Nonlinear Batch Process Monitoring Using Phase-Based Kernel-Independent Component Analysis–Principal Component Analysis (KICA–PCA)

ARTICLE *in* INDUSTRIAL & ENGINEERING CHEMISTRY RESEARCH · AUGUST 2009

Impact Factor: 2.59 · DOI: 10.1021/ie8012874

3 AUTHORS, INCLUDING:

Chunhui Zhao
Zhejiang University
91 PUBLICATIONS **580** CITATIONS
SEE PROFILE

Furong Gao
The Hong Kong University of Science and T…
267 PUBLICATIONS **3,770** CITATIONS
SEE PROFILE

Nonlinear Batch Process Monitoring Using Phase-Based Kernel-Independent Component Analysis−Principal Component Analysis (KICA−PCA)

Chunhui Zhao,[†,‡] Furong Gao,*[,†] and Fuli Wang[‡]

Department of Chemical and Biomolecular Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, and College of Information Science and Engineering, Northeastern University, Shenyang, Liaoning Province, P.R. China

In this article, the statistical modeling and online monitoring of nonlinear batch processes are addressed on the basis of the kernel technique. First, the article analyzes the conventional multiway kernel algorithms, which were just simple and conservative kernel extensions of the original multiway linear methods and thus inherited their drawbacks. Then, an improved nonlinear batch monitoring method is developed. This method captures the changes of the underlying nonlinear characteristics and accordingly divides the whole batch duration into different phases. Then, focusing on each subphase, both nonlinear Gaussian and non-Gaussian features are explored by a two-step modeling strategy usingkernel-independent component analysis−principal component analysis (KICA−PCA). Process monitoring and fault detection can be readily carried out online without requiring the estimation of future process data. Meanwhile, the dynamics of the data are preserved by exploring time-varying covariance structures. The idea and performance of the proposed method are illustrated using a real three-tank process and a benchmark simulation of fed-batch penicillin fermentation production.

1. Introduction

Batch processes play an important role in producing high-value-added products to meet today's rapidly changing market demand. Multivariate statistical analysis techniques[1−12] based on multiway principal component analysis (MPCA) and multiway partial least-squares (MPLS) have been widely applied to fault detection of batch processes. They can check and supervise whether a batch process is operating within the acceptable variation range. However, their achievable performance is limited because of the assumption that the monitored variables are normally distributed. Recently, independent component analysis (ICA),[13−15] a useful extension of PCA, has been given extensive attention as an alternative that can extract some statistically independent components (ICs) underlying process measurements. However, the original ICA algorithm[14,15] exhibits the typical problem that different search optimization solutions can be obtained because of the random initialization in the calculation procedure. This can give rise to some problems and inconvenience in the subsequent multivariate statistical analysis and process monitoring. A modified ICA method, designed by Lee et al.,[16] can determine the order of ICs and produce a unique and repeatable ICA solution by fixing the initialization instead of using random initialization as in the original ICA algorithm. This approach is based on the assumption that the space spanned by the major principal components (PCs) is a good starting point because ICA can be viewed as a natural way to "fine tune" PCA (centering and whitening) with an iterative search optimization procedure. Moreover, by maintaining the variance information of PCs during ICA feature extraction, critical process variation features are successfully captured.[16] A number of applications of ICA[16−23] have been reported, demonstrating the superiority of ICA over other methods in some statistical monitoring cases. On the other hand, ICA likely will not outperform PCA if process variables are normally distributed, as pointed out by Kano et al.[24] Because nonGaussianity is the basis of ICA operation, ICs are meaningless even if they can be calculated when this non-Gaussian assumption is incorrect. This motivates the combination of ICA and PCA[24,25] when some process variables follow normal distributions and others do not, which might be a more common situation in practice. As a unified framework, such an approach can integrate the advantages of both PCA and ICA.[24] In this way, both Gaussian and non-Gaussian features can be addressed, which thus can provide more meaningful and sufficient information for monitoring.

However, for some complicated industry batch processes that have especially nonlinear characteristics, PCA and ICA can exhibit compromised monitoring results because of their linearity assumption. To handle the nonlinear problem, the kernel technique has found increasing applications in recent years. Its basic idea is to transform the nonlinear original space into a high-dimensional linear kernel space by implementing a nonlinear kernel mapping. Then, the nonlinearity can be efficiently explored in the high-dimensional kernel space using only simple standard linear algebra while requiring no nonlinear optimization procedure. Given any algorithm that can be expressed solely in terms of dot products, this kernel method enables one to construct different nonlinear versions by means of simple integral operators and nonlinear kernel functions. Many linear statistical methods can be readily extended to their nonlinear kernel versions, including support vector machines (SVMs),[26] kernel PCA (KPCA),[27−31] kernel PLS (KPLS),[32] kernel Fisher discriminant analysis (KFDA),[33−35] and kernel ICA.[36−38] Lee and co-workers[27−30] developed a series of nonlinear process monitoring and fault diagnosis schemes using KPCA that highlighted the superiority of KPCA over linear PCA in solving nonlinear problems. Zhang and Qin[36] utilized multiway KICA (MKICA) to extract some dominant nonlinear ICs from nonlinear batches and propose them for process monitoring. They demonstrated that MKICA effectively captures the underlying nature of nonlinear batch processes and shows superior fault detectability compared to MICA. Moreover, Zhang[37]

* To whom correspondence should be addressed. E-mail: kefgao@ust.hk.
[†] The Hong Kong University of Science and Technology.
[‡] Northeastern University.

10.1021/ie8012874 CCC: $40.75
Published on Web 08/28/2009

combined KICA with SVMs to improve the performance of fault diagnosis in nonlinear cases. Generally, KICA[36−38] includes two steps: KPCA whitening and linear ICA iteration in the KPCA transformed space, which decomposes the nonlinearity based on higher-order statistics. Numerous successful reports[26−38] have highlighted the competency of the kernel-based approach for nonlinear cases.

Revisiting previously reported kernel investigations for nonlinear batch processes, we found that they[29,34,36] only simply and directly borrowed the idea and means of the conventional multiway linear statistical methods, here termed multiway kernel algorithms. That is, they arranged the three-way reference data batchwise unfolding and took the entire batch operation trajectory as a single object when implementing high-dimensional kernel mapping and kernel operation. Thus, they inherited the common defects of conventional multiway linear statistical methods; for example, they have to estimate the unknown future data at each time during online application, and it is difficult to reveal the changes of underlying nonlinear correlations over phases for multiphase batch processes. For linear batch processes, the data estimation problem during online application has been successfully solved based on various smart techniques.[6,39−44] Wold et al.[39] applied variablewise unfolding to three-way historical data to realize online monitoring without future trajectory estimation, followed by batchwise unfolding for batch-level monitoring. Moreover, their complete analysis of the batch data was performed in three steps, related to the individual observations, the evolution of the batches, and the whole batches. Lee et al.[40,41] combined data scaling based on batchwise unfolding and modeling based on variablewise unfolding to design a monitoring system that could be readily implemented online without data estimation. Rännar et al.[42] developed an adaptive modeling and monitoring method using hierarchical PCA, Ündey et al.[43] constructed an MPLSV regression model between the variablewise unfolded data matrix and the local batch time (or an indicator variable) for online process monitoring, and Lu et al.[44] derived multiple two-dimensional sub-PCA models by averaging those time-slice PCA loadings within the same phase. They all overcame the data estimation problem. On the other hand, the local modeling idea has also drawn much attention and been put into wide applications in linear batches.[2,4,11,43−51] Dong and McAvoy[45] illustrated that two-stage models were more powerful than a single model by utilizing a two-stage jacketed exothermic batch chemical reactor. Lu et al.[44] developed a phase division and thus phase-based statistical modeling method that could better reveal the changes of underlying linear characteristics over phases and thus improve process understanding and monitoring efficiency. Unfortunately, for nonlinear batch processes, the existing kernel-based methods[29,34,36] have never mentioned and addressed the above problems. Further studies are, thus, still necessary considering the significance of these issues.

In the present work, first, the limitations of conventional multiway kernel algorithms are analyzed, and the underlying cause is revealed. Then, a phase-based KICA−PCA two-step statistical analysis strategy is presented to solve these problems and improve the online monitoring performance of nonlinear batch processes. Compared with the existing multiway kernel methods,[29,34,36] it has the following advantages:

(a) Kernel-based statistical analysis is performed focusing on different phases that can better capture and reveal the changes of underlying nonlinear process characteristics.

(b) This approach does not require the estimation of future values, and at the same time, it holds the dynamics of nonlinear features using time-varying score covariance structures that make it beneficial to online implementation.

(c) The KICA−PCA two-step feature extraction strategy explores both Gaussian and non-Gaussian nonlinear behaviors and thus can supervise the underlying nonlinearity more comprehensively.

This article is organized as follows: First, the details of the proposed method, as well as the basic introduction of kernel operation, are described in section 2. Then, the performance and feasibility of the method are analyzed and illustrated in section 3. Finally, conclusions are drawn in section 4.

2. Methodology

In kernel operation, the kernel matrix is the basic analysis unit, whose size is the square of the number of samples. The kernel matrix will become problematic for large amounts of data, and its eigenvector decomposition will also be too demanding in terms of both time and memory. Therefore, previous works[29,34,36] unfolded three-way process data batchwise resulting from the consideration that there was little probability that the number of batches would become large and that the subsequent size of the kernel matrix would become problematic. Such an assumption, however, inevitably introduces the common inconveniences similar to those of conventional multiway linear statistical methods. In this section, a method is proposed to solve these problems that considers the multiplicity of process phases and can be readily put into online implementation with no data estimation.

Assume that J variables are measured at $k = 1, 2, ..., K$ time instances throughout the batch. Then, process data collected from I similar batches can be organized as a three-way array, $\underline{\mathbf{X}}(I \times J \times K)$. K sets of time-slice process data, $\mathbf{X}_k(I \times J)$, can be naturally separated and are normalized by default. In the present work, the batches are of synchronous trajectories without any special declaration so that the specific process time can indicate the operation progress.

2.1. Kernel-Based Phase Division. In linear cases, numerous research works[2,4,11,39−47] have been reported to deal with multiphase batch processes. Alternative phase division solutions have been designed with different advantages and specific applicabilities. A variant of a k-means clustering algorithm was developed by Lu et al.,[44] assuming that the changes of loading matrices can reflect the changes of underlying process characteristics and thus can be used to identify the variation over phases. By clustering those time-slice loadings, consecutive samples with similar underlying process characteristics are collected together in a phase, and dissimilar sample patterns are classified into different phases. However, previous phase identifications are essentially implemented from a linear viewpoint. For nonlinear batches, phase localization should be checked by revealing the changes in nonlinear behaviors throughout the operation duration.

According to Cover's theorem,[52] a nonlinear data structure is more likely to be linear after high-dimensional kernel mapping. Thus, kernel mapping provides a possible application platform for the existing linear phase division methods. In the present work, the clustering method of Lu et al.[44,51] was adopted. To better capture the multiplicity of nonlinear phases, the clustering should be implemented in the mapped high-dimensional feature space, where the nonlinearity underlying process measurement has been converted into linearity. Then, the key is to determine which analysis object can be used to reveal the changes of inherent nonlinear behaviors.

