Online Prediction of Quality-Related Variables for Batch Processes using a Sequential Phase Partition Method

Zheng Li,[1,2,3,4] Pu Wang,[1,2,3,4] Xuejin Gao,[1,2,3,4]* Yongsheng Qi[5] and Peng Chang[1,2,3,4]

1. *Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China*
2. *Engineering Research Centre of Digital Community, Ministry of Education, Beijing 100124, China*
3. *Beijing Laboratory for Urban Mass Transit, Beijing 100124, China*
4. *Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124, China*
5. *School of Electric Power, Inner Mongolia University of Technology, Hohhot 010051, China*

Batch processes inherently have multiple operation phases; different phases exhibit different characteristics. Hence, it is reasonable to partition the process into phases and build sub-phase models for online quality prediction. To this end, a sequential phase partition method based on the information increment is proposed. To address the multiphase behaviours in batch processes, this work utilizes a new information increment index to capture the dynamic characteristics of batch processes along a time direction and divides the process into sub-phases. Next, phase-based multiway partial least squares (MPLS) models are built to model within-phase characteristics and predict the quality-related variables online. Information increment is able to exploit the process evolution by focusing on the changing variable correlations derived from two adjacent extend time slice. It directly utilizes the available process measurements of successful history batch processes without data transformation or dimensionality reduction. The method is sequential and can overcome the limits of some phase partition methods that may divide the samples with discontinuous time sequence but similar characteristics into the same phase. In addition, the information increment is capable of reflecting the change of the process intuitively with high computation efficiency. Advantages of the proposed method are illustrated by two case studies, a penicillin simulation platform and an industrial application of *Escherichia coli* (*E. coli*) fermentation, respectively.

Keywords: batch processes, extend time slice, quality prediction, sequential phase partition

INTRODUCTION

Batch processes are widely used in fine chemistry, biochemical, pharmaceutical, and semiconductor industries. Unlike continuous processes, which treat high throughput as the main production goal, batch processes are considered as fine processing and are expected to be associated with high product quality. For most batch processes, guaranteeing the product quality is more economical than attaining a high throughput. High product quality is reflected particularly in consistent products, which means that the final products of different batches approach the same product specification at the termination of the batch. Quality control can guarantee consistent product and cost-saving. Generally, every link of a batch process in production plants is processed strictly according to the formulated production specifications, under the assumption that the desired product quality can be achieved by repeating the fixed discrete inputs. These policies may not be robust enough to overcome disturbances. Closed-loop control, which is often known as feedback control, is capable of rejecting disturbances by adjusting the input trajectory. Trajectory-tracking control and direct quality control are typical methods of closed-loop quality control. However, quality-related variables, i.e., important variables for product quality and quality variables, are often not measured online, however, this should be done when applying closed-loop quality control. Important variables that can reflect the final product quality, such as biomass concentration, are usually not available online, impeding their direct use in trajectory-tracking control or direct quality control. Although some variables can be measured online by expensive hardware analyzers, most hardware analyzers are difficult to maintain and still cause measurement delay. Quality variables are usually measured in the lab after a batch run, making it another key challenge in direct quality control. The online prediction of quality-related variables is serves as a premise of quality control, and this has become a crucial task in batch processes quality modelling and control.

Quality prediction approaches can be roughly classified into two categories: mechanism methods; and data-driven methods. Mechanism methods provide the most intuitive way to estimate quality-related variables, in which models are constructed by analyzing the chemical or physical reaction in the process. However, it is usually difficult to establish accurate mathematical models since complex biochemical reactions exist in most batch processes. Furthermore, it is impractical to carry out the exact parameters of mathematical models through costly production experiments. Through the use of accessible accumulated historical data, data-driven methods have become

* Author to whom correspondence may be addressed. E-mail address: gaoxuejin@bjut.edu.cn

Can. J. Chem. Eng. 97:2483–2497, 2019

DOI 10.1002/cjce.23494
Published online 05 April 2019 in Wiley Online Library
(wileyonlinelibrary.com).

an alternative means of solving the problems mentioned above. Data-driven based methods do not require prior knowledge of the process mechanism and only use the data obtained in the operation process. Partial least squares (PLS), principal component regression (PCR), artificial neural network (ANN), and subspace identification are effective data-driven quality prediction methods.[1–9] The methods mentioned above have attracted much attention for batch process modelling, monitoring, and quality prediction in recent years. The MPLS technique has become the most widely used multivariate statistical analysis technique for the quality prediction of batch processes.

The traditional MPLS model is built under the basic assumption that the whole batch data should come from a single operation phase. In reality, batch processes inherently have multiple operation phases, termed multiphase batch processes, due to the change of feed volume. Thus, the above assumption can be invalid. A similar scenario can be found in continuous processes, which is usually referred to as multimode. Indeed, many researchers have pointed out that the variable relationships may change among different operating phases.[10] As a result, using the entire batch data to establish a single model for prediction is insufficient to model the process and can inevitably cause prediction accuracy loss. Multimodel approaches divide the data into different groups such that each group represents a single operation phase/mode.[11] Multimodel approaches can be further classified into adaptive model methods and phase-based model methods. For adaptive model methods, such as just-in-time learning (JITL), a local model is built utilizing the most relevant available data whenever an estimation of a variable is needed. The correct selection of a sample is of importance for local modelling. Yuan et al.[12] proposed a double locally weighted sample selection method in which the samples for local modelling are selected in the supervised latent structure. Samples with different similarities are assigned with corresponding weights. The method evaluates the sample importance by these weights. However, these adaptive model methods are always time consuming when applied online, influencing the real time performance of the quality control system. A reasonable alternative method that is well suited to the process characteristic is to divide the batch process procedure into multiple phases and assign each phase a static model, which is referred to as phase-based model methods or phase partition methods.[13–16] Clustering-based methods, the Gaussian mixture model (GMM), and the hidden Markov model (HMM) are widely used phase partition methods. For continuous processes, phase partition methods are also used to identify different modes. Lou and Wang[17] presented a hidden semi-Markov model (HSMM). In the HSMM, each phase is represented by a group generated from the history data and the mode duration probability is integrated into the HMM for mode identification. Yu and Qin[18] presented a phase partition method based on the Gaussian mixture model (GMM). Lu et al.[19,20] proposed phase partition methods based on k-means clustering. In their works, k-means clustering is performed on the PCA loading matrices or PLS regression coefficients of time slices to divide the process into different phases. Hu et al.[21] proposed a phase partition method based on affinity propagation clustering (AP). Qi et al.[22] divided the time slices into different phases using fuzzy c-means clustering (FCM). However, these clustering-based phase partition methods can hardly cope with the sequence of process data. The samples that share similar characteristics can be divided into the same phase, but they may be far away from each other along the sample time direction. This may be accompanied by post-processing. To handle the sequence issue in batch processes, Luo et al.[23] presented a fuzzy phase partition method using the sequence-constrained fuzzy c-means (SCFCM) clustering algorithm. It integrates a sequential constrain into the object function of FCM, in order to improve the insufficient of clustering methods in terms of sequence problems.[23] The main drawbacks of these clustering-based methods is that the number of clusters needs to be determined in advance and the clustering results are easily affected by the selected similarity index. Camacho et al.[24] presented a multiphase (MP) method by searching the appropriate phase switch points that can improve the prediction accuracy of the global PLS or principal component analysis (PCA) model. However, it has a low computation efficiency. Ge et al.[25] presented a sequential phase partition method based on the repeatability factor of time slice. Zhao[26] and Qin et al.[27] proposed a series of step-wise sequential phase partition methods in which the switch points are decided in the time sequence by evaluating the quality prediction performance. However, the methods mentioned above determine whether or not the current sample time is a switch point by evaluating the prediction performance. It may not be sensitive enough to capture the dynamic characteristics of the process and may lead to time delay, to some extent. Batch processes exhibit dynamic characteristics. The correlation or autocorrelation between process variables will change in real time, which results in phase characteristics.[19] In the meantime, most of the above phase partition methods treat all of the process variables at each sample time as a macro unit for analyzing and they compute the similarity or build a model based on these units. This may neglect the influence of the correlation information between measurable process variables at each sample time, or the correlations between the measurable process variables and quality variables. In addition, the complex interactions between the hard-to-measure important variables and quality variables may also have an effect on the phase partition results, which is seldom explored.

In this article, we present an alternative phase partition method for predicting quality-related variables of multiphase batch processes using information increment matrix-partial least squares (IIMPLS). Instead of a traditional time slice that only consists of process variables, the extended time slice that consists of both process variables and quality variables is first proposed. It provides a new way to interpret the time varying relationship between the process variables and the quality-related variables, which may have a significant influence on the phase partition results. Secondly, the information increment index, which is derived from two adjacent extend time slices, is proposed to explore the dynamic change of the process evolution intuitively. Next, according to the trajectory of the information increment index, the process is divided into several sub-phases in the time sequence using a sliding time window. Afterwards, phase-based MPLS models are built and are used to estimate the quality-related variables online. The feasibility and effectiveness of the proposed method are illustrated by a penicillin simulation platform and an industrial application of *E. coli* fermentation, respectively.

The remainder of the article is organized as follows: The phase partition method based on the information increment is illustrated first; Then, the offline sub-phases modelling strategy that is used for predicting the quality-related variables is provided; Afterwards, the online prediction framework is developed; Finally, two experiments, i.e., a penicillin simulation platform and an industrial *E. coli* fermentation process, are used to validate the effectiveness of the new method.

METHODOLOGY

PLS and MPLS Method

PLS is one of the most widely used multivariate statistical analysis methods. It is gaining importance in fields of chemistry and industrial process control. PLS was firstly used by Wold et al.[28] in chemical applications. It extracts the information from X that is more relevant to Y, making it outperform the classical multiple linear regression and principal component regression in terms of the ability to interpret Y.

