Data-Driven Dynamic Modeling and Online Monitoring for Multiphase and Multimode Batch Processes with Uneven Batch Durations

Kai Wang[a], Lee Rippon[b], Junghui Chen[c*], Zhihuan Song[a*], and R.Bhushan Gopaluni[b*]

[a] State Key Laboratory of Industrial Control Technology, Zhejiang University,
Hangzhou, 310027, Zhejiang, China

[b] Department of Chemical and Biological Engineering, The University of British
Columbia,Vancouver, BC, Canada

[c] Department of Chemical Engineering, Chung Yuan Christian University, Chungli,
Taoyuan, Taiwan, 32023, Republic of China

[*] Corresponding author. E-mail: jason@wavenet.cycu.edu.tw (J. Chen); songzhihuan@zju.edu.cn (Z. Song); bhushan.gopaluni@ubc.ca (R.B. Gopaluni)

Abstract: Batch processes are often characterized by piecewise linear dynamics due to varying operating conditions. Multiphase and multimode modeling of batch processes is a common technique that offers insight into the process operation and improved online monitoring. However, existing monitoring methods have several drawbacks such as neglecting process dynamics, requiring separate treatment of transient behavior and relying on uniformity between batches. These challenges are addressed here by proposing a new strategy to construct a dynamic model for monitoring multimode and multiphase batch processes. A linear dynamic system partitions phases and describes local dynamic behavior before modes of operation are clustered based on the global differences between batches. Lastly, an expectation maximization algorithm for multibatch data in the same mode is applied to estimate phase parameters. Process monitoring results on a benchmark penicillin fermentation dataset suggest a significant improvement over previous methods.

Keywords: batch process, multimode, multiphase, online monitoring, dynamics

1. Introduction

Batch and semi-batch processes are a class of chemical and manufacturing processes carried out over a finite duration. They are widely used in modern industries such as pharmaceutical, food, biotechnology and semiconductors [1]. Monitoring batch processes is an important task to ensure safety, product quality and waste reduction. However, the variability within and between batches creates modeling complexities and challenges for online monitoring. Multiphase batch processes are characterized by different stages of dynamic behavior that yield different process variable characteristics within each batch duration such as those shown for the penicillin fermentation process in Figure 1. The penicillin fermentation process is characterized by four distinct phases: (1) exponential or lag phase, (2) growth phase, (3) stationary phase and (4) death phase. Due to feedback control and inherent process characteristics, the process variables often exhibit strong temporal correlation. Therefore, the data collected over the duration of a batch is a series of nonstationary dynamic sequences [2, 3]. Since the root cause of phases in batches comes from varying process operations, all variables are considered to be synchronized in the evolution of phases.

In addition to variability within each batch, there is often considerable variance between batches due to varied initial conditions, inconsistent raw materials and uneven batch durations. It is generally possible to further characterize batch processes by the number and duration of intra-batch phases in each batch. In the rest of the article, batches with similar phase characteristics are considered to exhibit the same *mode* and a batch process that exhibits multiple modes is referred to as the multimode batch process. For example, the second phase in Figure 1 may be either exponential or logarithmic, depending on the feed component or other factors. Batches with the exponential trend in the second phase have a particular mode that is distinguished from those with a logarithmic second phase trend given the other phases are the same. The various modes in a multimode batch process with data-driven strategies are determined by clustering a large number of batches based on similar phase characteristics. The multimode behavior results in considerable variability in process variable trajectories across various batches which in turn poses a significant challenge for online monitoring. In particular, obtaining an accurate model from first-principles knowledge is often difficult due to the aforementioned challenges.

With a vast amount of historical process data available, a number of data-driven monitoring techniques have been developed in the past two decades. In particular, multiway principal component analysis (MPCA) and multiway partial least squares (MPLS) are two widely used multivariate statistical analysis (MVA) algorithms for online monitoring of batch processes [4, 5]. However, these MVA techniques have significant practical limitations in that they require estimates of future data and treat the overall batch as a single phase. To address the multiphase aspects of batch processes, Lu et al. identify phases within a batch by clustering time-series data with similar characteristic features before applying PCA on each cluster [6]. However, this method is deficient when continuous transition patterns are severe. To improve monitoring for transition patterns a soft-transition multiple PCA method and an angle-based separation scheme were proposed [7, 8]. The partition results require temporal sorting which is accomplished through subjective rules in additional post-processing steps [9].

To avoid phase rearrangement and guarantee temporal order of phases, a sequential phase segmentation approach was developed for multiphase batch processes in [9-12]. These methods sequentially compare the dissimilarity of adjacent time-slice matrices to find the transition points of phases. Unfortunately, an underlying assumption made for the phase segmentation methods is that all batches have phase uniformity, i.e., the order and number of phases within each batch are invariant. This assumption is not necessarily true as operations can vary from batch to batch, in particular for operations requiring manual operator intervention.

Multimode modeling addresses variations between batches by clustering similar batches and treating each cluster as a mode of operation with high repeatability. Recently, a support vector data approach was used to address the multimode and multiphase variations in batch processes [13]. Alternatively, a strategy of concurrent phase partitioning and inter-mode statistical analysis was proposed with the assumption of known mode labels [14]. The concurrent strategy was developed further and a concurrent mode identification and phase division (CMIPD) method was proposed [15]. If mode labels are unavailable, the CMIPD method uses the k-means clustering algorithm to identify the modes.

The phase partitioning and mode identification algorithms referred above use an important concept called *time-slice matrix* in order to model and monitor batch processes. Let us assume that there are $I \in \mathrm{R}$ batches each with $J \in \mathrm{R}$ process variables where each variable of each batch is measured at $k = 1, 2, ..., K$ time instances under the assumption of the fixed batch duration. Then the three dimensional array of batch process data can be represented by the matrix $\mathbf{X} \in \mathrm{R}^{I \times J \times K}$. An element from *i*th batch, *j*th process variable and *k*th time sample is denoted by $\mathbf{X}(i, j, k)$. The *time-slice matrix* is the submatrix $\mathbf{X}_k \in \mathrm{R}^{I\times J}$obtained by aggregating the values of all process variables from all batches at time instant *k,*

$$\mathbf{X}_k = \left\{ \mathbf{X}_k\left(i,j\right) \in \mathrm{R} : \mathbf{X}_k\left(i,j\right) = \mathbf{X}\left(i,j,k\right) \in \mathbf{X}\,\forall\, i,j \right\}. \tag{1}$$

Similarly, a *time-segment matrix* $\mathbf{X}_{k_1:k_2} \in \mathrm{R}^{(k_2-k_1+1)I\times J}$ is defined as the matrix obtained by concatenating the *time-slice matrices* from time instant k_1 to k_2.Mathematically, the time-segment matrix is

$$\mathbf{X}_{k_1:k_2} = \begin{bmatrix} \mathbf{X}_{k_1} & \mathbf{X}_{k_1+1} & \cdots & \mathbf{X}_{k_2} \end{bmatrix} \tag{2}$$

A visual description of these matrices is provided in Figure 2. The majority of phase and mode identification algorithms rely on MVA of the time-slice and time-segment matrices.

It is worth mentioning that the raw batches are actually of different batch durations. When it is claimed that batch process data can be represented by a standard three dimensional array $\mathbf{X} \in \mathrm{R}^{I\times J\times K}$, the underlying assumption is that all batches with different lengths (K_i in Figure 2) have been aligned or synchronized into the same length. Commonly used techniques include interpolation, truncation, defining an indicator variable for resampling process variables, dynamic time warping (DTW) and correlation optimization warping (COW) [1, 16-21]. The drawbacks of these batch length equalization methods are well documented [22]. Specifically, interpolation can introduce incorrect information while truncation can result in the loss of useful information. The DTW and COW methods cannot be used for online monitoring because they require the complete batch lengths beforehand. In some applications, it is even difficult to find a suitable indicator variable to align with the other process variables of equal length. Above all, under batch alignment, the columns of time-slice or time segment matrices correspond to equal values of the alignment variable rather than equal elapsed time. This would corrupt the original dynamic correlations and change the original data distribution. Hence, these strategies may work when the uneven lengths are not severe. Methods without "equalizing" batches as a preprocessing step and instead directly learning models from the raw batch data with severe uneven batch durations have attracted more attention. Zhao et al. took batches with approximate durations as one mode/group. They analyzed the differences between the groups to extract between-group and within-group information for monitoring faults. However, it is not necessarily true to simply assume the duration is a unique factor determining the group whose batches have similar features [22]. Zhang et al. address the drawback of the regular time-slice data matrix by proposing a pseudo-time-slice matrix constructed using the k nearest neighbor (kNN) rule to search similar samples within a data window [15]. However, the employment of kNN requires training data to be saved for online monitoring. These algorithms have inherent limitations as they are primarily meant for independent and identically distributed (IID) data. They do not account for temporal correlations of each variable.

Previous experience with monitoring continuous systems indicates that ignoring temporal correlations - and hence the process dynamics - will yield inaccurate statistical models and result in an unacceptably large number of missed process fault alarms [23].

To address the issue of dynamic modeling, dynamic PCA for online monitoring of batch processes was proposed [24]. Two-dimensional dynamic PCA and two-dimensional subspace identification were then developed to simultaneously consider the dynamics within a batch and across batches [25, 26]. Recently, a global linear state-space model that represents the evolution of states with subspace identification was developed for designing a model predictive controller (MPC) [27, 28]. However, these dynamic models are limited in their description of the multimode and multiphase behaviors. Particle filtering is another well-known estimation algorithm that has been extended to estimate the states of batch processes but it has not been applied for online monitoring of batch processes [29, 30].

To solve these drawbacks, a novel linear dynamic system based mode identification and phase division (LDSMIPD) strategy is proposed in this article to deal with the issue of dynamic modeling and online monitoring with uneven length multimode and multiphase batch processes. A linear dynamic system (LDS) [31-33] is used to model each phase and a dissimilarity measure is designed based on LDS to partition different phases in a batch. LDS is a state-space model describing process dynamics and it can naturally deal with uneven-length data because LDS is a parametric model unlike MVA. A batch distance metric is then constructed to cluster similar batches into the same mode. The batch distance comprehensively takes phase features into consideration and can represent the difference in dynamics of two batches. The k-means clustering algorithm is employed to cluster the mode. Once the mode is determined, a modified expectation maximization (EM) algorithm is applied on data from all batches in a given mode to estimate models for the corresponding phases. The difference between these phase models and the LDS models identified for phase partitioning is that the phase models are trained with multibatch data in an identical mode and can thus represent a wider range of operation than the model determined from a single batch of data. Finally, monitoring strategies are designed for multimode and multiphase batch process fault detection according to corresponding mode information and phase information. In the following section the proposed LDS phase partitioning strategy is described. Afterwards, the techniques employed for clustering batches into modes and re-identifying the phase models are described in detail. A new online monitoring algorithm is then developed followed by a case study to validate our approach. Finally, some conclusions and recommendations are provided.

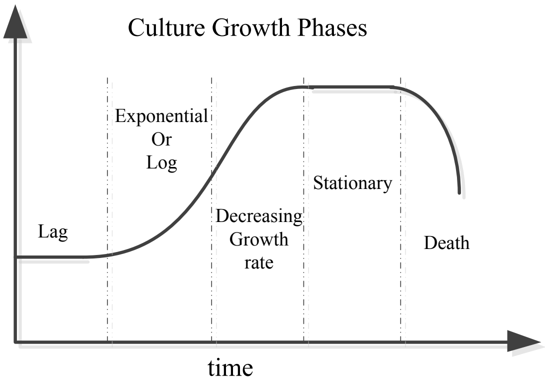


Figure 1: Illustration of culture growth phase in penicillin fermentation with four phases.

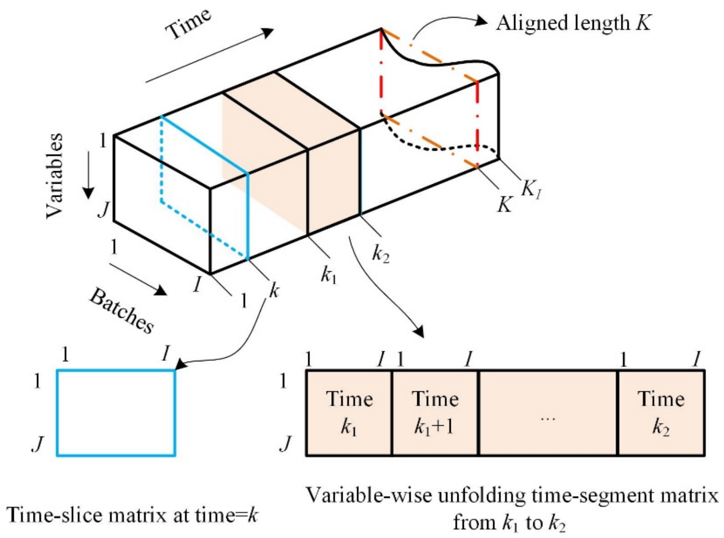


Figure 2. A visual depiction of *time-slice matrix* and *time-segment matrix*.

2. Sequential dynamic phase division

A standard linear dynamic system is formulated as follows:

$$\begin{aligned}\mathbf{t}(k+1) &= A\mathbf{t}(k)+\mathbf{w}(k)\\ \mathbf{x}(k) &= C\mathbf{t}(k)+\mathbf{v}(k)\end{aligned} \tag{3}$$

where $\mathbf{t}(k)\in \mathrm{R}^n$ represents the vector of latent variables governing the process dynamics at time *k*, $\mathbf{x}(k)\in \mathrm{R}^J$ denotes the vector of process variables also at time *k*. $\mathbf{w}(k)\in \mathrm{R}^n$ and $\mathbf{v}(k)\in \mathrm{R}^J$ are the process noises and measurement noises, respectively. The noise sequences are assumed to be normally distributed with zero mean such that $p(\mathbf{w}(k))=\mathrm{N}\ (0,Q)$ and. The initial state $\mathbf{t}(0)$ is also assumed to be normally distributed, i.e. $p(\mathbf{t}(0))=\mathrm{N}\ (\boldsymbol{\mu}(0),V(0))$. The unknown parameters of the LDS are combined into the parameter set $\Theta=\{\boldsymbol{\mu}(0),V(0),A,C,Q,R\}$ which are estimated with the EM algorithm [31, 34]. A stochastic subspace identification method is used to determine the choice of the hyper-parameters including the order of latent variables and the initial values of the iterative EM optimization[35].

Consider that for each batch, $i = 1,2,...,I$, there are J measured process variables, $\mathbf{x}^i\left(k\right) \in \mathrm{R}^J$ at time instances $k = 1,2,...,K_i$ where K_i denotes the duration of the corresponding batch. The data from each batch is represented using the set $X^i = \left\{\mathbf{x}^i\left(k\right)\right\}$, $k = 1,2,...,K_i$. The proposed sequential dynamic phase division with LDS is based on the following observations.