First, considering the nonlinear mapping Φ: $\mathbf{x} \in \mathbf{R}^J \rightarrow \mathbf{z} \in \mathbf{R}^h$, the normalized time-slice data $\mathbf{X}_k(I \times J)$ ($k = 1, 2, ..., K$) is transformed into the high-dimensional feature space, $\Phi(\mathbf{X}_k)$, where the dimension of the feature space, h, can be arbitrarily large or even infinite. Then, the time-slice covariance matrix in the feature space is given by

$$\mathbf{R}_k = \frac{1}{I}\tilde{\Phi}(\mathbf{X}_k)^{\mathrm{T}}\,\tilde{\Phi}(\mathbf{X}_k) \tag{1}$$

where $\tilde{\Phi}(\mathbf{X}_k)$ is the mean centered form of $\Phi(\mathbf{X}_k)$.

The covariance matrix covers the underlying process characteristics, which can be revealed by eigenvalue decomposition

$$\lambda_j \boldsymbol{\nu}_j = \mathbf{R}_k \boldsymbol{\nu}_j = \frac{1}{I}\tilde{\Phi}(\mathbf{X}_k)^{\mathrm{T}}\,\tilde{\Phi}(\mathbf{X}_k)\cdot \boldsymbol{\nu}_j \tag{2}$$

where λ_j, $\boldsymbol{\nu}_j \in \mathbf{R}^h$ are, respectively, the nonzero eigenvalue and the corresponding eigenvector of $\mathbf{R}_k$, representing the variance information and distribution direction respectively.

However, because of the unknown nonlinear mapping function $\Phi(\cdot)$, $\mathbf{R}_k$ is not explicitly known, and the inherent distribution information cannot be determined. Instead, an $I \times I$ kernel matrix, $\mathbf{K}_k = \Phi(\mathbf{X}_k)\Phi(\mathbf{X}_k)^{\mathrm{T}}$, is defined. Then, the mean-centered kernel matrix $\tilde{\mathbf{K}}_k = \tilde{\Phi}(\mathbf{X}_k)\tilde{\Phi}(\mathbf{X}_k)^{\mathrm{T}}$ can also be explicitly derived[27−29]

$$\tilde{\mathbf{K}}_k = \mathbf{K}_k - \mathbf{1}_I\mathbf{K}_k - \mathbf{K}_k\mathbf{1}_I + \mathbf{1}_I\mathbf{K}_k\mathbf{1}_I \tag{3}$$

where

$$\mathbf{1}_I = \frac{1}{I}\begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}$$

The eigenvalue solution of $\tilde{\mathbf{K}}_k$ is obtained as

$$\xi_j \boldsymbol{\alpha}_j = \tilde{\mathbf{K}}_k\boldsymbol{\alpha}_j = \tilde{\Phi}(\mathbf{X}_k)\,\tilde{\Phi}(\mathbf{X}_k)^{\mathrm{T}}\boldsymbol{\alpha}_j \tag{4}$$

Left-multiplying both sides of eq 4 by $\tilde{\Phi}(\mathbf{X}_k)^{\mathrm{T}}$, we obtain

$$\xi_j\tilde{\Phi}(\mathbf{X}_k)^{\mathrm{T}}\boldsymbol{\alpha}_j = \tilde{\Phi}(\mathbf{X}_k)^{\mathrm{T}}\,\tilde{\Phi}(\mathbf{X}_k)\,\tilde{\Phi}(\mathbf{X}_k)^{\mathrm{T}}\boldsymbol{\alpha}_j \tag{5}$$

Comparing eq 5 with eq 2 and simultaneously considering the normality constraint, it is readily derived that

$$\boldsymbol{\nu}_j = \frac{\tilde{\Phi}(\mathbf{X}_k)^{\mathrm{T}}\boldsymbol{\alpha}_j}{\sqrt{\xi_j}}$$

$$\lambda_j = \xi_j/I \tag{6}$$

From the above relationship, it is clear that $\tilde{\mathbf{K}}_k$ covers the underlying process distribution information directly related to $\mathbf{R}_k$. To solve the eigenvalue problem of eq 4 explicitly, a kernel function, $K(\mathbf{x}_i,\mathbf{x}_j) = \langle\Phi(\mathbf{x}_i),\Phi(\mathbf{x}_j)\rangle$, is introduced, which can avoid the need for both performing the nonlinear mapping and computing dot products in the feature space. According to Mercer's theorem of functional analysis,[53] there exists a mapping into a space where the kernel function acts as a dot product if the kernel function is a continuous kernel of a positive integral operator. The visual expression of $\tilde{\mathbf{K}}_k$ can thus be readily obtained based on the choice of specific kernel function. The kernel functions will be employed as the basic unit in the clustering procedure. The process duration is thus properly divided into C different nonlinear phases. The other details of the clustering procedure can be found in previous work[44,51] and will not be repeated here.

From the above analysis, the introduction of the kernel function is an important "kernel trick" through which the nonlinear relationship in input space can be readily extracted in the high-dimensional feature space by simple linear algebra. Moreover, a wide range of nonlinearities can be handled by employing different kernel functions. The representative kernel functions are as follows:[28,29]

Polynomial kernel

$$k(\mathbf{x},\mathbf{y}) = \langle\mathbf{x},\mathbf{y}\rangle^d$$

Sigmoid kernel

$$k(\mathbf{x},\mathbf{y}) = \tanh\langle\beta_0\langle\mathbf{x},\mathbf{y}\rangle + \beta_1\rangle$$

Radial basis kernel

$$k(\mathbf{x},\mathbf{y}) = \exp(-\frac{||\mathbf{x}-\mathbf{y}||^2}{g})$$

where d, β_0, β_1, and g are kernel parameters specified beforehand. The specific choice of a kernel function implicitly determines the form of mapping $\Phi(\cdot)$ and the feature space. The polynomial and radial basis kernels can always satisfy Mercer's theorem, whereas the sigmoid kernel can meet it only with certain parameter values.[52] In practice, the Gaussian kernel function is the most commonly used.[27−30,35,36,54] Generally, both kernel functions and their parameters can be set by trial and error for different cases.

2.2. Data Whitening Preprocessing and Arrangement. Before performing the ICA iteration on the data in linear cases, it is usually very useful to do some preprocessing work, that is, sphering or whitening data.[14,15] In the nonlinear KICA algorithm, generally, KPCA is used to sphere the data,[36−38] which is actually implemented in two steps: first nonlinear mapping of the initial process measurement into an invisible high-dimensional feature space and then PCA-based whitening of the mapped data into visible features. Thus, KICA modeling is actually iterative linear ICA on the KPCA-whitened data.[36−38] In previous work, KPCA-based whitening was carried out focusing on the entire batch trajectory on the basis of batchwise unfolding, as shown in Figure 1a. Such an approach, however, results in some defects as analyzed before. For linear cases, batchwise data normalization and variablewise modeling have been cleverly combined[6,40,41] and achieved successful applications. Inspired by such work, one might try to similarly prepare the batch data within each phase before embarking on kernel operation. Unfortunately, the involved kernel calculation increases polynomially with the number of data samples as indicated by the construction of kernel matrix $\mathbf{K}$ in subsection 2.1. The key is how to integrate the advantages of two unfoldings into the kernel operation.

In this study, the KPCA whitening is performed at each time point to obtain the time-slice whitened feature $\mathbf{Z}_k$, and then these features are variablewise arrayed in each phase, as shown in Figure 1b.

The KPCA time-specific whitening preprocessing is performed as follows:

In subsection 2.1, the eigenvalue solution of $\mathbf{R}_k$ is obtained based on eq 6. Here, correspondingly, the eigenvector matrix $\mathbf{V}_k$ can be briefly expressed in the form

$$\mathbf{V}_k = \tilde{\Phi}(\mathbf{X}_k)^{\mathrm{T}}\mathbf{H}_k\boldsymbol{\Lambda}_k^{-1/2} \tag{7}$$

where $\mathbf{H}_k = [\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,...,\boldsymbol{\alpha}_I]$ and $\boldsymbol{\Lambda}_k = \mathrm{diag}(\xi_1,\xi_2,...,\xi_I)$ are the eigenvector matrix and nonzero eigenvalues of $\tilde{\mathbf{K}}_k$, respectively, and $\mathbf{V}_k = [\boldsymbol{\nu}_1,\boldsymbol{\nu}_2,...,\boldsymbol{\nu}_I]$ is the eigenvector matrix of $\mathbf{R}_k$, all of which

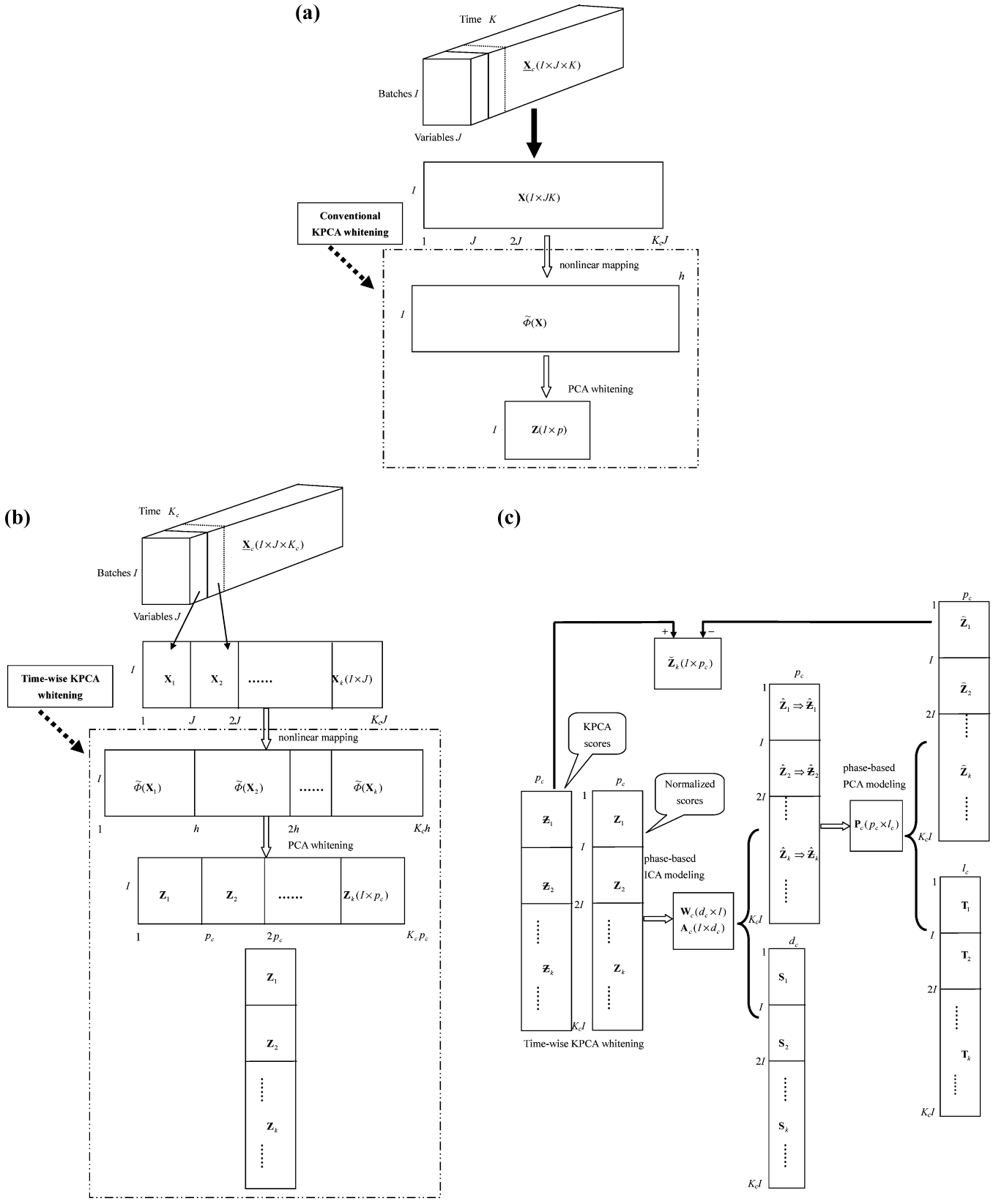

Figure 1. (a) Conventional KPCA whitening. (b) Timewise KPCA whitening. (c) Scheme of phase-based KICA−PCA two-step modeling algorithm.

