Batch-to-Batch Iterative Learning Control for End-Point Qualities Based on Kernel Principal Component Regression Model

Ganping Li
School of Information Engineering, Nanchang University, Nanchang City, P. R. China
Email: gplee1972@sina.com

Abstract—**A batch-to-batch model-based iterative learning control (ILC) strategy for the end-point product quality control in batch processes is proposed in this paper. A nonlinear model for end-point product quality is developed from process operating data using kernel principal component regression (KPCR). The ILC algorithm is derived to calculate the control policy by linearizing the KPCR model around the nominal trajectories and minimising a quadratic objective function concerning the end-point product quality. To overcome the detrimental effects of unknown process variations or disturbances, it is proposed in the paper that the KPCR model should be updated in a batchwise manner by removing the earliest batch data from the training data set and adding the latest batch data to the training data set. The ILC based on updated KPCR model shows adaptability for process variations or disturbances when applied to a simulated batch polymerization process. Comparisons between KPCR model and principal component regression (PCR) model based ILCs are also made in the simulations.**

Index Terms—**Iterative Learning Control, Kernel Principal Component Regression, Batch Process**

I. INTRODUCTION

Batch processes are very important in the industry to manufacture the low-volume and high-value products such as biochemicals, crystals and some speciality chemicals. Generally, most agile manufacturings are realized depending on batch processes because of their flexibility in handling many products. Those factors such as shorter product life cycle and more adaptive ability of time-to-market of batch processes are competitive factors in successful factories. Because batch processes playa more and more important role in economic growth, the control of batch processes and their optimal operation have received a great deal of attention in past decades.

Batch processes have some features not found in continuous processes, such as strong nonlinearity and without steady state of the system, which make batch process be difficult to control. With the development of control theories in past decades, newly emerged nonlinear control techniques were applied to batch processes, such as differential geometric approaches [1-3], nonlinear predictive control [4][5] and generic model control (GMC)[6]. However, these approaches rely on accurate process models. If model mismatches and disturbances exist, they may not be feasible to apply these methods for real processes. When a mechanistic model is not available or difficult to develop, data-driven model based control scheme provides a viable alternative for solving control problems associated with a nonlinear batch process [7].

Recently, a control strategy called "(ILC)" has been applied to batch process control. ILC was originally developed to improve the operation of robot manipulators under repetitive operations [8]. Since batch processes are intended to be run repeatedly, ILC can be extended to control batch processes. As far as ILC for batch process is concerned, the ILC is employed to track a desired trajectory or improve the product qualities batch-to-batch by feeding the previous output error back to the present batch, so that the output converges to the desired trajectory or product qualities. Such a kind of ILCs is referred as batch-to-batch ILC. Most ILC strategies are based on models, such as system inversion [9] or model predictive control (MPC) [10-12]. Since the development and validation of an accurate mechanistic model are very hard and the remaining uncertainties may still be high, data-driven based empirical model, on the other hand, have the advantages of ease in use. Thus empirical model-based ILC approaches are developed very fast in recent years. Among them, different ANNs were used in batch-to-batch ILC design due to their non-linear regression abilities [13-15].

In recent years, multivariable statistical technique such as partial least squares (PLS) has been incorporated into the ILC design for batch processes [16][17]. However, PLS is a linear approach that cannot describe nonlinear processes accurately. Recently, kernel methods such as kernel principal component analysis (KPCA) [18] and kernel partial least squares (KPLS) [19] are developed very fast in signal processing area. The basic idea of kernel methods is to first map the input space into a feature space via a nonlinear map which can be realized by a nonlinear kernel function and, then, extract the dominant components in the feature space. Kernel methods such as KPCA and KPLS are the nonlinear extensions of PCA and PLS in kernel feature subspace. By now, KPCA and KPLS have been applied to the monitoring [20] and control [21] of batch processes. Although some ILC strategies can reject persistent

doi:10.4304/jcp.8.12.3184-3190

disturbances of batch processes, to design a more adaptive ILC scheme is still of great importance. Because KPCA has showed good advantage of capturing process dynamics, thus KPCA can be incorporated into ILC design to achieve more adaptive control.

In this article, a batch-to-batch ILC strategy based on KPCR model is proposed. The training data from several normal batches are used to build the model base on the KPCR technique. Under the ILC strategy, the end-point qualities of the batch product will converge to the target value. Additionally, by updating the model batch-to-batch, the ILC can overcome the persistent batch-to-batch disturbances or process variations effectively. The ILC strategy is more adaptive than the ILC based on PCR model.

This paper is organized as follows. KPCR modeling technique is introduced in Section 2. ILC strategy based on KPCR model is proposed in Section 3, and then Section 4 gives a simulation example. Finally, the conclusions are made in Section 5.

