# Adaptive Monitoring Based on Independent Component Analysis for Multiphase Batch Processes with Limited Modeling Data

**ARTICLE** *in* INDUSTRIAL & ENGINEERING CHEMISTRY RESEARCH · MARCH 2008

Impact Factor: 2.59 · DOI: 10.1021/ie701680y



**5 AUTHORS**, INCLUDING:

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

Zhizhong Mao
Northeastern University (Shenyang, China)
**66** PUBLICATIONS **335** CITATIONS
SEE PROFILE

Mingxing Jia
Northeastern University (Shenyang, China)
**36** PUBLICATIONS **360** CITATIONS
SEE PROFILE

# Adaptive Monitoring Based on Independent Component Analysis for Multiphase Batch Processes with Limited Modeling Data

**Chunhui Zhao, Fuli Wang,* Zhizhong Mao, Ningyun Lu,† and Mingxing Jia**

*College of Information Science and Engineering, Northeastern University, Shenyang, Liaoning Province, People's Republic of China*

An adaptive monitoring method based on multiphase independent component analysis (ICA), termed AMPICA, is proposed for batch processes of long duration. Starting from limited modeling batches, the local process correlation structures are explored and thus multiple phase-specific models are developed, where each phase pattern can be faithfully approximated by different sub-ICA models. Then an adaptive updating strategy is adopted to accommodate more underlying process information and normal batch-to-batch slow-varying behaviors with the accumulation of new batch data. The idea and algorithm are illustrated with respect to the typical data collected from a benchmark simulation of fed-batch penicillin fermentation production. The simulation results show that the proposed method provides a new feasible statistical analysis solution for modeling and monitoring problems with limited data in long-cycle batch processes.

## 1. Introduction

Batch and semibatch processes play an important role in the processing of specialty chemicals and in the semiconductor, food, and biology industries for producing high-value-added products to meet today's rapidly changing market. Hence proper process monitoring and diagnosis is important to not only quality improvement but also process safety.[1−7] Multivariate statistical methods based on multiway principal component analysis (MPCA) and multiway partial least squares (MPLS) have been widely employed in batch process monitoring. Numerous review books and papers have been produced by many researchers,[8−13] where statistical analysis models are developed implicitly based on the Gaussian-distribution assumption from batch to batch. Recently, independent component analysis (ICA),[14−19] an alternative and useful extension of principal component analysis (PCA), has been given extensive attention in context with blind separation of independent sources for revealing hidden, statistically independent factors that underlie sets of non-Gaussian process measurements. Many published works[20−29] have shown that the ICA-based statistical analysis method exhibited superiority over other feature extraction techniques in wide practical applications since it is not based on the assumption of multivariate Gaussian distribution. When the measured variables are regarded to follow non-Gaussian distribution, ICA provides more meaningful latent feature information compared with PCA.

In most existing methods, it is assumed that reference batches collected for modeling are sufficient, which cover statistically all normal batch-to-batch variations, and thus can be employed in modeling to reveal the underlying process information. This is relatively easy for batch processes of short duration and processes that are inexpensive to conduct many trial runs. Slow batch processes, such as bio-related processes, take a long time to complete a batch run. In this case, it is not practical to obtain many cycles of reference data before statistical process analysis. A modeling and monitoring method, thus, is needed to start process understanding and analysis with limited batch cycles, i.e., just several batches. However, when conventional statistical methods are applied to limited modeling batches, several issues may arise: (i) How should the reference batch data be normalized prior to modeling and monitoring? (ii) How should one develop the monitoring system from the limited reference batches and then adaptively adjust the initial models with the accumulation of new normal batches? Thus it would not be feasible to develop a reliable monitoring system using conventional statistical analysis strategies to reveal the underlying batch-to-batch process information. One alternative statistical analysis method should be explored to develop the monitoring system only based on several modeling batches. Moreover, with the accumulation of newly available batches, the initial models should be adaptively improved and updated to accommodate more process information. This updating can also account for slow batch-to-batch varying behaviors.

In the present paper, starting from limited modeling batches, we develop an online monitoring and adaptive updating method based on multiphase ICA for batch processes of long duration. First, in the modeling procedure, the limited several batches are arranged variable-wise and generalized moving windows are employed in data normalization to prepare the process measurements prior to the development of monitoring models. Then the segments of the batch are detected based on a recursive optimizing step, and thus the entire process is properly divided into different phases. Therefore, the process behaviors can be well approximated by phase-specific sub-ICA models and the process monitoring can be performed focusing on each phase. Moreover, an adaptive updating strategy is formulated, which evaluates distinctively in each phase when it is necessary to update the models and how to do so. The updating can help the initial monitoring system well accommodate more and more process information as well as adapt to the common slow-varying dynamics over batches.

This paper is organized as follows. A brief introduction to the ICA algorithm is given in section 2, which is employed as the basic analysis tool in our work. Details of the proposed method are explained in section 3. A simulation case is shown in section 4, where the concept and algorithm presented in the work are illustrated effectively when applied to the simulated penicillin fermentation benchmark process. Conclusions are drawn in section 5.

* To whom correspondence should be addressed. Tel.: +86-024-83687434. Fax: +86-024-23890912. E-mail: flwang@mail.neu.edu.cn.

† College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu Province, People's Republic of China

10.1021/ie701680y CCC: $40.75 
Published on Web 03/15/2008

## 2. Independent Component Analysis (ICA)

Independent component analysis (ICA)[14−19] is a recently developed method in which the goal is to find the statistically independent non-Gaussian hidden factors, or as independent as possible, which constitute the observed variables through linear combination. Such a representation seems to capture the essential structure of the measurement data in many applications, including feature extraction and signal separation. There are further promising applications for ICA since it works in terms of higher order statistics than PCA.

In the ICA algorithm, it is assumed that the $J$ measured variables $x_1, x_2, ..., x_J$ can be described as a linear combination of $R$ (generally $R \leq J$) non-Gaussian independent components $s_1, s_2, ..., s_R$. The basic matter of ICA is to estimate both the latent components **s** and the mixing relationship **A** from only the process measurements **x** without any related prior knowledge, termed as the process of blind separation. The relationship between ICs and the process variables is expressed as follows:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{1}$$

where $\mathbf{x}(J \times 1)$ is the process vector. $\mathbf{A}(J \times R)$ represents the unknown mixing matrix. $\mathbf{s}(R \times 1)$ denotes the independent component vector, which has unit variance: $E(\mathbf{s}\mathbf{s}^T) = \mathbf{I}$.

In fact, the goal of ICA is equivalent to estimate the demixing matrix, $\mathbf{W}(R \times J)$, the inverse of mixing matrix, and obtain the independent component simply by

$$\mathbf{s} = \mathbf{W}\mathbf{x} \tag{2}$$

In developing the ICA algorithm, it is assumed that process variables are prewhitened, which is generally achieved by PCA, so that its components are uncorrelated and their variances are equivalent to unity. This whitening transformation can be expressed as

$$\mathbf{z} = \mathbf{\Lambda}^{-1/2}\mathbf{U}^T\mathbf{x} = \mathbf{Q}\mathbf{x} \tag{3}$$

where $\mathbf{Q} = \mathbf{\Lambda}^{-1/2}\mathbf{U}^T$ is the whitening matrix, in which **U** (orthogonal matrix of eigenvectors) and $\mathbf{\Lambda}$ (diagonal matrix of its eigenvalues) are generated from the eigen-decomposition on covariance matrix $E(\mathbf{x}\mathbf{x}^T)$.