were derived based on eqs 2−6. In some cases, the last few eigenvalues in $\mathbf{\Lambda}_k$ are very closely to zero. They can be excluded to avoid the singular problem in whitening preprocessing. In the present work, p_c eigenvectors were retained by unifying the dimensions of all $\mathbf{H}_k$ and $\mathbf{\Lambda}_k$ within the cth phase. According to practical experience, $I - 1$ is generally the preferred number. In the following subsections, both $\mathbf{\Lambda}_k$ and $\mathbf{H}_k$ represent the dimension-reduced matrices, and $\mathbf{V}_k$ satisfies $\mathbf{V}_k^{\mathrm{T}}\mathbf{V}_k = \mathbf{I}_{p_c}$ (where $\mathbf{I}_{p_c}$ denotes the p_c-dimensional identity matrix) at each time.

Then, the time-specific whitening models $\mathbf{Q}_k(h \times p_c)$ within the cth phase can be derived from eq 7 and expressed as follows

$$\mathbf{Q}_k = \mathbf{V}_k\left(\frac{1}{I}\mathbf{\Lambda}_k\right)^{-1/2} = \sqrt{I}\tilde{\Phi}(\mathbf{X}_k)^{\mathrm{T}}\mathbf{H}_k\mathbf{\Lambda}_k^{-1} \tag{8}$$

These models satisfy the condition

$$\mathbf{Q}_k^{\mathrm{T}}\mathbf{Q}_k = \left(\frac{1}{I}\mathbf{\Lambda}_k\right)^{-1/2}\mathbf{V}_k^{\mathrm{T}}\mathbf{V}_k\left(\frac{1}{I}\mathbf{\Lambda}_k\right)^{-1/2} = \left(\frac{1}{I}\mathbf{\Lambda}_k\right)^{-1/2}\left(\frac{1}{I}\mathbf{\Lambda}_k\right)^{-1/2} = I\mathbf{\Lambda}_k^{-1} \quad (9)$$

Thus, the mapped time-slice data in the feature space can be whitened using the transformation

$$\begin{aligned}\mathbf{Z}_k &= \tilde{\Phi}(\mathbf{X}_k)\mathbf{V}_k = \tilde{\Phi}(\mathbf{X}_k)\tilde{\Phi}(\mathbf{X}_k)^{\mathrm{T}}\mathbf{H}_k\mathbf{\Lambda}_k^{-1/2} \\ &= \tilde{\mathbf{K}}_k\mathbf{H}_k\mathbf{\Lambda}_k^{-1/2}\end{aligned}$$

$$\begin{aligned}\mathbf{Z}_k &= \tilde{\Phi}(\mathbf{X}_k)\mathbf{Q}_k = \tilde{\Phi}(\mathbf{X}_k)\mathbf{V}_k\left(\frac{1}{I}\mathbf{\Lambda}_k\right)^{-1/2} = \mathbf{Z}_k\left(\frac{1}{I}\mathbf{\Lambda}_k\right)^{-1/2} \\ &= \sqrt{I}\tilde{\mathbf{K}}_k\mathbf{H}_k\mathbf{\Lambda}_k^{-1}\end{aligned} \quad (10)$$

where $\mathbf{Z}_k(I \times p_c)$ represents the KPCA scores and $\mathbf{Z}_k(I \times p_c)$ represents their normalized forms, i.e., the whitened features. They are ordered according to the variances denoted by $(1/I)\mathbf{\Lambda}_k$, from both of which the correlations have been removed. Moreover, the model satisfies the condition $\mathrm{E}\{\mathbf{z}\mathbf{z}^{\mathrm{T}}\} = \mathbf{I}_{p_c}$, i.e., $(\mathbf{Z}_k^{\mathrm{T}}\mathbf{Z}_k)/I = \mathbf{I}_{p_c}$.

In this way, all invisible time-slice mapped data matrices, $\tilde{\Phi}(\mathbf{X}_k)$, have been transformed into visible whitened features in kernel space. Then, the phase-representative analysis unit of kernel features, $\mathbf{Z}_c(IK_c \times p_c)$ (where K_c is the number of samplings belonging to the cth phase), is steadily obtained by placing the whitened time slices $\mathbf{Z}_k$ belonging to the same phase c beneath one another. This pretreatment is a small but useful trick. It is important that the mapped data are first whitened at each time to get the visible time-slice feature $\mathbf{Z}_k$ and then arranged variablewise to form the phase-specific analysis unit. In this way, the underlying process characteristics can be revealed from an overall phase perspective and an analysis platform can be provided for the subsequent ICA−PCA two-step feature extraction. In contrast, if those batch data are first variablewise arranged and then whitened by KPCA, it is inevitable that one will be up against the calculation problem of kernel operation caused by large numbers of samples.

2.3. Phase-Based ICA−PCA Two-Step Modeling. The phase-based ICA−PCA two-step modeling procedure is shown in Figure 1c.

In the first step, the modified ICA algorithm[23] is used to extract the phase-specific demixing matrix $\mathbf{W}_c(d_c \times p_c)$ and the mixing matrix $\mathbf{A}_c(p_c \times d_c)$ (where d_c is the retained number of dominant ICs in phase c and $\mathbf{W}_c\mathbf{A}_c = \mathbf{I}_{d_c}$, where $\mathbf{I}_{d_c}$ is a d_c-dimensional identity matrix)

$$\mathbf{S}_c = \mathbf{Z}_c\mathbf{W}_c^{\mathrm{T}}$$

$$\hat{\mathbf{Z}}_c = \mathbf{Z}_c - \mathbf{S}_c\mathbf{A}_c^{\mathrm{T}} \quad (11)$$

where $\mathbf{S}_c(IK_c \times d_c)$ covers the phase-representative dominant ICs recovered from the kernel feature $\mathbf{Z}_c(IK_c \times p_c)$, which are ordered consistent with the original KPCA scores. A part of $\mathbf{Z}_c$ that is explained by the first d_c ICs can be reconstructed by $\mathbf{S}_c\mathbf{A}_c^{\mathrm{T}}$. $\hat{\mathbf{Z}}_c(IK_c \times p_c)$ is the phase-specific ICA residual of the whitened feature.

In the second step, the variance information at each time, $(1/I\mathbf{\Lambda}_k)^{1/2}$, which was used to normalize KPCA scores in whitening preprocessing, is returned to $\hat{\mathbf{Z}}_k(I \times p_c)$ by $\hat{\mathbf{Z}}_k = \hat{\mathbf{Z}}_k(1/I\mathbf{\Lambda}_k)^{1/2}$. Then the phase-specific modeling unit, $\hat{\mathbf{Z}}_c$ $(IK_c \times p_c)$, is further analyzed using PCA to reveal the Gaussian information within it

$$\mathbf{T}_c = \hat{\mathbf{Z}}_c\mathbf{P}_c$$

$$\widehat{\mathbf{Z}}_c = \hat{\mathbf{Z}}_c - \mathbf{T}_c\mathbf{P}_c^{\mathrm{T}} \quad (12)$$

where $\mathbf{T}_c(IK_c \times l_c)$ reveals the obtained phase-representative Gaussian PCs. $\mathbf{T}_c\mathbf{P}_c^{\mathrm{T}}$ reconstructs the part of $\hat{\mathbf{Z}}_c$ $(IK_c \times p_c)$ explained by the first l_c PCs. $\widehat{\mathbf{Z}}_c(IK_c \times p_c)$ is the phase-specific PCA residual of whitened feature, which is actually the final residual of the whitened feature after ICA−PCA two-step extraction. Moreover, the time-slice PCs and PCA residual, $\mathbf{T}_k(I \times l_c)$ and $\widehat{\mathbf{Z}}_k(I \times p_c)$, can be respectively separated from $\mathbf{T}_c$ and $\widehat{\mathbf{Z}}_c$.

Therefore, the total variations captured by ICA−PCA two-step models are derived at each time as

$$\check{\mathbf{Z}}_k = \mathbf{Z}_k - \widehat{\mathbf{Z}}_k = \mathbf{Z}_k\left(\frac{1}{I}\mathbf{\Lambda}_k\right)^{1/2} - \widehat{\mathbf{Z}}_k \quad (13)$$

This model can be inversely whitened, reconstructing the mapped data pattern at each time as follows

$$\check{\Phi}(\mathbf{X}_k) = \check{\mathbf{Z}}_k\mathbf{V}_k^{\mathrm{T}} \quad (14)$$

Compared with the initial mapped feature $\tilde{\Phi}(\mathbf{X}_k)$, the final residual unoccupied by ICA−PCA models, which actually involves the PCA residual and the losing part during the whitening procedure, can be expressed as

$$\tilde{\Phi}^{\mathrm{e}}(\mathbf{X}_k) = \tilde{\Phi}(\mathbf{X}_k) - \check{\Phi}(\mathbf{X}_k) \quad (15)$$

From the above modeling procedure, it is clear that KICA−PCA modeling is actually linear ICA−PCA performed on KPCA-transformed whitened data. Moreover, the retained number of ICs−PCs has a close relationship to the identification of how much process information is to be explained by ICA and how much by PCA. The monitoring performance will be best if both types of underlying information are properly explored. Based on such an awareness, in the present work, the proper number of components is determined by evaluating their effects on monitoring performance using cross-validation. The monitoring performance is checked by the overall type I error (OTI),[6,7,50] which is the proportion of false alarms in the normal duration. When an OTI value as close as possible to the significance level (here 1% is imposed) is obtained, the corresponding number of ICs−PCs is determined.