Unlike continuous processes data that is collected in two-dimensional arrays with attributes of variable and time, batch process data is in three-dimensional arrays with an additional attribute of batch. That means batch process data can be arranged in a three-way matrix X (batch × variable ×time). Batch-to-batch variations, which is also termed as the uneven lengths or uneven duration problem, could exist in many batch processes. Effective methods have been proposed to align batch data.[29–31] In addition, a series of dynamic models that do not suffer from this limitation have also been proposed recently. Corbett and Mhaskar[8] proposed a subspace identification-based modelling method to capture the dynamic evolution of a batch process and relate the subspace states to product quality. The method was further extended to account for more complicated mid-batch additions and address batch duration.[9] However, these methods mainly focus on quality control. In this article, the batch processes are assumed to have been aligned already, and thus all batches are supposed to have the same total sample time. MPLS is an extension of PLS and is more suitable to deal with three-dimensional batch process data. We can use MPLS to extract the information from the process measurement variables trajectories that is more relevant to the quality-related variables. It transforms the three-dimensional matrix into a two-dimensional matrix using a certain unfolding method and then perform PLS on the unfolded two-dimensional matrix.[32] In general, three methods that unfold the three-dimensional matrix X into a two-dimensional matrix are used in most of the research, i.e., NM unfolding, WFKH unfolding, and AT unfolding. NM unfolding is also called batch-wise unfolding, which was proposed by Nomikos and MacGregor.[33] Mean centering and normalizing are performed on each column of the two-dimensional matrix in order to emphasize the deviation of each batch from the average trajectory. It can eliminate the main non-linearity due to the dynamic behaviour of the process, but some sample measurements need to be estimated when predicting quality online. WFKH unfolding, which is often referred to as variable-wise unfolding, can avoid the sample estimation problem in NM unfolding. It was first proposed by Wold et al.[34] Mean centering and normalizing are performed on all of the measurements of each variable that was collected through all of the sample times of all of the batches, which can weaken the time varying characteristics of each variable. A third method, referred to as AT unfolding, draws advantages of the above two methods.[35] In AT unfolding, the three-dimensional matrix is pre-processed as with batch-wise unfolding, and then the pre-processed data is rearranged variable-wise. Therefore, not only the deviations from the average trajectory are studied but also the time varying characteristics of each variable.

In this article, we successfully perform AT unfolding on the three-dimensional matrix X of batch processes. Suppose a historical dataset of batch trajectory data consist of I batches is denoted a three-dimensional matrix $\tilde{X}(I \times J_x \times K)$, where J_x process variables are measured online over K sample times throughout the batch run. $\tilde{X}(I \times J_x \times K)$ is firstly unfolded into a two-dimensional matrix batch-wise, which can be further split into K time slices, $\tilde{X}_k(I \times J_x)$ $(k = 1, \ldots, K)$. Each time slice is consists of process variables measured online at the kth $(k = 1, \ldots, K)$ sample time of the normal batches. $\tilde{X}_k(I \times J_x)$ is then mean centered and normalized to $X_k(I \times J_x)$. Similarly, $\tilde{Y}(I \times J_y \times K)$ is the historical dataset of quality data and consists of I batches where J_y quality-related variables are measured offline over K sample time intervals. Two possible situations are illustrated here. One situation is that the important variables for product quality are assumed to be measured offline at every sample time. Another is that only the final measurements of the quality variables are available. For the latter, we can duplicate a final measurement for every sample time. The same preprocesses are performed on $\tilde{Y}(I \times J_y \times K)$ to achieve time slice $\tilde{Y}_k(I \times J_y)$ and the normalized $Y_k\left(I \times J_y\right)$. Thus, the first step of AT unfolding, batch-wise unfolding, has been performed on the three-dimensional matrix as a premise of the subsequent phase partition method.

Information Incremental Matrix

The information increment matrix utilizes the covariance matrix of process variables to capture the dynamic characteristics of the system along the time direction. The existing information increment matrix-based methods only concentrate on the process variables for fault diagnosis.[16,36] In our article, we integrate quality-related variables into time slices in order to emphasize the time varying relationship between the process variables and the product quality in batch processes. The relationship may have a significant influence on phase partition results. Extend time slice $\boldsymbol{Z}_k$ is defined as follows:

$$\boldsymbol{Z}_k = (\boldsymbol{X}_k | \boldsymbol{Y}_k) = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1J_x} & | & y_{11} & \cdots & y_{1J_y} \\ x_{21} & x_{22} & \cdots & x_{2J_x} & | & y_{21} & \cdots & y_{2J_y} \\ \cdot & \cdot & & & | & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & | & \cdot & \cdot & \cdot \\ \cdot & \cdot & & & | & \cdot & & \cdot \\ x_{I1} & x_{I2} & \cdots & x_{IJ_x} & | & y_{I1} & \cdots & y_{IJ_y} \end{pmatrix} \in R^{I \times J} \tag{1}$$

where $J = J_x + J_y$ and $k = 1, \ldots, K$. The covariance matrix of $\boldsymbol{Z}_k$ is then calculated as follows:

$$
\begin{aligned}
\boldsymbol{R}_k &= \frac{1}{I-1}\boldsymbol{Z}_k^T\boldsymbol{Z}_k \\
&= \frac{1}{I-1}\begin{pmatrix}
x_{11} & x_{12} & \cdots & x_{1J_x} & | & y_{11} & \cdots & y_{1J_y} \\
x_{21} & x_{22} & \cdots & x_{2J_x} & | & y_{21} & \cdots & y_{2J_y} \\
\cdot & \cdot & & & | & \cdot & & \cdot \\
\cdot & \cdot & \cdot & & | & \cdot & \cdot & \cdot \\
\cdot & \cdot & & & | & \cdot & & \cdot \\
x_{I1} & x_{I2} & \cdots & x_{IJ_x} & | & y_{I1} & \cdots & y_{IJ_y}
\end{pmatrix}^T
\begin{pmatrix}
x_{11} & x_{12} & \cdots & x_{1J_x} & | & y_{11} & \cdots & y_{1J_y} \\
x_{21} & x_{22} & \cdots & x_{2J_x} & | & y_{21} & \cdots & y_{2J_y} \\
\cdot & \cdot & & & | & \cdot & & \cdot \\
\cdot & \cdot & \cdot & & | & \cdot & \cdot & \cdot \\
\cdot & \cdot & & & | & \cdot & & \cdot \\
x_{I1} & x_{I2} & \cdots & x_{IJ_x} & | & y_{I1} & \cdots & y_{IJ_y}
\end{pmatrix} \\
&= \left(\begin{array}{cccc|ccc}
D(X_1) & cov(X_1,X_2) & \cdots & & cov(X_1,Y_1) & & cov(X_1,Y_{J_y}) \\
cov(X_2,X_1) & D(X_2) & \cdots & & cov(X_2,Y_1) & \cdots & cov(X_2,Y_{J_y}) \\
\cdot & \cdot & \cdots & & \cdot & \cdots & \cdot \\
\cdot & \cdot & \cdots & & \cdot & \cdots & \cdot \\
\cdot & \cdot & \cdots & & \cdot & \cdots & \cdot \\
\hdashline
cov(Y_1,X_1) & cov(Y_1,X_2) & \cdots & & D(Y_1) & \cdots & cov(Y_1,Y_{J_y}) \\
\cdot & \cdot & \cdots & & \cdot & \cdots & \cdot \\
\cdot & \cdot & \cdots & & \cdot & \cdots & \cdot \\
\cdot & \cdot & \cdots & & \cdot & \cdots & \cdot \\
cov(Y_{J_y},X_1) & cov(Y_{J_y},X_2) & \cdots & & cov(Y_{J_y},Y_1) & \cdots & D(Y_{J_y})
\end{array}\right) \\
&= \left(\begin{array}{c|c}
\boldsymbol{A}^k_{J_x\times J_x} & \boldsymbol{B}^k_{J_x\times J_Y} \\
\hdashline
\left(\boldsymbol{B}^k_{J_x\times J_Y}\right)^T & \boldsymbol{C}^k_{J_y\times J_Y}
\end{array}\right) \in R^{J\times J}
\end{aligned}
\tag{2}
$$

where $\boldsymbol{R}_k$ is a real symmetric matrix; $\boldsymbol{A}_{J_x\times J_x}$ is the submatrix that characterizes the correlation between process variables, in which the elements indicate the covariance between any two process variables or the variance of each process variable is displayed. $\boldsymbol{B}_{J_x\times J_y}$ is the submatrix that characterizes the correlation between process variables and quality-related variables. Each element $cov(X_i, Y_i)$ in $\boldsymbol{B}_{J_x\times J_y}$ represents the covariance between any one of the process variables $X_i(i = 1, \ldots, J_x)$ and any one of the quality-related variables $Y_j(j = 1, \ldots, J_y)$. $\boldsymbol{C}_{J_y\times J_y}$ is the submatrix that reflects the covariance between any two quality-related variables or the variance of each quality-related variable. Complex interactions between hard-to-measure important variables and quality variables can be presented via $\boldsymbol{C}_{J_y\times J_y}$. Furthermore, the information incremental matrix $\boldsymbol{D}_k$ is defined as follows:

$$
\begin{aligned}
\boldsymbol{D}_k &= \boldsymbol{R}_{k+1} - \boldsymbol{R}_k \\
&= \left(\begin{array}{c|c}
\boldsymbol{A}^{k+1}_{J_x\times J_x} - \boldsymbol{A}^k_{J_x\times J_x} & \boldsymbol{B}^{k+1}_{J_x\times J_Y} - \boldsymbol{B}^k_{J_x\times J_Y} \\
\hdashline
\left(\boldsymbol{B}^{k+1}_{J_x\times J_Y}\right)^T - \left(\boldsymbol{B}^k_{J_x\times J_Y}\right)^T & \boldsymbol{C}^{k+1}_{J_y\times J_Y} - \boldsymbol{C}^k_{J_y\times J_Y}
\end{array}\right) \\
&= \left(\begin{array}{c|c}
\boldsymbol{A}^k & \boldsymbol{B}^k \\
\hdashline
\left(\boldsymbol{B}^k\right)^T & \boldsymbol{C}^k
\end{array}\right) \\
&= \left(\begin{array}{c|c}
\boldsymbol{A}^k & 0 \\
\hdashline
0 & 0
\end{array}\right) + \left(\begin{array}{c|c}
0 & \boldsymbol{B}^k \\
\hdashline
\left(\boldsymbol{B}^k\right)^T & 0
\end{array}\right) + \left(\begin{array}{c|c}
0 & 0 \\
\hdashline
0 & \boldsymbol{C}^k
\end{array}\right)
\end{aligned}
\tag{3}
$$