II. KPCR

KPCR is a nonlinear extension of PCR in kernel feature subspace. Consider a nonlinear mapping

$$\Phi : R^m \to F, \quad x \mapsto \overline{X}. \tag{1}$$

where $\Phi(\cdot)$ is a nonlinear mapping function that projects sample data x from the input space to the high-dimensional feature space F. If the mapped data $\Phi(\mathbf{x})$ are centered in feature space F, that is $\frac{1}{n}\sum_{i=1}^{n}\Phi(\mathbf{x}_i)=0$, n is the sample number, then we have the covariance matrix in F:

$$\overline{\boldsymbol{C}} = \frac{1}{n}\sum_{i=1}^{n}\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_i)^T \tag{2}$$

The first step of KPCR is to perform KPCA for sample matrix $\mathbf{X}_{n\times m}$, which corresponds to solve the eigenvalue problem of $\overline{\boldsymbol{C}}$:

$$l\mathbf{v} = \overline{\boldsymbol{C}}\mathbf{v} \tag{3}$$

where l is the eigenvalue and $\mathbf{v}$ is the corresponding eigenvector of $\overline{\boldsymbol{C}}$. All solutions $\mathbf{v}$ with $l \neq 0$ lie in the span of $\Phi(\mathbf{x}_1),\Phi(\mathbf{x}_2),\cdots,\Phi(\mathbf{x}_n)$. Equation (3) is equivalent to the following eigenvalue problem [18]:

$$nl\boldsymbol{\alpha} = \boldsymbol{K}\boldsymbol{\alpha} \tag{4}$$

where $\boldsymbol{\alpha}=[\alpha_1,\alpha_2,\cdots,\alpha_n]^T$, α_i is the coefficient vector such that

$$\mathbf{v} = \sum_{i=1}^{n}\alpha_i\Phi(\mathbf{x}_i) \tag{5}$$

$\boldsymbol{K}$ is the $n\times n$ Gram kernel matrix defined as:

$$[\boldsymbol{K}]_{ij} = \boldsymbol{K}_{ij} = \left\langle\Phi(\mathbf{x}_i),\Phi(\mathbf{x}_j)\right\rangle = \boldsymbol{K}(\mathbf{x}_i,\mathbf{x}_j)$$

The PC vectors can be obtained in feature space as follows:

$$\begin{aligned}\beta_k(\mathbf{x}) &= \left\langle\mathbf{v}_k,\Phi(\mathbf{x})\right\rangle = \sum_{i=1}^{n}\alpha_i^k\left\langle\Phi(\mathbf{x}_i),\Phi(\mathbf{x})\right\rangle \\ &= \sum_{i=1}^{n}\alpha_i^k\hat{K}(\mathbf{x}_i,\mathbf{x})\end{aligned} \tag{6}$$

where k is the number of PCs reserved, $k=1,2,\cdots,r$. $\hat{\boldsymbol{K}}$ is the mean centered matrix of $\boldsymbol{K}$,

$$\hat{\boldsymbol{K}} = \boldsymbol{K} - \boldsymbol{E}_N\boldsymbol{K} - \boldsymbol{K}\boldsymbol{E}_N + \boldsymbol{E}_N\boldsymbol{K}\boldsymbol{E}_N \tag{7}$$

where

$$\boldsymbol{E}_N = \frac{1}{n}\begin{bmatrix}1 & \cdots & 1\\ \vdots & \ddots & \vdots\\ 1 & \cdots & 1\end{bmatrix}_{n\times n}. \tag{8}$$

The second step of KPCR is to perform linear regression between the nonlinear principal components and the observations in feature space. The linear regression model in feature space is:

$$\mathbf{Y} = \boldsymbol{\Phi}\boldsymbol{\xi} + \boldsymbol{\varepsilon} \tag{9}$$

where $\mathbf{Y}$ is a vector of n observations of an output, $\boldsymbol{\Phi}$ is $(n\times m)$ matrix of regressors whose i-th raw is $\boldsymbol{\Phi}(\mathbf{x}_i)$, $\boldsymbol{\xi}$ is a vector of regression coefficients, and $\boldsymbol{\varepsilon}$ is a vector of errors.