- LDS is a model that considers each batch of data as time series and as such it is not affected by the uneven batch durations unlike the standard MVA techniques which require extra cost and compromise to deal with the uneven batch duration problem.
- LDS inherently has the capability to account for steady-state as well as dynamic features of the batch process unlike the standard MVA techniques.

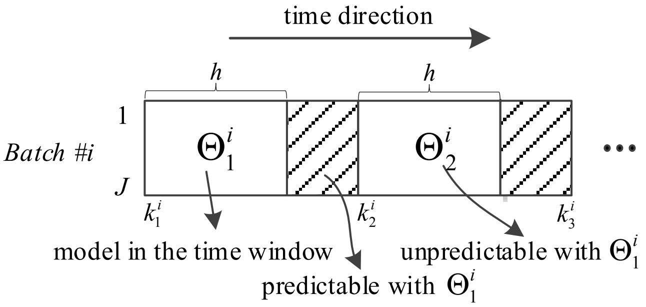


Figure 3. Illustration of phase segmentation

The phase segmentation process is applied on each batch data set. For each batch, the LDS corresponding to the first phase is identified by using an initial acceptable window of time. The data are deemed to be from a different phase if their predictability using the current model is unsatisfactory. As shown in Figure 3, samples that are predicted accurately by the trained model belong to this specific phase, which is identified by the "white" time window. In contrast, the division between a predictable and an unpredictable sample serves as the starting point for the next phase.

Based on the illustration in Figure 3, the procedure for the dynamic phase division is as follows:

Step1. *Initialization*. In the *i*-th batch the samples in a time window of length *h* used to identify the *p*-th phase are denoted as $X_p^i = \{\mathbf{x}_p^i(k)\}$, $k = 1, 2, \cdots, h$, where $\mathbf{x}_p^i(k) = \mathbf{x}^i(k_p^i + k)$ and k_p^i is the starting point of the p-th phase in the *i*-th batch run. The dataset X_p^i is pre-processed to have zero mean and unit standard deviation, i.e., $\bar{X}_p^i = \{\bar{\mathbf{x}}_p^i(k)\}$, $k = 1, 2, \cdots, h$.

Step 2. *Model training*. Train the LDS based on the *i*-th single batch data $\bar{X}_p^i$, given by

$$\begin{aligned} &\mathbf{t}_p^i(k+1) = A_p^i \mathbf{t}_p^i(k) + \mathbf{w}_p^i(k) \\ &\bar{\mathbf{x}}_p^i(k) = C_p^i \mathbf{t}_p^i(k) + \mathbf{v}_p^i(k) \\ &p(\mathbf{t}_p^i(0)) = N(\boldsymbol{\mu}_p^i(0), V_p^i(0)) \end{aligned} \quad . \tag{4}$$

Correspondingly, the set of parameters estimated in this case is given by

$$\Theta_p^i = \{\boldsymbol{\mu}_p^i(0), V_p^i(0), A_p^i, C_p^i, Q_p^i, R_p^i\}.$$

Step 3. *Phase extension*. A new phase is determined by detecting the time at which the model trained in the previous step is no longer able to adapt to samples, i.e., after $\bar{\mathbf{x}}_p^i(h)$. Generally, the distance between the true values, $\bar{\mathbf{x}}_p^i(k), k = h+1, h+2, \ldots$ and the predicted values $\hat{\mathbf{x}}_p^i(k), k = h+1, h+2, \ldots$ is used to determine whether the subsequent test samples correspond with the model. The details are as follows:

Step 3.1. *Distance measurement for the next sample*: The one-step ahead predicted marginal distribution associated with $\hat{\mathbf{x}}_p^i(k)$ and k is initialized as $h+1$, denoted

$$p\left(\hat{\mathbf{x}}_p^i(k)\right) = \mathrm{N}\left(\mathbf{z}_p^i(k), \Sigma_p^i(k)\right). \tag{5}$$

Given Eq.(4), the predicted mean and covariance matrix are respectively expressed as

$$\mathbf{z}_p^i(k) = C_p^i A_p^i \boldsymbol{\mu}_p^i(k-1) \tag{6}$$

and

$$\Sigma_p^i(k) = C_p^i \Gamma_p^i(k-1)\left(C_p^i\right)^T + R_p^i \tag{7}$$

where $\Gamma_p^i(k-1) = A_p^i V_p^i(k-1)\left(A_p^i\right)^T + Q_p^i$. Given that the predicted values of process variables at $k = h+1$ follow a multivariate Gaussian distribution, the squared Mahalanobis distance of the true values $\overline{\mathbf{x}}_p^i(k)$ from the predicted means can be given by

$$D_p^i(k) = \left(\overline{\mathbf{x}}_p^i(k) - \mathbf{z}_p^i(k)\right)^T \mathbf{\Sigma}_p^i(k)^{-1}\left(\overline{\mathbf{x}}_p^i(k) - \mathbf{z}_p^i(k)\right) \sim \chi^2(J) \tag{8}$$

Here $D_p^i(k)$ is the squared Mahalanobis distance and can be regarded as a sum of several unit Gaussians. Thus, $D_p^i(k)$ is subject to a χ^2distribution with J degrees of freedom. The control limit is determined by the χ^2 distribution with the confidence level α, denoted D_α. The implication of $D_p^i(k) > D_\alpha$is that the current model is not well-suited to describe the data. Note that the covariance matrix $\Sigma_p^i(k)$ may be ill-conditioned due to the linear relation among variables. The singular value decomposition (SVD) can be used to calculate pseudo-inverse by eliminating the effect of very small singular values. Therefore, the degrees of freedom of the χ^2 distribution and the control limit should be determined by the number of the remaining singular values, i.e., the values fairly larger than zero.

Step 3.2. *Updating the one-step-ahead predicted distribution forward:* By employing Kalman filtering, the non-steady state Kalman filter gain $K_p^i(k)$ can be obtained from Step 2 as

$$K_p^i(k) = \Gamma_p^i(k-1)\left(C_p^i\right)^T\left(C_p^i\Gamma_p^i(k-1)\left(C_p^i\right)^T + R_p^i\right)^{-1}. \tag{9}$$

Hence, the update rules of the means and the covariance matrix with respect to the latent states $\mathbf{t}_p^i(k)$ are given by

$$\boldsymbol{\mu}_p^i(k) = A_p^i\boldsymbol{\mu}_p^i(k-1) + K_p^i(k)\left(\overline{\mathbf{x}}_p^i(k) - C_p^iA_p^i\boldsymbol{\mu}_p^i(k-1)\right) \tag{10}$$

and

$$V_p^i(k) = \Gamma_p^i(k-1) - K_p^i(k)C_p^i\Gamma_p^i(k-1). \tag{11}$$

Once $\boldsymbol{\mu}_p^i(k)$ and $V_p^i(k)$ are obtained they can be used to update the one-step ahead predicted distributions $p\left(\hat{\mathbf{x}}_p^i(k+1)\right) = \mathrm{N}\left(\mathbf{z}_p^i(k+1), \Sigma_p^i(k+1)\right)$ similar to the Eqs.(5)~(7).

Step 3.3. *Iterative distance evaluation*: The distance between the true and predicted samples is evaluated in an iterative fashion according to Steps 3.1 and 3.2. If three consecutive samples exceed their control limits then the iterative procedure is stopped. The first sample exceeding the control limit is regarded as the starting point of the next phase and the entire procedure restarts at Step 1 to determine the next phase.

The phase division procedure is illustrated in Figure 4 with each color denoting a different phase. Note that the number of phases per batch is not necessarily constant. For example, the phase number of the first batch in Figure 4 is more than that of the second batch. Moreover, the length of each phase can differ from batch to batch even if the batches have the same number of phases. In Figure 4 the length of the first phase in the second batch is longer than the first phase in the I-th batch. These phenomena can be understood from the practical perspective that to achieve adequate performance some batches require additional operations or extra time. For example, in fermentation processes the fermentation tank should be cooled to a preferred temperature by either adding cooling water or elongating the cooling duration which will either introduce additional operations or prolong the cooling stage. The proposed sequential phase division is therefore better at adapting to the complexities of practical applications than techniques that are constrained by assumptions of equal phase length or identical phase evolution.

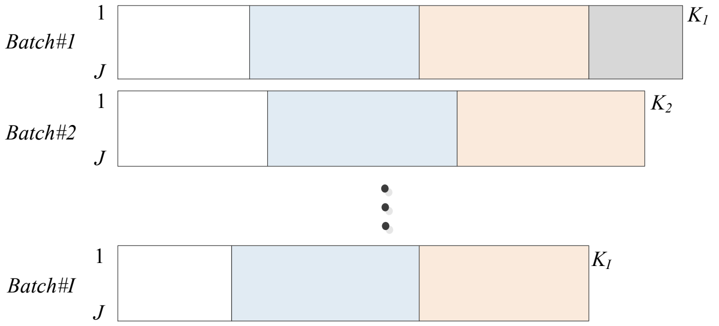


Figure 4. Illustration of phase division in different batches

3. Mode clustering and phase re-identification

The multiphase aspect of modeling describes the different operational stages along the time direction whereas the multimode aspect of modeling accounts for the operational variations between batches, i.e., batch-wise dissimilarity. Batches within a mode are defined to have a close global characteristic (i.e., batch distance), which compares batches over their entire respective duration. In the remainder of this section the mode clustering and phase parameter re-identification strategies are described. For mode clustering a batch distance index is designed to describe the difference between batches. Then, the k-means algorithm clusters the various batches into modes based on the designed batch distance [36]. Finally, based on the clusters the phase parameters are re-identified to homogenize the phase parameters for each mode.

3.1. Batch distance and mode clustering

It is clear that batches containing a different total number of phases are particularly dissimilar and should be classified into different modes. For batches with a different total number of phases, the batch distance can be regarded as infinite. However, it is not always correct to assume that batches with an equal number of phases are similar. Parameters of phases in the same order may vary significantly between batches even though the total number of phases is equivalent. Therefore, for mode clustering the definition of batch distance is suitable for batches with an equal number of phases.

The batch distance between the *i*-th batch and the *j*-th batch, denoted as $d^{i,j}$, is determined by the distances of corresponding phases in the same order, denoted as $d_p^{i,j}, p=1,2,...,P^i$ where P^i is the phase number of each batch. An illustration of the procedure for determining batch distance from the distances of corresponding phases is provided in Figure 5. It is clear from Figure 5 that any large variation between parallel phases will result in an obvious difference between batches. Therefore, the batch distance is determined by the maximum of the phase distances, i.e.,

$$d^{i,j} = \max_{p} d_p^{i,j}, p = 1, 2, \cdots, P^i. \tag{12}$$

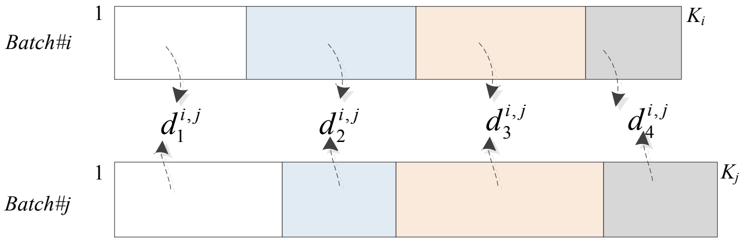


Figure 5. Determination of batch distance from phase distances.

Determination of the phase distance is based on the distribution of the phase data which is represented by the LDS after phase division. Two LDSs are considered equivalent when the following conditions are satisfied:

$$A_p^j = T^{-1} A_p^i T, \tag{13}$$

$$C_p^j = C_p^i T, \tag{14}$$

and

$$Q_p^j = T^{-1} Q_p^i \left(T^{-1}\right)^T, \tag{15}$$

where T is some nonsingular transformation matrix. Note that these conditions focus on the equivalence of process dynamics which are in part governed by the process noises (of covariance Q_p^i) and the inner parameters($\left(A_p^i, C_p^i\right)$) that describe the dynamic correlations of the process. Since each Q_p^i is a real symmetric matrix it can be written as $Q_p^i = B_p^i \left(B_p^i\right)^T$ and thus Eq.(15) can be reduced to

$$B_p^j = T^{-1} B_p^i. \tag{16}$$

An equivalent transformation of the two phases exists if one can find a nonsingular matrix T satisfying Eqs.(13), (14) and (16). The matrix T is determined by combining Eqs.(13) and (14) such that

$$\begin{pmatrix} C_p^j \\ C_p^j A_p^j \\ C_p^j \left(A_p^j\right)^2 \\ \vdots \end{pmatrix} = \begin{pmatrix} C_p^i \\ C_p^i A_p^i \\ C_p^i \left(A_p^i\right)^2 \\ \vdots \end{pmatrix} T. \tag{17}$$

The extended coefficient matrix of T can be truncated when they are full column rank. The estimate of T derived from Eq.(17) is denoted T_1 whereas the estimate of T derived by combining Eqs.(13) and (16) such that

$$\begin{pmatrix} B_p^j & A_p^j B_p^j & \left(A_p^j\right)^2 B_p^j & \cdots \end{pmatrix} = T^{-1} \begin{pmatrix} B_p^i & A_p^i B_p^i & \left(A_p^i\right)^2 B_p^i & \cdots \end{pmatrix} \tag{18}$$

is denoted T_2. In Eq.(18) the extended coefficient matrix of T^{-1} can be truncated when they are all full row rank. The phase distance is determined by comparing the estimate of T_1 to the estimate of T_2, i.e., if they are close the phases are considered equivalent. The distance between T_1 and T_2 is quantified by analyzing the distinction between eigenvalues of the two matrices via aligning the two matrices with the same eigenvectors [37]. The procedure for determining the phase distance is summarized as follows:

- Calculate $Z_1 = T_1 T_1^T$ and $Z_2 = T_2 T_2^T$ and let $Z = Z_1 + Z_2$.
- Decompose Z such that $Z = \Phi\Phi^T$.

- Calculate the eigenvalues $(\lambda_1, \lambda_2, ..., \lambda_n)$ of $\Phi^{-1} Z_1 \left(\Phi^{-1}\right)^T$.

The dissimilarity of T_1 and T_2 is then evaluated as follows

$$d(T_1, T_2) = \frac{4}{n} \sum_{z=1}^{n} (\lambda_z - 0.5)^2 \tag{19}$$

where $d(T_1, T_2)$ ranges from zero to one with a smaller value indicating more similarity between T_1 and T_2. Considering that $d(T_1, T_2)$ is a measurement of the distance between two LDSs, both of which have had their corresponding means (i.e., the set-points) removed. The phase distance should be determined by simultaneously considering the dissimilarity of dynamics and the difference between the set-points. Therefore, the phase distance $d_p^{i,j}$ is given by

$$d_p^{i,j} = d(T_1, T_2) + \beta \left\| \mathbf{m}_p^i - \mathbf{m}_p^j \right\|_2 \tag{20}$$

where $\mathbf{m}_p^i$ and $\mathbf{m}_p^j$ are sample averages of the p-th phase in i-th batch and j-th batch, respectively. The parameter β regulates the weighting between the two terms of Eq.(20). In this paper an adaptive β is selected, i.e.,

$$1 / \max_{i,j} \left\| \mathbf{m}_p^i - \mathbf{m}_p^j \right\|_2 \tag{21}$$

such that both terms in Eq.(20) range from zero to one. From the definition of phase distance it is clear that a value of $d_p^{i,j}$ close to zero indicates the two phases are sufficiently similar. Upon deriving the phase distance, the batch distance can be determined with Eq.(12) and used by the k-means clustering algorithm to cluster batches into several operating modes.

