Available online at www.sciencedirect.com

SCIENCE DIRECT®

Chemometrics and Intelligent Laboratory Systems 71 (2004) 151–163

Chemometrics and intelligent laboratory systems

www.elsevier.com/locate/chemolab

On-line monitoring of batch processes using multiway independent component analysis

Chang Kyoo Yoo[a,*], Jong-Min Lee[b], Peter A. Vanrolleghem[a,1], In-Beum Lee[b]

[a] *BIOMATH, Department of Applied Mathematics, Biometrics and Process Control, Ghent University, Coupure Links 653, B-9000 Gent, Belgium*
[b] *Department of Chemical Engineering, Pohang University of Science and Technology, San 31 Hyoja-Dong, Pohang 790-784, South Korea*

Received 30 January 2004; accepted 4 February 2004

Available online 12 April 2004

Abstract

Batch processes play an important role in the production of low-volume, high-value products such as polymers, pharmaceuticals, and biochemical products. Multiway principal component analysis (MPCA), a multivariate projection method, has been widely used to monitor batch processes. But in-control data of non-stationary processes in fact contain inherent non-Gaussian distributed data due to ramp changes, step changes, and even weak levels of autocorrelation. Monitoring charts obtained by applying MPCA to such non-Gaussian data may contain nonrandom patterns corresponding to the data characteristics. To obtain better monitoring performance in a batch process with non-Gaussian data, on-line batch monitoring method with multiway independent component analysis (MICA) is developed in this paper. MICA is based on a recently developed feature extraction method, called independent component analysis (ICA), whereas PCA looks for Gaussian components, whereas ICA searches for non-Gaussian components. MICA projects the multivariate data into a low-dimensional space defined by independent components (ICs). When the measured variables have non-Guassian distributions, MICA provides more meaningful statistical analysis and on-line monitoring compared to MPCA because MICA assumes that the latent variables are not Gaussian distributed. The proposed method was applied to the on-line monitoring of a fed-batch penicillin production. The simulation results demonstrate the power and advantages of MICA.

Keywords: Fault detection and diagnosis; Multiway independent component analysis (MICA); Multiway principal component analysis (MPCA); On-line batch process monitoring

1. Introduction

Batch processes play an important role in the production of low-volume, high-value products such as polymers, pharmaceuticals, and biological products. Most batch processes have predetermined starting and stopping points, and raw materials are introduced into the process in predefined amounts following a specific sequence. The manufacturing of a typical batch involves charging ingredients to the vessel, processing them under controlled conditions, and discharging the final product. A batch operation is considered successful if the values of the process variables remain within acceptable limits while following the recipe prescribed for the process, resulting in a uniform high-quality product. Unfortunately, some of the typical characteristics of batch processes complicate the on-line monitoring of such processes (e.g., finite duration, nonlinear behavior, and insufficient on-line sensors).

Small changes in the operating conditions during critical periods may degrade the quality and yield of the final product. Most batch processes exhibit batch-to-batch variations in the specified trajectory. These variations are of two types: normal variations due to common causes, and special variations due to unusual causes. However, product quality variables, the key indicators of process performance, are often examined off-line in a laboratory. The lack of on-line monitoring in most industrial batch processes means that, although operators may identify problems in product quality, they cannot recognize in advance the causes of the problems or when they occur. Therefore, the development of

* Corresponding author. Tel.: +32-9-264-5935, +32-9-264-6196; fax: +32-9-264-6220.

E-mail address: ChangKyoo.Yoo@biomath.UGent.be (C.K. Yoo).

[1] Tel.: +32-9-264-5935; fax: +32-9-264-6220.

0169-7439/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.chemolab.2004.02.002

effective methods for on-line monitoring and fault diagnosis of batch processes would significantly improve product quality because such methods would enable the detection of faults during process operation, making it possible to correct faults either prior to the completion of the batch or before the production of subsequent batches. Such early detection and correction of problems in batch processes would reduce the number of rejected batches.

Several techniques based on multivariate statistical analysis have been proposed for on-line monitoring and fault detection in batch processes. Nomikos and MacGregor [1,2] developed a new method known as multiway principal component analysis (MPCA) by extending the multivariate statistical process control (SPC) methods of principal component analysis (PCA) to batch processes. The key idea of their method is to compress the normal batch data and extract information by projecting the data onto a low-dimensional space that summarizes both the variables and their time trajectories. Having established the normal process behavior, the process of a given batch is then monitored by comparing the time progression of the projections in the reduced space with those of the normal batch data. The same authors also developed multiway partial least squares (MPLS) for monitoring processes for which both the process data and the product quality data are available [3]. Thus, MPCA and MPLS can be used to monitor batch processes if a model has been developed from nominal or good batch operation data. Numerous studies have investigated the application of MPCA to industrial batch monitoring [4–13]. Dong and McAvoy [14] used nonlinear principal component analysis (NLPCA) based on principal curves and neural networks to monitor batch processes. Rännar et al. [15] suggested an adaptive batch monitoring method using hierarchical PCA to overcome the need of estimating the missing data on trajectory deviation from the current time until the end of the batch in PCA.

It is well known that many of the variables monitored in process systems are not independent. The measured process variables may be combinations of independent variables that are not directly measurable (referred to as latent variables in probabilistic theory). Independent component analysis (ICA) can extract these underlying factors or components from multivariate statistical data. It defines a generative model for the observed multivariate data, which are typically in the form of a large database of samples. In this model, the data variables are assumed to be linear or nonlinear mixtures of some unknown latent variables, where the system governing the mixing of the latent variables is also unknown. The latent variables, which are called the independent components (ICs) of the observed data, are assumed to be non-Gaussian and mutually independent. ICA seeks to extract these ICs as well as the mixing process [16–18].

What distinguishes ICA from PCA is that it looks for components that are both statistically independent and non-Gaussian. PCA is a dimensionality reduction technique in terms of capturing the variance of the data which is capable of extracting uncorrelated latent variables from correlated data, while ICA is designed to separate the independent components (ICs) that are independent and constitute the observed variables. Furthermore, PCA can only impose independence up to second-order statistics information (mean and variance) while constraining the direction vectors to be orthogonal, whereas ICA has no orthogonality constraint and also involves higher-order statistics. In case of ICA, the most interesting directions are those that show the non-Gaussian distributions, whereas the directions of PCA are looking for the Gaussian distribution. Thus, both models have a different projection pursuit. ICA may reveal more meaningful information in the non-Gaussian data than PCA [18–20].

In this paper, we propose on-line batch monitoring using multiway independent component analysis (MICA) in order to obtain better monitoring performance in the batch process with non-Gaussian data. This article is organized as follows. MPCA is introduced in Section 2. In Section 3, the on-line batch monitoring method of MICA is described. The simulation results in a fed-batch penicillin production are given in Section 5. Finally, we present our conclusions in Section 6.

2. MPCA

MPCA is used for the analysis and monitoring of batch process data. Batch data are typically reported in terms of batch numbers, variables and times. Data are arranged into a three-dimensional matrix $\underline{\mathbf{X}}$ $(I \times J \times K)$, where I is the number of batches, J is the number of variables and K is the number of times each batch is sampled. This matrix can be decomposed using various three-way techniques, one of which is MPCA. MPCA is equivalent to performing ordinary PCA on a large two-dimensional matrix $\mathbf{X}$ constructed by unfolding the three-way data in the manner shown schematically in Fig. 1 [1].

MPCA decomposes the three-way array $\underline{\mathbf{X}}$ into a summation of the product of a score t_r and a loading matrix P_r plus a residual array $\underline{\mathbf{E}}$ that is minimized in the least squares sense as follows:

$$\underline{\mathbf{X}} = \sum_{r=1}^{R} t_r \otimes P_r + \underline{\mathbf{E}} = \sum_{r=1}^{R} t_r p_r^T + \mathbf{E} = \hat{\mathbf{X}} + \mathbf{E} \qquad (1)$$

where $\otimes$ denotes the Kronecker product ($\underline{\mathbf{X}} = t \otimes P$ is $\underline{\mathbf{X}}(i,j,k) = t(i)P(j,k)$), R denotes the number of principal components retained, t_r expresses the relationship among batches, p_r is related to variables and their time variation, $\mathbf{E}$ is the residual matrix. The first expression in Eq. (1) gives the 3-D decomposition while the second expression displays the more common 2-D decomposition.

