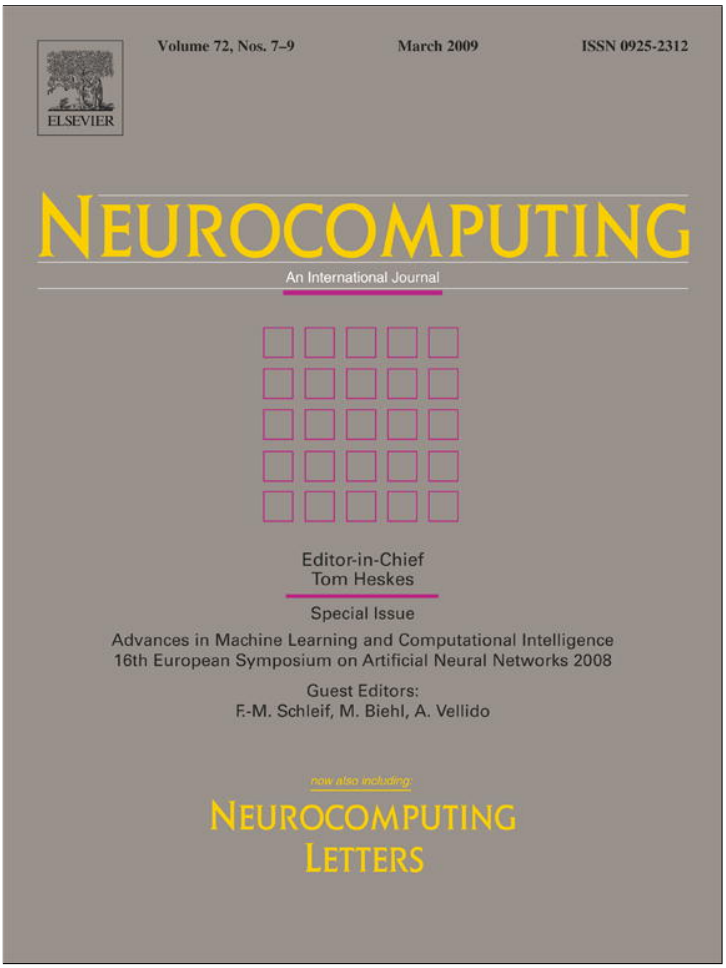

Neurocomputing 72 (2009) 1584–1596

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Multiway kernel independent component analysis based on feature samples for batch process monitoring ☆

Xuemin Tian [a], Xiaoling Zhang [a], Xiaogang Deng [a], Sheng Chen [b,*]

[a] College of Information and Control Engineering, China University of Petroleum (Hua Dong), Donying, Shandong 257061, China
[b] School of Electronics and Computer Science, University of Southampton, Highfield, Southampton SO17 1BJ, UK

ARTICLE INFO

Article history:
Received 7 May 2008
Received in revised form
21 August 2008
Accepted 18 September 2008
Communicated by T. Heskes
Available online 9 October 2008

Keywords:
Batch process
Nonlinearity
Kernel independent component analysis
Feature samples

ABSTRACT

Most batch processes generally exhibit the characteristics of nonlinear variation. In this paper, a nonlinear monitoring technique is proposed using a multiway kernel independent component analysis based on feature samples (FS-MKICA). This approach first unfolds the three-way dataset of a batch process into the two-way one and then chooses representative feature samples from the large two-way input training dataset. The nonlinear feature space abstracted from the unfolded two-way data space is then transformed into a high-dimensional linear space via kernel function and an independent component analysis (ICA) model is established in the mapped linear space. The proposed FS-MKICA method can significantly reduce the computational cost in extracting the kernel ICA model since it is based on the small subset of feature samples rather than on the entire input sample set. We supply two statistics, the I^2 statistic of process variation and the squared prediction error statistic of residual, for on-line monitoring of batch processes. The proposed method is applied to detecting faults in the fed-batch penicillin fermentation process. The standard linear ICA method for batch process monitoring, known as the multiway independent component analysis (MICA), is also applied to the same benchmark batch process. The simulation results obtained in this nonlinear batch process application clearly demonstrate the power and superiority of the new nonlinear monitoring method over the linear one. The FS-MKICA approach can extract the nonlinear features of the batch process while the MICA method cannot.

1. Introduction

Nowadays more and more products tend to be manufactured via batch or semi-batch operations. This is particularly the case for large range of consumer products, such as foods, cosmetics, chemical medicines, dyestuffs, dopes and so on, which are closely linked to people's daily life. Batch processes have advantages in lower investment costs and ventures, greater flexibility in manufacturing as well as easier to produce large varieties of products, compared to continuous ones. Most importantly, batch operation patterns allow corporations and companies to quickly make new products to adapt to drastic market competition. To ensure safety and stability of batch processes as well as high quality of final products, on-line monitoring and fault diagnosis in batch processes emerge as an essential and important task.

Advances in measurement technology and distributed control systems (DCSs) have facilitated collection of large amounts of data in today's process industry. Data-driven multivariate statistical methods, which are based on the theory of statistical process control (SPC), have attracted much attention from process engineers [6]. Principal component analysis (PCA) [22] is one of the classic data-driven multivariate statistical techniques which can handle high-dimensional and correlated process variables. Nomikos and MacGregor [19] developed a multiway PCA (MPCA) approach based on PCA for monitoring batch processes. However, MPCA is a static and linear projection method, which cannot effectively capture the nonlinear features existing in most batch processes. Dong and McAvoy [7] presented a nonlinear PCA method based on principal curves and neural networks to monitor batch process performance. In the work [15], a new nonlinear batch monitoring technique, referred to as the multiway kernel PCA (MKPCA), was proposed. This MKPCA first unfolds the three-way data of a normal batch process and then maps the unfolded data space into a high-dimensional feature space via a kernel function. Finally, principal components are extracted in the high-dimensional feature space.

☆ This work is supported by the National 863 Program of China sponsored Project (No. 2004AA412050) and the Natural Science Foundation of Shandong province, China (Grant no. Y2007G49).

* Corresponding author.
E-mail addresses: tianxm@hdpu.edu.cn (X. Tian), sqc@ecs.soton.ac.uk (S. Chen).

0925-2312/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2008.09.003

More recently, a multivariate statistical method, known as independent component analysis (ICA), has emerged to be a powerful tool in process monitoring. ICA is originally derived for blind source separation and has found wide-ranging applications in many areas including signal processing, feature extraction, telecommunications, audio signal separation, etc. [12,11]. In ICA, components capture high-order information and are non-Gaussian and dependent while components generated from PCA are merely de-correlated. Because of its specified merits compared to PCA, many researchers have implemented ICA for monitoring process behaviour. Kano and co-workers [13] directly monitored independent components obtained from the ICA algorithm for fault detection in a continuous process. Lee and co-workers [16] applied the ICA approach to statistical process monitoring, in which I^2, I_e^2 and the squared prediction error (SPE) charts are proposed as on-line monitoring charts, similar to the T^2 and SPE charts used for PCA, and they also considered the contribution plots of these statistics for fault identification. The authors of [17] developed a dynamic ICA algorithm for process monitoring to take into consideration the correlations that exist between variables. The ICA method was extended to the multiway ICA (MICA) for on-line monitoring of batch processes in [26,25], and in these two works the MICA is verified to be more efficient than the PCA because of the non-Gaussian distribution assumption of MICA. Albazzaza and Wang [1] proposed another method by introducing time lag shifts to include batch process dynamics in the ICA model.

ICA, however, is a statistical technique to separate linearly mixed latent sources. Thus, independent components obtained from ICA cannot explain the essential sources for nonlinear mixtures. The authors of [24] presented a two-phase nonlinear ICA algorithm for face recognition. This kernel ICA (KICA) approach developed in [24] is essentially kernel PCA (KPCA) plus ICA. Training data are whitened and mapped into a feature space by KPCA as linearly separable as possible. ICA is then performed to seek the projection directions in the KPCA-whitened space. In this study, we extend this KICA algorithm from face recognition to batch process monitoring. Direct adopting the KICA transformation approach requires that we first unfold the three-way batch data to a two-way one according to the method described in [23] before performing KICA and then compute a kernel matrix whose dimension is equivalent to the number of training samples. For an unfolded data matrix of batch process, however, the number of samples, n, is typically very large, and performing ICA on the kernel matrix for such a large dataset is costly and may involve prohibitive memory requirements. Hence, the efficiency of extracting feature components in such a high-dimensional feature space will be extremely poor and on-line monitoring performance will be impeded seriously. In fact, extracting the m dominant kernel principle components of the $n \times n$ kernel matrix has a computational complexity of $\mathcal{O}(m \cdot n^2)$ operations and memory requirements of $\mathcal{O}(n^2)$ [8].

In order to solve the computation problem associated with a huge kernel matrix, we introduce a novel feature sample extracting technique before implementing kernel transformation. Small subset of the d feature samples, which can describe the behaviours of the original data space, is selected from all the n training samples, where $d \ll n$ and $d \geqslant m$. Because the size of the kernel matrix is reduced significantly, computation of the ICA on the feature space becomes much simpler. This is simply because extracting the m dominant kernel principle components of the resulting $d \times d$ kernel matrix only has a computational complexity of $\mathcal{O}(m \cdot d^2)$ and memory requirements of $\mathcal{O}(d^2)$. This makes it possible to implement on-line batch processes monitoring efficiently. Both the I^2 and SPE charts are implemented for on-line monitoring purpose. The novel nonlinear monitoring method developed in this contribution is, therefore referred to as the multiway kernel independent component analysis based on feature samples (FS-MKICA). The remainder of this paper is organised as follows. ICA and KICA are briefly introduced in Sections 2 and 3, respectively. In Section 4, we describe our new FS-MKICA method and present an on-line batch monitoring strategy using the FS-MKICA. The proposed method is validated in Section 5 using a case study of the benchmark fed-batch penicillin fermentation, and its performance is evaluated in comparison with that of the standard MICA. Our conclusions are offered in Section 6.

