Monitoring Uneven Multistage/Multiphase Batch Processes using Trajectory-Based Fuzzy Phase Partition and Hybrid MPCA Models

Lijia Luo *

Institute of Process Equipment and Control Engineering, Zhejiang University of Technology, Hangzhou, China

Batch processes often have the traits of multiple operation stages/phases and uneven batch durations. These two traits bring difficulties to batch process modelling and monitoring. In this paper, a trajectory-based fuzzy phase partition (TBFPP) method and hybrid multiway PCA (MPCA) models are developed for monitoring multistage/multiphase batch processes with uneven durations. The TBFPP method divides each batch into several fuzzy operation phases by clustering trajectory data of phase-sensitive process variables using the sequence-constraint fuzzy c-means (SCFCM) clustering algorithm. This TBFPP method not only solves the uneven duration problem of batches, but also can identify transition regions between neighbouring operation phases. Fuzzy operation phases are further divided into "steady" and "transition" operation phases according to the membership degrees of samples. Hybrid modelling methods, consisting of phase-based (global) modelling and just-in-time (local) modelling, are used to cope with different process characteristics of the "steady" and "transition" operation phases. Offline phase-based MPCA models are built for "steady" operation phases to describe the steady process characteristics. Online just-in-time MPCA models are built for "transition" operation phases to handle the time-varying process characteristics. Based on the hybrid MPCA models, an online process monitoring method is proposed. The efficacy of the proposed methods is demonstrated through a simulation study of a fed-batch fermentation process.

Keywords: batch process, phase partition, hybrid MPCA, process monitoring, fault detection

INTRODUCTION

Batch process is playing an increasingly important role in biochemistry, pharmaceutical, semiconductor, and food industries.[1,2] Most batch processes have two inherent characteristics: uneven durations and multiple operation phases,[3] termed as uneven batch processes and multiphase batch processes, respectively. The duration of a real batch process is often not fixed due to fluctuations in initial conditions, process disturbances, and other factors, resulting in batches with uneven durations.[3–5] In many batch processes, a normal batch either goes through multiple processing units or runs in a single processing unit under multiple operation modes.[6] Moreover, the reactions in a batch may also dynamically vary along the operating time. Because of these factors, a batch process has multiple operation stages/phases. Those operation phases caused by changes of processing units or operation modes are explicit and easy to detect. However, those operation phases caused by dynamic changes of reactions or other time-varying factors may be latent and difficult to identify. Each operation phase has specific process behaviours that differ from those in other operation phases. Process behaviours in the transition region between two operation phases are very complicated. In addition, the multiphase characteristics aggravate the influence of the uneven duration problem. The same operation phase in different batches may have varying durations, regardless of whether or not the batches have equal total durations. The uneven duration problem and multiphase characteristics bring difficulties for modelling, control, and monitoring of batch processes.

In recent years, the uneven duration problem and multistage/multiphase characteristics in batch processes are gaining rapidly increasing attention. Many process modelling, process monitoring, and quality control methods have been developed for uneven batch processes or multiphase batch processes.[3,7,8] A popular method to cope with the multiphase characteristics is dividing the entire batch process into different operation phases and modelling each operation phase separately. A relatively precise process model thus can be built for each operation phase. Phase partition is a critical step before modelling. Three main types of phase partition methods, i.e., knowledge-based,[6] analysis-based,[9,10] and data-driven methods,[3] have been developed. Compared with the other two types of methods, data-driven methods are much easier to implement, as they only need process data and rarely rely on process knowledge. The representative data-driven phase partition methods include the multiphase (MP) algorithm,[11,12] the sub-PCA method,[13] the stepwise sequential phase partition (SSPP) algorithm,[14] the Gaussian mixture model (GMM)-based method,[15] and so on. Some data-driven phase partition methods were also proposed to handle the uneven duration problem in multiphase batch processes.[16–19] For example, Lu et al.[13] developed a revised sub-PCA phase partition method for multiphase batch processes with uneven-length operation phases. This method carries out phase partition by clustering the weighted loading matrices of time-slice PCA models via the K-means (KM) algorithm. Because the KM clustering algorithm cannot cope with the time sequence of samples, samples belonging to different time regions may be classified into the same phase. Such phase partition results are unreasonable, and thus post processing should be implemented to get time-consecutive operation phases.[16] Moreover, the revised sub-PCA method ignores the transition region between adjacent operation phases. Luo et al.[17]

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

Can. J. Chem. Eng. 97:178–187, 2019

DOI 10.1002/cjce.23220
Published online 6 April 2018 in Wiley Online Library
(wileyonlinelibrary.com).

proposed a phase partition method based on the warped K-means (WKM) clustering algorithm. This WKM-based method has better phase partition capability than the revised sub-PCA method.[17] However, it also ignores the between-phase transitions. Li et al.[18] extended the SSPP algorithm to the uneven batch case. The proposed method detects phase partition points sequentially by checking changes in variable correlations, while it does not distinguish phases and between-phase transitions.[18] To cope with the transitions between operation phases, Luo et al.[19] proposed a fuzzy phase partition method on the basis of the sequence-constraint fuzzy c-means (SCFCM) algorithm. The SCFCM clustering algorithm has two advantages. First, it can maintain the original time sequence of the samples by adding a hard sequentiality constraint in the clustering procedure. Second, the SCFCM algorithm generates fuzzy clusters. These two advantages enable the SCFCM-based phase partition method to directly obtain time-consecutive operation phases and to identify the between-phase transitions. However, the uneven duration problem was not considered by Luo et al.[19] It is thus necessary to develop effective phase partition methods that can simultaneously handle the uneven duration problem and transitions between operation phases.

On the other hand, operation phases in a multiphase batch process often have different process characteristics. Process characteristics in some operation phases may be steady and nearly linear, while those in other operation phases may be time-varying and strongly nonlinear. Process characteristics in the transition region between two adjacent operation phases are more complicated. Therefore, to build high performance process models, operation phases should be treated differently according to their own process characteristics. However, most existing methods model all operation phases using the same modelling technique,[12,14,16–18] ignoring the differences in process characteristics between operation phases. In fact, each modelling technique has a specific scope of application. It may be suitable for some operation phases, while not good for others. This drawback may result in low modelling accuracy for some operation phases and degrade the process monitoring performance. To overcome this drawback, Luo et al.[19] proposed a hybrid modelling strategy. The basic idea of hybrid modelling is to model each operation phase using an appropriate modelling technique respectively, by considering process characteristics, modelling accuracy, and modelling efficiency. So far, many different kinds of modelling techniques have been developed, such as principal component analysis (PCA),[20] support vector regression (SVR),[21] Gaussian mixture model (GMM),[22] just-in-time (JIT) modelling,[23] Tucker decomposition,[24] and so on. These modelling techniques have respective advantages in coping with linear, nonlinear, non-Gaussian, time-varying, or multiway process data. This enables us to implement hybrid modelling methods in multiphase batch processes.

