching.

Step2.1. Iterative searching. Set $t = t+1$, calculate new chaotic variables v^t by Eq. (8).

Step2.2. Variable mapping. Map the new chaotic variable into the optimized variable by linear mapping: $z_i^t = z_{i\min} + v_i^t(z_{i\max} - z_{i\min})$.

Step2.3. Searching for optimum. Based on present premise parameters x^t, consequent parameters a^t are estimated by LSE; and then compute $f(z^t,a^t)$. If $f(z^t,a^t) \le f^*$, set $f^* = f(z^t)$, $z^* = z^t$, $v^* = v^t$, $a^* = a^t$, $\alpha_1 = 0.05$, and go back to Step 2.1. Else if f^* isn't changed during M_1 times repeating step2, go to Step3.

Step3. Further searching. Add a disturb to v^* by $v^t = (1-\alpha_1)v^* + \alpha_1 v^t$, $\alpha_1 = \beta\alpha_1$, and go to step2.

With the search proceeding, the current optimum is deemed as approaching the best optimum more and more close, α_1 is reduced gradually. This helps searching follow different chaotic trajectory, and avoid the local optimum.

Step4. Judging. If f^* isn't changed during M_2 times repeating step3, the learning process is performed. Else return to Step2 to continuing searching.

4. Methodology

4.1. Batch-to-batch model-based iterative control for product quality

For the product quality control of batch processes, only the desired product quality variables at the end of a batch $y_d(t_f)$ are of interest, quality variables $y_k(t)$ is got on-line measured or analyzed after the *k*th batch. The batch-to-batch iterative optimization problem for product quality control can be formulated as:

$$\min_{U_k} J = \left\|\tilde{e}_k^f\right\|_Q^2 + \left\|\Delta U_k\right\|_R^2 \tag{9}$$

$$\text{s.t. } \tilde{e}_k^f = y_d(t_f) - \tilde{y}_k(t_f) \tag{10}$$

$$\Delta U_k = U_k - U_{k-1} \tag{11}$$

$$\hat{y}_k(t) = MRFNN[\hat{y}_k(t-1), u_k(t-1)] \tag{12}$$

$$\tilde{y}_k(t) = \hat{y}_k(t) + \alpha\bar{\hat{e}}_{k-1}(t) \tag{13}$$

$$\bar{\hat{e}}_{k-1}(t) = \frac{1}{\text{k - 1}}\sum_{i=1}^{k-1}\hat{e}_i(t) \tag{14}$$

$$\hat{e}_i(t) = y_i(t) - \hat{y}_i(t) \tag{15}$$

$$u^{low} \le U_k \le u^{hi} \tag{16}$$

$$y^{low} \le y_k(tf) \le y^{hi} \tag{17}$$

Given $y_i(t)$, $t \in (0,t_f)$, $i \le k$. Where $Q = \lambda_q \cdot I_n$ and $R = \lambda_r \cdot I_n$ are weighting matrices, $\tilde{e}_k^f$ is the difference between desired product qualities and modified predictions of MRFNN model $\tilde{y}_k(t_f)$,

$MRFNN[\cdot]$ represents the MRFNN model, $y_k(t)$ and $\hat{y}_k(t)$ are, respectively, the measured and predicted values of product quality variables at time *t* of *k*th batch, *t* is the discrete batch time, $\bar{\hat{e}}_{k-1}(t)$ is the average model error of all previous runs at time t, α_2 is a bias correction parameter and $0 < \alpha_2 \le 1$, u^{low} and u^{hi} are the lower and upper bounds of the input trajectory, y^{low} and y^{hi} are the lower and upper bounds of the final product qualities.

Here, the above optimization problem is solved by using chaotic search as described in Section 3. Xiong has proven that if Ineq. (18) holds, then $\lim_{k\to\infty} \Delta U_{k+1} = 0$. Therefore, we add Ineq. (18) to be another constrain of the optimization problem to make sure no worse solutions will be produced.

$$\left\| y_d - \tilde{y}_{k+1}(t_f) \right\| \le \left\| y_d - \tilde{y}_k(t_f) \right\|_Q^2 \qquad (18)$$

The control strategy is shown in Fig.2. It can be summarized as follows [7]:

Step1. At the current *k*th batch, the input U_k is implemented into the batch process and the outputs $y_k(t)$ are obtained by measurement or analysis.