3.2. Phase parameter re-identification

During the initial phase division procedure each model illustrating one specific phase is trained with data from a single batch. After mode clustering is accomplished, similar batches are classified into the same mode. From the definition of batch distance, each batch within a specific mode has the same number of phases in the same order. Therefore, for each mode there exists data from several batches that is available for estimating phase parameters. Phase parameter re-identification with data from multiple batches help refine model parameters and it can also help the re-identified model represent a wider range of external conditions since each batch is influenced by components like the distributions of initial latent states and the measurement noises.

The sequence of data contained in the p-th phase of the i-th batch within the m-th mode is denoted as $X_p^{m,i} = \{\mathbf{x}_p^{m,i}(k)\}$, with $k = 1, 2, \cdots, L_p^{m,i}$ where $L_p^{m,i}$ is the corresponding sequence length. Assuming there are I_m batches in the m-th mode, the multi-batch data used to re-identify the parameters for the p-th phase is denoted $\mathbf{X}_p^m = \{X_p^{m,1}, X_p^{m,2}, \cdots, X_p^{m,I_m}\}$.The multi-batch data after pre-processing to ensure zero mean and unit standard deviation for each variable is denoted $\bar{\mathbf{X}}_p^m = \{\bar{X}_p^{m,1}, \bar{X}_p^{m,2}, \cdots, \bar{X}_p^{m,I_m}\}$. Please note that unlike in the phase division procedure, the sample means and variances are calculated by the overall mode dataset, $\mathbf{X}_p^m$, instead of using a single batch.

The EM algorithm is used to estimate the parameter set

$\Theta_p^m = \{\boldsymbol{\mu}_p^m(0), V_p^m(0), A_p^m, C_p^m, Q_p^m, R_p^m\}$. The parameter set governs the following LDS for a specific phase in a mode:

$$\begin{aligned}
&\mathbf{t}_p^{m,i}(k+1) = A_p^m \mathbf{t}_p^{m,i}(k) + \mathbf{w}_p^m(k) \\
&\bar{\mathbf{x}}_p^{m,i}(k) = C_p^m \mathbf{t}_p^{m,i}(k) + \mathbf{v}_p^m(k) \\
&p\left(\mathbf{t}_p^{m,i}(0)\right) = \mathrm{N}\left(\boldsymbol{\mu}_p^m(0), V_p^m(0)\right)
\end{aligned} \quad . \tag{22}$$

The latent variables corresponding to the process variables $\bar{\mathbf{X}}_p^m$ are dentoted $\mathbf{T}_p^m$. The log likelihood function of the complete data is given by $\ln\left(p\left(\bar{\mathbf{X}}_p^m, \mathbf{T}_p^m | \Theta_p^m\right)\right)$. Generally, each batch can be considered as an independent process with little influence from or on other batches. Therefore, the log likelihood function over the mode can be further divided into the sum of the log likelihood functions for each batch, yet governed by the same parameters, i.e.,

$$\ln\left(p\left(\bar{\mathbf{X}}_p^m, \mathbf{T}_p^m | \Theta_p^m\right)\right) = \sum_{i=1}^{I_m} \ln\left(p\left(\bar{X}_p^{m,i}, T_p^{m,i}\right) | \Theta_p^m\right). \tag{23}$$

With Eq.(22), Eq.(23) is further divided as follows

$$\begin{aligned}
&\ln\left(p\left(\bar{\mathbf{X}}_p^m, \mathbf{T}_p^m | \Theta_p^m\right)\right) = \sum_{i=1}^{I_m} \sum_{k=1}^{L_p^{m,i}} \ln\left(p\left(\bar{\mathbf{x}}_p^{m,i}(k) | \mathbf{t}_p^{m,i}(k), C_p^m, R_p^m\right)\right) \\
&+ \sum_{i=1}^{I_m} \sum_{k=1}^{L_p^{m,i}} \ln\left(p\left(\mathbf{t}_p^{m,i}(k) | \mathbf{t}_p^{m,i}(k-1), A_p^m, Q_p^m\right)\right) \\
&+ \sum_{i=1}^{I_m} \ln\left(p\left(\mathbf{t}_p^{m,i}(0) | \boldsymbol{\mu}_p^m(0), V_p^m(0)\right)\right)
\end{aligned} \quad . \tag{24}$$

The EM algorithm is a two-stage iterative optimization technique consisting of the E-step and the M-step to determine the maximum likelihood solution.

In the E-step, with the estimated parameters Θ_p^m from the last iterative cycle, one can obtain the expectation of the log likelihood function from Eq.(24) associated with the posterior distribution of latent variables $p\left(\mathbf{T}_p^m | \bar{\mathbf{X}}_p^m, \Theta_p^m\right)$. Since batches in the same mode are independent and identically distributed, the E-step can be applied to each individual batch. To estimate the expectation, the posterior distributions of latent variables, $p\left(\mathbf{T}_p^m \mid \bar{\mathbf{X}}_p^m, \Theta_p^m\right)$, are derived by Kalman filtering and Kalman smoothing, which are forward recursion and backward recursion, respectively. The posterior filtering distributions for each batch are given by

$$\begin{aligned}
&p\left(\mathbf{t}_p^{m,i}(k) \mid \bar{\mathbf{x}}_p^{m,i}(0), \cdots \bar{\mathbf{x}}_p^i(k), \Theta_p^m\right) = N\left(\boldsymbol{\mu}_p^{m,i}(k), V_p^m(k)\right) \\
&\boldsymbol{\mu}_p^{m,i}(k) = A_p^m \boldsymbol{\mu}_p^{m,i}(k-1) + K_p^m(k)\left(\bar{\mathbf{x}}_p^{m,i}(k) - C_p^m A_p^m \boldsymbol{\mu}_p^{m,i}(k-1)\right) \\
&V_p^m(k) = \Gamma_p^m(k-1) - K_p^m(k) C_p^m \Gamma_p^m(k-1)
\end{aligned} \tag{25}$$

where $\Gamma_p^m(k-1) = A_p^m V_p^m(k-1)\left(A_p^m\right)^T + Q_p^m$.The Kalman filtering gain, $K_p^m(k)$, for the p-th phase and the m-th mode is given by

$$K_p^m(k) = \Gamma_p^m(k-1)\left(C_p^m\right)^T \left(C_p^m \Gamma_p^m(k-1)\left(C_p^m\right)^T + R_p^m\right)^{-1}. \tag{26}$$

Initial values of the forward recursion in Eq.(25) are $\boldsymbol{\mu}_p^m(0)$and $V_p^m(0)$. The posterior smoothing distribution associated with the latent variables of each batch is given by

$$\begin{aligned}
&p\left(\mathbf{t}_p^{m,i}(k) \mid \bar{X}_p^i, \Theta_p^m\right) = \mathrm{N}\left(\hat{\boldsymbol{\mu}}_p^{m,i}(k), \hat{V}_p^m(k)\right) \\
&\hat{\boldsymbol{\mu}}_p^{m,i}(k) = \boldsymbol{\mu}_p^{m,i}(k) + J_p^m(k)\left(\hat{\boldsymbol{\mu}}_p^{m,i}(k+1) - A_p^m \boldsymbol{\mu}_p^i(k)\right) \\
&\hat{V}_p^m(k) = V_p^m(k) + J_p^m(k)\left(\hat{V}_p^m(k+1) - \Gamma_p^m(k)\right) J_p^m(k)^T
\end{aligned} \tag{27}$$

where $J_p^m(k) = V_p^m(k)\left(A_p^m\right)^T \Gamma_p^m(k)^{-1}$. Initial values of this backward recursion for each batch are $\hat{\boldsymbol{\mu}}_p^{m,i}\left(L_p^{m,i}\right) = \boldsymbol{\mu}_p^{m,i}\left(L_p^{m,i}\right)$ and $\hat{V}_p^m\left(L_p^{m,i}\right) = V_p^m\left(L_p^{m,i}\right)$. Based on the posterior smoothing distributions, take the expectation of the log likelihood function in Eq.(24) results in the expectations of the following three statistics

$$E\left(\mathbf{t}_p^{m,i}(k)\right) = \hat{\boldsymbol{\mu}}_p^{m,i}(k) \tag{28}$$

$$E\left(\mathbf{t}_p^{m,i}(k)\mathbf{t}_p^{m,i}(k)^T\right)=\hat{\boldsymbol{\mu}}_p^{m,i}(k)\hat{\boldsymbol{\mu}}_p^{m,i}(k)^T+\hat{V}_p^m(k) \tag{29}$$

$$E\left(\mathbf{t}_p^{m,i}(k)\mathbf{t}_p^{m,i}(k-1)^T\right)=\hat{V}_p^m(k)J_p^m(k-1)^T+\hat{\boldsymbol{\mu}}_p^{m,i}(k)\hat{\boldsymbol{\mu}}_p^{m,i}(k-1)^T \tag{30}$$

where E is the expectation operator.

In the M-step, the expectation of the log likelihood function associated with the posterior distribution is maximized based on the results derived in the E-step. Unlike the E-step, the independent batches for each mode in the training data should be incorporated in the M-step to maximize the objective likelihood function over all batches in the same mode. The update procedure for the parameter set denoted $\Theta_p^{m,*}$in the M-step is given by the following set of equations:

$$\boldsymbol{\mu}_p^{m,*}(0)=\frac{1}{I_m}\sum_{i=0}^{I_m}E\left(\mathbf{t}_p^{m,i}(0)\right) \tag{31}$$

$$V_p^{m,*}(0)=\frac{1}{I_m}\sum_{i=1}^{I_m}E\left(\mathbf{t}_p^{m,i}(0)\mathbf{t}_p^{m,i}(0)^T\right)-\boldsymbol{\mu}_p^{m,*}(0)\boldsymbol{\mu}_p^{m,*}(0)^T \tag{32}$$

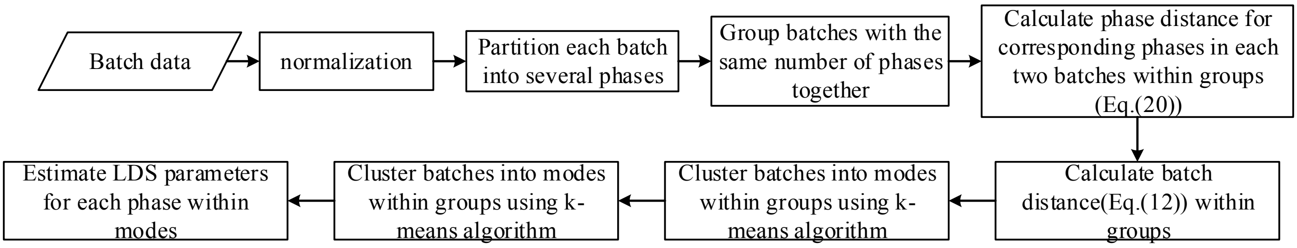
$$A_p^{m,*}=\left(\sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}}E\left(\mathbf{t}_p^{m,i}(k)\mathbf{t}_p^{m,i}(k-1)^T\right)\right)\left(\sum_{i=1}^{I_m}\sum_{k=0}^{L_p^{m,i}-1}E\left(\mathbf{t}_p^{m,i}(k)\mathbf{t}_p^{m,i}(k)^T\right)\right)^{-1} \tag{33}$$

$$\begin{aligned}Q_p^{m,*}=\frac{1}{\sum_i^{I_m}L_p^{m,i}}\sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}}\Big\{&E\left(\mathbf{t}_p^{m,i}(k)\mathbf{t}_p^{m,i}(k)^T\right)-A_p^{m,*}E\left(\mathbf{t}_p^{m,i}(k-1)\mathbf{t}_p^{m,i}(k)^T\right)\\ &-E\left(\mathbf{t}_p^{m,i}(k)\mathbf{t}_p^{m,i}(k-1)^T\right)\left(A_p^{m,*}\right)^T+A_p^{m,*}E\left(\mathbf{t}_p^{m,i}(k-1)\mathbf{t}_p^{m,i}(k-1)^T\right)\left(A_p^{m,*}\right)^T\Big\}\end{aligned} \tag{34}$$

$$C_p^{m,*}=\left(\sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}}\overline{\mathbf{x}}_p^{m,i}(k)E\left(\mathbf{t}_p^{m,i}(k)^T\right)\right)\left(\sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}}E\left(\mathbf{t}_p^{m,i}(k)\mathbf{t}_p^{m,i}(k)^T\right)\right)^{-1} \tag{35}$$

$$\begin{aligned}R_p^{m,*}=\frac{1}{\sum_i^{I_m}L_p^{m,i}}\sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}}\Big\{&\overline{\mathbf{x}}_p^{m,i}(k)\overline{\mathbf{x}}_p^{m,i}(k)^T-C_p^{m,*}E\left(\mathbf{t}_p^{m,i}(k)\right)\overline{\mathbf{x}}_p^{m,i}(k)^T\\ &-\overline{\mathbf{x}}_p^{m,i}(k)E\left(\mathbf{t}_p^{m,i}(k)^T\right)\left(C_p^{m,*}\right)^T+C_p^{m,*}E\left(\mathbf{t}_p^{m,i}(k)\mathbf{t}_p^{m,i}(k)^T\right)\left(C_p^{m,*}\right)^T\Big\}\end{aligned}\ . \tag{36}$$

Details for the above parameter updating formulas of the EM algorithm with multi-batch data are provided in Appendix A. The maximum likelihood estimate is obtained by conducting the E-step and the M-step iteratively until convergence is achieved. This optimization procedure does not require an equal length for each batch even though multiple batches for a given mode are considered together during phase parameter re-identification. The whole flow chart of process modeling described in Section 2 and Section 3 is shown in Figure 6.



Batch data
normalization
Partition each batch into several phases
Group batches with the same number of phases together
Calculate phase distance for corresponding phases in each two batches within groups (Eq.(20))
Calculate batch distance(Eq.(12)) within groups
Cluster batches into modes within groups using k-means algorithm
Cluster batches into modes within groups using k-means algorithm
Estimate LDS parameters for each phase within modes

Figure 6. The flow chart of process modeling.

4. Online monitoring

Once the phase division, mode clustering and phase re-identification procedures are completed, a set of structured dynamic models are obtained to describe the phase evolution in a clustered mode. One purpose of creating these dynamic models is to be able to perform online monitoring over the entire duration of new batches. For a specific phase in a mode, the methods for determining the monitoring statistic and the control limit are the same as the procedures for sequential phase division in Section 2. Firstly, with the p-th phase in the m-th mode, the statistic used to monitor the current sample is the squared Mahalanobis distance, $D_p^m\left(k\right)$constructed by the one-order ahead predicted distribution (Step 3.1 in Section 2). When $D_p^m\left(k\right)$ exceeds the control limit D_α with confidence α, the new point is considered to be abnormal. The parameters of the predicted distribution are updated by Kalman filtering when the new sample is obtained (Step 3.2 in Section 2). Note that the calculation of $D_p^m\left(k\right)$ in online monitoring requires parameters in phase re-identification to substitute the parameters in phase division.