After the whitening preprocessing, combined with eq 1, the following relationship can be expressed:

$$\mathbf{z} = \mathbf{Q}\mathbf{x} = \mathbf{Q}\mathbf{A}\mathbf{s} = \mathbf{B}\mathbf{s} \tag{4}$$

where $\mathbf{B} = \mathbf{Q}\mathbf{A}$ is an orthogonal matrix, given that $E(\mathbf{z}\mathbf{z}^T) = \mathbf{B}E(\mathbf{s}\mathbf{s}^T)\mathbf{B}^T = \mathbf{B}\mathbf{B}^T = \mathbf{I}$.

Naturally, we can estimate **s** from eq 4 by

$$\mathbf{s} = \mathbf{B}^T\mathbf{z} = \mathbf{B}^T\mathbf{Q}\mathbf{x} = \mathbf{A}^T\mathbf{Q}^T\mathbf{Q}\mathbf{x} \tag{5}$$

Combining eqs 2 and 5, we can readily derive the following relation between **W** and **A**:

$$\mathbf{W} = \mathbf{B}^T\mathbf{Q} = \mathbf{A}^T\mathbf{Q}^T\mathbf{Q} \tag{6}$$

Hyvärinen and Oja[18] presented a FastICA algorithm, whose procedure can be described as follows:

(a) Center and whiten process data. The FastICA algorithm employs PCA to remove correlation between the observed variables so that the whitened data are in fact the PCA scores with unit variance. In the FastICA algorithm procedure, we denote the preprocessed data just by **x** for simplicity.

(b) Determine the number of independent components $n$; set the counter $p = 1$.

(c) Choose an initial (e.g., random) weight vector $\mathbf{w}_p$.

(d) Let $\mathbf{w}^+ = E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - E(g'(\mathbf{w}^T\mathbf{x}))\mathbf{w}$, where $g(\cdot)$ are the derivatives of the nonquadratic function $G(\cdot)$.

(e) Orthogonalization and normalization are represented by

$$\mathbf{w}_p = \mathbf{w}_p - \sum_{j=1}^{p-1}\mathbf{w}_p{}^T\mathbf{w}_j\mathbf{w}_j; \qquad \mathbf{w}_p = \mathbf{w}_p/||\mathbf{w}_p||$$

(f) If $\mathbf{w}_p$ is not converged (here the convergence means that the dot product of the old and new values of **w** is almost equal to 1), go back to step d.

(g) Set $p = p + 1$. If $p \leq n$, return to step c. Otherwise, output the results.

According to Hyvärinen and Oja,[18] the following choices of $G(\cdot)$ have proved very useful:

$$G_1(u) = \frac{1}{a_1}\log\cosh(a_1 u) \qquad G_2(u) = -\exp\left(-\frac{u^2}{2}\right)$$

where $1 \leq a_1 \leq 2$ is some suitable constant.

The ICA algorithm holds the following ambiguities:

(i) We cannot determine the variances (energies) of the independent components.

(ii) We cannot determine the order of the independent components.

The first ambiguity is insignificant in most applications, whereas the second ambiguity should be treated with caution which results from the fact that both **s** and **A** are unknown. Therefore, we can freely change the order of the terms in the sum of the relationship expression

$$x_j = \sum_{i=1}^{R} a_{j,i}s_i \qquad (j = 1, 2, ..., J)$$

and call any of the independent components the first one. Such an ambiguity will exert an influence on the choice of phase partition algorithm, which will be further clarified in subsection 3.2.

## 3. AMPICA Algorithm for Slow Processes with Limited Batches

In each batch run, assume that $J$ variables are measured at $k = 1, 2, ..., K$ time instances throughout the batch. Then process data collected from similar $I$ batches can be organized as a three-way array $\boldsymbol{X}(I \times J \times K)$. In the present work, the batches are of equal length without special declaration so that the specific process time can be used as an indicator to data preprocessing, modeling, and online monitoring.

**3.1. Data Preprocessing.** Before performing statistical analysis, the three-way data matrix must be unfolded to a two-way form. Recently, applications have been presented in which the three-way process data matrix was unfolded variable-wise and scaled to zero mean and unit variance,[5,11] where $K$ number of vertical time-slice matrices, $\mathbf{X}_k(I \times J)$, are placed beneath one another to form a two-way data matrix $\mathbf{X}(KI \times J)$. This unfolding way does not require all batches to be of equal length, nor does it require estimation of future data for online monitoring. However, the conventional mean centering for a variable-unfolding matrix simply subtracts a constant, the grand mean of each variable over all batches and all time, from the trajectory of each variable. This still leaves the nonlinear and time-varying

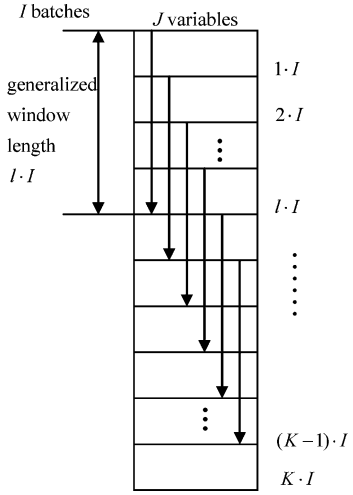


**Figure 1.** Illustration of the generalized moving window.

trajectories in the data matrix. Hence, direct unfolding and mean centering in the manner offer little benefit to process monitoring.

In this section, we employ generalized moving windows based on variable-unfolding data arrangement of several batches, which is the basic data preprocessing unit. Illustrated in Figure 1, data are arranged in each generalized window forming a two-way matrix, $\mathbf{X}((l \cdot I) \times J)$, in which sequential production order along the batch axis is preserved. $l$ is the time length of the moving window, so $l \cdot I$ is the data size of the generalized window. Here, $l$ should be properly chosen so that the data length of the generalized window, $L$ ($L = l \cdot I$), can satisfy at least 2 or 3 times the number of process variables to ensure a reliable statistical analysis as commonly practiced in most multivariable regressions.[31] Each row represents process measurements at each sampling time of a batch, and rows are arranged according to ascending time and batch. With this arrangement, $K - l + 1$ number of generalized moving windows are finally generated, designated as $\mathbf{X}^w((l \cdot I) \times J)$ ($w = 1, 2, ..., K - l + 1$). In each generalized moving window, since within the short period the process trajectory has not varied greatly, we subtract the mean of each column, which can approximately eliminate the main nonlinearity due to the dynamic behaviors of the process and look at the deviation from the average trajectory. Moreover, in each generalized moving window, each variable is scaled to unit variance to handle different measurement units, thus giving each equal weight. Then the means and standard deviations are orderly denoted with the specific process time, $k = l, l + 1, ..., K$, corresponding to the right side of each moving window, which will be used for online monitoring. Here, those samples within the first window, $\mathbf{x}_k$ (where $k = 1, 2, ..., l - 1$), which are not seated at the right side of any window, can borrow the mean and variance value corresponding to time $l$, i.e., the right side of the first moving window. Thus, during online application, the sampling point at each time, $\mathbf{x}_k$ ($k = 1, 2, ..., K$), can be normalized by directly employing the corresponding mean and standard values based on the denotation of process time. Consequently, the two-way data matrix $\mathbf{X}(KI \times J)$ is converted into the normalized form, $\bar{\mathbf{X}}(KI \times J)$.