2. Independent component analysis

The concept of ICA is illustrated in Fig. 1. Let us assume that the measured variables $\mathbf{x} = [x_1\ x_2\ \cdots\ x_m]^{\mathrm{T}} \in \mathbb{R}^m$ are given as the linear combinations of the l ($\leqslant m$) unknown independent components $\mathbf{s} = [s_1\ s_2\ \cdots\ s_l]^{\mathrm{T}} \in \mathbb{R}^l$. The relationship between $\mathbf{x}$ and $\mathbf{s}$ is given by

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

where $\mathbf{A} \in \mathbb{R}^{m \times l}$ is the mixing matrix. The aim of ICA is to estimate the latent components by finding a de-mixing matrix $\mathbf{W} \in \mathbb{R}^{l \times m}$ from the measured data. In addition, the components extracted must be as independent of each other as possible. Thus, the estimation of $\mathbf{s}$ can be expressed by

$$\hat{\mathbf{s}} = \mathbf{W} \cdot \mathbf{x}. \tag{2}$$

Note that the observation data need to be whitened before applying the ICA algorithm. The whitening transformation can be accomplished by means of PCA. The whitened data $\mathbf{y} = \mathbf{U}\mathbf{x}$ satisfies

$$E\{\mathbf{y}\mathbf{y}^{\mathrm{T}}\} = \mathbf{I}_m, \tag{3}$$

where $E\{\cdot\}$ denotes the expectation operator and $\mathbf{I}_m$ the $m \times m$ identity matrix. The whitening matrix $\mathbf{U}$ can be chosen explicitly as follows. Let $E\{\mathbf{x}\mathbf{x}^{\mathrm{T}}\} = \mathbf{E}\mathbf{D}\mathbf{E}^{\mathrm{T}}$, where $\mathbf{E}$ is the orthogonal matrix of eigenvectors of $E\{\mathbf{x}\mathbf{x}^{\mathrm{T}}\}$ and $\mathbf{D}$ the diagonal matrix of its eigenvalues. Then $\mathbf{U} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^{\mathrm{T}}$. In the sequel, the whitened data are still denoted by $\mathbf{x}$ for notational simplification.

There are several ways of obtaining the de-mixing matrix, and these include measures of non-Gaussianity such as kurtosis and negentropy, minimisation of mutual information and maximum likelihood estimation [12]. The FastICA [12] is a fixed-point iteration scheme based on negentropy, which is most widely used in ICA. The optimisation objective of the FastICA is defined by

$$\max_{\mathbf{w}} J(\mathbf{w}) = \max_{\mathbf{w}} (E\{G(\mathbf{w}^{\mathrm{T}}\mathbf{x})\} - E\{G(\upsilon)\})^2$$
$$\text{s.t. } E\{(\mathbf{w}^{\mathrm{T}}\mathbf{x})^2\} = 1, \tag{4}$$

where $\mathbf{w}^{\mathrm{T}}$ is a row vector of the matrix $\mathbf{W}$, υ is a Gaussian variable with zero mean and unit variance, and $G(\cdot)$ is a non-quadratic function. Several choices of the G-function were suggested in [12], and the following one

$$G(u) = \frac{1}{a}\log\cosh(au), \tag{5}$$

where $1 \leqslant a \leqslant 2$, is known to be a good general-purpose contrast function and is therefore used in this paper. The detailed algorithm of the FastICA can be found in the literature [12,11], and therefore it is not repeated here.

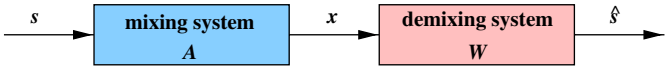

Fig. 1. Illustration of ICA.

ICA can alternatively be performed using the joint approximative diagonalisation of eigenmatrices (JADE) algorithm [5]. The JADE adopts fourth-order cumulant as the measure of non-Gaussianity, while the FastICA uses negentropy as the measure of non-Gaussianity. Negentropy is a more effective measure of non-Gaussianity. Also the FastICA has a fast convergence rate and is computationally much less complex than the JADE.

3. Kernel independent component analysis

It is known that most complex batch processes exhibit nonlinear characteristics and the linear ICA algorithm performs poorly in these cases. This section summarises the KICA algorithm [24]. Given a sequence of the observed data vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, where $\mathbf{x}_i \in \mathbb{R}^m$, the nonlinear mapping that maps the data onto the high-dimensional feature space $\mathscr{H}$ is defined by $\Phi : \mathbb{R}^m \to \mathscr{H}$ with $\mathbf{x} \mapsto \Phi(\mathbf{x})$. Assuming that the data in the mapped feature space has a linear separable structure, the ICA can readily be applied. As mentioned in the previous section, data should be whitened before applying the ICA algorithm. Similarly, the KICA algorithm requires whitening preprocessing, which can be achieved using PCA preprocessing in the feature space $\mathscr{H}$ [24].

3.1. KPCA preprocessing

Assuming that the observation vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ are nonlinearly mapped into the feature space $\mathscr{H}$, the covariance matrix of the mapped data can be expressed by

$$\mathbf{C}^{\mathscr{H}} = \frac{1}{n}\sum_{i=1}^{n} \Phi(\mathbf{x}_i)\Phi(\mathbf{x}_i)^{\mathrm{T}}. \tag{6}$$

For the time being assume that the mapped data are centred and satisfy $\sum_{i=1}^{n} \Phi(\mathbf{x}_i) = \mathbf{0}$. Define $\mathbf{Q} = [\Phi(\mathbf{x}_1)\ \Phi(\mathbf{x}_2)\ \cdots\ \Phi(\mathbf{x}_n)]$. Then an alternative expression of (6) is $\mathbf{C}^{\mathscr{H}} = (1/n)\mathbf{Q}\mathbf{Q}^{\mathrm{T}}$. Define the Gram matrix $\mathbf{K} = \mathbf{Q}^{\mathrm{T}}\mathbf{Q} \in \mathbb{R}^{n\times n}$, whose elements are defined by the given kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ as follows:

$$k_{i,j} = \Phi(\mathbf{x}_i)^{\mathrm{T}}\Phi(\mathbf{x}_j) = \langle \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \rangle = k(\mathbf{x}_i, \mathbf{x}_j). \tag{7}$$

Applying the eigendecomposition to the Gram matrix leads to $\mathbf{K} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}$, where $\mathbf{V} \in \mathbb{R}^{n\times n}$ is the matrix containing the orthonormal eigenvectors of $\mathbf{K}$ and $\mathbf{\Lambda} = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ is the diagonal matrix consisting of the eigenvalues of $\mathbf{K}$. Note that these eigenvalues satisfy $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n \geqslant 0$. Thus the expression of $\mathbf{C}^{\mathscr{H}}$ in the case of $\mathbf{K} = \mathbf{Q}^{\mathrm{T}}\mathbf{Q}$ is as follows:

$$\mathbf{C}^{\mathscr{H}} = (\mathbf{Q}\mathbf{V}\mathbf{\Lambda}^{-1/2})\frac{\mathbf{\Lambda}}{n}(\mathbf{Q}\mathbf{V}\mathbf{\Lambda}^{-1/2})^{-1}. \tag{8}$$

Evidently, the eigenvalues of $\mathbf{C}^{\mathscr{H}}$ are $\lambda_1/n, \lambda_2/n, \ldots, \lambda_n/n$ which are also arranged in descending order, and the associated matrix of orthonormal eigenvectors can be expressed as $\mathbf{Q}\mathbf{V}\mathbf{\Lambda}^{-1/2}$. Therefore, the whitened data in the feature space $\mathscr{H}$ can be deduced as follows:

$$\mathbf{y} = (\sqrt{n}\mathbf{Q}\mathbf{V}\mathbf{\Lambda}^{-1})^{\mathrm{T}}\Phi(\mathbf{x}) = \sqrt{n}\mathbf{\Lambda}^{-1}\mathbf{V}^{\mathrm{T}}\mathbf{k}_x, \tag{9}$$

where $\mathbf{k}_x = [k(\mathbf{x}_1, \mathbf{x})\ k(\mathbf{x}_2, \mathbf{x})\ \cdots\ k(\mathbf{x}_n, \mathbf{x})]^{\mathrm{T}} \in \mathbb{R}^n$.

As mentioned previously, the mapped data in the feature space $\mathscr{H}$ should be centred before applying the PCA projection. But it is difficult to centre the data in $\mathscr{H}$ because the nonlinear map $\Phi : \mathbb{R}^m \to \mathscr{H}$ is unknown. However, a slight modification of notation circumvents this difficulty, and one can centre the matrix $\mathbf{K}$ and the vector $\mathbf{k}_x$, respectively, as follows [24]:

$$\tilde{\mathbf{K}} = \mathbf{K} - \mathbf{Z}_n\mathbf{K} - \mathbf{K}\mathbf{Z}_n + \mathbf{Z}_n\mathbf{K}\mathbf{Z}_n, \tag{10}$$

$$\tilde{\mathbf{k}}_x = \mathbf{k}_x - \mathbf{Z}_n\mathbf{k}_x - \mathbf{K}\mathbf{z}_1 + \mathbf{Z}_n\mathbf{K}\mathbf{z}_1, \tag{11}$$

where $\mathbf{Z}_n$ is the $n \times n$ matrix whose elements are all $1/n$ and $\mathbf{z}_1$ is the $n \times 1$ vector whose elements are all $1/n$. Replacing $\mathbf{K}$ by $\tilde{\mathbf{K}}$, we can calculate the eigenvalues and corresponding orthonormal eigenvectors of $\tilde{\mathbf{K}}$, still denoted as $\mathbf{\Lambda}$ and $\mathbf{V}$, respectively. Similarly, replacing $\mathbf{k}_x$ by $\tilde{\mathbf{k}}_x$, the whitening operation (9) becomes

$$\mathbf{y} = \sqrt{n}\mathbf{\Lambda}^{-1}\mathbf{V}^{\mathrm{T}}\tilde{\mathbf{k}}_x. \tag{12}$$

3.2. Extracting essential sources using ICA

The next task after sphering data in the feature space $\mathscr{H}$ is to implement the ICA process. According to the ICA algorithm described in Section 2, we estimate the de-mixing matrix $\mathbf{W}$ and separate the independent components $\hat{\mathbf{s}}$ from the whitened variables $\mathbf{y}$ as

$$\hat{\mathbf{s}} = \mathbf{W} \cdot \mathbf{y}. \tag{13}$$

From $E\{\hat{\mathbf{s}}\hat{\mathbf{s}}^{\mathrm{T}}\} = \mathbf{W}E\{\mathbf{y}\mathbf{y}^{\mathrm{T}}\}\mathbf{W}^{\mathrm{T}} = \mathbf{W}\mathbf{W}^{\mathrm{T}} = \mathbf{I}_l$, we conclude that $\mathbf{W}$ is an orthogonal matrix which can of course be obtained by the FastICA algorithm [12]. Therefore, the KICA algorithm of [24] can be summarised into the two steps: firstly, sphering the data in the feature space $\mathscr{H}$ using the KPCA as described in (12) and, secondly, performing the ICA algorithm in the KPCA whitened space according to (13).