By projection of all original regressors onto the principal components, (9) can be rewritten as:

$$\mathbf{Y} = \boldsymbol{\Gamma}\mathbf{w} + \boldsymbol{\varepsilon} \tag{10}$$

where $\boldsymbol{\Gamma}=\boldsymbol{\Phi}\mathbf{V}$ is an $(n\times m)$ matrix of transformed regressors, and $\mathbf{V}$ is an $(m\times m)$ matrix whose kth column is the eigenvector $\mathbf{V}_k$. Define $\boldsymbol{A}=[\alpha_1,\alpha_2,\cdots,\alpha_n]$, then $\boldsymbol{\Gamma}$ can be calculated as [22]:

$$\begin{aligned}\boldsymbol{\Gamma} &= \hat{\boldsymbol{K}}\boldsymbol{A} \\ &= \sum_{i=1}^{n}\alpha_i\hat{\boldsymbol{K}}(\mathbf{x}_i,\mathbf{x})\end{aligned} \tag{11}$$

The least squares estimate of the coefficient vector $\mathbf{w}$ is:

$$\hat{\mathbf{w}} = (\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T\mathbf{Y} \tag{12}$$

Finally, using the first r nonlinear principal components in feature space, the KPCR model can be formulated as [23]:

$$\begin{aligned}y &= f(\mathbf{x},c) \\ &= \sum_{k=1}^{r}\hat{\mathbf{w}}_k\beta_k(\mathbf{x}) + b\end{aligned} \tag{13}$$

$$= \sum_{k=1}^{r} \hat{\mathbf{w}}_k \sum_{i=1}^{n} \alpha_i^k \hat{\boldsymbol{K}}(\mathbf{x}_i, \mathbf{x}) + b$$

$$= \sum_{i=1}^{n} c_i \hat{\boldsymbol{K}}(\mathbf{x}_i, \mathbf{x}) + b$$

where $\{c_i = \sum_{k=1}^{r} \hat{\mathbf{w}}_k \alpha_i^k\}_{i=1}^{n}$, b is a bias term.

III. Batch-to-batch ILC for end-point quality control

A. Batch-to-Batch ILC Based on Fixed KPCR Model

In batch processes, the end of batch product quality generally depends on the initiate condition and the control profile, which can be formed as

$$\mathbf{y}(t_f) = \boldsymbol{F}(\mathbf{X}_0, \mathbf{U}) \tag{14}$$

where $\mathbf{y}(t_f) = [y_1(t_f), y_2(t_f) \cdots y_N(t_f)]$ is a vector of end-point product qualities at batch end time t_f. $\mathbf{X}_0$ is the initiate condition of the batch process and $\mathbf{U} = [u_1, u_2, \cdots, u_L]^T$ is a vector of control inputs by dividing a batch process into L time segments of equal length. The nonlinear function vector $\boldsymbol{F}(\cdot,\cdot)$ is represented by KPCR model here based on training data of a batch process.

Based on the KPCR model formulated by (13), the optimal control policy $\mathbf{U}$ can be obtained by solving the following optimization problem:

$$\min_{\mathbf{U}} J[\mathbf{y}(t_f)] \tag{15}$$

s.t. product quality and process constraints

Generally, there are model mismatches and disturbances in real plants which will deteriorate the performance of a batch process if present control policy are applied. To overcome this problem, an ILC strategy is used for the improvement of the performance from batch to batch in the presence of model mismatches and disturbances. The ILC strategy uses information from previous batches to improve the operation of the next batch.

The first order Taylor series expansion of (14) around a nominal control profile can be expressed as [17]:

$$\hat{\mathbf{y}}(t_f) = \boldsymbol{F}_0 + \frac{\partial \boldsymbol{F}}{\partial u_1}\Delta u_1 + \frac{\partial \boldsymbol{F}}{\partial u_2}\Delta u_2 + \cdots + \frac{\partial \boldsymbol{F}}{\partial u_L}\Delta u_L \tag{16}$$

The actual end-point product quality for the kth batch can be written as

$$\mathbf{y}_k(t_f) = \hat{\mathbf{y}}_k(t_f) + \boldsymbol{e}_k \tag{17}$$

where $\mathbf{y}_k(t_f)$ are the actual product qualities and $\hat{\mathbf{y}}_k(t_f)$ are the predicted product qualities at the end of the kth batch respectively, and $\boldsymbol{e}_k$ is the model prediction error.

From (16), the prediction for the kth batch can be approximated using the first order Taylor series expansion based on the (k-1)th batch:

$$\begin{aligned}\hat{\mathbf{y}}_k(t_f) &= \hat{\mathbf{y}}_{k-1}(t_f) + \frac{\partial \boldsymbol{F}}{\partial u_1}\Big|_{U_{k-1}}(u_1^k - u_1^{k-1}) + \\ &\frac{\partial \boldsymbol{F}}{\partial u_2}\Big|_{U_{k-1}}(u_2^k - u_2^{k-1}) + \cdots + \frac{\partial \boldsymbol{F}}{\partial u_L}\Big|_{U_{k-1}}(u_L^k - u_L^{k-1}) \\ &= \hat{\mathbf{y}}_{k-1}(t_f) + \boldsymbol{G}_k^T \Delta \mathbf{U}^k\end{aligned} \tag{18}$$