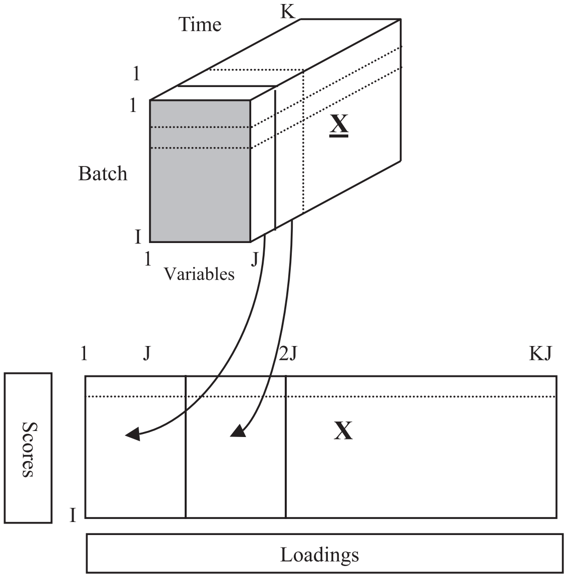

Fig. 1. Multiway unfolding of a three-way batch data set.

The three-way array $\underline{\mathbf{X}}$ can be unfolded in three ways, which give rise to the following two-dimensional matrices [6,14]:

- Batches × variables at each time (time-wise unfolding)
- Variables × time for each batch (batch-wise unfolding)
- Batches × times for each variable (variable-wise unfolding)

Time-wise unfolding is useful for analyzing the variability among samples, and batch-wise unfolding facilitates the analysis of the variability among batches by summarizing the information related to the measured variables and their variations over time [1]. Variable-wise unfolding can be used to obtain information about the variability among the batch variables [21,22]. To facilitate comparison with the MPCA method, we use the batch-wise unfolding scheme of the MICA method in this paper. Fig. 1 shows a schematic of the batch-wise unfolding method.

For on-line batch monitoring using MPCA, we know only the values from the start of batch to the current time; however, for on-line monitoring the incomplete set of real batch process should be augmented with predicted data to create a data set spanning the entire batch. Nomikos and MacGregor [2] suggested three different ways for variable trajectory estimation, i.e. to complete the remaining of the batches: (1) zero deviation, (2) current deviation, (3) PCA projection method. The choice of the most suitable approach depends on the characteristics of the batch process, but the second or the third filling method suggested by Nomikos and MacGregor [2] is mainly used.

The statistics used for monitoring multivariable batch processes are D-statistic and Q-statistic. The D-statistic, which is similar to the Hotelling's T^2 statistic, is a Mahalanobis distance between new data and the center of the normal operating condition data in a reduced dimension. The pattern of the residuals is monitored using the Q-statistic, which is a summation of the squared prediction error (SPE). The D-statistic monitors systematic variations in the principle component (PC) subspace, while the Q-statistic represents variations not explained by the retained PCs. That is, faults in the process that violate the normal correlation of variables are detected in the PC subspace by the D-statistic, whereas faults that violate the PCA models are detected in the residual space by the Q-statistic. At the end of a batch, D- and Q-statistics for batch i are calculated as follows:

$$D_i = \mathbf{t}_i^T \mathbf{S}_t^{-1} \mathbf{t}_i \sim \frac{R(I^2-1)}{I(I-R)} F_{R,I-R} \tag{2}$$

$$Q_i = \boldsymbol{e}_i \boldsymbol{e}_i^T = \sum_{c=1}^{KJ} \boldsymbol{E}(i,c)^2 \tag{3}$$

where $\mathbf{e}_i$ is the i-th row of $\mathbf{E}$, I is the number of batches in the reference set, $\mathbf{t}_i$ is a vector of R scores, $\mathbf{S}_t$ is the ($R\times R$) covariance matrix of the t-scores calculated during the model development, which is diagonal due to the orthogonality of the t score values, R is the number of principal components retained in the model, and $F_{R,I-R}$ is the F-distribution value with R and I-R degrees of freedom. The statistical limits on the D- and Q-statistics are computed by assuming that the data have a multivariate normal distribution. Weighting the distance by the inverse of the covariance matrix $\mathbf{S}_t$ accounts for differences in variation and the presence of correlations. Variables that have a larger normal variation are weighted less in D-statistic [23].

To calculate the D-statistic for on-line batch monitoring, the expression in Eq. (2) is implemented for each observation at time instant k of batch i.

$$D_i(k) = (\mathbf{t}_i^T)_k \mathbf{S}_t^{-1} (\mathbf{t}_i)_k \sim \frac{R(I^2-1)}{I(I-R)} F_{R,I-R} \tag{4}$$

where $(\mathbf{t}_i)_k$ is calculated from the filled observation vector. The SPE is a quadratic form of the errors at time interval k.

$$\begin{aligned}\mathrm{SPE}_i(k) &= \sum_{j=1}^{J} (x_{ij}(k) - \hat{x}_{ij}(k))^2 = \sum_{j=1}^{J} (\mathrm{e}_{ij}(k))^2 \\ &= \sum_{c=(k-1)J+1}^{Jk} \mathbf{e}_c^T \mathbf{e}_c \end{aligned} \tag{5}$$

where $\mathbf{e}_c$ is the c^{th} column of the matrix $\mathbf{E} = \mathbf{X}_{\text{new}} - \mathbf{t}_{\text{new}}\mathbf{P}^T$. The distribution of the SPE can be approximated by a weighted χ^2 distribution. The confidence limit of SPE can be obtained from the above approximate chi-squared distribution with a significance level of α, $\mathrm{SPE}_{k,\alpha}=(v_k/2m_k)\chi^2_{2m_k^2/v_k,\alpha}$, where m_k and v_k are the mean and variance of the SPE obtained for the data set used for the model development at time instant k and $\chi^2_{2m_k^2/v_k,\alpha}$ is the critical value of the χ^2 variable with $2m_k^2/v_k$ degrees of freedom at significance level α [23].

3. Batch monitoring with multiway ICA

3.1. Independent component analysis (ICA)

To introduce the ICA algorithm, it is assumed that d measured variables, $\mathbf{x}(k)$=$[x_1(k),\ldots,x_d(k)]^T$, at sample k can be expressed as linear combinations of m ($\leq d$) unknown independent components $s_1, s_2,\ldots,s_m$.

$$\mathbf{x}(k) = \sum_{j=1}^{m} \mathbf{a}_j s_j(k) = \mathbf{A}\mathbf{s}(k) \tag{6}$$

The ICs and the measured variables have means of zero. The relationship between them is given by

$$\mathbf{X} = \mathbf{AS} + \mathbf{E} \tag{7}$$

where $\mathbf{X}$=$[\mathbf{x}(1),\mathbf{x}(2),\ldots,\mathbf{x}(n)]\in R^{d\times n}$ is the data matrix, $\mathbf{A}$= $[\mathbf{a}_1,\ldots,\mathbf{a}_m]\in R^{d\times m}$ is the mixing matrix, $\mathbf{S}$=$[\mathbf{s}(1),\mathbf{s}(2),\ldots,\mathbf{s}(n)]\in R^{m\times n}$ is the independent component matrix, $\mathbf{E}\in R^{d\times n}$ is the residual matrix, and n is the number of samples. Here, we assume that $d\geq m$ (when $d=m$, the residual matrix, $\mathbf{E}$, becomes the zero matrix). The basic problem of ICA is to estimate the original components $\mathbf{S}$ or to estimate $\mathbf{A}$ from $\mathbf{X}$ without any knowledge of $\mathbf{S}$ or $\mathbf{A}$. Therefore, the objective of ICA is to calculate a separating matrix $\mathbf{W}$ so that the components of the reconstructed data matrix $\hat{\mathbf{S}}$, given as

$$\hat{\mathbf{S}} = \mathbf{WX} \tag{8}$$

become as independent of each other as possible. Using the ICA algorithm, we can obtain the rows of $\hat{\mathbf{S}}$ whose norm is 1 [18].