**3.2. Phase Partition and Phase-Based ICA Modeling.** For multiphase batch processes, each phase has its own underlying characteristics, and a batch process can exhibit significantly different behaviors over different phases. Considering that the multiplicity of operation phases is an inherent nature of many batch processes, it is desirable to partition the process duration into different phases and develop phase-based statistical analysis methods. To satisfy the monitoring requirements over different operation phases, Ündey et al.[32] have argued that several models should be developed to represent the multiphase nature of the process, which increases one's capability to identify events localized in different phases. In our previous work,[33,34] a variant $k$-means clustering algorithm is employed to check segments of the batch, which assumes that the changes of phase can be reflected by the changes of underlying process correlations. Then each model represents a specific phase and focuses on the local behaviors of the batch process, which allows one to unveil the underlying correlation structures specific to each phase and can effectively enhance one's process understanding and improve monitoring reliability. Camacho et al.[35,36] introduced a multiphase principal component analysis (MPPCA) algorithm to detect segments of the batch where the behavior of the process can be well approximated by one linear PCA model, which first attempted to divide the batch into different phases automatically to improve the prediction power of the monitoring model. All the phase-based principal component analysis methods have been successfully applied to multiphase batch processes. As mentioned before, an ICA-based statistical method should be selected when normal distribution cannot be satisfied well since the non-Gaussian latent signals can reveal more meaningful higher-order statistical distribution information. In the present work, we employed the ICA algorithm as the basic analysis tool for the modeling problem with limited batches. The key is how to divide batch processes into several phases only based on several reference batches.

In PCA, it is well-known that the order of the principal components is fixed according to their descending variance. However, as mentioned in section 2, the second ambiguity of ICA reveals that the order of the extracted independent components is different even when the same method is carried out focusing on the same process measurement. That is, we can call any independent component the first one. The uncertainty is determined by the fact that independent components are extracted through an iterative optimization procedure. Such a property determines that it is unsuitable to evaluate the dissimilarity between different ICA patterns by directly calculating the distance metric. Thus, it is improper to adopt the clustering idea in our previous work[33,34] to get the phase partition. Based on the above analysis, we adopt the basic idea of an initial MPPCA algorithm in variable-unfolded data by Camacho and Picó[35] to divide the process trajectory into different phases. Moreover, we further modify and develop the standard MPPCA algorithm[35] to be specifically fit for our application. Inspired by the nomenclature by Camacho and Picó,[35,36] here the partition algorithm is termed "MPICA".

The basic procedure of the MPICA algorithm in our present work is summarized as follows:

Step 1: Input the normalized data matrix $\bar{\mathbf{X}}(KI \times J)$.

Step 2: Perform the FastICA algorithm presented by Hyvärinen and Oja[18] on the data and get the initial model, where only one independent component (IC) is retained in the model for simplicity.

Step 3: For each time instant ($k$), if the subdivision in $k$ generates two segments of length larger than the minimum phase length predetermined prior, perform ICA on each of the two segments and compute mean squared errors (MSE) from the prediction of the initial model and the submodels, denoted as $MSE_0$ and $MSE_k$, respectively. Compare $MSE_k$ and choose the subdivision with the lowest value, $MSE_{min}$.

Step 4: If the best improvement of the MSE value, which is the difference between $MSE_{min}$ and $MSE_0$, does not reach a

threshold, then stop the branch; otherwise, accept the subdivision and recursively repeat steps 2 and 3, respectively, for the two resulting segments, which are now employed as the new input data in step 2.

The other details of the algorithm can be found in the work by Camacho and Picó[35,36] and will not be repeated here.

From the above algorithm, it can be seen that the phase partition result has a close relationship with the predetermined parameter: the minimum phase length. In multiphase batch processes, it is well-known that the minimum phase length, i.e., the minimum number of consecutive samples included in a phase, is up to the real underlying process characteristics. If the parameter can be set properly in accord with the actual process nature, it will not only reduce the computation complexity in the partition procedure but also benefit the correct division of subphases and thus get the accurate phase-specific monitoring models. In conclusion, the size of the minimum phase length should be set avoiding obtaining phases with insufficient sample points which will not provide stable and sufficient statistical information. Moreover, it should be selected not to cover too many process patterns to ensure the sensitivity to the changing of process characteristics over phases. However, up to now, there has been no definite criterion or uniform standard to strictly quantify the minimum phase length. Generally, prior expertise can provide an important reference standard for the general idea of minimum phase length. Moreover, it can also be selected by trial and error. Therefore, its determination is inevitably affected more or less by artificial subjectivity factors. The investigation on the determination of minimum phase length is meaningful and deserves further attention in the future.

The objective of the MPICA partition algorithm presented here is to find segments, i.e., phases, which can be well approximated by a linear ICA model, by performing an optimization search based on a division performance evaluation index MSE. Therefore, we can simply keep one IC in each model, which is enough to find out whether the partition is acceptable by fairly comparing the MSE values among them, although the models may not reach the best monitoring performance. After the phase partition, the proper number of ICs retained within each phase-specific model should be determined again to get more accurate monitoring models. Some methods have been outlined by Lee et al.,[29] although there is no standard criterion. In the present work, we use the simple approach of sorting the rows of the demixing matrix, $\mathbf{W}$, on the basis of their Euclidean norms.[21] That is, the ICs are sorted using Euclidean norms in order to show only those ICs that cause dominant changes in the processes.

From the partition procedure, there is no special requirement for the number of reference batches, which well meets the case dealt with in our present work. The output of MPICA is a partition of process trajectory along time direction. Process time can thus be naturally used as an indicator to identify the alternation of different subphases. After phase partition, the statistical analysis and modeling can then be performed on the phase-specific normalized data matrix $\bar{\mathbf{X}}^c(IK_c \times J)$, where $K_c$ is the operation duration of phase $c$.

**3.3. Monitoring Statistics and Confidence Limits.** In the above section, the following phase-representative relationships within each phase have been obtained for the phase-specific process data $\bar{\mathbf{X}}^c(IK_c \times J)$:

$$\mathbf{S}_c^{\mathrm{T}} = \mathbf{W}_c \bar{\mathbf{X}}^{c^{\mathrm{T}}}$$

$$\mathbf{E}_c^{\mathrm{T}} = \bar{\mathbf{X}}^{c^{\mathrm{T}}} - \mathbf{A}_c \mathbf{S}_c^{\mathrm{T}} \tag{7}$$

where $\mathbf{S}_c(IK_c \times R_c)$ is the phase-representative non-Gaussian independent components, $\mathbf{W}_c(R_c \times J)$ is the separating matrix, and $\mathbf{A}_c(J \times R_c)$ is the mixing matrix. $\mathbf{E}_c(IK_c \times J)$ is the residual matrix. $R_c$ is the remaining number of independent components in phase $c$.

In conventional multivariate monitoring systems, two types of statistics are commonly calculated: the $D$-statistic for the systematic part of the process variations captured by statistical models and the $Q$-statistic for the residual part of the process variations unoccupied by models. In previous work,[21−25] the $D$-statistic in ICA, also known as the $I^2$ statistic, is the sum of the squared independent scores at each time and is commonly defined as $I_k^2 = \mathbf{s}_k^{\mathrm{T}}\mathbf{s}_k$. In the present work, we appropriately calculate the $I^2$ statistic for the $i$th batch at the $k$th time within each phase as follows:

$$I_{i,k}^{\ 2} = (\mathbf{s}_{i,k} - \bar{\mathbf{s}}_c)^{\mathrm{T}} \mathbf{M}_c^{\ -1} (\mathbf{s}_{i,k} - \bar{\mathbf{s}}_c) \tag{8}$$

where $\mathbf{s}_{i,k}(R_c \times 1)$ is the independent component vector of the $i$th batch observation at the $k$th time within phase $c$; $\bar{\mathbf{s}}_c(R_c \times 1)$denotes the mean vector of the rows of the independent component matrix $\mathbf{S}_c(IK_c \times R_c)$; $\mathbf{M}_c(R_c \times R_c)$ corresponds to the phase-representative variance−covariance matrix of latent variable $\mathbf{S}_c(IK_c \times R_c)$, which actually is an identity matrix. Moreover, it should be noted that $\bar{\mathbf{s}}_c(R_c \times 1)$ is not necessarily zero vector since $\mathbf{S}_c(IK_c \times R_c)$ is not zero mean due to the use of generalized moving windows in the data preprocessing procedure.