2.4. Monitoring Statistics and Confidence Limits. Two types of statistics are calculated: the D^2 statistic for the systematic part of the process variations and the Q statistic for the residual part.

For systematic subspaces of ICA and PCA respectively:

$$D_{i,k}^{\ 2} = (\mathbf{s}_{i,k} - \bar{\mathbf{s}}_k)^{\mathrm{T}}\mathbf{O}_k^{-1}(\mathbf{s}_{i,k} - \bar{\mathbf{s}}_k) = \mathbf{s}_{i,k}^{\ \mathrm{T}}\mathbf{s}_{i,k}$$

$$T_{i,k}^{\ 2} = (\mathbf{t}_{i,k} - \bar{\mathbf{t}}_k)^{\mathrm{T}}\hat{\mathbf{O}}_k^{-1}(\mathbf{t}_{i,k} - \bar{\mathbf{t}}_k) = \mathbf{t}_{i,k}^{\ \mathrm{T}}\hat{\mathbf{O}}_k^{-1}\mathbf{t}_{i,k} \quad (16)$$

where $\mathbf{s}_{i,k}(d_c \times 1)$ and $\mathbf{t}_{i,k}(l_c \times 1)$ are the independent component vector and principal component vector, respectively, of the ith modeling batch at time k, and $\bar{\mathbf{s}}_k(d_c \times 1)$ and $\bar{\mathbf{t}}_k(l_c \times 1)$ denote the mean vectors of time-slice component matrices, $\mathbf{S}_k(I \times d_c)$ and $\mathbf{T}_k(I \times l_c)$, respectively, which actually are both zero vectors. $\mathbf{O}_k(d_c \times d_c)$ and $\hat{\mathbf{O}}_k(l_c \times l_c)$ are the time-varying covariance matrices calculated from $\mathbf{S}_k$ and $\mathbf{T}_k$, respectively. Here, $\mathbf{O}_k$ actually is an identity matrix due to the timewise data whitening. By means of $\hat{\mathbf{O}}_k$, the dynamics of the PCs are taken into the T^2-statistic monitoring chart.

For the error subspace, the squared prediction error (SPE), shows those process deviations from the expected trajectory that are not captured by the ICA−PCA models

$$\begin{aligned}\mathrm{SPE}_{i,k} &= \tilde{\Phi}^{\mathrm{e}}(\mathbf{x}_{i,k})^{\mathrm{T}}\tilde{\Phi}^{\mathrm{e}}(\mathbf{x}_{i,k}) \\ &= [\tilde{\Phi}(\mathbf{x}_{i,k}) - \breve{\Phi}(\mathbf{x}_{i,k})]^{\mathrm{T}}[\tilde{\Phi}(\mathbf{x}_{i,k}) - \breve{\Phi}(\mathbf{x}_{i,k})] \\ &= [\tilde{\Phi}(\mathbf{x}_{i,k})^{\mathrm{T}} - \breve{\mathbf{z}}_{i,k}{}^{\mathrm{T}}\mathbf{V}_k{}^{\mathrm{T}}][\tilde{\Phi}(\mathbf{x}_{i,k}) - \mathbf{V}_k\breve{\mathbf{z}}_{i,k}] \\ &= \tilde{k}(\mathbf{x}_{i,k}, \mathbf{x}_{i,k}) - 2\tilde{\Phi}(\mathbf{x}_{i,k})^{\mathrm{T}}\mathbf{V}_k\breve{\mathbf{z}}_{i,k} + \breve{\mathbf{z}}_{i,k}{}^{\mathrm{T}}\mathbf{V}_k{}^{\mathrm{T}}\mathbf{V}_k\breve{\mathbf{z}}_{i,k} \\ &= \tilde{k}(\mathbf{x}_{i,k}, \mathbf{x}_{i,k}) - 2\mathbf{z}_{i,k}{}^{\mathrm{T}}\breve{\mathbf{z}}_{i,k} + \breve{\mathbf{z}}_{i,k}{}^{\mathrm{T}}\breve{\mathbf{z}}_{i,k} \end{aligned} \quad (17)$$

This calculation shows that, although the final residual calculated in eq 15, $\tilde{\Phi}^e(\mathbf{X}_k)$, is not explicitly known, the SPE monitoring statistic can be readily and explicitly determined.

In the KPCA monitoring system, the premise of a Gaussian distribution provides an important basis for deriving the confidence limits of monitoring statistical criteria. Under such an assumption, the control limit in the systematic subspace at each time can be obtained from the indicated F distribution[7,55] with α significance factor according to the expression

$$T_k{}^2 \approx \frac{l_c(I^2 - 1)}{I(I - l_c)}F_{l_c, I-l_c, \alpha} \quad (18)$$

In the error subspace, the confidence limit for the SPE at each time can be approximated by a weighted χ^2 distribution[7]

$$\mathrm{SPE}_k \approx g_k\chi_{h^k,\alpha}{}^2 \quad (19)$$

where the distribution parameters at time k are estimated from $\mathrm{SPE}_k = [\mathrm{SPE}_{1,k}, \mathrm{SPE}_{2,k}, ..., \mathrm{SPE}_{I,k}]$; $g_k = \nu^k/2m^k$; and $h^k = 2(m^k)^2/\nu^k$, where m^k is the average of the SPE_k values and ν^k is the corresponding variance.

In KICA, the restriction of the non-Gaussianity of the ICs makes it impossible to calculate the control limit of the D^2 statistic simlarly to the case of the of Hotelling T^2 distribution in KPCA. An alternative approach based on nonparametric density estimation[56,57] is employed in the present work to define the normal coverage.

2.5. Online Monitoring. When a new batch sample, $\mathbf{x}_{\mathrm{new}}(J \times 1)$, is available, it is first normalized. Subsequently, its nonlinear mapping, $\tilde{\Phi}(\mathbf{x}_{\mathrm{new}})$, is preprocessed and whitened

$$\mathbf{z}_{\mathrm{new}}{}^{\mathrm{T}} = \tilde{\Phi}(\mathbf{x}_{\mathrm{new}})^{\mathrm{T}}\mathbf{V}_k = \tilde{\mathbf{k}}_{\mathrm{new},k}\mathbf{H}_k\mathbf{\Lambda}_k{}^{-1/2}$$

$$\mathbf{z}_{\mathrm{new}}{}^{\mathrm{T}} = \tilde{\Phi}(\mathbf{x}_{\mathrm{new}})^{\mathrm{T}}\mathbf{Q}_k = \mathbf{z}_{\mathrm{new}}{}^{\mathrm{T}}\left(\frac{1}{I}\mathbf{\Lambda}_k\right)^{-1/2} = \sqrt{I}\tilde{\mathbf{k}}_{\mathrm{new},k}\mathbf{H}_k\mathbf{\Lambda}_k{}^{-1} \quad (20)$$

where $\mathbf{z}_{\mathrm{new}}$ is the KPCA score and $\mathbf{z}_{\mathrm{new}}$ is its normalized form, i.e., the whitened feature vector.

Consequently, $\mathbf{z}_{\mathrm{new}}$ will be projected onto the subphase ICA−PCA model into which the current sample falls. In the ICA monitoring system, the ICs and the ICA residual of the whitened feature can be computed as

$$\mathbf{s}_{\mathrm{new}} = \mathbf{W}_c\mathbf{z}_{\mathrm{new}}$$

$$\hat{\mathbf{z}}_{\mathrm{new}} = \mathbf{z}_{\mathrm{new}} - \mathbf{A}_c\mathbf{s}_{\mathrm{new}} \quad (21)$$

In the PCA monitoring system, the PCs and the PCA residual are respectively calculated as

$$\hat{\mathbf{z}}_{\mathrm{new}} = \left(\frac{1}{I}\mathbf{\Lambda}_k\right)^{1/2}\hat{\mathbf{z}}_{\mathrm{new}}$$

$$\mathbf{t}_{\mathrm{new}}{}^{\mathrm{T}} = \hat{\mathbf{z}}_{\mathrm{new}}{}^{\mathrm{T}}\mathbf{P}_c$$

$$\widehat{\mathbf{z}}_{\mathrm{new}} = \hat{\mathbf{z}}_{\mathrm{new}} - \mathbf{P}_c\mathbf{t}_{\mathrm{new}} \quad (22)$$

Then, the final residual unoccupied by the KICA−PCA two-step model, $\tilde{\Phi}^e(\mathbf{x}_{\mathrm{new}})$, can be reconstructed at each time as

$$\breve{\mathbf{z}}_{\mathrm{new}} = \mathbf{z}_{\mathrm{new}} - \widehat{\mathbf{z}}_{\mathrm{new}}$$

$$\breve{\Phi}(\mathbf{x}_{\mathrm{new}}) = \mathbf{V}_k\breve{\mathbf{z}}_{\mathrm{new}}$$

$$\tilde{\Phi}^{\mathrm{e}}(\mathbf{x}_{\mathrm{new}}) = \tilde{\Phi}(\mathbf{x}_{\mathrm{new}}) - \breve{\Phi}(\mathbf{x}_{\mathrm{new}}) \quad (23)$$

Finally, the monitoring statistics of new sample can be readily calculated as

$$D_{\mathrm{new}}{}^2 = \mathbf{s}_{\mathrm{new}}{}^{\mathrm{T}}\mathbf{s}_{\mathrm{new}}$$

$$T_{\mathrm{new}}{}^2 = \mathbf{t}_{\mathrm{new}}{}^{\mathrm{T}}(\hat{\mathbf{O}}_k)^{-1}\mathbf{t}_{\mathrm{new}}$$

$$\begin{aligned}\mathrm{SPE}_{\mathrm{new}} &= \tilde{\Phi}^{\mathrm{e}}(\mathbf{x}_{\mathrm{new}})^{\mathrm{T}}\tilde{\Phi}^{\mathrm{e}}(\mathbf{x}_{\mathrm{new}}) \\ &= [\tilde{\Phi}(\mathbf{x}_{\mathrm{new}})^{\mathrm{T}} - \breve{\mathbf{z}}_{\mathrm{new}}{}^{\mathrm{T}}\mathbf{V}_k{}^{\mathrm{T}}][\tilde{\Phi}(\mathbf{x}_{\mathrm{new}}) - \mathbf{V}_k\breve{\mathbf{z}}_{\mathrm{new}}] \\ &= \tilde{k}(\mathbf{x}_{\mathrm{new}}, \mathbf{x}_{\mathrm{new}}) - 2\mathbf{z}_{\mathrm{new}}{}^{\mathrm{T}}\breve{\mathbf{z}}_{\mathrm{new}} + \breve{\mathbf{z}}_{\mathrm{new}}{}^{\mathrm{T}}\breve{\mathbf{z}}_{\mathrm{new}} \end{aligned} \quad (24)$$

Process monitoring is thus consecutively conducted online by judging whether the statistic indices are exceeding their normal limits.