where $\boldsymbol{D}_k$ is the differential matrix of sample time $k + 1$ and sample time k $(k = 1, \ldots, K - 1)$. It keeps changing throughout the entire operating procedure. Each element in $\boldsymbol{D}_k$ represents the differential of correlation coefficients between variables, which can characterize the dynamic variation of the process. To illustrate it more specifically, $\boldsymbol{A}^k$ captures the dynamic changing relationships of process variables, $\boldsymbol{B}^k$ captures the dynamic changing relationships between each process variable and each quality-related variable, and $\boldsymbol{C}^k$ explores the dynamic changing relationships of quality-related variables. Concentrating on the significant effects that the process variables may have on quality-related variables in particular, the weight of $\boldsymbol{B}^k$ needs to be raised and should be larger than the weights of both $\boldsymbol{A}^k$ and $\boldsymbol{C}^k$. For the convenience of calculation, the weights of both $\boldsymbol{A}^k$ and $\boldsymbol{C}^k$ are set to 1. The weight of $\boldsymbol{B}^k$ is denoted as w. The following weighed information incremental matrix can be defined as a reasonable interpretation for characterizing the dynamics in the whole batch run:

$$
\begin{aligned}
\boldsymbol{WD}_k &= \left(\begin{array}{c|c}
\boldsymbol{A}^k & 0 \\
\hdashline
0 & 0
\end{array}\right) + w\left(\begin{array}{c|c}
0 & \boldsymbol{B}^k \\
\hdashline
\left(\boldsymbol{B}^k\right)^T & 0
\end{array}\right) + \left(\begin{array}{c|c}
0 & 0 \\
\hdashline
0 & \boldsymbol{C}^k
\end{array}\right) \\
&= \left(\begin{array}{c|c}
\boldsymbol{A}^k & w\boldsymbol{B}^k \\
\hdashline
w\left(\boldsymbol{B}^k\right)^T & \boldsymbol{C}^k
\end{array}\right)
\end{aligned}
\tag{4}
$$

where $\boldsymbol{WD}_k$ is defined as a weighed information incremental matrix. On the basis of this, the information increment index is defined directly derived from $\boldsymbol{WD}_k$:

$$\delta_k = \frac{\sum_{i=1}^{J}\sum_{j=1}^{J}|\boldsymbol{WD}_k(i,j)|}{J^2} \tag{5}$$

where δ_k is a representative of the average changing rate of covariance and variance along the time direction.

Phase Partition Method

As the correlation between the process variables and the quality-related variables in the same phase may perform similarly, sequence $\delta_k(k = 1, \ldots, K-1)$ may keep fluctuating within a certain small range. However, it may perform significantly differently when it steps into the next adjacent phase due to a stronger nonlinearity and severe dynamics that resulted from the mechanism in the multiphase batch processes. In other words, δ_k performs a jump when it begins moving into another sub-phase. It becomes stable and fluctuates within another range after entering a new phase. A larger δ_k will appear in the phase switch time than those in steady phases. Therefore, we can divide the process into sub-phases according to information increment index δ_k. It should be noted that a large valued δ_k is not always a switch point, and it might be in a certain sub-phase with strong dynamics. A critical feature of a fake switch point is that its neighbouring δ_k also possess large values.

In this article, we use a sliding time window derived from the moving average to extract the local maximums of $\delta_k(k = 1, \ldots, K-1)$, which are considered to be switch points.[37] The phase partition procedure consists of the following steps, which are also illustrated in Figure 1:

(1) Determine the length of the time window s, the step size of the time window L, which satisfies $(L < s)$, and the total step that is denoted as *count*. Initialize the current step $c = 0$;
(2) Calculate the mean $\mu(c)$ and the standard deviation $\sigma(c)$ of δ_k in the current time window, i.e., the time interval $[1 + L\cdot c\cdot s + L\cdot c]$. For every $\delta_k[k = 1 + L\cdot c, \ldots, s + L\cdot c]$, only the δ_k that satisfy $\delta_k \geq \mu(c) + \theta\cdot\sigma(c)$ are selected as the switch points. θ is called a limiting factor. θ should not be too large to ensure a successful selection, which means that at least one δ_k can be determined. On the other hand, it should not be so small that too many δ_k are identified as switch points. Finally, the time window moves forward by updating the step $c = c + 1$;
(3) If $c = count$, then all of the switch points are obtained. If not, return to step (2).

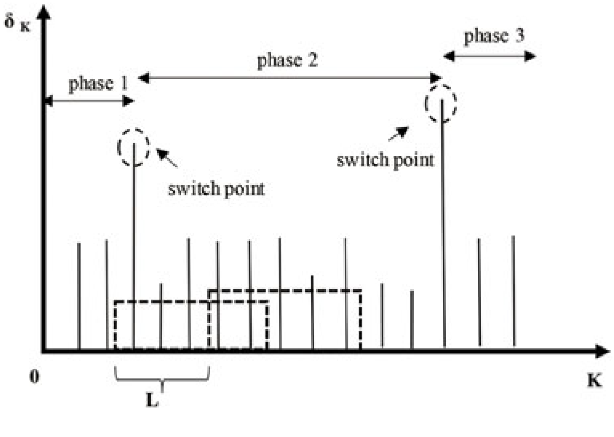

Figure 1. Phase partition process based on information increment index δ_k.

After the above calculations, an array of switch points denoted as $\delta_t(t = 1, \ldots, T)$ are obtained by a sliding time window, according to which the process is partitioned into D different sub-phases. The determination of weight w is important. A small w may not capable of emphasizing the effect the process variables have on the quality-related variables, whereas a large w may also weaken the self-interactions in the process variables and the quality-related variables. An appropriate w can guarantee an adequate selection of switch points and thus provides a reasonable phase partition result. The selection of δ_k is discussed later.

Sub-Phases Modelling

To improve process understanding and modelling performance, the MPLS models for each sub-phase are built for a batch process. Assuming that M extends time slices, denoted as $\boldsymbol{Z}_m(I\times J) = (X_m|Y_m)$ $(m = 1, \ldots, M, M \leq K-1)$, they have been partitioned into a certain sub-phase $d(1 \leq d \leq D)$ according to the information increment index. The M extend time slice are first rearranged into matrix $\boldsymbol{Z}_d(MI\times J)$ along the variable direction. In this way, the second step of AT unfolding is only implemented on the data set of each sub-phase. Then, $\boldsymbol{Z}_d(MI\times J)$ is further split into data set $\{\boldsymbol{X}_d(MI\times J_x), \boldsymbol{Y}_d(MI\times J_y)\}$ to build the following MPLS model:

$$\boldsymbol{X}_d = \boldsymbol{T}_d\boldsymbol{P}_d^T + \boldsymbol{E}_d \tag{6}$$

$$\boldsymbol{Y}_d = \boldsymbol{U}_d\boldsymbol{Q}_d^T + \boldsymbol{F}_d \tag{7}$$

This can be written in a regression form:

$$\hat{\boldsymbol{Y}}_d(MI\times J_y) = \boldsymbol{X}_d(MI\times J_x)\boldsymbol{\Theta}_d \tag{8}$$

where $\boldsymbol{T}_d$ and $\boldsymbol{U}_d$ are score matrices for $\boldsymbol{X}_d$ and $\boldsymbol{Y}_d$, respectively; $\boldsymbol{P}_d$ and $\boldsymbol{Q}_d$ are the corresponding loading matrices; $\boldsymbol{E}_d$ and $\boldsymbol{F}_d$ are the residual matrices; and $\boldsymbol{\Theta}_d(J_x\times J_y)$ is the regression coefficient matrix connecting $\boldsymbol{X}_d$ and $\boldsymbol{Y}_d$.

QUALITY PREDICTION STRATEGY

The quality prediction framework is illustrated in Figure 2. In offline modelling, the three-dimensional historical dataset is unfolded batch-wise first to form a two-dimensional dataset. After centering and normalizing batch-wise, the two-dimensional dataset is split into time slices, which are denoted as $\boldsymbol{Z}_k = (X_k|Y_k)$. Covariance matrix $\boldsymbol{R}_k$ and weighted information incremental matrix $\boldsymbol{WD}_k$ are calculated for each $\boldsymbol{Z}_k$ according to Equations (2)–(4). Then, information increment index δ_k is calculated according to Equation (5) as well. A sliding time window is introduced to extract the switch points of sequence $\delta_k(k = 1, \ldots, K-1)$. Finally, a MPLS model is established for each sub-phase. Relevant parameters, such as $\boldsymbol{\Theta}_d$, the mean and the standard deviation for each time slice, are stored as well.

When a new sample is acquired at the *kth* sample time, it is then assigned into a corresponding phase according to the current sample time k, which is supposed to be the *dth* sub-phase. Suppose the new sample is mean centered and normalized to $\boldsymbol{x}_{new}(J_x \times 1)$, the estimate of the current quality is calculated as follows:

$$\hat{\boldsymbol{y}}_{new} = \boldsymbol{\Theta}_d^T \boldsymbol{x}_{new} \tag{9}$$

where $\hat{y}_{new} = y_{new,1}, \ldots, y_{new,J_y}$ is a J_y dimensional quality-related variable vector.