There are two critical challenges in online monitoring for multimode and multiphase processes, i.e., mode selection and phase switching. Generally, the mode information is uncertain when a new batch begins since the sample size is too small to successfully determine the mode of the batch. In this work mode selection is performed with the voting principle. Firstly, the model in each mode is used to construct the Mahalanobis distance statistics, $D_p^m\left(k\right)$, $m=1,2,...,M$ where M is the mode number. The minimum statistic generates a vote for the corresponding mode. When the number of votes in one mode exceeds a predefined value v over other modes then the batch is considered to belong to the mode with the most votes. Subsequent samples are then monitored under the assumption that the batch belongs to this mode.

Before the mode is determined, the statistic is conventionally selected as the minimum Mahalanobis distance over all modes. To determine phase switching, the monitoring procedure starts at the first phase in each mode. The switch in phases is detected when the distance statistics from six continuous samples with the current phase model all exceed those calculated with the model from the subsequent phase. This phase switching condition indicates that the model for the next phase has a better prediction performance for the new samples than the model from the current phase.

5. Case studies

The Pensim benchmark for penicillin production in a fed-batch fermentor is applicable to a wide range of controller design and process monitoring challenges. Recently, an extensive reference data-set associated with the Pensim benchmark model has been developed for the purpose of process monitoring [38].Based on the existing Pensim model, this reference data-set carefully accounts for the following important factors: i) the variability across batches, ii) the type, magnitude and the onset time of a process upset and iii) the influence of measurement noise and process upsets on the process and the control system. Thus, the reference dataset is well suited to test and compare the efficacy of the proposed monitoring algorithm on multimode and multiphase batch processes. Specifically, the ten process variables that are listed in Table 1 are measured online and used to train the process models and perform online monitoring. Note that feed rate, one of the online measured variables, is discarded in this work because it starts at zero and tends to be a constant with a very small fluctuation, providing little dynamic information and potentially causing singularity problems in model building. The measurements are sampled every 0.2h and the running duration of a batch is approximately 460h with some variation due to uneven batch lengths.

5.1. Process modeling

The multimode reference data-set is generated by using two types of micro-organisms when producing the data collected over a total of 300 batches. The distribution of batch lengths is shown as a histogram in Figure 7. To provide some insight into the multimode and multiphase behavior, trends of the dissolved oxygen (DO) concentration of the batches are plotted in Figure 8. From Figure 8 it can be seen that the trends are split primarily into two clusters and the DO concentration is not static as a function of time. Early stages of each batch display complex transient behaviors whereas the later stages tend to be relatively steady.

Table 1. The description of process variables.

No.	Process variables
1	Fermentation volume
2	Dissolved oxygen concentration
3	Dissolved CO_2 concentration
4	Reactor temperature
5	pH
6	Feed temperature
7	Agitator power
8	Cooling/heating medium flow rate
9	Base flow rate
10	Acid flow rate

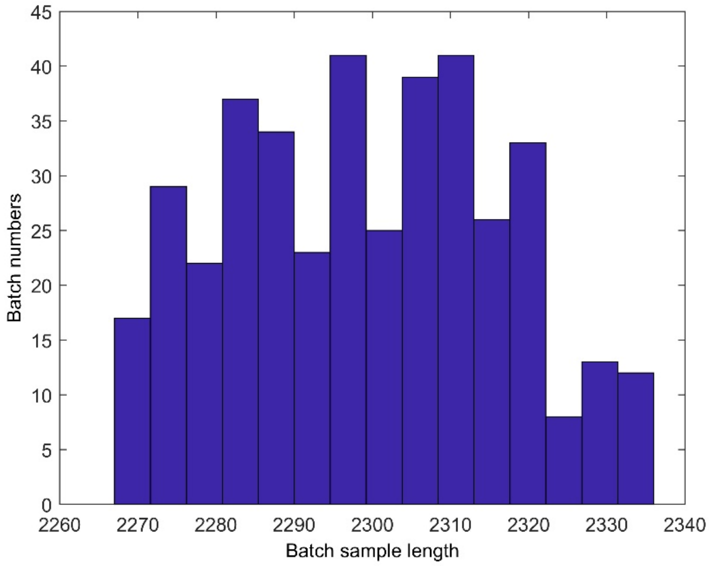
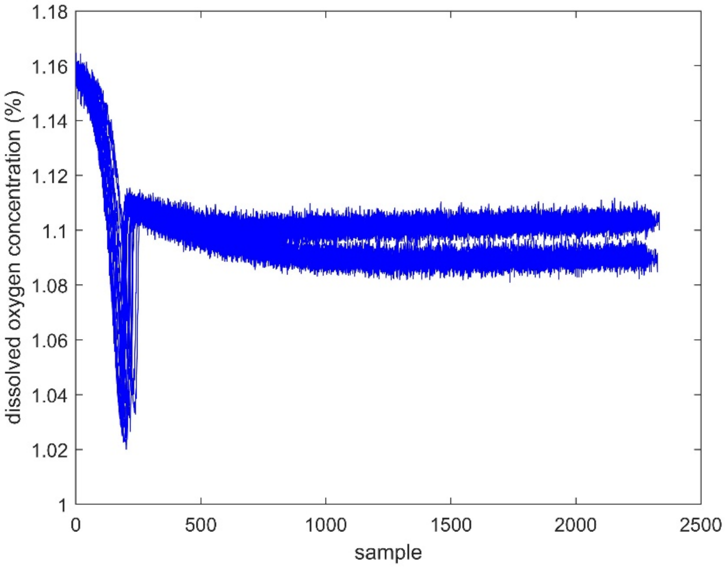


Figure 7. Distribution of batches according to the length of each batch in samples



8

Figure 8. The trends of dissolved oxygen concentration.

The phase division is performed with each batch and the length of the time window (*h*) to identify an LDS is set to be 100 samples. The specified window length is small enough to describe phase specifics and large enough to obtain a satisfactory estimate for the parameters of an LDS accordingly. As shown in Figure 9, after phase division the total number of phases per batch are primarily distributed between 8 phases and 13 phases. These normal batches can be initially classified into six modes temporarily according to their total number of phases. Figure 10 compares the duration of batches with a total of 8 phases (red circles) to the duration of batches with a total of 13 phases (blue stars). The increments in the horizontal axis in Figure 10 are distinct batches for a given total phase number. From Figure 10 it is clear that the total number of phases is somewhat independent of the batch duration, i.e., batches with a longer duration do not necessarily have more phases. Instead, the total number of phases is dependent upon the ability of a current phase to predict future samples, which ensures consistency of samples within the same phase.

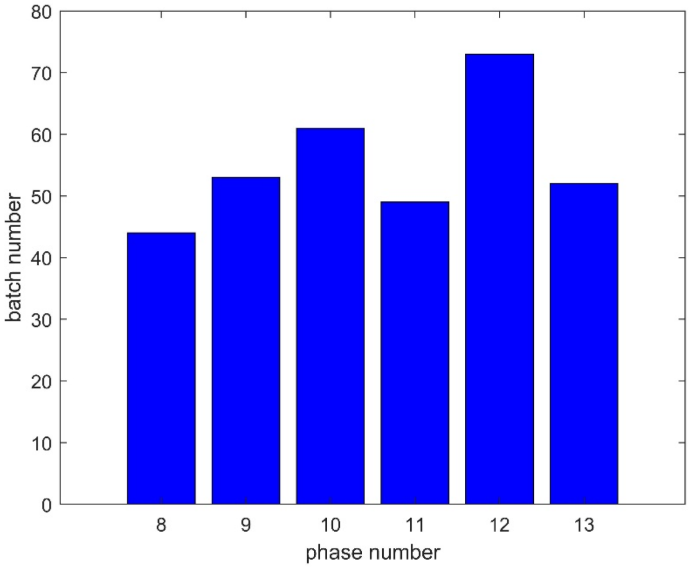


Figure 9. Distribution of batches according to total phase number

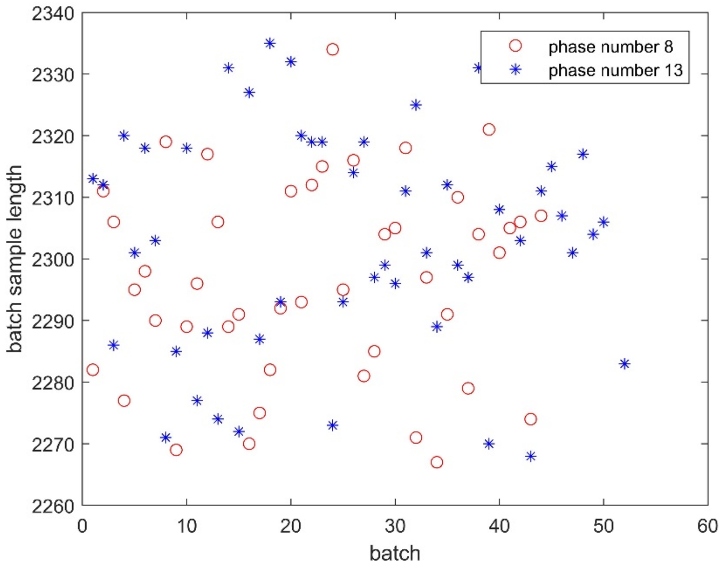


Figure 10. The lengths of batches with phase number 8 and 13.

Next, batches with the same phase number are clustered based on batch distance. Figure 11 shows the batch distances between the various batches with a total of 8 different phases. It can be seen that the defined batch distance clearly indicates two classes, which are clustered by the k-means algorithm. This clustering result is consistent with the condition that the training data-set is generated with two types of micro-organisms. For batches with more than 8 phases, the clustering results are similar to those in Figure 11, i.e., batches with the same phase number are separated into two clusters. Therefore, the entire training data-set is separated into 12 modes. Since clustering ensures similar properties of batches in a given mode, phase re-identification is performed for each phase utilizing more batches. The EM algorithm developed to deal with the multi-batch data is used to re-estimate the parameters for each phase. Figure 12 shows the trend of the log likelihood function for the first mode in batches with a total of 8 phases. Iteration results in a monotonous increase in the value of the log likelihood function until convergence is achieved.

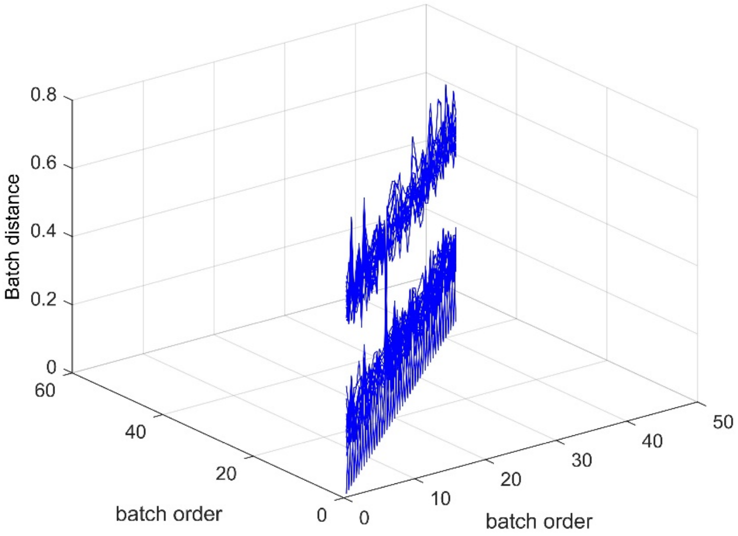


Figure 11. Batch distances between the various batches with a total of 8 different phases.

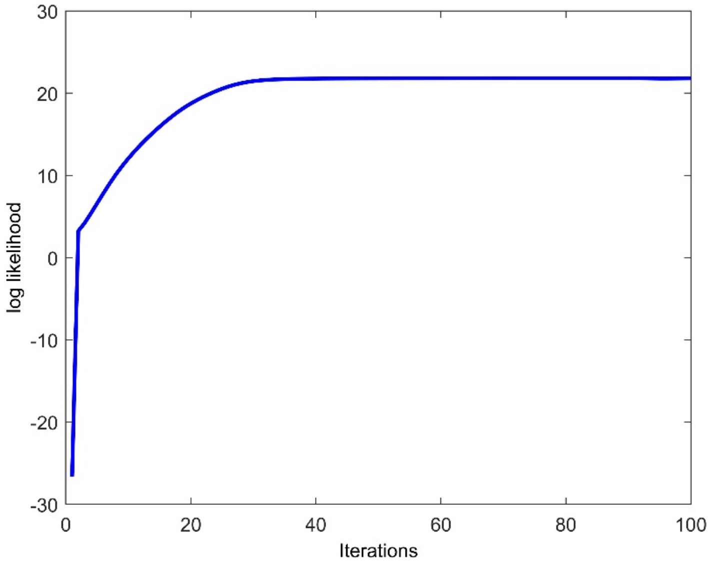


Figure 12. Log likelihood values as the iteration increases gradually.

5.2. Process monitoring

The structured LDSs obtained after phase division, mode clustering and phase re-identification are used to carry out the online monitoring when a new batch begins. The extensive reference data-set provides a broad range of faults with a variety of fault types, fault magnitudes and fault onset times. Specifically, there are 12 types of process faults and each type has several different fault magnitudes to evaluate the sensitivity of the monitoring methods. Moreover, each kind of fault with a specific magnitude has 200 different onset times when a process upset is triggered. Table 2 provides details of the various fault types and fault magnitudes. Taking the Fault 1 with a magnitude of -10% for example, the different onset time of this upset in 200 abnormal batches is plotted in Figure 13 which shows the fault onset time of the 200 faulty batches are uniformly distributed among four time ranges (i.e., 0-100h, 100h-200h, 200h-300h and 300h-400h). The onset time of other fault scenarios with a specific magnitude has an approximately equivalent distribution to that shown in Figure 13.