Assume that the m most dominant independent components $\hat{\mathbf{s}}_m$ are extracted, and denote the corresponding de-mixing matrix as $\mathbf{W}_m$. Then

$$\hat{\mathbf{s}}_m = \mathbf{W}_m \cdot \mathbf{y}. \tag{14}$$

The choice of m is an important subject entirely in itself. We adopt the following procedure to determine the value of m. Given all the independent components $\hat{\mathbf{s}}$, calculate the negentropy of each component, and arrange the components in descending order according to their negentropy values. Set a non-Gaussianity threshold, and only keep the components whose negentropy values are larger than the threshold. The desired threshold value, however, can only be found empirically.

4. Multiway KICA based on feature samples

The collected batch process data are a three-way matrix $\underline{X}(I \times J \times K)$, where I is the number of batch runs, J denotes the number of process variables and K is the number of samples in a batch run. In order to apply the KICA to on-line monitoring of batch processes, we need to unfold this three-way matrix first to obtain a two-way matrix $X(I \times KJ)$ using batch-wise unfolding, as illustrated in Fig. 2, and to normalise $X(I \times KJ)$ by mean centring and variance scaling. Then $X(I \times KJ)$ is rearranged into the two-way matrix $X(KI \times J)$ based on variable-wise unfolding [23,14], as

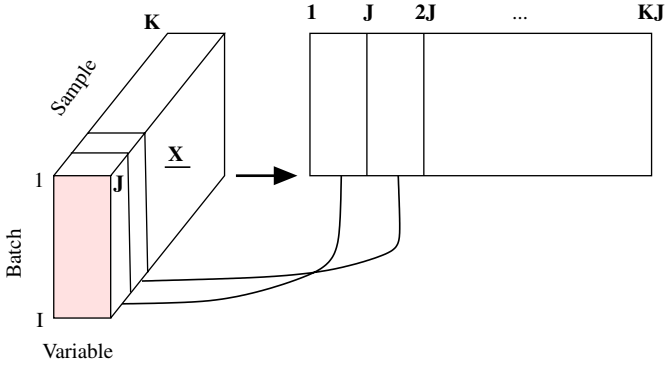

Fig. 2. Batch-wise unfolding.

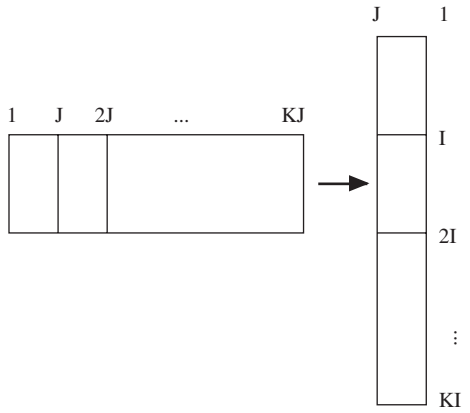

Fig. 3. Variable-wise unfolding.

illustrated in Fig. 3. The KICA is implemented on $X(KI \times J)$ and, consequently, a multiway KICA (MKICA) model is established for the batch process under normal operating conditions. This MKICA method for on-line monitoring of batch processes, obtained by directly applying the KICA algorithm of [24], however, may suffer from computational difficulty. This is because the number of samples, $n = KI$, for $X(KI \times J)$ is typically very large. Thus, when performing KICA in the data space $X(KI \times J)$, the computation of the $n \times n$ kernel matrix is extremely time consuming and moreover a very large computer memory space is required to store this kernel matrix. We present the novel MKICA method based on feature samples, referred to as the FS-MKICA, to overcome the shortcoming of the MKICA based on all the samples. Our proposed method selects a small subset of feature samples from $X(KI \times J)$ and compute the kernel matrix on this small subset, thus significantly reducing the computation cost and the memory space required.

4.1. Choosing feature samples

Let the subset of feature samples be $FS(d \times J) = \{\mathbf{fs}_1, \mathbf{fs}_2, \ldots, \mathbf{fs}_d\} \subset X(KI \times J)$. This subset should satisfy [9]

- $FS(d \times J)$ is the largest subset of $X(KI \times J)$ with $\text{rank}(\Phi_{FS}) = d$, and $\text{span}(\Phi_{FS}) \approx \text{span}(\Phi_X)$, where $\text{span}(\Phi_{FS})$ denotes the space spanned by the bases $\Phi(\mathbf{fs}_1), \ldots, \Phi(\mathbf{fs}_d)$, while $\text{span}(\Phi_X)$ denotes the feature space constructed from the entire dataset $X(KI \times J)$.

This implies that the kernel matrix in the mapped feature space $\mathscr{H}$, calculated based on $FS(d \times J)$, has the full rank d. Before describing how to choose $FS(d \times J)$, we give the following theorem.

Theorem 1. *Partition the $n \times n$ kernel matrix $\mathbf{K}_n$, calculated from the samples $\{\mathbf{x}_i\}_{i=1}^n$, into*

$$\mathbf{K}_n = \begin{bmatrix} \mathbf{K}_{n-1} & \mathbf{k}_{n-1,n} \\ \mathbf{k}_{n-1,n}^{\mathrm{T}} & k_{n,n} \end{bmatrix},$$

where $\mathbf{k}_{n-1,n}^{\mathrm{T}} = [k(\mathbf{x}_1, \mathbf{x}_n)\ k(\mathbf{x}_2, \mathbf{x}_n)\ \cdots\ k(\mathbf{x}_{n-1}, \mathbf{x}_n)]$ *and* $k_{n,n} = k(\mathbf{x}_n, \mathbf{x}_n)$. *Assume that* $\mathbf{K}_{n-1}$ *has the full rank. If* $\delta_n = k_{n,n} - \mathbf{k}_{n-1,n}^{\mathrm{T}} \mathbf{K}_{n-1}^{-1} \mathbf{k}_{n-1,n} = 0$, *then* $\mathbf{K}_n$ *has the reduced rank* $n-1$.

Proof. If the last column of $\mathbf{K}_n$ can be expressed as a linear combination of the first $n-1$ columns of $\mathbf{K}_n$, namely

$$\begin{bmatrix} \mathbf{k}_{n-1,n} \\ k_{n,n} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{n-1} \\ \mathbf{k}_{n-1,n}^{\mathrm{T}} \end{bmatrix} \boldsymbol{\alpha}_{n-1}, \tag{15}$$

where $\boldsymbol{\alpha}_{n-1} = [\alpha_1\ \alpha_2\ \cdots\ \alpha_{n-1}]^{\mathrm{T}}$ is a non-zero vector, then $\mathbf{K}_n$ has the reduced rank $n-1$. Rearrange (15) into

$$\mathbf{k}_{n-1,n} = \mathbf{K}_{n-1} \boldsymbol{\alpha}_{n-1}, \tag{16}$$

$$k_{n,n} = \mathbf{k}_{n-1,n}^{\mathrm{T}} \boldsymbol{\alpha}_{n-1}. \tag{17}$$

Since $\mathbf{K}_{n-1}$ has the full rank $n-1$ and therefore is invertible, from (16) we have $\boldsymbol{\alpha}_{n-1} = \mathbf{K}_{n-1}^{-1} \mathbf{k}_{n-1,n}$. Substituting this $\boldsymbol{\alpha}_{n-1}$ into (17) leads to $\delta_n = k_{n,n} - \mathbf{k}_{n-1,n}^{\mathrm{T}} \mathbf{K}_{n-1}^{-1} \mathbf{k}_{n-1,n} = 0$. □

To take into account the effect of noise in the data, $\mathbf{K}_n$ can be regarded to have approximately the same rank as that of $\mathbf{K}_{n-1}$, if $\delta_n \leqslant \varepsilon$, where $\varepsilon > 0$ is a small threshold value. Theorem 1 shows a way of selecting the feature sample subset $FS(d \times J)$, and the steps in extracting feature samples are now summarised as follows:

(1) Let the initial subset include a random sample (or alternatively the first sample) of the training set $X(KI \times J)$ and set $d = 1$. Then calculated the kernel matrix $\mathbf{K}_d$.
(2) Consider the samples of $X(KI \times J) \backslash FS(d \times J)$ one by one and calculate the corresponding test δ_{d+1}. If $\delta_{d+1} \leqslant \varepsilon$, the sample is abandoned; otherwise, set $d = d + 1$ and include the sample into $FS(d \times J)$ as well as modify the kernel matrix $\mathbf{K}_d$.
(3) After testing all the samples in $X(KI \times J)$, we obtain the feature sample subset $FS(d \times J)$ with d samples.

Appropriate value for the threshold ε depends on the noise level in the training data and the chosen kernel function. If a too large ε is used, the number of feature samples, d, will be too small and $\text{span}(\Phi_{FS})$ may not be an accurate approximation of the full feature space $\text{span}(\Phi_X)$. On the other hand, a too small ε will lead to an unnecessarily large number of feature samples or a nearly singular $\mathbf{K}_d$. We can use the following simple procedure to find an appropriate threshold value ε. Starting from a relatively large ε, gradually reduce ε to obtain the corresponding d values. This leads to the $\varepsilon - d$ curve. Look for the turning point of the $\varepsilon - d$ curve where the value of d changes from gradual increase to a sudden large increase or the corresponding kernel matrix $\mathbf{K}_d$ becomes nearly singular. The value of ε at this turning point is an appropriate threshold value with the corresponding d being the desired number of feature samples. This process of choosing appropriate threshold ε and subset sample size d is illustrated in Fig. 4.

The task of implementing the KICA method on the subset $FS(d \times J)$ is simplified significantly compared with the original one on the whole input sample set $X(KI \times J)$ and, moreover, a much smaller computer memory space is needed to store the kernel matrix. All these make it possible to efficiently execute our on-line monitoring technique for batch processes.