The prediction ability of models is quantified by the root mean square error (RMSE). In our article, batch-level RMSE and sample time level RMSE are adopted to evaluate the performance of the methods, respectively. The batch-level RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{K}\sum_{k=1}^{K} ||y_k - y_{kp}||^2} \tag{10}$$

where K is the total number of sample time; $y_k$is the measured value of a quality-related variable at sample time $k_;$ and y_{kp} is the corresponding prediction value. Batch-level RMSE is used to evaluate the model's prediction ability for a single batch run. Sample time level RMSE is also defined to

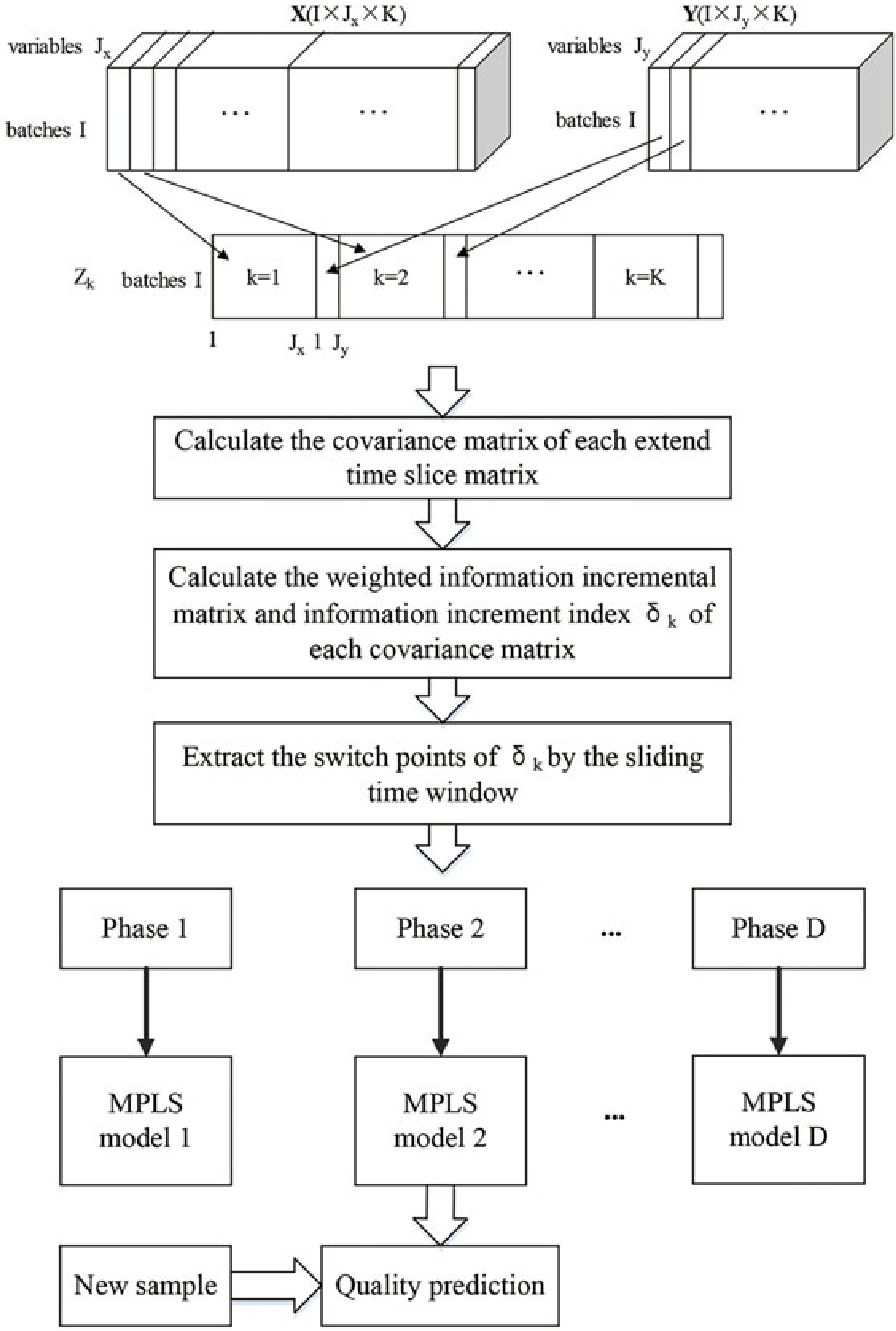
X(I×Jx×K)
variables Jx
batches I
Y(I×Jy×K)
variables Jy
batches I
Zk batches I
k=1
k=2
k=K
1
Jx 1 Jy
Calculate the covariance matrix of each extend time slice matrix
Calculate the weighted information incremental matrix and information increment index δ k of each covariance matrix
Extract the switch points of δ k by the sliding time window
Phase 1
Phase 2
Phase D
MPLS model 1
MPLS model 2
MPLS model D
New sample
Quality prediction

Figure 2. Global quality prediction strategy.

evaluate the prediction ability at each sample time, which is defined as follows:

$$RMSE(k) = \sqrt{\frac{1}{I_{test}}\sum_{j=1}^{I_{test}}||y_j^k - y_j^{kp}||^2} \tag{11}$$

where I_{test} is the number of test batches; y_j^k is the measured value at sample time k of test batch j; and y_j^{kp} is the corresponding prediction value of y_j^k.

CASE STUDY

Fed-Batch Penicillin Fermentation Simulation

Process description

Penicillin is a very common antiseptic that has been used to treat infections since 1942. The fed-batch penicillin fermentation process is a typical batch process with strong nonlinearity, dynamics, and multiphase characteristics. PensimV2.0 is an influential simulation platform of the penicillin fermentation process developed by the Illinois Institute of Technology. It has been widely used in the field of batch process monitoring and fault diagnosis since 2002. Relevant research has shown its practicality and effectiveness in simulating the mechanism of the penicillin fermentation process.[38,39]

Generally, the fed-batch penicillin fermentation process consists of two actual operating processes. One is the rapid consumption of the substrates period, during which the cells grow fast and there is no penicillin production. This operation phase lasts ~45 h. The other is the penicillin synthesis period. During this period, materials are fed into the fermentation from the substrate tank, which is consistent with the actual production process. This operation phase lasts ~355 h. Generally, the whole process lasts ~400 h. A total of 48 batches are generated by the PensimV2.0 under normal operating conditions. Among them, 40 batches are selected for modelling and 8 batches are used for testing. The duration of each batch is 400 h with a sample interval of 1 h. As shown in Table 1, 11 process variables and 2 quality-related variables are selected, respectively. Penicillin concentration is the quality variable in the penicillin fermentation process. Biomass concentration is an important variable that indicates the penicillin concentration. Both biomass concentration and penicillin concentration are available at every sample time in

Table 1. Process variables and quality-related variables in PensimV2.0

No.	Variable	Unit
x1	aeration rate	L/h
x2	agitator power	W
x3	substrate feeding rate	L/h
x4	substrate feeding temperature	K
x5	substrate (glucose) concentration	g/L
x6	DO saturation	%
x7	culture volume	L
x8	CO_2	mmol/L
x9	pH	-
x10	temperature	K
x11	generated heat	kJ/h
y1	biomass concentration	g/L
y2	penicillin concentration	g/L

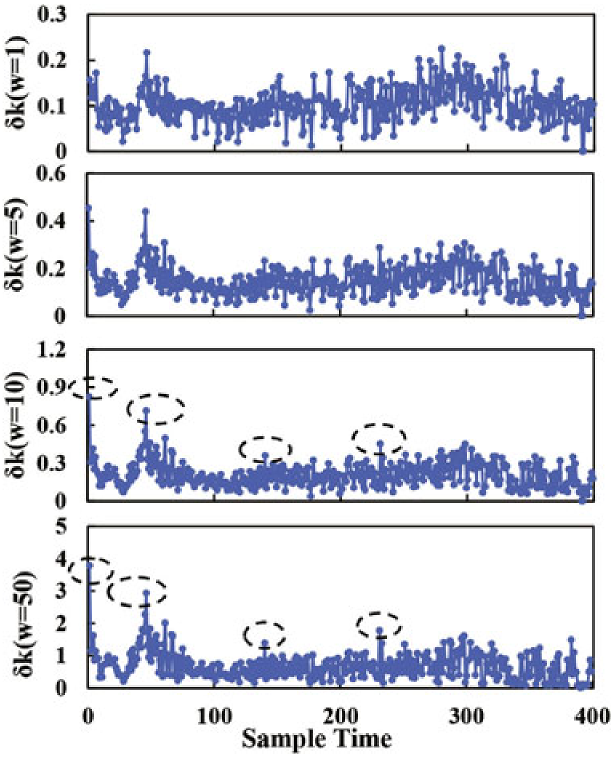

Figure 3. Results of the information increment index values under different weights using the PensimV2.0 data.

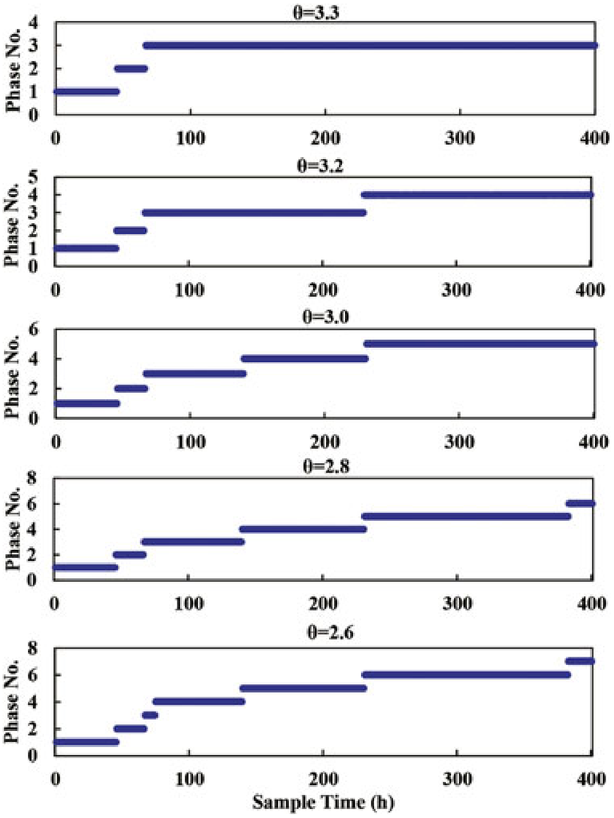

Figure 4. Phase partition results under different θ using the PensimV2.0 data.