Below we assume that d equals m except where specified otherwise. The initial step in ICA is whitening (also known as sphering) which eliminates all the cross-correlations among random variables. Consider a d-dimensional random vector $\mathbf{x}(k)$ at sample k with covariance $\mathbf{R_x}=E(\mathbf{x}(k)\mathbf{x}^T(k))$. The eigen-decomposition of $\mathbf{R_x}$ is given by

$$\mathbf{R_x} = \mathbf{U}\Lambda\mathbf{U}^T. \tag{9}$$

The whitening transformation is expressed as

$$\mathbf{z}(k) = \mathbf{Qx}(k) \tag{10}$$

where $\mathbf{Q}=\Lambda^{-1/2}\mathbf{U}^T$. One can easily verify that $\mathbf{R_z}$=$E(\mathbf{z}(k)\mathbf{z}^T(k))$ is the identity matrix under this transformation. After the whitening transformation we have

$$\mathbf{z}(k) = \mathbf{Qx}(k) = \mathbf{QAs}(k) = \mathbf{Bs}(k) \tag{11}$$

where $\mathbf{B}$ is an orthogonal matrix, as verified by the following relation:

$$E\{\mathbf{z}(k)\mathbf{z}^T(k)\} = \mathbf{B}E\{\mathbf{s}(k)\mathbf{s}^T(k)\}\mathbf{B}^T = \mathbf{BB}^T = \mathbf{I}. \tag{12}$$

We have therefore reduced the problem of finding an arbitrary full-rank matrix $\mathbf{A}$ to the simpler problem of finding an orthogonal matrix $\mathbf{B}$. Then, from Eq. (11), we can estimate $\mathbf{s}(k)$ as follows

$$\hat{\mathbf{s}}(k) = \mathbf{B}^T\mathbf{z}(k) = \mathbf{B}^T\mathbf{Qx}(k). \tag{13}$$

From Eqs. (8) and (13), the relation between $\mathbf{W}$ and $\mathbf{B}$ can be expressed as

$$\mathbf{W} = \mathbf{B}^T\mathbf{Q}. \tag{14}$$

To calculate $\mathbf{B}$, each column vector $\mathbf{b}_i$ is initialized and then updated so that the i-th independent component $\hat{s}_i$=$(\mathbf{b}_i)^T\mathbf{z}$ may have considerable non-Gaussianity. There are two classic measures of non-Gaussianity: kurtosis and negentropy. Kurtosis is the fourth-order cumulant of a random variable kurt$(y)=E\{y^4\}-3\{E\{y^2\}\}^2$. Non-Gaussianity can be measured by the absolute value of the kurtosis. These measures are zero for a Gaussian variable, and larger than zero for most non-Gaussian random variables. However, kurtosis calculation is sensitive to outliers. On the other hand, negentropy is based on the information-theoretic quantity of (differential) entropy. Entropy is a measure of the average uncertainty in a random variable and the differential entropy H of random variable y with density $f(y)$ is defined as

$$H(y) = -\int f(y)\log f(y)dy \tag{15}$$

A Gaussian variable has the largest entropy among all random variables with equal variance [16]. In order to obtain a measure of non-Gaussianity that is zero for a Gaussian variable, the negentropy J is defined as follows

$$J(y) = H(y_{\text{gauss}}) - H(y) \tag{16}$$

where y_{gauss} is a Gaussian random variable with the same variance as y. Negentropy is nonnegative and measures the departure of y from Gaussianity [18]. However, estimating negentropy using Eq. (16) would require an estimate of the probability density function. To estimate negentropy efficiently, Hyvärinen [16,17] suggested simpler approximations of negentropy as follows:

$$J(y) \approx [E\{G(y)\} - E\{G(v)\}]^2 \tag{17}$$

where y is assumed to be of zero mean and unit variance, v is a Gaussian variable of zero mean and unit variance, and G

is any non-quadratic function. By choosing G wisely, one obtains good approximations of negentropy. Hyvärinen [17] suggested a number of functions for G:

$$G_1(u) = \frac{1}{a_1}\operatorname{logcosh}(a_1 u) \tag{18}$$

$$G_2(u) = \exp(-a_2 u^2/2) \tag{19}$$

$$G_3(u) = u^4 \tag{20}$$

where $1 \le a_1 \le 2$ and $a_2 \approx 1$. Among these three functions, G_1 is a good general-purpose contrast function and was therefore selected for use in the present study.

Based on the approximate forms for negentropy, Hyvärinen [16–18] introduced a very simple and efficient fixed-point algorithm for ICA, calculated over sphered zero-mean vectors $\mathbf{z}$. This algorithm calculates one column of the matrix $\mathbf{B}$ and allows the identification of one independent component; the corresponding IC can then be found using Eq. (13). The algorithm is repeated to calculate each independent component. The algorithm is as follows,

1. Choose m, the number of ICs to estimate. Set counter $i \leftarrow 1$.
2. Take a random initial vector $\mathbf{b}_i$ of unit norm.
3. Let $\mathbf{b}_i \leftarrow E\{\mathbf{z}g(\mathbf{b}_i^T\mathbf{z})\} - E\{g'(\mathbf{b}_i^T\mathbf{z})\}\mathbf{b}_i$, where g is the first derivative and g' is the second derivative of G, where G takes the form of Eqs. (18), (19) or (20).
4. Perform the following orthogonalization: $\mathbf{b}_i \leftarrow \mathbf{b}_i - \Sigma_{j=1}^{i-1}(\mathbf{b}_i^{\mathrm{T}}\mathbf{b}_j)\mathbf{b}_j$
5. Normalize $\mathbf{b}_i \leftarrow \frac{\mathbf{b}_i}{\|\mathbf{b}_i\|}$
6. If $\mathbf{b}_i$ has not converged, go back to Step 3.
7. If $\mathbf{b}_i$ has converged, output the vector $\mathbf{b}_i$. Then, if $i \le m$ set $i \leftarrow i+1$ and go back to Step 2.

Note that the final vector $\mathbf{b}_i$ $(i=1,\ldots,m)$ given by the algorithm equals one of the columns of the (orthogonal) mixing matrix $\mathbf{B}$. After calculating $\mathbf{B}$, we can obtain $\hat{\mathbf{s}}(k)$ and demixing matrix $\mathbf{W}$ from Eqs. (13) and (14), respectively [16,19].

3.2. Ordering and dimension reduction of ICA

In chemical and biological processes, the measured variables are quantitative (e.g., temperature, pressure, and flow rate) and qualitative (e.g., key component concentration). Dimension reduction in ICA is based on the idea that these measured variables are the mixture of some independent variables [1,19]. The performance and interpretation of ICA monitoring depends on the correct choice of the ordering and dimension of the ICA model. Unlike PCA, there is no standard criterion for ordering of ICs, which complicates the ordering procedure. A number of methods for ordering ICs have been suggested. In the present study, we used the simple approach of sorting the rows of the demixing matrix, $\mathbf{W}$, on the basis of their Euclidean norms (L_2), where the L_2 norm of row $\mathbf{w}_i$ of $\mathbf{W}$ is [24]:

$$\arg\max_i \|\mathbf{w}_i\|_2 \tag{21}$$

That is, the ICs are sorted using an L_2 norm in order to show only those ICs that cause dominant changes in the process.

Once the ICs have been ordered, it is necessary to select the optimal number of ICs to be used for monitoring. This step is crucial because selecting too many ICs will magnify the noise and too small ICs will not be insufficient to catch the dominant characteristics of process, leading to poor monitoring performance. The data dimension can be reduced by selecting the first few rows of the ordered $\mathbf{W}$ based upon the assumption that the rows with the largest Euclidean norm have the greatest effect on the variation of $\mathbf{S}$. This approach is based on the idea that the dominant variation in a process can be monitored by considering the cumulative sums of only the first few dominant ICs. We used a graphical technique to determine the number of ICs similar to the SCREE test of PCA [19,25].

3.3. Multiway independent component analysis (MICA)

The monitoring method based on MICA is similar to that based on MPCA. MICA is equivalent to performing ICA on a large two-dimensional matrix $\mathbf{X}$ constructed by batchwise unfolding the three-way data matrix $\underline{\mathbf{X}}$ in the manner shown in Fig. 1. MICA decomposes the three-way array $\underline{\mathbf{X}}$ into a summation of the product of independent vectors $\mathbf{s}_r$ and loading matrices $\mathbf{A}_r$ plus a residual array $\underline{\mathbf{E}}$ so that the ICs $\mathbf{s}$ become as independent of each other as possible:

$$\underline{\mathbf{X}} = \sum_{r=1}^{R} \mathbf{s}_r \otimes \mathbf{A}_r + \underline{\mathbf{E}} = \sum_{r=1}^{R} \mathbf{s}_r \mathbf{a}_r^T + \mathbf{E} = \hat{\mathbf{X}} + \mathbf{E} \tag{22}$$

where $\otimes$ denotes the Kronecker product ($\underline{\mathbf{X}} = \mathbf{s} \otimes \mathbf{A}$ is $\underline{\mathbf{X}}(i,j,k) = s(i)A(j,k)$) and R denotes the number of ICs retained. The $\mathbf{S}$ and $\mathbf{A}$ matrices in Eq. (22) can be equivalent to the score matrix and loading matrix by analogy with MPCA, i.e. $\mathbf{S}$ can be regarded as the score matrix $\mathbf{T}$, and $\mathbf{A}$ can be treated as the loading matrix $\mathbf{P}$. The i-th elements of the independent vector $\mathbf{s}$ correspond to the i-th batch and summarize the overall variations in this batch with respect to the other batches over the entire history of the batch. The mixing matrix, $\mathbf{A}$, summarizes the time variations of the measured variables about their average trajectories; the elements of this matrix are the weights, which give the independent vectors $\mathbf{s}$ for a batch when applied to each variable at each time interval within that batch. “In MICA, an unfolded and scaled data matrix, $\mathbf{X}(\boldsymbol{JK} \times \boldsymbol{I})$, representing multiple batch runs is decomposed, $\mathbf{X} = \mathbf{U}_r \mathbf{\Lambda}_r^{1/2} \mathbf{Z}_r$, and (r x r) orthogonal rotation matrix, $\mathbf{B}$, is computed to produce rotated scores, $\mathbf{A}=\mathbf{U}_r\mathbf{\Lambda}_r^{1/2}\mathbf{B}$ and loadings, $\mathbf{S}=(\mathbf{B}^T\mathbf{Z}_r)$ giving $\mathbf{X}=\mathbf{U}_r\mathbf{\Lambda}_r^{1/2}\mathbf{B}\mathbf{B}^T\mathbf{Z}_r$, such that $\mathbf{B}\mathbf{B}^T=\mathbf{I}$."