In this paper, a trajectory-based fuzzy phase partition (TBFPP) method and hybrid multiway PCA (MPCA) models are developed for monitoring uneven multistage/multiphase batch processes. To simultaneously handle the uneven duration problem of batches and the transitions between operation phases, the TBFPP method implements fuzzy phase partition separately for each batch by clustering the trajectory data of phase-sensitive process variables using the SCFCM algorithm. Fuzzy operation phases of each batch are further divided into "steady" and "transition" operation phases according to membership degrees of samples. A phase identification combination index (PICI) is used for online phase identification. This PICI quantifies the time and spatial correlations between a new sample and all operation phases. Each new sample is assigned into the operation phase with the smallest PICI. Hybrid modelling methods are applied to model "steady" and "transition" operation phases separately, with offline phase-based MPCA models built for "steady" operation phases while online just-in-time MPCA models are built for "transition" operation phases. A process monitoring method is developed on the basis of hybrid MPCA models. The effectiveness and advantages of proposed methods are illustrated with a fed-batch penicillin fermentation process.

METHODOLOGY

Fuzzy Phase Partition and Online Phase Identification

In a multiphase batch process, process characteristics and variable correlations vary with operation phases. Besides, the transition region between two adjacent operation phases has very complicated process characteristics and variable correlations. Each operation phase and transition region between operation phases should be modelled separately to improve the modelling accuracy. Therefore, a batch process needs to be divided into different operation phases before modelling. The task of data-driven phase partition is to identify known and latent operation phases as well as their transition regions from historical batch data. Online phase identification aims to determine the current operation phase of a new batch. The uneven duration problem brings difficulties to both phase partition and online phase identification. In an uneven batch process, the same operation phase and the same transition region may have varying durations in different batches,[16,17] as shown in Figure 1. In other words, boundaries between operation phases are not fixed but vary with batches. In such a case, if performing phase partition on all batches simultaneously, it is difficult to get high phase partition accuracy for each batch. A better method is to carry out phase partition for each batch separately. The difficulty lies in ensuring consistency between phase partition results of all batches. In other words, all batches should have the same number of operation phases, and the same operation phase in different batches should have very similar process characteristics. On the other hand, because of the uneven duration problem, only using the operating time to determine the current operation phase of a new batch is not applicable for an uneven batch process. Thus, similarities between a new sample and historical samples in each operation phase should be measured and used for online phase identification. This increases the difficulty of online phase identification.

Trajectory-based fuzzy phase partition

Trajectories of process variables visually reflect the operation status of the batch process. Generally, switches between operation phases cause obvious trend variations in trajectories of some phase-sensitive variables. These phase-sensitive variables can be found out by inspecting the trajectories of all variables in a reference batch. As the operation phase switches from one to the other, the variables whose trajectories show inerratic trend changes are regarded as phase-sensitive variables; however, the trajectories of phase-insensitive variables only show irregular fluctuations or remain stable due to control actions. In many multiphase batch processes, some phase-sensitive variables serve as indicator variables for switching operation phases. The operation phase switches from one to the other if values of

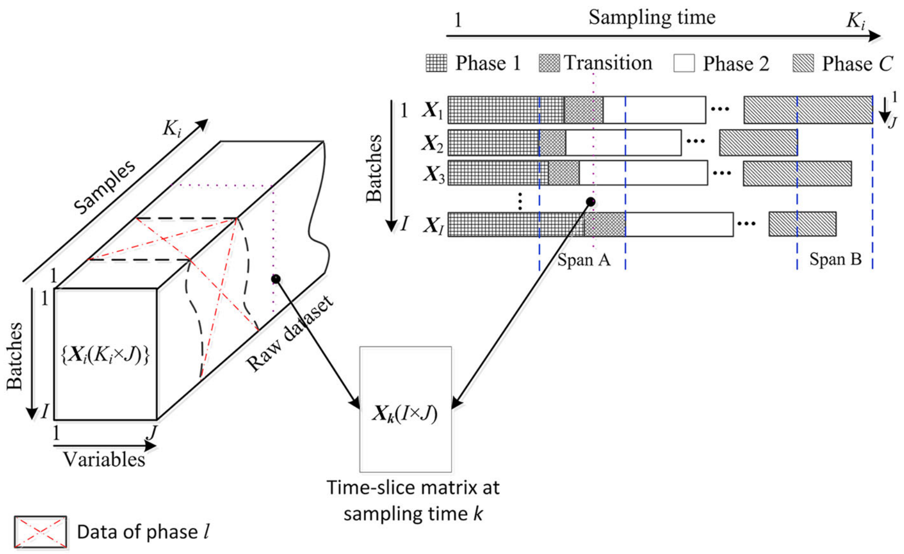

Figure 1. Operation phases in a multiphase batch process with uneven durations.

indicator variables exceed pre-set thresholds. Turning points in the trajectories of these indicator variables exactly correspond to switch points between operation phases. Moreover, owing to the good repeatability of a batch process, trend variations in variable trajectories in different batches are very similar regardless of whether or not batches have equal durations. This feature is able to ensure the consistency between phase partition results of different batches. Therefore, phase partition can be carried out by detecting changes in trajectories of phase-sensitive variables. According to this idea, a SCFCM-based fuzzy phase partition method has been proposed for the batch process with even durations.[19] This method divides the batch process into several fuzzy operation phases by clustering time slices $\{X_k(I \times J)\}_{k=1}^{L}$ of all batches (Figure 1) using the SCFCM algorithm. However, in an uneven batch process, a time slice X_{k} may consist of samples from different operation phases, and time slices at tails of batches have varying sizes, such as time slices in span A and span B in Figure 1. Due to the existence of these irregular time slices, the SCFCM-based fuzzy phase partition method is not applicable for the uneven batch process.

In this paper, a trajectory-based fuzzy phase partition (TBFPP) method is developed to deal with the uneven multiphase batch process. Let $\underline{X}_s = \{X_i(K_i \times J_s)\}_{i=1}^{I}$ denote the trajectory data of J_s phase-sensitive variables in I historical batches, where K_i is the duration of the ith batch. First, the data set $\underline{X}_{\mathrm{s}}$ is mean centered via the following two steps: (1) unfolding $\underline{X}_{\mathrm{s}} = \{X_i(K_i{\times}J_s)\}_{i=1}^{I}$ into a matrix $X_{\mathrm{s}}(N \times J_s)$ with $N = \sum_{i=1}^{I} K_i$ along the variable direction, and (2) centering $X_{\mathrm{s}}(N \times J_s)$ to $\bar{X}_s(N \times J_s)$. Second, the SCFCM clustering algorithm[19] is applied separately to each batch $\bar{X}_i(K_i \times J_s)$ $(i = 1, \ldots, I)$ to divide it into C fuzzy clusters (i.e., fuzzy operation phase). All samples of a batch belong to C fuzzy clusters in the form of membership degrees.[19] It is noteworthy that the SCFCM algorithm adds a hard sequentiality constraint into the clustering procedure to cope with the sequential nature of batch data.[19] As a consequence of this hard sequentiality constraint, the samples obtained at the beginning and at the end of a batch belong only to the first and final fuzzy clusters respectively, while other samples are assigned into two adjacent clusters (e.g., the ith and $(i+1)$th clusters), with the membership degrees defined as:[19]

$$\begin{cases} u_{kj} = \dfrac{1}{\sum\limits_{h=i}^{i+1}\left(\dfrac{d_{kj}}{d_{kh}}\right)^{2/(s-1)}}, \text{if } j = i \text{ or } i+1 \\ u_{kj} = 0, \text{otherwise} \end{cases} \qquad (1)$$

where u_{kj} is the membership degree of the kth sample x_{k} in the jth fuzzy cluster, $d_{kj} = ||x_k - m_j||$ is the Euclidean distance between the sample x_{k} and the centre m_{j} of the jth cluster, and $1 \le s < \infty$ is a parameter for adjusting the "blending" of different fuzzy clusters.[19] According to membership degrees of samples, the C fuzzy operation phases in each batch are finally divided into C "steady" operation phases and C-1 "transition" phases. As shown in Figure 2, the lth "steady" operation phase contains samples x_{k} that have membership degrees $u_{lk} \ge 0.9$ in the lth fuzzy phase. The "transition" phase between the lth and $(l+1)$th "steady" phases consists of samples x_{k} with membership degrees smaller than 0.9 in both the lth and $(l+1)$th "steady" phases.