Table 2. Overview of fault types and magnitudes

No.	Fault descriptions and magnitudes
Fault 1	Sudden change in feed substrate concentration (%) -10, -5, -2, -1, -0.5, 0.5, 1, 2, 5, 10
Fault 2	Change in coolant temperature(°C) -2, -1, -0.5, -0.2, -0.1, 0.1, 0.2, 0.5, 1, 2
Fault 3	Gradual change of feed rate (saturating at 0.04/0.08L/h for negative/positive drifts) (%/h) -0.30, -0.15, -0.05, 0.05, 0.15, 0.30
Fault 4	Gradual dissolved oxygen sensor drift (saturating at 0.2/2 for

	negative/positive drifts) (%/h) -0.10, -0.05, -0.02, -0.01, -0.005, 0.005, 0.01, 0.02, 0.05, 0.10
Fault 5	pH sensor drift (saturating at 2) (/h) 0.001, 0.002, 0.003, 0.004, 0.005, 0.010, 0.015, 0.025
Fault 6	Non-functional pH control (no acid or base flow for indicated duration) (h) 0.5, 1, 2, 5, 10, 20
Fault 7	Reduced pH control (control action and maximal control action reduced by indicated fraction) (%) -10, -20, -40, -60, -80, -90
Fault 8	Reactor temperature sensor bias (°C) -0.50, -0.10, -0.05, 0.05, 0.10, 0.50
Fault 9	Reactor temperature sensor drift (saturating at -5/+5°C for negative/positive drifts) (°C/h) -0.10, -0.05, -0.01, -0.005, 0.005, 0.01, 0.05, 0.10
Fault 10	Reduced temperature control (control action and maximal control action reduced by indicated fraction) (%) -5, -10, -20, -30, -40, -50
Fault 11	Reduced temperature control (control action reduced by indicated fraction, maximal flow not impacted) (%) -10, -20, -30, -40, -50, -60
Fault 12	Contamination (drift of substrate with -0.05/h to indicated level) -0.05, -0.10, -0.15, -0.20, -0.25

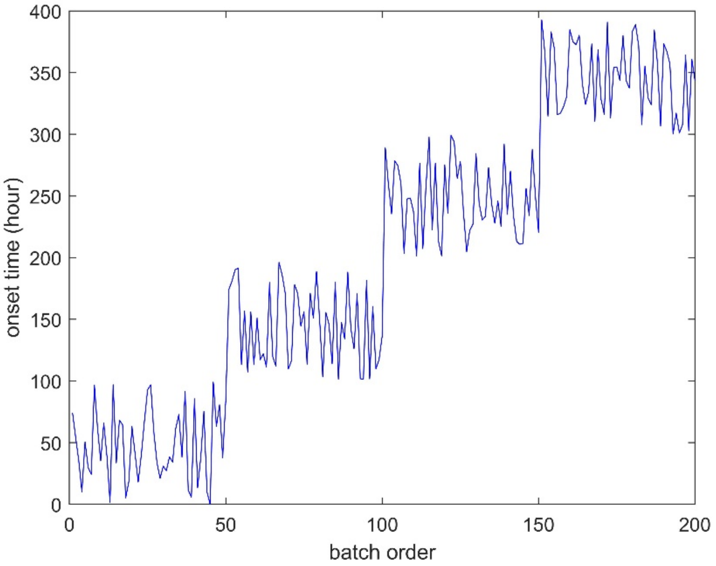


Figure 13. The abnormal onset time of these batches with respect to Fault 1 with magnitude -10%

In this paper, the newest state-of-the-art method, i.e., the CMIPD strategy based on time-slice matrices for monitoring multimode and multiphase batch processes with uneven durations is compared to the proposed LDSMIPD method. Note that the CMIPD is developed based on PCA so that the Hotelling T^2 and squared prediction error (SPE) statistics can be designed. However, these two statistics are asymmetric. The T^2 statistic can capture the main process fluctuations which still obey the process correlation whereas the SPE statistic can detect the abnormalities that deviate from the normal process variable correlations. The SPE statistic is sensitive and suited to detect process faults [39, 40]. Since this work aims to monitor abnormal process behavior, the SPE statistic in CMIPD is compared with the proposed method because the statistic designed in this paper is a counterpart of the SPE statistic in CMIPD. In this application, CMIPD has divided the batches into eight phases and each phase is clustered into modes according to differences in the direction of the principal component with respect to the time-slice matrix. The confidence level for setting the control limits in both the LDSMIPD and the CMIPD is set at 95%. Figure 14 compares the two methods for the training data (i.e., no faults). From Figure 14 it can be seen that both methods accurately monitor the majority of samples. The false alarm rates of CMIPD and LDSMIPD are 2.7% and 4.3%, respectively, both of which are less than 5% under the confidence level of 95%. However, the CMIPD as a static strategy presents a sharp peak at around the 250th sample where there exists a significant transient behavior according to Figure 8. By comparison, the dynamic model is smoother and provides a better fit of the transient stage.

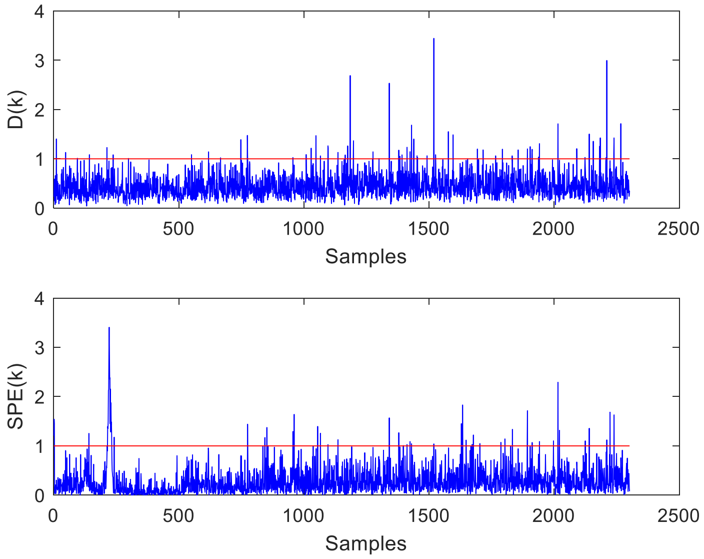


Figure 14. The monitoring for the normal case by two methods.

The performance for monitoring abnormal behavior is measured by fault detection rate (FDR), which is calculated by the ratio of the detected fault samples relative to the total number of fault samples. Using the second fault in Table 2 for example, the qualitative analysis of observing the monitoring curves of the statistics in Figures 14 to 19 is performed to intuitively demonstrate the performance of the proposed strategy. In Figures 14 to 19, the horizontal red line is the control limit for fault detection and the vertical red line is the true onset time of the fault. When these figures are plotted, the monitoring indices for all points are scaled by the control limit, i.e., $D_p^m\left(k\right)/\,D_\alpha$. Hence, the red line is always at 1 and correspondingly its log value is at 0. The normal case with five different magnitudes and onset times are tested with the second fault and the results are displayed in Figures 14 to 19. Together, these figures indicate that the two methods monitor the abnormal behavior well for a fault with a large absolute magnitude. However, the LDSMIPD displays a better performance in quickly capturing the incipient anomaly since the proposed method can react rapidly when the fault occurs. In comparison, the CMIPD method has an obvious detection delay. Similarly, Figure 16 and Figure 18 also indicate that the LDSMIPD has a higher FDR when the fault magnitude is medium. For a fault with a small magnitude such as the case in Figure 17, both methods ignore this small fault under the current confidence level. The benefit of the dynamic method is that the dynamic model is closer to the true distribution for time series than the distribution derived by the static method which assumes the data is independent and identically distributed.

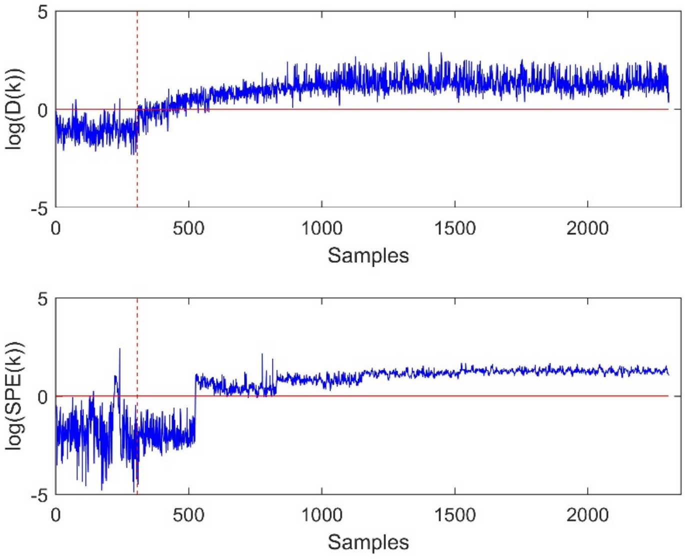


Figure 15. The comparison of LDSMIPD and CMIPD for Fault 2 with -2 magnitudes

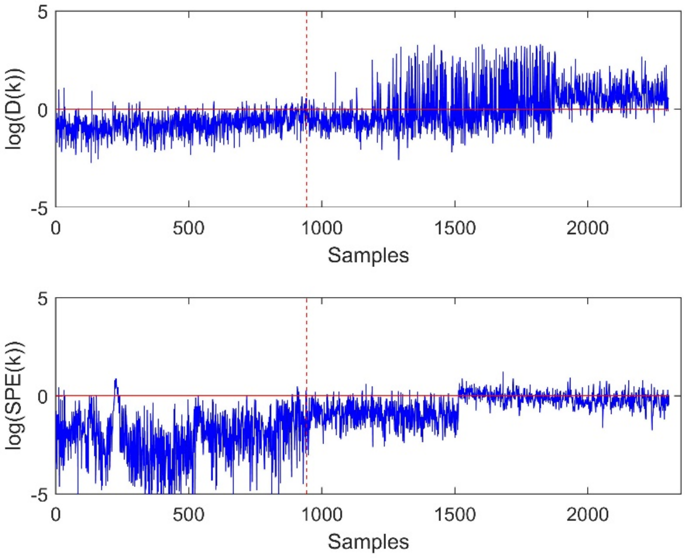


Figure 16. The comparison of LDSMIPD and CMIPD for Fault 2 with -0.5 magnitudes

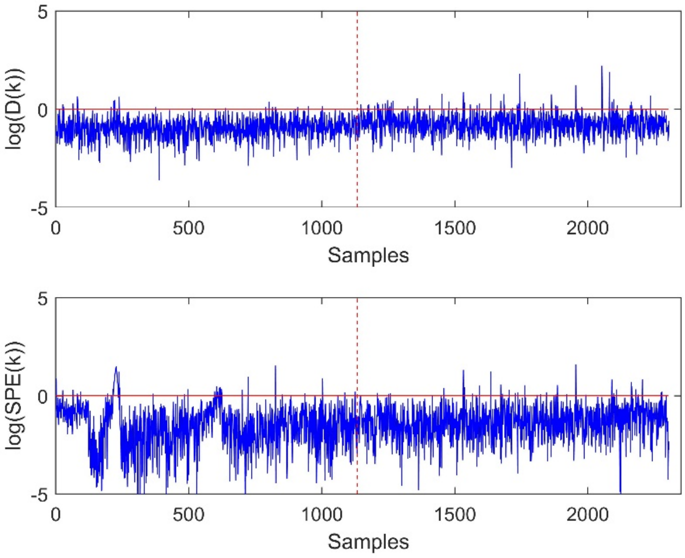


Figure 17. The comparison of LDSMIPD and CMIPD for Fault 2 with +0.1 magnitudes

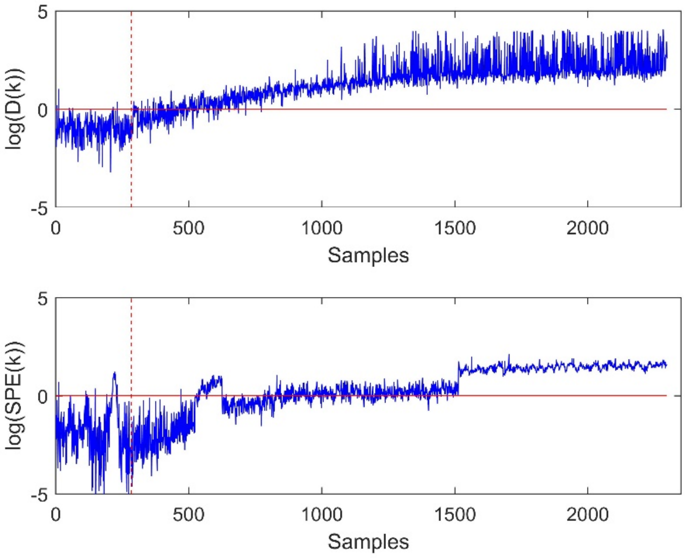


Figure 18. The comparison of LDSMIPD and CMIPD for Fault 2 with +1 magnitudes

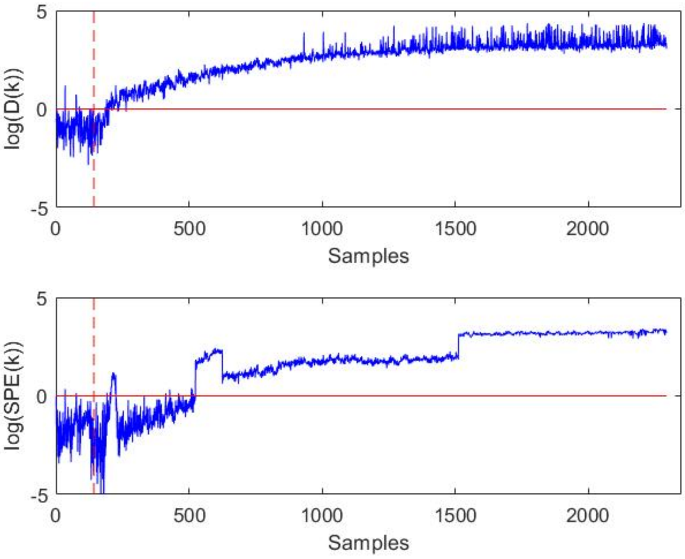


Figure 19. The comparison of LDSMIPD and CMIPD for Fault 2 with +2 magnitudes

The FDRs for all fault scenarios are quantitatively summarized in Table 3 to evaluate the performance of online monitoring. Table 3 gives the average of FDRs of 200 batches for a specific type of fault with a specific magnitude. The fault magnitudes in Table 3 for each fault are sequentially numbered as the order shown in Table 2. For example, the first fault contains ten different magnitudes so that they are numbered as 1-10 whereas the third fault has six different magnitudes which are numbered as 1-6 in Table 3. The numeric values in Table 3 are formatted such that the averaged FDR of LDSMIPD/CMIPD are presented for a convenient comparison with the bold cases indicating better performance for the LDSMIPD method. For example, 56/48 in Table 3 means the FDR of the first method is 56% while the FDR of the second method is 48%. Hence, Table 3 cannot only provide the detection rates of the two methods, it is also beneficial to give the intuition of the comparative performance of the two methods.

From this table, the overall monitoring performance is much better with the dynamic LDSMIPD technique than the static CMIPD method as evident by the higher FDR with LDSMIPD. Moreover, for the faults with very small magnitudes such as magnitudes 3-8 for the first fault, magnitudes 4-7 for the second fault and the first magnitude for the seventh fault, the FDR of CMIPD is almost still under 5%. Therefore, these scenarios will be misjudged as normal operation since the confidence level of the fault detection is set to be 95%. For some faults with medium or large magnitudes, LDSMIPD has a significant performance improvement such as the second magnitude of the fifth fault and magnitudes 3-5 of the seventh fault.