Step2. Compare $y_k(t)$ with the model prediction $\hat{y}_k(t)$ and calculate the model prediction error $\hat{e}_k(t)$ using Eq. (15) and store them. Using $\hat{e}_k(t)$ and those prediction errors stored in all previous runs to calculation the average model error $\bar{\hat{e}}_{k-1}(t)$ using Eq. (14).

Step3. Modify the MRFNN model predictions for the next batch and obtain $\tilde{y}_{k+1}(t)$ from Eq. (13). Based on the modified predictions $\tilde{y}_{k+1}(t_f)$, the optimization problem specified by Eqs. (9)-(18) is solved and an updated open-loop input U_{k+1} for the (*k+1*)th batch is calculated.

Step4. Increase the batch index *k* by 1 and go to Step1.

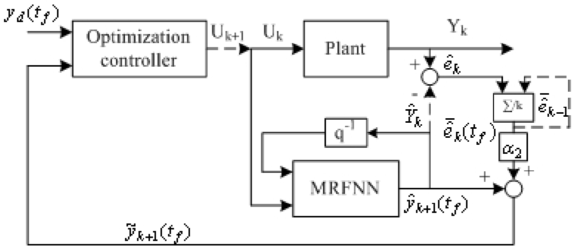

Fig. 2: Batch-to-Batch iterative optimal control diagram.

4.2. Batch-to-batch model-based iterative control for product quality

Based on the strategy described in 4.1 for product quality control of batch processes, a batch-to-batch model-based iterative control of temperature tracking for batch processes is proposed. For quality control, the main interest is on the final product quality and the reference trajectory is usually difficult to set. But for temperature control, the desired temperature trajectory is usually available.

Eqs. (9)-(10) and (17)-(18) are replaced by Eqs. (19)-(22). Then for every time *t*, the optimal input can be calculated by solving optimization problem specified by Eqs. (19)-(21) and (11)-(16). The control strategy is almost the same as described in Section 4.1 except that in Step (3), the optimization problem is solved based on the modified predictions $\tilde{y}_{k+1}(t)$ for every time *t*.

$$\min_{u_k(t)} J(t) = \left\| \tilde{e}_k(t) \right\|_Q^2 + \left\| \Delta u_k(t) \right\|_R^2 \qquad (19)$$

$$\text{s.t. } \tilde{e}_k(t) = y_d(t) - \tilde{y}_k(t) \qquad (20)$$

$$y^{low} \le y_k(t) \le y^{hi} \qquad (21)$$

$$\left\| y_d(t) - \tilde{y}_{k+1}(t) \right\| \le \left\| y_d(t) - \tilde{y}_k(t) \right\|_Q^2 \qquad (22)$$

5. Simulation

5.1. Production quality control of a batch reactor

The typical batch reactor is taken from [9,11]. The reaction scheme is $A \xrightarrow{k_1} B \xrightarrow{k_2} C$, the equations describing the reactor are:

$$\begin{aligned} \dot{x}_1 &= -4000\exp(-2500/T)x_1^2 \\ \dot{x}_2 &= 4000\exp(-2500/T)x_1^2 \\ &\quad - 6.2\times10^5\exp(-5000/T)x_2 \end{aligned} \qquad (23)$$

where x_1 and x_2 represent the dimensionless concentrations of A and B, *T* is the temperature of the reactor. For simplicity, *T* is scaled to dimensionless value as $T = (T - T_{\min})/(T_{\max} - T_{\min})$, the values of $T_{\min}$ and $T_{\max}$ are 298K and 398K respectively. The initial conditions are $x_1(0) = 1$ and $x_2(0) = 1$, and the constraint on the control $u = T_d$ is $0 \le T_d \le 1$. The performance index is to maximize the concentration of B $x_2(t_f)$ at the end of batch. The inputs and output of MRFNN model are $[T(t)\, x_2(t)]^T$ and $x_2(t+1)$ respectively.

11 batches of data are obtained from the mechanistic model (23) with T ranging from $T_{\min}$ to $T_{\max}$. There are 100 sets data in every batch with sampling time 0.01. Normally distributed random noise with zeros means with covariance 0.002^2 was added to corrupt x_2. Among these data, 10 batches were used for training and the remaining batch was used for testing. Set $M_1 = M_2 = 20$ and 5 rules in MRFNN, after learning, the modeling RMSE is 0.0011. Then the built MRFNN model is used for predicting tested by the unseen testing batch. The predicting RMSE is 0.0073, and the long-range prediction of x_2 from MRFNN on the unseen testing data is shown in Fig. 3. The modeling and testing results show that the built MRFNN model can fit training data well with good prediction ability.