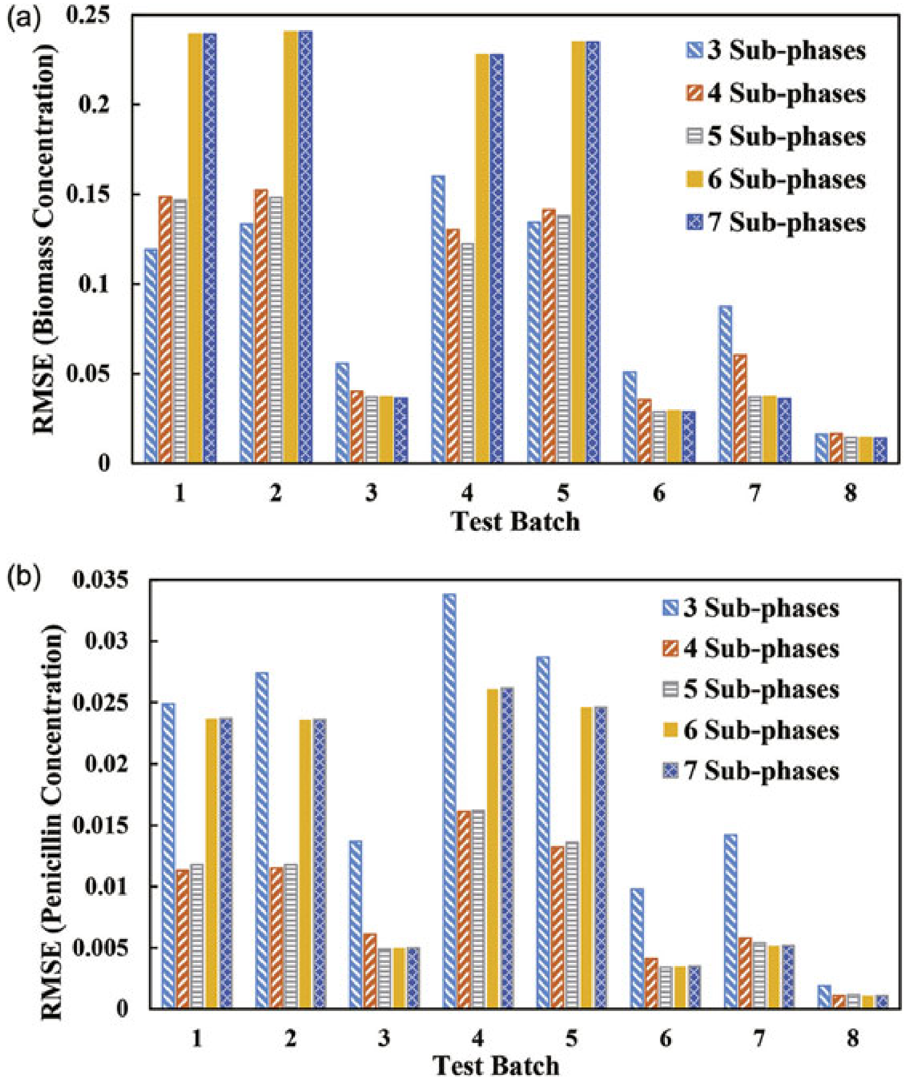

Figure 5. RMSEs of 8 test batches under different numbers of sub-phases: (a) RMSEs of biomass concentration; and (b) RMSEs of penicillin concentration.

PensimV2.0. The dataset for modelling can be denoted as $\{\tilde{X}(40 \times 11 \times 400), \tilde{Y}(40 \times 2 \times 400)\}$.

Phase partition results

The fed-batch penicillin fermentation process is divided into several sub-phases according to information increment index δ_k. The trajectory of δ_k offers a dynamic reflection of the variance and covariance among variables. Trajectories of δ_k under different weights are illustrated in Figure 3, in which w is set to different values. It can be seen that when the w value becomes larger, the switch points are captured more clearly in sequence δ_k. The potential switch points are marked out with dotted eclipses when $w = 10$ and $w = 50$. If w is too large, it will not be able to display the influences among process variables themselves. On the contrary, if w is too small, the influences of the interactions between the process variables and quality-related variables are weakened. Therefore, w is set to 10. It is also worth noting that if the width of the time window s is too small, the time window would be so sensitive that redundant points can be selected. On the other hand, if s is too large, for one extreme case that $s = K - 1$, some turn points that reflect the local abrupt may be omitted. Here, s is decided by experience and satisfies $K - 1 \approx (6 \sim 4)s$. To our knowledge, the step of the time window $L \approx (0.25 \sim 0.75)s$ for an effective selection. In this article, s is set to 70 and L is set to 33.

The limiting factor θ influences the selection of the switch points when using a sliding time window on sequence $\delta_k(k = 1, \ldots, K - 1)$. Phase partition results under different θ are shown in Figure 4. It can be seen that the smaller the θ is, the more switch points are selected. When θ is large enough, no switch points are selected. When $\theta = 3.3$, sample times 1, 46, and 67 are first selected as switch points and this divides the process into 3 sub-phases. It should be noted that sample time 46 is significant here because it is very close to the switching moment of the two actual operating processes, i.e., the rapid consumption of the substrates period and the penicillin synthesis period. As θ reduces in size, more switch points are identified gradually and more sub-phases emerge. When $\theta = 2.6$, sample times 1, 46, 67, 75, 149, 231, and 383 are identified as switch points and divide the process into 7 sub-phases. Here, θ is confined to 2–3 in term of the 3σ criterion from probability theory, which interprets that the probability of a sample dropping out of the interval $(\mu - 3\sigma, \mu + 3\sigma)$ is very small when the samples set is large enough.

The MPLS models for all sub-phases are built for quality prediction after phase partition. The batch-level RMSEs of 8 test batches under a different number of sub-phases are shown in Figure 5. It can be seen from Figure 5a that for most test batches, e.g., test batch 4, when the number of sub-phases is < 5, a higher prediction accuracy of biomass concentration is gradually achieved when the number of sub-phases becomes larger. However, the prediction accuracy decreases gradually when the number of sub-phases is > 5. The trend of the prediction accuracy of penicillin concentration is similar to that of the biomass concentration, as shown in Figure 5b. By considering the prediction results of both the biomass concentration and the penicillin concentration, we can determine that the cases with 4 sub-phases and 5 sub-phases have a better prediction performance. Computing the mean RMSEs of all of the test batches for cases with a different number of sub-phases shows that the case with the 5 sub-phases has the smallest mean RMSE for both biomass concentration and penicillin concentration and performs the best. Therefore, the optimal number of phases is selected as 5 when $\theta = 3$.

Online quality prediction

The prediction performance of the proposed IIMPLS method is compared with a single model method named SMPLS (single MPLS), a phase-based model method named MPMPLS (multiple phase-based MPLS), and an adaptive model method named D-LWPCR (double locally weighted principal component regression). SMPLS treats the whole batch as a single operation phase without the phase partition and performs AT unfolding on the three-dimensional historical batches for the MPLS modelling. MPMPLS was proposed by Luo et al.[23] It uses SCFCM, a type of clustering method, as a sequential phase partition method and uses variable-wise unfolding MPLS in each sub-operation phase for prediction. D-LWPCR was proposed by Yuan et al.[12] In D-LWPCR, the samples that were used to build a local model are first selected in a supervised latent structure, considering both the sample-importance and the variable-importance.[12] Then, the importance variable is predicted with a PCR model built by the selected samples. Table 2 shows the phase partition results of MPMPLS and IIMPLS, where the left time boundaries of the sub-phases are listed. We can see that the two methods identify the 45th and 46th sample time as phase switch points, respectively. The identified switch points correspond to the switch time between the cell growth period and the penicillin synthesis period in the actual production process. The results verify that the models possess physical meaning.

Table 3 lists the performance of each method on 8 test batches, including the RMSE mean and RMSE standard deviation of the 8 test batches over all of the sample times. It can be seen that IIMPLS is obviously better than SMPLS and D-LWPCR because it gives the smallest RMSE mean in terms of both biomass concentration and penicillin concentration. IIMPLS also performs slightly better than MPMPLS. In addition, we can see from the RMSE standard deviation that IIMPLS has the smallest predicting fluctuations for different test batches, which means that IIMPLS has the most robust prediction performance for various test batches.

Table 2. Left time boundaries of sub-phases in PensimV2.0

Phase No.	1	2	3	4	5	6	7	8	9
MPMPLS	1	27	34	39	**45**	94	127	196	223
IIMPLS	1	**46**	67	140	231	-	-	-	-

Table 3. RMSE statistical properties of each method in PensimV2.0

Case		SMPLS	MPMPLS	IIMPLS	D-LWPCR
Biomass concn.	RMSE mean	0.1199	0.0744	0.0736	0.1349
	RMSE SD	0.0576	0.0510	0.0420	0.1080
Penicillin concn.	RMSE mean	0.0155	0.0092	0.0085	0.0211
	RMSE SD	0.0072	0.0062	0.0054	0.0107

The prediction results of each method for a test batch are shown in Figure 6. From Figure 6a we can see that IIMPLS has the highest prediction accuracy of biomass concentration and that it correlates with the simulation value. MPMPLS also performs well. SMPLS provides an accurate prediction, except between 50–100 h, during which an obvious deviation occured. There is also a small deviation between 230–400 h. D-LWPCR performs well during most the of the sample times, except between 160–230 h and in some short time intervals, such as 260–270 h. In terms of the prediction of penicillin concentration, as shown in Figure 6b, both IIMPLS and MPMPLS provide accurate predictions. SMPLS also performs

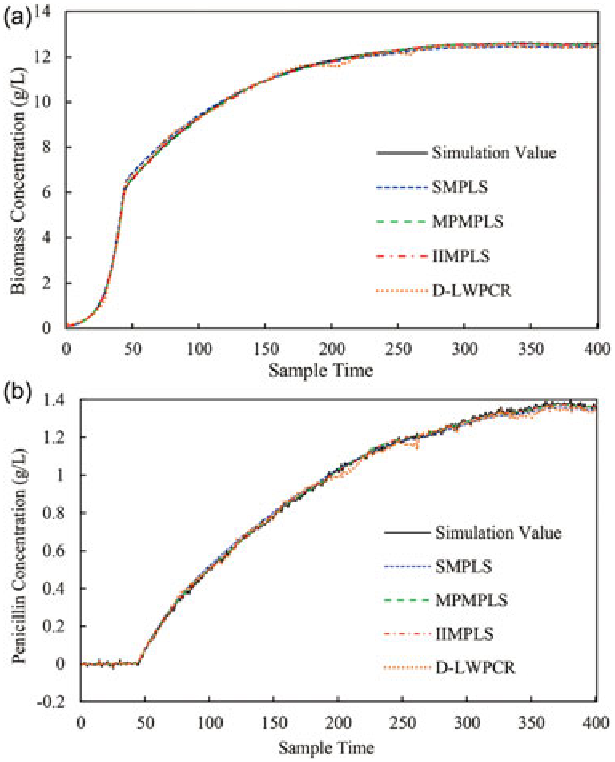

Figure 6. Prediction results of each method for a test batch: (a) prediction results of biomass concentration; and (b) prediction results of penicillin concentration.