Table 3. The averaged FDRs for each type of fault with different magnitudes

(format: FDR of LDSMIPD/ FDR of CMIPD unit: %)

No. of Magnitude	1	2	3	4	5	6	7	8	9	10
Fault 1	**56/48**	**45/19**	**13/5**	**13/4**	**14/4**	**12/4**	**13/4**	**16/6**	**48/31**	67/80
Fault 2	94/95	59/74	**41/19**	**9/5**	**9/4**	**9/4**	**11/6**	**51/34**	85/90	**99/96**
Fault 3	62/64	**59/52**	**55/31**	**45/37**	56/63	67/77	--	--	--	--
Fault 4	**97/96**	**94/92**	86/86	74/75	**59/56**	**63/56**	73/75	85/86	**94/93**	**97/95**
Fault 5	**10/4**	**58/10**	**74/30**	**81/50**	**87/64**	**96/91**	**98/97**	99/99	--	--
Fault 6	**6/4**	**6/4**	**7/4**	**8/6**	9/10	**57/19**	--	--	--	--
Fault 7	**9/3**	**16/3**	**37/3**	**70/4**	**78/8**	**86/42**	--	--	--	--
Fault 8	**49/22**	**7/5**	**6/4**	**6/4**	**7/5**	**55/36**	--	--	--	--
Fault 9	**99/97**	**97/95**	**68/67**	**50/43**	**66/53**	**81/72**	**98/95**	**99/97**	--	--

Fault 10	**6/4**	**7/4**	**14/5**	**71/41**	**78/48**	**92/88**	--	--	--	--
Fault 11	**7/4**	**12/5**	**18/6**	**63/23**	**81/48**	**93/88**	--	--	--	--
Fault 12	**16/7**	20/22	**50/33**	**55/46**	**58/52**	--	--	--	--	--

6. Conclusions

In this article, an LDS is used as a basic model for the batch process. The LDS is selected for its ability to fit the distribution of the stochastic dynamic system in a hidden Markov fashion, so that the sequential LDSs are well-suited for modeling multiphase dynamic processes and monitoring the process deviations. Moreover, unlike other MVA techniques, the proposed modeling procedure does not require an equal batch length because the LDS is a structured parametric model. Additionally, the multimode behavior is determined by evaluating the distance between batches. The proposed method is applied to a standard fermentation benchmark and compared to a static model. The detailed analyses of the dynamic model are illustrated and the results indicate a significant performance improvement with regards to monitoring the fermentation batch process used as a benchmark.

References

[1]. Nomikos, P. and J.F. MacGregor, *Multivariate SPC charts for monitoring batch processes.* Technometrics, 1995. **37**(1): p. 41-59.

[2]. Undey, C. and A. Cinar, *Statistical monitoring of multistage, multiphase batch processes.* IEEE control systems, 2002. **22**(5): p. 40-52.

[3]. Yao, Y. and F. Gao, *A survey on multistage/multiphase statistical modeling methods for batch processes.* Annual Reviews in Control, 2009. **33**(2): p. 172-183.

[4]. Nomikos, P. and J.F. MacGregor, *Monitoring batch processes using multiway principal component analysis.* AIChE Journal, 1994. **40**(8): p. 1361-1375.

[5]. Nomikos, P. and J.F. MacGregor, *Multi-way partial least squares in monitoring batch processes.* Chemometrics and intelligent laboratory systems, 1995. **30**(1): p. 97-108.

[6]. Lu, N., F. Gao, and F. Wang, *Sub-PCA modeling and on-line monitoring strategy for batch processes.* AIChE Journal, 2004. **50**(1): p. 255-259.

[7]. Zhao, C., et al., *Stage-based soft-transition multiple PCA modeling and on-line monitoring strategy for batch processes.* Journal of Process Control, 2007. **17**(9): p. 728-741.

[8]. Yao, Y. and F. Gao, *Phase and transition based batch process modeling and online monitoring.* Journal of Process Control, 2009. **19**(5): p. 816-826.

[9]. Qin, Y., C. Zhao, and F. Gao, *An iterative two-step sequential phase partition (ITSPP) method for batch process modeling and online monitoring.* AIChE Journal, 2016. **62**(7): p. 2358-2373.

[10]. Zhao, C. and Y. Sun, *Step-wise sequential phase partition (SSPP) algorithm based statistical modeling and online process monitoring.* Chemometrics and Intelligent Laboratory Systems, 2013. **125**: p. 109-120.

[11]. Chen, J. and Y.-C. Jiang, *Development of hidden semi-Markov models for diagnosis of multiphase batch operation.* Chemical engineering science, 2011. **66**(6): p. 1087-1099.

[12]. Sun, W., et al., *A method for multiphase batch process monitoring based on auto phase identification.* Journal of Process Control, 2011. **21**(4): p. 627-638.

[13]. Ge, Z., F. Gao, and Z. Song, *Batch process monitoring based on support vector data description method.* Journal of Process Control, 2011. **21**(6): p. 949-959.

[14]. Zhao, C., *Concurrent phase partition and between-mode statistical analysis for multimode and multiphase batch process monitoring.* AIChE Journal, 2014. **60**(2): p. 559-573.

[15]. Zhang, S., C. Zhao, and F. Gao, *Two-directional concurrent strategy of mode identification and sequential phase division for multimode and multiphase batch process monitoring with uneven lengths.* Chemical Engineering Science, 2018,**178**:p. 104-117.

[16]. Kourti, T., *Multivariate dynamic data modeling for analysis and statistical process control of batch processes, start-ups and grade transitions.* Journal of Chemometrics, 2003. **17**(1): p. 93-109.

[17]. Nielsen, N.-P.V., J.M. Carstensen, and J. Smedsgaard, *Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping.* Journal of Chromatography A, 1998. **805**(1): p. 17-35.

[18]. González-Martínez, J., A. Ferrer, and J. Westerhuis, *Real-time synchronization of batch trajectories for on-line multivariate statistical process control using dynamic time warping.* Chemometrics and Intelligent Laboratory Systems, 2011. **105**(2): p. 195-206.

[19]. Gins, G., P. Van den Kerkhof, and J.F. Van Impe, *Hybrid derivative dynamic time warping for online industrial batch-end quality estimation.* Industrial & Engineering Chemistry Research, 2012. **51**(17): p. 6071-6084.

[20]. Neogi, D. and C.E. Schlags, *Multivariate statistical analysis of an emulsion batch process.*

Industrial & engineering chemistry research, 1998. **37**(10): p. 3971-3979.

[21]. Fransson, M. and S. Folestad, *Real-time alignment of batch process data using COW for on-line process monitoring.* Chemometrics and Intelligent Laboratory Systems, 2006. **84**(1): p. 56-61.

[22]. Zhao, C., et al., *Statistical analysis and online monitoring for handling multiphase batch processes with varying durations.* Journal of Process Control, 2011. **21**(6): p. 817-829.

[23]. Li, G., S.J. Qin, and D. Zhou, *A new method of dynamic latent-variable modeling for process monitoring.* IEEE Transactions on Industrial Electronics, 2014. **61**(11): p. 6438-6445.

[24]. Chen, J. and K.-C. Liu, *On-line batch process monitoring using dynamic PCA and dynamic PLS models.* Chemical Engineering Science, 2002. **57**(1): p. 63-75.

[25]. Lu, N., et al., *Two-dimensional dynamic PCA for batch process monitoring.* AIChE Journal, 2005. **51**(12): p. 3300-3304.

[26]. Yao, Y. and F. Gao, *Subspace identification for two-dimensional dynamic batch process statistical monitoring.* Chemical Engineering Science, 2008. **63**(13): p. 3411-3418.

[27]. Corbett, B. and P. Mhaskar, *Subspace identification for data-driven modeling and quality control of batch processes.* AIChE Journal, 2016. **62**(5):p. 1581-1601.

[28]. Garg, A. and P. Mhaskar, *Subspace Identification Based Modeling and Control of Batch Particulate Processes.* Industrial & Engineering Chemistry Research, 2017. **56**(26):p. 7491-7502.

[29]. Zhao, Z., B. Huang, and F. Liu, *Parameter estimation in batch process using EM algorithm with particle filter.* Computers & Chemical Engineering, 2013. **57**: p. 159-172.

[30]. Zhao, Z., B. Huang, and F. Liu, *State estimation in batch process based on two-dimensional state-space model.* Industrial & Engineering Chemistry Research, 2014. **53**(50): p. 19573-19582.

[31]. Wen, Q., Z. Ge, and Z. Song, *Data-based linear Gaussian state-space model for dynamic process monitoring.* AIChE Journal, 2012. **58**(12): p. 3763-3776.

[32]. Ge, Z. and X. Chen, *Supervised linear dynamic system model for quality related fault detection in dynamic processes.* Journal of Process Control, 2016. **44**: p. 224-235.

[33]. Ding, S., et al., *Subspace method aided data-driven design of fault detection and isolation systems.* Journal of Process Control, 2009. **19**(9): p. 1496-1510.

[34]. Nasrabadi, N.M., *Pattern recognition and machine learning.* Journal of electronic imaging, 2007. **16**(4): p. 049-901.

[35]. Katayama, T., *Subspace Methods for System Identification.* Communications & Control Engineering, 2010. **34**(12): p. 1507–1519.

[36]. Jain, A.K., M.N. Murty, and P.J. Flynn, *Data clustering: a review.* ACM computing surveys (CSUR), 1999. **31**(3): p. 264-323.

[37]. Kano, M., et al., *Statistical process monitoring based on dissimilarity of process data.* AIChE Journal, 2002. **48**(6): p. 1231-1240.

[38]. Van Impe, J. and G. Gins, *An extensive reference dataset for fault detection and identification in batch processes.* Chemometrics and Intelligent Laboratory Systems, 2015. **148**: p. 20-31.

[39]. Joe Qin, S., *Statistical process monitoring: basics and beyond.* Journal of chemometrics, 2003. **17**(8-9): p. 480-502.

[40]. Wang, K., J. Chen, and Z. Song, *Concurrent Fault Detection and Anomaly Location in Closed-Loop Dynamic Systems With Measured Disturbances.* IEEE Transactions on Automation Science and Engineering, 2018: p. 1-13.

Appendix A. The derivations of EM algorithm dealing with multibatch data

Given the three expected statistics (28)~(30) obtained in E-step, the parameters in LDS need to be updated in the M-step. The maximum likelihood function has been given in Eq.(24). First, the maximum likelihood function with respect to $\boldsymbol{\mu}_p^m(0)$and $V_p^m(0)$ is

$$\begin{aligned} Cost_1 &= \sum_{i=1}^{I_m} \ln\left(p\left(\mathbf{t}_p^{m,i}(0) \mid \boldsymbol{\mu}_p^m(0), V_p^m(0)\right)\right) \\ &\propto -\sum_{i=1}^{I_m} \ln\left(\left|V_p^m(0)\right|\right) - \sum_{i=1}^{I_m}\left(\mathbf{t}_p^{m,i}(0)-\boldsymbol{\mu}_p^m(0)\right)^T V_p^m(0)^{-1}\left(\mathbf{t}_p^{m,i}(0)-\boldsymbol{\mu}_p^m(0)\right) \end{aligned} \tag{37}$$

Taking the expectation of Eq.(37) with $\mathbf{t}_p^{m,i}(0)$, then there is

$$\begin{aligned} & E\left(Cost_1\right) \propto \\ & -\sum_{i=1}^{I_m} \ln\left(\left|V_p^m(0)\right|\right) - \sum_{i=1}^{I_m} E\left(\left(\mathbf{t}_p^{m,i}(0)-\boldsymbol{\mu}_p^m(0)\right)^T V_p^m(0)^{-1}\left(\mathbf{t}_p^{m,i}(0)-\boldsymbol{\mu}_p^m(0)\right)\right) \\ & = -\sum_{i=1}^{I_m} \ln\left(\left|V_p^m(0)\right|\right) - \\ & \sum_{i=1}^{I_m}\left(tr\left(E\left(\mathbf{t}_p^{m,i}(0)\mathbf{t}_p^{m,i}(0)^T\right)\right) - 2\boldsymbol{\mu}_p^m(0)^T V_p^m(0)^{-1} E\left(\mathbf{t}_p^{m,i}(0)\right) + \boldsymbol{\mu}_p^m(0)^T V_p^m(0)^{-1} \boldsymbol{\mu}_p^m(0)\right) \end{aligned} \tag{38}$$

where $tr(\bullet)$ is the trace of one matrix. By maximizing the above $E\left(cost_1\right)$ with respect to $\boldsymbol{\mu}_p^m(0)$and $V_p^m(0)$, i.e., setting the derivative of $E\left(cost_1\right)$to be zero, the updated parameters can be calculated by Eqs.(31) and (32).

Similarly, from Eq.(24), the maximum likelihood function with respect to $A_p^m$and Q_p^m

$$
\begin{aligned}
Cost_2 &= \sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}} \ln\left(p\left(\mathbf{t}_p^{m,i}(k) \mid \mathbf{t}_p^{m,i}(k-1), A_p^m, Q_p^m\right)\right) \\
&\propto -\sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}} \ln\left(\left|Q_p^m\right|\right) \\
&-\sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}} \left(\mathbf{t}_p^{m,i}(k) - A_p^m \mathbf{t}_p^{m,i}(k-1)\right)^T \left(Q_p^m\right)^{-1} \left(\mathbf{t}_p^{m,i}(k) - A_p^m \mathbf{t}_p^{m,i}(k-1)\right)
\end{aligned}
\tag{39}
$$

Taking the expectation of $Cost_2$ and there is

$$
\begin{aligned}
&E\left(Cost_2\right) \propto -\sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}} \ln\left(\left|Q_p^m\right|\right) \\
&-\sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}} E\left(\left(\mathbf{t}_p^{m,i}(k) - A_p^m \mathbf{t}_p^{m,i}(k-1)\right)^T \left(Q_p^m\right)^{-1} \left(\mathbf{t}_p^{m,i}(k) - A_p^m \mathbf{t}_p^{m,i}(k-1)\right)\right) \\
&= -\sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}} \ln\left(\left|Q_p^m\right|\right) - \sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}} \left\{ tr\left(E\left(\mathbf{t}_p^{m,i}(k)\mathbf{t}_p^{m,i}(k)^T\right)\right)\left(Q_p^m\right)^{-1} \right. \\
&-tr\left(E\left(\mathbf{t}_p^{m,i}(k-1)\mathbf{t}_p^{m,i}(k)^T\right)\left(Q_p^m\right)^{-1} A_p^m\right) - tr\left(E\left(\mathbf{t}_p^{m,i}(k)\mathbf{t}_p^{m,i}(k-1)^T\right)\left(A_p^m\right)^T \left(Q_p^m\right)^{-1}\right) \\
&\left. +tr\left(E\left(\mathbf{t}_p^{m,i}(k-1)\mathbf{t}_p^{m,i}(k-1)^T\right)\left(A_p^m\right)^T \left(Q_p^m\right)^{-1} A_p^m\right)\right\}
\end{aligned}
\tag{40}
$$

By setting the derivative of $E\left(cost_2\right)$to be zero, the updated parameters of A_p^m and Q_p^m can be calculated by Eqs.(33) and (34).