In MICA, two types of statistics are deduced from the process model in normal operation: the *D*-statistic for the systematic part of the process variation and the *Q*-statistic for the residual part of the process variation. The *D*-statistic for batch *i*, also known as the I^2 statistic, is the sum of the squared independent scores and is defined as follows:

$$I^2(i) = \hat{\mathbf{s}}_d(i)^T\hat{\mathbf{s}}_d(i) \tag{23}$$

where $\hat{\mathbf{s}}_d(i)$ is the *i*-th column vector of $\hat{\mathbf{s}}_d$. The *i*-th element of the IC matrix represents the *i*-th batch and summarizes the overall variation in this batch with respect to the other batches in the historical database over its entire duration.

The *Q*-statistic for a batch *i*, also known as the SPE statistic, is defined as follows:

$$\mathrm{SPE}(i) = \mathbf{e}(i)^T\mathbf{e}(i) = (\mathbf{x}(i)-\hat{\mathbf{x}}(i))^T(\mathbf{x}(i)-\hat{\mathbf{x}}(i)) \tag{24}$$

where $\hat{x}$ can be calculated as follows:

$$\hat{\mathbf{x}}(i) = \mathbf{Q}^{-1}\mathbf{B}_d\hat{\mathbf{s}}(i) = \mathbf{Q}^{-1}\mathbf{B}_d\mathbf{W}_d\mathbf{x}(i) \tag{25}$$

Here, the I^2 value is used to detect faults associated with abnormal variations within an MICA model subspace, whereas the SPE value is used to detect new events that are not taken into account in an MICA model subspace.

In MPCA monitoring, the confidence limit is based on a specified distribution shown in Eqs. (4) and (5) based upon the assumption that the latent variables follow a Gaussian distribution. In MICA monitoring, however, the independent components do not conform to a specific distribution; hence, the confidence limits of the I^2 and SPE statistics cannot be determined directly from a particular approximate distribution. An alternative approach to defining the nominal operating regions is to use data-driven techniques such as non-parametric empirical density estimates using kernel density estimation (KDE) [12,26].

A univariate kernel estimator with kernel *K* is defined by:

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n}K\left\{\frac{x-x_i}{h}\right\} \tag{26}$$

where x is the data point under consideration, x_i is an observation value from the data set, *h* is the window width (also known as the smoothing parameter), *n* is the number of observations, and *K* is the kernel function. On the other hand, the disadvantages of kernel density estimation need a relatively large data set and are the necessity of the tuning (the selection of smoothing parameter). The problem of choosing how much to smooth is of crucial importance in density estimation. Many measures have been proposed for the estimation of *h*, the window width or smoothing parameter. If *h* is too large we oversmooth, erasing detail. If *h* is too small we undersmooth, and fail to filter out spurious detail. In this paper, the confidence limits of the two statistics, I^2 and SPE were obtained by kernel density estimation, where the Gaussian kernel and the least squares cross-validation (LSCV) methods for selecting *h* were used [27].

3.4. Off-line and on-line batch monitoring of the MICA model

Following are the detailed procedure of off-line and on-line monitoring method using the MICA model to supervise the batch process.

3.4.1. (A) Develop the normal operating condition (NOC) model

1. Acquire an operating data set during normal batch operation.
2. Unfold $\underline{\mathbf{X}}$ $(I\times J\times K)$ to $\mathbf{X}(I\times JK)$ using batch-wise unfolding scheme.
3. Normalize the data $\mathbf{X}(I\times JK)$ using the mean and standard deviation of each variable at each time in the batch cycle over all batches.
4. Transpose the scaled $\mathbf{X}(I\times JK)$. The transposed matrix is designated $\mathbf{X}_{\mathrm{normal}}(JK\times I)$.
5. Apply the whitening procedure to acquire the uncorrelated whitened matrix

   $$\mathbf{Z}_{\mathrm{normal}} = \mathbf{Q}\mathbf{X}_{\mathrm{normal}} \tag{27}$$

   where $\mathbf{Z}_{\mathrm{normal}}=[\mathbf{z}(1),\mathbf{z}(2),\ldots,\mathbf{z}(I)]\in R^{r\times I}$, Here, we can extract r columns of U in Eq. (9), *r* is the rank of the covariance matrix of $\mathbf{X}_{\mathrm{normal}}$.
6. Carry out the ICA procedure to obtain the following matrices **W**, **B**, and $\hat{\mathbf{S}}_{\mathrm{normal}}$ so that $\hat{\mathbf{S}}(n)=\mathbf{B}^T\mathbf{z}(n)$ has great non-Gaussianity.

   $$\hat{\mathbf{S}}_{\mathrm{normal}} = \mathbf{W}\mathbf{X}_{\mathrm{normal}} = \mathbf{B}^T\mathbf{Z}_{\mathrm{normal}} \tag{28}$$

   where $\mathbf{A}=(\mathbf{Q}^T\mathbf{Q})^{-1}\mathbf{Q}^T\mathbf{B}$ and $\mathbf{W}=\mathbf{B}^T\mathbf{Q}$.
7. Apply the ordering and dimension reduction of ICA. Thus, the dimension of the unfolded data matrix is reduced by selecting a few rows of **W** based upon the assumption that the rows with the largest sum of squares coefficient have the greatest effect on the variation of $\hat{\mathbf{S}}$. The *m* rows of **W** which are separated into deterministic part of **W** ($\mathbf{W}_d$), and the excluded part of **W**, ($\mathbf{W}_e$). The resulting matrices have the following forms:

   $$\hat{\mathbf{S}}_{\mathrm{normal}} = \begin{bmatrix}\hat{\mathbf{S}}_d\\ \hat{\mathbf{S}}_e\end{bmatrix} = \begin{bmatrix}\mathbf{W}_{\mathrm{d}}\mathbf{X}_{\mathrm{normal}}\\ \mathbf{W}_{\mathrm{e}}\mathbf{X}_{\mathrm{normal}}\end{bmatrix} \tag{29}$$

   $$\mathbf{A} = \begin{bmatrix}\mathbf{A}_d\\ \mathbf{A}_e\end{bmatrix} \tag{30}$$

   $$\mathbf{W} = \begin{bmatrix}\mathbf{W}_d\\ \mathbf{W}_e\end{bmatrix} \tag{31}$$

8. Finally, the MICA model is constructed. For each batch of $\mathbf{X}(I \times JK)$, $\mathbf{x}'$ $(1 \times JK)$ is projected into the reduced space of the MICA model. For all I batches, the ICs and residuals are calculated from $\hat{\mathbf{S}}_{\rm d}$ and SPE.
9. Calculate the I^2 statistic:

$$I^2(i) = \hat{\mathbf{s}}_d(i)^T \hat{\mathbf{s}}_d(i) \tag{23}$$

where $1 \leq i \leq$ I.

10. Calculate the Q-statistic for a batch i, also known as the SPE statistic, defined as follows:

$$\mathrm{SPE}(i) = \mathbf{e}(i)^T \mathbf{e}(i) = (\mathbf{x}(i) - \hat{\mathbf{x}}(i))^T (\mathbf{x}(i) - \hat{\mathbf{x}}(i)) \tag{33}$$

with Eq. (25)

11. Obtain the control limits of the I^2 and SPE statistics for off-line batch analysis using kernel density estimation.
12. Obtain the control limits of the SPE at each time for on-line monitoring. For batch i and time k, we have only $\mathbf{x}_{\rm normal}(kJ \times 1)$. For a vector filled with the future observation, $\mathbf{x}_{\rm pred}(JK \times 1)$

$$\mathrm{SPE}(i,k) = \sum_{j=(k-1)J+1}^{kJ} (\mathbf{x}_{\rm predj} - \hat{\mathbf{x}}_{\rm predj})^2 \tag{34}$$

where $\hat{\mathbf{x}}_{\rm pred} = \mathbf{A}_d \mathbf{W}_d \mathbf{x}_{\rm pred}$ and $\mathbf{x}_{\rm predj}$ is the j-th element of $\mathbf{x}_{\rm predj}$.