For the residual part of the nonsystematic variations, the $Q$-statistic of each batch, also known as the SPE statistic, can be also defined at each time $k$ as follows:

$$\mathrm{SPE}_{i,k} = \mathbf{e}_{i,k}^{\ \mathrm{T}} \mathbf{e}_{i,k} = (\mathbf{x}_{i,k} - \hat{\mathbf{x}}_{i,k})^{\mathrm{T}} (\mathbf{x}_{i,k} - \hat{\mathbf{x}}_{i,k}) \tag{9}$$

where $\mathbf{e}_{i,k}(J \times 1)$ is the estimated residual vector corresponding to $\mathbf{x}_{i,k}$.

In PCA monitoring, the control limits of monitoring statistics can be approximated by a specified distribution, which is rooted in the presupposition that the probability density of the process measurements follows a multivariate Gaussian distribution.[37] However, as mentioned before, the non-Gaussianity in ICA makes it impossible to calculate the control limits of $I^2$ statistic in a way similar to that for Hotelling-$T^2$ in PCA, as well as SPE. When the data distribution rule is unknown, an alternative approach, such as a data-driven technique, must be developed to define the normal operating regions of monitored statistics. Lee et al.[21−24] used a nonparametric density estimation method to calculate the control limits for ICA monitoring. However, the number of samples needed in the kernel density estimation is greatly larger than that by known distributions. With enough training samples, a reliable density fitting can be well obtained in theory. When the data are not enough, it is not possible to calculate a reliable enough density estimation result. In our proposed method, the control limits are developed focusing on each phase. However, the samples covered in each phase are different as indicated by $IK_c$, where there may be some short phases with insufficient number of data. Therefore, two different treatment ways are developed respectively in case of two different types of circumstances.

For the first case, the number of samples used during modeling, $IK_c$, is sufficient. The density estimation can be simply obtained based on all the reference data, and then the control limits can be naturally calculated based on the estimated density distribution.

For the second case, the number of reference data is not sufficient. The control limits can thus be calculated using a jackknife procedure in which each of the reference data is left out once, and the density distribution is estimated based on the remaining data. Therefore there will be the same number of density estimation results as the number of jackknife procedures. The mean of all the estimated density distributions can then be utilized as the representative density to calculate the final control limits.

Based on the above strategy, the density distributions of monitoring statistics are estimated. Thus the points occupying the 95% area of density functions can be calculated as the control limits with 5% confidence level.

When a new batch run is available, using the process time as an indicator, the new sampling data, $\mathbf{x}_{new}(J \times 1)$, should first be normalized using the relevant mean and variance obtained from the modeling procedure before calling the corresponding phase-representative ICA model. Then it is projected onto the phase-specific ICA model which the current sample falls in to calculate the independent components and derive the $I^2$ and SPE statistics:

$$\mathbf{s}_{new} = \mathbf{W}_c\mathbf{x}_{new}$$

$$\hat{\mathbf{x}}_{new} = \mathbf{A}_c\mathbf{s}_{new}$$

$$I_{new}^{\ 2} = \mathbf{s}_{new}^{\ T}\mathbf{M}_c^{\ -1}\mathbf{s}_{new}$$

$$SPE_{new} = (\mathbf{x}_{new} - \hat{\mathbf{x}}_{new})^T(\mathbf{x}_{new} - \hat{\mathbf{x}}_{new}) \qquad (10)$$

Process monitoring is conducted by continuously comparing the two statistics with the predetermined control limits. If the new process follows an operation sequence and variable trajectories similar to those of the reference batches, it can be considered a usual batch and should be archived in the normal database. Otherwise, the statistics will go beyond the control limits responding to a process abnormality.

**3.4. AMPICA Online Updating Strategy.** The initial monitoring system has been derived from a limited number of reference batches, more focusing on within-batch information. It can successfully supervise the process operation at the beginning since the neighboring subsequent batches will not change remarkably and hold underlying process features similar to the reference batches. Nevertheless, with process evolvement the accuracy of the monitoring system may be compromised since normal slow-varying behaviors are common in most real industrial processes. Therefore, with the accumulation of new successful batches, we should update the reference trajectory, the representative ICA monitoring models, and the online control limits to explore more batch-to-batch variation information and meanwhile to accommodate the normal slow-varying behaviors.

Generally, normal slow-varying behaviors are popular, resulting in the failure of the history models to properly explain the current underlying behaviors. Despite the gradual changing of underlying structure, it is regarded as normal because the change will not drastically affect the variable correlations and deteriorate the process operating performance, which is, in fact, often the reflection of real industrial processes. Moreover, the initial control limits may become too tight (or loose) in certain time spans for monitoring a new batch. Thus, with the accumulation of new successful batch data, the monitoring model structures should be adjusted to adapt to the current operating manner, as well as the control limits. On the one hand, the mean and variance used for data normalization should be corrected all the while; on the other hand, the structures of monitoring models and the control limits should be adjusted only when it is necessary. Here, the correction of mean and variance is termed "minor updating". By contrast, the revision of the model structures and the corresponding control limits is called "major updating".

The key is to judge when it is time to conduct the major updating adjustment of monitoring models and how to do so. Here the following analysis strategy is given: when a new batch run tracks the usual operation trajectory and stays well within the normal confidence limit, it can be considered a success and should be archived in the normal database. Otherwise, if there are consecutive samples going beyond the control limits, there are two potential causes. One may result from the real process abnormality, and the other may be due to the failure of historical monitoring models in pursuing the current process characteristics. To identify the two cases, the reliability of the old models should be checked online. That is, new monitoring models are recalculated based on the recently new normal batches, which are assured that only normal process information is included. Then the current evolving batch is reprojected onto the new models and the new monitoring statistic values, $I_k^2$ and $SPE_k$, are recalculated. If both statistics go below their control limits, it indicates that the previous alarm indications are caused by the failure of old models and thus the new models should replace the old ones as the current monitoring tool. Otherwise, if fault alarms are still exposed, it reveals that a real process abnormality has occurred, resulting from the unknown disturbances rather than the normal slow-varying dynamics. Correspondingly, the process should be analyzed in detail to give the possible cause for the fault. Some adaptive updating algorithms[38,39] have been developed to accommodate new underlying process behaviors. In their methods, new process information efficiently entered the old models recursively by properly combining the new models with old ones. Then the updated models are employed for the coming monitoring. However, in the present study, the updated models cannot be simply obtained by combining the old models with the new ones, which is determined by the second ambiguity of the ICA algorithm mentioned in section 2. That is, although the underlying structure patterns may be similar, ICs might be found in different orders, which thus generates different mixing matrices. Therefore, the updated ICA models must be recalculated using the FastICA algorithm[18] introduced in section 2.

In the proposed strategy, one issue should be concerned with the time interval cost by updating. In the present work, the model structures are updated focusing on each phase only when needed, which may reduce the computation complexity and burden compared with an alternative batch-sense updating strategy in which the models are updated after each normal batch. However, there may be some undesirable time delays for online monitoring before finishing the model correction, which is determined by the computation cost and the real case in process operation, e.g., the sampling interval. However, although the time delays will more or less influence the real-time monitoring of process operation within the current phase, it will not hinder the subsequent monitoring of its next neighboring phases since the statistical analysis and modeling are performed focusing on each phase. Moreover, the updating procedure can be feasibly finished during the time gap between the corresponding phases of two consecutive batches. Therefore, the influence of time delay is acceptable and the proposed updating strategy is feasible for practical online application.