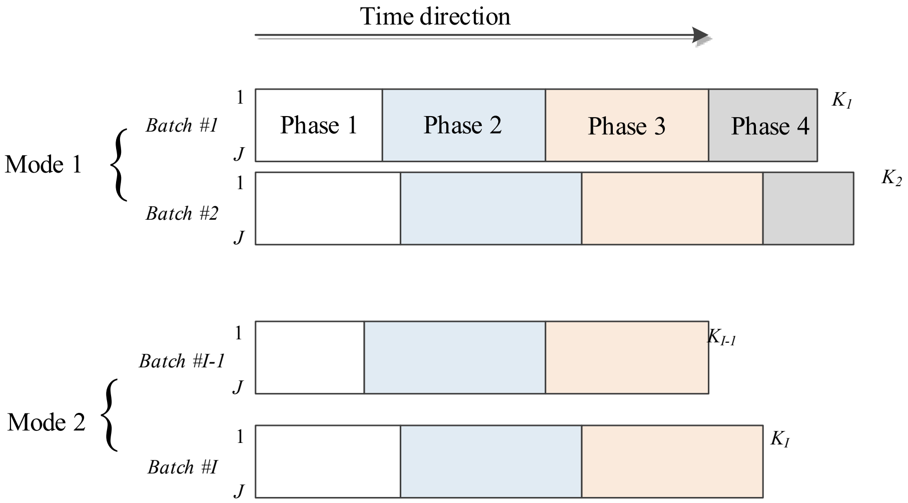
Finally, the maximum likelihood function with respect to $C_p^m$and R_p^m is given by

$$
\begin{aligned}
Cost_3 &= \sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}} \ln\left(p\left(\overline{\mathbf{x}}_p^{m,i}(k) \mid \mathbf{t}_p^{m,i}(k), C_p^m, R_p^m\right)\right) \\
&\propto -\sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}} \ln\left(\left|R_p^m\right|\right) \\
&-\sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}} \left(\overline{\mathbf{x}}_p^{m,i}(k) - C_p^m \mathbf{t}_p^{m,i}(k)\right)^T \left(R_p^m\right)^{-1} \left(\overline{\mathbf{x}}_p^{m,i}(k) - C_p^m \mathbf{t}_p^{m,i}(k)\right)
\end{aligned}
\tag{41}
$$

The expectation of $Cost_3$ is

$$
\begin{aligned}
E\left(Cost_3\right) &\propto -\sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}}\ln\left(\left|R_p^m\right|\right) \\
&-\sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}}E\left(\left(\bar{\mathbf{x}}_p^{m,i}\left(k\right)-C_p^m\mathbf{t}_p^{m,i}\left(k\right)\right)^T\left(R_p^m\right)^{-1}\left(\bar{\mathbf{x}}_p^{m,i}\left(k\right)-C_p^m\mathbf{t}_p^{m,i}\left(k\right)\right)\right) \\
&=-\sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}}\ln\left(\left|R_p^m\right|\right)-\sum_{i=1}^{I_m}\sum_{k=1}^{L_p^{m,i}}\left\{\bar{\mathbf{x}}_p^{m,i}\left(k\right)^T\left(R_p^m\right)^{-1}\bar{\mathbf{x}}_p^{m,i}\left(k\right)-2\bar{\mathbf{x}}_p^{m,i}\left(k\right)^T\left(R_p^m\right)^{-1}C_p^mE\left(\mathbf{t}_p^{m,i}\left(k\right)\right)\right. \\
&\left.+tr\left(E\left(\mathbf{t}_p^{m,i}\left(k\right)\mathbf{t}_p^{m,i}\left(k\right)^T\right)\left(C_p^m\right)^T\left(R_p^m\right)^{-1}C_p^m\right)\right\}
\end{aligned}
\tag{42}
$$

Setting the derivative of $E\left(Cost_3\right)$ to be zero, the updated parameters of $C_p^m$and R_p^m are given by Eqs.(35) and (36).



The Table of Contents graphic.

RESPONSE TO REVIEWERS

EDITORIAL REQUIREMENTS:

Q1. Move Figure 18 from the reference section.

A1. It has been moved.

--Figure 2 is not cited in the manuscript text.

A2. Figure 2 has been cited.

--Is permission needed to publish Table 2?

A3. Table 2 is from the Ref [38] that provides the simulated dataset. We think Table 2 with proper citation is allowed to present in this paper.

--Please add a Table of Contents graphic to the last page of the paper (instructions above).

A4. See the last page.

--References must be according to ACS guidelines--the name of the work cited should be listed.

A5. Yes, they have been added.

Reviewer(s)' Comments to Author:

Reviewer: 1

Comments to the Author

In this manuscript, a linear dynamic system based mode identification and phase division strategy is proposed to solve the process monitoring problem in multiphase and multimode batch processes with uneven batch durations. The performance of the proposed method is illustrated using a simulation process based on Penism fermentation. The idea in this manuscript may have some novelty. However, the motivation of this study is unclear and the description of the methodology is quite confused. In the following, I will present some comments for which I found particularly concern.

Q1. The format of the cited literature should be double checked.

A1. The format of the cited references has been revised to meet the requirements of the journal.

Q2. In Page 20, “mode clustering and phase re-identification procedures are complete” should be revised as “mode clustering and phase re-identification procedures are completed”. Please check other part of the article to avoid grammar mistakes.

A2. Thanks for pointing out the mistake. The whole paper has been double-checked to avoid these typos. See Line 429 on Page 23.

Q3. In Page 12, the authors mentioned Steps 3.1 and 3.2, but I cannot found these in the article. The authors should check their manuscript carefully to avoid similar mistakes.

A3. Thanks for this comment. These notes were missing in the first submission with

Latex. They have been added. Please see these modifications on Page 11-13.

Q4. In the abstract, the authors claim that “Batch processes are often characterized by non-linear dynamics and varying operating conditions”. Since the batch processes are generally non-linear, why do the authors develop monitoring model using linear dynamic system?

A4. We apologize for such a confusing statement about nonlinear dynamics in batch processes. Because the operating conditions vary, there are often multiphases/multistages in the batch process that produces a target product. In a lot of applications, the operation at each stage usually can be treated as a steady-state behavior, so each phase can be characterized as linear dynamics. Therefore, from the viewpoint of the whole batch, batch processes are often characterized by non-linear dynamics. To avoid this confusion, we think the phrase “piecewise linear dynamics” should be used to replace “non-linear dynamics”. Please see Line 1-2 on Page 1.

Q5. In the introduction, the authors claim that “Although there have been numerous studies to address uneven batch durations they generally are based on assumptions that are difficult to satisfy in practice”. However, I think the examples the authors provided are not sufficient for this statement. Hence, I suggest that the author spend more space to introduce the monitoring strategies for uneven batch processes.

A5. Yes. The methods of dealing with uneven batch durations can be divided into two classes. One uses an indirect way to align or synchronize uneven-length batches through some preprocessing procedures, such as interpolation, truncation, dynamic time warping, and so on. In the introduction, we focus more on the indirect methods. Also, there are other methods that can directly deal with the uneven length issue. Regarding the direct methods, more relevant explanations have been given as the reviewer suggested. The modifications are made as follows:

What is worth mentioning is that there often exist the raw batches with different durations. When it is claimed that batch process data can be represented by a standard three-dimensional array $\mathbf{X} \in \mathrm{R}^{I \times J \times K}$, the underlying assumption is that all the batches with different lengths (K_i in Figure 2) have been aligned or synchronized into the same length. Commonly used techniques include interpolation, truncation, defining an indicator variable for resampling process variables, dynamic time warping (DTW) and correlation optimization warping (COW) [1, 16-21]. The drawbacks of these batch length equalization methods are well documented [22]. Specifically, interpolation can introduce incorrect information while truncation can result in the loss of useful information. The DTW and COW methods cannot be used for online monitoring because they require complete batch lengths beforehand. In some applications, it is even difficult to find a suitable indicator variable to align with the other process variables of equal length. Furthermore, under batch alignment, the columns of time-slice or time segment matrices correspond to the equal value of the alignment variable rather than the equal elapsed time. This would corrupt the original dynamic correlations and change the original data distribution. Hence, these strategies may work when the uneven lengths are not severe. Without “equalizing” batches as a preprocessing step, directly learning models from the raw batch data with severe uneven batch durations has attracted more attention. Zhao et al. took the batches with approximate durations as one mode/group. They analyzed the differences between the groups to extract between-group and within-group information for fault monitoring. However, it is not necessarily true to simply assume that the duration is a unique factor determining the group whose batches have similar features [22]. Zhang et al. address the drawback of the regular time-slice data matrix by proposing a pseudo-time-slice matrix constructed using the k nearest neighbor (kNN) rule to search similar samples within a data window [15]. However, the employment of kNN requires training data to be saved for online monitoring. These algorithms have inherent limitations as they are primarily meant for independent and identically distributed (IID) data. They do not account for temporal correlations of each variable.

See Line 94-122 on Page 5-6.

Q6. In Page 7, I do not agree with the statement that “Linear dynamic system(LDS) is a model that is built using a small section of the time series data from each batch and as such it is not affected by the uneven batch durations unlike the standard multivariate statistical analysis (MVA) techniques”. According to the “Iterative distance evaluation” in Page 12, the proposed method also divides the multiphase process into different phases and then develops model for each phase. Besides, the MVA technique can also develop monitoring model using part of the data in a phase.

A6. Yes. We agree with this opinion. Both LDS and MVA can use part of the data in one phase to develop monitoring models. We think our statement in the original manuscript caused confusion. What is expected to present in the sentence is that LDS is able to naturally learn the dynamic model without considering or compromising batch durations. LDS is a parametric dynamic model that describes the time series. Hence, the LDS is trained using the batch data along the time direction. Take two batches in Fig.1 for example. The objective of maximizing the log-likelihood is given by

$$\max \ln p\left(X^1, X^2 \mid \boldsymbol{\theta}\right) = \ln p\left(X^1 \mid \boldsymbol{\theta}\right) + \ln p\left(X^2 \mid \boldsymbol{\theta}\right)$$

where X^1 has K_1 samples and X^2 has K_2 samples. $\boldsymbol{\theta}$ is the parameter set of LDS. The log-likelihood function of multi-batch data can be decomposed into the sum of the log-likelihood function of each individual batch in which the data are regarded as time series like a continuous process. Hence, LDS can naturally learn process dynamics no matter how long the batch duration is.

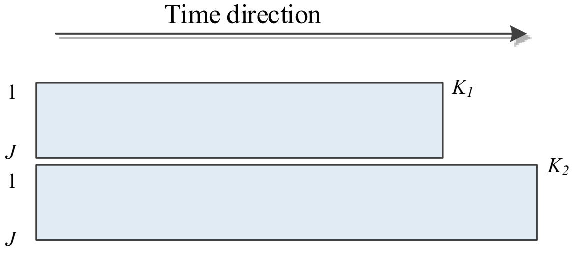


Figure. 1 The time direction for LDS

MVA mostly needs to analyze the statistical models in each time instance. Hence, MVA actually constructs statistical models along the batch direction. Because of uneven durations in different batches, MVA has to take extra costs to deal with the uneven duration issue, which has been reviewed in A5. The corresponding modification refers to Line 182-184 on Page 10.

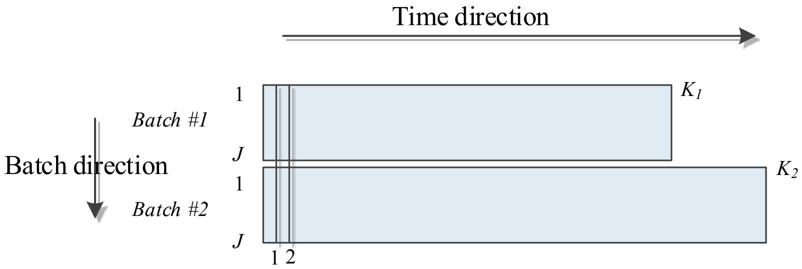


Figure 2. The batch direction for MVA

Q7. What is the training data of Eq. (4), the historical data of a single batch or the historical data of all the batches?

A7. Eq.(4) is trained with a single batch data for dividing the phase of this batch. To avoid confusion, this point has been emphasized in the revised manuscript. See Line 203 on Page 11.

Q8. In Eq. (8), the authors assume that statistic D follows chi square distribution. For monitoring strategies based on PCA and PLS, the extracted features are mutually orthogonal. Hence, these features are assumed to be independent. However, the variables in a real industrial process are generally related with each other, and the independence assumption is difficult to be satisfied. Therefore, I suggest the authors use kernel density estimation to calculate the control limit of statistic D.

A8. Eq.(8) is $D_p^i(k)=\left(\overline{\mathbf{x}}_p^i(k)-\mathbf{z}_p^i(k)\right)^T \mathbf{\Sigma}_p^i(k)^{-1}\left(\overline{\mathbf{x}}_p^i(k)-\mathbf{z}_p^i(k)\right)\sim\chi^2(J)$, where $\mathbf{\Sigma}_p^i(k)$ is a covariance matrix. With SVD, $\mathbf{\Sigma}_p^i(k)^{-1}=U_p^i(k)\Lambda_p^i(k)^{-1}U_p^i(k)^T$. $U_p^i(k)$ is a unitary matrix with mutually orthogonal columns so that the original variables are now rotated into orthogonal coordinates. Moreover, by multiplying $\Lambda_p^i(k)^{-1}$, the variables in the new coordinate system are all with unit variances. Hence, Eq.(8) is the sum of *J* unit Gaussians and it is still subjected to $\chi^2(J)$. The related statement has been revised to avoid confusion. See Line 226-228 on Page 12.

Q9. The methodology part of this article is quite confused. A flow chart of the proposed method may help the readers to better understand the proposed method.

A9. We are sorry to cause confusion in the methodology. The whole methodology procedure can be briefly summarized as follows.

a. Perform the sequential phase division for every single batch. The LDS model is used to judge the phase switching point.

b. Design the batch distance by comprehensively considering the phase features in each batch. The batch distance is used to measure the difference between two batches. And the k-means clustering is employed to group the similar batches as a mode.

c. Re-train LDS models for each phase using all the batch data in the same mode.

d. Design monitoring metrics and strategies for fault detection.

To avoid confusion, we made some proper modifications in the Introduction Section to clarify the motivations and enhance the organization of the paper. See Line 136-153 on Page 7.

As the reviewer suggested, the flow chart of the modeling training procedures is presented in Figure 3 as follows. See Figure 6 in the updated manuscript.