An important issue of phase partition is choosing a proper number of fuzzy operation phases. The phase number can be chosen according to process knowledge, such as the number of known operation phases or actual operation modes in the batch process. However, this method may be subjective and sometime fails to select a proper phase number due to the existence of latent operation phases. An alternative method is choosing an optimal phase number C^* from a set of phase numbers $\{1, \ldots, \bar{C}\}$ according to a partition performance combination index (PPCI) that is defined as:

$$PPCI_C = \gamma \hat{L}_C + (1-\gamma)\hat{C} \qquad (2)$$

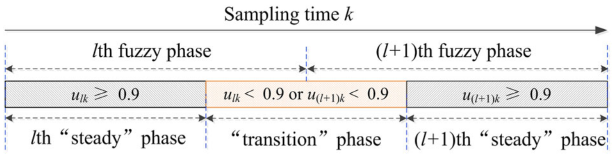

Figure 2. Partition of "steady" and "transition" phases.

with

$$\hat{L}_C = \frac{\tilde{L}_C - \mathrm{mean}(\tilde{\boldsymbol{L}})}{\mathrm{std}(\tilde{\boldsymbol{L}})}, \hat{C} = \frac{C - \mathrm{mean}(\boldsymbol{C})}{\mathrm{std}(\boldsymbol{C})} \tag{3}$$

$$\tilde{L}_C = \log(L_C), L_C = \sum\nolimits_{i=1}^{I} L_{i,C} \tag{4}$$

where $\gamma \in (0,1)$ is a tradeoff coefficient, $C = 1, \ldots, \bar{C}$ is the phase number, $\hat{L}_C$ and $\hat{C}$ are normalized values of $\tilde{L}_C$ and C, $\tilde{\boldsymbol{L}} = [\tilde{L}_1, \ldots, \tilde{L}_{\bar{C}}]$ and $\boldsymbol{C} = [1, \cdots, \bar{C}]$ are two vectors, mean$(\cdot)$ and std$(\cdot)$ denote mean and standard variance of a vector, and log$(\cdot)$ is the natural logarithm. The $L_{i,C}$ in Equation (4) is the sum-of-squared-error (SSE) of ith batch, which is computed by:

$$L_{i,C} = \sum_{l=1}^{C} \sum_{k=1}^{K_i} u_{lki}^{s} ||\boldsymbol{x}_{i,k} - \boldsymbol{m}_{l,i}||^2 \tag{5}$$

where u_{lki} is the membership degree of kth sample $\boldsymbol{x}_{\mathrm{i},k}$ in lth fuzzy phase in ith batch, $1 \leq s < \infty$ is a parameter used in the SCFCM algorithm, $\boldsymbol{m}_{\mathrm{l},i}$ is the centre of lth fuzzy phase in ith batch. The u_{lki} and $\boldsymbol{m}_{\mathrm{l},i}$ in Equation (5) are obtained by performing the SCFCM algorithm[19] on ith batch $\bar{X}_i(K_i \times J_s)$. The $\hat{L}_C$ in Equation (2) quantifies the phase partition performance for all batches, where a smaller $\hat{L}_C$ indicates a better partition performance. The $\hat{C}$ in Equation (2) represents the modelling complexity. The coefficient $\gamma \in (0,1)$ is used to adjust the tradeoff between partition performance and modelling complexity. A larger γ puts more importance on the partition performance than the modelling complexity. Generally, a larger phase number leads to better partition performance (i.e., a smaller $\hat{L}_C$), but increases the complexity of process modelling (i.e., a larger $\hat{C}$). Therefore, to obtain better partition performance and simultaneously to decrease the modelling complexity, the phase number corresponding to the smallest PPCI is optimal.

Online phase identification

A new sample of the current batch needs to be allocated into a certain operation phase before process monitoring. This online phase identification procedure can be implemented by evaluating the correlation between a new sample and each operation phase. The sampling time represents the time correlation between a sample and an operation phase, and the distance of a sample from the phase centre reflects their spatial correlation. Simultaneously considering time and spatial correlations, a phase identification combination index (PICI) of kth sample $\boldsymbol{x}_{\mathrm{k}}(1 \times J_s)$ in lth operation phase is defined as:

$$PICI_{k,l} = \beta \hat{d}_{k,l} + (1-\beta)\hat{t}_{k,l} \tag{6}$$

with

$$\hat{d}_{k,l} = \frac{\tilde{d}_{k,l} - \min(\tilde{\boldsymbol{d}}_k)}{\max(\tilde{\boldsymbol{d}}_k) - \min(\tilde{\boldsymbol{d}}_k)}, \hat{t}_{k,l} = \frac{\tilde{t}_{k,l} - \min(\tilde{\boldsymbol{t}}_k)}{\max(\tilde{\boldsymbol{t}}_k) - \min(\tilde{\boldsymbol{t}}_k)} \tag{7}$$

$$\tilde{d}_{k,l} = \sum\nolimits_{i=1}^{I} d_{i,k,l},\ \tilde{t}_{k,l} = \sum\nolimits_{i=1}^{I} t_{i,k,l} \tag{8}$$

$$t_{i,k,l} = \frac{|k - (b_{i,l+1} + b_{i,l})/2|}{b_{i,l+1} - b_{i,l}}, d_{i,k,l} = ||\boldsymbol{x}_k - \boldsymbol{\mu}_{i,l}||, \boldsymbol{\mu}_{i,l} = \sum_{k=b_{i,l}}^{b_{i,l+1}-1} \frac{\boldsymbol{x}_{i,k}}{n_{i,l}} \tag{9}$$

where k is the sampling time, $l = 1, \ldots, 2C$-1 is the phase index, $i = 1, \ldots, I$ is the batch index, $b_{i,l}$ denotes the start time of lth phase in ith batch, $n_{i,l}$ denotes the number of samples in lth phase in ith batch, $d_{i,k,l}$ and $t_{i,k,l}$ are spatial distance and time difference between $\boldsymbol{x}_{\mathrm{k}}$ and the centre $\boldsymbol{\mu}_{i,l}(1 \times J_s)$ of lth phase in ith batch, $\tilde{d}_{k,l}$ and $\tilde{t}_{k,l}$ are summations of spatial distances and time differences over all batches, $\hat{d}_{k,l}$ and $\hat{t}_{k,l}$ are normalized values of $\tilde{d}_{k,l}$ and $\tilde{t}_{k,l}$, $\tilde{\boldsymbol{d}}_k = \left\{\tilde{d}_{k,l}\right\}_{l=1}^{2C-1}$ and $\tilde{\boldsymbol{t}}_k = \left\{\tilde{t}_{k,l}\right\}_{l=1}^{2C-1}$ are two data sets that consist of spatial distances and time differences in all operation phases, and max$(\cdot)$ and min$(\cdot)$ denote maximal and minimal values in a data set. The $\beta \in [0,1]$ in Equation (6) is a weight coefficient for adjusting the importance of spatial distances and time differences to online phase identification. Using a larger β will increase the effect of the spatial distance on phase identification results. The sample $\boldsymbol{x}_{\mathrm{k}}$ is assigned into the operation phase with the smallest PICI. Figure 3 illustrates phase partition and online phase identification procedures.