3.4.2. (B) On-line monitoring

1. For new batch data up to time k, $\mathbf{X}_{\rm test}(k \times J)$, unfold it to $\mathbf{x}_{\rm test}(kJ \times 1)$. Apply the same scaling used in the modeling and fill-in the missing values to create $\mathbf{x}_{\rm pred_test}(JK \times 1)$using one of the three approaches suggested by Nomikos and MacGregor [2]. The vector filled with future observations, $\mathbf{x}_{\rm pred\_test}(JK \times 1)$, is created by the selected filling method.
2. For the scaled and filled matrix, $\mathbf{x}_{\rm pred\_test}(JK \times 1)$, calculate the ICs of $\mathbf{s}_{\rm testd}$

$$\hat{\mathbf{s}}_{\rm testd}(k) = \mathbf{W}_d \mathbf{x}_{\rm pred\_test} \tag{35}$$

3. Calculate $I_{\rm testd}{}^2(k)$ and SPE(k)

$$I^2_{\rm testd}(k) = \mathbf{s}_{\rm testd}(k)^T \mathbf{s}_{\rm testd}(k) \tag{36}$$

$$\mathrm{SPE}(k) = \sum_{j=(k-1)J+1}^{kJ} (\mathbf{x}_{\rm predj} - \hat{\mathbf{x}}_{\rm predj})^2 \tag{37}$$

where $\hat{\mathbf{x}}_{\rm pred} = \mathbf{A}_d \mathbf{W}_d \mathbf{x}_{\rm pred}$ and $\mathbf{x}_{\rm predj}$ is the j-th element of $\mathbf{x}_{\rm predj}$.

Once a fault is detected by the statistical monitoring method, the key to fault isolation using the MICA model is the use of contribution plots. By interrogating the underlying process model at the point where an event has been detected, contribution plots may reveal the group of process variables that most influence the model or the residuals. We suggest the following equations for calculating the variable contribution plot for $I^2_{\rm newd}(k)$ and SPE(k).

(a) Variable contribution for $I_{\rm testd}{}^2(k)$

$$\mathbf{x}_{cd}(k) = \frac{\mathbf{A}_d \hat{\mathbf{s}}_{\rm testd}(k)}{\| \mathbf{A}_d \hat{\mathbf{s}}_{\rm testd}(k) \|} \| \hat{\mathbf{s}}_{\rm testd}(k) \| \tag{38}$$

(b) Variable contribution for SPE(k)

$$\mathbf{x}_{\rm cspe}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k) \tag{39}$$

Generally, the aberrant variables will have the largest residuals. The residual at sample k, SPE(k), is defined as the sum of the squares of $\mathbf{e}(k)$. Thus, the vector $\mathbf{e}(k)$ contains information on the individual prediction errors of each process variable at sample k. By plotting $\mathbf{e}(k)$ as a bar graph, the contributions to SPE(k) can be viewed. The relative sizes of the bars indicate the contributions of the variables to the prediction error, or the lack of fit of a sample to the model.

4. Case study

The proposed monitoring method was applied to fault detection and diagnosis in a benchmark simulation of fed-batch penicillin production.

4.1. Fed-batch penicillin fermentation

The production of secondary metabolites such as penicillin has been much studied in both academia and industry. In industry, commercial quantities of secondary metabolites such as antibiotics are produced using filamentous microorganisms. It is well established that the formation of the target product (e.g., an antibiotic) is usually not associated with cell growth. Hence, it is common practice to first grow the microorganisms in a batch culture and then to promote the synthesis of the antibiotic by means of a fed-batch operation with glucose. The penicillin process has a nonlinear dynamics and a multistage characteristics. In typical penicillin fermentation, most of the necessary cell mass is generated during the initial preculture stage. The penicillin starts to be produced at the exponential growth phase and continues to be produced until cell growth reaches the stationary phase. Cell growth must continue at a certain minimum rate to maintain high penicillin productivity. It is for this reason that glucose is fed continuously into the

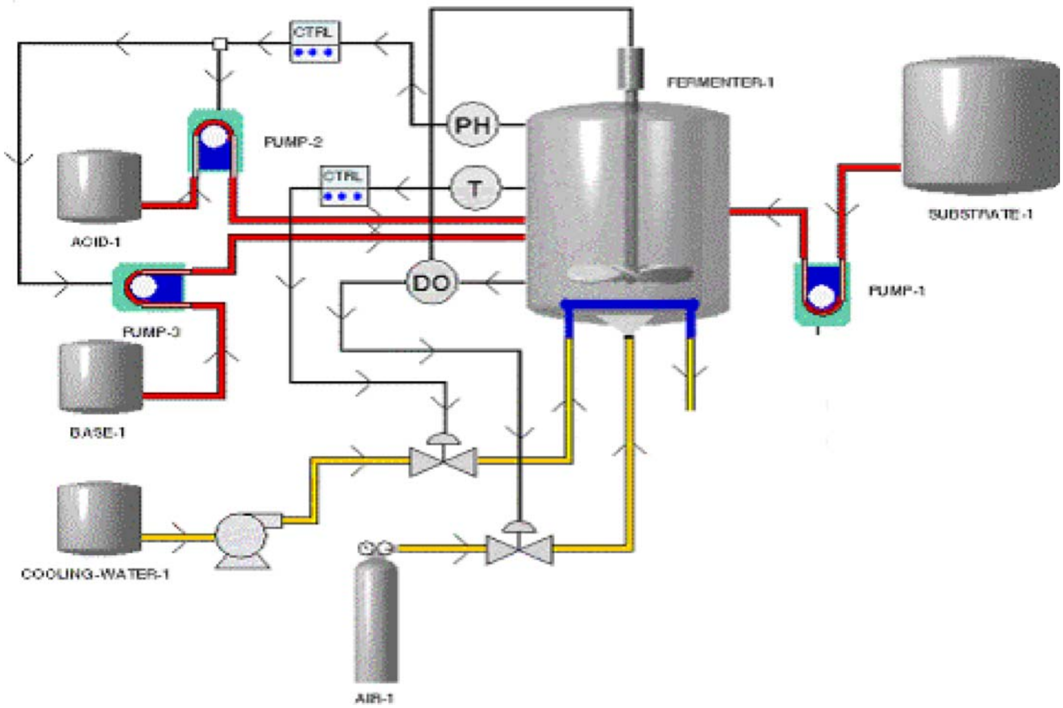

Fig. 2. Flow diagram of the penicillin fermentation process [28].

system during fermentation instead of being added all at once at the beginning [28].

The Monitoring and Control Group of the Illinois Institute of Technology has developed a simulator (PenSim v2.0) that is capable of simulating the concentrations of biomass, CO_2, hydrogen ion, penicillin, carbon source, and oxygen, as well as heat generation, during the production of penicillin under various operating conditions (http://www.chee.iit.edu/~cinar). These simulations are run under closed-loop control of pH and temperature while glucose addition is performed open-loop. A flow diagram of the penicillin fermentation process is illustrated in Fig. 2. In this process, important variables such as biomass and penicillin concentration are analyzed off-line by means of quality analysis experiments, resulting in a lag in the batch analysis. This means that primary quality variables are sampled after the batch is completed. On-line process monitoring provides a way to detect deviations in product quality during the process, making it possible to correct problems before they negatively impact the quality of the batch product.

5. Results and discussion

5.1. Off-line analysis of MPCA and MICA

In the present study, a total of 60 batches were simulated to create the normal reference batches. Variations typical of penicillin fermentation were introduced into these simulations, the results of which were used to develop the MPCA and MICA models. The 10 variables of the penicillin process taken into consideration in the present work are listed in Table 1. The duration of each batch was 400 h: about 45 h for the preculture stage and about 355 h for the fed-batch stage. The sampling interval was 1 h. Small variations were added to the simulation input data to mimic process variations in the normal operating conditions. In addition, measurement noise was added to each of the 10 monitored variables. Fig. 3 shows typical batch profiles of the 9 variables during penicillin fermentation.