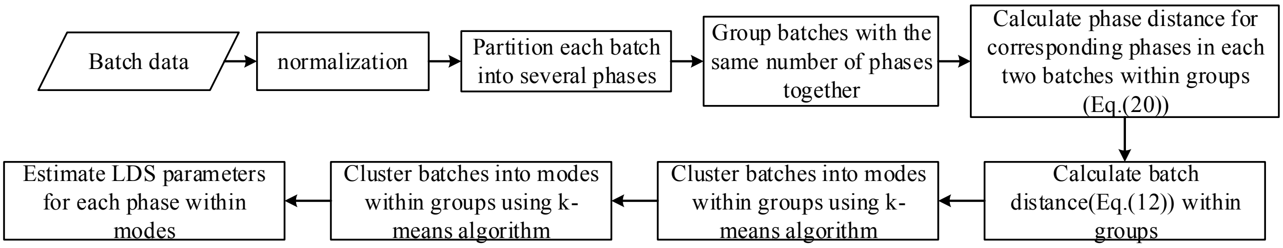

Batch data
normalization
Partition each batch into several phases
Group batches with the same number of phases together
Calculate phase distance for corresponding phases in each two batches within groups (Eq.(20))
Calculate batch distance(Eq.(12)) within groups
Cluster batches into modes within groups using k-means algorithm
Cluster batches into modes within groups using k-means algorithm
Estimate LDS parameters for each phase within modes

Figure 3. The flow chart of the methodology

Q10. In Penism fermentation process, the number of variables is 14. Why the author select these ten variables to do experiment.

A10. In the fed-batch penicillin fermentation process, 14 process variables are often used. However, this paper adopts the extensive dataset provided by "Van Impe, J. and G. Gins, *An extensive reference dataset for fault detection and identification in batch processes.* Chemometrics and Intelligent Laboratory Systems, 2015. **148**: p. 20-31", in which 11 process variables were measured online. We also copied the variable table here. Note that the feed rate variable is discarded in our paper. The feed rate at the start is zero and then it tends to be a constant with a very small fluctuation, so it is likely to cause singularity problems in model building. The trend of the feed rate in the normal condition is presented in Figure 4. That is why 10 variables are selected in this paper. To avoid confusion, this supplementary statement has been added (Line 470-473 on Page 25).

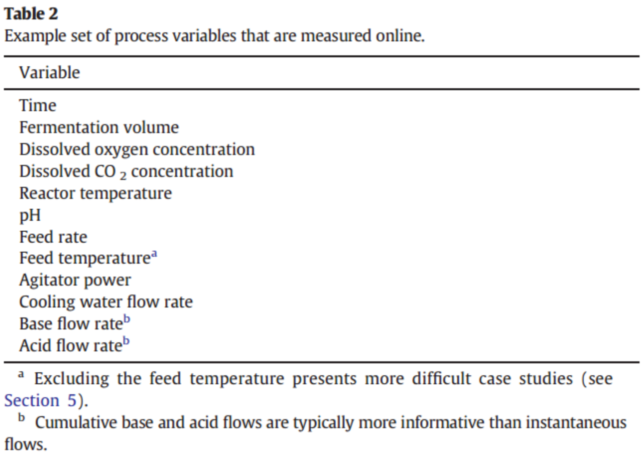


Table 2
Example set of process variables that are measured online.

Variable
Time
Fermentation volume
Dissolved oxygen concentration
Dissolved CO $_2$ concentration
Reactor temperature
pH
Feed rate
Feed temperature[a]
Agitator power
Cooling water flow rate
Base flow rate[b]
Acid flow rate[b]

[a] Excluding the feed temperature presents more difficult case studies (see Section 5).

[b] Cumulative base and acid flows are typically more informative than instantaneous flows.

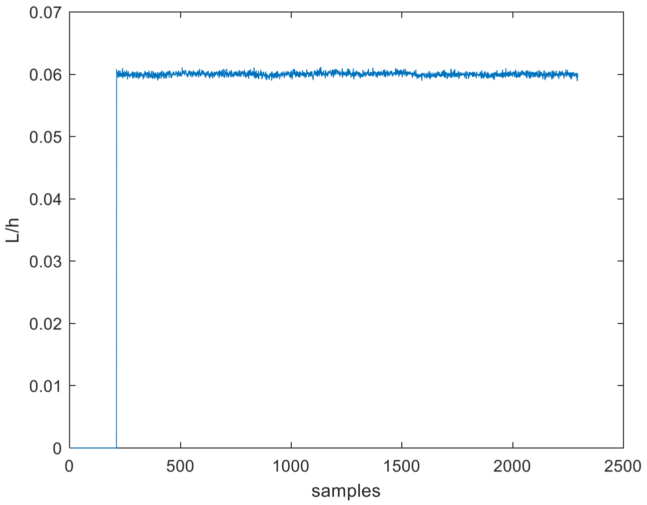


Figure 4. The trend of the feed rate.

Q11. In Figure 13, why the control limits of different phases are the same?

A11. Since the following squared Mahalanobis distance is used as the monitoring statistic,

$$D_p^i(k) = \left(\bar{\mathbf{x}}_p^i(k) - \mathbf{z}_p^i(k)\right)^T \mathbf{\Sigma}_p^i(k)^{-1}\left(\bar{\mathbf{x}}_p^i(k) - \mathbf{z}_p^i(k)\right) \sim \chi^2(J)$$

the control limit D_α is determined by the degree of freedom J (i.e., the variable number) and the confidence level α. Thus, it is right that all the points in different phases have the same control limit. In addition, all the monitoring statistics are scaled by the control limit in monitoring figures; i.e., the points in these monitoring figures (Figure 14-19 in the revised manuscript) are given by $D_p^m(k)/D_\alpha$, which indicates anomaly when it exceeds 1. That is why the control limit in the figures is always 1. To avoid confusion, the relevant illustrations have been added (Line 579-581 on Page 35).

Q12. In Table 3, the accurate detection rates of these two methods should also be provided.

A12. In Table 3, we use the format “FDR of LDSMIPD/ FDR of CMIPD” to present the results. For example, 56/48 means FDR of the first method is 56% while FDR of the second method is 48%. Table 3 can not only provide the detection rates of the two methods but also benefit readers by intuitively comparing the performance of the two methods. Hence, we think Table 3 has provided the detection rates the reviewer expected. To avoid misunderstanding, the explanations have been added to the manuscript. See Line 615-618 on Page 38.

Q13. Since the authors develop the LDSMIPD method to solve the drawbacks of the MVA methods, some MVA method should also be adopted to make a comparison.

A13. Yes. MVA has been widely applied to batch process modeling and monitoring. However, there are very few studies on multimode and multiphase batch processes with the uneven batch duration issue. The comparative method we chose in this work is CMIPD “Zhang, S., C. Zhao, and F. Gao, *Two-directional concurrent strategy of mode identification and sequential phase division for multimode and multiphase batch process monitoring with uneven lengths.* Chemical Engineering Science, 2018.”, which is the newest method dealing with the same problem as ours. The core of CMIPD is principal component analysis (PCA) for building phase models, so it is a typical MVA method. We think this comparison is fair and representative enough and can reflect the merits of our method. See Line 549-551 on Page 33.

Reviewer: 2

Comments to the Author

Main contributions of this manuscript to is as follows :

(1) Proposing a novel linear dynamic system based mode identification and phase division to deal with the issue of dynamic modeling and online monitoring with uneven length multimode and multiphase batch processes.

(2) Identifying different phases using a dissimilarity measure for linear dynamic systems to model each phase.

(3) Constructing a multimode model is using a model distance metric to cluster batches with similar phases.

(4) Estimating phase parameters by expectation maximization algorithm..

It is a challenging task to monitor multimode and multiphase batch processes. This manuscript is interesting and can be considered for a possible publication after a minor revision. The main questions are listed as follows.

Q1. What is “KI” in Figure 2. Why is it marked here?

A1. In practical batch processes, different batches certainly have different durations. Hence, K_I represents the I-th batch length different from the aligned length *K* in Figure 2. See Line 177-179 on Page 10.

Q2. “Ki3” should be corrected into “K3i” in Figure 3.

A2. Thank you for pointing out this typo. It has been changed. See Figure 3 on Page 10.

Q3. How to deal with the problem that different variable has different phase length.

A3. This problem is interesting. In view of practical process operations, the phase refers to a specific operation stage whose process characteristic varies a lot from the previous stage, i.e. phase change can be briefly understood as operation switching. The problem that different variables have different phase lengths is generally not severe in batch processes, so the phases in literature are defined as the sequential data subjected to a unified multivariate model or distribution instead of considering those variables separately. This is also the reason why we do not consider this problem in our work.

Regarding the problem the reviewer mentioned, some studies on continuous processes may be promising and helpful (Das L, Rengaswamy R, Srinivasan B. Data mining and control loop performance assessment: The multivariate case[J]. AIChE Journal, 2017, 63(8): 3311-3328.) The strategies generally segment sequential data for each variable and then concatenate these changing points from all the variables.

To avoid confusion, we added the relevant explanations to show all variables are considered to be synchronized in the evolution of phases. See Line 28-30 on Page 2.

Reviewer: 3

Comments to the Author

This is an interesting contribution related to monitoring in batch processes, but needs to improved before being accepted for publication.

Q1. Batch processes (at least in chemical engineering) are inherently nonlinear. Why are linear dynamic systems being used to model each phase, and why is the dissimilarity measure used to identify phase specific to linear systems? For the case study (benchmark data set) considered, this may not be an issue, but this does not seem to be a reasonable assumption for the general batch problem.

A1. We apologize for such a confusing statement about nonlinear dynamics in batch processes. As the operating conditions vary, there are often multiphases/multistages in the batch process that produces a target product. In a lot of applications, the operation in a stage usually can be treated as a steady-state behavior so that each phase can be characterized as linear dynamics. Therefore, from the viewpoint of the whole batch, batch processes are often characterized by non-linear dynamics. We think the phrase "piecewise linear dynamics" should replace "non-linear dynamics" to avoid confusion. Please see Line 1-2 on Page 1.

Q2. Also, were alternatives to the EM algorithm considered for the identification of the dynamic model?

A2. Sure. Subspace identification is a matrix projection and decomposition method as an alternative to the EM algorithm with the Bayesian method. See Line 129-131 on Page 7 and Ref.[27] *Corbett B, Mhaskar P. Subspace identification for data-driven modeling and quality control of batch processes[J]. AIChE Journal, 2016, 62(5): 1581-1601.*

Q3. I fail to understand the distinction (if there is any) between the notion of 'phase' and 'mode' used in the manuscript. Phases have been characterized as different stages of dynamic behavior that yield different process variable characteristics. The authors state that batches with similar phase characteristics are considered to exhibit the same mode and a batch process that exhibits multiple modes is referred to as the multimode batch process. This appears to indicate that the existence of a specific phase (a distinct dynamic signature) is what is used to classify a batch into a mode. Does this imply that any batch with more than one phase would necessarily exhibit multiple modes? Also, is there an assumption that batches with equal number of phases have the phases appearing in the same sequence?

A3. Let us take the fermentation process in the following figure for example. To achieve the final target, one batch needs to go through several different but indispensable stages, known as phases. However, in some phases, the operations can be adjusted slightly according to practical situations. For example, the second phase in Figure 4 may be exponential or log, determined by the component of feed or other factors. Batches with the exponential trend in the second phase can be taken as the same mode distinguished from the others with the log trend in the second phase. Hence, the difference between the phase and the mode can be summarized as:

A phase is about the varying process along the time direction and it is an inner property of one single batch. The mode represents the differences from batch to batch. The difference between the modes is caused by inconsistent operations at some specific stages.

Therefore, we think it is correct to say that that a specific phase (a distinct dynamic signature) is used to classify a batch into a mode. But one batch can just belong to one mode. And it is also right that the batches with the equal number of phases have the phases appearing in the same sequence because a good batch operation generally follows a fixed operation procedure with good repeatability. To avoid confusion, these explanations have been supplemented (Line 37-40 on Page 2).

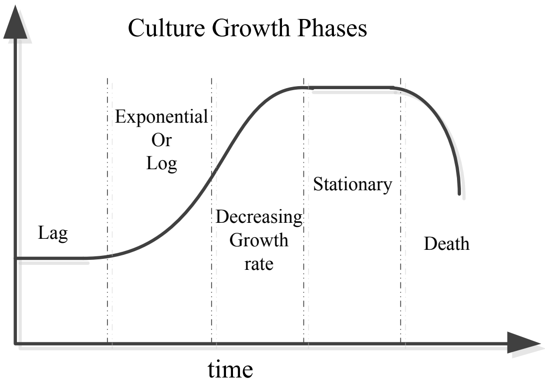


Figure 4. Illustration of 4 culture growth phases in penicillin fermentation

Q4. It is entirely possible that the grouping of batches into modes can result in only one batch belonging to a particular mode. In this case, I_m=1. Are there any issues for determining Xbar^m_p in this case?

A4. Sure. Mathematically, it is likely to group several batches into a particular mode. In this case, this batch is generally considered as an outlier batch. However, it does not make sense to use the mode with only one batch to construct a monitoring model. Our method is better as it works for the mode that only contains one batch because the linear dynamic system (LDS) uses time series to construct the model. Data in one batch are certainly time series like continuous processes. Actually, in the phase division (Section 2), the model is trained based on one single batch data. See Section 2.

Q5. On page 10, the explanation of how the degrees of freedom of the chi-square distribution and the control limit are determined ('by the number of remaining singular values') can be explained more clearly.

A5. Eq.(8) is $D_p^i(k)=\left(\bar{\mathbf{x}}_p^i(k)-\mathbf{z}_p^i(k)\right)^T \mathbf{\Sigma}_p^i(k)^{-1}\left(\bar{\mathbf{x}}_p^i(k)-\mathbf{z}_p^i(k)\right)\sim\chi^2(J)$, where $\mathbf{\Sigma}_p^i(k)$ is a covariance matrix and J is the number of variables. With SVD, $\mathbf{\Sigma}_p^i(k)^{-1}=U_p^i(k)\Lambda_p^i(k)^{-1}U_p^i(k)^T$. $U_p^i(k)$ is a unitary matrix with mutually orthogonal columns so that the original variables can now be rotated into orthogonal coordinates. Moreover, by multiplying $\Lambda_p^i(k)^{-1}$, all the variables in the new coordinate system have unit variances. Eq.(8) is the sum of *J* unit Gaussians and it is subjected to $\chi^2(J)$. If all the singular values in $\Lambda_p^i(k)$ are significantly larger than zero, it means no significant linear relationship exists in the variables, so the degree of the freedom is *J.* Otherwise, one should first delete the components with the singular values close to zero to avoid the singularity. The singular values larger than zero constitute the "remaining singular values", which represents the degree of freedom of the chi-square distribution. See Line 226-228 and Line 233-235 on Page 12-13.

Q6. In the section on dynamic phase segmentation, there are mysterious references to 'Step 1', 'Step 2' and 'Steps 3.1 and 3.2'. Given that there is no numbering of steps in the procedure, this makes the algorithm hard to follow, especially for iterative distance evaluation.

A6. Thanks for this comment. Those notes were missing in the first submission with Latex. Now they have been added. Please see these modifications on Page 11-13.