The updating strategy is summarized as follows:

(a) To reduce the error caused by data normalization, the current mean and standard deviation should be real-time

adjusted. During the monitoring progress, whenever a successful normal batch is archived into the normal database, the drift of data normalization information is introduced into the current monitoring tools by a real-time adjustment of the mean and variance. Then the newly calculated values will be employed in the data preprocessing of the coming batch.

(b) As the process exhibits significantly different underlying behaviors over different phases, monitoring models need to be updated respectively over different phases. That is, updating frequencies of phase-specific models may be different, which indeed are in accord with the underlying behavior of each specific phase.

(c) The key to the updating algorithm is to judge when it is necessary to adjust the current phase-representative old models. During the monitoring procedure, whenever the statistics are beyond the normal control limit, recalculate the new phase-specific model based on the recently monitored normal batches before the current batch within the current phase. Here for simplicity, the number of modeling batches remains the same. Then it is employed to calculate the new $I^2$ and SPE values. If both statistics go below their control limits, it reveals that the old model can no longer explain the correlation of the current data and the updating is necessary. Otherwise, a process abnormality has occurred.

(d) In the updating procedure, the phase-specific data used for estimating the distribution parameters of kernel density can be augmented to include the statistic values of the new normal batches. Then the new control limits can be naturally developed the same as introduced in section 3.3. By doing so, each phase will contain more and more batch-to-batch variations, and the estimated control limits will be correspondingly adjusted.

In the proposed updating algorithm, online monitoring results are made full use of to evaluate whether the current phase-specific model needs to be updated. This overcomes the drawbacks of uniformly updating all models. Whenever a certain phase-specific statistical model is recognized to be no longer proper, it should be adjusted based on recent normal batches, while it is not necessary to update the other phase-representative models. Using the simple updating procedure, it is feasible to provide a valid monitoring result since the updated models are only utilized to monitor the neighboring future batches which should track similar process characteristics with the modeling batches and will not vary greatly if under normal conditions. Moreover, with the accumulation of new successful batches and the consecutive updating procedure, the monitoring models will acquire new process operation information and accommodate the common nature of slow-varying process behaviors.

**3.5. Fault Diagnosis.** Through statistical process monitoring charts, it is possible to detect abnormal variations of a new batch failing to be captured by the phase-specific monitoring models. However, the monitoring charts can only acquaint one with the occurrence of unusual events uninvolved in the NOC process regions and cannot provide the information on what is wrong with the process, or which measurement variables cause the failure behaviors. To more directly reveal the main causal variables responsible for nonconforming operation, the contribution of each process variable to process variations should be further interrogated. Instead of summing the statistic information of all process variables at each time when online monitoring, they only explore the individual effect of each separate process variable on statistic indexes. The contribution plot,[40,41] a commonly used diagnosis tool, has been used to check the underlying fault variables in PCA statistical monitoring. In the present work, based on eq 10, the contribution to the $I^2$ statistic shared by each process variable of the new observation can be defined in the similar way as the calculation of contribution to $T^2$ in PCA:[41]

$$C_{I^2,j} = \mathbf{s}_{\text{new}}{}^{\text{T}}\mathbf{M}_c{}^{-1}\mathbf{w}_j x_{\text{new},j} \qquad (j = 1, 2, ..., J) \qquad (11)$$

where $\mathbf{w}_j$ is the $j$th column of the current phase-specific demixing matrix $\mathbf{W}_c$, which corresponds to the concerned $j$th process variable. $x_{\text{new},j}$ is the $j$th element of the new process measurement vector $\mathbf{x}_{\text{new}}(J \times 1)$.

Moreover, the contribution to SPE[41] is also captured based on the same fact that the contributions summed over all $J$ process variables equal the relevant statistics of the new batch.

$$C_{\text{SPE},j} = (x_{\text{new},j} - \hat{x}_{\text{new},j})^2 \qquad (12)$$

where $x_{\text{new},j}$ and $\hat{x}_{\text{new},j}$ are the real measurement and estimated value of the $j$th process variable, respectively.

When online monitoring, it is usually easy to identify more specific locations of out-of-control behaviors in a process by the phase-based statistical analysis. Then, combining contribution plots, one can directly zoom in during this period to inspect the contribution of each process variable to the violated statistical values to find those responsible. This can indeed enhance the process understanding and help considerably in diagnosing the causal source of the abnormality.

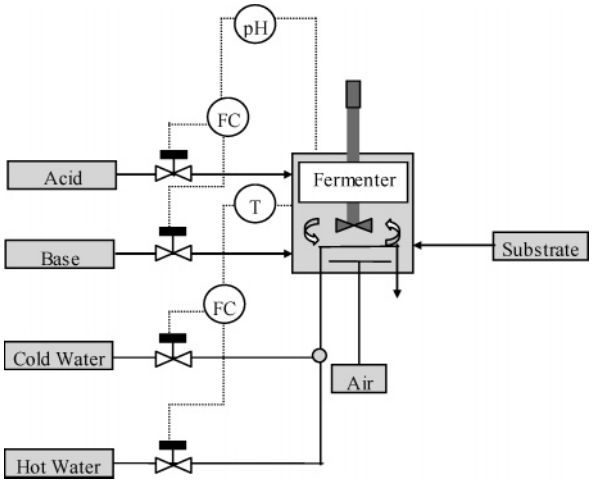


**Figure 2.** Flow diagram of the penicillin fermentation process.

## 4. Illustrations and Discussion

In this section, the proposed AMPICA algorithm is applied to the well-known benchmark simulation of fed-batch penicillin production.[42,43] A flow diagram of the penicillin fermentation process is given in Figure 2. The simulation results verify the effectiveness of the proposed method developed with limited batches from a relatively complex chemical industrial process of long duration.

**4.1. Fed-Batch Penicillin Fermentation.** The production of fed-batch penicillin fermentation is one of the most important chemical industrial cultivation processes. Due to its common nonlinear dynamics and multiphase characteristics during the production process, it has received wide attention as the subject of many studies showing its academic and industrial importance. A morphologically structured model has been formulated to describe trends in biomass and penicillin formation as well as substrate consumption for fed-batch cultivation.[42−44] In a typical operating procedure for the modeled fed-batch fermentation, most of the necessary cell mass is obtained during the initial

**Table 1. Variables Used in the Monitoring of the Benchmark Model**

| no. | 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 2. Summary of Fault Types Introduced at Different Times of Fermentation**

| fault no. | fault type | occurrence (h) |
|---|---|---|
| 1 | 15% step decrease in agitator power | 30 |
| 2 | 0.02 ramp decrease in substrate feed rate | 50 |
| 3 | 15% step decrease in substrate feed temperature | 20 |

preculture 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 due to the catabolite repressor effect. Consequently, glucose is fed continuously during fermentation instead of being added once at the beginning.

The simulation experiment is carried out using a simulator (PenSim v1.0 simulator) developed by the monitoring and control group of the Illinois Institute of Technology.[42] These simulations are run under closed-loop control of pH and temperature, while glucose addition is performed open-loop. The process variables selected for modeling in this work are listed in Table 1. The duration of each batch is 400 h, and the sampling interval is 1 h. The whole operation course consists of two physical operation stages: a preculture stage of about 45 h and a fed-batch stage of about 355 h.