3. Illustration and Discussion

3.1. Three-Tank System. A three-tank system, as shown in Figure 2, represents a typical nonlinear multiphase batch process in which five process variables, including two float inputs (F1 and F2) and three liquid levels (L1, L2, and L3), are measured every second. During the process, two levels, L1 and L2, are brought from their initial conditions to the set points (L1 = 300 mm and L2 = 200 mm), whereas L3 is left to float to reflect the interaction between tanks 1 and 2. The rising water levels

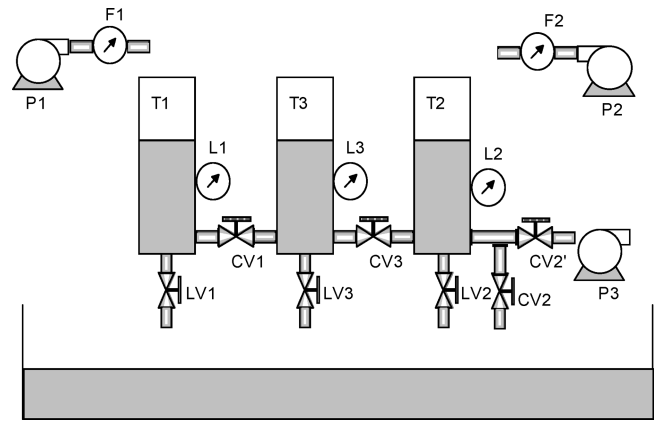

Figure 2. Scheme of three-tank system.

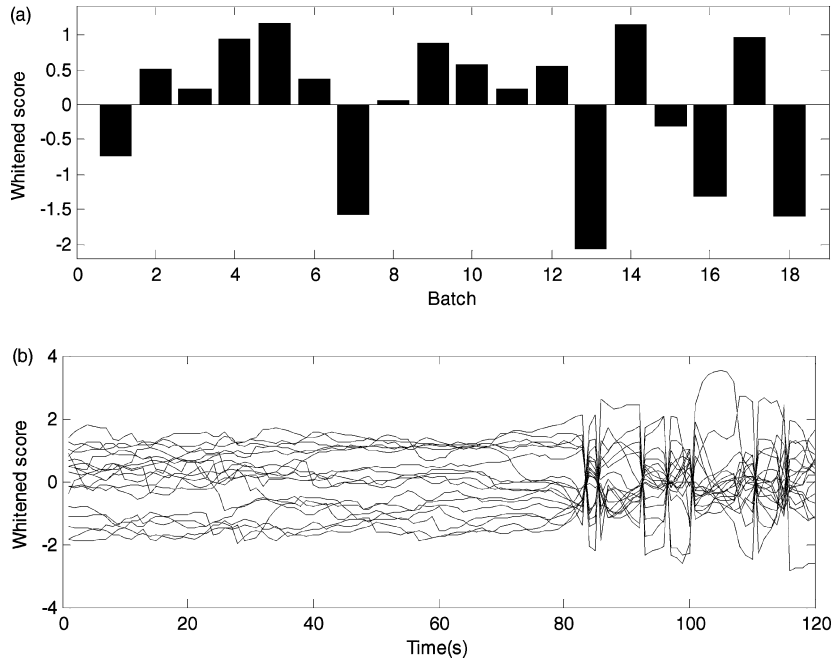

Figure 3. First whitened scores derived using (a) conventional KPCA whitening and (b) timewise KPCA whitening for 18 batches.

result in a time-varying process dynamics, and the closed-loop control of the liquid levels generates strong coupling relationships between the process variables. The process finishes after the three levels stabilize over a period of time. In this time, 120 points of historical data collected in each batch and 18 normal experiments carried out under the same conditions generate the modeling data matrix $\underline{\mathbf{X}}(18 \times 5 \times 120)$.

First, nonlinear mapping is applied to the normalized time-slice data sets, generating 120 kernel matrices $\tilde{\mathbf{K}}_k(18 \times 18)$, in which the radial basis kernel is employed to build the kernel space with kernel parameter c empirically chosen as 170. Five phases are obtained by clustering in the kernel space: phase I (1−33), in which manipulated variables F1 and F2 are both at saturation and three levels are increasing steadily; phase II (34−68), in which F1 is at saturation while F2 starts decreasing, and three levels keep increasing; phase III (69−80), in which both manipulated variables are decreasing, whereas all levels continue increasing; phase IV (81−97), in which both the decreasing trend of float inputs and the increasing of the water levels gradually become slow; phase V (98−120), in which all five variables tend to level off after mild fluctuations and, finally, L1 and L2 reach their set points. The phase localization agrees well with the changes in the underlying characteristics as a function of time.

Then, the timewise data whitening preprocessing is analyzed in comparison with the conventional batchwise whitening procedure.[36] The latter treats the whole batch trajectory as one object, meaning that only one whitened vector can be obtained for each batch. Taking as an example the first normalized KPCA component, the result is shown in Figure 3a. In contrast, the proposed timewise whitening preprocessing can reveal the time-varying trajectories of whitened features for 18 batches as shown in Figure 3b, which can provide more abundant process information. The phase-based analysis unit is thus arranged by variablewise stacking ofthe time-varying whitened vectors within the same phase. Figure 4 shows the graphical normality checking results respectively for phase-representative initial process variables, whitened scores, and ICA residuals, respectively. A normal probability plot is a useful means for graphical normality testing. If the data do come from a normal distribution, the plot

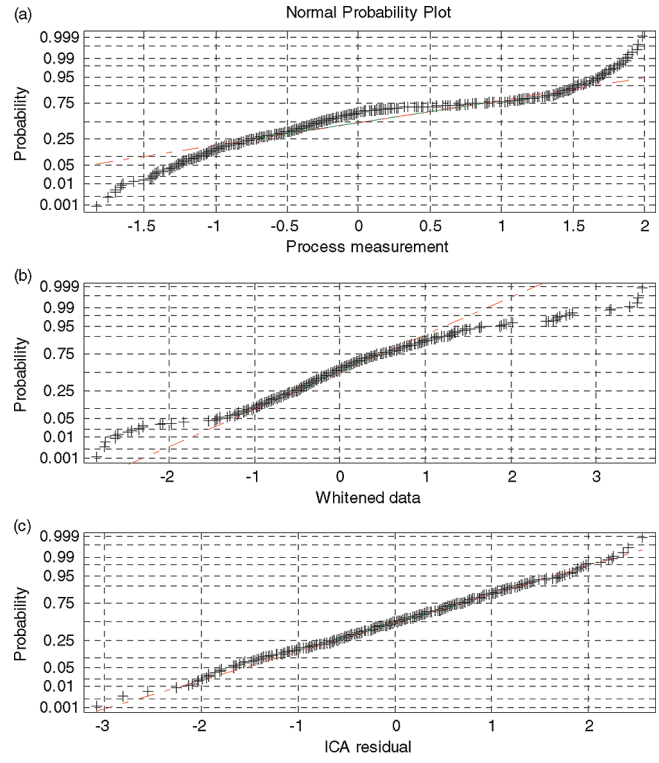

Figure 4. Normal probability comparison and check for (a) the initial process measurement, (b) the whitened data, and (c) the residual of whitening data after ICA.

will appear linear; otherwise, curvature will be introduced into the plot. Further, Table 1 shows the quantitative evaluation results of the whitened scores, ICs, and ICA residual in each phase using the Negentropy-based non-Gaussianity test. In the ICA algorithm, the model structure is determined by maximizing the non-Gaussianity of the extracted features,[15] which can be approximately measured by $J(s) \approx \{E[G(s)] - E[G(v)]\}^2$ [where the Gaussian variable v has the largest entropy among all random variables with equal variance, and $G(\cdot)$ is a nonquadratic

Table 2. Description of Fault Types Introduced into Three-Tank System

fault no.	fault type	description of disturbance	occurrence (samples)
1	leakage of tank 1	turn on valve LV1 intentionally	0
2	leakage of tanks 2 and 3	turn on valves LV2 and LV3 synchronously	45

function]. According to Hyvärinen and Oja,[15] the following choices of G(•) have proved very useful

$$G_1(s) = \frac{1}{a_1}\log[\cosh(a_1 s)] \qquad G_2(s) = -\exp\left(-\frac{s^2}{2}\right) \tag{25}$$

where constant a_1 satisfies $1 \le a_1 \le 2$. $G_1(s)$ is deemed to be a good general-purpose contrast function[14] and will be used here.

Therefore, in the present work, $J(s)$ is employed to perform the Negentropy-based non-Gaussianity test. The larger the value of $J(s)$, the greater the non-Gaussianity of variable s. Upon comparison of the calculation results in Table 1, it is clear that, after the first-step ICA extraction, the data will approach Gaussianity more obviously and then can be explored by PCA.

Two fault cases are manually introduced as described in Table 2. The first fault, a leak in tank 1 at the beginning, influences the process status of tank 2 indirectly because of their correlations through tank 3, as shown in Figure 2. For the second fault, namely, leaks in both tanks 2 and 3, undesirable disturbances will first be directly imposed on tank 1 due to the fault of tank 3, and further, the abnormal behavior of the tank 2 fault will also influence tank 1 through the communicating function of tank 3. For one normal case and the two abnormalities investigated in this illustration, the process trajectories are illustrated in Figure 5. Figure 6 shows the online monitoring results for the successful batch. From this plot, it can be seen that all monitoring indices stay well within the control bounds, which jointly indicate that the batch behaves free of any process abnormality throughout the duration. The monitoring results with respect to the two faults are respectively shown in Figures 7a and 8a. Both upsets can be quickly picked up by synthetically analyzing the monitoring charts, where the statistic values obviously go beyond the control limits almost immediately once the faults occur. Therefore, the monitoring models can effectively and in real time capture the abnormal variation of nonlinear process behaviors with no data estimation cost. Moreover, MPCA with variablewise unfolding is employed to develop the statistical model, and the monitoring results for the first fault are comparatively shown in Figure 7b, which yields an obvious detection delay. For fault 2, the monitoring results using conventional MPCA, MICA, multiway KPCA,[29] and multiway KICA[36] are shown in Figure 8b−e, respectively. Comparatively, the monitoring performance of the proposed method as shown in Figure 8a exhibits superior results. Using only multiway KICA, more ICs might be required to explain the systematic variations and therefore some Gaussian information may be mistakenly expressed as ICs. Moreover, some systematic information may not be extracted and thus left in the residuals. In contrast, for the proposed method, the second-step KPCA feature extraction can help to ease the burden of KICA when capturing the system variations. Comparing the results shown in Figure 8a and e, it can be seen that the KICA model in the proposed method can detect the abnormality faster than single KICA model. Moreover, after the second-step PCA feature extraction, the Gaussian PCs are extracted, and thus the SPE monitoring chart in Figure 8a can detect the fault earlier (at the 47th sampling time) than the case (alarm indication at the 52nd sampling time) shown in Figure 8e. These results provide a preliminary demonstration of the swiftness and effectiveness of the proposed method for process monitoring.