where

$\Delta \mathbf{U}^k = [\Delta u_1^k \quad \Delta u_2^k \quad \ldots \quad \Delta u_L^k]^T$,

$$\boldsymbol{G}_k^T = \left[\frac{\partial \boldsymbol{F}}{\partial u_1}\Big|_{U_{k-1}} \quad \frac{\partial \boldsymbol{F}}{\partial u_2}\Big|_{U_{k-1}} \quad \cdots \quad \frac{\partial \boldsymbol{F}}{\partial u_L}\Big|_{U_{k-1}}\right]^T.$$

The control input can be calculated employing the conventional quadratic objective function:

$$\min_{\Delta \mathbf{U}^k} J_k = \left\| \mathbf{y}_d - \mathbf{y}_{k-1}(t_f) - \boldsymbol{G}_k^T \Delta \mathbf{U}^k \right\|_Q^2 + \left\| \Delta \mathbf{U}^k \right\|_R^2 \tag{19}$$

where $\mathbf{y}_d$ is the objective value, $\boldsymbol{Q}$ is a weighting matrix for the end state errors and $\boldsymbol{R}$ is a weighting matrix for the control effort.

For the unconstrained case, set $\partial J / \partial \Delta \mathbf{U}^k = 0$, an analytical solution to the above minimization can be obtained as

$$\Delta \mathbf{U}^k = (\boldsymbol{G}_k \boldsymbol{Q} \boldsymbol{G}_k^T + \boldsymbol{R})^{-1} \boldsymbol{G}_k \boldsymbol{Q} (\mathbf{y}_d - \mathbf{y}_{k-1}(t_f)) \tag{20}$$

$$\mathbf{U}^k = \mathbf{U}^{k-1} + \Delta \mathbf{U}^k \tag{21}$$

Kernel function $\boldsymbol{K}(\mathbf{x}, \mathbf{x}_i)$ is a symmetric function satisfying the Mercer's condition. Different kernel functions can be selected to construct different KPCR model. Here we select RBF kernel function $\boldsymbol{K}(\mathbf{x}, \mathbf{x}_i) = \exp(-\left\| \mathbf{x} - \mathbf{x}_i \right\|_2^2) / 2\sigma^2$ to build KPCR model. The gradient of model output with respect to $\mathbf{U}$ and $\boldsymbol{G}$, can be calculated as

$$\begin{aligned}\boldsymbol{G}_k &= \frac{\partial \boldsymbol{F}}{\partial \mathbf{U}}\Big|_{\mathbf{U}_{k-1}} = \sum_{i=1}^{n} c_i \frac{\partial \boldsymbol{K}(\mathbf{U}, \mathbf{U}_i)}{\partial \mathbf{U}}\Big|_{\mathbf{U}_{k-1}} \\ &= \sum_{i=1}^{n} -c_i \frac{(\mathbf{U}_{k-1} - \mathbf{U}_i)}{\sigma^2} \exp(-\frac{\left\| \mathbf{U}_{k-1} - \mathbf{U}_i \right\|_2^2}{2\sigma^2})\end{aligned} \tag{22}$$

where $\boldsymbol{G}_k$ is the gain matrix of the ILC.

B. Batch-to-Batch ILC Based on updated KPCR Model

Generally there are unknown process variations and disturbances in real batch processes. To overcome the detrimental effects of unknown process variations or disturbances, an adaptive batch-to-batch ILC scheme based on updated KPCR models is proposed here. To capture the changed dynamics of batch processes, the KPCR model is updated from batch-to-batch by the method that the earliest batch

data are removed from the data set and the latest batch data are added to the data set to rebuild the KPCR model. The procedure can be formulated as

$$\begin{aligned} &\mathbf{X}_{k+1}^{(i)} = \mathbf{X}_{k}^{(i+1)}, i = 1, \cdots, n-1; \\ &\mathbf{X}_{k+1}^{(n)} = \mathbf{U}^{k} \end{aligned} \tag{23}$$

and

$$\begin{aligned} &\mathbf{Y}_{k+1}^{(i)} = \mathbf{Y}_{k}^{(i+1)}, i = 1, \cdots, n-1; \\ &\mathbf{Y}_{k+1}^{(n)} = \mathbf{y}_{k}(t_f) \end{aligned} \tag{24}$$

where i denotes the ith row of the data matrix which corresponds to the ith sample.