Forty normal batch runs are generated, among which the first four batches are used for initial modeling, and the other 36 cycles are used for model updating. To mimic the common slow batch-to-batch variations, we artificially set the initial culture volume of these batches to gradually increase from 100 to 104 L over batches and the other initial setting conditions are default. Despite the gradual increase in initial culture volume, the batches are deemed to be normal, which will not drastically affect the correlations of process variables and actually is a true reflection of real process behaviors. That is, in a real process, the remainder culture from a previous batch tends to increase the initial culture volume of the coming batch. Moreover, to gain more insight into the performance of the proposed method for online monitoring and fault detection, three types of fault data are introduced respectively under each of the three abnormal conditions (faults 1−3) presented in Table 2.

**4.2. Illustration of AMPICA Algorithm and Discussions.** The initial monitoring models are developed based on data from only four normal batches. The length of the moving time window $l$ is set to be 33 samples, and the size of the generalized window length is thus $L = 33 \times 4 = 132$. After data normalization, the modeling data are arranged into variable-wise unfolding $\bar{\mathbf{X}}(1600 \times 11)$. According to the MPICA algorithm, the whole process is automatically divided into five phases (1−44, 45−92, 93−151, 152−297, 298−400), reflecting the changing trend of process underlying behaviors. The partitioned phases may not be exactly the same with the real

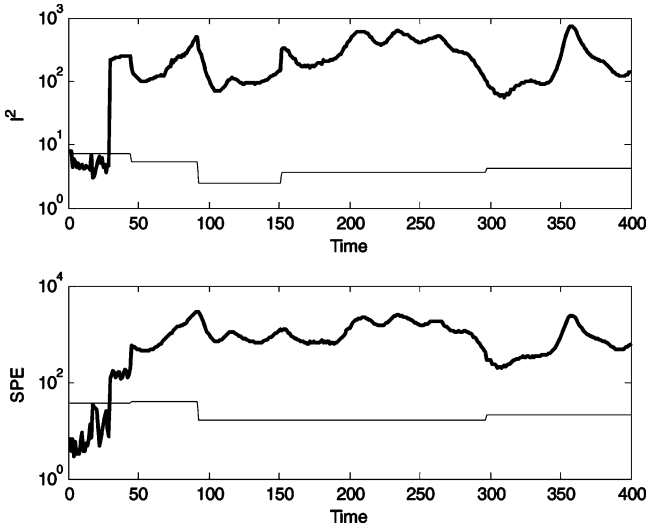


**Figure 3.** Monitoring results in the case of fault 1. (Solid line, 95% control limit; bold solid line, online monitoring statistics.)

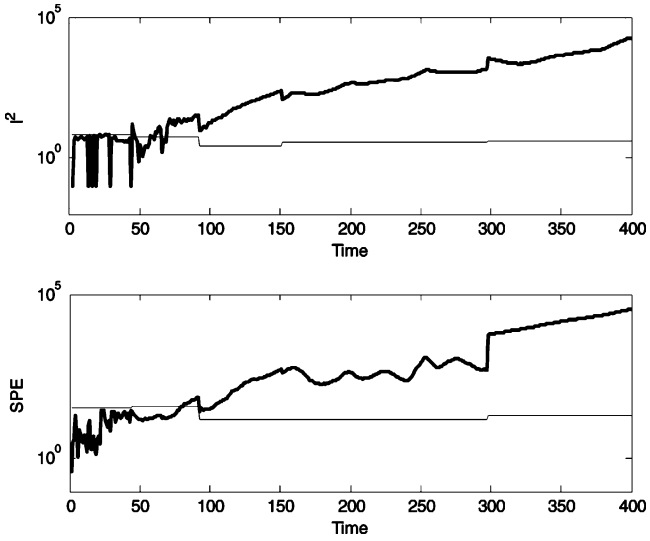


**Figure 4.** Monitoring results in the case of fault 2. (Solid line, 95% control limit; bold solid line, online monitoring statistics.)

operation stage in the fermentation cultivation but more focuses on the improvement of estimation ability of monitoring models. Without losing generality, each phase hints at different underlying process characteristics. Only three to five independent components are selected in each phase for the proposed modeling method.

Then the established ICA models are put into online testing of two fault batches and a normal batch. In the case of fault 1, a 15% step decrease in agitator power is introduced at 30 h and then maintained until the end of the batch. From the monitoring results shown in Figure 3, the abnormality can be clearly detected as $I^2$ and SPE values both go beyond the control limits immediately upon the occurrence of fault. The second process upset is generated by adding a 0.02 ramp increase in substrate feed rate from 50 h until the cultivation is over. From the monitoring results shown in Figure 4, both $I^2$ and SPE gradually increase from 50 h and then go beyond the confidence limits when approaching 70 h. The delay in the monitoring charts may result from the fact that the ramp increase of substrate feed rate may not change sharply the underlying process characteristics. In fact, its effect is gradually propagated to influence the systematic part and residual part after a period delay, where process variables also slowly depart from their expected

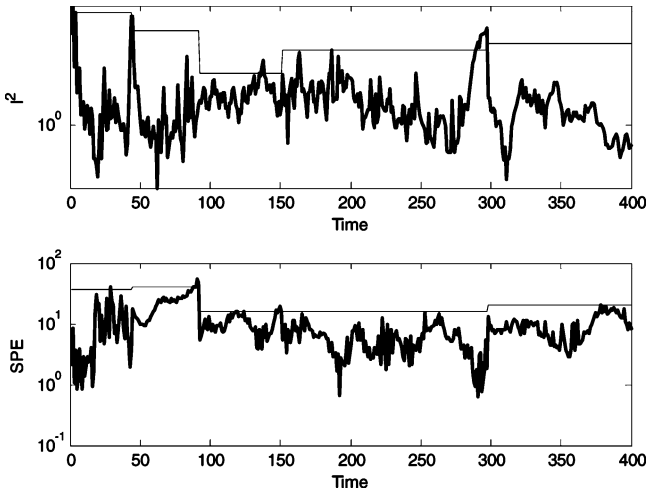


**Figure 5.** Online monitoring results for the first normal batch in updating database. (Solid line, 95% control limit; bold solid line, online monitoring statistics.)

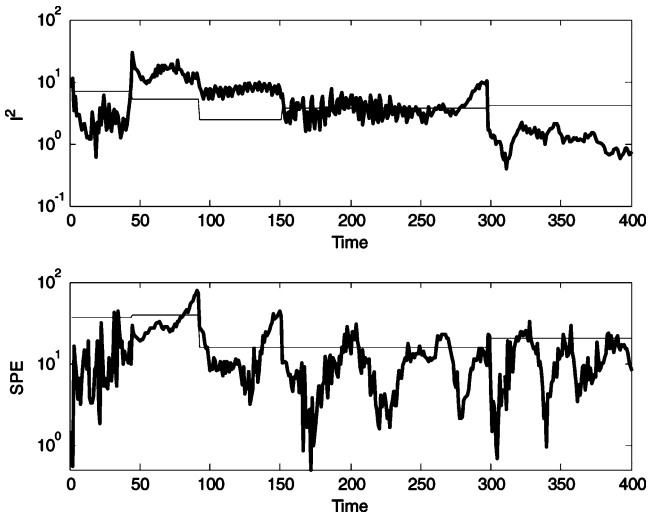


**Figure 6.** Online monitoring chart for the fifth normal batch without updating. (Solid line, 95% control limit; bold solid line, online monitoring statistics.)