A total of 60 batches thought to be normal were generated from the simulator. The reference data set was arranged as a three-way $\underline{\mathbf{X}}(I \times J \times K)$, where I corresponds to 60 batches, J corresponds to 10 process variables, and K corresponds to 400 time points. The MPCA and MICA models for the NOC models were constructed with the reference batch data $\underline{\mathbf{X}}$. In order to test if these batches are

Table 1
Variables used in the monitoring of the penicillin simulation benchmark

No.	Variables
1	Aeration rate (l/h)
2	Agitator power (W)
3	Glucose feed temperature (K)
4	Dissolved oxygen concentration (% saturation)
5	Culture volume (l)
6	Carbon dioxide concentration (mmol/l)
7	pH
8	Temperature (K)
9	Generated heat (kcal)
10	Cooling water flow rate (l/h)

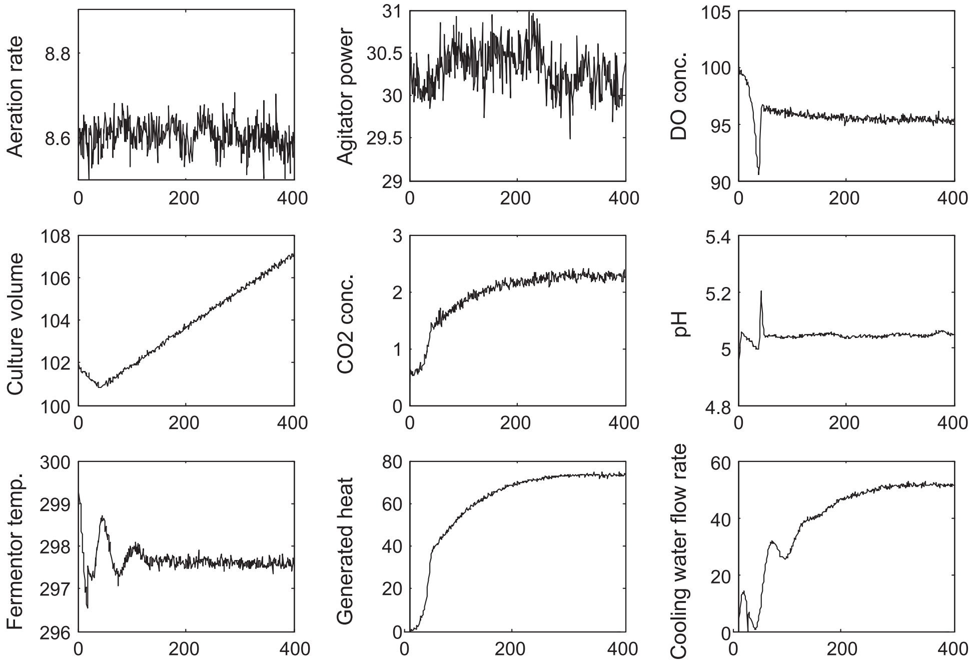

Fig. 3. Typical batch profiles of 9 variables during penicillin fermentation.

statistically normal, off-line analysis is performed. Fig. 4 illustrates the results of off-line analysis based on MPCA and MICA for the 60 reference batches, where the dotted lines correspond to 95% and 99% confidence limits which define the normal operation region. Four principal components of the MPCA model were determined by the cross-validation method [1], which explained about 59.3% of the total variability in the data. The remaining 41.7% of the variability was derived mainly from the measurement noise and random variation inherent in the normal batch operation. To order the ICA components, we used a Euclidean norm criterion to sort the rows of the demixing matrix. To ensure comparisons of equivalent models, four ICs were selected for the MICA model. Fig. 4 show the T^2 statistic and the SPE values of all the batches in the MPCA and MICA models, respectively. As shown in Fig. 4, some batches are above the 95% confidence limit at each model and the others are below 99% confidence limit. They can be used in the development of the reference model for on-line monitoring since all 60 batches lie within the specification limits, here 99% confidence limit. This difference between the MPCA and MICA mainly originates from the extracted feature components; both methods extract hidden information from a multidimensional data set, but PCA looks for Gaussian components, whereas ICA searches for non-Gaussian components.

Fig. 5 shows that normal probability plot and density estimates of the first score (t_1) do not conform to a Gaussian distribution but rather to a '*supergaussian distribution*' in which random variables take relatively more often values that are very close zero or very large. When MPCA is applied to this non-Gaussian data, the score values more

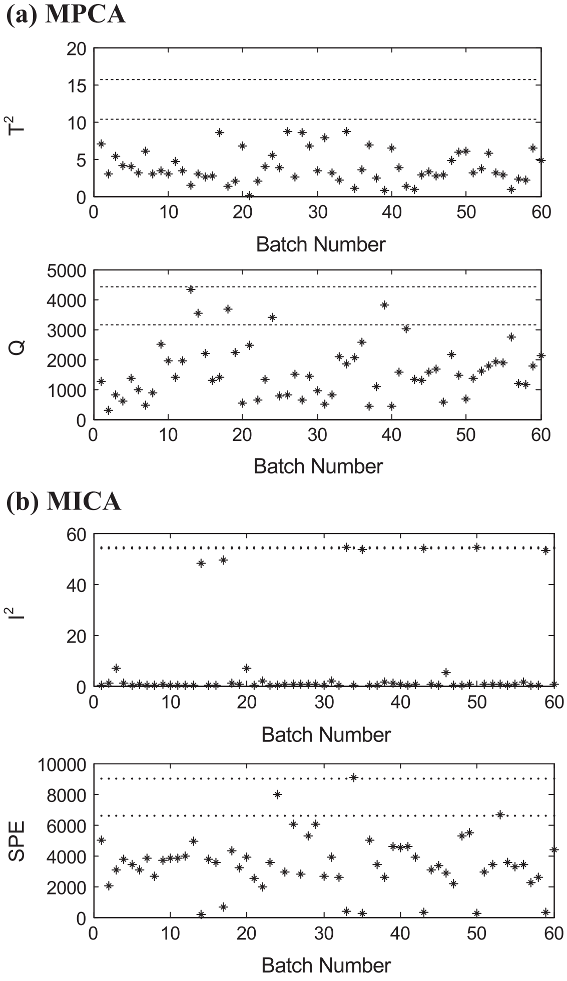

Fig. 4. Off-line analysis of (a) MPCA and (b) MICA for 60 reference batches.

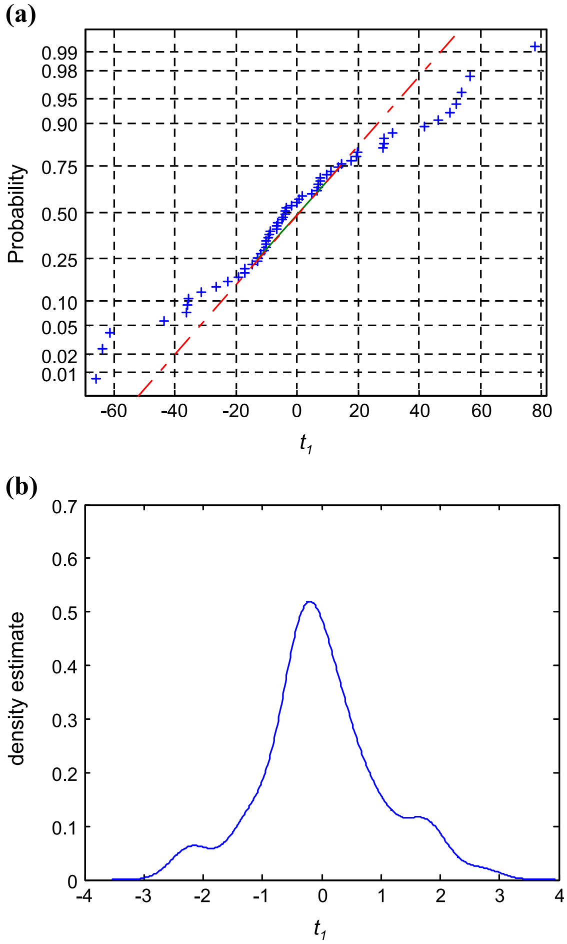

Fig. 5. Normal density plot and density estimate of the first score (t_1) obtained from MPCA.

frequently take on values that are very close to zero or very large. Thus, the monitoring charts of MPCA that are based on the assumption that the data are Gaussian distributed may deteriorate the reliability of the multivariate monitoring system.

5.2. On-line batch monitoring using MPCA and MICA

The MPCA and MICA models for on-line monitoring were developed from 60 reference batches. Three cases of the penicillin fermentation process in Table 2 were tested with MPCA and MICA: (1) normal operation, (2) a step decrease in the glucose feed rate, and (3) a linearly decrease in the glucose feed rate. Because the glucose feed rate is omitted from the monitored variables, the two faults relating the glucose feed rate become internal disturbances. If there was an equipment failure such as a leaky pump, the controller would still think it is pumping at the pre-programmed rate, thus the change in the glucose feed rate would not be directly observed, rather it would be indirectly observed through changes in other "measured" variables. We used the filling method 2 in which all future measurements are filled-in with their current deviation from the average batch because the current deviation most accurately matches the actual process measurement in the fed-batch process when the biomass was in its pre-culture [10]. Then the models of MPCA and MICA were tested against a new batch with a 99% control limit. The on-line monitoring charts were used to monitor the normal and abnormal batches at every time point k.