Table 1. Negentropy-Based Non-Gaussianity Test for Three-Tank System

phase no.	whitened scores		ICs		ICA residual	
	score 1	score 2	IC1	IC2	residual 1	residual 2
I	0.0010	0.0006	0.0726	0.0006	0.0005	0.0003
II	0.0014	0.00000006	0.1226	0.0012	0.0007	0.000023
III	0.0013	0.0015	0.0517	0.0165	0.0011	0.0010
IV	0.0006	0.0004	0.0323	0.0253	0.0000003	0.0001
V	0.0008	0.000024	0.0416	0.0019	0.0006	0.0000002

3.2. Fed-Batch Penicillin Fermentation Process. In this section, the proposed method is applied to a well-known benchmark process, fed-batch penicillin fermentation,[58,59] which has obvious nonlinear dynamics and multiphase characteristics. In a typical fed-batch fermentation process, most of the necessary cell mass is obtained during the initial preculture

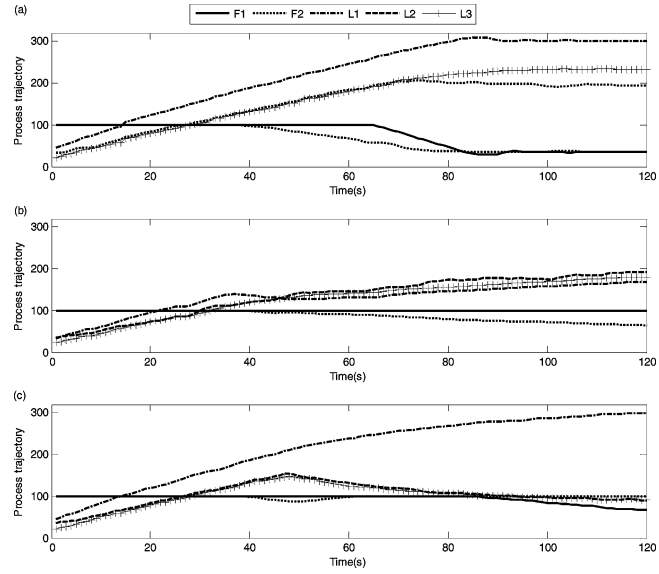

Figure 5. Process trajectories in the case of (a) normal process, (b) fault 1, and (c) fault 2.

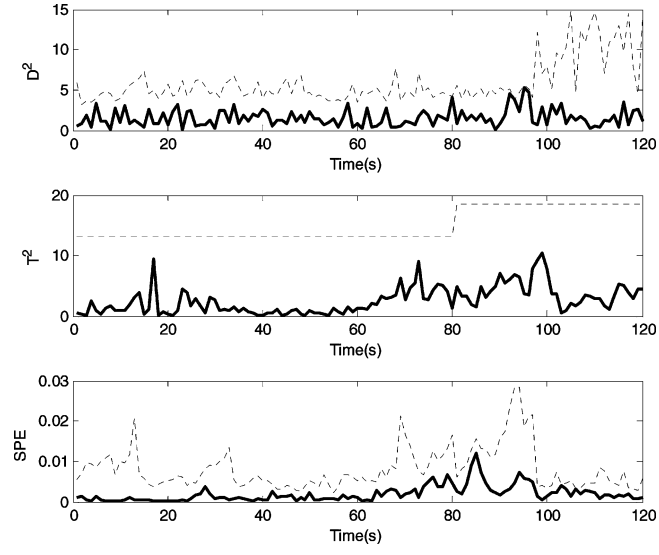

Figure 6. Monitoring charts for a normal batch using the proposed algorithm (dashed line, 99% control limit; bold solid line, online monitoring statistics).

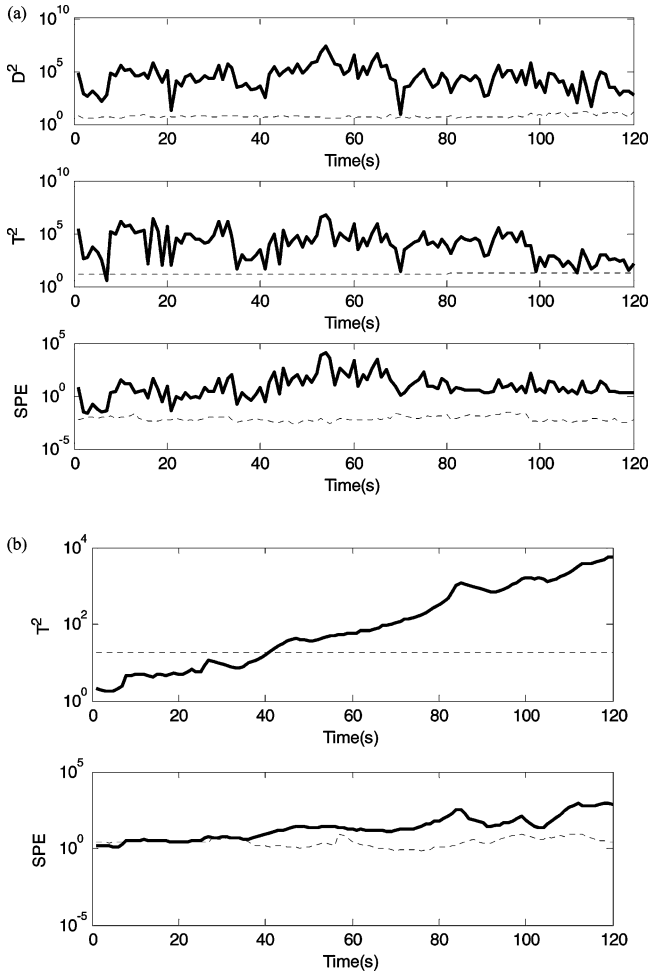

Figure 7. Monitoring plots of fault 1 using (a) the proposed algorithm and (b) MPCA with variablewise unfolding (dashed line, 99% control limit; bold solid line, online monitoring statistics).

phase. When most of the initially added substrate has been consumed by the microorganisms, the substrate feed begins. The penicillin starts to be generated at the exponential growth phase and continues to be produced until the stationary phase. A low substrate concentration in the fermentor is necessary for achieving a high product formation rate because of the catabolite repressor effect. Consequently, glucose is fed continuously during fermentation instead of being added one-off at the beginning.

In the present simulation experiment, a total of 30 reference batches are generated using a simulator (PenSim v1.0 simulator) developed by the monitoring and control group of the Illinois Institute of Technology. These simulations are run under closed-loop control of pH and temperature, while glucose addition is performed open-loop. Small variations are automatically added by the simulator to mimic the real normal operating conditions. The process variables selected for modeling in this work are listed in Table 3. With 1 h as the sampling interval, 400 points of 11 process variables collected in each batch and 30 normal experiments carried out under the same default initial conditions yield the reference modeling array $\underline{\mathbf{X}}(30 \times 11 \times 400)$.

According to the modeling procedure, the first 400 normalized time slices $\mathbf{X}_k(I \times J)$ are obtained from $\underline{\mathbf{X}}(30 \times 11 \times 400)$. In this case, the radial basis kernel function is employed to construct the kernel space, where the kernel parameter c is empirically chosen as 371. By the kernel trick, 400 kernel matrices $\tilde{\mathbf{K}}_k$ are generated, which are fed to the clustering

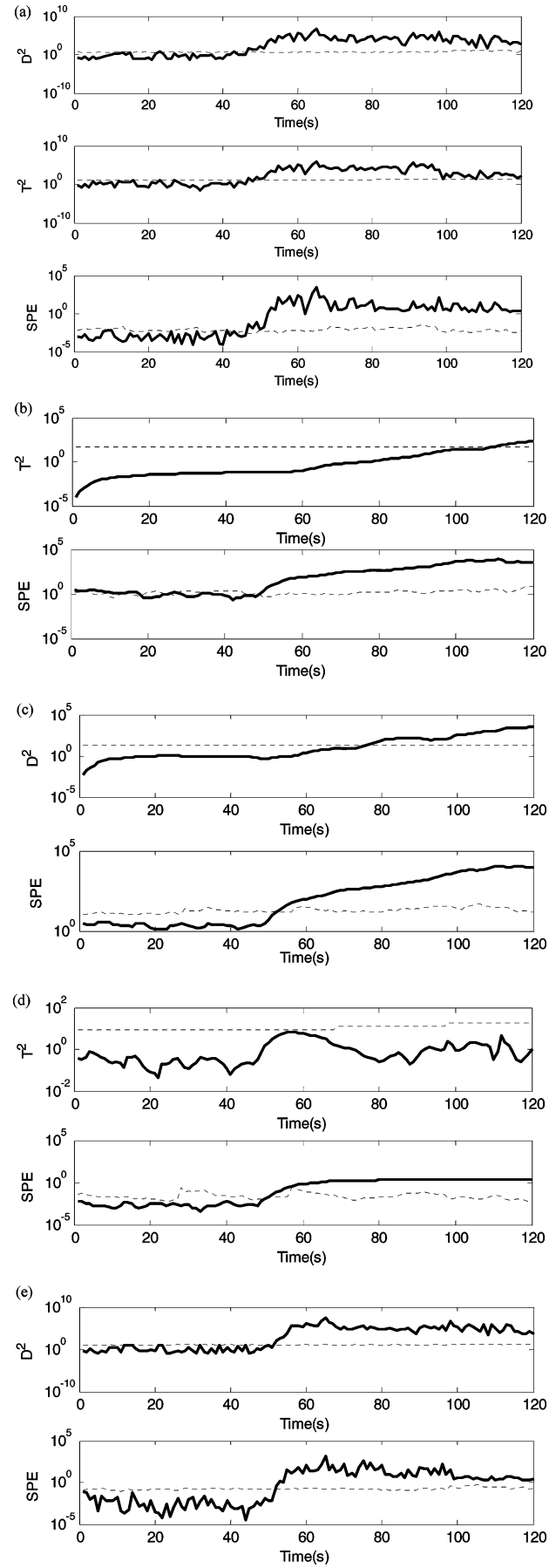

Figure 8. Monitoring plots of fault 2 using (a) the proposed method, (b) MPCA, (c) MICA, (d) multiway KPCA, and (e) multiway KICA (dashed line, 99% control limit; bold solid line, online monitoring statistics).

Table 3. Process Variables Used in the Modeling of the Benchmark Case

no.	process variable
1	aeration rate (L/h)
2	agitator power (W)
3	substrate feed rate (L/h)
4	substrate feed temperature (K)
5	dissolved oxygen concentration (g/L)
6	culture volume (L)
7	carbon dioxide concentration (g/L)
8	pH
9	fermentor temperature (K)
10	generated heat (kcal)
11	cooling water flow rate (L/h)

Table 4. Negentropy-Based Non-Gaussianity Test for Penicillin Fermentation Process

phase no.	whitened scores		ICs		ICA residual	
	score 1	score 2	IC1	IC2	residual 1	residual 2
I	0.0003	0.0004	0.0633	0.0326	0.0002	0.000048
II	0.0006	0.0003	0.0034	0.0007	0.000020	0.000019
III	0.0020	0.0004	0.0351	0.0005	0.000027	0.0001
IV	0.0001	0.0001	0.0344	0.0002	0.0001	0.0001
V	0.000006	0.000049	0.0038	0.0002	0.0000005	0.000002

Table 5. Summary of Fault Types Introduced at Different Times of Fermentation

fault no.	fault type	occurrence (h)
1	0.003 ramp decrease in substrate feed rate	50
2	15% step decrease in agitator power	20
3	pH controller failure	0

algorithm introduced in subsection 2.1, so that the whole process is generally divided into five phases (1−44, 45−70, 71−112, 113−322, and 323−400). The phase-clustering result might not be exactly the same as the real operation phase in the fermentation cultivation, but it places more emphasis on the changes of underlying nonlinear correlations along time direction.