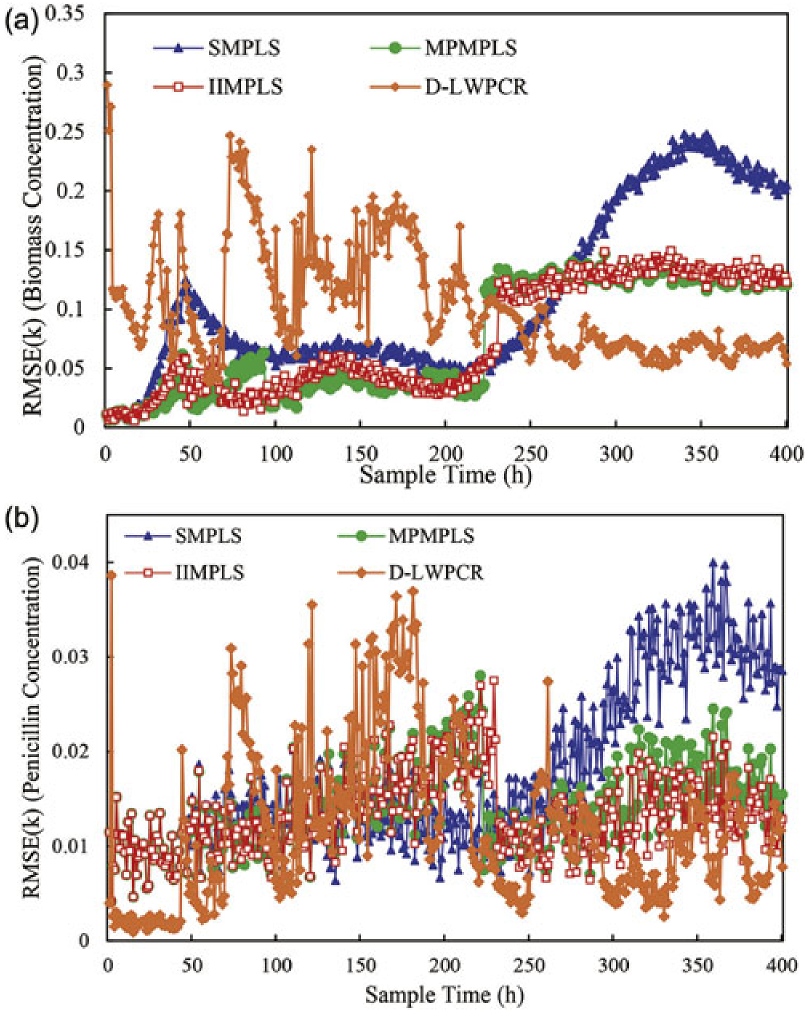

Figure 7. RMSE(k)s of 8 test batches for each method: (a) RMSE(k)s of biomass concentration; and (b) RMSE(k)s of penicillin concentration.

well during most of the sample times. However, the prediction accuracy decreases at the end of the fermentation process (350–400 h). D-LWPCR also shows performs well during most the of the sample times, except between 200–230 h and 250–270 h. Since the D-LWPCR selects the most relevant samples from all of the available historical data to build the local PCR model, the samples with similar characteristics but that are far from the current sample time are very likely to be chosen. The number of samples for local modelling is also fixed, which may have an influence on the modelling precision. In addition, the PensimV2.0 data is considered as a low-noise data set since it is generated by the simulation platform. Adaptive model methods, such as D-LWPCR, are more suitable for handling data with significant noises. In summary, the proposed method shows the best prediction efficiency for both biomass concentration and penicillin concentration.

Further, the sample time level RMSE(k)s of 8 test batches are shown in Figure 7. We can see from Figure 7a that IIMPLS predicts the biomass concentration slightly more accurately than the MPMPLS during most of the sample times. Obviously, SMPLS has relatively plain prediction performance. It shows a large prediction error, especially around 50 h and the end of the batch run. For 45–100 h, which corresponds to the end of the cell growth period and the beginning of the penicillin synthesis period, SMPLS cannot capture the local dynamics at the phase transition period, since SMPLS treats the whole batch run as a single phase. D-LWPCR has an unsatisfactory prediction accuracy with the unstable RMSE(k) fluctuation before 250 h. However, it shows excellent prediction accuracy after 250 h, compared with the other three methods.

Table 4. Process variables and quality-related variables in the *E. coli* fermentation process

No.	Variable	Unit
x1	pH	-
x2	DO saturation	%
x3	pressure	Pa
x4	temperature	°C
x5	aeration power	rpm
x6	added glucose	mL
x7	added nitrogen	mL
x8	aeration rate	Lm^{-1}
y1	*Escherichia coli* concentration, OD	-
y2	protein expression, IL-2	kg

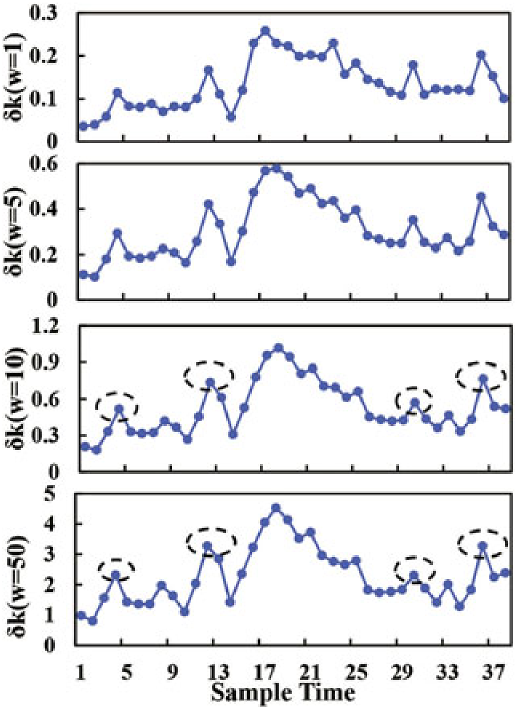

Figure 8. Results of the information increment index values under different weights.

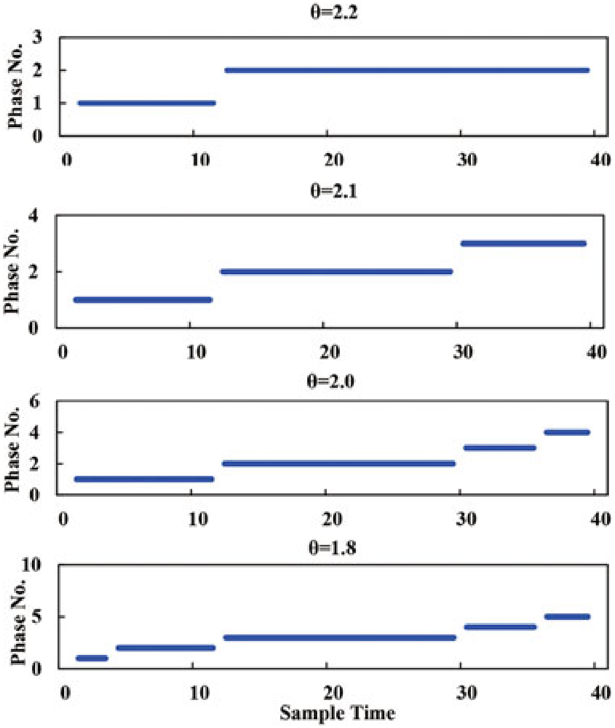

Figure 9. Phase partition results under different θ.

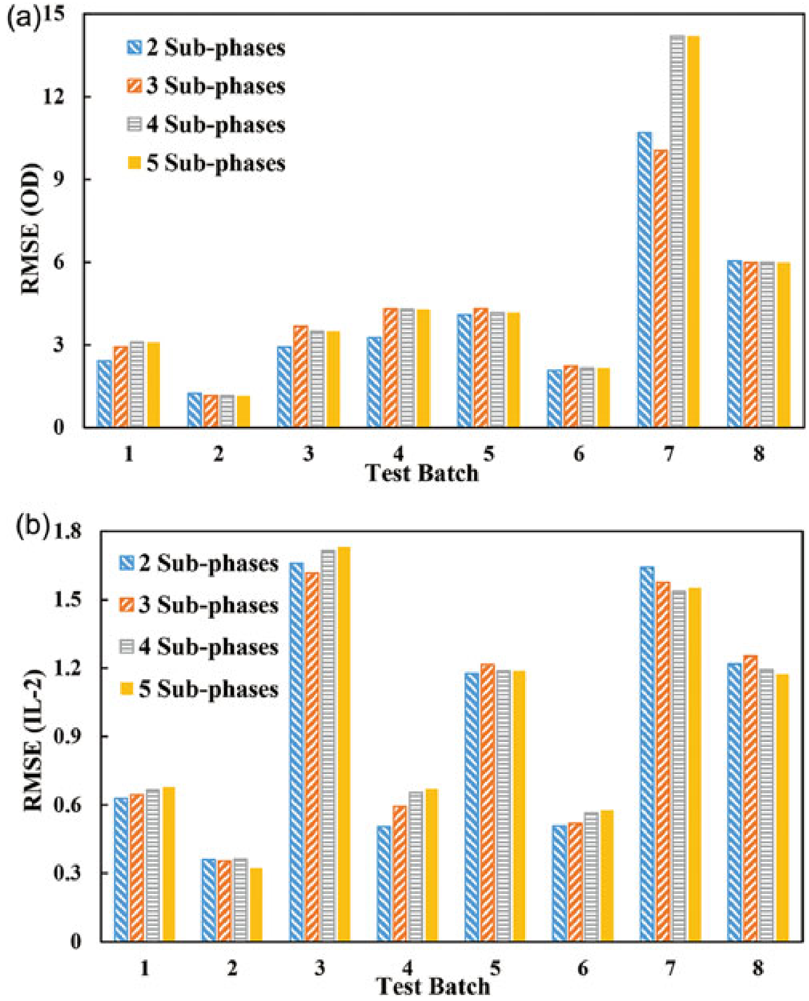

Figure 10. RMSEs of 8 test batches under different numbers of sub-phases: (a) RMSEs of OD; and (b) RMSEs of IL-2.