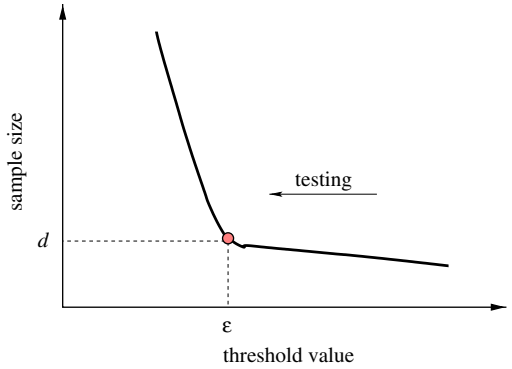

Fig. 4. Illustration of selecting appropriate threshold value and number of feature samples.

4.2. Batch process monitoring using FS-MKICA

The detailed FS-MKICA algorithm for batch process monitoring can readily be summarised. The steps for establishing the FS-MKICA model are as follows:

(1) Collect the batch data $\underline{X}(I \times J \times K)$ under normal operating condition. $\underline{X}(I \times J \times K)$ is unfolded into the two-way matrix $X(I \times KJ)$, which is normalised and then rearranged as the two-way data matrix $X(KI \times J)$.
(2) Use the procedure described in Section 4.1 to choose the subset $FS(d \times J)$ of d feature samples from the preprocessed data $X(KI \times J)$.
(3) Perform the KICA algorithm based on the feature sample set $FS(d \times J)$.

   (a) Sphere the feature sample data $FS(d \times J)$ using the KPCA. Calculate the eigenvalues and corresponding orthonormal eigenvectors of the Gram matrix $\mathbf{K}_d$ to obtain the whitened data $\mathbf{y}$ according to (12).
   Here further significant saving in computational complexity can be achieved by using only the k_L largest eigenvalues and corresponding orthonormal eigenvectors of $\mathbf{K}_d$ to perform the data whitening. A simple empirical criterion is that the chosen k_L largest eigenvalues should account for over 70–80% of the total energy (eigenvalues).
   (b) Extract the latent ingredients by utilising the ICA algorithm. We need to acquire the de-mixing matrix $\mathbf{W}$ and the mixing matrix $\mathbf{A}$ as well as to extract the independent components $\hat{\mathbf{s}}$ according to (13). Then choose the m most dominant independent components, denoted as $\hat{\mathbf{s}}_m$, and denote the corresponding de-mixing and mixing matrices as $\mathbf{W}_m$ and $\mathbf{A}_m$, respectively.
(4) Two types of statistics, I^2 and SPE statistics [1], are used to plot the process monitoring charts. The I^2 statistic for the systematic part of process variation is the sum of the squared independent components $\hat{\mathbf{s}}_m$ while the SPE statistic for the residual part is the sum of the square errors $\mathbf{e}$, where $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$ with $\hat{\mathbf{y}} = \mathbf{A}_m \cdot \mathbf{W}_m \cdot \mathbf{y}$.

More specifically, the I^2 and SPE statistics are defined as

$$I^2(i) = \hat{\mathbf{s}}_m(i)^{\mathrm{T}} \hat{\mathbf{s}}_m(i), \tag{18}$$

$$\mathrm{SPE}(i) = \mathbf{e}(i)^{\mathrm{T}} \mathbf{e}(i), \tag{19}$$

with $\hat{\mathbf{s}}_m(i)$ and $\mathbf{e}(i)$ denoting the i-th column vectors of $\hat{\mathbf{s}}_m$ and $\mathbf{e}$, respectively. The I^2 statistical control chart is used to detect special events entering the system, whereas the SPE statistic is a measure of non-systematic parts caused by random noise variation, and its chart is also a fault detection tool in process monitoring. The confidence limits of the above two statistics can be established by the kernel density estimation method [18].

Similarly, on-line monitoring of a batch process can be implemented in the following steps.

(1) Scale the new batch data in time k, $\mathbf{x}_{\mathrm{new}}(k)$, and whiten it into $\mathbf{y}_{\mathrm{new}}(k)$ according to (12).
(2) Calculate the new component vector $\hat{\mathbf{s}}_{\mathrm{new}}(k) = \mathbf{W}_m \cdot \mathbf{y}_{\mathrm{new}}(k)$ and the residual error vector $\mathbf{e}_{\mathrm{new}}(k) = \mathbf{y}_{\mathrm{new}}(k) - \mathbf{A}_m \cdot \mathbf{W}_m \cdot \mathbf{y}_{\mathrm{new}}(k)$. Then obtain the $I^2_{\mathrm{new}}(k)$ and $\mathrm{SPE}_{\mathrm{new}}(k)$ according to (18) and (19), respectively.
(3) The two statistics of the new batch data are compared with the corresponding confidence limits. If the upper control limits are exceeded, the abnormal behaviour of the process is detected.

The well-known kernel density estimation method estimates the probability density function (PDF) of the data sample set $\{x_i\}_{i=1}^{L}$ using [20,21]

$$p(x) = \frac{1}{\sigma L} \sum_{i=1}^{L} \varphi\left(\frac{x - x_i}{\sigma}\right), \tag{20}$$

where σ is the kernel width and $\varphi(\cdot)$ denotes the kernel function. Typically the Gaussian-kernel function

$$\varphi(u) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-u^2/2} \tag{21}$$

is used. The value $\tilde{x}$, which is defined by

$$\int_{-\infty}^{\tilde{x}} p(u)\,\mathrm{d}u = 0.99 \tag{22}$$

provides the 99% confidence limit.

5. A case study

The proposed approach was used to detect faults in the fed-batch penicillin cultivation process. The production of penicillin has been the subject of many studies because of its commercial and industrial importance [2].

5.1. Process description

The fed-batch penicillin cultivation process has the nature of nonlinear dynamics and multistage growth. In penicillin cultivation, most of the necessary cell mass is usually generated during the initial pre-culture stage and penicillin cells begin to be produced at the cell exponential growth phase. Cells continue to grow to be penicillin in the stationary phase. During this stage biomass growth rate must be kept constant to maintain high penicillin production. So a readily metabolisable sugar such as glucose is supplied continuously into the system instead of being added all at once at the beginning [4,3]. We obtain the batch data using a modular simulator (PenSim v2.0) for fed-batch fermentation developed by the monitoring and control group of the Illinois Institute of Technology in the year of 2002 [10]. The flowchart of the penicillin fermentation process is illustrated in Fig. 5. Environmental variables such as pH and temperature play an important role in the quality and quantity of the final product. In order to maintain the circumstances required by penicillin cultivation, acid or base additions are allowed to control pH at a certain value, while cooling or heating water are used to make the cultivation temperature stay constant. In addition, sugar concen-

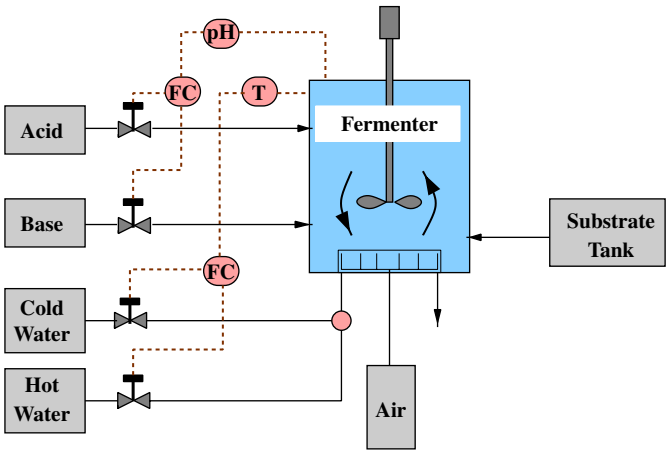

Fig. 5. Flowchart of a penicillin fermentation.

tration is controlled by feeding glucose during the penicillin production stage. In the simulator Pensim v2.0, simulations are run under a closed-loop control of pH and temperature while glucose addition is performed open-loop.

In this work, 11 variables listed in Table 1 are used to monitor the penicillin cultivation process. Table 2 shows the simulation initial conditions and set points of the batch fermentation process in Pensim v2.0. In order to achieve the on-line monitoring and fault detection of batch processes, a monitoring model under normal operation conditions needs to be established first. In this paper, 30 batch runs were generated from the simulator to model the process behaviour under normal operating conditions. Five additional batch runs were generated to test whether the FS-MKICA model can effectively monitor the batch process, in which the first one was the normal batch 31 and the other four batches were faulty batches as listed in Table 3.

5.2. Results and discussion

All the batch runs were the same duration of 400 h, consisting of a pre-culture stage of about 45 h and a fed-batch stage of about 355 h. The sampling interval was 0.5 h. Therefore the three-way matrix collected for modelling was denoted by $\underline{X}(30 \times 11 \times 800)$. According to the unfolding method discussed in Section 4, this three-way data matrix was unfolded and scaled into the two-way matrix $X(24\,000 \times 11)$. Then the subset of feature samples were chosen from these 24 000 data samples, and the KICA method was applied to the feature sample subset to obtain the FS-MKICA model. The kernel function was chosen to be the second-order polynomial function

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^{\mathrm{T}}\mathbf{y} + 1)^2. \tag{23}$$

The procedure of determining an appropriate number of feature samples is illustrated in Fig. 6. It can be seen that when the value of ε was reduced from 0.1 to 2×10^{-7}, the number of selected feature samples was a constant $d = 78$ while for $\varepsilon \leqslant 10^{-7}$, the kernel matrix became badly conditioned. Thus, the appropriate number of feature samples was $d = 78$. From the kernel matrix constructed based on the feature-sample set $FS(78 \times 11)$, $m = 12$ independent components were selected from the $d = 78$ non-linearly mapped variables in the feature space $\mathscr{H}$ by only retaining the components whose negentropy values were greater than the threshold 0.02, and the corresponding de-mixing and mixing matrices, $\mathbf{W}_m$ and $\mathbf{A}_m$, were established to complete the FS-MKICA model. The I^2 and SPE statistics were calculated using (18) and (19), respectively. The 99% confidence limits for these two statistics were obtained based on the kernel density estimation method [18].