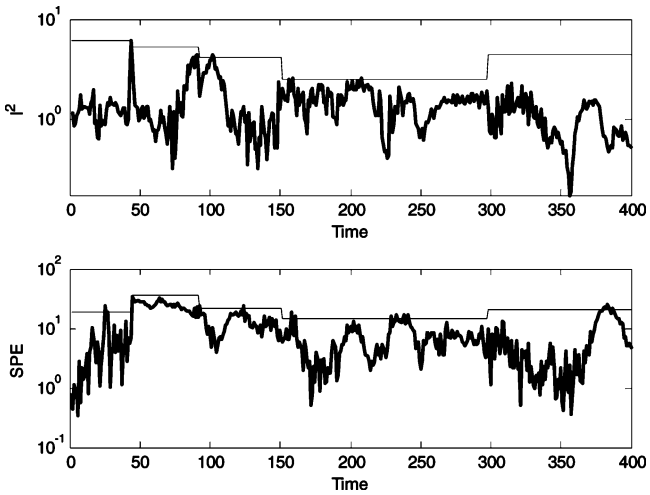


**Figure 7.** Online monitoring chart for the fifth normal batch using updated models. (Solid line, 95% control limit; bold solid line, online monitoring statistics.)

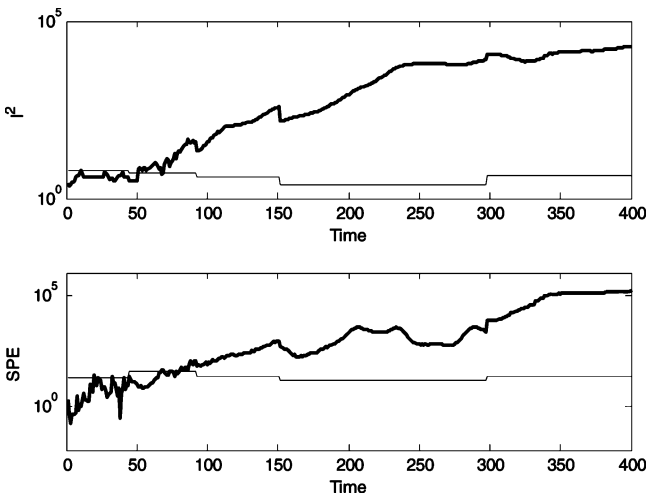


**Figure 8.** Online monitoring chart in the case of fault 2 using updated models. (Solid line, 95% control limit; bold solid line, online monitoring statistics.)

trajectories. Moreover, the monitoring result for the first normal batch in updating database is shown in Figure 5 using the initial models, where the values of $I^2$ and SPE statistics are both well below the control limits, indicating that the batch is free of any process abnormality. The above monitoring results illustratively demonstrate that the effectiveness of the initial monitoring models derived from limited batches, which ensures that, in the subsequent updating procedure, the models can be adjusted along a correct trend.

According to the proposed method, then the initial models are put into online adaptive updating based on the 36 updating batches in turn so that the new models and control limits are adjusted to accommodate more process information and adapt to the new operating status. Without updating, the process is first detected to be out of control at the fifth updating batch, as shown in Figure 6. However, it is well-known that it is a successful batch. Thus, the reliability of the old monitoring system is compromised. Figure 7 shows the monitoring results of the same normal batch using the updated models, which effectively eliminates the false alarms in Figure 6. Comparison between Figures 6 and 7 proves effectively that the proposed adaptive updating strategy can correctly capture the changing trend of process behaviors and accommodate the new operating information, which illustrates the necessity and significance of adaptive updating. The updated models are also put into fault detection. In the case of fault 2 presented in Figure 8, although the models are modified consecutively, the proposed updating strategy ensures that the models will not drift with the process abnormality, and thus yield good sensitivity to process faults. For the third abnormal batch, a step decrease is introduced into the substrate feed temperature at 20 h, which is maintained until the end of the batch. The updated models detect the deviation immediately after the onset of the fault since both of the monitoring indexes expose the process disorder with an abrupt increase outside the control limits, as shown in Figure 9. Therefore, we can naturally draw such a conclusion that the updated monitoring models not only eliminate the false alarms displayed in the monitoring charts of normal batches using the old models but also are sensitive to detect the real process disturbances.

Whenever the monitoring system is updated, seven fault batches with the current initial culture volume are generated under each of the three abnormal conditions (faults 1−3) presented in Table 2. Moreover, seven normal batches are also generated after each updating. The updated models are thus tested for both normal and fault cases before the next updating. Moreover, to gain more insight into the performance of the proposed method for batch process monitoring, it is compared

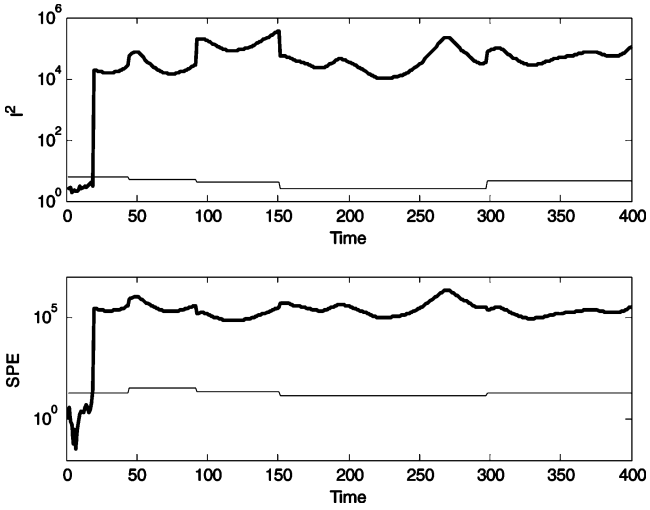


**Figure 9.** Online monitoring chart in the case of fault 3 using updated models. (Solid line, 95% control limit; bold solid line, online monitoring statistics.)

**Table 3. Summary of Monitoring Comparison Results During Five Updatings**

| | OTI (%) | | AST (%) | |
|---|---|---|---|---|
| updating no. | AMPICA | AMPPCA | AMPICA | AMPPCA |
| 1 | 2.5000 | 3.7500 | 3.7500 | 4.2143 |
| 2 | 1.9286 | 3.1071 | 2.7500 | 4.3929 |
| 3 | 1.7500 | 2.7857 | 3.3214 | 3.5357 |
| 4 | 2.0000 | 3.3571 | 2.2500 | 3.6429 |
| 5 | 1.6786 | 3.6786 | 2.2143 | 4.8929 |

with the adaptive algorithm based on multiphase PCA, here termed the AMPPCA algorithm using a naming rule similar to that for AMPICA. The monitoring results are contrastively summarized in Table 3 by means of two performance indices: the overall type I error (OTI)[5,9,35] and the relative action signal time (AST).[5,45] Following their definition, the OTI is the proportion of faults in the batches representing normal operation conditions: OTI = $nf/IK$, where $nf$ is the total number of the values of monitoring statistic outside the control limits in the NOC set. The AST is defined as the time elapsed between the introduction of an error and the out-of-control signal in the control chart. Here we calculate a relative AST value as a ratio between all the elapsed time and the total time length $IK$. Thus both the false alarms and the time delays in detecting fault can be availably taken into account. From the simulation results in Table 3 we can get a general and clear impression of the superior monitoring performance of the proposed method (AMPICA) compared with the AMPPCA algorithm. For example, using the AMPICA method, when there is no abnormality in the process, the monitoring statistical values stay well below the control limits throughout the process with few false alarms, indicating that the batch tracks a similar operation trajectory. For the abnormal operation, the proposed method can detect the fault with acceptable time delays. Compared with AMMPCA approaches, it generally shows better monitoring performance. All those efficiently support the effectiveness and reliability of the proposed algorithm.