First, Table 4 shows the results of the Negentropy-based non-Gaussianity test, which quantitatively evaluates the effects of ICA on non-Gaussian data. In addition to 20 normal batches, 20 fault batches are generated under the three abnormal conditions (faults 1−3) presented in Table 5. Figure 9 shows the online monitoring results for one successful batch, which reveals the success of the current batch given that the two-step monitoring systems do not give any false alarm. For the first

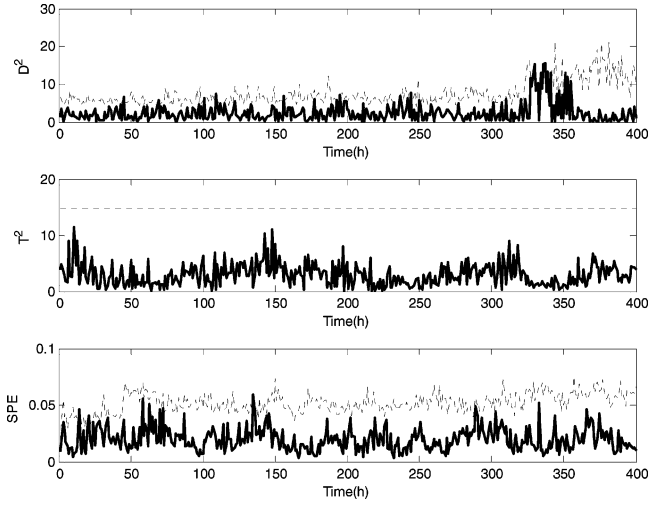

Figure 9. Monitoring charts for a normal batch using the proposed algorithm (dashed line, 99% control limit; bold solid line, online monitoring statistics).

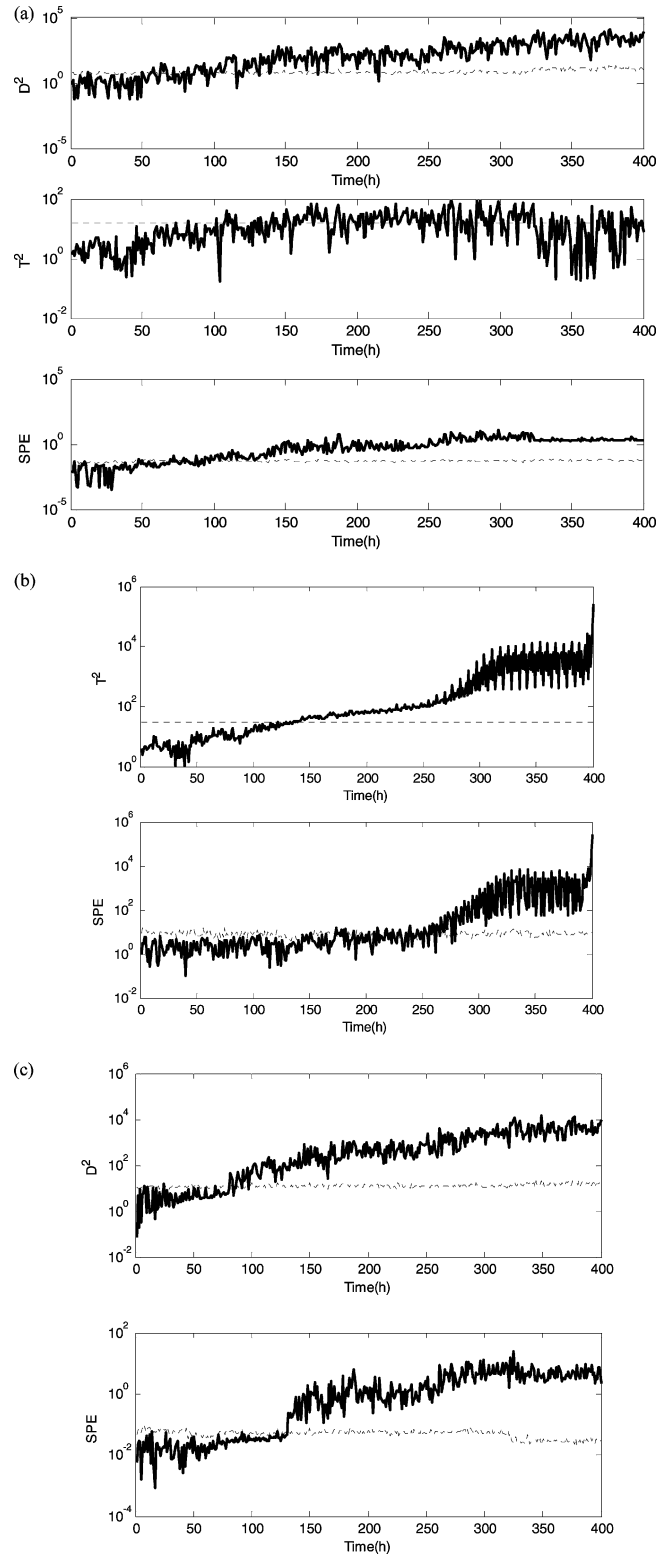

Figure 10. Monitoring plots of fault 1 using (a) the proposed algorithm, (b) MPCA with variablewise unfolding, and (c) multiway KICA (dashed line, 99% control limit; bold solid line, online monitoring statistics).

fault, a ramp decrease is imposed on the batch process, i.e., the substrate feed rate is decreased with a slope of 0.003 from 50 h to the end of process operation. A decrease in the substrate feed rate results in a reduction of the penicillin concentration because the substrate is the main carbon source to be fed during the fermentation,[57] although the process variation itself is slight. As shown in Figure 10a, the first-step KICA monitoring system yields out-of-control signals immediately when the process fault

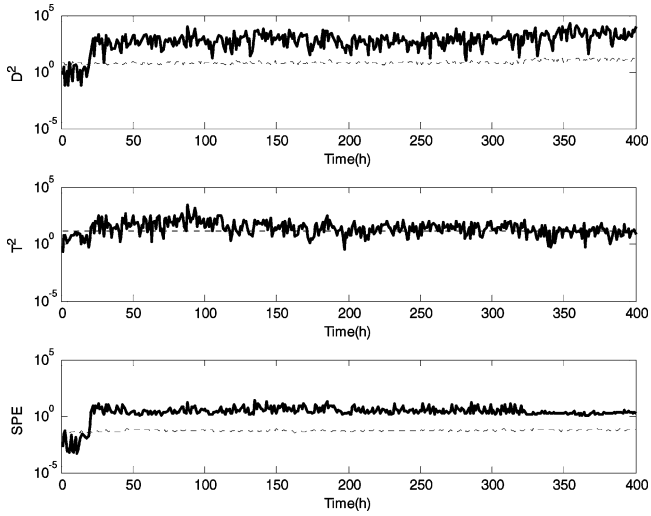

Figure 11. Monitoring plots using the proposed algorithm in the case of fault 2 (dashed line, 99% control limit; bold solid line, online monitoring statistics).

Table 6. Comparison Results of Monitoring Performance (Using 99% Control Limit)

method	performance indices	
	false alarm rate (%)	fault detection delay (h)
conventional KPCA	0.0405	32
		7
		12
conventional KICA	0.0410	25
		3
		10
proposed KICA−PCA	0.0130	13
		0
		0

occurs and continues until the end of process operation. The T^2 statistic values of the second-step KPCA monitoring system basically do not obviously escape from the control limits. This phenomenon might arise from the fact that, after KICA extraction, the impact of process disorder has been partly removed so that the systematic variation of KPCA is not corrupted as seriously as in the KICA system. On the other hand, this might indicate that the nonconformity mainly disturbs the non-Gaussian part of process operation. Because the process characteristics have been separated into two types and been subjected to different statistical analyses, correspondingly, the different influences of process disturbance are unveiled in the two monitoring systems. Comparatively, the monitoring results using MPCA with variablewise unfolding and multiway KICA which both yield fault detection delay, are shown in Figure 10b,c, respectively. Comparing Figure 10a and c, it can be seen that, without PCA modeling, the SPE monitoring chart does not give alarms until approaching 130 h as the major Gaussian information are not extracted. Therefore, the second-step PCA feature extraction is beneficial to the improvement of monitoring performance. For the second fault, a step change is imposed on the agitator power. The KICA−PCA monitoring charts make an instant response to the process abnormality as shown in Figure 11.

Further, Table 6 lists the quantitative comparison results among the proposed method, multiway KPCA,[29] and multiway KICA.[36] Two performance indices are used to evaluate how well each approach performs, namely, the false alarm rate (%) and the fault detection delay (h), in which the fault detection delay is checked for each introduced fault listed in Table 5. Compared with conventional multiway KPCA or multiway KICA, the proposed method shows similar monitoring performance and better results under some situations. In additino to its comprehensive analyses of both types of process behaviors, the better performance might also benefit from the identification of phase-specific characteristics and the introduction of time-varying covariances.

All of the above simulation results have effectively demonstrated that, based on the phase-specific KICA−PCA two-step modeling method, one can easily track the progress of each batch run without missing data estimation and detect the occurrence of the faults promptly.

4. Conclusions

In this article, nonlinear batch process monitoring methods based on kernel technique are improved by developing a phase-based KICA−PCA two-step modeling algorithm. Compared with conventional multiway kernel methods, the proposed algorithm considers the local phase characteristics, explores both Gaussian and non-Gaussian nonlinear information, reveals the time-varying dynamics, and can be readily put into online application without data estimation. In conclusion, the present approach provides a promising attempt to extend and improve kernel applications. Considerable further research is suggested.

Acknowledgment

Many thanks are due to Derya Tetiker in the Department of Chemical and Biological Engineering, Illinois Institute of Technology, who gave the authors kind help in using PenSim v1.0 simulator. This work was supported in part by the Hong Kong Research Grant Council under Project 613107, the China National 973 program (2009CB320601), and the National Natural Science Foundation of China (No. 60774068).

Literature Cited

(1) Kourti, T.; MacGregor, J. F. Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemom. Intell. Lab. Syst.* **1995**, *28*, 3.

(2) Kosanovich, K. A.; Dahl, K. S.; Piovoso, M. J. Improved process understanding using multiway principal component analysis. *Ind. Eng. Chem. Res.* **1996**, *35*, 138.

(3) Martin, E. B.; Morris, A. J.; Papazoglou, M. C.; Kiparissides, C. Batch process monitoring for consistent production. *Comput. Chem. Eng.* **1996**, *20*, 599.

(4) Louwerse, D. J.; Smilde, A. K. Multivariate statistical process control of batch processes based on three-way models. *Chem. Eng. Sci.* **2000**, *55*, 1225.