Table 2
Three simulation scenarios in penicillin fermentation

Test batch	Scenarios	Fault patterns
1	Normal batch	Normal batch
2	Step fault	Glucose feed rate is suddenly step-decreased by 15% at 50 h and maintained at that lower rate to the end of batch operation (400 h)
3	Ramp fault	Glucose feed rate is linearly decreased from 0.04 to 0.03 l/h from 60 h to the end of batch operation (400 h)

Fig. 6 show the monitoring results of the MPCA and MICA methods for the normal batch, respectively. In

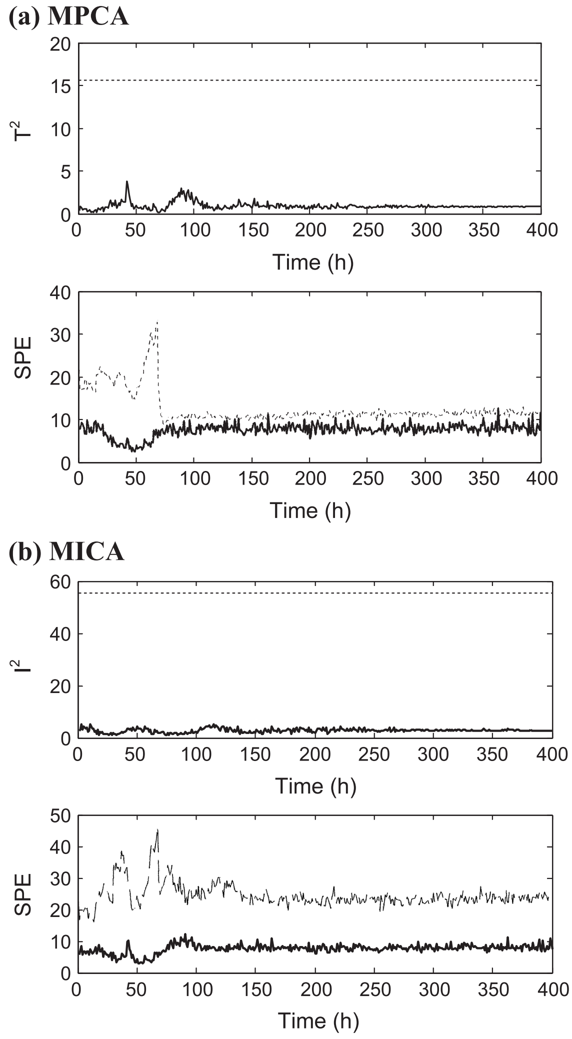

Fig. 6. On-line monitoring charts for (a) MPCA and (b) MICA in case of normal batch.

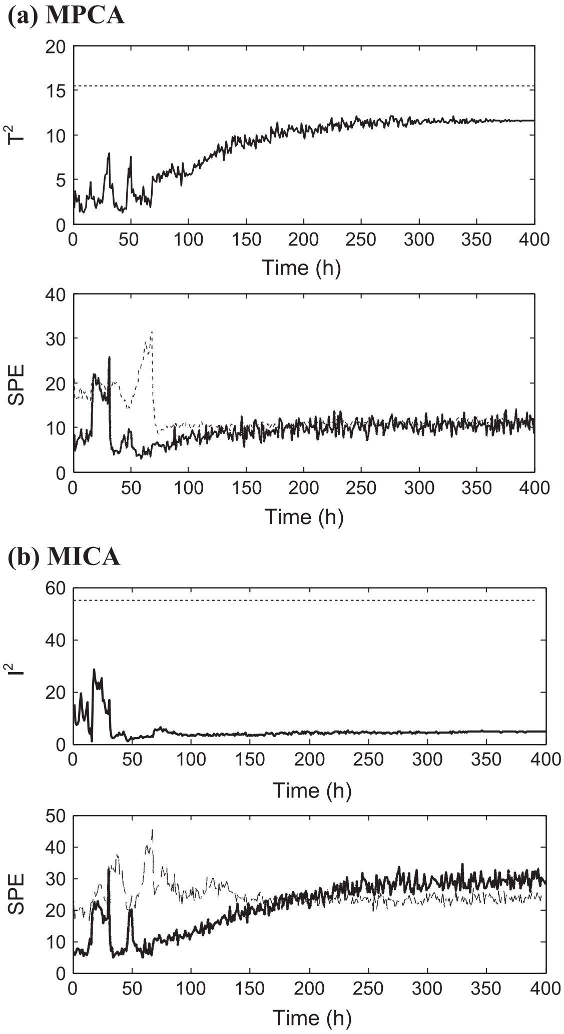

Fig. 7. On-line monitoring charts for (a) MPCA and (b) MICA in case of the first fault batch with a step decrease (99% confidence limits).

MICA model, this batch stays below the confidence limits for each case indicating that this batch behaves normally throughout the batch run. The SPE values of MICA are highest during the preculture stage of the batch (< 70). Furthermore, the SPE chart shows a maximum at the changeover between the preculture and fed-batch stages. In all batch runs, a batch culture has been followed by a fed-batch operation after the depletion of carbon source. The system switches itself to the fed-batch mode of operation when the level of glucose concentration reaches the threshold value chosen as 0.3 g/l [28]. In the simulations, the initial conditions of startup have different values for each batch. Different initial conditions cause large variations in the process variables at the start of the fed-batch phase. Then the process is using a setpoint trajectory until the end of the fed-batch. Therefore, variability is large in the early stage of batches, whereas variability in the late stage of batches is low. For this reason, the SPE control limits of the MPCA and MICA models have wide confidence intervals at the beginning of a batch and narrow confidence intervals at the end of a batch. This means that these data-based modeling methods can reflect the biological phenomena and relationships that occur during the penicillin fermentation. On the other hand, we observed a small deviation in the SPE chart of MPCA at 160, 170 and 360 h. Moreover, many samples lie in the vicinity of the 99% control limit after the preculture stage, making it difficult for the operator to decide the current status of the batch. Obviously, this deteriorates the reliability of the monitoring system. For this kind of multi-phase process, the multiple models approach in MPCA, which separates the total data into two or more stages, may give superior monitoring results and insights [23]. However, as shown above, single MICA model can adequately describe the two-phase characteristics of penicillin fermentation.

We now consider the process in which the glucose feed rate is suddenly decreased by 15% at 50 h and then maintained at that lower rate until the end of fermentation. Glucose is the main carbon source of the fed-batch fermen-

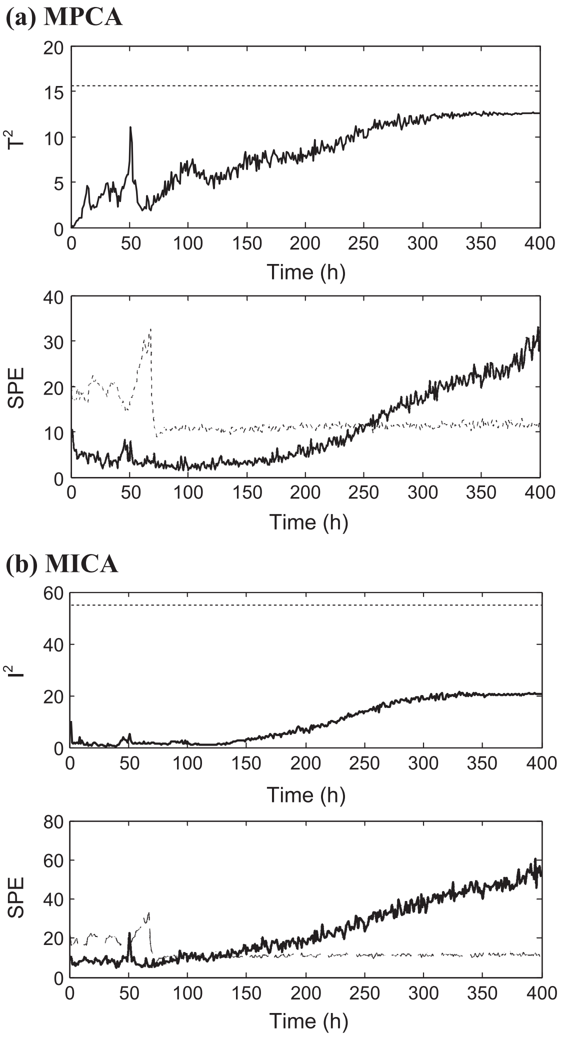

Fig. 8. On-line monitoring charts for (a) MPCA and (b) MICA in case of the second fault batch with a ramp decrease (99% confidence limits).