We compared our FS-MKICA approach employing the second-order polynomial kernel function with the MICA method [26], which is an important on-line monitoring model for batch processes. In a similar way to the construction of the FS-MKICA model, we constructed the MICA model. The three-way data matrix $\underline{X}(30 \times 11 \times 800)$ was first unfolded into the two-way data matrix $X(24\,000 \times 11)$. The ICA algorithm was then implemented in the unfolded data space, which selected $m = 4$ independent components from the $J = 11$ variables to establish the MICA model. We also supplied the I^2 and SPE statistics, calculated according to (18) and (19), for the MICA monitoring method. Using

Table 1
List of variables used for the monitoring of penicillin cultivation process

Number	Variables
1	Aeration rate (l/h)
2	Agitator power (W)
3	Substrate feed flow rate (l/h)
4	Substrate feed temperature (K)
5	Dissolved oxygen concentration (% saturation)
6	Culture volume (l)
7	Carbon dioxide concentration (m mol/l)
8	pH
9	Bioreactor temperature (K)
10	Generated heat (kcal)
11	Cooling flow rate (l/h)

Table 2
Initial conditions and set points used in the simulation of the penicillin fermentation process

Initial conditions	
Substrate concentration (g/l)	14–18
Dissolved oxygen concentration (% saturation)	1–1.2
Biomass concentration (g/l)	0
Penicillin concentration (g/l)	0
Culture volume (l)	100–104
Carbon dioxide concentration (m mol/l)	0.5–1
Bioreactor temperature (K)	295–301
Generated heat (kcal)	0
pH	4.5–5.5
Set points	
Aeration rate (l/h)	8–9
Agitator power (W)	29–31
Substrate feed flow rate (l/h)	0.039–0.045
Substrate feed temperature (K)	295–296
Bioreactor temperature (K)	297–298
pH	4.95–5.05

Table 3
Fault types

No.	Fault description
Fault 1	Substrate feed flow rate is suddenly step-decreased by 10% at 70 h and maintained to 150 h of batch operation.
Fault 2	Aeration rate is linearly increased with the ramp rate of $1\,l/h^2$ from 100 h to the end of batch operation (400 h).
Fault 3	Agitator rate is linearly increased with the ramp rate of $3\,W/h^2$ from 100 h to the end of batch operation (400 h).
Fault 4	Substrate feed flow rate is linearly increased with the ramp rate of $0.005\,1/h^2$ from 100 h to the end of batch operation (400 h).

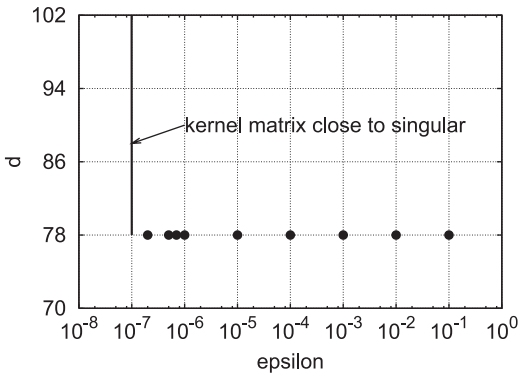

Fig. 6. Number of feature samples d versus threshold value ε for the second-order polynomial kernel function.

the kernel density estimation method again, we computed the 99% confidence limits for the two statistics.

Fig. 7(a) and (b) show the on-line monitoring results using the I^2 and SPE charts with the 99% confidence limits for the MICA and FS-MKICA methods, respectively, for the normal batch 31. For both the MICA and FS-MKICA models, the I^2 and SPE monitoring charts for the normal batch 31 stayed below the confidence limits for all the times except for a few instances during the preculture stage, which were allowed because the 1% false fault alarms might result from the 99% control limits. The two on-line monitoring methods

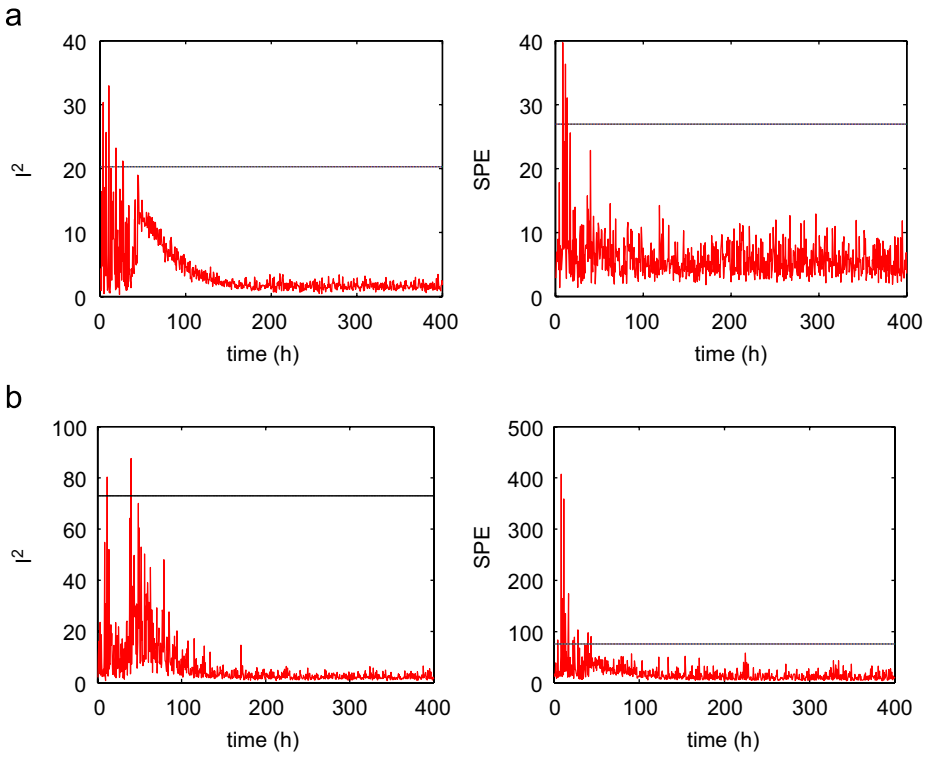

Fig. 7. On-line monitoring charts for (a) MICA and (b) FS-MKICA with polynomial function in the case of normal batch 31.

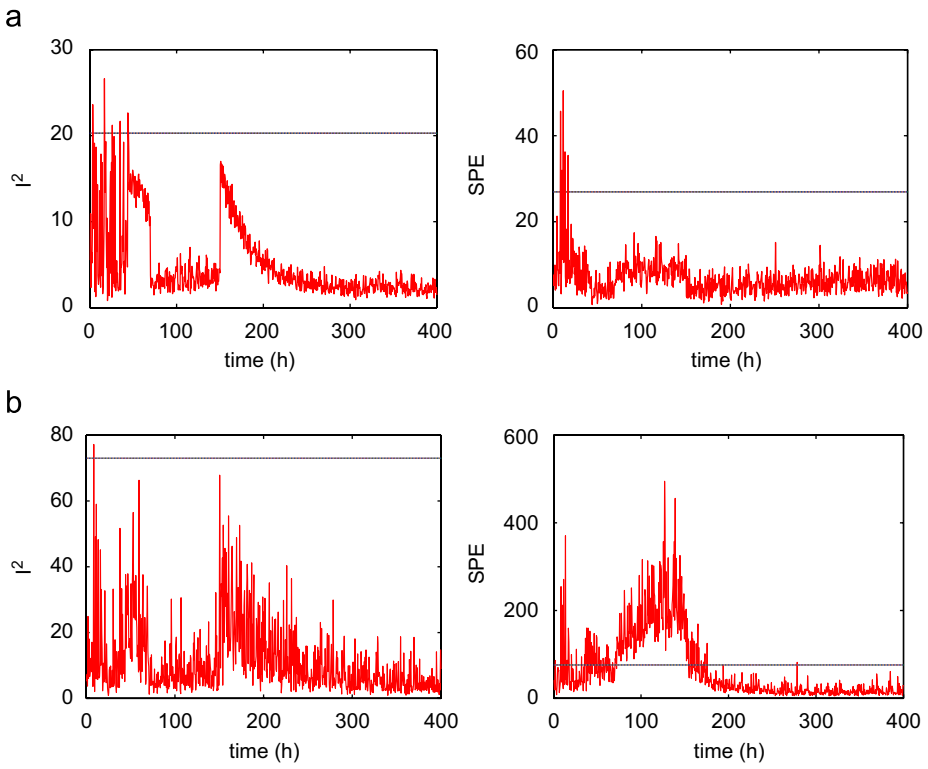

Fig. 8. On-line monitoring charts for (a) MICA and (b) FS-MKICA with polynomial function in the case of first-fault batch with a step decrease in substrate feed flow rate during the time period of 70–150 h.

confirmed that the batch was operating normally throughout the batch run.

Fig. 8 plots the on-line monitoring charts of the two models for the abnormal batch of fault 1 in which substrate (glucose) feed flow rate was suddenly decreased by 10% at the time of 70 h and lasted to 150 h. Glucose is the main carbon source of the fed-batch fermentation and thus a decrease in the glucose feed rate will lead to a reduction in penicillin production. From Fig. 8(a), neither the I^2 chart nor the SPE chart of the MICA model could detect the small step-decrease disturbance in the glucose feed rate. By contrast, the SPE chart of the FS-MKICA method definitely detected this fault during the time period of 70–150 h, as can be clearly seen from Fig. 8(b). This demonstrates that the proposed FS-MKICA approach has superior fault detection capability over the MICA method.

We next considered the second-fault batch, in which a ramp increase of aeration rate was imposed on the batch with the ramp rate of 1 $1/h^2$ from 100 h to the end of batch operation (400 h). The monitoring results using the MICA and FS-MKICA models for this abnormal process are shown in Fig. 9(a) and (b), respectively. The I^2 and SPE charts of both the models detected this faulty batch operation, as can be seen from Fig. 9. However, the I^2 chart of the MICA model exceeded its 99% control limit from 160 h approximately, whereas the value of the SPE chart for the FS-MKICA model exceeded its 99% control limit from around 150 h onward which was about 10 h earlier than the MICA method.