Table 5. Left time boundaries of sub-phases

Phase No.	1	2	3	4
MPMPLS	1	**11**	16	37
IIMPLS	1	**12**	-	-

Table 6. RMSE statistical properties of each method in the *E. coli* fermentation process

Case		SMPLS	MPMPLS	IIMPLS	D-LWPCR
OD	RMSE mean	9.6441	4.7807	4.0947	9.9925
	RMSE SD	6.8370	4.3671	3.0341	0.6854
IL-2	RMSE mean	2.8383	0.9816	0.9622	0.9701
	RMSE SD	0.5867	0.5272	0.5194	0.5452

IIMPLS and MPMPLS obviously perform better at the phase transition period. For the prediction of the RMSE(k)s of the penicillin concentration shown in Figure 7b, IIMPLS has the highest prediction accuracy for most of the sample times. SMPLS has a relatively small RMSE(k)s between 150–230 h, but the RMSE(k)s increases fast and fluctuates significantly until the end of the batch run. D-LWPCR has the same fluctuating pattern as MPMPLS and IIMPLS. However, the max RMSE of D-LWPCR is larger than that of MPMPLS and IIMPLS. Besides, D-LWPCR has the smallest RMSE(k). The prediction results mentioned above can indicate that IIMPLS offers an effective way to model batch processes for quality prediction.

Industrial *E. coli* Fermentation Process

Process description

The *E. coli* fermentation process is a typical batch process in which *E. coli* is fermented to produce pharmaceutical proteins. In this article, the methods are verified by the actual production data of the *E. coli* fermentation process from a biological pharmaceutical factory in Beijing. Interleukin-2 (IL-2) is produced and the production cycle is ~6–7 h. In order to implement a full cell growth, the fermentation process is conducted in a small fermenter during the first 2 h and then moved to a large fermenter for the remaining hours. Compared with the PensimV2.0 data, the actual production data can verify the significance and efficiency of the method more accurately.

In this work, 10 variables are selected, including 8 process variables and 2 quality-related variables, as listed in Table 4. *E. coli* concentration is an important variable

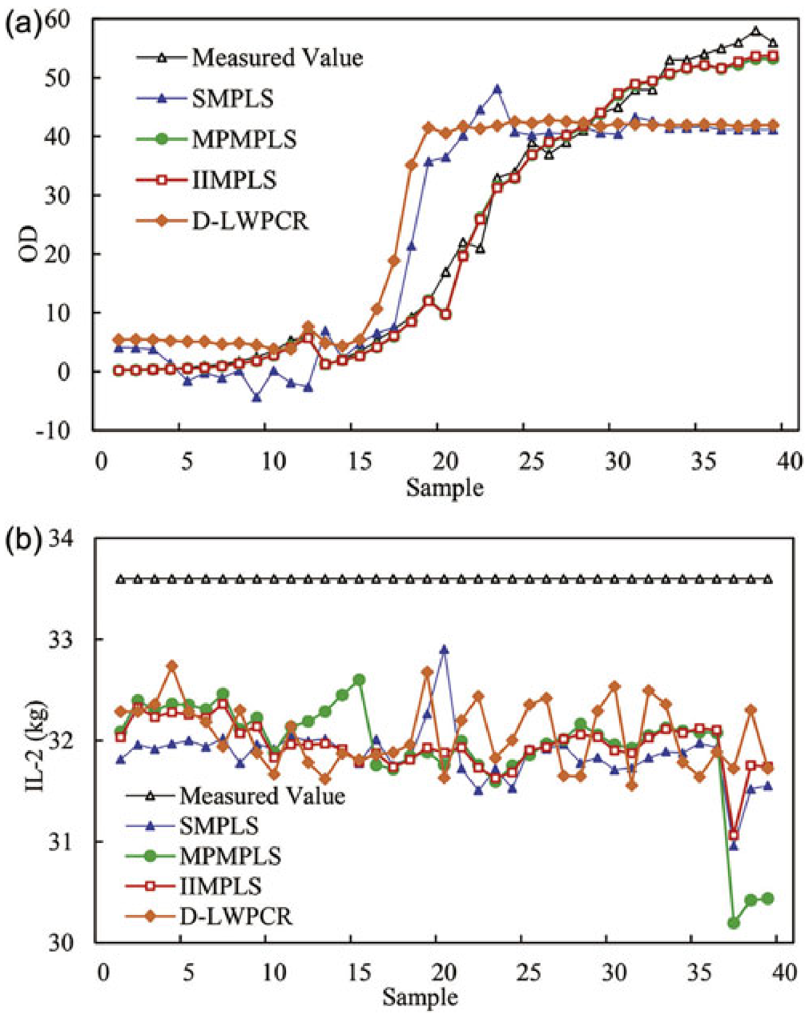

Figure 11. Prediction results of each method in the *E. coli* fermentation process: (a) prediction results of OD; and (b) prediction results of IL-2.

that indicates the final product IL-2. IL-2 is the quality variable. In the actual production, the *E. coli* concentration is indirectly measured by the optical density (OD). The fermentation duration is 6.5 h and the sample times are 10 min. A total of 28 normal batches are collected and processed to even length. Among them, 20 batches are randomly selected for modelling and the remaining 8 batches are used for testing. OD is usually measured offline and is available at every time interval. IL-2 is the final product quality only measured after the whole batch run, so we duplicate the final measured value for every sample time. Thus, the modelling dataset can be denoted as $\{\tilde{X}(20 \times 8 \times 39), \tilde{Y}(20 \times 2 \times 39)\}$.

Phase partition results

Similar to that in the PensimV2.0 simulation platform, the trajectories of δ_k under different weights are illustrated in Figure 8. The *E. coli* fermentation process is divided into several sub-phases according to the information increment index δ_k. It also can be seen that when the w value becomes larger, clearer switch points are captured in sequence δ_k. The potential switch points are marked out with dotted eclipses when $w = 10$ and $w = 50$. In this article, w is chosen to be 10, which is the same as that in the PensimV2.0 simulation platform. s is set to 8 and L is set to 5.

Phase partition results under different θ are shown in Figure 9. When $\theta = 2.2$, sample time 12 is first selected as a switch point and divides the process into 2 sub-phases. The sample time 12 is special because it is exactly the moment that a small fermenter is replaced by a large one. With the θ becoming smaller, more switch points are identified gradually and more sub-phases emerge. When $\theta = 2.1$, sample times 12 and 30 are identified as switch points and they divide the process into 3 sub-phases. When $\theta = 2.0$, sample time 36 is newly identified as a switch point and the process is divided into 4 sub-phases. Then, phase-based MPLS models are built for quality prediction after phase partition. The batch-level RMSEs of 8 test batches under different numbers of sub-phases are shown in Figure 10. After computing the mean RMSEs of all of the test batches under cases with different numbers of sub-phases, it was determined that the case with 2 sub-phases has the smallest mean RMSE for both OD and IL-2. Therefore, the optimal number of phases is selected as 2 when $\theta = 2.2$. In the meantime, we can see that the optimal number of sub-phases is not always large.

Online quality prediction

Table 5 shows the phase partition results of MPMPLS and IIMPLS, where left time boundaries of the sub phases are listed. We can see that MPMPLS identifies the 11th sample time as a phase switch point, which is close to the moment that a large fermenter is put into use for the production process. Furthermore, MPMPLS also selects sample times 16 and 37 as phase switch points. Table 6 lists the statistical performance of RMSE. It can be seen from the RMSE

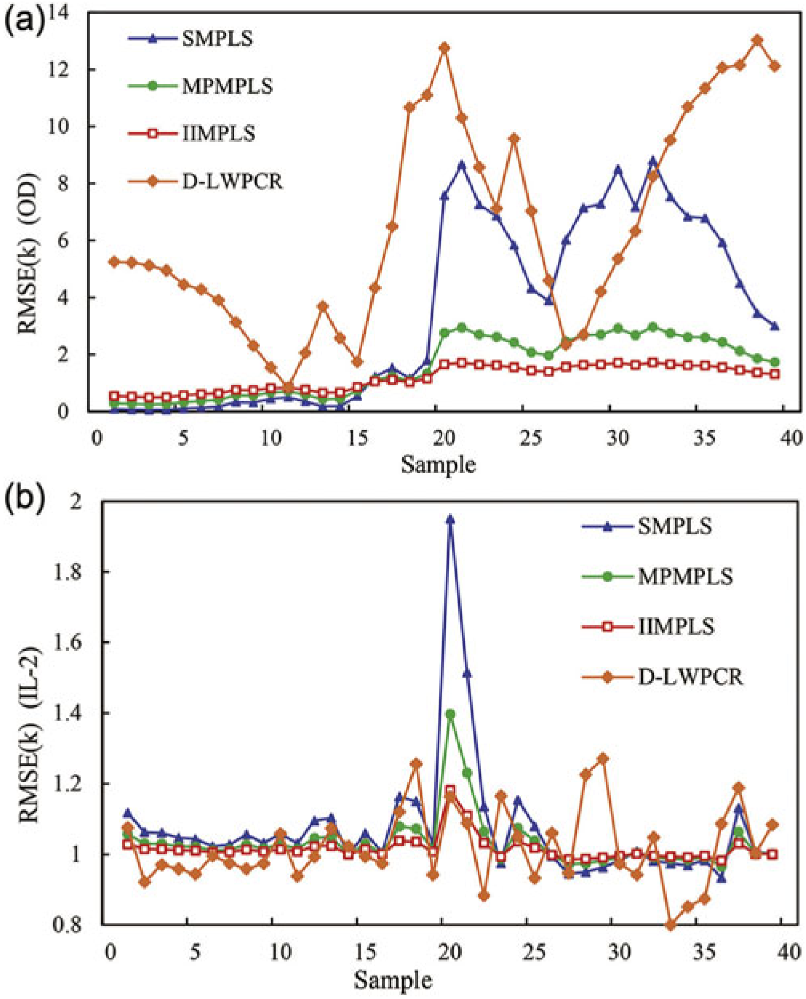

Figure 12. RMSE(k)s of 8 test batches for each method in the *E. coli* fermentation process: (a) RMSE(k)s of OD; and (b) RMSE(k)s of IL-2.

mean that the prediction accuracy of IIMPLS for OD is obviously better than SMPLS and D-LWPCR. IIMPLS also has a slightly better performance than MPMPLS. For the prediction of IL-2, IIMPLS, MPMPLS, and D-LWPCR obviously perform better than SMPLS. SMPLS has the worst prediction performance. We can also see from the RMSE standard deviation that IIMPLS has the most stable prediction performance for various test batches.