The batch length is divided into $N = 10$ equal stages. Set $y_d(t_f) = 0.43$, $\alpha_2 = 0.1$, $\lambda_q = 2000$ and $\lambda_r = 10$. Due to the MRFNN model-plant mismatches, the final product concentration optimized based on MRFNN is 0.4208. The calculated control policy is used as the first run of batch-to-batch iterative optimization based on MRFNN. After batch-to-batch iterative optimization for 7 batch runs, the product concentration is improved to 0.4267. The convergence of temperature trajectory is shown in Fig.4 and the tracking error is shown in Fig.5.

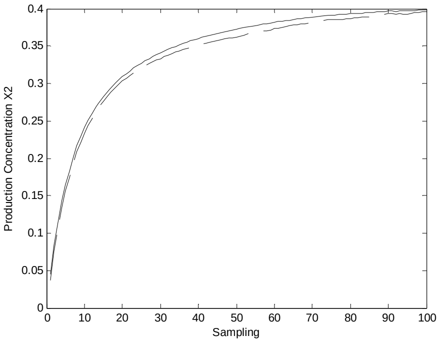

Fig.3: MRFNN predicting performance on product quality.

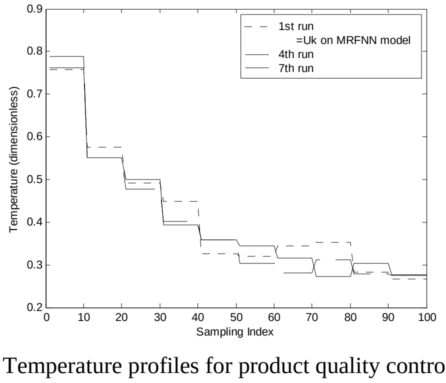

Fig.4: Temperature profiles for product quality control.

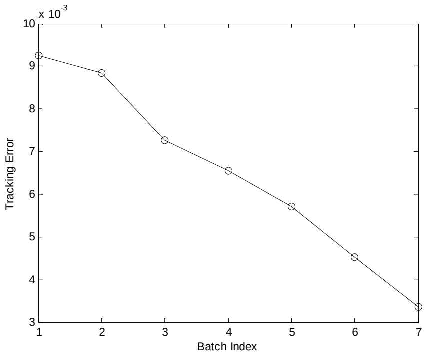

Fig.5: Tracking error for product quality control.

5.2. Production quality control of a batch reactor

A batch reactor is taken from [8] where a exothermic reaction $A \to B$ takes place. It is assumed that the reactor has a cooling jacket whose temperature is directly manipulated. The following equations describe the dynamics of the reactor system:

$$\begin{aligned} \frac{dT}{dt} &= -\frac{UA}{MC_p}(T - T_j) + \frac{(-\Delta H)V}{MC_p} k_0 e^{-E/RT} C_A^2 \\ \frac{dC_A}{dt} &= -k_0 e^{-E/RT} C_A^2 \end{aligned} \qquad (24)$$

where, $\frac{UA}{MC_p} = 0.09$ (l/min), $\frac{(-\Delta H)V}{MC_p} = 1.64$ (Kl/mol), $k_0 = 2.53 \times 10^{19}$ (l/mol·min), $E/R = 13550$ (°K), $T(0) = T_I = 25$ (℃), $C(0) = C_{AI} = 0.9$ (mol/l).

12 batches of data are obtained from the mechanistic model (23) with T_j ranging from 15 to 40. There are 80 sets data in every batch with sampling time is 1. Normally distributed random noise with zeros means with covariance 0.1^2 was added to corrupt T. Among these data, 10 batches were used for training and the remaining 2 batches were used for testing. The inputs and output of MRFNN model are $[T(t)\,T_{\mathrm{j}}(t)]^T$ and $T(t+1)$ respectively. Set $M_1 = M_2 = 20$ and 5 rules in MRFNN, after learning, the modeling RMSE is 0.2918. Then the built MRFNN model is used for predicting tested by the unseen testing batches, and the predicting RMSE is 0.6350.