Hybrid Modelling

After phase partition, process data in the same operation phase in all batches constitute phase data sets $\underline{\boldsymbol{X}}_l = \{\boldsymbol{X}_{i,l}(\boldsymbol{K}_{i,l} \times \boldsymbol{J})\}_{i=1}^{I}$, $l = 1, \ldots, 2C$-1. Process model for each operation phase can be built based on the phase data set. Because "steady" and "transition" phases have very different process characteristics, they are modelled using hybrid modelling methods consisting of phase-based (global) modelling and just-in-time (local) modelling. The phase-based modelling method builds a global phase model using all historical samples of an operation phase. This modelling method can be carried out offline and has higher modelling accuracy for the operation phase with steady process characteristics. The just-in-time (JIT) modelling method builds a local model for each new sample based on its relevant samples in historical batches.[23] This modelling method must be carried out online and can effectively cope with time-varying process characteristics, while the modelling efficiency is low because the whole modelling procedure is repeated for every new sample. In this paper, MPCA[1] is applied to implement hybrid modelling

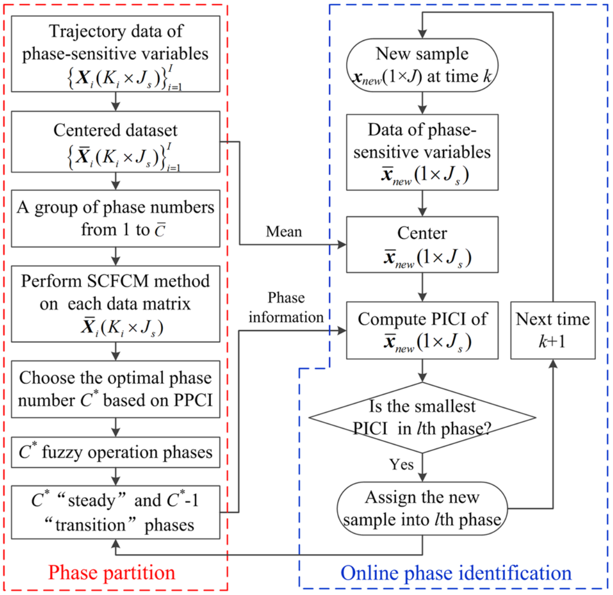
Trajectory data of phase-sensitive variables $\{X_i(K_i \times J_s)\}_{i=1}^{I}$
Centered dataset $\{\bar{X}_i(K_i \times J_s)\}_{i=1}^{I}$
A group of phase numbers from 1 to $\bar{C}$
Perform SCFCM method on each data matrix $\bar{X}_i(K_i \times J_s)$
Choose the optimal phase number C^* based on PPCI
C^* fuzzy operation phases
C^* "steady" and C^*-1 "transition" phases
Phase partition
Mean
Phase information
New sample $x_{new}(1\times J)$ at time k
Data of phase-sensitive variables $\bar{x}_{new}(1\times J_s)$
Center $\bar{x}_{new}(1\times J_s)$
Compute PICI of $\bar{x}_{new}(1\times J_s)$
Next time k+1
Is the smallest PICI in lth phase?
Yes
Assign the new sample into lth phase
Online phase identification

Figure 3. Phase partition and online phase identification procedures.

methods, resulting in phase-based MPCA (PMPCA) model and JIT-MPCA model, respectively.

Modelling steady operation phases

Since "steady" operation phases have relatively steady process behaviours and longer durations, they are modelled by the phase-based modelling method. Denote the data set of lth "steady" phase as $\{X_{i,l}(K_{i,l}\times J)\}_{i=1}^{I}$, where J is the number of process variables and $K_{i,l}$ is the duration of lth phase in ith batch. Before building PMPCA models, each phase data set $\{X_{i,l}(K_{i,l}\times J)\}_{i=1}^{I}$ is normalized to $\{\bar{X}_{i,l}(K_{i,l}\times J)\}_{i=1}^{I}$ via the batch-wise normalization method.[17] The batch-wise normalization can remove the average batch trajectory from all batches and highlight the batch-to-batch variation. The normalized data set $\{\bar{X}_{i,l}(K_{i,l}\times J)\}_{i=1}^{I}$ is then unfolded along the variable direction to $\{\hat{X}_l(N_l\times J)\}$ with $N_l=\sum_{i=1}^{I}K_{i,l}$. The PMPCA model of lth "steady" phase is built as:

$$\hat{X}_l=\sum_{r=1}^{R_l}\hat{t}_{r,l}\hat{p}_{r,l}^T+\hat{E}_l=\hat{T}_l\hat{P}_l^T+\hat{E}_l \tag{10}$$

where R_l is the number of latent variables, and $\hat{T}_l(N_l\times R_l)$, $\hat{P}_l(J\times R_l)$, and $\hat{E}_l(N_l\times J)$ are score, loading, and residual matrices, respectively.

Modelling transition phases

Due to the switch between two operation phases, a "transition" phase often has time-varying process behaviours and a shorter duration. Thus, the JIT modelling method is used for "transition" phases to improve the modelling accuracy. Suppose $x_{new,k}(J\times 1)$ is a new sample at kth sampling time in lth "transition" phase. A JIT-MPCA model is built for $x_{new,k}$ via the JIT modelling method. Firstly, H relevant samples of $x_{new,k}$ are selected from the lth phase data set $\{X_{i,l}(K_{i,l}\times J)\}_{i=1}^{I}$ according to the nearest neighbour rule. In other words, relevant samples are the H nearest neighbours of $x_{new,k}$. These relevant samples constitute a training data set $X_k(H\times J)$. After normalizing $X_k(H\times J)$ to $\tilde{X}_k(H\times J)$, the JIT-MPCA model for $x_{new,k}$ is built as:

$$\tilde{X}_k=\sum_{r=1}^{R_k}\tilde{t}_{r,k}\tilde{p}_{r,k}^T+\tilde{E}_k=\tilde{T}_k\tilde{P}_k^T+\tilde{E}_k \tag{11}$$

where all model parameters have similar definitions as those in Equation (10). Note that each JIT-MPCA model is only used for a new sample. It is discarded immediately after use. A totally new JIT-MPCA model is built for the next new sample.