(5) Wold, S.; Sjöström, M. Chemometrics, present and future success. *Chemom. Intell. Lab. Syst.* **1998**, *44*, 3.

(6) van Sprang, E. N. M.; Ramaker, H.-J.; Westerhuis, J. A.; Gurden, S. P.; Smilde, A. K. Critical evaluation of approaches for on-line batch process monitoring. *Chem. Eng. Sci.* **2002**, *57*, 3979.

(7) Nomikos, P.; MacGregor, J. F. Multivariate SPC charts for monitoring batch processes. *Technometrics* **1995**, *37*, 41.

(8) Nomikos, P.; MacGregor, J. F. Multiway partial least squares in monitoring batch processes. *Chemom. Intell. Lab. Syst.* **1995**, *30*, 97.

(9) Wise, B. M.; Gallagher, N. B.; Butler, S. W.; White, D. D., Jr.; Barna, G. G. A comparison of principal component analysis, multiway principal components analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process. *J. Chemom.* **1999**, *13*, 379.

(10) Ündey, C.; Tatara, E.; Çinar, A. Intelligent real-time performance monitoring and quality prediction for batch/fed-batch cultivations. *J. Biotechnol.* **2004**, *108*, 61.

(11) Martin, E. B.; Morris, A. J. Enhanced bio-manufacturing through advanced multivariate statistical technologies. *J. Biotechnol.* **2002**, *99*, 223.

(12) Wong, C. W. L.; Escott, R.; Martin, E. B.; Morris, A. J. The integration of spectroscopic and process data for enhanced process performance monitoring. *Can. J. Chem. Eng.* **2008**, *86*, 905.

(13) Comon, P. Independent component analysis, a new concept. *Signal Process.* **1994**, *36*, 287.

(14) Hyvärinen, A. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. Neural Networks* **1999**, *10*, 626.

(15) Hyvärinen, A.; Oja, E. Independent component analysis: algorithms and applications. *Neural Networks* **2000**, *13*, 411.

(16) Ahn, H. C.; Choi, E. S.; Han, I. G. Extracting underlying meaningful features and canceling noise using independent component analysis for direct marketing. *Expert Syst. Appl.* **2007**, *33*, 181.

(17) Lee, J.-M.; Yoo, C. K.; Lee, I.-B. Statistical process monitoring with independent component analysis. *J. Process Control* **2004**, *14*, 467.

(18) Lee, J.-M.; Yoo, C. K.; Lee, I.-B. Statistical monitoring of dynamic processes based on dynamic independent component analysis. *Chem. Eng. Sci.* **2004**, *59*, 2995.

(19) Yoo, C. K.; Lee, J.-M.; Vanrolleghem, P. A.; Lee, I.-B. On-line monitoring of batch processes using multiway independent component analysis. *Chemom. Intell. Lab. Syst.* **2004**, *71*, 151.

(20) Kano, M.; Tanaka, S.; Hasebe, S.; Hashimoto, I.; Ohno, H. Monitoring independent components for fault detection. *AIChE J.* **2003**, *49*, 969.

(21) Kano, M.; Hasebe, S.; Hashimoto, I.; Ohno, H. Evolution of multivariate statistical process control: Application of independent component analysis and external analysis. *Comput. Chem. Eng.* **2004**, *28*, 1157.

(22) Albazzaz, H.; Wang, X. Z. Statistical process control charts for batch operations based on independent component analysis. *Ind. Eng. Chem. Res.* **2004**, *43*, 6731.

(23) Lee, J.-M.; Qin, S. J.; Lee, I.-B. Fault detection and diagnosis based on modified independent component analysis. *AIChE J.* **2006**, *52*, 3501.

(24) Kano, M.; Tanaka, S.; Hasebe, S.; Hashimoto, I.; Ohno, H. Combination of independent component analysis and principal component analysis for multivariate statistical process control. In *Proceedings of the International Symposium on Design, Operation and Control of Chemical Plants (PSE Asia 2002)*; Taipei, Taiwan, Dec 4−6, 2002; p 319.

(25) Ge, Z. Q.; Song, Z. H. Process monitoring based on independent component analysis−principal component analysis (ICA−PCA) and similarity factors. *Ind. Eng. Chem. Res.* **2007**, *46*, 2054.

(26) Widodo, A.; Yang, B.-S. Application of nonlinear feature extraction and support vector machines for fault diagnosis of induction motors. *Expert Syst. Appl.* **2007**, *33*, 241.

(27) Cho, J.-H.; Lee, J.-M.; Choi, S. W.; Lee, D. K.; Lee, I.-B. Fault identification for process monitoring using kernel principal component analysis. *Chem. Eng. Sci.* **2005**, *60*, 279.

(28) Lee, J.-M.; Yoo, C. K.; Choi, S. W.; Vanrolleghem, P. A.; Lee, I.-B. Nonlinear process monitoring using kernel principal component analysis. *Chem. Eng. Sci.* **2004**, *59*, 223.

(29) Lee, J.-M.; Yoo, C. K.; Lee, I.-B. Fault detection of batch processes using multiway kernel principal component analysis. *Comput. Chem. Eng.* **2004**, *28*, 1837.

(30) Choi, S. W.; Lee, C.; Lee, J.-M.; Park, J. H.; Lee, I.-B. Fault detection and identification of nonlinear processes based on kernel PCA. *Chemom. Intell. Lab. Syst.* **2005**, *75*, 55.

(31) Cui, P. L.; Li, J. H.; Wang, G. Z. Improved kernel principal component analysis for fault detection. *Expert Syst. Appl.* **2008**, *34*, 1210–1219.

(32) Nicolaï, B. M.; Theron, K. I.; Lammertyn, J. Kernel PLS regression on wavelet transformed NIR spectra for prediction of sugar content of apple. *Chemom. Intell. Lab. Syst.* **2007**, *85*, 243.

(33) Yang, J.; Frangi, A. F.; Yang, J.-Y.; Zhang, D.; Jin, Z. KPCA plus LDA: A complete kernel Fisher discriminant framework for feature extraction and recognition. *IEEE Trans. Pattern Anal.* **2005**, *27*, 230.

(34) Cho, H.-W. Nonlinear feature extraction and classification of multivariate process data in kernel feature space. *Expert Syst. Appl.* **2007**, *32*, 534.

(35) Cho, H.-W. Identification of contributing variables using kernel-based discriminant modeling and reconstruction. *Expert Syst. Appl.* **2007**, *33*, 274.

(36) Zhang, Y. W.; Qin, S. J. Fault detection of nonlinear process using multiway kernel independent component analysis. *Ind. Eng. Chem. Res.* **2007**, *46*, 7780.

(37) Zhang, Y. W. Fault detection and diagnosis of nonlinear processes using improved kernel independent component analysis (KICA) and support vector machine (SVM). *Ind. Eng. Chem. Res.* **2008**, *47*, 6961.

(38) Yang, J.; Gao, X.; Zhang, D.; Yang, J.-Y. Kernel ICA: An alternative formulation and its application to face recognition. *Pattern Recogn.* **2005**, *38*, 1784.

(39) Wold, S.; Kettaneh, N.; Friden, H.; Holmberg, A. Modelling and diagnostics of batch processes and analogous kinetic experiments. *Chemom. Intell. Lab. Syst.* **1998**, *44*, 331.

(40) Lee, J.-M.; Yoo, C. K.; Lee, I.-B. Enhanced process monitoring of fed-batch penicillin cultivation using time-varying and multivariate statistical analysis. *J. Biotechnol.* **2004**, *110*, 119.

(41) Lee, J.-M.; Yoo, C. K. On-line batch process monitoring using different unfolding method and independent component analysis. *J. Chem. Eng. Jpn.* **2003**, *36*, 1384.

(42) Rännar, S.; MacGregor, J. F.; Wold, S. Adaptive batch monitoring using hierarchical PCA. *Chemom. Intell. Lab. Syst.* **1998**, *41*, 73.

(43) Ündey, C.; Ertunç, S.; Çinar, A. Online batch/fed-batch process performance monitoring, quality prediction, and variable-contribution analysis for diagnosis. *Ind. Eng. Chem. Res.* **2003**, *42*, 4645.

(44) Lu, N. Y.; Gao, F. R.; Wang, F. L. A sub-PCA modeling and on-line monitoring strategy for batch processes. *AIChE J.* **2004**, *50*, 255.

(45) Dong, D.; McAvoy, T. J. Multi-stage Batch Process Monitoring. In *Proceedings of the American Control Conference*; IEEE Press: Piscataway, NJ, 1995; p 1857.

(46) Lennox, B.; Hiden, H.; Montague, G.; Kornfeld, G.; Goulding, P. Application of multivariate statistical process control to batch operations. *Comput. Chem. Eng.* **2000**, *24*, 291.

(47) Lane, S.; Martin, E. B.; Kooijmans, R.; Morris, A. J. Performance monitoring of a multi-product semi-batch process. *J. Process Control* **2001**, *11*, 1.

(48) Ündey, C.; Cinar, A. Statistical monitoring of multistage, multiphase batch processes. *IEEE Control Syst. Mag.* **2002**, *22*, 40.

(49) Ramaker, H.-J.; van Sprang, E. N. M.; Westerhuis, J. A.; Smilde, A. K. Fault detection properties of global, local and time evolving models for batch process monitoring. *J. Process Control* **2005**, *15*, 799.

(50) Camacho, J.; Picó, J. Online monitoring of batch processes using multi-phase principal component analysis. *J. Process Control* **2006**, *16*, 1021.

(51) Zhao, C. H.; Wang, F. L.; Mao, Z. Z.; Lu, N. Y.; Jia, M. X. Improved knowledge extraction and phase-based quality prediction for batch processes. *Ind. Eng. Chem. Res.* **2008**, *47*, 825.

(52) Haykin, S. *Neural Networks*; Prentice-Hall: Englewood Cliffs, NJ, 1999.

(53) Christianini, N., Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*; Cambridge University Press: Cambridge, U.K., 2000.

(54) Silverman, B. W. *Density Estimation for Statistics and Data Analysis*; Chapman & Hall: London, 1986.

(55) Ryan, T. P. *Statistical Methods for Quality Improvement*; Wiley: New York, 1989.

(56) Chen, Q.; Wynne, R. J.; Goulding, P.; Sandoz, D. The application of principal component analysis and kernel density estimation to enhance process monitoring. *Control Eng. Pract.* **2000**, *8*, 531.

(57) Martin, E. B.; Morris, A. J. Non-parametric confidence bounds for process performance monitoring charts. *J. Process Control* **1996**, *6*, 349.

(58) Birol, G.; Ündey, C.; Parulekar, S. J.; Cinar, A. A morphologically structured model for penicillin production. *Biotechnol. Bioeng.* **2002**, *77*, 538.

(59) Birol, G.; Ündey, C.; Cinar, A. A modular simulation package for fed-batch fermentation: Penicillin production. *Comput. Chem. Eng.* **2002**, *26*, 1553.

Received for review August 25, 2008
Revised manuscript received July 7, 2009
Accepted July 21, 2009

IE8012874