The updated KPCR model can be established using the same KPCR technique mentioned in section 2. The ILC algorithm based on the updated KPCR model can be reformulated as

$$\Delta \mathbf{U}^{k} = [\tilde{\boldsymbol{G}}_k \boldsymbol{Q} \tilde{\boldsymbol{G}}_k^{T} + \boldsymbol{R}]^{-1} \tilde{\boldsymbol{G}}_k \boldsymbol{Q} (\mathbf{y}_d - \mathbf{y}_{k-1}(t_f)) \tag{25}$$

where $\tilde{\boldsymbol{G}}_k$ is the updated gain matrix.

IV. Numerical Simulations

In this section, the proposed ILC method is tested through simulations for the thermally initiated bulk polymerization of styrene in a batch reactor [21]. The state equation describing the polymerization system is shown as follows:

$$\begin{aligned} \dot{x}_1 &= \frac{\rho_0^2 \rho}{M_{\mathrm{m}}} (1-x_1)^2 \exp(2x_1 + 2M_{\mathrm{n}} x_1^2) A_{\mathrm{m}} \exp(-\frac{E_{\mathrm{m}}}{T}) \\ \dot{x}_2 &= \frac{\dot{x}_1 x_2}{1+x_1} \left(1 - \frac{1400 x_2}{A_{\mathrm{w}} \exp(B/T)} \right) \\ \dot{x}_3 &= \frac{\dot{x}_1}{1+x_1} \left(\frac{A_{\mathrm{w}} \exp(B/T)}{1500} - x_3 \right) \\ \rho &= \frac{1-x_1}{r_1 + r_2 T_{\mathrm{c}}} + \frac{x_1}{r_3 + r_4 T_{\mathrm{c}}} \\ \rho_0 &= r_1 + r_2 T_{\mathrm{c}} \\ T_{\mathrm{c}} &= T - 273.15 \end{aligned} \tag{26}$$

where x_1 is the conversion, $x_2 = x_{\mathrm{n}} / x_{\mathrm{nf}}$ and $x_3 = x_{\mathrm{w}} / x_{\mathrm{wf}}$ are the dimensionless number-average and weight-average chain lengths, respectively. $u = T / T_{\mathrm{ref}}$ is the dimensionless reactor temperature as control variable; T_{c} is the temperature in degrees Celsius; A_{w} and B are coefficients in the relation between the weight-average chain lengths and the temperature obtained from experiments; A_{m} and E_{m} are the frequency factor and energy of the overall monomer reaction respectively; the constants $r_1 - r_4$ are density-temperature corrections; M_{m} and M_{n} are the monomer molecular weight and polymer-monomer interaction parameter, respectively; and t_f is the final time of the batch. The initial values of the states are $x_1(0) = 0$, $x_2(0) = 1$, and $x_3(0) = 1$. The process parameter values are given in Table 1.

TABLE I.
Parameter values for the batch polymerization process

Parameter	Value
M_{m}	104 kg/mol
A_{w}	0.033454
A_{m}	4.26×10^5 m^3/(kmol s)
E_{m}	10103.5 K
B	4364 K
r_1	0.9328×10^3 kg/m^3
r_2	-0.87902 kg/(m^3 °C)
r_3	1.0902×10^3 kg/m^3
r_4	-0.59 kg/(m^3 °C)
T_{ref}	399.15 K
M_{n}	0.33
t_f	300 min
x_{nf}	700
x_{wf}	1500

To apply the ILC strategy, the sample time of the batch process is set as 20 minutes. The control profile is divided into 15 equal stages. During each stage, the control variable is kept constant. Then the control input is formed as $\mathbf{U} = [u_1, u_2, \cdots, u_{15}]^T$ (L=15). The control outputs are the end-point qualities $\mathbf{y}(t_f) = [x_1(t_f) \;\; x_2(t_f) \;\; x_3(t_f)]^T$.

In order to generate the training data to build the KPCR model, random changes with normal distribution ($NID(0, 0.1^2)$) are added to the nominal trajectory and data of 30 batch runs are generated as training data. Then a KPCR model is built from the training data which takes $\mathbf{U}(30 \times 15)$ as the model input and $\mathbf{y}(t_f)(30 \times 3)$ as the model output. Table 2 shows the root mean squared (RMS) errors of the KPCR model outputs and the PCR model outputs. It can be seen from Table 2 that KPCR model has more accuracy than PCR model.