The prediction results of the four methods for a test batch are shown in Figure 11. From Figure 11a we can see that both MPMPLS and IIMPLS agree well with the measured value of OD, whereas SMPLS and D-LWPCR show poor prediction performance during most of the sample times. Figure 11b illustrates the prediction results for IL-2. The prediction of IL-2 is not significantly different than the prediction of OD, except at the end of the batch run. All of the methods, except D-LWPCR, do not perform well during sample time 36–37. IIMPLS performs best at the end of the batch run and has the highest prediction accuracy for the final value of IL-2. In conclusion, IIMPLS shows the best prediction efficiency for both the OD and the final protein expression IL-2.

The RMSE(k)s of 8 test batches are shown in Figure 12. From Figure 12a we can see that MPMPLS and IIMPLS have a better prediction accuracy than both SMPLS and D-LWPCR. D-LWPCR has a higher RMSE(k)s during most of the sample times, except sample times 11, 15, 28, and 29. It performs better than SMPLS during sample times 28–33. During the first 10 sample times, SMPLS performs slightly better than MPMPLS and IIMPLS, but later the RMSE(k)s of SMPLS increase sharply and far exceed the RMSE(k)s of MPMPLS and IIMPLS. RMSE(k)s of MPMPLS also show an increase from sample time 15, but the increase is not as obvious as that in SMPLS. IIMPLS performs best because the RMSE(k)s of IIMPLS increases slowly and keeps fluctuating in a small range under the value of 2 until the end of the batch runs. In terms of the prediction of IL-2 illustrated in Figure 12b, IIMPLS has the smallest RMSE(k)s for most of the sample times. It is worth noting that SMPLS encounters a severe increase at sample time 21. Since SMPLS treats the process as a single phase and gathers all of the data from all of the sample times together equally to build a model without distinguishing their time varying properties, it may be difficult for SMPLS to distinguish the local dynamics of the process. MPMPLS also shows an obvious increase at sample time 21, but it performs much better than SMPLS and D-LWPCR. By contrast, IIMPLS has the smallest RMSE(k) and has the slightest increase at sample time 21. The above prediction results indicate that IIMPLS outperforms the other three methods for quality prediction in an industrial *E. coli* fermentation process.

CONCLUSIONS

A sequential phase partition method based on the information increment is proposed, which utilizes the covariance matrix of the extend time slice matrix to capture the dynamic characteristics of batch processes along the time direction and divides the process into sub-phases. The proposed partition method has two main advantages. First, it is a sequential method and can overcome the limits of some phase partition algorithms that may divide the samples with a discontinuous time sequence but with similar characteristics into the same phase. Second, it takes into consideration the effects that the quality-related variables have on the phase partition results in terms of each sample time. Besides, the method is easy to implement and has a high computation efficiency. The feasibility and effectiveness of the proposed method are illustrated by a penicillin simulation platform and an industrial application of *E. coli* fermentation, respectively. The results can indicate that IIMPLS can correctly identify the switch moment of the actual operating processes as phase switch points, which enhances the interpretation of the process. It also has a higher prediction accuracy and is more robust.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China under grant 61640312, 61763037, and 61803005, the Natural Science Foundation of Beijing Municipality under grant 4172007, and the Beijing Municipal Commission of Education.

REFERENCES

[1] W. Q. Li, C. H. Zhao, *Can. J. Chem. Eng*. **2017**, *95*, 1817.

[2] X. C. Wang, P. Wang, X. J. Gao, Y. S. Qi, *Chemometr. Intell. Lab*. **2016**, *158*, 138.

[3] Y. Wang, Q. C. Jiang, B. B. Li, L. Z. Cui, *IEEE Access* **2018**, *6*, 13005.

[4] C. H. Zhao, F. L. Wang, F. R. Gao, *Can. J. Chem. Eng*. **2009**, *87*, 466.

[5] Z. Q. Ge, *Control Eng. Pract*. **2014**, *31*, 9.

[6] Y. C. He, L. Zhou, Z. Q. Ge, Z. H. Song, *Can. J. Chem. Eng*. **2018**, *96*, 1541.

[7] S. Y. He, Y. Q. Wang, C. Q. Liu, *Can. J. Chem. Eng*. **2018**, *96*, 444.

[8] B. Corbett, P. Mhaskar, *AIChE J*. **2016**, *62*, 1581.

[9] B. Corbett, P. Mhaskar, *Ind. Eng. Chem. Res*. **2017**, *56*, 6962.

[10] C. H. Zhao, F. L. Wang, Y. Yao, F. R. Gao, *Acta Automatica Sinica* **2010**, *36*, 366.

[11] Y. Q. Wang, Y. B. Si, B. Huang, Z. J. Lou, *Can. J. Chem. Eng*. **2018**, *96*, 2073.

[12] X. F. Yuan, B. Huang, Z. Q. Ge, Z. H. Song, *Chemometr. Intell. Lab*. **2016**, *153*, 116.

[13] J. X. Liu, T. Liu, J. Zhang, *Ind. Eng. Chem. Res*. **2016**, *55*, 9229.

[14] X. F. Ye, P. L. Wang, Z. Y. Yang, *IEEE Access* **2018**, *6*, 1249.

[15] Y. Liu, T. Chen, J. H. Chen, *Ind. Eng. Chem. Res*. **2015**, *54*, 5037.

[16] R. X. Guo, K. Guo, J. K. Dong, *IEEE T. Autom. Sci. Eng*. **2017**, *14*, 1582.

[17] Z. J. Lou, Y. Q. Wang, *Ind. Eng. Chem. Res*. **2017**, *56*, 13800.

[18] J. Yu, S. J. Qin, *Ind. Eng. Chem. Res*. **2009**, *48*, 8585.

[19] N. Y. Lu, F. R. Gao, F. L. Wang, *AIChE J*. **2004**, *50*, 255.

[20] N. Y. Lu, F. R. Gao, *Ind. Eng. Chem. Res*. **2005**, *44*, 3547.

[21] Y. Hu, X. Gao, Y. Li, Y. Qi, P. Wang, *Journal of Chemical Industry and Engineering (China)* **2016**, *67*, 1989.

[22] Y. S. Qi, P. Wang, X. J. Gao, *IET Control Theory A*. **2012**, *29*, 754.

[23] L. J. Luo, S. Y. Bao, J. F. Mao, D. Tang, Z. L. Gao, *Ind. Eng. Chem. Res*. **2016**, *55*, 4045.

[24] J. Camacho, J. Pico, A. Ferrer, *J. Chemometr*. **2008**, *22*, 632.

[25] Z. Q. Ge, L. P. Zhao, Y. Yao, Z. H. Song, F. R. Gao, *J. Process Contr*. **2012**, *22*, 599.

[26] C. H. Zhao, *IEEE T. Autom. Sci. Eng*. **2014**, *11*, 983.

[27] Y. Qin, C. H. Zhao, X. Z. Wang, F. R. Gao, *Chem. Eng. Sci.* **2017**, *166*, 130.

[28] S. Wold, H. Martens, H. Wold, *Lect. Notes Math.* **1983**, *973*, 286.

[29] A. Kassidas, J. F. MacGregor, P. A. Taylor, *AIChE J.* **1998**, *44*, 864.

[30] P. Nomikos, J. F. Macgregor, *Technometrics* **1995**, *37*, 41.

[31] H. R. Choi, E. J. Kim, T. Y. Kim, *I. Symp. Consum. Electr.* **2014**, *5057*, 1.

[32] P. Nomikos, J. F. MacGregor, *Chemometr. Intell. Lab.* **1995**, *30*, 97.

[33] P. Nomikos, J. F. Macgregor, *AIChE J.* **1994**, *40*, 1361.

[34] S. Wold, N. Kettaneh, H. Friden, A. Holmberg, *Chemometr. Intell. Lab.* **1998**, *44*, 331.

[35] D. Aguado, A. Ferrer, J. Ferrer, A. Seco, *Chemometr. Intell. Lab.* **2007**, *85*, 82.

[36] C. L. Wen, Y. C. Hu, *Acta Automatica Sinica* **2012**, *38*, 832.

[37] J. B. Xiong, Q. R. Wang, J. F. Wan, B. Y. Ye, W. C. Xu, J. Q. Liu, *Sensor Lett.* **2013**, *11*, 877.

[38] G. Birol, C. Undey, A. Cinar, *Comput. Chem. Eng.* **2002**, *26*, 1553.

[39] C. Undey, E. Tatara, A. Cinar, *J. Biotechnol.* **2004**, *108*, 61.

Manuscript received July 19, 2018; revised manuscript received November 12, 2018; accepted for publication December 17, 2018.