Once the process upset is recognized, a contribution plot is employed to interrogate the assignable cause. Take, for example, fault 1, where the fault variable is variable 2 (agitator power); the contribution values of each process variable at different times are shown in Figure 10. Upon the recognition of process failure, it can be seen that agitator power (variable 2) has the largest contribution values to $I^2$, which agrees well with the real

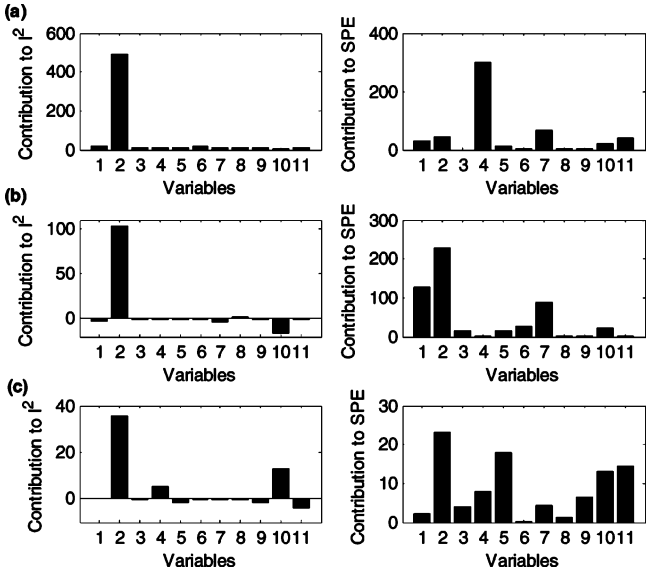


**Figure 10.** Contribution plots in the case of fault 1 at (a) 30, (b) 100, and (c) 300 h.

circumstances. In the case of the SPE diagnosis chart, although there are other variables which also possess large contribution rate, variable 2 overwhelms them over a period of time after the occurrence of fault. Thus, the diagnosis ability of the models is demonstrated illustratively.

The above-described experimental results indicate that the proposed modeling and monitoring scheme developed from limited batches can reveal process information, monitor process behaviors, and provide reasonable explanations for the detected faults.

## 5. Conclusions

In this study, an adaptive statistical modeling, monitoring, and updating algorithm based on ICA is proposed for slow batch processes, starting with limited cycles. Based on the use of generalized moving windows, data are normalized and then organized into variable-wise unfolding. The whole process duration is divided into different phases by executing the MPICA algorithm on the variable-unfolded matrix, and multiphase ICA models are thus developed for each phase. An adaptive algorithm is formulated focusing on each specific phase with the accumulation of newly available normal batches concerning when and how to update the old model structure. The proposed updating method can not only give a valid initial monitoring scheme with limited reference batches, but can also accommodate more process information and the slow time-varying behaviors over batches. The feasibility and effectiveness of the proposed idea and algorithm are successfully illustrated with respect to a complex fed-batch penicillin fermentation chemical batch process. As most chemical industrial processes experience lengthy duration as well as slow batch-to-batch variations, the developed statistical modeling and adaptive updating scheme will be promising and is expected to have broad applicability in practical batch processes. In conclusion, the AMPICA algorithm gives a fresh perspective to the modeling and monitoring problem for slow batch processes of lengthy duration. This paper has not explored in detail how well the proposed method can function in other industrial batch processes, which deserves further investigations.

## Acknowledgment

The authors would like to acknowledge the anonymous reviewers for their helpful comments. This work was supported in part by the National Natural Science Foundation of China (No. 60374003 and No. 60774068) and Project 973 (No. 2002CB312200), China.

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

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

(6) Kourti, T. Multivariate dynamic data modeling for analysis and statistical process control of batch processes, start-ups and grade transitions. *J. Chemom.* **2003**, *17*, 93.

(7) Wold, S.; Geladi, P.; Esbensen, K. Multi-way principal components-and PLS-analysis. *J. Chemom.* **1987**, *1*, 41.

(8) Nomikos, P.; MacGregor, J. F. Monitoring of batch processes using multi-way principal component analysis. *AIChE J.* **1994**, *40*, 1361.

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

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

(11) 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.

(12) Wang, X. Z. *Data mining and knowledge discovery for process monitoring and control*; Springer: London, 1999.

(13) Smilde, A. K. Comments on Three-Way Analyses used for Batch Process Data. *J. Chemom.* **2001**, *15*, 19.

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

(15) Hyvärinen, A.; Oja, E. A fast fixed-point algorithm for independent component analysis. *Neural Comput.* **1997**, *9*, 1483.

(16) Hyvärinen, A. The fixed-point algorithm and maximum likelihood estimation for independent component analysis. *Neural Process. Lett.* **1999**, *10*, 1.

(17) Hyvärinen, A. Survey on independent component analysis. *Neural Comput. Surv.* **1999**, *2*, 94.

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

(19) Girolami, M. *Self-organising neural networks: Independent component analysis and blind source separation*; Springer: London, 1999.

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

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

(22) 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.

(23) Lee, J.-M.; Yoo, C. K.; Lee, I.-B. Statistical process monitoring with multivariate exponentially weighted moving average and independent component analysis. *J. Chem. Eng. Jpn.* **2003**, *36*, 563.

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

(25) 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.

(26) Kano, M.; Tanaka, S.; Ohno, H.; Hasebe, S.; Hashimoto, I. The use of independent component analysis for multivariate process control. In *Proceedings of International Symposium on Advanced Control of Industrial Processes*, Kumamoto, Japan, June 2002; p 423.

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

(28) 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.

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

(30) Chen, J.; Wang, X. Z. A New Approach to Near-Infrared Spectral Data Analysis Using Independent Component Analysis. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 992.

(31) Johnson, R. A.; Wichern, D. W. *Applied multivariate statistical analysis*; Springer: London, 2002.

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

(33) 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.

(34) Zhao, C. H.; Wang, F. L.; Lu, N. Y.; Jia, M. X. Stage-based soft-transition multiple PCA modeling and on-line monitoring strategy for batch processes. *J. Process Control* **2007**, *17*, 728.

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

(36) Camacho, J.; Picó, J. Multi-phase principal component analysis for batch processes modeling. *Chemom. Intell. Lab. Syst.* **2006**, *81*, 127.

(37) Jensen, D. R.; Solomon, H. A. A Gaussian approximation to the distribution of a definite quadratic form. *J. Am. Stat. Assoc.* **1972**, *67*, 898.

(38) Zhao, C. H.; Wang, F. L.; Gao, F. R.; Lu, N. Y.; Jia, M. X. Adaptive Monitoring Method for Batch Processes Based on Phase Dissimilarity Updating with Limited Modeling Data. *Ind. Eng. Chem. Res.* **2007**, *46*, 4943.

(39) Qin, S. J. Recursive PLS algorithms for adaptive data modeling. *Comput. Chem. Eng.* **1998**, *22*, 503.

(40) Miller, P.; Swanson, R. E.; Heckler, C. E. Contribution plots: A missing link in multivariate quality control. *Appl. Math. Comput. Sci.* **1998**, *8*, 775.

(41) Westerhuis, J. A.; Gurden, S. P.; Smilde, A. K. Generalized contribution plots in multivariate statistical process monitoring. *Chemom. Intell. Lab. Syst.* **2000**, *51*, 95.

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

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

(44) Zangirolami, T. C.; Johansen, C. L.; Nielsen, J. Simulation of penicillin production in fed-batch cultivations using a morphologically structured model. *Biotechnol. Bioeng.* **1996**, *56*, 593.

(45) 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.

*Received for review* December 10, 2007
*Revised manuscript received* February 2, 2008
*Accepted* February 4, 2008

IE701680Y