The ILC based on the fixed KPCR model are applied to the batch process. First, we test the ILC performance under normal condition. The ILC uses the law of (20)

TABLE II.
Modeling errors using PCR and KPCR technique in polymerization

$\mathbf{y}(t_f)$	PCR model		KPCR model	
	RMS training error	RMS testing error	RMS training error	RMS testing error
$x_1(t_f)$	0.0044	0.0090	0.0002	0.0072
$x_2(t_f)$	0.0050	0.0114	0.0002	0.0077
$x_3(t_f)$	0.0034	0.0069	0.0002	0.0050

where $\boldsymbol{G}_k$ is derived from a fixed KPCR model. The desired end-point qualities are set as $\mathbf{y}_d = [0.8 \quad 1 \quad 1]^T$. The weighting matrix $\boldsymbol{Q}$ is selected as $\boldsymbol{Q} = diag(1000, 5100, 75000)$, and the weighting matrix $\boldsymbol{R}$ is selected as $\boldsymbol{R} = 0.08\boldsymbol{I}$. The ILC performance is compared with that using PCR model. The results are shown in Fig. 1. It can be seen from Fig. 1 that the RMS control errors ($\|\mathbf{y}_d - \mathbf{y}_k(t_f)\|_2$) of both control strategies converge quickly within several batches. But as demonstrated in Fig. 1, the ILC based on KPCR model converges faster than the ILC based on PCR model and has less RMS errors.

To illustrate the control performance of the adaptive ILC law of (25) under disturbance condition, two disturbance cases are considered: disturbance case 1, constant batchwise disturbances; case 2, batch-to-batch

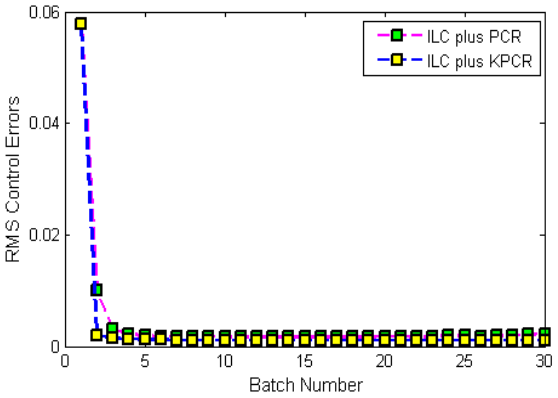


Figure 1. Comparison of RMS control errors using fixed PCR model based ILC and fixed KPCR model based ILC under normal condition

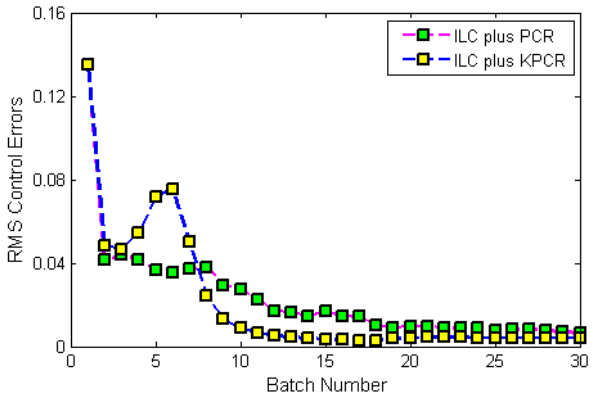


Figure 2. Comparison of RMS control errors using updated PCR model based ILC and updated KPCR model based ILC in disturbance case 1

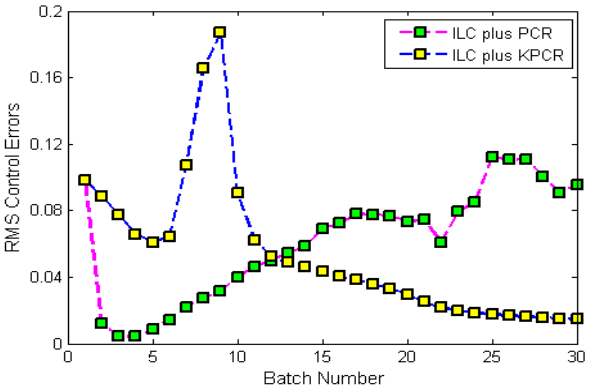


Figure 3. Comparison of RMS control errors using updated PCR model based ILC and updated KPCR model based ILC in disturbance case 2

parametric disturbances. In disturbance case 1, it is assumed that the system is affected by a constant batchwise disturbance in A_w (+10%). The desired end-point qualities are also set as $\mathbf{y}_d = [0.8 \quad 1 \quad 1]^T$. The weighting matrix $\boldsymbol{Q}$ is selected as $\boldsymbol{Q} = diag(900, 15000, 15000)$, and $\boldsymbol{R}$ is selected as $\boldsymbol{R} = 0.01\boldsymbol{I}$. The results of updated PCR model based and updated KPCR model based ILC strategies are shown in Fig. 2. It can be seen from Fig. 2 that in the previous seven batch runs, updated KPCR model based ILC shows larger error than updated PCR model based ILC, but in the next batch runs, the updated KPCR model based ILC converges more quickly than the updated PCR model based ILC.