Process Monitoring

The T^2 and SPE statistics are used for process monitoring. They monitor data variations in model subspace and residual subspace, respectively. The T^2 and SPE statistics of a sample $x_{k,l}(1\times J)$ in lth "steady" phase are defined as:

$$T_{k,l}^2=\left(t_{k,l}-\bar{t}_l\right)S_l^{-1}\left(t_{k,l}-\bar{t}_l\right)^T \tag{12}$$

$$SPE_{k,l}=e_{k,l}e_{k,l}^T=\sum_{j=1}^{J}e_{j,k,l}^2 \tag{13}$$

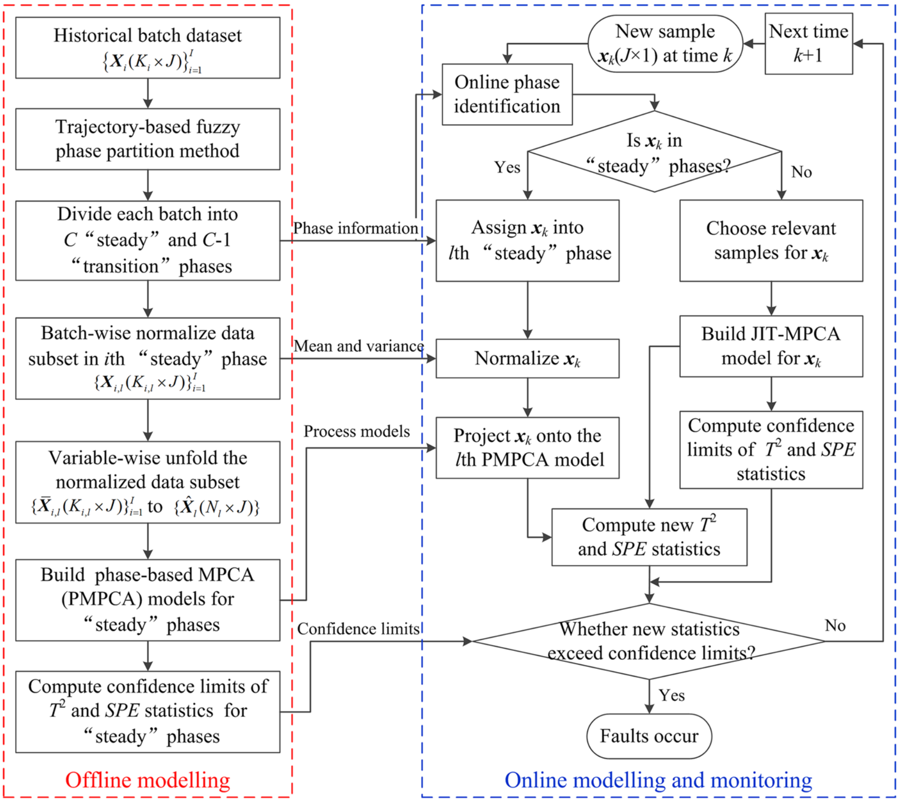
Historical batch dataset
$\{X_i(K_i\times J)\}_{i=1}^{I}$
Trajectory-based fuzzy phase partition method
Divide each batch into C "steady" and C-1 "transition" phases
Batch-wise normalize data subset in ith "steady" phase $\{X_{i,l}(K_{i,l}\times J)\}_{i=1}^{I}$
Variable-wise unfold the normalized data subset $\{\bar{X}_{i,l}(K_{i,l}\times J)\}_{i=1}^{I}$ to $\{\hat{X}_l(N_l\times J)\}$
Build phase-based MPCA (PMPCA) models for "steady" phases
Compute confidence limits of T^2 and SPE statistics for "steady" phases
Offline modelling
Phase information
Mean and variance
Process models
Confidence limits
New sample $x_k(J\times 1)$ at time k
Next time k+1
Online phase identification
Is x_k in "steady" phases?
Yes
No
Assign x_k into lth "steady" phase
Choose relevant samples for x_k
Normalize x_k
Build JIT-MPCA model for x_k
Project x_k onto the lth PMPCA model
Compute confidence limits of T^2 and SPE statistics
Compute new T^2 and SPE statistics
Whether new statistics exceed confidence limits?
No
Yes
Faults occur
Online modelling and monitoring

Figure 4. Modelling and online monitoring procedures.

where $\boldsymbol{t}_{k,l}(1 \times R_l)$ and $\boldsymbol{e}_{k,l}(1 \times J)$ are scores and residuals of $\boldsymbol{x}_{k,l}$, $\bar{\boldsymbol{t}}_l$ is the row mean of the score matrix $\hat{\boldsymbol{T}}_l(N_l \times R_l)$ in Equation (9), and $\boldsymbol{S}_l(R_l \times R_l)$ is the covariance matrix of centered $\hat{\boldsymbol{T}}_l$. For a sample in a "transition" phase, its T^2 and SPE statistics are also defined by Equation (12) and Equation (13) after replacing $\hat{\boldsymbol{T}}_l$ with $\tilde{\boldsymbol{T}}_k$.

Confidence limits of T^2 and SPE statistics are calculated by the kernel density estimation (KDE) method.[25] For lth "steady" phase, the distribution of T^2 values of all training samples in the data set $\hat{X}_l(N_l \times J)$ is estimated by KDE:[25]

$$\hat{f}_l(T^2) = \frac{1}{N_l\theta}\sum_{n=1}^{N_l} K\left(\frac{T^2 - T_n^2}{\theta}\right) \tag{14}$$

where T_n^2 denotes the T^2 value of the nth training sample, N_l is the total number of training samples, θ is a smoothing parameter, and $K(\cdot)$ is a Gaussian kernel function. Similarly, for the kth sample in a "transition" phase, the distribution of T^2 values of all training samples in the data set $\tilde{X}_k(H \times J)$ is estimated by Equation (14) after replacing N_l with H. The confidence limit of the T^2 statistic at the significant level α is then computed by the inverse cumulative distribution function of $\hat{f}_l(T^2)$. The confidence limit of the SPE statistic is computed in the similar way as the T^2 statistic.

In online monitoring, when a new sample $\boldsymbol{x}_{new,k}(J \times 1)$ is measured at sampling time k, it is assigned into an operation phase via the online phase identification method. Then, $\boldsymbol{x}_{new,k}(J \times 1)$ is projected on the corresponding model to compute new T^2 and SPE values. If $\boldsymbol{x}_{new,k}$ is in lth "steady" phase, its scores $\boldsymbol{t}_{new,k}$ and residuals $\boldsymbol{e}_{new,k}$ are computed by:

$$\boldsymbol{t}_{new,k} = \hat{\boldsymbol{P}}_l^T \boldsymbol{x}_{new,k} \quad \boldsymbol{e}_{new,k} = \boldsymbol{x}_{new,k} - \hat{\boldsymbol{P}}_l \hat{\boldsymbol{P}}_l^T \boldsymbol{x}_{new,k} \tag{15}$$

where $\hat{\boldsymbol{P}}_l$ is the loading matrix in Equation (10). If $\boldsymbol{x}_{new,k}$ is in a "transition" phase, its scores and residuals are computed via Equation (15) after replacing $\hat{\boldsymbol{P}}_l$ with $\tilde{\boldsymbol{P}}_k$. Substituting scores and residuals of $\boldsymbol{x}_{new,k}$ into Equation (12) and Equation (13), new T^2 and SPE values are obtained. If the T^2 or SPE value exceeds the corresponding confidence limit, a fault is detected. The complete modelling and online procedures are illustrated in Figure 4.

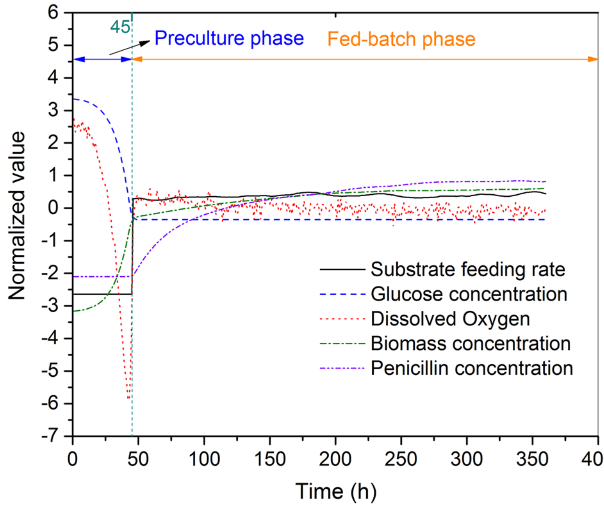

Figure 5. Normalized trajectories of 5 phase-sensitive process variables in a normal batch.