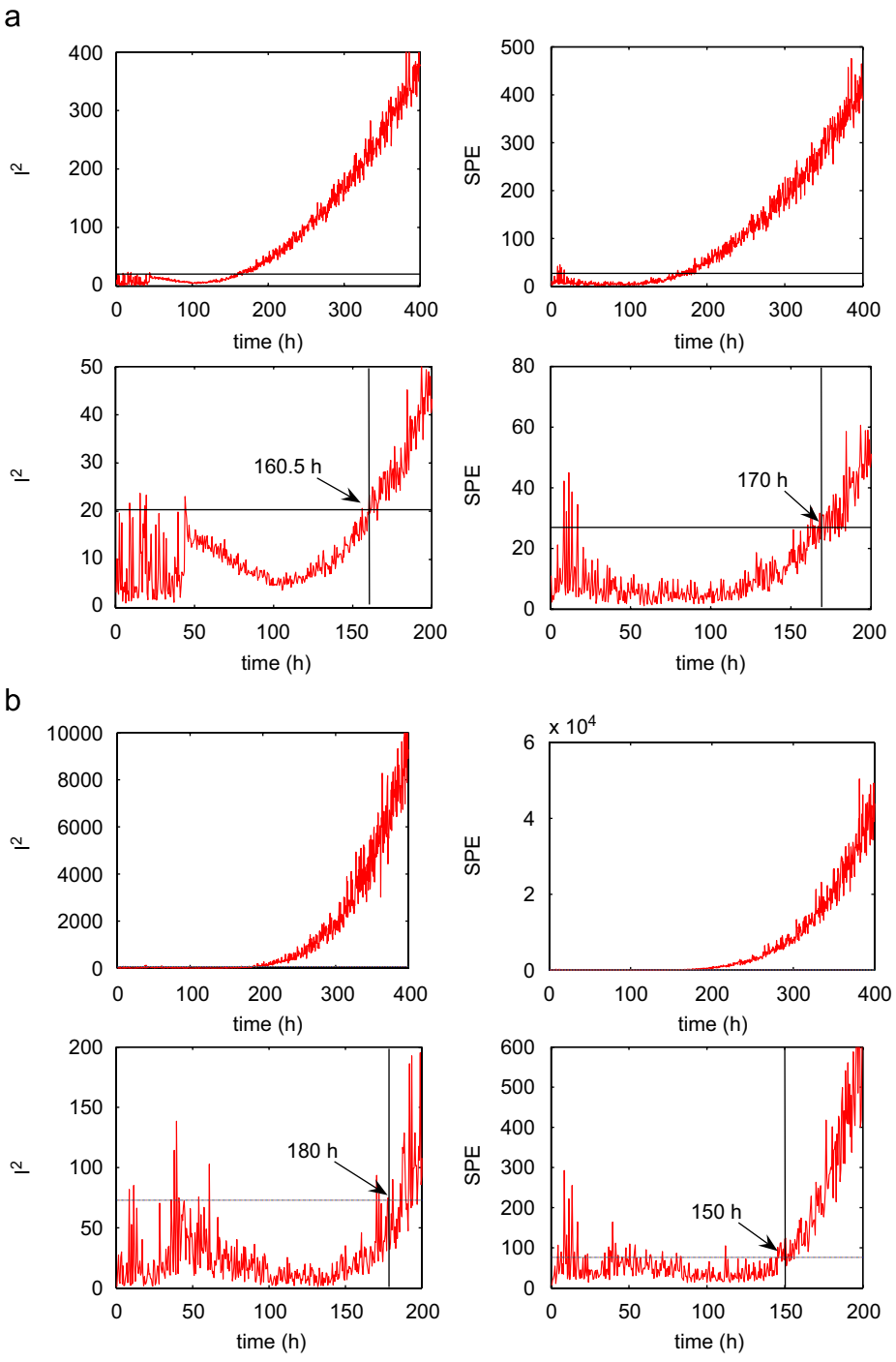

Fig. 9. On-line monitoring charts for (a) MICA and (b) FS-MKICA with polynomial function in the case of second-fault batch.

We then introduced the third-fault batch of Table 3. Fig. 10(a) and (b) show the monitoring results of the MICA and FS-MKICA methods, respectively. Again, the I^2 and SPE charts of both the methods detected this faulty batch operation. However, from Fig. 10, we know that the MICA model detected this disturbance from 180 h onward, while the FS-MKICA model detected the faulty batch operation from 165 h onward, which was 15 h earlier than the MICA model.

The last fault of Table 3 was also imposed on the batch so that the glucose feed rate was linearly increased at the ramp rate of

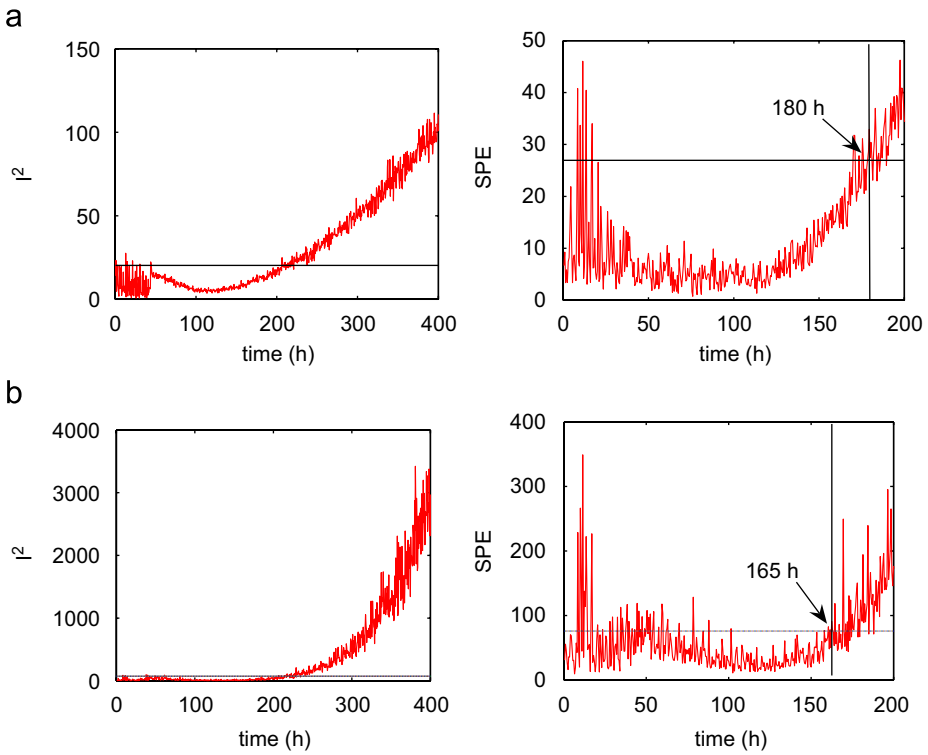

Fig. 10. On-line monitoring charts for (a) MICA and (b) FS-MKICA with polynomial function in the case of third-fault batch.

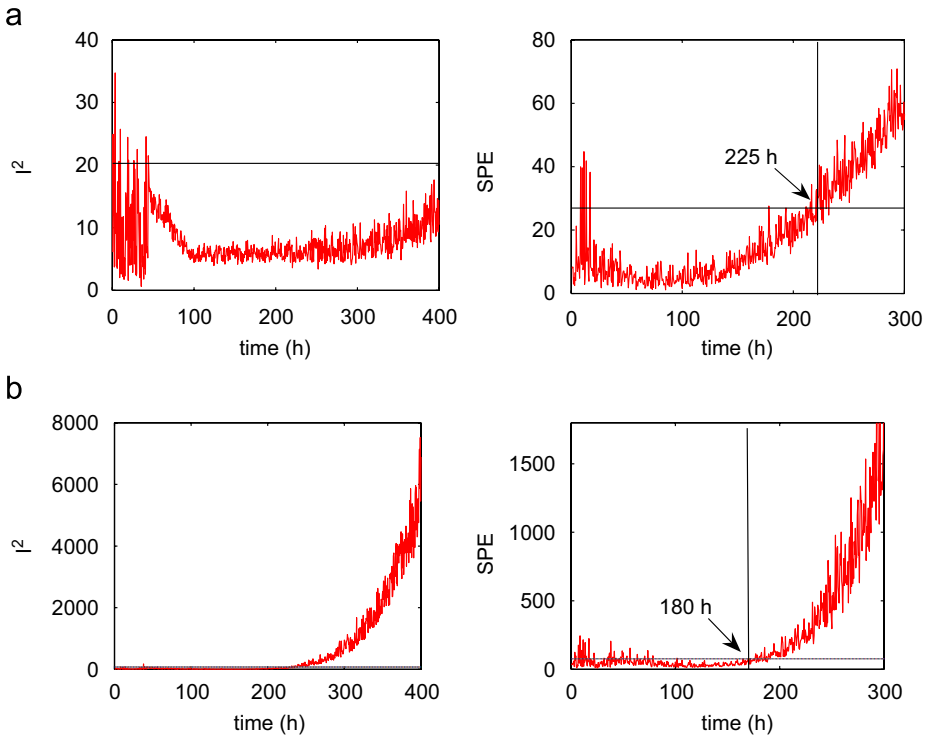

Fig. 11. On-line monitoring charts for (a) MICA and (b) FS-MKICA with polynomial function in the case of fourth-fault batch.

$0.005\,1/h^2$ from 100 h to the end of batch operation (400 h). The on-line monitoring results using the MICA model were depicted in Fig. 11(a), where it can be seen that the I^2 chart of the MICA model could not detect this type of fault with the ramp change in the substrate feed flow rate. The SPE chart of the MICA model did detect this faulty batch operation at 225 h. As shown in Fig. 11(b), both the I^2 and SPE charts of the FS-MKICA model correctly detected this faulty batch process and, moreover, the SPE chart of the FS-MKICA model detected the fault at 180 h, far earlier than the SPE chart of the MICA model did. This clearly demonstrates that the proposed FS-MKICA approach can detect a fault or disturbance more rapidly during the fermentation process.