In disturbance case 2, we assume A_w first ramps up and then stays constant after 15 batch runs are completed. The disturbance case with batch-to-batch parametric changes is formulated as

$$A_w = \begin{cases} A_{w0} \times \left(1 + 0.01 \times (k-1)\right), k \le 15 \\ 1.14 A_{w0}, k > 15 \end{cases} \tag{27}$$

where A_{w0} is the nominal value of parameter A_w given in Table 1.

In this case, the desired end-point product qualities are still set as $\mathbf{y}_d = [0.8 \quad 1 \quad 1]^T$. The weighting matrix $\boldsymbol{Q}$ is selected as $\boldsymbol{Q} = diag(11220, 22851, 22364)$, and $\boldsymbol{R}$ is selected as $\boldsymbol{R} = 0.01\boldsymbol{I}$. The performance of updated PCR model based ILC and that of updated KPCR model based ILC are shown in Fig. 3. It can be seen from Fig. 3 that, although the RMS errors of updated KPCR model based ILC are larger than those of updated PCR model based ILC in the previous 12 batch runs, the error gradually

TABLE III.
THE END-POINT QUALITIES IN THE 30TH BATCH RUN OF ALL CASES USING PCR MODEL BASED ILC AND KPCR MODEL BASED ILC

$\mathbf{y}_{30}(t_f)$			with disturbances			
	without disturbances		Case1		Case2	
	PCR-ILC	KPCR- ILC	PCR-ILC	KPCR- ILC	PCR-ILC	KPCR- ILC
$x_1(t_f)$	0.7907	0.7923	0.8053	0.7991	0.8162	0.8099
$x_2(t_f)$	0.9981	0.9974	1.0005	1.0007	1.0576	1.0089
$x_3(t_f)$	0.9923	0.9915	1.0032	0.9984	1.0747	1.0073

decreases after the 9th batch run as batch number increases afterwards. In contrast, although the RMS error of updated PCR model based ILC decreases in the first four batch runs, it increases gradually when batch number increases afterwards. It indicates that updated KPCR model based ILC can adapt the process variations in the case while updated PCR model based ILC cannot. The results demonstrate that KPCR model can capture the batchwise changed dynamics more effectively than PCR model and leads to more adaptive performance control. Table 3 presents the control performances of the two ILC strategies in the 30th batch run of all simulated cases.

V. Conclusions

A batch-to-batch model-based iterative learning control strategy for the tracking control of the end point product quality of batch processes is proposed. To address the problem of nonlinearities in batch processes, a nonlinear model for end-point product quality prediction, linearized around the nominal batch trajectories, is identified from process operating data using KPCR technique. On the basis of the linearized KPCR model, an ILC law is obtained explicitly by calculating the optimal control profile. To address the problems of unknown disturbances and process variations, the KPCR model should be updated from batch to batch to capture the changed batch process dynamics. After each batch run, the latest data are added to the data set and the earliest data are removed from the data set to rebuild the model. Based on the updated model, an adaptive ILC is derived to overcome process variations or disturbances. The proposed technique is illustrated on a simulated batch polymerization process. The results demonstrate that the ILC using KPCR model can adapt process variations or disturbances better than the ILC using PCR model.

Acknowledgment

This work was supported by the Chinese 863 Program under Grant 2013AA013804 and the National Natural Science Foundation of China under Grant 61064004.

References

[1] T. Clarke-Pringle, and J. F. MacGregor, “Nonlinear adaptive temperature control of multi-product, semi-batch polymerization reactors,” *Computers & Chemical Engineering*, vol. 21, pp.1395–1409, 1997.

[2] C. Gentric, P. Filiberto, M. A. Latifi, and J. P. Corriou, “Optimization and non-linear control of a batch emulsion polymerization reactor,” *Chemical Engineering Journal*, vol. 75, pp.31–46, 1999.

[3] W. Xie, S. Rohani, and A. Phoenix, “Extended kalman filter based nonlinear geometric control of a seeded batch cooling crystallizer,” *The Canadian Journal of Chemical Engineering*, vol. 80, pp.167–172, 2002.

[4] M. V. Lelann, M. Cabassud, and G. Casamatta, “Modeling, optimization and control of batch chemical reactors in fine chemical production,” *Annual Reviews in Control*, vol. 23, pp.25–34, 1999.

[5] J. Valappil, and C. Georgakis, “State estimation and nonlinear model predictive control of end-use properties in batch reactors,” *Proceedings of the American Control Conference*, pp.999–1004, June 2001.