CASE STUDY

Process Description

Proposed methods are tested in a fed-batch penicillin fermentation process.[26] This process is a typical multiphase batch process that contains two real operation phases: a preculture phase and a fed-batch phase. Figure 5 shows normalized trajectories of 5 phase-sensitive process variables in a normal batch. The process starts with the preculture operation phase during which biomass grows quickly by consuming the glucose in the initial culture substrate. When the glucose concentration decreases to a threshold value, the process immediately switches from the preculture phase to fed-batch phase. During the fed-batch phase, additional glucose is fed into the fermentor to produce penicillin. This batch process is simulated in a simulator named PenSim v2.0.[26] A total of 60 normal batches with small random variations in startup initialization are generated. These normal batches constitute a training data set. The durations of batches vary between 350 and 400 h, where the preculture phase lasts about 45 h and the fed-batch phase lasts longer than 305 h. As listed in Table 1, 13 process variables are measured every 1 h during the batch run. In addition, 12 fault batches are generated under the fault modes in Table 2.

Phase Partition Results

Although the above batch process only has two actual operation phases, process behaviours in each operation phase are not consistent, but obviously change along the operating time, as shown in Figure 5. This implies that there exist latent sub-phases in two actual operation phases. To improve the modelling accuracy, two actual operation phases and the latent sub-phases in each batch are identified by the trajectory-based fuzzy phase partition method. Trajectory data of 5 phase-sensitive variables, i.e., the substrate concentration, biomass concentration, dissolved oxygen saturation, substrate feeding rate, and generated heat, are used for phase partition. Phase numbers ranging from 1–10 are tested to choose an optimal one. The PPCI of each phase number is computed by Equation (2) with $\gamma = 0.6$, as shown in Figure 6. Obviously, the optimal phase number is 5 because it has the smallest PPCI.

Figure 7 shows membership degrees of all samples in 5 fuzzy operation phases in a training batch. It can be seen that 5 fuzzy operation phases sequentially distribute along the operating time,

Table 1. Process variables in the fed-batch penicillin fermentation process

No.	Process variables	Units
1	Aeration rate	L/h
2	Agitator power input	W
3	Substrate feeding rate	L/h
4	Substrate feeding temperature	K
5	Substrate (glucose) concentration	g/L
6	Dissolved oxygen saturation	%
7	Culture volume	L
8	CO_2 concentration	mmol/L
9	pH	
10	Bioreactor temperature	K
11	Generated heat	kcal/h
12	Biomass concentration	g/L
13	Penicillin concentration	g/L

Table 2. Fault batches in the fed-batch penicillin fermentation process

No.	Fault variable	Magnitude	Fault type	Start time (h)	End time (h)	Batch duration (h)
1	v2	+5 %	Step	20	40	380
2	v3	−5 %	Step	46	100	376
3	v2	+5 W	Ramp	20	150	371
4	v1	−5 %	Step	38	150	351
5	v3	+5 %	Step	46	150	392
6	v3	+0.02 L/h	Ramp	60	300	393
7	v1	−2 L/h	Ramp	100	200	356
8	v1	−3 %	Step	150	250	364
9	v1	−3 L/h	Ramp	60	300	373
10	v3	+0.005 L/h	Ramp	46	150	382
11	v3	−5 %	Step	46	100	397
12	v2	−5 %	Step	20	100	363

and each fuzzy operation phase consists of consecutive sampling instants. Besides, each sample either belongs to one fuzzy operation phase or two neighbouring fuzzy operation phases. These features ensure that the phase partition result is reasonable and interpretable. According to whether or not membership degrees of samples are larger than 0.9, 5 fuzzy operation phases are divided into 5 "steady" phases and 4 "transition" phases. The ranges of start time and durations of 9 phases in all training batches are listed in Table 3. Note that the 5th operation phase starts at around 45 h in all batches, which agrees well with the switch time between two actual operation phases, i.e., preculture phase and fed-batch phase. Therefore, the first 4 phases are latent sub-phases of the preculture phase, and the last 5 phases are latent sub-phases of the fed-batch phase.

Online Process Monitoring Results

PMPCA models are built for "steady" phases 1, 3, 5, 7 and 9, while JIT-MPCA models are built in "transition" phases 2, 4, 6 and 8. To illustrate advantages of these hybrid MPCA (HMPCA) models, they are compared with a single MPCA (SMPCA) model. The SMPCA model is built by treating the entire batch as a single operation phase and modelling it without using phase partition. In all MPCA models, the number of principal components (PCs) is determined by the cumulative percent variance (CPV) method. The retained PCs explain more than 85 % of the variance in data. The 12 fault batches in Table 2 are used to test the monitoring abilities of HMPCA and SMPCA models. The 99 % confidence limits, namely at the significance level of 0.01, of T^2 and *SPE* statistics are used for fault detection. The monitoring performance of each model is quantified by three indexes: fault detection rate (FDR), false alarm rate (FAR), and fault detection delay (FDD). FDR and FAR are defined as FDR $= n_f/n_{tf}$ and FAR $= n_o/n_{to}$, where n_f or n_o denotes the number of faulty samples detected in fault conditions or normal conditions, n_{tf} or n_{to} is the total number of samples in fault conditions or normal conditions. FDD is defined as FDD $= t_d$ - t_c, where t_c is the fault occurrence time and t_d is the fault detection time.

Table 4 compares monitoring results of two models for 12 fault batches. Because the FAR is equal to zero in all cases, its value is not listed. Table 4 shows that HMPCA gives higher FDRs than SMPCA for all the 12 faults. Especially for faults 5, 8, 11, and 12, FDRs of T^2 and *SPE* statistics of HMPCA are much higher than those of SMPCA. Mean FDRs (MFDRs) of T^2 and *SPE* statistics of HMPCA increase by more than 6 % as compared to SMPCA, indicating the significant improvement in monitoring performance. Note that the *SPE* statistic of HMPCA

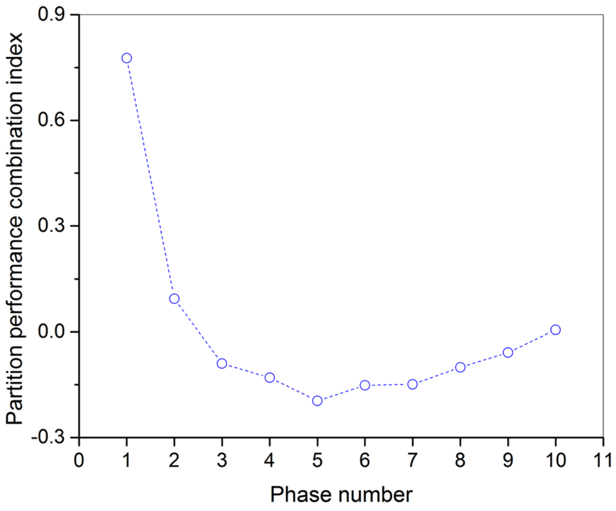

Figure 6. Partition performance combination indexes of different phase numbers.