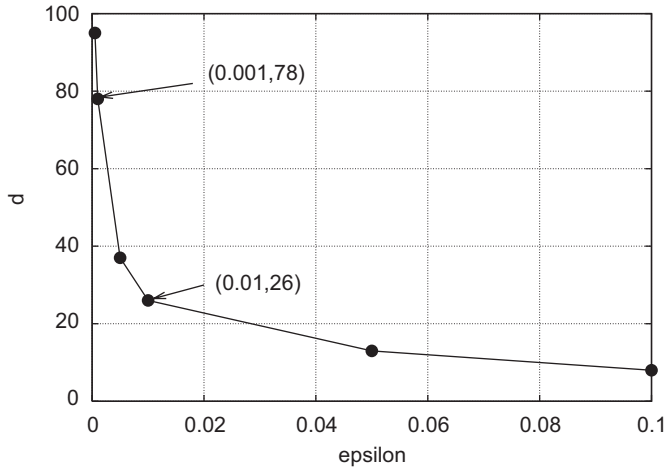

Fig. 12. Number of feature samples d versus threshold value ε for the Gaussian-kernel function.

5.3. Further results using Gaussian kernel

It can be seen that the FS-MKICA model with the second-order polynomial kernel function performed well. The polynomial function (23) is a commonly used kernel function and it is simple and easy to implement. Another commonly used kernel function is the Gaussian function defined by

$$k(\mathbf{x},\mathbf{y};\rho) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/2\rho^2}, \tag{24}$$

where ρ is the kernel width. For a comparison purpose, we also used the Gaussian-kernel function (24) to form the FS-MKICA model. Appropriate value for the Gaussian-kernel width was found empirically to be $\rho = 15$. To determine an adequate number of feature samples, the $\varepsilon - d$ curve is plotted in Fig. 12, where it can be seen that an appropriate number of selected feature samples was $d = 26$. From the Gaussian-kernel matrix constructed based on the feature-sample set $FS(26 \times 11)$, $m = 10$ independent components were retained as their negentropy values were greater than the threshold 0.02. The corresponding de-mixing and mixing matrices, $\mathbf{W}_m$ and $\mathbf{A}_m$, were established to complete the Gaussian-kernel based FS-MKICA model. The I^2 and SPE statistics were then calculated using (18) and (19), respectively. The 99% confidence limits for these two statistics were obtained again based on the kernel density estimation method [18].

The on-line monitoring performance of this constructed FS-MKICA model with Gaussian kernel and $d = 26$ are depicted in Figs. 13–16. Compared with the performance of the FS-MKICA model with polynomial kernel and $d = 78$ shown in Figs. 8(b)–11(b), it can be seen that the performance of the Gaussian-kernel based FS-MKICA model was poorer. Specifically, it did not clearly detect the fault in the first-fault batch, and it took longer time to detect the other three faulty batch processes. We concluded that the second-order polynomial kernel based

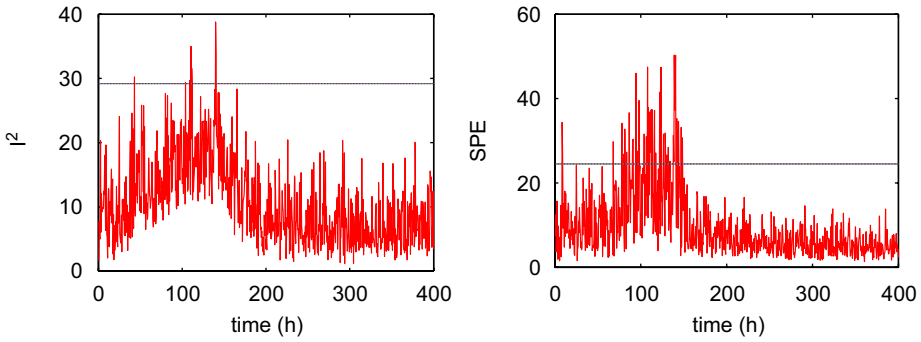

Fig. 13. On-line monitoring charts for FS-MKICA with Gaussian function and $d = 26$ in the case of first-fault batch with a step decrease in substrate feed flow rate during the time period of 70–150 h.

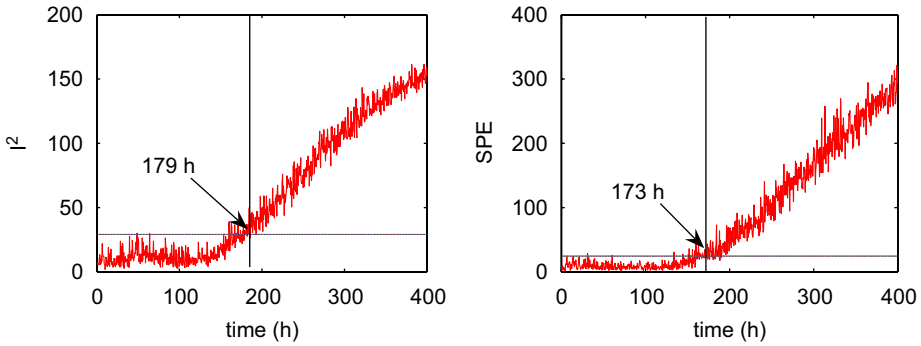

Fig. 14. On-line monitoring charts for FS-MKICA with Gaussian function and $d = 26$ in the case of second-fault batch.

FS-MKICA model had superior fault detection capability over the Gaussian-kernel based FS-MKICA model for this benchmark process.

The Gaussian-kernel based FS-MKICA model constructed from the $d = 26$ feature samples was actually optimal for Gaussian-kernel function. To demonstrate this, we further investigated using the $d = 78$ feature samples, corresponding to the $\varepsilon - d$ curve of Fig. 12 at the point ($\varepsilon = 0.001$, $d = 78$), to construct the FS-MKICA model and we also retained the first $m = 12$ dominant independent components just as in the case of polynomial kernel. The on-line monitoring charts of this constructed FS-MKICA model are shown in Figs. 17–20, respectively, for the four faulty batch processes. It is clear that the fault detection capability of this Gaussian-kernel based FS-MKICA model was inferior to that of the Gaussian-kernel based FS-MKICA model constructed from only $d = 26$ feature samples.

6. Conclusions

In this work, a new approach has been proposed for on-line monitoring of batch processes which exhibit significant nonlinear characteristics. Our proposed FS-MKICA method can effectively handle very large datasets. The key idea of the proposed approach is to choose a small subset of feature samples from the large unfolded two-way training batch data efficiently, which ensures that the constructed low-dimensional kernel space approximates the full nonlinear feature space well. In the constructed low-dimensional nonlinear feature space, the KICA method can readily be applied, without suffering the difficulty of prohibitive computation, to establish the FS-MKICA monitoring model. The I^2 and SPE statistical charts have been supplied to monitor new batch runs on-line. The proposed FS-MKICA model has been investigated in the simulation

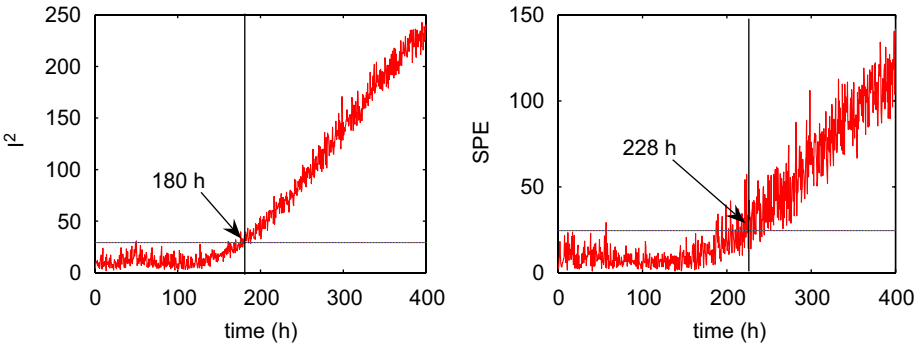

Fig. 15. On-line monitoring charts for FS-MKICA with Gaussian function and $d = 26$ in the case of third-fault batch.

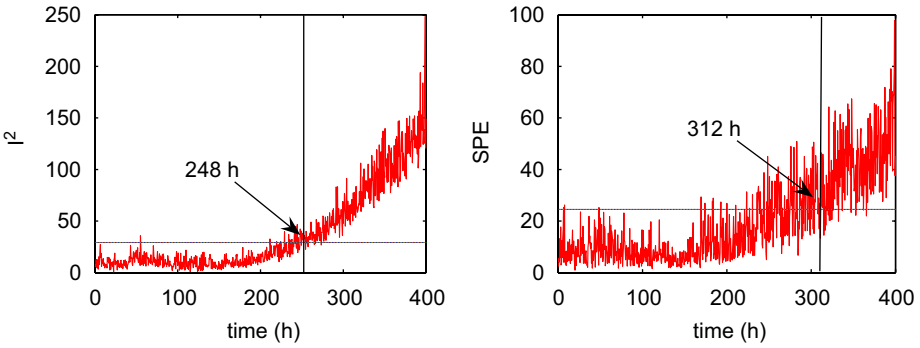

Fig. 16. On-line monitoring charts for FS-MKICA with Gaussian function and $d = 26$ in the case of fourth-fault batch.

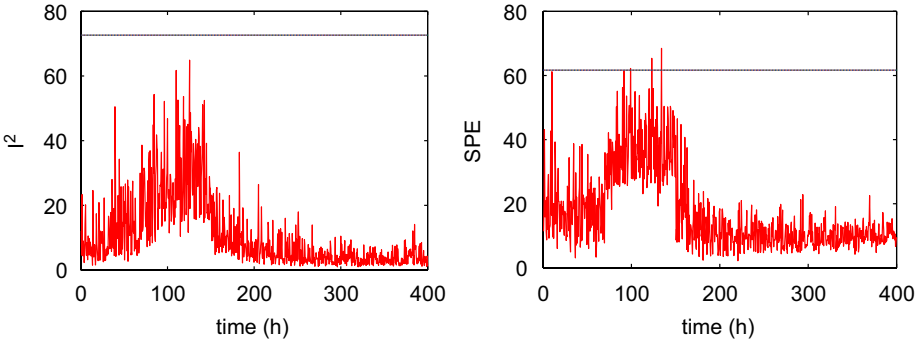

Fig. 17. On-line monitoring charts for FS-MKICA with Gaussian function and $d = 78$ in the case of first-fault batch with a step decrease in substrate feed flow rate during the time period of 70–150 h.