Set $\alpha_2 = 0.25$, $\lambda_q = 1$ and $\lambda_r = 0.1$. Due to the model-plant mismatches, the optimal input T_j computed based on MRFNN model applied to the batch reactor leads big errors, see the first run shown in Fig.6. After iterative optimization for 7 runs, the actual temperature *T* can track the reference trajectory well although small tracking errors still exist.

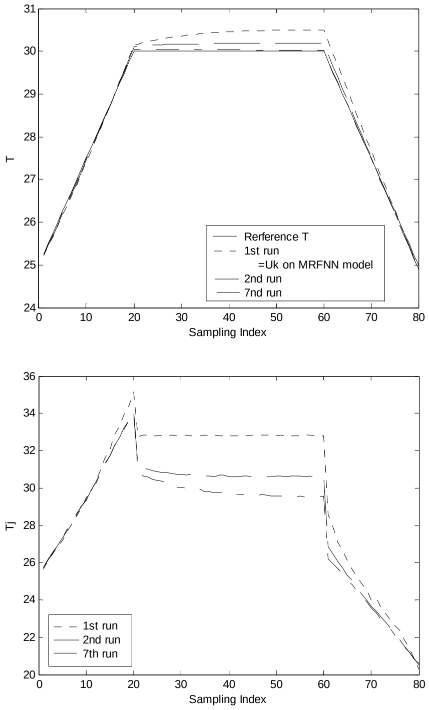

Fig.6: Results of iterative temperature control.

6. Conclusions

The batch-to-batch control model-based iterative optimal control strategy for product quality control of batch processes is realized based on multilayer recurrent fuzzy neural network (MRFNN) and chaotic search. MRFNN are used to model batch processes for long-range predictions. With local feedback connections in both membership layer and rule layer, MRFNN can solve the temporal problem of batch processes. The premise parameters of MRFNN are learned by modified chaotic search (CS) algorithm, while consequent coefficients are updated by least square estimation (LSE) for fast convergence. To find the optimal control profile is the optimization problem, which is solved by CS. Due to model-plant mismatches and unmeasured disturbances, the calculated optimal control profile may not be optimal when applied to the actual process. Current predictions are improved by prediction errors from previous batches, and the model errors are gradually reduced from batch-to-batch. This control strategy is then extended to address the temperature tracking control problem of batch processes. The effectiveness is verified on simulated batch reactors.

References

[1] J. Zhang, Batch process modeling and optimal control based on neural network models. *Acta Automatica Sinica*, 31:19-31, 2005.

[2] J. S. Chang, S. C. Lu, Y. L. Chiu, Dynamic modeling of batch polymerization reactors via the hybrid neural-network rate-function approach. *Chemical Engineering Journal*, 130:19-28, 2007.

[3] J. Zhang, A. J. Morris, E. B. Martin, Long-term prediction models based on mixed order locally recurrent neural networks. *Computers and Chemical Engineering,* 22: 1051-1063, 1998.

[4] J. Zhang, A neural network-based strategy for the integrated batch-to-batch control and within-batch control of batch processes. *Transaction of the Institute of Measurement and Control*, 27: 391-410, 2005.

[5] C.J. Lin and C. C. Chin, Prediction and identification using wavelet-based recurrent fuzzy neural networks. *IEEE Transaction on Systems Man and Cybernetics - Part B: Cybernetics*, 34: 2144-2154, 2004.

[6] C.J. Lin, C. H. Chen, A compensation-based recurrent fuzzy neural network for dynamic system identification. *European Journal of Operational Research*, 172: 696–715, 2006.

[7] Z.H. Xiong, J. Zhang, A batch-to-batch iterative optimal control strategy based on recurrent neural network models. *Journal of Process Control*, 15: 11-21, 2005.

[8] J.H. Lee, K. S. Lee, W. C. Kim, Model-based iterative learning control with a quadratic criterion for time-varying linear systems. *Automatica*, 36: 641-657, 2000.

[9] Z. H. Xiong, J. Zhang, X. Wang and Y. M. Xu, Run-to-run iterative optimization control of batch processes based on recurrent neural networks. *Advances in Neural Networks - ISNN2004, Lecture Notes in Computer Science*, pp. 97-103, 2004.

[10] B. Li, W.S. Jiang, Optimizing complex function by chaos search. *Cybern Syst.*, 29: 409-419, 1998.

[11] W.H. Ray, editor. *Advanced Process Control*, McGraw-Hill, New York, 1981.