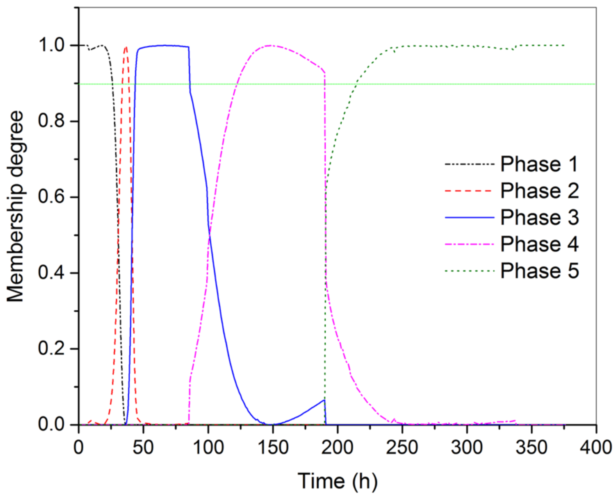

Figure 7. Membership degrees of all samples in 5 fuzzy operation phases in a training batch.

Table 3. Ranges of start time and durations of 9 operation phases in 60 training batches

Phase No.	1	2	3	4	5	6	7	8	9
Start time (h)	1	26–28	33–35	38–40	43–45	80–94	113–129	172–201	198–220
Duration (h)	24–27	7–8	5–6	4–5	37–50	29–37	56–72	24–33	154–177

detects all faults without time delay and with FDDs higher than 99 %. However, SMPCA has FDDs of 2 h for faults 2, 5, 10, and 11 that are caused by disturbances in the substrate feeding rate (v3) at 46 h.

Figure 8 and Figure 9 show monitoring charts of two models for fault batches 8 and 11, respectively. Fault 8 occurred at 150 h and ended at 250 h, corresponding to operation phases 7, 8, and 9. As shown in Figure 8a, T^2 and *SPE* statistics of SMPCA miss several faulty samples in fault batch 8. These undetected faulty samples are wrongly regarded as normal samples, because the SMPCA model cannot accurately describe the specific variable correlation in each operation phase. Different from the SMPCA model, HMPCA models are able to capture important process characteristics of all operation phases in that each operation phase is modelled separately. As a result, all of the faulty samples are successfully detected by T^2 and *SPE* statistics of HMPCA (Figure 8b). Especially, in all "transition" phases, confidence limits of T^2 and *SPE* statistics vary with the operating time, owing to the ability of JIT-MPCA models in tracking dynamic process characteristics. Fault 11 occurred in operation phases 5 and 6. As shown in Figure 9a, SMPCA detects fault 11 at 48 h, which lags behind the fault occurrence time by 2 h. Besides, some faulty samples in fault batch 11 are undetected by T^2 and *SPE* statistics of SMPCA, resulting in a smaller FDR. Figure 9b shows that T^2 and *SPE* statistics of HMPCA successfully detect all faulty samples in fault batch 11 with no time delay. The above results validate that HMPCA has better monitoring ability than SMPCA.

To further illustrate the differences between the SMPCA and HMPCA models, the score plots of the first two principal components of two models for all samples in a normal batch are shown in Figure 10. It can be seen from Figure 10a that data points in the preculture phase are far from those in the fed-batch phase. Data points in the preculture phase (from 1–41 h) are scattered in the upper left corner of Figure 10a, while data points (from 46–386 h) in the fed-batch phase gather together in the lower right region of Figure 10a. In particular, data points obtained in the time period from 42–45 h are located between the preculture phase and the fed-batch phase. These features indicate that process characteristics vary with operation phases and a SMPCA model is not adequate to accurately model all operation phases. Consequently, the monitoring performance of SMPCA is lower. Figure 10b shows that the data points of each operation phase group together in a small area. This implies that the HMPCA models are able to capture specific process characteristics of each operation phase, and therefore they have better monitoring performance than the SMPCA model.

Table 4. FDR and FDD of SMPCA and HMPCA models for fault batches

	SMPCA				HMPCA			
	FDR (%)		FDD (h)		FDR (%)		FDD (h)	
Fault no.	T^2	*SPE*	T^2	*SPE*	T^2	*SPE*	T^2	*SPE*
1	95	95	0	0	100	100	0	0
2	84	96	2	2	100	100	0	0
3	93	95	0	0	100	99	0	0
4	95	95	0	0	100	100	0	0
5	82	90	2	2	**89**	**100**	0	0
6	93	93	0	0	85	99	0	0
7	95	98	0	0	98	100	0	0
8	91	91	0	0	**100**	**100**	0	0
9	96	96	0	0	100	100	0	0
10	82	95	2	2	100	100	0	0
11	84	84	2	2	**100**	**100**	0	0
12	93	93	0	0	**100**	**100**	0	0
MFDR	90.2	93.5	\	\	**97.6**	**99.8**	\	\

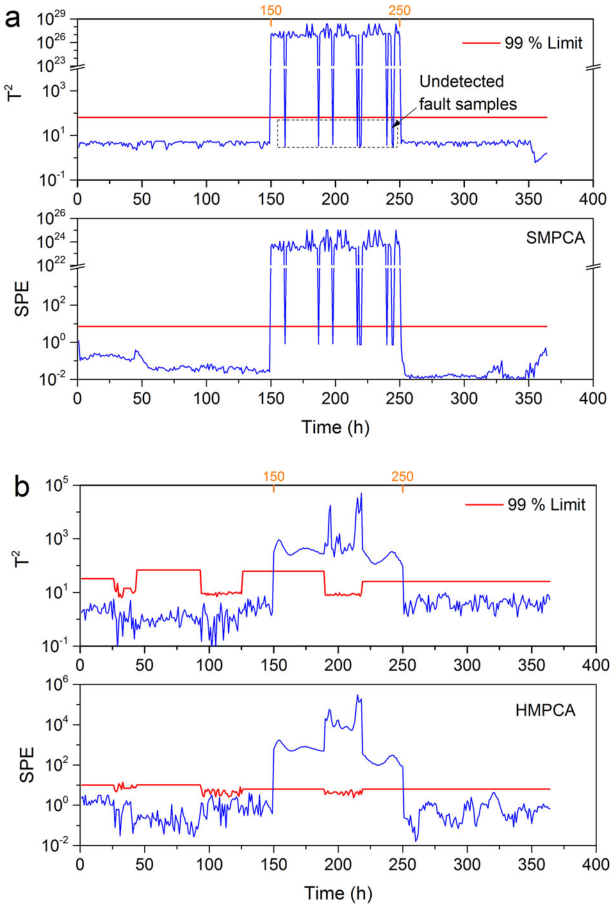

Figure 8. Monitoring charts of (a) SMPCA and (b) HMPCA for fault batch 8.