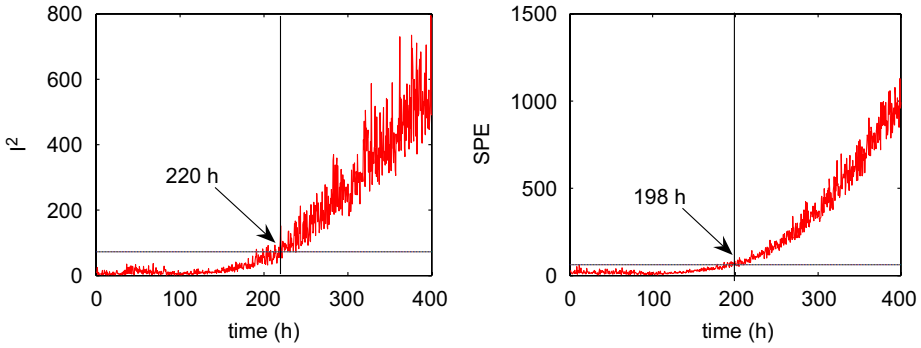

Fig. 18. On-line monitoring charts for FS-MKICA with Gaussian function and $d = 78$ in the case of second-fault batch.

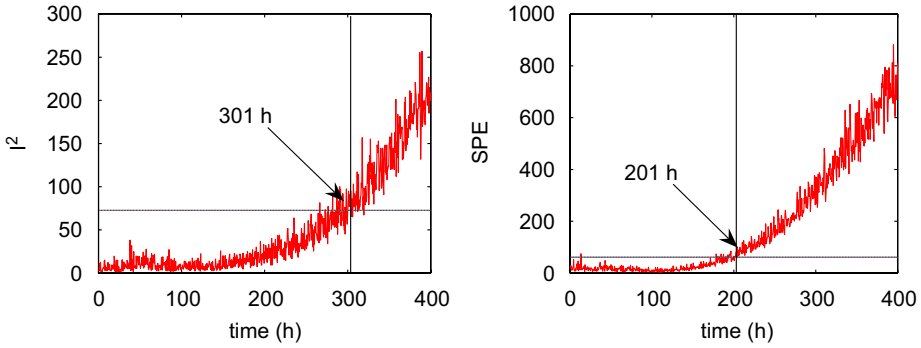

Fig. 19. On-line monitoring charts for FS-MKICA with Gaussian function and $d = 78$ in the case of third-fault batch.

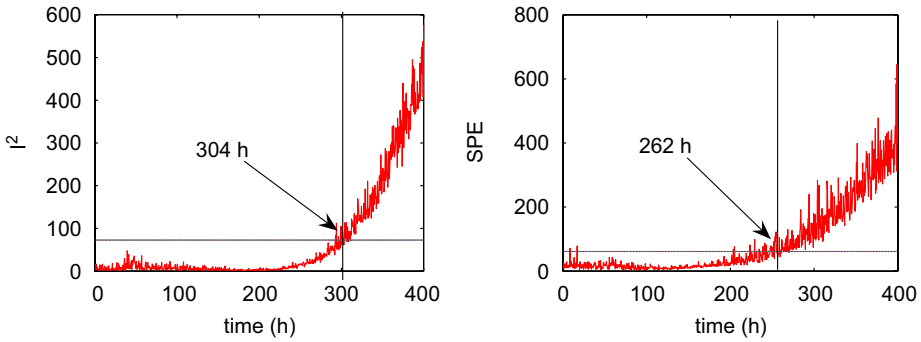

Fig. 20. On-line monitoring charts for FS-MKICA with Gaussian function and $d = 78$ in the case of fourth-fault batch.

study of a fed-batch penicillin fermentation process with the standard linear MICA model as the benchmark. The simulation results obtained have clearly demonstrated that the FS-MKICA model can detect faults or disturbance more accurately and rapidly, especially for batch processes having nonlinear characteristics.

References

[1] H. Albazzaza, X.Z. Wang, Multivariate statistical batch process monitoring using dynamic independent component analysis, in: W. Marquardt, C. Pantelides (Eds.), Computer Aided Chemical Engineering 21A: 16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering, Amsterdam, Elsevier, 2006, pp. 1341–1346.
[2] B. Atkinson, F. Mavituna, Biochemical Engineering and Biotechnology Handbook, second ed., Stockton Press, New York, 1991.
[3] G. Birol, C. Ündey, A. Çinar, A modular simulation package for fed-batch fermentation: penicillin production, Comput. Chem. Eng. 26 (11) (2002) 1553–1565.
[4] G. Birol, C. Ündey, S.J. Parulekar, A. Çinar, A morphologically structured model for penicillin production, Biotechnol. Bioeng. 77 (5) (2002) 538–552.
[5] J.F. Cardoso, A. Soloumiac, Blind beamforming for non-Gaussian signals, IEE Proc. F 140 (6) (1993) 362–370.
[6] G. Chen, T.J. McAvoy, Predictive on-line monitoring of continuous processes, J. Process Control 8 (5–6) (1998) 409–420.
[7] D. Dong, T.J. McAvoy, Batch tracking via nonlinear principal component analysis, AIChE J. 42 (8) (1996) 2199–2208.
[8] G.H. Golub, C.F.V. Loan, Matrix Computations, third ed., The John Hopkins University Press, Baltimore, MD, 1996.
[9] S. Harmeling, A. Ziehe, M. Kawanabe, K.R. Müller, Kernel feature spaces and nonlinear blind source separation, in: T.G. Kietterich, S. Brecker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems, vol. 14, MIT Press, Cambridge, MA, 2002, pp. 761–768.
[10] ⟨http://www.chee.iit.edu/~cinar⟩.
[11] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis, Wiley, New York, 2001.
[12] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications, Neural Networks 13 (4–5) (2000) 411–430.
[13] M. Kano, S. Tanaka, S. Hasebe, I. Hashimoto, H. Ohno, Monitoring independent components for fault detection, AIChE J. 49 (4) (2003) 969–976.
[14] J.-M. Lee, C.K. Yoo, I.-B. Lee, On-line batch process monitoring using a consecutively updated multiway principal component analysis model, Comput. Chem. Eng. 27 (12) (2003) 1903–1912.

[15] J.-M. Lee, C.-K. Yoo, I.-B. Lee, Fault detection of batch processes using multiway kernel principal component analysis, Comput. Chem. Eng. 28 (9) (2004) 1837–1847.
[16] J.-M. Lee, C.-K. Yoo, I.-B. Lee, Statistical process monitoring with independent component analysis, J. Process Control 14 (5) (2004) 467–485.
[17] J.-M. Lee, C.-K. Yoo, I.-B. Lee, Statistical monitoring of dynamic processes based on dynamic independent component analysis, Chem. Eng. Sci. 59 (14) (2004) 2995–3006.
[18] E.B. Martin, A.J. Morris, Non-parametric confidence bounds for process performance monitoring charts, J. Process Control 6 (6) (1996) 349–358.
[19] P. Nomikos, J.F. MacGregor, Monitoring batch processes using multiway principal component analysis, AIChE J. 40 (8) (1994) 1361–1375.
[20] E. Parzen, On estimation of a probability density function and mode, Ann. Math. Stat. 33 (1962) 1066–1076.
[21] B.W. Silverman, Density Estimation, Chapman & Hall, London, 1996.
[22] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemometrics Intell. Lab. Syst. 2 (1–3) (1987) 37–52.
[23] S. Wold, P. Geladi, K. Esbensen, J. Öhman, Multi-way principal components and PLS analysis, J. Chemometrics 1 (1) (1987) 41–56.
[24] J. Yang, X.-M. Gao, D. Zhang, J.-Y. Yang, Kernel ICA: an alternative formulation and its application to face recognition, Pattern Recognition 38 (10) (2005) 1784–1787.
[25] C.-K. Yoo, D.-S. Lee, P.A. Vanrolleghem, Application of multiway ICA for on-line process monitoring of a sequencing batch reactor, Water Res. 38 (7) (2004) 1715–1732.
[26] C.-K. Yoo, J.-M. Lee, P.A. Vanrolleghem, I.-B. Lee, On-line monitoring of batch processes using multiway independent component analysis, Chemometrics Intell. Lab. Syst. 71 (2) (2004) 151–163.

Xuemin Tian received his Bachelor of Engineering from Huadong Petroleum Institute, Dongying, China, in January 1982, and his M.S. degree from Beijing University of Petroleum, Beijing, China, in June 1994. From September 2001 to June 2002, he served as visiting professor at Central of Process Control, University of California in Santa Barbara.

He is a professor of Process Control at China University of Petroleum (Hua Dong). Professor Tian's research interests are in modelling, advanced process control and optimisation for petrol-chemical processes as well as fault detection and diagnosis, and process monitoring.

Xiaoling Zhang was born in Shandong, China, on March 29, 1982. She received her B.Sc. degree in 2005 and her M.S. degree in July 2008, both from China University of Petroleum.

She is taking a lecturer post at ChengDe Petroleum College, China. Her current research interests focus on fault detection and diagnosis in batch processes.

Xiaogang Deng was born in Shandong, China, on August 5, 1981. He obtained his Bachelor degree in 2002 and his Ph.D. degree in chemical engineering and technology in 2008, both from China University of Petroleum.

He is currently working as a research assistant in College of Information and Control Engineering, China University of Petroleum. His research interests include process modelling and simulation, fault diagnosis of industrial processes, and controller performance monitoring.

Sheng Chen received his Bachelor of Engineering from Huadong Petroleum Institute, Dongying, China, in January 1982, and his Ph.D. degree in control engineering from the City University, London, UK, in 1986. He was awarded the Doctor of Sciences (D.Sc.) degree by the University of Southampton, Southampton, UK, in 2005.

He joined the School of Electronics and Computer Science, the University of Southampton, Southampton, in September 1999. He previously held research and academic appointments at the University of Sheffield, Sheffield, the University of Edinburgh, Edinburgh, and the University of Portsmouth, Portsmouth, all in UK. Professor Chen's research interests include wireless communications, machine learning and neural networks, finite-precision digital controller design, and evolutionary computation methods. He has published over 350 research papers.

In the database of the world's most highly cited researchers, compiled by Institute for Scientific Information (ISI) of the USA, Dr. Chen is on the list of the highly cited researchers in the engineering category.