[6] N. M. Aziz, M. A. Hussain, and I. M. Mujtaba, “Performance of different types of controllers in tracking optimal temperature profiles in batch reactors,” *Computers and Chemical Engineering*, vol. 24, pp.1069–1075, 2000.

[7] K. Y. Rani, and S. C. Patwardhan, “Data-driven model based control of a multi-product semi-batch polymerization reactor,” *Chemical Engineering Research and Design*, vol. 85, pp.1397–1406, 2007.

[8] Suguru Arimoto, Sadao Kawamura, and Fumio Miyazaki, “Bettering operation of robots by learning,” *Journal of Robotic systems*, vol. 1, pp.123–140, 1984.

[9] K. L. Moore, *Iterative Learning Control for Deterministic System*, Springer, New York, 1993.

[10] K.S. Lee, I. S. Chin, H. J. Lee, and Jay H. Lee, “Model predictive control technique combined with iterative learning for batch processes,” *AIChE Journal*, vol. 45, pp.2175–2187, 1999.

[11] J. H. Lee, K. S. Lee, and W. C. Kim, “Model-based iterative learning control with a quadratic criterion for time-varying linear systems,” *Automatica*, vol. 36, pp.41–659, 2000.

[12] K. S. Lee, and J. H. Lee, “Iterative learning control-based batch process control technique for integrated control of end product properties and transient profiles of process variables,” *Journal of Process Control*, vol. 13, pp.607–621, 2003.

[13] Z. H. Xiong, and J. Zhang, “A batch-to-batch iterative optimal control strategy based on recurrent neural network models,” *Journal of Process Control*, vol. 15, pp.11–21, 2005.

[14] J. Zhang, “Batch-to-batch optimal control of a batch polymerisation process based on stacked neural network models,” *Chemical Engineering Science*, vol. 63, pp.1273–1281, 2008.

[15] Z. H. Xiong, Y. X. Xu, J. Zhang, and J. Dong, “Batch-to-batch control of fed-batch processes using control-affine feedforward neural network,” *Neural Computing & Applications*, vol. 17, pp.425–432, 2008.

[16] J. F. Cerrillo, and J. F. MacGregor, “Iterative learning control for final batch product quality using partial least squares models,” *Industrial and Engineering Chemistry Research*, vol. 44, pp. 9146–9155, 2005.

[17] J. H. Chen, and K. C. Lin, “Batch-to-batch iterative learning control and within-batch on-line control for end-point qualities using MPLS-based dEWMA,” *Chemical Engineering Science*, vol. 63, pp.977–990, 2008.

[18] B. Schölkopf, A. Smola, and K. B. Müller, “Nonlinear component analysis as a kernel eigenvalue problem,” *Neural Computation*, vol. 10, pp.1299–1319, 1998.

[19] R. Rosipal, and L. J. Trejo, “Kernel partial least squares regression in reproducing kernel hilbert space,” *Journal of Machine Learning Research*, vol. 2, pp.97–123, 2001.

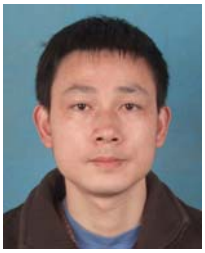
[20] J. M. Lee, C. K. Yoo, and I. B. Lee, “Fault detection of batch processes using multiway kernel principal component analysis,” *Computers and Chemical Engineering*, vol. 28, pp.1837–1847, 2004.

[21] Y. W. Zhang, Y. P. Fan, and P .C. Zhang, “Combining kernel partial least-squares modeling and iterative learning control for the batch-to-batch optimization of constrained nonlinear processes,” *Industrial Engineering and Chemical Research*, vol. 49, pp.7470–7477, 2010.

[22] X. Zhang, S. L. Ma, W. W. Yan, X. Zhao, and H. H. Shao, “A novel systematic method of quality monitoring and prediction based on FDA and kernel regression,” *Chinese*

Journal of Chemical Engineering, vol. 17, pp.427–436, 2009.

[23] R. Rosipal, M. Girolami, L. J. Trejo, and A. Cichocki, "Kernel PCA for feature extraction and de-noising in nonlinear regression," *Neural Computing & Applications*, vol. 10, pp.231–243, 2001.



Ganping Li was born in Nanchang City, China on June 4, 1972. He received the B.S. degree in electrical engineering from Tianjin University, Tianjin, China, in 1995, the M.S. degree in chemical mechanical engineering from Nanchang University, Nanchang, China, in 2002, and the Ph.D. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2007.

From 1995 to 1999, he was an Assistant Engineer at Tianjin Electrical Drive Design and Research Institute, Tianjin, China. He is currently a lecturer of electrical engineering at Nanchang University, Nanchang, China. His technical interests are in intelligent control, process control and optimization.