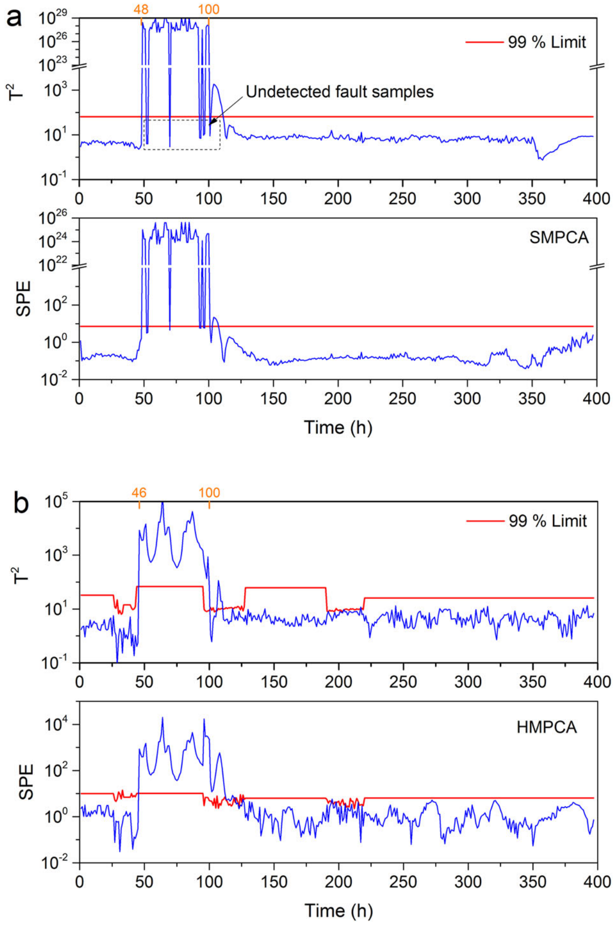

Figure 9. Monitoring charts of (a) SMPCA and (b) HMPCA for fault batch 11.

CONCLUSIONS

A trajectory-based fuzzy phase partition (TBFPP) method and hybrid MPCA models are developed for monitoring uneven multistage/multiphase batch processes. The TBFPP method divides each batch into fuzzy operation phases by clustering trajectory data of the phase-sensitive process variable using the SCFCM algorithm. Fuzzy operation phases in each batch are further divided into "steady" and "transition" phases according to membership degrees of samples. A phase identification combination index (PICI) is used for online phase identification. Each sample of a new batch is assigned into the operation phase with the smallest PICI. The "steady" and "transition" phases are modelled using hybrid modelling methods consisting of phase-based modelling and just-in-time modelling, with phase-based MPCA (PMPCA) built for "steady" phases and JIT-MPCA models built for "transition" phases. An online process monitoring method is developed on the basis of the hybrid MPCA models. The proposed methods are applied to a fed-batch penicillin fermentation process. The results indicate that the TBFPP method not only identifies two actual operation phases in the batch process accurately, but also reveals latent operation sub-phases and transition regions between neighbouring phases. In comparison with the single MPCA (SMPCA) model, the hybrid MPCA (HMPCA) models are more suitable for describing the different process characteristics of "steady" and "transition" phases. Therefore, the HMPCA-based monitoring method outperforms the SMPCA-based method in terms of higher fault detection rate and shorter fault detection delay.

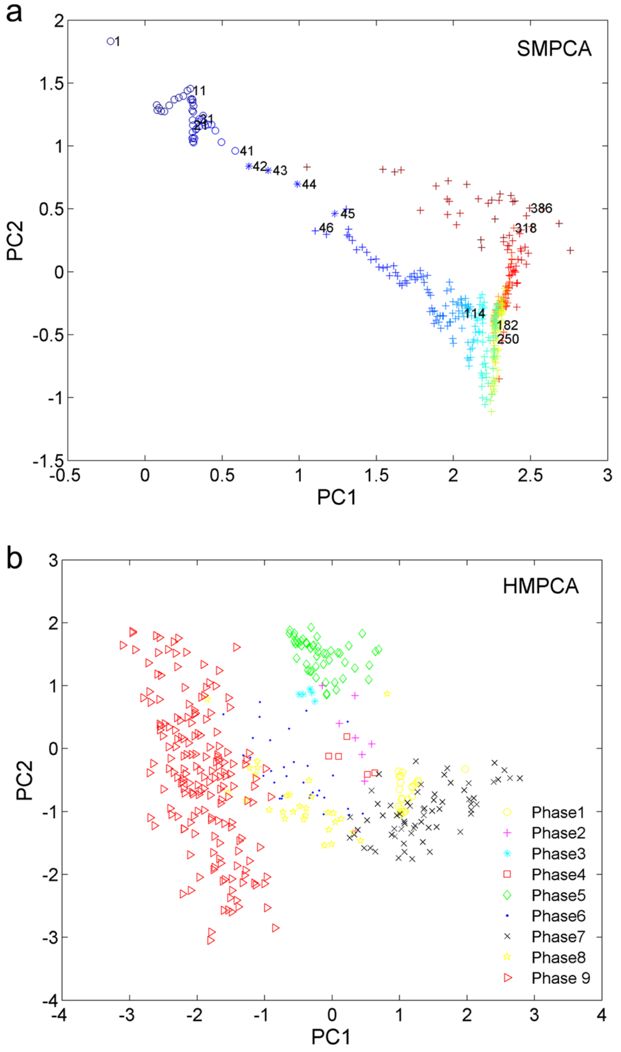

Figure 10. Score plots of first two principal components (PC) of (a) SMPCA and (b) HMPCA for all samples in a normal batch. Numbers denotes the sampling time of data points.

ACKNOWLEDGEMENT

This study was supported by the National Natural Science Foundation of China (No. 61304116).

REFERENCES

[1] P. Nomikos, J. F. MacGregor, *AIChE J.* **1994**, *40*, 1361.

[2] J. F. MacGregor, A. Cinar, *Comput. Chem. Eng.* **2012**, *47*, 111.

[3] Y. Yao, F. R. Gao, *Annu. Rev. Control* **2009**, *33*, 172.

[4] L. Luo, S. Bao, Z. Gao, *Chemometr. Intell. Lab.* **2015**, *143*, 28.

[5] L. Luo, S. Bao, Z. Gao, J. Yuan, *Ind. Eng. Chem. Res.* **2014**, *53*, 15101.

[6] C. Undey, A. Cinar, *IEEE Contr. Syst. Mag.* **2002**, *22*, 40.

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

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

[9] K. Gollmer, C. Posten, *Control Eng. Pract.* **1996**, *4*, 1287.

[10] X. T. Doan, R. Srinivasan, *Comput. Chem. Eng.* **2008**, *32*, 230.

[11] J. Camacho, J. Pico, *J. Process Contr.* **2006**, *16*, 1021.

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

[13] N. Y. Lu, F. R. Gao, F. L. Wang, *AIChE J.* **2004**, *50*, 55.

[14] C. H. Zhao, Y. X. Sun, *Chemometr. Intell. Lab.* **2013**, *125*, 109.

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

[16] N. Y. Lu, F. R. Gao, Y. Yang, F. L. Wang, *Ind. Eng. Chem. Res.* **2004**, *43*, 3343.

[17] L. Luo, S. Bao, J. Mao, D. Tang, *Ind. Eng. Chem. Res.* **2016**, *55*, 2035.

[18] W. Q. Li, C. H. Zhao, F. R. Gao, *Ind. Eng. Chem. Res.* **2015**, *54*, 10020.

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

[20] S. Wold, K. Esbensen, P. Geladi, *Chemometr. Intell. Lab.* **1987**, *2*, 37.

[21] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York **1995**.

[22] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York **2006**.

[23] H. Kaneko, K. Funatsu, *AIChE J.* **2016**, *62*, 717.

[24] L. R. Tucker, *Psychometrika* **1966**, *31*, 279.

[25] E. B. Martin, A. J. Morris, *J. Process Contr.* **1996**, *6*, 349.

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

Manuscript received October 9, 2017; revised manuscript received February 14, 2018; accepted for publication March 15, 2018.