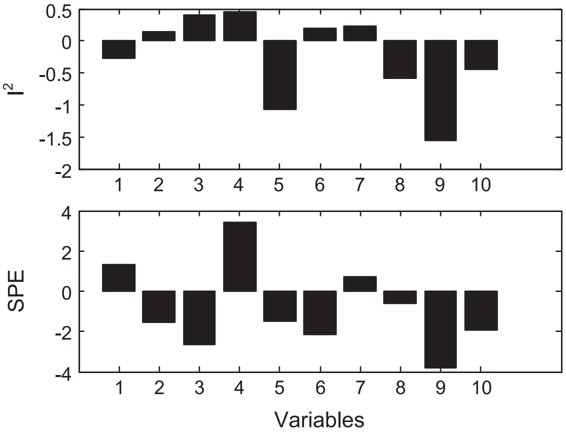

Fig. 9. Contribution plots of the MICA model for the second abnormal batch with a ramp decrease at 380 h.

tation, and thus a decrease in the glucose feed rate will lead to a reduction in penicillin production. The monitoring results of the MPCA and MICA models for this abnormal process are shown in Fig. 7. This type of fault of the step decrease in the glucose feed rate is difficult to detect in the T^2 and SPE charts of MPCA. The SPE chart of MPCA deviates slightly at 90 h and then increases slightly and continuously until the end of fermentation. In contrast, the SPE chart of MICA successfully detects this type of fault; this chart shows a sharp rise at about 70 h and then increases continuously to the end of fermentation. Compared with MPCA, MICA shows relatively correct and more rapid fault detection ability. The SPE value of MICA exceeds its control limit for the first time at around 180 h, which was delayed 130 h after the event had occurred. Furthermore, the I^2 values of MICA suddenly increased at 70 h, which corresponds to the characteristics of the fault type.

We now consider the fault in which a ramp decrease is imposed on the batch, specifically, the glucose feed rate is linearly decreased from 0.04 to 0.03 l/h from 60 h to the end of batch operation (400 h) due to the leaky pump. A decrease in its feed results in reduction in penicillin production since glucose is the main carbon source to be fed during the fed-batch fermentation [28]. As shown in Fig. 8, the SPE chart of MPCA definitely detects the disturbance at 260 h, which was delayed 200 h after the commencement of the disturbance. From this figure, we know that MPCA monitoring can detect this small external disturbance with a significant delay. It shows that MICA detects the ramp type fault earlier than MPCA. The SPE chart of MICA detects the fault at 100 h, which is 160 h earlier than the detection time of MPCA. Both T^2 values of MPCA and the I^2 values of MICA increase linearly from 70 h to the end of batch operation (400 h), which corresponds to the characteristics of the fault.

Because these monitoring charts only detect non-conforming batches, contribution plots can be used to diagnose the event so as to assign a cause. The contribution plots indicate which variables are predominantly responsible for the deviations from the normal batch behavior. Fig. 9 shows the contribution plots for the I^2 and SPE charts of MICA at 380 h. The deviation of SPE come from the dissolved oxygen concentration (variable 4), the carbon dioxide concentration (variable 6) and the generated heat (variable 9), respectively. Thus, it can be inferred that the deviation of the glucose feed rate (internal disturbance) affects other variables, especially the dissolved oxygen concentration and the generated heat. The decrease of the glucose feed rate reduces the microorganism reaction rate, leading to dissolved oxygen concentration and decrease of the generated heat.

6. Conclusions

A new approach to monitoring the progress of batch processes has been described. The key idea of the proposed method is to exploit the ability of MICA to extract statistical independent features from three-way batch data with a non-Gaussian distribution by projecting the data onto a low-dimensional space that summarizes both the variables and their time trajectories. Given that ICA is more meaningful property than PCA for extracting features from data sets containing non-Gaussian components, the use of ICA can improve the monitoring performance. The simulation study of a fed-batch fermentation showed the power and advantages of MICA.

Acknowledgements

This work was financially supported by the Post-doctoral Fellowship Program of the Korea Science and Engineering Foundation (KOSEF) and a Visiting Postdoctoral Fellowship of the Fund for Scientific Research-Flanders (FWO).

References

[1] P. Nomikos, J.F. MacGregor, Monitoring batch processes using multiway principal component analysis, AIChE J. 40 (8) (1994) 1361–1375.

[2] P. Nomikos, J.F. MacGregor, Multivariate SPC charts for monitoring batch processes, Technometrics 37 (1995) 41–59.

[3] P. Nomikos, J.F. MacGregor, Multi-way partial least square in monitoring batch processes, Chem. Intell. Lab. Sys. 30 (1995) 97–108.

[4] N.B. Gallagher, B.M. Wise, Application of multi-way principal component analysis to nuclear waste storage tank monitoring, Comput. Chem. Eng. 20 (1996) S739–S744.

[5] P. Nomikos, Detection and diagnosis of abnormal batch operations based on multi-way principal component analysis, ISA Trans. 35 (1996) 259–266.

[6] K.A. Kosanovich, K.S. Dahl, M.M. Piovoso, Improved process understanding using multiway principal component analysis, Ind. Eng. Chem. Res. 35 (1996) 138–146.

[7] D.J. Tates, A.A. Louwerse, A.K. Smilde, G.L.M. Koot, H. Berndt, Monitoring a PVC batch process with multivariate statistical process control charts, Ind. Eng. Chem. Res. 38 (1998) 4769–4776.

[8] J.A. Westerhuis, T. Kourti, J.F. MacGregor, Comparing alternative approaches for multivariate statistical analysis of batch process data, J. Chemometrics 13 (1999) 397–413.

[9] L. Gregersen, S.B. Jorgensen, Supervision of fed-batch fermentations, Chem. Eng. J. 75 (1999) 69–76.

[10] S. Albert, R.D. Kinley, Multivariate statistical monitoring of batch processes: an industrial case study of fermentation supervision, Trends Biotechnol. 19 (2) (2001) 53–62.

[11] B. Lennox, G.A. Montague, H.G. Hiden, G. Kornfeld, P.R. Goulding, Process monitoring of an industrial fed-batch fermentation, Biotechnol. Bioeng. 74 (2) (2001) 125–135.

[12] Q. Chen, R.J. Wynne, P. Goulding, D. Sandoz, The application of principal component analysis and kernel density estimation to enhance process monitoring, Cont. Eng. Prac. 8 (2000) 531–543.

[13] T. Kourti, Multivariate dynamic data modeling for analysis and statistical process control of batch processes, start-ups and grade transitions, J. Chemom. 17 (2003) 93–109.

[14] D. Dong, T.J. McAvoy, Batch tracking via nonlinear principal component analysis, AIChE J. 42 (8) (1996) 2199–2208.

[15] S. Rãnnar, J.F. MacGregor, S. Wold, Adaptive batch monitoring using hierarchical PCA, Chem. Intell. Lab. Sys. 41 (1998) 73–81.

[16] A. Hyvärinen, Fast and robust fixed-point algorithms for independent component analysis, IEEE Trans. Neural Netw. 10 (1999) 626–634.

[17] A. Hyvärinen, Survey on independent component analysis, Neural Comput. Surv. 2 (1999) 94–128.

[18] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis, Wiley, USA, 2001.

[19] J.-M. Lee, C.K. Yoo, I.B. Lee, Statistical process monitoring with independent component analysis, J. Process Control 14 (2004) 467–485.

[20] M. Kano, S. Tanaka, H. Ohno, S. Hasebe, I. Hashimo, Monitoring independent components for fault detection, AIChE J. 49 (2003) 969–976.

[21] S. Wold, N. Kettanhe, H. Friden, A. Holmberg, Modelling and diagnostics of batch processes and analogous kinetic experiments, Chem. Intell. Lab. Sys. 44 (1998) 331–340.

[22] X. Meng, J. Morris, B. Martin, On-line monitoring of batch processes using PARAFAC representation, J. Chemom. 17 (2003) 65–81.

[23] C. Ündey, A. Çinar, Statistical monitoring of multistage, multiphase batch processes, IEEE Control Syst. Mag. 10 (2002) 40–52.

[24] J.-F. Cardoso, A. Soulomica, Blind beamforming for non-Gaussian Signals, IEE Proc. F. 140 (6) (1993) 362–370.

[25] J.E. Jackson, A User's Guide to Principal Components, Wiley, USA, 1991.

[26] E.B. Martin, A.J. Morris, Non-parametric confidence bounds for process performance monitoring charts, J. Process Control 6 (6) (1996) 349–358.

[27] B.W. Silverman, Density Estimation for Statistics and Data Analysis, Chapman & Hall, UK, 1986.

[28] B. Birol, C. Undey, A. Cinar, A modular simulation package for fed-batch fermentation: penicillin production, Comput. Chem. Eng. 26 (2002) 1553–1565.